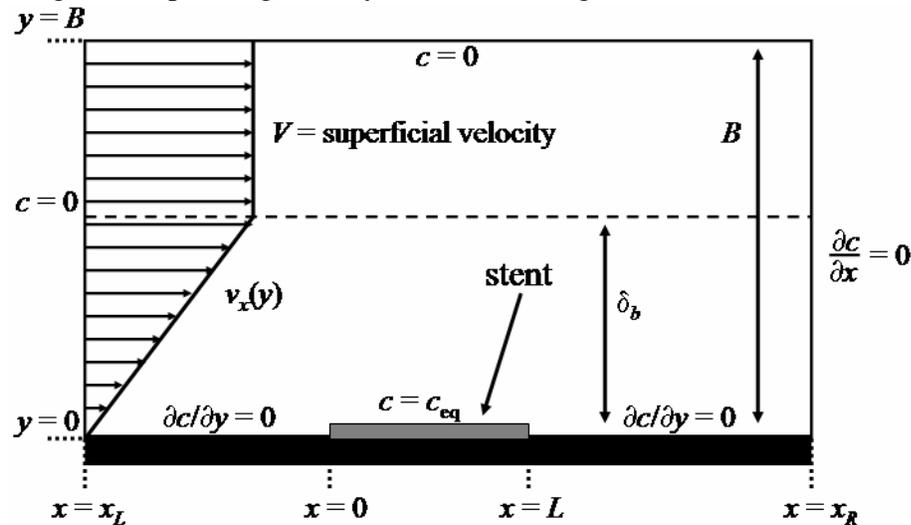


Fall 2005 10.34 Exam II. Wed. Nov. 9, 2005

(SOLUTION)

Read through the entire exam before beginning work, and budget your time.

You are researching drug eluting stents, and wish to understand better how the drug release rate depends upon the properties of the fluid in which it is immersed. You develop a model using the simplified geometry shown in the figure below.



The stent is placed on a solid impermeable surface, and to simplify the flow problem, we assume that the drug stent is very thin and does not affect the local nature of the flow. Also, we assume that the stent is very wide in the out-of-plane direction so that we can neglect variation in the \$z\$ direction. The local velocity of the fluid goes from zero at the solid surface (no-slip BC) to the superficial velocity \$V\$ at a distance \$\delta_b\$ from the surface. We assume a linear velocity profile,

$$v_x(y) = \begin{cases} \frac{Vy}{\delta_b}, & 0 \leq y \leq \delta_b \\ V, & y > \delta_b \end{cases} \quad (\text{EQ 1})$$

Given this velocity profile, we wish to model the external mass transfer resistance to release of drug from the stent. We solve the 2-D convection/diffusion equation

$$\frac{\partial c}{\partial t} = -v_x \frac{\partial c}{\partial x} + D \frac{\partial^2 c}{\partial x^2} + D \frac{\partial^2 c}{\partial y^2} \quad (\text{EQ 2})$$

on the domain \$x_L \leq x \leq x_R\$ and \$0 \leq y \leq B\$ to obtain \$c(x, y)\$ at steady state.

B is a distance sufficiently far away from the surface that we can assume that the drug concentration is zero there, $c(x, B) = 0$. At the left “inlet” boundary $x = x_L$, we use the BC $c(x_L, y) = 0$. At the right “outlet” boundary, we use the BC $\partial c / \partial x|_{(x_R, y)} = 0$. At the solid surface $y = 0$, the stent occupies the region $0 \leq x \leq L$ and the remaining regions are impermeable to the drug. At the stent surface, we use the boundary condition $c(0 \leq x \leq L, 0) = c_{\text{eq}}$, where c_{eq} is the drug concentration in the fluid that is in equilibrium with the stent. At all other regions on the impermeable surface, we use the no-flux BC $\partial c / \partial y|_{(x, 0)} = 0$.

The parameters for this problem are $V, \delta_b, D, c_{\text{eq}}, L, x_L, x_R, B$.

