CAAM 453/553 · NUMERICAL ANALYSIS I

Problem Set 7

Posted Tuesday, 24 November 2009. Due Friday, 4 December 2009.
CAAM 453 students should complete 60 points worth of problems.
CAAM 553 students should complete 80 points worth of problems.

1. [40 points]

   Convection–diffusion equations play an important role in fluid dynamics. In one dimension, the simplest such equation takes the form

   $$-\varepsilon u''(x) + u'(x) = 0, \qquad u(0) = a, \quad u(1) = b.$$

   (The second derivative term, Δu in higher dimensions, gives diffusion; the first derivative term, $\mathbf{w}^T\nabla u$ in higher dimensions, gives convection in the direction of the 'wind', $\mathbf{w}$.)

   Note that this convection–diffusion equation is a *boundary value problem*, rather than an initial value problem. As stated, it is easy enough to solve by hand, but it will be useful to develop a numerical method that we could also apply to more difficult problems. The *shooting method* is one option:

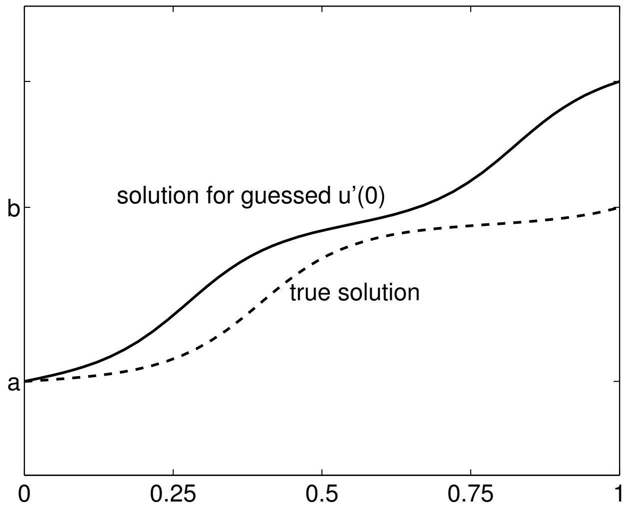
   - Write this second-order ODE as a system of two first-order ODEs:

     $$\begin{aligned} u_1'(x) &= u_2(x) \\ u_2'(x) &= \varepsilon^{-1}u_2(x). \end{aligned}$$

   - Guess some value for $u'(0)$.
   - Integrate this system (e.g., using a Runge–Kutta method) for $x \in [0,1]$ with the initial values $u_1(0) = u(0) = a$ (given by the problem) and $u_2 = u'(0)$ (guessed).
   - Unless you are lucky, the solution you obtain will not match the boundary condition $u(1) = b$, because the guessed value for $u'(0)$ is not correct. One can use a nonlinear root-finding algorithm (e.g., bisection, *regula falsi*, the secant method, or Newton's method) to adjust the guess $u'(0)$ until the integrated value at $x = 1$ agrees with the desired $u(1) = b$. That is, one seeks a zero of the objective function

     $$f(\xi) = b - (u(1) \text{ computed with } u'(0) = \xi).$$

   The following figure shows a schematic view of the shooting method (for a different differential equation). The solid line is the solution to the ODE with the correct value $u(0) = a$, but the incorrect $u'(0)$. Since this initial slope is incorrect, the corresponding value for $u(1)$ is also wrong. The dashed line shows the true solution, which satisfies $u(1) = b$. The challenge is to adjust the guessed value for $u'(0)$ so that the computed $u(1)$ satisfies the boundary condition $u(1) = b$.



Your task is to solve the convection-diffusion equation.

(a) Implement the shooting method to solve the above convection-diffusion boundary value problem with $\varepsilon = 1/10$, $u(0) = 0$ and $u(1) = 1$. Please use MATLAB's built-in ODE integrator, `ode45`; you may use any root-finding algorithm you like, but please implement it yourself. If you use the bisection or *regula falsi* algorithms, use $u'(0) = 0$ and $u'(0) = 1$ to obtain your initial bracket. If you use the secant method or Newton's method, try $u'(0) = 0$ as an initial guess.

Please present your code, a plot of $u(x)$ for $x \in [0, 1]$, and the value of $u'(0)$ that gives $u(1) = 1$.

(b) Repeat the same experiment for $\varepsilon = 1/50$. The exact solution demonstrates a *boundary layer* near $x = 1$.

(c) Derive the exact solution for this convection-diffusion problem. In particular, what are the exact values for $u'(0)$ in parts (a) and (b)?

2. [20 points]

Suppose that Gaussian elimination with partial pivoting is applied to some nonsingular matrix $\mathbf{A} \in \mathbb{C}^{n\times n}$ to compute the factorization $\mathbf{PA} = \mathbf{LU}$. Prove that the growth factor

$$\rho := \frac{\max_{j,k} |u_{jk}|}{\max_{j,k} |a_{jk}|}$$

must satisfy $\rho \le 2^{n-1}$.

