FF505

Computational Science

Matrix Calculus

Marco Chiarandini (marco@imada.sdu.dk)

Department of Mathematics and Computer Science (IMADA)
University of Southern Denmark

Resume

- MATLAB, numerical computing vs symbolic computing
- MATLAB Desktop
- Script files
- 1D and 2D arrays
- Plot
- Interacting with matlab
- Matrix vs array operations
- Car market assignment

Other topics:

- matrices and vectors
- solving linear systems
- determinants
- linear transformation
- eigenvalues and eigenvectors
- diagonalization?

Outline

Creating Matrices

```
eye(4)  % identity matrix
zeros(4)  % matrix of zero elements
ones(4)  % matrix of one elements
```

```
A=rand(8)
triu(A)  % upper triangular matrix
tril(A)
diag(A)  % diagonal
```

```
>> [ eye(2), ones(2,3); zeros(2),
     [1:3;3:-1:1] ]

ans =

    1  0  1  1  1
    0  1  1  1  1
    0  0  1  2  3
    0  0  3  2  1
```

Can you create this matrix in one line of code?

```
 -5     0     0     0     0     0     0     1     1     1     1
  0    -4     0     0     0     0     0     0     1     1     1
  0     0    -3     0     0     0     0     0     0     1     1
  0     0     0    -2     0     0     0     0     0     0     1
  0     0     0     0    -1     0     0     0     0     0     0
  0     0     0     0     0     0     0     0     0     0     0
  0     0     0     0     0     0     1     0     0     0     0
  1     0     0     0     0     0     0     2     0     0     0
  1     1     0     0     0     0     0     0     3     0     0
  1     1     1     0     0     0     0     0     0     4     0
  1     1     1     1     0     0     0     0     0     0     5
```

Reshaping


```
%% reshape and replication
A = magic(3) % magic square
A = [A [0;1;2]]
reshape(A,[4 3]) % columnwise
reshape(A,[2 6])
v = [100;0;0]
A+v
A + repmat(v,[1 4])
```

Outline

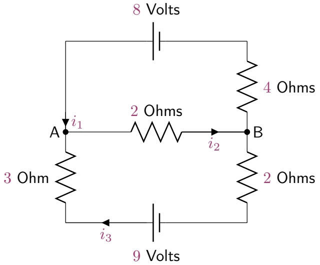
Dot and Cross Products

dot(A,B) inner or scalar product: computes the projection of a vector on the other. eg. dot(Fr,r) computes component of force $\mathbf{F}$ along direction $\mathbf{r}$

```
v=1:10
u=11:20
u*v' % inner or scalar product
ui=u+i
ui'
v*ui' % inner product of C^n
norm(v,2)
sqrt(v*v')
```

cross(A,B) cross product: eg: moment $\mathbf{M} = \mathbf{r} \times \mathbf{F}$

Electrical Networks



$$\begin{aligned} i_1 - i_2 + i_3 &= 0 \quad \text{node A} \\ -i_1 + i_2 - i_3 &= 0 \quad \text{node B} \\ 4i_1 + 2i_2 &= 8 \quad \text{top loop} \\ 2i_2 + 5i_3 &= 9 \quad \text{bottom loop} \end{aligned}$$

We want to determine the amount of current present in each branch.

Kirchoff's Laws

- At every node, the sum of the incoming currents equals the sum of the outgoing currents
- Around every closed loop, the algebraic sum of the voltage gains must equal the algebraic sum of the voltage drops.

Voltage drops V (by Ohm's law)

$$V = iR$$

Matrix Multiplication

$$
\begin{aligned}
i_1 - i_2 + i_3 &= 0 && \text{node A} \\
-i_1 + i_2 - i_3 &= 0 && \text{node B} \\
4i_1 + 2i_2 &= 8 && \text{top loop} \\
2i_2 + 5i_3 &= 9 && \text{bottom loop}
\end{aligned}
$$

$$
\begin{bmatrix} 1 & -1 & 1 \\ -1 & 1 & -1 \\ 4 & 2 & 0 \\ 0 & 2 & 5 \end{bmatrix}
\begin{bmatrix} i_1 \\ i_2 \\ i_3 \end{bmatrix}
=
\begin{bmatrix} 0 \\ 0 \\ 8 \\ 9 \end{bmatrix}
\qquad A\mathbf{x} = \mathbf{b}
$$

Chemical Equations

$$x_1 \mathrm{CO}_2 + x_2 \mathrm{H}_2\mathrm{O} \to x_3 \mathrm{O}_2 + x_4 \mathrm{C}_6\mathrm{H}_{12}\mathrm{O}_6$$

To balance the equation, we must choose x_1, x_2, x_3, x_4 so that the numbers of carbon, hydrogen, and oxygen atoms are the same on each side of the equation.

$$
\begin{aligned}
x_1 &= 6x_4 && \text{carbon atoms} \\
2x_1 + x_2 &= 2x_3 + 6x_4 && \text{oxygen} \\
2x_2 &= 12x_4 && \text{hydrogen}
\end{aligned}
$$

Matrix Multiplication

$$x_1 = 6x_4 \qquad \text{carbon atoms}$$

$$2x_1 + x_2 = 2x_3 + 6x_4 \qquad \text{oxygen}$$

$$2x_2 = 12x_4 \qquad \text{hydrogen}$$

$$\begin{bmatrix} 1 & 0 & 0 & -6 \\ 2 & 1 & 2 & 6 \\ 0 & 2 & 0 & 12 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \qquad A\mathbf{x} = \mathbf{0}$$

Matrix-Matrix Multiplication

In the product of two matrices A * B,
the number of columns in A must equal the number of rows in B.

