Computational Physics II: Introduction to Matlab and Numerical Calculus

รศ.ดร. ชรินทร์ โหมดชัง

Assoc. Prof. Dr. Charin Modchang

Department of Physics,
Faculty of Science, Mahidol University

1.0 Basic matrix operations

Definition of Matrix Addition

If $A = [a_{ij}]$ and $B = [b_{ij}]$ are matrices of size $m \times n$, then their **sum** is the $m \times n$ matrix given by

$$A + B = [a_{ij} + b_{ij}].$$

The sum of two matrices of different sizes is undefined.

Definition of Scalar Multiplication

If $A = [a_{ij}]$ is an $m \times n$ matrix and c is a scalar, then the **scalar multiple** of A by c is the $m \times n$ matrix given by

$$cA = [ca_{ij}].$$

Definition of Matrix Multiplication

If $A = [a_{ij}]$ is an $m \times n$ matrix and $B = [b_{ij}]$ is an $n \times p$ matrix, then the **product** AB is an $m \times p$ matrix

$$AB = [c_{ij}]$$

where

$$c_{ij} = \sum_{k=1}^{n} a_{ik}b_{kj} = a_{i1}b_{1j} + a_{i2}b_{2j} + a_{i3}b_{3j} + \cdot\cdot\cdot + a_{in}b_{nj}.$$

1.0 Basic matrix operations

Definition of the Determinant of a 2 × 2 Matrix

The **determinant** of the matrix

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$$

is given by

$$\det(A) = |A| = a_{11}a_{22} - a_{21}a_{12}.$$

Definitions of Minors and Cofactors of a Matrix

If A is a square matrix, then the **minor** M_{ij} of the element a_{ij} is the determinant of the matrix obtained by deleting the ith row and jth column of A. The **cofactor** C_{ij} is given by

$$C_{ij} = (-1)^{i+j}M_{ij}.$$

Definition of the Determinant of a Matrix

If A is a square matrix (of order 2 or greater), then the determinant of A is the sum of the entries in the first row of A multiplied by their cofactors. That is,

$$\det(A) = |A| = \sum_{j=1}^{n} a_{1j}C_{1j} = a_{11}C_{11} + a_{12}C_{12} + \cdot\ \cdot\ \cdot + a_{1n}C_{1n}.$$

1.1 Basic Elements of Matlab

Variables

Matlab has only one data type: matrix.

$$x = 5; \quad y = 8; \quad \vec{a} = \begin{bmatrix} 3 & 8 & -1 \end{bmatrix}; \quad \vec{b} = \begin{bmatrix} 1 \\ 5 \\ 3 \end{bmatrix}$$

$$C = \begin{bmatrix} 2 & 5 & 7 \\ 8 & 3 & 1 \\ 0 & -5 & 8 \end{bmatrix}; \quad D = \begin{bmatrix} 3 & x & 6 \\ pi & 4 & 2 \\ 1 & \sqrt{-1} & 3 \end{bmatrix}$$

```
x = 5;
y = 8;
a = [3 8 -1];
b = [1; 5; 3];
C = [2 5 7; 8 3 1; 0 -5 8];
D = [3 x 6; pi 4 2; 1 sqrt(-1) 3];
```

1.1 Basic Elements of Matlab

Arithmetic Operations

All arithmetic operations in Matlab are matrix operations.

```
z = x + y;
g = a*b;
E = C + D;
F = C*D;
G = C/D;   %this is equivalent to C*inv(D)
H = C^2;   %power operator
J = D'   %Hemitian conjugate
K = C.'   %Transpose
```

Element by element operations:

```
L = C.*D;
M  = C./D;
N = C.^2;
```

1.1 Basic Elements of Matlab

Loops and conditionals

```
for i=1:5   %Your basic loops; i goes from 1 to 5 with the default step 1
   p(i) = 2^i;
end         %This is the end of the loop

for i=1:2:5   %Your basic loops; i goes from 1 to 5 with the step 2
   q(i) = 2^i;
end         %This is the end of the loop

while (x>1)
   x = x/2
end
```

