**15. Stability analysis and computer experiment:**

Test the stability of the scheme from problem 13 for $d = 0$:

$$\frac{0.5T_j^{n-1} - 2T_j^n + 1.5T_j^{n+1}}{\Delta t} - \frac{\alpha(T_{j-1} - 2T_j + T_{j+1})^n}{\Delta x^2} = 0$$

with the von Neumann method. (Hint: This requires the solution of a quadratic equation in the amplification factor $g$. You can abbreviate $s\sin^2(k\Delta x/2)$ with $h$ to simplify the expression. Discuss separately the cases for which $g$ has real and complex solutions).

**16. Implementation of a different integration scheme:**

Modify the program sim1 to introduce the scheme from problem 13 in place of the FTCS (forward time, centered space) scheme:

For the very first time step use the formula $\partial T/\partial t \approx (T^{n+1} - T^n)/\Delta t$ with $d = 0$ (because $T^{n-1}$is not available at $t = 0$.)

**(a)** Obtain solutions with $\Delta x = 0.1$, 0.05, 0.025 for $s = 0.3$ using $d = 0$ and $d = 1 - \frac{1}{12s}$, determine the error, and compare with the solution produced by the FTCS scheme.

**(b)** Is the accuracy for decreasing $\Delta x$ consistent with your truncation error for problem 13? For $d = 1 - \frac{1}{12s}$ the scheme is supposedly 4th order accurate. Is the result from the simulation consistent with this expectation and comment/interpret on your result? (Hint: the error in the simulation is actually not 4th order - why may this be the case different from the theoretical prediction?).

**17. How fast is your computer?**

Modify and use the program count.f (on the website) to determine the execution time of basic operations (assign, add, subtract, multipl, div) and important mathematical library calls (sin, exp, sqrt) for your computer. You can time the program execution with the command 'time' on unix (linux) machines or run the program sufficiently long to measure the time with a stopwatch - provided there is no other activity on the computer with which the program has to share time. You can change the the loop dimension with the parameter n to adjust execution times to a reasonable physical time of a few to a few ten seconds. The comparison of the execution of different combinations of operations can provide additional consistency for the execution times of basic operations. The program contains specifically loops that are recursive (i.e., cannot be vectorized or parallelized) and array operations where multiple operations can be executed in a single cycle. Edit the operations within the loop as you see fit to determine the execution times. These depend on the type of variables and - for modern computers - also on the possibility of vectorization of parallelization. Repeat the timing tests with different optimization options to detect possible differences and the potential of faster execution for more aggressive optimization (note that results may change for more aggressive optimization). Generate a list for the results which should also include information on what type of computer/processor you run the program and which compiler (including options) you used. Interpret your results where you may compare specific execution time to the basic time of a single operation (entering results into a spreadsheet is a good idea for this comparison).

---

Please turn in the solutions to the homework on Monday, 3/4/2013