# Foundations of Computational Math I Exam 1
## In-class Exam
## Open Notes, Textbook, Homework Solutions Only
## Calculators Allowed
## No collaborations with anyone
## Friday 26 October, 2012

| Question | Points Possible | Points Awarded |
|---|---|---|
| 1. Basic Floating Point | 25 | |
| 2. Basic Analysis | 25 | |
| 3. Factorization | 30 | |
| 4. Backward Error | 20 | |
| Total Points | 100 | |

**Name:**_____

**Alias:** _____

**to be used when posting anonymous grade list.**

# Problem 1 (25 points)

Each question below has a brief answer and justification.

**1.a (5 points)** Explain the idea of a "hidden bit" in a floating point system with base $\beta = 2$ and the benefit achieved by using it.

**Solution:**

The hidden bit is taken to be a 1 for normalized floating point numbers in an FP system with $\beta = 2$ and 0 for subnormal floating point numbers. The benefit is that it allows $t$ bits of precision while actually storing only $t - 1$.

It does not work with $\beta \neq 2$ since in that case there is more than one value for the first digit in a normalized FP number, i.e., $0, 1, \ldots, \beta - 1$.

**1.b (10 points)** Consider a floating point representation system with base $\beta = 8$, precision $t = 12$ and exponent range determined by $L = -15$ and $U = 16$. How many bits are required to represent a floating point number in this system?

**Solution:**

- Each digit requires 3 bits since $\beta = 8$. The mantissa therefore requires $3t = 36$ bits.
- A sign bit is required.
- The exponent range contains 32 integers, therefore, 5 bits are required in the exponent field.

This yields $36 + 5 + 1 = 42$ bits. Since $\beta = 8$ hidden bit normalization is not applicable.

**1.c (5 points)** Suppose $x$, $y$ and $z$ are floating point numbers in a standard model floating point arithmetic system. Does the standard model imply that floating point arithmetic is associative, i.e.,

$$(x \boxed{op} (y \boxed{op} z)) = ((x \boxed{op} y) \boxed{op} z) \ ?$$

**Solution:**

Standard model floating point is not associative. The example of cancellation from the notes is a required counterexample:

$$x = 472635.0000 \quad y = 27.5013 \quad z = -472630.0000$$
$$fl(fl(472635.0000 + 27.5013) - 472630) = 33$$
$$fl(27.5013 + fl(472635 - 472630)) = 32.5013$$

**1.d (5 points)** A communications laboratory has a signal processor that computes with precision high enough to satisfy easily all of the requirements by applications of interest to the laboratory. Suppose for one of the applications the data must be collected using sensors that measure the appropriate signals in the environment and the resulting problem to be solved using the data has a condition number of $\kappa \approx 10^5$.

If at least 2 decimal digits of accuracy in the solution are required, how many decimal digits must the sensors accurately measure in the input signals?

**Solution:** The relative forward error is related to the uncertainty in the problem data by the condition number, i.e.,

$$\frac{|f(d+e) - f(d)|}{|f(d)|} \leq \kappa \frac{|e|}{|d|}$$

For the forward error to be smaller than $10^{-2}$ with $\kappa = 10^5$ we need the uncertainty in the data to be less than $10^{-7}$.

# Problem 2 (25 points)

## 2.a (10 points)

Suppose $A \in \mathbb{R}^{m\times n}$ and consider the matrix 2-norm

$$\|A\|_2 = \max_{\|x\|_2=1} \|Ax\|_2$$

Show that $\|A\|_2 \geq \|A_1\|_2$ where

$$A = \begin{pmatrix} A_1 \\ A_2 \end{pmatrix},$$

$m = m_1 + m_2$, $A_1 \in \mathbb{R}^{m_1\times n}$, and $A_2 \in \mathbb{R}^{m_2\times n}$.

**Solution:**

There are at least three ways to prove this using material from the class.

