

ABSTRACT

Title of Document: A SPATIALLY-EXPLICIT FRAMEWORK
FOR INVESTIGATING PATCHINESS EFFECTS
IN AQUATIC ECOSYSTEMS

David Louis Scheurer, Doctor of Philosophy, 2006

Directed By: Professor Robert H. Gardner, Marine-Estuarine-
Environmental-Science, University of Maryland
Center for Environmental Science

Aquatic ecosystems display complex spatially-varying patterns of growth and decay. These patterns are produced by the interaction of numerous physical and biological processes that result in characteristic scales of patchiness with important ecological consequences. Although these interactions and processes have been studied extensively, it is still unclear under what conditions and to what degree one process dominates the other and how the dynamics change across scales. This dissertation uses a spatial modeling approach to examine how processes and patterns translate across spatial and temporal scales and how the spatial distribution of resources in turn, influences these processes and patterns. This is accomplished through the development of a novel spatially-explicit simulation framework which utilizes 1) a nutrient-phytoplankton-zooplankton-detritus (NPZD) ecosystem model; 2) realistic physical exchanges between individual model cells; 3) spatially varying forcing functions and 4) robust pattern analysis techniques, to produce a consistent and reliable method for extrapolating detailed, fine-grained dynamics to broad-scale patterns within aquatic environments. Application of the framework required the development of two novel components, an

NPZD ecosystem model to simulate biological processes and a method to simulate turbulent mixing at fine and intermediate scales. Experiments testing the robustness of these components are presented along with results from simulations applying the framework to investigate species and ecosystem level response to spatial and temporal heterogeneity in nutrient forcing. Major results of the work and potential applications for investigating scale-dependent patterns in aquatic ecosystems are discussed.

A SPATIALLY-EXPLICIT FRAMEWORK FOR INVESTIGATING
PATCHINESS EFFECTS IN AQUATIC ECOSYSTEMS

By

David Louis Scheurer

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park, in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2006

Advisory Committee:

Professor Robert Gardner, Chair
Associate Professor Raleigh Hood
Professor Edward Houde
Associate Professor Patrick Kangas, Dean's Rep.
Professor Michael Kemp
Professor Larry Sanford

© Copyright by
David Louis Scheurer
2006

This work is dedicated to my wife

Anne

ACKNOWLEDGEMENTS

This work would not be possible without the support, patience, perseverance, and insight of my Advisor, Dr. Robert Gardner. I am also thankful for the assistance, commitment, and time of my Committee members, both past and present. I am appreciative of the flexibility and support of Drs. Rob Magnien and David Johnson of the NOAA Costal Ocean Program for helping me to balance my work and school activities. Thanks is also extended to the MEES Program for keeping me on track and in school and I would especially like to thank Debbie Morrin-Nordlund and Beth Zinecker for answering my many questions over the years and for their compassion. I am grateful for the stable support provided by the EPA STAR program as part of the Multiscale Experimental Ecosystem Center at the University of Maryland Center for Environmental Science. I would like to thank Dan Fiscus for helping to develop the SAS code and Eric Farris and Clayton Kingdon for trouble shooting many computer and software problems. I would also like to thank John Peterson, Chung-Chi Chen, and Debbie Hinkle for assistance with development and testing of the model and numerous helpful discussions. I would also like to thank all my friends, family and coworkers who have provided encouragement and support over the years. A debt of gratitude is due to my parents for raising me and instilling in me an ongoing desire to learn.

TABLE OF CONTENTS

Chapter 1: General Introduction.....	1
1.1 The Issue of Extrapolation.....	1
1.2 Physical and Biological Scales of Patchiness in Aquatic Systems.....	3
1.3 Patchiness Issues Associated with Experimental Ecosystems.....	7
1.4 Linking Experimental Ecosystems to Marine Systems.....	10
1.5 Dissertation Objectives.....	12
1.6 Components of Modeling System.....	13
1.7 Dissertation Structure.....	18
Chapter 2: Sensitivity analysis of an NPZD pelagic ecosystem model under changing chemical, biological, and physical conditions20	
Abstract.....	20
2.1 Introduction.....	21
2.2 Methods.....	23
2.2.1 General Model Description.....	23
2.2.2 Model Parameters.....	23
2.2.3 Model equations.....	28
2.2.4 Sensitivity Analysis.....	36
2.2.5 Experimental Treatments.....	37
2.3 Results.....	40
2.3.1 General sensitivity dynamics.....	40
2.3.2 Sensitivity response to variation in external drivers (Independent effects).....	44
2.3.3 Simultaneous variation of external drivers (Factorial experiments).....	47
2.3.4 Dominant model parameters.....	54
2.4 Discussion.....	59

2.4.1 Sensitivity Dynamics.....	60
2.4.2 Response to changing environmental conditions.....	62
2.4.3 State variable effects on sensitivity results.....	63
2.4.4 Model structure and form effects on sensitivity results.....	64
2.4.5 Comparison of sensitivity results between studies.....	65
2.5 Conclusions.....	67
Chapter 3: A novel grid-based method for simulating idealized turbulence in aquatic systems.....	72
Abstract.....	72
3.1 Introduction.....	73
3.2 Methods.....	76
3.2.1 Simulation Platform.....	76
3.2.2 Eulerian seeded eddy model (ESEM).....	79
3.2.3 Subgrid Diffusive Mixing.....	90
3.2.4 Experimental Treatments.....	92
3.3 Results.....	97
3.3.1 Breakdown of a passive tracer.....	102
3.3.2 Instantaneous velocity spectrum.....	104
3.3.3 Spreading rate of a patch.....	105
3.4 Discussion.....	111
3.4.1 Benefits of the ESEM method.....	112
3.4.2 Limitations, assumptions and potential improvements.....	118
3.4.3 Applications of the ESEM.....	120
3.5 Conclusion.....	123
Chapter 4: Response of an NPZD pelagic ecosystem model to spatially and temporally varying nutrient input.....	124
Abstract.....	124
4.1 Introduction.....	125
4.2 Methods.....	127

4.2.1 Simulation Platform.....	127
4.2.2 Multifractal Map Generation.....	129
4.2.3 Simulation Domain and Parameters.....	130
4.2.4 Experimental Design.....	134
4.2.5 Statistical Analysis.....	136
4.3 Results.....	139
4.3.1 Qualitative response to experimental treatments.....	139
4.3.2 Mean Effects.....	143
4.3.3 Pattern Effects.....	143
4.3.4 Trophic Effects/Physical and biological interactions.....	149
4.4 Discussion.....	151
4.4.1 Simulation Platform and Overall Model Dynamics.....	151
4.4.2 Question 1: Does patchiness in nutrient inputs affect ecosystem function?.....	153
4.4.3 Question 2: Does patchiness in nutrient inputs affect spatial distributions?.....	155
4.4.4 Question 3: Can biological dynamics overcome the influence of physical mixing?.....	159
4.4.5 Management Implications.....	161
4.5 Conclusion.....	162
Appendix A: Model state variable and parameter descriptions.....	164
Appendix B: Sensitivity Analysis.....	173
Appendix C: Chi-Square Analysis.....	177
Appendix D: Numerical Diffusion.....	181
Appendix E: Turbulence Intensity Calculation.....	184
Appendix F: Spectral Analysis.....	187
Appendix G: Computer Code.....	189
References.....	265

Chapter 1

General Introduction

1.1 The Issue of Extrapolation

The extrapolation of theoretical and experimental information across spatial and temporal scales is a significant challenge facing the scientific community. Extrapolation across scales directly impacts our ability to address broad-scale environmental problems that often require integration of information obtained at multiple scales (Root and Schneider 1995). During the last century, environmental problems such as eutrophication (Vitousek et al. 1997), habitat loss (Laurance et al. 1997), overharvesting of marine systems (Pauly et al. 1998) and greenhouse gas emissions (Root and Schneider 1995) have steadily grown to become broad-scale global issues. Traditional ecological research, however, has usually been conducted over fine spatial scales of short duration (Kareiva and Anderson 1988) leaving ecologists with the difficult task of extrapolating results to larger scales (Frost et al. 1988; Levin 1992). The extent to which information from these fine-scale studies (e.g., field plots, mesocosms, experimental ecosystems) can be used to address broad-scale issues remains a pressing problem (Schneider 1994; Carpenter et al. 1995).

The issue of scale has played an increasingly important role in the design and interpretation of ecological experiments (Stommel 1963; Allen 1977; Frost et al. 1988; Wiens 1989; Levin 1992). This increasing role is due, in part, to a growing awareness that the extrapolation of local conditions (i.e., mean values estimated from fine-scale measurements) do not adequately characterize the variance and nonlinear dynamics that

often define natural ecosystems (Allen and Hoekstra 1992; Giller et al. 1992; Peterson and Parker 1998). Spatial heterogeneity caused by the complex interaction of physical and biological processes over multiple scales is often cited as a major factor limiting extrapolation (Levin 1992; Holling 1992). The resulting “patchiness”, or spatial heterogeneity, is a significant source of scale-dependent effects, and also a source of error when results are extrapolated to broader scales (for reviews see Allen and Hoekstra 1992; Levin 1992; Holling 1992; Kemp et al. 2001). Failure to account for the effects of patchiness in modeling and empirical studies may severely limit the applicability of fine-scale studies to current environmental issues (Tilman and Kareiva 1997).

Unfortunately, employing a suite of experimental units (e.g., field plots or experimental ecosystems) of different sizes to study the scale-dependent nature of a particular ecological issue may exceed the budget of many projects. Instead, one is often forced to extrapolate beyond the confines of their particular experimental system. The implicit assumption here is that the experimental system of study is an analog of the larger system, very similar to what is done in engineering studies when a small-scale model is created for testing various effects which are then scaled up to the real system or object. But within an ecological context, a simple linear extrapolation is usually not possible (Schneider 1994). The relative importance of controlling biological and physical processes can change with scale, creating difficulties in extrapolation (Steele 1978; Harris 1980; Powell 1989). Often, the scale-dependent effects of these processes can neither be reproduced nor eliminated from studies with limited temporal and spatial domains (Tilman 1989; Peters 1991). Moving from the “simplified”, truncated, and usually homogenous assumptions, typical of many of these studies, to the spatially heterogeneous

natural environment can cause serious problems for extrapolation (Frost et al. 1988; Carpenter et al. 1995).

1.2 Physical and Biological Scales of Patchiness in Aquatic Systems

Physical and biological scales of patchiness, along with their interactions, have a long history in oceanographic and limnologic research (Platt 1972; Powell et al. 1975; Haury et al. 1978; Harris 1980; Okubo 1980; Mackas et al. 1985; Powell 1989; Steele 1991). Since the early 1960's when Hutchinson observed the "paradox of the plankton" (Hutchinson 1961) and Stommel created his famous space-time diagram of physical scales for the oceanic environment (Stommel 1963) patchiness (i.e., spatial heterogeneity) has been considered extremely important in aquatic systems. Patchiness, both physical and biological, exists at all scales (Steele 1978) and, in the highly dispersive aquatic environment, there is a general trade-off between physical (advective, dispersive) and biological (growth, death, and behavior) forces that results in characteristic scales of patchiness with important biological consequences. Surprisingly, it is still unclear under what conditions and to what degree one process dominates the other and how the dynamics change across scales (Giller et al. 1992; Powell and Steele 1995).

There are many variance-generating structures within the oceanic environment that create patchiness over a broad range of space (10^{-3} m to 10^4 km) and time (seconds to years) scales. At the broadest scales ($>10,000$ km), the atmosphere and ocean couple to generate basin-wide circulation (gyres) and current patterns that give rise to eddies and rings at the 10 to 1000 km scale. Other processes, such as internal waves and fronts,

occur in the 100 m to 10 km range and give way to turbulence and viscous dissipation at the finest scales (Kolmogorov 1941; Mackas et al. 1985). Even though there is a continuous transfer of variance from broad to fine scales, with the largest variability occurring at the longest time and broadest space scales, certain processes dominate others at particular scales (Saunders 1992). For example, at broad scales the oceanic environment resembles a two-dimensional “flat” system. Geostrophic motions (i.e., oceanic current systems) drive the dynamics in the horizontal direction while vertical processes play a minor role. At finer scales, the variability is more three-dimensional because turbulence becomes the main driving force rather than geostrophic motion (Denman 1992).

Recognition that marine organisms are not randomly distributed but are patchy in time and space dates back to the early part of the last century (Hardy and Gunther 1935). Since then, many investigators have searched for mechanisms that may produce biological patchiness within an environment dominated by physical processes (e.g., Steele 1978; Okubo 1980; Mackas et al. 1985). The early work of Skellam, Kierstead, and Slobodkin showed that biological patchiness in plankton could be maintained through reproductive means despite the highly dispersive physical environment found in aquatic systems (Skellam 1951; Kierstead and Slobodkin 1953). In the natural environment under optimal conditions (high nutrients, stable water column, convergence zones), phytoplankton patch maintenance through reproductive means can be achieved and sustained such as when an algal bloom occurs. But in general, growth-induced patches are ephemeral because nutrients become exhausted, predators respond to higher densities, or large mixing events break down and disperse the patch. Consequently, for

passively dispersed plankton their distribution or patchiness is largely governed by physical processes (Denman 1976; Denman and Platt 1976; Gower et al. 1980).

As organism size increases, generation time also increases reducing the ability of larger organisms to grow at a sufficient rate to create local patches of high density (Sheldon et al. 1972). However, larger organisms are usually more mobile (e.g., zooplankton, fish, etc.) and have the ability to create biological patchiness through directed movement in the form of aggregations, schools, swarms, and as a result of taxis behavior (Hamner et al. 1983; Okubo 1986; Norris and Schilt 1987). This mobility decouples biota from the physical environment that plays such a dominant role in structuring groups of strictly planktonic organisms, producing effects at smaller scales than predicted from physical effects alone (Weber et al. 1986; Levin et al. 1989; Schneider 1992; Horne and Schneider 1997).

When physical and biological processes are in synchrony there is a potential for the dynamics to become amplified, further enhancing patch generation (Denman and Powell 1984; Abbot 1993). For example, frontal zones are often areas of enhanced biological productivity. Organisms, especially planktonic species, can become trapped in convergence zones where nutrient concentrations may be higher than in the surrounding water column (Franks 1992). If the planktonic organisms are retained long enough, they can take advantage of the elevated nutrients and reduced physical dispersal to greatly increase biomass levels (Strass 1992). This effect is not only limited to frontal boundaries but can apply to any physical discontinuity from the scale of a thermocline up to mesoscale oceanic eddies (Gower et al. 1980; Venrick 1982; Oschlies and Garcon 1998).

Patchiness in the pelagic ocean is important for many reasons. Patches concentrate resources and are, therefore, critical for organisms living in areas where the average distribution of a resource is so dilute that growth and survival are reduced (LeBrasseur et al. 1969; Lasker 1975). Patchiness is also important because it provides protection from predation (Turner and Pitcher 1986; Inman and Krebs 1987). In a statistical sense, the chances of being consumed decrease as the school (i.e., patch) gets larger and saturation of predators due to handling time constraints and satiation becomes more probable. Patches can create advantages for reproduction (Brown 1975). Mate acquisition is facilitated in aggregations of individuals, and synchronized spawning adaptations can increase fertilization success. From a life-history standpoint, a large reproductive output released at one time will improve survival probabilities, thereby augmenting recruitment and future cohorts. In addition, the magnitude or degree of patchiness will influence processes such as the exchange of material and organisms across boundaries, predator-prey interactions, effects of disturbance, life-history traits, and genetic transfer between populations (Marquet et al. 1993).

Many organisms exploit the physical and biological patch-generating processes to increase their survival and reproductive success. The congregation of sea birds, pelagic fish, and marine mammals on krill patches are all examples of this type of behavior (Gaskin 1976; Sund et al. 1981; Schneider 1991). Vertical migration is used by some planktonic organisms to either maintain their position within a favorable habitat or to move to a new location. This is seen in estuaries when larvae of some organisms migrate into the surface water on the flood tide and then migrate to the bottom on the ebb tide (Staples 1979; Rothlisberg et al. 1983). Occurrence of breeding grounds in favorable

hydrographic environments provides free transport mechanisms that can aid in the survival of larval offspring (Sinclair 1987).

1.3 Patchiness Issues Associated with Experimental Ecosystems

Typical mesocosm experiments have limited temporal (days to months) and spatial (1 to 10's m) scales that create corresponding truncated physical and biological dynamics (Petersen et al. 1999). Given these constraints, the full suite of patch-generating mechanisms operating in the marine environment cannot be reproduced in experimental ecosystems, which can lead to scale-dependent results and a corresponding limit on extrapolation (Kemp et al. 2001). The next section discusses how processes affecting patch generation are either modified, distorted, or absent from experimental ecosystems.

Space: Marine mesocosms are small, typically isolated, and mixed differently from marine systems. Phenomena such as clumping, patterns of species coexistence, and stabilization of predator/prey cycles are examples of phenomena altered by enclosures (Tilman and Kareiva 1997). Isolation caused by mesocosms also produces patterns of species extinctions, either from competitive exclusion or unstable population cycles, that differ from natural systems and also alter ecosystem function.

The walls of mesocosms will affect the spatial dynamics and environment that an organism experiences. Some effects include the increased attenuation of light through the water column (Petersen et al. 1997), altered benthic/pelagic coupling due to enhanced exchanges with wall organisms (Chen et al. 1997), and artificial gradients (e.g., dead zones) resulting from uneven mixing (Sanford 1997). While such effects occur in the natural environment from interactions with the bottom and attached structures, within

mesocosms, the effect is much greater because of the higher surface area to volume ratio found within these enclosed systems. For pelagic systems, bottom or wall effects are usually non-existent.

In the natural environment, the exchange of material and individuals (fluxes) is a key process (Marquet et al. 1993). At the mesocosm scale, these processes (emigration, immigration, transport) are difficult to include and are an often-cited reason why mesocosms diverge from natural systems over time (Bloesch et al. 1988). The biological development of a mesocosm can be highly sensitive to the assemblage of species present at the start of an experiment and to mesocosm size. Consequently, mesocosm measures of extinction, competitive exclusion, or coexistence are likely to have limited applicability to “open” systems where populations are constantly replenished from neighboring regions (NRC 1995).

Physical Processes: The type of mixing (rate, frequency, etc.) applied to mesocosms affects a number of important physically mediated ecosystem processes (Sanford 1997). Effects include the rate of diffusion across boundaries, uptake kinetics, predator-prey contact rates, the time scales for mixing and flushing, and shear effects on organisms. When mesocosms are designed to mimic realistic water column turbulence, unrealistically high fluxes across the sediment-water interface are produced by artificial stirring. A mixing rate designed to achieve a particular light regime may produce higher shear velocities or predator-prey contact rates. Consequently, fine-scale studies in aquatic experimental ecosystems tend to be dominated by diffusive processes, while advection plays a more prominent role in the larger domain of natural ecosystems.

Because the strength of turbulence depends on spatial scale (Denman 1992), the extrapolation of turbulent effects is challenging in mesocosms. Turbulent effects may include those due to shear flows across boundaries (pycnoclines, thermoclines), patchiness generated through advection and eddies, and stratification/destratification events. Mesocosms are generally too limited in their size to include these important larger-scale effects that affect the growth and development of the biotic community and ecosystem-level processes such as nutrient recycling.

Biological Processes: A potentially serious problem of mesocosm studies is the lack of trophic complexity that arises because of truncated space and time scales. The effects of larger organisms or organisms with broad home ranges are particularly problematic (Carpenter et al. 1995). These organisms can have important feedbacks with ecosystem processes or can have a direct “top-down” effect on dynamics. The effects created by organisms at the top of the food chain can be dramatic but very transient or “patchy” in nature. The extrapolation of mesocosm predation rates, for example, based on average values can over- or under-estimate the true rates occurring in the “patchy” non-linear natural system.

Movement patterns are also altered in mesocosms. For example, diurnal vertical migration of zooplankton species may be restricted in a spatially constrained mesocosm with realistic levels of turbulence. Movement is unrealistically restricted for nekton species that are maintained at non-natural densities or confined to restricted spaces that preclude important dynamics such as schooling (Heath and Houde 2001). In addition, the confined spaces of smaller mesocosms prevent or alter escape mechanisms normally

employed by prey species, resulting in predator-prey contact rates between species that can exceed those under natural conditions.

In marine systems, physical and biological variability generally increase at broader temporal and spatial scales (Steele 1978). This means that patch-generating processes at these broad, long-term scales have relatively large effects on ecosystem level dynamics. Examples of such processes include climate shifts, the North Atlantic oscillation, and the ENSO cycle in the Pacific. These events have profound effects on community dynamics and ecosystem functioning (Southward 1980). For instance, a common consequence of the cessation of upwelling off the Peruvian coast due to an El Niño event is the collapse of populations that depend on the high supply of nutrients (Barber and Chavez 1986). The Russell cycle is another important large-scale process (Southward 1980). It results in a dramatic shift in the dominant species of the North Atlantic, thus affecting recruitment and growth in important fisheries such as the cod fishery. Mesocosm experiments cannot address questions at these broad time-space scales, leaving them at a disadvantage to tackle current environmental issues (e.g., fish stock fluctuation, marine pollution, marine mammal-fishery interactions, effects of aquaculture, or establishment of marine protected areas).

1.4 Linking Experimental Ecosystems to Marine Systems

The net result of the issues outlined above is that it is difficult, if not impossible, to simulate the full suite of marine biological-physical interactions in individual mesocosms. Short of enclosing extremely large volumes of water, the inclusion of patch effects within mesocosm experiments remains a significant challenge. The Multiscale

Experimental Ecosystem Research Center (MEERC) was established to develop a fundamental understanding of these scale-dependent effects and their potential implications for understanding natural estuarine ecosystems. Through this center, scientists worked to understand principles for scaling the structure, function, and dynamics of estuarine ecosystems through the use of experimental ecosystems, hydrodynamic experiments, and numerical and simulation models. Research within MEERC used experimental ecosystems (i.e., mesocosms) of various sizes, shapes, and ecological complexity to evaluate scale-related hypotheses (Gardner et al. 2001; Petersen et al. 2003). Experiments using mesocosms of different sizes can capture the change in process that occurs as a result of changes in the geometric dimension of the mesocosm, thus allowing the effects of artifacts due to enclosure to be quantified (Chen et al. 1997; Sanford 1997; Berg et al. 1999; Heath and Houde 2001). The results of these experiments have also uncovered fundamental scaling relationships related to primary productivity in natural systems (Petersen et al. 1997), the impacts of turbulent mixing (Petersen et al. 1998), the design of experimental ecosystems (Petersen et al. 1999), fluxes across boundaries (Sanford and Crawford. 2000) and impacts of fish predation (Mowitt et al. 2006).

Alternatively, modeling studies can provide a cost-effective means of examining scaling relationships and investigating the scale-dependence of a particular system. Models enable researchers to simulate the system of interest (including appropriate levels of physical and biological patchiness) over a range of spatial, temporal and complexity scales. The models can be adjusted to give similar results as fine-scale empirical studies and then used as a tool to investigate dynamics beyond the boundaries imposed by their

restricted time, space, and organismal scales. A wide variety of different habitat configurations, physical conditions, and biotic components can be explored, thereby providing a valuable link between isolated empirical experiments and the natural system. Many of these experiments would not be possible or are too cost prohibitive to conduct within the field.

1.5 Dissertation Objectives

The continued development of scaling relationships from a purely experimental approach will remain limited by the scope of fine-scale experiments and the time and resources necessary to perform them. The ability to detect and explain fundamental scaling relations in aquatic ecosystems must be extended beyond these limitations. The explicit use of simulation models to bridge this gap should prove useful in the identification of scale-dependent processes and the characterization of the domains over which they apply (Gardner et al. 2001; Kemp et al. 2001; Petersen et al. 2003). Thus, a model adequate to explain the fine-grained dynamics of individual mesocosms can also be used to extrapolate results to the scales of the natural system.

My dissertation uses a strong, theoretical modeling approach to examine how processes and patterns translate across spatial and temporal scales and how the spatial distribution (e.g., heterogeneity, patchiness) of resources influences these processes and patterns. To produce a consistent and reliable method for extrapolating detailed, fine-grained dynamics to broad-scale patterns within aquatic environments I developed a robust spatially-explicit modeling framework capable of incorporating a wide variety of different physical, biological, and input components. This was accomplished through the

development of several components: 1) a spatially-explicit simulation framework; 2) a nutrient-phytoplankton-zooplankton-detritus ecosystem model for fine-scale pelagic interactions; 3) realistic physical exchanges between grid-cells; 4) spatially and temporally varying forcing functions and 5) robust pattern analysis and visualization techniques. Each of these components will be described below.

1.6 Components of Modeling System

Modeling Platform: A general framework, the SLS (spatial lattice system), for simulating spatial effects in aquatic systems was developed to explore linkages between temporally and spatially varying ecosystem dynamics over a wide range of scales. The SLS involves the spatial linkage of fine scale ecosystem models (e.g., resolution of meters) into a gridded landscape with exchanges among grid sites simulated by sets of difference equations. The bulk flow of constituents and organisms past a fixed point are recorded (i.e., Eulerian method) as opposed to following particles around as they move within a pre-defined model domain (i.e., Lagrangian method). This methodology has already been implemented for a number of terrestrial systems, including fire (Gardner et al. 1996), population dynamics of plant species (Lavorel et al. 1994), and dispersal of insects and pests (With et al. 1997). The advantage of the SLS is that the solution technique, based on linear rates of exchange via the Euler integration method, is computational efficient allowing large spatial systems to be simulated with high resolution. The method however is restricted to two-dimensional systems.

The model domain can range from one pixel (non-spatial) up to a one million pixel map (current computational limit). A typical simulation is run on a square map, but

other map configurations such as a rectangle are possible. The map boundary conditions can be set as absorbing to approximate an isolated section of water or wrapped to minimize boundary effects. The modeling framework and ecosystem models are written in FORTRAN 95 and run on a Pentium-PC with a Windows operating system. A flexible user interface for the SLS allows generated spatial patterns to be visualized in “real-time” along with corresponding statistical summaries. Output from the simulations can also be sent to data files for later analysis by spatial statistics software (e.g., SAS). Incorporation of a wide range of input functions is also possible (spatially variable nutrient inputs or fish predation) and when fully configured the SLS framework is capable of reproducing physical and biological exchanges typical of the natural environment while handling a range of mixing and or physical forcing scenarios. The SLS also allows a single ecosystem to be simulated in isolation for direct comparison with data from experimental mesocosms. For additional information about the SLS platform see Chapter 2.

Unit Model: Each grid cell of the SLS platform is represented by an aquatic ecosystem model. Replicate models are arrayed across the grid requiring that the model be both simple yet of sufficient detail to represent relevant physical and biological processes affecting estuarine ecosystems. The unit model employed in my dissertation considers nutrient, phytoplankton, zooplankton, and detritus dynamics (NPZD). The NPZD model is composed of sets of generally accepted functional relationships that have been broadly applied and tested. The values of the model parameters were based on rate constants published for similar models and environmental conditions (including MEERC experiments) with parameters estimated to approximate dynamics occurring in the

Chesapeake Bay under stratified summer time conditions. Additional information about the NPZD model can be found in Chapter 2.

Cell-to-cell exchanges: Two physical mixing routines representing the processes of advection and diffusion were included in the SLS. The advection routine utilized a simple technique where the direction and magnitude of the flow could vary with time while the diffusion routine uses a isotropic mass-balance approach which is identical to Fickian diffusion methods. Numerical comparison of the diffusion algorithm to the common advection-diffusion equations found in the oceanographic literature gave similar results. However, the SLS method was more efficient, reducing computation times by nearly an order of magnitude for two-dimensional maps.

Complex flow patterns are a particular challenge because the full equations of motion are not considered by the SLS. Therefore, a novel technique for 2-dimensional turbulence in aquatic systems was developed from a seeded eddy model for particle systems (Dyke and Robertson 1985; Abraham 1998). This method was adapted to a gridded framework, preserving the cascade of turbulent energy from broad- to fine-scales. The technique is theoretically based, computationally efficient, and statistically accurate. The technique also exhibits no numerical diffusion, can simulate scale-dependent mixing and preserves fine-scale concentration gradients. When combined with the diffusion routine to simulate sub-grid mixing, the method has the added advantage of being able to introduce realistic fluid effects within a Eulerian framework without having to resort to complex fluid dynamical models or oversimplifications of turbulent mixing. Additional information about this novel method can be found in Chapter 3.

Spatially and temporally varying forcing functions: Three specially written functions were developed to control model initialization of nutrients and biota into the spatial domain, allow time-varying patterns of nutrient input to be simulated, and to systematically examine the effect of biotic and physical variables on ecosystem processes. These functions are described below.

In many systems the frequency, duration, and magnitude of inputs can have a profound effect on system dynamics (Roughgarden 1978; Reynolds 1993; Petchey et al. 1997). The first function addressed this issue by producing a correlated time-series of model inputs at prescribed frequencies. This input function included random variation (white noise), low-frequency variation (red noise), high-frequency variation (blue noise), as well as constant inputs.

The second function created spatially varying patterns of the forcing functions for nutrients and fish predation. This routine created a circular impact zone within the simulation domain similar to a disturbance in terrestrial systems. The size of the impact ranged from a single pixel to the size of the whole simulation domain with the magnitude and frequency of the impact varied by prescribed values. In addition, the routine allowed the location of impact to vary either randomly or via a correlated random walk. This routine was used for conducting simulated “pulse” and “press” experiments and to simulate spatially and temporally effects (biomass loss) due to fish predation.

The third function generated spatially-explicit patterns based on multifractal map generation techniques (Saupe 1998) to produce two-dimensional patterns of variation in nutrients similar to those observed in natural systems. The method can simulate a range of correlation structures from almost no correlation structure all the way to highly

correlated. Through alterations in this correlation structure, different degrees of “patchiness” can be simulated and matched to what might be found in natural systems for a particular substance or organism.

Statistical Toolbox: A final component of the dissertation was the implementation of a set of quantitative methods to analyze and compare the effects of model experiments on predicted spatial and temporal variation in ecosystem dynamics.

Testing and development of the newly developed NPZD biological model was conducted through sensitivity analysis. The Gardner and Trabalka (1985) method of sensitivity analysis was used to identify the key parameters of the model over a broad range of environmental conditions and to determine conditions for which the model provides reliable predictions. This numerical method used Latin hypercube sampling to simultaneously vary all parameters by +/- 1% of their default value giving an unbiased indication of a model’s sensitivity to a minimal change in parameter value while also taking into account the interactive effects of the other parameters. The technique has been used to address a wide range of issues which include: plankton productivity (Bartell et al. 1988a), toxicological effects (O’Neill et al. 1983), top-down and bottom-up controls on productivity (Bartell et al. 1988b) and forest development (Dale et al. 1988).

Additional information on this method and its application can be found in Chapter 2.

The second component was the statistical analysis of model output in time and space to characterize the spatial heterogeneity of ecosystem dynamics. Spectral analysis has been extensively used for examining patchiness and scale-dependencies in aquatic systems (Platt 1972; Powell et al. 1975; Denman et al. 1977; Lekan and Wilson 1978; Haury et al. 1978; Steele 1985; Weber et al. 1986) and was used here to characterize the

change in variance in data series through time and space (Weber et al., 1986). Details regarding the methods used and the results produced by spectral analysis can be found in Chapters 3 and 4.

1.7 Dissertation Structure

This dissertation is based on a novel framework for assessing scale-dependent effects in aquatic ecosystems. The following chapters present experiments testing each model component followed by an application of the framework testing the effect of spatial and temporal variation in nutrient forcing. Major results and potential applications are also discussed. The appendices provide a listing of the FORTRAN code for the SLS. Each chapter is designed to be submitted as a stand-alone peer-reviewed publication resulting in some duplication of materials among chapters. The structure is as follows:

Chapter 2: The dynamic effects of changing chemical, biological and physical conditions on the predictability of an NPZD pelagic ecosystem model. This chapter describes the NPZD model and uses sensitivity analysis to understand model behavior over a wide range of environmentally relevant conditions.

Chapter 3: A grid-based method for simulating idealized turbulence in aquatic systems is presented. The chapter describes the mixing processes used by the simulation framework (turbulence and diffusion) and validation of the accuracy and utility of the turbulent mixing routine.

Chapter 4: The response of an NPZD model to spatially and temporally varying nutrient input is evaluated. Special emphasis is given to the analysis of nutrient input and

physical and biological processes that occur during advection through an idealized estuary.

The integration of experimental methods, empirical studies, and modeling is required for extrapolation of results across temporal and spatial scales. The SLS is a useful platform for bridging the gap between experimental ecosystems and natural systems through inclusion of “realistic” ecosystem dynamics and providing a quantitatively consistent means for identifying scale-dependent theoretical and empirical “rules” across space, time, and ecological complexity scales.

Chapter 2

Sensitivity analysis of an NPZD pelagic ecosystem model under changing chemical, biological, and physical conditions

Abstract

Marine ecosystem models of nutrient, phytoplankton, zooplankton and detrital dynamics (commonly referred to as NPZD models) simulate ecological systems that can vary greatly as a function of changing environmental conditions (e.g., light, temperature, nutrient inputs, etc.). Sensitivity analysis was applied to identify the dominant parameters and processes affecting the predictions of a generalized NPZD ecosystem model. Systematic experiments were performed over a broad range of environmental conditions, including variation in nutrient input, fish predation, and water exchange rates, to assess the range of relationships between parameters and predictions. Results demonstrated that when environmental conditions were well known only three to five of 28 total parameters directly affected model predictions but the relative importance of individual parameters shifted dramatically when environmental conditions changed. When results were pooled over all experiments, a total of 17 parameters were found to significantly affect model results. Parameters representing the zooplankton maximum growth rate and assimilation efficiency were consistently sensitive across all experiments, supporting previous studies identifying the dominant role of these processes in NPZD models. Other parameters were sensitive over a restricted sub-set of environmental conditions, making the importance of accurate estimates of these parameters situationally dependent. Sensitivity analysis proved to be a powerful means for understanding model

dynamics and identifying where parameter accuracy most affects model predictions. These methods are generally applicable to a broad range of marine biogeochemical models.

2.1 Introduction

Marine biogeochemical models such as the NPZ, NPZD, and similar formulations have been increasingly used to address a wide range of important oceanographic questions including: broad scale dynamics of oceanic plankton production (Dadou et al. 1996), the effects of physical forcings (Goericke and Welschmeyer 1998), iron limitation (Loukos et al. 1997; Denman and Pena 1999), frontal zone dynamics (Franks and Walstad 1997), effects of stochasticity (Steele and Henderson 1992b), nutrient cycling (Oguz et al. 1999), predator/prey dynamics (Edwards and Brindley 1996), impacts of predation (Steele and Henderson 1992a), mixotrophy (Stickney et al. 2000), bloom processes (McGillicuddy et al. 1995), and nutrient enrichment (Murray and Parslow 1999; Kemp et al. 2001). These simple but comprehensive ecosystem models are attractive because they have well characterized dynamics, utilize generally accepted functional response relationships, have been broadly utilized, and have a limited number of parameters to estimate.

As with all models, NPZD models are idealized representations of complex ecosystems. In spite of this simplicity, simulation results are often dependent on uncertain parameters which may vary over a wide range of values depending on the experimental conditions being examined. Ignoring the error or uncertainty associated with each parameter can seriously affect the reliability of model predictions especially if

the processes they affect vary in time and space (O'Neill 1973; Gardner et al. 1982). Determining the effects of parameter variability on model dynamics is therefore one of the most important steps in model development (i.e., design, testing, calibration, validation) and should be rigorously quantified whenever possible (Miller 1974; Oreskes et al. 1994; Caswell 2000).

An important method of model characterization is the use of sensitivity analysis to identify key model parameters and associated processes. Sensitivity analysis assumes that if a model coefficient or a group of coefficients representing a process is found to be highly sensitive then a change in that parameter/coefficient will cause a dramatic change in the model output (Brylinsky 1972; Tomovic and Vukobratovic 1972). Thus, sensitivity analysis highlights the parameters having the greatest effect on model output (e.g., changes in mean levels, productivity) which can then aid model development efforts.

I used the Gardner and Trabalka (1985) method of sensitivity analysis to identify the key parameters of a newly developed NPZD aquatic ecosystem model. This numerical method simultaneously varies parameters by +/- 1% of their default value and statistically estimates the effect of these perturbations on model response. Experiments were performed over a broad range of environmental conditions in order to: 1) identify the most sensitive parameters for each set of conditions; 2) determine conditions for which the model provides reliable predictions, and 3) develop a more general understanding of the dynamics of this NPZD model. To my knowledge, this form of sensitivity analysis has yet to be attempted on the NPZD class of models (i.e., marine biogeochemical) and should complement other studies that have used related methods to

investigate model behavior (e.g., Fasham 1995; Druon and Fevre 1999; Edwards and Brindley 1999; Edwards 2001; Fennel et al. 2001; Halvorsen et al. 2001).

2.2 Methods

2.2.1 General Model Description

An NPZD (nutrient-phytoplankton-zooplankton-detritus) pelagic ecosystem model was developed as one component of a spatially-explicit simulation platform. This work was part of an EPA-funded project that was established to investigate how to extrapolate results from fine-grained experiments to the scales of natural ecosystems (Multiscale Experimental Ecosystem Research Center (see Chapter 1). My model needed to approximate basic dynamics in fine-scale aquatic systems (i.e. mesocosms), both in complexity and behavior, but still be applicable to broader-scale natural systems. Since the model was to be arrayed on a grid to allow for extrapolation through the inclusion of spatial dynamics, the basic structure had to be simple. The NPZD model structure seemed ideally suited for my goals.

The model simulates a well-mixed pelagic water column typical of summer conditions in the Chesapeake Bay and the MEERC mesocosm experiments (Harding et al. 1986; Kemp et al. 1990; Chen et al. 1997; Petersen et al. 1997). Various equations and formulations were taken from other biogeochemical models found in the literature and then modified to meet the needs of the MEERC experiments. Additional information is given in section 2.2.3 below.

Important model features and assumptions are as follows. Seasonal variation in environmental conditions such as changes in temperature and light was not simulated.

Phytoplankton, zooplankton, and detritus were measured in carbon units (gC) while nutrients are in nitrogen units (gN). The fluxes between the state variables are shown in Figure 2.1. Phytoplankton and zooplankton are represented as a single “generic” species, bulk nitrogen is the only nutrient considered and the detrital compartment represents a “static” microbial loop. Phytoplankton growth is affected by nutrient concentration, irradiance level and self-shading effects. Zooplankton can prey on phytoplankton and detritus based on a feeding preference and their respective densities. Nutrients enter through diffusive mixing across the thermocline while exports occur either through the sinking of phytoplankton and detritus or by fish predation. Unit dimensions are in m^3 with a five-minute time-step used to solve the model difference equations via the Euler integration method. The model is written and compiled in FORTRAN 95 and runs on a PC platform.

2.2.2 Model Parameters

The NPZD model contains 28 parameters (see Table 2.1 and Appendix A for a complete listing) with values based on rate constants published for similar models and environmental conditions. Because there was a large variation in published values for parameters, final estimates were determined by a general consensus (i.e., highest frequency) or as a simple average when a consensus did not exist. In most cases, values representing conditions typical of the Chesapeake Bay and the MEERC mesocosm experiments were selected. The model was then calibrated to give stable dynamics (i.e., no oscillations or predator/prey cycles) and equilibrium biomass and concentration levels that might be observed during the summer months in the Chesapeake Bay.

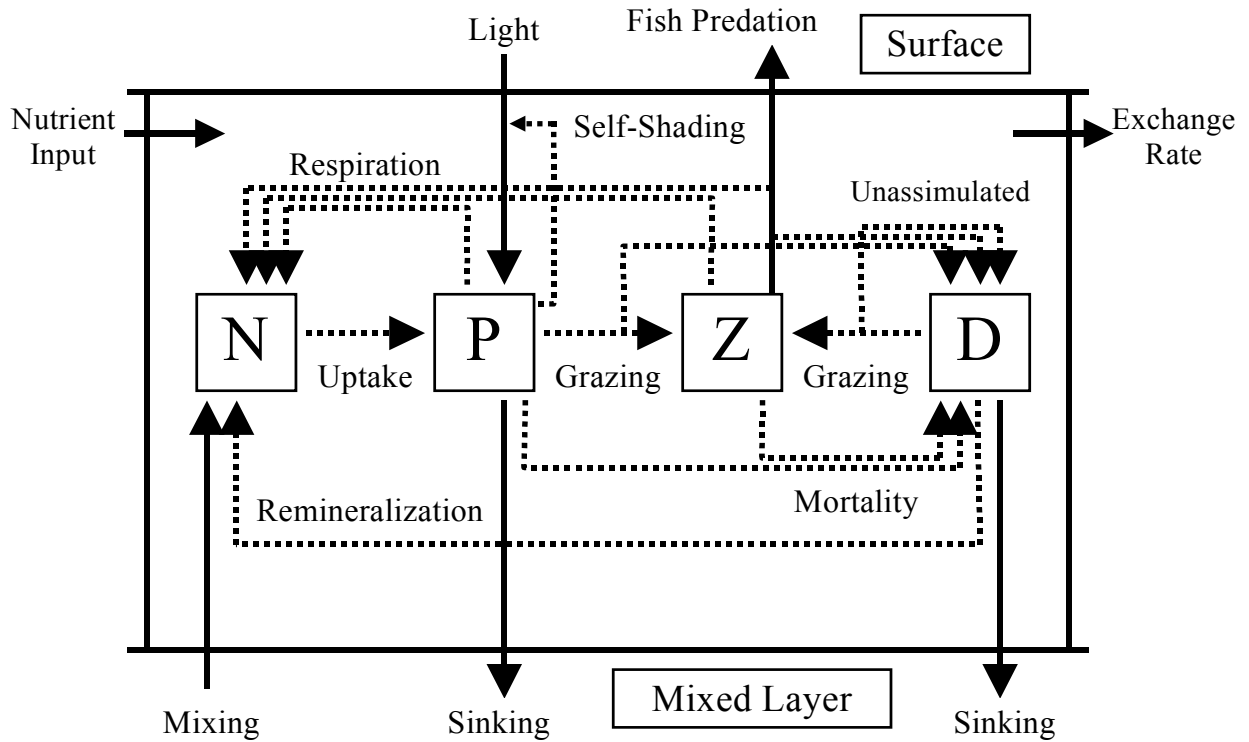


Figure 2.1. Conceptual diagram of the NPZD model used for the sensitivity analysis. External inputs and outputs are in solid lines and internal fluxes between the state variables are represented as dashed lines. See text for additional details.

Table 2.1. Description, symbols and nominal values of model state variables and parameters for sensitivity experiments. Parameters are separated into the state variable in which they appear.

Description	Symbol	Value¹	Units
<i>State variables</i>			
Nutrient	N	0.002	gN m^{-3}
Phytoplankton	P	0.380	gC m^{-3}
Zooplankton	Z	0.056	gC m^{-3}
Detritus	D	0.160	gC m^{-3}
<i>Nutrient parameters, (N)</i>			
Concentration of N below mixed layer	N_o	0.30	gN m^{-3}
Exchange rate across mixed layer	N_d	0.02	day^{-1}
Respiratory losses for fish	R_f	0.10	day^{-1}
<i>Phytoplankton parameters, (P)</i>			
Maximum P growth rate	P_{max}	2.80	day^{-1}
Respiratory losses for P	R_p	0.05	day^{-1}
Mortality losses for P	M_p	0.05	day^{-1}
Sinking losses for P	S_p	0.05	day^{-1}
Light half-saturation constant	I_n	10.00	$\text{E m}^{-2} \text{day}^{-1}$
Surface light intensity	I_o	26.00	$\text{E m}^{-2} \text{day}^{-1}$
Self-shading effects of P	K_c	0.40	$\text{m}^2 \text{gC}^{-1}$
Light attenuation coefficient	K_w	0.20	m^{-1}
Depth of mixed layer	K_z	5.00	m
Nutrient half-saturation constant	K_n	0.02	gN m^{-3}
<i>Zooplankton parameters, (Z)</i>			
Maximum Z ingestion rate	Z_{max}	1.00	day^{-1}
Assimilation efficiency	A_z	0.70	day^{-1}
Respiratory losses for Z	R_z	0.25	day^{-1}
Quadratic mortality losses for Z	M_z	1.00	$\text{day}^{-1} (\text{gC m}^{-3})^{-1}$
Grazing half-saturation constant	k	0.10	gC m^{-3}
Preference for P over D	P_{pref}	0.70	unitless

¹Initial values of state variables, nominal values for parameters, and range of values for external drivers.

Table 2.1 (cont'd). Description, symbols and nominal values of model state variables and parameters for sensitivity experiments. Parameters are separated into the state variable in which they appear.

Description	Symbol	Value¹	Units
<i>Detritus parameters, (D)</i>			
Remineralization rate	R_d	0.20	day ⁻¹
Sinking rate of D	S_d	0.05	day ⁻¹
Unassimilated losses for fish	F_d	0.40	day ⁻¹
<i>External drivers, (Sensitivity Experiments)</i>			
Nutrient input rate	η	0.0-0.64	gN m ⁻³ day ⁻¹
Water exchange rate	μ	0.0-0.20	day ⁻¹
Fish predation rate	ν	0.0-0.80	day ⁻¹

¹Initial values of state variables, nominal values for parameters, and range of values for external drivers.

2.2.3 Model equations

Four coupled ordinary difference equations were used to describe changes in the concentration of nutrient, N , phytoplankton, P , zooplankton, Z , and detritus, D , in a well-mixed water column separated from deeper water by a thermocline.

Phytoplankton: Changes in P concentration were affected by growth, respiration, mortality, sinking, grazing and mixing as indicated in Eq. (1):

$$\frac{\Delta P}{\Delta t} = P_{\max} \text{Min}(f(N), f(L))P - R_p P - M_p P - S_p P - (P_{\text{graz}})Z - \mu P \quad (\text{Eq. 1})$$

P growth is a function of nutrient uptake, $f(N)$, and photosynthesis, $f(L)$, each of which returns a value between zero and one. Based on Liebig's law of the minimum (Liebig 1840), the minimum value of these two functions was used to constrain the maximum phytoplankton growth rate, P_{\max} . The effect of N concentration on P growth rate is modeled as a Michaelis-Menten formulation (Monod 1942):

$$f(N) = \frac{N}{K_n + N} \quad (\text{Eq. 2})$$

N uptake rate, $f(N)$, is a function of the ambient nutrient concentration, N , with K_n as the half-saturation coefficient. The storage of nutrients by P and the separation of nutrients into specific compounds (i.e., nitrate, ammonia, etc) were not considered. Phosphorous and other potential limiting chemical constituents (e.g., silicate, iron) were not modeled because they were assumed to be non-limiting over the conditions tested. The relationship between light intensity and the rate of photosynthesis, $f(L)$, is based on an

analytical solution of the hyperbolic photosynthesis-irradiance curve from formulations by Kirk (1994) and Huisman and Weissing (1994) and is briefly explained below.

Light intensity, I , at a specific depth, K_z , in the water column is formulated with the Beer-Lambert exponential light attenuation equation:

$$I(K_z) = I_o e^{(-K_d K_z)} \quad (\text{Eq. 3})$$

where I_o is the irradiance level just under the water surface and K_d is the light attenuation coefficient due to water. The P biomass production rate, $P(I)$, as a function of light intensity at a specific depth, can then be modeled with a hyperbolic equation where I_n is the light half-saturation constant:

$$P(I) = \frac{I(K_z)}{I_n + I(K_z)} \quad (\text{Eq. 4})$$

Since phytoplankton were assumed to be well-mixed and experience the whole light field, the integral of Eq. (4) was taken over the mixed layer depth to integrate the production rate and divided by water column depth, K_z , to convert from areal to volumetric units:

$$f(L) = \frac{\ln\left(\frac{I_n + I_o}{I_n + I_o e^{(-K_d K_z)}}\right)}{(K_d K_z)} \quad (\text{Eq. 5})$$

Although other production-irradiance formulations could have been used (Cullen 1990), I chose the above formulation because an interpretable analytical solution was available providing an accurate productivity rate with minimal computational costs. This

formulation does not consider photoinhibition effects which were assumed negligible in a well-mixed water column.

The effects of water column turbidity and P biomass on the light attenuation coefficient, K_d , were based after the linear formulation used in Frost (1997):

$$K_d = K_w + K_c P \quad (\text{Eq. 6})$$

The light attenuation coefficient used in Eq. (3) governs how rapidly light is attenuated with depth. Water column turbidity, K_w , is parameterized according to the amount of suspended non-living material present in the water. Turbidity caused by living biomass is parameterized by K_c and reflects the degree to which phytoplankton biomass will block light as it travels through the water column. Eq. (6) provides a realistic feedback where unconstrained P growth, such as from a lack of predation or excess nutrients, will be prevented due to self-shading.

The remaining terms in Eq. (1) represent P losses associated with respiration, R_p , mortality, M_p , sinking, S_p , and dilution of biomass from mixing with the external environment (i.e., water exchange rate, μ , see section 2.5 for additional details regarding the sensitivity experiments). Respiration is the loss of P biomass due to metabolic activities while mortality losses were due to cell senescence. Mortality losses from P were assumed to result in particulate material that remains suspended within the upper water column. Sinking results in biomass lost due to transport across the thermocline and is exported from the model. The grazing term is the amount of biomass lost due to Z predation on P and will be explained further below.

Zooplankton: Changes in Z biomass were affected by growth, respiration, mortality, predation, and mixing as indicated in Eq. (7):

$$\frac{\Delta Z}{\Delta t} = A_z(P_{graz} + D_{graz})Z - R_z Z - M_z Z^q - vZ - \mu Z \quad (\text{Eq. 7})$$

Z growth is a function of the combined P and D grazing rates (P_{graz} , D_{graz}) multiplied by the assimilation efficiency, A_z . The grazing rates were, in turn, a function of the Z maximum grazing rate, Z_{max} , the half-saturation coefficient for grazing, k , and the relative concentration of P and D as shown in Eqs. (8a) and (8b).

$$P_{graz} = Z_{max} \frac{P^M}{k^M + (P + D)^M} \quad (\text{Eq. 8a})$$

$$D_{graz} = Z_{max} \frac{D^M}{k^M + (P + D)^M} \quad (\text{Eq. 8b})$$

The equations for Z grazing were based on the formulation of Fasham et al. (1990) with modifications of the shape factor, M , suggested by Matsuda et al. (1986) and Fasham (1995). The Holling type-2 functional response (i.e., hyperbolic curve) occurs when the coefficient, M , is equal to 1 while a Holling type-3 functional response (i.e., sigmoid curve, Holling 1959) occurs when M is equal to 2. I used a value for M of 1 for all simulations. The functions represented by Eqs. (8a) and (8b) were formulated to return a value between 0 and 1 with the combined value never exceeding 1. This constraint is necessary since Z_{max} represents the maximum daily biomass specific grazing rate regardless of whether Z are feeding on P or D .

An additional term was added to Eqs. (8a) and (8b) to account for a feeding

preference by the Z , producing a final grazing term similar to that of Loukos et al. (1997):

$$P_{graz} = Z_{\max} \frac{(P_{rate} P)^M}{k^M + (P_{rate} P + D_{rate} D)^M} \quad (\text{Eq. 9a})$$

$$D_{graz} = Z_{\max} \frac{(D_{rate} D)^M}{k^M + (P_{rate} P + D_{rate} D)^M} \quad (\text{Eq. 9b})$$

where P_{rate} and D_{rate} were formulated as follows:

$$P_{rate} = \frac{P_{pref} P}{P_{pref} P + (1 - P_{pref}) D} \quad (\text{Eq. 10a})$$

$$D_{rate} = \frac{(1 - P_{pref}) D}{P_{pref} P + (1 - P_{pref}) D} \quad (\text{Eq. 10b})$$

The modified grazing terms in equations (10a) and (10b) are dimensionless functions that account for the relative density of P and D and the preference, P_{pref} , that Z have for feeding on P over D . For example, if the Z show no preference for either P or D then P_{pref} equals 0.5. If P and D densities are equal then P_{rate} and D_{rate} would also be equal (i.e., 0.5); if P are twice as dense then the P_{rate} would be 0.66 and the D_{rate} would be 0.33. I choose Eqs. (9a) and (9b) over other formulations in the literature (e.g., Ivlev) because the expression allows Z to feed on multiple food sources by specifying only two coefficients, P_{pref} and k .

The loss terms of the Z equation were from respiration, mortality, predation, and mixing. Mixing losses result from the exchange of biomass with the surrounding environment and is parameterized with the, μ , coefficient (see section 2.2.5 for additional details regarding the sensitivity experiments). Respiratory losses were parameterized as a

linear loss term controlled by the Z respiration coefficient, R_z , while mortality, M_z , may be either a linear ($q = 1$) or quadratic ($q = 2$) function of Z biomass. The Z mortality term is often used as a “closure term” for ecosystem models to represent higher-order effects such as predation or internal density-dependent regulation factors (e.g., cannibalization).

The specific form of the Z mortality term (e.g., linear, quadratic, etc.) is still an active area of debate due to the dramatic impact that its parameterization can have on model dynamics (Steele and Henderson 1992a; Edwards and Brindley 1996). I choose a quadratic mortality function to simulate increasing mortality due to cannibalism at higher biomass levels. The quadratic formulation also has the added benefit of reducing instabilities and attenuating swings in predator-prey cycles that may occur with linear loss terms (but see Caswell and Neubert 1998; Edwards and Bees 2001). Mortality losses from Z were assumed to result in particulate material that remains suspended within the water column.

The “closure” term for this NPZD model represents the removal of Z biomass due to mobile predators such as fish. The fish predation rate, ν , is used to approximate this relationship as a linear loss of Z biomass. Half of the Z biomass removed by fish predation is incorporated as fish biomass (i.e., exported from the model) while the rest is recycled as input to the N and D pools. This partitioning of Z biomass allows the NPZD model to be incorporated into a spatially-explicit modeling framework (i.e., future simulations) to consider how spatially variable fish predation affects spatial patterns in aquatic systems. In this chapter, I use the (ν) coefficient to vary the amount of fish predation in the sensitivity experiments (see section 2.2.5 for additional details).

Detritus: The detrital material, D , found in the water column is simulated as a simplified “microbial” loop consisting mainly of bacteria and protist feeding on particulate material (i.e., fecal pellets, dead phytoplankton and zooplankton; see: Steele 1998; Fasham et al. 1990; Loukos et al. 1997). Eq. (11) represents the key processes affecting the amount of D within the water column as:

$$\frac{\Delta D}{\Delta t} = (M_p P + M_z Z^q) + (1 - A_z)(P_{graz} + D_{graz})Z + F_d v Z - R_d D - S_d D - (D_{graz})Z - \mu D \quad (\text{Eq. 11})$$

Inputs to the D compartment were from the P and Z mortality loss terms ($M_p P + M_z Z^q$) as well as unassimilated biomass from grazing by zooplankton and fish. The unassimilated portion due to fish predation on Z is represented by the F_d coefficient while $(1 - A_z)$ is the proportion of grazed P and D that is unassimilated from the Z grazing formulations. The organisms within the compartment break down the particulate material, returning a portion back into soluble nitrogen, a linear process regulated by the remineralization coefficient, R_d . Much of the suspended particulate material will settle out of the water column at a rate proportional to the D sinking velocity, S_d . Any material lost through this mechanism is assumed to be mixed below the thermocline and exported from the model. The Z grazing rate on D is the loss of D biomass due to Z grazing and has been previously described in the Z state variable section. The final loss term for D is the removal of biomass due to mixing with the external environment (i.e., water exchange rate, μ , see section 2.2.5 for additional details regarding the sensitivity experiments). The explicit consideration of a D compartment enables a wider range of possible processes to be simulated, including the separation of N regeneration into a fast and slow turnover pool and for Z to feed on an additional food source other than P .

Nutrients: Changes in the nutrient pool, N , are shown in Eq. (12):

$$\frac{\Delta N}{\Delta t} = \eta + N_d(N_o - N) + (R_p P + R_z Z + R_f v Z) + R_d D - P_{\max} \text{Min}(f(N), f(L)) P - \mu N \quad (\text{Eq. 12})$$

The nutrient state variable, N , is the total nitrogen dissolved in the water column (i.e., mainly nitrate and ammonia). Many studies have modeled these two separate forms of nitrogen as distinct compartments to address a variety of issues or questions such as separating new versus regenerated production (e.g., McGillicuddy et al. 1995) or examining the inhibitory effects of ammonium on nitrate uptake by phytoplankton (e.g., Fasham 1995). This level of detail was not required in my simulations so only changes to bulk nitrogen were considered.

Inputs to the N state variable occur via external input, mixing, respiration, and regeneration. Mixing occurs via the diffusive exchange of nutrients from below the thermocline, modeled after Steele and Henderson (1981) and Edwards and Brindley (1996) where N_d is the fraction of the mixed layer which is exchanged daily with the deeper water. The N pool below the mixed layer, N_o , remains constant under the assumption that sources of N below the mixed layer balance losses due to diffusion. Water column inputs of N occur through P , Z , and fish respiration and the regeneration of N through microbial breakdown of D . The R_f coefficient represents that portion of the Z predation loss term that is returned to the N state variable due to fish respiration. Nutrients can also enter the model externally through a spatially and or temporally varying input function, η , allowing N inputs from a wide variety of sources to be simulated (e.g., pulsed upwelling events, inputs from riverine sources). For the purposes of this chapter, I used this parameter to force the model under a range of nutrient input

scenarios as part of the sensitivity analysis (see section 2.2.5 for further details regarding the sensitivity experiments).

The only two loss terms for the N compartment were through uptake by P to generate biomass and from mixing with the external environment (i.e., water exchange rate, μ , see section 2.2.5 for additional details regarding the sensitivity experiments). Since nutrients were modeled in (gN), as opposed to (gC) for the other state variables, inputs from respiration and regeneration and losses from growth were first converted into nitrogen units using a standard stoichiometric ratio of 0.1761 based on Redfield stoichiometry and accounting for difference in weight between nitrogen and carbon (Redfield et al. 1963; Denman and Pena 1999).

2.2.4 Sensitivity Analysis

The classical sensitivity index, first developed by Tomovic and Karplus (1963), is the partial derivative of the model predictions, Y_i , with respect to each model parameter, P_i . Numerical estimates of parameter sensitivities for the NPZD model were estimated using PRISM, a program that performs Monte Carlo simulations with parameter values generated via Latin Hypercube sampling (Gardner and Trabalka 1985). The data set of parameters and associated model predictions is obtained by iterative model simulations. The resulting data sets are then analyzed via linear regression of the predicted state variable value on the parameter values to estimate model sensitivities (Gardner et al. 1983). As long as the variance of the model parameters is small (i.e., $\pm 1\%$), the slopes of the regressions are equal to the analytic estimates of the partial derivative (Gardner et al. 1981). For each of the sensitivity experiments outlined in the next section, the

normalized sensitivity index, U_i , was estimated by dividing the sum of squares of the regression by the total sum of squares multiplied by 100. The value U_i ranges from zero to one hundred for all parameters regardless of the units with which they were measured (Gardner and Trabalka 1985; Gardner et al. 1990). See Appendix B for additional details concerning this and other methods of sensitivity analysis.

2.2.5 Experimental Treatments

Previous studies on model behavior and sensitivities have shown that parameter sensitivities are dynamic and can change depending on the state of the model (e.g., top-down/bottom-up control, threshold and saturation effects, nutrient status, etc.) at the time of the sensitivity analysis (Bartell et al. 1986; Bartell et al. 1988b; Fong et al. 1997; Klepper 1997; Pastres et al. 1997). To address these issues and obtain a broad estimate of parameter-prediction dependencies, the sensitivity analysis was performed across a wide range of nutrient input, fish predation, and water exchange rates. In all cases the sensitivity analysis was conducted after equilibrium was reached at 90-days.

Treatment Series 1: Varying nutrient input rate (η). Changes in nutrient state, from oligotrophic to eutrophic conditions, were investigated by progressively increasing the daily input of nutrients into the water column (Table 2.2). The range of conditions considered approximate seasonal changes in ambient nutrient concentration that might be experienced in the Chesapeake Bay from variations in riverine input or exchange rate across the thermocline (Boynton et al. 1995; Boynton and Kemp 2000).

Treatment Series 2: Varying water exchange rate (μ). Losses due to mixing with the environment outside of the model domain, as might be experienced due to variations

Table 2.2. The sensitivity experiments. A. Three independent series of sensitivity analysis were performed varying only the nutrient addition (η), water exchange (μ) or fish predation rates (ν). B. Interactive effects were evaluated by two factorial experiments: (I) all combinations of η with μ (36 treatment combinations), and (II) all combinations of η with ν (24 treatment combinations). 100 Monte Carlo iterations were performed for each sensitivity experiment. See text for additional details.

A. Independent effects		
η (gN m ⁻³ day ⁻¹)	μ (% day ⁻¹)	ν (% day ⁻¹)
0.00	0.00	0.00
0.04	1.25	20.00
0.08	2.50	40.00
0.16	5.00	80.00
0.32	10.00	
0.64	20.00	
B. Interactive effects (factorial experiments)		
I		$\eta \times \mu$
II		$\eta \times \nu$

in advection and turbulence were reflected by changes in the water exchange rate, μ . I varied this parameter from the baseline case representing a closed system to a range of exchange rates approximating residence times from 2.5 days to 320 days (Table 2).

Treatment Series 3: Varying fish predation rate (ν). Loss of zooplankton via fish predation is simulated as a linear loss term, ν . Since fish predation is variable in space and time, I tested model sensitivity to a range of possible values from no predation to very high predation rates (80% loss rate day⁻¹, Table 2.2). This range reflects differing degrees of predation by larger/more numerous fish schools or additional losses to zooplankton from disease or cannibalism beyond that already accounted for by the natural mortality, M_z , parameter.

