Adaptive Quadrature

> restart;

nalib

> libname:="c:/nalib",libname;

libname := "nalib", "/Library/Frameworks/Maple.framework/Versions/15/lib",

"/Library/Frameworks/Maple.framework/Versions/15/toolbox/NAG/lib"

> with(numanal);

[SOR, SOR_dir, adaptq, adaptq_dir, bezier, bezier_dir, bisection, bisection_dir, chop, chop_dir,
clamped_spline, clamped_spline_dir, divided_diff, divided_diff_dir, extrap, extrap_dir,
falseposition, falseposition_dir, fixedpoint, fixedpoint_dir, gaussseidel, gaussseidel_dir, hermite,
hermite_dd, hermite_dd_dir, hermite_dir, horner, horner_dir, jacobi, jacobi_dir, muller,
muller_dir, natural_spline, natural_spline_dir, newton, newton_dir, romberg, romberg_dir,
secant, secant_dir, steffensen, steffensen_dir]

Page 148 #1a&2a

We enter the formula for $S$ for adaptive Simpson as a Maple function.

> S:=(f,a,b)->(b-a)/6*(f(a)+4*f((a+b)/2)+f(b));

We enter our function as a Maple function.

> f:=x->x^2*ln(x);

We compute $S(1,1.5)$ for $f$.

> S1:=S(f,1,1.5);

We compute $S(1,1.25)$ for $f$.

> S2:=S(f,1,1.25);

We compute $S(1.25,1.5)$ for $f$.

> S3:=S(f,1.25,1.5);

We compute our integral estimate.

> estimate:=S2+S3;

We compute our preliminary error estimate, which is approximately 15 times the error estimate.

> errorest:=abs(S1-(S2+S3));

We compute the approximate error estimate.

> approxerror:=errorest/15;

We compute the actual integral value.
We compute the actual error in the approximation.

```maple
> err:=int(f(x),x=1..1.5)-S2-S3;
errorr := 8.972 10^{-7}
```

We are going to use adaptive quadrature with Simpson's rule to approximate

\[
\int_{1}^{3} \frac{100 \sin \left( \frac{10}{x} \right)}{x^2} \, dx
\]

with tolerance \(10^{-4}\). We will use the procedure `adaptq` in the class library `nalib`. We first check the directions for `adaptq`.

> adaptq_dir();

The arguments for adapt are:
1) the function being integrated
2) the lower limit of integration
3) the upper limit of integration
4) tolerance
5) maximum number of levels
6) variable for returning the integral approximation

If assigning the result to a variable, have the variable and the 6th argument the same.

If S is the variable for returning the integral's value and has already been given a value, the procedure should be preceded by the statement: `S:='S'`

We enter the function.

```maple
> f:=(100/x^2)*sin(10/x);
f := \frac{100 \sin \left( \frac{10}{x} \right)}{x^2}
```

We evaluate the approximation with `adaptq`.

> S:=adaptq(f,1,3,10^(-4),100,S);

The integral of F from 1.00000000 to 3.00000000 is -1.42601481 to within 1.00000000e-04
The number of function evaluations is: 93

```maple
S := -1.426014813
```

We find the actual value of the integral.

> actual:=evalf(int(f,x=1..3));

```maple
actual := -1.426024757
```

We find the actual error.

> err:=abs(S-actual);

```maple
err := 0.000009944
```

`NumericalAnalysis`
We do the same problem using the command `AdaptiveQuadrature` in the `NumericalAnalysis` package.

```maple
> with(Student[NumericalAnalysis]);

[AdaptiveQuadrature, AddPoint, ApproximateExactUpperBound, ApproximateValue, BackSubstitution, BasisFunctions, Bisector, CubicSpline, DataPoints, Distance, DividedDifferenceTable, Draw, Euler, EulerTutor, ExactValue, FalsePosition, FixedPointIteration, ForwardSubstitution, Function, InitialValueProblem, InitialValueProblemTutor, Interpolant, InterpolantRemainderTerm, IsConvergent, IsMatrixShape, IterativeApproximate, IterativeFormula, IterativeFormulaTutor, LeadingPrincipalSubmatrix, LinearSolve, LinearSystem, MatrixConvergence, MatrixDecomposition, MatrixDecompositionTutor, ModifiedNewton, NevilleTable, Newton, NumberOfSignificantDigits, PolynomialInterpolation, Quadrature, RateOfConvergence, RelativeError, RemainderTerm, Roots, RungeKutta, Secant, SpectralRadius, Steffensen, Taylor, TaylorPolynomial, UpperBoundOfRemainderTerm, VectorLimit]
```

We find the approximation using `output=value`.

```maple
> ans := AdaptiveQuadrature(f, x=1..3, method=simpson, output=value);

ans := -1.42601481
```

