Analiza numeryczna

Wyklad 4. Rozwiązywanie rownan nieliniowych

Pawel Wozny

Wroclaw, 25 października 2023 r.

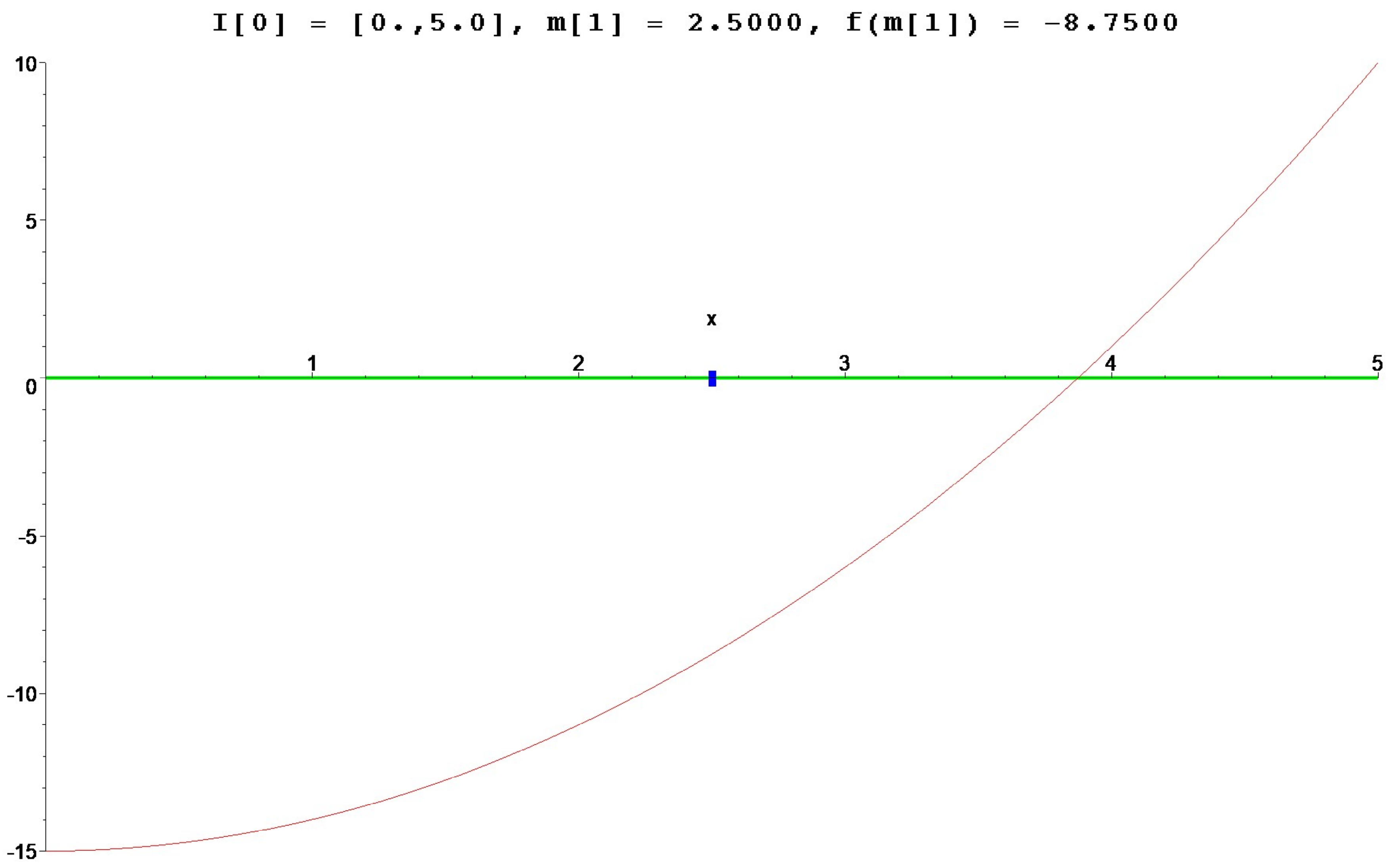
Przyklady

1. Metoda bisekcji

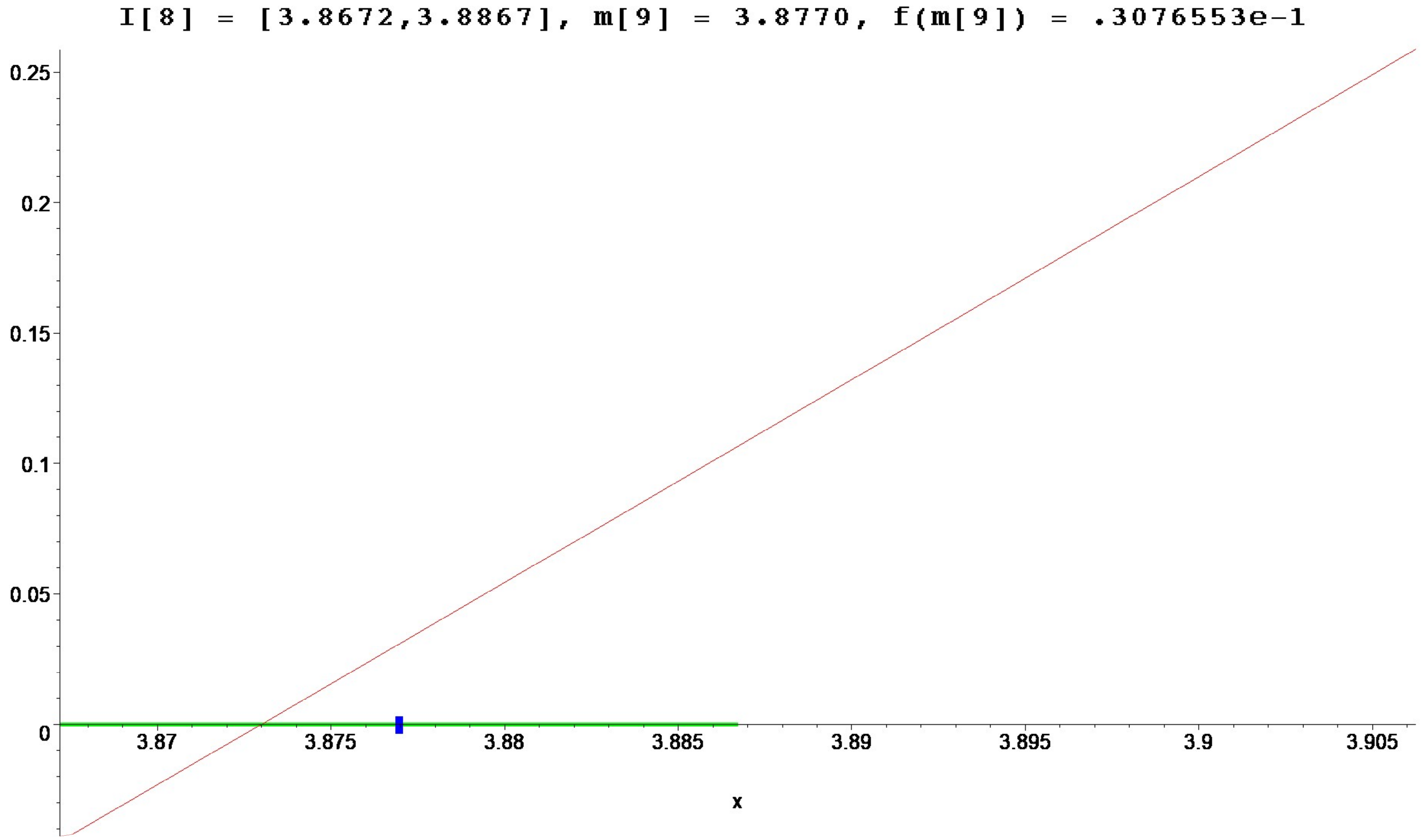
1.1. Prezentacja graficzna

```
>
>
> restart:
>
> N:=7: M:=25:
>
> a[0]:=0.0: b[0]:=5.0: m[1]:=(a[0]+b[0])/2:
>
> f:=x->x^2-15:
>
> plot_w:=plot(f(x),x=a[0]..b[0],scaling=unconstrained):
>
> plot_I:=plot([[a[0],0],[b[0],0]],x=a[0]..b[0],color=green,thickness=5):
> plot_m:=plot([[m[1],-0.25],[m[1],0.25]],x=a[0]..b[0],color=blue,thickness=10):
>
> as:=convert(evalf(a[0],5),string):
> bs:=convert(evalf(b[0],5),string):
> ms:=convert(evalf(m[1],5),string):
> fms:=convert(evalf(f(m[1]),5),string):
>
> opis:="I["||0||"] = ["||as||","||bs||"], "||"m["||1||"] = "||ms||", f(m["||1||"]) = "||fms:
>
> klatka[0]:=plots[display](plot_w,plot_m,plot_I,title=opis,titlefont=[COURIER,BOLD,15]):
>
> for i from 1 to N
   do
       if f(m[i])<0 then
         a[i]:=m[i]:
         b[i]:=b[i-1]
       else
         a[i]:=a[i-1]:
         b[i]:=m[i]
      fi:

     m[i+1]:=(a[i]+b[i])/2:

     plot_I:=plot([[a[i],0],[b[i],0]],x=a[0]..b[0],color=green,thickness=5):
```

```
      plot_m:=plot([[m[i+1],-0.25],[m[i+1],0.25]],x=a[0]..b[0],color=blue,thickness=10):

      as:=convert(evalf(a[i],5),string):
      bs:=convert(evalf(b[i],5),string):
      ms:=convert(evalf(m[i+1],5),string):
      fms:=convert(evalf(f(m[i+1]),10),string):

      opis:="I["||i||"] = ["||as||","||bs||"], "||"m["||(i+1)||"] = "||ms||
                    ", f(m["||(i+1)||"]) = "||fms:

      klatka[i]:=plots[display](plot_w,plot_m,plot_I,title=opis,titlefont=[COURIER,BOLD,15])
    od:

>
> plot_w:=plot(f(x),x=a[N]..b[N],scaling=unconstrained):
>
> for i from N+1 to M
    do
        if f(m[i])<0 then
          a[i]:=m[i]:
          b[i]:=b[i-1]
        else
          a[i]:=a[i-1]:
          b[i]:=m[i]
       fi:

      m[i+1]:=(a[i]+b[i])/2:

      plot_I:=plot([[a[i],0],[b[i],0]],x=a[N]..b[N],color=green,thickness=5):
      plot_m:=plot([[m[i+1],-0.01/3],[m[i+1],0.01/3]],x=a[N]..b[N],color=blue,thickness=10):

      as:=convert(evalf(a[i],5),string):
      bs:=convert(evalf(b[i],5),string):
      ms:=convert(evalf(m[i+1],5),string):
      fms:=convert(evalf(f(m[i+1]),10),string):

      opis:="I["||i||"] = ["||as||","||bs||"], "||"m["||(i+1)||"] = "||ms||
                    ", f(m["||(i+1)||"]) = "||fms:

      klatka[i]:=plots[display](plot_w,plot_m,plot_I,title=opis,titlefont=[COURIER,BOLD,15])
    od:

>
> plots[display]([seq(klatka[i],i=0..N)],insequence=true);
>
>
>
>
```




```
> 
> 
> plots[display]([seq(klatka[i],i=N+1..M)],insequence=true);
> 
> 
> 
```