Interactive Effects. The interactive effects of the nutrient input rate, η , water exchange rate, μ , and fish predation rate, ν , were examined by two factorial experiments. The first series examined the interactive effects of the nutrient input rate, η , and the water exchange rate, μ . Under these treatment combinations 36 scenarios were conducted over the range of values listed in Table 2.2 with the fish predation amount held constant at a value of (20%) to reflect average predation pressure on the zooplankton by fish and other external mechanisms. The second series examined the interactive effects of the nutrient input rate, η , and the fish predation rate, ν . Under these treatment combinations 30 scenarios were conducted over the range of values listed in Table 2.2 with the water exchange rate set to zero to eliminate the effects of this treatment variable in this series of simulations.

2.3 Results

Significant changes in the relationship between model parameters and predictions were produced by changes in the external drivers (i.e., nutrient addition, η , fish predation, ν , and water exchange, μ). I will discuss these patterns of change in terms of the overall model performance, and then examine the direct and interactive effects of changes in the external drivers.

2.3.1 General sensitivity dynamics

The model state variables adjusted to changes in the external drivers by establishing a new equilibrium concentration after a short period of time. Figure 2.2 shows the model response for one combination of external drivers and Figure 2.3 shows the equilibrium concentration of each state variable over the range of treatment combinations for the two factorial experiments. Enrichment with nutrients causes a bloom in P biomass and subsequent increases in Z and D biomass. An increase in fish predation results in lower Z and D biomass and in higher P while an increase in the water exchange rate reduces the overall concentration levels of all the state variables, except N which tended to increase. Under higher nutrient input levels the rate of growth of P , Z , and D were maximized causing N concentration to build up over time (data not shown).

Parameter sensitivities were estimated once the model state variables adjusted to the new equilibrium (Figure 2.4). For the purpose of this chapter, I define an “important” parameter as one whose sensitivity, U , explains greater than 10% of the variance in model output (i.e., % variance explained). For example, in Figure 2.4, the parameters

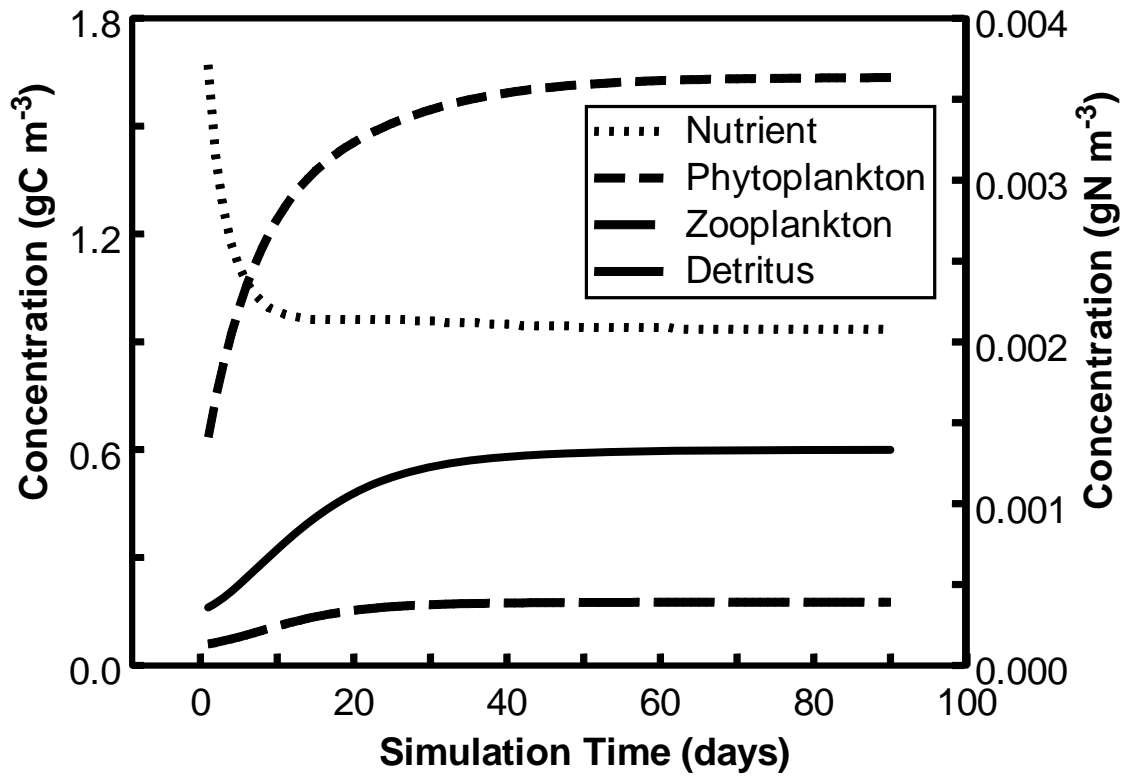


Figure 2.2. Temporal response of each state variable to a typical treatment combination. In most cases a new equilibrium is established after a short period of transient dynamics. Values on the right axis are for the nutrient state variable. Case shown ($\eta = 0.04 \text{ gN m}^{-3} \text{ day}^{-1}$, $\mu = 2.5\% \text{ day}^{-1}$).

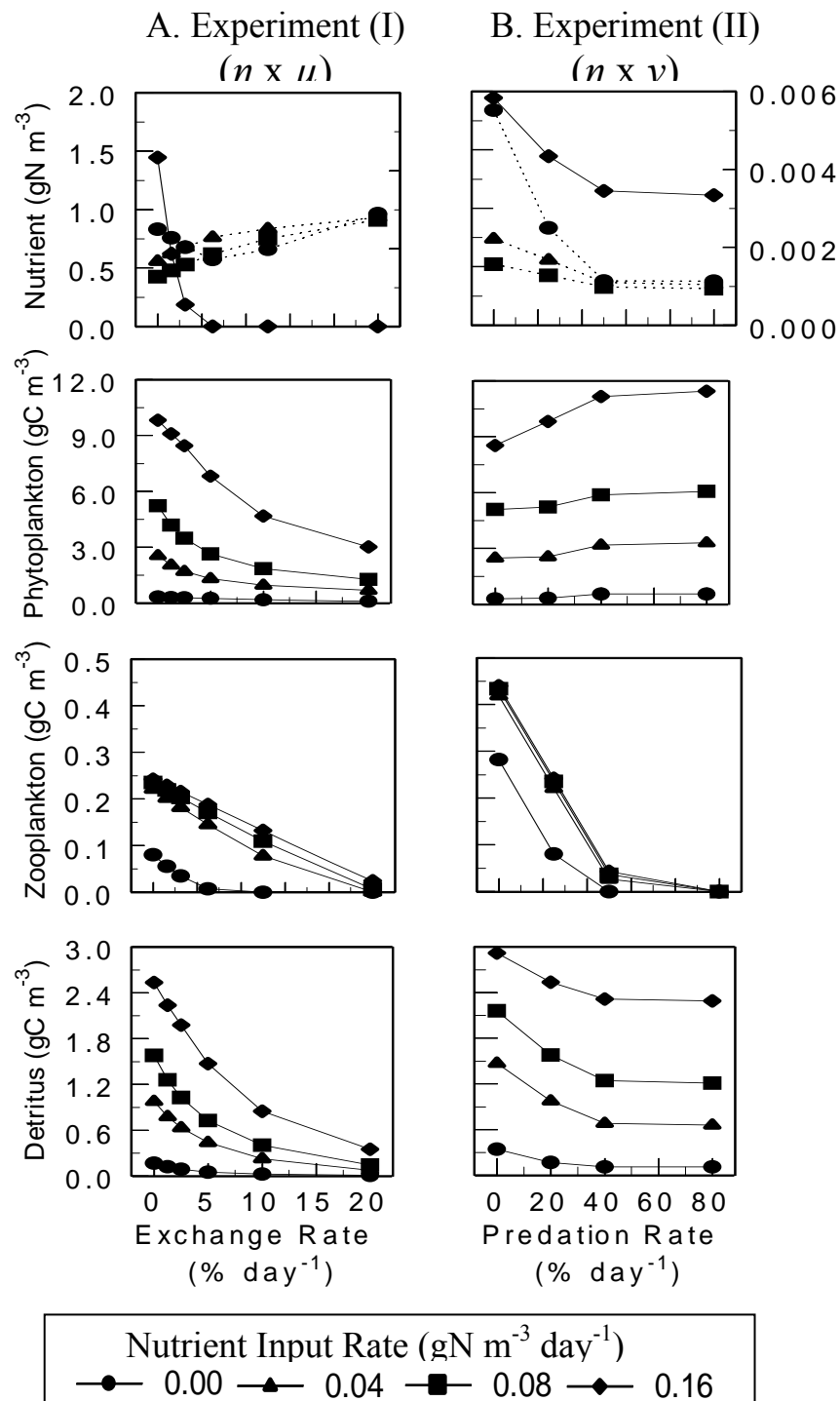


Figure 2.3. Equilibrium concentration values for each state variable under the various treatment combinations. The effects of increasing the water exchange and fish predation rates are shown in the left (A) and right panels (B), respectively for each of the nutrient input treatment amounts. Only the first four nutrient input treatments are shown because the dynamics saturate at higher nutrient input amounts for the P , Z , and D state variables. The N state variable at the higher nutrient input amounts shows a similar response as the ($\eta = 0.16 \text{ gN m}^{-3} \text{ day}^{-1}$) nutrient input treatment except the mean concentration levels observed are elevated. The dashed lines for the top two panels are scaled to the right hand axes for the N state variable graphs.

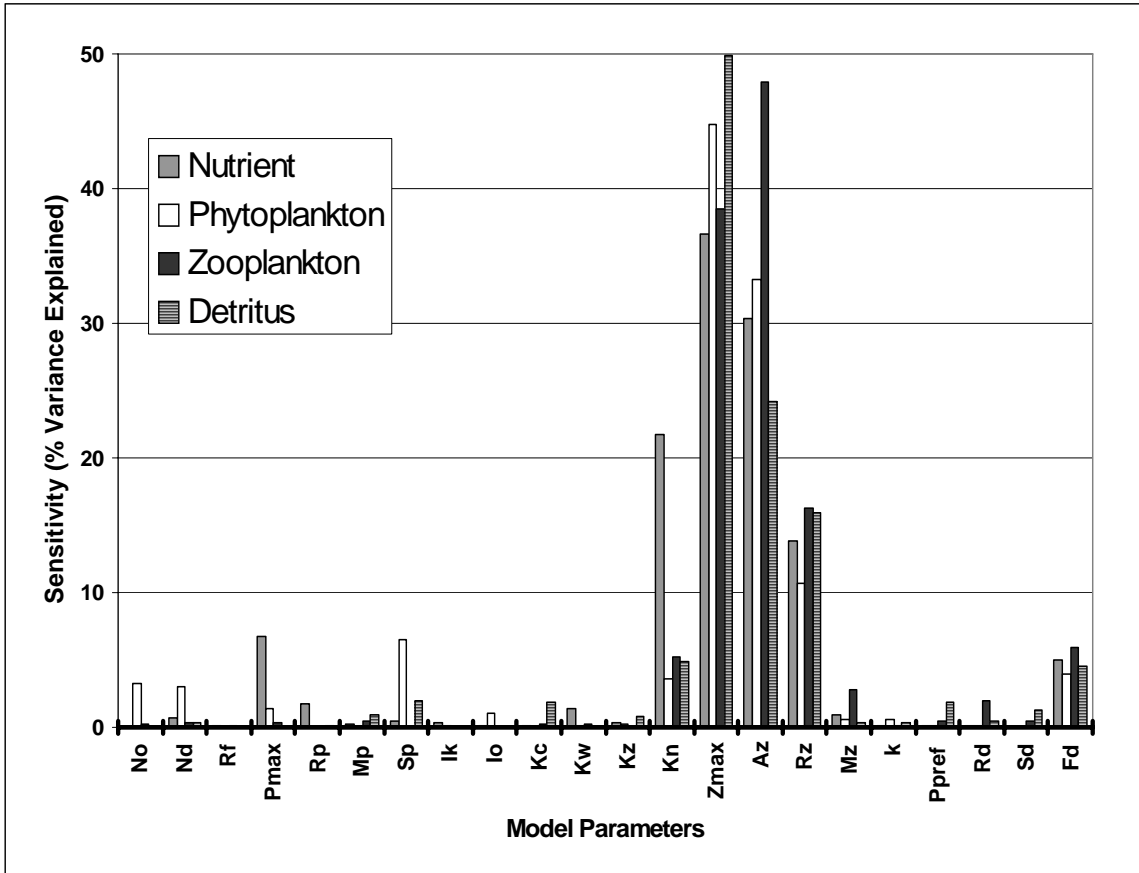


Figure 2.4. Graph showing sensitivity data for each parameter from a representative treatment combination. Case shown ($\eta = 0.04 \text{ gN m}^{-3} \text{ day}^{-1}$, $\mu = 2.5\% \text{ loss day}^{-1}$). Only parameters explaining greater than 10% of the model variance were retained for further analysis.

Z_{max} , A_z , R_z , was important for all the state variables while K_n would be important for only the N state variable. The 10% criterion is deemed reasonable because typically three to five parameters exceed this value and explain most of the variance in model output for a particular state variable. Only parameters identified as important for at least one set of treatment conditions were retained for further analysis.

2.3.2 Sensitivity response to variation in external drivers (*Independent effects*)

Parameter sensitivities may vary as a function of the external drivers (η , μ , or ν) and state variables (N , P , Z or D) being considered. Table 2.3 illustrates the changes in sensitivities measured for each external driver, while holding the other drivers constant at baseline values (Table 2.2a). The direct effect of the broad range of conditions examined shows that six parameters were consistently important: Z_{max} , A_z , R_z under all conditions; N_o and N_d were each important in six of 12 cases; while P_{max} was important in five of 12 cases. Although the prediction of N , P and D involves the specification of 5-10 parameters, changes in Z required accurate determination of only three to four parameters (Table 2.3). Changes in sensitivity exhibited four basic types of response over the broad ranges of conditions tested: positive, negative, no response and complex. These patterns of response were summarized in Table 2.3 and discussed below for each of the external drivers.

The effect of variable nutrient input rates: Nutrient additions had a broad effect on patterns of parameter sensitivities. For this treatment series, all of the parameters except one (i.e., R_p) were sensitive in at least one of the state variables. The response

Table 2.3. Sensitivity analysis for independent effects (see Table 2.2A). Symbols indicate that increasing values of η , μ or ν resulted in significant sensitivities that either: increased (+), decreased (-), were unresponsive (=), or were complex (*)². Many parameters demonstrated a critical threshold to increasing values of η (not shown). See text for additional details.

Parm ¹	<i>N</i>			<i>P</i>			<i>Z</i>			<i>D</i>		
	η^3	μ	ν	η^3	μ	ν	η^3	μ	ν	η^3	μ	ν
N_o				-	+	*				-	*	=
N_d	+				+	*				-	*	=
P_{max}	*	+	+	=						=		
R_p			+									
M_p										=	*	+
S_p				+		+						+
K_c	-			=						=		
K_z	-			=						=		
K_n	+	*	+									
Z_{max}	-	-	-	-	-	*	=	-	=	*	*	*
A_z	-	-	*	-	-	*	=	-	-	*	*	
R_z	-	-	*	-	*	*	=	-	*	*	*	-
M_z							-		-			
P_{pref}										-		-

¹Only parameters with sensitivities greater than 10% are shown. See Table 1.1 for parameter definitions.

²Complex sensitivities were those that demonstrated a non-consistent response to variations in the external drivers. All of the parameters in this classification responded positively at first and then declined except for the N_o and N_d parameters which had the opposite response.

³For the η treatment series the results were combined for the ($\eta \times \mu$), and the ($\eta \times \nu$) factorial baseline cases.

(e.g., positive, negative) of a parameter to increasing nutrients often changed between state variables. For example, the Z_{max} , A_z , and R_z parameters became less sensitive with increasing nutrients in the N and P state variables, were unresponsive in the Z state variable, and in the D state variable responded positively at first and then decreased. The parameters with the strongest response to nutrient additions were the $N_d(N)$, $S_p(P)$, and $Z_{max}(D)$ parameters (data not shown). While none of the parameters in the Z state variable responded very strongly to changes in the rate of nutrient additions the $A_z(Z)$ parameter was the most sensitive. Many of the parameters sensitive to nutrient additions also exhibited a threshold response where the parameter either became sensitive (e.g., K_c , K_z , P_{max}) or insensitive (e.g., Z_{max} , A_z , R_z) above the ($\eta = 0.16 \text{ gN m}^{-3} \text{ day}^{-1}$) treatment level.

The effect of variable water exchange rates: Increasing the water exchange rate resulted in fewer parameters exhibiting sensitivity than the increases in the nutrient input rate listed above. Only eight out of 14 parameters were sensitive in at least one of the state variables. The K_c and K_z parameters were no longer sensitive for any of the state variables and the P_{max} parameter became insensitive for the P and D state variables. Although the parameters exhibited a variety of responses to changing exchange rates, no examples of a threshold response were observed. The strongest responses in the N and P state variables were from the P_{max} and Z_{max} parameters respectively and the A_z parameter had the strongest response in the Z and D state variables (data not shown). All of the sensitive parameters in the D state variable had a complex (i.e., curvilinear response) to increases in the water exchange rate (Table 2.3).

The effect of variable fish predation rates: The sensitivity dynamics observed when the fish predation rate was increased were very similar to the other two treatment series. A majority of the parameters sensitive for this series of experiments were also sensitive for the other two treatment series (e.g., K_n , Z_{max} , R_z). The R_p parameter in the N state variable was the only parameter sensitive solely in the fish predation series. Some parameters were sensitive in only the nutrient input and predation series (e.g., M_z , P_{pref}) while the $N_d(P)$ parameter was the only one sensitive for the water exchange and fish predation series. The parameters with the strongest response to increases in the fish predation rate were the Z_{max} parameter in the N and P state variables, the A_z parameter in the Z state variable and the N_o parameter in the D state variable (data not shown).

2.3.3 Simultaneous variation of external drivers (Factorial experiments)

The previous sections described how the parameter sensitivities changed as a function of increases in the magnitude of the external drivers. In the following sections I examine the interactive effects by two factorial experiments (Table 2.2b), the combined changes in the nutrient input and water exchange rates ($\eta \times \mu$) and nutrient input and fish predation rates ($\eta \times \nu$). These experiments provide important insights into the domain over which each parameter affects model predictions. Tables 2.4-2.5 and Figures 2.5-2.6 illustrate the patterns observed for each experiment and are discussed below.

Overall sensitivity patterns: Consideration of all treatment combinations identified 15 sensitive parameters with 11 parameters common to both experiments (Table 2.4). The parameters with the highest average sensitivity were $P_{max}(N)$, $Z_{max}(P)$, $A_z(Z)$, and

Table 2.4. Results for the factorial series of sensitivity experiments (Table 2.2B). Sensitivities of each state variable are shown for the two factorial experiments: (A) nutrient input versus water exchange ($\eta \times \mu$), and (B) nutrient input versus fish predation ($\eta \times \nu$). The average sensitivity over all the treatment combinations (*mean*), the min/max sensitivity values (*range*) and the frequency of sensitivity values $>$ the 10% cutoff criterion for sensitive parameters (*f*) is displayed.

A. $\eta \times \mu$		State Variables							
Parm¹	<i>Nutrient (N)</i>		<i>Phytoplankton (P)</i>		<i>Zooplankton (Z)</i>		<i>Detritus (D)</i>		
	mean (range)	<i>f</i>	mean (range)	<i>f</i>	mean (range)	<i>f</i>	mean (range)	<i>f</i>	
<i>Z_{max}</i>	14 (0-38)	0.47	21 (3-47)	0.50	36 (12-40)	1.00	18 (0-50)	0.50	
<i>A_z</i>	13 (0-35)	0.50	14 (0-37)	0.50	46 (26-49)	1.00	13 (0-32)	0.56	
<i>R_z</i>	7 (0-22)	0.39	5 (0-15)	0.19	17 (15-24)	1.00	10 (1-19)	0.47	
<i>P_{max}</i>	18 (3-49)	0.64	9 (0-21)	0.42			8 (0-19)	0.42	
<i>K_z</i>	14 (0-42)	0.42	15 (0-36)	0.42			9 (0-24)	0.42	
<i>K_c</i>	7 (0-21)	0.33	9 (0-22)	0.42			12 (1-26)	0.42	
<i>N_d</i>	7 (0-69)	0.17	5 (0-42)	0.08			4 (0-25)	0.17	
<i>N_o</i>			5 (1-41)	0.11			4 (0-30)	0.17	
<i>K_n</i>	16 (0-38)	0.58							
<i>S_p</i>			7 (0-47)	0.17					
<i>M_p</i>							11 (0-27)	0.56	

¹ Only parameters with sensitivities greater than 10% are shown. See Table 2.1 for parameter definitions.

Table 2.4 (cont'd). Results for the factorial series of sensitivity experiments (Table 2.2B). Sensitivities of each state variable are shown for the two factorial experiments: (A) nutrient input verses water exchange ($\eta \times \mu$), and (B) nutrient input versus fish predation ($\eta \times \nu$). The average sensitivity over all the treatment combinations (*mean*), the min/max sensitivity values (*range*) and the frequency of sensitivity values > the 10% cutoff criterion for sensitive parameters (*f*) is displayed.

B. $\eta \times \nu$		State Variables							
Pm¹	<i>Phytoplankton (P)</i>		<i>Zooplankton (Z)</i>		<i>Detritus (D)</i>				
	<i>mean (range)</i>	<i>f</i>	<i>mean (range)</i>	<i>f</i>	<i>mean (range)</i>	<i>f</i>	<i>mean (range)</i>	<i>f</i>	
<i>Z_{max}</i>	11 (0-43)	0.33	12 (0-44)	0.33	29 (0-40)	0.96	11 (0-48)	0.29	
<i>A_z</i>	8 (0-33)	0.33	8 (0-34)	0.25	36 (0-50)	0.96	6 (0-21)	0.21	
<i>R_z</i>	5 (1-20)	0.25	3 (0-14)	0.04	15 (0-22)	0.96	6 (0-14)	0.29	
<i>N_d</i>	18 (0-70)	0.33	6 (0-35)	0.13			4 (0-24)	0.17	
<i>P_{max}</i>	20 (3-40)	0.71	11 (2-22)	0.50			9 (0-18)	0.50	
<i>K_z</i>	14 (0-35)	0.50	19 (0-38)	0.50			13 (0-27)	0.50	
<i>K_c</i>	9 (0-21)	0.50	11 (0-24)	0.50			15 (2-26)	0.50	
<i>S_p</i>			21 (1-89)	0.42			10 (4-34)	0.30	
<i>N_o</i>			6 (1-39)	0.17			6 (0-32)	0.17	
<i>M_z</i>					5 (0-15)	0.25			
<i>R_p</i>	4 (1-12)	0.17							
<i>K_n</i>	16 (0-41)	0.50							
<i>R_d</i>							4 (0-14)	0.08	
<i>P_{pref}</i>							1 (0-14)	0.04	
<i>M_p</i>							17 (0-43)	0.75	

Table 2.5. The degree of interaction observed between treatment factors for each factorial experiment ($\eta \times \mu$, $\eta \times \nu$). Illustrated is the frequency (percentage) that the sensitivity results seen along one of the treatment axes (η , μ , ν) for each parameter will be indicative of the sensitivity dynamics when the other factor is varied. A higher percentage indicates less interaction and consistency of results across the other treatment series. A low probability signifies a significant interaction with the other treatment series. Significant interactions at the ($p=0.01$) probability level were identified through Chi-square analysis and are marked in bold.

Pm¹	N			P			Z			D		
	η^2	μ	ν	η^2	μ	ν	η^2	μ	ν	η^2	μ	ν
N_o				--		10				0/0	10	10
N_d	80/	80	10							0/0	10	10
P_{max}	57/	83	83	50/	90	10				50/4	90	10
R_p	--		78									
M_p										47/7	80	67
S_p				80/	73	89				--/65		61
K_c	60/	80	10	50/	90	10				50/4	90	10
K_z	50/	90	10	50/	90	10				50/4	90	10
K_n	50/	90	10									
Z_{max}	50/	77	78	47/	80	78	100/	10	95	47/6	80	83
A_z	53/	87	78	47/	80	78	100/	10	95	40/7	67	83
R_z	--		67	43/	77	94	100/	10	95	50/6	77	72
M_z							--		0			
P_{pref}										--/75		83
R_d										--/90		89

¹Only parameters with sensitivities greater than 10% are shown. See Table 2.1 for parameter definitions.

²Since the nutrient treatment series, η , appears in both factorial experiments, there are two probability values, the first percentage is for the ($\eta \times \mu$) series and the second value is for the ($\eta \times \nu$) series.

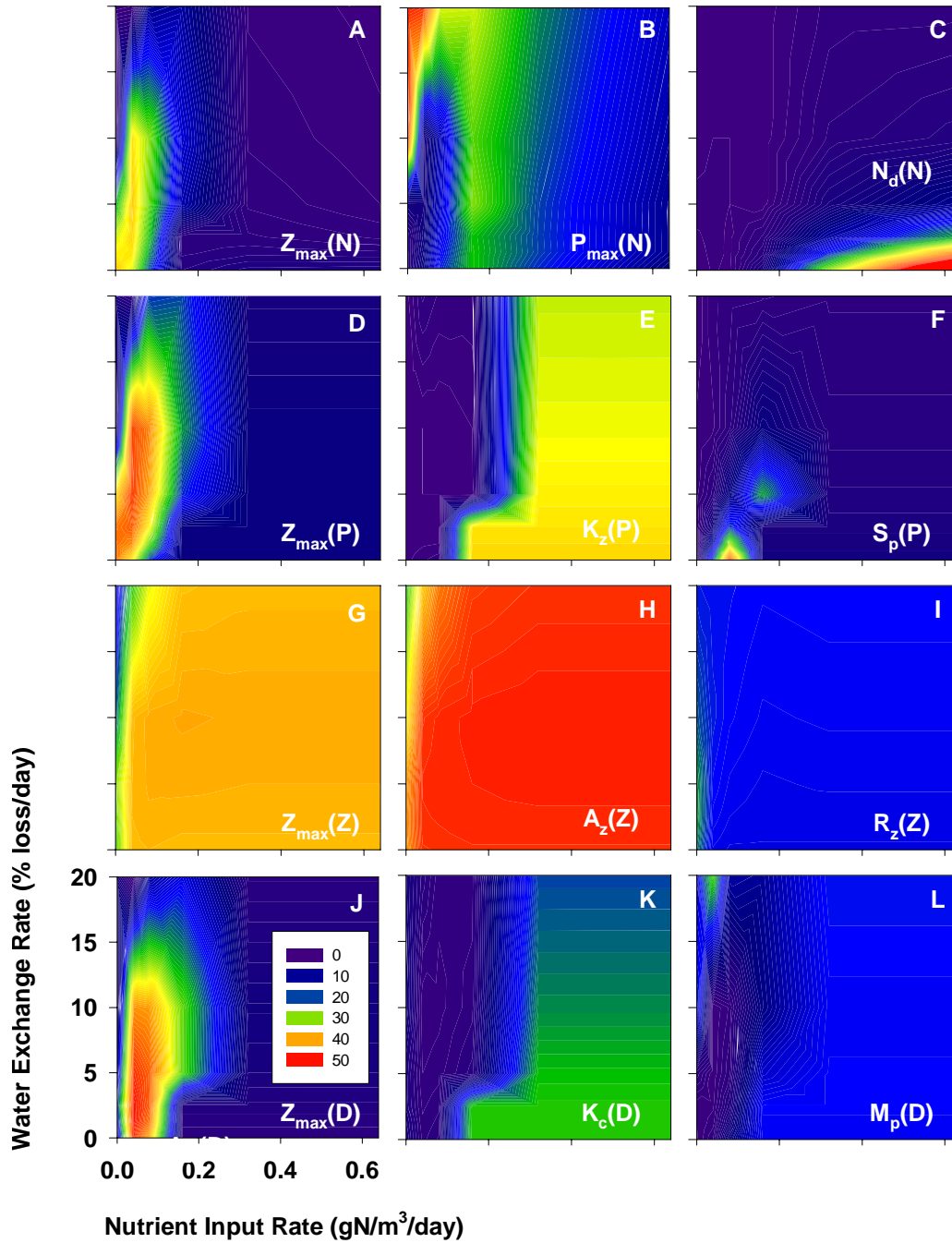


Figure 2.5. Contour plots of the effects of nutrient input (η , x axis) and water exchange (μ , y axis) on the sensitivity of selected parameters for each state variable. Each row of plots represents a single state variable (A-C = N, D-F = P, G-I = Z, J-L = D). The Z_{max} parameter is represented in each state variable (plots A, D, G, J). Parameters were chosen to illustrate the various types of patterns (e.g., uniform, isolated, threshold, gradients) observed over the state space covered by the treatment combinations and to allow comparison between the two factorial experiments (see Figure 2.6).

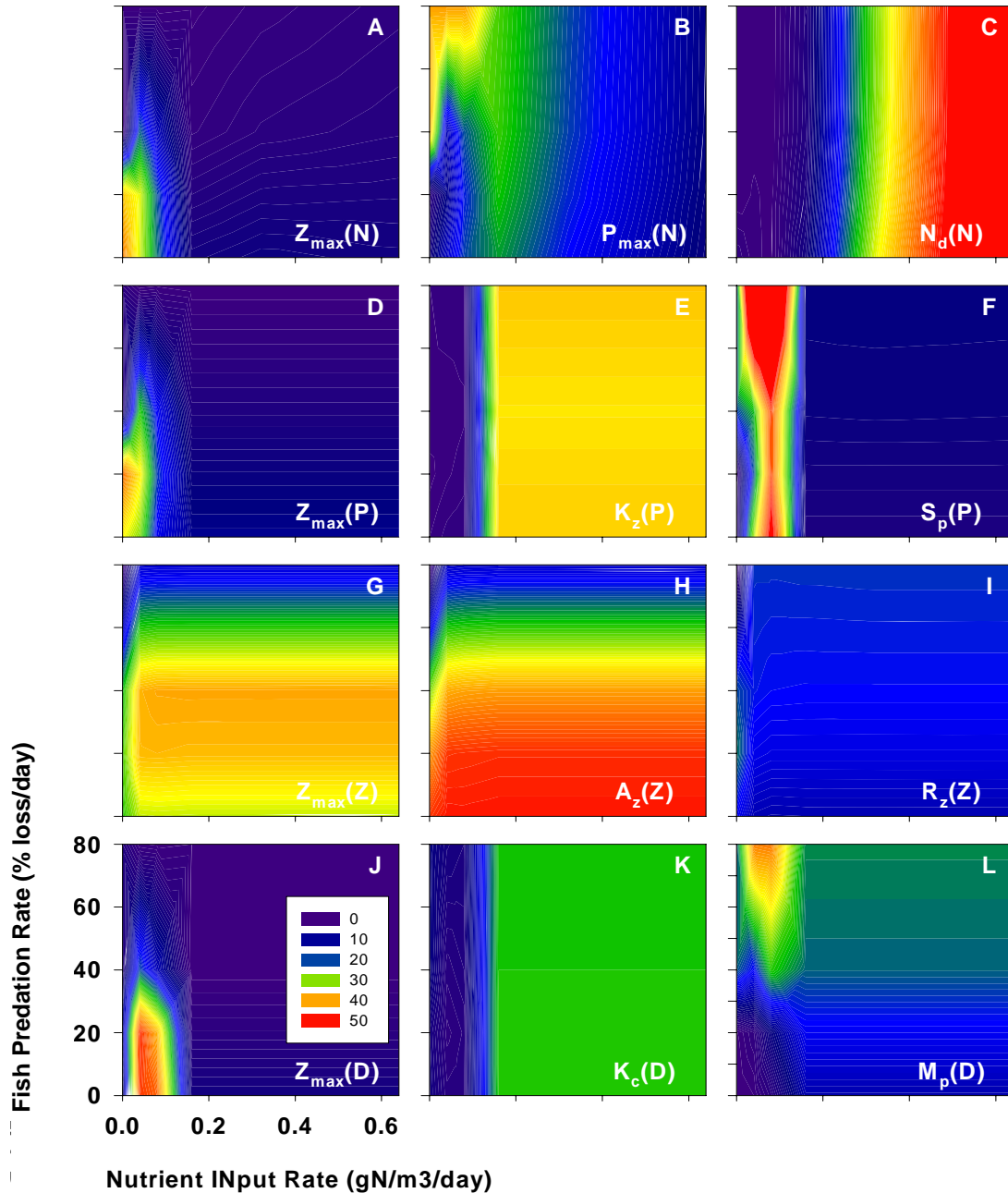


Figure 2.6. Contour plots of the effects of nutrient input (η , x axis) and fish predation (v , y axis) on the sensitivity of selected parameters for each state variable. Each row of plots represents a single state variable (A-C = N , D-F = P , G-I = Z , J-L = D). The Z_{max} parameter is represented in each state variable (plots A, D, G, J). Parameters were chosen to illustrate the various types of patterns (e.g., uniform, isolated, threshold, gradients) observed over the state space covered by the treatment combinations and to allow comparison between the two factorial experiments (see Figure 2.5).

$Z_{max}(D)$ for experiment I ($\eta \times \mu$) and $P_{max}(N)$, $S_p(P)$, $A_z(Z)$, and $M_p(D)$ for experiment II ($\eta \times v$). Some parameters were highly sensitive for a narrow range of conditions while other parameters were consistently sensitive over a broad range of treatment combinations. Only the Z_{max} , A_z , R_z , parameters were sensitive for all the state variables and for both experiments.

Interactive and Non-interactive parameters: The overall sensitivity patterns described in Table 4 were expanded upon in Figure 2.5 and 2.6 to illustrate the full range of sensitivity dynamics and to highlight interactive and non-interactive parameters. From visual inspection of Figures 2.5 and 2.6, it can be seen that most parameters exhibit some degree of interaction effect due to the simultaneous variation in the drivers for the two factorial experiments (i.e., dynamics changed due to variations in one or both of the drivers). For example, the $N_d(N)$ parameter (Figure 2.6, C) was unresponsive to the amount of fish predation but highly responsive to the level of nutrient inputs. The $K_c(D)$ parameter (Figure 2.6, K) was also unresponsive to the fish predation level but showed a threshold effect to nutrient inputs. The $A_z(Z)$ parameter had an opposite response in that fish predation had a strong impact on sensitivity dynamics while, in many cases, changes in nutrient inputs had no impact (Figure 2.6, H). Some parameters also had interactive effects to both treatment factors such as the $P_{max}(N)$ and $Z_{max}(P)$ parameters (Figure 2.5, B and D).

The degree of interaction observed between treatment factors for each factorial experiment was determined through frequency and Chi-Square analysis based on a

probability function (see Appendix C for additional information) the results of which are shown in Table 2.5. A higher percentage indicates less interaction and consistency of results across that treatment series and a low probability signifies a significant interaction. Those parameters which failed the χ^2 test at the ($P = 0.01$) level, indicating a low level of predictability and a significant interaction effect, are highlighted in bold.

The analysis showed that many of the parameters could be predicted based on the sensitivity results observed in the control series. This was especially true for changes in the water exchange rate (μ) and fish predation rate (ν) which demonstrated no significant interactions except for the M_z (Z) parameter. Sensitivity dynamics (i.e., sensitive or non-sensitive) stayed consistent across the series allowing prediction based on only the dynamics seen in the control case. In contrast, changes in the nutrient input rate (η) resulted in significant interactions in many of the cases (e.g., P_{max} , K_c , K_z), which will complicate efforts to predict the sensitivity dynamics based on only the control cases. The predictability of parameters to changes in η , were sometimes impacted by the other factor (μ or ν) and reflects the slightly different baseline conditions between the two factorial experiments.

2.3.4 Dominant model parameters

Another way to examine trends in parameter sensitivity due to variations in the external drivers is to identify and examine the interactions between the most sensitive parameters for each treatment combination. An analysis of this type highlights which parameter, and its associated process, dominates for a given set of conditions and how parameters change importance under the changing environmental conditions simulated in

the model. These often complicated, internal model dynamics are described below for each of the external drivers and are illustrated in Table 2.6.

Nutrient input effects: The amount of nutrients entering the model system caused dramatic shifts in dominant parameters which give insight into the internal model dynamics. Under the low nutrient input conditions ($\eta = 0.0-0.04 \text{ gN m}^{-3} \text{ day}^{-1}$) N availability is mainly controlled through regenerative processes and diffusive exchange across the thermocline. Under these conditions only low biomass levels of P and Z were supported and the parameter sensitivities reflect an ecosystem that is nutrient and grazer limited as indicated by the importance of the nutrient (N_o) and grazer (Z_{max} , A_z) related processes. N and P were both controlled by Z_{max} , either through direct grazing or the regeneration of nutrients from grazing processes, while the Z state variable is mainly influenced by internal growth processes as reflected by the sensitivity to the feeding efficiency (A_z). The low overall biomass in the ecosystem leaves the D compartment starved for biomass input and thus controlled by Z grazing (Z_{max}) and the supply of nutrients to the system (N_o).

Intermediate levels of η ($0.08-0.16 \text{ gN m}^{-3} \text{ day}^{-1}$) creates a more productive ecosystem that is less impacted by Z predation and input of N from below the thermocline. The increased biomass of P leads to an increased demand on nutrients increasing model sensitivity to the K_n and S_p parameters which regulate how rapidly nutrients are removed from the water column. The K_c and K_z parameters also become important as P growth starts to become limited due to self-shading effects. The D state variable is now sensitive to the input of P biomass (K_c) rather than the input from Z biomass (Z_{max}).

Table 2.6. Parameter ranking as a function of treatment combination for each factorial experiment and state variable. Only the most sensitive parameter for a given treatment combination is shown. Additional information on selected cases are discussed further in the text.

State Variable η (gN/m ³ /day)	A. $\eta \times \mu$						B. $\eta \times v$			
	μ (% loss/day)						v (% loss/day)			
	0	1.25	2.5	5	10	20	0	20	40	80
N										
0.0	Z_{max}	Z_{max}	A_z	A_z	P_{max}	P_{max}	Z_{max}	Z_{max}	K_n	P_{max}
0.04	Z_{max}	Z_{max}	Z_{max}	Z_{max}	Z_{max}	P_{max}	Z_{max}	Z_{max}	K_n	P_{max}
0.08	K_n	K_n	K_n	Z_{max}	Z_{max}	K_n	K_n	K_n	K_n	P_{max}
0.16	K_z	K_z	K_z	K_n	P_{max}	K_n	K_z	K_z	K_z	K_z
0.32	N_d	K_z	K_z	K_z	K_z	K_z	N_d	K_z	N_d	K_z
0.64	N_d	N_d	K_z	K_z	K_z	K_z	N_d	N_d	N_d	N_d
P										
0.0	Z_{max}	Z_{max}	Z_{max}	Z_{max}	N_d	N_d	Z_{max}	Z_{max}	N_o	N_o
0.04	Z_{max}	Z_{max}	Z_{max}	Z_{max}	Z_{max}	N_d	S_p	Z_{max}	A_z	S_p
0.08	S_p	S_p	Z_{max}	Z_{max}	Z_{max}	Z_{max}	S_p	S_p	S_p	S_p
0.16	K_z	K_z	K_z	S_p	Z_{max}	Z_{max}	K_z	K_z	K_z	K_z
0.32	K_z	K_z	K_z	K_z	K_z	K_z	K_z	K_z	K_z	K_z
0.64	K_z	K_z	K_z	K_z	K_z	K_z	K_z	K_z	K_z	K_z
Z										
0.0	A_z	A_z	A_z	A_z	A_z	A_z	A_z	A_z	A_z	R_z
0.04	A_z	A_z	A_z	A_z	A_z	A_z	A_z	A_z	A_z	R_z
0.08	A_z	A_z	A_z	A_z	A_z	A_z	A_z	A_z	A_z	R_z
0.16	A_z	A_z	A_z	A_z	A_z	A_z	A_z	A_z	A_z	R_z
0.32	A_z	A_z	A_z	A_z	A_z	A_z	A_z	A_z	A_z	R_z
0.64	A_z	A_z	A_z	A_z	A_z	A_z	A_z	A_z	A_z	R_z
D										
0.0	N_o	N_o	R_z	A_z	N_o	N_o	P_{pref}	R_z	M_p	M_p
0.04	Z_{max}	Z_{max}	Z_{max}	Z_{max}	Z_{max}	M_p	Z_{max}	Z_{max}	Z_{max}	M_p
0.08	Z_{max}	Z_{max}	Z_{max}	Z_{max}	Z_{max}	A_z	Z_{max}	Z_{max}	S_p	M_p
0.16	K_c	K_c	K_c	Z_{max}	Z_{max}	A_z	K_c	K_c	K_z	K_z
0.32	K_c	K_c	K_c	K_c	K_c	K_c	K_c	K_c	K_z	K_z
0.64	K_c	K_c	K_c	K_c	K_c	K_c	K_c	K_c	K_z	K_z

At high levels of η (0.32-0.64 gN m⁻³ day⁻¹) input a number of significant changes occur. P biomass is now limited by self-shading effects and, despite further nutrient additions, is unable to increase in biomass. Nutrients build-up in the ecosystem and the dynamics shift from nutrient-limitation to light-limitation. The most sensitive parameters for the P and D state variables are now the depth of the water column, K_z and the light attenuation coefficient for phytoplankton self-shading, K_c as these parameters control how many phytoplankton can be produced at a given light intensity. Since zooplankton were unable to control the phytoplankton due to saturated feeding at these high P biomass levels, Z is still mainly controlled by internal processes (A_z). Due to the build-up of nutrients, the N state variable is now sensitive to the flux of nutrients across the thermocline (N_d).

Water exchange effects: The changes in parameter sensitivities due to increases in the water exchange rate were primarily dependent on the nutrient status of the ecosystem. Under the low nutrient input cases ($\eta = 0.0$ -0.04 gN m⁻³ day⁻¹), the biomass in all the state variables was reduced by the increased flushing due to the higher water exchange rate but the largest impact seemed to result from the decrease in Z predation from the loss of zooplankton from the system. The reduction in Z grazing shifted both the P and N dominant process from Z_{max} to N_d and P_{max} respectively as these state variables became controlled by the supply of nutrient rather than predation and regenerated material. The Z biomass, under most conditions, was dominated by the zooplankton feeding efficiency, A_z , while the sensitivity of the D parameters were more complicated due to interactions between the input of new material from physical (N_o) and biological (R_z, A_z, Z_{max}, M_p) sources.

With the addition of more nutrients ($\eta = 0.08-0.16 \text{ gN m}^{-3} \text{ day}^{-1}$) there is now greater biomass and higher P and Z growth rates. As the water exchange rate is increased, N goes from control by phytoplankton uptake (K_n), to zooplankton control (Z_{max}) and then back to phytoplankton control (K_n). P goes from phytoplankton (S_p) control back to zooplankton control (Z_{max}) whereas D is controlled mainly by zooplankton process (Z_{max}, A_z) across the range of water exchange rates. The overall impact of the higher exchange rate was to remove nutrients and biomass from the system which shifted the model from the early stages of light-limitation and P control into a more nutrient-limited state controlled mainly by Z processes.

At higher nutrient addition rates ($\eta = 0.32-0.64 \text{ gN m}^{-3} \text{ day}^{-1}$) the effects of the water exchange rate become masked by excess biomass in the system and the dominance of the P parameters. The N state variable was controlled by the thermocline exchange rate, N_d , when the water exchange rate was low. The K_z parameter then takes over at higher water exchange rates. With such high nutrient input rates there were always excess nutrients resulting in a physically driven system, first by internal inputs and then by external losses as the exchange rate increases. P is controlled directly through the depth parameter, K_z , whereas D is controlled indirectly through the self-shading coefficient for the phytoplankton, K_c .

Fish predation effects: The impact of changing the fish predation rate was largely determined by the level of nutrients entering the system which in turn affected the level of phytoplankton biomass available to support zooplankton production. With the baseline case ($\eta = 0.00 \text{ gN m}^{-3} \text{ day}^{-1}$) there was no fish predation and no external nutrient inputs. Under these conditions, Z can have their greatest impact on the other state

variables. As a result, the dynamics were dominated by the Z parameters Z_{max} , A_z , and P_{pref} . When the fish predation rate was increased the system went from being Z dominated to N and P dominated. N became controlled by P growth (K_n , P_{max}) while P were limited by the amount of N below the thermocline, N_o . D was dominated by inputs from P losses (M_p) instead of Z related grazing and growth terms (P_{pref} , R_z). The zooplankton go from being internally regulated by their feeding efficiency (A_z) to regulation by losses due to respiration, R_z .

When nutrients were increased ($\eta = 0.16 \text{ gN m}^{-3} \text{ day}^{-1}$) the P , N , and D state variables were no longer controlled by Z even under the low fish predation treatments. Here the P and N biomass values were largely controlled by the K_z parameter indicating that the system is light-limited. The D state variable is mainly controlled by the K_c and K_z parameters which limit the potential amount of P biomass available as input to the D state variable.

At higher nutrient addition rates ($\eta = 0.64 \text{ gN m}^{-3} \text{ day}^{-1}$) and for all predation rates considered, the N state variable was controlled by the diffusion rate across the thermocline, N_d , as opposed to the depth of the water column, K_z . The excess in N , along with P growth limitation from self-shading, effectively de-coupled N from P control. The dominant processes for the P , Z and D state variables remain unchanged from the previous nutrient input series.

2.4 Discussion

The identification of parameters and processes that most affect model predictions is a key step in the development, calibration, and testing of any model (Gardner et al.

1981; Oreskes et al. 1994; Caswell 2000). Because parameters may be poorly known or may affect feedbacks and non-linear interactions, cursory inspection of model equations will not provide useful insight for determining parameter-prediction dependencies. Sensitivity analysis, through the systematic variation of input parameters and comparison with corresponding predictions, is a critical tool that should be universally employed during all stages of model development, testing, and application.

Useful ecosystem models must be capable of considering a broad range of environmental drivers and internal conditions. Consequently, a sensitivity analysis should be conducted over the widest practical range of conditions. For pelagic ecosystem models, variations in predator/prey dynamics, nutrient pulses, and water exchange rate are all important processes to consider. The wide variability of parameter values reported in the literature for these models also requires an equally extensive analysis. Parameters that are insensitive under a wide range of conditions are unlikely to be a concern for model development and application (Gardner et al. 1982). However, high uncertainty combined with high sensitivity will identify parameters of concern (Gardner and O'Neill 1983; Gardner et al. 1990). Model development and testing must focus on these parameters and conditions in order to improve model precision and the reliability of predictions. The results reported here illustrate the importance of these analyses for the NPZD model.

2.4.1 Sensitivity Dynamics

The sensitivity analysis of the NPZD model for each treatment combination showed that only three to five of 28 parameters were important for any fixed set of

conditions. When results were combined over all scenarios a much larger parameter set (i.e., 17 parameters) was found to be sensitive for at least one set of conditions. The most sensitive parameters for each state variable and experiment were the P_{max} (N , Exp I, II), A_z (Z , Exp I, II), Z_{max} (P and D , Exp I), S_p (P , Exp II), and M_p (D , Exp II) parameters (see Table 4). Had I limited the analysis to only one set of conditions, as is sometimes done in sensitivity analysis, a number of potentially important parameters would have been overlooked.

Most parameters were sensitive over a narrow set of conditions. Only the zooplankton maximum grazing rate, Z_{max} , and assimilation efficiency, A_z , parameters were sensitive over most of the treatment combinations tested. Increased precision in the estimates of these two parameters will most improve the precision of model predictions for the model. Other broadly sensitive parameters were the N_d , P_{max} , K_z , and K_c parameters. These parameters were sensitive for many cases in the N , P , and D state variables, playing a major role in the dynamics of the NPZD model.

The R_f , I_n , I_o , K_w , k , S_d , and F_d parameters were never sensitive despite the broad range of environmental conditions considered. Under the conditions tested, parameter values, and model structure, these parameters had little influence on the model dynamics. To reduce model complexity, unimportant parameters, such as those above, could be eliminated or combined with other parameters and processes. For the remaining parameters, the sensitivity results were variable depending on the conditions being simulated and the state variable being considered.

2.4.2 Response to changing environmental conditions

The changing environmental conditions simulated (i.e., increases in productivity from higher nutrient inputs, greater top-down control from increases in fish predation and higher mixing losses from an increased water exchange rate) should impact the model and the parameter sensitivities in different ways. I anticipated finding different parameter groupings that would change depending on the treatment series but the actual trends observed were very complicated with few patterns. Only key parameters were able to be identified for each treatment series and state variable. These key parameters were: (1) the N_d (for N), S_p (for P), A_z (for Z), and Z_{max} (for D) parameters for the nutrient input series; (2) P_{max} (for N), Z_{max} (for P), and A_z (for Z, D) parameters for the water exchange rate series; and (3) the Z_{max} (for N, P), A_z (for Z), and N_o (for D) parameters for the fish predation series. As before, the Z_{max} and A_z parameters dominated the sensitivity dynamics.

The factorial design allowed me to examine interactions between the simultaneous variation of the nutrient input rate with each of the other two drivers. In the absence of significant interactions, simple main effects (i.e., the edges of the factorial) would be adequate for predicting if a parameter would remain sensitive or insensitive. Many parameters within the water exchange rate and fish predation rate treatment series met these conditions (see Table 2.5). However, the nutrient input rate parameters displayed significant interactions due to a pronounced threshold effect. These results suggest that my NPZD model will be more predictable to changes in the water exchange and fish predation rate than changes in the nutrient input rate under this particular set of parameter values.

The factorial design also provided valuable insight into the internal dynamics of my model. Through the identification of the most sensitive parameter for each treatment combination the interactions between parameters was exposed. For instance, over the course of the nutrient input treatment series, the dominant parameters shifted from those associated with nutrient limitation and biological control (i.e., Z_{max} , K_n , S_p) under low nutrient conditions to those associated with light limitation and physical control (i.e., K_z , K_c) under high nutrient conditions. This was due to the increase in phytoplankton biomass and nutrients combined with the inability of the zooplankton to regulate the phytoplankton at the higher nutrient input rates. Section 3.4 describes additional examples which reveal interesting internal model dynamics.

2.4.3 State variable effects on sensitivity results

The level of sensitivity for most parameters was dependent on the state variable (i.e., N , P , Z , or D) being considered. High sensitivities usually occurred when a parameter was present in the equation for that state variable. Some parameters were only sensitive for one or two state variables (e.g., K_n , S_p , M_z , M_p). Parameters with limited sensitivity can provide a powerful tool for understanding the complex interactions between state variables. For example, why was the natural mortality rate for phytoplankton, M_p , only sensitive in the D compartment while the sinking rate for phytoplankton, S_p , was sensitive in both the P and D state variables? Both parameters have the same value (i.e., 10% loss day⁻¹) and appear in the same equation (i.e., P) with the same structure (i.e., first order loss term) yet have different sensitivity dynamics. A probable reason for the sensitivity of the S_p parameter in the D state variable is that under

low nutrient conditions the D compartment is “starved” for material and any process that affects how much new material is available for recycling is critical. The dynamics related to the M_p parameter occur because this parameter is a direct input to the D compartment while for P , it occurs as one of several loss terms (Eq. 1).

2.4.4 Model structure and form effects on sensitivity results

Choices in model structure and form play a significant role in determining model sensitivities. For several parameters (e.g., K_z) there was a threshold effect with increasing nutrient input. This sudden change in sensitivity was associated with an equally sudden switch from nutrient-limitation to light-limitation as represented in the photosynthesis equation (see Eq. 1). In this equation, phytoplankton growth was limited by the minimum value from either the nutrient or light function as opposed to a multiplicative function where both processes can simultaneously limit phytoplankton growth. The formulation employed in Eq. 1 has the effect of removing the non-limiting process from the model dynamics (i.e., the model no longer sees that part of the equation). Both functions are used in the literature (Cullen et al. 1993; Haney and Jackson 1996; Denman and Pena 1999) so investigation of this and other possible threshold effects (e.g., minimum feeding concentration for the Ivlev function) warrant further attention for their potential impact on parameter sensitivity and model dynamics.

Other effects of model structure on parameter sensitivity include the types of functional responses employed (e.g., linear, quadratic, saturating), number of feedbacks, and types of processes modeled. For example, this model does not include temperature effects which are often parameterized with a Q^{10} formulation. Other studies have found

this parameter to be highly important (e.g., Bartell et al. 1988b; Pastres et al. 1997) and had I included a temperature related parameter I probably would have found similar results. Another example includes the type of functional response utilized. My simulations would be quite different if I had formulated the phytoplankton so they were not limited by self-shading effects as has been done in other models (e.g., Franks and Chen 1996; Kemp et al. 2001). The plateau seen in many of the parameter sensitivities at the higher nutrient concentrations would probably not exist.

2.4.5 Comparison of sensitivity results between studies

The comparison of the results of sensitivity analyses for different studies can be problematic. Even though the ecosystem is the same (i.e., a pelagic ecosystem) the model formulations may differ in scope and detail; parameter values vary depending on the location and species simulated; and the technique for performing the sensitivity analysis is often different. Several studies have addressed this latter problem by comparing the performance of different sensitivity techniques (Hamby 1995; Homma and Saltelli 1996). Although no standardized method has emerged, a range of methods appear to give similar results (Hamby 1995).

Comparisons between studies are also difficult when parameter sensitivities are analyzed for transient rather than equilibrium conditions. Although analysis of transients are important, cross-model comparisons are complicated by the dynamic changes in parameter sensitivities that may occur over a seasonal cycle or from changes in food-web structure (e.g., Bartell et al. 1988b; Fong et al. 1997; Klepper 1997; Pastres et al. 1997). An additional difficulty with comparing sensitivity studies is that the response variable

for the sensitivity analysis often differs. I was interested in parameter sensitivities relative to the equilibrium biomass concentration. Other studies have also used this response variable (e.g., Bax 1985; Fasham 1995; Druon and Fevre 1999), but alternative endpoints are possible including carbon export dynamics, primary and secondary production rates, and patterns of phytoplankton growth and decline (e.g., Bartell et al. 1988a; Fasham et al. 1990; McGillicuddy et al. 1995).

Despite these difficulties, several generalizations are possible. I showed the parameters affecting Z also affected the equilibrium values for all the state variables under a wide range of conditions. Many studies have also found that the dynamics of Z play a major role in the flow of energy directly via predation or indirectly through excretion and fecal pellet production (McGillicuddy et al. 1995; Frost 1997; Edwards et al. 2000; Edwards 2001; Halvorsen et al. 2001). A surprising result was that the quadratic mortality term for the zooplankton was not sensitive for a majority of the cases. Other studies (Steele and Henderson 1992a; Edwards and Brindley 1999; Murray and Parslow 1999; Kemp et al. 2001) have found the mortality term to be important in controlling the dynamics of these simple ecosystem models. One possible explanation is the population levels for Z never reached a high enough value to overcome the effects of the Z growth rate and assimilation efficiency. This is further supported by the fact that the only time the quadratic mortality term was important was when the fish predation rate was low allowing the Z biomass to achieve their highest values. If the quadratic mortality term was parameterized with a lower value (i.e., weaker control of biomass) or if there were conditions where the Z could achieve higher population levels (i.e., higher assimilation efficiency, lower losses from excretion, lower grazing saturation) then a

more prominent role should be expected for the mortality term.

Some parameters, like the half-saturation constant for Z grazing, k , and the parameters related to photosynthesis (e.g. I_o , I_n), were never sensitive although other studies have found them to be important (e.g., Druon and Fevre 1999). A possible reason for my results was that the light intensity for my simulations was greater than the half-saturation constant for light extinction. Under saturated light conditions small variations in the half-saturation constant will have no measurable effect on model output. The fluctuations in P biomass caused by the treatment combinations did cause the self-shading parameter to be important under the high biomass cases. If simulations would have been run over the course of a year, with variations in light intensity, then the half-saturation constant for light uptake most likely would have been important especially during times of low light intensity. The same can be said for the Z half-saturation constant for grazing, since for a majority of the cases the P equilibrium value was much higher than the half-saturation constant value. These examples all highlight the need to use caution when interpreting results of sensitivity experiments and especially when comparing between models with different parameter values, structures and endpoints.

2.5 Conclusions

Sensitivity analysis (in the strictest sense) is the partial derivative of the state variable with respect to the parameter (Tomovic 1963). Whether this derivative is estimated analytically or numerically, it represents a single point within the larger domain of all possible parameter-prediction relationships. Consequently, I used sets of factorial experiments that varied model conditions to estimate parameter-prediction relationships

across a broad range of conditions.

The results of these experiments have shown that: 1) only two to three parameters are usually required to “explain” predictions if environmental conditions (i.e., model drivers) are well-defined; 2) although many different parameters may be sensitive when the domain of possible environmental conditions are explored, only a few parameters were consistently important across all experiments; and 3) interactive effects among parameters were revealed by the systematic variation of environmental conditions.

The information provided by a comprehensive sensitivity analysis is also useful for the analysis of model structure. Shifts in parameter sensitivity, including threshold effects, indicate an associated shift in process-prediction dependencies. Dominant parameters under a particular set of conditions are often not applicable to a new set of conditions. Such shifts may be easily verified by empirical observations to verify the validity of a particular model formulation. The failure to obtain empirical verification may mean that alternative model formulations are warranted.

The analysis of parameter sensitivities can also be used to reveal where model simplification may be possible. Parameters, and associated processes, that are never important may indicate unneeded model complexity. Alternative functional forms with fewer parameters may be substituted without loss of predictive power (Gardner et al. 1982). The tedious process of model calibration should focus on sensitive parameters, but with the caveat that such calibration may not be suitable if environmental conditions (and associated parameters sensitivities) are significantly altered.

Based on the analysis reported here, the newly developed NPZD model behaves in a manner consistent with what would be expected as a result of changes in nutrient

additions, predation pressure, and water exchange rate. Enrichment with nutrients causes a bloom in P biomass and subsequent increases in Z and D biomass. An increase in fish predation results in lower Z and D biomass and in higher P while an increase in the water exchange rate reduces the overall concentration levels of all the state variables, except N which tended to increase (see Figure 2.3). The most sensitive parameters have been identified and their range in sensitivities characterized. The results also provide insight into the internal dynamics of the model for this particular set of parameter values and experimental conditions (see discussion in section 2.3.4).

Based on this knowledge, a number of possible future directions are possible. The results from this analysis are primarily limited the particular values used for each of the model parameters. These parameters can however vary over a wide range of values. One potential way to extend the applicability of my results is to conduct an uncertainty analysis to understand how parameter variability changes when the parameters are varied over a realistic range. Other papers have used this technique to gain additional insights into model dynamics (Gardner and O'Neill 1983; Dale et al. 1988; Gardner et al 1990; Rose et al. 1991).

I can use the sensitivity results to make more efficient future efforts at model development, calibration and testing. Some of the identified non-sensitive parameters could be combined with other parameters or processes to remove unnecessary parameters to reduce computation time and errors due to parameter estimation. Efforts at calibration will be facilitated by knowing which parameters are likely to cause a change in a particular state variable with only a minimal change in the parameter value (i.e., the sensitive parameters). Many of the parameters used in my model have a wide range of

possible values reported in the literature. Parameters which are highly variable and also highly sensitive need careful attention since errors in the estimation of these parameters can drastically impact the observed model dynamics.

Through the sensitivity analysis, I discovered that the top-down control by zooplankton is rather weak in my model, allowing phytoplankton to increase rapidly with only a minimal amount of nutrient addition. Stronger top-down control by the zooplankton was prevented by the low growth rate of the zooplankton (reflected in the Z_{max} parameter) combined with high losses from respiration (R_z parameter) and self-regulation due to the quadratic mortality term (M_z parameter). If I wish to simulate a system where there is increased coupling between phytoplankton and zooplankton I will have to alter some of these terms so that the zooplankton can achieve higher biomass concentrations. Additional changes are possible, based on insights gained from the sensitivity analysis, but these will be largely driven by the focus of future experiments.

In summary, sensitivity analysis should be performed during all stages of model formulation, development and application (Gardner et al. 1981; Caswell 2000). The existence of efficient Monte Carlo techniques allows this analysis to be efficiently explored for most ecosystem models. Sensitivity analysis enables the identification of the most sensitive parameters for a particular parameter set or environmental conditions. Through my analysis I was able to investigate parameter behavior under a wide range of environmental conditions which should aid in the extrapolation of the model dynamics and behavior to other conditions outside of the experimental domain. The volume of data produced however, continues to be a bottle-neck, making the analysis of results awkward and difficult to explore. Opportunities for the application of advanced analysis

techniques (e.g., Beres and Hawkins 2001) should be explored to make the benefits of sensitivity analysis universally available.

Chapter 3

A novel grid-based method for simulating idealized turbulence in aquatic systems

Abstract

Algorithms based on Fickian diffusion are often used to represent or parameterize turbulent mixing in spatially-explicit ecosystem models. These methods, however, suffer from a number of limitations including the accelerated smoothing/elimination of existing concentration gradients and an inability to represent the heterogeneity generated by scale-dependent mixing. I developed a computationally efficient and statistically accurate method for simulating idealized two-dimensional turbulence in aquatic systems which avoids these problems. Based on a “seeded eddy” model developed for particle systems, I have adapted the technique for a gridded framework while still preserving the cascade of turbulent energy from broad- to fine-scales. Results from tracer decay studies showed that this method reproduces the velocity spectrum of homogeneous isotropic turbulence and accurately represents the spreading rate of particles from scale-dependent turbulent mixing. The technique is extremely fast and preserves concentration gradients down to the resolution of the simulation domain thereby allowing the simulation of realistic concentration gradients over a broad range of scales. These attributes make the method ideal for incorporation into spatially-explicit ecosystem models for the theoretical and applied investigation of physical-biological interactions over a range of scales, especially where the preservation of spatial patterns is important.

