L. Vandenberghe
EE103

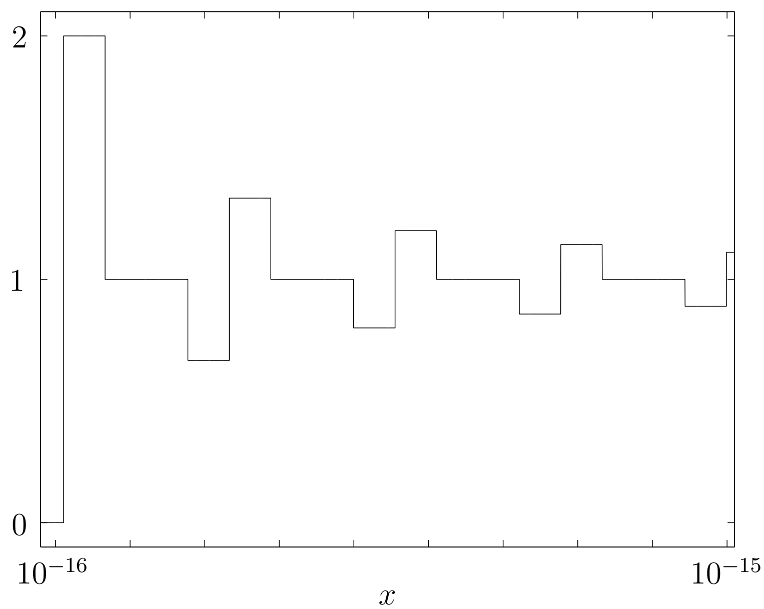
6/16/05

Final Exam Solutions

Problem 1 (20 points). The figure shows the function

$$f(x) = \frac{(1+x)-1}{1+(x-1)}$$

evaluated in IEEE double precision arithmetic in the interval $[10^{-16}, 10^{-15}]$, using the Matlab command `((1+x)-1)/(1+(x-1))` to evaluate $f(x)$.