NOTE:

This problem was inspired by an actual numerical study performed by a student here at MIT. In that case, she was looking at the spatial distribution of drug release into the underlying tissue by solving a reaction/diffusion equation in the underlying surface underneath the stent to model drug uptake. She also solved, coupled to the convection/diffusion equation in the fluid stream, the Navier-Stokes equations to compute the exact laminar velocity profile around the stent. In practice, the drug release into the underlying tissue is not spatially uniform, but is concentrated just below and downstream of the stent. Here, I have simplified the problem by making the underlying surface impermeable to the drug and by neglecting any perturbation in the flow field by the stent.

You wish to compute the steady state concentration profile $c(x, y)$ using finite differences. You wish your solution to approximate the case of an isolated stent on an infinitely large solid surface in contact with an infinitely large body of fluid, with specified values of the superficial velocity V , boundary layer thickness δ_b , drug equilibrium concentration c_{eq} , stent size L , and drug diffusivity D .

(1.a) (5 pts) Describe how would you choose the remaining parameter values in a set of one or more finite difference calculations to approximate the solution in this limiting case.

ANSWER:

To simulate a single stent on an infinitely large surface we want $x_L \rightarrow \infty$ and $x_R \rightarrow \infty$. To simulate contact with an infinitely large body of fluid, we want $B \rightarrow \infty$. But, for our finite difference calculations, all of these values must be finite. So, we start with moderately large values of these distances, and increase them until they no longer seem to affect the local behavior of the solution near the stent.

(1.b) (5 pts) How would you determine a sufficient number of grid points to use in your calculations?

ANSWER:

We only obtain the exact solution, for fixed x_L, x_R, B , in the limit where the grid point spacing goes to zero. But again, in practice we can only treat a finite number N_x or grid points in the x direction and a finite number N_y in the y direction. Again, we must increase these until the local solution near the stent no longer appears to change significantly with an increase in the number of grid points.

Also, as an optional additional factor that might influence our decision, we see that this is a convection/diffusion problem, and so we should use upwind differencing to avoid the unphysical oscillations that occur when the grid point spacing is too large. But, these differences are less accurate than central differences. So, we might want to choose N_x large enough that the local Peclet number $V(\Delta x)/L$ is less than two, so that we could safely use the more accurate central finite difference approximation.

(2.a) (30 pts) For specified values of all parameters $V, \delta_b, D, c_{eq}, L, x_L, x_R, B$ and a specified grid of $N_x \times N_y$ grid points, derive the set of algebraic equations that you would solve for the concentration values at the grid points $c(x_i, y_j)$.

DO NOT ASSUME UNIFORM GRID SPACING, as it is probably much more efficient to use a narrower grid point spacing near the stent than far away from it. Assume that you already have specified the grid coordinates,

$$x_L < x_1 < x_2 < \dots < x_{N_x} < x_R \quad 0 < y_1 < y_2 < \dots < y_{N_y} < b \quad (\text{EQ 3})$$

Derive first the general equation that is satisfied at all “interior” points $1 < i < N_x$ and $1 < j < N_y$. Then for each grid point next to a boundary, show how you modify this equation to enforce the appropriate boundary condition.

When writing your equations, for convenience retain using the indices (i, j) to identify each grid point $c(x_i, y_j)$.

ANSWER:

At each grid point (x_i, y_j) , we want the PDE to be satisfied locally,

$$0 = -[v_x(y_j)] \frac{\partial c}{\partial x} \Big|_{(x_i, y_j)} + D \frac{\partial^2 c}{\partial x^2} \Big|_{(x_i, y_j)} + D \frac{\partial^2 c}{\partial y^2} \Big|_{(x_i, y_j)}$$

To avoid any chance of unphysical oscillations when the grid point spacing is too large, we use upwind differences for the convection term,

$$\left. \frac{\partial c}{\partial x} \right|_{(x_i, y_j)} \approx \frac{c(x_i, y_j) - c(x_{i-1}, y_j)}{x_i - x_{i-1}}$$

However, we could (optionally) use the more-accurate central finite difference approximation, but only if we explicitly check that the local Peclet number is less than two.