1.1 Basic Elements of Matlab

Loops and conditionals

```
if (k > 5)    % A simple conditional
   a = 2;
end

if (m>= 5 && m <= 8)  %Another conditional using else
   a = 3;
else
   a = 4;
end

if (m>= 5 && m <= 8)
   a = 3;
elseif (m == 10)     %Conditional using elesif
   a = 4;
end
```

1.1 Basic Elements of Matlab

Colon operator

The colon operator can be used to create a vector.

Consider

Demonstration: colon_op1.m and colon_op2.m

(1)

```
tau = 0.1;
for i=1:100
    time(i) = tau*i;
end
```

(2)

```
tau = 0.1;
i = 1:100;    %The colon operator can be used to create a vector
time = tau*i;
```

How can we increase the speed of (1)?

The colon operator is also useful for selecting parts of a matrix.

- z = B(:,2) assigns the second column of matrix B to the vector z.

1.1 Basic Elements of Matlab

Input and Output

```
x = input('Enter the value of x: ');
a = input('Answer <yes> or <no>: ','s')
```

```
disp('The value of x is ')
disp(x)
```

```
fprintf('The value of x is %g \n',x);
```

other useful commands:

save :save all variables in the workspace
load :load the saved data

```
>>save ABC A B C %store the values of A,B,C into the file ‘ABC.mat’
>>load ABC A C %read the values of A,C from the file ‘ABC.mat’
>>clear A C %clear the memory of MATLAB about A,C
```

1.1 Basic Elements of Matlab

Some useful build-in functions

Table 1.1: Selected MATLAB mathematical functions.

`abs(x)`	Absolute value or complex magnitude
`norm(x)`	Magnitude of a vector
`sqrt(x)`	Square root
`sin(x)`, `cos(x)`	Sine and cosine
`tan(x)`	Tangent
`atan2(y,x)`	Arc tangent of `y/x` in $[0, 2\pi]$
`exp(x)`	Exponential
`log(x)`, `log10(x)`	Natural logarithm and base-10 logarithm
`rem(x,y)`	Remainder (modulo) function (e.g., `rem(10.3,4)=2.3`)
`floor(x)`	Round down to nearest integer (e.g., `floor(3.2)=3`)
`ceil(x)`	Round up to nearest integer (e.g., `ceil(3.2)=4`)
`rand(N)`	Uniformly distributed random numbers from the interval $[0, 1)$. Returns $N \times N$ matrix.
`randn(N)`	Normal (Gaussian) distributed random numbers (zero mean, unit variance). Returns $N \times N$ matrix.