3.1 Introduction

Turbulence plays a prominent role in structuring physical and biological interactions in aquatic ecosystems. Physical processes are affected through the alteration of mixing, flushing, and mass transfer rates (Gargett 1989), the breakdown of large-scale concentration gradients (Okubo 1980), the transfer of variability from large to small scales (Ottino 1990), and the regulation of exchanges between interfaces such as the air-sea, pycnocline, and benthic boundary layers (Hopfinger 1987). For organisms in the aquatic environment, turbulence affects a wide range of ecological processes at scales of individuals, populations, communities, and ecosystems. Some of these include; feeding rates (Kiorboe and Saiz 1995), uptake kinetics (Lazier and Mann 1989), predator-prey interactions (Hwang et al. 1994), encounter rates (Rothschild and Osborn 1988), aggregation and disaggregation processes (Squires and Yamazaki 1995), species selection (Margalef 1978), and ecosystem productivity (Nixon 1988). Together, these turbulence-mediated, physical and biological processes interact in complex ways to influence resource availability (e.g., nutrients), distribution of physical (e.g., temperature) and biological (e.g., phytoplankton) properties and ultimately ecosystem structure (Mackas et al. 1985; Weber et al. 1986; Powell 1989; Kiorboe 1993).

Turbulence in the aquatic environment is the result of unstable flows from forces moving water over large domains (e.g., tides, currents, upwelling, etc.). These flows create a cascade of energy and variance from large-scale structures down to ever-decreasing scales (Richardson 1922; Kolmogorov 1941). Large scale swirls and eddies break down into smaller swirls and eddies and eventually into random chaotic motions. Finally, at the finest scales (i.e., centimeter scale), the turbulent energy is dissipated into

heat (Batchelor 1967). The large eddies are responsible for a majority of the stirring associated with turbulent mixing and are often inhomogeneous and anisotropic (Garrett 1989). As eddies breakdown into smaller structures, this cascade of turbulent energy becomes more spatially uniform and independent of orientation (Richardson 1922). The behavior of the turbulent mixing and associated eddies can then be statistically described by the idealized theory of homogeneous, isotropic turbulence (Batchelor 1967).

Numerous comprehensive reviews of aquatic turbulence and associated parameters are available (see especially Tennekes and Lumley 1972; Landahl and Mollo-Christensen 1986; Hopfinger 1987; Yamazaki and Osborn 1988; Gargett 1989; Ottino 1990; Sanford 1997).

The dominant role that turbulence often plays in the physical environment has led to substantial research to incorporate turbulence, and the mixing it facilitates, into aquatic ecosystem models. The chaotic nature of turbulent flows makes it difficult to simulate directly (Tennekes and Lumley 1972; Yamazaki et al. 1991). Often it is easier to simulate turbulence with an appropriate statistical approximation (Holloway 2004). For example, rather than solve the non-linear terms of the Navier-Stokes equations, the turbulence may be parameterized as a Fickian diffusion or random walk process (Okubo 1980; Steele and Henderson 1992; Visser 1997). An alternative approach is to represent the flow directly rather than simulate the associated physics. For instance, prescribed flow fields or multifractal maps may be used to characterize a fully developed turbulent flow field (Dyke and Roberston 1985; Powell and Okubo 1994; Abraham 1998; Marguerit et al. 1998; Mariani et al. 2005). Since each method has its limitations, the use of one method over another will depend on the application of the model, the desired level

of accuracy, the computational costs, the type of simulation method (e.g., Eulerian, Lagrangian), and the temporal and spatial scale of the model domain, process, or species being studied (Chen and Annan 2000; James 2002; Umlauf and Burchard 2005).

Fickian diffusion methods are often used to parameterize turbulent mixing (i.e., turbulent diffusion) in spatially-explicit aquatic ecosystem models. Originally developed to describe molecular diffusive processes, Fickian diffusion methods assume that turbulence can be modeled as a diffusive random process with the rate of the mixing governed by the strength of the turbulence (Okubo 1980). The approach has been very successful in addressing a number of important questions related to the impact of mixing processes on ecosystem dynamics and in understanding observed spatial patterns in aquatic systems. Some of these include analyzing factors that regulate phytoplankton blooms (Kierstead and Slobodkin 1953), interpreting and understanding observed concentration gradients of physical and biological variables in the aquatic environment (Wroblewski and O'Brien 1976; Steele and Henderson 1992), and simulating animal movement processes (Grunbaum 1994). There is also a large body of theory which has been developed based upon Fickian diffusion methods (Okubo 1980) and the approach has been expanded to similar areas such as reaction-diffusion (Gurney et al. 1998), cellular automata (Wolfram 1986), and random walk models (Durrett and Levin 1994).

There are, however, a number of problems with extrapolating the Fickian diffusion approach to the scales at which turbulent processes dominate (i.e., meters to kilometer scales). The primary concern is that the rate of mixing caused by turbulence is dependent on the scale of the problem (i.e., scale-dependent diffusion) and the Fickian diffusion approach assumes the same mixing rate regardless of scale, which can over- or

underestimate the true mixing rates occurring in natural systems (Okubo 1980, Petrovskii 1999). Another issue is that the Fickian diffusion approach models turbulent mixing as a diffusive process which will result in gradients or patterns becoming smeared or dissipated thereby tending to eliminate spatial concentration gradients over time or to prevent the generation of pattern that occurs during turbulent mixing (Abraham 1998).

A novel technique for simulating idealized two-dimensional turbulence in aquatic systems which is free from many of the limitations of traditional Fickian diffusion methods is presented in this chapter. Based on a seeded eddy model developed for Lagrangian particle systems, (Dyke and Robertson 1985; Abraham 1998) I have adapted the technique to work within a gridded Eulerian framework while still preserving the cascade of turbulent energy from broad- to fine-scales. This new method, the Eulerian seeded eddy model (ESEM), is theoretically based, computationally efficient, and statistically accurate. Results from tracer decay studies are used to demonstrate that the ESEM method reproduces the velocity spectrum of homogeneous isotropic turbulence and accurately models the spreading rate of particles from scale-dependent turbulent mixing. Because the ESEM is extremely fast and exhibits no numerical diffusion, this method allows the simulation of realistic concentration gradients over a broad range of scales.

3.2 Methods

3.2.1 Simulation Platform

As part of my dissertation research efforts investigating the use of spatially-explicit simulation models, I developed a spatial lattice framework (SLS) for simulating

spatial effects in aquatic systems (Figure 3.1). My approach has been to embed a fine-scale aquatic ecosystem model that simulates the dynamics of nutrients, phytoplankton, zooplankton, and detritus (see Chapter 1) within a gridded aquatic landscape with exchanges among grid cells controlled by sets of difference equations. The bulk flow of constituents and organisms past a fixed point are recorded, analogous to a Eulerian framework, and rules are used to establish exchange of material, energy, and/or organisms between adjacent grid sites.

The SLS simulation platform was ideal for investigating one of the main focus areas of my dissertation: how processes and patterns translate across spatial and temporal scales and how the spatial distribution of resources and organisms influences these processes and patterns. To study these biological and physical interactions within a spatially explicit framework, I needed a way to simulate realistic cell-to-cell physical exchanges. I also planned to compare the spatial patterns produced by the model simulations to field data that were spatially extensive (30-50 km) but also had fine spatial resolution (30 m). Over these spatial scales, turbulence is a dynamic process with numerous swirls and eddies, so any technique had to be capable of reproducing a range of mixing scenarios typical of the natural environment while also preserving fine-scale concentration gradients.

Since my simulation platform was based on a Eulerian framework, at fine spatial resolutions, I was limited in the types of methods to incorporate realistic turbulent mixing effects. Circulation models for the system of study (i.e., Chesapeake Bay) could not produce realistic turbulent flows and preserve fine-scale pattern (i.e., less than 100 m) at the scales of interest (e.g., Cerco and Cole 1993; Hood et al. 1999). The use of

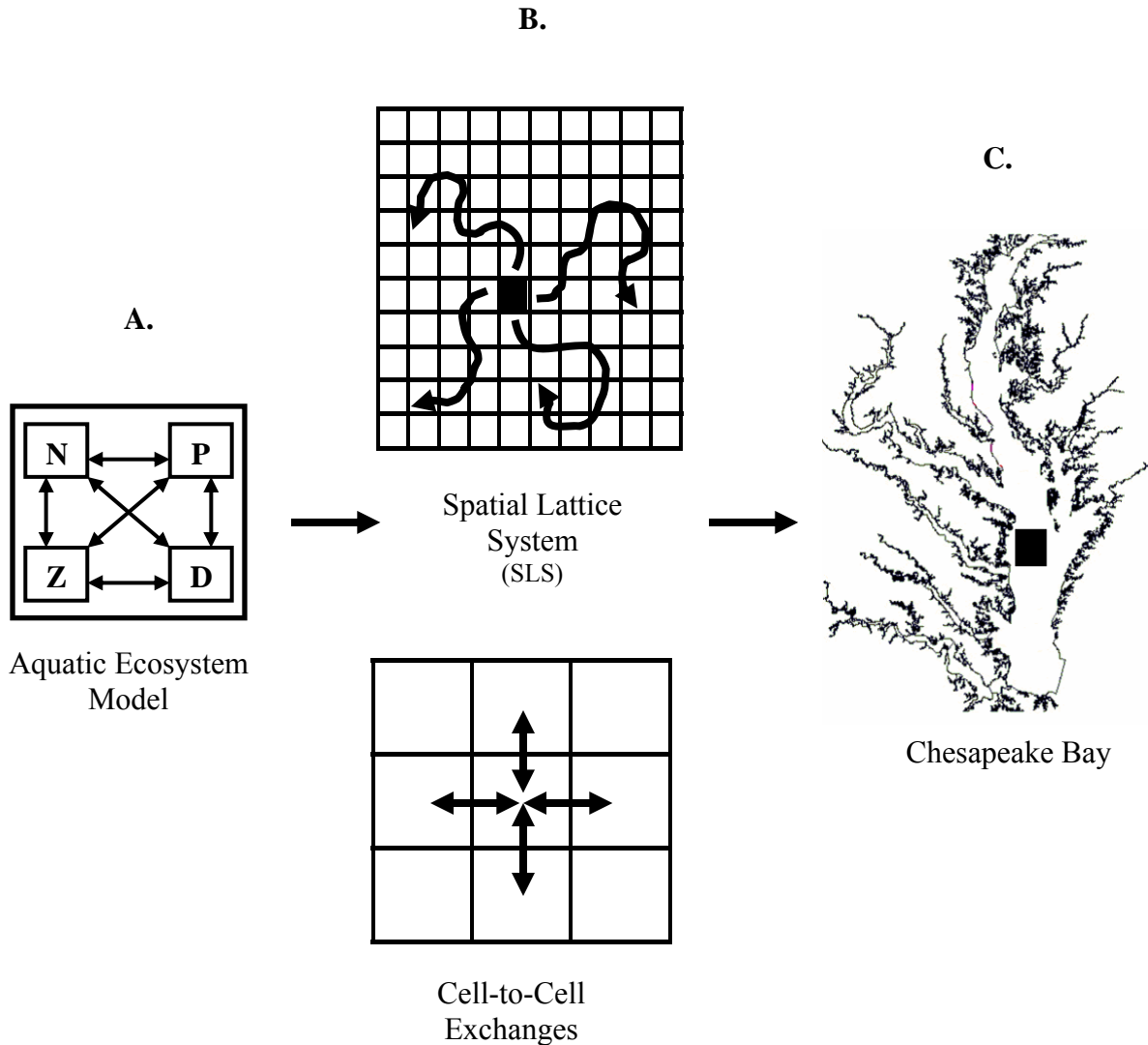


Figure 3.1. Schematic showing the Spatial Lattice System (SLS). The developed system had to be capable of reproducing physical (e.g., turbulence, advection, diffusion) and biological (predator-prey dynamics, fish movement) exchanges typical of the natural environment while handling a range of mixing and or physical forcing scenarios which would allow results to be extrapolated to natural systems such as the Chesapeake Bay. Panel A illustrates the biological component of the SLS which is inserted into each grid-cell of the simulation framework. In future experiments, I plan on using a nutrient-phytoplankton-zooplankton-detritus ecosystem model. Panel B illustrates the physical component of the SLS which consists of routines to approximate diffusive and turbulent mixing. The turbulent mixing routine is the subject of this chapter. Panel C illustrates the application of the SLS to a section of the Chesapeake Bay.

multifractal maps to produce statistically accurate flow fields was possible (Marguerit et al. 1998) but since I wished to study the “dynamic” interaction between turbulent mixing and biological processes and its effect on spatial patterns in aquatic systems, the static nature of this method would not be very satisfactory. Another method would be to parameterize the mixing as a Fickian diffusion process. With this method, fine-grids can be simulated, but because it is a diffusion-based process within a grid-based system, concentration gradients are not accurately represented through time (see Appendix D).

3.2.2 Eulerian seeded eddy model (ESEM)

Because there was no satisfactory method to incorporate realistic mixing effects without resorting to complex three-dimensional hydrodynamic equations or oversimplifications of the physical environment, I developed an approach that used seeded eddies and multifractals to represent turbulent mixing processes in aquatic systems (Abraham 1998; Seuront et al. 1999). These methods are conceptually based on Komologorov’s equilibrium theory of turbulence (Kolmogorov 1941) and have been applied by Dyke in the simulation of offshore turbulent dispersion using seeded eddies (Dyke and Robertson 1985).

The basic concept is that mixing energy is injected at large scales and then cascades to smaller scales until it is eventually dissipated by heat. The input and dissipation of energy is assumed to be in equilibrium and can be viewed as a series of nested eddies. At the largest scales there is a parent eddy that recursively breaks into four daughter eddies that also break up into four smaller eddies until the field is filled with nested eddies of decreasing size (Figure 3.2). The papers by Seuront et al. (1999) and

Lovejoy et al. (2001) provide an overview of these methods and some recent applications.

The seeded eddy model described above, however, was developed for use in particle (i.e., Lagrangian) systems where the location of material is known exactly. To make this method applicable to my system I modified the technique to work within a gridded (i.e., Eulerian) framework where concentration gradients between grid cells must be averaged. My method uses the same basic mechanics used by Abraham (1998) with modifications to allow application within a grid-based system. A schematic of the steps involved with the method is described below and illustrated in Figure 3.3.

First a distribution of eddies is created based on the dimensions of the simulation domain. Since the method uses multifractal generation techniques to create the distribution of eddies, the simulation domain must be a power of two (e.g., 16 x 16 simulation domain equals 2^4). For this chapter, I used a 1024x1024 gridded map which is equal to 2^{10} cells on each side. The exponent, in this case 10, sets the number of map levels (n). The number of eddies at each map level ($E_{(n)}$) is then determined as follows:

$$E_{(n)} = 4^{(n-1)} \tag{Eq. 1}$$

At the largest scale (i.e., map level 1) there would be one “parent” eddy which fills up the entire simulation domain. At the next scale down (i.e., map level 2) there would be four eddies with each eddy taking up one-fourth of the simulation domain. This is repeated until the highest map level is reached. The radius for each eddy at a particular map level ($R_{(n)}$) is determined as follows:

$$R_{(n)} = L/2^n \quad (\text{Eq. 2})$$

Where

L = Length of the simulation domain

At each map level there are four times the number of eddies compared to the previous level and the radius is half of the previous level. To conserve energy as the eddies decrease in size and increase in number, the velocity ($V_{(n)}$) of each eddy ($E_{(n)}$) relative to that of the largest eddy ($E_{(1)}$) is calculated as follows:

$$V_{(n)} = (E_{(1)}/E_{(n)})^{1/3} \quad (\text{Eq. 3})$$

Equation 3 is based on the turbulent cascade of energy in the inertial subrange of turbulence (Kundu 1990; Moum 1996; Sanford 1997). Since the largest eddy is providing all the energy, the smaller eddy velocities are scaled to the velocity of the largest eddy. For example, eddies at map level (3) would have a velocity which was (0.63) that of the eddy at map level (1).

The scaled velocity of each eddy is then converted into a rotation frequency based on the time-step and desired turbulence intensity for a particular scenario. The rotation frequency determines the duration of time, measured in the time-step of the model, between the start of a rotation cycle for eddies of a given radius. For example, eddies

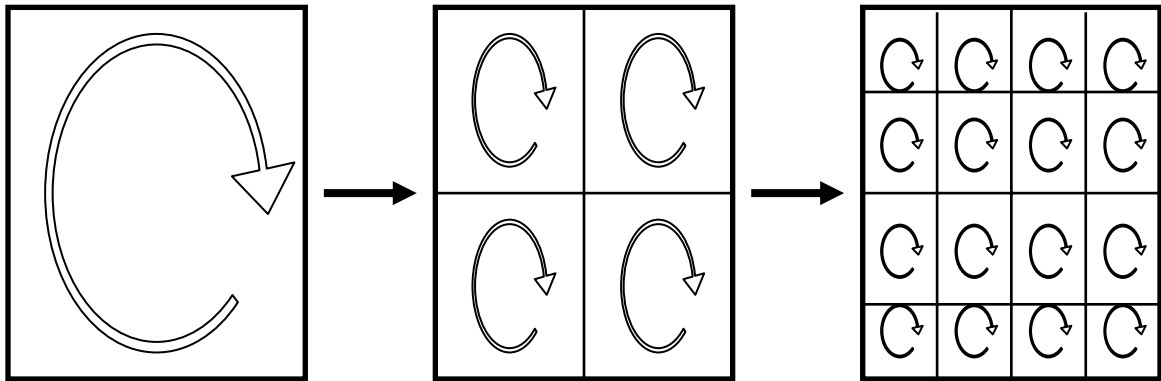
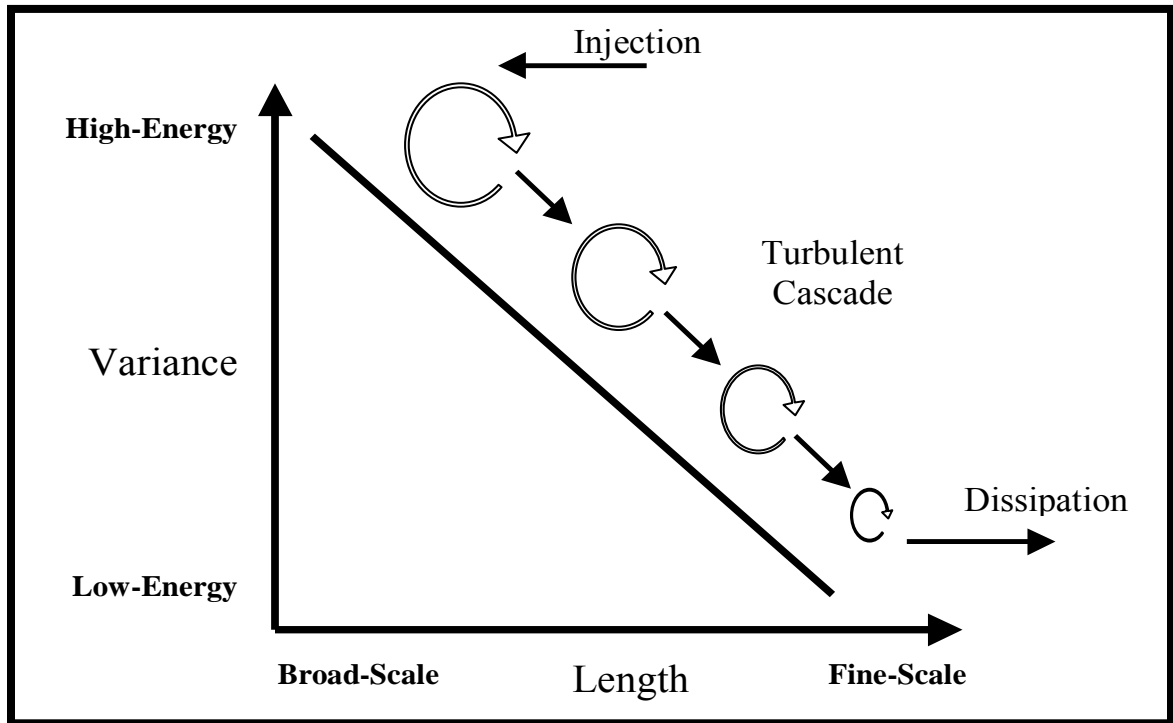


Figure 3.2. Conceptual diagram showing the cascade of turbulent energy from broad- to fine-scales within the multifractal and seeded-eddy framework. At each scale, large eddies break up into space-filling, sub-eddies until eventual dissipation into heat. Diagram is adapted from Seuront et al. (1999).

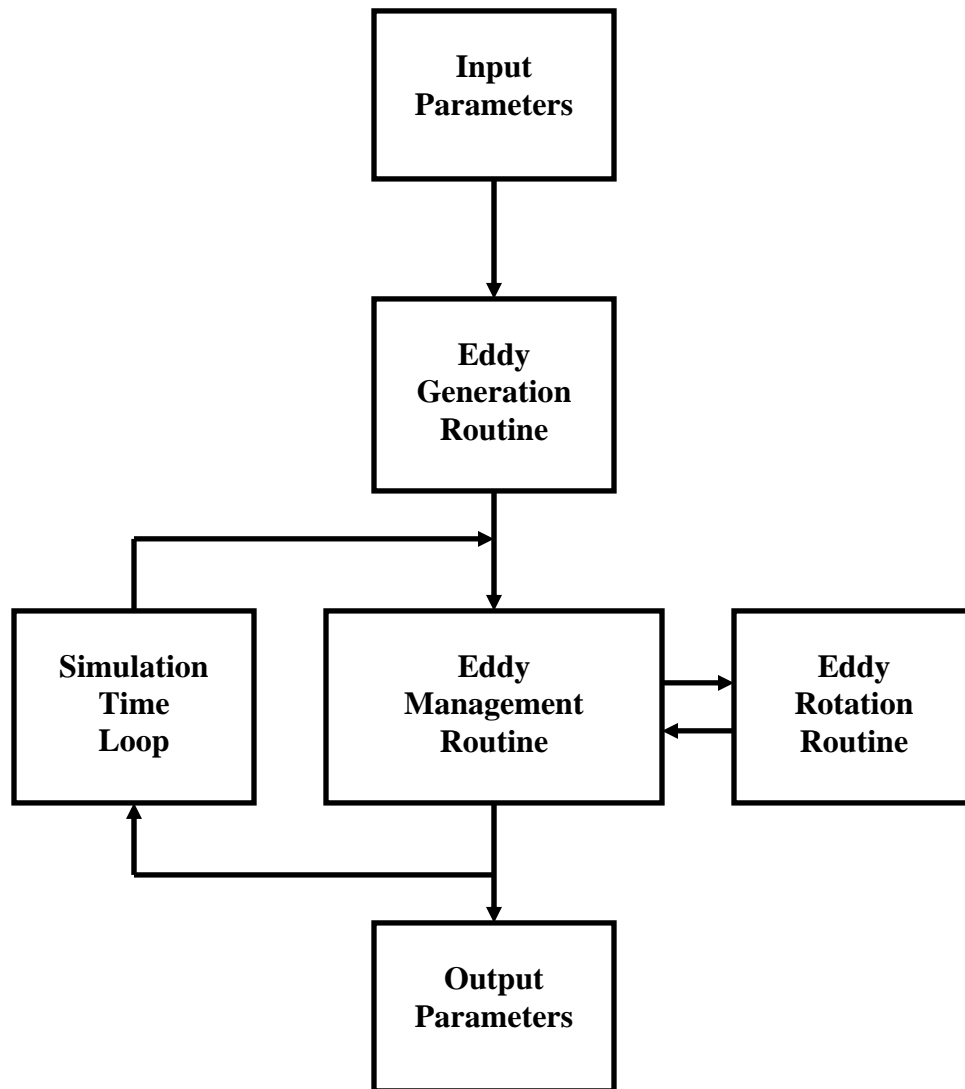


Figure 3.3. Schematic of the steps and routines used to generate turbulent mixing using the turbulence generation method described in this chapter for use in an Eulerian simulation domain. See text for additional details.

with a radius of 128 cells will begin a rotational cycle every 322 time-steps (see also appendix E). Table 1 summarizes the parameters associated with the eddy distribution at each map level. Each eddy is then given a random location on the map and a random spin direction. For a 1024x1024 map, the method will create a distribution of 350,000 eddies of various sizes to be tracked during a particular simulation (Figure 3.4).

The next issue which had to be addressed was how to rotate the eddies within a gridded framework. When a particular eddy rotates a given distance, the outside of the eddy travels farther than the inside. Figure 3.5a illustrates this effect for square eddies within a Eulerian framework. In this example, the eddy has an overall radius of five cells. If the eddy is broken up into five sub-eddies (radius of 1, 2, 3, 4, and 5 cell widths) and rotated one-eighth of a turn, the outer edge (radius) will have traveled five cells and the inner edge will have traveled only one cell distance. A typical solution to account for the changes in rotation speed would be to use different velocities depending on the location within the eddy (i.e., sub-radius five will move five times faster than sub-radius one).

Within my framework, however, material cannot move more than one cell width in a time-step to maintain the conservation of mass. Using the example above, when the outer part of the circle has rotated one cell width, the rest of the circle will have rotated less than one cell width (i.e., some fraction of 100%). Since material within a Eulerian framework is averaged between grid cells, using this approach has the inherent problem of numerical diffusion (see Appendix D). Due to the large number of eddies (i.e., 350,000) and the number of times each eddy rotates during a typical simulation, the amount of error introduced can become quite large. In the initial development of my

Table 3.1. Summary of parameters associated with eddies at a particular map level. The values are based on a simulation domain of 1024x1024 cells. See text for additional details.

Map Level (n)	Eddy Number	Radius (cells)	Scaled Velocity (unitless)	Rotation Frequency (cycles/dt)
1	1	512	1.00	512
2	4	256	0.79	406
3	16	128	0.63	322
4	64	64	0.50	256
5	256	32	0.40	204
6	1,024	16	0.32	163
7	4,096	8	0.25	128
8	16,384	4	0.20	102
9	65,536	2	0.16	81
10	262,144	1	0.13	66

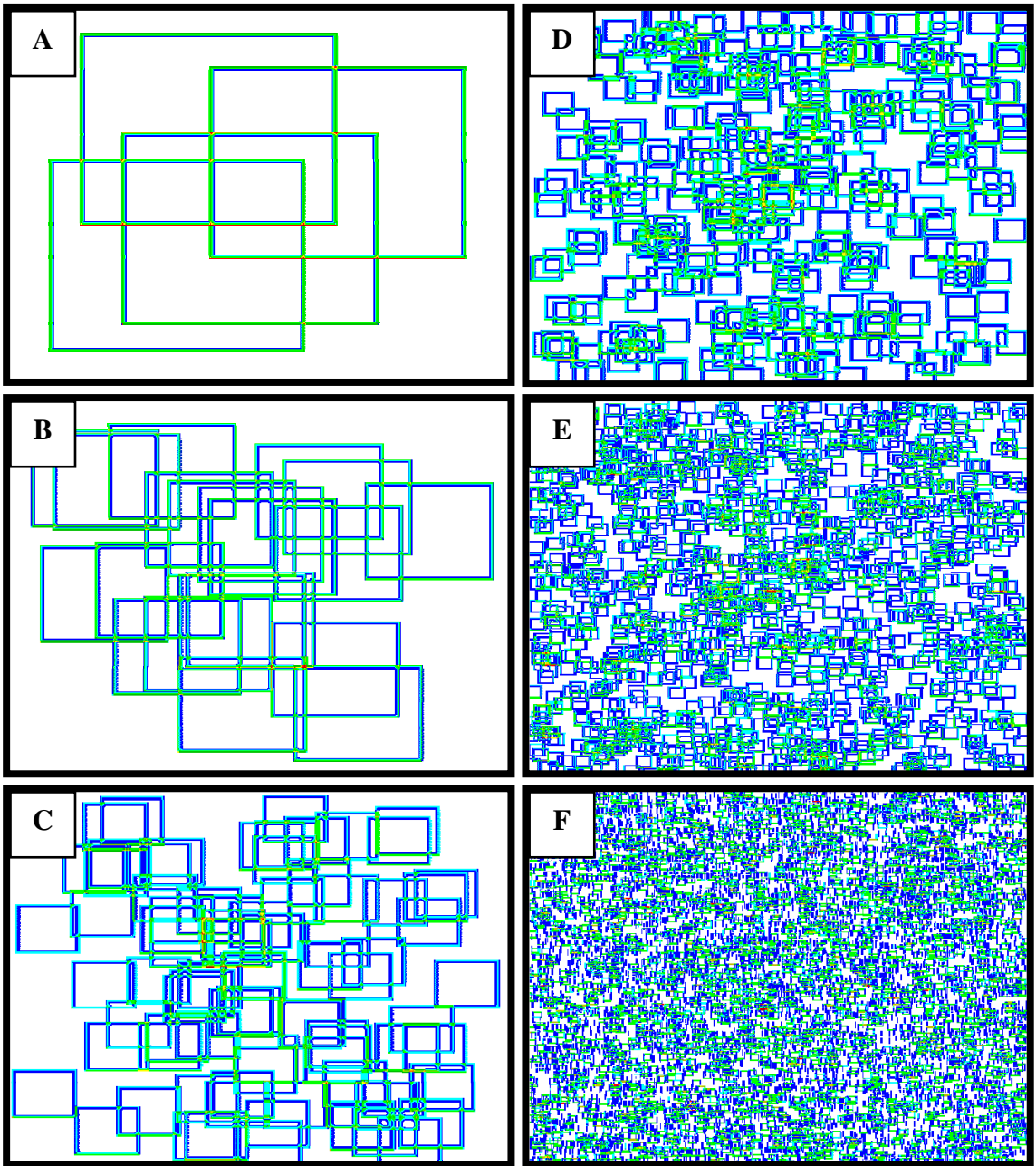


Figure 3.4. Examples of the eddy distribution at various map levels shown by tracing the outer edge of each eddy. Only map levels 2-7 are shown for a 1024x1024 map (A= radius (256) with 4 eddies; B = radius (128) with 16 eddies; C = radius (64) with 64 eddies; D = radius (32) with 256 eddies; E = radius (16) with 1,024 eddies; and F = radius (8) with 4,096 eddies).

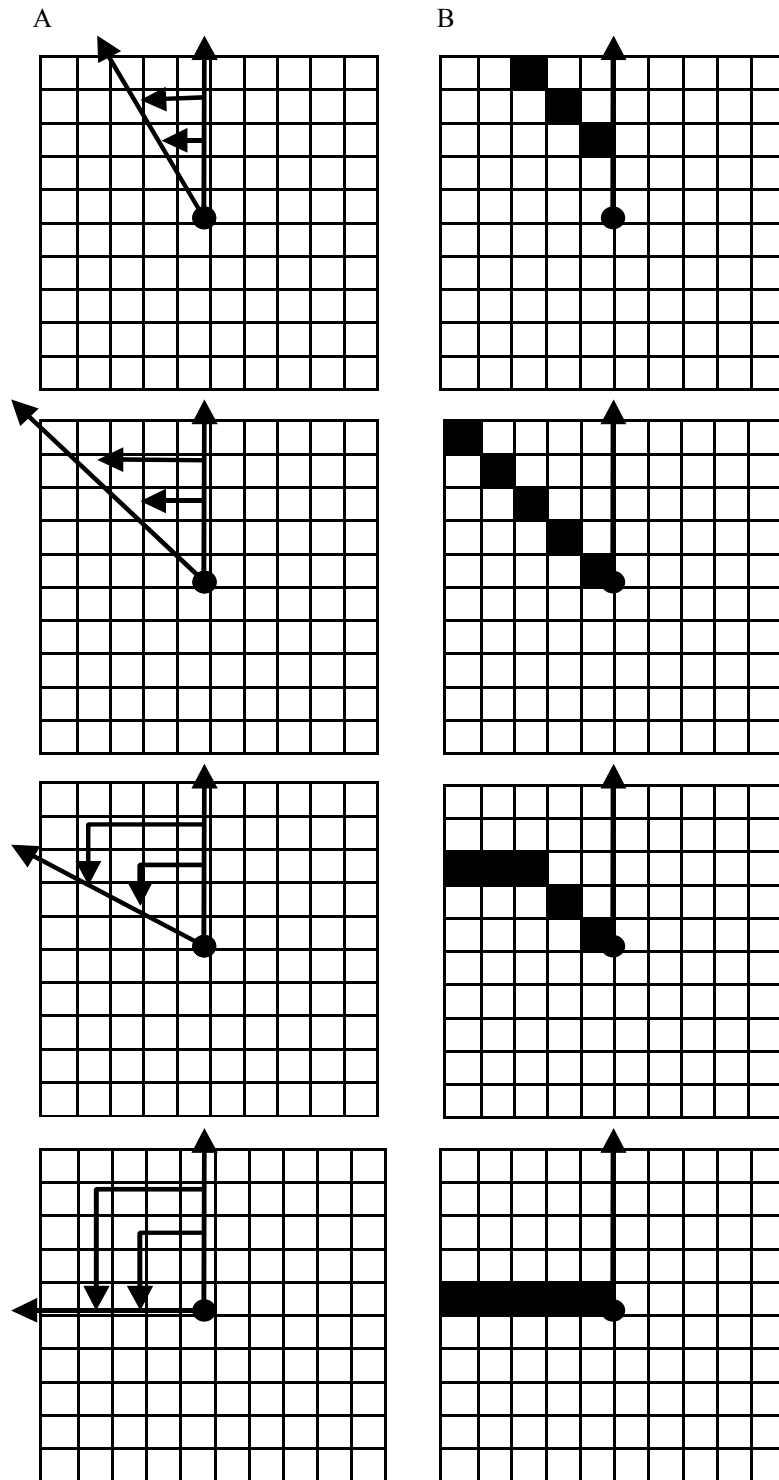


Figure 3.5. Illustration of the eddy rotation algorithm used to move each eddy. Example shown is for an eddy with a radius of five cells, undergoing a quarter turn rotation, for (A) a conceptual realization of circular movement within a gridded framework with square eddies, and (B) the modified routine which had to be developed to prevent excessive smearing of materials due to numerical diffusion errors present when using the conceptual framework. See text for additional details.

method I found this approach created excessive smearing and artificial spreading of substances which prevented realistic concentration gradients from being achieved.

To overcome the potential numerical errors a routine was developed to allow each eddy to rotate sequentially from the outer edge. The routine works by only fluxing material between cells within an eddy, when the amount of material to be fluxed is exactly 100 percent. For example, the outer ring of cells of the eddy would be rotated one grid-cell distance while all inner cells would remain stationary. In the next time-step the outer ring would again rotate along with the next inner ring of cells. This process would be incrementally repeated until the inner-most radius was reached (see Figure 3.5b). The method creates a slight temporal delay but results in no numerical diffusion errors, a key attribute of the ESEM method.

Eddy movement (i.e., rotation) within the ESEM method is done sequentially rather than concurrently as in natural systems. The complexity of the eddy field within the simulation framework (i.e., over 350,000 individual eddies) and the requirement that material cannot flux more than one-grid cell prevent the simultaneous movement of eddies within the simulation domain. Each eddy instead completes its rotational cycle independently which precludes dynamics such as accelerated or inhibited movement due to the simultaneous interaction of multiple eddies. Material transported by the eddies can however move large distances within a time-step due to the movement caused by multiple eddies sequentially affecting the trajectory of a particular particle.

In summary, the ESEM method requires three separate movement routines to work properly (Figure 3.3). The first movement routine determines when to rotate each individual eddy. If I was trying to simulate an idealized turbulent cascade starting from a

large (i.e., parent) eddy breaking up into smaller eddies which then break up into smaller eddies and so on, I would rotate each eddy level in sequence. In natural systems, with fully developed turbulence, there is a field of interacting eddies of different sizes, with new eddies being added and others dissipating into heat. To reproduce this field of eddies I gave each eddy a random start time to begin rotating. Once an eddy starts rotating it continues to rotate until it reaches one full revolution. To conserve rotational momentum, after a full rotation, the eddy remains dormant a prescribed length of time (i.e., rotation frequency) determined by the size of the eddy as described in equation 3 above. As a result, larger eddies are active more often than smaller eddies because the smaller eddies take less time to complete a full rotation. For example, from Table 1, an eddy with a radius of four cells would begin a rotation cycle every 102 time-steps and take 32 steps to complete a rotation while an eddy with a radius of one cell would begin a rotation cycle every 66 time-steps and take eight time-steps to complete a rotation. In a simulation containing 700 time-steps an eddy having a radius of four would have completed six revolutions while an eddy with a radius of one would have completed nine revolutions.

The second movement routine controls the movement of the sub-radii found within each eddy. As described above, the inner part of each circle has less distance to travel than the outer part. Once an eddy starts rotating, it continues to rotate from the outer most ring down to the inner most ring. This movement routine keeps track of where in the rotation cycle (i.e., sub-radii) each eddy is currently located. Using the example above with an eddy of radius four, the outermost ring would move one cell each time-step (32 cells total), the next sub-radius would move one cell every second time-step

(24 cells total), the next sub-radius would move one cell every third time-step (16 cells total) and the inner-most radius would move one cell every fourth time-step (8 cells total).

The third and final movement routine controls how often to call the turbulent mixing sub-routine. This is controlled by the turbulence intensity and the time-step. At a given time-step, the method has an intrinsic turbulence intensity. If the routine were called at each time-step this would be the maximum turbulence intensity for that time-step. By measuring the instantaneous velocity fluctuations (see appendix E) a turbulence intensity can be calculated. If this intensity is too high for planned simulations the frequency at which the sub-routine is called can be adjusted to achieve the correct turbulence intensity. If the intensity is too low, then the time-step can be shortened to achieve the desired intensity, keeping in mind that shorter time steps increase the computational cost. Through adjustments in the time-step or the frequency at which the turbulent mixing sub-routine is called, a wide range of turbulence intensities can be simulated.

3.2.3 Subgrid Diffusive Mixing

As with other turbulent simulation methods in Eulerian based platforms, mixing processes below the resolution of the simulation domain (i.e., grainsize or cellsize) have to be addressed. These subgrid-scale mixing processes are often parameterized as random exchanges between adjacent cells. In my simulation platform, I employ a similar approach and approximate (i.e., parameterize) subgrid-scale mixing processes based on Fick's law, which states that the rate of transfer of matter will be proportional to the

difference in concentration between two finite volumes. Therefore, the flux (J) = $-D(dC/dx)$ where D is the diffusivity and C is the concentration of matter (Okubo 1980). Rearranging the equation and substituting for the concentration difference, the following formulation can be used to solve for the transfer probability (k) between two volumes or adjacent grid cells (Fischer et al., 1979):

$$k = D (dt) / (dx)^2 \quad (\text{Eq. 4})$$

Where

k = Transfer probability

D = Diffusivity coefficient (i.e., eddy diffusivity)

dt = duration of time (i.e., time-step of model)

dx = width of the volume (i.e., grainsize of simulation domain)

With this formulation it is possible to calculate a transfer probability between two grid cells based on the specification of a diffusion coefficient. Since the grainsize of my simulations was 100m, I calculated a transfer probability based on a value which would produce a $-5/3$ spectral slope over the inertial sub-range for a passively mixed tracer. This value was found to be $700 \text{ cm}^2/\text{sec}$ and is within the range of values found in natural systems at the 100m scale (Okubo 1980). Using equation 4 with a value of $700 \text{ cm}^2/\text{sec}$ for the eddy diffusivity, 100 m for the width of the volume (i.e., grainsize) and 1,344 s for

the duration of time (i.e., time-step of the model) the transfer probability can be calculated as follows:

$$K_{700} = 700 \text{ cm}^2/\text{sec} (1,344\text{s}) / (10,000\text{cm})^2 = 0.9408 \text{ \% loss per cell side}$$

per time-step

In my simulation platform, material within a cell is exchanged with the four adjacent cells. This will result in a total flux from each cell per time-step of 3.7632 percent. Experiments testing the accuracy of this method to approximate diffusive mixing gave good agreement with the analytical solution to the diffusion equation for the evolution of a concentration gradient over time (data not shown).

3.2.4 Experimental Treatments

Three experiments were conducted to demonstrate the robustness of the ESEM method at simulating idealized two-dimensional homogeneous turbulence. Turbulent mixing has several key characteristics: mixing time, instantaneous velocity spectrum, and particle spreading rate (Fischer et al. 1979; Okubo 1980; Garrett 1989; Sanford 1997) which I attempted to reproduce and validate through a series of experiments described below.

In all experiments I used a 1024x1024 simulation domain with a pixel size (i.e., grainsize) of 100 m. This allowed examination of the robustness of the technique over a wide range of scales, from 100 m to 100 km, where the use of homogeneous, isotropic, steady-state turbulence to simulate turbulent mixing would be appropriate. The time-step

employed was set at 22.4 min to achieve a turbulence intensity of 10 cm/s for the simulations. See Appendix E for a description of the process used to set the turbulence intensity and Table 3.2 for descriptive statistics for the mean and variability of the turbulence intensity (U_{rms}) of the simulation method. The value used for simulations is within the range found for turbulence intensity over these spatial scales (Middleton 1985). The map boundaries were wrapped to prevent edge effects and simulations were not initiated until one full turbulence cascade was complete (approximately eight days) so that all eddies would have had a chance to begin their rotational cycles. The maximum and minimum eddy radii were 512 pixels and one pixel respectfully. Per the procedure described in section 3.2.2, over 350,000 eddies of various sizes were generated and randomly placed on the map. I replicated each experiment 10 times utilizing a different random distribution of eddies to understand the degree to which the seeding of the eddy field will affect the results of each experiment. When spectral analysis was utilized I sampled multiple locations on the map which allowed an ensemble spectral slope to be calculated (see Appendix F for details on the spectral analysis technique).

Experiment #1-Rate of Pattern Breakdown: A key characteristic of turbulence is that it acts to accelerate the breakdown of scalar and concentration gradients through mixing caused by eddies within the turbulent field (Fischer et al. 1979; Okubo 1980). The rate at which turbulent mixing breaks down gradients is a function of the turbulent mixing intensity. When a scalar or gradient is subjected to turbulent mixing the initial pattern (i.e., variance) found within the field should be transferred to smaller and smaller scales where it is acted until by diffusive mixing until the pattern is eventually dissipated to background levels. In the first experiment, to determine if the method is capable of

Table 3.2. Descriptive statistics for the mean and variability of the turbulence intensity (U_{rms}) of the simulation method. For each map replication, the RMS was calculated for 10,000 particles after one solution-step. This was repeated for 100 solution-steps to determine the variability within a map over-time as eddies cycle on and off. The grand mean is the average RMS value of the ten replications. See Appendix E for additional details.

Map Replication (#)	Mean	Min	Max	Std Err	CV
1	1.33	1.13	1.56	0.012	9.27
2	1.34	0.87	1.54	0.013	9.87
3	1.41	1.01	1.64	0.015	10.71
4	1.34	0.96	1.54	0.019	14.09
5	1.33	1.09	1.52	0.011	8.63
6	1.29	1.05	1.51	0.013	10.33
7	1.34	0.98	1.57	0.018	13.66
8	1.37	1.06	1.56	0.010	7.62
9	1.39	1.14	1.72	0.017	12.07
10	1.29	0.88	1.50	0.013	9.72

Grand Mean = 1.34

mixing a substance at an appropriate rate for a given mixing intensity, I measured the length of time for the method to remove the initial variance from a passive scalar variable. Similar to Abraham's (1998) approach, for a scalar variable I used a cosine function to create pattern within the simulated domain at the largest scale possible, the width of the map. After initialization of the map with the cosine function, I ran simulations for 20 days and measured the rate at which the pattern was altered by the turbulence mixing routine. Twice per day, 90 evenly spaced transects along the length of the map were saved and then statistically analyzed for changes in variance.

Experiment #2- Instantaneous Velocity Fluctuations: A second important characteristic of turbulent mixing is the velocity spectrum of the turbulence field of eddies. The velocity spectrum measures the eddy size distribution of turbulence intensity at a given instant (Garrett 1989). Since the turbulent field is made up of eddies of all different sizes, superimposed on one another, some areas of the turbulent field will undergo more rapid mixing than other areas. This field as a whole, however, should be correlated to some degree with the various length scales of the eddies and is often found to have a spectral slope of -1.67 over the inertial sub-range of turbulent mixing (Kundu 1990).

For the second experiment, I measured the instantaneous velocity fluctuations at various locations on the map, over the course of a simulation, to determine the correlation structure of the overall turbulence field produced by the ESEM method. The velocity fluctuations were measured by calculating the distance that individual particles, along a transect, moved after several simulation time-steps. This analysis was performed on 50 evenly spaced transects with a length of 512 grid cells. Because the spectral analysis

requires the transect length to be a power of two, and since the simulation domain was 1024x1024 grid cells, a length of 512 was the largest possible dimension that could be analyzed to avoid potential edge effects. The 50 transects were then broken up into north-south and east-west velocity components and the ensemble spectral slope, β , was calculated. The analysis was repeated at 20 additional time-points over the course of a three-day simulation. This high intensity sampling was performed to understand how the velocity fluctuations change during the simulation as eddies within the turbulent field begin and end their respective rotational cycles.

Experiment #3 – Scale Dependent Diffusion: The third characteristic of turbulent mixing that I tested was the ability of the ESEM to reproduce the spreading rate of a patch due to scale-dependent diffusion. Spreading of a patch, defined as a substance with an elevated concentration above a background level for that substance (e.g., nutrient concentration, phytoplankton biomass), subjected to turbulent mixing should accelerate over time as the patch grows larger. For example, when a patch is small it can be transported whole by eddies larger than the width of the patch and only eddies smaller than the width of the patch will cause it to increase in size. As the patch increases in size more and more eddies will be able to cause the patch to increase in size and thus increase the rate of spread (Okubo 1976; Okubo 1980). Fickian diffusion based methods are limited to constant linear rates of spread which diminishes their utility for addressing environmental (e.g., pollution transport) and biological issues (e.g., bloom dynamics, chemical signaling) where the time evolution of concentration gradients is important.

For the third experiment, I measured the rate of spread of a patch subjected to the eddy field within the ESEM to determine if the method was capable of reproducing scale-

dependent mixing. I initialized the center of the simulation grid with a square patch consisting of 1000 particles (i.e., one particle per grid cell) and then recorded the movement of each particle over the course of a simulation. The variance and average displacement (RMS, mean squared radius of all particles) of the patch after each time-step was calculated. Since the center of the patch changed over time, the RMS was calculated from a center location determined by the average coordinates for each particle in the patch. Simulations were run for seven days, which was long enough to calculate spreading rates and to prevent edge effects from affecting particle travel. The RMS value was calculated similar to the methodology employed in Appendix B to determine the turbulence intensity.

3.3 Results

The series of simulations indicated that the overall performance of the ESEM was excellent. As seen in Figure 3.6, panel A, the turbulence generation routine was capable of realistically mixing a cosine pattern over a 20-day simulation. The cosine pattern was transformed from a uniform pattern, with only variability at the largest scale (i.e., the scale of the map), to a well-mixed pattern, with very little evidence of the initial pattern. As the pattern is mixed, the characteristic swirls and eddies of turbulence are evident over a range of scales.

Because the ESEM method produces no numerical errors, the maps in the Figure 3.6, panel A series become increasingly pixilated over time. As a result the ESEM method will accumulate variance over time rather than dissipate variance as is seen in natural systems through fine-scale diffusive mixing processes. The ESEM method must

therefore be coupled with a subgrid-scale mixing routine to accurately simulate the mixing of concentration gradients over time.

To simulate subgrid diffusive mixing (i.e., mixing below the grainsize of the simulation platform) within my simulation domain I coupled the ESEM method with a Fickian diffusion routine to produce the realistic fine-grained patterns shown Figure 3.6, panel B. The exchange rate between the grid cells was calibrated to an eddy diffusivity value of $700 \text{ cm}^2/\text{sec}$ and was chosen through a series of experiments (data not shown) where the eddy diffusivity was varied between $100 \text{ cm}^2/\text{sec}$ and $1000 \text{ cm}^2/\text{sec}$, values which bracket those typically seen in natural systems at the 100 m scale (Okubo 1980). The $700 \text{ cm}^2/\text{sec}$ value produced the most realistic change in variance with scale for a passively mixed tracer over the inertial subrange and was able to reproduce/maintain a $-5/3$ spectral slope over this range during the 20-day simulation (see Figure 3.7 panel B).

Since Fickian diffusion methods are sometimes used to parameterize turbulent mixing processes at the scale of the simulation domain I conducted an experiment using only the Fickian diffusion method employed above as a comparison to the mixing dynamics produced by the ESEM method. To provide an accurate comparison I calibrated the eddy diffusivity value so that by the end of the simulation the variance removed from the cosine pattern by the Fickian diffusion routine would be roughly equal to that removed by the ESEM method with subgrid mixing. The eddy diffusivity value chosen was $1.25 \times 10^6 \text{ cm}^2/\text{sec}$ and is typical of values at the 100 km scale (Okubo 1980). The simulation dynamics are shown in Figure 3.6, panel C. As can be seen, the initial cosine pattern was gradually reduced to mean levels but without any of the intricate spatial patterns observed in panels A and B.

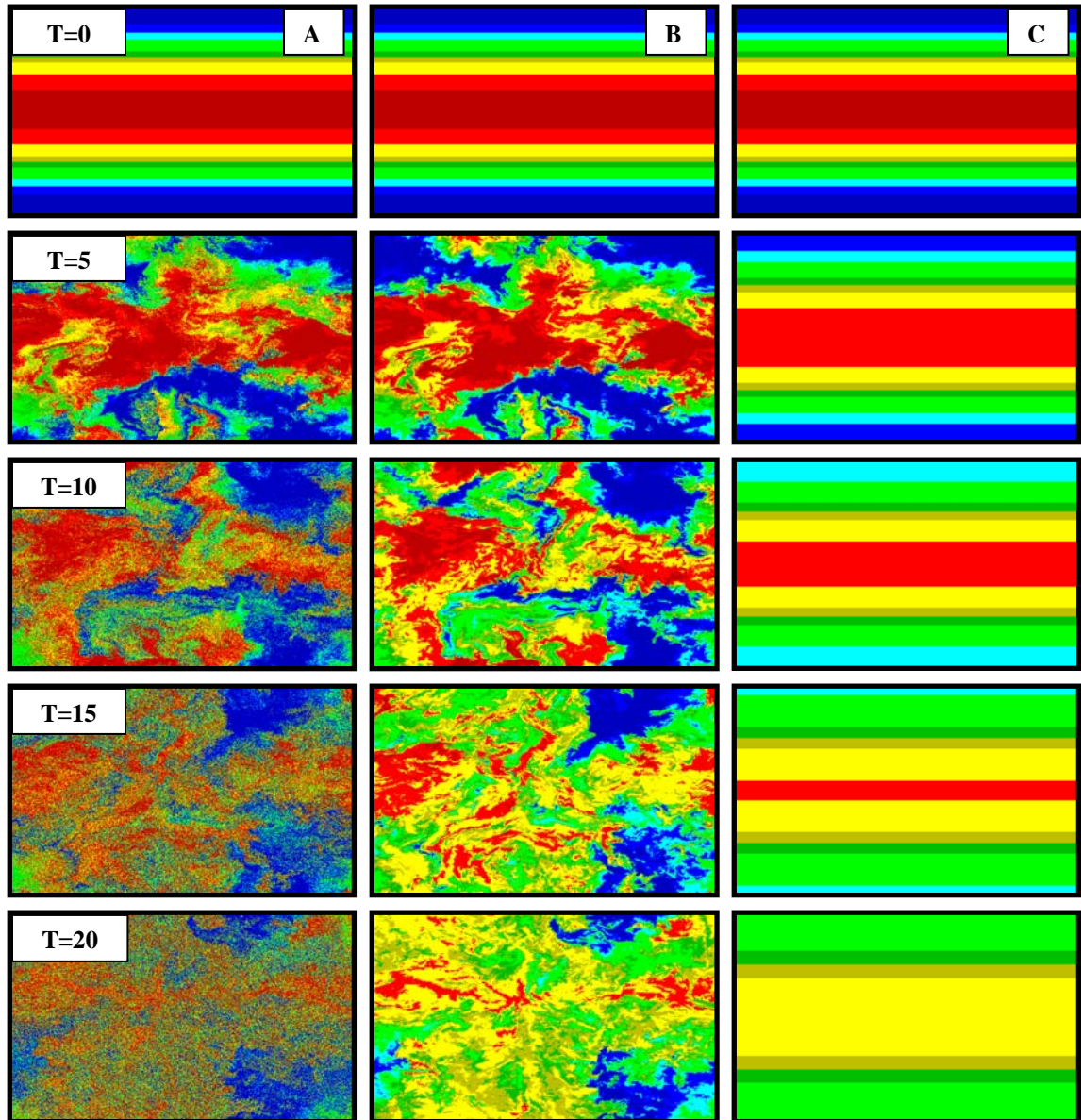


Figure 3.6. The series of panels illustrate the mixing effects of the turbulence generation and Fickian diffusion routines on a map initialized with a cosine function. The simulation domain is 1024x1024 pixels with a pixel width of 100 m with values from one to zero and a corresponding color range from red to blue. The simulation was run for 20-days with snapshots of the pattern taken a five-day intervals. The panel A series shows the breakdown of the pattern using only the ESEM method without the incorporation of any subgrid diffusive mixing. Panel B incorporates subgrid diffusive mixing (i.e., eddy diffusivity) at a value of $700 \text{ cm}^2/\text{sec}$ which is within the range found at the 100m scale for natural systems ($100\text{-}1000 \text{ cm}^2/\text{sec}$). Panel C shows the breakdown of the pattern using only Fickian diffusion to approximate turbulent mixing effects at the scale of the simulation domain (i.e., a value of $1.25 \times 10^6 \text{ cm}^2/\text{sec}$). See text for additional details.

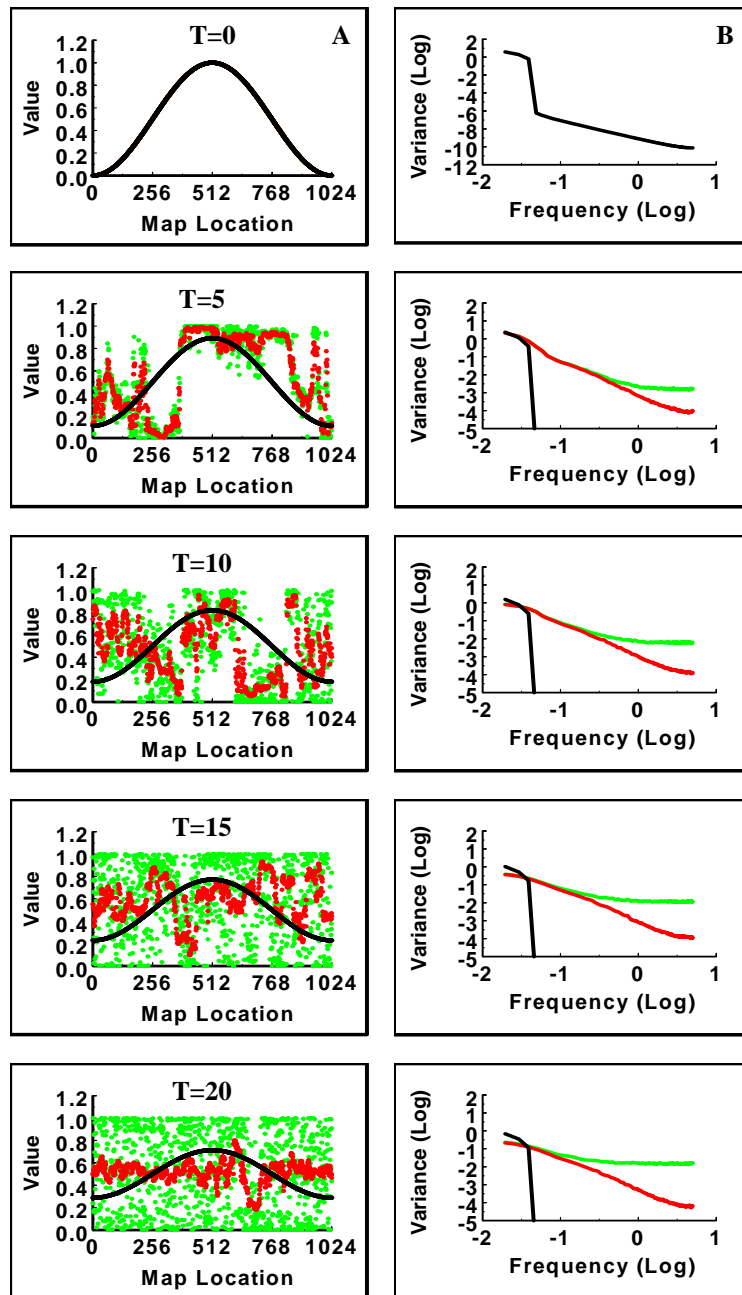


Figure 3.7. Panel A follows a representative transect near the middle of the simulation domain at each snapshot in time over the 20-day simulation represented in Figure 6. Panel B characterizes the change in the pattern over the simulation domain as represented by the change in spectral signature calculated from a composite spectral analysis of 90 evenly spaced transects across the simulation domain. The lines correspond to simulations utilizing only the ESEM method (green line), the ESEM method plus subgrid diffusive mixing (red line), and only diffusive mixing parameterized to approximate turbulent mixing at the scale of the simulation domain (black line).

At the level of individual transects across the simulation domain, Figure 3.7, panel A, illustrates the effect of the turbulent mixing under the three cases illustrated in Figure 3.6. The larger eddies can cause rapid mixing as seen in the sharp changes in value along the transect line (e.g., $T=5$) for the green and red lines. The smaller eddies are also acting to mix the pattern and their impact is more evident as the simulation progresses due to the continual breakdown of the large-scale pattern into smaller fragments. As the simulation progresses, the impact of the subgrid diffusive mixing is clearly seen in the “smoothing” of the transects (red line). If the simulations were conducted over longer time periods the case with subgrid diffusive mixing would eventually reach mean (i.e., the equilibrium condition) as represented by a flat horizontal line with no evidence of the previous pattern. The simulation without any subgrid mixing would remain well homogenized and approximate a random noise pattern (green line). For the simulation with only Fickian diffusion (black line), the cosine pattern gradually flattens over time toward the mean level without the sharp gradients seen in the other two cases.

In panel B of Figure 3.7, the change in variance as a function of scale is statistically shown through the use of spectral analysis. An analysis of the initial pattern (i.e., $T=0$) shows the slope is dominated by variance at the broadest scales. Over the simulation, the mixing caused this large-scale variance to decline with a corresponding transfer of variance to finer scales (i.e., the turbulence is injecting small-scale variability). This result is a flattening of the spectral slope, β (green line). If the simulation were allowed to run long-enough, the spectral slope will become uniformly flat with equal variance at all scales (i.e., $\beta=0$). In natural systems, this variance is dissipated by

diffusive processes which necessitated the inclusion of subgrid diffusive mixing to the ESEM method. When this is done, the small scale variance is indeed removed as seen in the red line. The diffusive mixing removed the small scale variance and reproduced the 5/3 spectral slope typically observed in natural conditions. Without the ESEM method, and using only Fickian diffusion to simulate turbulent mixing, the variance at the largest scales is maintained over the course of the simulation (black line).

3.3.1 Breakdown of a passive tracer

The rate at which turbulent mixing breaks down concentration gradients is a function of the turbulent mixing intensity. For the simulations I used a turbulent mixing intensity of 10 cm/s, to correspond to the value used in Abraham (1998) for mid-latitude systems. For a given turbulence intensity a mixing time can be calculated which is the time necessary to transfer the variance from the largest scales down to the smallest scales. At a turbulence intensity of 10cm/s the mixing time should be approximately 15 days in aquatic systems for fully developed two-dimensional homogeneous turbulence (Klein and Hua 1990).