The first is based on the discussion of structural orthogonality and the relationship between orthogonality and inner product-based vector norms. Recall that $\forall y \in \mathbb{R}^m$, we have

$$y = \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} \rightarrow \|y\|_2^2 = \|y_1\|_2^2 + \|y_2\|_2^2$$
$$\therefore \forall x \in \mathbb{R}^n \quad \|A_1 x\|_2^2 \leq \|A_1 x\|_2^2 + \|A_2 x\|_2^2$$

It follows that

$$\|A\|_2^2 = \max_{\|x\|_2=1} \|Ax\|_2^2 = \max_{\|x\|_2=1} \{\|A_1 x\|_2^2 + \|A_2 x\|_2^2\} \geq \max_{\|x\|_2=1} \|A_1 x\|_2^2 = \|A_1\|_2^2.$$

Therefore, the conclusion

$$\|A\|_2^2 \geq \|A_1\|_2^2$$

is proven.

The second approach uses the fact that there exists $x_1 \in \mathbb{R}^n$ such that

$$\|A_1 x_1\|_2 = \|A_1\|_2.$$

Therefore,

$$\|A\|_2^2 \geq \|Ax_1\|_2^2 = \|A_1 x_1\|_2^2 + \|A_1 x_1\|_2^2 \geq \|A_1 x_1\|_2^2 = \|A_1\|_2^2$$

Note this proof does not work if you start with $z \in \mathbb{R}^n$ such that $\|Az\|_2 = \|A\|_2$.

A third and more elegant proof uses the consistency of the family of matrix 2-norms. Define

$$A = \begin{pmatrix} A_1 \\ A_2 \end{pmatrix} \quad B = \begin{pmatrix} I_{m_1} & 0 \end{pmatrix}$$

Note that it is easily seen from the definition of the matrix 2-norm and $B$ that

$$\|B\|_2 = 1$$

We therefore have

$$A_1 = BA \rightarrow \|A_1\|_2 = \|BA\|_2 \leq \|B\|_2 \|A\|_2 = \|A\|_2$$

as desired.

## 2.b (15 points)

Consider the vector space $\mathbb{R}^3$ and the subspace

$$\mathcal{S} = span[v_1, \ v_2]$$
$$v_1 = \begin{pmatrix} 1 \\ -1 \\ 0 \end{pmatrix}, \quad v_2 = \begin{pmatrix} 1 \\ 0 \\ -1 \end{pmatrix}$$

(i) **(5 points)** Show that $v_1$ and $v_2$ are linearly independent and a basis for $\mathcal{S}$.

**Solution:** The linear independence is evident from the corresponding positions of 0 and $-1$ in $v_1$ and $v_2$. To prove this formally we must show that if $Vx = y$, where $x \in \mathbb{R}^2$, $y \in \mathbb{R}^3$, and $V = \begin{pmatrix} v_1 & v_2 \end{pmatrix}$, then $x, y$ is a unique pair.

We have

$$Vx = y$$
$$\begin{pmatrix} 1 & 1 \\ -1 & 0 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} \xi_1 \\ \xi_2 \end{pmatrix} = \begin{pmatrix} \eta_1 \\ \eta_2 \\ \eta_3 \end{pmatrix} \Leftrightarrow \begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} \xi_1 \\ \xi_2 \end{pmatrix} = \begin{pmatrix} \eta_2 \\ \eta_3 \end{pmatrix}, \quad \eta_1 = \xi_1 + \xi_2$$

Since the $2 \times 2$ matrix is nonsingular the pair is unique.

(ii) **(10 points)** Derive from the vectors $\{v_1, \ v_2\}$ a basis $\{q_1, \ q_2\}$ of the subspace $\mathcal{S}$ where the vectors $q_1$ and $q_2$ are orthonormal vectors. Justify your answer using reasoning based on material presented in the lectures and notes.

**Solution:** It is clear that $v_1$ and $v_2$ are not orthogonal since

$$v_1^T v_2 = 1$$

We can use the notion of projection and the unique decomposition of a vector into two pieces give an subspace that we have discussed relative to linear least squares.

