Homework 3 solutions

1. Exercise 2.13. We first note that $A = PD$ where $P$ is a permutation matrix and $D$ is diagonal:

$$A = \begin{bmatrix}
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 \\
0 & 1 & 0 & 0 \\
1 & 0 & 0 & 0
\end{bmatrix} \begin{bmatrix}
10^{-2} & 0 & 0 & 0 \\
0 & 10^{-3} & 0 & 0 \\
0 & 0 & -10^4 & 0 \\
0 & 0 & 0 & -10
\end{bmatrix}.$$

(a) We have

$$\|A\| = \max_{x \neq 0} \frac{\sqrt{(10^{-2}x_1)^2 + (10^{-3}x_2)^2 + (-10^4x_3)^2 + (-10x_4)^2}}{\sqrt{x_1^2 + x_2^2 + x_3^2 + x_4^2}}$$

$$= 10^4.$$

We achieve the maximum in the definition of $A$ by choosing

$$x_1 = 0, \quad x_2 = 0, \quad x_3 = 1, \quad x_4 = 0.$$

The argument is the same as when we derived the norm of a diagonal matrix. An alternative proof is to use the expression $A = PD$:

$$\|Ax\| = \|PDx\| = (x^TDP^TPDx)^{1/2} = (x^TDDx)^{1/2} = \|Dx\|$$

because $P^TP = I$ for a permutation matrix. Therefore

$$\frac{\|Ax\|}{\|x\|} = \frac{\|Dx\|}{\|x\|}$$

for all $x$, and hence the norm of $A$ will be the same as the norm of $D$.

(b) The inverse is

$$A^{-1} = \begin{bmatrix}
0 & 0 & 0 & 10^2 \\
0 & 0 & 10^3 & 0 \\
-10^{-4} & 0 & 0 & 0 \\
0 & -10^{-1} & 0 & 0
\end{bmatrix}.$$

We can find this answer by solving $AX = I$ column by column.
Alternatively, we can write $A^{-1} = (PD)^{-1} = D^{-1}P^{-1} = D^{-1}P^T$. Therefore

$$A^{-1} = \begin{bmatrix} 10^2 & 0 & 0 & 0 \\ 0 & 10^3 & 0 & 0 \\ 0 & 0 & -10^{-4} & 0 \\ 0 & 0 & 0 & -10^{-1} \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

$$= \begin{bmatrix} 0 & 0 & 0 & 10^2 \\ 0 & 0 & 10^3 & 0 \\ -10^{-4} & 0 & 0 & 0 \\ 0 & -10^{-1} & 0 & 0 \end{bmatrix}.$$  

(c) As in part 1, we have

$$\|A\| = \max_{x \neq 0} \frac{\sqrt{(-10^{-4}x_1)^2 + (-10^{-1}x_2)^2 + (10^3 x_3)^2 + (10^2 x_4)^2}}{\sqrt{x_1^2 + x_2^2 + x_3^2 + x_4^2}}$$

$$= 10^3.$$  

The counterpart of the second method in part 1 is to note that $\|x\| = \|P^T x\|$ if $P$ is a permutation matrix. Therefore

$$\frac{\|Ax\|}{\|x\|} = \frac{\|D^{-1}P^T x\|}{\|x\|} = \frac{\|D^{-1}P^T x\|}{\|P^T x\|}$$

and maximizing this over $x \neq 0$ is the same as maximizing $\|D^{-1}y\|/\|y\|$ over $y \neq 0$. Therefore $\|D^{-1}P^T\| = \|D^{-1}\|$.  

(d) $\kappa(A) = \|A\|\|A^{-1}\| = 10^7$.

2. Exercise 2.15.

- $\|A\| \geq \|Ax\|/\|x\|$ for all $x \neq 0$, so

$$\|A\| \geq \max\left\{ \frac{\|Ax^{(1)}\|}{\|x^{(1)}\|}, \frac{\|Ax^{(2)}\|}{\|x^{(2)}\|}, \frac{\|Ax^{(3)}\|}{\|x^{(3)}\|}, \frac{\|Ax^{(4)}\|}{\|x^{(4)}\|} \right\}$$

$$= \max\{100, 10^{-2}, 10, 10^5\}$$

$$= 10^5.$$  

- $\|A^{-1}\| \geq \|A^{-1}y\|/\|y\|$ for all $y \neq 0$. Therefore,

$$\|A^{-1}\| \geq \max\left\{ \frac{\|x^{(1)}\|}{\|Ax^{(1)}\|}, \frac{\|x^{(2)}\|}{\|Ax^{(2)}\|}, \frac{\|x^{(3)}\|}{\|Ax^{(3)}\|}, \frac{\|x^{(4)}\|}{\|Ax^{(4)}\|} \right\}$$

$$= \max\{10^{-2}, 100, 10^{-1}, 10^{-5}\}$$

$$= 100.$$  

2
\[ \kappa(A) = \|A\|\|A^{-1}\| \geq 10^7 \] (by combining the bounds for \(\|A\|\) and \(\|A^{-1}\|\)).