Figure 3.8 shows the change in variance of the initial cosine function over the 20-day simulation for each map series represented in Figure 3.6 and 3.7. In the case where no subgrid mixing was added the variance remained relatively constant over the duration of the simulation. In the case with subgrid mixing set to $700 \text{ cm}^2/\text{sec}$ the variance generally decreases with time and by the end of the 20-day simulation almost all of the variance has been removed. This is in general agreement with the time scales necessary to transfer variance from large- to fine-scales in natural systems of similar size and

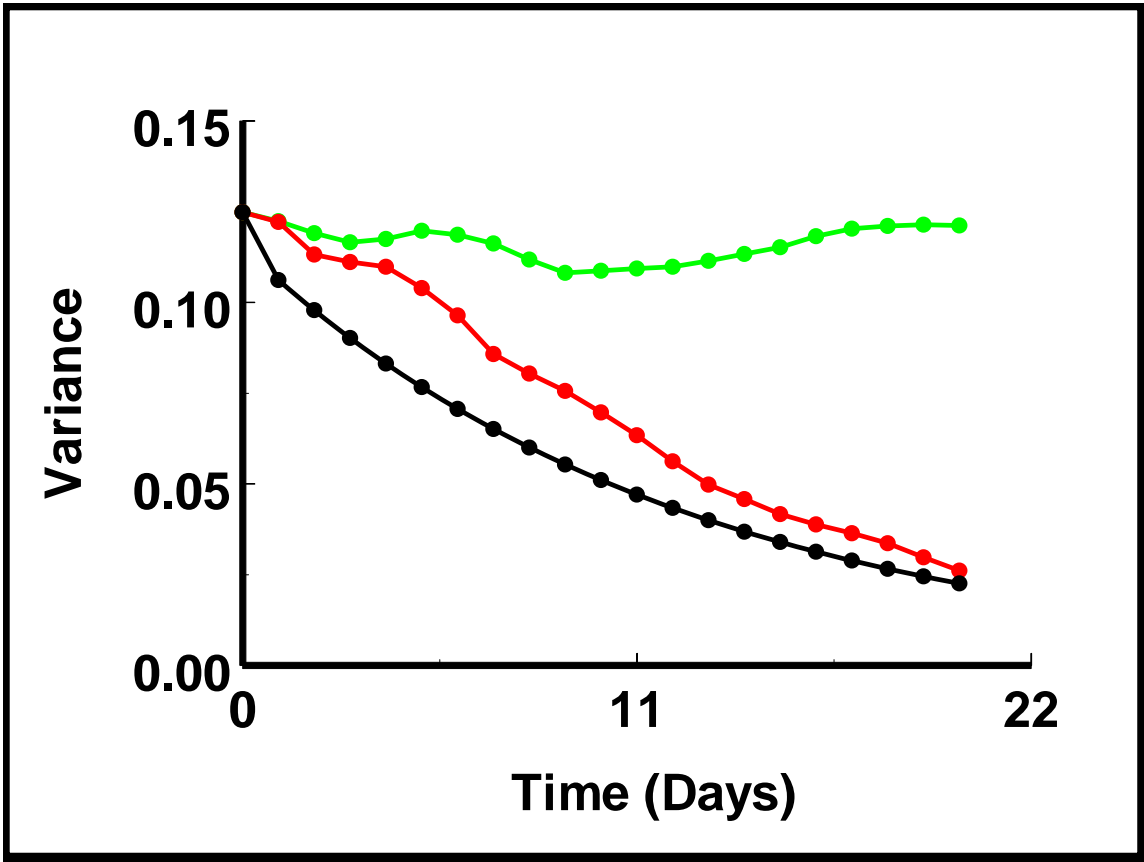


Figure 3.8. Shows the change in variance over the 20-day simulation for each map series represented in Figure 6. The variance was calculated from 90 evenly spaced transects across the simulation domain at one day intervals. The lines correspond to simulations utilizing only the ESEM method (green line), the ESEM method plus subgrid diffusive mixing (red line), and only diffusive mixing parameterized to approximate turbulent mixing at the scale of the simulation domain (black line).

mixing intensity as that used in my simulations (Klein and Hua 1990). In the case with only Fickian diffusive processes (i.e., without the ESEM), the variance also decreases with time but at an exponential rate which is faster at first and then slower than the case with the ESEM and subgrid diffusive mixing. The value used in my simulations to approximate turbulent mixing with Fickian diffusion was calibrated to give the same final variance over a 20-day simulation as seen in the case with the ESEM and subgrid diffusive mixing.

3.3.2 Instantaneous velocity spectrum

For experiment two, my goal was to confirm that the method accurately reproduced the velocity spectrum for fully developed, homogeneous, isotropic turbulence. The velocity spectrum measures the degree to which velocity fluctuations at different scales are correlated. According to theory, the spectrum of the spatial distribution of instantaneous velocity fluctuations should have a spectral slope (β) of -1.67 (Kundu 1990). This value captures the structure of the turbulence created by the rotation and interaction of the range of eddy sizes present within the turbulent field within the inertial sub-range.

To estimate the instantaneous velocity fluctuations, I measured the displacement of a line of particles within the simulation domain after three solution-steps of the turbulence routine. The recorded displacement was broken into an (X) and (Y) component and then analyzed through spectral analysis to obtain the spectral slope. Figure 3.9 shows the analysis of one example transect. For this example, the β was 1.68 which agrees well with the empirical prediction.

I also wanted to measure changes in β over the course of a simulation as the various eddies underwent their rotational cycles and determine how the spectral slope changed when different eddy fields were utilized. Table 3.3 shows the result of this analysis. As evidenced in the low coefficient of variation (max CV was 4.8%) within each map replication, the spectral slope remained fairly consistent over the course of a simulation. When different eddy fields are used (i.e., map replication) β ranged from 1.66 to 1.77 with an average value of 1.72. While the average spectral slope of all the map replications was slightly higher than the expected value of 1.68, all of the individual eddy distributions produced velocity fluctuations which were close to the expected value and should be capable of simulating a realistic turbulence field and associated correlation structure.

3.3.3 Spreading rate of a patch

In addition to the breakdown of concentration gradients, as seen in Figure 3.6 and 3.7, turbulent mixing also causes a patch to increase in size as eddies disperse parts of the patch away from the patch edge. This advection causes an increase in the variance of the patch with time as well as dilution (i.e., breakup and diffusion) of the patch with non-patch material. Figure 3.10 shows the spreading of a patch caused by the turbulence generation routine over a seven-day simulation. Notice how the patch becomes asymmetrical at first and then begins to break-up into smaller patches and fragments.

For the third experiment, I calculated the change in the patch size, as measured by the RMS, of the scenario illustrated in Figure 3.10. I also tested the impact of changing the eddy distribution on the spreading rate by running nine additional experiments with a

different random number (i.e., a different eddy field). The results of the analysis are shown in Figure 3.11. For these experiments and analysis the simulation were conducted using the case without any subgrid mixing added. Because the analysis required the tracking of individual particles, the addition of subgrid mixing would have greatly complicated the calculation of the RMS value (see Appendix E). Over the time scales of the analysis (i.e., seven days) the added spreading caused by subgrid diffusive mixing was minor to the spreading caused by the ESEM turbulent mixing routine (see Figure 3.6, T=5 series) indicating that the use of the ESEM method without subgrid diffusive mixing was appropriate.

In Figure 3.11, panel A, the RMS values generally increased with time. The overall increase was approximately linear with each line having periods of accelerated or decreased changes in spreading rate over the course of the simulation. The variations in the spreading rate could have been caused by a number of factors such as the various eddies switching on and off; from the patch undergoing mixing by eddies larger (accelerated spreading) or smaller (no spreading) than the patch; and by the patch being mixed so that it wraps onto itself (decrease in spreading rate). The wide range in RMS values between experiments is probably due to there being only one patch on the map which will tend to emphasize the importance of the random placement of the eddies, especially the larger ones which are fewer in number. In the previous experiments, I sampled multiple locations on the map which allowed integration of the entire eddy field and hence less variance between replications.

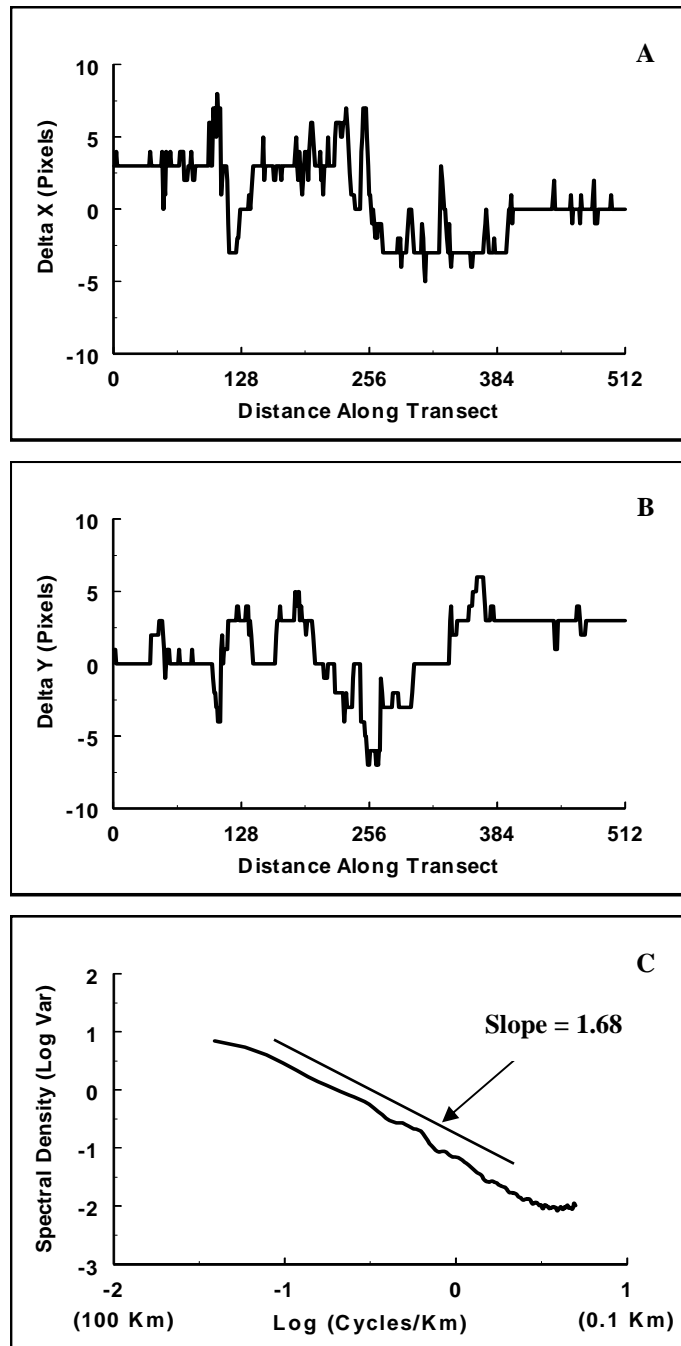


Figure 3.9. Velocity fluctuations and subsequent spectral analysis for a representative transect. For each map replication, the displacement of simulated particles was recorded after three time-steps, for 41 evenly spaced transects one-half the length of the map. Figure A and B show the number of map pixels traveled in the (X) and (Y) direction for one representative transect. Figure C is the composite spectral analysis and resulting slope of the individual transects for all 41 transects. This analysis was repeated for 20 additional time-points for each map replication, the results of which are shown in Table 3.3. See text for additional details.

Table 3.3. Descriptive statistics for the mean and variability of the spectral slope of the instantaneous velocity fluctuations. For each map replication, the composite spectral slope was calculated from 82 evenly spaced transects (one-half the map length). This was repeated 20 times over the course of a simulation and then averaged to develop the descriptive statistics shown for each map replication. The grand mean is the average composite spectral slope over the ten replications.

Map Replication (#)	Composite Spectral Slope	Min	Max	Std Err	CV
1	1.67	1.53	1.79	0.018	4.79
2	1.77	1.59	1.83	0.011	2.88
3	1.76	1.62	1.83	0.011	2.75
4	1.72	1.63	1.82	0.012	3.20
5	1.66	1.58	1.75	0.011	2.99
6	1.70	1.61	1.77	0.010	2.69
7	1.74	1.61	1.84	0.014	3.52
8	1.77	1.62	1.88	0.013	3.38
9	1.69	1.60	1.83	0.015	3.93
10	1.68	1.61	1.79	0.011	3.04

Grand Mean = 1.72

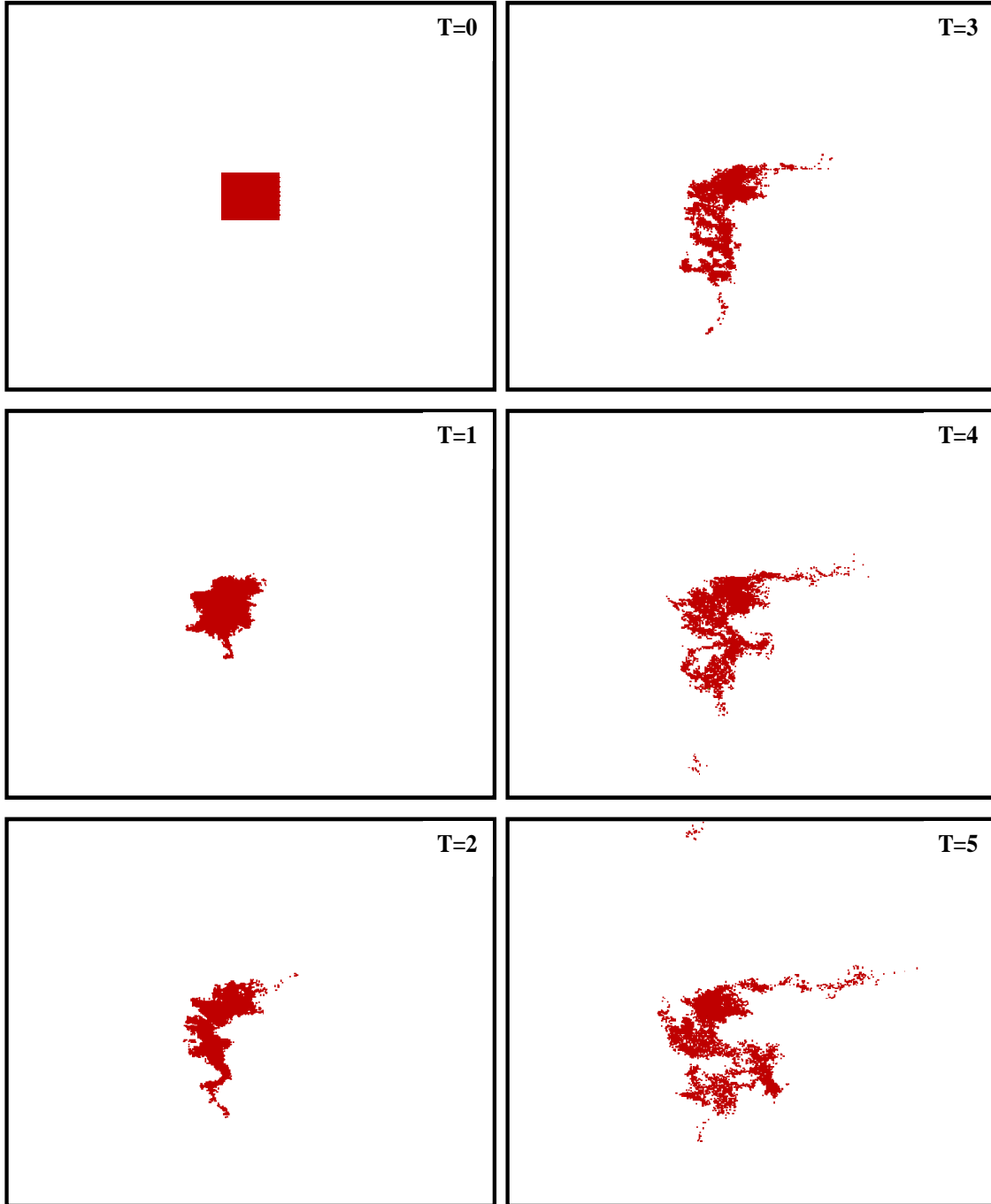


Figure 3.10. Sequence of pictures showing the spreading dynamics associated with a patch of particles over a seven-day simulation. The map ($10,485 \text{ km}^2$) was initialized with 14,641 particles in a square pattern (146 km^2) in the center of the map. Only days one through five are shown. See text for additional details.

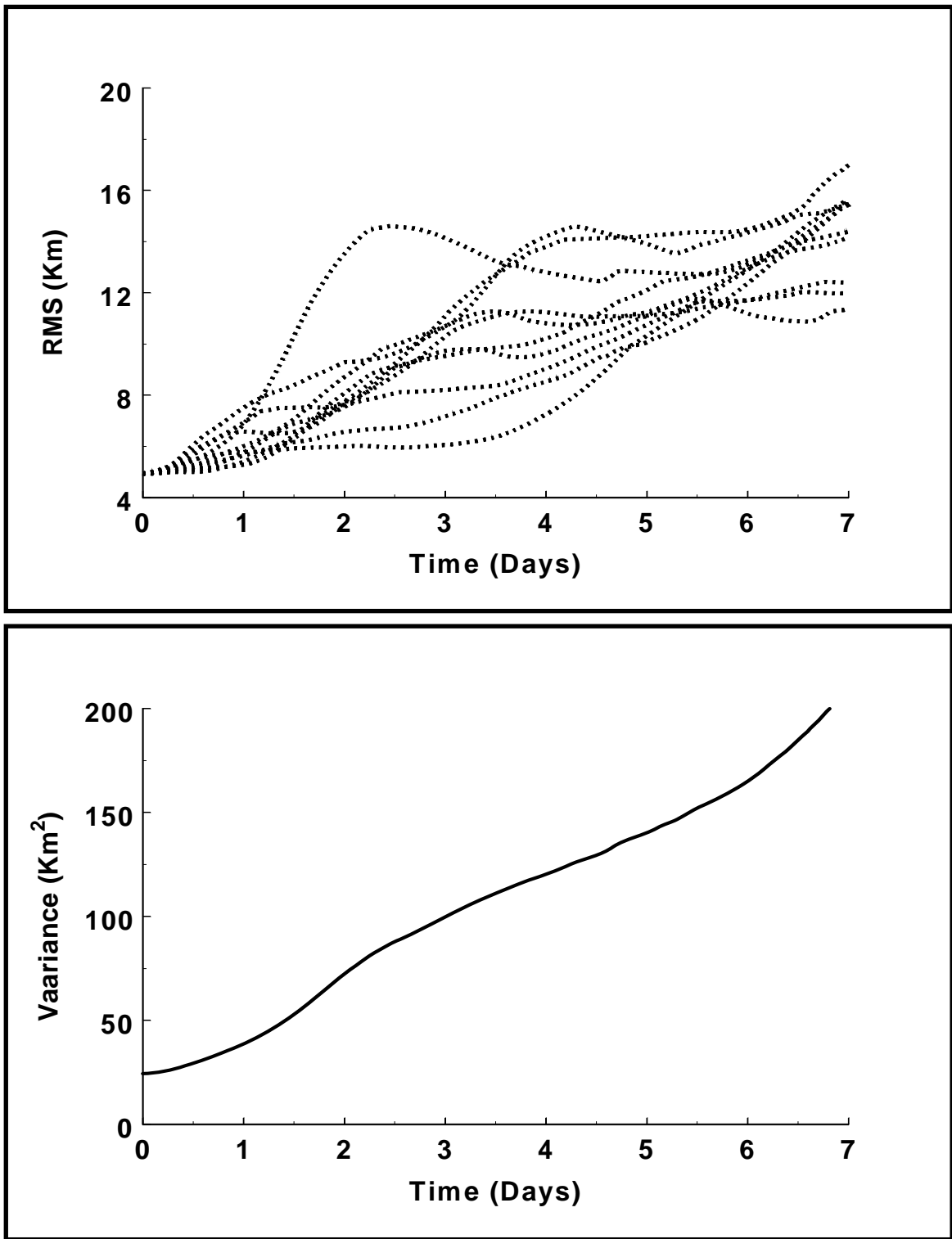


Figure 3.11. Panel A illustrates the change in patch size, as measured by RMS, over a seven-day simulation. Each line represents a simulation initialized with a different eddy distribution with an initial patch size the same as in Figure 3.10. Panel B is the change in variance with time for the average of the RMS values from Panel A.

In fully developed turbulence, the variance in the size of the patch should increase according to the square of the displacement with time because as the patch grows in size more and more eddies will be capable of increasing the size of the patch (Okubo 1980). When the RMS value is converted to variance the patch did show a non-linear increase with time as would be expected in natural systems undergoing turbulent mixing (Figure 3.11, panel B). Since I was limited to seven days (i.e., to prevent edge effects) in which to conduct the simulations, additional experiments will be necessary to determine if the variance does continue to accelerate with time and if the relationship holds under different initial patch sizes. The ability to simulate scale-dependent mixing is a key benefit of the ESEM method over the simulation of turbulent mixing via only Fickian diffusion methods.

3.4 Discussion

There are a limited number of options available for simulating the effects of turbulent mixing within aquatic ecosystem models. Some of these include the direct simulation of turbulence through complex hydrodynamic simulations involving the equations of motion (e.g., Halloway 1986; James 1996; Vested et al. 1996; Kantha and Clayson 2000; Walters 2005.), the use of prescribed flow fields, multifractal maps or statistical approximations (e.g., Aref 1984; Marguerit et al. 1998; Van Dan et al. 1999; Lopez et al. 2001; Reigada et al. 2003; Halloway 2004), or the use of Fickian diffusion and random walk methods to parameterize the turbulent mixing as a diffusive, random process (e.g. Okubo 1980; Powell and Okubo 1994; Visser 1997; Riddle 1998; Brentnall et al. 2003). For my planned dissertation experiments, I required a turbulence generation

method which could simulate the dynamic nature of turbulent mixing, over a wide range of scales, while also preserving fine-scale concentration gradients. These constraints limit the applicability of existing turbulent mixing simulation methods to address my particular needs.

To overcome the limitations of the methods above, I developed the ESEM to simulate idealized two-dimensional turbulence in aquatic systems. Based on a seeded eddy model developed for particle systems, I adapted the technique for a Eulerian framework while preserving the cascade of turbulent energy from broad- to fine-scales (Dyke and Roberston 1985; Abraham 1998). Results from the three experiments reported here demonstrate the adequacy of this method for simulating steady-state turbulent mixing over scales ranging from 100 m to 100 km. The ESEM method was capable of 1) reproducing the velocity spectrum of homogeneous, isotropic, turbulence; 2) accurately reproducing the spreading rate of particles from scale-dependent turbulent mixing and 3) breaking down large-scale concentration gradients over realistic time-frames while maintaining realistic small-scale gradients. Together, these capabilities give my method the advantage of being able to introduce realistic fluid effects within a Eulerian framework without having to resort to complex hydrodynamic models or oversimplifications of turbulent mixing with Fickian based diffusion methods.

3.4.1 Benefits of the ESEM method

Fickian diffusion methods are often used to parameterize turbulent mixing effects in aquatic ecosystem models, either as a replacement for the direct simulation of turbulent mixing via Navier-Stokes equations or to parameterize mixing effects below the grid

resolution of hydrodynamic models (e.g., Okubo 1980; Steele and Henderson 1992; McGillicuddy et al. 1995; Franks and Chen 2001; Brentnall et al. 2003; Patel et al. 2004). One reason for the popularity of the approach is that Fickian diffusion methods are fairly simple to incorporate into grid-based systems and there is an extensive theory developed based on their use. There are some limitations, however, with using Fickian diffusion and related random walk methods to approximate turbulent mixing in aquatic ecosystem models. These include high computational cost of fine grids, numerical diffusion, an inability to inject spatial heterogeneity during the mixing process and an inability to incorporate scale-dependent diffusion (i.e., variable eddy diffusivity). In the next sections, I discuss how the ESEM method is capable of overcoming these critical limitations and can become a viable replacement for traditional Fickian diffusion methods used to parameterize turbulent mixing effects in aquatic ecosystem models.

Preservation of fine-scale concentration gradients: The benefit of the ESEM method is that it is able to inject spatial heterogeneity during mixing and to preserve these fine-scale concentration gradients during the mixing process. Since Fickian diffusion methods are based on approximating turbulent mixing as a diffusive process, any concentration gradients within the turbulent field will be randomly mixed over time, resulting in mixing but also the eventual loss of any concentration gradients. Studies have shown that the mixing caused by a turbulent field will inject pattern (i.e., heterogeneity) into concentration gradients (Abraham 1998). The mixing acts to break up large-scale gradients and transfer that pattern to finer and finer scales until eventual dissipation by diffusive processes once the gradients are thoroughly mixed. Fickian diffusion methods are unable to inject the necessary spatial pattern at short and

intermediate time scales and can give misleading results when trying to interpret observed spatial patterns in aquatic systems.

The ESEM method is able to preserve concentration gradients and inject spatial variability since the full spectrum of eddies, present within a turbulent field, is also present within the turbulence generation routine through the generation of a space-filling field of eddies. Although the ESEM is a simple representation of a complex physical process, the method reproduces the velocity spectrum of turbulence and generates the characteristic swirls caused by turbulent mixing. It is these swirls that act to break down the concentration gradients and inject variability from large to fine scales as opposed to the random mixing of Fickian diffusion methods which are unable to generate variability and actually artificially destroy gradients which may be present.

The ESEM method has the same limitations as other grid-based turbulence generation methods. The method must be coupled with a Fickian diffusion routine to approximate mixing effects below the grid resolution of the simulation domain. Without this coupling the ESEM method retains all of the original variance of the initial pattern and there is no mechanism to dissipate the concentration gradients as happens in natural systems through random diffusive mixing (see Figure 3.6, panel A). When combined with a Fickian diffusion routine the ESEM method is able to realistically dissipate the variance over appropriate time-scales (see Figure 3.6, panel B) without removing the pattern introduced by the ESEM turbulence generation routine. In high resolution simulations, with grid sizes on the order of centimeter to meter scales, realistic mixing should be achieved since the parameterization of the Fickian diffusion fluxes would correspond to the true diffusive fluxes occurring in the natural environment. When

Fickian diffusion methods are used to simulate turbulent mixing processes at larger scales (see Figure 3.6, panel C), problems can arise since it is not possible to simulate the heterogeneity which is introduced during the mixing process (see Figure 3.6, panel A and B).

Ability to incorporate scale-dependent diffusion: Another potential problem with using Fickian diffusion methods to approximate turbulent mixing effects in aquatic systems is that the mixing intensity has to be set for the whole simulation domain. In natural systems, when concentration gradients (e.g., patches of a substance) undergo mixing within a turbulent field, the rate of breakdown in the gradients or spreading of the substance is dependent on scale (Okubo 1980; Petrovskii 1999). For example, when a patch is small (i.e., smaller than most of the eddies present in the turbulent field) most of the eddies will simply move the patch around without much mixing while eddies substantially smaller than the patch will only be capable of small-scale mixing within the patch, similar to diffusive mixing. As a patch grows larger it will interact with more of the eddies within the eddy field causing accelerated mixing, especially the larger eddies with greater velocities (Okubo 1980). Each patch is therefore subject to its own unique turbulent mixing intensity based on the local eddy field. Fickian diffusion methods cannot have a variable “eddy diffusivity” parameter, although recent advances are helping to minimize this impact especially for Lagrangian systems (Ross and Sharples 2004). Depending on the scale at which the eddy diffusivity parameter is set, the spreading rate of small patches will be overestimated while those of large patches will be underestimated. In addition, the spreading rate of the substance will be linear with time

rather than non-linear as the patch grows larger and interacts with additional eddies within the turbulent field.

With the ESEM method, the turbulence intensity will vary according to the local eddy field around a particle, patch, or concentration gradient. The overall turbulence intensity is set globally based on averaging the velocity fluctuations over the whole simulation domain (see Appendix E). Once set, each of the eddy sizes rotates with its own velocity and frequency of rotation. Since I am explicitly simulating the movement of each eddy within the turbulence field, the ability to simulate scale-dependent mixing effects is inherent within the ESEM technique. I verified this property by simulating the spreading rate and variance of a single patch (see Section 3.3.3) and found that the spreading rate was variable through time due to periods of accelerated spreading and sometimes periods of reducing spreading. While unexpected, these dynamics seem accurate because as the simulated patch was being broken apart, it was sometimes mixed onto itself causing a reduction in the spreading rate. The overall variance of the patch did accelerate with time as would be expected by a patch undergoing turbulent mixing and interacting with larger and more energetic eddies. My method is especially powerful when simulating multiple patches or gradients which span many scales. To my knowledge, the ability to simulate scale-dependent diffusion within an Eulerian framework is currently not possible except with the direct simulation of turbulent mixing. Thus, the ESEM represents a significant improvement over the use of Fickian diffusion methods to approximate turbulent mixing effects.

Elimination of numerical diffusion: Another inherent problem with Fickian diffusion methods, especially within a Eulerian, or grid-based framework, is numerical

diffusion (see Appendix D). These methods exchange material between adjacent cells through a flux parameter which is often based on an eddy diffusivity parameter. When this parameter is scaled to represent turbulent mixing at broad scales the amount fluxed between grid cells with each time-step can be large. Material within each grid cell is assumed to be uniformly mixed so any material moved will travel farther than the true distance based on the flux parameter, causing an accelerated rate of spreading (i.e., numerical diffusion). Reducing the effects of numerical diffusion, within grid-based systems is an active area of research and many techniques have been devised to minimize its impact on concentration gradients (e.g., James 1996; Vested et al. 1996; James 2002). Most involve schemes where the simulation grid is subdivided or the simulation time-step is altered. Many of these schemes are highly technical and involve high computational cost and therefore may not be feasible for high-resolution grids or theoretical studies.

The ESEM method avoids many of the limitations above because it greatly reduces numerical diffusion effects. Through the use of the unique eddy movement methodology I was able to completely eliminate numerical diffusion at scales larger than the grid resolution. The effects of numerical diffusion are eliminated by allowing the eddies to rotate at prescribed frequencies, starting from the outside of the circle and then moving inward. It was necessary to allow a slight temporal delay in the rotation of each eddy and in the cycling of the eddies between periods of rotation and non-rotation but the introduced temporal delay had a negligible effect on the ability to accurately approximate turbulent mixing effects. I was able to preserve the proper cascade of turbulent energy from broad- to fine-scales with the benefit of simulating realistic concentration gradients down to the grid resolution of the simulation platform. Since I am approximating the

turbulent mixing as a field of eddies rather than the direct numerical simulation of turbulence, either through Navier-Stokes equations or diffusive flux based equations, the ESEM offers substantial numerical efficiency in addition to the ability to preserve fine-scale concentration gradients.

The ESEM method does however still require the use of a Fickian diffusion routine to simulate subgrid-scale mixing so the method is not entirely free of numerical diffusion errors. These errors are limited to the grainsize of the simulation platform, which in my case was 100m. This will result in much less artificial spreading than the use of Fickian diffusion to simulate turbulent mixing at larger scales within the simulation domain. The advantage of the ESEM method is that it can be used to simulate turbulent mixing effects at these larger scales where it is inappropriate to use Fickian diffusion based routines.

3.4.2 Limitations, assumptions and potential improvements

The ESEM, as presently formulated, is designed to simulate idealized (i.e., statistical) two-dimensional homogeneous, isotropic turbulence which is in equilibrium with the surrounding environment. When applied to this type of environment, the ESEM is capable of simulating turbulent mixing but the method does have a number of limitations and assumptions which will limit its applicability to other situations or types of turbulent mixing.

The primary assumption is that the turbulence is in steady-state, meaning that the input of energy (i.e., variance) is equal to the output of energy. The method is unable to simulate the development or decay of turbulent mixing (e.g., going from laminar flow to

turbulent flow) and associated eddies. The eddies in the simulation domain are static and do not undergo intensification, decay or movement. The method should only be applied in situations where a fully developed turbulent field is of interest.

The ESEM method, as currently formulated, is limited to homogeneous and isotropic turbulence. This means that the turbulence is space-filling with respect to eddies and contains a full spectrum of eddy sizes so that the energy spectrum follows that predicted by a $-5/3$ spectral signature (Kolmogorov spectrum). Often in natural systems, turbulence is not isotropic and has “intermittency” so that the energy within the turbulent field is not uniform (Holloway 1986). Such intermittency has a number of implications for physical and biological coupling within aquatic systems such as predator/prey interactions, chemical signaling, and shear effects (Seuront and Schmitt 2005). I was unable to incorporate the effects of intermittency within the current formulation but do not see any limitations to adding this feature at a future time.

The method is only capable of simulating two-dimensional turbulence in its current configuration. Since the eddies also interact sequentially (i.e., the eddies take turns rotating within each time-step) and are stationary, the simulation of convergence or divergence (e.g., upwelling or downwelling) effects are not possible. Unfortunately, the incorporation of stratification and boundary layers effects are not possible with the current formulation of the ESEM method. The inclusion of these potentially critical dynamics will not be possible without major revisions in the way the method simulates the movement of the eddies. Currently, there are no plans to address these limitations or to explicitly include a third dimension within the simulation platform.

The ESEM method is also based on seeded eddies and multifractals which introduce some additional constraints. The simulation domain must be a power of two. This is normally not a problem except at larger map sizes as the interval between simulation domains increases exponentially (e.g., 128, 256, 512, 1024, 2048, etc.). The eddy diameters and size also change in prescribed intervals based on theory (Seuront et al. 1999) which is typically not a problem unless irregular eddy distributions and size classes are to be utilized. Finally, the simulation domain must be configured as a square array limiting applications to situations where a regular grid is necessary.

3.4.3 Applications of the ESEM

Even with the assumptions and limitations mentioned above, the ESEM method should have numerous applications for turbulent mixing and ecosystem dynamics in aquatic systems. The main advantage of the method is its ability to introduce realistic fluid effects within an Eulerian framework without many of the problems inherent with Fickian diffusion methods or turbulent mixing parameterizations. The method can simulate turbulent mixing effects over a range of scales (i.e., meters to tens of kilometers) and in many key aspects the method reproduces statistically accurate turbulent mixing.

The technique is extremely flexible in that a wide range of cases, both theoretical and natural, can be considered. For example, the length of the turbulent cascade can be changed. A full cascade is typically simulated (i.e., from a parent eddy down to the grain size of the map) but this cascade can be truncated at one or both ends to simulate conditions where a full turbulent cascade might not apply (e.g., such as where there is an upper to lower limit to the eddy sizes within the turbulent field).

There are also a wide range of turbulence intensities possible with alterations in the time-step or frequency with which the turbulence generation routine is called. Since, a random number routine is used to initialize the eddy field, determine the direction of rotation for each eddy, and eddy rotation start times, Monte Carlo replications are possible. With this capability the effects of the eddy field and the experimental treatment can be investigated systematically. The simulation domain, while not flexible with respect to geometry, can be configured with wrapped or reflecting boundaries for additional options for simulation conditions.

Hydrodynamic models, with routines to simulate fully dynamic turbulence are capable of simulating turbulence directly down to the resolution of the simulation grid. These models have been very useful in studying turbulent mixing effects on physical and biological processes (e.g., McGillicuddy et al. 1995; Franks and Chen 2001; Patel et al. 2004; Grieco et al. 2005; Proehl et al. 2005). The problem with these models is with the grid size and the parameterization (e.g., eddy diffusivity) of turbulent mixing below the grid resolution. This has been done typically with a Fickian diffusion approximation which is best suited for simulating mixing effects at fine spatial or long temporal scales. Hydrodynamic models often lack the resolution necessary (e.g., typically down to km scales) to reach the scales where random mixing processes dominate (i.e., less than meter scales). The ESEM method is a potential bridge to connect these two domains. One application might be to use the ESEM method to simulate intermediate scales (meters to kilometers) through linkages with a hydrodynamic model. If the need for accurate, dynamic flow fields is not necessary then the ESEM method can be used to replace

complex hydrodynamic models with the added advantage that with the computational savings, finer grid resolutions can be utilized.

The ESEM method also has potential application to theoretical and applied studies involving the investigation of turbulent mixing effects on species interactions and spatial patterns in aquatic systems. To date, these studies have been largely limited to approximating turbulent mixing with Fickian diffusion based methods or the use of static methods such as multifractal maps to approximate turbulent fields (e.g., Steele and Henderson 1992; Petrovskii 1999; James 2002; Petrovskii et al. 2002; Ghosal and Mandre 2003; Reigada et al. 2003; Vilar et al. 2003). Potential applications include the investigation of critical patch size dynamics (e.g., “KISS” dynamics), nutrient and pollution transport/mixing studies, and the role of physical and biological dynamics in the maintenance of spatial patterns in aquatic systems. All of these applications depend on the ability to maintain accurate concentration gradients, to simulate scale-dependent diffusion effects on the spreading and transport of a substance, and to approximate the effects of turbulent mixing in a robust dynamic manner. The ability to couple the ESEM method with ecosystem models to investigate physical-biological interactions within a spatially-explicit framework is another key attribute of the method. As an example, in Chapter 4, I couple the ESEM simulation platform to a nutrient-phytoplankton-zooplankton-detritus (NPZD) ecosystem model to gain a better understanding of what processes may be responsible for the observed patterns of nutrients, phytoplankton and zooplankton within the Chesapeake Bay ecosystem.

3.5 Conclusion

In this chapter, I developed a novel technique for simulating idealized two-dimensional turbulence in aquatic systems which is free from many of the limitations inherent with using Fickian diffusion methods to approximate turbulent mixing in aquatic systems. Based on a seeded eddy model developed for particle systems, I adapted the technique to work within a gridded framework while still preserving the cascade of turbulent energy from broad- to fine-scales. The technique is theoretically based, computationally efficient, and statistically accurate. The advantages of this technique are that it is extremely fast, exhibits no numerical diffusion, can simulate scale-dependent mixing and preserves fine-scale concentration gradients. These attributes make the method ideal for incorporation into spatially-explicit ecosystem models for the theoretical and applied investigation of physical-biological interactions over a range of scales, especially where the preservation of spatial patterns is important.

Chapter 4

Response of an NPZD pelagic ecosystem model to spatially and temporally varying nutrient input

Abstract

Spatial and temporal variability of nutrient input is a major driver of ecosystem productivity in estuarine systems. I used a spatially-explicit NPZD (nutrient, phytoplankton, zooplankton, detritus) ecosystem model to investigate how patterns of nutrient input are affected by and interact with physical and biological processes during transit through an estuary. Multifractal map generation techniques were used to create large (30 km²) spatially-extensive patterns of nutrient input to the pelagic water column. A factorial design was then used to vary the mean nutrient concentrations, turbulent mixing and diffusion, and biological processing of the NPZD model. Results were recorded by taking transects across the simulation domain of changes in the mean levels and pattern of model state variables and a suite of ecosystem parameters. Statistical analysis shows that the mean nutrient level is a better predictor of ecosystem dynamics than the level of variability in the nutrient input signal. Changes in spatial patterns were strongly controlled by the variability present in the nutrient input maps and physical mixing processes. Biologically induced pattern formation was observed under elevated nutrient conditions but patterns were rapidly dissipated by turbulent mixing and diffusion. The ability of this framework to isolate physical and biological processes with a spatially-explicit NPZD model provides new insight for understanding the nonlinear response to changing spatial patterns within estuarine ecosystems.

4.1 Introduction

Nutrient loading, primarily in the form of nitrogen, is a major driver of ecosystem function in estuarine systems (Boynton et al. 1982; Nixon and Pilson 1983). Numerous studies have documented changes in nutrient levels and subsequent alteration in key ecosystem properties including changes in nutrient and carbon cycling (Ward 1996), nutrient and carbon export to the benthic community (Kemp et al. 1999), production and respiration (Boynton et al. 1982), and trophic transfer (Glibert 1998). These nutrient induced “bottom-up” effects interact in complex ways to alter patterns of species abundances (Venrick 1982), primary and secondary productivity (Postma et al. 1984) and ultimately the value of ecosystems for human use (e.g., fisheries yield). Excess nutrients in estuarine systems can lead to eutrophication often with negative impacts such as harmful algal blooms, hypoxia, loss of habitat, and alteration in community composition (Ryther and Dunstan 1971; Cloern 2001; Nixon 1995; Malone et al. 1996; Murdoch et al. 1998). Consequently, management of nutrient input levels has become a critical issue for most North American estuarine systems (Rosenberg 1995; Bricker et al. 1999).

Nutrient inputs to estuarine systems typically vary in time and space (Boynton et al. 1995). The effect of nutrient variability on biotic processes depends on both the direct effects of the nutrient concentration gradients, which may overwhelm the nutrient uptake or the assimilation capacity of the ecosystem, or on indirect effects produced from the coupled dynamics of multiple trophic levels (D'Elia et al. 1986; Diaz and Rosenberg 1995; Kemp et al. 2001). Nutrient variability will also interact with the physical environment through diffusive and turbulent mixing processes (Kierstead and Slobodkin

1953; Haury et al. 1978) to influence observed biomass distributions (Gower et al. 1980). Because of these interacting and often non-linear dynamics, characterization of the effects caused by spatial heterogeneity of nutrient inputs may be required to predict changes in population and ecosystem dynamics.

The Chesapeake Bay is one of many estuarine systems around the US which is heavily impacted by elevated nutrient levels (Boynton and Kemp 2000; NRC 200; Kemp et al. 2005). In this system there have been a number of recent efforts undertaken to gain a greater understanding of factors affecting ecosystem productivity and the spatial distribution of resources and organisms. Long-term efforts have included the Chesapeake Bay LMER project called Trophic Interactions in Estuarine Systems (TIES) and the Multiscale Experimental Ecosystem Research Center (MEERC). Both efforts collected extensive data on ecosystem properties and responses of those systems to changes in nutrient inputs. While a great deal has been learned from these studies (Petersen et al. 1997; Petersen et al. 2003), it is difficult to tease apart many of the interacting drivers and processes which can affect an ecosystem's response to nutrient variations (e.g., changes in temperature, light levels, water depth, species composition, predation pressure, mixing, etc).

This chapter presents results from simulation studies designed to elucidate some of the controlling physical and biological processes which may be responsible for determining how spatial and temporal patterns of nutrient inputs affect key ecosystem properties and biomass distributions. I developed a spatially-explicit simulation platform which incorporates an NPZD (nutrient, phytoplankton, zooplankton, detritus) ecosystem model coupled with a novel technique for simulating mixing effects to represent the

biological and physical interactions occurring within an idealized estuarine ecosystem representative of the Chesapeake Bay. Multifractal map generation techniques were used to generate spatially-extensive patterns of nutrient input ($\sim 30 \text{ km}^2$ regions) of the pelagic water column with a spatial resolution (i.e., grain size) of 30 m. A complete factorial design was used to independently vary the concentration levels and heterogeneity of nutrients to simulate conditions ranging from oligotrophic to eutrophic and from fine-scale to broad-scale patterns of nutrient patchiness. Treatment effects on key ecosystem properties and biomass distributions were statistically analyzed for changes in mean concentrations and variability of spatial pattern. The relative influences of physical and biological processes on the resulting spatial patterns of model state variables (e.g., nutrient, phytoplankton, zooplankton, detritus) were also investigated through the sequential variation of each model component. This suite of experimental treatments was designed to reveal pattern-process relationships in the Chesapeake Bay.

4.2 Methods

4.2.1 Simulation Platform

A spatial lattice framework (SLS) was developed to simulate spatial effects in aquatic systems and to extrapolate results from fine-grained experiments to the scales of natural ecosystems. The approach I developed utilizes a gridded aquatic landscape with exchanges among grid sites controlled by sets of difference equations. The bulk flow of constituents and organisms past a fixed point were recorded and rules were used to establish exchange of material, energy, and/or organisms between adjacent grid sites. The advantage of this approach is that the solution technique is fast and efficient,

allowing large spatial extents to be simulated at fine resolution. Because the method is flexible, it is relatively easy to consider different biological and physical components within the simulation framework. The framework also allows for the sequential consideration of processes—while holding other key variables constant—to evaluate the relative impact of biological and physical variables on model response. Please see Chapter 1 for additional information about the simulation platform.

Biological Component: A fine-scale aquatic ecosystem model was embedded within each grid cell of the SLS platform simulating the dynamics of nutrients, phytoplankton, zooplankton, and detritus within the upper water column. The model was calibrated to the stratified summer conditions of the Chesapeake Bay with rate constants based on values published for similar models and environmental conditions. The model was adjusted to insure stable dynamics (i.e., no oscillations or predator/prey cycles) and equilibrium biomass and concentration levels that might be observed during the summer months in the Chesapeake Bay. While the model has not been tested against independent field data, a sensitivity analysis of the model indicated that it adequately represents many of the key biological processes, interactions, and responses occurring in estuarine systems. For a complete description of the model, equations, parameter values, and sensitivity analysis please see Chapter 2.

Physical Component: Since the simulation platform was based on a Eulerian framework at fine spatial resolutions, I was limited in the types of methods that could incorporate realistic physical mixing between adjacent biological models at larger scales. Complex flow patterns are a particular challenge because the full equations of motion are not considered by the SLS. To avoid these limitations, a seeded eddy model developed

for particle systems (Dyke and Robertson 1985; Abraham 1998) was adapted for the SLS to simulate idealized turbulence mixing within the simulation domain. The technique is theoretically based, computationally efficient, statistically accurate and simulates scale-dependent mixing while preserving fine-scale concentration gradients. When combined with a diffusive mixing routine to simulate subgrid mixing, this method has the added advantage of being able to introduce realistic fluid effects without having to resort to complex hydrological models or oversimplifications of turbulent mixing. Additional information about the physical mixing component of the platform can be found in Chapter 3.

4.2.2 Multifractal Map Generation

Multifractal map generation techniques were used to create realistic, spatially-extensive patterns of nutrient input to the simulated estuarine system. A special program was developed using the mid-point displacement method (Saupe 1998) to generate a series of two-dimensional maps of spatially varying nutrient inputs. The algorithm can simulate a range of structures from nearly random to strongly correlated spatial structures that are similar to those observed in natural systems. Because the mid-point displacement method allows the spatial patterns to be varied independently of the mean concentration levels, this method was ideally suited to the simulation I wished to perform. Variations in H control the correlation structure of the maps with high H values producing “smooth” maps and low H values producing “rough” maps. This allows a wide range of maps to be produced which different degrees of “patchiness” through changes in H value. Examples of generated maps can be found in the results section.

4.2.3 Simulation Domain and Parameters

The simulation domain was configured to approximate a stationary section of the pelagic mesohaline Chesapeake Bay. The dimensions were 30 km on a side and within the domain was a gridded-framework of one million individual square cells of 30 m. These dimensions were selected to approximate the width of the Chesapeake Bay and to correspond in resolution to data collected for the TIES program. This configuration also allowed examination of changes in the simulated spatial distributions of model state variables over a range of scales covering three orders of magnitude. Table 4.1 provides a summary of the simulation parameters with additional information about these parameters and the simulation domain described below.

Rather than a stationary simulation domain, material within the domain was advected from north to south to reflect a flow through system typical of net estuarine transport. Constituents (i.e., nutrients and biomass variables) entered the northern edge of the domain (i.e., map), traveled down the map and then were exported at the southern edge of the domain. To approximate the fixed boundaries of a shoreline, the western and eastern edges of the map were configured to have reflecting boundaries. By utilizing a flow-through simulation I was able to evaluate the combined effects of changes in spatial and temporal patterns of nutrient inputs, follow individual transects as they move through the simulation domain, examine temporal changes in the state variables and better approximate dynamics within an estuarine environment (see Figure 4.1). A fixed-frame simulation platform did not allow this flexibility.

Table 4.1. Description and nominal values of key simulation parameters. See text for additional details.

Description	Value	Units
Simulation Domain Extent	~30	km ²
Simulation Domain Grainsize	30	m
Cell Number	1024	per side
Simulation Length	11	days
Time-step	5	min
Residual advection velocity	3.3	cm/sec
Turbulence Intensity	~1.0	cm/sec
Eddy Diffusivity	50	cm ² /sec
Residence Time Cell	15	min
Residence Time Map	~11	days
Transects Along Map	21	unitless
Spacing Between Transects	1.44	km

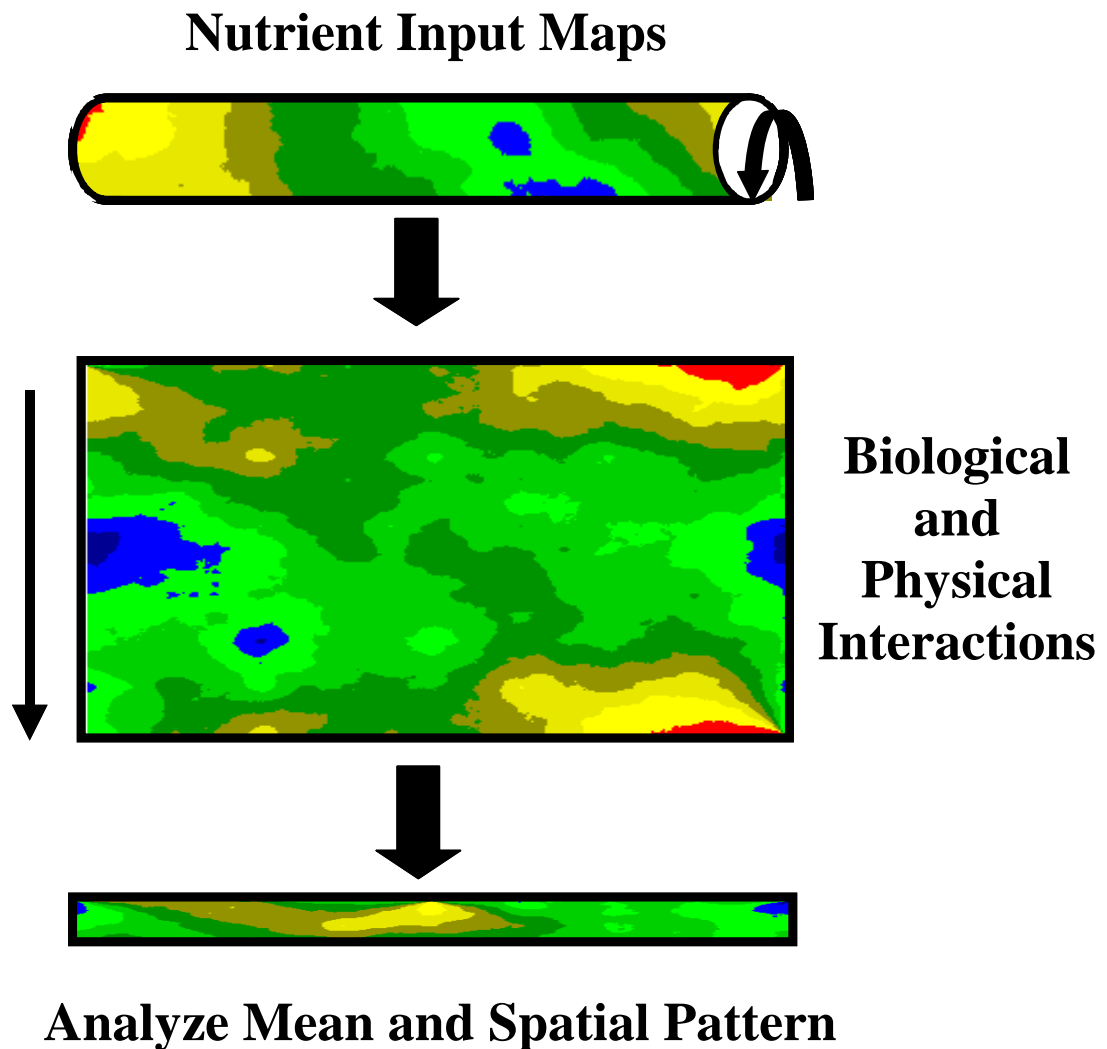


Figure 4.1. Schematic showing the flow through system utilized to simulate an idealized estuarine ecosystem. Nutrient input maps are fed into the simulation domain at the northern boundary which then interacts with the various physical and biological components until exiting the system at the southern edge. Changes in the mean and pattern of selected variables are measured at 21 locations down the simulation domain.

The advection speed for the simulations was set at 3.3 cm/s. For a substance to traverse the length of the map took approximately 11 days. This residual advection velocity was set to allow adequate time for physical and biological dynamics to occur before being advected beyond the model domain. As material moves through the model domain it interacts with the biological components of the model and is mixed by the physical processes of diffusion and turbulence. The turbulence intensity was set at 1.0 cm/s, which is within the range of turbulence intensities found in the Chesapeake Bay (Petersen et al. 1998). The scale of turbulent mixing was limited to the resolution of the grid with subgrid mixing determined by an eddy diffusivity of 50 cm²/sec. This value is also within the range found in aquatic environments at the grain size of the model (Okubo 1980). These parameters describing turbulent and diffusive mixing were held constant for all scenarios that included these physical processes. The solution step (i.e., time-step) used to solve the differential equations for the model state variables and to regulate the advection, turbulence, and diffusion routines was set to five minutes. The simulations were started after the nutrient inputs maps had completed one cycle through the simulation domain which allowed adequate time to initialize the model before the start of each experiment.

The initial values for the four state variables (i.e., nutrient, N , phytoplankton, P , zooplankton, Z , detritus, D) of the biological model are as follows: $N = 0.0043 \text{ gN/m}^3$, $P = 0.22 \text{ gC/m}^3$, $Z = 0.02 \text{ gC/m}^3$, $D = 0.08 \text{ gC/m}^3$. These values were determined by conducting a non-spatial simulation of the model without any external nutrient inputs and allowing the simulation to run until equilibrium conditions were achieved. These initial values then became the input values for the P , Z , and D state variables at the top of the

simulation domain (i.e., uniform input values across the width of the simulation domain). For the N state variable, I generated multifractal nutrient maps and used these spatial patterns as input to the simulation domain.

4.2.4 Experimental Design

Multifractal map generation techniques were used to create input maps with realistic, spatially-extensive nutrient patterns (see section 4.2.2). The concentration levels and heterogeneity of these nutrient input maps were varied to simulate a range of conditions from oligotrophic to eutrophic and to vary the spatial distributions of nutrient input (Figure 4.2). Three levels of mean nutrient input were simulated: low (0.01 gN/m³), medium (0.1 gN/m³), or high (1.0 gN/m³) ambient nutrient concentration. The spatial heterogeneity (i.e., patchiness) of each map was also varied over three levels from low ($H=0.99$), med ($H=0.60$) and high ($H=0.01$) levels of patchiness. Variations in H control the correlation structure of the maps with high H values producing “smooth” maps and low H values producing “rough” maps. Variation in the mean level of nutrients and the spatial variation of nutrient levels allowed these two effects to be systematically and independently evaluated.

A series of model cases were run varying the physical and biological processes in the SLS. The various combinations are as follows: Case A (control) was simply the nutrient input map (no biology or physical mixing); Case B (Control + Diffusion); Case C (Control + Turbulence); Case D (Control + Diffusion + Turbulence); Case E (Control + Biology); Case F (Control + Biology + Diffusion); Case G (Control + Biology + Turbulence); Case H (Control + Biology + Diffusion + Turbulence). Case A, the

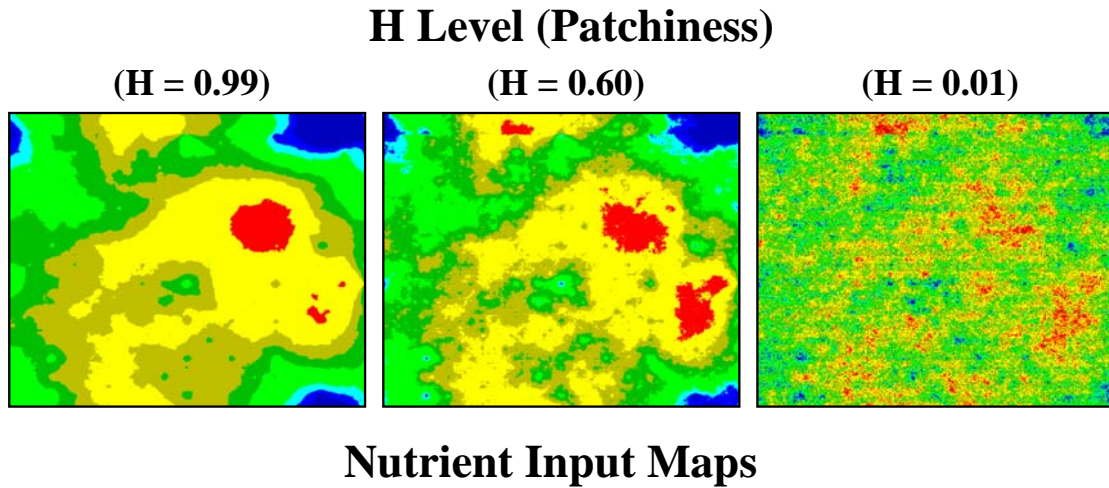


Figure 4.2. Nutrient input maps used in the model simulations. The maps shown all have the same mean nutrient concentration but differ in the amount of patchiness present (i.e., H level). The color scale goes from low (blue) to high (red) values. Three mean nutrient levels for the input maps were utilized for each H level.

“control case”, provides a baseline to which the effects of the various treatments (changes in nutrient level, level of patchiness, and model structure) can be compared. Cases B-D test the impacts of the physical components of the model without any biological effects; Case E tests the biological component without any physical effects; while Case H is the full model with all biological and physical processes active. Thus, there are 72 treatment combinations for the full factorial (three nutrient level treatments x three patchiness level treatments x eight model cases, Table 4.2).

4.2.5 Statistical Analysis

Model predictions for each treatment combination were evaluated by transects across the width of the simulation domain at 21 equally spaced locations, roughly 1.4 km apart. Samples were repeated every 12 hours resulting in 22 replicates for each transect over the 11-day simulation (a total of $21 \times 22 = 462$ transects). The mean concentration of each state variable (i.e., N , P , Z , D) and a suite of six ecosystem parameters were recorded for each transect. These included phytoplankton and zooplankton growth rates, phytoplankton and zooplankton production amounts and the amount of total biomass transferred to higher trophic levels and the benthos.

As a measure of the spatial variability (i.e., patchiness) the Proc Spectral routine of SAS (SAS 2001) was used to conduct a spectral analysis of each transect collected for the four state variables. The method works by dividing the data series into a range of sines and cosines of decreasing periods where the amplitude gives the intensity of variability at a particular period. The spectrum is created by graphing the log of the spectral density (the variance at each frequency) versus the log of the frequency or wave

Table 4.2. Description of the experimental treatments utilized in the model simulations. A 3x3x8 factorial design was employed to vary the nutrient input level, H level, and model case over a range of conditions to simulate oligotrophic to eutrophic conditions, high levels to low levels of patchiness, and the interaction of biological and physical processes. See text for additional details.

<i>Nutrient Input Level</i> (gN m ⁻³ day ⁻¹)	H Level (Patchiness)	Model Case (Physical/Biological Properties)
0.01	0.01	A - (Input Map)
0.1	0.66	B - (Input Map + Diffusion)
1.0	0.99	C - (Input Map + Turbulence)
		D - (Input Map + Diffusion + Turbulence)
		E - (Biology)
		F - (Biology + Diffusion)
		G - (Biology + Turbulence)
		H - (Biology + Diffusion + Turbulence)

number. From these spectra I calculated a spectral slope (i.e., Beta) by conducting a linear regression of the spectrum.

A General Linear Model (GLM) of SAS (SAS 2001) was used to statistically identify the effects of different treatment combinations (i.e., nutrient level, *H* level, model case) on the mean levels of the variables measured by each transect (i.e., changes in ecosystem parameters or in the level of patchiness). These data and methods of analysis allow three questions to be investigated:

Question 1: Does patchiness in nutrient inputs affect ecosystem function?

Question 2: Does patchiness in nutrient inputs affect spatial distributions?

Question 3: Can biological dynamics overcome the influence of physical mixing?

The questions above were chosen because of their direct relevance to the management of estuarine ecosystems many of which are now suffering from excess nutrient inputs (Turner and Rabalais 1994; Howarth et al. 1996). Through the proposed studies I hoped to gain a better understanding of the nonlinear response of estuarine systems to changing spatial patterns, the identification of conditions under which spatial dynamics-interactions are important, and what processes may be responsible for the observed spatial patterns found in estuarine systems such as Chesapeake Bay.

4.3 Results

4.3.1 Qualitative response to experimental treatments

The spatial dynamics of the model simulations are illustrated in Figures 4.3-4.5. The characteristic patterns produced by variation in the nutrient level entering the simulation domain (via the multifractal input maps) are illustrated in Figure 4.3 for each state variable (N , P , Z , D). Elevated nutrient levels produced phytoplankton blooms followed by gradual declines once nutrients become exhausted. Elevated phytoplankton concentrations enabled higher zooplankton and detritus levels. These responses were spatially lagged, becoming more pronounced in size and transported further down the simulation domain with higher nutrient levels (see A-D of Figure 4.3). Changes caused by variations in H (i.e., pattern) under medium nutrient levels are illustrated in Figure 4.4. Decreases in H caused finer spatial patterns to be evident for all four state variables with little impact on mean biomass concentrations as was seen in Figure 4.3.

Changes caused by variations in model case (i.e., presence or absence of diffusion, turbulence or biotic processes) for a given nutrient and H is illustrated in Figure 4.5. Highlighted in the physical series (cases A-D) is the effect of mixing on the pattern of input defined by the multifractal maps. The E-H cases include the added effects of biological dynamics as represented by the phytoplankton state variable. Both Case A (no mixing, no biology) and E (biology but no mixing) are reference cases against which the effects of mixing and biology were compared. Diffusive mixing tends to smooth out spatial pattern and turbulence adds additional heterogeneity. With both processes operating and intermediate pattern between these two extremes is created (see Figure 4.5).