Taking $\mathcal{S} = \mathcal{R}(v_1)$ we have that $q_1 = v_1/\|v_1\|_2$ is an orthonormal basis for $\mathcal{S}$. $q_2 \perp \mathcal{S}$ can be found by projecting $v_2$ onto $\mathcal{S}^\perp$, i.e.,

$$\tilde{v}_2 = v_2 - q_1 q_1^T v_2, \quad q_2 = \frac{\tilde{v}_2}{\|\tilde{v}_2\|_2}$$

This is equivalent to applying the results of the homework problem that considered the least squares approximation of one vector, $x$, by another, $y$, and produced an orthogonal decomposition of $x$, i.e.,

$$min_\alpha \|x - \alpha y\|_2, \quad x = z + \alpha_{min} y, \quad \alpha_{min} = y^T x / y^T y$$

We have therefore

$$\tilde{v}_2 = v_2 - \frac{v_2^T v_1}{v_1^T v_1} v_1 = \begin{pmatrix} 1/2 \\ 1/2 \\ -1 \end{pmatrix}$$

$$\tilde{v}_2^T v_1 = 0$$

and

$$q_1 = \frac{1}{\|v_1\|_2} v_1 = \begin{pmatrix} 1/\sqrt{2} \\ -1/\sqrt{2} \\ 0 \end{pmatrix}$$

$$q_2 = \frac{1}{\|\tilde{v}_2\|_2} \tilde{v}_2 = \frac{1}{\sqrt{3/2}} \begin{pmatrix} 1/2 \\ 1/2 \\ -1 \end{pmatrix} = \begin{pmatrix} 1/\sqrt{6} \\ 1/\sqrt{6} \\ -\sqrt{2/3} \end{pmatrix}$$

We then have the desired results

$$q_1^T q_2 = 0, \quad q_1^T q_1 = q_2^T q_2 = 1, \quad \mathcal{S} = span[q_1, \ \ q_2] = span[v_1, \ \ v_2]$$

This is a simple example of a more general procedure called the Gram-Schmidt method to produce an orthonormal basis from a basis that is not orthonormal.

Note that you could also use the technique based on Householder reflectors presented for solving least squares problems to produce and orthogonal basis.

# Problem 3 (30 points)

Consider $S \in \mathbb{R}^{n \times n}$ whose nonzero elements have the following pattern for $n = 8$:

$$S = \begin{pmatrix} 1 & 0 & 0 & 0 & \mu_1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & \mu_2 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & \mu_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & \alpha & \beta & 0 & 0 & 0 \\ 0 & 0 & 0 & \gamma & \delta & 0 & 0 & 0 \\ 0 & 0 & 0 & \delta_1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & \delta_2 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & \delta_3 & 0 & 0 & 0 & 1 \end{pmatrix}$$

**3.a (10 points)** We have considered several basic transformations ( Gauss transforms, Gauss-Jordan transforms, elementary permutations, Householder reflectors) that can be used to compute factorizations efficiently. Assume that $S$ is diagonally dominant (both row-wise and column-wise).

Using whatever combination of these transformations you think appropriate, describe an algorithm to compute stably a factorization of $S$ that can be used to solve $Sx = b$. **Your algorithm should be designed to require as few computations as possible.** Your solution must include a description of how you exploit the structure of the matrix and its factors.

**3.b (10 points)** Assume that you have the factorization of $S$ defined by your algorithm from Part (3.a), describe an algortihm to solve $Sx = b$. **Your algorithm should be designed to require as few computations as possible.** Your solution must include a description of how you exploit the structure of the matrix and its factors.

**3.c (5 points)** Determine the order of computational complexity, i.e., give $k$ in $O(n^k)$, when your factorization algorithm is applied to a matrix of any dimension $n$.

**3.d (5 points)** Determine the order of computational complexity, i.e., give $k$ in $O(n^k)$, when your algorithm to solve $Sx = b$ given the factorization is applied to a matrix of any dimension $n$.

