ChE 505
Final Exam (100 points)
Advanced Mathematics for Engineers
Administered: 8:00-10:00 A.M. Saturday December 9, 2000

Problem 1. (50 points)

When we dealt with the numerical solution to parabolic PDEs, we split the study up into linear and nonlinear problems. With linear problems, we used a Crank-Nicholson method, which allowed us to calculate a Jacobian and perform one matrix inversion in order to solve the entire problem at all positions and times in one fell swoop. With nonlinear problems, we used a Runge-Kutta type method, where we did not require any linear algebra, but instead looped through time, advancing one time-step each pass through the loop.

When we dealt with ODEs, we ignored treating the linear problem individually and just used the Runge-Kutta type solution to solve both linear and nonlinear ODEs. However, an analogous linear method could have been used to solve ODEs. Outline the procedure to solve a linear ODE at all times with a single matrix inversion. Consider this single linear ODE:

$$\frac{dy}{dx} = ay + bx + c \tag{1}$$

$$y(x = x_o) = y_o \tag{2}$$

You are asked to solve over the range $x_o \le x \le x_f$ using n_x intervals.
Use the forward finite difference formula:

$$\frac{dy}{dx} \approx \frac{y_{j+1} - y_j}{\Delta x} \tag{3}$$

Use a second-order method, namely

$$\frac{y_{j+1} - y_j}{\Delta x} = \frac{1}{2}\left[K_j + K_{j+1}\right] \tag{4}$$

where K_j is the right hand side of equation (1) evaluated at (x_j, y_j).
In outlining your procedure,

(a) provide an algorithm.
(b) provide the generic equation, which you will end up solving.
(c) show the first and second row of the matrix
(d) comment on the structure and bandwidth of the matrix.
(e) point out any loops in the algorithm.
(f) comment on convergence.
(g) comment on stability. What can make the program crash?

Solution:

Substitute equation (1) into equation (4)

$$\frac{y_{j+1}-y_j}{\Delta x}=\frac{1}{2}\left[ay_j+bx_j+c+ay_{j+1}+bx_{j+1}+c\right]$$

The y variables at any j but j =0 are unknown. Put all knowns on right hand side and unknowns on left hand side.

$$y_{j+1}-\frac{\Delta x}{2}ay_{j+1}-\frac{\Delta x}{2}ay_j-y_j=\frac{\Delta x}{2}\left[bx_j+c+bx_{j+1}+c\right]$$

(b) $$\left(1-\frac{\Delta x}{2}a\right)y_{j+1}-\left(1+\frac{\Delta x}{2}a\right)y_j=\frac{\Delta x}{2}\left[b(x_j+x_{j+1})+2c\right] \qquad (5)$$

This equation has the unknown on the left hand side.

(a) The algorithm:

1. calculate $\Delta x=\frac{x_f-x_0}{n_x}$. $x_i=i\cdot\Delta x$ for i = 0 to n_x
2. Write equation (5), for j = 0 to n_x-1.
3. Solve the resulting matrix equation.

The matrix equation would look like:

$$\underline{\underline{J}}\,\underline{y}=\underline{R}$$

Let $J_d=\left(1-\frac{\Delta x}{2}a\right)$ $J_o=-\left(1+\frac{\Delta x}{2}a\right)$

(c)

$$\underline{\underline{J}}=\begin{bmatrix}J_d & 0 & 0 & 0\\ J_o & J_d & 0 & 0\\ 0 & \ddots & \ddots & 0\\ 0 & 0 & J_o & J_d\end{bmatrix} \qquad \underline{y}=\begin{bmatrix}y_1\\ y_2\\ \vdots\\ y_{n_x}\end{bmatrix} \qquad \underline{R}=\begin{bmatrix}\frac{\Delta x}{2}\left[b(x_0+x_1)+2c\right]-J_oy_0\\ \frac{\Delta x}{2}\left[b(x_1+x_2)+2c\right]\\ \vdots\\ \frac{\Delta x}{2}\left[b(x_{n_x-1}+x_{n_x})+2c\right]\end{bmatrix}$$

(d) comment on the structure and bandwidth of the matrix.

The matrix is banded with a width of two.

(e) point out any loops in the algorithm.

There are no loops in the algorithm. You just solve the system of equations once.

(f) comment on convergence.

There is no iterative loop for convergence so convergence is not an issue.

This method will work as long as the determinant of the matrix is non-zero, which it will always be due to the structure and origin of the matrix.

(g) comment on stability. What can make the program crash?

