Topic : Newton Raphson Method - Roots of Equations
Simulation : Pitfall - Oscillation around a maxima or minima
Language : Mathmatica 4.1
Authors : Nathan Collier, Autar Kaw
Date : 11 July 2002
Abstract : This simulation illustrates a pitfall of the Newton-Raphson method where one is getting oscilation around a local maxima or minima of a function.

■ INPUTS: Enter the Following

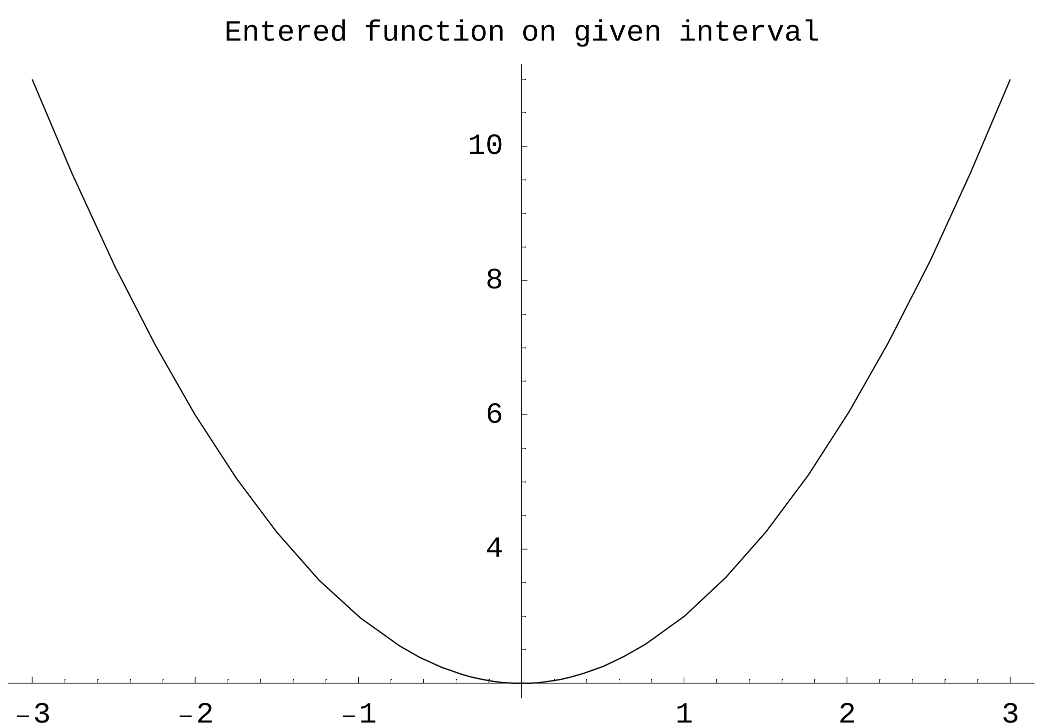
Function in $f[x]$ 0

In[487]:=
```
f[x_] := x^2 + 2
```

Range of 'x' you want to see the function

In[488]:=
```
x_begin = -3;
x_end = 3;
```

In[490]:=
```
curve = Plot[f[x], {x, x_begin, x_end}, PlotLabel →
    "Entered function on given interval", TextStyle → {FontSize → 11}];
```



Initial guess

In[491]:=
```
x0 = -1;
```

Maximum number of iterations

In[492]:=
```
n_maximum = 100;
```

Because this method uses a line tangent to the function at the initial guess, we must calculate the derivative of the function to find the slope of the line at this point. Here we will define the derivative of the function $f(x)$ as $g(x)$.