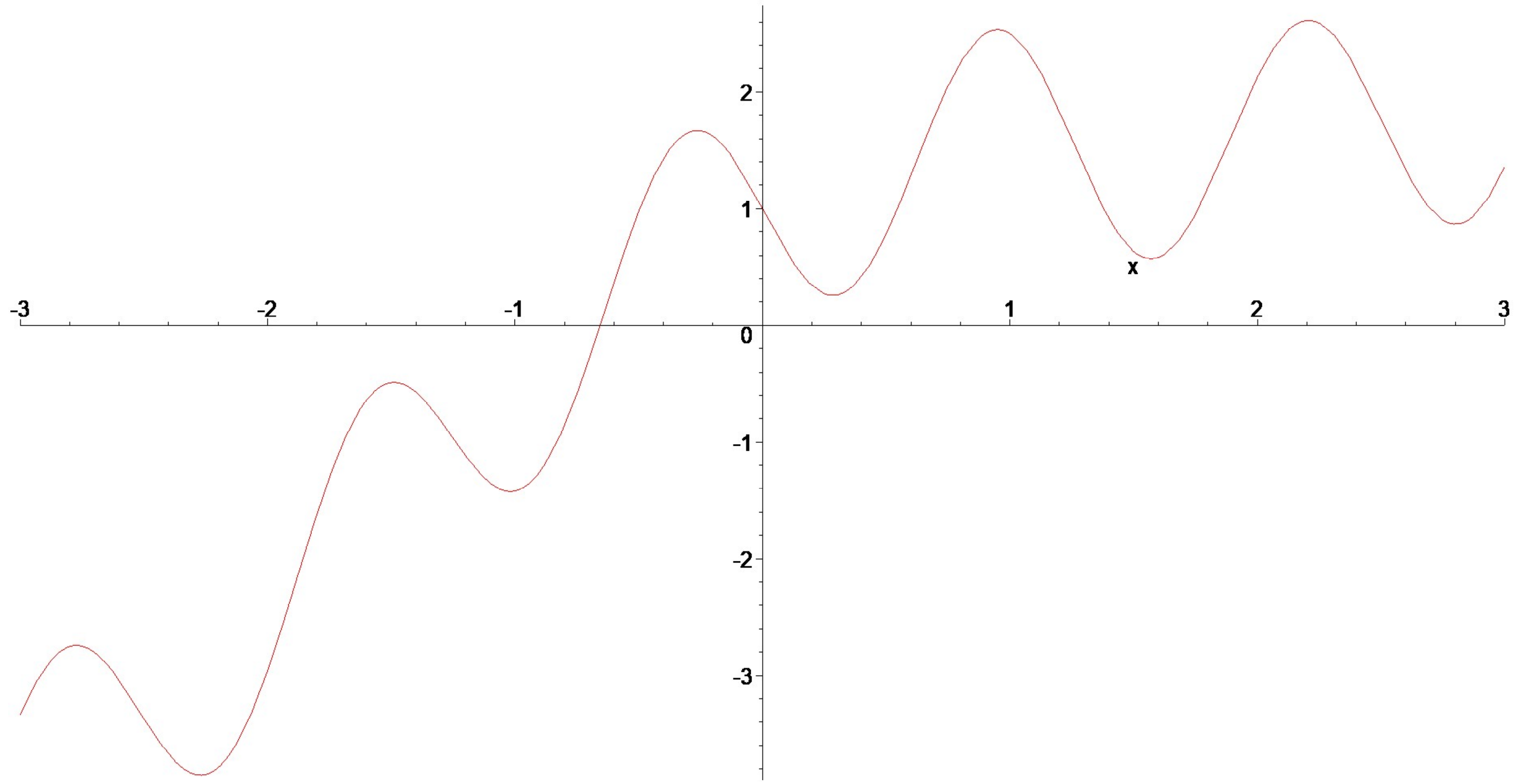


>

1.2. Przyklad

>
>
```
> restart:
>
> f:=x->cos(x)-sin(5*x)+x;
>
> plot(f(x),x=-3..3);
>
```

$$f := x \to \cos(x) - \sin(5\,x) + x$$




```
>
>
> restart;
>
> Digits:=20:
>
> f:=x->cos(x)-sin(5*x)+x:
>
> N:=30:
>
> print(``);print(`========================================`);print(``);
> print(`Metoda bisekcji`);
> print(``);
> f(x)=0;
> print(``);
> print(a[0]=-1,`       `,b[0]=0);
> print(``);print(`========================================`);print(``);
>
> printf("================================================\n");
> printf(" k    a[k]            b[k]            m[k+1]\n");
> printf("================================================\n");
>
> a[0]:=-1.0: b[0]:=0.0:
>
> for k from 0 to N
    do
      m[k+1]:=(a[k]+b[k])/2:

      printf("$%2d$ & $%1.10f$ & $%1.10f$ & $%1.10f$ \\\\\n",k,a[k],b[k],m[k+1]);
      #printf("%2d & %1.10f & %1.10f & %1.10f \n",k,a[k],b[k],m[k+1]);

       if is(f(m[k+1])<0) then
```

```
            a[k+1]:=m[k+1];
            b[k+1]:=b[k]
          else
            a[k+1]:=a[k];
            b[k+1]:=m[k+1]
         fi

       od:
>
>
```