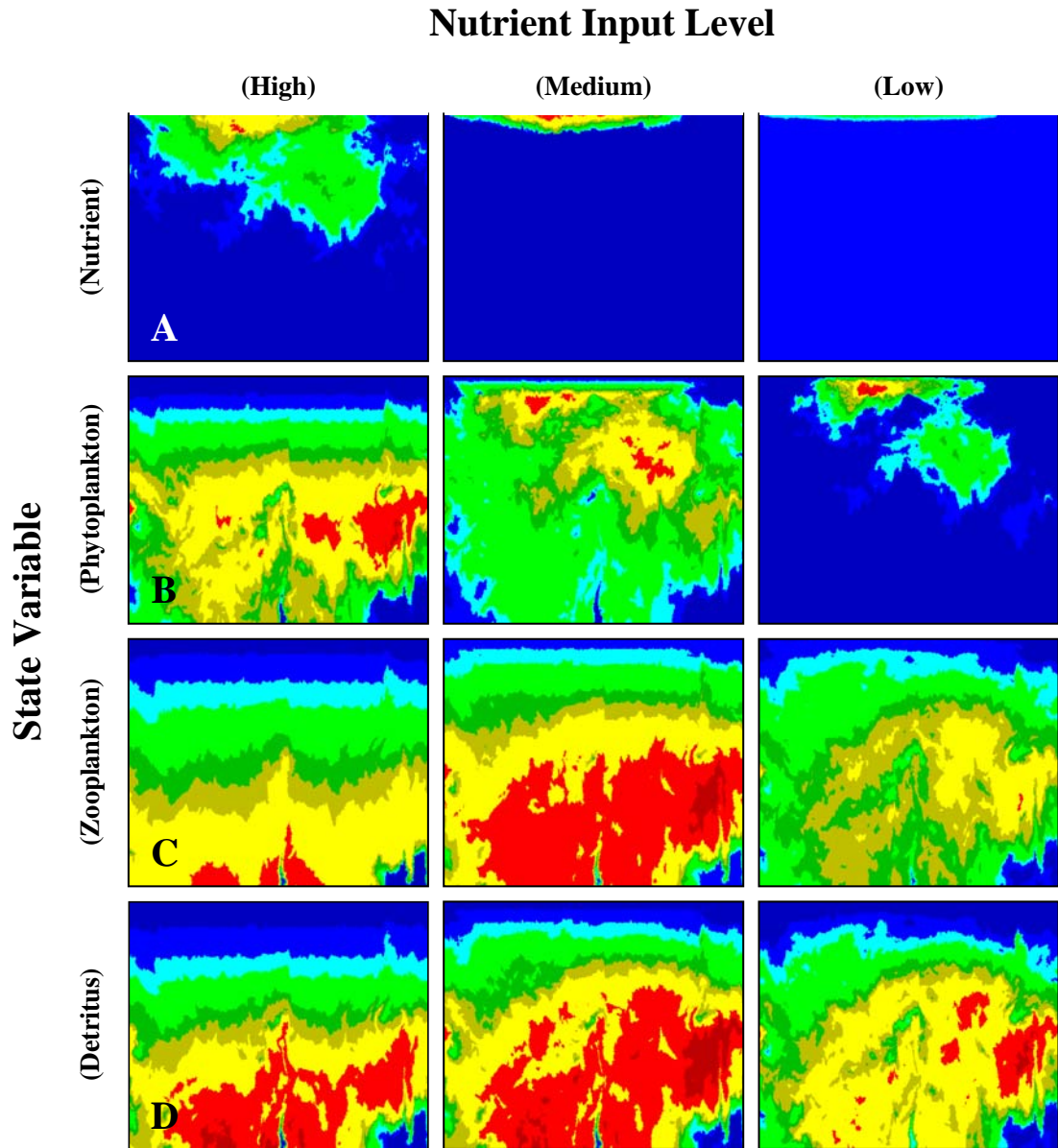


Figure 4.3. A snapshot of typical ecosystem dynamics for the nutrient (N), phytoplankton (P), zooplankton (Z), and detritus (D) state variables under the three nutrient input levels tested. The patchiness level is medium ($H = 0.60$) and the model case is the full model with all physical and biological processes active. The color scale (range) is set to maximize the observed pattern within each map as follows (gm^{-3}):
 Nutrient Input Level High ($N:0.0-0.015$, $P:0.4-0.46$, $Z:0.55-0.065$, $D:0.16-0.18$);
 Nutrient Input Level Medium ($N:0.0-0.015$, $P:0.4-1.20$, $Z:0.55-0.095$, $D:0.16-0.30$);
 Nutrient Input Level Low ($N:0.0-0.020$, $P:0.4-7.00$, $Z:0.55-0.155$, $D:0.16-1.10$).

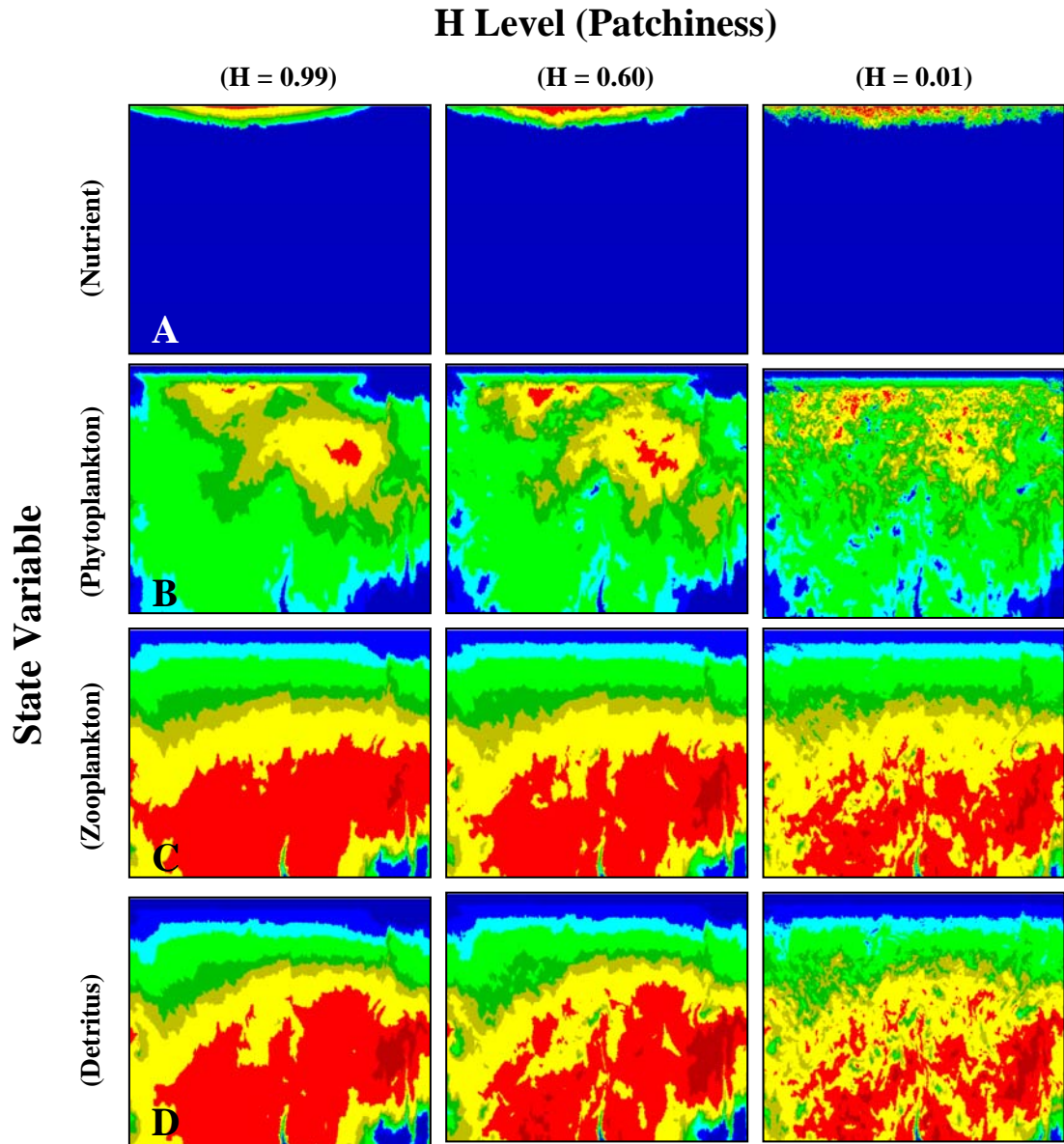


Figure 4.4. A snapshot of typical ecosystem dynamics for the nutrient (N), phytoplankton (P), zooplankton (Z), and detritus (D) state variables under the three H levels tested. The nutrient level is medium (0.1) and the model case is the full model with all physical and biological processes active. The color scale (range) is set to maximize the observed pattern within each map as follows (gm^{-3}): (N :0.0-0.015, P :0.4-1.20, Z :0.55-0.095, D :0.16-0.30).

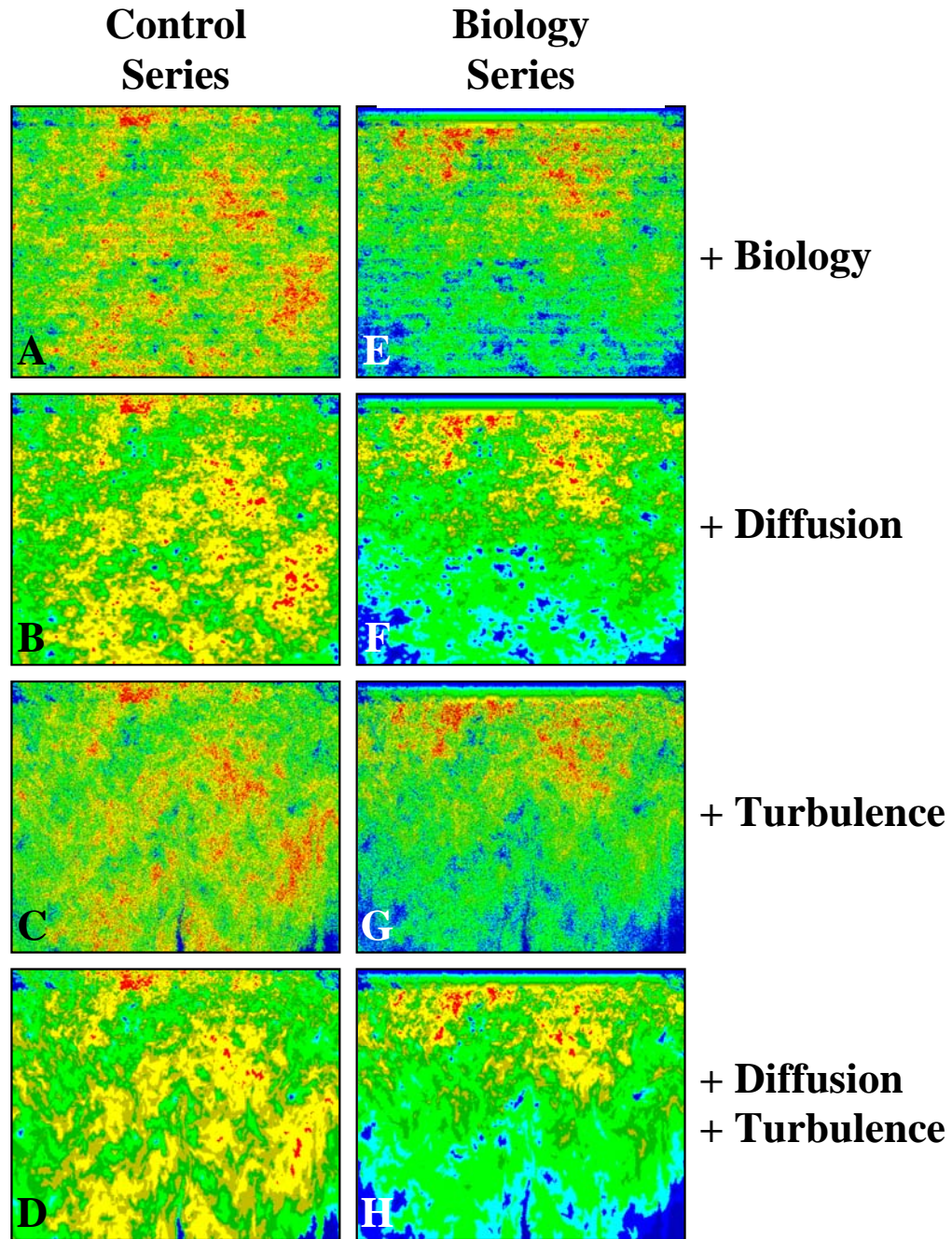


Figure 4.5. A snapshot illustrating the impact of the model case on concentration and spatial pattern. Figures A-D highlight the physical effects (i.e., control series, no biology) while figures E-H highlight the additional effects of biological dynamics (i.e., biology series). For the control series the nutrient state variable is used as a tracer and the phytoplankton state variable is highlighted for the biology series. The nutrient input (0.1) and H (0.66) level are both set to medium. The color scale (range) is set to maximize the observed pattern within each map as follows (gm^{-3}): ($N:0.0-0.2$, $P:0.4-1.20$).

4.3.2 Mean Effects

A primary question of this chapter was to determine if the patchiness generated by changes in the H of the nutrient input maps (see Figure 4.4) would result in significant changes to the mean level of each state variable (i.e., N, P, Z, D) and a suite of six ecosystem parameters (i.e., phytoplankton and zooplankton growth rates, phytoplankton and zooplankton production amount and the amount of total biomass transferred to higher trophic levels and the benthos). Table 4.3 shows the results of the GLM analysis for change in mean phytoplankton biomass at three representative locations corresponding to the start, middle, and end of the simulation.

The nutrient level treatment was highly significant ($P < 0.0001$) for all three locations, accounting for increasing levels of the variation with distance. The effects of H (the spatial pattern of nutrient input) and model case did not have a significant effect on mean phytoplankton levels over a majority of the treatment combinations (data not shown). When H level or case level were found to be significant the amount of variation associated with these variables was always small (i.e., a low R^2). These results are consistent for each of the other nine response variables tested and over the 21 map locations (data not shown).

4.3.3 Pattern Effects

The second question addressed was whether the spatial distributions of the model state variables would be significantly affected by the experimental treatments. The impact of experimental treatments on spatial patterns was assessed by spectral analysis which provides a means of measuring the change in variance (i.e., patchiness) with scale

Table 4.3. Results of GLM for the main effects of nutrient level, H level, and model case on the mean concentration found at location 1, 11, and 21 of the phytoplankton state variable. At each location there were 22 replications for each of the nutrient, H and model case (3x3x4 factorial). Only the nutrient treatment was found to be significant (Pr>F) and important (R-Square) over the conditions tested.

Location 1				
Effect ²	df	F Value	Pr > F	R-Square
Nutrient Level	1	395.18	<.0001	0.33
Patchiness Level	1	0.62	0.4316	<.01
Model Case ¹	3	0.08	0.9716	<.01

Location 11				
Effect ²	df	F Value	Pr > F	R-Square
Nutrient Level	1	35815.6	<.0001	0.97
Patchiness Level	1	0.04	0.8499	<.01
Model Case ¹	3	0.79	0.4992	<.01

Location 21				
Effect ²	df	F Value	Pr > F	R-Square
Nutrient Level	1	16709.9	<.0001	0.97
Patchiness Level	1	1.08	0.2999	<.01
Model Case ¹	3	0.17	0.9171	<.01

¹Only 4 model cases were run (E-H)

²MSE = 0.003943, 0.0696, 0.0950, for location 1,11, and 21 respectively with 786 df

(section 4.2.5). The results indicated that a “break” in the spectral signature occurred at approximately the one kilometer scale. Figure 4.6 illustrates this impact for a series of model cases without (A-D) and with (E-H) biological dynamics. Turbulence preferentially caused the variance to increase while diffusion caused the variance to decrease at scales less than one kilometer. Due to the way the spectral analysis is calculated (i.e., with more data points at finer frequencies), the observed flattening in the relationship between variance and frequency biases the estimation of the spectral slope when calculated over the full spectral signature. Therefore, I calculated the spectral slope on only the linear part of the line roughly corresponding to data to the left of the scale break.

Table 4.4 illustrates the general results of the GLM analysis of the change in the mean spectral slope of phytoplankton biomass at map location 11 (mid-way down the simulation domain). The sum of squares for each treatment (biology, diffusion, and turbulence and their interaction terms) are separately defined in Table 4.4. The results show that H , diffusion, and turbulence were all significant at $P < 0.001$ and also explained a large fraction of the total variation in simulation results ($R^2 > 0.1$). These effects were consistent across the other state variables (i.e., N , Z , D) and map locations (data not shown). Variation in nutrient levels and biology were statistically significant, but the effect of these variables declined with distance (data not shown). All of the other treatment variables were almost always non-significant.

Figure 4.7 illustrates the changes in variance explained (i.e., the R^2 estimated from the partial sums of squares) for each of the treatments and state variables through time. Most of the variability was attributed to changes in H and the processes of

Table 4.4. Results of GLM for the main and interaction effects of nutrient level, H level, and model case on the mean spectral slope for the phytoplankton state variable at map location 11. For this location there were 22 replications for each of the nutrient, H and model cases (3x3x8 factorial).

Spectral Slope				
Effect ¹	df	F Value	Pr > F	R-Square
Nutrient Level	1	13.55	0.0002	<0.01
H Level	1	6014.9	<.0001	0.59
Model Case	7			
Biology	(1)	2.79	0.0951	<0.01
Diffusion	(1)	929.04	<.0001	0.09
Turbulence	(1)	1588.27	<.0001	0.16
Biology*Diffusion	(1)	0.61	0.434	<0.01
Biology*Turbulence	(1)	1.13	0.2872	<0.01
Diffusion*Turbulence	(1)	1.94	0.1644	<0.01
Biology*Diffusion*Turbulence	(1)	1.09	0.297	<0.01

¹MSE = 0.1085 for the Spectral slope variable with 1574 df

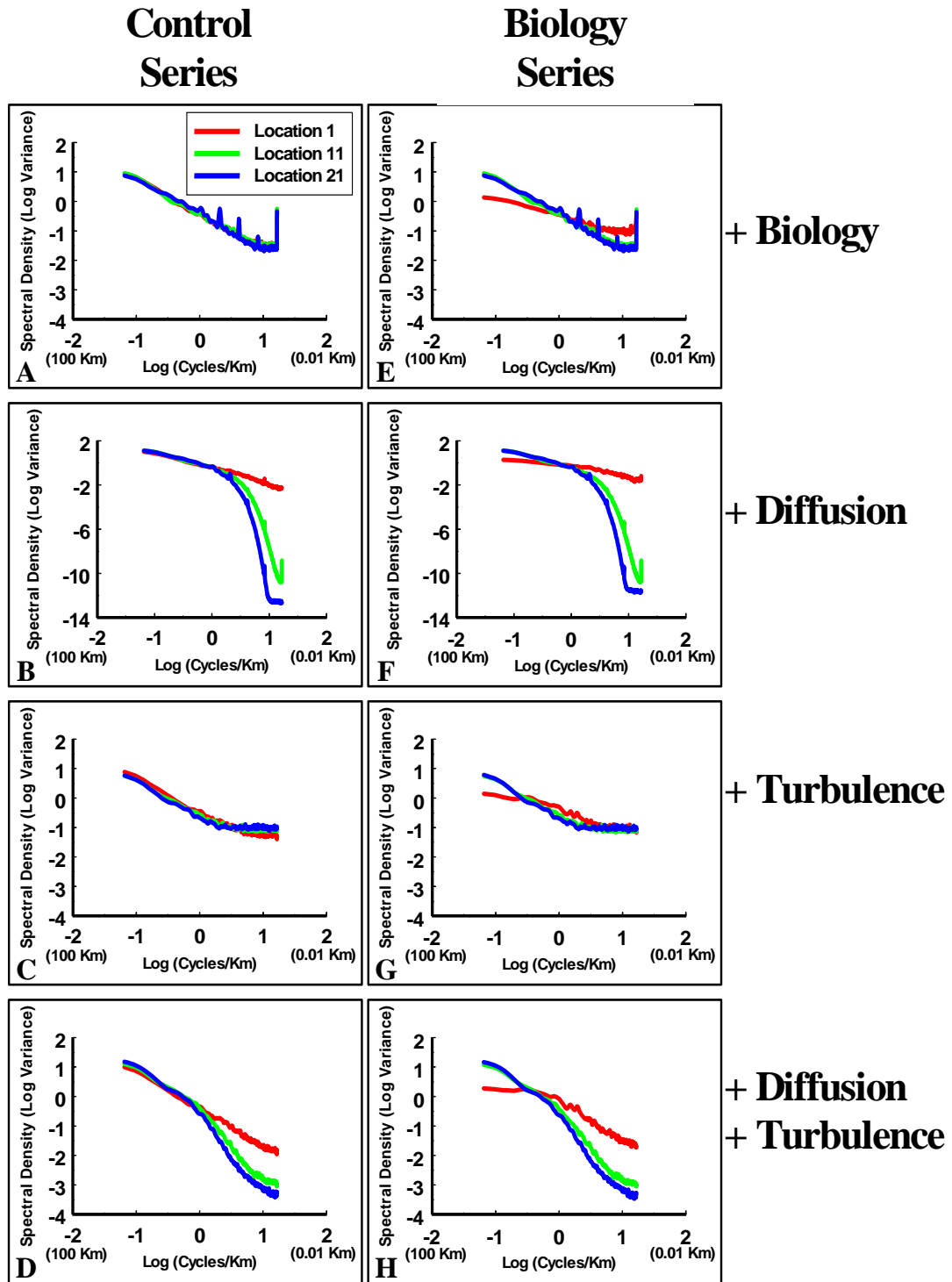


Figure 4.6. Series of graphs illustrating the impact of the model case on spatial pattern at three locations down the simulation domain. Pattern is measured as the average spectral signature of 22 transects at each location. Figures A-D highlight the physical effects (i.e., control series, no biology) while figures E-H highlight the additional effects of biological dynamics (i.e., biology series). For the control series the nutrient state variable is used as a tracer and the phytoplankton state variable is highlighted for the biology series. The nutrient input and H levels are set to medium (0.1) and low (0.01) respectively.

Spectral Slope

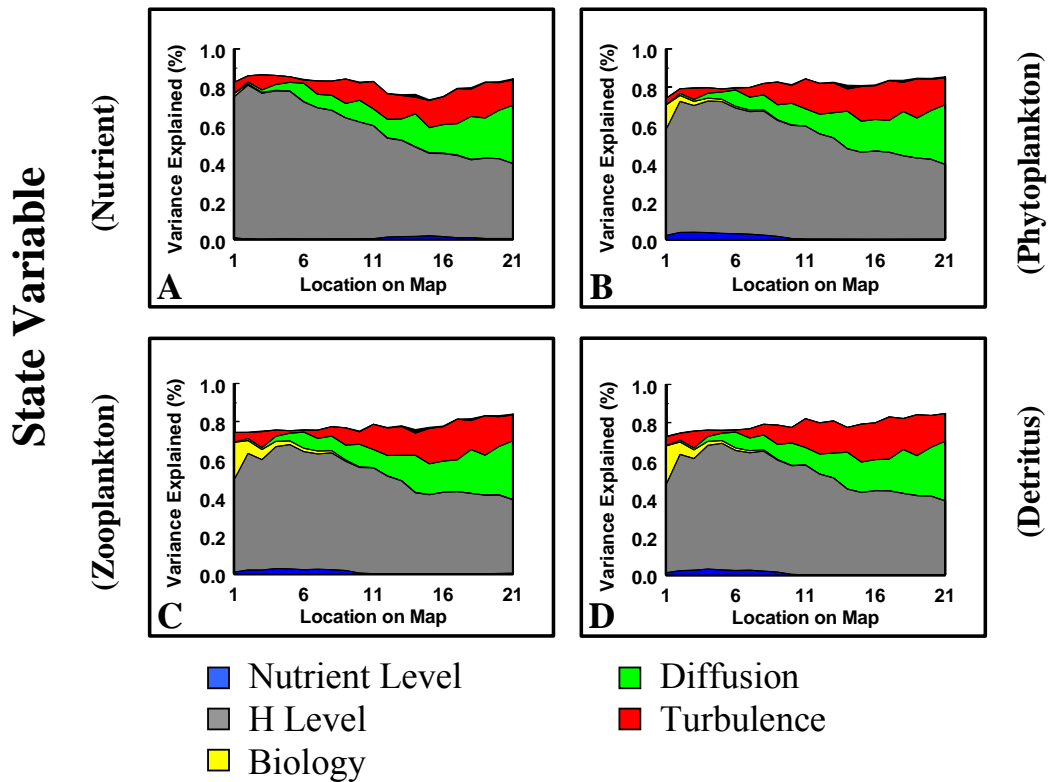


Figure 4.7. Amount of variance explained (i.e., partial r-squared) in the spectral slope response variable for each of the main treatment effects in each state variable and map location down the simulation domain. The model case has been broken out into three sub-levels (i.e., Biology, Diffusion, Turbulence). See text for additional details.

diffusion and turbulence. Over time the initially high importance of H declines and is replaced by increased importance of turbulence and diffusion. These physical mixing processes are approximately equally important in controlling pattern formation within the simulation framework. Across all the conditions tested neither biology nor nutrient input statistically affected patterns measured by spectral analysis.

4.3.4 Trophic Effects/Physical and biological interactions

The third question addressed was whether the spatial distributions of the model state variables could overcome the influence of the physical mixing processes as seen in the previous section. To evaluate, I examined the change in spectral slope as a function of H and nutrient level. The full model case was run (with all physical and biological processes operating) as these conditions would most closely match dynamics occurring in natural systems.

Figure 4.8 highlights representative results of the analysis which focused on the endpoints of the H and nutrient level treatments (i.e., high and low nutrient level and high and low H level). Illustrated are changes in the spectral slope over time for each state variable with a control line corresponding to the equivalent model case without any biological processes (i.e., only physical processes). The difference between the control line spectral slope and the state variable line spectral slope, identify potential areas where biological dynamics are affecting the simulated patterns as opposed to physical processes.

The greatest difference between the control line and the lines for the state variables (and hence impact of biological processes on spatial patterns) was found in Figures 4.8c and 4.8d. These are the high nutrient cases. This decoupling is strongly

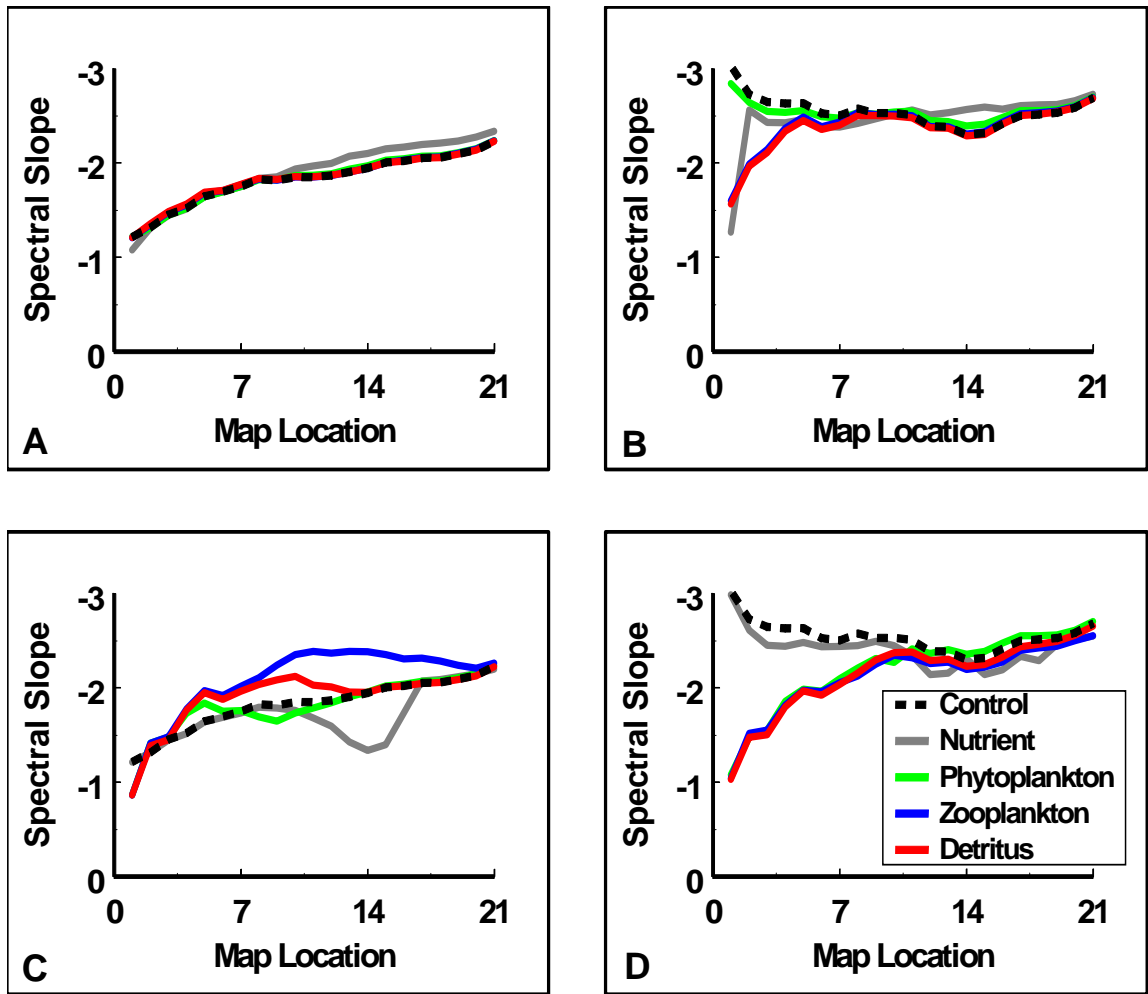


Figure 4.8. Series of graphs illustrating the change in spectral slope for each state variable and map location across nutrient input and H levels under the full model case (i.e., full model with all processes included). The dotted back line is the corresponding control case with the biological dynamics turned off. Nutrient level for each graph is as follows (A=0.01, B=0.01, C=1.0, D=1.0). H level for each graph is as follows: (A=0.01, B=0.99, C=0.01, D=0.99).

evident by the dramatic decreases and increases in the spectral slopes of the state variables which sometimes deviated from the control line together (Figure 4.8d) and sometimes in separate directions (Figure 4.8c) with changes in the H level of the map. By the end of the simulation (i.e., bottom of the map) all of the state variables were tracking the control signal in that they had similar spectral slopes. Under the low nutrient combinations (Figure 4.8a and 4.8b) there was little separation between the spectral slopes of the control and state variable lines indicating that their distributions were largely controlled by the pattern generated through the physical mixing processes operating in the model. Overall, the spectral slope tended to increase with time.

4.4 Discussion

4.4.1 Simulation Platform and Overall Model Dynamics

Previous studies have found that spatial and temporal heterogeneity in nutrient input is a key factor affecting ecosystem productivity and biomass distributions in estuarine systems (Baird et al. 1995; Smith and Kemp 1995; Yeager et al. 2005). It is, therefore, critical to understand how nutrient inputs are processed and, in turn, affect biomass distributions within these ecosystems. The complex effects of mixing (e.g., turbulent mixing, advection, stratification), biological uptake, and environmental forcing (e.g. temperature, solar radiation) on ecosystem response in time and space can be difficult to determine. My simulation platform allows these physical-biological interactions to be separately examined within a spatially-explicit framework, limiting the impact of confounding factors usually present in field studies or experimental ecosystems.

The simulation platform (SLS) is coupled to an NPZD (nutrient, phytoplankton, zooplankton, detritus) ecosystem model and physical transport/mixing routines that simulate advection, idealized turbulent mixing, and diffusive processes. When forced with multifractal maps, that represented spatial variation in nutrient input, spatially-extensive patterns of ecosystem dynamics and interaction were generated. Similar studies are possible with existing coupled biophysical ecosystem models (e.g., Wang et al. 2006) but it is the combined integrated use of the various components (i.e., nutrient input maps, physical mixing routines, biological models, statistical analysis) that make my approach unique and attractive as a potential alternative, to assess change at fine scales and broad extents.

Despite the simplicity of the SLS, I was able to simulate spatial dynamics within an idealized estuarine ecosystem over a broad range of nutrient input scenarios from oligotrophic to eutrophic and from fine-scale to broad-scale patchiness. The biological model responded appropriately to elevated nutrient levels as demonstrated by the uptake, bloom, predation, and decay dynamics of the *N*, *P*, *Z* and *D* state variables (see Figure 4.3 and 4.4). The magnitude (i.e., mean concentrations) and separation in space of the responses were typical of what might be expected for estuarine ecosystems under similar nutrient and environmental conditions. The physical processes operating in the SLS were also able to capture the general aspects of turbulent mixing (i.e., increased heterogeneity) and diffusion (i.e., decreased heterogeneity) which are typically operating in estuarine systems over a 30 m to 30 km scale range (see Figure 4.5 and 4.6). Given these encouraging results I believe that the SLS can address the questions posed regarding the effects of heterogeneity of nutrient input on ecosystem function and spatial patterns.

4.4.2 *Question 1: Does patchiness in nutrient inputs affect ecosystem function?*

A central purpose of these simulations was determination of the effect of the spatial pattern (i.e. patchiness) of nutrient inputs on ecosystem behavior. Nutrient input levels are, of course, a key driver of ecosystem dynamics (Kemp et al. 2005). I wished to investigate whether the variability of nutrient inputs (i.e., changes in mean levels) or the heterogeneity of nutrient inputs (holding mean level constant) had equivalent effects on simulated estuarine ecosystem response. Specifically, would local variability have no lasting broad-scale effect, or would this variability propagate through time and space, significantly changing ecosystem dynamics? The experiments reported here provided insight into the conditions under which these changes in the level and heterogeneity of nutrients can be examined.

Over the range of conditions reported here there was no significant effect of patchiness of nutrient input on ecosystem function. None of the mean values for the four state variables and suite of six ecosystem variables (i.e., phytoplankton and zooplankton growth rates, phytoplankton and zooplankton production amount and the amount of total biomass transferred to higher trophic levels and the benthos) were sensitive to changes in the degree of patchiness of nutrient inputs. No significant effects were found even under optimal conditions of varying the level of patchiness under the highest nutrient levels or at locations where rapid growth was occurring. Only changes in nutrient input level (independent of patchiness) were found to have a significant impact on mean levels of the state variables and ecosystem variables. This result was expected since the nutrient input

level was varied over three orders of magnitude allowing higher production and corresponding greater changes to biomass production and ecosystem dynamics.

Numerous studies, both empirical and modeling, have described the effect of nutrient input levels on ecosystem dynamics (D'Elia et al. 1986; Chen et al. 2000; Kemp et al. 2001). Many of these studies reported change as a function of variation in the mean level of nutrient input. Although my results produced similar changes in biomass, primary and secondary productivity, and export to higher trophic levels and to the benthos, my research is the first to attempt to isolate the effects of spatial heterogeneity in nutrient inputs on estuarine ecosystem dynamics. Within a given nutrient input level however, changes in the pattern of nutrient input produced no significant differences in mean levels over the conditions tested in the simulations.

The lack of a spatial effect was surprising because others studies have found that the patchiness of inputs does affect ecosystem dynamics especially phytoplankton blooms, predator/prey interactions, feeding, and survival dynamics (Denman and Powell 1984; Strass 1992; Abbot 1993). The different outcome could be due to the analysis method I employed. In my simulations, I tested for change by averaging across a transect at selected locations down the simulation domain. Nutrient input values along the transect may vary widely, yet average to a constant mean level, obscuring the relationship between spatial patterns and model response. Averaging by horizontal transects would underestimate the impact of large patches traveling down the simulation domain.

Another complication which may obscure potential response could be due to the type of simulation framework employed. Because the model domain was designed to be an open flow-through system any differences in patchiness were either rapidly mixed

(through turbulence), diluted (through diffusion) or ephemeral in nature (removed by nutrient uptake by phytoplankton) and thus, did not have any lasting impacts. Under the conditions simulated, there were no means of creating patterns fixed in space or persistent in time via differences in depth, convergence and divergence zones, frontal zone, and behavior dynamics. These unconsidered effects can have a pronounced effect on ecosystem properties (Okubo 1986; Franks 1992; Hood et al. 1999; Roman et al. 2005).

Given the caveats, results indicate that the heterogeneity of nutrient inputs, in a flow-through open system, will not of itself cause persistent changes in ecosystem dynamics. Even though changes in the pattern of nutrient inputs resulted in changes in the distributions (i.e., spatial patterns) of the state variables in the model, it may only be necessary to know the mean level of nutrients entering the estuary to predict ecosystem response. The wide range of conditions tested within our simulation framework, consisting of over a three fold nutrient loading, a wide range of patchiness levels, and idealized mixing processes, suggests that these results may be robust. Additional experiments would be necessary to ensure that the issues discussed in the previous paragraphs do not invalidate my preliminary results presented here.

4.4.3 Question 2: Does patchiness in nutrient inputs affect spatial distributions?

Material and organisms in the aquatic environment are rarely uniform across space, but tend to be patchy in nature (Hardy and Gunther 1935; Mackas et al. 1985). This patchiness is caused by a number of physical and biological processes which interact at a number of scales to influence ecosystem dynamics (Denman and Powell 1984). I wished to investigate whether heterogeneity of the nutrient input signal would also affect

the patchiness of other components of the ecosystem, e.g., the nutrient, phytoplankton, zooplankton, and detritus distributions. Impacts of spatial heterogeneity on spatial patterns in natural systems may have important local effects ranging from predator/prey dynamics to species survival (Lasker 1975; Harris 1980). I did not address these higher trophic levels effects in this Chapter.

Changes in pattern in the SLS were measured by spectral analysis of the spatial distribution of biomass of each state variable. Spectral analysis calculates the change in variance with scale. The slope of this relationship, estimated by linear regression, provides a convenient summary of variation with scale in aquatic systems (Platt and Denman 1975; Weber et al. 1986). The estimation of the spectral slope is simple as long as the relationship remains linear. However, non-linearity or break-points in the spectral slope are often present requiring estimation of the slope for only a portion of the data. I calculated the spectral slope for data on the left side of the break (see Figure 4.6), the region dominated by effects at scales > 1 km.

Over the conditions tested within the SLS, there were highly significant effects of the three treatment variables (nutrient level, H , and model case) on the mean spectral slope of the model state variables (N , P , Z , D). When the model case treatments were broken out further there were additional significant effects due to the biology, diffusion, and turbulence sub-treatments (see Table 4). These results suggest that the model state variables were not only responding to differences in pattern caused by variations in H (which is expected) but also to differences caused by the physical and biological processing of the nutrient input signal.

Physical mixing processes played a dominant role in affecting the mean spectral slope of the model state variables. Mixing caused by the turbulence and diffusion processes operating in the model tended to become more important with time (Figure 4.7). Turbulence and diffusion both decrease variance but do so in different ways. Turbulence breaks up (i.e., mixes) any gradients and then diffusion acts to smooth out the remaining pattern (Okubo 1980; Abraham 1998). In my simulations, this may be the case, with turbulence rapidly breaking up the pattern and becoming less important with time, and diffusion smoothing out the patchiness created by the turbulence and hence becoming more important with time (i.e., increased surface area).

Other studies have found similar results indicating the importance of physical mixing effects on spatial patterns of scalar (e.g., temperature) and biomass distributions such as phytoplankton and zooplankton (Denman and Platt 1976; Gower et al. 1980). My results seem intuitive given the transient nature of biological interactions especially in a flow-through system like my simulation domain where nutrients are depleted over time. Even with the strong diffusion and turbulence signal, H was also found to significantly impact the spatial patterns of the state variables. H has its strongest impact early in the simulation and still accounted for approximately 50% of the explained variance in the model by the end of the simulation. This is probably attributed to the state variables responding to the pattern created by the input maps at the start of the simulation before turbulence, through the breakdown of concentration gradients and diffusion through the smoothing of concentration gradients, have a chance to alter patterns created by the nutrient input maps. Likewise, the brief importance of biological dynamics early in the

simulation is due to initial processing of the input signal, through high growth rates, before the mixing processes can dominate the pattern formation in the ecosystem.

The various levels of spatial heterogeneity (H) of nutrient input provides the baseline conditions (i.e., pattern and spectral slope) upon which the biological and physical processes can be compared. In the absence of any biological and physical processes, the spectral slopes of the input map are unchanged with time and range from a slope of -1.0 under low H to a slope of -3.0 under the high H (data not shown). Adding only diffusive mixing causes the slope to increase with time while adding turbulent mixing causes the spectral slope to decrease with time. These isolated effects are consistent with previous studies indicating that diffusive processes cause a “reddening” of the spectral signature while turbulent mixing causes a “whitening” of the spectral signature (Steele and Henderson 1981; Abraham 1998). These changes are caused by the injection of variability through turbulent mixing which in the absence of diffusive mixing will tend to accumulate variance causing the spectral slope to decrease. In contrast, diffusive mixing without turbulent mixing will tend to decrease variance through time through the dissipation of concentration gradients causing the spectral slope to increase. When both diffusion and turbulence are operating at the same time (i.e., the full physical model) the overall spectral slope tends to increase with time and may even reach an equilibrium condition at longer time-scales. My physical mixing signal is higher, (trending toward 4.2.2) than that predicted from the theoretical $-5/3$ Kolmogorov slope but similar in magnitude to spectral slopes calculated for passive tracers conducted in other studies (TIES data unpublished, Denman 1976).

4.4.4 Question 3: Can biological dynamics overcome the influence of physical mixing?

Pattern in aquatic systems is often controlled by physical mixing processes (Denman and Powell 1984), and much of the pattern variability in the SLS simulations was also driven by the same physical processes. As part of my simulation studies I was interested in identifying specific conditions where biological dynamics might be able to impact the spatial patterns seen in the model state variables. To investigate these interactions further I examined how the spectral slope might change under natural conditions as approximated under the full model case with all biological and physical processes operating. Figure 4.8 compares the spectral slope due physical mixing (diffusion and turbulence) with the corresponding case with biological processes included for each state variable. Deviation from the control line indicates zones where biological processes affect observed patterns.

Under low nutrient input levels the spectral slope of the model state variables almost always followed that of the physical mixing signal (i.e., control signal). The growth rates under these conditions are typically too low to support any type of biological induced pattern generation. Any growth that does occur quickly maps onto the signal created by the physical mixing processes. Under low nutrient input and H levels, the nutrient state variable did have a slight deviation from the control line for the second half of the simulation which may indicate the influence of regenerative processes from the other state variables as the system winds down. Under high H the zooplankton and detritus state variables responded slower to the control signal than did the phytoplankton and nutrient state variables. This is probably due to the time that it takes for the zooplankton to respond to the phytoplankton pattern which is in turn responding to the

nutrient input signal. The phytoplankton are able to respond more quickly to the control signal.

Biological alteration of the control signal was most prevalent under the high nutrient input treatments but this response was typically transient in nature. Usually by the end of the simulation the pattern of all the state variables matched that of the physical mixing signal. The biological signal was strongest under the higher nutrient input levels likely due to enhanced growth and predator-prey dynamics. Changes in the level of patchiness under high nutrient conditions had a dramatic impact on the types of changes seen in the model state variables. Under high H all of the state variables behaved in a similar manner while under low H each state variable had a different dynamic trajectory, an effect possibly due to smaller patch size of the low H maps. The large patches of the high H level maps may alter the dynamics which in previous studies have often been characterized as having their greatest influence at small scales (Lovejoy et al. 2001; Seuront et al. 2002). These results seem to indicate that biological dynamics will have their greatest impact on biomass distributions under conditions of high nutrients and a high degree of patchiness (i.e., low H level). My interpretation is that the biological interactions do not have the power to change broad scale patterns and instead impact fragmented patterns to a greater extent by enhancing the initial patterns present and adding additional biological induced variance. My results seem to suggest that the zooplankton distributions were less impacted by physical processes (see Figure 4.8c). Further analysis would be necessary to understand this observation within the context of these simulations.

4.4.5 Management Implications

Previous studies examining spatial patterns within the Chesapeake Bay indicate a system characterized by heterogeneous populations and nutrient inputs both in time and space. Understanding ecosystem response to nutrient variability is also important for the management of these systems, especially for management issues that are often dependent not only on mean levels of nutrient inputs but also their pattern in space and time (e.g., harmful algal blooms, hypoxia, and fisheries production).

My results indicate that heterogeneity in nutrient inputs is likely to have a relatively short-term effect. Any pulses in nutrients, as were approximated by the SLS simulations, were rapidly mixed and dispersed and subsequently drawn down by biological uptake and growth processes. Thus, any initial patterns rapidly disappear leaving little time for alterations in ecosystem processes. Therefore, knowledge of the mean level is of primary concern for predicting ecosystem response to changing nutrient inputs. This has implications for the management of estuarine ecosystems related to the timing of nutrient releases (e.g., sewage treatment plants) or the design of sampling programs (e.g., characterizing nutrient fields)

The SLS simulations also indicate that the pattern of nutrient inputs can impact the spatial distributions of organisms such as phytoplankton and zooplankton. Under high nutrient conditions changes in the level of patchiness impacted the spatial distributions of these key resources. The phytoplankton and zooplankton state variables were able to decouple their distributions from the underlying physical dynamics. While this may not impact ecosystem dynamics there could be smaller scale impacts that affect the likelihood of a phytoplankton bloom or the resource field available for fish predation.

Given the transient nature of these dynamics, efforts at predicting the spatial distributions of these resources should focus on characterizing the underlying physical field.

4.5 Conclusion

In this chapter I presented the application of a novel simulation framework (SLS) for addressing issues of pattern, scale, and physical-biological interactions within aquatic ecosystems. The platform utilizes a spatially-explicit nutrient-phytoplankton-zooplankton-detritus ecosystem model, realistic physical mixing techniques, and multifractal map generation techniques to investigate how patterns of nutrient input are affected by and interact with physical and biological processes during transit through an estuary. A factorial design was used to vary the concentration levels and heterogeneity of nutrients to simulate a spectrum of conditions ranging from oligotrophic to eutrophic and from fine-scale to broad-scale patterns of patchiness.

Results indicate that variations in the pattern of the nutrient input maps did not result in significant changes in mean levels of key ecosystem variables. The mean nutrient level is a better predictor of ecosystem dynamics than the level of variability in the nutrient input signal. The spatial patterns observed in the state variables, however, were strongly controlled by the nutrient input maps and physical mixing processes. The template pattern provided by the nutrient input maps is rapidly mixed through the physical mixing processes resulting in a general increase in the spectral slope with time. Biologically induced pattern formation was seen under some treatment combinations especially under high nutrient input conditions, but tended to be transient in nature, eventually merging with the physical mixing signal by the end of the simulation. When a

biological signal is present, the degree of patchiness affects the individual response of the various state variables causing deviations under low patchiness levels and coherence under high patchiness levels.

Identification of conditions under which spatial dynamics-interactions are important will aid the extrapolation of experimental results to the broader-scales of natural systems where spatial heterogeneity is a key component. The simulation framework used in this chapter allows the separation of effects due to physical and biological processes through the sequential elimination of processes thereby allowing one to gain a better understanding of the controlling processes on ecosystem productivity, biological dynamics, and spatial patterns in aquatic systems. This “spatial and temporal decomposition” provides a rigorous means of evaluating the range over which experimental results may be reliably extrapolated.

Appendix A: Model state variable and parameter descriptions

Listed below is a description of the state variables and parameters shown in Table 1.1 and a brief explanation of why a particular value was chosen for the simulations.

State Variables

N = The amount of dissolved inorganic nitrogen (gN) in the water column per cubic meter. The nitrogen components are for ammonia, nitrite and nitrate. Other sources of nitrogen are not modeled (e.g., urea). The initial value used for the simulations was based on equilibrium values.

P = The amount of phytoplankton biomass (gC) in the water column per cubic meter. The phytoplankton compartment consists mainly of a range of size classes from flagellates to diatoms aggregated together. The initial value used for the simulations was based on equilibrium values.

Z = The amount of zooplankton biomass (gC) in the water column per cubic meter. The zooplankton compartment consists mainly of micro and macro zooplankton species. The initial value used for the simulations was based on equilibrium values.

D = The amount of particulate material (gC) in the water column per cubic meter. This compartment is meant to approximate a microbial loop and to consist of both living and

non-living material (i.e., dead phytoplankton and zooplankton, bacteria, protozoan). The initial value used for the simulations was based on equilibrium values.

Parameters

N_o = The concentration of dissolved inorganic nitrogen below the mixed layer. The 0.30 gN m⁻³ value used is within the range found for sub-thermocline nitrogen values in the mainstem of the Chesapeake Bay during the summer months (Kemp et al. 1990). This pool supplies all the nitrogen inputs to the model when there are no external nutrient inputs and is assumed to be unaffected by the other processes occurring in the model.

N_d = The diffusion rate across the thermocline. This parameter is highly variable and can be parameterized with a wide range of values depending on the stability of the thermocline or the amount of turbulence in the mixed layer. Edwards and Brindley (1999), based on a mixed layer depth of 12.5 m found a range of values (0.0008-0.13 day⁻¹) from the literature. I chose a value of 0.02 day⁻¹ which can be converted into a vertical eddy diffusivity with the following equation: $Diffusivity = percent\ flux * ((\delta X)^2 / (\delta T))$. Using a δX of 5 m and a δT of 86,400 sec, you obtain a vertical diffusivity of 5.8x10⁻⁶ cm² sec⁻¹ which is within the range found for stratified interior conditions (Sanford 1997) and similar to other values used in models parameterized to the Chesapeake Bay (Johnson et al. 1993). The value was set at the low range to simulate the stratified conditions in the summer months and using this value and the dissolved

nitrogen concentration below the mixed layer results in a maximum nitrogen input to the mixed layer of $0.006 \text{ gN day}^{-1}$ assuming no nitrogen is present in the mixed layer.

R_f = The respiratory losses for fish is the fraction of the zooplankton biomass that is removed by fish predation and then recycled directly back to the nutrient compartment. The parameter represents excretion components lost from the fish during movement and feeding within the model domain and was arbitrarily set at 0.10 day^{-1} .

P_{max} = The maximum phytoplankton growth rate under non-limiting conditions and optimal temperature. I chose a growth rate of 2.8 day^{-1} similar to a value used by Fasham et al. (1990) and within the range ($0.14\text{-}8.11 \text{ day}^{-1}$) reported for natural populations in marine and coastal areas (Parsons et al. 1984).

R_p = The respiratory losses for phytoplankton is the proportion of biomass loss each day due to cell maintenance and respiration. A wide range of values is found in the literature so I used the common value of 0.05 day^{-1} found in many models of this type.

M_p = The mortality losses for phytoplankton is the proportion of biomass lost each day due to natural mortality such as cell senescence. Many values can be used for this parameter depending on if other mortality factors are included (e.g., predation, starvation, etc). Since I have separated zooplankton predation from this term, a low natural mortality rate of 0.05 day^{-1} seemed appropriate and consistent with values used in other models.

S_p = The sinking losses for phytoplankton is the proportion of biomass lost each day due to the loss of phytoplankton from the water column from sinking and mixing below the thermocline. I used a value of 0.05 day^{-1} which, when combined with a mixed layer depth of 5 m, gives a sinking rate of 0.25 m day^{-1} . The value used is within the range found for natural mixed assemblage communities (Bienfang 1981).

I_n = The light half-saturation constant is the intensity of light that is one-half the amount needed to give maximum productivity. I chose a value of $10 \text{ E m}^{-2} \text{ d}^{-1}$ to correspond roughly to half of the surface light intensity so that in the absence of self-shading and light attenuation effects, the phytoplankton could grow near their light saturated maximal values. Kirk (1994) reports values for this parameter for marine microalgae in the range of $(4.32-70.7 \text{ E m}^{-2} \text{ d}^{-1})$.

I_o = The surface light intensity after correcting for reflectance due to the water surface. The value of $26 \text{ E m}^{-2} \text{ d}^{-1}$ reflects an actual surface intensity of $52 \text{ E m}^{-2} \text{ d}^{-1}$ which is within the range of values reported for the Chesapeake Bay (Harding et al. 1986).

K_c = The attenuation of light in the water column due to phytoplankton biomass. The value $0.4 \text{ m}^2 \text{ g}^{-1}$ was taken from Edwards and Brindley (1996) and falls within the range found in the literature $(0.3-1.2 \text{ m}^2 \text{ g}^{-1})$.

K_w = The attenuation of light in the water column due to non-living particulate and dissolved material. A value of 0.2 m^{-1} was used so that with equilibrium phytoplankton concentrations (0.38 gC m^{-3}) the total light attenuation (K_d) coefficient would be 0.352 m^{-1} which falls within the range ($0.1\text{-}3.0 \text{ m}^{-1}$) typical for coastal waters (Lorenzen 1972) and found in the Chesapeake Bay (Harding et al. 1986).

K_z = The depth of the mixed layer in the Chesapeake Bay over the summer stratification period. I used a value of 5 m and assumed that this is a typical mixed layer depth in the mesohaline regions of the Chesapeake Bay based on a range of values reported from cruises in Harding et al. (1985).

K_n = The nutrient half-saturation constant is the concentration of nitrogen that will give half of the maximum nutrient uptake rate. A value of 0.02 gN m^{-3} was used which falls within the range of values reported for large and small phytoplankton (Eppley et al. 1969; Goldman and Glibert 1983).

Z_{max} = The maximum zooplankton ingestion rate of phytoplankton biomass under optimal temperature and prey densities. Edwards and Brindley (1996) report a range for this value of ($0.6\text{-}1.4 \text{ day}^{-1}$) from other marine biogeochemical models. I chose a mean value of 1.0 day^{-1} which is also the value used by several other models (Fasham et al. 1990; Doney et al. 1996; Denman and Pena 1999)

A_z = The assimilation efficiency reflects the proportion of consumed phytoplankton biomass that is converted into zooplankton biomass. Typical values for this parameter range from (0.7-0.9 day⁻¹). Several models (Dadou et al. 1996; Denman and Pena 1999; Druon and Fevre 1999) use a value of 0.7 day⁻¹ which is the value I decided to use for my simulations.

R_z = The respiratory losses for zooplankton is the fraction of zooplankton biomass that is lost due to excretion and respiration. The value used in the simulations (0.25 day⁻¹) is within the wide range of values (0.05-0.80 day⁻¹) found in the literature (Fasham et al. 1990; Edwards and Brindley 1996; Druon and Fevre 1999).

M_z = The mortality losses for zooplankton reflect the amount of biomass lost due to natural and density dependent mortality factors (e.g., starvation, cannibalism). This parameter is a quadratic function of biomass and was given a value of 1.0 day⁻¹ (gC m⁻³)⁻¹ after Edwards and Brindley (1996) who also found a range of (0.25-2.0) for this parameter in the literature. Using this parameter value translates into a loss rate equal to the zooplankton biomass value, which under equilibrium zooplankton densities, becomes a loss rate of 0.056 day⁻¹. The implications of using this particular parameter value and formulation is that the zooplankton biomass can never exceed 1.0 gC m⁻³ assuming there are no other loss terms for the zooplankton.

k = The half-saturation constant for zooplankton grazing represents the concentration of phytoplankton or detritus that will result in a grazing rate that is one-half the maximal

value. A value of 0.10 gC m^{-3} was used based on similar values by Evans and Parslow (1985) and Armstrong (1994). The parameter is the same for both phytoplankton and detritus.

P_{pref} = The preference for phytoplankton over detritus by zooplankton. A value of 0.5 will give equal preference. In my simulations I used a value of 0.7 to give phytoplankton a higher preference over detrital material similar to Loukos et al. (1997) and Fasham (1995).

R_d = The remineralization rate of detrital material into dissolved inorganic nitrogen due to bacteria decompositional processes. A rate of 0.05 or 0.1 day^{-1} has typically been used in other NPZD models (Fasham 1995; Doney et al. 1996; Edwards 2001) but rates as high as 0.2 and 0.3 day^{-1} are also used (Dadou et al. 1996; Kemp et al. 2001). I chose a value of 0.2 day^{-1} for my simulations to reflect the higher decomposition rates occurring during the summer months in the Chesapeake Bay.

S_d = The sinking loss rate for detritus is the proportion of biomass lost each day due to the loss of detritus from the water column from sinking and mixing below the thermocline. I used a value of 0.05 day^{-1} to correspond to the value used for the phytoplankton sinking rate. When combined with a mixed layer depth of 5 m this value gives a sinking rate of 0.25 m day^{-1} . A higher sinking rate, as is often used in other models, is probably warranted especially considering that the detrital material also

consists of fecal pellets and dead zooplankton (Fasham et al. 1990; Druon and Fevre 1999; Edwards 2001; Fennel et al. 2001)

F_d = The unassimilated losses from fish is the fraction of the zooplankton biomass that is removed by fish predation and then is recycled back to the detrital compartment. The parameter represents mostly unassimilated zooplankton biomass and some fecal pellet production that occurs during fish movement and feeding within the model domain. The parameter was arbitrarily set at 0.40 day^{-1} . An alternative pathway for this variable would be to have this material exported from the model as if often done with other studies (Fasham et al. 1990; Edwards 2001) due to rapid sinking associated with fish fecal material providing little time for remineralization.

External Drivers (Sensitivity Experiments)

η = The nutrient input rate is the amount of nutrient input to the model from outside sources. For the sensitivity experiments, a range of values ($0.0\text{-}0.64 \text{ gN m}^{-3} \text{ day}^{-1}$) were utilized to force the model to go from an oligotrophic to a eutrophic state as described in section (2.2.5).

μ = The water exchange rate represents the amount of material that is lost from the model domain due to dilution effects with the surrounding water. The loss affects all the state variables equally. For the sensitivity treatments, I used a range of values ($0.0\text{-}0.20$

day⁻¹) to simulate changes in the size or amount of flushing occurring in the model as described in section (2.2.5).

v = The fish predation rate for zooplankton is the proportion of zooplankton biomass that is removed by fish predation. This parameter was used as a treatment variable in my sensitivity experiments and was varied from (0.0-0.80 day⁻¹) to simulate a gradient in the degree of top-down control as described in section (2.2.5).

Appendix B: Sensitivity Analysis

There are numerous techniques available to conduct a sensitivity analysis on model simulation results (e.g., Brylinsky 1972; Rose and Swartzman 1981; Dale et al. 1988; Hakanson 2000; for recent reviews see Hamby 1995; Homma and Saltelli 1996). Most of the methods can be categorized under two headings, relative and global. A relative sensitivity analysis is the technique most often seen in the literature and is sometimes called individual parameter perturbation experiments. With this technique parameters are systematically increased or decreased over a predetermined range (e.g., +/- 10%), with the subsequent change in model output recorded. Global sensitivity analysis is similar to a relative sensitivity analysis except that the parameters are varied over the whole range found in the literature for a particular parameter. While useful for a number of applications, these methods of sensitivity analysis can lead to an incomplete picture of the relationship between sensitive parameters. Since only one parameter at a time is varied there is no way to understand the interactions between model parameters and previous studies have also demonstrated that when there are large errors in all parameters, nonlinear and higher order effects dominate the outcome and seriously limit the applicability of results (Gardner et al. 1980a; Gardner et al. 1980b; Gardner et al. 1981). Sensitivity analysis is most powerful when the parameter errors are small and each parameter contributes independently to prediction error.

An alternative to the above types of sensitivity analysis is a method developed by Gardner and Trabalka (1985). With this method, Latin hypercube sampling is used to simultaneously vary all parameters by +/- 1% of their default value giving an unbiased

indication of a model's sensitivity to a minimal change in parameter value while also taking into account the interactive effects of the other parameters. The benefits of this method are that it greatly reduces the number of simulations needed, allows for extensive replication through Monte Carlo analysis, and provides an unbiased estimation of parameter importance with only a minimal change in parameter value. The technique has been used to address a wide range of issues which include: plankton productivity (Bartell et al. 1988a), toxicological effects (O'Neill et al. 1983), top-down and bottom-up controls on productivity (Bartell et al. 1988b) and forest development (Dale et al. 1988). The method is even more powerful when combined with other complementary analysis techniques like uncertainty and error analysis (Gardner et al. 1980a; Gardner et al. 1980b; Gardner et al. 1981; Gardner 1984; Bartell et al. 1986; Gardner et al. 1990; Rose et al. 1991).

Each sensitivity analysis conducted in chapter 2 followed the steps outlined in Figure. B.1. First, an input file is created which contains all of the baseline values for each of the parameters to be varied in the sensitivity analysis. This file also specifies the range that the parameters will be varied (0-100 percent), the type of sampling distribution from which to take values (normal, log, uniform, exponential) and the number of replications (should be at least 100). Second, the file from step one is input into a program (i.e., PRISM) that uses a Latin hypercube sampling algorithm to generate random parameter sets based on the input file specifications (Gardner et al. 1983). For chapter 2, the specifications were to vary each of the parameters by 1% over a normal distribution and to create 100 independent parameter sets. Third, the generated output file from the PRISM program, containing the 100 distributions for each of the

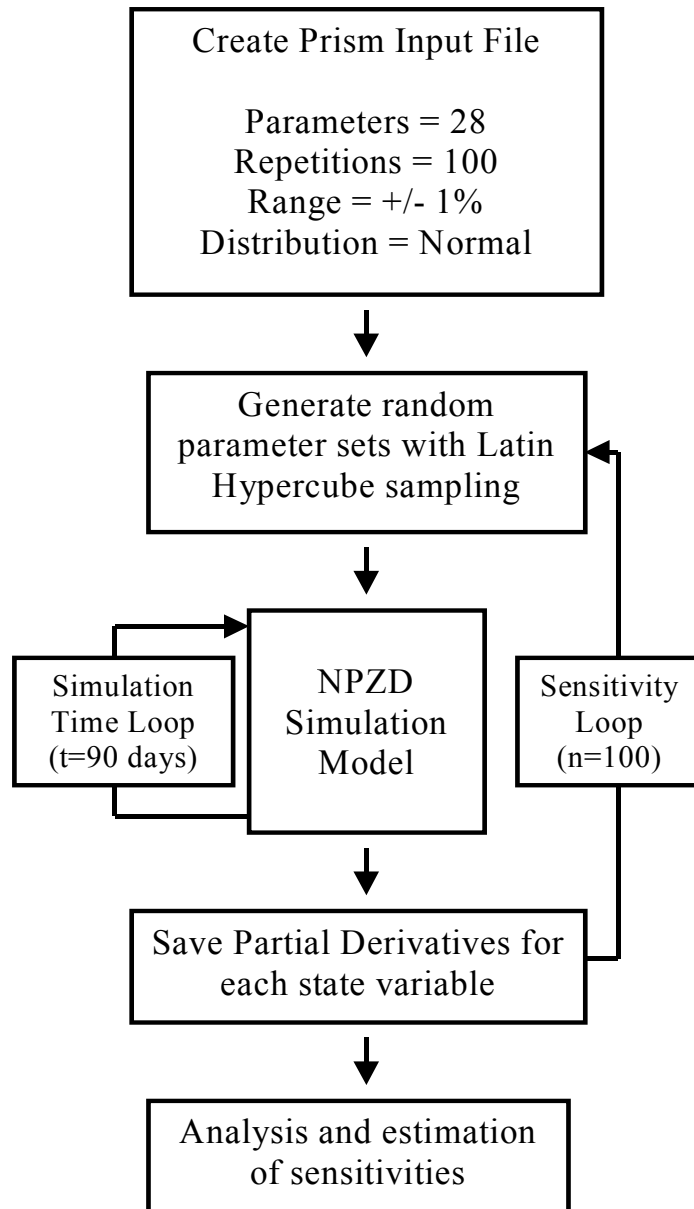


Figure B.1. Schematic showing the steps involved in conducting a sensitivity analysis for each of the treatment combinations.

parameters, is sequentially read in by the simulation model. Fourth, at the end of each simulation run the partial derivatives for the four state variables and the input parameter values for that particular simulation are saved to a separate file. Fifth, this output file is then read in by another program which calculates the actual sensitivities for each parameter, along with, other statistics such as mean, max, min, coefficient of variation, and parameter correlations.

