The Romberg Integration Algorithm

by applying Richardson extrapolation repeatedly on the trapezoidal rule. The estimates generate a triangular array.

Richardson Extrapolation: it's a method used to improve the results of numerical integration by computing, from two estimates of an integral, a third one that is more accurate approximation.

Using

$$h_n = \frac{1}{2^n}(b-a)$$

the method can be inductively defined by

$$R(0,0) = h_1(f(a)+f(b))$$

$$R(n,0) = \tfrac{1}{2}R(n-1,0) + h_n \sum_{k=1}^{2^{n-1}} f(a+(2k-1)h_n)$$

$$R(n,m) = R(n,m-1) + \tfrac{1}{4^m-1}(R(n,m-1) - R(n-1,m-1))$$

or

$$R(n,m) = \tfrac{1}{4^m-1}(4^m R(n,m-1) - R(n-1,m-1))$$

<u>MATLAB CODE:</u>

```
>> [I, nit] = romberg(fn,0,pi,10^-6,10)
I = -4.9348
nit =  5
```

```
function[I, nit]=romberg(func,a,b,etol,maxit)
% Romberg integrates function 'func' of
% one variable within a given interval
% INPUTS:
%     func = the function whose integration is desired
%     a, b = the interval of integration
%     etol = error tolerance
%     maxit = maximum number of ierations
% OUTPUTS:
%     I = the integral of the function within the interval
%     nit = actual number of iterations

  Iprev = 0;
  R(1) = ((b-a)/2)*(func(a)+func(b));
  nit = 1;
  while(nit < maxit)
      R(nit+1) = trapz(func,a,b,nit+1,R(nit));
      for j=nit:-1:1
          p = 4^(nit-j+1);
          R(j) = (p*R(j+1)-R(j))/(p-1);
      end
      if abs(R(1)-Iprev) <= etol*abs(Iprev); break; end
      Iprev = R(1);
      nit = nit+1;
  end
  I = R(1);
end
```

```
function[I]=trapz(func,a,b,n,Ip)
  I = 0;
  h = (b-a)/(2^(n));
  for i=1:2^(n-1)
    I = I+func(a+(2*i-1)*h);
  end
  I = 0.5*Ip+h*I;
end
```