Math 292B (Winter 1998, Instructor: M. Holst)

Homework #2 (Finite Element Methods for Parabolic Equations)

Handed out: 9 March 1998
Due in class: 20 March 1998

You have been provided with the MATLAB finite element code FEMBIF which can treat the following class of problems numerically (for $d = 2$):

$$\begin{aligned} -\nabla \cdot (a(x)\nabla u(x)) + b(x,u) &= f(x), \quad \text{in } \Omega \subset \mathbb{R}^d, \\ n(x) \cdot (a(x)\nabla u(x)) + c(x,u) &= h(x), \quad \text{on } \partial_N\Omega, \\ u(x) &= g(x), \quad \text{on } \partial_D\Omega, \\ \partial\Omega &= \partial_D\Omega \cup \partial_N\Omega, \\ \{\} &= \partial_D\Omega \cap \partial_N\Omega. \end{aligned}$$

The MATLAB code FEMBIF discretizes this problem with Petrov-Galerkin finite element methods, based on piecewise-linear basis functions with local support. The resulting implicit nonlinear algebraic equations are then solved with a Newton iteration. Each Newton iteration requires the solution of a large, sparse, unstructured linear algebraic system of equations (the linearization or Jacobian system). These equations are solved using MATLAB's builtin sparse direct solver (some variant of sparse Gaussian elimination), or any of the iterative methods (classical, CG, or multigrid) that you implemented in the previous homework.

Our tasks in this homework are to extend the code to handle nonlinear parabolic equations.

- **Problem 1.** (Semi-Discretization in Space.)

  Adding a *momentum* term to the above system, along with an initial condition, and allowing the various coefficients to now vary also in time, gives the following nonlinear parabolic equation:

$$\begin{aligned} \rho\dot{u}(x,t) - \nabla \cdot (a(x,t)\nabla u(x,t)) + b(x,t,u) &= f(x,t), \quad \text{in } \Omega \times (0,T), \qquad \Omega \subset \mathbb{R}^d, \\ n(x) \cdot (a(x,t)\nabla u(x,t)) + c(x,t,u) &= h(x,t), \quad \text{on } \partial_N\Omega \times (0,T), \\ u(x,t) &= g(x,t), \quad \text{on } \partial_D\Omega \times (0,T), \\ u(x,0) &= u^0(x), \quad \text{in } \Omega, \\ \partial\Omega &= \partial_D\Omega \cup \partial_N\Omega, \\ \{\} &= \partial_D\Omega \cap \partial_N\Omega. \end{aligned}$$

  when we discretize the original elliptic system using the usual Galerkin approach with N continuous piecewise-linear basis functions $\{\phi\}_{j=1}^N$ over triangles, we obtain the nonlinear algebraic system

$$N(\alpha) = f,$$

  where $u(x) = \sum_{j=1}^N \alpha_j \phi_j(x)$, and where $N(\cdot)$ is implicitly defined. For the now parabolic problem, we semi-discretize in space by looking for a solution of the form

$$u(x,t) = \sum_{j=1}^{N} \alpha_j(t)\phi_j(x).$$

  Employing the weak formulation from the lectures, with test functions taken to be the same as our trial functions $\phi_j(x)$, we obtain the nonlinear system of ODEs:

$$M\dot{u} + N(u) = f,$$

where M is the mass matrix discussed in the lectures.

It is clear then that to semi-discretize the parabolic problem, all we need to do is assemble the mass matrix M when we assemble the nonlinear residual $N(\cdot)$ (or the linearization of $N(\cdot)$ used in the Newton iteration, namely the linearized stiffness matrix A). For this first problem, simply add the required code to `assem.m` to form M as you form $N(\cdot)$ or A.

- **Problem 2.** (Method of Lines – Explicit and Implicit Methods.)

  Now that you have the semi-discretization in space provided by the mass matrix, implement the following three numerical methods for solving the resulting system of ODEs (see the lecture notes for the detailed descriptions of the algorithms):

  1. Forward Euler
  2. Backward Euler
  3. Crank-Nicolson (extra credit)

  For simplicity, go ahead an invert M when you need to (by solving a system, not by forming M^{-1} explicitly!!!), rather than doing something like "mass-lumping" (which we have not discussed yet).

  Note that for the implicit methods, you would normally have to use the Newton iteration to solve the implicit equations at each time step. However, for simplicity, you can just implement these methods for the linear case if you prefer.

  As a test of the three methods, solve the following model problem with your implementation, on the domain $\Omega = [0,1] \times [0,1] \subset \mathbb{R}^2$:

$$\begin{aligned} \dot{u}(x,y,t) - \nabla^2 u(x,y,t) &= (\pi^2 - 1)e^{-t}\sin(\pi x)\sin(\pi y), \quad \text{in } \Omega \times (0,T), \\ u(x,y,t) &= 0, \quad \text{on } \partial\Omega \times (0,T), \\ u(x,y,0) &= \sin(\pi x)\sin(\pi y), \quad \text{on } \Omega. \end{aligned}$$

  See if you recover the analytical solution (if I've done the math right):

$$u(x,y,t) = e^{-t}\sin(\pi x)\sin(\pi y).$$

  Does the stability restriction discovered in class for the explicit method rear its head? I.e., does the method blow up if you take step-sizes which are too large? What happens to the implicit methods for large stepsizes? Does the accuracy in space and time, as you refine the mesh and shorten the time step, agree with the predictions derived in class?

- **Problem 3.** (The elusive *a priori* estimate.)

  Consider the initial boundary-value problem:

$$\begin{aligned} \rho\dot{u}(x,t) - Lu(x,t) &= 0, \quad \text{in } \Omega \times (0,T), \qquad \Omega \subset \mathbb{R}^d, \\ u(x,t) &= 0, \quad \text{on } \partial\Omega \times (0,T), \\ u(x,0) &= u^0(x), \quad \text{in } \Omega, \end{aligned}$$

  where as usual, L is a uniformly elliptic operator, generating a bounded and coercive bilinear weak form (not necessarily symmetric). Prove the following very general *a priori* estimate which eluded us for several lectures this quarter.

Theorem:

$$\|\dot{u}(x,t)\|_{L^2(\Omega)} \le \frac{C}{t}\|u(x,0)\|_{L^2(\Omega)}, \qquad t \text{ in } (0,T).$$

Hints: We have already shown, in this general setting, the following *a priori* estimate (via a Gronwall inequality):

$$\|u(x,t)\|_{L^2(\Omega)} \le \|u(x,0)\|_{L^2(\Omega)}, \qquad t \text{ in } (0,T).$$

To prove the new estimate, start as we did in class in the proof of the estimate above, beginning with the weak form:

$$\text{Find } u(x,t) \text{ such that } (\dot{u},v) + < Lu, v >= 0, \quad \forall v \in H_0^1(\Omega), t \in (0,T),$$

$$(u(x,0),v) = (u^0(x),v), \quad \forall v \in H_0^1(\Omega),$$

where $< Lu, v >$ represents the bilinear weak form after appropriate integration by parts. The analysis trick (revealed to me by Professor Xu) is to use a special test function ($\phi = 2t^2\dot{u}$) in the weak form. You then employ the coercivity assumption on $< Lu, v >$ (coming from the ellipticity assumption on L), and integrate, to obtain a differential inequality for $< Lu, u >$. Now employ the weak form again, with a second special test function ($\phi = 2\dot{u}$), use the differential inequality you just obtained for $< Lu, u >$, multiply the result by t, and then integrate. This should give you the above estimate...good luck (it is still not trivial, even with this hint).