CSCI 5304 HW 2

Jingxiang Li

October 6, 2014

Problem 1

Consider the two mathematically equivalent formulas

1. $a = x^2 - y^2$
2. $a = (x + y) * (x - y)$

Problem a

Compute both formulas in 2 digit decimal arithmetic using $x = 11$, $y = 10$. Which formula gives the best answer? Ignore any limits on the exponent.

Solution

$x = .11 \times 10^2$, $y = .10 \times 10^2$

1. $x^2 = .0121 \times 10^4 = .12 \times 10^3$, $y^2 = .0100 \times 10^4 = .10 \times 10^3$,
   $a = x^2 - y^2 = .02 \times 10^3 = .20 \times 10^2$
2. $x + y = .21 \times 10^2$, $x - y = .01 \times 10^2 = .10 \times 10^1$,
   $a = (x + y) \times (x - y) = .21 \times 10^2 * .10 \times 10^1 = .21 \times 10^2$

Note that formula 2 gives the best answer.

Problem b

Compute both formulas in 3-bit binary arithmetic using $x = 3/2$ (dec) and $y = 1$. Again, which formula gives the most accurate answer? Ignore any limits on the exponent.

Solution

$x = .110 \times 2^1$, $y = .100 \times 2^1$

1. $x^2 = .100100 \times 2^2 = .100 \times 2^2$, $y^2 = .010000 \times 2^2 = .100 \times 2^1$,
   $a = x^2 - y^2 = .100 \times 2^2 - .0100 \times 2^2 = .010 \times 2^2 = .100 \times 2^1$
2. $x + y = 1.010 \times 2^1 = .101 \times 2^2$, $x - y = .010 \times 2^1 = .100 \times 2^0$,
   $a = (x + y) \times (x - y) = .010100 \times 2^2 = .101 \times 2^1$

Note that formula 2 gives the best answer.

Problem c

In your favorite programming language and platform, find two numbers x, y such that the two formulas above give different answers. Which answer is more accurate. Report the precision used, the machine epsilon (unit round-off) and the general description of your machine. Can you find two numbers for which one of the answers has no accuracy whatsoever, but the other is almost OK? Note: in Matlab all arithmetic is in double precision, but you can force single precision by using the single function: `a = single(5/4)` forces all arithmetic involving a to be on single precision.

Solution

Progarmming Language: Octave 64-bit
Precision: Double Precision
epsilon: 2^{-52}
Machine: Intel Core i5-3210M CPU @ 2.50GHz×4, 64-bit

Let $x = \frac{1}{2} + 4 \times \text{eps}$, $y = \frac{1}{2}$

```
x = 1/2 + 4 * eps;
y = 1/2;
format hex;
a1 = x^2 - y^2;
a2 = (x + y) * (x - y);
a1
> a1 = 3cb0000000000000
a2
> a2 = 3cd0000000000004
```

Note that the second formula has more accuracy.

Problem 2

Consider the system

$$Ax \equiv \begin{pmatrix} -0.001 & 1.001 \\ 0.001 & -0.001 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \equiv b$$

whose solution is $x_1 = x_2 = 1$ and the system

$$(A + \Delta A)y = b + \Delta b$$

where $\Delta A = \epsilon|A|$, $\Delta b = \epsilon|b|$. Here $|A|$ means take the absolute value of all the elements individually. In the following we let $\epsilon = 10^{-4}$.

Problem a

Compute $k_\infty(A)$. Compute the actual value of $||x - y||_\infty / ||x||_\infty$ and its estimate obtained from using the (standard) condition number $k_\infty(A)$ (Theorem 2 in notes).

Which is quite far from the actual error.

Solution

At first we calculate the actual value using Octave.

```
A = [-0.001, 1.001; 0.001, -0.001];
b = [1; 0];
epsilon = 1e-04;

A_new = A + epsilon .* abs(A);
b_new = b + epsilon .* abs(b);
y = inv(A_new) * b_new;
x = [1; 1];
norm(x - y, inf) / norm(x, inf)
> 2.0038e-04
```

As the result, the actual error is $2.0038\mathrm{e} - 04$

Then we calculate its estimate using the following formula

$$\frac{||x - y||_\infty}{||x||_\infty} \leq \frac{2\epsilon}{1 - \rho} |||A^{-1}||A|||_\infty$$

where $\rho = \epsilon k_\infty(A)$

```
rho = epsilon * cond(A, inf);
2 * epsilon / (1 - rho) * norm(abs(A) * abs(inv(A)), inf)
> 6.6785e-04
```

Note that the estimation of error bound is $6.6785\mathrm{e} - 04$

Problem b

Now repeat the above, but this time use $\Delta(b) = \epsilon \begin{pmatrix} 0 \\ 1 \end{pmatrix}$

Solution

At first we calculate the actual value using Octave.

```
A = [-0.001, 1.001; 0.001, -0.001];
b = [1; 0];
epsilon = 1e-04;

A_new = A + epsilon .* abs(A);
b_new = b + epsilon .* abs([0; 1]);
y = inv(A_new) * b_new;
x = [1; 1];
norm(x - y, inf) / norm(x, inf)
> 0.099790
```

As the result, the actual error is 0.099790

Then we calculate its estimate

```
norm(inv(A), inf) * norm(A, inf) / ...
(1 - norm(inv(A), inf) * norm(epsilon * abs(A), inf)) *...
(norm(epsilon * [0;1], inf) / norm(b, inf) + norm(epsilon * abs(A), inf) /...
norm(A, inf))
> 0.22321
```

Note that the estimation of error bound is 0.22321

Problem 3

Problem a

Show that the following matrix is singular

$$A = \begin{pmatrix} 1 & 2 & -1 \\ 2 & 1 & 1 \\ -1 & 1 & -2 \end{pmatrix}$$

Solution

Let A_i be the ith column of matrix A, then we have $A_1 - A_2 - A_3 = 0$, suggesting $\text{rank}(A) = 2 < 3$, so that A is singular.