1.1 Basic Elements of Matlab

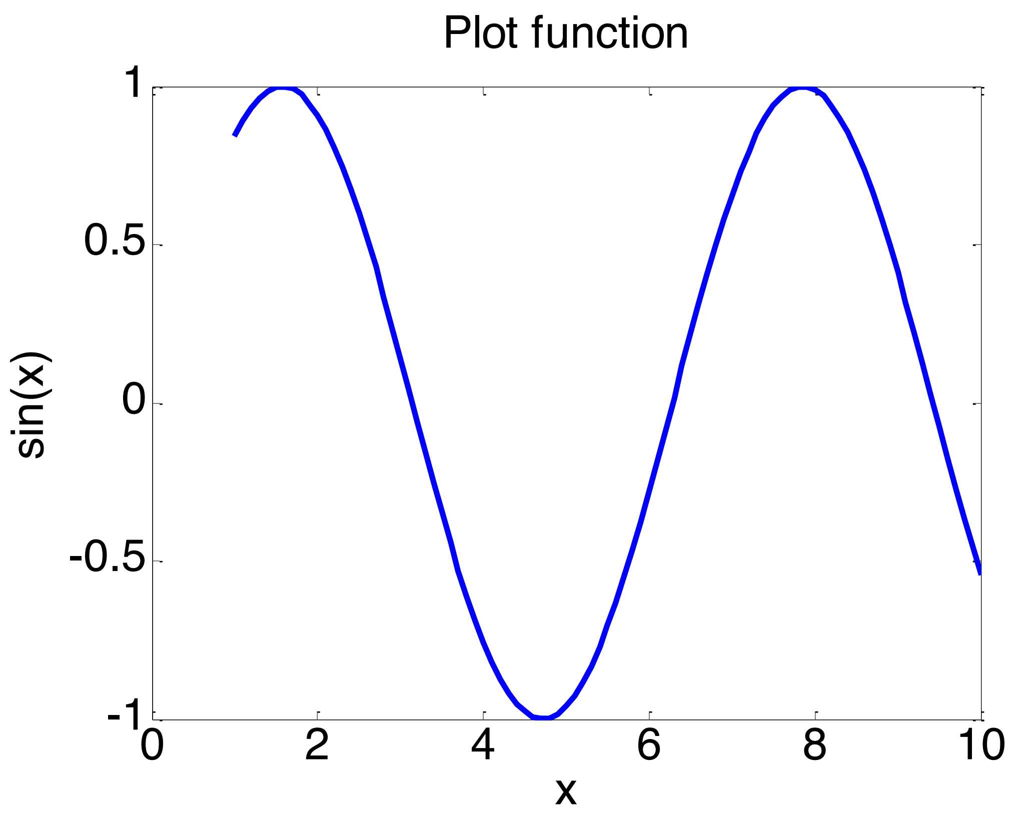
Some useful build-in functions

inv(x)	:Inverse of the matrix x
plot(x,y)	:Plot vector x versus vector y
semilogx(x,y)	:Semilog plot
semilogy(x,y)	:Semilog plot
loglog(x,y)	:loglog plot
polar(theta,rho)	:Polar plot
zeros(N)	:Create an N-by-N matrix with all elements set to zero
ones(N)	:Create an N-by-N matrix with all elements set to one

1.1 Basic Elements of Matlab

2D Graphic

```
x=1:0.1:10;
y=sin(x);
plot(x,y)
xlabel('x')
ylabel('sin(x)')
title('Plot function')
```



1.1 Basic Elements of Matlab

2D Graphic

```
%nm114_2: plot several types of graph
th = [0: .02:1]*pi;
subplot(221), polar(th,exp(-th))
subplot(222), semilogx(exp(th))
subplot(223), semilogy(exp(th))
subplot(224), loglog(exp(th))
pause, clf
subplot(221), stairs([1 3 2 0])
subplot(222), stem([1 3 2 0])
subplot(223), bar([2 3; 4 5])
subplot(224), barh([2 3; 4 5])
pause, clf
y = [0.3 0.9 1.6 2.7 3 2.4];
subplot(221), hist(y,3)
subplot(222), hist(y,0.5 + [0 1 2])
```