==

Metoda bisekcji

$\cos(x) - \sin(5\,x) + x = 0$

$a_0 = \text{-}1, \quad , b_0 = 0$

==

```
================================================
 k   a[k]           b[k]            m[k+1]
================================================
$ 0$ & $-1.0000000000$ & $0.0000000000$ & $-.5000000000$ \\
$ 1$ & $-1.0000000000$ & $-.5000000000$ & $-.7500000000$ \\
$ 2$ & $-.7500000000$ & $-.5000000000$ & $-.6250000000$ \\
$ 3$ & $-.7500000000$ & $-.6250000000$ & $-.6875000000$ \\
$ 4$ & $-.6875000000$ & $-.6250000000$ & $-.6562500000$ \\
$ 5$ & $-.6562500000$ & $-.6250000000$ & $-.6406250000$ \\
$ 6$ & $-.6562500000$ & $-.6406250000$ & $-.6484375000$ \\
$ 7$ & $-.6562500000$ & $-.6484375000$ & $-.6523437500$ \\
$ 8$ & $-.6562500000$ & $-.6523437500$ & $-.6542968750$ \\
$ 9$ & $-.6562500000$ & $-.6542968750$ & $-.6552734375$ \\
$10$ & $-.6562500000$ & $-.6552734375$ & $-.6557617188$ \\
$11$ & $-.6562500000$ & $-.6557617188$ & $-.6560058594$ \\
$12$ & $-.6560058594$ & $-.6557617188$ & $-.6558837891$ \\
$13$ & $-.6558837891$ & $-.6557617188$ & $-.6558227539$ \\
$14$ & $-.6558227539$ & $-.6557617188$ & $-.6557922363$ \\
$15$ & $-.6557922363$ & $-.6557617188$ & $-.6557769775$ \\
$16$ & $-.6557769775$ & $-.6557617188$ & $-.6557693481$ \\
$17$ & $-.6557693481$ & $-.6557617188$ & $-.6557655334$ \\
$18$ & $-.6557693481$ & $-.6557655334$ & $-.6557674408$ \\
$19$ & $-.6557674408$ & $-.6557655334$ & $-.6557664871$ \\
$20$ & $-.6557674408$ & $-.6557664871$ & $-.6557669640$ \\
$21$ & $-.6557674408$ & $-.6557669640$ & $-.6557672024$ \\
$22$ & $-.6557674408$ & $-.6557672024$ & $-.6557673216$ \\
$23$ & $-.6557673216$ & $-.6557672024$ & $-.6557672620$ \\
$24$ & $-.6557672620$ & $-.6557672024$ & $-.6557672322$ \\
$25$ & $-.6557672620$ & $-.6557672322$ & $-.6557672471$ \\
$26$ & $-.6557672471$ & $-.6557672322$ & $-.6557672396$ \\
$27$ & $-.6557672396$ & $-.6557672322$ & $-.6557672359$ \\
$28$ & $-.6557672396$ & $-.6557672359$ & $-.6557672378$ \\
$29$ & $-.6557672378$ & $-.6557672359$ & $-.6557672368$ \\
$30$ & $-.6557672378$ & $-.6557672368$ & $-.6557672373$ \\
```

```
>
>
> Digits:=30:
>
> wynik:=fsolve(f(x),x):
>
> print(``);print(`==========================================`);print(``);
> f(x)=0;
> print(``);
> print(`Rozwiazanie  `=wynik);
> print(``);print(`==========================================`);print(``);
> print(`Metoda bisekcji`);
> print(``);
```

```
> printf("%46s m[%d] = %1.20f","",N,m[N]);
> print(``);print(`=========================================`);print(``);
> print(`Blad wzgledny`);
> print(``);
> print(`blad `=evalf(abs((m[N]-wynik)/wynik)));
> print(``);print(`=========================================`);print(``);
>
>
>
>
```

