Numerical Analysis
Problem Sheet 3

Stuart Golodetz

February 6, 2006

1. By performing Gauss Elimination (without pivoting), solve

$$\begin{bmatrix} 2 & 1 & 1 & 0 \\ 4 & 3 & 3 & 1 \\ 8 & 7 & 9 & 5 \\ 6 & 7 & 9 & 8 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = \begin{bmatrix} 3 \\ 8 \\ 24 \\ 25 \end{bmatrix}$$

From your calculations, write down an LU factorisation of the matrix A above, and verify that $LU = A$. Then by successive back and forwards substitutions (and without further factorisation) solve $Ax = b_2$ where $b_2 = [4\ 7\ 9\ 2]^T$.

Answer

See separate sheets.

2. What is the determinant of the matrix A in question 1 above? (Note one of the few algebraic properties of the determinant is that $det(BC) = det(B)det(C)$ and you might also want to consider what is the determinant of a triangular matrix.)

Answer

Since $A = LU$, clearly $det(A) = det(L)det(U)$. The determinant of a triangular matrix is just the product of its diagonal elements, a result which is readily verified by induction. So here:

$$\begin{aligned} & det(A) \\ = \ & det(L)det(U) \\ = \ & 1 \times 8 \\ = \ & 8 \end{aligned}$$

3. Given an LU factorisation of a matrix A, how might one calculate a column of the inverse of A? Estimate the computational work in calculating A^{-1} and hence in solving $Ax = b$ via explicit computation of A^{-1} and multiplication by b.

   Are you now convinced that this is *not* the way to solve linear systems of equations in practice?!

   An even worse technique would be to apply GE separately for each column: what would the computational cost be then?

Answer

(a) To calculate column i of the inverse, just calculate (using forwards and back substitution as before) the vector x s.t.

$$Ax = [\underbrace{0 \cdots 0}_{i-1\ times}\ 1\ 0 \cdots 0]^T$$

For example, the 2^{nd} column of the inverse of the matrix A above could be calculated (as in question 1) via:

$y_0 = 0$
$2y_0 + y_1 = 1 \Rightarrow y_1 = 1$
$4y_0 + 3y_1 + y_2 = 0 \Rightarrow y_2 = -3$
$3y_0 + 4y_1 + y_2 + y_3 = 0 \Rightarrow y_3 = -1$

$2d = -1 \Rightarrow d = -\frac{1}{2}$
$2c + 2d = -3 \Rightarrow c = -1$
$b + c + d = 1 \Rightarrow b = \frac{5}{2}$
$2a + b + c = 0 \Rightarrow a = -\frac{3}{4}$

$$\begin{pmatrix} -0.75 \\ 2.5 \\ -1 \\ -0.5 \end{pmatrix}$$

(b) TODO

(c) TODO

4. [M] In MATLAB the 'backslash', \, solves linear systems of equations via Gauss Elimination with partial pivoting: thus $x = A\backslash b$ will perform a permuted LU factorisation (sometimes called a PLU factorisation: $PA = LU$) of A and solves $Ly = Pb$ via forward substitution and then $Ux = y$ via back substitution, giving back just the solution vector, x.

   Verify your solutions to the two linear systems in question 1 above. (Note to define a matrix in MATLAB - originally short for MATrix LABoratory! - you just need [to start,] to end, commas between entries in a row and semi-colons between different rows, thus

   $A = [2, 1, 1, 0; 4, 3, 3, 1; 8, 7, 9, 5; 6, 7, 9, 8]$

   defines the matrix in question 1 above and

   $b = [3; 8; 24; 25]$

   defines the first right-hand-side column vector.)

   Answer

   The solutions produced by MATLAB match those I calculated. (Since I checked them on my graphics calculator at the time, this is not altogether unexpected!)

5. Prove that the product of two lower-triangular matrices is lower-triangular and that the inverse of a non-singular lower-triangular matrix is lower-triangular. Deduce similar results for upper-triangular matrices.

   Answer

   Given an $m \times n$ matrix $A = (a_{i,j})$ and an $n \times p$ matrix $B = (b_{i,j})$, recall that if $AB = C = (c_{i,j})$ then $c_{i,j} = \sum_{k=1}^{n} a_{i,k} b_{k,j}$.

   (a) Suppose A and B are both lower-triangular. In other words, suppose that if $i < j$ then $a_{i,j} = b_{i,j} = 0$. We want to show that if $i < j$ then $c_{i,j} = 0$ as well. Well, as noted before:

$$c_{i,j} = \sum_{k=1}^{n} a_{i,k} b_{k,j}$$

Our strategy is to show that if $i < j$ then every term of this sum is 0, i.e. either $a_{i,k}$ or $b_{k,j}$ is 0. This is fairly easy: if $i < k$ or $k < j$ then obviously the relevant factor is going to be 0, so suppose for a contradiction that this isn't the case. Then $k \leq i$ and $k \geq j$. But this clearly can't be the case, for then $k \leq i < j \leq k$! So our assumption was wrong and one of the factors must be 0. Hence the sum evaluates to 0 and all the elements of C above the diagonal are 0, whence C is lower-triangular.