For the second derivative in the x direction, we first define the mid-point locations $x_{i \pm 1/2} = \frac{1}{2}(x_i + x_{i \pm 1})$, such that a finite difference approximation of the outer derivative yields

$$\left. \frac{\partial^2 c}{\partial x^2} \right|_{(x_i, y_j)} \approx \frac{\left. \frac{\partial c}{\partial x} \right|_{(x_{i+1/2}, y_j)} - \left. \frac{\partial c}{\partial x} \right|_{(x_{i-1/2}, y_j)}}{x_{i+1/2} - x_{i-1/2}}$$

We next use central finite difference approximations for each first derivative,

$$\left. \frac{\partial c}{\partial x} \right|_{(x_{i+1/2}, y_j)} \approx \frac{c(x_{i+1}, y_j) - c(x_i, y_j)}{x_{i+1} - x_i}$$

$$\left. \frac{\partial c}{\partial x} \right|_{(x_{i-1/2}, y_j)} \approx \frac{c(x_i, y_j) - c(x_{i-1}, y_j)}{x_i - x_{i-1}}$$

to obtain the approximation for the second derivative in x,

$$\left. \frac{\partial^2 c}{\partial x^2} \right|_{(x_i, y_j)} \approx \frac{\left[\frac{c(x_{i+1}, y_j) - c(x_i, y_j)}{x_{i+1} - x_i} \right] - \left[\frac{c(x_i, y_j) - c(x_{i-1}, y_j)}{x_i - x_{i-1}} \right]}{x_{i+1/2} - x_{i-1/2}}$$

Collecting terms, we have

$$\left. \frac{\partial^2 c}{\partial x^2} \right|_{(x_i, y_j)} \approx a_{x, lo}^{(i)} c(x_{i-1}, y_j) + a_{x, mid}^{(i)} c(x_i, y_j) + a_{x, hi}^{(i)} c(x_{i+1}, y_j)$$

where

$$\begin{aligned}
a_{x, \text{lo}}^{(i)} &= (x_i - x_{i-1})^{-1} (x_{i+1/2} - x_{i-1/2})^{-1} \\
a_{x, \text{hi}}^{(i)} &= (x_{i+1} - x_i)^{-1} (x_{i+1/2} - x_{i-1/2})^{-1} \\
a_{x, \text{mid}}^{(i)} &= -(a_{x, \text{lo}}^{(i)} + a_{x, \text{hi}}^{(i)})
\end{aligned}$$

We do exactly the same thing for the second derivative in y to obtain

$$\left. \frac{\partial^2 c}{\partial y^2} \right|_{(x_i, y_j)} \approx a_{y, \text{lo}}^{(j)} c(x_i, y_{j-1}) + a_{y, \text{mid}}^{(j)} c(x_i, y_j) + a_{y, \text{hi}}^{(j)} c(x_i, y_{j+1})$$

where with $y_{j\pm 1/2} = \frac{1}{2}(y_j + y_{j\pm 1})$, we have

$$\begin{aligned}
a_{y, \text{lo}}^{(j)} &= (y_j - y_{j-1})^{-1} (y_{j+1/2} - y_{j-1/2})^{-1} \\
a_{y, \text{hi}}^{(j)} &= (y_{j+1} - y_j)^{-1} (y_{j+1/2} - y_{j-1/2})^{-1} \\
a_{y, \text{mid}}^{(j)} &= -(a_{y, \text{lo}}^{(j)} + a_{y, \text{hi}}^{(j)})
\end{aligned}$$