Appendix C: Chi-Square Analysis

The degree of interaction observed between the treatment factors for each factorial experiment ($\eta \times \nu$ and $\eta \times \mu$) was determined through frequency and Chi-Square analysis. I wanted to determine if the sensitivity dynamics (i.e., if a parameter was sensitive or non-sensitive) seen when only one of the treatments is varied, while the other is held at the baseline value, would be predictive of the dynamics over the range of non-baseline treatment combinations. My null hypothesis was that the sensitivity dynamics seen along the edges of the factorial experiment (i.e., control cases, direct effects) are sufficient to predict the sensitivity dynamics over the rest of the factorial treatment combinations. Deviation from this prediction shows sensitivity dynamics cannot be predicted from the direct effects alone and that an interaction exists between the two treatment axes within the factorial experiment. The degree of interaction was characterized by the frequency that the null hypothesis was proven to be correct which was then converted to a probability with the following calculation:

$$Probability = (observed / expected) * 100 \quad (C.1)$$

where *expected* is the total number of treatment combinations (i.e., cases) exclusive of the control cases and *observed* is the number of cases where the results in the control cases matched the results seen in the non-control cases.

The above calculation is repeated for each of the individual treatment combinations along a particular treatment axis. Two examples are illustrated below

where the values shown in the table are the sensitivity values for each treatment combination. The control values are highlighted in dark gray bold whereas light gray is used to highlight treatment combinations where the sensitivity dynamics were successfully predicted by the corresponding non-control case. The number of times that a control case successfully predicts the non-control cases is summed over all the control cases to calculate the overall probability for that treatment series. Higher probably values correspond to higher predictive capability and a lower degree of interaction between the main and indirect effects. The probability values are shown in Table 2.5 for each parameter and each treatment (η , ν , μ). Since the nutrient treatment series, η , appears in both factorial experiments, there are two probability values, the first percentage is for the ($\eta \times \mu$) series and the second value is for the ($\eta \times \nu$) series.

Example 1: Factorial experiment ($\eta \times \nu$), parameter (K_z), phytoplankton state variable (P)

Nutrient Input (η)	Fish Predation (ν)			
	0	20	40	80
0	1.0	0.3	0.9	1.0
0.04	1.7	0.9	0.6	1.5
0.08	1.9	1.3	0.9	1.4
0.16	37.0	36.4	35.4	37.6
0.32	37.0	36.4	35.4	37.6
0.64	37.0	36.4	35.4	37.6

Fish Predation Treatment (ν) Expected = 18
 Observed = 18
 Probability = 100%

1st series (times correct) = 3
 2nd series (times correct) = 3
 3rd series (times correct) = 3
 4th series (times correct) = 3
 5th series (times correct) = 3
 6th series (times correct) = 3
 Total = 18 out of 18

Nutrient Input (η)	Fish Predation (ν)			
	0	20	40	80
0	1.0	0.3	0.9	1.0
0.04	1.7	0.9	0.6	1.5
0.08	1.9	1.3	0.9	1.4
0.16	37.0	36.4	35.4	37.6
0.32	37.0	36.4	35.4	37.6
0.64	37.0	36.4	35.4	37.6

Nutrient Input Treatment (η) Expected = 20
 Observed = 8
 Probability = 40%

1st series (times correct) = 2
 2nd series (times correct) = 2
 3rd series (times correct) = 2
 4th series (times correct) = 2
 Total = 8 out of 20

Example 2: Factorial experiment ($\eta \times \mu$), parameter (Z_{max}), Nutrient state variable (N)

Nutrient Input (η)	Water Exchange (μ)					
	0	1.25	2.5	5	10	20
0	37.6	34.7	32.1	24.1	0.0	0.0
0.04	32.8	35.3	36.6	37.5	35.8	0.5
0.08	20.1	23.5	26.0	28.8	30.0	9.3
0.16	3.5	2.9	2.6	11.4	13.9	10.4
0.32	4.7	3.4	2.6	1.9	1.4	0.9
0.64	4.9	3.2	1.9	1.0	0.6	0.3

Water Exchange Treatment (μ) Expected = 30
 Observed = 23
 Probability = 77%

1st series (times correct) = 3
 2nd series (times correct) = 4
 3rd series (times correct) = 4
 4th series (times correct) = 2
 5th series (times correct) = 5
 6th series (times correct) = 5
 Total = 23 out of 30

Nutrient Input (η)	Water Exchange (μ)					
	0	1.25	2.5	5	10	20
0	37.6	34.7	32.1	24.1	0.0	0.0
0.04	32.8	35.3	36.6	37.5	35.8	0.5
0.08	20.1	23.5	26.0	28.8	30.0	9.3
0.16	3.5	2.9	2.6	11.4	13.9	10.4
0.32	4.7	3.4	2.6	1.9	1.4	0.9
0.64	4.9	3.2	1.9	1.0	0.6	0.3

Nutrient Input Treatment (η) Expected = 30
 Observed = 15
 Probability = 50%

1st series (times correct) = 2
 2nd series (times correct) = 2
 3rd series (times correct) = 2
 4th series (times correct) = 3
 5th series (times correct) = 2
 6th series (times correct) = 4
 Total = 15 out of 30

Probability values (i.e., frequency values) were then tested for significance at the ($p=0.01$) level by comparing *observed* versus *expected* results based on a one-tailed chi-squared distribution with one degree of freedom. The chi-squared value for each treatment series was calculated with the following equation:

$$\chi^2 = (\textit{observed} - \textit{expected})^2 / \textit{expected} \quad (\text{C.2})$$

where *observed* and *expected* are defined as in C.1 with χ^2 based on a (2x2) contingency table consisting of (*observed/expected*) and (*sensitive/non-sensitive*) for each axis of the table. Chi-square values exceeding the value for a one-tailed chi-squared distribution ($\chi^2 = 6.63$) with one degree of freedom (number rows-1 * number columns-1) are indicated in bold in Table 2.5 (actual χ^2 values are not shown) for each of the sensitive parameters from both factorial experiments.

For the examples shown above the Chi-squared values would be 0.00 (ν) and 7.20 (η) for example 1 and 1.63 (μ) and 7.50 (η) for example 2. The nutrient input rate (η) for each example does exhibit an interaction effect, indicating that changes in this variable alter the sensitivity dynamics over those seen in the baseline nutrient input case. The fish predation rate (ν) and water exchange rate (μ) on the other hand, did not have an interaction effect, which should allow the baseline case sensitivities to predict the sensitivity dynamics in the non-baseline cases with some degree of accuracy.

Appendix D: Numerical Diffusion

Numerical diffusion refers to the error introduced into grid-based (i.e., Eulerian) simulation frameworks when concentration gradients are averaged between grid cells. Numerical diffusion occurs because it is not possible to represent a particle or calculate a concentration between grid points. The assumption which has to be implemented for these systems is that anything not fluxed 100% to the next grid cell is assumed to be uniformly mixed between the grid points. What this means in practice is that anything fluxed into an adjacent cell is automatically transported to the center of that cell resulting in an accelerated spreading rate of a particle or a concentration gradient (i.e., numerical diffusion).

For example, if there is a series of five grid cells which are 100 m in diameter (Figure D.1). The left-most cell is filled with a substance while the rest of the cells are empty. There is an external flux (e.g., such as from an external turbulence field) applied to the system which will move everything in each cell to the right at a rate of 20 m per second. Under these conditions, to completely empty the cell of material should take five seconds. If the time-step of the simulation is fixed at one second intervals then the equivalent flux out of the cell would be 20 percent of the material at each time-step.

Because the grid cell distance is 100 m, at each time-step, the 20 percent of the material which is fluxed into the cell is assumed to be uniformly mixed within that cell rather than at the leftmost edge. At the next time-step, the material which is now in the second cell is fluxed to the third cell when there should be no material in that cell until at least the sixth time-step since it would take at least five time-steps to travel the distance

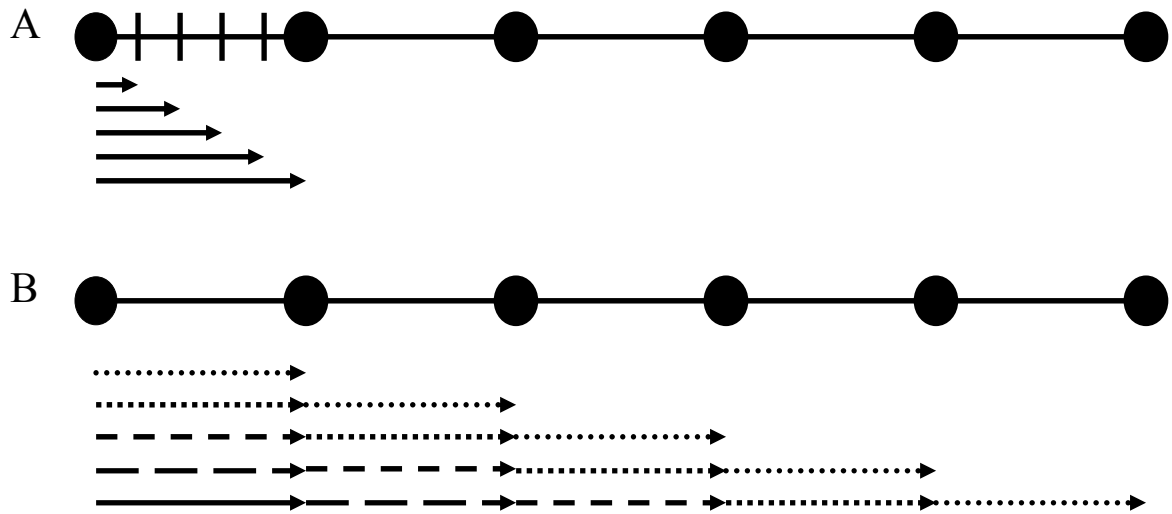


Figure D.1. Diagram illustrating the effects of numerical diffusion within grid-based simulation frameworks. Panel A shows the idealized transport of a substance subjected to a 20 percent flux rate each time-step without numerical diffusion errors. Each arrow represents the next time-step. Panel B shows how the transport would be changed due to numerical diffusion error. See text for additional details.

of a grid cell. With each additional time-step the material gets transported farther and farther.

After five time-steps, instead of having 100 percent of the material in the second cell there is material in all five cells. While the error from the numerical diffusion decreases exponentially away from the original cell (i.e., 20% of 20% of 20% and so on) this artificial spreading can be a problem in cases where accurate concentration gradients or the precise location of objects are required.

Numerical diffusion errors can be minimized through various means. One method is to increase the time-step of the simulation so more material is fluxed for a given time-step. The grid size can also be reduced so there is less distance to travel between grid cells. For some applications, the opposite approach can be taken where the time-step is decreased or the grid size increased to minimize the amount of material which is fluxed. Other methods use complicated numerical and grid-solutions (e.g., James 1996; Vested et al. 1996; James 2002) and while all of these methods can help to alleviate some of the error introduced through numerical diffusion there unfortunately is no way to eliminate numerical diffusion errors within an Eulerian simulation framework.

Appendix E: Turbulence Intensity Calculation

The turbulence intensity is an important measure of how rapidly a turbulent field will stir (i.e., mix) a substance contained within that field. Before I could apply the turbulence generation routine it was necessary to calculate the intrinsic turbulence intensity of the field of eddies within the simulation framework and also to determine how changes in the eddy distribution would influence the value obtained. Once the intrinsic turbulence intensity was determined I could then adjust the time-step or the frequency in which I call the turbulence generation routine to approximate a range of turbulence intensities which might be of interest to real-world applications.

For the turbulence intensity calculation I used a map size of 1024x1024 cells, a grain size of 100m and a time-step of 10min. I ran the turbulence simulation routine until the turbulent field was fully developed (approximately 512 calls to the turbulence generation routine) and then seeded the map with 10,000 evenly spaced particles within the simulation domain. After one time-step (i.e., one call of the turbulence generation routine) the displacement of each particle was recorded. The displacement of each particle was then converted into a RMS (i.e., root mean squared error) value for the whole map at that time-point based on the following calculations:

$$U_{(RMS)} = \text{sqrt} ((\text{Sum}(D)^2) / N) / \text{Delta}T \quad (\text{Eq. E1})$$

where

N = the number of particles seeded on the map

D = distance traveled for each particle = $\sqrt{(x_2-x_1)^2 + (y_2-y_1)^2}$

X_1 and Y_1 are the initial x and y coordinates of each particle

X_2 and Y_2 are the final x and y coordinates of each particle

ΔT = time-step of the model

The RMS value was then calculated for 100 subsequent time-steps (spaced five time-steps apart, resetting the particles after each time-step) to determine the variability in the RMS value as the various eddies complete and start their rotational cycles. An average RMS value and other descriptive statistics (i.e., min, max, standard error and coefficient of variation) was then calculated based on these 100 time-points. This whole experiment was repeated 10 times using a different random eddy distribution with a final grand mean calculated based on all 1,000 RMS values.

As shown in Table 3.2, the overall RMS value was found to be 1.34 based on the 10 eddy distributions tested which means for every solution-step (each time the turbulence generation routine is called) the average distance traveled by a particle within the simulation domain is 1.34 pixels. The mean varied from 1.29 to 1.41 between the different eddy distributions. The variability is caused by the eddies cycling on and off over the course of a simulation and can be seen in the coefficient of variation (CV) within each map replication, which ranged from 7.62 to 14.09 percent. For each map snapshot, the number of pixels traveled by an individual particle could be as high as four or five pixels or as low as zero pixels but, as the grand mean value indicates, is typically around one or two pixels.

For my planned simulations, I wished to obtain a turbulence intensity of 10 cm/s. Each cell in the domain was set at a size of 100 m which meant that for each solution-step a particle, on average, would move 134 m distance. This translates into an intrinsic turbulence intensity of 22.3 cm per solution-step. To achieve the desired turbulence intensity of 10cm/s this meant that each solution-step (i.e., time-step) would have to be 1,340 sec or 22.4 min. The frequency at which the turbulence generation routine was called, which in this case is once per time-step, could also have been changed if for some reason the time-step or cell size were constrained to a set value.

Appendix F: Spectral Analysis

Spectral analysis is a linear statistical technique in which the variance around the mean of a data series is broken up into contributing frequencies (Platt and Denman 1975). Spectral diagrams are an important and extensively-used method because they enable partitioning of the total variance in a data series among contributions having different characteristic length scales (Weber et al. 1986). In my experiments I used spectral analysis to identify the spatial patterns of key variables within the simulation domain. The method is appealing because it allowed partitioning of the data into component wavenumbers to expose the underlying pattern, to show how the variance of the dataset changed with scale, and to reveal persistent patterns in the data.

The method works by dividing the data series into a range of sines and cosines of decreasing wavelengths where the amplitude gives the intensity of variability at each scale. The spectrum is created by graphing the log of the spectral density (i.e., the variance of the amplitude squared) versus the log of the wavenumber. From the velocity spectrum a spectral slope can be calculated (i.e., Beta). Spectra are often described in terms of color, analogous to wavelength characteristics of visible light, such that “white” spectra (flat slope) indicate that variance is constant across length scales, “red” spectra (negative slope) indicate the relative importance of large length scales, and “blue” spectra (positive slope) indicate dominance of processes acting over small length scales (Ripa et al. 1999). There is a large literature illustrating how changes in spectral pattern can be related to changes in physical and biological processes. For a good review of the application of spectral analysis for addressing terrestrial and aquatic questions in ecology

please see Platt and Denman (1975). A recent review by Franks (2005) highlights some of the pitfalls with using spectral analysis for the analysis of biological data, especially for understanding plankton patchiness over the inertial subrange of three-dimensional isotropic turbulence.

For the analysis in chapters 3 and 4, I utilized the Proc Spectral routine contained within the SAS statistical software package. Before analysis I detrended each transect and conducted a spectral analysis on an ensemble sample which allowed combination of transects obtained from the simulation domain into one composite spectral slope. Creating a composite spectral slope is a very powerful means of bringing out persistent patterns in the data and allows the creation of variance and confidence estimates around the spectral slope data.

Appendix G: Computer Code

The information in this appendix contains the computer code used to create the SLS simulation framework. The computer language used for the platform is FORTRAN 95 and after compilation with an appropriate compiler, will run on a Windows or DOS based operating system. All of the necessary input files, modules, subroutines, functions, and main program necessary to create and executable file is included. Please note however, that the code was developed over many years and contains variables or sections which may no longer be in use or may have parameter names which differ from those listed in the dissertation. I have attempted to remove as much of this code as possible without removing key parts of the current operating sections. Parameter values listed are for illustrative purposes only and do not correspond to any particular experiment conducted as part of this dissertation. No guarantees are implied to the accuracy of experiments conducted with the computer code provided. Unfortunately, it was not possible to provide a flowchart or a user manual for this version of the SLS framework as it is still undergoing active development. This code should only be considered a beta version or “proof of concept”.

Modules

The variables contained in the 3 modules to follow are globally declared and used in multiple subroutines. This provides consistency among subroutines and allows a variable to be declared only once.

```
MODULE global_variables
```

```
    IMPLICIT NONE
```

```
    INTEGER :: boundary_input,visual_switch
    REAL, ALLOCATABLE :: Nutinput(:, :)
    REAL :: Nut_input_amount
    INTEGER :: AdvectionFreq
    REAL :: Init(4)
    INTEGER :: NutInputFreq
    INTEGER :: map_location
    INTEGER :: map_input
    INTEGER :: output_dimension, circle_dimension
    INTEGER :: Nut_chapter_switch,stat_switch
    INTEGER :: map_save

    Real :: time, t, interval,tt
```

```

INTEGER :: counter
INTEGER :: tmax,numrow,numcol,zzzz !direction of advection
INTEGER :: flag,iseed_input
REAL :: DT

REAL :: CN, Ppref, Io, M !depth (now spatial)
REAL :: Dk, Kn !No(now spatial)
REAL :: Pmax, Sp, Rp, Mp, Kw, Kc, Ik
REAL :: Zmax, Az, Rz, Mz, k3 , Pz
REAL :: Rd, Sd
REAL :: Nutrient, Light, Kd, Drate, Prate, Dgraz, Pgraz
REAL :: Input, Regeneration, Remineralization
REAL :: Uptake, PhytoSink, PhytoResp, PhytoMort, PhytoPred
REAL :: ZooResp, ZooMort, ZooPred
REAL :: NaturalMort, ZooFecal, Remineralize, DetritalSink, DetritalPred
REAL :: Nnew,Dnew
REAL, ALLOCATABLE :: N(:,,:), P(:,,:), Z(:,,:), D(:,,:)
REAL, Allocatable :: Pnew(:,,:),Znew(:,,:),
BiomassOut(:,,:),PredOut(:,,:),PhytoGrowth(:,,:),ZooGrowth(:,,:)
REAL :: Fz, Ff, BiomassOutNew, PredOutNew, FishResp, FishFecal

REAL :: No,Kz
!REAL, ALLOCATABLE :: No(:,,:), depth(:,,:)
REAL :: diffcoeff(4)
REAL :: advcoeff(4)
REAL :: turcoeff(4)
REAL :: SumN, SumP,SumZ, SumD
REAL :: Ntotal,Ptotal,Ztotal,Dtotal
REAL, ALLOCATABLE :: Nmax(:), Pmax(:), Zmax(:), Dmax(:)
REAL, ALLOCATABLE :: Nmin(:), Pmin(:), Zmin(:), Dmin(:)
REAL, ALLOCATABLE :: Nave(:), Pave(:), Zave(:), Dave(:)
REAL, ALLOCATABLE :: Nvar(:), Pvar(:), Zvar(:) ,Dvar(:)
REAL, ALLOCATABLE :: Nsd(:), Psd(:), Zsd(:) ,Dsd(:)
REAL, ALLOCATABLE :: Ncv(:), Pcv(:), Zcv(:) ,Dcv(:)
INTEGER :: Mapsize
REAL, ALLOCATABLE :: Nmap(:,,:), Pmap(:,,:), Zmap(:,,:), Dmap(:,,:)
REAL, Allocatable :: Narraymin(:),Narraymax(:),Parraymin(:),Parraymax(:)
REAL, Allocatable :: Zarraymin(:),Zarraymax(:),Darraymin(:),Darraymax(:)
REAL :: Nmapmax, Nmapmin, Pmapmax, Pmapmin, Zmapmax, Zmapmin, Dmapmax,
Dmapmin

INTEGER :: maxlevel_input, fractalinit,stateinit
Real :: hclump_input,meanzoo_input,xxsd_input
INTEGER :: scaleviz,nutpipe,nutconst,nutslug
REAL :: nutpipevalue,nutconstamount,advections witch
INTEGER :: visualization

```

```

INTEGER :: xlocation, ylocation, nut_spot
REAL :: coorangle, direction,meanturn, newangle,cv_input,threshold
INTEGER :: ylocationcoor,pred_spot,return_freq,fishend, transectlocation
REAL :: return_freq2
REAL :: nut_amount,nut_move,nut_angle
REAL :: pred_amount,pred_move,pred_angle
INTEGER :: nut_radius,pred_radius, nut_frequency, pred_frequency,disturbance
REAL :: cos_map_max, depth_max, pool_max,nutconst_max
INTEGER :: depth_input, pool_input,nutconst_input
INTEGER :: DT_trigger, file_save_freq,
        stat_counter,simlength,nut_spike_freq,first_nut_spike
REAL :: simtime
INTEGER :: TurbulentFreq,max_radius,min_radius
INTEGER :: FileSaveFreq, StartNutSpike, EndNutSpike, NutSpikeFreq, StartFishPred,
        EndFishPred,&FishPredFreq, StartNutMove, EndNutMove, NutMoveFreq,
        StartFishMove, EndFishMove, FishMoveFreq
REAL :: xlocation_fish, ylocation_fish, x_location_fish, y_location_fish
REAL :: x_new_fish, y_new_fish
REAL :: angle_fish, distance_fish
INTEGER :: counter_fish

REAL :: xlocation_nut, ylocation_nut, x_location_nut, y_location_nut
REAL :: x_new_nut, y_new_nut
REAL :: angle_nut, distance_nut
INTEGER :: counter_nut
Integer :: nutradius
Real :: NutSlugAmount
Integer :: Fishradius
REAL :: FishExt,Fishfood
INTEGER :: nut_input,counter2, stat_counter2
REAL :: nut_max
INTEGER :: MzSwitch
REAL :: linmort,quadmort
INTEGER :: nrep, nval,nval2
INTEGER :: StartSen, EndSen, SenFreq
INTEGER :: senflag,senswitch
INTEGER, ALLOCATABLE :: xcenter(:),ycenter(:),rotation(:),&
        circlesize(:),starttime(:),freqswitch(:)
INTEGER, ALLOCATABLE ::
        cascade_switch(:),cascade_trigger(:),radius_stop_location(:)
INTEGER :: totalcir
INTEGER :: diff_switch,adv_switch,file_switch,min_radius2,x1_input,x2_input,turran
INTEGER :: y1_input,y2_input,line_init,tur_switch,repetition,radius_flag,tur_trigger
INTEGER :: turmove_flag,bio_switch
REAL :: ratelimit

```



```
END MODULE
```

```
MODULE map_variables  
  IMPLICIT NONE
```

```
  INTEGER :: maptype_input, wrap_input
```

```
END MODULE
```

```
MODULE sensitivity  
  implicit none
```

```
  REAL :: parm(10000)  !keeps an array of the values  
  INTEGER :: np
```

```
END MODULE
```

Input Files

The SLS platform uses two input files to read in commonly used model parameters and constants. These two files should be read in as external text files. By using input files these model parameters and constants can be changed without the need to recompile the code of the SLS platform.

! This file contains all of the constants used for the NPZD model (Biology Subroutine)
! This file must be called inputpar.dat

0.1761 ! CN (converts gC to gN) (unitless)
0.020 ! Kn (half-saturation constant for nitrogen uptake) (gN/m³)
2.8 ! Pmax (maximum phytoplankton growth rate) (gC/gC/d)
0.05 ! Rp (respiratory losses from phytoplankton) (%/d)
0.05 ! Mp (mortality losses from phytoplankton) (%/d)
0.05 ! Sp (sinking losses from phytoplankton) (%/d)
10.0 ! Ik (half-saturation constant for photosynthesis) (E/m²/day)
5.0 ! Kz (depth of the mixed layer) (m)
26.0 ! Io (surface light intensity) (E/m²/d)
0.4 ! Kc (self-shading effects of phytoplankton) (m²/gC)
0.2 ! Kw (extinction coefficient for light attenuation (/m)
1.0 ! Zmax (maximum ingestion(growth) rate) (gC/gC/d)
0.70 ! Az (assimilation efficiency) (%/d)
0.25 ! Rz (respiratory losses from zooplankton) (%/d)
0.25 ! Pz (higher predator losses from zooplankton (%/d)
0.7 ! Ppref (preference for phytoplankton over detritus) (unitless)
0.10 ! k3 (half-saturation constant for grazing rate (gC/m³)
0.20 ! Rd (reminalization rate of detritus into nitrogen) (%/d)
0.05 ! Sd (sinking rate of detritus out of the mixed layer (%/d)
0.20 ! Mz (linear mortality coeff)
1.0 ! Mz (quadratic mortality coeff)
0.1 ! Fz (excretion losses for fish) (%/d)
0.4 ! Ff (combined fecal and unassimilated losses for fish) (%/day)

!This file contains commonly used simulation parameters used by some or all of the model subroutines and main program
! This file must be called input.dat

2 !Simulation length (# days), divide by DT to get number of time steps
1024 !Numrow
1024 !Numcol
10 !Map level, have to coordinate with size of map (6=64x64 map)
2 !Visualization (1=on 2=off)
5 !Xscale for visualization (1=(0-.01),2=(0-.1),3=(0-1),4=(0-10), 5=min/max)

```

1      !Granularity for contour graphs
1      !Pick y coordinate for transect (bottom to top counting)
-513341332  !Random number seed input value
1      !Type of initialization (1=multifractal init, 2=regular init, 3=cos init)
1      !Which multifractal init(1=n,2=p,3=z,4=d)
1      !Advection switch (1=user input constant ,2=correlation)
4      !Direction of advection (1=SW,2=W,3=NW,4=S,5=N,6=SE,7=E,8=NE)
1      !Autocorrelation line, 1=fixed, 2=max whole sim 3=max each step
2      !Maptype for multifractal 1=gradient 2=regmultifractal
1      !Wrapped boundaries or not 1=wrapped boundaries 2=not wrapped
1      !Type of depth initialization (1=reg,2=multifract,3=cos)
1      !Type of nutrient pool initialization(1=reg,2=multifract,3=cos)
1      !Type of constant nutrient input init(1=reg,2=multifract,3=cos)
144    !Frequency of file saving (based on DT)
60000  !Start of nutrient spike (based on DT)
60000  !End of nutrient spike (based on DT)
10000  !How often spike within time-frame above (ex: every 10 DT)
10000  !Start of fish predation (based on DT)
70000  !End of fish predation (based on DT)
10000  !How often fish predate within time-frame above
70000  !Start of nutrient movement (based on DT)
70000  !End of nutrient movement (based on DT)
70000  !How often move nutrients within time-frame above
70000  !Start of fish movement (based on DT)
70000  !End of fish movement (based on DT)
10000  !How often move fish within time-frame above
5      !Size of spot (1=spot,10=uniform,2-9 decreasing size,11=very small) nut
1      !Size of spot (1=spot,10=uniform,2-9 decreasing size) fish
2      !Type of zooplankton mortality (1=linear, 2=quadratic)
2      !Type of zooplankton pred functional response (1=MM, 2=Holling3)
1      !Number of repetitions for the sensitivity analysis (set to 1 if not performing one)
80     !# of parameters for sen analysis * how often output (set to 4 if only want at end)
2      !Flag for spatial or temporal sensitivity analysis (1=spatial, 2=temporal)
3      !Switch for sensitivity (only for temporal sensitivity analysis)
2      !Diffusion switch(2=off)
2      !Advection switch(2=off)
1      !File save switch(2=off)
1      !Value for minimum radius (cannot be greater than numrow/4)
1      !Switch for random or prescribed turbulence (1=random 2=prescribed)
60     !Xlocation for first eddy of prescribed
70     !Xlocation for second eddy of prescribed
64     !Ylocation for first eddy
64     !Ylocation for second eddy
4      !Line initialization (1=line,2=spots,3=one spot,4=blank) need reg init also
2      !Switch to turn turbulence on and off(2=off)
1      !Switch to turn radius keeping on/off (2=off)-inside circle moves faster than out

```

2 !Switch to turn move timing on and off (2=off)=big/small circles move same)
2 !Switch to turn off biology (2=off)
4 !Boundary cond(1=abs,2=reflecting,3=wrapped 4=nut chapter) only for diffusion
3 !Set nut input map flag (only for nut chapter) 1=multifracta,2=uniform,3=none
3 !Determines freq of advection (DT)-nut chapter-want advection heading south
3 !Determines frequency of input for nut map (DT) has to be same as advec freq
1 !Nutrient chapter switch (if turn on =1) then normal stats are turned off
2 !Switch to turn on or off stats for (sd,cv,var) (1=on,2=off)
14 !Turbulence freq
1 !Min_radius
256 !Max_radius
1 !Start of map file saving
0.02 !Initial value for N
0.03 !Initial value for P
0.056 !Initial value for Z
0.160 !Initial value for D
0.01146 !Diffusion N
0.01146 !Diffusion P
0.1146 !Diffusion Z
0.1147 !Diffusion D
1.00 !Advection N
1.00 !Advection P
1.00 !Advection Z
1.00 !Advection D
1.0 !Turbulence N
1.0 !Turbulence P
1.0 !Turbulence Z
1.0 !Turbulence D
0.99 !Clumping on multifractal maps
0.33 !Cv for multifractal map (do not exceed 33%-(0-.33))
0.0 !Diffusion rate across thermocline
0.30 !Concentration of N below thermocline
60.0 !Correlation angle for advection
1.0 !Max for cos init
1.0 !Max depth for cos init
1.0 !Max for nutrient pool cos init
1.0 !Max for nutrient constant input cos init
10.0 !Correlation angle for spot movement nut
0.0 !Distance to move spot each instance nut
10.0 !Correlation angle for spot movement fish
0.0 !Distance to move spot each instance fish
0.0 !Amount of nutrient input per event (gN/m3) per DT
0.0 !Amount of fish predation per event (gC/m3) removed per DT
0.003472 !DT time-step of model
1.0 !Mean value for the nutrient input map (for nut input chapter)
0.002 !Mean value for nutrient input map (mean value outside simulation domain)

0.380 !Mean phyto input value due to advection enters top of map
0.056 !Mean zoo input value due to advection enters top of map
0.160 !Mean detrital input value due to advection enters top of map

Main Program

Below is the code for the main program of the SLS platform. This program reads in the input files and calls the various subroutines responsible for all of the processes included within the platform, and controls the outer time loop of the simulation platform.

```
PROGRAM SLS
```

```
    USE global_variables  
    IMPLICIT NONE
```

```
    INTEGER :: irep,iii
```

```
! Read in initial conditions from (input.dat) and (inputpar.dat) files
```

```
    CALL initval
```

```
! Open files which will store simulation output
```

```
    CALL fileopen
```

```
! Open up files for the sensitivity analysis if needed (set nrep to 1 if don't want sensitivity analysis)
```

```
    IF (nrep .ne. 1) THEN  
        open (41, FILE='pr2in.dat',STATUS='old')  
        open (42, FILE='pr3in.dat',STATUS='unknown')  
    END IF
```

```
! Begin outer loop of simulation (depends on number of repetitions for sensitivity analysis)
```

```
! If not doing sensitivity analysis this loop will disappear-done only once
```

```
    DO irep = 1, nrep
```

```
! Read in the input values from the sensitivity analysis data file if necessary
```

```
    IF (nrep .ne. 1) THEN  
        CALL prism_Rd(irep)  
    END IF
```

```
! Create map initialization and initialize movement routines
```

```
    CALL mapinit  
    call nutmove  
    call fishmove
```

```
! Initialize counters and arrays
```

```
    counter = 0  
    stat_counter = 0
```

```
counter_nut = 0
counter_fish = 0
map_location = 1
```

```
Nmax=0.0
Nmin=0.0
Pmax=0.0
Pmin=0.0
Zmax=0.0
Zmin=0.0
Dmax=0.0
Dmin=0.0
Nvar=0.0
Pvar=0.0
Zvar=0.0
Dvar=0.0
Nsd=0.0
Psd=0.0
Zsd=0.0
Dsd=0.0
Ncv=0.0
Pcv=0.0
Zcv=0.0
Dcv=0.0
Nave=0.0
Pave=0.0
Zave=0.0
Dave=0.0
```

```
! If running turbulence create distribution of eddies and save
  IF (tur_switch .eq. 1) THEN  !done so that don't get deallocation error when
    CALL eddy_generation(numcol,iseed_input)
  END IF
```

```
! Time loop of the simulation
```

```
DO WHILE (counter .LE. tmax)

  simtime = float(counter)*DT
  write (70,*) 'day=', simtime
```

```
! Calculates spot movement-input, fish movement-input, and how often save files
  call switches(irep)
```

```
! Contains the biological interactions
```

```

        IF (bio_switch .eq. 1) THEN
            CALL biology
        END IF

! Calculates turbulent movement
        IF (tur_switch .eq. 1) THEN
            DT_trigger = ABS(MOD(counter,TurbulentFreq))
            IF (DT_trigger .eq. 0) THEN
                CALL
                turbulentmixing(counter,max_radius,min_radius)
            END IF
        END IF

! Calculates advective movement
        IF (adv_switch .eq. 1) THEN
            DT_trigger = ABS(MOD(counter,AdvectionFreq))
            IF (DT_trigger .eq. 0) THEN
                CALL advect(N,advcoeff(1),1)
                CALL advect(P,advcoeff(2),2)
                CALL advect(Z,advcoeff(3),3)
                CALL advect(D,advcoeff(4),4)
            END IF
        END IF

! Calculates diffusive movement
        IF (diff_switch .eq. 1) THEN
            CALL diffusion(N,diffcoeff(1))
            CALL diffusion(P,diffcoeff(1))
            CALL diffusion(Z,diffcoeff(1))
            CALL diffusion(D,diffcoeff(1))
        END IF

! Calculates map nutrient input frequency (only for nut input chapter)
! Only do this if running nutrient input map
        IF (map_input .ne. 3) THEN
            DT_trigger = ABS(MOD(counter,NutInputFreq))
            IF (DT_trigger .eq. 0) THEN
! Wraps map to beginning if reaches end
                IF (map_location .gt. numrow) THEN
                    map_location = 1
                END IF
! Read map location and input as nutrient map starting at the northern edge
                N(1:numcol,numcol) = Nutinput(1:numcol,map_location)
                map_location = map_location + 1
            END IF
        END IF

```



```
! Updates loop (based on DT) and counter
      counter = counter + 1

      END DO

!End of simulation timeloop

! If running sensitivity analysis write key outputs to prism file-this file will be input into
prism 3
! for temporal sensitivity analysis-linked with how often save files

      IF (nrep .ne. 1) THEN
            CALL Prism_out(irep)
      END IF

      END DO      ! end prism loop

! Close open files and deallocate arrays
      CALL cleanup

END PROGRAM
```

Data Input Routines

This subroutine reads in the data provided in the two input files and assigns the value to the corresponding model variable.

```
SUBROUTINE initval
```

```
! Reads in data from input file
! Input file is separated into (integer) and (real) data
```

```
USE global_variables
USE map_variables
IMPLICIT NONE
```

```
INTEGER :: iii,fileposition
INTEGER :: UserDataInteger(100)
REAL :: UserDataReal(100)
REAL :: UserInput(100)
```

```
OPEN (20, FILE='input.dat', STATUS='old')
OPEN (21, FILE='inputpar.dat', STATUS='old')
```

```
! Integer data input
```

```
  iii=1
  DO fileposition=iii,64
    READ (20,FMT=*) UserDataInteger(iii)
    iii=iii+1
  END DO
```

```
! Real data input
```

```
  iii=1
  DO fileposition=iii,37
    READ (20,FMT=200) UserDataReal(iii)
    iii=iii+1
  END DO
  200 format (F6.6)
```

```
! Read data from second input file (inputpar.dat)
```

```
  iii=1
  DO fileposition=iii,23
    READ (21,FMT=300) UserInput(iii)
    iii=iii+1
  END DO
  300 format (F6.6)
```

CLOSE (20)

CLOSE (21)

! Use data to initialize simulation variables

! Integer input data

```
simlength = UserDataInteger(1)
numrow = UserDataInteger(2)
numcol = UserDataInteger(3)
maxlevel_input = UserDataInteger(4)
visualization = UserDataInteger(5)
scaleviz = UserDataInteger(6)
granular = UserDataInteger(7)
ylocationcoor = UserDataInteger(8)
iseed_input = UserDataInteger(9)
fractalinit = UserDataInteger(10)
stateinit = UserDataInteger(11)
advectionswitch = UserDataInteger(12)
zzzz = UserDataInteger(13)
transectlocation = UserDataInteger(14)
maptype_input = UserDataInteger(15)
wrap_input = UserDataInteger(16)
depth_input = UserDataInteger(17)
pool_input = UserDataInteger(18)
nut_input = UserDataInteger(19)
FileSaveFreq = UserDataInteger(20)
StartNutSpike = UserDataInteger(21)
EndNutSpike = UserDataInteger(22)
NutSpikeFreq = UserDataInteger(23)
StartFishPred = UserDataInteger(24)
EndFishPred = UserDataInteger(25)
FishPredFreq = UserDataInteger(26)
StartNutMove = UserDataInteger(27)
EndNutMove = UserDataInteger(28)
NutMoveFreq = UserDataInteger(29)
StartFishMove = UserDataInteger(30)
EndFishMove = UserDataInteger(31)
FishMoveFreq = UserDataInteger(32)
nutradius = UserDataInteger(33)
Fishradius = UserDataInteger(34)
MzSwitch = UserDataInteger(35)
M = UserDataInteger(36)
nrep = UserDataInteger(37)
nval = UserDataInteger(38)
```

```

senflag = UserDataInteger(39)
senswitch = UserDataInteger(40)
diff_switch = UserDataInteger(41)
adv_switch = UserDataInteger(42)
file_switch = UserDataInteger(43)
min_radius2 = UserDataInteger(44)
turrans = UserDataInteger(45)
x1_input = UserDataInteger(46)
x2_input = UserDataInteger(47)
y1_input = UserDataInteger(48)
y2_input = UserDataInteger(49)
line_init = UserDataInteger(50)
tur_switch = UserDataInteger(51)
radius_flag = UserDataInteger(52)
turmove_flag = UserDataInteger(53)
bio_switch = UserDataInteger(54)
boundary_input = UserDataInteger(55)
map_input = UserDataInteger(56)
AdvectionFreq = UserDataInteger(57)
NutInputFreq = UserDataInteger(58)
Nut_chapter_switch = UserDataInteger(59)
stat_switch = UserDataInteger(60)
TurbulentFreq = UserDataInteger(61)
min_radius = UserDataInteger(62)
max_radius = UserDataInteger(63)
map_save = UserDataInteger(64)

DT = UserDataReal(32) !DT for simulation
tmax = INT(float(simlength)/DT)
circle_dimension = INT(numcol/min_radius)**1.82
output_dimension = INT(tmax/FileSaveFreq) + 10

```

! Allocate arrays based on simulation length and mapsize

call allocation

```

N = UserDataReal(1)
P = UserDataReal(2)
Z = UserDataReal(3)
D = UserDataReal(4)

```

! Real input data

```

diffcoeff(1) = UserDataReal(5)
diffcoeff(2) = UserDataReal(6)
diffcoeff(3) = UserDataReal(7)

```

diffcoeff(4) = UserDataReal(8)
advcoeff(1) = UserDataReal(9)
advcoeff(2) = UserDataReal(10)
advcoeff(3) = UserDataReal(11)
advcoeff(4) = UserDataReal(12)
turcoeff(1) = UserDataReal(13)
turcoeff(2) = UserDataReal(14)
turcoeff(3) = UserDataReal(15)
turcoeff(4) = UserDataReal(16)
hclump_input = UserDataReal(17)
cv_input = UserDataReal(18)
Dk = UserDataReal(19)
No = UserDataReal(20)
coorangle = UserDataReal(21)
cos_map_max = UserDataReal(22)
depth_max = UserDataReal(23)
pool_max = UserDataReal(24)
nut_max = UserDataReal(25)
angle_nut = UserDataReal(26)
distance_nut = UserDataReal(27)
angle_fish = UserDataReal(28)
distance_fish = UserDataReal(29)
NutSlugAmount = UserDataReal(30)
FishExt = UserDataReal(31)
Nut_input_amount = UserDataReal(33)
Init(1) = UserDataReal(34)
Init(2) = UserDataReal(35)
Init(3) = UserDataReal(36)
Init(4) = UserDataReal(37)

CN = UserInput(1)
Kn = UserInput(2)
Pmax = UserInput(3)
Rp = UserInput(4)
Mp = UserInput(5)
Sp = UserInput(6)
Ik = UserInput(7)
Kz = UserInput(8)
Io = UserInput(9)
Kc = UserInput(10)
Kw = UserInput(11)
Zmax = UserInput(12)
Az = UserInput(13)
Rz = UserInput(14)
Pz = UserInput(15)
Ppref = UserInput(16)

```
k3 = UserInput(17)
Rd = UserInput(18)
Sd = UserInput(19)
linmort = UserInput(20)
quadmort = UserInput(21)

IF (MzSwitch .eq. 1) THEN
    Mz = linmort
END IF
IF (MzSwitch .eq. 2) THEN
    Mz = quadmort
END IF

Fz = UserInput(22)
Ff = UserInput(23)

END SUBROUTINE
```

NPZD Module

The equations and solution technique associated with the NPZD model utilized in the dissertation chapters is contained within the biology subroutine below. Initial values for the state variables and constants are obtained from the two input files.

SUBROUTINE biology

USE global_variables
IMPLICIT NONE

INTEGER :: j,k

! Loop over map and solve Eulers method based on the DT

DO j = 1, numrow
DO k = 1, numcol

! Functions

Nutrient = N(j,k) / (Kn + N(j,k))
Kd = Kw + Kc*P(j,k)
Light = LOG((In + Io)/(In + Io*EXP(-Kd*Kz))) / (Kd*Kz)
ratelimit = MIN(Nutrient,Light)
PhytoGrowth(j,k) = Pmaxg*ratelimit*P(j,k)
Drate = ((1-Ppref)*D(j,k)) / ((Ppref*P(j,k)) + ((1-Ppref)*D(j,k)))
Prate = (Ppref*P(j,k)) / ((Ppref*P(j,k)) + ((1-Ppref)*D(j,k)))
Dgraz = Zmax*((Drate*D(j,k))**M / ((k3**M) + ((Prate*P(j,k)) + (Drate*D(j,k)))**M))
Pgraz = Zmax*((Prate*P(j,k))**M / ((k3**M) + ((Prate*P(j,k)) + (Drate*D(j,k)))**M))

!For numerical stability

IF (Dgraz .lt. 0.0) THEN
Dgraz = 0.0
END IF

IF (Pgraz .lt. 0.0) THEN
Pgraz = 0.0
END IF

! Inflows/Outflows

! Zooplankton State Variable

$$\text{ZooGrowth}(j,k) = (\text{Pgraz} + \text{Dgraz}) * \text{Az} * \text{Z}(j,k)$$

$$\text{ZooResp} = \text{Rz} * \text{Z}(j,k)$$

$$\text{ZooMort} = \text{Mz} * (\text{Z}(j,k) ** \text{MzSwitch})$$

$$\text{ZooPred} = \text{Pz} * \text{Z}(j,k)$$

$$\text{FishResp} = \text{ZooPred} * \text{Fz}$$

$$\text{FishFecal} = \text{ZooPred} * \text{Ff}$$

! Nutrient State Variable

$$\text{Input} = \text{Dk} * (\text{No} - \text{N}(j,k))$$

$$\text{Regeneration} = ((\text{Rp} * \text{P}(j,k)) + (\text{Rz} * \text{Z}(j,k)) + \text{Fz} * \text{ZooPred}) * \text{CN}$$

$$\text{Remineralization} = (\text{Rd} * \text{D}(j,k)) * \text{CN}$$

$$\text{Uptake} = \text{PhytoGrowth}(j,k) * \text{CN}$$

! Phytoplankton State Variable

$$\text{PhytoSink} = \text{Sp} * \text{P}(j,k)$$

$$\text{PhytoResp} = \text{Rp} * \text{P}(j,k)$$

$$\text{PhytoMort} = \text{Mp} * \text{P}(j,k)$$

$$\text{PhytoPred} = \text{Pgraz} * \text{Z}(j,k)$$

! Detritus State Variable

$$\text{NaturalMort} = ((\text{Mp} * \text{P}(j,k)) + (\text{Mz} * (\text{Z}(j,k) ** \text{MzSwitch})))$$

$$\text{ZooFecal} = (1 - \text{Az}) * (\text{Dgraz} + \text{Pgraz}) * \text{Z}(j,k)$$

$$\text{Remineralize} = \text{Rd} * \text{D}(j,k)$$

$$\text{DetritalSink} = \text{Sd} * \text{D}(j,k)$$

$$\text{DetritalPred} = \text{Dgraz} * \text{Z}(j,k)$$

! New Growth

$$\text{Nnew} = (\text{Input} + \text{Regeneration} + \text{Remineralization} - \text{Uptake}) * \text{DT}$$

$$\text{Pnew}(j,k) = (\text{PhytoGrowth}(j,k) - \text{PhytoSink} - \text{PhytoResp} - \text{PhytoMort} - \text{PhytoPred}) * \text{DT}$$

$$\text{Znew}(j,k) = (\text{ZooGrowth}(j,k) - \text{ZooResp} - \text{ZooMort} - \text{ZooPred}) * \text{DT}$$

$$\text{Dnew} = (\text{NaturalMort} + \text{ZooFecal} + \text{FishFecal} - \text{Remineralize} - \text{DetritalSink} - \text{DetritalPred}) * \text{DT}$$

! Updated Growth

$N(j,k) = N(j,k) + N_{new}$

$P(j,k) = P(j,k) + P_{new}$

$Z(j,k) = Z(j,k) + Z_{new}$

$D(j,k) = D(j,k) + D_{new}$

IF (N(j,k) .lt. 0.0) THEN

 N(j,k) = 0.0

END IF

IF (P(j,k) .lt. 0.0) THEN

 P(j,k) = 0.0

END IF

IF (Z(j,k) .lt. 0.0) THEN

 Z(j,k) = 0.0

END IF

IF (D(j,k) .lt. 0.0) THEN

 D(j,k) = 0.0

END IF

 END DO

END DO

END SUBROUTINE

Turbulent Mixing Modules

These subroutine conduct the calculations necessary to implement turbulent mixing between the grid cells of the SLS platform. See chapter 3 for additional information on each of the various components utilized.

```
SUBROUTINE eddy_generation(maplength,iseed)
```

```
! This routine creates the initial distribution of eddies and saves them in an array for future use
```

```
    USE global_variables  
    IMPLICIT NONE
```

```
    INTEGER :: i,j,k  
    REAL :: ran1, ratio  
    INTEGER :: eddy_level, eddy_radius  
    INTEGER :: iseed, maplength
```

```
! Determine the number of eddies to generate based on the mapsize  
! For this routine the mapsize must be a power of 2  
! Always generate a full cascade starting at (r=1/4 mapsize) down to (r=1 pixel)  
! Can truncate the cascade in the next subroutine by specifying the max/min radius  
! Eddy sizes will decrease by 1/2 with each cascade level  
! Since the eddies at each size will be space filling the number of eddies  
! at each level goes up by 4 raised to the level
```

```
    eddy_level = 0  
    totalcir = 0  
    eddy_radius = maplength/4
```

```
    DO WHILE (eddy_radius .GE. 1)  
        eddy_level = eddy_level + 1  
        eddy_radius = eddy_radius/2  
        totalcir = totalcir + 4**eddy_level  
    END DO
```

```
! Allocate storage arrays based on the total number of eddies  
! Declare these variables outside of this subroutine so that don't have to pass huge arrays)  
! totalcir variable also needs to be declared externally
```

```
    ALLOCATE (xcenter(totalcir),ycenter(totalcir),rotation(totalcir),&  
             circlesize(totalcir),starttime(totalcir),freqswitch(totalcir))
```

```

ALLOCATE
(cascade_switch(totalcir),cascade_trigger(totalcir),radius_stop_location(totalcir))

! Give each eddy a random X,Y center except the (X) center can be wrapped (left and
right) and the
! (Y) center has to be within the simulation domain (can't go over the top and bottom
edge)
! Each eddy is also given a random rotation, and a starttime for rotation
! Eddies in arrays are arranged from the largest circles down to the smallest

      k = 0
      eddy_radius = maplength/4

      DO i = 1, eddy_level
        DO j = 1, 4**i  !loop for the number of eddies in that level
          k = k + 1
          circlesize(k) = eddy_radius  !assign an eddy radius

! Initialize the (Y) coordinate to (1,1) to force the routine to pick a random location for
each new circle

          ycenter(k) = 1

! Keep on randomly creating (Y) circle centers till condition is meet

          DO WHILE (ycenter(k) .LE. eddy_radius .OR. ycenter(k) .GE.
(maplength-eddy_radius))
            ycenter(k) = Int(ran1(iseed) * float(maplength-1)) + 1
          END DO

! Randomly choose an (X) center

          xcenter(k) = Int(ran1(iseed) * float(maplength-1)) + 1

! Randomly choose a rotation
! Routine produces value between (0-1) + 1 = 1 and 2 (1=clockwise, 2=counterclockwise)

          rotation(k) = Int(ran1(iseed) * float(2)) + 1

! Randomly choose a start time based on an interval determined by the largest eddy

          starttime(k) = Int(ran1(iseed) * float(maplength/4-1)) + 1

! Determine frequency of oration based on the largest eddy and a 1/3 scaling of energy
dissipation rate

```

```

        ratio = (float(eddy_radius)/float(maplength/4))
        freqswitch(k) = Int((ratio**0.333)*float(maplength/4))
    END DO
    eddy_radius = eddy_radius/2
END DO

```

END SUBROUTINE

SUBROUTINE turbulentmixing(timestep,maxradius,minradius)

! This subroutine determines which eddies to rotate can contains a call to the subroutine
! which does the actual rotation of the eddy. The routine can be called each timestep or at
! some frequency of the basic timestep
! Only cycle through the circles specified by the maxradius and minradius user inputs
! This will allow partial turbulent cascades to be simulated

```

    USE global_variables
    IMPLICIT NONE

```

```

    INTEGER :: i, j
    INTEGER :: timestep, maxradius, minradius

```

```

!loop over all the circles
    DO i = 1, totalcir
!check for random starttime

```

```

        IF (timestep .ge. starttime(i) .and. circlesize(i) .le. maxradius .and.
circlesize(i) .ge. minradius) THEN
!if trigger=0 then start cascade
!reset the radius stop location for that circle to the outer edge
            IF (cascade_switch(i) .eq. 0) THEN
                cascade_trigger(i) = ABS(MOD((timestep-
starttime(i)),freqswitch(i)))
                radius_stop_location(i) = circlesize(i)
            END IF

```

```

!start cascade if trigger condition is met
!reset cascade switch so that cascade will go down to a radius of 1
!reset temporary radius counter to edge of circle each time
            IF (cascade_trigger(i) .eq. 0) THEN
                cascade_switch(i)=1
                j=circlesize(i)

```

```

!cascade inward till reach radius stop location

```

```

DO WHILE (j .GE. radius_stop_location(i))

!Call turbulence movement routine (needs the x and y center, rotation, temporary location
on circle(j)
!Also needs turbulence exchange rate (typically will be 1.0)
!This routine does all the number crunching

CALL
turbulence_move(turcoeff(1),j,xcenter(i),ycenter(i),rotation(i),N,numcol)
CALL
turbulence_move(turcoeff(2),j,xcenter(i),ycenter(i),rotation(i),P,numcol)
CALL
turbulence_move(turcoeff(3),j,xcenter(i),ycenter(i),rotation(i),Z,numcol)
CALL
turbulence_move(turcoeff(4),j,xcenter(i),ycenter(i),rotation(i),D,numcol)

!reduce temporary location on circle by 1
j=j-1

!cascade has reached the innermost radius so exit and start a new cascade trigger check
cycle
IF (j .eq. 0) THEN
cascade_switch(i) = 0
END IF

END DO

!decrease radius stop location by 1
radius_stop_location(i) = radius_stop_location(i) - 1

END IF
END IF
END DO

END SUBROUTINE

```

```

SUBROUTINE turbulence_move(turrate,r,x,y,dir,map,sidelength)

```

```

! This routine does all the numerical processing for the turbulence movement. Each call
!will only move one ring of cells of a specific radius for a specific circle
! The method has complete conservation of material
! Map is the state variable of interest that you want to simulate should be a global
!variable and declared outside of the routine

```

```
USE global_variables
IMPLICIT NONE
```

```
INTEGER :: i,r,x,y,dir
REAL :: Past, Future, turrate
INTEGER :: xloc,yloc
INTEGER :: rplus1,rminus1
INTEGER :: xloc_past,yloc_past
INTEGER :: sidelength
REAL :: map(sidelength,sidelength)
```

```
! The code is broken down into clockwise and counterclockwise movement routines
! Code also allows for wrapped boundaries which makes more complicated
! Routine is very efficient because only have to store one piece of information
! It just gets swapped between past, present, and future
```

```
! Always start the circle at the top left edge and recode variables to reduce complication
```

```
xloc = x-r
yloc = y-r
rplus1=r+r+1
rminus1=r+r-1
```

```
! Check for edge position (left edge and top)
```

```
IF (xloc .lt. 1) xloc = numcol+xloc
IF (yloc .lt. 1) yloc = numcol+yloc
```

```
! Set the temporary past variable
```

```
yloc_past = yloc + 1
```

```
! Check for edge
```

```
IF (yloc_past .gt. numcol) yloc_past = yloc_past-numcol
```

```
! Set the real past variable
```

```
past = map(xloc,yloc_past) * turrate
```

```
! Now simulate movement of cells around circle
! Have employed a cell swap routine-very efficient
! Only works because cells are moved in sequence
```

```
IF (dir .eq. 1) THEN !clockwise (to the right)
```

```

DO i = 1,rplus1 !across top
  future = map(xloc,yloc) * turrate
  map(xloc,yloc) = (map(xloc,yloc) - future) + past
  past = future
  xloc=xloc+1
  IF (xloc .gt. numcol) xloc = xloc-numcol !Check for edge
END DO

```

```

DO i = 1,rminus1 !down right side
  future = map(xloc,yloc) * turrate
  map(xloc,yloc) = (map(xloc,yloc) - future) + past
  past = future
  yloc=yloc+1
  IF (yloc .gt. numcol) yloc = yloc-numcol !Check for edge
END DO

```

```

DO i = 1,rplus1 !along bottom
  future = map(xloc,yloc) * turrate
  map(xloc,yloc) = (map(xloc,yloc) - future) + past
  past = future
  xloc=xloc-1
  IF (xloc .lt. 1) xloc = numcol+xloc !Check for edge
END DO

```

```

DO i = 1,rminus1 !up left side
  future = map(xloc,yloc) * turrate
  map(xloc,yloc) = (map(xloc,yloc) - future) + past
  past = future
  yloc=yloc-1
  IF (yloc .lt. 1) yloc = numcol+yloc !Check for edge
END DO

```

ELSE !turn counter-clockwise (to the left) same code as above except in opposite direction

```

xloc = x-r
yloc = y-r

```

```

IF (xloc .lt. 1) xloc = numcol+xloc
IF (yloc .lt. 1) yloc = numcol+yloc
xloc_past = xloc + 1
IF (xloc_past .gt. numcol) xloc_past = xloc_past-numcol
past= map(xloc_past,yloc) * turrate

```

```

DO i = 1,rplus1 !down left side

```

```

        future = map(xloc,yloc) * turrate
        map(xloc,yloc) = (map(xloc,yloc) - future) + past
        past = future
        yloc=yloc+1
        IF (yloc .gt. numcol) yloc = yloc-numcol  !Check for edge
    END DO

```

```

    DO i = 1,rminus1  !along bottom
        future = map(xloc,yloc) * turrate
        map(xloc,yloc) = (map(xloc,yloc) - future) + past
        past = future
        xloc=xloc+1
        IF (xloc .gt. numcol) xloc = xloc-numcol  !Check for edge
    END DO

```

```

    DO i = 1,rplus1  !up right side
        future = map(xloc,yloc) * turrate
        map(xloc,yloc) = (map(xloc,yloc) - future) + past
        past = future
        yloc=yloc-1
        IF (yloc .lt. 1) yloc = numcol+yloc      !Check for edge
    END DO

```

```

    DO i = 1,rminus1  !along top
        future = map(xloc,yloc) * turrate
        map(xloc,yloc) = (map(xloc,yloc) - future) + past
        past = future
        xloc=xloc-1
        IF (xloc .lt. 1) xloc = numcol+xloc      !Check for edge
    END DO

```

END IF

END SUBROUTINE

Diffusion Module

This subroutine does the calculations necessary to implement diffusive mixing between the grid cells of the SLS platform. It is based on Fickian diffusion methods. The different cases allow various boundary conditions to be simulated

! This routine calculates the diffusive losses based on an eddy diffusivity
! An eddy diffusivity value is transformed into a percent loss flux in one direction
! based on the time-step and the grainsize
! This value is then multiplied by 4 to get the loss that would occur in 4 directions which
! then becomes the value for the diffusion coefficient used as input to the model !
and passed to this subroutine
! Losses only occur to the adjacent 4 cells because exchanges are based on a grid system

! Numerical constraint !!!!!!!

! The percent loss per time-step in one direction should not exceed (0.2 = 20%) so that
!! the total flux passed to this routine does not exceed 80%. This is to ensure that
!!! the spread of a passive substance resembles a normal distribution.