(b) *The textbook approach*

As per p.47 of Süli and Mayers, we prove that the inverse of a non-singular $n \times n$ lower-triangular matrix L is lower-triangular by induction.

Base Case (n = 2)

Well if

$$L = \begin{pmatrix} a & 0 \\ c & d \end{pmatrix}$$

then by the usual method

$$L^{-1} = \frac{1}{ad} \begin{pmatrix} d & 0 \\ -c & a \end{pmatrix} = \begin{pmatrix} \frac{1}{a} & 0 \\ -\frac{c}{ad} & \frac{1}{d} \end{pmatrix}$$

This is clearly lower-triangular.

Inductive Hypothesis

The inverse of a non-singular lower-triangular matrix of size[1] $\in [2, k]$ is lower-triangular.

Inductive Step

An arbitrary non-singular lower-triangular matrix L of size $k+1$ can be written as follows:

$$L = \begin{pmatrix} L_k & \mathbf{0} \\ \mathbf{a}^T & \alpha \end{pmatrix}$$

where L_k is a non-singular lower-triangular matrix of size $k \times k$, $\mathbf{a}$ and $\mathbf{0}$ are $k \times 1$ column vectors and α is a real number. Its inverse, L^{-1}, can be written as:

$$L^{-1} = \begin{pmatrix} X & \mathbf{b} \\ \mathbf{c}^T & \beta \end{pmatrix}$$

where X is a matrix of size $k \times k$, $\mathbf{b}$ and $\mathbf{c}$ are $k \times 1$ column vectors and β is a real number. Now, obviously we have that $LL^{-1} = I_{k+1}$, whence:

$$\begin{pmatrix} L_k & \mathbf{0} \\ \mathbf{a}^T & \alpha \end{pmatrix} \begin{pmatrix} X & \mathbf{b} \\ \mathbf{c}^T & \beta \end{pmatrix} = \begin{pmatrix} I_k & \mathbf{0} \\ \mathbf{0}^T & 1 \end{pmatrix}$$

This straightforwardly gives us that: $L_k X = I_k$, $L_k \mathbf{b} = \mathbf{0}$, $\mathbf{a}^T X + \alpha \mathbf{c}^T = \mathbf{0}^T$ and $\mathbf{a}^T \mathbf{b} + \alpha\beta = 1$. This means that $X = L_k^{-1}$, which is lower-triangular by hypothesis and that $\mathbf{b} = \mathbf{0}$. Furthermore, since $\alpha \neq 0$ we can deduce that $\mathbf{c}^T = -\frac{\mathbf{a}^T X}{\alpha}$ and $\beta = \frac{1 - \mathbf{a}^T \mathbf{b}}{\alpha}$. So L^{-1} is lower-triangular and we're done. □

[1] Note that we're talking about square matrices here.

A slightly more esoteric method

We observe (by looking at the Gaussian Elimination method) that an arbitrary unit $n \times n$ lower-triangular matrix L is a product of $M(i, j, \lambda)$ matrices, that the inverses of these are all lower-triangular and that as we just saw multiplying lots of lower-triangular matrices together (as we'll do to get the inverse of L) gives us a lower-triangular matrix. So the inverse of a unit lower-triangular matrix is lower-triangular.

What about non-unit lower-triangular matrices? Well, they're equal to a scaling matrix times a unit lower-triangular matrix. For instance:

$$\begin{pmatrix} 2 & 0 \\ 1 & 4 \end{pmatrix} = \begin{pmatrix} 2 & 0 \\ 0 & 4 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ \frac{1}{4} & 1 \end{pmatrix}$$

Both scaling matrices and, more importantly, their inverses, are lower-triangular. So when we calculate the inverse of L, we're just multiplying in yet another lower-triangular matrix and the resulting inverse is still lower-triangular. More explicitly, we have, for some M matrices and with S being the scaling matrix just mentioned, that:

$$L = SM_1 \cdots M_k \text{ for some integer } k \geq 1$$

So:

$$L^{-1} = M_k^{-1} \cdots M_1^{-1} S^{-1}$$

This is a product of lower-triangular matrices and hence lower-triangular.

(c) The product of two upper-triangular matrices is upper-triangular. This can be deduced as follows: given two upper-triangular matrices A and B, observe that their transposes are lower-triangular. We want to show that their product AB is upper-triangular, for which it suffices to show that $(AB)^T$ is lower-triangular. Well, $(AB)^T = B^T A^T$ and as just noted B^T and A^T are lower-triangular, so, by the proof above, their product, $(AB)^T$, is also.

(d) Given an upper-triangular matrix A, observe that its transpose is lower-triangular. We want to show that if $AB = I$ then B is upper-triangular. Well, $(AB)^T = B^T A^T = I^T = I$ and we know from before that since A^T is lower-triangular, its inverse, B^T, is as well, so B itself must be upper-triangular.

