Subject : The following demonstrates the convergence of the Gaussian method of estimating integrals of continuous functions.

Authors : Nathan Collier, Autar Kaw, Loubna Guennoun

Date : 27 October 2005

■ Introduction

auss Quadrature Rule is another method of estimating an integral. The theory behind the two point Gauss Quadrature Rule is to approximate the integral by taking the area under a straight line connecting any two points on the curve that are not predetermined as *a* and *b*, but as unknowns x_1 and x_2. For *n*-points rules, the general form to approximate the integral is

$$\int_a^b f(x)\,dx = c_0 \cdot f(x_0) + c_1 \cdot f(x_1) + \ldots. + c_n \cdot f(x_n)$$

where c_i and x_i are the weighting factors and function arguments used in Gauss Quadrature formulas, respectively. However, these factors and arguments are already defined to approximate any integral from -1 to 1. To be able to use them, the limits of the integral of the function *f(x)* need to be changed to [-1,1].

$$\int_a^b f(x)\,dx = \int_{-1}^{1} f\left(\frac{b-a}{2} \cdot x + \frac{b+a}{2}\right) \cdot \frac{b-a}{2}\,dx$$

NOTE: Weighting factors *c* and function arguments *x* used in Gauss Quadrature Rule have already been defined in the textbook for up to six points.

The following procedure will illustrate the Gauss Quadrature Rule of integration. The user may enter any function *f(x)*, the lower and upper limit for the function, and the number of points *n* in the data section (up to six points). By entering this data, the program will calculate the exact value of the integral, followed by the results using the Gauss Quadrature Rule with *n* points. The program will also display the true error, the absolute relative true percentage error, the approximate error, the absolute relative approximate percentage error, and the number of significant digits that are at least correct.

■ Inputs

Integrand $f[x] = 0$

```
In[389]:= f[x_] := 300.0 * x / (1.0 + Exp[x]);
```

Lower and upper limit of the integral, *a* and *b* respectively.

```
In[390]:= a = 0.0;
          b = 10.0;
```

Maximum number of points. This number must be between 1 and 6.