USE global_variables
IMPLICIT NONE

INTEGER :: i,j,xxx,yyy,north,east,south,west,totval
REAL :: diff
REAL :: Yvalues(numrow,numcol), v(numrow,numcol)

! Multiply entire Yvalues matrix by loss (percent per time-step) due to diffusion in the 4
cardinal directions
! Put these values into a separate v matrix and subtract from the original values
! Divide v matrix by 4 to get the loss in each direction (will be the same)
! Need the v matrix so that can update all the values at one time

v=0
v = diff * (Yvalues)
Yvalues = (Yvalues - v)
v = v/4

! Determine location on map
! Use location to determine what case to run

! Outer select case gives the boundary conditions based on an input file
! case 1 = reflecting, case 2 = absorbing, case 3 = wrapped , case 4 = nutinput chapter
(modified absorbing,wrapped)

```

! Update sequence for v (from the north, from the south, from the west, from the east)
! Yvalues(i,j) = Yvalues(i,j) + v(i,j+1) + v(i,j-1) + v(i-1,j) + v(i+1,j)
!           (from N) (from S) (from W) (from E)

```

```

        SELECT CASE (boundary_input)

```

```

CASE (1) !absorbing boundaries

```

```

! Upper left corner

```

```

        Yvalues(1,numcol) = Yvalues(1,numcol) + v(1,numcol-1) +
        v(2,numcol)

```

```

! Upper right corner

```

```

        Yvalues(numrow,numcol) = Yvalues(numrow,numcol) +
        v(numrow,numcol-1) + v(numrow-1,numcol)

```

```

! Lower left corner

```

```

        Yvalues(1,1) = Yvalues(1,1) + v(1,2) + v(2,1)

```

```

! Lower right corner

```

```

        Yvalues(numrow,1) = Yvalues(numrow,1) + v(numrow,2) +
        v(numrow-1,1)

```

```

! Do just the top edge of map

```

```

        DO i=2,numrow-1

```

```

            Yvalues(i,numcol) = Yvalues(i,numcol) + v(i,numcol-1) + v(i-1,numcol) +
            v(i+1,numcol)

```

```

        END DO

```

```

! Do just the bottom edge

```

```

        DO i=2,numrow-1

```

```

            Yvalues(i,1) = Yvalues(i,1) + v(i,2) + v(i-1,1) + v(i+1,1)

```

```

        END DO

```

```

! Do just the left side

```

```

        DO j=2,numcol-1

```

```

            Yvalues(1,j) = Yvalues(1,j) + v(1,j+1) + v(1,j-1) + v(2,j)

```

```

        END DO

```

```

! Do just the right side

```

```

        DO j=2,numcol-1

```

```

            Yvalues(numrow,j) = Yvalues(numrow,j) + v(numrow,j+1) + v(numrow,j-1) +
            v(numrow-1,j)

```

```

        END DO

```

```

! Do just the interior of the map

```

```

        DO i=2,numrow-1

```

```

            DO j=2,numcol-1

```

```

                Yvalues(i,j) = Yvalues(i,j) + v(i,j+1) + v(i,j-1) + v(i-1,j) + v(i+1,j)

```

```

            END DO

```

```

        END DO

```

```

CASE (2) !reflecting boundaries

```

```

! Upper left corner
  Yvalues(1,numcol) = Yvalues(1,numcol) + v(1,numcol) + v(1,numcol-1) +
    v(1,numcol) + v(2,numcol)
! Upper right corner
  Yvalues(numrow,numcol) = Yvalues(numrow,numcol) + v(numrow,numcol) +
    v(numrow,numcol-1) + v(numrow-1,numcol) + v(numrow,numcol)
! Lower left corner
  Yvalues(1,1) = Yvalues(1,1) + v(1,2) + v(1,1) + v(1,1) + v(2,1)
! Lower right corner
  Yvalues(numrow,1) = Yvalues(numrow,1) + v(numrow,2) + v(numrow,1) +
    v(numrow-1,1) + v(numrow,1)
! Do just the top edge of map
  DO i=2,numrow-1
    Yvalues(i,numcol) = Yvalues(i,numcol) + v(i,numcol) + v(i,numcol-1) + v(i-1
,numcol) + v(i+1,numcol)
  END DO
! Do just the bottom edge
  DO i=2,numrow-1
    Yvalues(i,1) = Yvalues(i,1) + v(i,2) + v(i,1) + v(i-1,1) + v(i+1,1)
  END DO
! Do just the left side
  DO j=2,numcol-1
    Yvalues(1,j) = Yvalues(1,j) + v(1,j+1) + v(1,j-1) + v(1,j) + v(2,j)
  END DO
! Do just the right side
  DO j=2,numcol-1
    Yvalues(numrow,j) = Yvalues(numrow,j) + v(numrow,j+1) + v(numrow,j-1) +
    v(numrow-1,j) + v(numrow,j)
  END DO
! Do just the interior of the map
  DO i=2,numrow-1
    DO j=2,numcol-1
      Yvalues(i,j) = Yvalues(i,j) + v(i,j+1) + v(i,j-1) + v(i-1,j) + v(i+1,j)
    END DO
  END DO

```

CASE (3) !wrapped boundaries

```

! Do the corners of the map first
!upper left corner
  Yvalues(1,numcol) = Yvalues(1,numcol) + v(1,1) + v(1,numcol-1) +
    v(numrow,numcol) + v(2,numcol)
!upper right corner
  Yvalues(numrow,numcol) = Yvalues(numrow,numcol) + v(numrow,1) +
    v(numrow,numcol-1) + v(numrow-1,numcol) + v(1,numcol)
!bottom left corner

```

```

        Yvalues(1,1) = Yvalues(1,1) + v(1,2) + v(1,numcol) + v(numrow,1) + v(2,1)
!bottom right corner
        Yvalues(numrow,1) = Yvalues(numrow,1) + v(numrow,2) + v(numrow,numcol)
        + v(numrow-1,1) + v(1,1)
! Do just the top edge of map
        DO i=2,numrow-1
            Yvalues(i,numcol) = Yvalues(i,numcol) + v(i,1) + v(i,numcol-1) + v(i-
1,numcol) + v(i+1,numcol)
        END DO
! Do just the bottom edge
        DO i=2,numrow-1
            Yvalues(i,1) = Yvalues(i,1) + v(i,2) + v(i,numcol) + v(i-1,1) + v(i+1,1)
        END DO
! Do just the left side
        DO j=2,numcol-1
            Yvalues(1,j) = Yvalues(1,j) + v(1,j+1) + v(1,j-1) + v(numrow,j) + v(2,j)
        END DO
! Do just the right side
        DO j=2,numcol-1
            Yvalues(numrow,j) = Yvalues(numrow,j) + v(numrow,j+1) + v(numrow,j-1) +
v(numrow-1,j) + v(1,j)
        END DO
! Do just the interior of the map
        DO i=2,numrow-1
            DO j=2,numcol-1
                Yvalues(i,j) = Yvalues(i,j) + v(i,j+1) + v(i,j-1) + v(i-1,j) + v(i+1,j)
            END DO
        END DO

```

CASE (4) !wrapped boundaries (W-E), absorbing (N-S)

```

! Do the corners of the map first
!upper left corner
        Yvalues(1,numcol) = Yvalues(1,numcol) + v(1,numcol-1) +
        v(numrow,numcol) + v(2,numcol)
!upper right corner
        Yvalues(numrow,numcol) = Yvalues(numrow,numcol) + v(numrow,numcol-1)
+ v(numrow-1,numcol) + v(1,numcol)
!bottom left corner
        Yvalues(1,1) = Yvalues(1,1) + v(1,2) + v(numrow,1) + v(2,1)
!bottom right corner
        Yvalues(numrow,1) = Yvalues(numrow,1) + v(numrow,2) + v(numrow-1,1)
        + v(1,1)
! Do just the top edge of map
        DO i=2,numrow-1
            Yvalues(i,numcol) = Yvalues(i,numcol) + v(i,numcol-1) + v(i-1,numcol) +
v(i+1,numcol)
        END DO

```

```

        END DO
! Do just the bottom edge
        DO i=2,numrow-1
            Yvalues(i,1) = Yvalues(i,1) + v(i,2) + v(i-1,1) + v(i+1,1)
        END DO
! Do just the left side
        DO j=2,numcol-1
            Yvalues(1,j) = Yvalues(1,j) + v(1,j+1) + v(1,j-1) + v(numrow,j) + v(2,j)
        END DO
! Do just the right side
        DO j=2,numcol-1
            Yvalues(numrow,j) = Yvalues(numrow,j) + v(numrow,j+1) + v(numrow,j-1) +
v(numrow-1,j) + v(1,j)
        END DO
! Do just the interior of the map
        DO i=2,numrow-1
            DO j=2,numcol-1
                Yvalues(i,j) = Yvalues(i,j) + v(i,j+1) + v(i,j-1) + v(i-1,j) + v(i+1,j)
            END DO
        END DO

        END SELECT
END SUBROUTINE

```

Advection Modules

These subroutines move material between grid cells based on a correlated random walk methodology. The routines are extremely flexible allowing complex patterns of movement. These routines are also used to simulate spatially and temporally variable nutrient inputs and fish predation.

```
SUBROUTINE advectmovement
```

```
  USE global_variables
```

```
  IMPLICIT NONE
```

```
! Pick a random angle deviation (this will determine how correlated the values are
```

```
! Small (SD) will give a small turning radius
```

```
! Pick a random distance
```

```
! Do not allow negative values so take the absolute value of the returned result
```

```
! The (gauss) function returns a value between (-6 and 6) which are normally distributed
```

```
! most of the values fall between (-2 and 2) roughly 95%
```

```
! so a (SD) of (180) will give the occasional 360 degree turn
```

```
! pick an (SD) which is the same as the turning radius that you want
```

```
! well at least where you want most of the values to fall
```

```
  meanturn = 0.0
```

```
  newangle = (Gauss (0.0,1.0,iseed_input) * coorangle) + meanturn
```

```
! Add the new angle to the old angle from outside of the loop
```

```
  direction = direction + newangle
```

```
! If past (360) start a new circle
```

```
  direction = circle(direction)
```

```
  IF (direction .ge. 332.5 .and. direction .le. 360.0) THEN
```

```
    zzzz = 5
```

```
  END IF
```

```
  IF (direction .ge. 0.0 .and. direction .lt. 22.5) THEN
```

```
    zzzz = 5
```

```
  END IF
```

```
  IF (direction .ge. 22.5 .and. direction .lt. 67.5) THEN
```

```
    zzzz = 8
```

```
  END IF
```

```
  IF (direction .ge. 67.5 .and. direction .lt. 112.5) THEN
```

```

        zzzz = 7
    END IF

    IF (direction .ge. 112.5 .and. direction .lt. 157.5) THEN
        zzzz = 6
    END IF

    IF (direction .ge. 157.5 .and. direction .lt. 202.5) THEN
        zzzz = 4
    END IF

    IF (direction .ge. 202.5 .and. direction .lt. 247.5) THEN
        zzzz = 1
    END IF

    IF (direction .ge. 247.5 .and. direction .lt. 292.5) THEN
        zzzz = 2
    END IF

    IF (direction .ge. 292.5 .and. direction .lt. 332.5) THEN
        zzzz = 3
    END IF

END SUBROUTINE

```

SUBROUTINE advect(Yvalues,adrate,signal)

```

    USE global_variables
    IMPLICIT NONE

```

```

    REAL :: adrate, Yvalues(numrow,numcol)
    REAL :: scratchmatrix(numrow,numcol)
    INTEGER :: j,k
    INTEGER :: directionarray(8)
    INTEGER :: signal
    DATA directionarray/1,2,3,4,5,6,7,8/ ! 1=upperleft,3=upperright, etc

```

!user supplied switch on advection=number tells how often to switch advection
! use spot movement routine

```

    scratchmatrix=0
    scratchmatrix = adrate * (Yvalues)
    Yvalues = (Yvalues - scratchmatrix)

```

```

DO j=1,numrow
DO k=1,numcol

    SELECT CASE (directionarray(zzzz))

    CASE (1) !Southwest
    IF (j .eq. numrow .and. k .eq. numcol ) THEN !bottom right corner
        Yvalues(j,k) = Yvalues(j,k) + scratchmatrix(1,1)
    ELSE IF (j .eq. numrow) THEN !bottom
        Yvalues(j,k) = Yvalues(j,k) + scratchmatrix(1,k+1)
    ELSE IF (k .eq. numcol) THEN !right
        Yvalues(j,k) = Yvalues(j,k) + scratchmatrix(j+1,1)
    Else !all other places
        Yvalues(j,k) = Yvalues(j,k) + scratchmatrix(j+1,k+1)
    END IF

    CASE (2) !West
    IF (j .eq. numrow) THEN !bottom of map
        Yvalues(j,k) = Yvalues(j,k) + scratchmatrix(1,k)
    Else !all other places
        Yvalues(j,k) = Yvalues(j,k) + scratchmatrix(j+1,k)
    END IF

    CASE (3) !Northwest
    IF (j .eq. numrow .and. k .eq. 1) THEN !left corner
        Yvalues(j,k) = Yvalues(j,k) + scratchmatrix(1,numcol)
    ELSE IF (k .eq. 1) THEN !left
        Yvalues(j,k) = Yvalues(j,k) + scratchmatrix(j+1,numcol)
    ELSE IF (j .eq. numrow) Then !bottom
        Yvalues(j,k) = Yvalues(j,k) + scratchmatrix(1,k-1)
    ELSE !all other places
        Yvalues(j,k) = Yvalues(j,k) + scratchmatrix(j+1,k-1)
    END IF

    CASE (4) !South
    IF (k .eq. numcol) THEN !bottom of map !right edge of map
        !put in modification so that if goes off edge it gets absorbed
        !adds value from (Init)-mean level outside of simulation domain from
        north
        Yvalues(j,k) = Init(signal)
        ! Yvalues(j,k) = Yvalues(j,k) + scratchmatrix(j,1)
    Else !all other places
        Yvalues(j,k) = Yvalues(j,k) + scratchmatrix(j,k+1)
    END IF

    CASE (5) !North

```



```

IF (k .eq. 1) THEN !left edge of map
    Yvalues(j,k) = Yvalues(j,k) + scratchmatrix(j,numcol)
Else !all other places
    Yvalues(j,k) = Yvalues(j,k) + scratchmatrix(j,k-1)
END IF

CASE (6) !Southeast
IF (j .eq. 1 .and. k .eq. numcol) THEN !upper right corner
    Yvalues(j,k) = Yvalues(j,k) + scratchmatrix(numrow,1)
ELSE IF (j .eq. 1) THEN !top edge of map
    Yvalues(j,k) = Yvalues(j,k) + scratchmatrix(numcol,k+1)
ELSE IF (k .eq. numcol) THEN !right edge of map
    Yvalues(j,k) = Yvalues(j,k) + scratchmatrix(j-1,1)
ELSE !all other places
    Yvalues(j,k) = Yvalues(j,k) + scratchmatrix(j-1,k+1)
END IF

CASE (7) !East
IF (j .eq. 1) THEN !top of map
    Yvalues(j,k) = Yvalues(j,k) + scratchmatrix(numrow,k)
Else !all other places
    Yvalues(j,k) = Yvalues(j,k) + scratchmatrix(j-1,k)
END IF

CASE (8) !Northeast
IF (j .eq. 1 .and. k .eq. 1) THEN !upper left corner
    Yvalues(j,k) = Yvalues(j,k) + scratchmatrix(numrow,numcol)
ELSE IF (j .eq. 1) THEN !top side of map
    Yvalues(j,k) = Yvalues(j,k) + scratchmatrix(numrow,k-1)
ELSE IF (k .eq. 1) THEN !left side of map
    Yvalues(j,k) = Yvalues(j,k) + scratchmatrix(j-1,numcol)
ELSE !all other places
    Yvalues(j,k) = Yvalues(j,k) + scratchmatrix(j-1,k-1)
END IF

CASE DEFAULT
END SELECT

END DO
END DO

END SUBROUTINE

```

Multifractal Map Generation Routines

These routines generate the multifractal maps utilized in chapter 4. A wide variety of map can be generated from low correlation to high correlation as well as map approximating environmental gradients. The routine is adapted from the citation below and modified to interface with the SLS platform.

```
Subroutine hfraction(maxlevel,hclump,meanmap,cv,mapout,iseed_map)
```

```
! program written from algorithm MidPointFM2D in the
! Science of Fractal Images, M.F. Barnsley, R. L. Devaney
! B. B. Mandelbrot, H.-O. Peitgen, D. Saupe, and R. F. Voss
! Springer-Verlag
```

```
! maxprm = length of a side-needs to be a multiple of 2
! maxlevel = gives different sized maps-powers of 2
! not sure what maxcurd means at this point
```

```
INTEGER,PARAMETER :: maxprm = 2048
!INTEGER,PARAMETER :: maxlevel = 5 !gives and 8 sided map 2x2x2
!INTEGER,PARAMETER :: iseed_map = -12345678
```

```
! maxprm = 4 so that it gives a 16 cell map
  INTEGER maxlevel, iseed_map
  integer N, icnt, i, j, aa, bb
  real X(0:maxprm,0:maxprm)
  real xmin, xmax, xmean, xsd
  logical addition
  real :: hclump, meanmap
  REAL :: Xnorm(0:2**maxlevel,0:2**maxlevel)
  Real :: XXmean,cv,xxsd,Xnormave, Xnormmin,Xnormmax,XXtotal
  REAL :: mapout(1:2**maxlevel,1:2**maxlevel)
```

```
  xxsd = cv * meanmap
  N = 2**maxlevel
  if (N .gt. maxprm) stop
  addition = .true.
```

```
! call 2d multi-fractal generator
  call mpfm2d (X, addition, iseed_map,hclump,maxlevel)
```

```
! initialize statistics
  xmin = 1e30
  xmax = -1e30
  xmean = 0.0
```

```

xsd = 0.0
icnt = 0

! obtain statistics
do i = 0,N
  do j = 0,N
    if (X(i,j) .lt. xmin) xmin = X(i,j)
    if (X(i,j) .gt. xmax) xmax = X(i,j)
    xmean = xmean + X(i,j)
    xsd = xsd + X(i,j)*X(i,j)
    icnt = icnt+1
  enddo
enddo
xsd = sqrt((xsd - (xmean*xmean)/icnt) / (icnt-1))
xmean = xmean / icnt

200  format (/10x,'Hfract Summary'/10x,'Mean = ',g12.6 &
/10x,'St.dev. = ',g12.6 /10x,'Minimum = ',g12.6 &
/10x,'Maximum = ',g12.6)

!read in truncated array to Xnorm check on mean value
Xnorm = X(0:N,0:N)
Xnormave = SUM(Xnorm)/((N+1)*(N+1))
!subtract out mean
Xnorm = Xnorm - xmean
Xnormave = SUM(Xnorm)/((N+1)*(N+1))
Xnormmax = MAXVAL(Xnorm)
Xnormmin = MINVAL(Xnorm)
!divide by SD
Xnorm = Xnorm/xsd
Xnormave = SUM(Xnorm)/((N+1)*(N+1))
Xnormmax = MAXVAL(Xnorm)
Xnormmin = MINVAL(Xnorm)
!re-normalize for mean of map
Xnorm = Xnorm*xxsd + meanmap
XXmean = SUM(Xnorm)/((N+1)*(N+1))
Xnormmax = MAXVAL(Xnorm)
Xnormmin = MINVAL(Xnorm)
XXtotal = SUM(Xnorm)
!remove any negative numbers
DO aa = 0,N
DO bb = 0,N
  IF (Xnorm(aa,bb) .lt. 0.0) then
    Xnorm(aa,bb) = 0.0
  END IF

```

```
END DO
END DO
```

```
    XXmean = SUM(Xnorm)/((N+1)*(N+1))
    Xnormmax = MAXVAL(Xnorm)
    Xnormmin = MINVAL(Xnorm)
    XXtotal = SUM(Xnorm)
    mapout = Xnorm(1:2**maxlevel,1:2**maxlevel)

    END subroutine
```

Subroutine mpfm2d (X, addition, izeed,H,maxlevel2)

```
    USE map_variables

    ! USE parmlist
    ! USE fracblock
    ! USE mapblock
    INTEGER,PARAMETER :: maxprm = 2048
    INTEGER :: maxlevel2
    real :: H
    real X(0:maxprm,0:maxprm)
    integer N, DD, d, izeed
    logical addition
    integer :: maptype,wrap
    REAL :: sigma,delta
    Real :: southwest
    Real ::southeast
    Real ::northwest
    Real ::northeast

    southwest = 15.0
    southeast = 15.0
    northwest = 9.0
    northeast = 9.0
    sigma = 1 !indicates amount of variance(SD)
    maptype = maptype_input !1=gradient ! don't want gradient
    wrap = wrap_input !2=no wrapped boundaries want wrapped boundaries
    ! addition = 0 !turn random additions off
    !
    N = 2**maxlevel2
    delta = sigma
    X(0,0) = 0.0
    X(0,N) = 0.0
    X(N,0) = 0.0
    X(N,N) = 0.0
```

```

if (maptype .eq. 1) then !multifractal map with a gradient
  X(0,0) = southwest
  X(0,N) = southeast
  X(N,0) = northwest
  X(N,N) = northeast
endif
!
DD = N
d = N / 2
!
do is = 1, maxlevel2
! going from grid type I to type II
  delta = delta * 0.5**(0.5 * H)
  do ix = d, N-d, DD
    do iy = d, N-d, DD
      X(ix,iy) = f4(delta, X(ix+d,iy+d), X(ix+d,iy-d), &
        X(ix-d,iy+d), X(ix-d,iy-d),sigma)
    enddo
  enddo
! displace other points also if needed
  if (addition) then
    do ix = 0, N, DD
      do iy = 0, N, DD
        X(ix,iy) = X(ix,iy) + delta * Gauss(0.0,sigma,iseed)
      enddo
    enddo
  endif
! going from grid type II to type I
  delta = delta * 0.5**(0.5*H)
! interpolate and offset boundary grid points
  do ix = d, N-d, DD
    X(ix,0) = f3(delta, X(ix+d,0), X(ix-d,0), X(ix,d),sigma)
    X(ix,N) = f3(delta, X(ix+d,N), X(ix-d,N), X(ix,N-d),sigma)
    X(0,ix) = f3(delta, X(0,ix+d), X(0,ix-d), X(d,ix),sigma)
    X(N,ix) = f3(delta, X(N,ix+d), X(N,ix-d), X(N-d,ix),sigma)
    if (wrap .eq. 1) then ! if want wrapped boundaries
      X(ix,N) = X(ix,0)
      X(N,ix) = X(0,ix)
    endif
  enddo
! interpolate and offset inter grid points
  do ix = d, N-d, DD
    do iy = DD, N-d, DD
      X(ix,iy) = f4 (delta, X(ix,iy+d), X(ix,iy-d), &
        X(ix+d,iy), X(ix-d,iy),sigma)
    enddo
  enddo

```

```

        enddo
    enddo
do ix = DD, N-d, DD
    do iy = d, N-d, DD
        X(ix,iy) = f4(delta, X(ix,iy+d), X(ix,iy-d), &
            X(ix+d,iy), X(ix-d,iy),sigma)
    enddo
enddo
! displace other points also if needed
if (addition) then
    do ix = 0, N, DD
        do iy = 0, N, DD
            X(ix,iy) = X(ix,iy) + delta * Gauss(0.0,sigma,iseed)
        enddo
    enddo
do ix = d, N-d, DD
    do iy = d, N-d, DD
        X(ix,iy) = X(ix,iy) + delta * Gauss(0.0,sigma,iseed)
    enddo
enddo
endif
DD = DD / 2
d = d / 2
enddo
return
end
! statement function definition
REAL function f3(delta,x0,x1,x2,sigma)
    REAL delta,x0,x1,x2
    Real sigma
    INTEGER iseed
    iseed = -12345678
    f3 = (x0+x1+x2) / 3 + delta * Gauss(0.0,sigma, iseed)
    return
end

REAL function f4(delta,x0,x1,x2,x3,sigma)
    REAL delta,x0,x1,x2,x3
    Real sigma
    INTEGER iseed
    iseed = -12345678
    f4 = (x0+x1+x2+x3) / 4 + delta * Gauss(0.0,sigma,iseed)
    return
end

```

Sensitivity Analysis Routines

These routines conduct the sensitivity analysis utilized in chapter 2. Output from these routines is then input into a separate program to calculate the estimated sensitivity values (not shown). The routines listed allow for spatial and temporal sensitivity analysis to be conducted. Only a temporal sensitivity analysis at day 90 was conducted in chapter 2. These routines are adapted from methods developed by Gardner and Trabalka (1985) and is used with permission.

```
SUBROUTINE Prism_rd(irep)
```

```
USE sensitivity
USE global_variables
IMPLICIT NONE
```

```
INTEGER :: ii,izz,ndum,jj,j,irep,loop,loopcounter,iii,jjj
INTEGER :: niter,nseed
CHARACTER(LEN = 60) :: attl
CHARACTER(LEN = 15) :: pnam(10000)    !keeps an array of the names
CHARACTER(LEN= 3) :: timepoint(130)  !maximum size map or number of
                                     time-points
```

```
DATA timepoint /'001','002','003','004','005','006','007','008','009','010', &
                '011','012','013','014','015','016','017','018','019','020', &
                '021','022','023','024','025','026','027','028','029','030', &
                '031','032','033','034','035','036','037','038','039','040', &
                '041','042','043','044','045','046','047','048','049','050', &
                '051','052','053','054','055','056','057','058','059','060', &
                '061','062','063','064','065','066','067','068','069','070', &
                '071','072','073','074','075','076','077','078','079','080', &
                '081','082','083','084','085','086','087','088','089','090', &
                '091','092','093','094','095','096','097','098','099','100', &
                '101','102','103','104','105','106','107','108','109','110', &
                '111','112','113','114','115','116','117','118','119','120', &
                '121','122','123','124','125','126','127','128','129','130/'
```

```
! re-write header information
```

```
!! nval = 4    !number of output variables interested in !hard wired
```

```
IF (irep .eq. 1) THEN
```

```
! np = number of parameters, ndum = ***, niter = number of iterations
```

```
! nseed = random number seed !nval = number of output parameters
```

```
    read (41,103) attl
```

```
    write (42,103) attl
```

```
103  format (a60)
```

```
    read (41,*) np, ndum, niter, nseed
```

```

        write (42,*) np, nval, niter, nseed
!!    PRINT *, np, nval, niter, nseed

! read in parameter names
        read (41,105) (pnam(j),j=1,np)
105    format (5(2x,a10))

! map outputs into parameter vector
! flag to only output sensitivity at end of simulation

        IF (nval .eq. 4) THEN

                IF (senswitch .eq. 1) THEN
                        pnam(np+1) = ' Nave'
                        pnam(np+2) = ' Pave'
                        pnam(np+3) = ' Zave'
                        pnam(np+4) = ' Dave'
                END IF

                IF (senswitch .eq. 2) THEN
                        pnam(np+1) = ' Npatch'
                        pnam(np+2) = ' Ppatch'
                        pnam(np+3) = ' Zpatch'
                        pnam(np+4) = ' Dpatch'
                END IF

                IF (senswitch .eq. 3) THEN
                        pnam(np+1) = ' Nbiomass'
                        pnam(np+2) = ' Pbiomass'
                        pnam(np+3) = ' Zbiomass'
                        pnam(np+4) = ' Dbiomass'
                END IF

                IF (senswitch .eq. 4) THEN
                        pnam(np+1) = ' Narea'
                        pnam(np+2) = ' Parea'
                        pnam(np+3) = ' Zarea'
                        pnam(np+4) = ' Darea'
                END IF

                IF (senswitch .eq. 5) THEN
                        pnam(np+1) = ' Ncir'
                        pnam(np+2) = ' Pcir'
                        pnam(np+3) = ' Zcir'
                        pnam(np+4) = ' Dcir'
                END IF

```


Else

IF (senswitch .eq. 1) THEN

```
loopcounter = 0
loop = (nval/4)    ! nval = total number of point (10 output * 4 parameters)
DO iii = 1, 4      !hardwired for the number of output parameters
DO jjj = 1, loop   !number of output points
```

```
loopcounter = loopcounter + 1
```

```
IF (iii .eq. 1) THEN
    pnam(np+loopcounter) = 'Nave'//timepoint(jjj)
END IF
```

```
IF (iii .eq. 2) THEN
    pnam(np+loopcounter) = 'Pave'//timepoint(jjj)
END IF
```

```
IF (iii .eq. 3) THEN
    pnam(np+loopcounter) = 'Zave'//timepoint(jjj)
END IF
```

```
IF (iii .eq. 4) THEN
    pnam(np+loopcounter) = 'Dave'//timepoint(jjj)
END IF
```

```
END DO
END DO
```

END IF

IF (senswitch .eq. 2) THEN

```
loopcounter = 0
loop = (nval/4)    ! nval = total number of point (10 output * 4 parameters)
DO iii = 1, 4      !hardwired for the number of output parameters
DO jjj = 1, loop   !number of output points
```

```
loopcounter = loopcounter + 1
```

```
IF (iii .eq. 1) THEN
    pnam(np+loopcounter) = 'Npatch'//timepoint(jjj)
END IF
```

```
IF (iii .eq. 2) THEN
```

```

        pnam(np+loopcounter) = 'Ppatch'//timepoint(jjj)
        END IF

        IF (iii .eq. 3) THEN
            pnam(np+loopcounter) = 'Zpatch'//timepoint(jjj)
            END IF

            IF (iii .eq. 4) THEN
                pnam(np+loopcounter) = 'Dpatch'//timepoint(jjj)
                END IF
            END DO
        END DO
    END IF

    IF (sensswitch .eq. 3) THEN

        loopcounter = 0
        loop = (nval/4)    ! nval = total number of point (10 output * 4 parameters)
        DO iii = 1, 4      !hardwired for the number of output parameters
            DO jjj = 1, loop    !number of output points

                loopcounter = loopcounter + 1

                IF (iii .eq. 1) THEN
                    pnam(np+loopcounter) = 'Nbiomass'//timepoint(jjj)
                    END IF

                    IF (iii .eq. 2) THEN
                        pnam(np+loopcounter) = 'Pbiomass'//timepoint(jjj)
                        END IF

                        IF (iii .eq. 3) THEN
                            pnam(np+loopcounter) = 'Zbiomass'//timepoint(jjj)
                            END IF

                            IF (iii .eq. 4) THEN
                                pnam(np+loopcounter) = 'Dbiomass'//timepoint(jjj)
                                END IF
                            END DO
                        END DO
                    END IF

                IF (sensswitch .eq. 4) THEN

```

```

loopcounter = 0
loop = (nval/4)    ! nval = total number of point (10 output * 4 parameters)
DO iii = 1, 4      !hardwired for the number of output parameters
  DO jjj = 1, loop  !number of output points

    loopcounter = loopcounter + 1

    IF (iii .eq. 1) THEN
      pnam(np+loopcounter) = 'Narea'//timepoint(jjj)
    END IF

    IF (iii .eq. 2) THEN
      pnam(np+loopcounter) = 'Parea'//timepoint(jjj)
    END IF

    IF (iii .eq. 3) THEN
      pnam(np+loopcounter) = 'Zarea'//timepoint(jjj)
    END IF

    IF (iii .eq. 4) THEN
      pnam(np+loopcounter) = 'Darea'//timepoint(jjj)
    END IF

  END DO
END DO
END IF

```

```

IF (senswitch .eq. 5) THEN

```

```

  loopcounter = 0
  loop = (nval/4)    ! nval = total number of point (10 output * 4 parameters)
  DO iii = 1, 4      !hardwired for the number of output parameters
    DO jjj = 1, loop  !number of output points

      loopcounter = loopcounter + 1

      IF (iii .eq. 1) THEN
        pnam(np+loopcounter) = 'Ncir'//timepoint(jjj)
      END IF

      IF (iii .eq. 2) THEN
        pnam(np+loopcounter) = 'Pcir'//timepoint(jjj)
      END IF

      IF (iii .eq. 3) THEN
        pnam(np+loopcounter) = 'Zcir'//timepoint(jjj)
      END IF
    END DO
  END DO
END IF

```

```

                                END IF

                                IF (iii .eq. 4) THEN
                                    pnam(np+loopcounter) = 'Dcir'//timepoint(jjj)
                                END IF
                            END DO
                        END DO
                    END IF

                END IF

            END IF

            izz = INT((np+nval)/5)
            IF (izz .gt. 0) THEN !write parameter names
                do jj = 1,izz
                    write (42,105) (pnam(j),j=(jj-1)*5+1,jj*5)
                end do
            END IF
            write (42,105) (pnam(j),j=(izz*5)+1,np+nval)
        END IF

        ! read in parameters

        read (41,*) ii          !should be positioned in file at the iteration number
        IF (ii .ne. irep) THEN
            !! PRINT *, '    Error -- prism read out of order'
            !! PRINT *, '    Stopping execution'
            stop
        END IF
        read (41,*) (parm(j),j=1,np)
        !! write (6,*) (parm(j),j=1,np)
        ! map parameters to model

        Kn = parm(1)
        Pmax = parm(2)
        Rp = parm(3)
        Mp = parm(4)
        Sp = parm(5)
        Ik = parm(6)
        Kz = parm(7)
        Io = parm(8)
        Kc = parm(9)
        Kw = parm(10)
        Zmax = parm(11)
        Az = parm(12)
        Rz = parm(13)
        Mz = parm(14)

```

```

Pz = parm(15)
Ppref = parm(16)
k3 = parm(17)
Rd = parm(18)
Sd = parm(19)
Dk = parm(20)
No = parm(21)
N = parm(22)
P = parm(23)
Z = parm(24)
D = parm(25)
diffcoeff(1) = parm(26)
return
end

```

```

SUBROUTINE Prism_out(irep)

```

```

    USE global_variables
    USE sensitivity
    implicit none

```

```

    INTEGER :: iz,j,jj,irep,loop,loopcounter,iii,jjj

```

```

    !output the results of prism2 to file to be input to prism 3
    ! irep = number of the simulation
    ! np = number of input parameters
    ! nval = number of output parameters

```

```

!Spatial Sen Analysis !only output at end of simulation and along a given map axis
IF (senflag .eq. 1) THEN

```

```

    loopcounter = 0
    loop = (nval/4)    ! nval2 = total number of pixels along a given dimension *
                      ! number of output parameters
                      ! example (10 cells * 4 parameters) = 40
    DO iii = 1, 4      !hardwired for the number of output parameters
        DO jjj = 1, loop    !number of output points

```

```

                loopcounter = loopcounter + 1    !does sensitivity analysis only on the
                                                cell means

```

```
IF (iii .eq. 1) THEN
  parm(np+loopcounter) = Nmap(stat_counter,jjj,ylocationcoor)
END IF
```

```
IF (iii .eq. 2) THEN
  parm(np+loopcounter) = Pmap(stat_counter,jjj,ylocationcoor)
END IF
```

```
IF (iii .eq. 3) THEN
  parm(np+loopcounter) = Zmap(stat_counter,jjj,ylocationcoor)
END IF
```

```
IF (iii .eq. 4) THEN
  parm(np+loopcounter) = Dmap(stat_counter,jjj,ylocationcoor)
END IF
```

```
END DO
END DO
```

```
END IF
```

```
!Temporal Sensitivity Analysis
```

```
IF (senflag .eq. 2) THEN
```

```
IF (nval .eq. 4) THEN    !Output only at end of simulation
```

```
IF (senswitch .eq. 1) THEN
  parm(np+1) = Nave(stat_counter)
  parm(np+2) = Pave(stat_counter)
  parm(np+3) = Zave(stat_counter)
  parm(np+4) = Dave(stat_counter)
END IF
```

```
Else
```

```
IF (senswitch .eq. 1) THEN
  loopcounter = 0
  loop = (nval/4)    ! nval = total number of point (10 output * 4
                    ! parameters)
  DO iii = 1, 4    !hardwired for the number of output parameters
    DO jjj = 1, loop    !number of output points

      loopcounter = loopcounter + 1
```

```

        IF (iii .eq. 1) THEN
            parm(np+loopcounter) = Nave(jjj)
        END IF

        IF (iii .eq. 2) THEN
            parm(np+loopcounter) = Pave(jjj)
        END IF

        IF (iii .eq. 3) THEN
            parm(np+loopcounter) = Zave(jjj)
        END IF

        IF (iii .eq. 4) THEN
            parm(np+loopcounter) = Dave(jjj)
        END IF

            END DO
        END DO
    END IF
END IF

write (42,'(1x,i5)') irep      ! prints out a line for each replication
iz = INT((np + nval)/5) !how many times to perform output

! output parameter variables in 5 columns

if (iz .gt. 0) then
    do jj = 1,iz
        write (42,'(5(2x,g12.6))') (parm(j),j=(jj-1)*5+1,jj*5)
    end do
end if
write (42,'(5(2x,g12.6))') (parm(j),j=(iz*5)+1,np+nval)
return
end

```

File Saving and Statistical Analysis Routines

These routines save simulation output for data analysis with 3rd party statistical analysis programs (e.g., SAS, EXCEL, AXUM). The routines are also set up to conduce real-time analysis of selected state variables for common statistical measures (e.g., mean, variance, CV, etc).

```
SUBROUTINE stats
```

```
    USE global_variables  
    IMPLICIT NONE
```

```
    REAL :: xmin,xmax  
    INTEGER :: iii,i,j
```

```
    stat_counter = stat_counter + 1
```

```
!This section contains the statistics for the nut input chapter  
IF (Nut_chapter_switch .eq. 1) THEN
```

```
! Saves entire nutrient input map once at beginning of the simulation  
! Reproduces map exactly starting from the top and going down  
! so bottom of map (last lines in file will be what is input first)
```

```
If (map_input .ne. 3) then
```

```
IF (stat_counter .eq. 1) THEN  
    Write (44) numcol  
    Write (44) numrow  
    Write (44) counter  
    Write (44) Nutinput  
END IF
```

```
End IF
```

```
write (70,*) 'counter=', counter  
Write (40) numcol  
Write (40) numrow  
Write (40) counter  
Write (40) N
```

```
Write (41) numcol  
Write (41) numrow  
Write (41) counter
```


Write (41) P

Write (42) numcol
Write (42) numrow
Write (42) counter
Write (42) Z

Write (43) numcol
Write (43) numrow
Write (43) counter
Write (43) D

Write (46) numcol
Write (46) numrow
Write (46) counter
Write (46) Pnew

Write (47) numcol
Write (47) numrow
Write (47) counter
Write (47) Znew

Write (48) numcol
Write (48) numrow
Write (48) counter
Write (48) BiomassOut

Write (49) numcol
Write (49) numrow
Write (49) counter
Write (49) PredOut

Write (50) numcol
Write (50) numrow
Write (50) counter
Write (50) PhytoGrowth

Write (51) numcol
Write (51) numrow
Write (51) counter
Write (51) ZooGrowth

!This section contains stats calculated for regular simulations

Else

! initialize statistical arrays at the beginning of each time-step

Ntotal=0.0

Ptotal=0.0

Ztotal=0.0

Dtotal=0.0

SumN=0.0

SumP=0.0

SumZ=0.0

SumD=0.0

Nmax(stat_counter)=MAXVAL(N)

Nmin(stat_counter)=MINVAL(N)

Pmax(stat_counter)=MAXVAL(P)

Pmin(stat_counter)=MINVAL(P)

Zmax(stat_counter)=MAXVAL(Z)

Zmin(stat_counter)=MINVAL(Z)

Dmax(stat_counter)=MAXVAL(D)

Dmin(stat_counter)=MINVAL(D)

!Calculates the max and min over the whole simulation

!Used for creating line and contour graphs calculate at end of simulation

Nmapmax = MAXVAL(Nmax)

Nmapmin = MINVAL(Nmin)

Pmapmax = MAXVAL(Pmax)

Pmapmin = MINVAL(Pmin)

Zmapmax = MAXVAL(Zmax)

Zmapmin = MINVAL(Zmin)

Dmapmax = MAXVAL(Dmax)

Dmapmin = MINVAL(Dmin)

! Calculate max,min, and average

Ntotal=SUM(N)

Nave(stat_counter)=(Ntotal/(numrow*numcol))

Ptotal=SUM(P)

Pave(stat_counter)=(Ptotal/(numrow*numcol))

Ztotal=SUM(Z)

Zave(stat_counter)=(Ztotal/(numrow*numcol))

```

Dtotal=SUM(D)
Dave(stat_counter)=(Dtotal/(numrow*numcol))

! Only do this if want variance, cv, and sd calculated
IF (stat_switch .eq. 1) THEN

!Sum of all the Y's squared ,am using the computational formula
SumN = SUM(N * N)
SumP = SUM(P * P)
SumZ = SUM(Z * Z)
SumD = SUM(D * D)
Mapsize = numrow * numcol

! Calculate the Variance
Nvar(stat_counter) = SumN - ((Ntotal*Ntotal)/Mapsize)
Pvar(stat_counter) = SumP - ((Ptotal*Ptotal)/Mapsize)
Zvar(stat_counter) = SumZ - ((Ztotal*Ztotal)/Mapsize)
Dvar(stat_counter) = SumD - ((Dtotal*Dtotal)/Mapsize)

! Prevents round off error
IF (Nvar(stat_counter) .lt. .0001) THEN
    Nvar(stat_counter) = 0.0000000000000000
END IF

IF (Pvar(stat_counter) .lt. .0001) THEN
    Pvar(stat_counter) = 0.0000000000000000
END IF

IF (Zvar(stat_counter) .lt. .0001) THEN
    Zvar(stat_counter) = 0.0000000000000000
END IF

IF (Dvar(stat_counter) .lt. .0001) THEN
    Dvar(stat_counter) = 0.0000000000000000
END IF

! Calculate the SD
Nsd(stat_counter) = SQRT(ABS(Nvar(stat_counter)))
Psd(stat_counter) = SQRT(ABS(Pvar(stat_counter)))
Zsd(stat_counter) = SQRT(ABS(Zvar(stat_counter)))
Dsd(stat_counter) = SQRT(ABS(Dvar(stat_counter)))

! Calculate the CV
Ncv(stat_counter) = (Nsd(stat_counter)*100)/Nave(stat_counter)
Pcv(stat_counter) = (Psd(stat_counter)*100)/Pave(stat_counter)
Zcv(stat_counter) = (Zsd(stat_counter)*100)/Zave(stat_counter)

```

```

Dcv(stat_counter) = (Dsd(stat_counter)*100)/Dave(stat_counter)

END IF

! Write stats to output file
IF (file_switch .eq. 1) THEN
  WRITE (30,FMT=*) Nmax(stat_counter),Nmin(stat_counter),Nave(stat_counter),
&
  Nvar(stat_counter),Nsd(stat_counter),Ncv(stat_counter)
  WRITE (31,FMT=*) Pmax(stat_counter),Pmin(stat_counter),Pave(stat_counter),
&
  Pvar(stat_counter),Psd(stat_counter),Pcv(stat_counter)
  WRITE (32,FMT=*) Zmax(stat_counter),Zmin(stat_counter),Zave(stat_counter),
&
  Zvar(stat_counter),Zsd(stat_counter),Zcv(stat_counter)
  WRITE (33,FMT=*) Dmax(stat_counter),Dmin(stat_counter),Dave(stat_counter),
&
  Dvar(stat_counter),Dsd(stat_counter),Dcv(stat_counter)
  WRITE (50,*) Nutrient, Light, Drate, Prate, Dgraz, Pgraz, Kd, ratelimit
  END IF

END IF

END SUBROUTINE

```

General Model Subroutines

!This routine determines when to call the various subroutines within the SLS based on a predetermined frequency.

SUBROUTINE switches(irep)

USE global_variables
IMPLICIT NONE

INTEGER :: irep

! Calculate map statistics and save output

If (simtime .ge. map_save) then
DT_trigger = ABS(MOD(counter,FileSaveFreq))
IF (DT_trigger .eq. 0) THEN
CALL stats
END IF
END If

! Calculate external nutrient addition

! need to check if a spot or an area

IF (counter .gt. StartNutSpike .and. counter .lt. EndNutSpike) THEN
DT_trigger = ABS(MOD(counter,NutSpikeFreq))
IF (DT_trigger .eq. 0) THEN
call nutspot

!this is done so that the nutrient pulse isn't
confounded

END IF
END IF

!with the time-step of the model

!if were in biology would get fractionated by
the DT

!could leave in but would have to divide the value
wanted by DT

!to get the real value

! Calculate fish predation

! need to check for a spot or an area

IF (counter .gt. StartFishPred .and. counter .lt. EndFishPred) THEN
DT_trigger = ABS(MOD(counter,FishPredFreq))
IF (DT_trigger .eq. 0) THEN
call fishspot

END IF
END IF

```

! Calculate nutrient spot movement
  IF (counter .gt. StartNutMove .and. counter .lt. EndNutMove) THEN
    DT_trigger = ABS(MOD(counter,NutMoveFreq))
    IF (DT_trigger .eq. 0) THEN
      CALL nutmove
    END IF
  END IF

! Calculate fish predation movement
  IF (counter .gt. StartFishMove .and. counter .lt. EndFishMove) THEN
    DT_trigger = ABS(MOD(counter,FishMoveFreq))
    IF (DT_trigger .eq. 0) THEN
      CALL fishmove
    END IF
  END IF

END SUBROUTINE

```

!This routine allocates the dimensions of all the arrays based on the mapsize and the simulation length

```

SUBROUTINE allocation
  USE global_variables
  IMPLICIT NONE

```

! Dimensions arrays according to the input provided by the user

```

  ALLOCATE(N(numrow,numcol))
  ALLOCATE(P(numrow,numcol))
  ALLOCATE(Z(numrow,numcol))
  ALLOCATE(D(numrow,numcol))
  ALLOCATE(Nutinput(numrow,numcol))
  ALLOCATE(Pnew(numrow,numcol))
  ALLOCATE(Znew(numrow,numcol))
  ALLOCATE(BiomassOut(numrow,numcol))
  ALLOCATE(PredOut(numrow,numcol))
  ALLOCATE(PhytoGrowth(numrow,numcol))
  ALLOCATE(ZooGrowth(numrow,numcol))
  ALLOCATE(Ntransect(numrow),Ptransect(numrow),Ztransect(numrow),
    Dtransect(numrow))
  ALLOCATE(Nautocorr(numrow/2),Pautocorr(numrow/2),Zautocorr(numrow/2),
    Dautocorr(numrow/2))

```

```

ALLOCATE(Nmap(visual_switch,numrow,numcol))
ALLOCATE(Pmap(visual_switch,numrow,numcol))
ALLOCATE(Zmap(visual_switch,numrow,numcol))
ALLOCATE(Dmap(visual_switch,numrow,numcol))
ALLOCATE(Nmax(output_dimension),Pmax(output_dimension),
          Zmax(output_dimension),Dmax(output_dimension))
ALLOCATE(Nmin(output_dimension),Pmin(output_dimension),
          Zmin(output_dimension),Dmin(output_dimension))
ALLOCATE(Nave(output_dimension),Pave(output_dimension),
          Zave(output_dimension),Dave(output_dimension))
ALLOCATE (Nvar(output_dimension), Pvar(output_dimension), &
          Zvar(output_dimension) ,Dvar(output_dimension))
ALLOCATE (Nsd(output_dimension), Psd(output_dimension),&
          Zsd(output_dimension) ,Dsd(output_dimension))
ALLOCATE (Ncv(output_dimension), Pcv(output_dimension), &
          Zcv(output_dimension) ,Dcv(output_dimension))
ALLOCATE (Narraymin(output_dimension),Narraymax(output_dimension),&
          Parraymin(output_dimension),Parraymax(output_dimension))
ALLOCATE (Zarraymin(output_dimension),Zarraymax(output_dimension),&
          Darraymin(output_dimension),Darraymax(output_dimension))

```

END SUBROUTINE

! This routine opens all the files which will be read or saved to during a simulation
SUBROUTINE fileopen

```

USE global_variables
IMPLICIT NONE

```

```

OPEN (30, FILE='Nstats.dat', STATUS='unknown')
OPEN (31, FILE='Pstats.dat', STATUS='unknown')
OPEN (32, FILE='Zstats.dat', STATUS='unknown')
OPEN (33, FILE='Dstats.dat', STATUS='unknown')
OPEN (34, FILE='NPZDtrans.dat', STATUS='unknown')

OPEN (40, FILE='Nmap.dat', STATUS='unknown',form='unformatted')
OPEN (41, FILE='Pmap.dat', STATUS='unknown',form='unformatted')
OPEN (42, FILE='Zmap.dat', STATUS='unknown',form='unformatted')
OPEN (43, FILE='Dmap.dat', STATUS='unknown',form='unformatted')
OPEN (44, FILE='Ninputmap.dat', STATUS='unknown',form='unformatted')

OPEN (46, FILE='Pnewmap.dat', STATUS='unknown',form='unformatted')

```

```
OPEN (47, FILE='Znewmap.dat', STATUS='unknown',form='unformatted')
OPEN (48, FILE='BiomassOut.dat', STATUS='unknown',form='unformatted')
OPEN (49, FILE='PredOut.dat', STATUS='unknown',form='unformatted')
  OPEN (50, FILE='PhytoGrowth.dat', STATUS='unknown',form='unformatted')
  OPEN (51, FILE='ZooGrowth.dat', STATUS='unknown',form='unformatted')
```

```
OPEN (52, FILE='rates.dat', STATUS='unknown')
OPEN (70, FILE='junk.dat', STATUS='unknown')
```

```
  Write (70,*) 'contains time data'
```

```
END SUBROUTINE
```

!This routine closes all files which were opened during a simulation and deallocates arrays

```
SUBROUTINE cleanup
```

```
  USE global_variables
  IMPLICIT NONE
```

```
Endfile(40)
Endfile(41)
Endfile(42)
Endfile(43)
Endfile(44)
Endfile(46)
Endfile(47)
Endfile(48)
Endfile(49)
Endfile(50)
Endfile(51)
```

```
! Close all opened files
```

```
  CLOSE(30)
  CLOSE(31)
  CLOSE(32)
  CLOSE(33)
  CLOSE(34)
  CLOSE(50)
  CLOSE(51)
  CLOSE(52)
  CLOSE(70)
```

```
! Removes allocated space from memory
```



```

DEALLOCATE(N,P,Z,D)
DEALLOCATE (Pnew,Znew,BiomassOut,PredOut,PhytoGrowth,ZooGrowth)
DEALLOCATE (Nmap,Pmap,Zmap,Dmap)
DEALLOCATE(Nutinput)
DEALLOCATE(Nmax,Pmax,Zmax,Dmax)
DEALLOCATE(Nave,Pave,Zave,Dave)
DEALLOCATE(Nmin,Pmin,Zmin,Dmin)
DEALLOCATE(Nvar,Pvar,Zvar,Dvar)
DEALLOCATE(Nsd,Psd,Zsd,Dsd)
DEALLOCATE(Ncv,Pcv,Zcv,Dcv)
DEALLOCATE (xcenter,ycenter,rotation,&
             circlesize,starttime,freqswitch)
DEALLOCATE (cascade_switch,cascade_trigger,radius_stop_location)

```

```
END SUBROUTINE
```

This subroutine initializes the simulation domain and other spatial variables. The initialization can also vary by state variable in the NPZD model.

```
SUBROUTINE mapinit
```

```

    USE global_variables
    IMPLICIT NONE

```

```
    INTEGER :: xx,yy
```

```
! Flag to turn on and off multifractal initialization 1=turn on
```

```
IF (fractalinit .eq. 1) THEN
```

```
    IF (stateinit .eq. 1) THEN
```

```
    Call hfraction(maxlevel_input,hclump_input,N(1,1),cv_input,N,iseed_input)
```

```
    END IF
```

```
    IF (stateinit .eq. 2) THEN
```

```
    Call hfraction(maxlevel_input,hclump_input,P(1,1),cv_input,P,iseed_input)
```

```
    END IF
```

```
    IF (stateinit .eq. 3) THEN
```

```
    Call hfraction(maxlevel_input,hclump_input,Z(1,1),cv_input,Z,iseed_input)
```

```
    END IF
```

```
    IF (stateinit .eq. 4) THEN
```

```
    Call hfraction(maxlevel_input,hclump_input,D(1,1),cv_input,D,iseed_input)
```

```
    END IF
```

```
END IF
```

```
IF (fractalinit .eq. 3) THEN !Cosine gradient
```

```

DO xx = 1,numrow
DO yy = 1,numcol
  IF (stateinit .eq. 1) THEN
    N(xx,yy) = cos_map_max * ((1-COS((2*3.14*yy)/numcol))/2)
  END IF
  IF (stateinit .eq. 2) THEN
    P(xx,yy) = cos_map_max * ((1-COS((2*3.14*yy)/numcol))/2)
  END IF
  IF (stateinit .eq. 3) THEN
    Z(xx,yy) = cos_map_max * ((1-COS((2*3.14*yy)/numcol))/2)
  END IF
  IF (stateinit .eq. 4) THEN
    D(xx,yy) = cos_map_max * ((1-COS((2*3.14*yy)/numcol))/2)
  END IF
END DO
END DO
END IF

```

!Done to create map for nutrient input chapter
!Takes initial value for nutrients and creates a separate nutrient input map
! the same size as the simulation domain

```

  IF (map_input .eq. 1) THEN
    Call
    hfract(maxlevel_input,hclump_input,Nut_input_amount,cv_input,Nutinput,iseed_input)
  END IF

  IF (map_input .eq. 2) THEN
    Call
    hfract(maxlevel_input,hclump_input,Nut_input_amount,cv_input,Nutinput,iseed_input)
    Nutinput = Nut_input_amount !sets the generated map to the map
    mean
  END IF

  IF (map_input .eq. 3) THEN
    !default so that no map is created
  END IF

END SUBROUTINE

```

General Model Functions

!The following functions are used in various subroutines to do special calculations

FUNCTION circle(angle)

```
REAL :: angle
circle = angle
IF (angle .gt. 360.0) THEN
    circle = (angle - 360.0)
END IF
IF (angle .lt. 0.0) THEN
    circle = (angle + 360.0)
END IF
```

END

! Function will round a real number then convert it to an integer
! location is the real representation of the number
! round returns the integer representation of the number
! If the number is negative it will round up (for example)
! -3.876 will round to -4
! -3.212 will round to -3

Function round(location)

```
REAL :: location
INTEGER :: round

IF (location .lt. 0.0) THEN
    round = INT(location - 0.5)
ELSE
    round = INT(location + 0.5)
END IF
```

END

! Function will take a distance and an angle and convert it into an (x distance) use (trig) functions

Function convert_x(length, angle)

```
REAL :: length, angle, conver
! PRINT *,length, angle
```

```

! First quadrant
IF (angle .gt. 0.0 .and. angle .lt. 90.0) THEN
    conver = angle * 0.0175          ! conversion to radians
    convert_x = length * COS(conver) ! x location
END IF

! Second quadrant
IF (angle .gt. 90.0 .and. angle .lt. 180.0) THEN
    conver = angle - 90.0            ! quadrant modification
    conver = conver * 0.0175        ! conversion to radians
    cosine = -SIN(conver)           ! quadrant modification
    convert_x = length * cosine      ! x location
END IF

! Third quadrant
IF (angle .gt. 180.0 .and. angle .lt. 270.0) THEN
    conver = angle - 180.0          ! quadrant modification
    conver = conver * 0.0175        ! conversion to radians
    cosine = -COS(conver)           ! quadrant modification
    convert_x = length * cosine      ! x location
END IF

! Fourth quadrant
IF (angle .gt. 270.0 .and. angle .lt. 360.0) THEN
    conver = angle - 270.0          ! quadrant modification
    conver = conver * 0.0175        ! conversion to radians
    cosine = SIN(conver)            ! quadrant modification
    convert_x = length * cosine      ! x location
END IF

END

```

END

! Function will take a distance and an angle and convert it into an (y distance) use (trig) functions

Function convert_y(length, angle)

```

REAL :: length, angle, conver

! PRINT *, length, angle

! First quadrant
IF (angle .gt. 0.0 .and. angle .lt. 90.0) THEN
    conver = angle * 0.0175          ! conversion to radians
    convert_y = length * SIN(conver) ! y location

```

```

END IF

! Second quadrant
IF (angle .gt. 90.0 .and. angle .lt. 180.0) THEN
    conver = angle - 90.0           ! quadrant modification
    conver = conver * 0.0175       ! conversion to radians
    sine = COS(conver)             ! quadrant modification
    convert_y = length * sine      ! y location
END IF

! Third quadrant
IF (angle .gt. 180.0 .and. angle .lt. 270.0) THEN
    conver = angle - 180.0        ! quadrant modification
    conver = conver * 0.0175     ! conversion to radians
    sine = -SIN(conver)          ! quadrant modification
    convert_y = length * sine    ! y location
END IF

! Fourth quadrant
IF (angle .gt. 270.0 .and. angle .lt. 360.0) THEN
    conver = angle - 270.0       ! quadrant modification
    conver = conver * 0.0175     ! conversion to radians
    sine = -COS(conver)         ! quadrant modification
    convert_y = length * sine    ! y location
END IF

```

END

! creates a normal number by summing 12 uniform random numbers
! creates a number between (-6 and 6)
! 95% of the numbers fall between 0 and 2 in a normal distribution

```

function Gauss(xm, xs, izeed)
real xm, xs, sume
sume = 0.0
do i = 1,12
    sume = sume + ran1(izeed)
enddo
Gauss = (sume - 6.0) * xs + xm
return
end

```

real function ran1 (idum)
! returns a uniform random deviate between 0.0 and 1.0.

```

! Set IDUM to any negative value to initialize or reinitialize
! the sequence.
! (C) Copr. 1986-92 Numerical Recipes Software
!
integer idum, ia, im, iq, ir, ntab, ndiv
real am, eps, rnmx
parameter (ia=16807, im=2147483647, am=1./im, iq=127773, ir=2836, &
& ntab=32, ndiv=1+(im-1)/ntab, eps=1.2e-7, rnmx=1.-eps)
integer j, k, iv(ntab), iy
save iv, iy
data iv /ntab*0/, iy /0/
!
if (idum .le. 0 .or. iy .eq. 0) then
  idum = max(-idum,1)
  do j = ntab+8, 1, -1
    k = idum / iq
    idum = ia * (idum - k*iq) - ir*k
    if (idum .lt. 0) idum = idum + im
    if (j .le. ntab) iv(j) = idum
  enddo
!
  iy = iv(1)
endif
!
k = idum / iq
idum = ia * (idum - k*iq) - ir*k
if (idum .lt. 0) idum = idum+im
j = 1 + iy / ndiv
iy = iv(j)
iv(j) = idum
ranl = min(am*iy, rnmx)
!
return
end

```

Misc Model Subroutines (Prototype)

The following subroutines were not used in the chapters presented in the final dissertation but were used in preliminary experiments investigating the effects of spatial and temporal heterogeneity in nutrient input and fish predation on spatial patterns and biological dynamics. Some of these routines are still in development but are included here as addition code which may be useful for studies involving these key ecosystem forcing functions.

```
SUBROUTINE nutmove
```

```
    USE global_variables  
    IMPLICIT NONE
```

```
! Local variables
```

```
    REAL :: ran_dir, ran_angle, mean_angle
```

```
!Functions
```

```
    INTEGER :: round
```

```
    REAL :: ran1, Gauss, circle
```

```
    REAL :: convert_x, convert_y
```

```
    counter_nut = counter_nut + 1
```

```
    IF (counter_nut .eq. 1) THEN
```

```
! Pick a random location for the predation starting location
```

```
    !     xlocation_nut = ran1(iseed_input) * numrow
```

```
    !     ylocation_nut = ran1(iseed_input) * numcol
```

```
! Set up for center of map-for sensitivity chapter
```

```
    xlocation_nut = numrow/2
```

```
    ylocation_nut = numcol/2
```

```
! Pick a random direction (outside of main loop)
```

```
! Random number between (0-1)
```

```
! Use the ran1 function
```

```
! (iseed)=random seed number used to initialize the routine
```

```
! ran_dir gives a percent of 360 based on the random number generated
```

```
    ran_dir = ran1(iseed_input) * 360
```

```
    Else
```

```
! Main Loop
```

```
! Pick a random angle deviation (this will determine how correlated the values are to one another)
```

```

! Small (SD) will give a small turning radius
! Pick a random distance
! Do not allow negative values so take the absolute value of the returned result

! The (gauss) function returns a value between (-6 and 6) which are normally distributed
! most of the values fall between (-2 and 2) roughly 95%
! so a (SD) of (180) will give the occasional 360 degree turn
! pick an (SD) which is the same as the turning radius that you want
! well at least where you want most of the values to fall

! Need to initialize predation pressure so that it doesn't keep that pressure
! at that spot over the whole simulation period

    !! nut_predation = 0.0

! Set mean_angle to (0.0) so that angle has no bias but could set to other values to give a
bias
    mean_angle = 0.0

! Calculate a new random angle based on standard deviation
    ran_angle = (Gauss (0.0,1.0,iseed_input) * angle_nut) + mean_angle

! Add the new angle to the old angle from outside of the loop
    ran_dir = ran_dir + ran_angle

! If past (360) start a new circle
    ran_dir = circle(ran_dir)

! Based on angle (ran_dir) and (distance) determine the new map location (absolute units
!from the original position) --used trigonometry functions and convert to Cartesian
!coordinates
! There is an error in the (Y) direction in Cartesian coordinates
    x_location_nut = convert_x(distance_nut,ran_dir)
    y_location_nut = convert_y(distance_nut,ran_dir)

! Round and make an integer value
! Also convert to grid coordinates for the (units only) as the (X) units are already correct
! The grid command counts down in the (Y) direction as opposed to up in Cartesian
coordinates

! Need to place a negative in front of the value for (Y)
    x_new_nut = round(x_location_nut)
    y_new_nut = -(round(y_location_nut))

! Move to the new position
    xlocation_nut = xlocation_nut + x_new_nut

```



```

        ylocation_nut = ylocation_nut + y_new_nut

! Need to check for the map edge (for wrapped boundaries) need to wrap the location
    IF (xlocation_nut .gt. numcol) THEN
        xlocation_nut = xlocation_nut - numcol
    END IF

    IF (xlocation_nut .lt. 1) THEN
        xlocation_nut = numcol + xlocation_nut
    END IF

    IF (ylocation_nut .gt. numrow) THEN
        ylocation_nut = ylocation_nut - numrow
    END IF

    IF (ylocation_nut .lt. 1) THEN
        ylocation_nut = numrow + ylocation_nut
    END IF

    END If

END SUBROUTINE

