PROGRAMMING TEST PROBLEMS

OUTLINE

- MATLAB Plot Function
- MATLAB Symbolic Integration
- MATLAB Built-in Numerical Integration
- Simpson’s Rule

MATLAB PLOT FUNCTION

Functions

plot	2-D line plot
plotyy	2-D line plots with y-axes on both left and right side
plot3	3-D line plot

loglog	Log-log scale plot
semilogx	Semilogarithmic plot
semilogy	Semilogarithmic plot

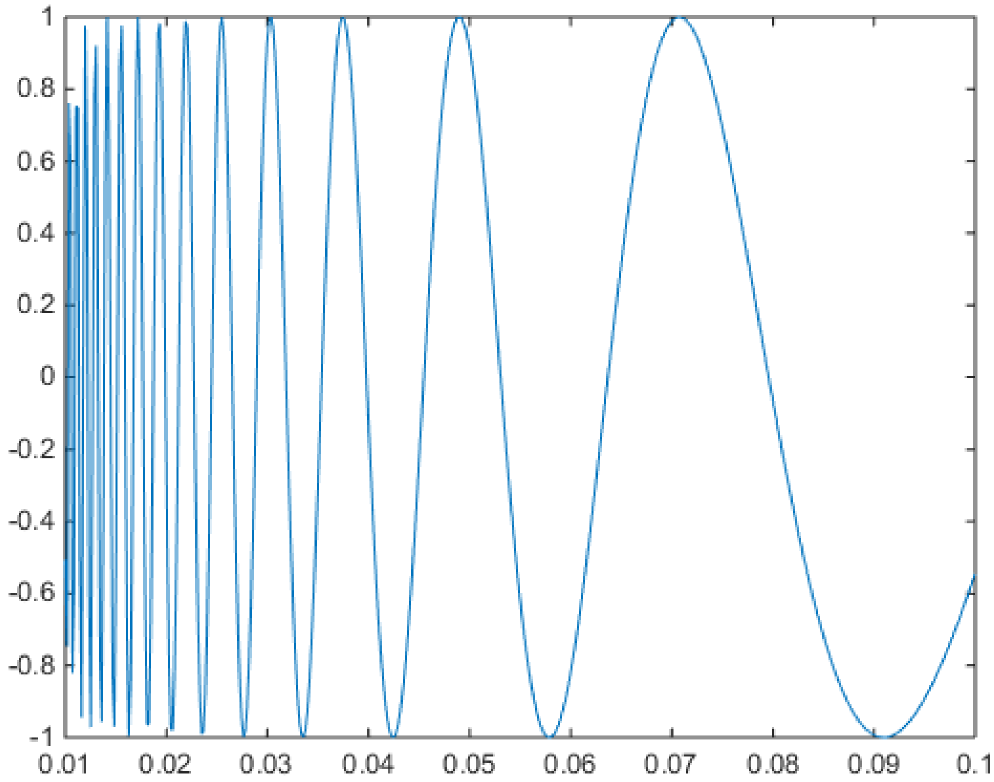
errorbar	Plot error bars along curve

fplot	Plot function between specified limits
ezplot	Easy-to-use function plotter
ezplot3	Easy-to-use 3-D parametric curve plotter

MATLAB PLOT FUNCTION

Create a function handle from an anonymous function. Plot the function from 0.01 to 0.1.

```
sn = @(x) sin(1./x);
fplot(sn,[0.01,0.1])
```



MATLAB SYMBOLIC INTEGRATION

Indefinite Integral of a Univariate Expression

Find an indefinite integral of this univariate expression:

```
syms x
int(-2*x/(1 + x^2)^2)
```

```
ans =
1/(x^2 + 1)
```

MATLAB SYMBOLIC INTEGRATION

Approximate Definite Integrals

Compute this definite integral:

```
syms x
F = int(cos(x)/sqrt(1 + x^2), x, 0, 10)
```

```
F =
int(cos(x)/(x^2 + 1)^(1/2), x, 0, 10)
```

If `int` cannot compute a closed form of a definite integral, try approximating that integral numerically using `vpa`. For example, approximate `F` with five significant digits:

```
vpa(F, 5)
```

```
ans =
0.37571
```

MATLAB BUILT-IN NUMERICAL INTEGRATION

Functions

`integral`	Numerical integration
`integral2`	Numerically evaluate double integral
`integral3`	Numerically evaluate triple integral
`quad`	Numerically evaluate integral, adaptive Simpson quadrature
`quadgk`	Numerically evaluate integral, adaptive Gauss-Kronrod quadrature
`quad2d`	Numerically evaluate double integral, tiled method
`cumtrapz`	Cumulative trapezoidal numerical integration
`trapz`	Trapezoidal numerical integration
`polyint`	Integrate polynomial analytically