===

$\cos(x) - \sin(5\,x) + x = 0$

Rozwiazanie = -0.655767236930589026442857078583

===

Metoda bisekcji

m[30] = -.65576723683625459670

===

Blad wzgledny

blad = $0.143853526724516875981983823099\ 10^{-9}$

===

```
>
```

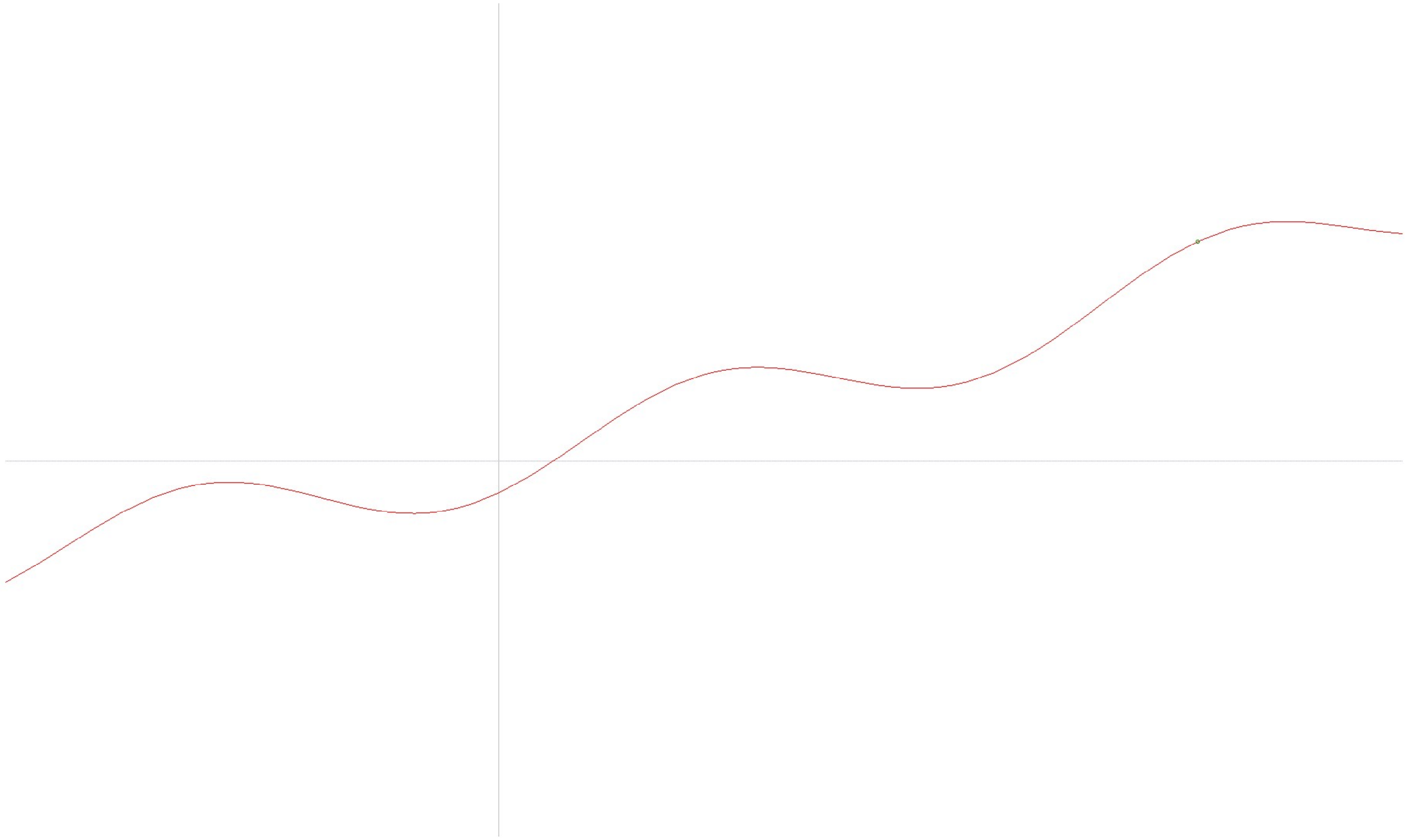
2. Metoda Newtona

2.1. Prezentacja graficzna

```
>
>
> ############################################################################
> ### Metoda Newtona szukania miejsca zerowego funkcji - prezentacja graf. ###
> ###                                                                      ###
> ### Autor: Pawel Keller                                                  ###
> ############################################################################
>
> restart:
>
> Digits:=30:
>
> with(plots):
>
>
> newton:=proc(f,x0,N,xRange,yRange) local p0,p00,p01,o1,o2,p1,p2,p3,p4,fx,
      fr,i,j,x,xi,opis1,opis2,opis;
> # Pierwszy obrazek ---------------------
    i:=0: xi:=x0:
    p00:=plot(f(x),x=xRange,labels=[``,``]):
    p01:=pointplot([[x0,f(x0)]],symbol=circle,color=COLOUR(RGB,0,0.5,0),
```

```
      labels=[``,``]):
    o1:=listplot([[op(1,xRange),0],[op(2,xRange),0]],color=gray,labels=[``,``]);
    o2:=listplot([[0,op(1,yRange)],[0,op(2,yRange)]],color=gray,labels=[``,``]);
    opis1:=convert(evalf(x0,10),string):
    opis:="x(0) = "||opis1:
    p0:=display([p00,p01,o1,o2],title=opis,titlefont=[COURIER,BOLD,15],
      labels=[``,``],axes=none):
> # Metoda i pozostale obrazki -----------
>   while i<N do
>     p1:=pointplot([[xi,0],[xi,f(xi)]],symbol=circle,
        color=COLOUR(RGB,0,0.5,0),labels=[``,``]):
      p2:=listplot([[xi,0],[xi,f(xi)]],color=yellow,labels=[``,``]):
      p3:=plot(f(xi)+(x-xi)*D(f)(xi),x=xRange,color=COLOUR(RGB,0.7,0.7,1),
        labels=[``,``]):
      xi:=xi-f(xi)/D(f)(xi): i:=i+1:
      p4:=pointplot([[xi,0]],symbol=circle,color=COLOUR(RGB,0.7,0.1,0.4),
        labels=[``,``]):
      fx:=evalf(f(xi)): fx:=evalf(fx,10):
      opis1:=convert(evalf(xi,10),string):
      opis2:=convert(fx,string):
      opis:="x("||i||") = "||opis1||", f(x("||i||")) = "||opis2:
      fr[i]:=display([p00,o1,o2,p1,p2,p3,p4],title=opis,
        titlefont=[COURIER,BOLD,15],labels=[``,``]):
>   od:
>   display([p0,seq(fr[j],j=1..N)],insequence=true,
      view=[xRange,yRange],labels=[``,``]):
>
> end:
>
> f:=x->sin(x-1)+((x-1)/10)^2+(x-1)/2+0.5:
>
> newton(f,8.5,12,-6..11,-10..12);
>
Warning, the name changecoords has been redefined
```

x(0) = 8.5



>

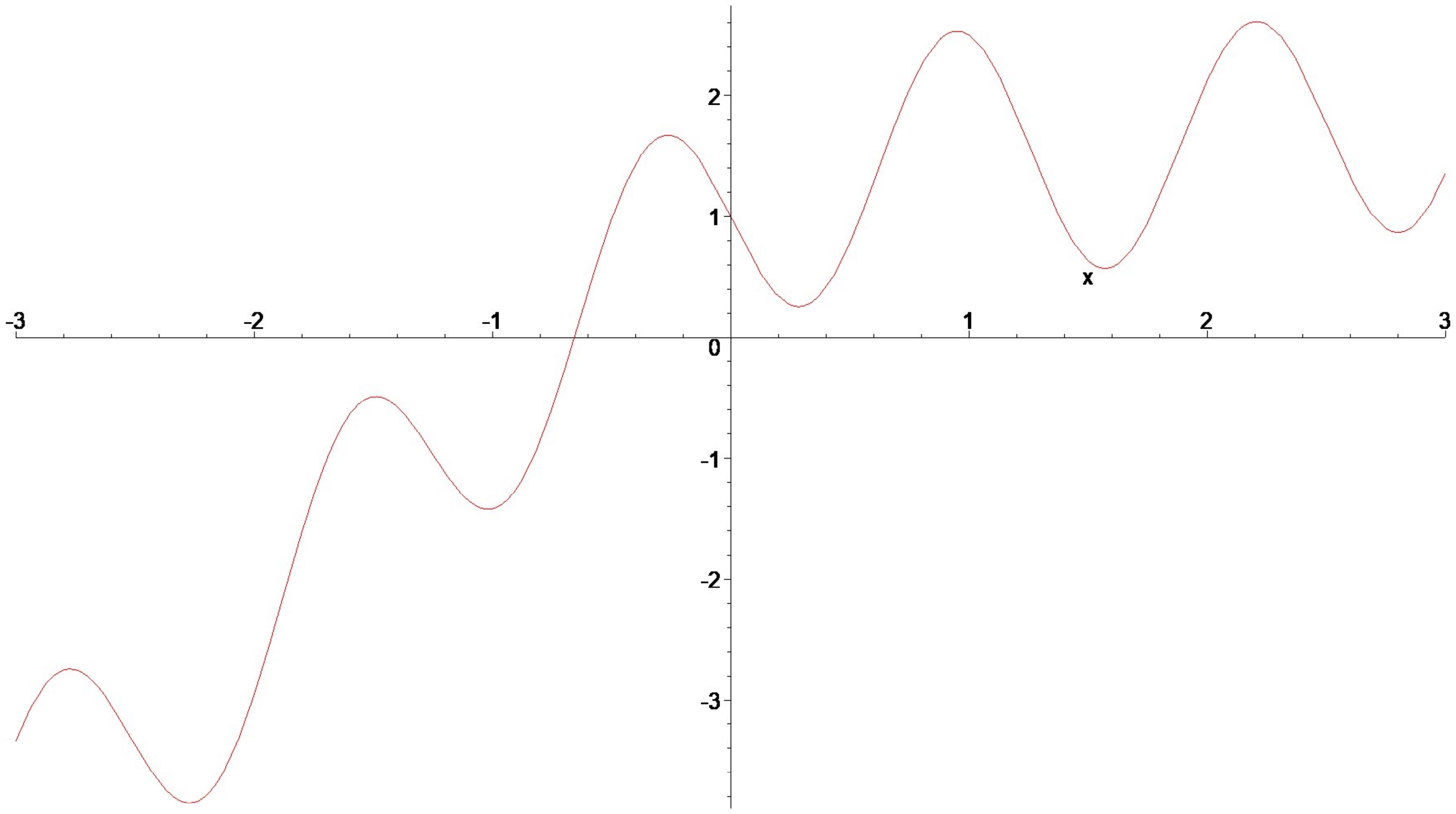
2.2. Przyklad

>

>

```
> restart:
>
> f:=x->cos(x)-sin(5*x)+x;
>
> plot(f(x),x=-3..3);
>
```

$$f := x \to \cos(x) - \sin(5\,x) + x$$




```
>
>
>
> restart;
>
> Digits:=20:
>
> f:=x->cos(x)-sin(5*x)+x:
>
> df:=unapply(diff(f(x),x),x):
>
> N:=10:
>
> print(``);print(`========================================`);print(``);
> print(`Metoda Newtona`);
> print(``);
> f(x)=0;
> print(``);
> x[n+1]=x[n]-f(x[n])/df(x[n]);
> print(``);
> print(x[0]=-0.5);
> print(``);print(`========================================`);print(``);
>
> printf("==========================\n");
> printf(" k    x[k]\n");
> printf("==========================\n");
>
> x[0]:=-0.5:
>
> for k from 0 to N
    do
```

```
x[k+1]:=x[k]-f(x[k])/df(x[k]);

printf("$%2d$ & $%1.20f$ \\\\\n",k,x[k]);
#printf("%2d   %1.20f \n",k,x[k])
od:
>
>
>
>
```

==

Metoda Newtona

$$\cos(x) - \sin(5\,x) + x = 0$$

$$x_{n+1} = x_n - \frac{\cos(x_n) - \sin(5\,x_n) + x_n}{-\sin(x_n) - 5\cos(5\,x_n) + 1}$$

$$x_0 = -0.5$$

==

```
===========================
 k   x[k]
===========================
$ 0$ & $-.50000000000000000000$ \\
$ 1$ & $-.67794515043998232787$ \\
$ 2$ & $-.65559800934709186178$ \\
$ 3$ & $-.65576723122823670992$ \\
$ 4$ & $-.65576723693058901993$ \\
$ 5$ & $-.65576723693058902645$ \\
$ 6$ & $-.65576723693058902645$ \\
$ 7$ & $-.65576723693058902645$ \\
$ 8$ & $-.65576723693058902645$ \\
$ 9$ & $-.65576723693058902645$ \\
$10$ & $-.65576723693058902645$ \\
```

```
>
>
> Digits:=30:
>
> wynik:=fsolve(f(x),x):
>
> print(``);print(`==========================================`);print(``);
> f(x)=0;
> print(``);
> print(`Rozwiazanie  `=wynik);
> print(``);print(`==========================================`);print(``);
> print(`Metoda Newtona`);
> print(``);
> printf("%46s x[%d] = %1.20f","",N,x[N]);
> print(``);print(`==========================================`);print(``);
> print(`Blad wzgledny`);
> print(``);
> print(`blad `=evalf(abs((x[N]-wynik)/wynik)));
> print(``);print(`==========================================`);print(``);
>
>
>
>
```

