MATH 335: Numerical Analysis

Problem Set 7, Final version

Due Date: Thurs., Feb. 26, 2009

Read Sections 4.1, 4.2, 4.3 and 4.4 in Kharab & Guenther.

AP 3.16 A van der Walls fluid is one which satisfies the equation of state

$$p = \frac{R\theta}{v-b} - \frac{a}{v^2}\,.$$

Here R, a, b are positive constants, p is the pressure, θ the absolute temperature, and v the volume. Show that if one sets

$$P = \frac{27b^2}{a}p, \qquad T = \frac{27Rb}{8a}\theta, \qquad V = \frac{v}{3b}\,,$$

the equation of state takes the dimensionless form

$$\left(P + \frac{3}{V^2}\right)(3V-1) = 8T\,.$$

Observe that $V > 1/3$ since $T > 0$.

Find V to five decimal places using a numerical method when

(a) $P = 6, T = 2$.

(b) $P = 4, T = 1$.

(c) $P = 5, T = 5$.

★8 Consider the matrix problem $\mathbf{Ax} = \mathbf{b}$, with

$$\mathbf{A} = \begin{bmatrix} 0.0002 & 0.2 \\ 2 & 2 \end{bmatrix} \qquad \text{and} \qquad \mathbf{b} = \begin{bmatrix} 0.2 \\ 4 \end{bmatrix}.$$

(a) Solve this linear system using naive Gaussian elimination (i.e., no pivoting) and three-digit (decimal) arithmetic. Compare your result to the true answer $\begin{bmatrix} 1.\overline{001} & 0.\overline{998} \end{bmatrix}^T$.

(b) Solve the same system using Gaussian elimination with scaled partial pivoting and three-digit arithmetic.

(c) Repeat part (a) with $\mathbf{b} = \begin{bmatrix} 0.2 & 5.6 \end{bmatrix}^T$ (for which the true solution is is approximately $\begin{bmatrix} 1.8018 & 0.9982 \end{bmatrix}^T$). Then multiply both sides of $\mathbf{Ax} = \mathbf{b}$ by the matrix

$$\mathbf{S} = \begin{bmatrix} 10 & 0 \\ 0 & 1 \end{bmatrix},$$

giving the equivalent system $\mathbf{SAx} = \mathbf{Sb}$, and then repeat part (a). This corresponds to performing the elementary row operations of multiplying row 1 by a constant and row 2 by a constant before beginning the naive Gaussian elimination. Does it help?

(d) We say that a matrix exhibits *artificial ill-conditioning* if its poor behavior with respect to Gaussian elimination is due in large measure to it being poorly scaled: The entries in the matrix vary widely, over several orders of magnitude, which is always a warning sign that we should be on the lookout for poor numerical behavior. Mutiply both sides of the matrix equation

$$\begin{bmatrix} 0.007 & -0.8 \\ -0.1 & 10 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0.7 \\ 10 \end{bmatrix}$$

by the matrix $\mathbf{S}$ of part (c), and solve the resulting system by naive Gaussian elimination in three-digit arithmetic. Does this help? (If not, the matrix may be truly, not artificially, ill-conditioned.) This technique is sometimes referred to as *balancing* the matrix, that is, making all entries, or the sums along all rows (in absolute value), more nearly equal.

★9 The function `naiveGE.m` (not part of the OCTAVE distribution) implements *naive* Gaussian elimination in OCTAVE (so using IEEE 64-bit floating point arithmetic). That is, when you call `naiveGE` sending it a matrix, it will (so long as your matrix has at least as many columns as rows, and at every step has a nonzero number which can serve as a pivot) return an echelon form for that matrix.

(a) Use `naiveGE` and back substitution to solve the linear system of Exercise 4.3.1 (Kharab & Guenther):

$$\begin{bmatrix} -1 & 2 & -1 & 0 & 4 \\ 1 & 2 & 0 & 3 & 0 \\ 0 & -3 & 1 & 1 & 2 \\ 1 & 0 & 2 & -1 & 3 \\ 2 & -2 & 2 & -2 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 4 \\ 0 \\ 3 \end{bmatrix}.$$

(b) Modify the `naiveGE` routine to accept the coefficient matrix $\mathbf{A}$ and right-hand side $\mathbf{b}$ as separate parameters, and so that it returns *just* the solution of the problem (i.e., so the backward substitution is carried out in the routine itself). Make sure the returned solution for the problem above agrees with the one you got in part (a).

(c) Modify the routine yet again so that it carries out scaled partial pivoting. (Make sure throughout this problem you are understanding and modifying *my program*, not replacing it with some pre-written program which already does what is desired.)

★10 Solve $\mathbf{LUx} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 2 & 4 & 4 \\ 0 & 1 & 2 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 2 \\ 0 \\ 2 \end{bmatrix}$ without multiplying **L** and **U** to find **A**.