

Cap. 2 - Resolução Numérica de Equações não Lineares

Implementação computacional do método da bisseção usando o Matlab

```
function [zero,res,niter] = Met_Bissecao(f,a,b,tol,nmax)
% Met_Bissecao tenta encontrar um zero da função f no intervalo [a,b]
%          usando o método da bisseção
% f : função que descreve a equação f(x)=0
% a,b : extremos do intervalo inicial
% nmax : número máximo de iterações (critério de paragem)
% tol : tolerância para o erro (critério de paragem)
% zero,res,niter : valor aproximado para o zero, f(zero), número de
%          iterações no qual a aproximação para zero foi calculada
if f(a) == 0
    zero=a; res=0; niter=0;
    fprintf('Foi encontrada a raiz. ');
    return;
elseif f(b) == 0
    zero=b; res=0; niter=0;
    fprintf('Foi encontrada a raiz. ');
    return;
elseif f(a) * f(b) > 0
    error('f(a) e f(b) têm o mesmo sinal');
end;
niter=1; d=(b-a)/2;
aux=[a, (a+b)/2, b];
while (d >= tol) && (niter < nmax)
    if f(aux(2)) == 0
        d=0;
        fprintf('Foi encontrada a raiz. ');
    else
        if f(aux(1))*f(aux(2))<0
            aux(3) = aux(2);
        else
            aux(1) = aux(2);
        end;
        aux(2)=(aux(1)+aux(3))/2;d=d/2;
    end;
    niter=niter + 1;
end;
if niter == nmax
    fprintf('Foi atingido o número máximo de iterações. ');
end;
zero=aux(2); res=f(zero);
end
```