Math221 Homework # 3 Solutions

Instructor: James W. Demmel

September 20, 2009

2.2.

It enough to give only the high order terms (n^3 and mn^2 with the right coefficients) in your answer, but for completeness we give the exact answer.

It costs $\frac{4n^3-3n^2-n}{6} \approx \frac{2n^3}{3}$ to factor $A = PLU$. Solving $Ux = b$ costs n^2. Solving $Ly = b_1$ costs $n^2 - n$ for a general b_1 (we have no divisions; L has ones on the diagonal) and only $i^2 - i$ for $b_1 = e_i$, where e_i is the i-th column of the identity matrix (since we only have to solve a smaller system for e_i ($x_j = 0$ for $j > i$). Total cost of solving $Ax = b$ having $A = PLU$ is $2n^2 - n$ for a general right hand side and $n^2 + i^2 - i$ for $b = e_i$.

Solving $AX = B$ after computing $A = PLU$ means solving m equations which costs $m(2n^2 - n)$. The total cost for this method is

$$C_1 = \frac{4n^3 - 3n^2 - n}{6} + m(2n^2 - n) \approx \frac{2n^3}{3} + 2mn^2.$$

To find A^{-1} after computing $A = PLU$ we means solve n linear equations with right hand sides e_i which costs $\sum_{i=1}^{n}(n^2 + i^2 - i) = \frac{4n^3-n}{3}$. Then to find $A^{-1}B$ we have another $mn(2n-1)$ flops.

The total cost of computing $A^{-1}B$ is

$$C_2 = \frac{4n^3 - 3n^2 - n}{6} + \frac{4n^3 - n}{3} + mn(2n-1) \approx 2n^3 + 2mn^2$$

$$C_2 - C_1 = \frac{4n^3 - n}{3} + 2mn^2 - mn - (2mn^2 - mn) = \frac{4n^3 - n}{3} \approx \frac{4n^3}{3}$$

2.7.

D is nonsingular, so $D_{ii} \neq 0$. $A = A^T$ implies $LDM^T = MDL^T$. Therefore $DM^TL^{-T} = L^{-1}MD$. $L^{-1}MD$ is lower triangular with diagonal D_{ii}, DM^TL^{-T} is upper triangular with diagonal D_{ii}, so $DM^TL^{-T} = L^{-1}MD = D$. Therefore $L^{-1}M = I$ and $M = L$.

2.8.

We generate random matrices A and vectors b as described in the problem, and solve the problem using GEPP to get solution x_1 and using Cramer's rule to get solution x_2. We then compute the backward error according to Theorem 2.2 in the book for both solutions: $r_i = Ax_i - b$ and then $\|r_i\|_1/(\|A\|_1\|x_i\|_1)$; if the method is backward stable then this last expression should be on the order of machine precision, about 10^{-16}. By running the following Matlab code, we get a table of 20 examples, for which the backward from GEPP is always small, but for Cramer's rule is it frequently as large as 10^{-3}.

```
% Answer for Question 2.8 in the text book
% For each example matrix:
%   cond = accurate condition number norm(A,2)*norm(inv(A),2)
%   berr(GEPP) = backward error for GEPP; should be O(1e-16)
%   berr(Cramer) = backward error for Cramer's rule; might be larger
clear res; pert = 1e-14;
for i=1:20,
    a=rand(2,2);a(:,2)=a(:,1)+pert*randn(2,1);
    b=a(:,2)+pert*randn(2,1);
    x1 = a\b;
    r1 = a*x1-b; nr1 = norm(r1,1)/(norm(a,1)*norm(x1,1));
    determ = a(1,1)*a(2,2)-a(1,2)*a(2,1);
    clear x2;
    x2(1) = (a(2,2)*b(1)-a(1,2)*b(2))/determ;
    x2(2) = (-a(2,1)*b(1)+a(1,1)*b(2))/determ;
    x2=x2';
    r2 = a*x2-b; nr2 = norm(r2,1)/(norm(a,1)*norm(x2,1));
    res(i,:)=[i,cond(a),nr1,nr2];
end
disp('    example       cond       berr(GEPP)  berr(Cramer)')
res
```

2.11.

For $x = te_i - e_j$ we have $0 < x^T Ax = t^2 a_{ii} - 2ta_{ij} + a_{jj}$ for all t. This is only possible when the discriminant is negative, i.e. $a_{ij}^2 < a_{ii}a_{jj}$.

2.13.

Part 1: We consider only the more general case, the Sherman-Morrison-Woodbury formula. Assume $T = I + V^T A^{-1} U$ is nonsingular. Then multiply out

$$
\begin{aligned}
&(A + UV^T)(A^{-1} - A^{-1}UT^{-1}V^T A^{-1}) \\
&= I + UV^T A^{-1} - UT^{-1}V^T A^{-1} - UV^T A^{-1}UT^{-1}V^T A^{-1} \\
&= I + UV^T A^{-1} - UT^{-1}V^T A^{-1} - U(T - I)T^{-1}V^T A^{-1} \\
&= I + UV^T A^{-1} - UT^{-1}V^T A^{-1} - UV^T A^{-1} + UT^{-1}V^T A^{-1} = I
\end{aligned}
$$

to see that $A + UV^T$ is nonsingular with the desired inverse.

Conversely, if T is singular, then $Tx = 0$ for some nonzero x, so $(I + V^T A^{-1}U)x = 0$. Note that this implies $Ux \neq 0$, for otherwise $Tx = x = 0$, a contradiction. Thus $y = A^{-1}Ux \neq 0$, and $(A + UV^T)y = UTx = 0$, so $A + UV^T$ has a nonzero null vector y and is singular.

Part 2: $B = A + uv^T$, $By = c$ implies

$$y = B^{-1}c = \left(A^{-1} - \frac{A^{-1}uv^TA^{-1}}{1 + v^TA^{-1}u} \right) c = A^{-1}c - \frac{A^{-1}uv^TA^{-1}c}{1 + v^TA^{-1}u}$$

So the algorithm is as follows:

1. Solve $Ax = c$ for x
2. Solve $Az = u$ for z
3. Compute $y = x - z\frac{v^Tx}{1+v^Tz}$.

2.17

`A\b` returns the solution x of Ax=b computed using Gaussian elimination with partial pivoting. b'/A returns the solution x^T of $b^T = x^TA$, where b^T and x^T are row vectors. A/b gives an error message, because the dimensions don't match correctly. `A\b` and inv(A)*b both solve Ax=b, but inv(A)*b computes the inverse explicitly and multiplies b by it.

2.18.1.

We use the fact that the LU decomposition is unique at every step (for unit lower triangular L). Let $A_{11} = L_{11}U_{11}$. We have

$$\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} = \begin{bmatrix} L_{11} & 0 \\ L_{21} & I \end{bmatrix} \begin{bmatrix} U_{11} & U_{12} \\ 0 & S \end{bmatrix}$$

where $A_{21} = L_{21}U_{11}$ (so $L_{21} = A_{21}U_{11}^{-1}$), $A_{12} = L_{11}U_{12}$ (so $U_{12} = L_{11}^{-1}A_{12}$), and

$$A_{22} = S + L_{21}U_{12} = S + A_{21}U_{11}^{-1}L_{11}^{-1}A_{12} = S + A_{21}A_{11}^{-1}A_{12}$$

Thus after k steps of Gaussian Elimination A_{22} is overwritten by $S = A_{22} - A_{21}A_{11}^{-1}A_{12}$.

Extra credit

See Algorithm 8 (RTRSM) on page 27 of
www.eecs.berkeley.edu/Pubs/TechRpts/2009/EECS-2009-29.pdf.