Thus, at each interior point $1 < i < N_x$, $1 < j < N_y$, the local PDE

$$0 = -[v_x(y_j)] \left. \frac{\partial c}{\partial x} \right|_{(x_i, y_j)} + D \left. \frac{\partial^2 c}{\partial x^2} \right|_{(x_i, y_j)} + D \left. \frac{\partial^2 c}{\partial y^2} \right|_{(x_i, y_j)}$$

yields the linear algebraic equation

$$\begin{aligned}
0 = -[v_x(y_j)] &\left[\frac{c_{i,j} - c_{i-1,j}}{x_i - x_{i-1}} \right] + D[a_{x, \text{lo}}^{(i)} c_{i-1,j} + a_{x, \text{mid}}^{(i)} c_{i,j} + a_{x, \text{hi}}^{(i)} c_{i+1,j}] \\
&+ D[a_{y, \text{lo}}^{(j)} c_{i,j-1} + a_{y, \text{mid}}^{(j)} c_{i,j} + a_{y, \text{hi}}^{(j)} c_{i,j+1}]
\end{aligned}$$

where $c_{i,j} \equiv c(x_i, y_j)$. Collecting terms, we have

$$0 = A_{x, \text{lo}}^{(i,j)} c_{i-1,j} + A_{y, \text{lo}}^{(i,j)} c_{i,j-1} + A_{\text{mid}}^{(i,j)} c_{i,j} + A_{y, \text{hi}}^{(i,j)} c_{i,j+1} + A_{x, \text{hi}}^{(i,j)} c_{i+1,j}$$

with

$$A_{x, \text{lo}}^{(i,j)} = \left[\frac{v_x(y_j)}{x_i - x_{i-1}} \right] + Da_{x, \text{lo}}^{(i)} \quad A_{y, \text{lo}}^{(i,j)} = Da_{y, \text{lo}}^{(j)}$$

$$A_{\text{mid}}^{(i,j)} = - \left[\frac{v_x(y_j)}{x_i - x_{i-1}} \right] + D[a_{x, \text{mid}}^{(i)} + a_{y, \text{mid}}^{(j)}]$$

$$A_{y, \text{hi}}^{(i,j)} = Da_{y, \text{hi}}^{(j)} \quad A_{x, \text{hi}}^{(i,j)} = Da_{x, \text{hi}}^{(i)}$$

For the grid points that neighbor a boundary, we must modify these equations slightly, as the equations call for the value of a concentration at a point outside of our grid.

Inlet BC at $x = 0$:

At all grid points near the inlet, $i = 1$, we have

$$0 = A_{x, \text{lo}}^{(1,j)} c_{0,j} + A_{y, \text{lo}}^{(1,j)} c_{1,j-1} + A_{\text{mid}}^{(1,j)} c_{1,j} + A_{y, \text{hi}}^{(1,j)} c_{1,j+1} + A_{x, \text{hi}}^{(1,j)} c_{2,j}$$

with the “hypothetical” grid point $x_0 = x_L$. Here, we enforce the Dirichlet BC, $c_{0,j} = c(x_L, y_j) = 0$, so that our modified equations enforcing the inlet BC are

$$0 = A_{y, \text{lo}}^{(1,j)} c_{1,j-1} + A_{\text{mid}}^{(1,j)} c_{1,j} + A_{y, \text{hi}}^{(1,j)} c_{1,j+1} + A_{x, \text{hi}}^{(1,j)} c_{2,j}$$

Bulk BC at $y = B$:

Here, our grid points are $j = N_y$, for which we have

$$0 = A_{x, \text{lo}}^{(i, N_y)} c_{i-1, N_y} + A_{y, \text{lo}}^{(i, N_y)} c_{i, N_y-1} + A_{\text{mid}}^{(i, N_y)} c_{i, N_y} + A_{y, \text{hi}}^{(i, N_y)} c_{i, N_y+1} + A_{x, \text{hi}}^{(i, N_y)} c_{i+1, N_y}$$

We have our hypothetical grid point at $y_{N_y+1} = B$, and enforce the BC by setting $c_{i, N_y+1} = c(x_i, B) = 0$, so that our modified equations are

$$0 = A_{x, \text{lo}}^{(i, N_y)} c_{i-1, N_y} + A_{y, \text{lo}}^{(i, N_y)} c_{i, N_y-1} + A_{\text{mid}}^{(i, N_y)} c_{i, N_y} + A_{x, \text{hi}}^{(i, N_y)} c_{i+1, N_y}$$

