Zadania 8

Krzysztof Kozubek (Matlab)

Metoda Netwon's:

```
function apprx = newt(func,gss,ite,tol)
f = inline(func); fp =
inline(diff(sym(func))); iter = ite;
for n=1:iter;     oldgss = gss;
newgss = gss - (f(gss)/fp(gss));
gss = newgss; end
if (abs(newgss-oldgss)) < tol
apprx = gss;     disp(iter); else
iter = ite+1;     apprx =
newt(func,gss,iter,tol); end end
```

newt('(x^2-1)*sinh(x)^3', -1.1 , 100 , 1e-6) wynik: -1 newt('(x^2-1)*sinh(x)^3', 0 , 100 , 1e-6) wynik: 0 newt('(x^2-1)*sinh(x)^3', 2 , 100 , 1e-6) wynik: 1

Metoda Secant:

```
function root = secant(f,xo,x1,epsilon)
f = inline(f); if (nargin < 3)
error('Less input arguments'); elseif
(nargin == 3)      epsilon = 1e-6;
Iter = 0;      fprintf('xo = %d',xo);
fprintf('\n');      fprintf('f(xo) =
%f',f(xo));      fprintf('\n\n');
    disp('Iteration           x1            x2             f(x1)');
disp('=========        ========     ========     ============');
    while(abs(f(x1)) >= epsilon)
Iter = Iter + 1;
if(f(x1)-f(xo) == 0)
            fprintf('f(x1)-f(xo) = 0 on iteration number %d',Iter);
fprintf('\n');              disp('We can not find the root');
return;          end
        x2 = x1 - (f(x1)*((x1-xo)/(f(x1)-f(xo))));
fprintf('%3d',Iter);         fprintf('%22.4f',x1);
fprintf('%13.4f',x2);
fprintf('%16.4f',f(x1));         fprintf('\n');
x1 = x2;      end      root = x2;
elseif(nargin == 4)      Iter =
0;      fprintf('xo = %d',xo);
fprintf('\n');
fprintf('f(xo) = %f',f(xo));
fprintf('\n\n');
    disp('Iteration           x1            x2             f(x1)');
disp('=========        ========     ========     ============');
    while(abs(f(x1)) >= epsilon)
Iter = Iter + 1;
if(f(x1)-f(xo) == 0)
            fprintf('f(x1)-f(xo) = 0 on iteration number %d',Iter);
fprintf('\n');              disp('We can not find the root');
return;          end
        x2 = x1 - (f(x1)*((x1-xo)/(f(x1)-f(xo))));
fprintf('%3d',Iter);         fprintf('%22.4f',x1);
```

