# CPSC 302 Final Solution

Uri Ascher

October 2001

## Question 1

1. For the first method we have $n$ subtractions, $n-1$ additions and $n+1$ multiplications/divisions fot a total of $3n$ flops.

   For the second method we have $n+2$ multiplications/divisions and $n-1$ additions, but only one subtraction for a total of $2n+2$ flops.

   The **second method** is therefore cheaper.

2. The roundoff error accumulated when summing up nonnegative numbers is proportional to their size. A difference in accuracy may arise when the values $|x_i - \bar{x}| \ll |x_i|$. Then the **first method** is more accurate.

   Example: suppose $x_i = \bar{x} + \delta_i$, where $\delta_i$ are so small that in the precision used $\bar{x}^2 + \delta_i^2 = \bar{x}^2$ while $\bar{x} + \delta_i = x_i$ is calculated accurately. Then using method (1) we get

   $$s^2 = \frac{1}{n}\sum_{i=1}^{n} \delta_i^2 > 0$$

   which is correct, whereas using method (2) we get

   $$\begin{aligned} \sum x_i^2 &= \sum (\bar{x} + \delta_i)^2 = \sum (\bar{x}^2 + 2\delta_i \bar{x} + \delta_i^2) \\ &= n\bar{x}^2. \\ \Rightarrow\ s^2 &= 0, \end{aligned}$$

   which is incorrect.

## Question 2

a) A floating point number is assumed to be represented as $y = \pm a \times 2^e$, where $e$ is an exponent and $0.5 \le a < 1$. Here it also assumed that $y > 0$. Thus,

$$y^{1/3} = a^{1/3} \times (2^{1/3})^e.$$

We can write $e = 3\hat{e} + j$ where $j = 0,\ 1$ or $2$. Then

$$y^{1/3} = [a^{1/3} \cdot (2^{1/3})^j] \times 2^{\hat{e}}.$$

So, upon storing the constants $2^{1/3}$ and its square we can get $y^{1/3}$ from $a^{1/3}$ in at most 3 flops. The range of the numbers to be considered below is that of the fraction $a$.

b) Define $f(x) = x^3 - a$. Then the equation $f(x) = 0$ obviously has the required root. Now, $f'(x) = 3x^2$, so the Newton iteration is

$$x_{k+1} = x_k - \frac{x_k^3 - a}{3x_k^2} = \frac{2x_k^3 + a}{3x_k^2}, \qquad k = 0, 1, 2, \ldots.$$

The flop count (recognizing $x^3 = x^2 \cdot x$) is 6 operations.

Bonus: write the iteration as

$$x_{k+1} = \frac{2}{3}x_k + \frac{a/3}{x_k^2}.$$

Compute constants $\frac{2}{3}$ and $a/3$ once and store them. This then yields per iteration 4 operations.

c) The range of $a$ is so small that the initial guess does not matter so much. Choose, e.g., $x_0 = 0.9$. Then certainly $|x^* - x_0| \le .25 = 2^{-2}$. Now,

$$2^{-52} \le |x^* - x_k| \le M|x^* - x_{k-1}|^2 \le \cdots \le M^k(.25)^{2^k}.$$

So, roughly, $2^k = 21$, hence $k \approx 5$.

## Question 3

a) Using Jacobi we get $\mathbf{x}_1 = 1.2(1,1,1)^T$, $\mathbf{e}_1 = 0.2(1,1,1)^T$, $\|\mathbf{e}_1\|_1 = 0.6$; then $12 - 1.2 - 1.2 = 9.6$, so $\mathbf{x}_2 = 0.96(1,1,1)^T$, $\mathbf{e}_2 = 0.04(1,1,1)^T$, $\|\mathbf{e}_2\|_1 = 0.12$.

Using Gauss-Seidel we get $x_1^1 = 1.2$, $x_2^1 = (12 - 1.2)/10 = 1.08$, $x_3^1 = (12 - 1.2 - 1.08)/10 = 0.972$. So, $\mathbf{e}_1 = (.2, .08, -.028)^T$, and $\|\mathbf{e}_1\|_1 = 0.308$. The Gauss-Seidel iteration seems faster.

b) We have to check the norm of $T = I - M^{-1}A$, with

$$A = \begin{pmatrix} 10 & 1 & 1 \\ 1 & 10 & 1 \\ 1 & 1 & 10 \end{pmatrix}, \quad M = \begin{pmatrix} 10 & 0 & 0 \\ 0 & 10 & 0 \\ 0 & 0 & 10 \end{pmatrix}.$$

We obtain

$$T = -0.1 \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix}$$

and, since $\|T\|_\infty = 0.2 < 1$ convergence is guaranteed.

c) Now we have $\mathbf{x}_1 = 6(1,1,1)^T$, $\mathbf{x}_2 = -24(1,1,1)^T$. The error obviously grows rapidly.

Here

$$T = -2.5 \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix}.$$

Clearly in all norms we can quickly assess $\|T\| > 1$. But this is not a proof of divergence. For the latter we need to show that $\rho(T) > 1$.

Note that $\lambda = 2.5$ is an eigenvalue because the matrix $\lambda I - T$ is then clearly singular (consisting of repetitions of one value). Hence, $\rho(T) \geq 2.5 > 1$, proving divergence.

## Question 4

Multiplying $A\mathbf{x} = \mathbf{b}$ by $T$ we have

$$T\mathbf{b} = TA\mathbf{x} = LU\mathbf{x}.$$

So for $T\mathbf{b}$ we can apply forward and backward substitutions. Our algorithm is:

1. Form $\hat{\mathbf{b}} = T\mathbf{b}$.
2. Solve $L\mathbf{y} = \hat{\mathbf{b}}$ for $\mathbf{y}$.
3. Solve $U\mathbf{x} = \mathbf{y}$ for $\mathbf{x}$.

The multiplication $T\mathbf{b}$ costs about $2n^2$ flops, whereas forward and backward substitutions cost about $n^2$ flops each. The total operation count is therefore $4n^2 + O(n) = O(n^2)$.

In detail:

1.

$$\begin{aligned} &for\ i = 1:n \\ &\hat{b}_i = \sum_{j=1}^{n} t_{ij} b_j \end{aligned}$$

2.

$$\begin{aligned} &for\ i = 1:n \\ &y_i = \hat{b}_i - \sum_{j=1}^{i-1} l_{ij} y_j \end{aligned}$$

3.

$$\begin{aligned} &for\ i = n:-1:1 \\ &x_i = \frac{y_i - \sum_{j=i+1}^{n} u_{ij} x_j}{u_{ii}} \end{aligned}$$