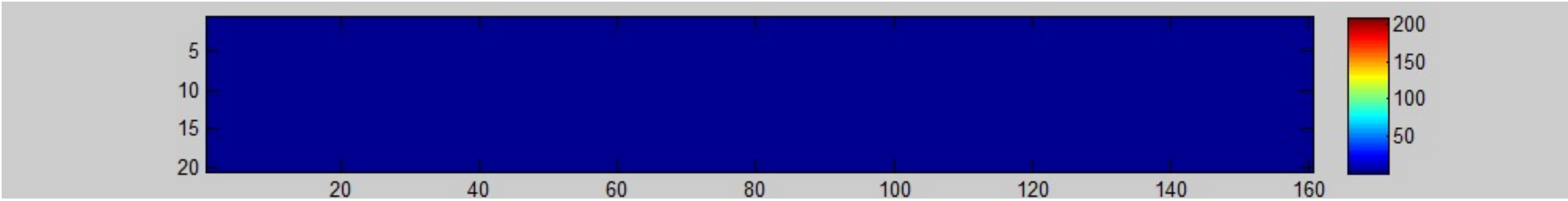
1.1 Basic Elements of Matlab

3D Graphic

```
%nm115: to plot 3D graphs
t = 0:pi/50:6*pi;
expt = exp(-0.1*t);
xt = expt.*cos(t); yt = expt.*sin(t);
%dividing the screen into 2 x 2 sections
subplot(221), plot3(xt, yt, t), grid on %helix
subplot(222), plot3(xt, yt, t), grid on, view([0  0 1])
subplot(223), plot3(t, xt, yt), grid on, view([1 -3 1])
subplot(224), plot3(t, yt, xt), grid on, view([0 -3 0])
pause, clf
x = -2:.1:2;  y = -2:.1:2;
[X,Y] = meshgrid(x,y); Z = X.^2 + Y.^2;
subplot(221), mesh(X,Y,Z), grid on %[azimuth,elevation] = [-37.5,30]
subplot(222), mesh(X,Y,Z), view([0,20]), grid on
pause, view([30,30])
subplot(223), contour(X,Y,Z)
subplot(224), contour(X,Y,Z,[.5,2,4.5])
```

