Computer Practise 5

Computer Practise with MATLAB: functions of one and more variables, integrals and linear system's application

5.1 Functions of one and more variables.

When we are working with functions, sometimes it is interesting to find the roots of a one variable function (“fzero”). The algorithm used in this command was ideated by T. Dekker and it uses a combination of the bisection method, the secant method and the inverse quadratic interpolation method.
Its syntax is

```
fzero(fun,x0,'optiones').
```

Some examples are:

```
x=0:0.01:23;
plot(x, cos(x))
fzero(@cos,[1 23])
%
x=-2:0.01:2;
plot(x,sin(x.*x))
fzero(@(x)sin(x*x),2);
fzero(@(x)sin(x*x),0)
```

We can use the “fzero” command also with user defined functions

```
x = fzero(@myfun,x0);
```

where “myfun” is a M-file that defines a function as

```
function f = myfun(x)
f = ...            % Compute function value at x
```

For instance, we want to look for the roots of de $f(x) = x^3 - 2\,x - 5$ close to $x_0 = 2$. We can do with an anonymous function

```
f = @(x)x.^3-2*x-5;
%
z = fzero(f,2)
```

or defining the function

```
function f = myfun(x)
f = x.^3-2*x-5
%
x=0:0.01:2;
plot(x,myfun(x))
x = fzero(@(x) myfun(x),2)
```

Since the function of this example is a polynomial one, it is convenient to use the ‘‘root’’ command, that allows to compute all its roots (the command arguments are the coefficients of the different powers of x).

```
roots([1 0 -2 -5])
```

Also, when working with functions, it could be interesting to find the minimum value of the function $f(x)$ in a fixed interval (‘‘fminbnd’’). The same command can be used to find the maximum value, that will be the minimum of the opposite of the function ($-f(x)$ reflects the function about a horizontal axis).
Its syntax is

```
x = fminbnd(fun,x1,x2,options)
```

where x_1 and x_2 represent the search interval. The options that appear are optimization options and it is convenient to look al the help manual for its use.
Some examples are:

```
x=-3:0.01:3;
plot(x,sin(x.*x))
fminbnd(@(x)sin(x*x),-1,1)
fminbnd(@(x)(-sin(x*x)),-2,2)
fminbnd(@(x)(-sin(x*x)),0,2)
```

As before, we can use it with anonymous function

```
f = @(x)x.^3-2*x-5;
%
x = fminbnd(f, 0, 2)
```

or with user defined functions

```
function f = myfun(x)
f = x^3-2*x-5
%
x = fminbnd(@(x) myfun(x),0,2)
```

In the case of functions of several variables, the “fminsearch” command is used.
Its syntax is

```
x = fminsearch(fun,x0,options)
```

where x_0 represents an initial approximation used to start the search.
With the same example:

```
x=-3:0.01:3;
plot(x,sin(x.*x))
fminsearch(@(x) sin(x*x),1)
fminsearch(@(x) sin(x*x),2)
%
x=-3:0.01:3;
plot(x,-sin(x.*x))
fminsearch(@(x) -sin(x*x),1)
fminsearch(@(x) -sin(x*x),2)
fminsearch(@(x) -sin(x*x),3)
```

Another one:

```
x=-4:.1:4; y=-4:.1:4;
[xx,yy]=meshgrid(x,y);
zz=xx.^2+yy.^2;
figure
surf(xx,yy,zz)
xlabel('x axis'); ylabel('y axis'); zlabel('z axis');
title('f(x,y) = x^2 + y^2');
%
tazon = @(x) x(1)^2+x(2)^2;
%
[x,fval] = fminsearch(tazon,[1, 1])
```

“fminsearch” can be used also for user defined functions.

```
x = fminsearch(@(x) myfun(x),2)
```

A classic example for the multidimensional minimization is the Rosenbrock banana function.