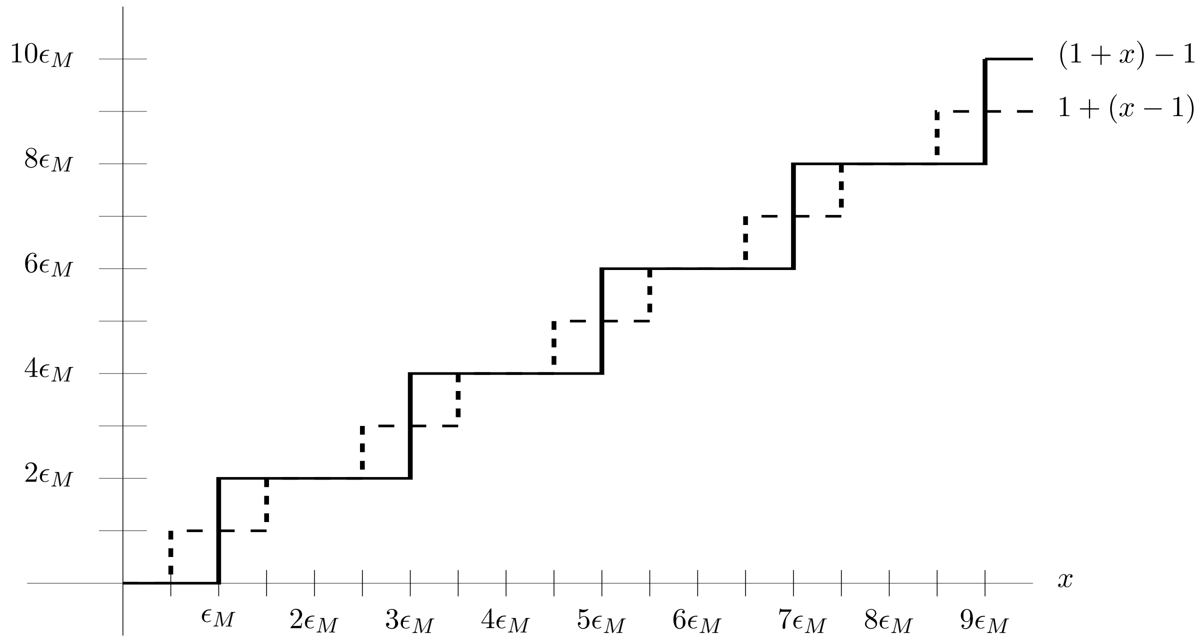


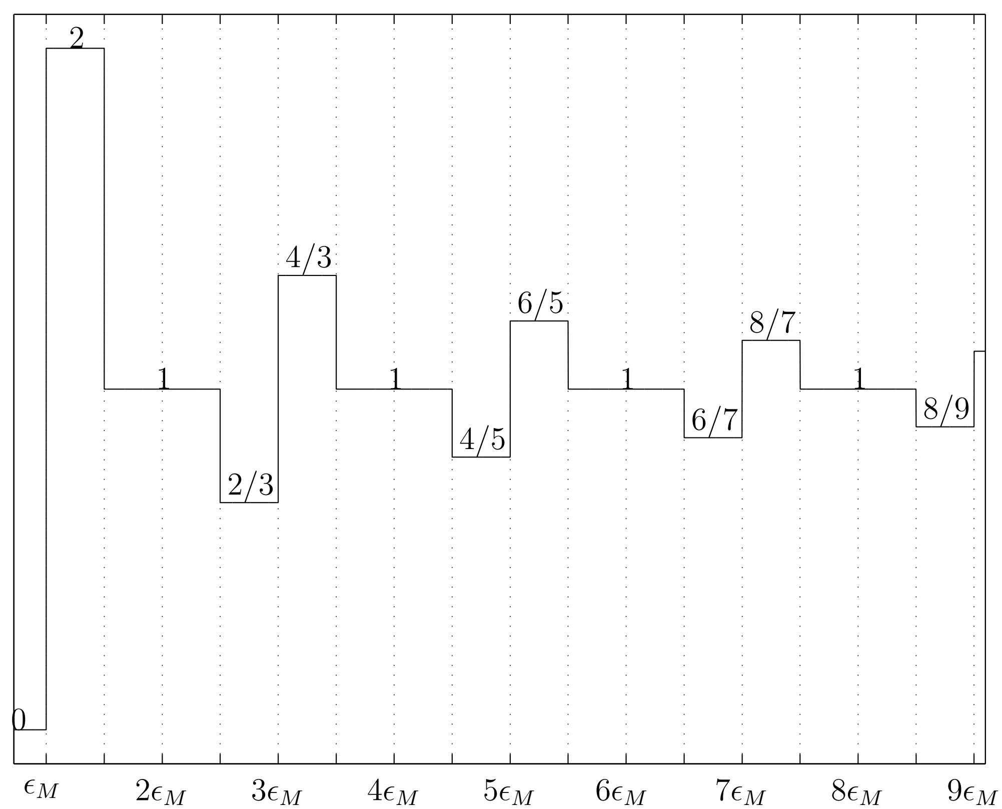
We notice that the computed function is piecewise-constant, instead of a constant 1.

1. What are the endpoints of the intervals on which the computed values are constant?
2. What are the computed values on each interval?

Carefully explain your answers.

Solution. The first figure shows the rounded values of the numerator and denominator. The second plot shows the result of the division.

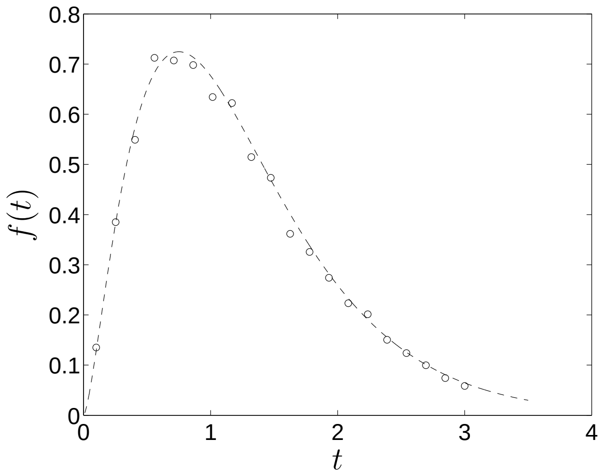




Problem 2 (20 points). The figure shows $m = 20$ points $(t_i,\ y_i)$ as circles. These points are well approximated by a function of the form

$$f(t) = \alpha t^\beta e^{\gamma t}.$$

(An example is shown in dashed line.)



Explain how you would compute values of the parameters α, β, γ such that

$$\alpha t_i^\beta e^{\gamma t_i} \approx y_i, \quad i = 1, \ldots, m, \tag{1}$$

using the following two methods.