==

$$\cos(x) - \sin(5x) + x = 0$$

Rozwiazanie = -0.6557672369305890264428570785 83

==

Metoda Newtona

```
x[10] = -.65576723693058902645
```

==

Blad wzgledny

$$blad = 0.108924646044127644831225528694\ 10^{-19}$$

==

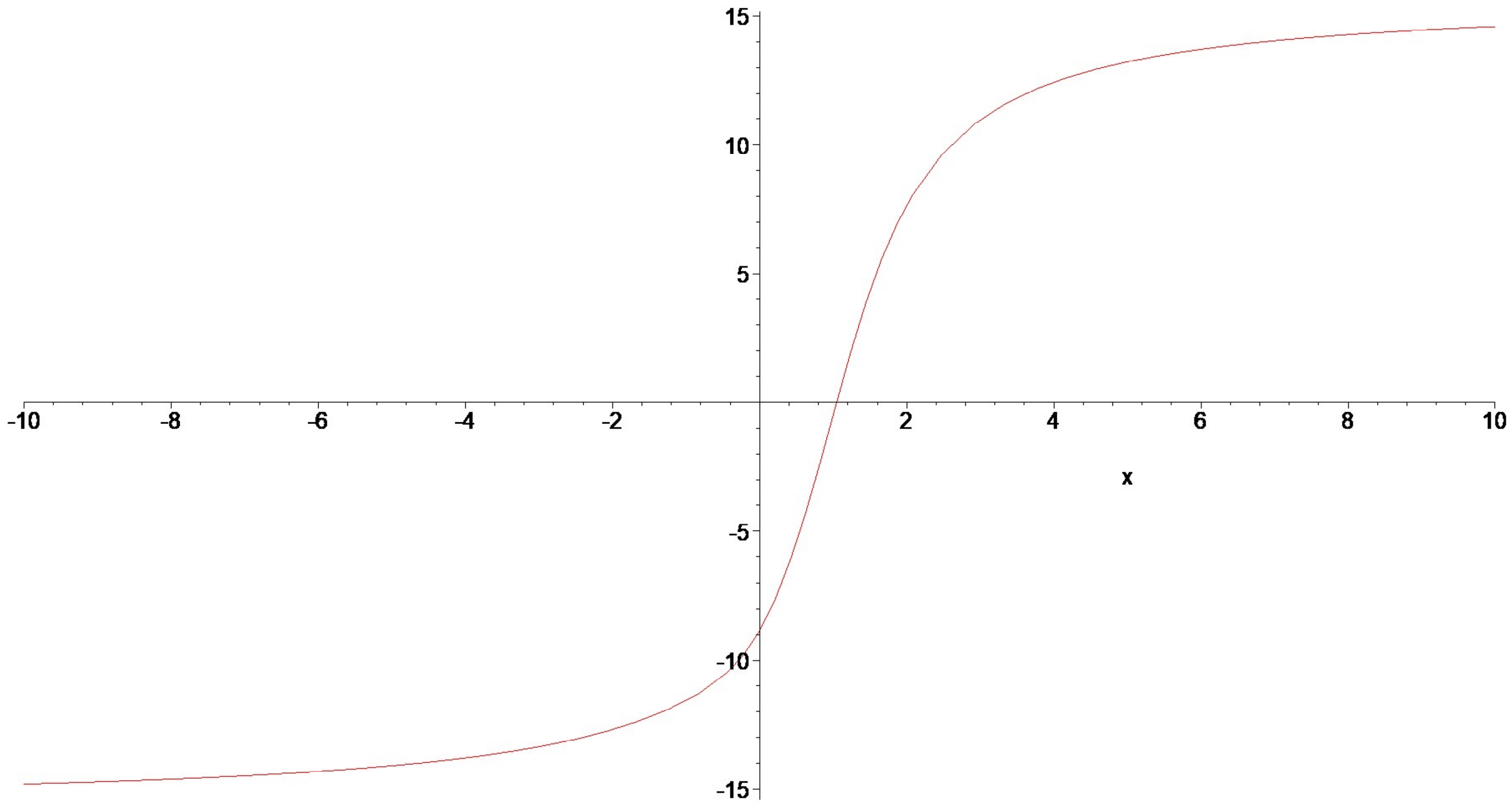
>
>

2.3. Niebezpieczenstwa

2.3.1 Przyklad

```
> 
> 
> restart:
> 
> Digits:=18:
> 
> f:=x->10*arctan(x-1)-1/(x^2+1);
> 
> df:=unapply(diff(f(x),x),x):
> 
> plot(f(x),x=-10..10);
> 
> 
```

$$f := x \rightarrow 10\arctan(x-1) - \frac{1}{x^2+1}$$




```
>
>
> N:=15:
>
> x[0]:=2.3604; ## x[0]:=2.3604; x[0]:=2.3605;
>
> for n from 0 to N
    do
      x[n+1]:=x[n]-f(x[n])/df(x[n]);
    od;
>
> wynik:=fsolve(f(x),x):
>
> print(``);
>
> `Rozwiazanie`=wynik;
>
> print(``);print(`========================================`);print(``);
> print(`Blad wzgledny`);
> print(``);
> print(`blad `=evalf(abs((x[N]-wynik)/wynik)));
> print(``);print(`========================================`);print(``);
>
>
>
```

$$x_0 := 2.3604$$

$$x_1 := -0.18766477730275611$$

$$x_2 := 2.35977266008974346$$

$$x_3 := -0.18613681572809727$$

$$x_4 := 2.35393979300013453$$

$$x_5 := -0.171981950054416000$$
$$x_6 := 2.30072163776969476$$
$$x_7 := -0.0471370447556 6936$$
$$x_8 := 1.89511457444064931$$
$$x_9 := 0.659271346118787 61$$
$$x_{10} := 1.07393532610916290$$
$$x_{11} := 1.04764856116119073$$
$$x_{12} := 1.04770760329720930$$
$$x_{13} := 1.04770760353829477$$
$$x_{14} := 1.04770760353829477$$
$$x_{15} := 1.04770760353829477$$
$$x_{16} := 1.04770760353829477$$