**Solution:**

Consider $P^T S P$ where

$$S = \begin{pmatrix} 1 & 0 & 0 & 0 & \mu_1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & \mu_2 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & \mu_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & \alpha & \beta & 0 & 0 & 0 \\ 0 & 0 & 0 & \gamma & \delta & 0 & 0 & 0 \\ 0 & 0 & 0 & \delta_1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & \delta_2 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & \delta_3 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$P = \begin{pmatrix} e_1 & e_2 & e_3 & e_6 & e_7 & e_8 & e_4 & e_5 \end{pmatrix}$$

$$\tilde{S} = P^T S P = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & \mu_1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & \mu_2 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & \mu_3 \\ 0 & 0 & 0 & 1 & 0 & 0 & \delta_1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & \delta_2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & \delta_3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \alpha & \beta \\ 0 & 0 & 0 & 0 & 0 & 0 & \gamma & \delta \end{pmatrix}$$

$\tilde{S} = P^T S P$ is a block upper triangular matrix with 1 in $n-2$ of the diagonal elements and a trailing $2 \times 2$ diagonal block

$$M = \begin{pmatrix} \alpha & \beta \\ \gamma & \delta \end{pmatrix}$$

Therefore,$\tilde{S}$ is nonsingular if and only if $M$ is nonsingular. The same follows for $S$.

The factorization algorithm to solve $Sx = b$ is therefore simply factoring $M$. This is clearly $O(1)$ and the best that can be done.

We have the factorization for $n = 8$ that clearly generalizes to any $n$:

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & \mu_1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & \mu_2 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & \mu_3 \\ 0 & 0 & 0 & 1 & 0 & 0 & \delta_1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & \delta_2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & \delta_3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \alpha & \beta \\ 0 & 0 & 0 & 0 & 0 & 0 & \gamma & \delta \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \gamma/\alpha & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & \mu_1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & \mu_2 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & \mu_3 \\ 0 & 0 & 0 & 1 & 0 & 0 & \delta_1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & \delta_2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & \delta_3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \alpha & \beta \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \omega \end{pmatrix}$$

The $2 \times 2$ forward and backward solve is $O(1)$ in complexity. Computing the remaining $n-2$ elements of $P^T x$ requires a multiplication and addition for each giving a complexity of

$$2n + O(1)$$

for the solution procedure given the factorization and, in fact, in this case for the entire procedure.

The use of permutations here results in the familiar $LU$ factorization of a permuted matrix. The algorithm does not, however, require the permutation. $\gamma$ can be eliminated in its original position with a single transformation involving only two rows of the matrix and therefore two elements of the righthand-side vector during the solution process.

$$TSx = Tb$$

$$\begin{pmatrix} 1 & 0 & 0 & 0 & \mu_1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & \mu_2 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & \mu_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & \alpha & \beta & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \omega & 0 & 0 & 0 \\ 0 & 0 & 0 & \delta_1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & \delta_2 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & \delta_3 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \xi_1 \\ \xi_2 \\ \xi_3 \\ \xi_4 \\ \xi_5 \\ \xi_6 \\ \xi_7 \\ \xi_8 \end{pmatrix} = \begin{pmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \\ \beta_4 \\ \beta_5' \\ \beta_6 \\ \beta_7 \\ \beta_8 \end{pmatrix}$$

The solution $x$ is determined by solving the scalar equation for $\xi_5$ then recovering the rest of the solution in $O(n)$ computations as before. It is easily seen that this is essentially the same algorithm as above but without the permutation the factorization is a very simple unit lower triangular matrix times a matrix with a pattern that is essentially the same as $S$. This factorization is related to the so-called Spike factorization algorithm used as the basis for a well-known parallel algorithm for banded systems.

