TECHNICAL RESEARCH REPORT

Software and Other Teaching Tools Applied to Modeling and Analysis of Distributed Parameter Systems

by Raymond A. Adomaitis

T.R. 97-78

Sponsored by
the National Science Foundation
Engineering Research Center Program,
the University of Maryland,
Harvard University,
and Industry

Software and Other Teaching Tools Applied to Modeling and Analysis of Distributed Parameter Systems

Raymond A. Adomaitis

Department of Chemical Engineering and Institute for Systems Research
University of Maryland
College Park, MD 20742

From the Proc. Mid-Atlantic Sec. ASEE Fall Regional Conf, 1997

Abstract: Over the last two years, we have been developing a library of Matlab subprograms integrated with on-line lecture notes in the form of WWW documents which are used in both undergraduate and graduate-level Chemical Engineering Applied Mathematics classes. The goal has been to make as transparent as possible the relationships between model development, solution, and analysis of systems described by partial differential equation models. This paper presents the results of our initial efforts to create computational modules which have a one-to-one correspondence with each step of implementing eigenfunction expansion, Galerkin's, and other weighted residual methods. Examples representing heat transfer in a cylinder, and gas flow and heat transfer in a chemical vapor deposition reactor are presented.

1. Motivation

The motivation for developing a library of computational techniques to simplify the numerical solution of distributed parameter system problems grew out of the desire to work industrially-relevant problems in the classroom and to unify some of the computational techniques used in our research. Our approach evolved from experiences with students that were proficient at using state-of-the-art simulators to compute what *appeared* to be solutions to fluid flow and other transport problems, but were unable to answer basic questions about the solutions, such as whether it was a true, converged solution that satisfied the original modeling equations, or to derive useful information from the solution, such as rates of heat transfer or surface reaction rates.

The advent of powerful numerical and symbolic manipulation languages, such as Matlab[1], makes it possible to implement sophisticated numerical techniques without getting lost in programming details demanded by Fortran or other, traditional programming languages. While Matlab retains the advantages of the latter, and so facilitates open-ended interpretation of simulation results, its ability to perform array operations, flexibility in passing function arguments, and library of mathematical and graphics functions allow students to concentrate on problem solving. Given this flexibility, our goal was to develop a set of software tools and routines for implementing the method of weighted residuals (MWR), rather than produce a self-contained package for solving partial differential equation models. Our preliminary results are presented in this paper.

[1] http://www.mathworks.com

2. Documentation

In our approach to the numerical solution of distributed parameter system problems, the user must have an understanding of the particular MWR numerical solution procedure they wish to use. Thus, the solution procedure intricacy relative to "smarter" packages, where these details are performed automatically, means the documentation for our tools must be more extensive. This problem was solved by embedding the software subroutines in the on-line class lecture notes. A Unix script was written to perform this integration, and so maintenance of the programs and documentation is simplified. Lecture notes are maintained in LaTex and the Unix script combines the Matlab routines

into the text in the proper locations to produce a postscript file when the entire LaTex file was compiled. This document, plus the library of Matlab functions, is then automatically copied to a directory accessible by students through a web browser. The LaTex file is also processed with the LaTex to html translator (latex2html[2]) to create a set of web pages of the lecture notes[3] that have direct links to the relevant Matlab files – students simply click on the automatically generated links to download the Matlab functions and scripts.

3. The MWR Library

The current library of functions for simplifying implementation of weighted residual method numerical techniques contains routines for defining the problem domain and the trial functions used, general routines to simplify discretizing problems, including collocation methods, and utilities specially modified for numerically integrating and visualizing the results. A sample of the programs available is given below.

physical_domain.m Sets up the physical domain and defines the differential operators according to the domain and any imposed problem symmetries – normally, the first routine called.

gdf.m Generates discretized representations of a sequence of trial functions. These can be in the form of eigenfunctions generated by a Sturm-Liouville problem or an arbitrary sequence of functions chosen as part of a Galerkin discretization.

gs.m A Gram-Schmidt orthogonalization procedure.

wip.m The weighted inner product of two sets of functions – one of the most heavily used subprograms, since inner products are used in nearly every MWR step.

projection.m Projects a function or a set of functions onto a set of trial functions.

[2] http://www-dsed.llnl.gov/files/programs/unix/latex2html/manual/manual.html

[3] http://www.isr.umd.edu/~adomaiti/ench453/lectures/lectures.html

rdf.m Computes the roots of a discretized function – used in nonlinear eigenvalue problems and in the collocation subroutines.

colpts.m Computes collocation points at the roots of the highest-frequency trial function as part of an orthogonal collocation procedure.

colmat.m Generates the interpolation, discretization, and quadrature weight arrays for orthogonal collocation discretizations.

odae12.m An ordinary differential plus algebraic equation integrator.