MATLAB BUILT-IN NUMERICAL INTEGRATION

- **Q = quad(fun,xmin,xmax)** numerically integrates function fun from xmin to xmax using **adaptive Simpson quadrature** and default error tolerances

$$\int_0^2 \frac{1}{x^3 - 2x - 5}\,dx,$$

```
Q = quad(@myfun,0,2)

Q =

   -0.4605
```

```
function y = myfun(x)
y = 1./(x.^3-2*x-5);
```

```
F = @(x)1./(x.^3-2*x-5);
Q = quad(F,0,2);
```

MATLAB BUILT-IN NUMERICAL INTEGRATION

- **Q = trapz(X,Y)** integrates Y with spacing increment X

Create a domain vector, X.

```
X = 0:pi/100:pi;
```

Calculate the sine of X and store the result in Y.

```
Y = sin(X);
```

Integrate the function values contained in Y using `trapz`.

```
Q = trapz(X,Y)
```

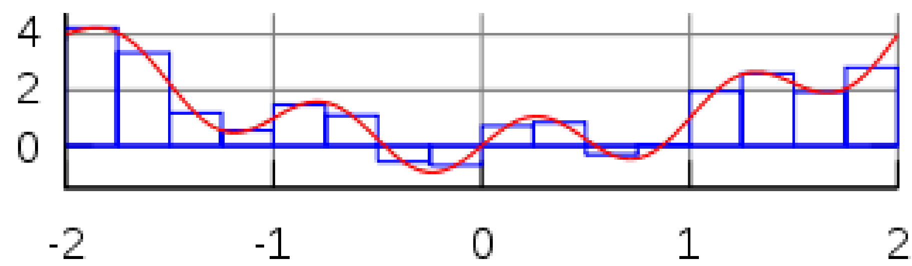
```
Q =

    1.9998
```

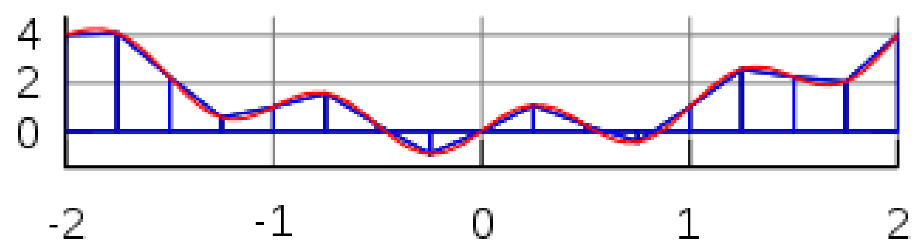
NUMERICAL INTEGRATION

Midpoint/Rectangle Rule vs Trapezoidal Rule

$$\int_a^b f(x)\,dx \approx (b-a)\, f\left(\frac{a+b}{2}\right).$$

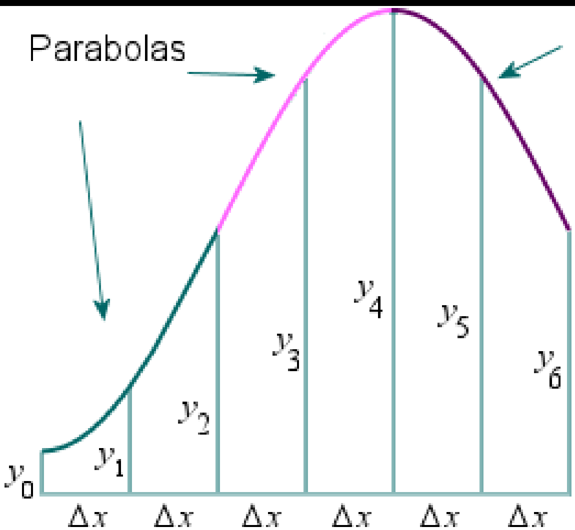


$$\int_a^b f(x)\,dx \approx (b-a)\,\frac{f(a)+f(b)}{2}.$$



Numerical Integration Demonstration

SIMPSON'S RULE



Simpson's Rule

$$\text{Area} = \int_a^b f(x)\,dx \approx \frac{\Delta x}{3}(y_0 + 4y_1 + 2y_2 + 4y_3 + 2y_4 \ldots + 4y_{n-1} + y_n)$$

where $\Delta x = \dfrac{b-a}{n}$.

Note: In Simpson's Rule, n must be EVEN.