Solid surface at $y = 0$:

Here, at $j = 1$, we must modify our equations

$$0 = A_{x, \text{lo}}^{(i, 1)} c_{i-1, 1} + A_{y, \text{lo}}^{(i, 1)} c_{i, 0} + A_{\text{mid}}^{(i, 1)} c_{i, 1} + A_{y, \text{hi}}^{(i, 1)} c_{i, 2} + A_{x, \text{hi}}^{(i, 1)} c_{i+1, 1}$$

to account for the hypothetical grid point at $y_0 = 0$. If $0 \leq x_j \leq L$, we use the Dirichlet BC $c_{i, 0} = c_{\text{eq}}$, such that our modified equations are

$$0 = A_{x, \text{lo}}^{(i, 1)} c_{i-1, 1} + A_{y, \text{lo}}^{(i, 1)} c_{\text{eq}} + A_{\text{mid}}^{(i, 1)} c_{i, 1} + A_{y, \text{hi}}^{(i, 1)} c_{i, 2} + A_{x, \text{hi}}^{(i, 1)} c_{i+1, 1}$$

Else, our grid point is next to the part of the surface that is impermeable and we enforce the BC $\partial c / \partial y|_{(x_i, 0)} = 0$. Using Lagrange interpolation, we approximate $c(x_i, y)$ and its derivative near $y = 0$ as

$$\begin{aligned} c(x_i, y) &= c_{i, 0} L_0(y) + c_{i, 1} L_1(y) + c_{i, 2} L_2(y) \\ \left. \frac{\partial c}{\partial y} \right|_{(x_i, y)} &= c_{i, 0} L_0'(y) + c_{i, 1} L_1'(y) + c_{i, 2} L_2'(y) \end{aligned}$$

where the Lagrange polynomials and their derivatives are

$$L_j(y) = \prod_{\substack{k=0 \\ k \neq j}}^2 \frac{(y - y_k)}{(y_j - y_k)} \quad L_j'(y) = \sum_{\substack{k=0 \\ k \neq j}}^2 \frac{1}{(y_j - y_k)} \prod_{\substack{m=0 \\ m \neq j \\ m \neq k}}^2 \frac{(y - y_k)}{(y_j - y_k)}$$

Thus, from our no-flux BC we have

$$0 = c_{i, 0} L_0'(0) + c_{i, 1} L_1'(0) + c_{i, 2} L_2'(0)$$

so that the necessary modification to our equations is to make the substitution

$$c_{i, 0} = \left[\frac{L_1'(0)}{L_0'(0)} \right] c_{i, 1} + \left[\frac{L_2'(0)}{L_0'(0)} \right] c_{i, 2}$$

Outlet BC at $x = x_R$:

Finally, we consider the points at $i = N_x$ where we have

$$0 = A_{x, \text{lo}}^{(N_x, j)} c_{N_x-1, j} + A_{y, \text{lo}}^{(N_x, j)} c_{N_x, j-1} + A_{\text{mid}}^{(N_x, j)} c_{N_x, j} + A_{y, \text{hi}}^{(N_x, j)} c_{N_x, j+1} + A_{x, \text{hi}}^{(N_x, j)} c_{N_x+1, j}$$

Here, we remove the dependence on the hypothetical grid value $c_{N_x+1,j} = c(x_R, y_j)$ by enforcing the outlet BC $\partial c / \partial x|_{(x_R, y_j)} = 0$. To do so, we again use Lagrange interpolation,

$$c(x, y_j) = c_{N_x+1,j} L_{N_x+1}(x) + c_{N_x,j} L_{N_x}(x) + c_{N_x-1,j} L_{N_x-1}(x)$$

$$\left. \frac{\partial c}{\partial x} \right|_{(x, y_j)} = c_{N_x+1,j} L_{N_x+1}'(x) + c_{N_x,j} L_{N_x}'(x) + c_{N_x-1,j} L_{N_x-1}'(x)$$

where like before,

$$L_j(x) = \prod_{\substack{k=N_x+1 \\ k \neq j}}^{N_x-1} \frac{(x-x_k)}{(x_j-x_k)} \quad L_j'(x) = \sum_{\substack{k=N_x+1 \\ k \neq j}}^{N_x-1} \frac{1}{(x_j-x_k)} \prod_{\substack{m=N_x+1 \\ m \neq j \\ m \neq k}}^{N_x-1} \frac{(x-x_k)}{(x_j-x_k)}$$