4. Heat Transfer in a Cylinder

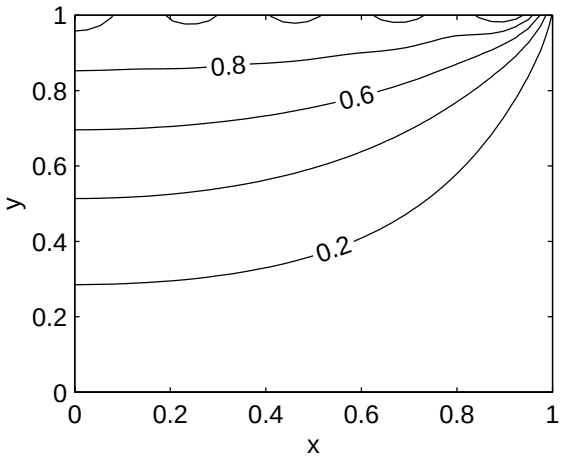


Figure 1: *Cylinder temperature profile computed from the MWR subroutines.*

As an example demonstrating how the MWR library is used, consider the problem of steady-state heat transfer inside a cylinder which also transfers energy to/from the surrounding environment. Inside the cylinder, we have the heat equation

$$\nabla^2 T(r,z) = 0$$

subject to boundary conditions $\partial T/\partial r = -10T$ at $r = 1$, symmetry at $r = 0$, $T = 0$ at $z = 0$, and $T = 1$ at $z = 1$. This problem is an excellent example of apparently inconsistent boundary conditions (notice the corner where $T = 1$ at $z = 1$ and $\partial T/\partial r = -10T$ at $r = 1$ meet) and so demonstrates how different solution procedures overcome this potential problem. We can write explicitly the

separation of variables solution as

$$T(r,z) = \sum_{n=1}^{\infty} a_n \sinh \lambda_n z J_0(\lambda_n r)$$

with

$$a_n = \frac{\int_0^1 J_0(\lambda_n r)\, r dr}{\sinh \lambda_n \int_0^1 J_0^2(\lambda_n r)\, r dr}$$

with the eigenvalues computed from the nonlinear equation $\lambda_n : \lambda J_1(\lambda) - 10 J_0(\lambda) = 0$.

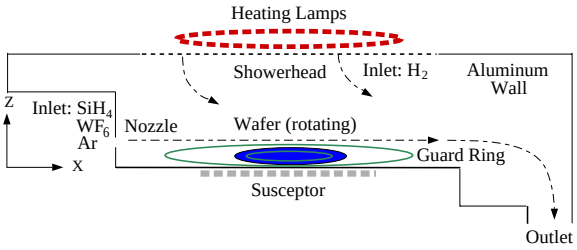
The steps for solving this problem with a Galerkin technique can be written as follows – we include references to the Matlab MWR routines used to implement each step.

1. We must first define the physical domain over which the modeling partial differential equation is defined (`physical_domain.m`);
2. The cylinder temperature must be defined by two sets of trial functions: the first η_m to account for the nonhomogeneous boundary condition at $z = 1$, and the second $\psi_{m,n}$ is used to minimize the residual inside the physical domain. The function `gdf.m` is used for both, with trial functions based on a combination of Bessel's functions in the radial direction, and polynomial or trigonometric functions in the axial direction. Both can be converted to orthonormal sets with `gs.m`;
3. The mode amplitude coefficients a_m of η_m are computed directly by projecting $T = 1$ onto these trial functions with `projection.m`. Alternatively, they can be computed one coefficient at a time with `wip.m`;
4. As the final step, the entire trial function expansion is substituted into the heat equation to form a residual. This residual can be projected onto the $\psi_{m,n}$ to compute the corresponding mode amplitude coefficients directly with `projection.m` as part of a Galerkin technique; a least-squares solution procedure to find the $b_{m,n}$ can be implemented using the differentiation tensors generated by `physical_domain.m`.

These steps give the final form of the cylinder temperature solution as

$$T(r,z) = \sum_{m=1}^{M} a_m \eta_m(r,z) + \sum_{m,n=1}^{M,N} b_n \psi_{m,n}(r,z).$$

(a) Side View



(b) Top View

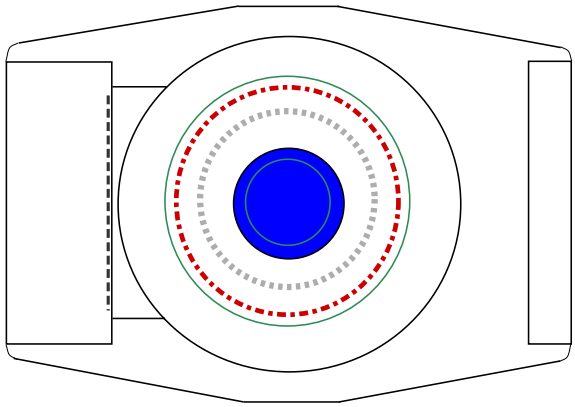


Figure 2: *Geometry of the tungsten chemical vapor deposition reactor system.*

Representative results are shown in Fig. 1.