In[392]:= $n_{maximum} = 6;$

■ Weighting Factors and Functions Arguments for up to 6 Points

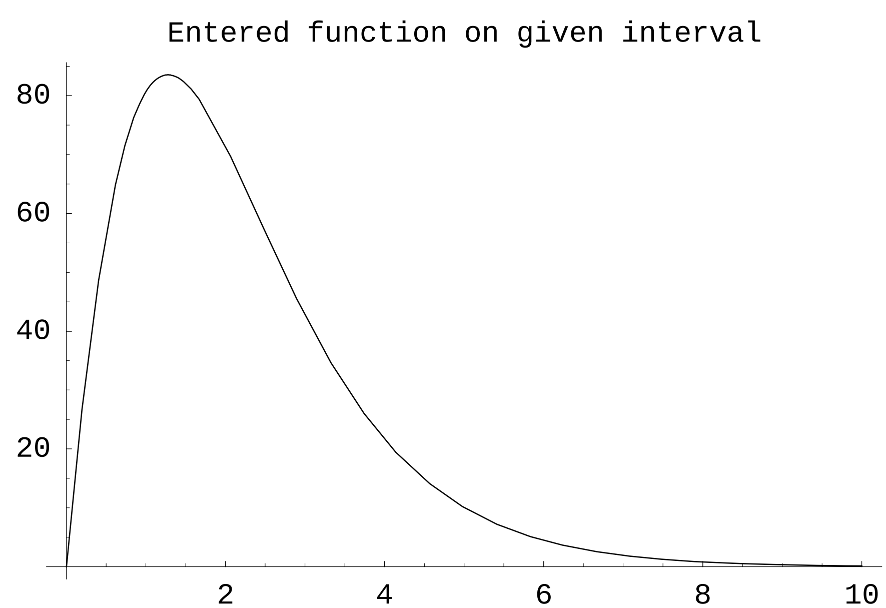
Weighting Factors for 1-6 Points

In[393]:=
$C_{1,0} = 2.0;$
$C_{1,1} = 1.0;$
$C_{2,1} = 1.0;$
$C_{1,2} = 0.5555556;$
$C_{2,2} = 0.8888889;$
$C_{3,2} = 0.5555556;$
$C_{1,3} = 0.347854845;$
$C_{2,3} = 0.652145155;$
$C_{3,3} = 0.652145155;$
$C_{4,3} = 0.347854845;$
$C_{1,4} = 0.236926885;$
$C_{2,4} = 0.478628670;$
$C_{3,4} = 0.568888889;$
$C_{4,4} = 0.478628670;$
$C_{5,4} = 0.236926885;$
$C_{1,5} = 0.171324492;$
$C_{2,5} = 0.360761573;$
$C_{3,5} = 0.467913935;$
$C_{4,5} = 0.467913935;$
$C_{5,5} = 0.360761573;$
$C_{6,5} = 0.171324492;$

Function Arguements for 1-6 Points

In[414]:=
```
X1,0 = 0.0;
X1,1 = -0.5773503;
X2,1 = 0.5773503;
X1,2 = -0.774596669;
X2,2 = 0.0;
X3,2 = 0.774596669;
X1,3 = -0.861136312;
X2,3 = -0.339981044;
X3,3 = 0.339981044;
X4,3 = 0.861136312;
X1,4 = -0.906179846;
X2,4 = -0.538469310;
X3,4 = 0.0;
X4,4 = 0.538469310;
X5,4 = 0.906179846;
X1,5 = -0.932469514;
X2,5 = -0.661209386;
X3,5 = -0.238619186;
X4,5 = 0.238619186;
X5,5 = 0.661209386;
X6,5 = 0.932469514;
```

In[435]:=
```
curve = Plot[f[x], {x, a, b}, PlotLabel → "Entered function on given interval",
   TextStyle → {FontSize → 11}];
General::spelll Off;
```



■ True Solution

This is the solution found by Mathematica

In[437]:=
```
Actual = Integrate[f[x], {x, a, b}]
```
Out[437]=
```
246.59
```

■ Value of integral as a function of iterations

The integral given above has the limits of[a,b].It needs to be converted into an integral with limits[-1,1] fnew(x) is the new function that will be used for evaluating the integral using the Gauss Quadrature rule

In[438]:=
```
fnew[x_] := f[(b - a) * x / 2 + (b + a) / 2] * (b - a) / 2;
```

Gaussian

In[439]:=
```
Array[AV, n_maximum];
```

In[440]:=
```
For[i = 1, i ≤ n_maximum, i++,
  AV[i] = Sum[C_{j,i-1} * fnew[X_{j,i-1}], {j, 1, i}]]
```

■ Absolute true error

In[441]:=
```
Array[Et, n_maximum];
```

In[442]:=
```
For[i = 1, i <= n_maximum, i++, Et[i] = Abs[Actual - AV[i]]]
```

■ Absolute relative true error

In[443]:=
```
Array[∈t, n_maximum];
```

In[444]:=
```
For[i = 1, i <= n_maximum, i++, ∈t[i] = Abs[Et[i] / Actual * 100]]
```

■ Absolute approximate error

In[445]:=
```
Array[Ea, n_maximum];
```

In[446]:=
```
For[i = 1, i <= n_maximum, i++, If[i <= 1, Ea[i] = 0, Ea[i] = Abs[AV[i] - AV[i - 1]]]]
```

■ Absolute relative approximate error

In[447]:=
```
Array[∈a, n_maximum];
```

In[448]:=
```
For[i = 1, i <= n_maximum, i++, If[i <= 1, ∈a[i] = 0, ∈a[i] = Abs[Ea[i] / AV[i] * 100]]]
```

