Programming Homework Assignment

Assigned Dec 31, 2011. Due Jan 13, 2012.

1. The objective of this homework is to solve numerically the solution of the heat equation

$$u_t(x,y,t) = \nu(u_{xx}(x,y,t) + u_{yy}(x,y,t)), \quad (x,y) \in (0,1)^2, t > 0 \tag{1}$$

with initial and boundary conditions

$$u(x,y,0) = \sin(\pi x)\sin(\pi y), \quad u(x,0,t) = u(x,1,t) = u(0,y,t) = u(1,y,t) = 0, \tag{2}$$

in order to get the solution $u(x,y,t)$ at $t = 1$. Here ν is a material dependent parameter. We take $\nu = 0.1$ in this problem.

The discrete approximation of $u(x,y,t)$ is defined on (x_i, y_j, t^n) and is denoted by $u^n_{i,j}$, where $x_i = i\Delta x$, $y_j = j\Delta y$, $t^n = n\Delta t$. For simplicity, we take $\Delta x = \Delta y = \Delta t = h = \frac{1}{N}$. The numerical solution is obtained through an updating process, from $\{u^n_{i,j}\}_{i,j=1,\cdots N-1}$ to $\{u^{n+1}_{i,j}\}_{i,j=1,\cdots N-1}$ until one reaches $n = N$, $t^n = t^N = 1$, using the following formula

$$\frac{u^{n+1}_{i,j} - u^n_{i,j}}{\Delta t} = \nu \cdot \frac{1}{2}\Big(\frac{u^{n+1}_{i+1,j} - 2u^{n+1}_{i,j} + u^{n+1}_{i-1,j}}{(\Delta x)^2} + \frac{u^{n+1}_{i,j+1} - 2u^{n+1}_{i,j} + u^{n+1}_{i,j-1}}{(\Delta y)^2} \\ \frac{u^n_{i+1,j} - 2u^n_{i,j} + u^n_{i-1,j}}{(\Delta x)^2} + \frac{u^n_{i,j+1} - 2u^n_{i,j} + u^n_{i,j-1}}{(\Delta y)^2} \Big) \tag{3}$$

One can verify easily that the exact solution for (1), (2) is $u^e(x,y,t) = e^{-2\pi^2\nu t}\sin(\pi x)\sin(\pi y)$. If all things are done correctly, you should see the error at $t = 1$ to scale like

$$\max_{i,j=1,\cdots N-1} |u^e(x_i, y_j, 1) - u^N_{i,j}| = O(\Delta x^2) + O(\Delta t^2) \quad (= O(N^{-2}) \text{ in this problem}) \tag{4}$$

when you change N from 100 to 200, or from 200 to 400, etc.

   (a) Start with the 1D case, where $u^e(x,t) = e^{-\pi^2\nu t}\sin(\pi x)$. Analyze possible approaches (ie. direct methods, iterative methods, etc) for solving the corresponding linear system and determine the quickest method and estimate the number of multiplication needed in terms of N. If you cannot do the 2D case, analyze, implement and hand in the 1D case instead for (minimal) partial credit.

   (b) Continue on the 2D case. Analyze possible methods you learned from this class and explain why you chose your method. Hand in your analysis and your test result for (4) and your code (An email account for handing in the code will be announced later).

   (c) (This one is a little harder and carries extra credit. Do it if time permits and be sure to consult me before and during implementation)

   Solve numerically for

$$u_{xx}(x,y) + u_{yy}(x,y) - u^3(x,y) = \sin(\pi x)\sin(\pi y), \quad (x,y) \in (0,1)^2.$$

with the boundary condition

$$u(x,0) = u(x,1) = u(0,y) = u(1,y) = 0.$$

Now you have a nonlinear system to solve using, for example, the fixed point iteration or Newton's method. A good initial guess for the fixed point/Newton iteration is given by $u^0(x,y) = \frac{-1}{2\pi^2}\sin(\pi x)\sin(\pi y)$ (which is obtained by dropping the u^3 term). In addition, there is another direct/iterative method needed for the linear system in either of the iteration.

Analyze possible combination of methods mentioned above and explain the reason for your choices.

Check that your answer is second order accurate (that is, the error is $O(N^{-2})$). Note that the exact solution $u^e(x,y)$ is not known and you have to think about how to do this.