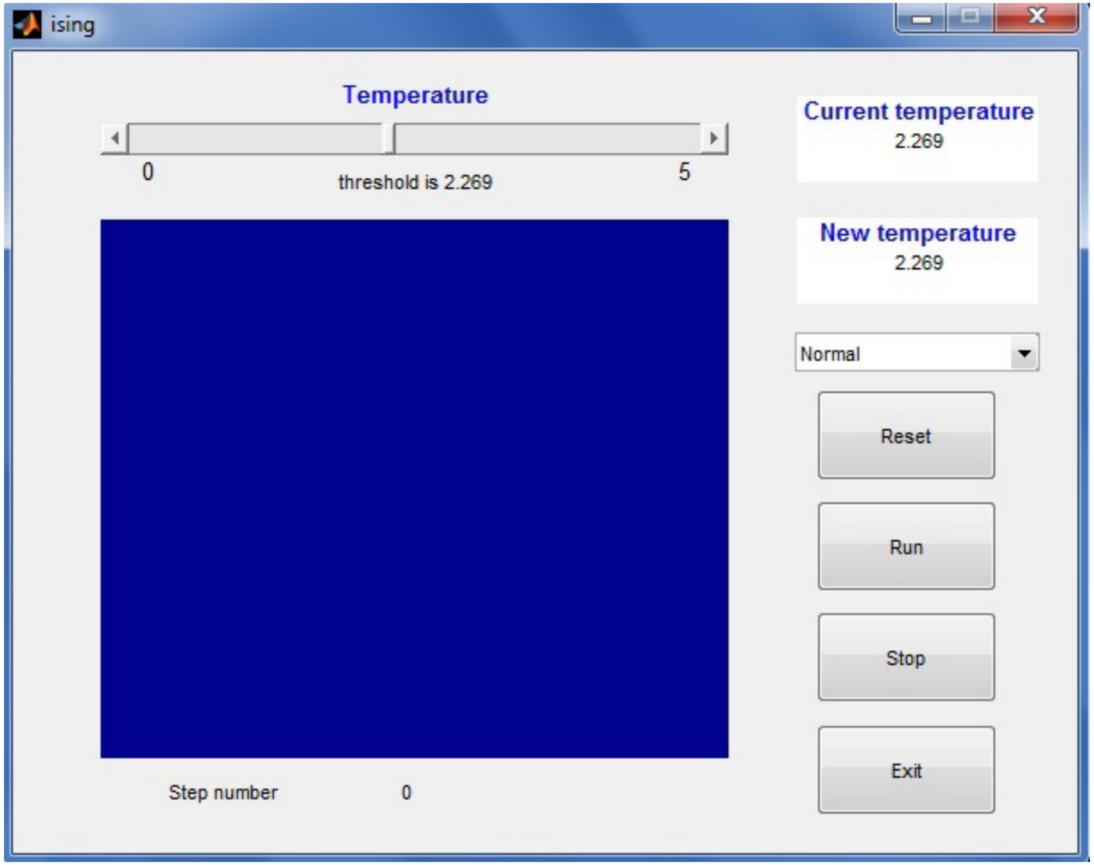
1.1 Basic Elements of Matlab

Movies



Graphic User Interface (GUI)

Ising.m



1.1 Basic Elements of Matlab

If you don’t know, ask Matlab

1. If you know the function name, but don’t know how to use it: help ‘function name’ (e. g. help for)

2. If you want to find the MATLAB commands/functions which are related with a job: lookfor ‘job’ (e.g. lookfor repeat)

1.1 Basic Elements of Matlab

Simple Matlab Program

Orthogonality program: orthog.m

Table 1.4: Outline of program `orthog`, which evaluates the dot product of a pair of three dimensional vectors.

- Initialize the vectors $\mathbf{a}$ and $\mathbf{b}$.
- Evaluate the dot product as $\mathbf{a} \cdot \mathbf{b} = \mathbf{a}_1\mathbf{b}_1 + \mathbf{a}_2\mathbf{b}_2 + \mathbf{a}_3\mathbf{b}_3$.
- Print dot product and state whether vectors are orthogonal.

Exercise: Modify the orthogonality program so that it can handle vectors of any length.

1.1 Basic Elements of Matlab

Simple Matlab Program

If we have 3 data points, what is the simplest equation of a curve which passes through all data points?

The Lagrange form of the interpolation polynomial which passes 3 data points is

$$p(x) = \frac{(x-x_2)(x-x_3)}{(x_1-x_2)(x_1-x_3)}y_1 + \frac{(x-x_1)(x-x_3)}{(x_2-x_1)(x_2-x_3)}y_2 + \frac{(x-x_1)(x-x_2)}{(x_3-x_1)(x_3-x_2)}y_3$$

where (x_1,y_1), (x_2, y_2), (x_3,y_3) are the three data points to be fitted. Commonly, such polynomials are used to interpolate between data points.

1.1 Basic Elements of Matlab

Simple Matlab Program

Interpolation program: interp.m, intrpf.m

Table 1.6: Outline of function `intrpf`, which evaluates the Lagrange quadratic (1.5).

- *Inputs*: $\mathbf{x} = [x_1\ x_2\ x_3]$, $\mathbf{y} = [y_1\ y_2\ y_3]$, and x^*
- *Outputs*: y^*
- Calculate $y^* = p(x^*)$ using the Lagrange polynomial (1.5).

1.1 Basic Elements of Matlab

Simple Matlab Program

Interpolation program: interp.m, intrpf.m

In general, if we have *n* data points, **(ASK)**

$$p(x) = \sum_{j=1}^{n} p_j f_{nj}(x)$$

$$\text{where } f_{nj}(x) = \prod_{k \neq j}^{n} \frac{x - x_k}{x_j - x_k}$$

1.2 Numerical Errors

1. Round-off error

In Matlab eps = 2.2204e-016

Try to calculate the following in Matlab

1. (1+eps)-1
2. (2+eps)-2

So, what is the value of this expression $$\frac{10^{-20}}{\left(3+10^{-20}\right)-3}$$

2. Range Error

Try to calculate these in Matlab: 10^400 and 10^(-400). Check if 10^400 = 10^500

The maximum range of number that Matlab can represent is about $10^{\pm 308}$

Inf = Infinity

NaN = Not a Number, e. g., 0/0

1.2 Numerical Errors

3. Loss of Significant

$$f_1(x) = \sqrt{x}(\sqrt{x+1} - \sqrt{x}), \qquad f_2(x) = \frac{\sqrt{x}}{\sqrt{x+1} + \sqrt{x}}$$