In[493]:=
```
g[x_] := f'[x]
```

Iteration 1

In[494]:=
```
x1 = x0 - f[x0] / g[x0]
```

Out[494]= $\frac{1}{2}$

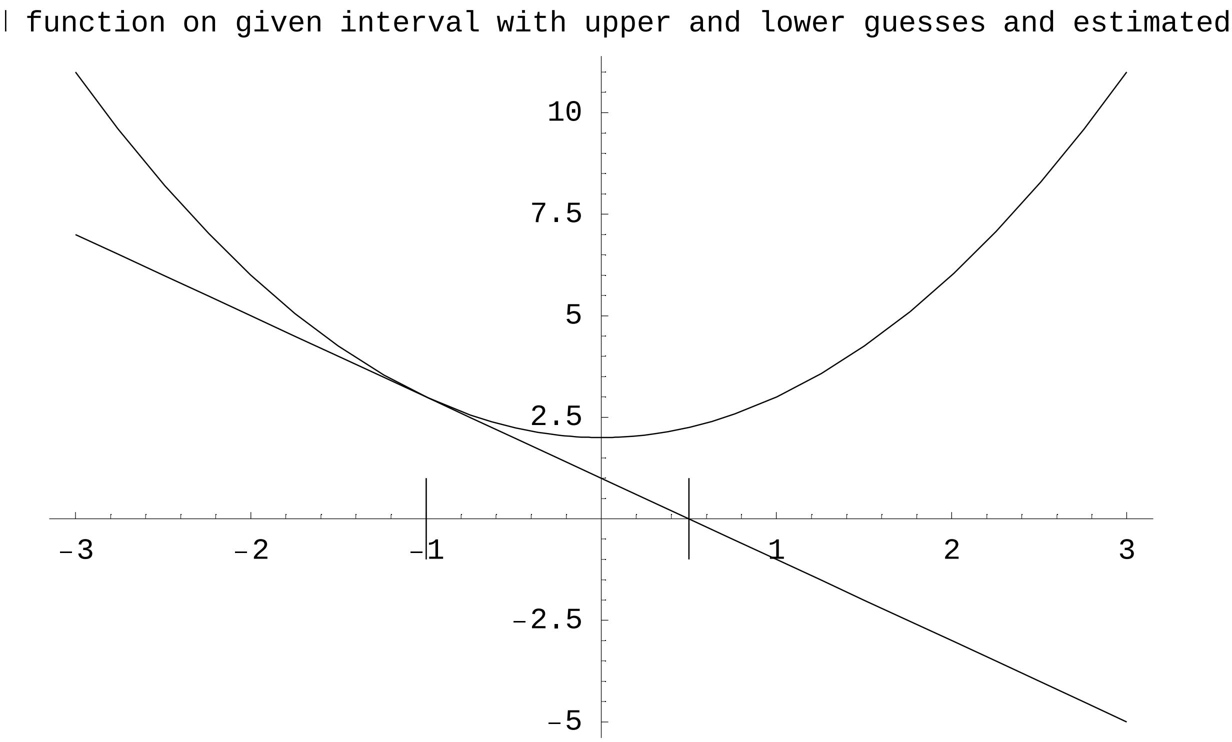
In[495]:=
```
ϵ_a = Abs[(x1 - x0) / x1 * 100]
```

Out[495]= 300

In[496]:=
```
tanline[x_] := f[x0] + ((0 - f[x0]) / (x1 - x0)) * (x - x0)
```

In[497]:=
```
tline = Plot[tanline[x], {x, x_begin, x_end}];
```

In[498]:=
```
Show[Graphics[Line[{{x0, 1}, {x0, -1}}]], curve,
  Graphics[Line[{{x1, 1}, {x1, -1}}]], tline, Axes → True,
  PlotLabel → "Entered function on given interval with upper and
     lower guesses and estimated root", TextStyle → {FontSize → 11}];
```



