M3.1 Calcule um zero da função

$$f(x) = e^x - x^2 - 2x - 2,$$

usando $x^{(1)} = 2$.

function[f] = teste(x)

f = exp(x) – x^2 – 2*x – 2

_____----_____

[x,f,exitflag,output] = fsolve(teste,2)

$$\begin{cases} (x_1^4 + 0.06823x_1) - (x_2^4 + 0.05848x_2) - 0.01509 = 0 \\ (x_1^4 + 0.05848x_1) - (2x_2^4 + 0.11696x_2) = 0 \end{cases}$$

Resolva o sistema utilizando para aproximação inicial $(0.30, 0.30)$, fornecendo a informação acerca das derivadas

Exemplo: fsolve

```
function [F,d] = teste(x)
F(1) = [(x(1)^4+0.06823*x(1))-(x(2)^4+0.05848*x(2))-0.01509];
F(2) = [(x(1)^4+0.05848*x(1))-(2*x(2)^4+0.11696*x(2))];
% fornecendo as primeiras derivadas
if nargout>1
    d = [4*x(1)^3+0.06823 -4*x(2)^3-0.05848; 4*x(1)^3+0.05848 -8*x(2)^3-0.11696];
end
>> x0=[0.30 0.30]
>> options=optimset('Jacobian','on')
%para que a rotina use as primeiras derivadas fornecidas na m-file
>> [xsol,fsol,exitflag,output]=fsolve('teste',x0,options)
xsol =
    0.2928    0.1879
fsol =
  1.0e-005 *
   -0.1429   -0.2858
exitflag =
     1
output =
       iterations: 3
       ...
```