**CS 421:** Fall 1997

# Solutions to Assignment 2

1. Column oriented upper triangular solver:

```
function x=UTriSol(U,b)
% x=UTriSol(U,b)
% Column oriented upper triangular solve.
n=size(U,2);
x=zeros(n,1);
for j=n:-1:1
    x(j) = b(j)/U(j,j);
    b(1:j-1) = b(1:j-1)-x(j)*U(1:j-1,j);
end
```

Testing code:

```
% Script for testing UTriSol
fprintf('  n    2-norm of    relative      time\n')
fprintf('       residual      error        (sec)\n\n')
for n=10:10:100
    U=triu(rand(n,n));    % Random upper triangular matrix
    z=rand(n,1);          % Random known solution
    b=U*z;                % Generate right hand side
    tic
    x=UTriSol(U,b);
    t=toc;
    res_norm=norm(b-U*x);
    rel_error=norm(x-z)/norm(z);
    fprintf('%4d %12.4e %12.4e %9.5f\n',n,res_norm,rel_error,t)
end
```

This script gave the following output on a Sparc 1:

```
  n    2-norm of    relative      time
       residual      error        (sec)

  10   0.0000e+00   8.2150e-15   0.04651
  20   8.8818e-16   5.5166e-13   0.10648
  30   8.8818e-16   1.0719e-13   0.14024
  40   3.6078e-15   4.0158e-08   0.19118
  50   4.6788e-15   5.6675e-07   0.24266
  60   4.1954e-15   3.4937e-08   0.29251
  70   7.9005e-15   1.1356e-08   0.38617
  80   8.9509e-15   9.1389e-06   0.40162
  90   4.4588e-14   3.3018e+02   0.46407
 100   1.3594e-14   2.4183e+01   0.52461
```

2. Let $y$ be an approximate solution to $Ax = b$. For the following, we need to assume that $y \neq 0$.

   (a) Let $r = b - Ay$ and $E = ry^T/(y^Ty)$. Then

   $$(A+E)y = (A + \frac{ry^T}{y^Ty})y = Ay + \frac{ry^Ty}{y^Ty} = Ay + r = b.$$

   (b) Note that $(A+F)y = b \Leftrightarrow Fy = r$. There are several ways of constructing matrices $F$ different from $E$ and such that $Fy = r$. Here are few:

      i. Let $F = rz^T/(z^Ty)$ where $z \neq 0$. Then $Fy = r$ and $F \neq E$ if $z$ is not a multiple of $y$. This gives infinetly many rank-1 choices for $F$.

      ii. If all $y_i \neq 0$ can choose $F = \text{diag}([r_1/y_1, \ldots, r_n/y_n])$. If some $y_i = 0$, then can choose $F$ to be equal to $E$, except we arbitrarly change the $i$-th column.

      iii. Let $f_j$ denote the $j$-th column of $F$. We want to choose the columns such that $\sum_{j=1}^n y_j f_j = r$. At least one of the components of the vector $y$ is nonzero, say $y_k \neq 0$. Then we can choose $f_j$ for $j \neq k$ to be arbitrary and set $f_k = (r - \sum_{j\neq k} y_j f_j)/y_k$. This gives an $F$ such that $Fy = r$ and shows that there are infinitely many $F$ satisfying $(A+F)y = b$.

   (c) First compute the 2-norm of $E$. For all $z \neq 0$,

   $$||Ez||_2 = ||\frac{ry^Tz}{y^Ty}||_2 = \frac{|y^Tz|}{y^Ty}||r||_2 = \frac{||r||_2}{||y||_2^2}|y^Tz| \leq \frac{||r||_2}{||y||_2}||z||_2$$

   because $|y^Tz| \leq ||y||_2||z||_2$ (Cauchy-Schwartz inequality). This shows that $||E||_2 \leq ||r||_2/||y||_2$. But

   $$||E||_2 = \max_{z\neq 0} \frac{||Ez||_2}{||z||_2} \geq \frac{||Ey||_2}{||y||_2} = \frac{||r||_2}{||y||_2},$$

   so we have that $||E||_2 = ||r||_2/||y||_2$.
   Need to show that any matrix $F$ satisfying $(A+F)y = b$ has at least as big 2-norm as $E$. This is indeed the case, since $(A+F)y = b$ implies $Fy = r$ and therefore

   $$||F||_2 = \max_{z\neq 0} \frac{||Fz||_2}{||z||_2} \geq \frac{||Fy||_2}{||y||_2} = \frac{||r||_2}{||y||_2} = ||E||_2.$$