```

```

SUBROUTINE fishmove

    USE global_variables
    IMPLICIT NONE

! Local variables
    REAL :: ran_dir, ran_angle, mean_angle

!Functions
    INTEGER :: round
    REAL :: ran1, Gauss, circle
    REAL :: convert_x, convert_y

        counter_fish = counter_fish + 1
    IF (counter_fish .eq. 1) THEN

! Pick a random location for the predation starting location
        !   xlocation_fish = ran1(iseed_input) * numrow
        !   ylocation_fish = ran1(iseed_input) * numcol
    
```

```
xlocation_fish = numrow/2
ylocation_fish = numcol/2
```

```
! Pick a random direction (outside of main loop)
! Random number between (0-1)
! Use the ran1 function
! (iseed)=random seed number used to initialize the routine
! ran_dir gives a percent of 360 based on the random number generated
  ran_dir = ran1(iseed_input) * 360
```

```
  Else
```

```
! Main Loop
! Pick a random angle deviation (this will determine how correlated the values are to one
another)
! Small (SD) will give a small turning radius
! Pick a random distance
! Do not allow negative values so take the absolute value of the returned result

! The (gauss) function returns a value between (-6 and 6) which are normally distributed
! most of the values fall between (-2 and 2) roughly 95%
! so a (SD) of (180) will give the occasional 360 degree turn
! pick an (SD) which is the same as the turning radius that you want
! well at least where you want most of the values to fall
```

```
! Need to initialize predation pressure so that it doesn't keep that pressure
! at that spot over the whole simulation period
```

```
  !! fish_predation = 0.0
```

```
! Set mean_angle to (0.0) so that angle has no bias but could set to other values to give a
bias
```

```
  mean_angle = 0.0
```

```
! Calculate a new random angle based on standard deviation
  ran_angle = (Gauss (0.0,1.0,iseed_input) * angle_fish) + mean_angle
```

```
! Add the new angle to the old angle from outside of the loop
```

```
  ran_dir = ran_dir + ran_angle
```

```
! If past (360) start a new circle
```

```
  ran_dir = circle(ran_dir)
```

```

! Based on angle (ran_dir) and (distance) determine the new map location (absolute units
!from the original position) --used trigonometry functions and convert to Cartesian
!coordinates
! There is an error in the (Y) direction in Cartesian coordinates
  x_location_fish = convert_x(distance_fish,ran_dir)
  y_location_fish = convert_y(distance_fish,ran_dir)

! Round and make an integer value
! Also convert to grid coordinates for the (units only) as the (X) units are already correct
! The grid command counts down in the (Y) direction as opposed to up in Cartesian
coordinates
! Need to place a negative in front of the value for (Y)
  x_new_fish = round(x_location_fish)
  y_new_fish = -(round(y_location_fish))

! Move to the new position
  xlocation_fish = xlocation_fish + x_new_fish
  ylocation_fish = ylocation_fish + y_new_fish

! Need to check for the map edge (for wrapped boundaries) need to wrap the location
  IF (xlocation_fish .gt. numcol) THEN
    xlocation_fish = xlocation_fish - numcol
  END IF

  IF (xlocation_fish .lt. 1) THEN
    xlocation_fish = numcol + xlocation_fish
  END IF

  IF (ylocation_fish .gt. numrow) THEN
    ylocation_fish = ylocation_fish - numrow
  END IF

  IF (ylocation_fish .lt. 1) THEN
    ylocation_fish = numrow + ylocation_fish
  END IF

  END If

END SUBROUTINE

```

```

SUBROUTINE Nutspot

```

```

  USE global_variables
  IMPLICIT NONE

```

```

INTEGER :: iii, jjj, map_area, spot_radius_round
INTEGER :: ylocation_new, xlocation_new, new_loc_y, new_loc_x
REAL :: spot_area, spot_radius, spot_diameter, spot_distance

```

! Check for a spot, radius, or uniform effect

! If equal to 1 = spot (values = 2-9 = are circles of different radius)

```

IF (nutradius .eq. 1 .or. nutradius .eq. 10) then

```

```

    IF (nutradius .eq. 1) THEN      !nutrient spot effect
        N(ylocation_nut,xlocation_nut) = N(ylocation_nut,xlocation_nut) +
        NutSlugAmount
    END IF

```

```

    IF (nutradius .eq. 10) THEN    !uniform nutrient addition
        N = N + NutSlugAmount
    END IF

```

Else

! Decide on size of spot

! Minimum mapsize = 16x16 below this will get errors because will be less than one cell

! Spotsize will exceed the resolution of the grid

! Draw a circle around the selected point on the map

```

SELECT CASE (nutradius)
    CASE (2) ! (1/2 the area of the map)
        map_area = 2
    CASE (3) ! (1/4 the area of the map)
        map_area = 4
    CASE (4) ! (1/8 the area of the map)
        map_area = 8
    CASE (5) ! (1/16 the area of the map)
        map_area = 16
    CASE (6) ! (1/32 the area of the map)
        map_area = 32
    CASE (7) ! (1/64 the area of the map)
        map_area = 64
    CASE (8) ! (1/128 the area of the map)
        map_area = 128
    CASE (9) ! (1/256 the area of the map)
        map_area = 256
    case (11)
        map_area = 512

```

END SELECT

```
! Now calculate the area of the circle based on the mapsize
  spot_area = ((numrow * numcol)/(map_area))
! Calculate the radius of the circle
  spot_radius = SQRT(spot_area/3.14159)
! Round radius up to the nearest whole number
  spot_radius_round = INT(spot_radius + 0.5)
! Calculate diameter of the circle
  spot_diameter = spot_radius_round * 2

! Loop over square which encompasses circle
  DO iii = 1, (spot_diameter + 1)    !down columns (Y)
    DO jjj = 1, (spot_diameter + 1) !across rows (X)

! Initialize location from value outside of subroutine (old location on map)
      ylocation_new = ylocation_nut
      xlocation_new = xlocation_nut

! Find location within square
      new_loc_y = (iii-(spot_radius_round + 1))
      new_loc_x = (jjj-(spot_radius_round + 1))
      spot_distance = SQRT(Real((new_loc_y * new_loc_y) + (new_loc_x *
new_loc_x)))

! Check to see if location falls within circle
! If it does then make this the new location
      IF (spot_distance .le. spot_radius) THEN

          ylocation_new = ylocation_new + new_loc_y
          xlocation_new = xlocation_new + new_loc_x

! Need to check for the map edge (for wrapped boundaries) need to wrap the location

          IF (xlocation_new .gt. numcol) THEN
              xlocation_new = xlocation_new - numcol
          END IF

          IF (xlocation_new .lt. 1) THEN
              xlocation_new = numcol + xlocation_new
          END IF

          IF (ylocation_new .gt. numrow) THEN
              ylocation_new = ylocation_new - numrow
          END IF
```

```

        IF (ylocation_new .lt. 1) THEN
            ylocation_new = numrow + ylocation_new
        END IF

! Create effect at location
        N(ylocation_new,xlocation_new) = N(ylocation_new,xlocation_new) +
NutSlugAmount
        END IF

        END DO
        END DO
    end if

END SUBROUTINE

```

SUBROUTINE fishspot

```

    USE global_variables
    IMPLICIT NONE

    INTEGER :: iii, jjj, map_area, spot_radius_round, i, j
    INTEGER :: ylocation_new, xlocation_new, new_loc_y, new_loc_x
    REAL :: spot_area, spot_radius, spot_diameter, spot_distance

! Check for a spot, radius, or uniform effect
! If equal to 1 = spot (values = 2-9 = are circles of different radius)
    IF (fishradius .eq. 1 .or. fishradius .eq. 10) then

        IF (fishradius .eq. 1) THEN      !nutrient spot effect
            Z(ylocation_fish,xlocation_fish) = Z(ylocation_fish,xlocation_fish) -
            FishExt

            Fishfood = FishExt      !!!!!!!!!!!!!!!!!!!!!
            IF (Z(ylocation_fish,xlocation_fish) .lt. 0.0) THEN
                Z(ylocation_fish,xlocation_fish) = 0.0
                Fishfood = FishExt - (FishExt - Z(ylocation_fish,xlocation_fish))
            END IF

        END IF

        END IF

        IF (fishradius .eq. 10) THEN      !uniform fish addition
            Z = Z - FishExt

```

```
Fishfood = FishExt
```

```
DO i = 1,numcol
```

```
DO j = 1,numrow
```

```
IF (Z(i,j) .LT. 0.0) THEN !check for zero values
```

```
Z(i,j) = 0.0
```

```
Fishfood = FishExt - (FishExt - Z(i,j))
```

```
END IF
```

```
END DO !allows to keep track of bioenergetics
```

```
END DO !can allow movement based on food supply
```

```
!array operation not sure if will work
```

```
END IF
```

```
Else
```

```
! Decide on size of spot
```

```
! Minimum mapsize = 16x16 below this will get errors because will be less than one cell
```

```
! Spotsize will exceed the resolution of the grid
```

```
! Draw a circle around the selected point on the map
```

```
SELECT CASE (fishradius)
```

```
CASE (2) ! (1/2 the area of the map)
```

```
map_area = 2
```

```
CASE (3) ! (1/4 the area of the map)
```

```
map_area = 4
```

```
CASE (4) ! (1/8 the area of the map)
```

```
map_area = 8
```

```
CASE (5) ! (1/16 the area of the map)
```

```
map_area = 16
```

```
CASE (6) ! (1/32 the area of the map)
```

```
map_area = 32
```

```
CASE (7) ! (1/64 the area of the map)
```

```
map_area = 64
```

```
CASE (8) ! (1/128 the area of the map)
```

```
map_area = 128
```

```
CASE (9) ! (1/256 the area of the map)
```

```
map_area = 256
```

```
END SELECT
```

```
! Now calculate the area of the circle based on the mapsize
```

```
spot_area = ((numrow * numcol)/(map_area))
```

```
! Calculate the radius of the circle
```

```
spot_radius = SQRT(spot_area/3.14159)
```

```
! Round radius up to the nearest whole number
```

```
spot_radius_round = INT(spot_radius + 0.5)
```

```

! Calculate diameter of the circle
    spot_diameter = spot_radius_round * 2

! Loop over square which encompasses circle
    DO iii = 1, (spot_diameter + 1)    !down columns (Y)
      DO jjj = 1, (spot_diameter + 1) !across rows (X)

! Initialize location from value outside of subroutine (old location on map)
        ylocation_new = ylocation_fish
        xlocation_new = xlocation_fish

! Find location within square
        new_loc_y = (iii-(spot_radius_round + 1))
        new_loc_x = (jjj-(spot_radius_round + 1))
        spot_distance = SQRT(Real((new_loc_y * new_loc_y) + (new_loc_x *
new_loc_x)))

! Check to see if location falls within circle
! If it does then make this the new location
        IF (spot_distance .le. spot_radius) THEN

            ylocation_new = ylocation_new + new_loc_y
            xlocation_new = xlocation_new + new_loc_x

! Need to check for the map edge (for wrapped boundaries) need to wrap the location

            IF (xlocation_new .gt. numcol) THEN
                xlocation_new = xlocation_new - numcol
            END IF

            IF (xlocation_new .lt. 1) THEN
                xlocation_new = numcol + xlocation_new
            END IF

            IF (ylocation_new .gt. numrow) THEN
                ylocation_new = ylocation_new - numrow
            END IF

            IF (ylocation_new .lt. 1) THEN
                ylocation_new = numrow + ylocation_new
            END IF

! Create effect at location
            Z(ylocation_new,xlocation_new) = Z(ylocation_new,xlocation_new) -
FishExt

```



```
Fishfood = FishExt

IF (Z(ylocation_new,xlocation_new) .lt. 0.0) THEN
  Z(ylocation_new,xlocation_new) = 0.0
  Fishfood = FishExt - (FishExt - Z(ylocation_new,xlocation_new))
END IF

END IF

END DO
END DO

end if

END SUBROUTINE
```

References

- Abbott, M.R. 1993. Phytoplankton patchiness: Ecological implications and observation methods. Pages 37-49 in S.A. Levin, T.M. Powell, and J.H. Steele (eds.). *Patch Dynamics*. Lecture Notes in Biomathematics 96. Springer-Verlag, New York.
- Abraham, E.R. 1998. The generation of plankton patchiness by turbulent stirring. *Nature* 391:577-580.
- Allen, T.F.H. 1977. Scale in microscopic algal ecology: A neglected dimension. *Phycologia* 16:253-257.
- Allen, T.F.H. and T.W. Hoekstra. 1992. *Toward a Unified Ecology*. Columbia University Press, New York.
- Aref, H. 1984. Stirring by chaotic advection. *Journal of Fluid Mechanics* 143:1-21.
- Armstrong, R.A. 1994. Grazing limitation and nutrient limitation in marine ecosystems: Steady state solutions of an ecosystem model with multiple food chains. *Limnology and Oceanography* 39:597-608.
- Baird, D., R.E. Ulanowicz and W.R. Boynton. 1995. Seasonal nitrogen dynamics in the Chesapeake Bay: A network approach. *Estuarine Coastal and Shelf Science* 41:137-162.
- Barber, R.T. and F.P. Chavez. 1986. Ocean variability in relation to living resources during the 1982-83 El Nino. *Nature* 319:279-285.
- Bartell, S.M., J.E. Breck, R.H. Gardner, A.L. Brenkert. 1986. Individual parameter perturbation and error analysis of fish bioenergetics models. *Canadian Journal of Fisheries and Aquatic Sciences* 43:160-168.
- Bartell, S.M., A.L. Brenkert and S.R. Carpenter. 1988a. Parameter uncertainty and the behavior of a size-dependent plankton model. *Ecological Modelling* 40:85-95.
- Bartell, S.M., A.L. Brenkert, R.V. O'Neill and R.H. Gardner. 1988b. Temporal variation in regulation of production in a pelagic food web model. Pages 101-118 in S.R. Carpenter (ed). *Complex Interactions in Lake Communities*. Springer-Verlag, New York.
- Batchelor, G.K. 1967. *An Introduction to Fluid Mechanics*. Cambridge University Press, Cambridge.

- Bax, N.J. 1985. Application of multi- and univariate techniques of sensitivity analysis to SKEBUB, a biomass-based fisheries ecosystem model, parameterized to Georges Bank. *Ecological Modelling* 29:353-382.
- Beres, D.L. and D.M. Hawkins. 2001. Plackett-Burman techniques for sensitivity analysis of many-parametered models. *Ecological Modelling* 141:171-183.
- Berg, G.M., P.M. Glibert and C.-C. Chen. 1999. Dimension effects of enclosures on ecological processes in pelagic systems. *Limnology and Oceanography* 44:1331-1340.
- Bienfang, P.K. 1981. Sinking rates of heterogeneous, temperate phytoplankton populations. *Journal of Plankton Research* 3:235-253.
- Bigelow, H.B. and W.C. Schroeder. 1953. Fishes of the Gulf of Maine. *Fishery Bulletin* 74:1-577.
- Bloesch, J., P. Bossard, H. Buhner, H.R. Burgi and U. Uehlinger. 1988. Can results from limnocorral experiments be transferred to in situ conditions? *Hydrobiologia* 159:297-308.
- Boynton, W.R., W.M. Kemp and C.W. Keefe. 1982. A comparative analysis of nutrients and other factors influencing estuarine phytoplankton production. Pages 69-90 in V.S. Kennedy (ed). *Estuarine Comparisons*. Academic Press, New York.
- Boynton, W.R., J.H. Garber, R. Summers and W.M. Kemp. 1995. Inputs, transformations, and transport of nitrogen and phosphorus in Chesapeake Bay and selected tributaries. *Estuaries* 18: 285-314.
- Boynton, W.R. and W.M. Kemp. 2000. Influence of river flow and nutrient loading on selected ecosystem processes and properties in Chesapeake Bay. Pages 269-298 in J. Hobbie (ed). *Estuarine science: A Synthetic Approach to Research and Practice*. Island Press, Washington, DC.
- Brentnall, S.J., K.J. Richards, J. Brindley and E. Murphy. 2003. Plankton patchiness and its effect on larger-scale productivity. *Journal of Plankton Research* 25:121-140.
- Bricker, S.B., C.G. Clement, D.E. Pirhalla, S.P. Orlando and D.R.G. Farrow. 1999. National Estuarine Eutrophication Assessment. Effects of Nutrient Enrichment in the Nation's Estuaries. National Ocean Service, National Oceanic and Atmospheric Administration, Silver Spring, Maryland.
- Brown, J.L. 1975. *The Evolution of Behavior*. Norton, New York.

- Brylinsky, M. 1972. Steady-state sensitivity analysis of energy flow in a marine ecosystem. Pages 81-101 in B.C. Patten (ed). *Systems Analysis and Simulation in Ecology*. Academic Press, New York.
- Carpenter, S.R., S.W. Chisholm, C.J. Krebs, D.W. Schindler and R.F. Wright. 1995. Ecosystem experiments. *Science* 269:324–327.
- Caswell, H. 2000. *Matrix population models*. Sinauer, Sunderland, Massachusetts.
- Caswell, H. and M.G. Neubert. 1998. Chaos and closure terms in plankton food chain models. *Journal of Plankton Research* 20:1837-1845.
- Cerco, C.F. and T. Cole. 1993. Three-dimensional eutrophication model of Chesapeake Bay. *Journal of Environmental Engineering* 119:1006-1025.
- Chen, C.-C., J.E. Petersen, and W.M. Kemp. 1997. Spatial and temporal scaling of periphyton growth on walls of estuarine mesocosms. *Marine Ecology Progress Series* 155:1–15.
- Chen, C.-C., J. Petersen and W.M. Kemp. 2000. Nutrient uptake in experimental estuarine ecosystems: Scaling and partitioning rates. *Marine Ecology Progress Series* 200:103-116.
- Chen, Fa and J.D. Annan. 2000. The influence of different turbulence schemes on modeling primary production in a 1D coupled physical-biological model. *Journal of Marine Systems* 26:259-288.
- Cloern, J.E. 2001. Review. Our evolving conceptual model of the coastal eutrophication problem. *Marine Ecology Progress Series* 210: 223-253.
- Cullen, J.J. 1990. On models of growth and photosynthesis in phytoplankton. *Deep-Sea Research* 37: 667-683.
- Cullen, J.J., R.J. Geider, J. Ishizaka, D.A. Kiefer, J. Marra, E. Sakshaug and J.A. Raven. 1993. Toward a general description of phytoplankton growth for biogeochemical models. Pages 153-176 in G.T. Evans and M.J.R. Fasham (eds). *Towards a model of ocean biogeochemical processes*. Springer-Verlag, Berlin.
- Dadou, I., V. Garcon, V. Andersen, G.R. Flierl and C.S. Davis. 1996. Impact of the North Equatorial Current meandering on a pelagic ecosystem: A modeling approach. *Journal of Marine Research* 51:311-342.
- Dale, V.H., H.I. Jager, R.H. Gardner and A.E. Rosen. 1988. Using sensitivity and uncertainty analysis to improve predictions of broad-scale forest development. *Ecological Modelling* 42:165-178.

- D'Elia, C.F., J.G. Sanders and W.R. Boynton. 1986. Nutrient enrichment studies in a coastal plain estuary: phytoplankton growth in large scale, continuous cultures. *Canadian Journal of Fisheries and Aquatic Science* 43:397-406.
- Denman, K.L. 1976. Covariability of chlorophyll and temperature in the sea. *Deep-Sea Research* 23:539-550.
- Denman, K.L. 1992. Scale-determining biological-physical interactions in oceanic food webs. Pages 377-402 in P.S. Giller, A.G. Hildrew, and D.G. Raffaelli (eds). *Aquatic Ecology: Scale, Pattern and Process*. Oxford, England: Blackwell Science.
- Denman, K.L. and M.A. Pena. 1999. A coupled 1-D biological/physical model of the northeast subarctic Pacific Ocean with iron limitation. *Deep-Sea Res. II* 46:2877-2908.
- Denman, K.L. and T. Platt. 1976. The variance spectrum of phytoplankton in a turbulent ocean. *Journal of Marine Research* 34:593-601.
- Denman, K.L. and T. M. Powell. 1984. Effects of physical processes on planktonic ecosystems in the coastal ocean. *Oceanography and Marine Biology: An Annual Review* 22:125-165.
- Denman, K., A. Okubo and T. Platt. 1977. The chlorophyll fluctuation spectrum in the sea. *Limnology and Oceanography* 22:1033-1038.
- Diaz, R. and R. Rosenberg. 1995. Marine benthic hypoxia: A review of its ecological effects and the behavioural responses of benthic macrofauna. *Oceanography and Marine Biology: An Annual Review* 33: 245-303.
- Doney, S.C., D.M. Glover and R.G. Najjar. 1996. A new coupled, one-dimensional biological-physical model for the upper ocean: Applications to the JGOFS Bermuda Atlantic time-series study (BATS) site. *Deep-Sea Res. II* 43:591-624.
- Druon, J.N. and J.L. Fevre. 1999. Sensitivity of a pelagic ecosystem model to variations of process parameters within a realistic range. *Journal of Marine Systems* 19:1-26.
- Durrett, R. and S.A. Levin. 1994. Stochastic spatial models: A user's guide to ecological applications. *Philosophical Transactions of the Royal Society of London B*. 343:329-350.
- Dyke, P.P.G. and T. Robertson. 1985. The simulation of offshore turbulent dispersion using seeded eddies. *Applied Mathematical Modeling* 9:429-433.
- Edwards, A.M. 2001. Adding detritus to a nutrient-phytoplankton-zooplankton model: A dynamical-systems approach. *Journal of Plankton Research* 23:389-413.

- Edwards, A.M. and M.A. Bees. 2001. Generic dynamics of a simple plankton population model with a non-integer exponent of closure. *Chaos Soliton Fractal* 12:289-300.
- Edwards, A.M. and J. Brindley. 1996. Oscillatory behavior in a three-component plankton population model. *Dynamics and stability of systems* 11:347-370.
- Edwards, A.M. and J. Brindley. 1999. Zooplankton mortality and the dynamical behavior of plankton population models. *Bulletin of Mathematical Biology* 61:303-339.
- Edwards, C.A., H.P. Batchelder and T.M. Powell. 2000. Modeling microzooplankton and macrozooplankton dynamics within a coastal upwelling system. *Journal of Plankton Research* 22:1619-1648.
- Eppley, R., J. Roger and J. McCarthy. 1969. Half-saturation constants for uptake of nitrate and ammonia by marine phytoplankton. *Limnology and Oceanography* 14: 912-920.
- Evans, G.T. and J.S. Parslow. 1985. A model of annual plankton cycles. *Biological Oceanography* 3:327-347.
- Fasham, M.J.R. 1995. Variations in the seasonal cycle of biological production in subarctic oceans: A model sensitivity analysis. *Deep-Sea Research I* 42:1111-1149.
- Fasham, M.J.R., H.W. Ducklow and S.M. McKelvie. 1990. A nitrogen-based model of plankton dynamics in the oceanic mixed layer. *Journal of Marine Research* 48: 591-639.
- Fennel, K., M. Losch, J. Schroter and M. Wenzel. 2001. Testing a marine ecosystem model: Sensitivity analysis and parameter optimization. *Journal of Marine Systems* 28:45-63.
- Fischer, H.B., E.J. List, R.C.Y. Koh, J. Imberger and N.H. Brooks. 1979. *Mixing in Inland and Coastal Waters*. Academic Press, New York.
- Fong, P., M.E. Jacobson, M.C. Mescher, D. Lirman and M.C. Harwell. 1997. Investigating the management potential of a seagrass model through sensitivity analysis and experiments. *Ecological Applications* 7:300-315.
- Franks, P.J.S. 1992. Phytoplankton blooms at fronts: Patterns, scales, and physical forcing mechanisms. *Reviews in Aquatic Sciences* 6:121-137.
- Franks, P.J.S. 2005. Plankton patchiness, turbulent transport and spatial spectra. *Marine Ecology Progress Series* 294:295-309.

- Franks, P.J.S. and C. Chen. 1996. Plankton production in tidal fronts: A model of Georges Bank in summer. *Journal of Marine Research* 54:631-651.
- Franks, P.J.S., and C. Chen. 2001. A 3-D prognostic numerical model study of the Georges Bank ecosystem. Part II: biological-physical model. *Deep-Sea Research II* 48:457-482.
- Franks, P.J.S. and L.J. Walstad. 1997. Phytoplankton patches at fronts: A model of formation and response to wind events. *Journal of Marine Research* 55:1-29.
- Frost, B.W. 1997. Grazing control of phytoplankton stock in the open subarctic Pacific Ocean: A model assessing the role of mesozooplankton, particularly the large calanoid copepod *Neocalanus spp.*. *Marine Ecology Progress Series* 39:49-68.
- Frost, T.M., D.L. DeAngelis, S.M. Bartell, D.J. Hall and S.H. Hurlbert. 1988. Scale in the design and interpretation of aquatic community research. Pages 229-258 in S.R. Carpenter (ed). *Complex interactions in lake communities*. Springer-Verlag, New York.
- Gardner, R.H. 1984. A unified approach to sensitivity and uncertainty analysis. Pages 155-157 in M.H. Hamza (ed). *Applied Simulation and Modelling*. Proceedings of the IASTED International Symposium, San Francisco, U.S.A., June 4-6, 1984. ACTA Press, Anaheim, California.
- Gardner, R.H. and R.V. O'Neill. 1983. Parameter uncertainty and model predictions: A review of Monte Carlo results. Pages 245-257 in M.B. Beck and G. Van Straten (eds). *Uncertainty and Forecasting of Water Quality*. Springer-Verlag, New York.
- Gardner, R.H. and J.R. Trabalka. 1985. *Methods of Uncertainty Analysis for a Global Carbon Dioxide Model*. Department of Energy Technical Report, Washington, DC.
- Gardner, R.H., W.G. Cale and R.V. O'Neill. 1982. Robust analysis of aggregation error. *Ecology* 63:1771-1779.
- Gardner, R.H., V.H. Dale and R.V. O'Neill, R.V. 1990. Error propagation and uncertainty in process modeling. Pages 208-219 in R.K. Dixon, R.S. Meldahl, G.A. Ruark and W.G. Warren (eds). *Process Modeling of Forest Growth Responses to Environmental Stress*. Timber Press, Portland, Oregon.
- Gardner, R. H., W.W. Hargrove, M.G. Turner and W.H. Romme. 1996. Climate change, disturbances and landscape dynamics. Pages 149-172 in B.H. Walker and W.L. Steffen (eds). *Global Change and Terrestrial Ecosystems*. Cambridge University Press, Cambridge.

- Gardner, R.H., D.D. Huff, R.V. O'Neill, J.B. Mankin, J. Carney and J. Jones. 1980a. Application of error analysis to a marsh hydrology model. *Water Resources Research* 16:659-664.
- Gardner, R.H., R.V. O'Neill, J.B. Mankin and D. Kumar. 1980b. Comparative error analysis of six predator-prey models. *Ecology* 61:323-332.
- Gardner, R.H., R.V. O'Neill, J.B. Mankin and J.H. Carney. 1981. A comparison of sensitivity analysis and error analysis based on a stream ecosystem model. *Ecological Modelling* 12:177-194.
- Gardner, R.H., B. Rojder and U. Bergstrom. 1983. PRISM: A Systematic Method for Determining the Effect of Parameter Uncertainties on Model Predictions. Studsvik Energiteknik AB report/NW, Nykoping, Sweden.
- Gardner, R.H., W.M. Kemp, V.S. Kennedy and J.E. Petersen. 2001. *Scaling relations in experimental ecology*. Columbia University Press, New York.
- Gargett, A.E. 1989. Ocean turbulence. *Annual Reviews of Fluid Mechanics* 21:419-451.
- Gaskin, D.E. 1976. The evolution, zoogeography and ecology of Cetacea. *Oceanography and Marine Biology: An Annual Review* 43:163-178.
- Ghosal, S. and S. Mandre. 2003. A simple model illustrating the role of turbulence on phytoplankton blooms. *Journal of Mathematical Biology* 46:333-346.
- Giller P.S., A.G. Hildrew and D.G. Raffaelli. 1992. *Aquatic Ecology: Scale, Pattern and Process*. Blackwell Science, Oxford, England.
- Glibert, P. M. 1998. Interactions of top-down and bottom-up control in planktonic nitrogen cycling. *Hydrobiologia* 363:1-12.
- Goericke, R. and N.A. Welschmeyer. 1998. Response of Sargasso Sea phytoplankton biomass, growth rates and primary production to seasonally varying physical forcing. *Journal of Plankton Research* 20:2223-2249.
- Goldman, J.C. and P.M. Glibert. 1983. Kinetics of inorganic nitrogen uptake by phytoplankton. Pages 233-274 in E.J. Carpenter and D.G. Capone (ed). *Nitrogen in the Marine Environment*. Academic Press, New York.
- Gower, J.F.R., K.L. Denman and R.J. Holyer. 1980. Phytoplankton patchiness indicates the fluctuation spectrum of mesoscale oceanic structure. *Nature* 288:13-15.
- Grieco, L, L.-B. Tremblay and E. Zambianchi. 2005. A hybrid approach to transport processes in the Gulf of Naples: an application to phytoplankton and zooplankton population dynamics. *Continental Shelf Research* 25:711-728.

- Grunbaum, D. 1994. Translating stochastic density-dependent individual behavior with sensory constraints to an Eulerian model of animal swarming. *Journal of Mathematical Biology* 33:139-161.
- Gurney, W.S.C., A.R. Veitch, I. Cruickshank and G. McGeachin. 1998. Circles and Spirals: Population Persistence in a Spatially Explicit Predator-Prey Model. *Ecology* 79:2516-2530.
- Hakanson, L. 2000. The role of characteristic coefficients of variation in uncertainty and sensitivity analyses, with examples related to the structuring of lake eutrophication models. *Ecological Modelling* 131:1-20.
- Halvorsen, E., O.P. Pedersen, D. Slagstad, K.S. Tande, E.S. Fileman and S.D. Batten. 2001. Microzooplankton and mesozooplankton in an upwelling filament off Galicia: Modelling and sensitivity analysis of the linkages and their impact on the carbon dynamics. *Progress in Oceanography* 51:499-513.
- Hamby, D.M. 1995. A comparison of sensitivity analysis techniques. *Health Physics* 68:195-204.
- Hamner, W.M., P.P. Hamner, S.W. Strand and R.W. Gilmer. 1983. Behavior of Antarctic krill, *Euphausia superba*: Chemoreception, feeding, schooling, and molting. *Science* 220:433-435.
- Haney, J.D. and G.A. Jackson. 1996. Modeling phytoplankton growth rates. *Journal of Plankton Research* 18:63-85.
- Harding, L.W., B.W. Meeson and T.R. Fisher. 1985. Photosynthesis patterns in Chesapeake Bay phytoplankton: Short- and long-term responses of P-I curve parameters to light. *Marine Ecology Progress Series* 26:99-111.
- Harding, L.W., B.W. Meeson and T.R. Fisher. 1986. Phytoplankton production in two east coast estuaries: Photosynthesis-light functions and patterns of carbon assimilation in Chesapeake and Delaware Bays. *Estuarine Coastal and Shelf Science* 23:773-806.
- Hardy, A.C. and E.R. Gunther. 1935. The plankton of the South Georgia whaling grounds and adjacent waters, 1926–1927. *Discovery Reports* 11:1-456.
- Harris, G.P. 1980. Temporal and Spatial Scales in Phytoplankton Ecology. Mechanisms, Methods, Models, and Management. *Canadian Journal of Fisheries and Aquatic Sciences* 37:877-900.
- Haury, L.R., J.A. McGowan and P.H. Wiebe. 1978. Patterns and processes in the time-space scales of plankton distributions. Pages 277-327 in J.H. Steele (ed). *Spatial*

Pattern in Plankton Communities. NATO Conference Series IV, Volume. 3. New York: Plenum.

- Heath, M.R. and E.D. Houde. 2001. Evaluating and modeling foraging performance of planktivorous and piscivorous fish: effects of containment and issues of scale. Pages 191-222 in R.H. Gardner, W.M. Kemp, V.S. Kennedy and J.E. Petersen (eds). *Scaling relations in experimental ecology*. Columbia University Press, New York.
- Holling, C.S. 1959. The components of predation as revealed by a study of small mammal predation on the European pine sawfly. *Canadian Entomologist* 91:293-320.
- Holling, C.S. 1992. Cross-scale morphology, geometry, and dynamics of ecosystems. *Ecological Monographs* 62:447-502.
- Holloway, G. 1986. Eddies, waves, circulation and mixing: statistical geofluid mechanics. *Annual Review of Fluid Mechanics* 18:91-147.
- Holloway, G. 2004. From classical to statistical ocean dynamics. *Surveys in Geophysics* 25:203-219.
- Homma, T. and A. Saltelli. 1996. Importance measures in global sensitivity analysis of nonlinear models. *Reliability Engineering and System Safety* 52:1-17.
- Hood, R.R., H.V. Wang, J.E. Purcell, E.D. Houde and L.W. Harding. 1999. Modeling particles and pelagic organisms in Chesapeake Bay: convergent features control plankton distributions. *Journal of Geophysical Research* 104:1223-1243.
- Hopfinger, E.J. 1987. Turbulence in stratified fluids: a review. *Journal of Geophysical Research* 92:5287-5303.
- Horne, J.K. and D.C. Schneider. 1997. Spatial variance of mobile aquatic organisms: Capelin and cod in Newfoundland coastal waters. *Philosophical Transactions of the Royal Society of London B* 352:633-642.
- Howarth, R.W., G. Billen, D. Swaney, A. Townsend, N. Jaworski, K. Lajtha, J.A. Downing, R. Elmgren, N. Caraco, T. Jordan, F. Berendse, J. Freney, V. Kudeyarov, P. Murdoch and Z. Zhao-Liang. 1996. Regional nitrogen budgets and riverine N & P fluxes for the drainages to the North Atlantic Ocean: Natural and human influences. *Biogeochemistry* 35:75-79.
- Huisman, J. and F.J. Weissing. 1994. Light-limited growth and competition for light in well-mixed aquatic environments: An elementary model. *Ecology* 75:507-520.

- Hutchinson, G.E. 1961. The paradox of the plankton. *The American Naturalist* 95:137-145.
- Hwang, J.S., J.H. Costello and J.R. Strickler. 1994. Copepod grazing in turbulent flow: elevated foraging behavior and habituation of escape responses. *Journal of Plankton Research* 16:421-431.
- Inman, A.L. and J. Krebs. 1987. Predation and group living. *Trends in Ecology and Evolution* 2:31-32.
- James, I.D. 1996. Advection schemes for shelf sea models. *Journal of Marine Systems* 8:237-254.
- James, I.D. 2002. Modelling pollution dispersion, the ecosystem and water quality in coastal waters: a review. *Environmental Modelling and Software* 17:363-385.
- Johnson, B.H., Keu.W. Kim, R.E. Heath, B.B. Hsieh and H.Lee. Bulter. 1993. Validation of three-dimensional hydrodynamic model of Chesapeake Bay. *Journal of Hydraulic Engineering* 119:3-20.
- Kantha, L.H. and C.A. Clayson. 2000. *Numerical Models of Oceans and Oceanic Processes*. Academic Press. San Diego.
- Kareiva, P. and M. Andersen. 1988. Spatial Aspects of Species Interactions: The Wedding of Models and Experiments. Pages 35-50 in A. Hastings (ed). *Community Ecology*. Springer-Verlag, New York.
- Kemp, W.M., M.T. Brooks and R.R. Hood. 2001. Nutrient enrichment, habitat variability and trophic transfer efficiency in simple models of pelagic ecosystems. *Marine Ecology Progress Series* 223:73-87.
- Kemp, W.M., P. Sampou, J. Caffrey, M. Mayer, K. Henriksen and W.R. Boynton. 1990. Ammonium recycling versus denitrification in Chesapeake Bay sediments. *Limnology and Oceanography* 35:1545-1563.
- Kemp, W.M., J. Faganeli, S. Pskaric, E.M. Smith and W.R. Boynton. 1999. Pelagic-benthic coupling and nutrient cycling. Pages 295-340 in T.C. Malone, A. Malej, L. Harding, N Smodlaka and R.E. Turner (eds). *Ecosystems at the Land-Sea Margin*. American Geophysical Union, Washington, DC.
- Kemp, W.M., J.E. Petersen and R.H. Gardner. 2001. Scale-dependence and the problem of extrapolation: Implications for experimental and natural coastal ecosystems. Pages 3-57 in R.H. Gardner, W.M. Kemp, V.S. Kennedy and J.E. Petersen (eds.). *Scaling Relations in Experimental Ecology*. Columbia University Press, NY.

- Kemp, W.M., W.R. Boynton, J.E. Adolf, D.F. Boesch, W.C. Boicourt, G. Brush, J.C. Cornwell, T.R. Fisher, P.M. Glibert, J.D. Hagy, L.W. Harding, E.D. Houde, D.G. Kimmel, W.D. Miller, R.I.E. Newell, M.R. Roman, E.M. Smith and J.C. Stevenson. 2005. Eutrophication of Chesapeake Bay: historical trends and ecological interactions. *Marine Ecology Progress Series* 303:1-29.
- Kierstead, H. and L.B. Slobodkin. 1953. The size of water masses containing plankton blooms. *Journal of Marine Research* 12:141-147.
- Kiorboe, T. 1993. Turbulence, phytoplankton cell size, and the structure of pelagic food webs. *Advances in Marine Biology* 29:1-72.
- Kiorboe, T. and E. Saiz. 1995. Planktivorous feeding in calm and turbulent environments, with emphasis on copepods. *Marine Ecology Progress Series* 122:135-145.
- Kirk, J.T.O. 1994. *Light and Photosynthesis in Aquatic Ecosystems*. Cambridge University Press, New York.
- Klein, P. and B.L. Hua. 1990. The mesoscale variability of the sea surface temperature: An analytical and numerical model. *Journal of Marine Research* 48:729-763.
- Klepper, O. 1997. Multivariate aspects of model uncertainty analysis: Tools for sensitivity analysis and calibration. *Ecological Modelling* 101:1-13.
- Kolmogorov, A.N. 1941. The local structure of turbulence in an incompressible viscous fluid for very large Reynolds number. *Akademiia Nauk SSSR Comptes Rendus* 30:299-303.
- Kundu, P.K. 1990. *Fluid mechanics*. Academic Press, Inc, San Diego.
- Landahl, M.T. and E. Mollo-Christensen. 1986. *Turbulence and random processes in fluid mechanics*. Cambridge University Press, Cambridge.
- Lasker, R. 1975. Field criteria for survival of anchovy larvae: The relation between inshore chlorophyll maximum layers and successful first feeding. *Fishery Bulletin* 73:453-462.
- Laurance W.F., S.G. Laurance, L.V. Ferreira, J.M. Rankin de Merona, C. Gascon and T.E. Lovejoy. 1997. Biomass collapse in Amazonian forest fragments. *Science* 278:1117-1118.
- Lovejoy, S., W.J.S. Currie, Y. Tessier, M.R. Claereboudt, E. Bourget, J.C. Roff, and D. Schertzer. 2001. Universal multifractals and ocean patchiness: phytoplankton, physical fields and coastal heterogeneity. *Journal of Plankton Research* 23: 117-141.

- Lavorel, S., R.V. O'Neill and R.H. Gardner. 1994. Spatio-temporal dispersal strategies and annual plant species coexistence in a structured landscape. *Oikos* 71:75-88.
- Lazier, J.R.N. and K.H. Mann. 1989. Turbulence and the diffusive layers around small organisms. *Deep-Sea Research* 36:1721-1733.
- LeBrasseur, R.J., W.E. Barraclough, O.D. Kennedy and T.R. Parsons. 1969. Production studies in the strait of Georgia. Part III. Observations on the food of larval and juvenile fish in the Fraser River plume, February to May, 1967. *Journal of Experimental Marine Biology and Ecology* 3:39-50.
- Lekan, J.F. and R.E. Wilson. 1978. Spatial variability of phytoplankton biomass in the surface waters of Long Island. *Estuarine Coastal Marine Sciences* 6:239-251.
- Levin, S.A. 1992. The problem of pattern and scale in ecology. *Ecology* 73:1943-1967.
- Levin, S.A., A. Morin and T.M. Powell. 1989. Patterns and processes in the distribution and dynamics of Antarctic krill. Pages 281-299 in Scientific Committee for the Conservation of Antarctic Marine Living Resources Selected Scientific Papers Part 1, SC-CAMLR-SSP/5, CCAMLR, Hobart, Tasmania, Australia.
- Liebig, von J. 1840. *Organic Chemistry and its Application to Agriculture and Physiology*. Taylor and Walton, London.
- Lopez, C., Hernandez-Garcia E., O. Piro, A. Vulpiani, and E. Zambianchi. 2001. Population dynamics advected by chaotic flows: A discrete-time map approach. *Chaos* 11:397-403.
- Lorenzen, C.J., 1972. Extinction of light in the ocean by phytoplankton. *J. Cons. Int. Explor. Mer.* 34, 262-267.
- Loukos, H., B. Frost, D.E. Harrison and J.W. Murray. 1997. An ecosystem model with iron limitation of primary production in the equatorial Pacific at 140W. *Deep-Sea Research II* 44:2221-2249.
- Lovejoy, S., W.J.S. Currie, Y. Tessier, M.R. Claereboudt, E. Bourget, J.C. Roff and D. Schertzer. 2001. Universal multifractals and ocean patchiness: phytoplankton, physical fields and coastal heterogeneity. *Journal of Plankton Research* 23: 117-141.
- Mackas, D.L., K.L. Denman and M.R. Abbott. 1985. Plankton Patchiness: Biology in the Physical Vernacular. *Bulletin of Marine Science* 37:652-674.
- Malone, T.C., D.J. Conley, P.M. Glibert, L.W. Harding, Jr. and K. Sellner. 1996. Scales of nutrient limited phytoplankton productivity: the Chesapeake Bay example. *Estuaries* 19:371-385.

- Margalef, R. 1979. Life-forms of phytoplankton as survival alternatives in an unstable environment. *Oceanol Acta* 1:493-509.
- Marguerit, C., D. Schertzer, F. Schmitt and S. Lovejoy. 1998. Copepod diffusion within multifractal phytoplankton fields. *Journal of Marine Systems* 16:69-83.
- Mariani, P., V. Botte and M.R. d'Alcala. 2005. An object-oriented model for the prediction of turbulence effects on plankton. *Deep-Sea Research II*. 52:1287-1307.
- Marquet, P.A., M.-J. Fortin, J. Pineda, D.O. Wallin, J. Clark, Y. Wu, S. Bollens, C.M. Jacobi and R.D. Holt. 1993. Ecological and evolutionary consequences of patchiness: A marine-terrestrial perspective. Pages 277-304 in S.A. Levin, T.M. Powell and J.H. Steele (eds). *Patch Dynamics. Lecture Notes in Biomathematics* 96. New York: Springer-Verlag.
- Matsuda, H., K. Kawasaki, N. Shigesada, E. Teramoto and L.M. Ricciardi. 1986. Switching effect on stability of the prey-predator system with three trophic levels. *Journal of Theoretical Biology* 122:251-262.
- McGillicuddy Jr. D.J., A.R. Robinson and J.J. McCarthy. 1995. Coupled physical modeling of the spring bloom in the North Atlantic: three dimensional bloom and post-bloom processes. *Deep Sea Research I* 42:1359-1398.
- Middleton J.F. 1985. Drifter spectra and diffusivities. *Journal of Marine Research* 43:37-55.
- Miller, D.R., 1974. Sensitivity analysis and validation of simulation models. *Journal of Theoretical Biology* 48 :345-360.
- Monod, J., 1942. *Recherches sur la croissance des cultures bacteriennes*. Hermann, Paris, France.
- Moum J.N. 1996. Energy-containing scales of turbulence in the ocean. *Journal of Geophysical Research* 101:14095-14109.
- Mowitt, W.P., E.D. Houde, D. Hinkle and A. Sanford. 2006. Growth of planktivorous bay anchovy *Anchoa mitchilli*, top-down control, and scale-dependence in estuarine mesocosms. *Marine Ecology Progress Series* 308:255-269.
- Murdoch, W.W., R.M. Nisbet, E. McCauley, A.M. DeRoos and W.S.C. Gurney. 1998. Plankton abundance and dynamics across nutrient levels: tests of hypotheses. *Ecology* 79: 1339-1356.

- Murray, A.G. and J.S. Parslow. 1999. The analysis of alternative formulations in a simple model of a coastal ecosystem. *Ecological Modelling* 119:149-166.
- National Research Council (NRC). 1995. *Understanding Marine Biodiversity*. National Academy Press, Washington, DC.
- National Research Council (NRC). 2000. *Clean coastal waters. Understanding and reducing the effect of nutrient pollution*. National Academy Press, Washington, DC.
- Nixon, S.W. and M.E.Q. Pilson. 1983. Nitrogen in estuarine and coastal marine ecosystems. Pages 565-648 in E.J. Carpenter and D.G. Capone (eds). *Nitrogen in the Marine Environment*. Plenum Press, New York.
- Nixon, S.W. 1988. Physical energy inputs and the comparative ecology of lake and marine ecosystems. *Limnology and Oceanography* 33:1005-1025.
- Nixon, S.W. 1995. Coastal marine eutrophication: A definition, social causes, and future concerns. *Ophelia* 41:199-219.
- Norris, K.S. and C. R. Schilt. 1987. Cooperative societies in three-dimensional space: On the origins of aggregations, flocks, and schools, with special reference to dolphins and fish. *Ethology and Sociobiology* 9:149-179.
- O'Neill, R.V., S.M. Bartell and R.H. Gardner. 1983. Patterns of toxicological effects in ecosystems: A modeling study. *Environmental Toxicology and Chemistry* 2:451-461.
- Oguz, T., H.W. Ducklow, P. Malanotte-Rizzoli, J.W. Murray, E.A. Shushkina, V.I. Vedernikov, U. Unluata. 1999. A physical-biochemical model of plankton productivity and nitrogen cycling in the Black Sea. *Deep-Sea Research I* 46:597-636.
- Okubo, A. 1976. Remarks on the use of 'diffusion diagrams' in modeling scale-dependent diffusion. *Deep-Sea Research* 23:1213-1214.
- Okubo, A. 1980. *Diffusion and Ecological Problems: Mathematical Models*. Biomathematics 10. Springer-Verlag, New York.
- Okubo, A. 1986. Dynamical aspects of animal grouping: Swarms, schools, flocks, and herds. *Advanced Biophysics* 22:1-94.
- O'Neill, R.V. 1973. Error analysis of ecological models. Page 1035 in D.J. Nelson (ed). *Radionuclides in ecosystems, Proceedings of the Third National Symposium on Radioecology*, National Technical Information Service, Springfield, Virginia.

- Oreskes, N., K. Shrader-Frechette and K. Belitz. 1994. Verification, validation, and confirmation of numerical models in the earth sciences. *Science* 263:641-646.
- Oschlies, A. and V. Garçon. 1998. Eddy-induced enhancement of primary production in a model of the North Atlantic Ocean. *Nature* 394:266-269.
- Ottino, J.M. 1990. Mixing, chaotic advection and turbulence. *Annual Review of Fluid Mechanics* 22:207-254.
- Parsons, T.R., M. Takahashi and B. Hargrave. 1984. *Biological Oceanographic Processes*. Pergamon Press, New York.
- Pastres, R., D. Franco, G. Pecelik, C. Solidoro and C. Dejak. 1997. Local sensitivity analysis of a distributed parameters water quality model. *Reliability Engineering & System Safety* 57:21-30.
- Patel, D., K. Gugesarajah and B. Thake. 2004. Modelling diatom growth in turbulent waters. *Water Research* 38:2713-2725.
- Pauly, D., V. Christensen, J. Dalsgaard, R. Froese and F. Torres Jr. 1998. Fishing down marine food webs. *Science* 279:860-863.
- Petchey O.L., A. Gonzalez and H.B. Wilson. 1997. Effects on population persistence: the interaction between environmental noise colour, intraspecific competition and space. *Proceedings of the Royal Society of London Series B-Biological Sciences* 264:1841-1847.
- Peters, R.H. 1991. *A Critique for Ecology*. Cambridge University Press, Cambridge, England.
- Petersen, J. E., C.-C. Chen and W. M. Kemp. 1997. Scaling aquatic primary productivity: Experiments under nutrient- and light-limited conditions. *Ecology* 78:2326-2338.
- Petersen, J.E., L.P. Sanford and W.M. Kemp. 1998. Coastal plankton responses to turbulent mixing in experimental ecosystems. *Marine Ecology Progress Series* 171:23-41.
- Petersen, J.E., J.C. Cornwell and W.M. Kemp. 1999. Implicit scaling in the design of experimental aquatic ecosystems. *Oikos* 85:3-18.
- Petersen JE, W. Kemp, R. Bartelson, W. Boynton, C-C. Chen, J. Cornwell, R. Gardner, D. Hinkle, E. Houde, T. Malone, W. Mowitt, L. Murray, L. Sanford, J. Stevenson, K. Sundberg and S. Suttles. 2003. Multiscale experiments in coastal ecology: improving realism and advancing theory. *Bioscience* 53:1181-1197.

- Peterson, D.L. and V.T. Parker. 1998. *Ecological Scale. Complexity in Ecological Systems*. Columbia University Press, New York.
- Petrovskii, S.V. 1999. On the plankton front waves accelerated by marine turbulence. *Journal of Marine Systems* 21:179-188.
- Petrovskii, S.V., M.E. Vinogradov and A.Y. Morozov. 2002. Formation of the patchiness in the plankton horizontal distribution due to biological invasion in a two species model with account for the Allee effect. *Oceanology* 42:363-372.
- Platt, T. 1972. Local phytoplankton abundance and turbulence. *Deep-Sea Research* 19:183-187.
- Platt, T. and K.L. Denman. 1975. Spectral Analysis in Ecology. *Annual Review of Ecology and Systematics* 6:189-210.
- Postma, H., W.M. Kemp, J.M. Colebrook, J. Horwood, I.R. Joint, R. Lampitt, S.W. Nixon, M.E.Q. Pilson and F. Wulff. 1984. Nutrient cycling in estuarine and coastal marine ecosystems. Pages 651-662 in M.J.R. Fasham (ed). *Flows of energy and material in marine ecosystems*. Plenum Press, New York.
- Powell, T.M. 1989. Physical and Biological Scales of Variability in Lakes, Estuaries, and the Coastal Ocean. Pages 157-176 in J. Roughgarden, R.M. May and S.A. Levin (eds). *Perspectives in Ecological Theory*. Princeton University Press, Princeton, NJ.
- Powell, T.M. and J.H. Steele. 1995. *Ecological Time Series*. Chapman and Hall, New York.
- Powell, T.M. and A. Okubo. 1994. Turbulence, diffusion and patchiness in the sea. *Philosophical Transactions of the Royal Society of London B* 343:11-18.
- Powell, T.M., P.J. Richerson, T.M. Dillon, B.A. Agee, B.J. Dozier, D.A. Godden and L.O. Myrup. 1975. Spatial Scales of Current Speed and Phytoplankton Biomass Fluctuations in Lake Tahoe. *Science* 189:1088-1090.
- Proehl, J.A., D.R. Lynch, D.J. McGillicuddy Jr. and J.R. Ledwell. 2005. Modeling turbulent dispersion on the North Flank of Georges Bank using Lagrangian Particle Methods. *Continental Shelf Research* 25:875-900.
- Redfield, A.C., B.H. Ketchum and F.A. Richards. 1963. The influence of organisms on the composition of sea water. Pages 26-27 in M.N. Hill (ed). *The Sea, Ideas and Observations on Progress in the Study of the Seas*. Interscience, New York.

- Reigada, R., R.M. Hillary, M.A. Bees, J.M. Sancho and F. Sagues. 2003. Plankton blooms induced by turbulent flows. *Proceedings of the Royal Society of London B* 270:875-880.
- Reynolds, C.S. 1990. Temporal scales of variability in pelagic environments and the response of phytoplankton. *Freshwater Biology* 23:25-53.
- Richardson, L.F. 1922. *Weather prediction by numerical process*. Cambridge University Press, Cambridge.
- Riddle, A.M. 1998. The specification of mixing in random walk models for dispersion in the sea. *Continental Shelf Research* 18:441-456.
- Ripa J., P. Lundberg and V. Kaitala. 1998. A general theory of environmental noise in ecological food webs. *American Naturalist* 151:256-263.
- Roman, M., X. Zhang, C. McGilliard and W. Boicourt. 2005. Seasonal and annual variability in the spatial patterns of plankton biomass in Chesapeake Bay. *Limnology and Oceanography* 50:480-492.
- Root, T.L. and S.H. Schneider. 1995. Ecology and climate: Research strategies and implications. *Science* 269:334-341.
- Rose, K.A. and G.L. Swartzman. 1981. A review of parameter sensitivity methods applicable to ecosystem models. U.S. Nuclear Regulatory Commission, Washington, DC.
- Rose, K.A., E.P. Smith, R.H. Gardner, A.L. Brenkert and S.M. Bartell. 1991. Parameter sensitivities, monte carlo filtering, and model forecasting under uncertainty. *Journal of Forecasting* 10:117-133.
- Rosenberg, R. 1985. Eutrophication-the future marine coastal nuisance? *Marine Pollution Bulletin* 16:227-231.
- Ross, O.N. and J. Sharples. 2004. Recipe for 1-D Lagrangian particle tracking models in space-varying diffusivity. *Limnology and Oceanography Methods* 2:289-302.
- Rothlisberg, P.C., J.A. Church and A.M.G. Forbes. 1983. Modelling the advection of vertically migrating shrimp larvae. *Journal of Marine Research* 41:511-538.
- Rothschild, B.J. and T.R. Osborn. 1988. Small-scale turbulence and plankton contact rates. *Journal of plankton Research* 10:465-474.
- Roughgarden, J.D. 1978. Influence of competition on patchiness in a random environment. *Theoretical Population Biology* 14:185-203.

- Ryther, J. and W. Dunstan. 1971. Nitrogen, phosphorus, and eutrophication in the coastal marine environment. *Science* 171: 1008-1112.
- Sanford, L.P. 1997. Turbulent mixing in experimental ecosystem studies. *Marine Ecology Progress Series* 161:265-293.
- Sanford, L.P. and S.M. Crawford. 2000. Mass transfer versus kinetic control of uptake across solid-water boundaries. *Limnology and Oceanography* 45:1180-1186.
- SAS. 2001. The SAS System for Windows, Version 8.01. Windows Version 5.0.2195, SAS Institute Inc., Cary, NC.
- Saunders, P.M. 1992. Space and time variability of temperature in the upper ocean. *Deep-Sea Research* 19:467-480.
- Saupe, D. 1988. Algorithms for random fractals. Pages 71-113 in H.-O. Petigen, and D. Saupe (eds). *The science of fractal images*. New York, Springer Verlag.
- Schneider, D.C. 1991. The role of fluid dynamics in the ecology of marine birds. *Oceanography and Marine Biology: An Annual Review* 29:487-521.
- Schneider, D.C. 1992. Scale-dependent patterns and species interactions in marine nekton. Pages 441-467 in P.S. Giller, A.G. Hildrew and D.G. Raffaelli (eds). *Aquatic Ecology: Scale, Pattern and Process*. Oxford, England: Blackwell Science.
- Schneider, D.C. 1994. *Quantitative Ecology: Spatial and Temporal Scaling*. Academic Press, San Diego.
- Seuront, L. and F.G. Schmitt. 2005. Multiscaling statistical procedures for the exploration of biophysical couplings in intermittent turbulence. Part I. Theory. *Deep-Sea Research II* 52:1308-1324.
- Seuront, L., F. Schmitt, Y. Lagadeuc, D. Schertzer and S. Lovejoy. 1999. Universal multifractal analysis as a tool to characterize multiscale intermittent patterns: example of phytoplankton distribution in turbulent coastal waters. *Journal of Plankton Research* 21:877-922.
- Seuront, L., V. Gentilhomme and Y. Lagadeuc. 2002. Small-scale nutrient patches in tidally mixed coastal waters. *Marine Ecology Progress Series* 232: 29-44.
- Sheldon, R., W.A. Prakash and W.H. Sutcliffe Jr. 1972. The size distribution of particles in the ocean. *Limnology and Oceanography* 17:327-340.
- Sinclair, M. 1987. *Marine Populations*. University of Washington Press, Seattle.

- Skellam, J.G. 1951. Random dispersal in theoretical populations. *Biometrika* 38:196-218.
- Smith, E.M. and W.M. Kemp. 1995. Seasonal and regional variations in plankton community production and respiration for Chesapeake Bay. *Marine Ecology Progress Series* 116: 217-231.
- Southward, A.J. 1980. The western English Channel: An inconstant ecosystem? *Nature* 285:361-366.
- Squires, K.D. and H Yamazaki. 1995. Preferential concentration of marine particles in isotropic turbulence. *Deep Sea Research I* 42:1989-2004.
- Staples, D.J. 1979. Seasonal migration patterns of postlarval and juvenile banana prawns, *Penaeus merguensis* de Man, in the major rivers of the Gulf of Carpentaria, Australia. *Australian Journal of Marine and Freshwater Research* 30:143-157.
- Steele, J.H. 1978. *Spatial Pattern in Plankton Communities*. NATO Conference Series IV, Vol. 3 Plenum, New York.
- Steele, J.H. 1985. A comparison of terrestrial and marine ecological systems. *Nature* 313:355-358.
- Steele, J.H. 1991. Marine ecosystem dynamics: Comparison of scales. *Ecological Research* 6:175-183.
- Steele, J.H., 1998. Incorporating the microbial loop in a simple plankton model. *Proc. R. Proceedings of the Royal Society of London Series B-Biological Sciences* 265: 1771-1777.
- Steele, J.H. and E.W. Henderson. 1981. A simple plankton model. *American Naturalist* 117: 676-691.
- Steele, J.H. and E.W. Henderson, 1992a. The role of predation in plankton models. *Journal of Plankton Research* 14:157-172.
- Steele, J.H. and E.W. Henderson. 1992b. A simple-model for plankton patchiness. *Journal of Plankton Research* 14:1397-1403.
- Stickney, H.L., R.R. Hood and D.K. Stoecker. 2000. The impact of mixotrophy on planktonic marine ecosystems. *Ecological Modelling* 125:203-230.
- Stommel, H. 1963. Varieties of oceanographic experience. *Science* 139:572-576.
- Strass, V.H. 1992. Chlorophyll patchiness caused by mesoscale upwelling at fronts. *Deep-Sea Research* 39:75-96.

- Sund, P.N., M. Blackburn and F. Williams. 1981. Tunas and their environment in the Pacific Ocean: A review. *Oceanography and Marine Biology: An Annual Review* 19:443-512.
- Tennekes, H. and J.L. Lumley. 1972. *A first course in turbulence*. MIT press, Cambridge, MA.
- Tilman, D. 1989. Ecological experimentation: Strengths and conceptual problems. Pages 136-157 in G.E. Likens (ed). *Long-Term Studies in Ecology*. New York: Springer-Verlag.
- Tilman, D. and P. Kareiva. 1997. *Spatial Ecology: The Role of Space in Population dynamics and Interspecific Interactions*. Princeton University Press, Princeton, NJ.
- Tomovic, R. and W.J. Karplus. 1963. *Sensitivity Analysis of Dynamic Systems*. McGraw-Hill, New York.
- Tomovic, R. and M. Vukobratovic. 1972. *General Sensitivity Theory*. Elsevier, New York.
- Turner, G. and T.J. Pitcher. 1986. Attack dilution, search, and abatement. *American Naturalist* 128:228-240.
- Turner, R.E. and N.N. Rabalais. 1994. Coastal eutrophication near the Mississippi river delta. *Nature* 368: 619-621.
- Umlauf, L. and H. Burchard. 2005. Second-order turbulence closure models for geophysical boundary layers. A review of recent work. *Continental Shelf Research* 25:795-827.
- Van Dam, G.C., R.V. Ozmidov, K.A. Korotenko and Jo M. Suijlen. 1999. Spectral structure of horizontal water movement in shallow seas with special reference to the North Sea, as related to the dispersion of dissolved matter. *Journal of Marine Systems* 21:207-228.
- Venrick, E.L. 1982. Phytoplankton in an oligotrophic ocean: Observations and questions. *Ecological Monographs* 52:129-154.
- Vested, H.J., J.W. Baretta, L.C. Ekebjærg and A. Labrosse. 1996. Coupling of hydrodynamical transport and ecological models for 2D horizontal flow. *Journal of Marine Systems* 8: 255-267.
- Vilar, J.M.G., R.V. Sole and J.M. Rubi. 2003. On the origin of plankton patchiness. *Physica A-Statistical Mechanics and its Applications* 317:239-246.

- Visser, A.W. 1997. Using random walk models to simulate the vertical distribution of particles in a turbulent water column. *Marine Ecology Progress Series* 158:275-281.
- Vitousek, P.M., J.D. Aber, R.W. Howarth, G.E. Likens, P.A. Matson, D.W. Schindler, W.H. Schlesinger and D.G. Tilman. 1997. Human alteration of the global nitrogen cycle: Sources and consequences. *Ecological Applications* 7:737-750.
- Walters, R.A. 2005. Coastal ocean models: two useful finite element methods. *Continental Shelf Research* 25:775-793.
- Wang, P., L.C. Linker, R. Batiuk and C. Cerco. 2006. Surface analysis of Chesapeake Bay water quality response to different nutrient and sediment loads. *Journal of Environmental Engineering-ASCE* 132:377-383.
- Ward, B.B. 1996. Nitrification and denitrification: probing the Nitrogen Cycle in aquatic environments. *Microbial Ecology* 32:247-261.
- Weber, L.H., S.Z. El-Sayed and I. Hampton. 1986. The variance spectra of phytoplankton, krill and water temperature in the Antarctic Ocean south of Africa. *Deep-Sea Research* 33:1327-1343.
- Wiens, J.A. 1989. Spatial scaling in ecology. *Functional Ecology* 3:385-397.
- With, K.A., R.H. Gardner and M.G. Turner. 1997. Landscape connectivity and population distributions in heterogeneous environments. *Oikos* 78:151-169.
- Wolfram, S. 1986. *Theory and Application of Cellular Automata*. World Scientific, Singapore.
- Wroblewski, J.S. and J.J. O'Brien. 1976. A spatial model of phytoplankton patchiness. *Marine Biology* 35:161-175.
- Yamazaki, H., and T.R. Osborn. 1988. Review of oceanic turbulence: implications for biodynamics. Pages 215-234 in B.J. Rothschild (ed). *Toward a theory on biological-physical interactions in the world ocean*. Kluwer Academic Publishers, Dordrecht.
- Yamazaki, H., T.R. Osborn and K. D. Squires. 1991. Direct numerical simulation of planktonic contact in turbulent flow. *Journal Plankton Research* 13:629-643.
- Yeager, C.L.J., L.W. Harding and M.E. Mallonee. 2005. Phytoplankton production, biomass and community structure following a summer nutrient pulse in Chesapeake Bay. *Aquatic Ecology* 32: 135-149.