L. Vandenberghe
EE103 10/28/08

Midterm Solutions

Problem 1 (10 points). Express the following problem as a set of linear equations. Find a cubic polynomial

$$f(t) = c_1 + c_2(t - t_1) + c_3(t - t_1)^2 + c_4(t - t_1)^2(t - t_2)$$

that satisfies

$$f(t_1) = y_1, \qquad f(t_2) = y_2, \qquad f'(t_1) = s_1, \qquad f'(t_2) = s_2.$$

The numbers t_1, t_2, y_1, y_2, s_1, s_2 are given, with $t_1 \neq t_2$. The unknowns are the coefficients c_1, c_2, c_3, c_4. Write the equations in matrix-vector form $Ax = b$, and solve them.

Solution. We first derive expressions for $f(t_1)$, $f(t_2)$, $f'(t_1)$, $f'(t_2)$:

$$f(t_1) = c_1, \qquad f(t_2) = c_1 + c_2 h + c_3 h^2, \qquad f'(t_1) = c_2, \qquad f'(t_2) = c_2 + 2c_3 h + c_4 h^2$$

where $h = t_2 - t_1$. In matrix notation, the four interpolation conditions are

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & h & h^2 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 2h & h^2 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ s_1 \\ s_2 \end{bmatrix}.$$

If we exchange the second and third rows, we can solve this by forward substitution. The solution is

$$\begin{aligned} c_1 &= y_1, \\ c_2 &= s_1, \\ c_3 &= \frac{y_2 - c_1 - hc_2}{h^2} \\ &= \frac{(y_2 - y_1)/h - s_1}{h}, \\ c_4 &= \frac{s_2 - c_2 - 2hc_3}{h^2} \\ &= \frac{s_2 - s_1 - 2((y_2 - y_1)/h - s_1)}{h^2} \\ &= \frac{s_2 + s_1 - 2(y_2 - y_1)/h}{h^2}. \end{aligned}$$

Problem 2 (10 points). A diagonal matrix with diagonal elements $+1$ or -1 is called a *signature matrix*. The matrix

$$S = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix}$$

is an example of a 3×3 signature matrix. If S is a signature matrix, and A is a square matrix that satisfies

$$A^T S A = S, \tag{1}$$

then we say that A is *pseudo-orthogonal* with respect to S.

1. Suppose S is an $n \times n$ signature matrix, and u is an n-vector with $u^T S u \neq 0$. Show that the matrix
$$A = S - \frac{2}{u^T S u} u u^T$$
is pseudo-orthogonal with respect to S.

2. Show that pseudo-orthogonal matrices are nonsingular. In other words, show that any square matrix A that satisfies (1) for some signature matrix S is nonsingular.

3. Describe an efficient method for solving $Ax = b$ when A is pseudo-orthogonal. 'Efficient' here means that the complexity is at least an order of magnitude less than the $(2/3)n^3$ complexity of the standard method for a general set of linear equations. Give the complexity of your method (number of flops for large n).

4. Show that if A satisfies (1) then $ASA^T = S$. In other words, if A is pseudo-orthogonal with respect to S, then A^T is also pseudo-orthogonal with respect to S.

Solution.

1. A is symmetric and
$$\begin{aligned} A^T S A &= (S - \frac{2}{u^T S u} u u^T) S (S - \frac{2}{u^T S u} u u^T) \\ &= S^3 - \frac{2}{u^T S u} S^2 u u^T - \frac{2}{u^T S u} u u^T S^2 + \frac{4}{(u^T S u)^2} u u^T S u u^T \\ &= S - \frac{2}{u^T S u} u u^T - \frac{2}{u^T S u} u u^T + \frac{4}{u^T S u} u u^T \\ &= S. \end{aligned}$$
(We used the fact that $S^2 = I$.)

2. We show that A has a zero nullspace. Suppose $Ax = 0$. Then
$$(SA^T S)Ax = 0 \quad \Longrightarrow \quad S(A^T S A)x = 0 \quad \Longrightarrow \quad S^2 x = x = 0.$$

3. To solve $Ax = b$, we can multiply both sides of the equation with SA^TS on the left to get
$$x = SA^TSb.$$
The multiplications with S involve only sign changes. The multiplication with A^T costs $2n^2$ flops.