■ Significant digits at least correct

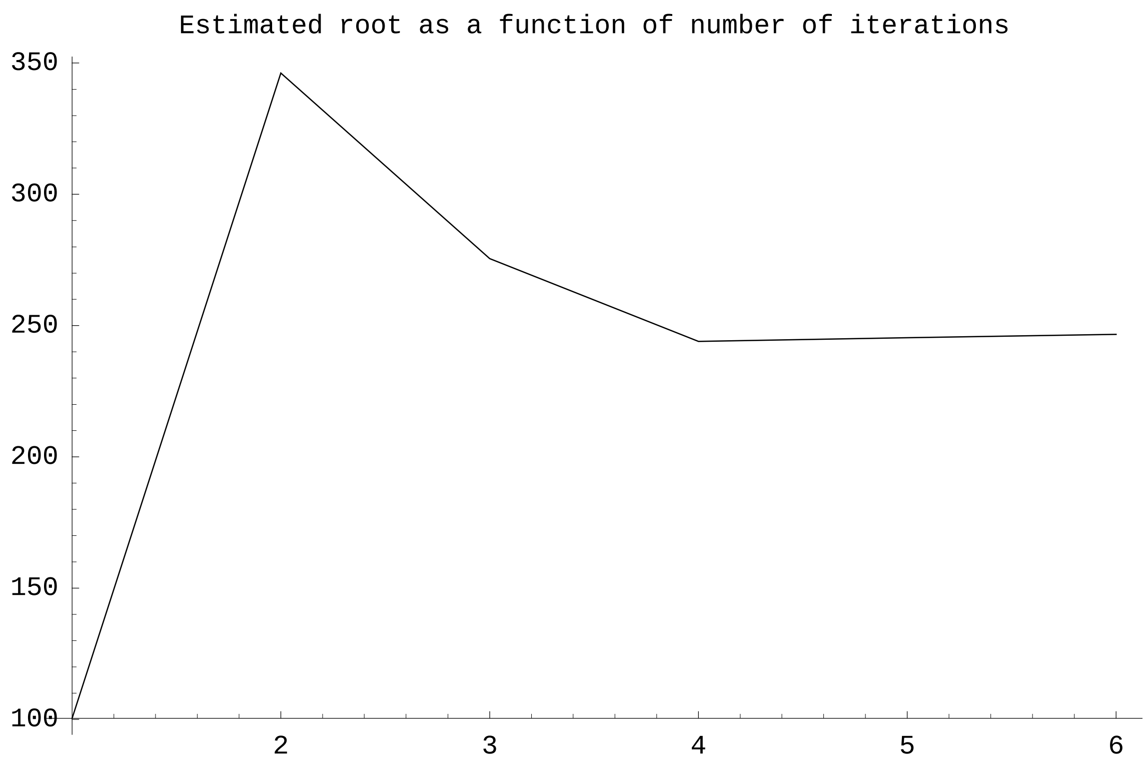
In[449]:=
```
Array[sigdig, nmaximum];
```

In[450]:=
```
For[i = 1, i <= nmaximum, i++, If[(∈a[i] ≥ 5) || (i <= 1),
   sigdig[i] = 0, sigdig[i] = Floor[(2 - Log[10, Abs[∈a[i] / 0.5]])]]]
```

