Computer exercise: vectors and matrices

The concepts vector and matrix are very important in many areas, and it is used when you solve linear systems of equations. The Matlab stands for MATrix LABoratory, suggesting that Matlab is designed especially for matrices. Here you will get a little introduction to matrices and vectors in Matlab.

Introduction

A vector is usually seen as a quantity having direction as well as magnitude. In physics a vector is denoted with an arrow, like \( \vec{v} \). In computer science you often consider a vector as a way of storing data (like in a table), and it is often called array. In Matlab we do not really distinguish between a vector and a scalar (a single number). You can consider a scalar to be a vector of length one.

Vectors can be defined as row vectors or column vectors: \( \mathbf{v} = \left( \begin{array}{c} 1 \\ 3 \end{array} \right) \) or \( \mathbf{u} = ( 1 \ 3 ) \). In Matlab you define these vectors with the commands: \( \mathbf{v} = [1; 3] \) and \( \mathbf{u} = [1 \ 3] \). Matrices and vectors are surrounded by brackets, and semicolon inside the brackets separate the rows.

To do

Define the following vectors in Matlab:

\[
\mathbf{a} = \left( \begin{array}{c} 2 \\ 0 \end{array} \right), \quad \mathbf{b} = \left( \begin{array}{c} 3 \\ 3 \end{array} \right), \quad \mathbf{c} = \left( \begin{array}{c} 0 \\ 1 \end{array} \right), \quad \mathbf{d} = \left( \begin{array}{c} -2 \\ 2 \end{array} \right), \quad \mathbf{e} = \left( \begin{array}{c} 1 \\ 3 \\ 2 \end{array} \right)
\]

and

Store the vectors in variables \( \mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d} \) and \( \mathbf{e} \).

1. In Matlab’s Command window, type the following (and try to understand the results):
   a) \( \mathbf{e}(1) \)
   b) \( \mathbf{e}(3) \)
   c) \( \mathbf{e}(2) = 4 \) (Note, this is an assignment)
   d) \( \mathbf{e}(4) \)
   e) \( \mathbf{e}(0) \)

2. You can ‘flip’ a vector/matrix by transposing it. In Matlab the character \( \prime \) is used for transpose:
   Try \( \mathbf{a}’ \) and \( \mathbf{b}’ \)

3. Vector Multiplication and vector addition works in Matlab according to rules in Algebra. Type (and try to understand):
   a) \( \mathbf{a} + \mathbf{b} \)
   b) \( \mathbf{b} - \mathbf{a} \)
c) \( c + e \) It does not work. Why?
d) \( a*b \) It does not work. Why?
e) \( a*a \). It does not work. Why?
f) \( a^2 \). The character ^ means 'power of'. Thus, it is the same as \( a*a \)
g) \( a'*b \) This is the dot product)
h) \( a.^2 \). denotes element-by-element powers. Try to understand what it is.
i) \( a.*a \) . * denotes element-by-element multiplication.

4. Assume you would like to plot the function \( f(x) = x^2 \) on the interval \([-2 2]\). According to the previous part of the lab, you perform the following steps:
- Create an x-axis, \( x = \text{linspace}(-2,2) \). Try it in Matlab!
- Next step is to calculate the corresponding function values. It will be either \( fx=x^2 \) or \( fx=x.*2 \) in Matlab. Which one will work and why? Try in Matlab!
- Finally, when \( x \) and \( fx \) are defined, plot the function: \( \text{plot}(x,fx) \)

5. Matrices (and vectors) are used when solving linear systems of equations.
One example: The equation system
\[
\begin{align*}
2x_1 + 4x_2 - 6x_3 &= -4 \\
x_1 + 5x_2 + 3x_3 &= 10 \\
x_1 + 3x_2 + 2x_3 &= 5
\end{align*}
\]
is written in matrix-vector notation \( Ax = b \), where
\[
A = \begin{pmatrix} 2 & 4 & -6 \\ 1 & 5 & 3 \\ 1 & 3 & 2 \end{pmatrix}, \quad b = \begin{pmatrix} -4 \\ 10 \\ 5 \end{pmatrix}
\]
Now, the system can be solved by \( x = A^{-1}b \), where \( A^{-1} \) is defined as the invers of \( A \).

Translate this to Matlab:
- In the Command window, define the matrix \( A \)
- Define the column vector \( b \)
- Finally solve the system: \( x = \text{inv}(A)*(b) \) (\( \text{inv}(A) \) is the inverse of \( A \) in Matlab)

If the result is correct \( Ax \) must be equal to \( b \). Check that in Matlab (compare \( A*x \) to the vector \( b \))!

6. Investigate some of the rules for matrix computations in Matlab:
   a) What is \( A.*A \) ?
   b) What is \( A.^2 \) ?
   c) What is \( A*A \) ?
   d) What is \( A(2,3) \) ?
   e) What does \( A(3,1) = -10 \) mean?
   f) Try to understand the command \( A(:,2) \)
   g) Try to understand the command \( A(3,:) \)
   h) Try to understand the command \( A(2:3,3) \)
   i) Let Matlab output the third column of \( A \) on the screen
7. There are a number of special, predefined matrices in Matlab. The most important ones are \texttt{zeros(m,n)}, \texttt{ones(m,n)}, \texttt{rand(m,n)} and \texttt{eye(n)}, where \(m\) and \(n\) are the number of rows and columns, respectively. Investigate these matrices in Matlab:
   
a) Type \texttt{zeros(8,6)} What is the result? Also read the help text (\texttt{help zeros}).
   
b) Type \texttt{ones(2,4)} What is the result? Also, read the help text.
   
c) Type \texttt{eye(5)} What is this matrix called? Again, read the help text.
   
d) Type \texttt{rand(6,4)} What is the result? Read the help text (\texttt{rand} is short for random).

8. In exercise 5 above you solved a linear system of equations using the inverse. However, using the inverse is not the best way to solve equation systems. It’s ‘expensive’, e.g. it takes more time and computer power than an alternative method, called Gaussian elimination in Algebra. In Matlab this more efficient method is implemented in the ‘backslash operator’, \texttt{\textbackslash}.

   It does not matter when you solve little problems (like the one in 5, with three unknowns), but it makes sense when working with thousands of unknowns. We will here do a little test, just to show the difference.

   Do the following in Matlab (it’s very important that you end the lines with semicolon):
   \begin{verbatim}
   m = 4000;
   A = rand(m,m);
   b = rand(m,1);
   tic; x = inv(A)*b; toc
   \end{verbatim}

   Here we create a random linear system in 4000 unknowns (4000 rows) and solve it using the inverse. The command \texttt{tic} and \texttt{toc} work together and measure the computation time. It will clock the time of the computations going on in between \texttt{tic} and \texttt{toc} (\texttt{tic} will start the clock and \texttt{toc} will end it).

   How long time did it take to solve the equation system (approximately)?

8. Now, repeat the last command again, but change the inverse-method to Gaussian elimination, e.g. backslash (use arrow-up on keyboard to repeat command, and arrow-left to change it):
   \begin{verbatim}
   tic; x = A\b; toc
   \end{verbatim}

   How long time did it take? How much quicker is Gaussian elimination?

   Both methods are solving exactly the same problems, but with different performance. This illustrates of the importance of choosing the best method (if possible).

\textbf{When you solve linear equations in Matlab, use backslash, \textbackslash}

\textbf{Three steps involved when you solve a linear system in Matlab:}
- Define the matrix (i.e. A)
- Define the right hand side (i.e. b)
- Solve the system with backslash (i.e. x = A\b )