Memory aid

We can re-write Simpson's Rule by grouping it as follows:

$$\int_a^b f(x)\,dx \approx \frac{\Delta x}{3}(y_0 + 4(y_1 + y_3 + y_5 + \ldots) + 2(y_2 + y_4 + y_6 + \ldots) + y_n)$$

This gives us an easy way to remember Simpson's Rule:

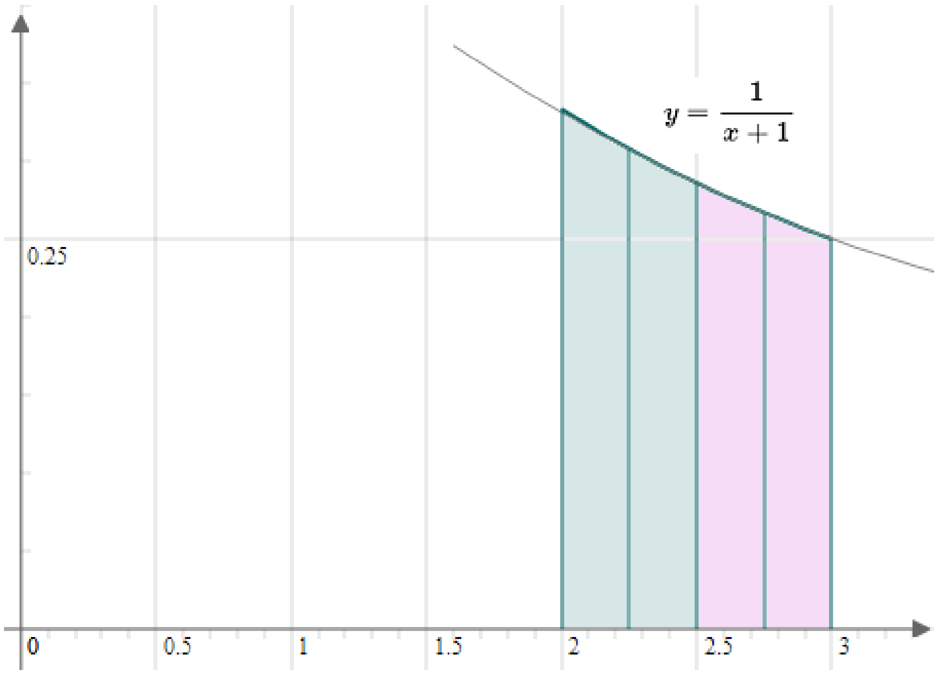
$$\int_a^b f(x)\,dx \approx \frac{\Delta x}{3}(\text{FIRST} + 4(\text{sum of ODDs}) + 2(\text{sum of EVENs}) + \text{LAST})$$

SIMPSON'S RULE

$$\text{Area} = \int_a^b f(x)\mathrm{d}x$$

$$\approx \frac{0.25}{3}(0.333333 + 4(0.3076923) + 2(0.2857142) + 4(0.2666667) + 0.25)$$

$$= 0.2876831$$



$$\Delta x = \frac{b-a}{n} = \frac{3-2}{4} = 0.25$$

$$y_0 = f(a) = f(2) = \frac{1}{2+1} = 0.3333333$$

$$y_1 = f(a+\Delta x) = f(2.25) = \frac{1}{2.25+1} = 0.3076923$$

$$y_2 = f(a+2\Delta x) = f(2.5) = \frac{1}{2.5+1} = 0.2857142$$

$$y_3 = f(a+3\Delta x) = f(2.75) = \frac{1}{2.75+1} = 0.2666667$$

$$y_4 = f(b) = f(3) = \frac{1}{3+1} = 0.25$$

SIMPSON'S RULE

Editor - E:\Downloads\simpsons\simpsons.m

simpsons.m

```matlab
function I = simpsons(f,a,b,n)
    h=(b-a)/n; xi=a:h:b;
    I= h/3*(f(xi(1))+2*sum(f(xi(3:2:end-2)))+4*sum(f(xi(2:2:end)))+f(xi(end)));
```

Command Window

New to MATLAB? See resources for Getting Started.

```
>> f=@(x) 1./(x+1);
>> format long
>> I=simpsons(f,2,3,4)

I =

   0.287683150183150
```

Command Window

New to MATLAB? See resources for Getting Started.

```
>> f=@(x) ((x-1).*x./2).*((x-1).*x./2);
>> I=simpsons(f,-1,1,2)

I =

    0.3333
```

COMMAND M-FILE

Editor - E:\Downloads\simpsons\Command.m

Command.m

```
f=@(x) x.^2.*exp(x);
fplot(f,[1,3]);
syms x;
fanalytic=int(x.^2.*exp(x))
areaanalytic=int(x.^2.*exp(x),1,3)
areaval=vpa(areaanalytic)
x=1:0.1:3;
areatrapz=trapz(x,f(x))
areaquad=quad(f,1,3)
areasimpsons=simpsons(f,1,3,10)
```

REFERENCE WEBSITES

- **Line Plots**

  http://www.mathworks.com/help/matlab/line-plots.html

- **Numerical Integration and Differentiation**

  http://www.mathworks.com/help/symbolic/int.html

  http://www.mathworks.com/help/matlab/ref/quad.html

  http://www.mathworks.com/help/matlab/numerical-integration-and-differentiation.html

- **Comparing Basic Numerical Integration Methods**

  http://demonstrations.wolfram.com/ComparingBasicNumericalIntegrationMethods/