[Trefethen and Bau, problem 22.1]

3. [20 points]

Suppose $\mathbf{A} \in \mathbb{C}^{n\times n}$ is *strictly column diagonally dominant* (SCDD), which means that for each k,

$$|a_{kk}| > \sum_{j\neq k} |a_{jk}|.$$

Show that if Gaussian elimination with partial pivoting is applied to $\mathbf{A}$, no row interchanges take place. (In practice, this means that if one knows a matrix is SCDD, then there is no need to check for pivots.)

[Trefethen and Bau, problem 21.6]

4. [20 points]

Suppose that we have the LU factorization $\mathbf{A} = \mathbf{LU}$ of $\mathbf{A} \in \mathbb{R}^{n\times n}$.

(a) Explain how one can efficiently compute $\mathbf{A}^{-1}$. How many further operations are required?

(b) Show how to compute the (j, k) entry of $\mathbf{A}^{-1}$ in approximately $(n - j)^2 + (n - k)^2$ floating point operations. [Golub and Van Loan]

Supplemental Problem

S1. Suppose that the matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$ has upper band width $b > 0$ and lower band width $\ell > 0$; that is,

$$a_{j,k} = 0$$

if $k - j > b$ or $j - k > \ell$. To leading order, what is the complexity of Gaussian elimination for such a matrix? Assume that no pivoting is required.

S2. [15 points]

Apply the QR eigenvalue algorithm (without upper Hessenberg reduction or shifting) to the matrix

$$\mathbf{A} = \begin{pmatrix} 325 & 309 & 634 & 250 & 0 \\ -104 & -96 & -200 & -70 & 8 \\ -102 & -98 & -200 & -84 & -4 \\ -20 & -20 & -40 & -14 & 0 \\ 34 & 34 & 76 & 44 & 16 \end{pmatrix}.$$

What are the eigenvalues of $\mathbf{A}$?

For output, submit only $\mathbf{A}_{10}$, $\mathbf{A}_{20}$, ... Continue until you decide that convergence has occurred.

Now use the `hess` command reduces to reduce $\mathbf{A}$ to upper Hessenberg form, $\mathbf{A} = \mathbf{U}\mathbf{H}\mathbf{U}^*$:

$$[\mathtt{U},\mathtt{H}] = \mathtt{hess}(\mathtt{A})$$

Repeat the above experiment with the upper Hessenberg matrix $\mathbf{H}$.

S3. Recall that a Hermitian matrix $\mathbf{M}$ is positive definite provided $\mathbf{x}^*\mathbf{M}\mathbf{x} > 0$ for all nonzero $\mathbf{x} \in \mathbb{C}^n$, or, equivalently, if all its eigenvalues are positive real numbers.

(a) Show that if $\mathbf{M} \in \mathbb{C}^{n\times n}$ is Hermitian positive definite, then $\mathbf{M}^{-1}$ is Hermitian positive definite.

(b) Show that if, in addition, $\mathbf{B} \in \mathbb{C}^{n\times m}$ has full rank with $n > m$, then $\mathbf{B}^*\mathbf{M}^{-1}\mathbf{B}$ is Hermitian positive definite.

(c) Let $\mathbf{M}$ and $\mathbf{B}$ be as in parts (a) and (b). Linear systems of the form

$$\begin{pmatrix} \mathbf{M} & \mathbf{B} \\ \mathbf{B}^* & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix} = \begin{pmatrix} \mathbf{f} \\ \mathbf{g} \end{pmatrix}$$

play an essential role in applications such as optimization and fluid dynamics.
This matrix is Hermitian but not positive definite, and hence it does not generally have a Cholesky factorization. However, the matrix does have a triangular factorization

$$\mathbf{A} = \begin{pmatrix} \mathbf{L} & \mathbf{0} \\ \mathbf{K}^* & \mathbf{H} \end{pmatrix} \begin{pmatrix} \mathbf{L}^* & \mathbf{K} \\ \mathbf{0} & -\mathbf{H}^* \end{pmatrix}.$$

Here $\mathbf{L}\mathbf{L}^*$ and $\mathbf{H}\mathbf{H}^*$ are Cholesky factorizations of two different matrices.

— Determine the matrix that $\mathbf{L}\mathbf{L}^*$ must equal in terms of $\mathbf{M}$ and/or $\mathbf{B}$, and explain why this matrix must have a Cholesky factorization.

— Specify what $\mathbf{K}$ must equal in terms of $\mathbf{B}$ and/or $\mathbf{L}$.

— Determine the matrix that $\mathbf{H}\mathbf{H}^*$ must equal in terms of $\mathbf{M}$ and/or $\mathbf{B}$, and explain why this matrix must have a Cholesky factorization.

(This procedure provides an efficient alternative to computing an $\mathbf{LU}$ factorization of the original $(n+m)\times(n+m)$ matrix.)