(a) \(\kappa(A_1) = 1181, \kappa(A_2) = 3 \cdot 10^{17}, \kappa(A_3) = 95.\)

A very high condition number indicates that the matrix is very close to a singular matrix. In fact, the exact \(A_2\) is singular (see part c), so \(\kappa(A_2)\) is not defined (or might be defined as \(+\infty\)). The fact that Matlab returns a very high finite number for \(A_2\) is due to the finite precision.

(b) We prefer experiment 3.

(c) Experiment 2 is certainly a bad choice, because sensor 3 will measure the same as sensor 1, and sensor 2 gives the same measurement as sensor 4. So we really only have two independent measurements. Therefore \(A_2\) is singular.

The comparison between experiments 1 and 3 is less clear. If the four sensors are at the same position, the matrix is be singular (condition number \(+\infty\)). So we would expect that as we move the sensors closer together, the condition number gets worse.


(a) The columns of \(A\) are linearly dependent when \(\epsilon = 0\): the sum of the first two columns is equal to the third column.

To verify the expression for the inverse, we multiply the two matrices to get

\[
\frac{1}{\epsilon} \begin{bmatrix} 1 + \epsilon & 1 & 2 \\ 1 & -1 & 0 \\ 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & -2 \\ 1 & 1 - \epsilon & -2 \\ -1 & -1 & 2 + \epsilon \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.
\]

(b) Use the inequality \(\|A\| \geq \|Ax\|/\|x\|\) with \(x = (0, 0, 1)\) to get \(\|A\| \geq \sqrt{3}\).

Use the inequality \(\|A^{-1}\| \geq \|A^{-1}x\|/\|x\|\) with \(x = (1, 0, 0)\) to get \(\|A\| \geq \sqrt{3/|\epsilon|}\).

Multiplying the two bounds gives

\[ \kappa(A) = \|A\|\|A^{-1}\| \geq \frac{\sqrt{15}}{|\epsilon|}. \]

(c) The solution of \(Ax = b\) is

\[ x = A^{-1}b = \frac{1}{\epsilon} \begin{bmatrix} 1 & 1 & -2 \\ 1 & 1 - \epsilon & -2 \\ -1 & -1 & 2 + \epsilon \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ -1 \\ 1 \end{bmatrix}. \]

Choose \(\Delta b = (1, 0, 0)\). Then

\[ \Delta x = A^{-1}\Delta b = \frac{1}{\epsilon} \begin{bmatrix} 1 & 1 & -2 \\ 1 & 1 - \epsilon & -2 \\ -1 & -1 & 2 + \epsilon \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \frac{1}{\epsilon} \begin{bmatrix} 1 \\ 1 \\ -1 \end{bmatrix}. \]
which gives \[
\frac{\|\Delta x\|}{\|x\|} = \frac{\sqrt{3}}{|c|\sqrt{2}}, \quad \frac{\|\Delta b\|}{\|b\|} = \frac{1}{\sqrt{3}}.
\]


(a) \(n^2 + n\) for forming \(A = D + uv^T\), followed by \((2/3)n^3\) for solving \(Ax = b\).

(b) Using the expression for the inverse, we can write \(x\) as
\[
x = (D^{-1} + \frac{1}{1 + v^T D^{-1} u} D^{-1} uv^T D^{-1}) b
\]
\[
= D^{-1} b + \frac{1}{1 + v^T D^{-1} u} D^{-1} uv^T D^{-1} b
\]
\[
= D^{-1} b + \frac{v^T D^{-1} b}{1 + v^T D^{-1} u} D^{-1} u
\]

We can evaluate this as follows:
- calculate \(y = D^{-1} b\) \((n\) flops\)
- calculate \(z = D^{-1} u\) \((n\) flops\)
- calculate \(\alpha = v^T y/(1 + v^T z)\) \((4n + 2)\)
- calculate \(x = y + \alpha z\) \((2n)\)

The total is \(8n\).