1. The Gauss-Newton method applied to the nonlinear least-squares problem

$$\text{minimize} \quad \sum_{i=1}^{m} \left(\alpha t_i^\beta e^{\gamma t_i} - y_i\right)^2$$

   with variables α, β, γ. Your description should include a clear statement of the linear least-squares problems you solve at each iteration. You do not have to include a line search.

2. Solving a single linear least-squares problem, obtained by selecting a suitable error function for (1) and/or making a change of variables. Clearly state the least-squares problem, the relation between its variables and the parameters α, β, γ, and the error function you choose to measure the quality of fit in (1).

Solution.

1. This problem is a nonlinear least-squares problem

$$\text{minimize} \quad g(x) = \textstyle\sum_{i=1}^m r_i(x)^2$$

   with variables $x = (\alpha, \beta, \gamma)$ and $r_i(\alpha, \beta, \gamma) = \alpha t_i^\beta e^{\gamma t_i} - y_i$. Starting at some initial estimate x (for example, computed with the method of part 2), we repeatedly solve linear least-squares problems

$$\text{minimize} \quad \|A^{(k)}x - b^{(k)}\|^2$$

with $b^{(k)} = Ax^{(k)} - r^{(k)}$ and

$$A^{(k)} = \begin{bmatrix} t_1^\beta e^{\gamma t_1} & \alpha(\log t_1) t_1^\beta e^{\gamma t_1} & \alpha t_1^{\beta+1} e^{\gamma t_1} \\ t_2^\beta e^{\gamma t_2} & \alpha(\log t_2) t_2^\beta e^{\gamma t_2} & \alpha t_2^{\beta+1} e^{\gamma t_2} \\ \vdots & \vdots & \vdots \\ t_m^\beta e^{\gamma t_m} & \alpha(\log t_m) t_m^\beta e^{\gamma t_m} & \alpha t_m^{\beta+1} e^{\gamma t_m} \end{bmatrix}, \qquad r^{(k)} = \begin{bmatrix} \alpha t_1^\beta e^{\gamma t_1} - y_1 \\ \alpha t_2^\beta e^{\gamma t_2} - y_2 \\ \vdots \\ \alpha t_m^\beta e^{\gamma t_m} - y_m \end{bmatrix}.$$

(The elements in the ith row are the derivatives of r_i with respect to α, β and γ.) We take the solution of this LS problem as the next iterate $x^{(k+1)}$. We terminate when the gradient $\nabla g(x) = 2{A^{(k)}}^T r^{(k)}$ is sufficiently small.

2. Taking the logarithm on both sides we can write the equations as

$$\log \alpha + \beta \log t_i + \gamma t_i \approx \log y_i.$$

We can solve this as a linear least-squares problem if we use the error function

$$\sum_{i=1}^{m} (\log \alpha + \beta \log t_i + \gamma t_i - \log y_i)^2$$

and take as variables $x = (\log \alpha, \beta, \gamma)$. In matrix form, we minimize $\|Ax - b\|^2$ where

$$A = \begin{bmatrix} 1 & \log t_1 & t_1 \\ 1 & \log t_2 & t_2 \\ \vdots & \vdots & \vdots \\ 1 & \log t_m & t_m \end{bmatrix}, \qquad b = \begin{bmatrix} \log y_1 \\ \log y_2 \\ \vdots \\ \log y_m \end{bmatrix}.$$

Problem 3 (20 points). Let $\hat{x}$ and $\hat{y}$ be the solutions of the least-squares problems

$$\text{minimize} \quad \|Ax-b\|^2, \qquad \text{minimize} \quad \|Ay-c\|^2$$

where $A \in \mathbf{R}^{m\times n}$, $b \in \mathbf{R}^m$, $c \in \mathbf{R}^m$ with $\mathbf{rank}(A) = n$. We assume that $A\hat{x} \neq b$.

1. Show that the $m \times (n+1)$ matrix $\begin{bmatrix} A & b \end{bmatrix}$ has rank $n+1$.
2. Show that the solution of the least-squares problem

$$\text{minimize} \quad \left\| \begin{bmatrix} A & b \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} - c \right\|^2,$$

   with variables $u \in \mathbf{R}^n$, $v \in \mathbf{R}$, is given by

$$\hat{u} = \hat{y} - \frac{b^Tc - b^TA\hat{y}}{b^Tb - b^TA\hat{x}}\hat{x}, \qquad \hat{v} = \frac{b^Tc - b^TA\hat{y}}{b^Tb - b^TA\hat{x}}.$$

3. Describe an efficient method for computing $\hat{x}$, $\hat{y}$, $\hat{u}$, $\hat{v}$, given A, b, c, using the QR factorization of A. Clearly describe the different steps in your algorithm. Give a flop count for each step and a total flop count. In the total flop count, include all terms that are cubic (n^3, mn^2, m^2n, m^3) and quadratic (m^2, mn, n^2). If you know several methods, give the most efficient one (least number of flops for large m and n).