Program: nm122

1.3 Numerical Differentiation

Naïve Formula

Definition $f'(x) \equiv \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}$

We also have the Taylor series expressed as

$$f(x+h) = f(x) + hf'(x) + \frac{1}{2}h^2 f''(x) + \ldots$$

So, the first order derivative can be written as

$$\boxed{f'(x) = \frac{f(x+h) - f(x)}{h} + O(h)}$$

Truncation Error

1.3 Numerical Differentiation

Centered Formula

An equivalent definition for the derivative is

$$f'(x) \equiv \lim_{h \to 0} \frac{f(x+h) - f(x-h)}{2h}$$

Using the Taylor series expansion

$$f(x+h) = f(x) + hf'(x) + \frac{1}{2}h^2 f''(x) + \frac{1}{6}h^3 f'''(x) + \ldots$$

$$f(x-h) = f(x) - hf'(x) + \frac{1}{2}h^2 f''(x) - \frac{1}{6}h^3 f'''(x) + \ldots$$

What is the truncation error?

The first derivative can be written as

$$\boxed{f'(x) = \frac{f(x+h) - f(x-h)}{2h} + O(h^2)}$$

Truncation Error

1.3 Numerical Differentiation

Exercise: Derive the second derivative using the Taylor expansion

Answer: $$\boxed{f''(x) = \frac{f(x+h) - 2f(x) + f(x-h)}{h^2} + O\left(h^2\right)}$$

Demonstration: deriv.m and fund_deriv.m

1.4 Numerical Integration

Trapezoid rule

Consider the integral

$$I = \int_a^b f(x)dx$$

Our strategy for estimating *I* is to evaluate *f*(*x*) at a few points and fit a simple curve (e.g., piecewise linear) through these points.

Trapezoid rule

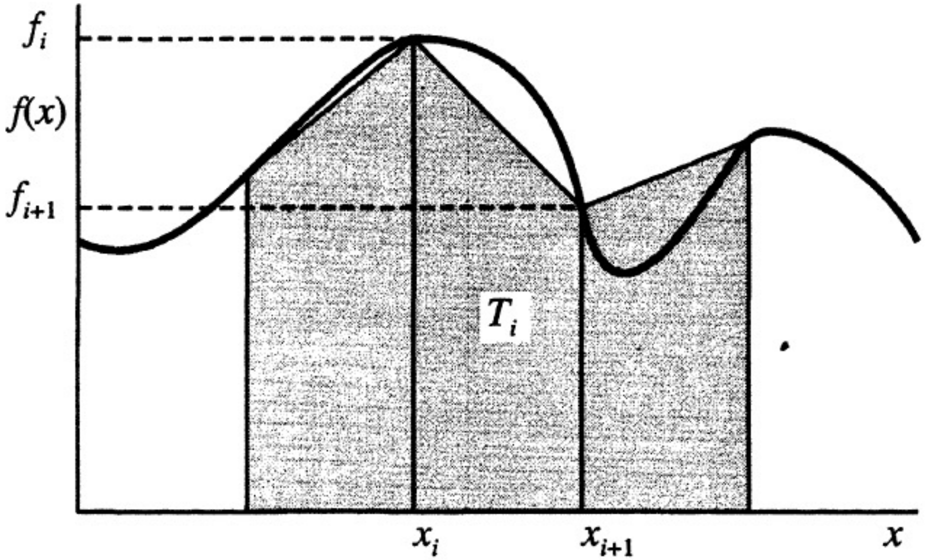


Figure 10.3: General trapezoidal rule for estimating integrals.