Note that using $LU$ factorization on the original form of $S$ causes fill-in to appear but the complexity stays $O(n)$. This can be seen after one step (assume pivoting is not needed for stability) that eliminates $\gamma, \delta_1, \ldots, \delta_3$.

$$S = \begin{pmatrix} 1 & 0 & 0 & 0 & \mu_1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & \mu_2 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & \mu_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & \alpha & \beta & 0 & 0 & 0 \\ 0 & 0 & 0 & \gamma & \delta & 0 & 0 & 0 \\ 0 & 0 & 0 & \delta_1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & \delta_2 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & \delta_3 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$M_4^{-1}S = \begin{pmatrix} 1 & 0 & 0 & 0 & \mu_1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & \mu_2 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & \mu_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & \alpha & \beta & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \delta' & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \omega_1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & \omega_2 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & \omega_3 & 0 & 0 & 1 \end{pmatrix}$$

On the next step, the $\omega_i$ must be eliminated but this does not cause fill-in.

$$M_5^{-1}M_4^{-1}S = \begin{pmatrix} 1 & 0 & 0 & 0 & \mu_1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & \mu_2 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & \mu_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & \alpha & \beta & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \delta' & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

so the algorithm remains $O(n)$ as desired but slightly more work is performed for the factorization and the solution phases.

Applying Householder reflectors to $S$ starting with eliminating column 4 below the diagonal causes the trailing part of the matrix to fill in completely in general therefore requiring the subsequent reduction of a dense matrix which yields and $O(n^3)$ computation. This can be seen by noting that a relector designed to perform

$$H \begin{pmatrix} \alpha \\ \gamma \\ \delta_1 \\ \delta_2 \\ \delta_3 \end{pmatrix} = \begin{pmatrix} \alpha' \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

has the form

$$H = I - 2uu^T$$

where $u$ is a dense vector in general. Therefore

$$H \begin{pmatrix} \beta & 0 & 0 & 0 \\ \delta' & 0 & 0 & 0 \\ \omega_1 & 1 & 0 & 0 \\ \omega_2 & 0 & 1 & 0 \\ \omega_3 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} * & * & * & * \\ * & * & * & * \\ * & * & * & * \\ * & * & * & * \\ * & * & * & * \end{pmatrix}$$

# Problem 4 (20 points)

The following backward error lemma for finite precision computation of a matrix vector product is true. **You do not have to prove it.**

**Lemma 4.1.** *If $M \in \mathbb{R}^{n\times n}$ and $x \in \mathbb{R}^n$ then the matrix vector product $\hat{y} \in \mathbb{R}^n$ computed in finite precision satisfies*

$$\hat{y} = (M + \Delta M)x$$
$$|\Delta M| \leq \omega_n |M|$$
$$\omega_n = \frac{nu}{1 - nu}$$

Let $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times n}$, and $C \in \mathbb{R}^{n\times n}$ with $C = AB$. Consider the matrix $\hat{C} \in \mathbb{R}^{n\times n}$ that is the product $AB$ computed in finite precision.

**4.a (10 points)** Use the lemma to characterize the error $|C - \hat{C}|$.

**4.b (10 points)** With no further assumptions, does your characterization say that the computation of a matrix product backward stable? Justify your answer.

**Solution:** Recall that matrix multiplication can be expressed in terms of multiple matrix vector products. We have

$$\hat{c}_j = (A + E_j)b_j$$
$$|E_j| \leq \omega_n |A|$$
$$\therefore |C - \hat{C}| \leq \omega_n |A||B|$$

In general, we do not have backward stability since each column is backward stable with a different perturbation, i.e., we cannot find $\Delta A$ and $\Delta B$ such that

$$\hat{C} = (A + \Delta A)(B + \Delta B).$$

We must add other assumptions such as a nonsingular $B$ or $A$ to get a backward bound.

If, for example, $B$ is nonsingular then we have

$$\hat{C} = C + \hat{E} = AB + \hat{E}$$
$$\hat{C}B^{-1} = A + \hat{E}B^{-1}$$
$$\hat{C} = (A + \hat{E}B^{-1})B$$

which is a backward error as desired.