Iteration 2

In[499]:=
```
x2 = x1 - f[x1] / g[x1]
```

Out[499]= $-\frac{7}{4}$

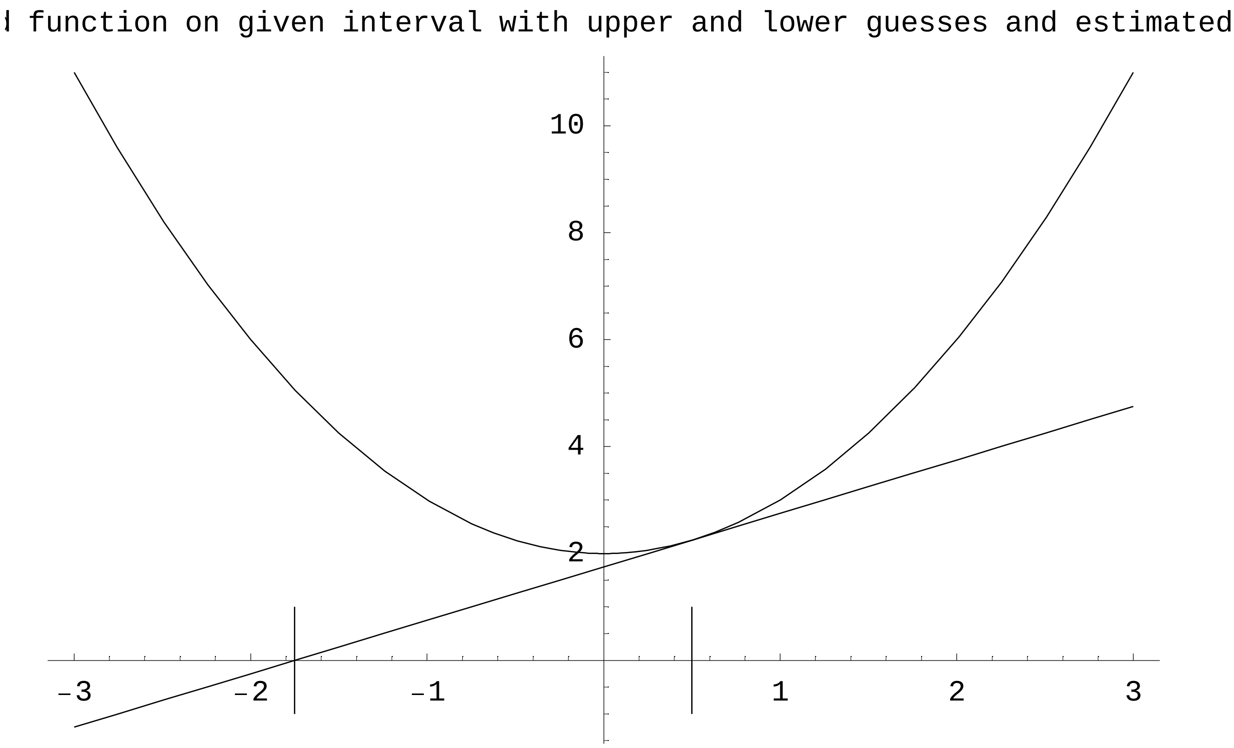
In[500]:=
```
ϵa = Abs[(x2 - x1) / x2 * 100]
```

Out[500]= $\frac{900}{7}$

In[501]:=
```
tanline[x_] := f[x1] + ((0 - f[x1]) / (x2 - x1)) * (x - x1)
```

In[502]:=
```
tline = Plot[tanline[x], {x, xbegin, xend}];
```

In[503]:=
```
Show[Graphics[Line[{{x1, 1}, {x1, -1}}]], curve,
  Graphics[Line[{{x2, 1}, {x2, -1}}]], tline, Axes → True,
  PlotLabel → "Entered function on given interval with upper and
     lower guesses and estimated root", TextStyle → {FontSize → 11}];
```