Remark:

Other possible discretization for (1) include

$$\frac{u_{i,j}^{n+1} - u_{i,j}^{n}}{\Delta t} = \nu\left(\frac{u_{i+1,j}^{n+1} - 2u_{i,j}^{n+1} + u_{i-1,j}^{n+1}}{(\Delta x)^2} + \frac{u_{i,j+1}^{n+1} - 2u_{i,j}^{n+1} + u_{i,j-1}^{n+1}}{(\Delta y)^2}\right) \tag{5}$$

or

$$\frac{u_{i,j}^{n+1} - u_{i,j}^{n}}{\Delta t} = \nu\left(\frac{u_{i+1,j}^{n} - 2u_{i,j}^{n} + u_{i-1,j}^{n}}{(\Delta x)^2} + \frac{u_{i,j+1}^{n} - 2u_{i,j}^{n} + u_{i,j-1}^{n}}{(\Delta y)^2}\right) \tag{6}$$

Both discretization give rise to error of $O(\Delta x^2) + O(\Delta t)$ ($= O(N^{-1})$ for this problem). In addition, for (6) to be applicable, it is necessary that $\Delta t \le C\Delta x^2$ (otherwise, you will get 'NaN', which means 'not a number', an indication for overflow, almost immediately).

Ideally, the best solution would be (3) together with a fast solver for the corresponding linear system. Such a faster solver ($O(N^2)$ operations for the $(N-1)^2 \times (N-1)^2$ matrix) does exist, but is beyond the scope of this course.

The next best method is probably (6) with $\Delta t = C\Delta x^2$. Since the purpose of this homework is to review your techniques for solving linear systems, we will not consider this option here.

2. (The Schrödinger equation (simplified to 2D case). This is the replacement problem for the final exam)

   Solve numerically for the smallest eigenvalue λ and corresponding eigenfunction $u(x,y)$ from

$$-(u_{xx} + u_{yy})(x,y) + V(x,y)u(x,y) = \lambda u(x,y) \quad (x,y) \in (0,1)^2. \tag{7}$$

   with the boundary condition

$$u(x,0) = u(x,1) = u(0,y) = u(1,y) = 0. \tag{8}$$

Here $V(x,y)$ is a given function.

We take $V(x,y) = \alpha \sin(\pi x)\sin(\pi y)$ on $(x,y) \in (0,1)^2$ with $\alpha = 1$ in this problem.

In general, the eigenvalues and eigenfunctions may be complex-valued. However, if you discretize the left hand side of (7), it is easy to see that the corresponding matrix A is real and symmetric. Therefore the unknown eigenvalue we are looking for must be real. What is less obvious is that A is also positive definite (you can take it for granted). Therefore $\lambda > 0$. The goal of the problem is to find $\lambda^{(1)}$, the smallest one and the corresponding eigenfunction $u^{(1)}(x,y)$ (or, for the discretized problem, the corresponding eigenvector $u^{(1)}_{i,j} \in R^{(N-1)^2}$).

The procedure to solve for $\lambda^{(1)}$ can be explained as follows. Take any vector $u \in R^{(N-1)^2}$ and expand it in terms of the orthonormal eigen-basis of A:

$$u = \sum_{k=1}^{(N-1)^2} a^{(k)} u^{(k)}. \tag{9}$$

where $Au^{(k)} = \lambda^{(k)} u^{(k)}$ and $\lambda^{(1)} \le \lambda^{(2)} \le \cdots \le \lambda^{((N-1)^2)}$. Since

$$A^n u = \sum_{k=1}^{(N-1)^2} a^{(k)} (\lambda^{(k)})^n u^{(k)}. \tag{10}$$

It follows that the $u^{((N-1)^2)}$ component dominates in $A^n u$ as $n \to \infty$. On the other hand, the $u^{(1)}$ component will dominate as $n \to -\infty$. Therefore the strategy to get $u^{(1)}$ is to apply A^{-1} to u repeatedly until it aligns with $u^{(1)}$.

$$\begin{array}{l}
\text{do } k = 1, \cdots, \\
\quad \text{solve } w \text{ from } Aw = u_k \\
\quad u_{k+1} = w / \|w\|_2 \\
\quad \lambda_k = u_{k+1}^T A u_{k+1} \\
\text{end do}
\end{array} \tag{11}$$

You can set the stopping criterion by, for example, examining whether $|\lambda_k - \lambda_{k+1}| < tol$ holds. Since $u^{(1)}$ is an eigenvector, so is any multiple of $u^{(1)}$. Therefore it is necessary to normalize the eigenvector as in (11). At convergence, you get is the ground state $(\lambda^{(1)}, u^{(1)})$.

(a) Analyze possible approaches for solving $(\lambda^{(1)}, u^{(1)})$ and implement the method you choose.

(b) Do the same for the nonlinear Schrödinger equation

$$\begin{array}{l}
-(u_{xx} + u_{yy})(x,y) + (V(x,y) + |u(x,y)|^2)u(x,y) = \lambda u(x,y) \quad (x,y) \in (0,1)^2 \\
\int_0^1 \int_0^1 |u(x,y)|^2 = 1.
\end{array} \tag{12}$$

with the same boundary condition (8). That is, the original V now contains an additional term involving the unknown (normalized) eigenfunction. This equation is better known as the Gross-Pitaevskii equation and is related to the theory of Bose-Einstein condensate. If you are interested and have background in physics, you should be able to find some document from google to get more understanding.

This is again an harder problem. You should really consult me before and during implementation.