MATLAB – Numerical Integration

Trapezoidal rule with equal segments. To compute $\int_a^b f(x)dx$ using trapezoidal rule, one can use the Matlab built-in function `trapz(x,y)` or the following function:

```
function I = trap(func,a,b,n,varargin)
% trap: composite trapezoidal rule quadrature
% I = trap(func,a,b,n,pl,p2,...):
% composite trapezoidal rule
% input:
% func = name of function to be integrated
% a, b = integration limits
% n = number of segments (default = 100)
% pl,p2,... = additional parameters used by func
% output:
% I = integral estimate
if nargin<3,
    error('at least 3 input arguments required'),
end
if ~(b>a),
    error('upper bound must be greater than lower'),
end
if nargin<4|isempty(n),
    n = 100;
end

x = a;
h = (b - a)/n;
s = func(a,varargin{:});
for i = 1 : n - 1
    x = x + h;
    s = s + 2*func(x,varargin{:});
end
s = s + func(b,varargin{:});
```

```
I = (b - a) * s/(2*n);
end
```

Use both functions to find an estimate for $\int_0^1 \sin x \, dx$ with $n = 50$.

Trapezoidal rule with unequal segments. To compute $\int_a^b f(x)dx$ when the segments are not equal, use the following function `trapuneq(x,y)`.

```
function I = trapuneq(x,y)
% trapuneq: unequal spaced trapezoidal rule quadrature
% I = trapuneq(x,y):
% Applies the trapezoidal rule to determine the integral
% for n data points (x, y) where x and y must be of the
% same length and x must be monotonically ascending
% input:
% x = vector of independent variables
% y = vector of dependent variables
% output:
% I = integral estimate
if nargin<2,
    error('at least 2 input arguments required'),
end
if any(diff(x)<0),
    error('x not monotonically ascending'),
end
n = length(x);
if length(y)~= n,
    error('x and y must be same length');
end
s = 0;
for k = 1:n - 1
    s = s + (x(k+l) - x(k))*(y(k)+ y(k +l))/2;
end
I = s;
```

For example, try the following

```
>> x = [0 .12 .22 .32 .36 .4 .44 .54 .64 .7 .8];
>> y = 0.2 + 25*x - 200*x.^2 + 675*x.^3 - 900*x.^4 + 400*x.^5;
>> trapuneq(x,y)
```

Cumulative integral using trapezoidal rule. The following built-in function computes the cumulative integral:

```
>> I=cumtrapz(x,y)
```

Try the following

```
>> x = [0 .12 .22 .32 .36 .4 .44 .54 .64 .7 .8];
>> y = 0.2 + 25*x - 200*x.^2 + 675*x.^3 - 900*x.^4 + 400*x.^5;
>> cumtrapz(x,y)
```

Double and Triple Integrals. MATLAB has functions to implement both double (integral2) and triple (integral3) integration. A simple representation of the syntax for integral2 is

```
>> q = integral2(fun, xmin, xmax, ymin, ymax)
```

where `q` is the double integral of the function `fun` over the ranges from `xmin` to `xmax` and `ymin` to `ymax`. Here is an example of how this function can be used to compute the double integral:

```
>> q = integral2(@(x,y) 2*x*y + 2*x - x.^2 - 2*y.^2 + 72,0,8,0,6)
```

Example. Integrate $f(x,y) = \frac{1}{(1+x+y)^2\sqrt{x+y}}$ over the triangle $0 \le x \le 1$, $0 \le y \le 1 - x$. Note that, the integrand is infinite at (0,0).

```
% In Cartesian coordinates:
>> fun = @(x,y) 1./( sqrt(x + y) .* (1 + x + y).^2 )
>> ymax = @(x) 1 - x
>> Q = integral2(fun,0,1,0,ymax)

% In polar coordinates
>> polarfun = @(theta,r) fun(r.*cos(theta),r.*sin(theta)).*r
>> rmax = @(theta) 1./(sin(theta) + cos(theta))
>> Q = integral2(polarfun,0,pi/2,0,rmax)
```