$$f(x_1, x_2) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2.$$

The minimum stays at $(1, 1)$ with zero value. Easily we can plot the 3-D graphic.

```
x=-2:0.05:2; y=-2:0.05:2;
[xx,yy]=meshgrid(x,y);
zz=100*(yy-xx.^2).^2 + (1-xx).^2;
figure
surf(xx,yy,zz)
xlabel('x axis'); ylabel('y axis'); zlabel('z axis');
```

and solve the minimization problem with an anonymous function:

```
banana = @(x)100*(x(2)-x(1)^2)^2+(1-x(1))^2;
%
[x,fval] = fminsearch(banana,[-1.2, 1])
```

or with a user defined function

```
function f = banana(x)
f = 100*(x(2)-x(1).^2).^2 + (1-x(1)).^2;
%
x = fminsearch(@(x) banana(x),[-1.2,1])
```

In the Web page
https://es.mathworks.com/help/optim/examples/banana-function-minimization.html
you can see the solved example from a numerical point of view, using different search algorithms. You can try the example in the browser or in a MATLAB command window typing

```
openExample('optim/BananaFunctionMinimizationExample')
```

5.2 Integral of functions of one and more variables.

The most used command in the case of one variable function

```
q = integral(fun,xmin,xmax)
```

while it is not recommended to use the command

```
q = quad(fun,a,b)
```

The *fun* variable indicates the anonymous function or the user defined function and *xmin*, *xmax* (or a y b) represent the boundary points of the integration interval.
For instance, in the case of the integral

$$\int_0^2 \frac{1}{x^3 - 2\,x - 5}\,dx$$

two MATLAB instructions are enough for finding its value:

```
F = @(x)1./(x.^3-2*x-5);
Q = integral(F,0,2)
Q = quad(F,0,2)
```

In the case of functions of two variables, in order to compute double integrals the used commands are:

```
q = integral2(fun,xmin,xmax,ymin,ymax)
q = quad2d(fun,a,b,c,d)
```

For instance, in the case of the integral

$$\int_0^1 \int_0^{1-x} \left[\sqrt{x+y}\,(1+x+y)^2\right]^{-1}\, dy\, dx = \frac{\pi}{4} - \frac{1}{2}$$

let us first plot the graphics of the surface and then we check the integral value.
To do this last step, we notice that the inner integral in dy depends on the x variable, therefore not only we have to define the function that appears in the integrand but also we have to define the upper bound of the integral $ymax$.

```
x=pi:0.05:2*pi; y=0:0.1:pi;
[X,Y]=meshgrid(x,y);
funci5 =  1./(sqrt(X + Y) .* (1 + X + Y).^2 );
surf(x,y,funci5)
xlabel('X'); ylabel('Y');
rotate3d
%
fun = @(x,y) 1./(sqrt(x + y) .* (1 + x + y).^2 );
ymax = @(x) 1 - x;
Q = quad2d(fun,0,1,0,ymax)
Q = integral2(fun,0,1,0,ymax)
pi/4 - 0.5
```

In the case of functions of three variables, in order to compute triple integrals the used command is:

```
q = integral3(fun,xmin,xmax,ymin,ymax,zmin,zmax)
```

While the bounds *ymin*, *ymax*, *zmin* y *zmax* may be functions, *xmin* y *xmax* have to be real numbers.
For instance, the triple integral

$$\int_0^{\pi} \int_0^1 \int_{-1}^1 (y\, \sin(x) + z\, \cos(x))\; dz\, dy\, dx$$

two MATLAB instructions are enough for finding its value:

```
fun = @(x,y,z) y.*sin(x)+z.*cos(x);
q = integral3(fun,0,pi,0,1,-1,1)
```

5.3 Exercises.

1. The exercises can be done individually or in groups of at most two people. Exercises ''shared'' or copied from your companions or form previous years will be graded with 0 points.

2. Each group has to email to **virginia.muto@ehu.eus** a zip file with the name *group.zip* that has to contain the script files that solve the questions stated at the end of the page from 5 (a) to 5 (d) and from 6 (a) to 6 (c). The files have to contain the explication of the used commands. Each group has one week to solve the exercises and late deliver will be penalized with half the mark.

3. The name(s) and surname(s) of the author(s) have to appear also in the first line of every Matlab file. Unnamed files will not be graded.

4. Run always the codes before submitting them.

5. Besides the functions used in the previous classes, you will have to use MATLAB commands seen in these notes in order to compute roots, *fzero* and *roots*, to compute the maximum or minimum of a function of one or more variables, *fminbnd* or *fminsearch*, and to compute multiple integrals, *integral*, *integral2* or *integral3*.