$$Rozwiazanie = 1.04770760353829477$$

==

Blad wzgledny

$$blad\ = 0.$$

==

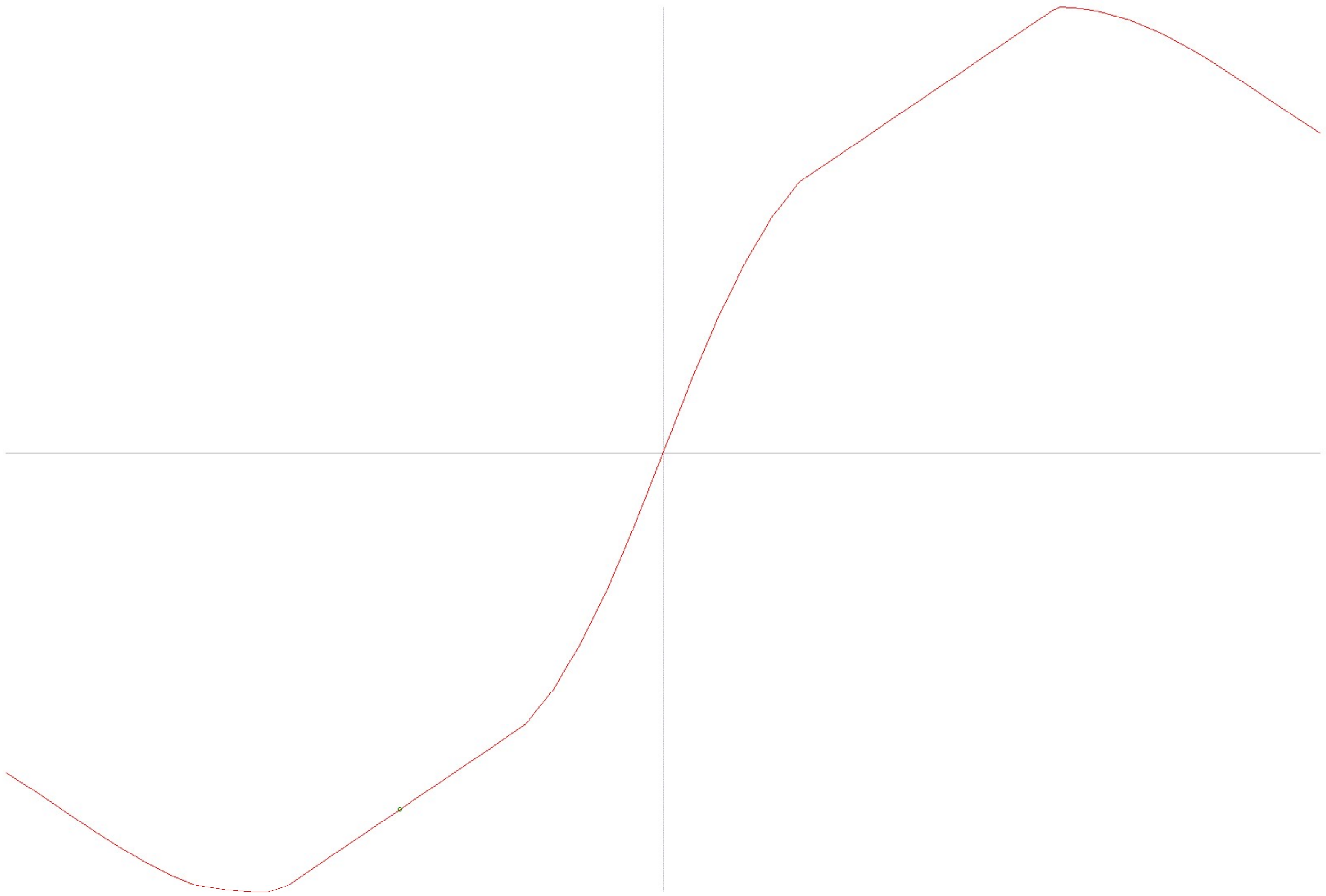
```
>
```

2.3.2 Przyklad

```
>
>
> ##############################################################################
> ### Metoda Newtona szukania miejsca zerowego funkcji - prezentacja graf. ###
> ###                                                                      ###
> ### Autor: Pawel Keller                                                  ###
> ##############################################################################
>
> restart:
>
> Digits:=30:
>
> with(plots):
>
>
> newton:=proc(f,x0,N,xRange,yRange) local p0,p00,p01,o1,o2,p1,p2,p3,p4,fx,
      fr,i,j,x,xi,opis1,opis2,opis;
> # Pierwszy obrazek ---------------------
    i:=0: xi:=x0:
    p00:=plot(f(x),x=xRange,labels=[``,``]):
    p01:=pointplot([[x0,f(x0)]],symbol=circle,color=COLOUR(RGB,0,0.5,0),
      labels=[``,``]):
    o1:=listplot([[op(1,xRange),0],[op(2,xRange),0]],color=gray,labels=[``,``]);
    o2:=listplot([[0,op(1,yRange)],[0,op(2,yRange)]],color=gray,labels=[``,``]);
    opis1:=convert(evalf(x0,10),string):
    opis:="x(0) = "||opis1:
    p0:=display([p00,p01,o1,o2],title=opis,titlefont=[COURIER,BOLD,15],
      labels=[``,``],axes=none):
> # Metoda i pozostale obrazki -----------
>   while i<N do
```

```
>     p1:=pointplot([[xi,0],[xi,f(xi)]],symbol=circle,
        color=COLOUR(RGB,0,0.5,0),labels=[``,``]):
      p2:=listplot([[xi,0],[xi,f(xi)]],color=yellow,labels=[``,``]):
      p3:=plot(f(xi)+(x-xi)*D(f)(xi),x=xRange,color=COLOUR(RGB,0.7,0.7,1),
        labels=[``,``]):
      xi:=xi-f(xi)/D(f)(xi): i:=i+1:
      p4:=pointplot([[xi,0]],symbol=circle,color=COLOUR(RGB,0.7,0.1,0.4),
        labels=[``,``]):
      fx:=evalf(f(xi)): fx:=evalf(fx,10):
      opis1:=convert(evalf(xi,10),string):
      opis2:=convert(fx,string):
      opis:="x("||i||") = "||opis1||", f(x("||i||")) = "||opis2:
      fr[i]:=display([p00,o1,o2,p1,p2,p3,p4],title=opis,
        titlefont=[COURIER,BOLD,15],labels=[``,``]):
>   od:
>   display([p0,seq(fr[j],j=1..N)],insequence=true,
      view=[xRange,yRange],labels=[``,``]):
>
> end:
>
> f:=x->piecewise(x<-3,-cos(x+3)-4,x>-3 and x<-1,x-2,x>-1 and x<1,arctan(x)/(Pi/12),x>1 and x<3,x+2,x>3,cos(x-3)+4):
>
> newton(f,-2,15,-5..5,-5..5);
>
>
Warning, the name changecoords has been redefined
```

x(0) = -2.