Solution.

1. We have to show that $Ax + bt = 0$ is only possible if x and t are zero. If $t \neq 0$, we have $-(1/t)Ax = b$, but this is impossible because it would mean that $-(1/t)x$ is equal to the least-squares solution $\hat{x}$ and therefore $A\hat{x} = b$. If $t = 0$, we must have $x = 0$ because A has rank n.

2. The normal equations for the least-squares problem are

$$\begin{bmatrix} A^TA & A^Tb \\ b^TA & b^Tb \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} A^Tc \\ b^Tc \end{bmatrix},$$

   i.e.,

$$A^TAu = A^Tc - vA^Tb, \qquad b^TAu + b^Tbv = b^Tc.$$

   From the first equation,

$$u = (A^TA)^{-1}A^Tc - (A^TA)^{-1}A^Tbv = \hat{y} - v\hat{x}. \tag{2}$$

   Substituting in the second equation gives

$$b^TA(\hat{y} - v\hat{x}) + b^Tbv = b^Tc$$

   and solving for v gives

$$\hat{v} = \frac{b^Tc - b^TA\hat{y}}{b^Tb - b^TA\hat{x}}.$$

   The denominator is nonzero because $b^T(b - A\hat{x}) = (b - A\hat{x})^T(b - A\hat{x}) = \|b - A\hat{x}\|^2 \neq 0$. Substituting $\hat{v}$ in (2) gives the expression for $\hat{u}$.

3. We use the standard method for computing $\hat{x}$ and $\hat{y}$.

   (a) QR factorization of A ($2mn^2$ flops).

   (b) Compute $\hat{x}$ by solving $Rx = Q^T b$ ($2mn$ for the matrix-vector multiplication and n^2 for the backsubstitution).

   (c) Compute $\hat{y}$ by solving $Ry = Q^T c$ ($2mn + n^2$).

   The expressions for $\hat{u}$ and $\hat{v}$ can be simplified by noting that

$$b^T(b - A\hat{x}) = b^T b - b^T QR\hat{x} = b^T b - (Q^T b)^T (Q^T b)$$

   and

$$b^T(c - A\hat{y}) = b^T c - b^T QR\hat{y} = b^T c - (Q^T b)^T (Q^T c).$$

   Since we already computed $Q^T b$ and $Q^T c$, this only requires linear operations ($2m$ for the inner products $b^T b$ and $b^T c$, and $2n$ for the inner products $(Q^T b)^T (Q^T b)$ and $(Q^T b)^T (Q^T c)$). Finally, we make a vector addition to get $\hat{u}$.

   The total cost is $2mn^2 + 2n^2 + 4mn$.

Problem 4 (15 points). Suppose A and B are $n \times n$ matrices with A nonsingular, and b, c and d are vectors of length n. Describe an efficient algorithm for solving the set of linear equations

$$\begin{bmatrix} A & B & 0 \\ 0 & A^T & B \\ 0 & 0 & A \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} b \\ c \\ d \end{bmatrix}$$

with variables $x_1 \in \mathbf{R}^n$, $x_2 \in \mathbf{R}^n$, $x_3 \in \mathbf{R}^n$. Give a flop count for your algorithm, including all terms that are cubic or quadratic in n. If you know several methods, give the most efficient one (least number of flops for large n).

Solution.

1. LU factorization $A = PLU$ ($(2/3)n^3$).
2. Solve $PLUx_3 = d$ in three steps:
   - Solve $P\hat{x}_3 = d$ (0 flops).
   - Solve $L\tilde{x}_3 = \hat{x}_3$ by forward substitution (n^2 flops).
   - Solve $Ux_3 = \tilde{x}_3$ by backsubstitution (n^2 flops).
3. Compute $\tilde{c} = c - Bx_3$ ($2n^2$ flops) and solve $U^TL^TP^Tx_2 = \tilde{c}$ in three steps:
   - Solve $U^T\hat{x}_1 = \tilde{c}$ by forward substitution (n^2 flops).
   - Solve $L^T\tilde{x}_1 = \hat{x}_1$ by backsubstitution (n^2 flops).
   - Solve $P^Tx_1 = \tilde{x}_1$ (0 flops).
4. Compute $\tilde{b} = b - Bx_2$ ($2n^2$ flops) and solve $PLUx_1 = \tilde{b}$ in three steps as in step 2: ($2n^2$ flops).