Thus, we enforce the BC by making the substitution

$$c_{N_x+1,j} = \left[\frac{L_{N_x}'(x_R)}{L_{N_x+1}'(x_R)} \right] c_{N_x,j} + \left[\frac{L_{N_x-1}'(x_R)}{L_{N_x+1}'(x_R)} \right] c_{N_x-1,j}$$

Finally, we will have to account for the fact that at each of the four corners, we have to make two of the modifications above. This completes the derivation of the set of linear algebraic equations that we solve to obtain the steady-state concentrations at each grid point.

(2.b) (5 pts) What numerical issues were you considering when you chose the particular finite difference approximations that you used?

ANSWER:

When discretizing the first derivative in the convection term, upwind one-sided differences were used to avoid any spurious oscillations. When discretizing the first derivative in the boundary conditions, Lagrange interpolation was used to provide just as high accuracy as is used when discretizing the second derivatives.

Once you have derived these equations, you must solve them in MATLAB.

(3.a) (5 pts) Explain how you would do this, and describe your procedure for putting the problem into a standardized format that one of the MATLAB solvers knows how to handle.

ANSWER:

These are linear algebraic equations, and so I would put them in the form $A\phi = \underline{b}$, using a labeling scheme that assigns the unique integer $k \in [1, N_x N_y]$ to each unknown $c(x_i, y_j)$ by the rule $k = (i-1)N_y + j$ (other rules are acceptable as long as they are regular and assign labels uniquely). Above, I have derived the values of the matrix coefficients and the only non-zero components of \underline{b} arise from the Dirichlet BC at the stent surface.

(3.b) (10 pts) How would the total computational effort require to solve the problem scale as powers of N_x and N_y ?

ANSWER:

Based on the rule above, I see from the general form of the equations,

$$0 = A_{x, lo}^{(i,j)} c_{i-1,j} + A_{y, lo}^{(i,j)} c_{i,j-1} + A_{mid}^{(i,j)} c_{i,j} + A_{y, hi}^{(i,j)} c_{i,j+1} + A_{x, hi}^{(i,j)} c_{i+1,j}$$

that the linear system is sparse with only five non-zero elements per row. Using $\phi_k = c(x_i, y_j)$ with $k = (i-1)N_y + j$ to label the unknowns, this yields

$$0 = A_{x, lo}^{(i,j)} \phi_{k-N_y} + A_{y, lo}^{(i,j)} \phi_{k-1} + A_{mid}^{(i,j)} \phi_k + A_{y, hi}^{(i,j)} \phi_{k+1} + A_{x, hi}^{(i,j)} \phi_{k+N_y}$$

so that the matrix is banded with bandwidth $m = N_y$. When a matrix is banded, Gaussian elimination is much faster, with the computational time scaling as Nm^2 , where N is the number of unknowns and m is the bandwidth. Here, $N = N_x N_y$ and $m = N_y$, so that the total computational effort scales no more than $N_x N_y^3$. Note that if $N_y < N_x$, it would be advantageous to change the labeling scheme to make the bandwidth N_x .

Let us say that you have now computed the steady state concentration field $c(x, y)$ using the method outlined above.

(4) (15 pts) Explain in detail how you would compute from your numerical solution the net flux of drug out of the stent, in units of moles of drug per unit width in the z direction per unit time. "In detail" means set up the exact equation(s) that you would solve to obtain this value, and describe what MATLAB routine(s) you would use.