Problem b

What is the range or column space of A ? What is its null space? Give a basis for each subspace.

Solution

Since $\text{rank}(A) = 2$, the column space of A should be the linear combination of any two columns of A, which is $\{x : x = \alpha A_1 + \beta A_2, \alpha, \beta \in \mathbb{R}\}$, where $A_1 = \begin{pmatrix} 1 \\ 2 \\ -1 \end{pmatrix}$, $A_2 = \begin{pmatrix} 2 \\ 1 \\ 1 \end{pmatrix}$.

The basis of the column space of A can be $\{A_1, A_2\}$

Then, the null space will be $\{x : x = \alpha \begin{pmatrix} -1 \\ 1 \\ 1 \end{pmatrix}, \alpha \in \mathbb{R}\}$ the basis of the null space of A can be $\{\begin{pmatrix} -1 \\ 1 \\ 1 \end{pmatrix}\}$

Problem c

Consider the matrix B obtained from A by adding $\eta = 0.001$ to the entry (1, 3) (So $B = A + \eta e_1 e_3^T$). Without computing the inverse of B, show that $||B^{-1}||_1 \geq 3,000$.

Solution

Let $x = \begin{pmatrix} -1 \\ 1 \\ 1 \end{pmatrix}$, then we have $Bx = \begin{pmatrix} -0.001 \\ 0 \\ 0 \end{pmatrix}$, then we can have the lower bound of

$$||B^{-1}||_1 \geq \frac{||x||_1}{||Bx||_1} = 3,000$$

Q.E.D.

Problem d

Find a lower bound for the condition number $k_1(B)$.

Solution

$$k_1(B) = ||B||_1 \cdot ||B^{-1}||_1 \geq 4 \times 3,000 = 12,000$$

Problem 4

Consider the $n \times n$ matrix

$$A_n = \begin{pmatrix} 1 & & & & & \\ -2 & 1 & & & & \\ & -2 & 1 & & & \\ & & \ddots & \ddots & & \\ & & & \ddots & \ddots & \\ & & & & -2 & 1 \end{pmatrix}$$

What is A_n^{-1}? [Hint: Write $A_n = I - E_n$ and use expansion $I + E + E^2 + \cdots$.]

Calculate the condition numbers $k_1(A_n)$ and $k_\infty(A_n)$. Verify your results with matlab for the case $n = 10$.

Solution

Note that we can rewrite $A_n = I - E$ where

$$E = \begin{pmatrix} 0 & & & & & \\ 2 & 0 & & & & \\ & 2 & 0 & & & \\ & & \ddots & \ddots & & \\ & & & \ddots & \ddots & \\ & & & & 2 & 0 \end{pmatrix}$$

It's easy to see that $E^n = 0$ and

$$(I - E) \cdot (I + E + E^2 + \cdots + E^{n-1}) = I - E^n = I$$

So that

$$A_n^{-1} = (I + E + E^2 + \cdots + E^{n-1})$$

Then we verify the result using Octave for the case $n = 10$

```
n = 10;
A = eye(n);
for i = 1 : n
    for j = 1 : n
        if (i - j == 1)
            A(i, j) = -2;
        end
    end
end
E = -(A - eye(n));

result = zeros(n);
```

```
for i = 0 : (n - 1)
    result = result + E ^ i;
end

%% A_inv is the inverse matrix of A obtained from formula above

A_inv = result
>
     1     0     0     0     0     0     0     0     0     0
     2     1     0     0     0     0     0     0     0     0
     4     2     1     0     0     0     0     0     0     0
     8     4     2     1     0     0     0     0     0     0
    16     8     4     2     1     0     0     0     0     0
    32    16     8     4     2     1     0     0     0     0
    64    32    16     8     4     2     1     0     0     0
   128    64    32    16     8     4     2     1     0     0
   256   128    64    32    16     8     4     2     1     0
   512   256   128    64    32    16     8     4     2     1

%% Verify A_inv is the inverse of A.
%% If so, A * A_inv should be the identity matrix.

A * A_inv
>
     1    0    0    0    0    0    0    0    0    0
     0    1    0    0    0    0    0    0    0    0
     0    0    1    0    0    0    0    0    0    0
     0    0    0    1    0    0    0    0    0    0
     0    0    0    0    1    0    0    0    0    0
     0    0    0    0    0    1    0    0    0    0
     0    0    0    0    0    0    1    0    0    0
     0    0    0    0    0    0    0    1    0    0
     0    0    0    0    0    0    0    0    1    0
     0    0    0    0    0    0    0    0    0    1
```