Set 15: Adaptive Quadrature

Kyle A. Gallivan

Department of Mathematics

Florida State University

Foundations of Computational Math 2
Spring 2013

Overview

- Extrapolation
  - Combine two different methods.
  - Use error estimates as if they were exact.
  - Generate new method with better error properties.
- Step halving
  - Use same composite method with two different stepsizes.
  - Exploit earlier work if possible in an incremental update approach.
  - Combine with error estimates to to control number of halvings.
- Adaptive Quadrature
  - Use error estimates to update stepsize, i.e., not only halving.
  - Selective subinterval refinement.

Extrapolation

Consider the numerical approximation of a definite integral via Simpson's first and second rule.

We have

$$
\begin{aligned}
I^* &= I_2 + E_2 = I_2 - \frac{1}{90} h_2^5 f^{(4)}(\eta_2) \\
I^* &= I_3 + E_3 = I_3 - \frac{3}{80} h_3^5 f^{(4)}(\eta_3)
\end{aligned} \tag{1}
$$

$$
\frac{E_2}{E_3} = \frac{\frac{(b-a)^5}{2880} f^{(4)}(\eta_2)}{\frac{(b-a)^5}{6480} f^{(4)}(\eta_3)}
$$

$$
\eta_2 \approx \eta_3 \rightarrow E_2 \approx \frac{9}{4} E_3 \tag{2}
$$

Extrapolation

Assume equality in (2), define $\hat{I}^*$ via equality in (1), and solve for $\hat{I}^*$:

$$\hat{E}_2 = \frac{9}{4}\hat{E}_3$$

$$\hat{I}^* = I_2 + \hat{E}_2 \rightarrow \hat{I}^* = I_2 + \frac{9}{4}\hat{E}_3$$

$$\hat{I}^* = I_3 + \hat{E}_3$$

$$-\frac{4}{9}\hat{I}^* = -\frac{4}{9}I_2 - \hat{E}_3$$

$$\hat{I}^* = I_3 + \hat{E}_3$$

Extrapolation

$$\left.\begin{array}{c} -\frac{4}{9}\hat{I}^* = -\frac{4}{9}I_2 - \hat{E}_3 \\ \hat{I}^* = I_3 + \hat{E}_3 \end{array}\right. \to \hat{I}^* = \frac{9}{5}I_3 - \frac{4}{5}I_2 \tag{3}$$

We have a new method $\hat{I}^* \approx I^*$,

$$I_{3,2}(f) = \frac{9}{5}I_3 - \frac{4}{5}I_2 = \sum_j \alpha_j f_j$$

This combination idea is behind many extrapolation methods.

Alternate View of Extrapolation

Attempt to estimate E_3 with $\hat{E}_3$ and define new method

$$I_{3,2}(f) = I_3 + \hat{E}_3$$

Recall the approximations we treat as equalities,

$$\hat{E}_2 = \frac{9}{4}\hat{E}_3$$

$$\hat{I}^* = I_2 + \frac{9}{4}\hat{E}_3$$

$$\hat{I}^* = I_3 + \hat{E}_3$$

$$0 = I_3 - I_2 + \hat{E}_3 - \frac{9}{4}\hat{E}_3 \rightarrow \hat{E}_3 = \frac{4}{5}(I_3 - I_2)$$

Alternate View of Extrapolation

This yields

$$I_{3,2}(f) = I_3 + \hat{E}_3 = I_3 + \frac{4}{5}(I_3 - I_2)$$

Easy to see that as before,

$$I_{3,2}(f) = \frac{9}{5} I_3 - \frac{4}{5} I_2$$

However, this yields a different implementation strategy, i.e., $I_{3,2}(f)$ is not written as independent method, but as a correction to I_3 given I_2. Hence the complexity of the combined evaluation of the methods is crucial to this kind of approach.

Example

Combine midpoint and trapezoidal rules:

$$I_0 = (b-a)f(\frac{a+b}{2}), \quad E_0 = \frac{(b-a)^3}{24} f''(\eta_0)$$

$$I_1 = \frac{(b-a)}{2}\left[f(a) + f(b)\right], \quad E_1 = -\frac{(b-a)^3}{12} f''(\eta_1)$$

$$\begin{array}{c} \hat{E}_1 = -2\hat{E}_0 \\ \hat{I}^* = I_1 + \hat{E}_1 = I_0 + \hat{E}_0 \end{array} \rightarrow \begin{array}{c} \hat{I}^* = I_1 - 2\hat{E}_0 \\ 2\hat{I}^* = 2I_0 + 2\hat{E}_0 \end{array}$$