The product AB has the same number of rows as A and the same number of columns as B. For example

```
>> A=randi(10,3,2) % returns a 3-by-2 matrix containing pseudorandom integer values
      drawn from the discrete uniform distribution on 1:10
A =
     6 10
    10 4
     5 8

>> C=randi(10,2,3)*100
C =
       1000 900 400
        200 700 200
>> A*C % matrix multiplication
ans =
       8000 12400 4400
      10800 11800 4800
       6600 10100 3600
```

Exercise: create a small example to show that in general, $AB \neq BA$.

Matrix Functions

Eigenvalues and eigenvectors:

```
A = ones(6)
trace(A)
A = A - tril(A)-triu(A,2)
eig(A)

diag(ones(3,1),-1)
[V,D]=eig(diag(1:4))

rank(A) % rank of A
orth(A) % orthonormal basis
```

Visualizing Eigenvalues

```
A=[5/4,0;0,3/4];
eigshow(A) %effect of operator A on unit
      verctor
```

Outline

Matrix Operations


```
%% matrix operations
A * C % matrix multiplication
B = [5 6; 7 8; 9 10] * 100 % same dims as A
A .* B % element-wise multiplcation
% A .* C or A * B gives error - wrong dimensions
A .^ 2
1./B
log(B) % functions like this operate element-wise on vecs or matrices
exp(B) % overflow
abs(B)
v = [-3:3] % = [-3 -2 -1 0 1 2 3]
-v % -1*v

v + ones(1,length(v))
% v + 1 % same

A' % (conjuate) transpose
```

Matrix and Array Operations

- Matrix operations follow the rules of linear algebra (not compatible with multidimensional arrays).
- Array operations execute element-by-element operations and support multidimensional arrays.
- The period character (.) distinguishes the array operations from the matrix operations.
- Array operations work on corresponding elements of arrays with equal dimensions
- scalar expansion: scalars are expanded into an array of the same size as the other input

Matrix vs Array Operations

- **Addition/Subtraction**: trivial
- **Multiplication**:
  - of an array by a scalar is easily defined and easily carried out.
  - of two arrays is not so straightforward:
    MATLAB uses two definitions of multiplication:
    - array multiplication (also called element-by-element multiplication)
    - matrix multiplication
- **Division and exponentiation** MATLAB has two forms on arrays.
  - element-by-element operations
  - matrix operations

$\rightsquigarrow$ Remark:
the operation division by a matrix is not defined. In MatLab it is defined but it has other meanings.

Array Operations (Element-by-Element)

Symbol	Operation	Form	Examples
+	Scalar-array addition	A + b	[6,3]+2=[8,5]
-	Scalar-array subtraction	A - b	[8,3]-5=[3,-2]
+	Array addition	A + B	[6,5]+[4,8]=[10,13]
-	Array subtraction	A - B	[6,5]-[4,8]=[2,-3]
.*	Array multiplication	A.*B	[3,5].*[4,8]=[12,40]
./	Array right division	A./B	[2,5]./[4,8]=[2/4,5/8]
.\	Array left division	A.\B	[2,5].\[4,8]=[2\4,5\8]
.^	Array exponentiation	A.^B	[3,5].^2=[3^2,5^2]
			2.^[3,5]=[2^3,2^5]
			[3,5].^[2,4]=[3^2,5^4]

Matrix Operations

`*`	Matrix multiplication	`C = A*B` is the linear algebraic product of the matrices A and B. The number of columns of A must equal the number of rows of B.
`\`	Matrix left division (`mldivide`)	`x = A\B` is the solution to the equation $Ax = B$. Matrices A and B must have the same number of rows.
`/`	Matrix right division (`mrdivide`)	`x = B/A` is the solution to the equation $xA = B$. Matrices A and B must have the same number of columns. In terms of the left division operator, `B/A = (A'\B')'`.
`^`	Matrix power	`A^B` is A to the power B, if B is a scalar. For other values of B, the calculation involves eigenvalues and eigenvectors.
`'`	Complex conjugate transpose	`A'` is the linear algebraic transpose of A. For complex matrices, this is the complex conjugate transpose.

Matrix division

Backslash or matrix left division `A\B`
It is roughly like `INV(A)*B` except that it is computed in a different way:
`X = A\B` is the solution to the equation `A*X = B` computed by Gaussian elimination.

Slash or right matrix division `A/B`
`X = A/B` is the solution to the equation `X*A = B`. It is the matrix division of `B` into `A`, which is roughly the same as `A*INV(B)`, except it is computed in a different way. More precisely, `A/B = (B'\A')'`.

Algorithms:
http://www.maths.lth.se/na/courses/NUM115/NUM115-11/backslash.html