This method is stable regardless of time step, because the system is linear. However, inaccurate results will be obtained if large time steps are taken.

Problem 2. (50 points)

Consider an isothermal batch reactor containing monomer molecules. The monomers and polymers can react to form longer polymers of length i, P_i. For example, P_6, can be created by these reactions

$$P_5 + P_1 \rightarrow P_6, \qquad P_4 + P_2 \rightarrow P_6, \qquad \text{and} \qquad 2P_3 \rightarrow P_6$$

Also, P_6, can be consumed by these reactions

$$P_6 + P_1 \rightarrow P_7, \qquad P_6 + P_2 \rightarrow P_8, \qquad \text{and generally} \quad P_6 + P_{n-6} \rightarrow P_n$$

A mole balance can be written for the number of moles of a polymer of length i, P_i, for every value of i:

accumulation = in - out + generation - consumption

$$\frac{dP_i}{dt} = 0 - 0 + \sum_{j=1}^{i/2} k_f(i-j,j)P_{i-j}P_j - \sum_{n=i+1}^{\infty} k_f(i,n-i)P_iP_{n-i} \qquad (2.1)$$

where $k_f(i-j,j)$ is a reaction rate constant which is a function of the length of the two reactants, P_{i-j} and P_j.

Assume you have a smooth function that describes $k_f(i-j,j)$ as a function of its two arguments (i,j) from (1,1) to $(n_{max}-i,i)$ where is $n_{max} = 10^6$.

The reactor is initially filled with 100% monomer, P_1.

Using the model in equation (2.1), answer the following questions:

(a) What sort of equations do you have? (ODE, PDE, AE, IE)?
(b) Are the equations linear or nonlinear?
(c) How many equations do you have?
(d) What technique would you use to solve this problem?
(e) What problems would you expect to encounter using this technique?

Since the summation in the mole balance (2.1) is over a large range ($n_{max} = 10^6$), we could approximate the sum with an integral and assume that P_i is a continuous function over i. The mole balance becomes:

$$\frac{dP(x)}{dt} = \int_{y=1}^{x/2} k_f(x-y,y)P(x-y)P(y)dy - \int_{n=x}^{n_{max}} k_f(x,n-x)P(x)P(n-x)dn \qquad (2.2)$$

Answer questions (a) through (e) for equation (2.2).

Solution:

Using the model in equation (2.1), answer the following questions:

(a) What sort of equations do you have? (ODE, PDE, AE, IE)?

Ordinary differential equations

(b) Are the equations linear or nonlinear?

This issue of linearity is determined by the linearity of unknowns. The terms which contain $P_{i-j}P_j$ are nonlinear in the unknowns.

(c) How many equations do you have?

You can write this equation for i = 1 to $n_{max} = 10^6$

(d) What technique would you use to solve this problem?

You would use a method for integrating a system of nonlinear ODEs, like the Runge-Kutta method or the Euler method.

(e) What problems would you expect to encounter using this technique?

Since you have $n_{max} = 10^6$ ODEs, it's going to take a long time to solve them all. Plus all that information is not going to be useful. The only other problem one might expect is the same problem one always encounters with Runge-Kutta and Euler methods, namely that the code may crash as Δt becomes large.

Using the model in equation (2.2), answer the following questions:

(a) What sort of equations do you have? (ODE, PDE, AE, IE)?

Integro-differential equation.

(b) Are the equations linear or nonlinear?

nonlinear due to the product $P_{i-j}P_j$

(c) How many equations do you have?

1 equation. (for P(x,t)).

(d) What technique would you use to solve this problem?

I would discretize P along x. Then I would use a Runge-Kutta technique to integrate through time. At each Runge-Kutta evaluation, I would integrate the right-hand-side of equation (2.2) using a Trapezoidal rule.

(e) What problems would you expect to encounter using this technique?

Don't see any problems right off the bat. The advantage in solving the problem as a single integro-differential equation rather than a system of ODEs is that you have 10^6 ODEs but only 1 integro-differential equation (IDE). When you discretize the IDE (in order to evaluate the integrals using Trapezoidal rule), you won't need to discretize it into 10^6 intervals. Perhaps you will use only 10^2 or 10^3 intervals. Therefore, the calculation time will be much shorter.

There is no question of convergence in this problem, if you use a method like Runge-Kutta. You know P(x,t=0). Therefore, you can evaluate the integrals on the right hand side of equation (2.2). Once you know that, you can use RK to obtain $P(x,t_1)$. So there is no iterative convergence scheme required.