4. Multiplying $A^TSA = S$ on the left with S gives $SA^TSA = I$. This means that $A^{-1} = SA^TS$. For a square matrix $AA^{-1} = A^{-1}A = I$. Therefore
$$ASA^T = ASA^TS^2 = A(SA^TS)S = AA^{-1}S = S.$$

Problem 3 (10 points). Define a block matrix

$$K = \begin{bmatrix} A & B \\ B^T & -C \end{bmatrix},$$

where the three matrices A, B, C have dimension $n \times n$. The matrices A and C are symmetric and positive definite. Show that K can be factored as

$$K = \begin{bmatrix} L_{11} & 0 \\ L_{21} & L_{22} \end{bmatrix} \begin{bmatrix} I & 0 \\ 0 & -I \end{bmatrix} \begin{bmatrix} L_{11}^T & L_{21}^T \\ 0 & L_{22}^T \end{bmatrix},$$

where L_{11} and L_{22} are lower triangular matrices with positive diagonal elements. The blocks L_{11}, L_{21}, L_{22}, and the two identity matrices on the righthand side, all have size $n \times n$.

What is the cost of computing this factorization (number of flops for large n)? Carefully explain your answers.

Solution. If we work out the product in the factorization we get

$$K = \begin{bmatrix} L_{11}L_{11}^T & L_{11}L_{21}^T \\ L_{21}L_{11}^T & L_{21}L_{21}^T - L_{22}L_{22}^T \end{bmatrix}.$$

This gives three conditions

$$A = L_{11}L_{11}^T, \qquad B = L_{11}L_{21}^T, \qquad C + L_{21}L_{21}^T = L_{22}L_{22}^T.$$

This shows that L_{11} is the Cholesky factor of A, and L_{22} is the Cholesky factor of $C + L_{21}L_{21}^T$. (This last matrix is positive definite because

$$x^T(C + L_{21}L_{21}^T)x = x^TCx + x^TL_{21}L_{21}^Tx = x^TCx + \|L_{21}^Tx\|_2^2 > 0$$

for all nonzero x if C is positive definite.)

The factorization can therefore be computed as follows:

- Cholesky factorization $A = L_{11}L_{11}^T$. ($(1/3)n^3$ flops).
- Compute $L_{21} = B^TL_{11}^{-T}$ by solving $L_{11}L_{21}^T = B$. (n^3 flops because each column of L_{21}^T can be computed by forward substitution in n^2 flops.).
- Calculate $D = C + L_{21}L_{21}^T$. (n^3 flops for the matrix-matrix product, if we take into account that the result is a symmetric matrix.)
- Cholesky factorization $D = L_{22}L_{22}^T$. ($(1/3)n^3$ flops.)

The total is $(8/3)n^3$ flops, the same complexity as a Cholesky factorization of order $2n$.

Problem 4 (10 points). Describe an efficient method for solving the equation

$$\begin{bmatrix} 0 & A^T & I \\ A & 0 & 0 \\ I & 0 & D \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} b \\ c \\ d \end{bmatrix}.$$

The nine blocks in the coefficient matrix have size $n \times n$. The matrix A is nonsingular, and the matrix D is diagonal with nonzero diagonal elements. The vectors b, c, and d in the righthand side are n-vectors. The variables are the n-vectors x, y, z.

If you know several methods, give the most efficient one. Clearly state the different steps in your algorithm, give the complexity (number of flops) of each step, and the total number of flops.

Solution. We can first solve the second equation $Ax = c$ for x, using the standard method based on the LU factorization. Given x, we solve the third equation $Dz = d - x$ for z. Finally we solve $A^T y = b - z$ for y. We can reuse the LU factorization of A to solve the equation in y.

1. LU factorization $A = PLU$ ($(2/3)n^3$ flops).
2. Solve $PLUx = c$ by forward and backward substitution ($2n^2$ flops).
    - $x_1 = P^T c$ (0 flops).
    - Solve $Lx_2 = x_1$ by forward substitution (n^2 flops).
    - Solve $Ux = x_2$ by backsubstitution (n^2 flops).
3. Compute $d - x$ and solve $Dz = d - x$ ($2n$ flops).
4. Compute $b - z$ and solve $U^T L^T P^T y = b - z$ by forward and backward substitution ($2n^2$ flops.)
    - Solve $U^T y_1 = b - z$ by forward substitution (n^2 flops).
    - Solve $L^T y_2 = y_1$ by backsubstitution (n^2 flops).
    - $y = Py_2$ (0 flops).

The dominant term in the total complexity is $(2/3)n^3$ for the LU factorization.