

Meteorology 5344, Fall 2005
Computational Fluid Dynamics
Instructor: Dr. Ming Xue

Homework #4: Numerical Solution of Heat Transfer Equation

Distributed Tuesday Oct. 11, 2005

Due Thursday Nov. 3, 2005

The goal of this computer problem is to examine the stability and accuracy of explicit and implicit schemes for solving a 1-D heat transfer equation.

1. The 1-D heat transfer equations is

$$\frac{\partial T}{\partial t} = \sigma \frac{\partial^2 T}{\partial x^2} \quad (1)$$

where σ is a positive and constant diffusivity. The problem consists of a 0.30 m thick wall of infinite lateral extent that is initially at a uniform temperature $T_{\text{initial}} = 100$ K. The surface temperatures at the two sides (T_{side}) are suddenly increased to and maintained at a temperature of 300 K. The wall is composed of a Nickel alloy that has a conductivity $\sigma = 3.0 \times 10^{-6} \text{ m}^2 \text{ s}^{-1}$. The analytic solution for the temperature distribution can be written

$$T(x, t) = T_{\text{side}} + 2(T_{\text{initial}} - T_{\text{side}}) \sum_{m=1}^{\infty} \left[e^{-(m\pi/L)^2 \sigma t} \frac{1 - (-1)^m}{m\pi} \sin\left(\frac{m\pi x}{L}\right) \right] \quad (2)$$

where $0 < x < L$. The size of m determines the accuracy of the solution.

a. Using the FTCS scheme, solve (1) out to time = 30 minutes and plot your solution for all x points every 5 minutes using $\Delta x = 0.015$ m and $\Delta t = 20$ and 60 sec. Explain your results.

Consider the following two questions and make sure you know the answers. You are not required to perform the actual computation, although you can do it as an exercise.

b. Using the exact solution given by (2), determine the convergence rate of the FTCS scheme for this particular problem by running a series of experiments in which Δx is successively refined over a wide range of values (the choices of which are up to you). Remember to vary Δt such that $\mu = \sigma \Delta t / (\Delta x)^2$ remains constant. Make plots of the RMS error versus resolution, and the convergence rate versus resolution. Discuss your results in light of the theoretical order of the FTCS scheme and other issues from lecture.

c. Choosing two "high resolution" solutions from part b, use Richardson extrapolation to determine a fourth-order solution. Then, compare this solution with that obtained from your code using $\mu = 1/6$, i.e., a direct fourth-order solution. How do the two compare? To what

degree is the Richardson extrapolation result a function of the grid size used for the two "high resolution" solutions. You may wish to rerun a few coarsened-grid cases to answer this last question.

2. For the same heat transfer equation given in problem 1:

- a. Using the Crank-Nicholson scheme, solve (1) out to $t = 30$ minutes and plot your solution for all x points every 5 minutes using $\Delta x = 0.015$ m and $\Delta t = 20$ sec. Employ the Thomas algorithm (see also Appendix A of your textbook) to solve the tridiagonal matrix.
- b. Compute additional solutions with successively larger timesteps (of your choice), and compare the plotted results with the analytic solution given in problem 1. At what point does the solution error become intolerably large, and is the scheme stable for all choices of Δt as predicted by linear theory?

Additional question for you to consider. Actual computation not required but you need to be sure you know the expected results.

- c. Choose your favorite solution (i.e., a given Δt and Δx) from your FTCS experiments in problem 1 and determine the CPU time and error relative to the analytic solution. Rerun this same case with your C-N code and compare the CPU time and error. Given that a larger timestep is possible in the C-N code, make two additional sets of runs:
 - 1). For the Δx used in part c above, increase Δt in the C-N run until the solution error deviates substantially (say 20%) from that of the FTCS experiment. How much savings (in terms of CPU time) did you get by using an implicit scheme? In other words, how much cheaper is the C-N scheme (at larger Δt) than the FTCS scheme (at smaller Δt) for equivalent solution accuracy and Δx ?
 - 2) Assuming that you can use no more CPU time than from your first experiment in part c, compute the best solution possible with the C-N scheme. That is, use the smallest Δx possible, coupled with the largest Δt , to get the "best" solution within the CPU time constraint given from your first experiment in part c.