Since `method=simpson` is the default, that option can be skipped. We do this with `output=information`.

```maple
> AdaptiveQuadrature(f, x=1..3, output=information);

INTEGRAL: Int(100/x^2*sin(10/x), x=1..3) = -1.42602476

APPROXIMATION METHOD: Adaptive Simpson's Rule

---------------------------------- INFORMATION TABLE ------------------
----------------
Approximate Value         Absolute Error         Relative Error
-1.42601481               9.946e-06         0.0006975 %
---------------------------------- ITERATION HISTORY ------------------
---------------------
Interval           Status         Present Stack
1..3             fail           EMPTY
1..2             fail           [2, 3]
1..3/2           fail           [[1], [3/2, 2]]
1..5/4           fail           [[2], [5/4, 3/2]]
1..9/8           fail           [[3], [9/8, 5/4]]
1..17/16          fail           [[4], [17/16, 9/8]]
1..33/32         PASS           [[5], [33/32, 17/16]]
33/32..17/16       PASS           [[4], [17/16, 9/8]]
17/16..9/8          fail           [[3], [9/8, 5/4]]
17/16..35/32       PASS           [[4], [35/32, 9/8]]
35/32..9/8          fail           [[3], [9/8, 5/4]]
9/8..5/4           fail           [[2], [5/4, 3/2]]
9/8..19/16          fail           [[3], [19/16, 5/4]]
9/8..37/32         PASS           [[4], [37/32, 19/16]]
37/32..19/16        PASS           [[3], [19/16, 5/4]]
19/16..5/4          PASS           [[2], [5/4, 3/2]]
5/4..3/2           fail           [[1], [3/2, 2]]
5/4..11/8          fail           [[2], [11/8, 3/2]]
5/4..21/16         PASS           [[3], [21/16, 11/8]]
21/16..11/8        PASS           [[2], [11/8, 3/2]]
11/8..3/2           fail           [[1], [3/2, 2]]
11/8..23/16        PASS           [[2], [23/16, 3/2]]
23/16..3/2         PASS           [[1], [3/2, 2]]
3/2..2           fail           [2, 3]
```
<table>
<thead>
<tr>
<th>Interval</th>
<th>Result</th>
<th>Error Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>3/2..7/4</td>
<td>fail</td>
<td>[[1], [7/4, 2]]</td>
</tr>
<tr>
<td>3/2..13/8</td>
<td>fail</td>
<td>[[2], [13/8, 7/4]]</td>
</tr>
<tr>
<td>3/2..25/16</td>
<td>PASS</td>
<td>[[3], [25/16, 13/8]]</td>
</tr>
<tr>
<td>25/16..13/8</td>
<td>PASS</td>
<td>[[2], [13/8, 7/4]]</td>
</tr>
<tr>
<td>13/8..7/4</td>
<td>fail</td>
<td>[[1], [7/4, 2]]</td>
</tr>
<tr>
<td>13/8..27/16</td>
<td>PASS</td>
<td>[[2], [27/16, 7/4]]</td>
</tr>
<tr>
<td>27/16..7/4</td>
<td>PASS</td>
<td>[[3], [7/4, 2]]</td>
</tr>
<tr>
<td>7/4..2</td>
<td>fail</td>
<td>[2, 3]</td>
</tr>
<tr>
<td>7/4..15/8</td>
<td>PASS</td>
<td>[[1], [15/8, 2]]</td>
</tr>
<tr>
<td>15/8..2</td>
<td>PASS</td>
<td>[2, 3]</td>
</tr>
<tr>
<td>2..2</td>
<td>fail</td>
<td>EMPTY</td>
</tr>
<tr>
<td>2..5/2</td>
<td>fail</td>
<td>[5/2, 3]</td>
</tr>
<tr>
<td>2..9/4</td>
<td>fail</td>
<td>[[1], [9/4, 5/2]]</td>
</tr>
<tr>
<td>2..17/8</td>
<td>PASS</td>
<td>[2, 17/8, 9/4]</td>
</tr>
<tr>
<td>17/8..9/4</td>
<td>PASS</td>
<td>[[1], [9/4, 5/2]]</td>
</tr>
<tr>
<td>9/4..3/2</td>
<td>fail</td>
<td>[5/2, 3]</td>
</tr>
<tr>
<td>9/4..19/8</td>
<td>PASS</td>
<td>[[1], [19/8, 5/2]]</td>
</tr>
<tr>
<td>19/8..5/2</td>
<td>PASS</td>
<td>[5/2, 3]</td>
</tr>
<tr>
<td>5/2..3</td>
<td>fail</td>
<td>EMPTY</td>
</tr>
<tr>
<td>5/2..11/4</td>
<td>PASS</td>
<td>[11/4, 3]</td>
</tr>
<tr>
<td>11/4..3</td>
<td>PASS</td>
<td>EMPTY</td>
</tr>
</tbody>
</table>

Number of Function Evaluations: 93

Compare the information above to the use of `output=plot`.

> `AdaptiveQuadrature(f, x=1..3, output=plot);`
An Approximation of the Integral of

\[ f(x) = \frac{100 \sin \left( \frac{10}{x} \right)}{x^2} \]

on the Interval [1, 3]

Using Adaptive Simpson's rule

Integral Value: -1.426024756
Approximation: -1.426014810

The default tolerance here is $10^{-4}$. To get a different tolerance, say $10^{-6}$, we use `method=[simpson,10^(-6)]` instead of `method=simpson`.

> `betterans := AdaptiveQuadrature(f, x=1..3, method=[simpson,10^(-6)], output=value);`

```
betterans := -1.426024678
```