Iteration 3

In[504]:=
```
x3 = x2 - f[x2] / g[x2]
```

Out[504]= $-\frac{17}{56}$

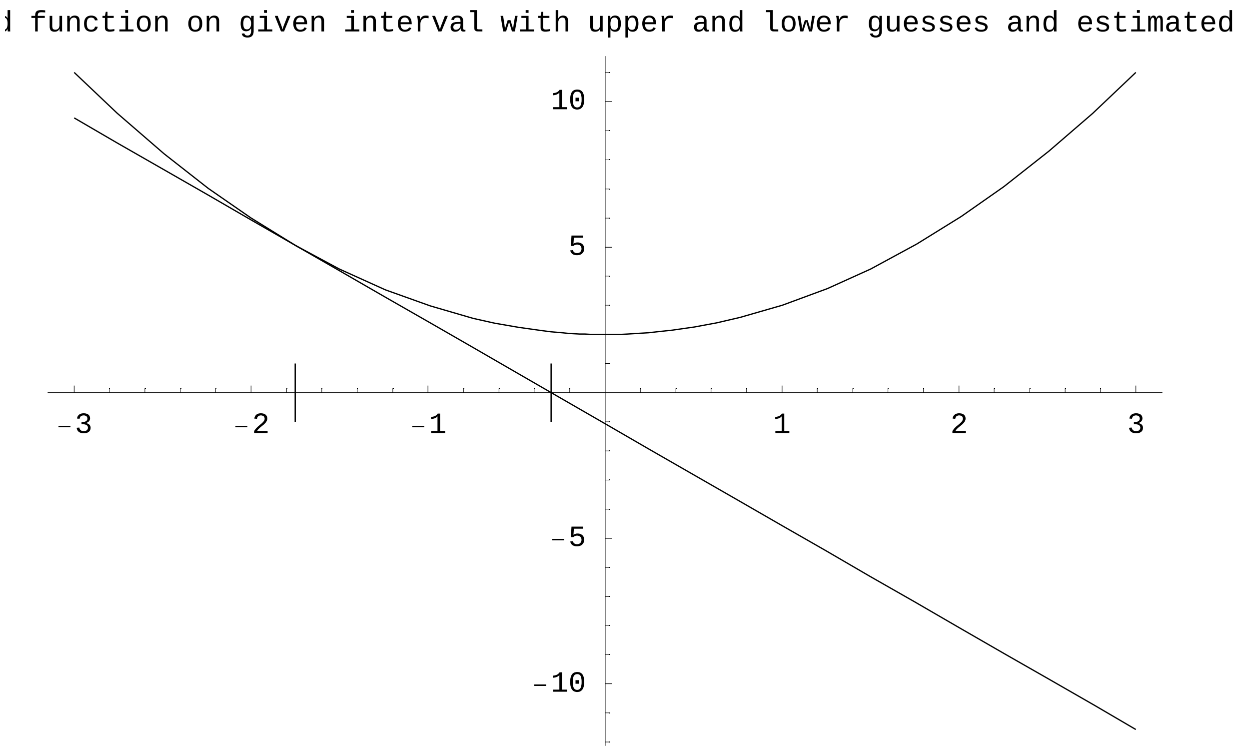
In[505]:=
```
ϵa = Abs[(x3 - x2) / x3 * 100]
```

Out[505]= $\frac{8100}{17}$

In[506]:=
```
tanline[x_] := f[x2] + ((0 - f[x2]) / (x3 - x2)) * (x - x2)
```

In[507]:=
```
tline = Plot[tanline[x], {x, xbegin, xend}];
```

In[508]:=
```
Show[Graphics[Line[{{x3, 1}, {x3, -1}}]], curve,
  Graphics[Line[{{x2, 1}, {x2, -1}}]], tline, Axes → True,
  PlotLabel → "Entered function on given interval with upper and
     lower guesses and estimated root", TextStyle → {FontSize → 11}];
```