Total: $(2/3)n^3 + 10n^2$.

Problem 5 (15 points). Let S be a square matrix that satisifes $S^T = -S$. (This is called a *skew-symmetric* matrix.)

1. Show that $I - S$ is nonsingular. (Hint: first show that $x^T S x = 0$ for all x.)
2. Show that $(I + S)(I - S)^{-1} = (I - S)^{-1}(I + S)$. (This property does not rely on the skew-symmetric property; it is true for any matrix S for which $I - S$ is nonsingular.)
3. Show that the matrix
$$A = (I + S)(I - S)^{-1}$$
is orthogonal.
4. What is the condition number of A?

Solution.

1. We show that $(I - S)x = 0$ implies $x = 0$:
$$(I - S)x = 0 \quad \Longrightarrow \quad x^T(I - S)x = 0 \quad \Longrightarrow \quad x^T x = 0 \quad \Longrightarrow \quad x = 0.$$
In step 3 we used the fact that $x^T S x = (x^T S x)^T = x^T S^T x = -x^T S x$ which is only possible if $x^T S x = 0$.
2. We have (for any matrix S)
$$(I - S)(I + S) = I - S + S - S^2 = (I + S)(I - S).$$
Multiplying with $(I - S)^{-1}$ on both sides gives
$$(I + S)(I - S)^{-1} = (I - S)^{-1}(I + S).$$
3. We show that $A^T A = I$:
$$\begin{aligned} A^T A &= (I - S)^{-T}(I + S)^T(I + S)(I - S)^{-1} \\ &= (I - S^T)^{-1}(I + S^T)(I + S)(I - S)^{-1} \\ &= (I + S)(I - S)^{-1}(I + S)(I - S)^{-1} \\ &= (I - S)^{-1}(I + S)(I + S)^{-1}(I - S) \\ &= I. \end{aligned}$$
4. The condition number of an orthogonal matrix is 1. The norm of an orthogonal matrix is 1:
$$\|A\| = \max_{x \neq 0} \frac{\|Ax\|}{\|x\|} = \max_{x \neq 0} \frac{\sqrt{x^T A^T A x}}{\|x\|} = \max_{x \neq 0} \frac{\sqrt{x^T x}}{\|x\|} = 1.$$
Furthermore $\|A^{-1}\| = \|A^T\| = 1$ because if A is square and orthogonal, A^T is also orthogonal.

Problem 6 (10 points). Very large sets of linear equations

$$Ax = b \tag{3}$$

are sometimes solved using *iterative* methods instead of the standard non-iterative methods (based on LU factorization), which can be too expensive or require too much memory when the dimension of A is large. A simple iterative method works as follows. Suppose A is an $n \times n$ matrix with nonzero diagonal elements. We write A as

$$A = D - B$$

where D is the diagonal part of A (a diagonal matrix with diagonal elements $D_{ii} = A_{ii}$), and $B = D - A$. The equation $Ax = b$ is then equivalent to $Dx = Bx + b$, or

$$x = D^{-1}(Bx + b).$$

The iterative algorithm consists in running the iteration

$$x^{(k+1)} = D^{-1}(Bx^{(k)} + b), \tag{4}$$

starting at some initial guess $x^{(0)}$. The iteration (4) is very cheap if the matrix B is sparse, so if the iteration converges quickly, this method may be faster than the standard LU factorization method.

Show that if $\|D^{-1}B\| < 1$, then the sequence $x^{(k)}$ converges to the solution $x = A^{-1}b$.

Solution. We show that $\|x^{(k)} - x\|$ goes to zero. Subtracting $x = D^{-1}(Bx + b)$ from (4) we have

$$x^{(k+1)} - x = D^{-1}B(x^{(k)} - x)$$

and taking norms on both sides,

$$\|x^{(k+1)} - x\| = \|D^{-1}B(x^{(k)} - x)\| \leq \|D^{-1}B\|\|x^{(k)} - x\|.$$

If $\|D^{-1}B\| < 1$ this shows that $\|x^{(k)} - x\|$ goes to zero.