Lab X: Romberg integration

Due on Monday March 21th, noon.
By email to beerli@fsu.edu [use Subject: ISC5315: lab X]

1. Familiarize yourself with the Richard's extrapolation and Romberg integration, use the the material on the next page
2. Write a program to take any polynomial up to degree=7 and a given accuracy (for example up to 8 digits) calculate the integral of that polynomial using Romberg integration with the trapezoidal rule ($R(n, 0)$) and Simpson's rule ($R(n, 1)$). Use either C++ or Java. Assume that the user either gives the polynomial by hand or then has several polynomials in a file (one per line). Ideally your program will work with and without command-line options.
3. Your program should print out the original formula, the accuracy, the integration result.
4. Allow for an option in your program so that a user can see the convergence process by showing the results and the improvement.
5. Write a report that shows how you implemented the program, in particular I will look for section that uses pseudocode to explain the structure of your program.
6. Send report (PDF file) and the program as tar.gz archive.

Romberg Integration

- Trapezoid formula
  - with n subintervals gives error ~ $1/n^2$
  - I repeat the calculation with $2n$ subintervals
  - Can I combine to get a better estimate?
- Error in trapezoidal

$$E = \frac{(b-a)^3}{12n^2} \underbrace{\max_{x\in[a,b]} f''(x)}$$

if I assume that this
is independent of n

$$\boxed{E = \frac{C}{n^2}}$$

Romberg Integration

- If we assume that "C" is roughly constant
  - I_{true} = true value of the integral
  - I_n = numerical approximation using "n" steps
- We have

$$I_{true} = I_n + E(n)$$

$$\begin{aligned} E(n) &\approx \frac{C}{n^2} &= I_{true} - I_n \\ E(2n) &\approx \frac{C}{4n^2} &= I_{true} - I_{2n} \end{aligned}$$

eliminate C/n^2

$$\boxed{I_{true,est} = I_{2n} + \frac{I_{2n} - I_n}{3}}$$

Romberg Integration

- Example
  - vertical distance covered by a rocket between 8 and 30 s.

$$x = \int_8^{30} \left[2000 \ln\left(\frac{140000}{140000 - 2100t}\right) - 9.8t\right] dt$$

Exact Answer:
11061 m

n	Value	E_t	RelErr
1	11868	807	7.296
2	11266	205	1.854
3	11153	91.4	0.8265
4	11113	51.5	0.4655
5	11094	33.0	0.2981
6	11084	22.9	0.2070
7	11078	16.8	0.1521
8	11074	12.9	0.1165

Multistep trapezoidal rule

Romberg Integration

- Use Richardson extrapolation

$$I_2 = 11266$$
$$I_4 = 11113$$
$$I_{true,est} = I_4 + \frac{I_4 - I_2}{3} = 11062$$

- If you already have done I_4, then very little extra work
  - I_2 uses a subset of the same function evaluations

Romberg Integration

- Usually much better estimates

n	Trapezoidal Rule	e_t for Trapezoidal Rule	Richardson's Extrapolation	e_t for Richardson's Extrapolation
1	11868	7.296	--	--
2	11266	1.854	11065	0.03616
4	11113	0.4655	11062	0.009041
8	11074	0.1165	11061	0.0000

Error Estimate

- It can be shown that

$$E = \frac{C_1}{n^2} + \frac{C_2}{n^4} + \frac{C_3}{n^6} + \cdots$$

- For the scheme

$$I_{true,est} = I_{2n} + \frac{I_{2n} - I_n}{3}$$

- The leading error is O(*1/n*^{4})
  - like Simpson's
  - In Numerical Recipes, this is how Simpson's is implemented

Successive Refinement

- General Expression for Romberg Integration

$$I_{2n}^{(k)} = \frac{4^k I_{2n}^{(k-1)} - I_n^{(k-1)}}{4^{k-1} - 1} \qquad k \geq 2$$

  - index k represents order of extrapolation
  - $I_n^{(1)}$ represents regular trapezoidal with *n* segments
  - k = 2 represents values using the true estimate as O(h^2)
  - $I_n^{(k)}$ has an error of order $1/n^{2k}$

Successive Refinement

- In our particular example

		First Order (k=2)	Second Order (k=3)	Third Order (k=4)
1-segment	11868			
		11065		
2-segment	11126		11062	
		11062		11061
4-segment	11113		11061	
		11061		
8-segment	11074			

Romberg's method

From Wikipedia, the free encyclopedia

In numerical analysis, **Romberg's Method** (Romberg 1955) generates a triangular array consisting of numerical estimates of the definite integral

$$\int_a^b f(x)\,dx$$

by applying Richardson extrapolation (Richardson 1910) repeatedly on the trapezium rule or the rectangle rule (midpoint rule). Romberg's method is a Newton–Cotes formula -- it evaluates the integrand at equally-spaced points. The integrand must have continuous derivatives, though fairly good results may be obtained if only a few derivatives exist. If it is possible to evaluate the integrand at unequally-spaced points, then other methods such as Gaussian quadrature and Clenshaw–Curtis quadrature are generally more accurate.