The above shows that the 2-norm of $E$ is a good measure for the backward error and since $||E||_2 = ||r||_2/||y||_2$

$$\frac{||E||_2}{||A||_2} = \frac{||r||_2}{||A||_2||y||_2} = \frac{||b - Ay||_2}{||A||_2||y||_2},$$

i.e., the relative backward error equals the relative residual.

3. Pascal matrix:

```
function P=Pascal(n)
% The Pascal matrix of order n
P=ones(n,n);
for i=2:n
    for j=2:n
        P(i,j)=P(i,j-1)+P(i-1,j);
    end
end
```

```
% Solve the linear system A*x=b where A is the Pascal matrix
% of order n and b=A*ones(n,1)
format short e
for j=1:5
    n(j)=4*j;
    disp(['n=',num2str(n(j))])
    A=Pascal(n(j));
    e=ones(n(j),1);
    b=A*e;
    tic;
    [L,U,P]=lu(A);
    y=L\(P*b);
    x=U\y
    t(j)=toc;
    r=b-A*x;
    rel_res(j)=norm(r,1)/(norm(A,1)*norm(x,1));
    rel_error(j)=norm(x-e,1)/norm(e,1);
    cond_est(j)=1/rcond(A);
end
fprintf('  n    relative     relative   cond. number    time\n')
fprintf('       residual      error       estimate      (sec)\n\n')
fprintf('%3d %12.4e %12.4e %12.4e %12.4e\n',[n; rel_res; rel_error; cond_est; t])
semilogy(n,rel_error,'-',n,cond_est,'--',n,cond_est*eps,':')
legend('Relative error','Cond. number estimate','Estimate of error')
xlabel('n')
```

The computed solutions:

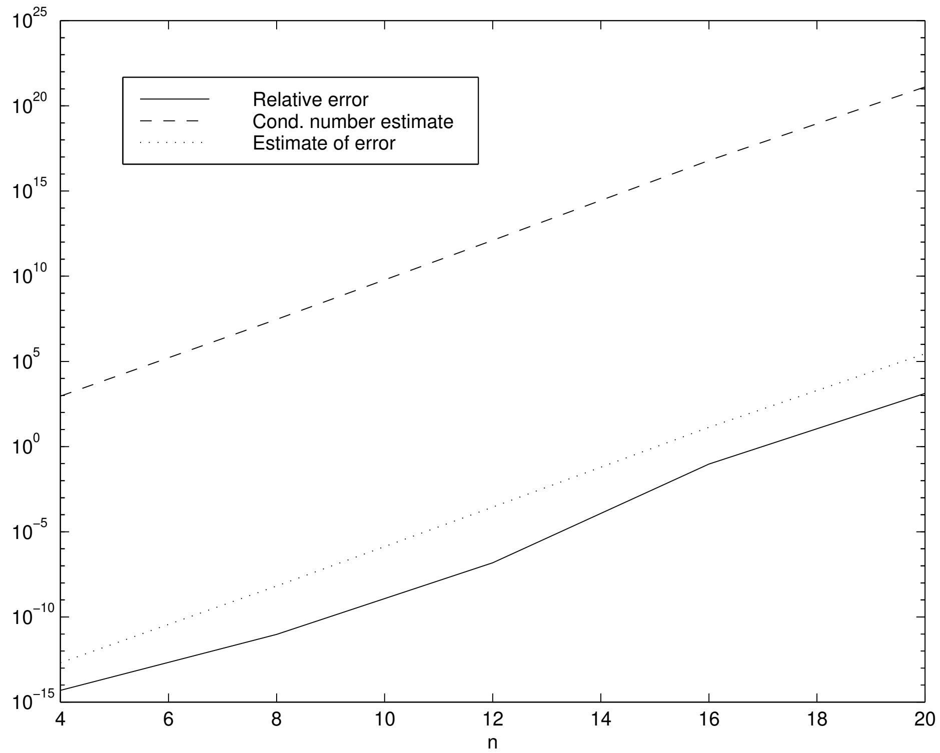
| n=4 | n=8 | n=12 | n=16 | n=20 |
|---|---|---|---|---|
| x = | x = | x = | x = | x = |
| 1.0000e+00 | 1.0000e+00 | 1.0000e+00 | 1.0001e+00 | 1.3467e+00 |
| 1.0000e+00 | 1.0000e+00 | 1.0000e+00 | 9.9899e-01 | -4.4874e+00 |
| 1.0000e+00 | 1.0000e+00 | 1.0000e+00 | 1.0065e+00 | 4.2134e+01 |
| 1.0000e+00 | 1.0000e+00 | 1.0000e+00 | 9.7382e-01 | -1.9272e+02 |
| | 1.0000e+00 | 1.0000e+00 | 1.0737e+00 | 6.4220e+02 |
| | 1.0000e+00 | 1.0000e+00 | 8.4695e-01 | -1.5799e+03 |
| | 1.0000e+00 | 1.0000e+00 | 1.2423e+00 | 3.0024e+03 |
| | 1.0000e+00 | 1.0000e+00 | 7.0289e-01 | -4.4698e+03 |
| | | 1.0000e+00 | 1.2843e+00 | 5.2705e+03 |
| | | 1.0000e+00 | 7.8777e-01 | -4.9089e+03 |
| | | 1.0000e+00 | 1.1225e+00 | 3.5716e+03 |
| | | 1.0000e+00 | 9.4637e-01 | -1.9586e+03 |
| | | | 1.0173e+00 | 7.4294e+02 |
| | | | 9.9615e-01 | -1.3023e+02 |
| | | | 1.0005e+00 | -4.4472e+01 |
| | | | 9.9997e-01 | 4.6214e+01 |
| | | | | -1.6977e+01 |
| | | | | 5.2198e+00 |
| | | | | 4.2746e-01 |
| | | | | 1.0350e+00 |

| n | relative residual | relative error | cond. number estimate | time (sec) |
|---|---|---|---|---|
| 4 | 0.0000e+00 | 4.8850e-15 | 9.1782e+02 | 6.1110e-03 |
| 8 | 1.9185e-17 | 9.4618e-12 | 2.9203e+07 | 8.5023e-02 |
| 12 | 1.9742e-17 | 1.5209e-07 | 1.2651e+12 | 4.0547e-02 |
| 16 | 2.1083e-17 | 9.3388e-02 | 6.1671e+16 | 1.4195e-02 |
| 20 | 1.0504e-19 | 1.3316e+03 | 1.2799e+21 | 1.1562e-01 |

In all cases the relative residual is less than machine epsilon (which means it is as accurate as one can hope for).

On the other hand, the relative error grows exponentially with $n$. This is caused by an exponential growth in the condition number. In fact, we should expect $\mathrm{cond}(A)\epsilon_{\mathrm{mach}}$ to be a reasonable estimate for the relative error, which it is, as can be seen from the plot. The big condition number comes from widely varying scales in the matrix; as you follow the diagonal the numbers grow exponentially.

The leading term in the flop count for LU-factorization is $2n^3/3$, so we should expect a cubic growth of time as a function of $n$. We don't see this here and I found my times to vary a lot between runs. This variation is because for such a small problem size, tasks other than the computation have relatively big effect on the running time. To get better time measurement we need much bigger $n$, so that the LU-factorization becomes the dominant work.




The estimate of error is the estimated condition number multiplied by machine epsilon.