Il comando quad

```
>> Q = quad(FUN,A,B)
```

"Tries to approximate the integral of scalar-valued function FUN from A to B to within an error of 1.e-6 using recursive adaptive Simpson quadrature. FUN is a function handle. The function Y=FUN(X) should accept a vector argument X and return a vector result Y, the integrand evaluated at each element of X."

Function handle

"A function handle is a MATLAB value that provides a means of calling a function indirectly. You can pass function handles in calls to other functions."

```
>> fun = @functionname
```

"Returns a handle to the specified MATLAB function."

```
>> fun = @(arglist)anonymous_function
```

"Constructs an anonymous function and returns a handle to that function. The body of the function, to the right of the parentheses, is a single MATLAB statement or command. arglist is a comma-separated list of input arguments."

Simpson

```
function sol=simpson(fun,a,b,N)
H=(b-a)/N;
x=[a:H/2:b];
fx=feval(fun,x);
sol=H/6*(fx(1)+4*sum(fx(2:2:end-1))+2*sum(fx(3:2:end-2))+fx(end));
```

Scrivere una funzione di Matlab per approssimare

$$\int_a^b f(x)\,dx$$

con errore stimato minore di `toll` usando la formula di Simpson.

Stima a posteriori del errore di quadratura

La formula di Simpson è una formula di quadratura accurata di ordine 4 rispetto a H.

$$I(f) - I_H^S(f) = -\frac{b-a}{180}\left(\frac{H}{2}\right)^4 f^{(iv)}(\xi)$$

$$I(f) - I_{H/2}^S(f) \approx \frac{1}{2^4}\left[I(f) - I_H^S(f)\right]$$

$$\begin{aligned} I(f) - I_{H/2}^S(f) &= I(f) - I_H^S(f) + I_H^S(f) - I_{H/2}^S(f) \\ &\approx 2^4\left[I(f) - I_{H/2}^S(f)\right] + I_H^S(f) - I_{H/2}^S(f) \end{aligned}$$

$$(2^4 - 1)\left[I(f) - I_{H/2}^S(f)\right] \approx I_{H/2}^S(f) - I_H^S(f)$$

$$I(f) - I_{H/2}^S(f) \approx \frac{1}{2^4 - 1}\left[I_{H/2}^S(f) - I_H^S(f)\right]$$

Simpson (dando la tolleranza)

```
function [sol,N,flag]=simpsontoll(fun,a,b,toll)
N=1;
solold=simpson(fun,a,b,N);
flag=1;
for k=1:20
    N=2*N;
    sol=simpson(fun,a,b,N);
    err=abs(sol-solold)/15;
    if err < toll, flag=0; return, end
    solold=sol;
end
```

La formula di Simpson adattiva

```
function [sol,v,flag]=simpsonadatt(fun,a,b,toll)
hmin=(b-a)/1000;
sol=0;
c=a;
d=b;
v=[a];
flag=0;
while c~=b
   if d-c < hmin
      flag=1;
      return
   end
   [solp,err]=estimaerr(fun,c,d);
   if err<toll/2*(d-c)/(b-a)
      sol=sol+solp;
      v=[v d];
      c=d;
      d=b;
   else
      d=(c+d)/2;
   end
end
```

La formula di Simpson adattiva

```
function [I2,err]=estimaerr(fun,a,b)
h=(b-a)/4;
x=[a:h:b];
fx=feval(fun,x);
I1=(b-a)/6*(fx(1)+4*fx(3)+fx(5));
I2=(b-a)/12*(fx(1)+4*fx(2)+2*fx(3)+4*fx(4)+fx(5));
err=abs(I2-I1)/15;
```