COMPUTER ASSIGNEMENT NUMBER 3
Due Friday, November 17, 2009

Computer Assignments are to be submitted electronically no later than 9:00 AM. Submit the routines and scriptfiles in a single zip file. Name the file using your team number followed by an underscore and the number of the computer assignment.

Learning Objectives

1. Be able to program compound Newton-Cotes quadrature rules to compute integrals.
2. Be able to study the performance of these rules on different examples.

Assignment

The object of this lab is to implement the recursive trapezoidal rule for evaluating integrals. As we have already studied in Lecture 9, the algorithm for this recursive scheme is given by:

Algorithm: Recursive trapezoidal rule to compute $\int_a^b f(x)dx$ with an absolute error less than ϵ.

Input: $\epsilon > 0$, $f(x)$, b, a, $n \in \mathbb{N}$.
Output: $I \approx \int_a^b f(x)\,dx$.

- $h = (b-a)/n$; $I = \frac{h}{2}(f(a)+f(b))$;
- IF $n > 1$ THEN $I = I + h\sum_{j=1}^{n-1} f(a+jh)$;
- $\Delta = 1 + \epsilon$;
- DO WHILE $\Delta > \epsilon$

  $h = h/2; I_0 = I$;

  $I = I_0/2 + h\sum_{j=1}^{n} f(a+(2j-1)h)$;

  $n = 2n$;

  $\Delta = |I - I_0|$;

We will implement this algorithm in Matlab introducing a couple of changes:

1. First, we will consider the relative error instead of the absolute error as stopping criterium. This means that will take Δ as $\Delta = |1 - I/I_0|$. Note that some precautions have to be taken with this change (for instance, it is not possible to compute the relative error when $I_0 = 0$)

2. Second, we will consider both a minimum (4) and a maximum number (16) of iterations for the trapezoidal rule.

The first two lines of the Matlab function implementing the recursive trapezoidal will be:

```
function [ti,n]=trap(func,eps,a,b)
funci=inline(func);
```

We will use the command **feval** to evaluate the function funci at different points.
The meaning of the input variables of the **trap** function will be:

1. func: alphanumeric chain describing the function to be integrated; for example, 'x^2', if we want to compute $\int_a^b x^2 dx$.
2. eps: tolerance to the relative error (we will take 10^{-8})
3. a, b: extreme points of the integration interval.

On the other hand, the output variables will be:

1. ti: numerical value of the integral.
2. n: number of iterations used for the trapezoidal rule.

As a firts test of the algorithm, we could consider a linear function, as we already know that the trapezoidal rule is exact for this kind of functions. For example:

```
>> [it,n]=trap('10*x-4',1.e-8,0,1)
```

the outputs of the function have to be it=1, n=4.
We will study the performance of our algorithm on the following examples:

1. $\int_{-1}^{1} x^2 dx$
2. $\int_{-\pi/2}^{\pi/2} cos(x) dx$
3. $\int_{-\infty}^{+\infty} cos(mx) exp(-x^2) dx$, para $m = 0, 4, 8, 9, 10$

Note that the last example corresponds to an improper integral. In this case, we have to consider a finite approximation for the infinite interval. We could take, for instance:

$$\int_{-\infty}^{+\infty} f(x)dx \approx \int_{-10}^{10} f(x)dx$$

It will also interesting to analyze the sensitivity of the numerical results to the approximation of the infinite interval.
Finally, remember that the analytical value of the improper integrals is given by:

$$\int_{-\infty}^{+\infty} cos(mx) e^{-x^2} dx = \sqrt{\pi} e^{-m^2/4}$$