Iteration 4

In[509]:=
```
x4 = x3 - f[x3] / g[x3]
```

Out[509]= $\frac{5983}{1904}$

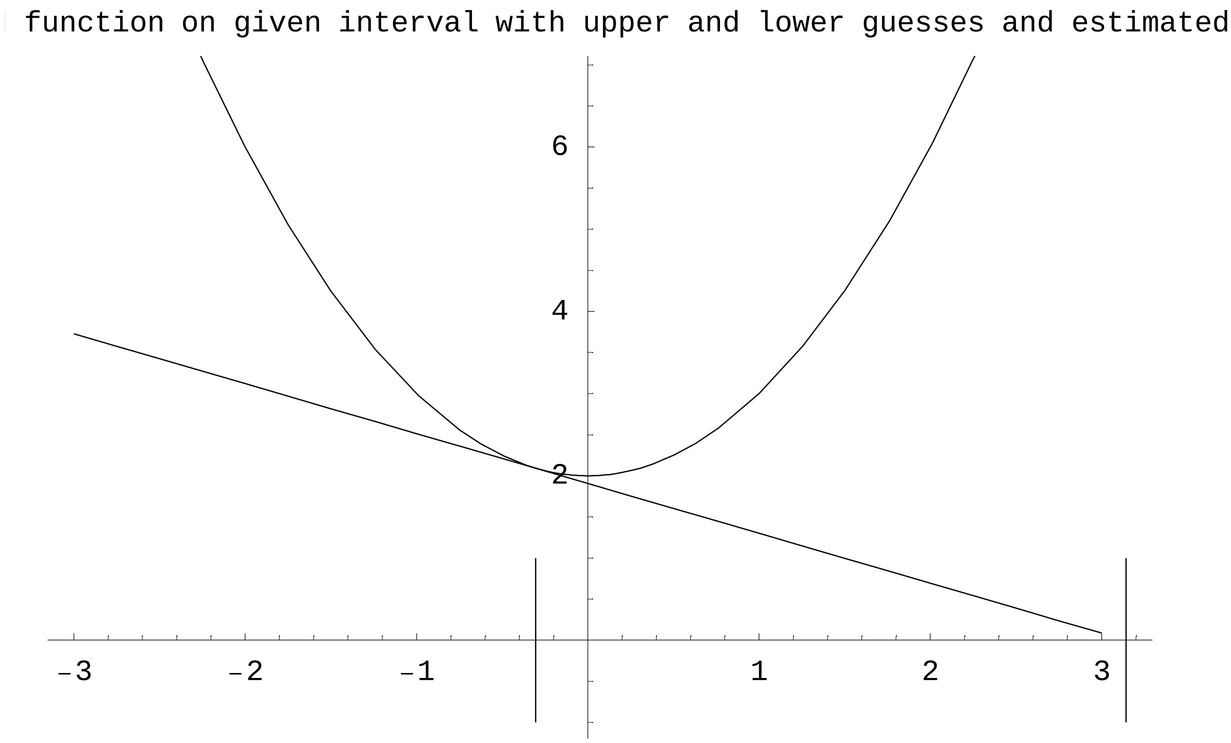
In[510]:=
```
ϵa = Abs[(x4 - x3) / x4 * 100]
```

Out[510]= $\frac{656100}{5983}$

In[511]:=
```
tanline[x_] := f[x3] + ((0 - f[x3]) / (x4 - x3)) * (x - x3)
```

In[512]:=
```
tline = Plot[tanline[x], {x, xbegin, xend}];
```

In[513]:=
```
Show[Graphics[Line[{{x3, 1}, {x3, -1}}]], curve,
  Graphics[Line[{{x4, 1}, {x4, -1}}]], tline, Axes → True,
  PlotLabel → "Entered function on given interval with upper and
    lower guesses and estimated root", TextStyle → {FontSize → 11}];
```