6. Suppose A is a real $n \times n$ matrix with $n \geq 2$ and that the permutation matrix

$$P = \begin{bmatrix} 0 & \cdots & 0 & 1 \\ 0 & \cdots & 1 & 0 \\ \vdots & . & \vdots & \vdots \\ 1 & \cdots & 0 & 0 \end{bmatrix}$$

Show that premultiplication of A by P reverses the order of the rows of A.

If $A = LU$ is an LU factorisation of A (without pivoting), what is the structure of PLP? Hence describe how to calculate a factorisation $A = \hat{U}\hat{L}$ where $\hat{U}$ is unit upper-triangular and $\hat{L}$ is lower-triangular.

Answer

(a) Let $B = (b_{i,j}) = PA$. Then:

$$b_{i,j} = \sum_{k=1}^{n} p_{i,k} a_{k,j}$$

Now $p_{i,j} = \delta_{i,(n+1-j)}$, i.e. $p_{i,j} = 1$ if $i + j = n + 1$ and $p_{i,j} = 0$ otherwise. So:

$$b_{i,j} = p_{i,(n+1-i)}\, a_{(n+1-i),j} = a_{(n+1-i),j}$$

since all the other terms in the summation above evaluate to 0. But this new formula says exactly what we wanted to derive, namely that the order of the rows has been reversed, since $n + 1 - i$ is just the i^{th} row in from the bottom of the matrix.

(b) The structure of PLP is unit upper-triangular. This is intuitively obvious, since we're essentially just reversing the order of all the rows and the columns, but if we wanted to be more precise about it, we could note that if $A' = (a'_{i,j}) = PLP$ then we have that $a'_{i,j} = a_{(n+1-i),(n+1-j)}$. In particular, this implies the following things:

- The diagonal elements of A' are all 1, as for any i we have that $a'_{i,i} = a_{(n+1-i),(n+1-i)}$ and all the diagonal elements of A were 1.
- Any element of A' below the diagonal is 0, since if $i > j$ then $n + 1 - i < n + 1 - j$, i.e. $a'_{i,j}$ equals an element of A which was above the diagonal and hence 0.

Given these two facts, A' is obviously unit upper-triangular.

(c) We've observed that if L is unit lower-triangular then PLP is unit upper-triangular. For similar reasons, if U is upper-triangular then PUP is lower-triangular.

Now, we observe that $PP = I$ and consider $A' = PAP$. If we can perform an LU factorisation of A (i.e. $\forall i \cdot a_{i,i} \neq 0$) then we can perform one of A', since the set of diagonal elements remains the same (note that $a'_{i,i} = a_{(n+1-i),(n+1-i)}$).

Suppose then that we have derived L and U such that $A' = LU$. Then:

$$A = (PP)A(PP) = P(PAP)P = P(A')P = P(LU)P = P(LPPU)P = (\underbrace{PLP}_{\hat{U}})(\underbrace{PUP}_{\hat{L}})$$

where $\hat{U}$ is unit upper-triangular and $\hat{L}$ is lower-triangular. So to calculate a factorisation as described, all we do is LU factorise PAP and calculate $\hat{U} = PLP$ and $\hat{L} = PUP$.

7. Perform an LU factorisation with partial pivoting on the matrix

$$\begin{bmatrix} \frac{2}{3} & 1 & 3 \\ 2 & 1 & 4 \\ 1 & \frac{3}{2} & 4 \end{bmatrix}$$

and write down the permutation matrix P, the unit lower-triangular matrix of multipliers and the upper-triangular matrix U such that $PA = LU$ which you should verify.

[M] Check your answer with MATLAB:

$[L, U, P] = lu(A)$

will compute P, L and U for you - see how much easier! (but you must do this by hand once in your life and if you've correctly done the above then join the club! else look and see why you've gone wrong).

Answer

See separate sheet. The way I've found L is rather convoluted; is there a better way?

8. For the problem description see the sheet (it's too long to type out).

Answer

The results I got were as follows:

```
for k=4:10, A = randn(2^k); tic, [L,U,P] = lu(A); toc, end

(Elapsed time is 0.002361 seconds.)
Elapsed time is 0.000140 seconds.
Elapsed time is 0.000530 seconds.
Elapsed time is 0.002364 seconds.
Elapsed time is 0.014012 seconds.
Elapsed time is 0.102702 seconds.
Elapsed time is 0.499340 seconds.

0.499340/0.102702 = 4.8620
0.102702/0.014012 = 7.3296
0.014012/0.002364 = 5.9272
0.002364/0.000530 = 4.4604
0.000530/0.000140 = 3.7857

(4.8620+7.3296+5.9272+4.4604+3.7857)/5 = 5.2730
```

(The reason I started from 4 rather than 5 was that I noticed the first result was always larger than expected: perhaps setup time or something?)

The time to compute an LU factorisation doesn't grow by a consistent amount as you double the dimension (it would be rather strange if it did, in fact), but the factors are all roughly similar: their average is about 5, as the above calculation shows. This isn't a particularly meaningful number in itself, it's more of a ballpark figure.