(a) Costs \((8/3)n^3\).

Instead of calculating the inverse it is more efficient to solve \(Ax = b\), and then make the inner product with \(c\) as in \texttt{val = c'*(A\textbackslash b)}; The cost is \((2/3)n^3\) \(\text{(for solving } Ax = b\}.

(b) The cost is \((8/3)n^3\) for calculating the inverse, plus \(2n^2m\) for the matrix-matrix product \(A^{-1} B\), plus \(2nm\) for the product with \(c^T\). Ignoring lower-order terms we get \((8/3)n^3 + 2n^2m\).

There are two problems here. First we should use the backslash operator instead of computing the inverse. Second we can improve the code by observing that
\[
c^T A^{-1} B = (A^{-T} c)^T B.
\]

It is therefore more efficient to first compute \(A^{-T} c\) (\(i.e.,\) solve \(A^T x = c\)), and then calculate \(c^T A^{-1} B\) as \(x^T B\). In Matlab: \texttt{val = (A'\textbackslash c)'*B}. The cost of this method is roughly \(2n^3/3\) flops (for solving \(A^T x = c\). The product \(x^T B\) cost \(2nm\) flops, which is negligible compared with \(2n^3/3\).
(c) The cost is \((2/3)(2n)^3 = (16/3)n^3\).

We can improve this by noting that the equations

\[
\begin{bmatrix}
A & 0 \\
0 & B
\end{bmatrix}
\begin{bmatrix}
x_1 \\
x_2
\end{bmatrix} =
\begin{bmatrix}
b \\
c
\end{bmatrix}
\]

(where \(x_1 \in \mathbb{R}^n\) and \(x_2 \in \mathbb{R}^n\)) are really two separate sets of linear equations

\[Ax_1 = b, \quad Bx_2 = c.\]

Therefore we can also find the solution as follows:

\[
\begin{align*}
&>> x1 = A\backslash b; \\
&>> x2 = B\backslash c; \\
&>> x = [x1; x2];
\end{align*}
\]

This costs \((4/3)n^3\) flops.

(d) We can take advantage of the structure of the set of equations

\[
\begin{bmatrix}
A & B \\
C & I
\end{bmatrix}
\begin{bmatrix}
x_1 \\
x_2
\end{bmatrix} =
\begin{bmatrix}
b \\
c
\end{bmatrix}.
\]

Using the second equation we can express \(x_2\) in terms of \(x_1\) as

\[x_2 = c - Cx_1.\]

Substituting this in the second equation, we can eliminate \(x_2\):

\[Ax_1 + B(c - Cx_1) = b,\]

\[
i.e.,
(A - BC)x_1 = b - Bc.
\] (1)

We can first solve this set of equation for \(x_1\), and then obtain \(x_2\) using \(x_2 = c - Cx_1\). The Matlab code is

\[
\begin{align*}
&>> x1 = (A-B*C)\backslash (b-B*c); \\
&>> x2 = c-C*x1; \\
&>> x = [x1; x2];
\end{align*}
\]

What is the cost of this implementation? First we have to form the matrix \(A - BC\), which requires \(20n^3\) flops. We have to construct the right hand side \(b - Bc\), which costs \(20n^2\) flops. Then we have to solve the set of equations (1), which takes \((2/3)n^3\) flops. Finally, we have to calculate \(x_2 = c - Cx_1\), which takes \(20n^2\) flops. If we collect the \(n^3\) terms, we end up with a total of \((20 + 2/3)n^3\), which is roughly 100 times less than \(2(11n)^3/3\).
(e) We can partition the variable \( x \) and the righthand side \( b \) in two blocks and write the equations \( Ax = b \) as

\[
\begin{bmatrix}
I & B \\
C & I
\end{bmatrix}
\begin{bmatrix}
x_1 \\
x_2
\end{bmatrix}
= 
\begin{bmatrix}
b_1 \\
b_2
\end{bmatrix}.
\]

In other words,

\[ x_1 + Bx_2 = b_1, \quad Cx_1 + x_2 = b_2. \]

We can either eliminate \( x_1 \) from the first equation, or eliminate \( x_2 \) from the second. If we eliminate \( x_1 \) by substituting \( x_1 = b_1 - Bx_2 \) in the second equation, we obtain

\[(I - CB)x_2 = b_2 - Cb_1.\]

We can solve these equations for \( x_2 \) and then compute \( x_1 = b_1 - Bx_2 \). The steps in the algorithm are

i. Form \( D = I - CB \) (\( 2n^2m + n \approx 2n^2m \) flops) and \( d = b_2 - Cb_1 \) (\( 2nm + n \approx 2nm \)).

ii. Solve \( Dx_2 = d \) using the standard method for dense matrices (\( (2/3)n^3 \)).

iii. Compute \( x_1 = b_1 - Bx_2 \) (\( 2mn + m \approx 2mn \)).