Iteration 5

In[514]:=
```
x5 = x4 - f[x4] / g[x4]
```

Out[514]= $\frac{28545857}{22783264}$

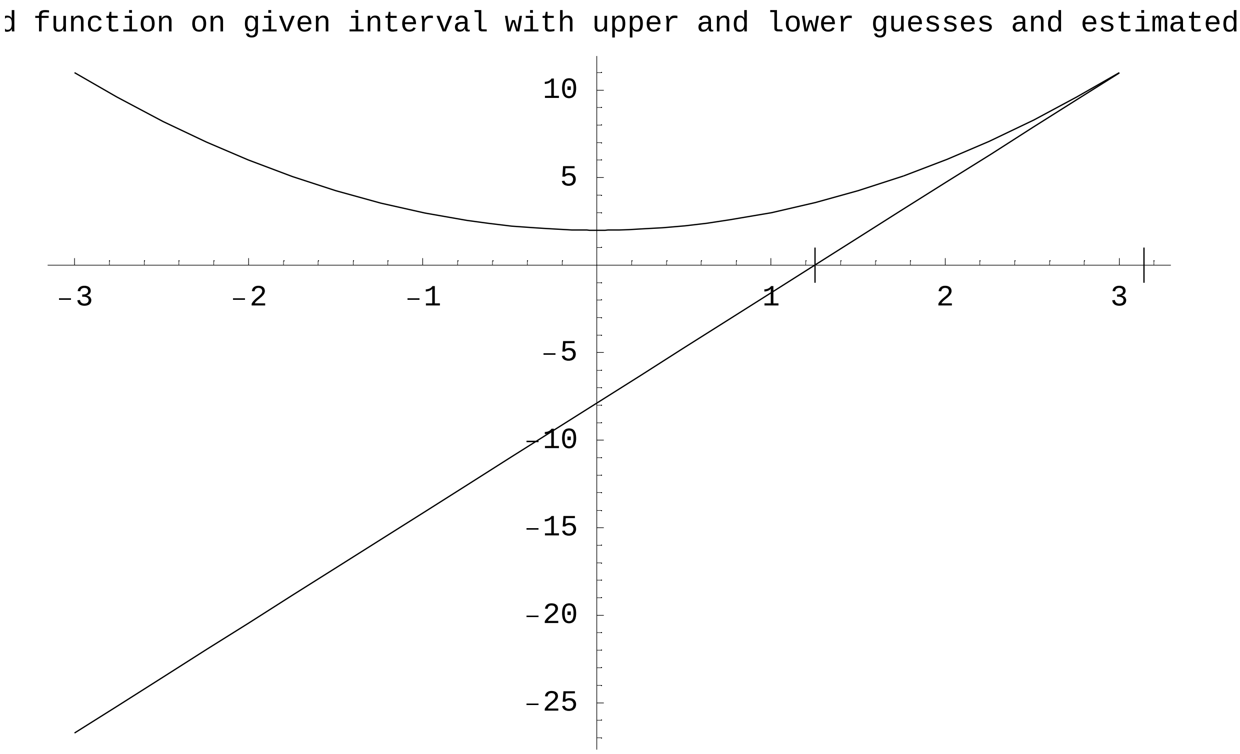
In[515]:=
```
ϵa = Abs[(x5 - x4) / x5 * 100]
```

Out[515]= $\frac{4304672100}{28545857}$

In[516]:=
```
tanline[x_] := f[x4] + ((0 - f[x4]) / (x5 - x4)) * (x - x4)
```

In[517]:=
```
tline = Plot[tanline[x], {x, xbegin, xend}];
```

In[518]:=
```
Show[Graphics[Line[{{x5, 1}, {x5, -1}}]], curve,
  Graphics[Line[{{x4, 1}, {x4, -1}}]], tline, Axes → True,
  PlotLabel → "Entered function on given interval with upper and
    lower guesses and estimated root", TextStyle → {FontSize → 11}];
```



Summary

■ Value of root as a function of iterations

Here the Newton-Raphson method algorithm is applied to generate the values of the roots, thrue error, absolute relative true error, approximate error, absolute relative approximate error, and the number of significant digits at least correct in the estimated root as a function of number of iterations.

In[519]:=
```
Clear[xr];
```

In[520]:=
```
Array[xr, n_maximum];
```

In[521]:=
```
For[i = 1; xr[0] = x0, i <= n_maximum, i++,
  xr[i] = xr[i - 1] - f[xr[i - 1]] / g[xr[i - 1]] // N]
```

In[522]:=
```
xrplot = Table[xr[i], {i, 0, n_maximum}];
```

■ Absolute relative approximate error

In[523]:=
```
Array[ϵa, n_maximum];
```

In[524]:=
```
For[i = 1, i <= n_maximum, i++, ϵa[i] = Abs[(xr[i] - xr[i - 1]) / xr[i] * 100]]
```

In[525]:=
```
ϵaplot = Table[ϵa[i], {i, 0, n_maximum - 1}];
```

In[526]:=
```
ListPlot[ϵaplot, PlotJoined → True,
  PlotRange -> All, AxesOrigin -> {1, Min[ϵaplot]},
  PlotLabel -> "Absolute relative approximate error
     as a function of number of iterations"];
```

Absolute relative approximate error as a function of number of iterations