■ Graphs

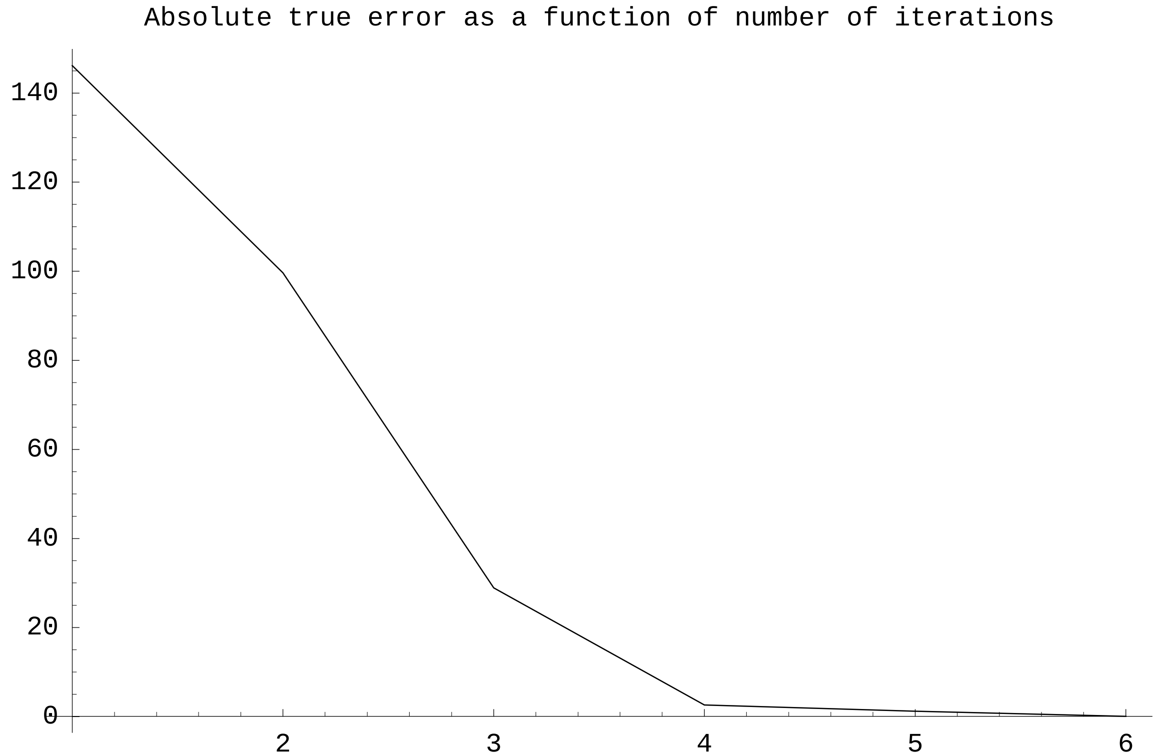
In[451]:=
```
xrplot = Table[AV[i], {i, 1, nmaximum}];
```

In[452]:=
```
ListPlot[xrplot, PlotJoined → True,
 PlotRange -> All, AxesOrigin -> {1, Min[xrplot]},
 PlotLabel -> "Estimated root as a function of number of iterations"];
```



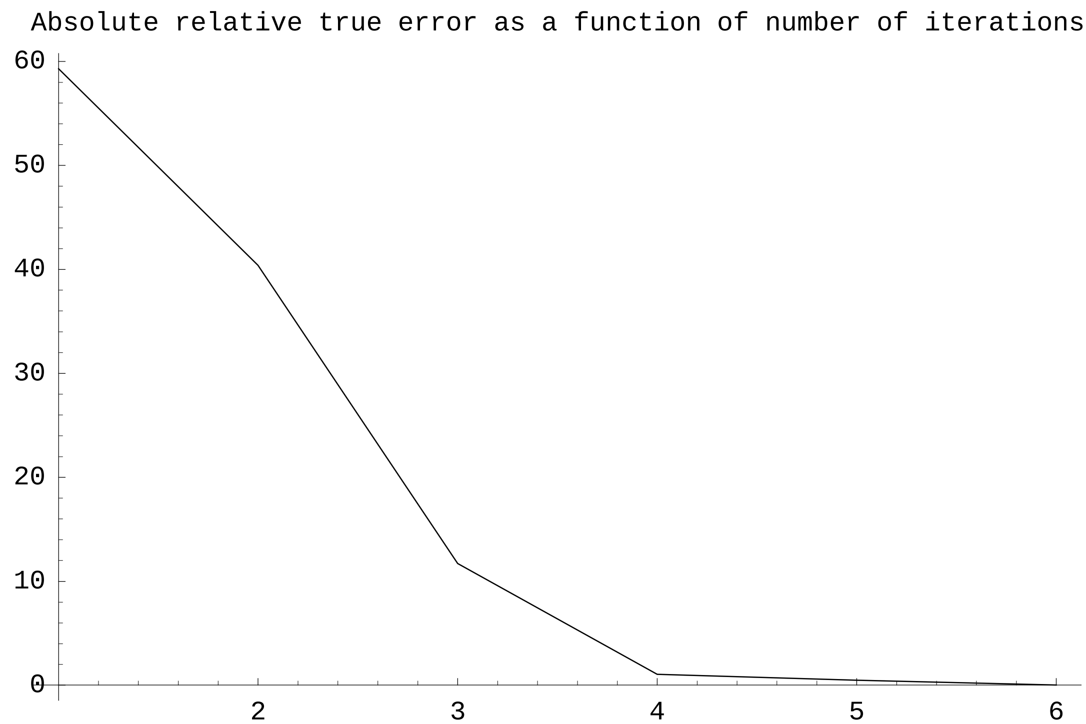
In[453]:=
```
Etplot = Table[Et[i], {i, 1, nmaximum}];
```