The area of a single trapezoid is $$T_i = \frac{1}{2}(x_{i+1} - x_i)(f_{i+1} + f_i)$$

where $f_i \equiv f(x_i)$

1.4 Numerical Integration

Trapezoid rule

The true integral is estimated as the sum of the areas of the trapezoids, so

$$I \approx I_T = T_1 + T_2 + \ldots + T_{N-1}$$

If we take equal spacing $h = \frac{b-a}{N-1}$, so $x_i = a + (i-1)h$

Then $$T_i = \frac{1}{2}h(f_{i+1} + f_i)$$

The trapezoidal rule for equally spaced points is

$$\boxed{\begin{aligned} I_T(h) &= \frac{1}{2}hf_1 + hf_2 + hf_3 + \ldots + hf_{N-1} + \frac{1}{2}hf_N \\ &= \frac{1}{2}h(f_1 + f_N) + h\sum_{i=2}^{N-1} f_i \end{aligned}}$$

1.4 Numerical Integration

Trapezoid rule

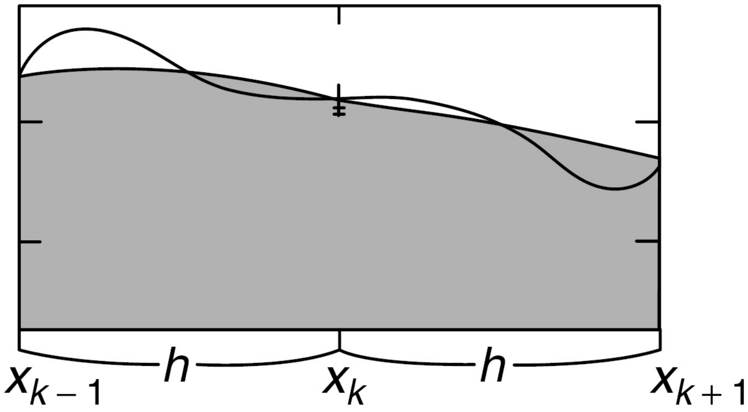
Most numerical analysis texts give the truncation error for the trapezoidal rule as

$$I - I_T(h) = -\frac{1}{12}(b-a)h^2 f''(\zeta)$$

Where there exists $a \le \zeta \le b$

1.4 Numerical Integration

Simpson rule



$$f(x) = \frac{(x - x_i)(x - x_{i+1})}{(x_{i-1} - x_i)(x_{i-1} - x_{i+1})} f_{i-1} + \frac{(x - x_{i-1})(x - x_{i+1})}{(x_i - x_{i-1})(x_i - x_{i+1})} f_i$$

$$+ \frac{(x - x_{i-1})(x - x_i)}{(x_{i+1} - x_{i-1})(x_{i+1} - x_i)} f_{i+1} + O(h^3).$$

$$\boxed{S = \frac{h}{3} \sum_{j=0}^{n/2-1} (f_{2j} + 4 f_{2j+1} + f_{2j+2}) + O(h^4),}$$

Homework

(1)

Operations on Vectors

(a) Find the mathematical expression for the computation to be done by the following MATLAB statements.

```
>>n = 0:100; S = sum(2.^-n)
```

(b) Write a MATLAB statement that performs the following computation.

$$\left(\sum_{n=0}^{10000} \frac{1}{(2n+1)^2} \right) - \frac{\pi^2}{8}$$

(c) Write a MATLAB statement which uses the commands `prod()` and `sum()` to compute the product of the sums of each row of a 3×3 random matrix.

Homework

2. Write a Malab code to numerically differentiate the following functions using right, left, and center derivatives. Your program should also allow the user to specify the desired maximum error.

1.1 $x^2 + x + 5$

1.2 $\cos(x)$

1.3 $(\sin(x) - 1)^4$

ส่งการบ้านมาที่

Email: homework.charin@gmail.com

อย่าลืมเขียน ชื่อ นามสกุล และรหัสนักศึกษาทุกครั้ง ทั้งในตัวอีเมลและในไฟล์การบ้าน