\>

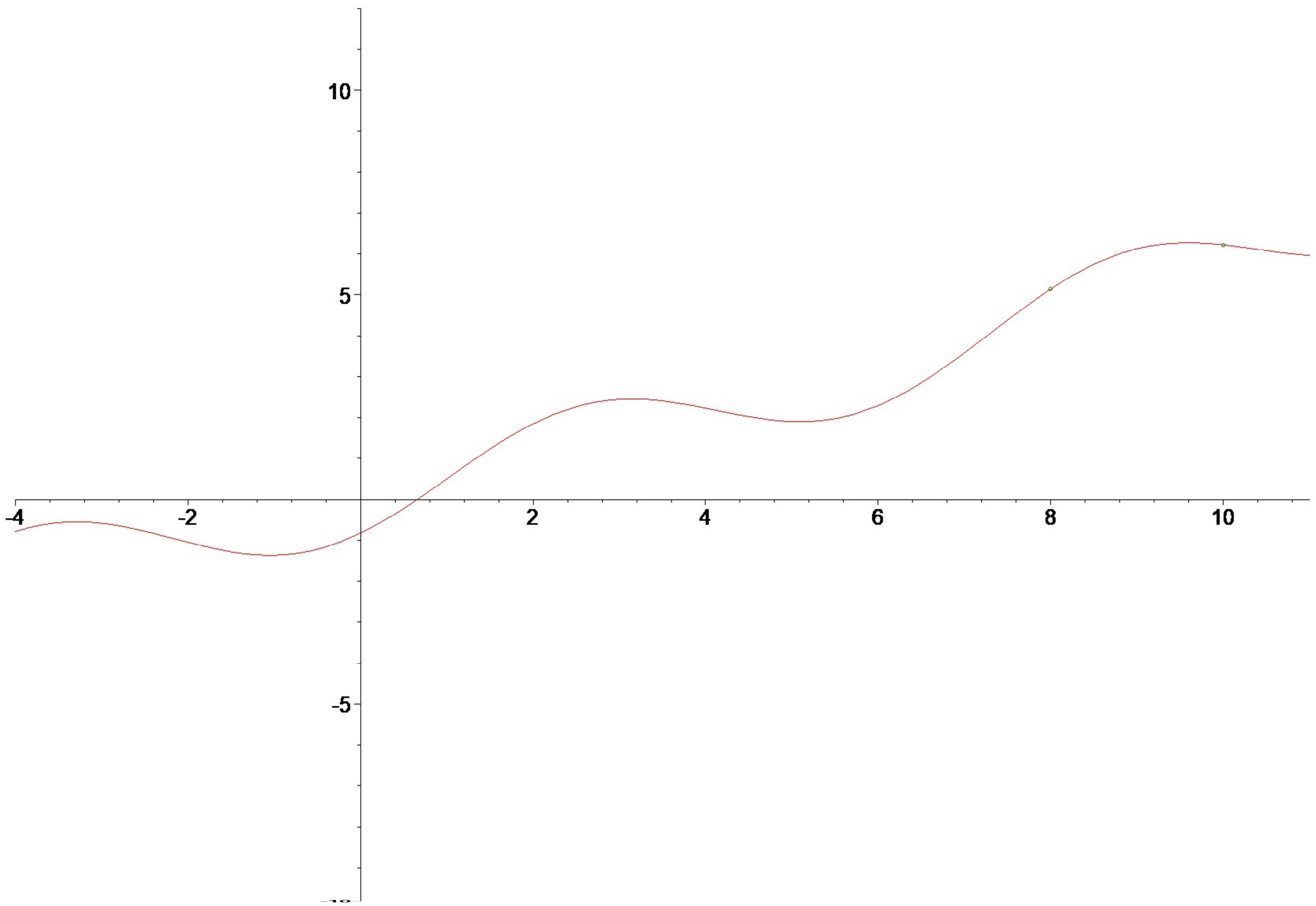
3. Metoda siecznych

3.1. Prezentacja graficzna

```
>
>
> #############################################################################
> ### Metoda siecznych szukania miejsca zerowego funkcji - prezentacja graf. ###
> ###                                                                        ###
> ### Autor: Pawel Keller                                                    ###
> #############################################################################
>
> restart:
>
> Digits:=30:
>
> with(plots):
>
> sieczne:=proc(f,x0,x1,N,xRange,yRange) local p0,p00,p01,p1,p2,p2a,p3,p4,
     fr,i,j,x,xi,xii,xiii,opis1,opis2,opis;
> # Pierwszy obrazek ---------------------
   i:=1: xi:=x0: xii:=x1:
   p00:=plot(f(x),x=xRange,labels=[``,``]):
```

```
    p01:=pointplot([[x0,f(x0)],[x1,f(x1)]],symbol=circle,
      color=COLOUR(RGB,0,0.5,0),labels=[``,``]):
    opis1:=convert(evalf(x1,16),string):
    opis:="x(1) = "||opis1:
    p0:=display([p00,p01],title=opis,titlefont=[COURIER,BOLD,15],labels=[``,``]):
> # Metoda i pozostale obrazki -----------
>   while i<N do
>     p1:=pointplot([[xi,0.0],[xi,f(xi)],[xii,0],[xii,f(xii)]],symbol=circle,
        color=COLOUR(RGB,0,0.5,0),labels=[``,``]):
      p2:=listplot([[xi,0.0],[xi,f(xi)]],color=yellow,labels=[``,``]):
      p2a:=listplot([[xii,0.0],[xii,f(xii)]],color=green,labels=[``,``]):
      p3:=plot(f(xii)+(x-xii)*(f(xii)-f(xi))/(xii-xi),x=xRange,
        color=COLOUR(RGB,0.7,0.7,1),labels=[``,``]):
      if (xii-xi=0) then xii:=xii
                    else xiii:=xii-f(xii)*(xii-xi)/(f(xii)-f(xi)) fi:
      i:=i+1:
      p4:=pointplot([[xiii,0]],symbol=circle,color=COLOUR(RGB,0.7,0.1,0.4),
        labels=[``,``]):
      opis1:=convert(evalf(xiii,16),string):
      opis2:=convert(f(xiii),string):
      opis:="x("||i||") = "||opis1||", f(x("||i||")) = "||opis2:
      fr[i]:=display([p00,p1,p2,p2a,p3,p4],title=opis,
        titlefont=[COURIER,BOLD,15],labels=[``,``]):
      xi:=xii: xii:=xiii:
>   od:
>   display([p0,seq(fr[j],j=2..N)],insequence=true,
      view=[xRange,yRange],labels=[``,``]):
> end:
>
>
> f:=x->sin(x-1)+((x-1)/10)^2+(x-1)/2+0.5:
>
> sieczne(f,8.0,10.0,12,-4..11,-10..12);
>
Warning, the name changecoords has been redefined
```

x(1) = 10.0



>

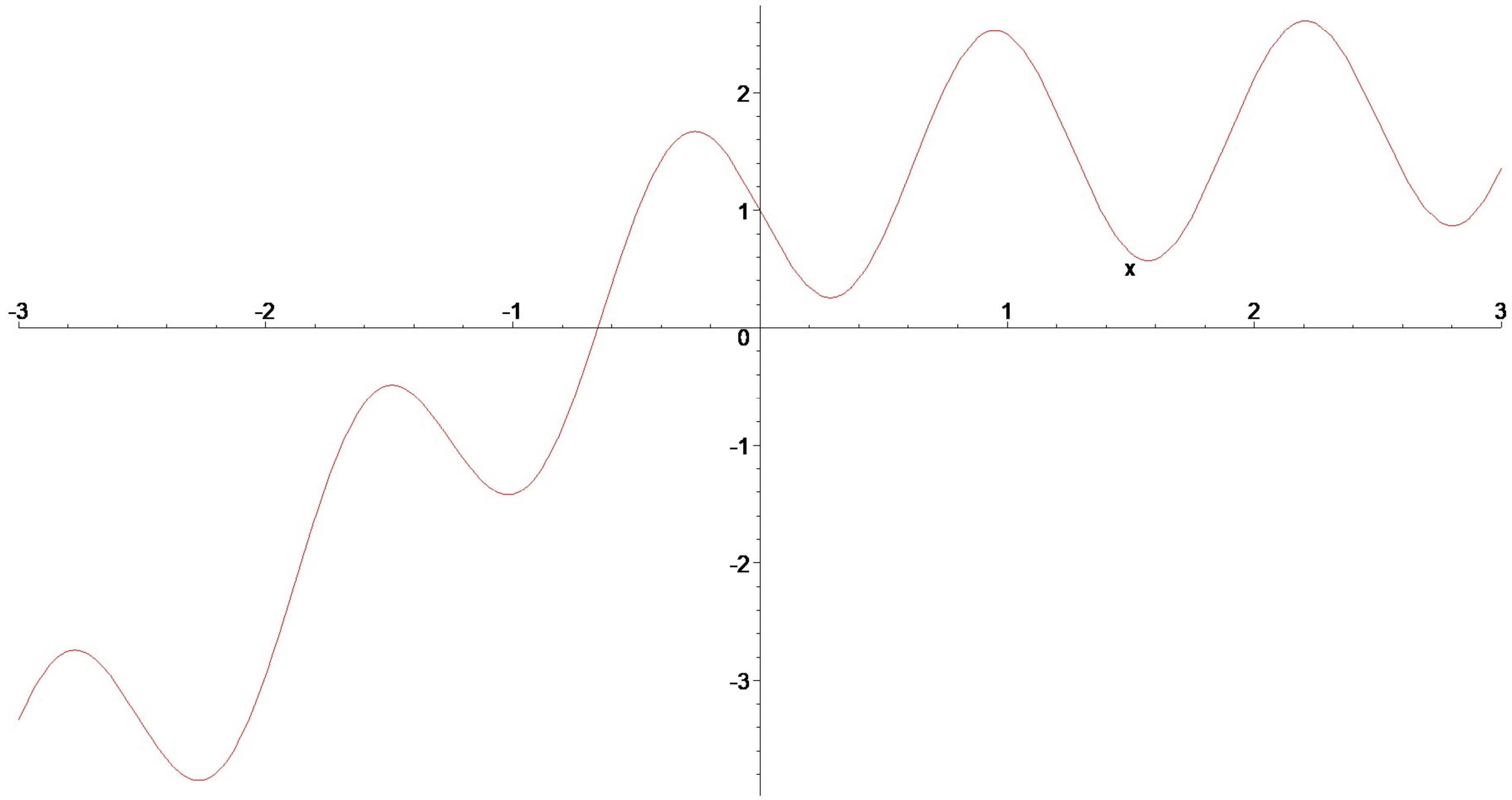
3.2 Przyklad

>

>

```
> restart:
>
> f:=x->cos(x)-sin(5*x)+x;
>
> plot(f(x),x=-3..3);
>
```

$$f := x \to \cos(x) - \sin(5\,x) + x$$




```
>
>
> restart;
>
> Digits:=20:
>
> f:=x->cos(x)-sin(5*x)+x:
>
> N:=12:
>
> print(``);print(`=========================================`);print(``);
> print(`Metoda siecznych`);
> print(``);
> f(x)=0;
> print(``);
> x[n+1]=x[n]-f(x[n])*(x[n-1]-x[n])/(f(x[n-1])-f(x[n]));
> print(``);
> print(x[0]=-1.0,`    `,x[1]=0.0);
> print(``);print(`=========================================`);print(``);
>
> printf("===========================\n");
> printf(" k    x[k]\n");
> printf("===========================\n");
>
> x[0]:=-1.0:
> x[1]:=0.0:
>
> for k from 1 to N+1
    do
      x[k+1]:=x[k]-f(x[k])*(x[k-1]-x[k])/(f(x[k-1])-f(x[k]));

      printf("$%2d$ & $%1.20f$ \\\\\n",k-1,x[k-1]);
```

```
      #printf("%2d   %1.20f \n",k-1,x[k-1])
    od:
>
>
```