Total cost: \( 2n^2m + (2/3)n^3 \).

If we eliminate \( x_2 \) by substituting \( x_2 = b_2 - Cx_1 \) in the first equation, we have to solve

\[(I - BC)x_1 = b_1 - Bb_2\]

for \( x_1 \), and then compute \( x_2 = b_2 - Cx_1 \).

i. Form \( D = I - BC \) (\( 2nm^2 + m \approx 2nm^2 \) flops) and \( d = b_1 - Bb_2 \) (\( 2nm + m \approx 2nm \)).

ii. Solve \( Dx_2 = d \) using the standard method (\( (2/3)m^3 \) flops).

iii. Compute \( x_2 = b_2 - Cx_1 \) (\( 2mn + n \approx 2mn \)).

Total: \( 2nm^2 + (2/3)m^3 \).

We conclude that method 2 is faster if \( n > m \).


(a) The inverse \( A^{-1} \) is the unique matrix \( X \) that satisfies \( AX = I \). To determine \( A^{-1} \) we solve the equation \( AX = I \) column by column.

The elements of the \( i \)th column of \( A^{-1} \) are the solutions of the equations

\[
\begin{align*}
a_{11}x_1 & = 0 \\
a_{21}x_1 + a_{22}x_2 & = 0 \\
a_{31}x_1 + a_{32}x_2 + a_{33}x_3 & = 0
\end{align*}
\]
\[ a_{i-1,1}x_1 + a_{i-1,2}x_2 + \cdots + a_{i-1,i-1}x_{i-1} = 0 \]
\[ a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{ii}x_i = 1 \]
\[ \vdots \]
\[ a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n = 0, \]

which can be solved by forward substitution. From the first \( i - 1 \) equations, we see that
\[ x_1 = x_2 = \cdots = x_{i-1} = 0. \]

This shows that \( A^{-1} \) is lower triangular.

(b) A short (and correct) answer is that we need to solve \( AX = I \), column by column, by solving the \( n \) triangular sets of equations given in part 1. The cost is equal to \( n \) forward substitutions, i.e., \( n^3 \).

A closer look shows that we can save some work because the righthand sides are unit vectors. As noted in part 1, the first \( i - 1 \) elements of column \( i \) of \( A^{-1} \) are zero. The remaining coefficients are the solution of

\[
\begin{bmatrix}
  a_{ii} & 0 & \cdots & 0 \\
  a_{i+1,i} & a_{i+1,i+1} & \cdots & 0 \\
  \vdots & \vdots & \ddots & \vdots \\
  a_{n,i} & a_{n,i+1} & \cdots & a_{nn}
\end{bmatrix}
\begin{bmatrix}
  x_i \\
  x_{i+1} \\
  \vdots \\
  x_n
\end{bmatrix} = 
\begin{bmatrix}
  1 \\
  0 \\
  \vdots \\
  0
\end{bmatrix}. \tag{2}
\]

This is a lower triangular set of equations of dimension \( n - i + 1 \), and can be solved via forward substitution in \( (n - i + 1)^2 \) operations. The total is
\[ n^2 + (n - 1)^2 + (n - 2)^2 + \cdots + 2^2 + 1^2 = (1/3)n^3 \]
operations, plus lower order terms.

8. Exercise 2.27.

\[
L = \begin{bmatrix}
  1 & 0 & 0 & 0 \\
  -2 & 1 & 0 & 0 \\
  1 & -2 & 1 & 0 \\
  -4 & 3 & 2 & 1
\end{bmatrix}, \quad U = \begin{bmatrix}
  -3 & 2 & 0 & 3 \\
  0 & -2 & 0 & -6 \\
  0 & 0 & -1 & 1 \\
  0 & 0 & 0 & 3
\end{bmatrix}
\]