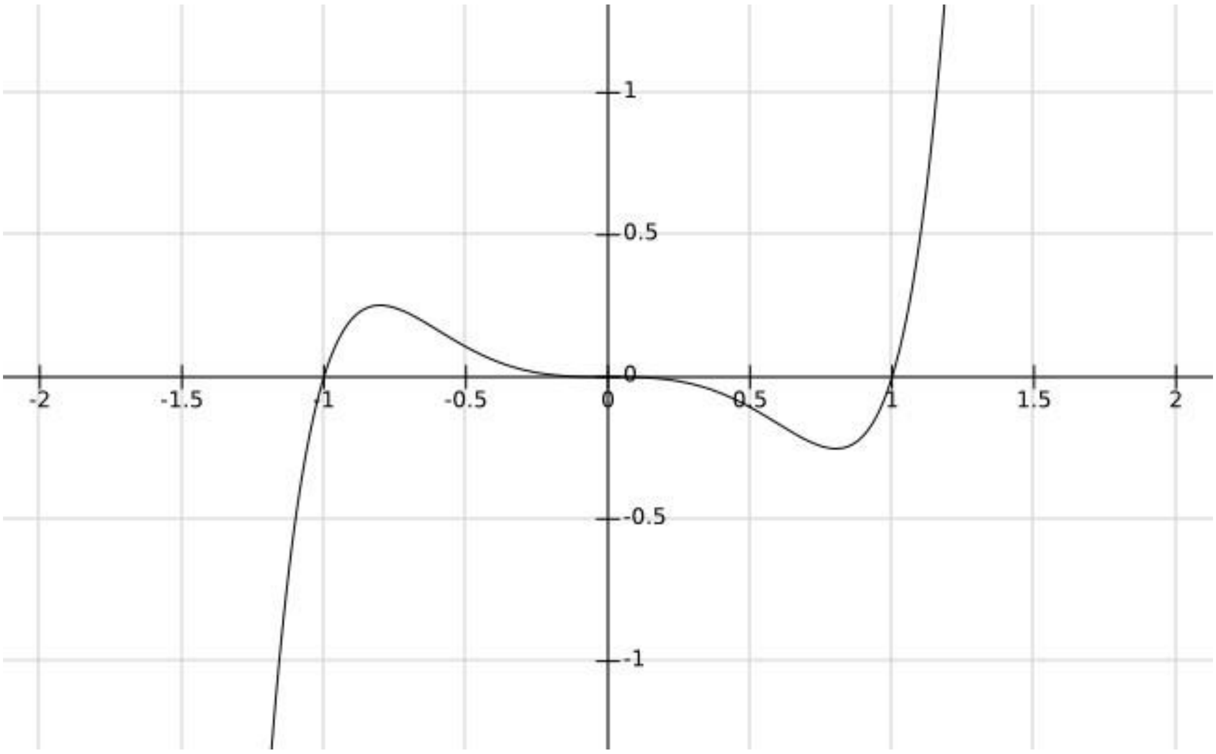
```
fprintf('%13.4f',x2);
fprintf('%16.4f',f(x1));         fprintf('\n');
x1 = x2;      end      root = x2;
end
```

secant('(x^2-1)*sinh(x)^3', -1.1 , 100 , 1e-6) wynik = -1 secant('(x^2-1)*sinh(x)^3', 0, 100 , 1e-6) wynik = 0 secant('(x^2-1)*sinh(x)^3', 1, 100 , 1e-6) wynik = 1

Metoda Bisect

```
function [ xs ] = bisect( funch, a, b, tol )
funch = inline(funch); x1=a;
f1=feval(funch,x1); x2=b;
f2=feval(funch,x2); if f1*f2>=0
    error('no root in interval');
end tic; for n=1:1/eps     if
f1*f2<0
        x3=0.5*(x1+x2); fxns(n)=feval(funch,x3);
if f1*fxns(n)<0             x2=x3;
            x3=0.5*(x1+x2); fxns(n)=feval(funch,x3);
end         if f2*fxns(n)<0             x1=x3;
x3=0.5*(x1+x2);         end     end     xns(n)=x3;
err(n)=abs((x2-x1)/x2);
    if (err(n) < tol), break;  end %stop iterating if error is less
than tolerance end t=toc;
fprintf('iteration\t|\t\txns\t\t\t|\t\tf(xns)\t\t|\t\terror\n');
fprintf('-------------------------------------------------------------
---------------\n'); for
i=2:length(xns)
    fprintf('%5d\t\t|\t%10.5f\t\t|\t%10.5f\t\t|\t%10.5f\n',i-
1,xns(i),fxns(i),err(i));
end
fprintf('-------------------------------------------------------------
---------------\n'); format
long; xs=xns(length(xns));
fprintf('\nfinal solution: \n\tx = %-10.10f\n',xns(length(xns)));
fprintf('time elapsed in milliseconds: \n\tt = %-10.10f\n',t*10^3);
end
```

bisect ('(x^2-1)*sinh(x)^3', -1.1 , 100 , 1e-6) wynik = -1.000000123... bisect ('(x^2-1)*sinh(x)^3', 0, 100 , 1e-6) wynik = 0.000000145... bisect ('(x^2-1)*sinh(x)^3', 1, 100 , 1e-6) wynik = 1.000000123...



Rozwiązaniem, miejscami zerowymi równania (x^2-1) sinh^3 x=0 jest:

$ans_1 = -1$ $ans_2 = 0$ $ans_3 = 1$

Moim zdaniem, najlepszą metodą jest metoda Secant(Siecznych):

- Wykonuje mniej iteracji z taką samą dokładnością
- Jest dokładniejsza od reszty metod