==

Metoda siecznych

$$\cos(x) - \sin(5\,x) + x = 0$$

$$x_{n+1} = x_n - \frac{(\cos(x_n) - \sin(5\,x_n) + x_n)\,(x_{n-1} - x_n)}{\cos(x_{n-1}) - \sin(5\,x_{n-1}) + x_{n-1} - \cos(x_n) + \sin(5\,x_n) - x_n}$$

$$x_0 = \text{-}1.0, \quad , x_1 = 0.$$

==

```
==========================
 k   x[k]
==========================
$ 0$ & $-1.00000000000000000000$ \\
$ 1$ & $0.00000000000000000000$ \\
$ 2$ & $-.41345857802582439236$ \\
$ 3$ & $1.08367156761733602380$ \\
$ 4$ & $-2.63459485977556400280$ \\
$ 5$ & $-.55385141479322637280$ \\
$ 6$ & $-.93566705646569705430$ \\
$ 7$ & $-.67979388694116828879$ \\
$ 8$ & $-.64597678242853749324$ \\
$ 9$ & $-.65582509370868559267$ \\
$10$ & $-.65576733272725822950$ \\
$11$ & $-.65576723692947838382$ \\
$12$ & $-.65576723693058902646$ \\
```

```
>
>
> Digits:=30:
>
> wynik:=fsolve(f(x),x):
>
> print(``);print(`==========================================`);print(``);
> f(x)=0;
> print(``);
> print(`Rozwiazanie  `=wynik);
> print(``);print(`==========================================`);print(``);
> print(`Metoda siecznych`);
> print(``);
> printf("%46s x[%d] = %1.20f","",N,x[N]);
> print(``);print(`==========================================`);print(``);
> print(`Blad wzgledny`);
> print(``);
> print(`blad `=evalf(abs((x[N]-wynik)/wynik)));
> print(``);print(`==========================================`);print(``);
>
```

==

$$\cos(x) - \sin(5\,x) + x = 0$$

Rozwiazanie $= \text{-}0.655767236930589026442857078583$

==

Metoda siecznych

x[12] = -.65576723693058902646

==

Blad wzgledny

$blad = 0.261417778314754786229084657197\ 10^{-19}$

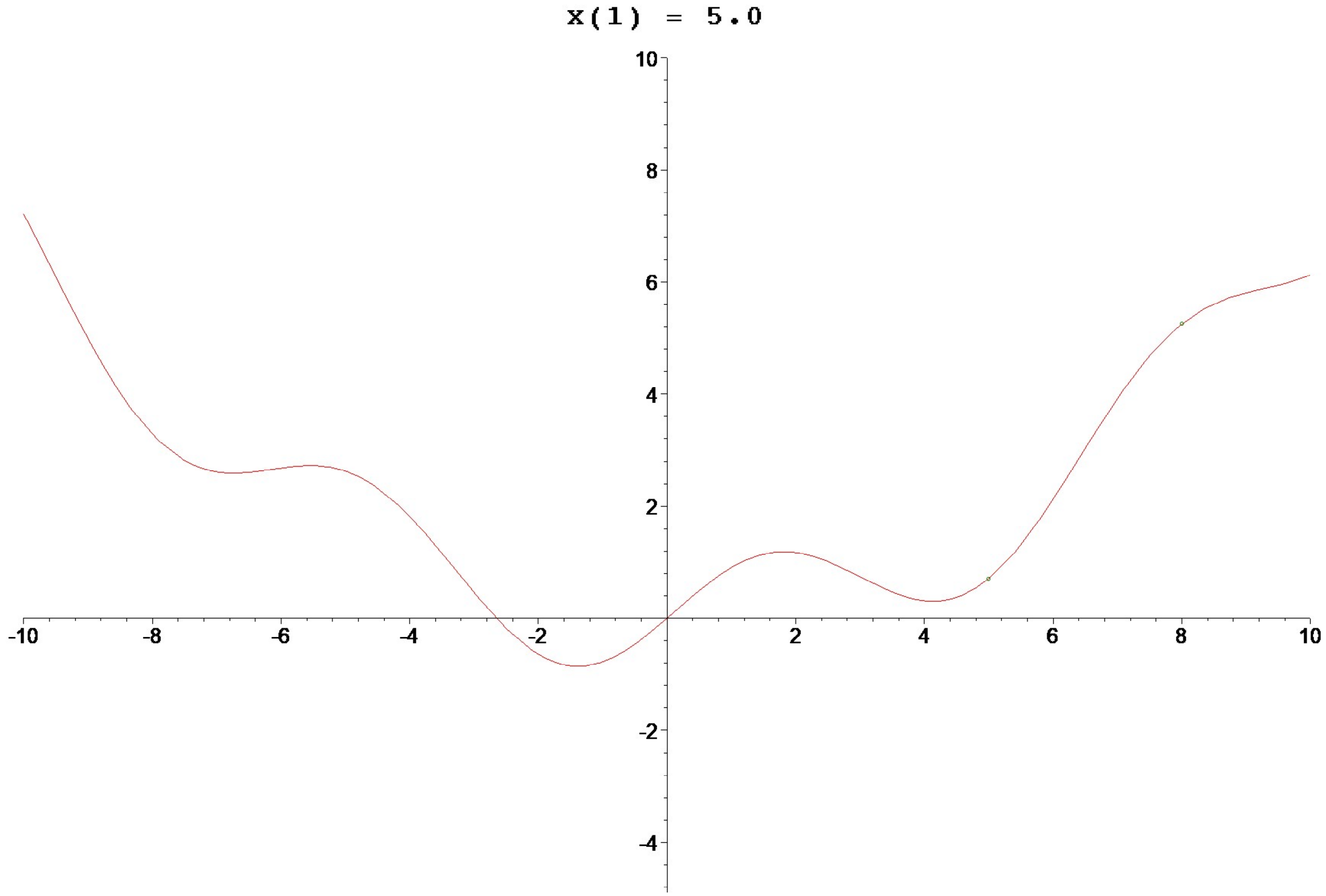
==

\>

3.3 Pewna uwaga

```
>
>
> ##############################################################################
> ### Metoda siecznych szukania miejsca zerowego funkcji - prezentacja graf. ###
> ###                                                                        ###
> ### Autor: Pawel Keller                                                    ###
> ##############################################################################
>
> restart:
>
> Digits:=30:
>
> with(plots):
>
> sieczne:=proc(f,x0,x1,N,xRange,yRange) local p0,p00,p01,p1,p2,p2a,p3,p4,
     fr,i,j,x,xi,xii,xiii,opis1,opis2,opis;
> # Pierwszy obrazek ---------------------
   i:=1: xi:=x0: xii:=x1:
   p00:=plot(f(x),x=xRange,labels=[``,``]):
   p01:=pointplot([[x0,f(x0)],[x1,f(x1)]],symbol=circle,
     color=COLOUR(RGB,0,0.5,0),labels=[``,``]):
   opis1:=convert(evalf(x1,16),string):
   opis:="x(1) = "||opis1:
   p0:=display([p00,p01],title=opis,titlefont=[COURIER,BOLD,15],labels=[``,``]):
> # Metoda i pozostale obrazki -----------
   while i<N do
>    p1:=pointplot([[xi,0.0],[xi,f(xi)],[xii,0],[xii,f(xii)]],symbol=circle,
       color=COLOUR(RGB,0,0.5,0),labels=[``,``]):
     p2:=listplot([[xi,0.0],[xi,f(xi)]],color=yellow,labels=[``,``]):
     p2a:=listplot([[xii,0.0],[xii,f(xii)]],color=green,labels=[``,``]):
     p3:=plot(f(xii)+(x-xii)*(f(xii)-f(xi))/(xii-xi),x=xRange,
       color=COLOUR(RGB,0.7,0.7,1),labels=[``,``]):
     if (xii-xi=0) then xii:=xii
                   else xiii:=xii-f(xii)*(xii-xi)/(f(xii)-f(xi)) fi:
     i:=i+1:
     p4:=pointplot([[xiii,0]],symbol=circle,color=COLOUR(RGB,0.7,0.1,0.4),
       labels=[``,``]):
     opis1:=convert(evalf(xiii,16),string):
     opis2:=convert(f(xiii),string):
     opis:="x("||i||") = "||opis1||", f(x("||i||")) = "||opis2:
     fr[i]:=display([p00,p1,p2,p2a,p3,p4],title=opis,
       titlefont=[COURIER,BOLD,15],labels=[``,``]):
     xi:=xii: xii:=xiii:
```

```
>   od:
>   display([p0,seq(fr[j],j=2..N)],insequence=true,
      view=[xRange,yRange],labels=[``,``]):
> end:
>
>
> f:=x->sin(x)+x^2/15:
>
> sieczne(f,8.0,5.0,40,-10..10,-5..10);
>
Warning, the name changecoords has been redefined
```




```
>
```

4. Numeryczne wyznaczanie rzedu metody

5. Kto wygra?