In[454]:=
```
ListPlot[Etplot, PlotJoined → True,
 PlotRange -> All, AxesOrigin -> {1, Min[Etplot]},
 PlotLabel -> "Absolute true error as a function of number of iterations"];
```



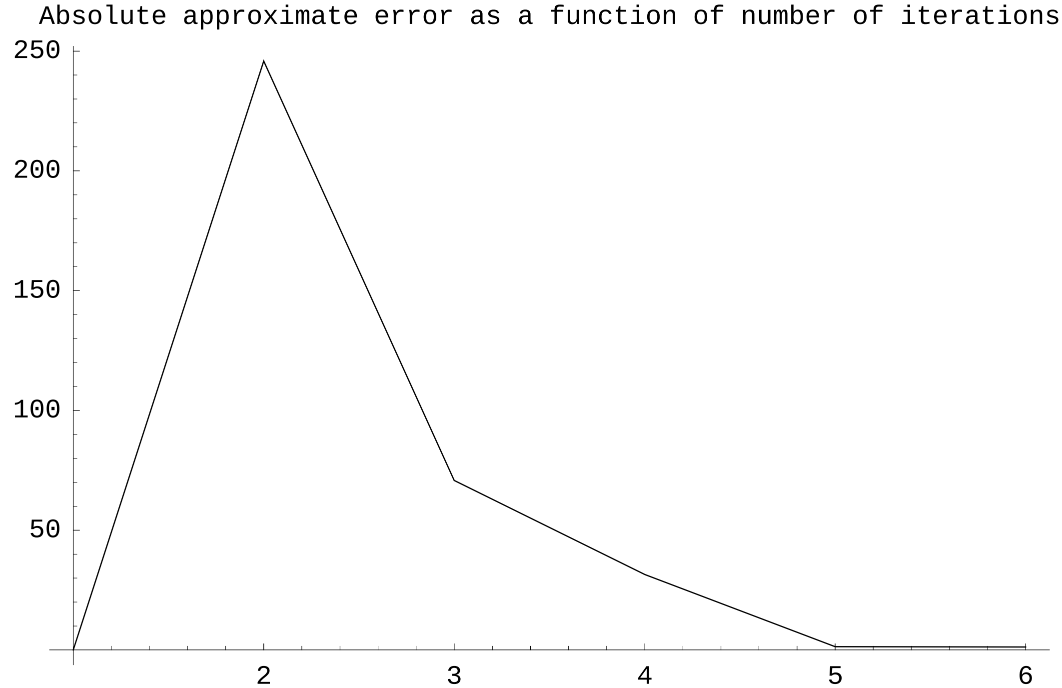
In[455]:=
```
ϵtplot = Table[ϵt[i], {i, 1, n_maximum}];
```

```
In[456]:= ListPlot[ϵtplot, PlotJoined → True,
  PlotRange -> All, AxesOrigin -> {1, Min[ϵtplot]}, PlotLabel ->
   "Absolute relative true error as a function of number of iterations"];
```




```
In[457]:= Eaplot = Table[Ea[i], {i, 1, n_maximum}];
```