   (a) Find the three roots (''fzero'' command), the relative minimum and maximum values (''fminbnd'' and ''fminsearch'' commands) for the function

   $$f(x) = x^3 - 6x^2 - 4x + 25.$$

   Before doing anything, plot a graphics of the function to discover the correct intervals.

   (b) Build a MATLAB program/script that generates the surface for

   $$z = x^3 + y^3 + 3\,e^{xy}$$

   and its tangent plane in the point $x = 1, y = 0$. Moreover, label the axis and define a title for the graphics.

   Note: you have to obtain the equation for the tangent plane by yourself (paper and pencil).

   (c) Look for relative minimum and maximum values for the function in $D = [-1, 1] \times [-1, 1]$

   $$f(x, y) = x\,y\,(1 - x^2 - y^2).$$

   Before doing anything, generate a graphics of the function.

(d) Write a MATLAB program/script that allows to compute the following integrals and, when it is possible, to plot the graphics of the functions that appear in the integrand.

$$\int_0^2 \frac{x^2-9}{x^3-2\,x-5}\,dx, \qquad \int_0^{20} \log(x+1)\sin(x)\,dx,$$

$$\int_0^2\int_0^2 \left(x^3+y^2\right)\,dy\,dx, \qquad \int_{-1}^1\int_{-x^2}^{x^2} \left(x^2+y^2\right)\,dy\,dx,$$

$$\int_1^2\int_{-1}^1\int_0^1 (2\,x+3\,y+z)\,dz\,dy\,dx,$$

$$\int_0^1\int_0^{1-x}\int_0^3 x^2\,y\,dz\,dy\,dx.$$

6. Remember that in the first computer practise we saw an example that solved the linear system "$Ax = b$" and computed the euclidean norm

```
A=[-3 0 1; 2 5 -7;-1 4 8];
b = [4;5;6];
x = A\b
x = mldivide(A,b)
norm(A*x - b)      % euclidean norm
```

In this problem we want to see one applications for linear systems: a method used to solve an Ordinary Differential Equation (ODE).

For this problem you should use the *syms* command for the Symbolic Math Toolbox. Useful commands are *simplify* and *eval*.

(a) Consider the Boundary Value Problem in the interval $[0,1]$ given by the ODE:

$$(1+x^2)\,u''(x)+2\,x\,u'(x)=2+6\,x^2+2\,x\,\cos(x)-(1+x^2)\,\sin(x), \tag{5.1}$$

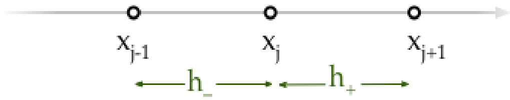
with the boundary values

$$u(0)=1, \qquad u(1)=2+\sin 1. \tag{5.2}$$

Check analytically that the exact solution is

$$u_e(x)=x^2+\sin x+1.$$

(b) In order to obtain an approximate solution of the above written problem, we can consider a step $h=\frac{1}{n+1}$ and the grid of equidistant points $x_j = j\,h; j=1,\ldots,n$, with $x_0=0$ and $x_{n+1}=1$



We approximate the derivatives in such points using

$$u''(x_j) = \frac{u_{j+1} - 2u_j + u_{j-1}}{h^2}, \qquad u'(x_j) = \frac{u_{j+1} - u_{j-1}}{2\,h},$$

where $u_j = u(x_j)$ is the j-th component of the unknown vector $\mathbf{u_h}$.
Write for $j = 1, \ldots, n$ the approximate equations for (5.1) with the boundary values (5.2) and define the matrix $\mathbf{A_h}$ and the vector $\mathbf{b_h}$ such that the set of these equations can be considered a system of linear equations

$$\mathbf{A_h}\,\mathbf{u_h} = \mathbf{b_h}.$$

(c) Write a MATLAB program/script that allows to solve the obtained linear system for $n = 10, 20, 40, 80, 160$, and compute the euclidean norm for the errors

$$||\mathbf{u_h} - \mathbf{u_e}||_2 = \left(\sum_{j=1}^{n} (u_j - u_e(x_j))^2 \right)^{1/2}.$$

Plot in the same graphic the discrete approximated solutions and the exact continuous one.