ANSWER:

The diffusive flux of drug out of the stent surface is $-D \frac{\partial c}{\partial y} \Big|_{(x, 0)}$, in units of moles of drug per unit area of stent per unit time. Integrating over x , we obtain the net flux out of the stent in units of moles of drug per unit width in the z direction per unit time,

$$\text{net flux} = \int_0^L \left[-D \frac{\partial c}{\partial y} \Big|_{(x, 0)} \right] dx$$

Using the same Lagrangian interpolation method as for the von Neumann BC at the solid surface, we can estimate the value of the derivative at each grid point $0 \leq x_j \leq L$ on the stent surface. We then obtain the net flux by integrating over these values with trapz.

Let's say that we now want to simulate a dynamic release, where initially the fluid contains no drug and then at time zero, the release from the drug begins until steady state is reached. Assume that you will have to do this calculation several times for different parameter values, and so computational time is a concern.

(5.a) (5 pts) What MATLAB routine would you use to perform the calculation, and why do you choose this particular one?

ANSWER:

I would use **ode15s**, as sets of coupled ODEs obtained by discretizing PDEs such as this one are quite stiff for any moderately large number of grid points.

To use this MATLAB routine, you have to supply additional routine(s) as well to provide information about the dynamics of your specific system.

(5.b) (10 pts) What information will you provide, and how do you obtain it from the derivations you performed for your steady-state calculation above?

ANSWER:

I would have to supply the time derivative vector, $f(t, \phi) = A\phi - b$, where A and b are the same quantities derived above. Also, as **ode15s** is implicit, I should supply the Jacobian matrix to avoid **ode15s** having to estimate it by costly finite difference approximations. Luckily, I already have the Jacobian matrix, which is just A .

Here we have neglected variation in the out-of-plane z direction by assuming infinite width of the stent in that direction.

(6) (10 pts) If we wanted to relax this approximation and model a stent of finite length and width, what new numerical issues would arise in computing the steady state solution?

ANSWER:

By relaxing this approximation, we have a 3-D boundary value problem. There is nothing conceptually new about how we discretize the diffusion equation or enforce the boundary conditions. What would change is the method used to solve the set of linear algebraic equations, since Gaussian elimination cannot be used with 3-D systems due to extreme fill-in problems. Therefore, I would have to use an iterative solver that does not have this problem, along the lines of the methods described in class (e.g. **gmres**).

(7) (5 pts) Your @\$% professor has just given you another exam that is far too long to complete in just 55 minutes. Scanning the test, you see that there are N problems, where each problem $j = 1, 2, \dots, N$ is worth P_j points. You estimate that problem j would take T_j minutes to finish completely, and you are confident you would get the right answer if you had enough time. Due to partial credit, you feel the score that you would receive on each question is proportional to the time allotted to it, so that if you spend $\frac{1}{2}T_j$ minutes on problem j , you receive $\frac{1}{2}P_j$ points. Express the problem of finding the best time-management strategy in terms of a constrained minimization that you can solve numerically.

ANSWER:

I want to set up this problem as a minimization of a cost function subject to some number of equality or inequality constraints. The set of variables that we vary are the times $\{t_1, t_2, \dots, t_N\}$ allocated to each problem. We want to maximize the score, so with the effect of partial credit, we minimize the cost function

$$F_c = - \sum_{j=1}^N P_j \left(\frac{t_j}{T_j} \right)$$

subject to the constraints that the times spent on each problem cannot exceed the total necessary time to solve it completely, and that the total amount of time spent on the exam cannot exceed 55 minutes,

$$\begin{aligned} 0 \leq t_j \leq T_j & \quad j = 1, 2, \dots, N \\ (t_1 + t_2 + \dots + t_N) \leq 55 \end{aligned}$$

This type of resource allocation problem is common in industry, often with the use of linear cost and constraint functions such as found here. In this instance, it is easy to see that the optimal strategy is to work through the problems in order of their decreasing rate of point generation, P_j/T_j ; however, the general linear programming approach outlined here is far more general and can be applied to many problems where the optimal strategy is not so evident.