Numerical Integration and Ordinary Differential Equations.

S.Ithaya Ezhil Manna

P.G. and Research Department of Mathematics,
St.Joseph's College (Autonomous), Tiruchirappalli - 620 002, India.

15.03.2018

Numerical Integration(Quadrature)

Numerical evaluation of the integral $\int f(x)dx$ is called quadrature. Most often, integrand f (x) is quite complicated and it may not be possible to carry out the integration analytically.
In such cases,we resort to numerical integration.
However we can only evaluate definite integrals,i.e.,$\int_a^b f(x)\,dx$.
Matlab provides the following built in functions for numerical integration.
1. *Quad* integrates a specified function over specified limits, based on adaptive *Simpson's rule*.

Definition

In numerical analysis, *Simpson's rule* is a method for numerical integration, the numerical approximation of definite integrals. Specifically, it is the following approximation: $\int_a^b f(x)\,dx \approx \frac{b-a}{6}\left(f(a)+4f\left(\frac{a+b}{2}\right)+f(b)\right)$

2. *quadl* (the last letter is an ell, as in QUADL, not 1) integrates a specified function over specified limits, based on adaptive *Lobatto quadrature*. This one is more accurate than quad but it also uses more function evaluations.

Definition

A quadrature formula of highest algebraic degree of accuracy for the interval [a,b]=[-1,1] and weight p(x)=1 with two fixed nodes: the end-points of [-1,1].The *Lobatto quadrature* formula has the form
$\int_a^b f(x)\,dx = A[f(-1)+f(1)] + \sum_{j=1}^n C_j f(x_j)$.

Numerical Integration Cont..

The general call syntax for both quad and quad1 is as follows:

$$Integral = quad('your - function', a, b, tol, trace, p1, p2,)$$

To use*quad1* , you just replace *quad* in the syntax. As shown in the syntax, both functions require you to supply the integrand as a user-written function.
The optional input argument *tol* specifies absolute tolerance (the default value is 10^{-6}).
A nonzero value of the other optional argument,*trace*, shows some intermediate calculations at each step. The optional arguments*p1, p2 etc.,* are simply passed on to the user-defined function as input arguments in addition to x.

$\boxed{\text{Example:1}}$ Compute $\int_{1/2}^{3/2} e^{-x^2}\,dx$.

The simplest syntax:
function y = erfcousine(x);

y = e^{-x^2};
y = quad ('erfcousine',1/2,3/2)
y =
0.3949.

Double integration:

To evaluate integral of the form

$$\int_{xmin}^{xmax} \int_{ymin}^{ymax} f(x, y) dydx.$$

MATLAB provides a function *dblquad*. The calling syntax for dblquad is

$$I = dblquad(fxy - fun, xmin, xmax, ymin, ymax, tol, @method).$$

Where *tol* and *method* are optional input arguments. The optional argument tol Specifies tolerance (default is 10-6), as previously discussed for 1-D integration, and method specifies a choice that the user makes about the method of integration to be used, e.g., quad or quad1. The default method is quad. The user-defined integrand function, $fxy - fun$, must be written such that it can accept a vector x and a scalar y while evaluating the integrand.

$\boxed{\text{Example:2}}$

Compute $\int_{-1}^{1} \int_{0}^{2} (1 - 6x^2 y) dydx$.

The simplest syntax:

function y = erf(x);

y = $1 - 6x^2 y$;

y = dblquad ('erf',-1,1,0,2)

y =

4.0000.

Ordinary Differential Equations

There is a separate suite of ordinary differential equation solvers in MATLAB. Long ago, MATLAB used to have just two built-in functions for solution of ODEs-ode23 and ode45.
The syntax of use of ode23 is

$$[time, solution] = ode23(your - function, tspan, x_0)$$

Example:3

Solve the first-order linear differential equation $\frac{dx}{dt} = x + t$ with the initial condition $x(0) = 0$.

The simplest syntax:

```
function xdot = simpode(t, x);
          xdot = x + t;
          tspan = [0 2];
          x0 = 0;
          [t,x] = ode23 ('simpode', tspan, x0)
          plot (t, x);
          xlabel( ' t ' );
          ylabel( ' x ' );
```