(a) We first determine the first row of \( U \) and the first column of \( L \):
\[
\begin{bmatrix}
  -3 & 2 & 0 & 3 \\
  6 & -6 & 0 & -12 \\
  -3 & 6 & -1 & 16 \\
  12 & -14 & -2 & -25
\end{bmatrix} = \begin{bmatrix}
  1 & 0 & 0 & 0 \\
  -2 & 1 & 0 & 0 \\
  1 & l_{32} & 1 & 0 \\
  -4 & l_{42} & l_{43} & 1
\end{bmatrix} \begin{bmatrix}
  -3 & 2 & 0 & 3 \\
  0 & u_{22} & u_{23} & u_{24} \\
  0 & u_{32} & u_{33} & u_{34} \\
  0 & 0 & 0 & u_{44}
\end{bmatrix}.
\]
(b) Next we examine the submatrix

\[
\begin{bmatrix}
-6 & 0 & -12 \\
6 & -1 & 16 \\
-14 & -2 & -25
\end{bmatrix}
= \begin{bmatrix}
-2 \\
1 \\
-4
\end{bmatrix}
\begin{bmatrix}
2 & 0 & 3 \\
l_{32} & 1 & 0 \\
l_{42} & l_{43} & 1
\end{bmatrix}
\begin{bmatrix}
u_{22} & u_{23} & u_{24} \\
u_{33} & u_{34} \\
u_{43} & u_{44}
\end{bmatrix}.
\]

If we combine the constants, this reduces to

\[
\begin{bmatrix}
-2 & 0 & -6 \\
4 & -1 & 13 \\
-6 & -2 & -13
\end{bmatrix}
= \begin{bmatrix}
1 & 0 & 0 \\
l_{32} & 1 & 0 \\
l_{42} & l_{43} & 1
\end{bmatrix}
\begin{bmatrix}
u_{22} & u_{23} & u_{24} \\
u_{33} & u_{34} \\
u_{43} & u_{44}
\end{bmatrix}.
\]

The second column of \( L \) and the second row \( U \) follow immediately:

\[
\begin{bmatrix}
-2 & 0 & -6 \\
4 & -1 & 13 \\
-6 & -2 & -13
\end{bmatrix}
= \begin{bmatrix}
1 & 0 & 0 \\
-2 & 1 & 0 \\
3 & l_{43} & 1
\end{bmatrix}
\begin{bmatrix}
-2 & 0 & -6 \\
0 & u_{33} & u_{34} \\
0 & 0 & u_{44}
\end{bmatrix}.
\]

(c) Next we examine the submatrix

\[
\begin{bmatrix}
-1 & 13 \\
-2 & -13
\end{bmatrix}
= \begin{bmatrix}
-2 \\
3
\end{bmatrix}
\begin{bmatrix}
0 & -6 \\
l_{43} & 1
\end{bmatrix}
\begin{bmatrix}
u_{33} & u_{34} \\
0 & u_{44}
\end{bmatrix},
\]

\[\text{i.e.,}\]

\[
\begin{bmatrix}
-1 & 1 \\
-2 & 5
\end{bmatrix}
= \begin{bmatrix}
1 & 0 \\
l_{43} & 1
\end{bmatrix}
\begin{bmatrix}
u_{33} & u_{34} \\
0 & u_{44}
\end{bmatrix}.
\]

This gives us the third row of \( U \) and the third column of \( L \):

\[
\begin{bmatrix}
-1 & 1 \\
-2 & 5
\end{bmatrix}
= \begin{bmatrix}
1 & 0 \\
-1 & 1
\end{bmatrix}
\begin{bmatrix}
u_{33} & u_{34} \\
0 & u_{44}
\end{bmatrix}.
\]

(d) Finally, \( u_{44} \) follows from

\[5 = 2 + u_{44},\]

hence \( u_{44} = 3 \).

9. **Exercise 2.28.**

(a) LU-factorization \( A = PLU \) ((2/3)\( n^3 \) flops).

(b) Calculate \( y = A^{-1}b \) by solving \( PLUy = b \):

i. Calculate \( v = P^Tb \) (0 flops, because \( P^Tb \) is a permutation of \( b \))

ii. Solve \( Lw = v \) by forward substitution (\( n^2 \) flops)

iii. Solve \( Uy = w \) by backward substitution (\( n^2 \) flops)

(c) Calculate \( v = A^{-2}b = A^{-1}y \) by solving \( PLUv = y \) (2\( n^2 \))

(d) Calculate \( w = A^{-3}b = A^{-1}v \) by solving \( PLUw = v \) (2\( n^2 \))

(e) \( x = b + y + v + w \) (3\( n \)).

Total: (2/3)\( n^3 \) + 6\( n^2 \) + 3\( n \).