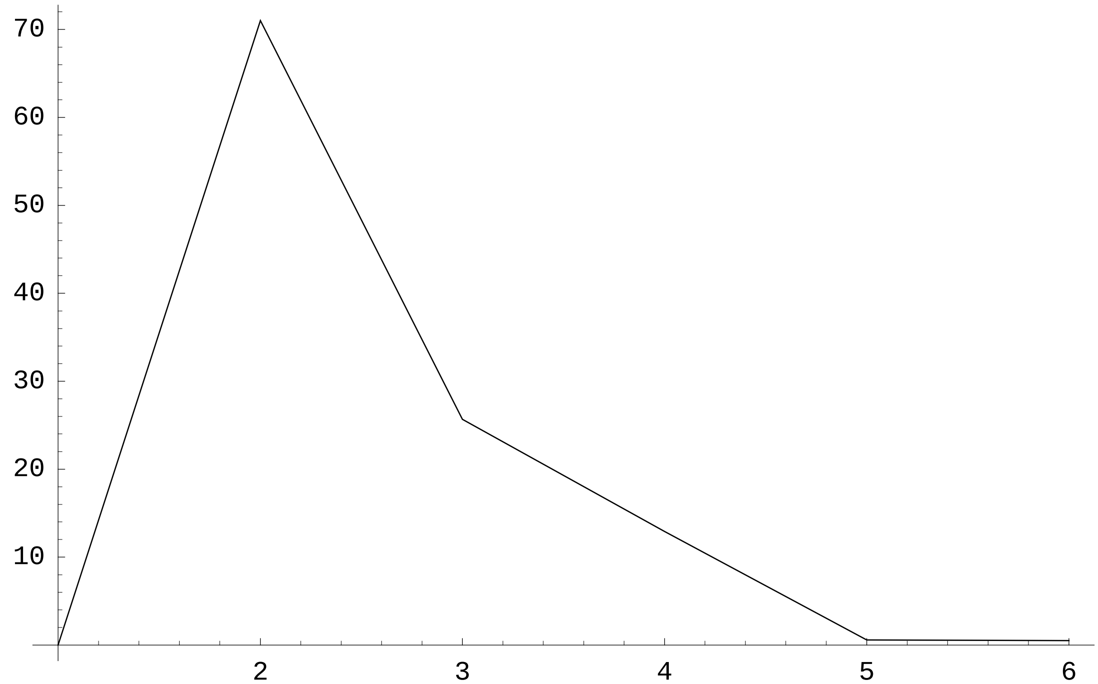
In[458]:=
```
ListPlot[Eaplot, PlotJoined → True,
 PlotRange -> All, AxesOrigin -> {1, Min[Eaplot]}, PlotLabel ->
  "Absolute approximate error as a function of number of iterations"];
```



In[459]:=
```
ϵaplot = Table[ϵa[i], {i, 1, n_maximum}];
```

In[460]:=
```
ListPlot[∈aplot, PlotJoined → True,
 PlotRange -> All, AxesOrigin -> {1, Min[∈aplot]},
 PlotLabel -> "Absolute relative approximate error
    as a function of number of iterations"];
```

Absolute relative approximate error as a function of number of iterations



In[461]:=
```
sigdigplot = Table[sigdig[i], {i, 1, n_maximum}];
```

In[462]:=
```
<< Graphics`Graphics`
```

In[463]:=
```
BarChart[sigdigplot];
```