5. A Chemical Vapor Deposition Problem

The development of the MWR library was partly motivated by the desire to introduce industrially-relevant distributed parameter system modeling problems in the classroom. As one example of such a system, consider the selective tungsten chemical vapor deposition (CVD) system shown in Fig. 2. This process is one unit operation in manufacturing the microelectronic devices of integrated circuits.

The geometry of the this CVD reactor system is such that reactant gases are introduced into the reactor from two sources: a gas mixture of silane and tungsten hexafluoride is injected through a thin nozzle slot on the reactor chamber side wall, and hydrogen is pumped in through a shower head at the chamber top. Gases mix in the chamber and react at the surface of a 4 inch diameter wafer, which is held by the edge on the slowly rotating (4 rpm) quartz susceptor by a quartz

guard ring. The wafer is heated to 300 C by a ring of incoherent tungsten-halogen lamps through transparent shower head. The CVD runs last for approximately 5 minutes after the operating temperature is reached.

6. CVD Reactor Model Solution

An important question that can be answered by modeling and simulation of this system is how heat transfer to the gas phase from the wafer undergoing processing compares to the radiant heating rates. The solution procedure is similar in structure to the cylinder heat transfer example: the modeling equations for the gas flow field and temperature, with the associated boundary conditions, are defined first. The steady, fully-developed gas flow field is computed with a Galerkin expansion [1], and the result is included in a 2-dimensional approximation to the gas temperature field defined along the center axis of the CVD reactor. As in the cylinder problem, the temperature field is approximated by two trial function series: the first is fit to the nonhomogeneous boundary condition resulting from the heated wafer on the reactor bottom surface. The entire trial function expansion is, again, substituted into the gas temperature modeling partial differential equation. After projecting the resulting nonhomogeneous terms onto the second trial (eigen)function set, the remaining linear problem is solved to compute the second set of mode amplitude coefficients.

The results for the steady gas temperature field for representative operating conditions are shown in Fig. 3. The rate of heat transfer from the wafer to the gas phase is found to be approximately 40 W/m^2, which is small relative to the radiant heating rates, which can be on the order of 100,000 W/m^2. These quantities are easy to compute, since the numerical techniques for computing derivatives (normal to the wafer surface in this case) of the gas temperature field necessary for flux calculations are already in place. Additional information regarding these applications and the underlying numerical techniques can be found in [2, 3].

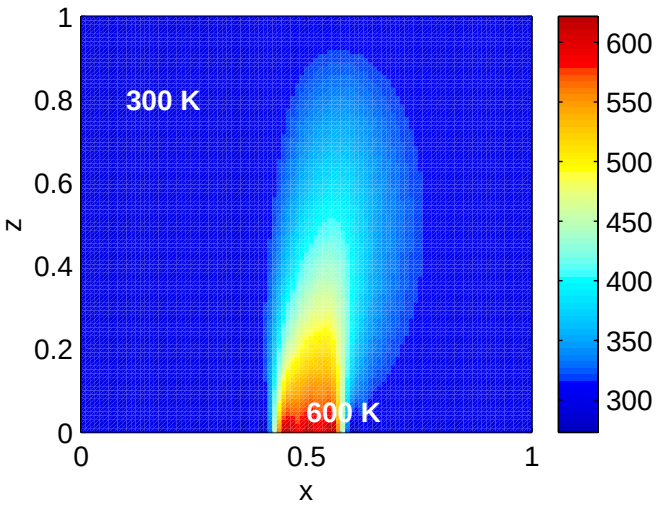


Figure 3: *Computed chemical vapor deposition system gas temperature profile showing the heated gas plume.*

References

[1] MacCluer, C. R. 1994, *Boundary Value Problems and Orthogonal Expansions* IEEE Press, New York.

[2] Chang, H.-Y., Y.-h. Lin, and R. A. Adomaitis 1997, Statistically-determined trial functions for RTCVD model reduction and simulation. AIChE Annual Meeting, Los Angeles, CA.

[3] Adomaitis, R. A. 1997, An orthogonal collocation technique for rapid thermal processing system discretization. *AIChE J.*, submitted; **also** *ISR TR 97-63.*