Contents

Method

The method can be defined inductively

$$R(0,0) = \frac{1}{2}(b-a)(f(a)+f(b))$$

$$R(n,0) = \frac{1}{2}R(n-1,0) + h_n \sum_{k=1}^{2^{n-1}} f(a+(2k-1)h_n)$$

$$R(n,m) = R(n,m-1) + \frac{1}{4^m-1}(R(n,m-1) - R(n-1,m-1))$$

or

$$R(n,m) = \frac{1}{4^m-1}(4^m R(n,m-1) - R(n-1,m-1))$$

where

$$n \geq 1$$

$$m \geq 1$$

$$h_n = \frac{b-a}{2^n}.$$

In big O notation, the error for $R(n, m)$ is (Mysovskikh 2002):

$$O\left(h_n^{2m+2}\right).$$

The zeroeth extrapolation, $R(n, 0)$, is equivalent to the trapezoidal rule with $2^n + 1$ points; the first extrapolation, $R(n, 1)$, is equivalent to Simpson's rule with $2^n + 1$ points. The second extrapolation, $R(n, 2)$, is equivalent to Boole's rule with $2^n + 1$ points. Further extrapolations differ from Newton Cote's Formulas. In particular further Romberg extrapolations expand on Boole's rule in very slight ways, modifying weights into ratios similar as in Boole's rule. In contrast, further Newton Cotes methods produce increasingly differing weights, eventually leading to large positive and negative weights. This is indicative of how large degree interpolating polynomial Newton Cotes methods fail to converge for many integrals, while Romberg integration is more stable.

When function evaluations are expensive, it may be preferable to replace the polynomial interpolation of Richardson with the rational interpolation proposed by Bulirsch & Stoer (1967).

Example

As an example, the Gaussian function is integrated from 0 to 1, i.e. the error function $\mathrm{erf}(1) \approx 0.842700792949715$. The triangular array is calculated row by row and calculation is terminated if the two last entries in the last row differ less than 10^{-8}.

```
0.77174333
0.82526296  0.84310283
0.83836778  0.84273605  0.84271160
0.84161922  0.84270304  0.84270083  0.84270066
0.84243051  0.84270093  0.84270079  0.84270079  0.84270079
```

The result in the lower right corner of the triangular array is accurate to the digits shown. It is remarkable that this result is derived from the less accurate approximations obtained by the trapezium rule in the first column of the triangular array.

Implementation

Here is an example of a computer implementation of the Romberg method (In the C programming language), it needs one vector and one variable; it also needs a sub-routine trapeze:

```
  #include <stdio.h>
  #include <stdlib.h>
  #include <math.h>
  #define MAX 6

 int main()
 {
     double s[MAX];
     int i,k;
```

```
    double var ;
    for (i = 1; i< MAX; i++)
        s[i] = 1;

    for (k=1; k< MAX; k++)
    {
        for (i=1; i <=k; i++)
        {
            if (i==1)
            {
                var = s[i];
                s[i] = Trap(0, 1, pow(2, k-1));     // sub-routine tra
            }                                       // integrated from
                                                    /* pow() is the nu
            else
            {
                s[k]= ( pow(4 , i-1)*s[i-1]-var )/(pow(4, i-1) - 1);

                var = s[i];
                s[i]= s[k];
            }
         }
    for (i=1; i <=k; i++)
        printf ("  %f  ", s[i]);
        printf ("\n");
    }

    return 0;
}
```

References

- Richardson, L. F. (1911), "The Approximate Arithmetical Solution by Finite Differences of Physical Problems Involving Differential Equations, with an Application to the Stresses in a Masonry Dam" (http://links.jstor.org/sici?sici=0264-3952(1911)210%3C307%3ATAASBF%3E2.0.CO%3B2-J) , *Philosophical Transactions of the Royal Society of London. Series A* **210**: 307–357, doi:10.1098/rsta.1911.0009 (http://dx.doi.org/10.1098%2Frsta.1911.0009) , http://links.jstor.org/sici?sici=0264-3952(1911)210%3C307%3ATAASBF%3E2.0.CO%3B2-J
- Romberg, W. (1955), "Vereinfachte numerische Integration", *Det Kongelige Norske Videnskabers Selskab Forhandlinger* (Trondheim) **28** (7): 30–36
- Thacher, Jr., Henry C. (July 1964), "Remark on Algorithm 60: Romberg integration" (http://portal.acm.org/citation.cfm?id=364520.364542) , *Communications of the ACM* **7** (7): 420–421, doi:10.1145/364520.364542 (http://dx.doi.org/10.1145%2F364520.364542) , http://portal.acm.org/citation.cfm?id=364520.364542
- Bauer, F.L.; Rutishauser, H.; Stiefel, E. (1963), Metropolis, N. C., et al., ed., "New aspects in numerical quadrature", *Experimental Arithmetic, high-speed computing and mathematics, Proceedings of Symposia in Applied Mathematics* (AMS) (15): 199–218
- Bulirsch, Roland; Stoer, Josef (1967), "Handbook Series Numerical Integration. Numerical quadrature by extrapolation" (http://www-gdz.sub.uni-goettingen.de/cgi-bin/digbib.cgi?PPN362160546_0009) ,

Numerische Mathematik **9**: 271–278, http://www-gdz.sub.uni-goettingen.de/cgi-bin/digbib.cgi?PPN362160546_0009
- Mysovskikh, I.P. (2002), "Romberg method" (http://eom.springer.de/r/r082570.htm) , in Hazewinkel, Michiel, *Encyclopaedia of Mathematics*, Springer-Verlag, ISBN 1-4020-0609-8, http://eom.springer.de/r/r082570.htm

External links

- ROMBINT (http://www.mathworks.com/matlabcentral/fileexchange/loadFile.do?objectId=34&objectType=file) -- code for MATLAB (author: Martin Kacenak)
- Module for Romberg Integration (http://math.fullerton.edu/mathews/n2003/RombergMod.html)
- Free online integration tool using Romberg, Fox-Romberg, Gauss-Legendre and other numerical methods (http://www.hvks.com/Numerical/webintegration.html)