$$\hat{I}^* = \frac{1}{3} I_1 + \frac{2}{3} I_0$$

Extrapolation

$$I^* \approx \hat{I}^* = \frac{1}{3} I_1 + \frac{2}{3} I_0$$

$$\begin{aligned}
\hat{I}^* &= \frac{(b-a)}{6}\left[f(a) + f(b)\right] + \frac{4(b-a)}{6} f(\frac{a+b}{2}) \\
&= \frac{(b-a)}{2}\left[\frac{1}{3} f(a) + \frac{4}{3} f(\frac{a+b}{2}) + \frac{1}{3} f(b)\right] \\
&= \frac{h}{3}\left[f(a) + 4 f(\frac{a+b}{2}) + f(b)\right]
\end{aligned}$$

$$\text{where } h = \frac{(b-a)}{2}$$

yields Simpson's rule.

Alternate View of Extrapolation

$$\hat{E}_0 = -\frac{1}{2}\hat{E}_1$$

$$I^* = I_0 - \frac{1}{2}\hat{E}_1$$

$$I^* = I_1 + \hat{E}_1$$

$$0 = I_1 - I_0 + \frac{3}{2}\hat{E}_1$$

$$\hat{E}_1 = \frac{2}{3}(I_1 - I_0)$$

$$I_{1,0} = I_1 - \frac{2}{3}(I_1 - I_0)$$

which is Simpson's rule.

Coarse and Fine Grid Errors

Suppose we have a composite quadrature formula with infinitesimal order r and degree of exactness m,

$$E = Kh^r f^{(m+1)}(\eta)$$

where dependence of K on $L = b - a$ is implied.

Choose a stepsize h and consider using it and $h/2$ for a coarse and a fine grid. We have

$$E_c = Kh^r f^{(m+1)}(\eta)$$

$$E_f = K\left(\frac{h}{2}\right)^r f^{(m+1)}(\mu)$$

Coarse and Fine Grid Errors

Assuming that η vs μ does not affect the value of $f^{(m+1)}(x)$ significantly, we have

$$E_c = Ch^r$$

$$E_f \approx \hat{E}_f = C\left(\frac{h}{2}\right)^r = \frac{1}{2^r}E_c$$

e.g., composite trapezoidal rule $\quad E_{ct} = -\dfrac{(b-a)}{12}f^{(2)}(\eta)h^2 = Ch^2$

Note the equality for E_c, hence, C is not known and must be eliminated.

Error Estimation for Step Halving

$$E_c = Ch^r$$

$$\hat{E}_f = C\left(\frac{h}{2}\right)^r = \frac{1}{2^r}E_c$$

$$I^* = I_c + E_c = I_f + \hat{E}_f \quad \text{assumed}$$

$$0 = (I_c - I_f) + (E_c - \hat{E}_f)$$

Choice of two subsitutions $\longrightarrow$ estimate E_c or E_f.

Error Estimation for Step Halving

$$0 = (I_c - I_f) + E_c(1 - \frac{1}{2^r})$$

$$E_c(1 - \frac{1}{2^r}) = (I_f - I_c)$$

$$E_c = \frac{2^r}{(2^r - 1)}(I_f - I_c)$$

$$0 = (I_c - I_f) + \hat{E}_f(2^r - 1)$$

$$\hat{E}_f = \frac{1}{(2^r - 1)}(I_f - I_c)$$

Error Estimation for Step Halving

We compute $(I_f - I_c)$ and we estimate

$$E_c = \frac{2^r}{(2^r - 1)}(I_f - I_c)$$

$$E_f \approx \hat{E}_f = \frac{1}{(2^r - 1)}(I_f - I_c)$$

Step Halving

Let $n = 8$, $h_8 = (b-a)/8$ and consider the composite Trapezoidal rule

$$I_8 = \frac{h_8}{2}\left[f_0 + f_8 + 2(f_1 + \cdots + f_7)\right]$$

Let $h_1 = (b-a)$ and $h_{2i} = h_i/2$. We have

$$I_1 = \frac{h_1}{2}\left[f_0 + f_8\right], \quad I_2 = \frac{h_2}{2}\left[f_0 + f_4\right] + \frac{h_2}{2}\left[f_4 + f_8\right]$$

$$I_4 = \frac{h_4}{2}\left[f_0 + f_2\right] + \frac{h_4}{2}\left[f_2 + f_4\right] + \frac{h_4}{2}\left[f_4 + f_6\right] + \frac{h_4}{2}\left[f_6 + f_8\right]$$

$$I_8 = \frac{h_8}{2}\left[f_0 + f_1\right] + \frac{h_8}{2}\left[f_1 + f_2\right] + \frac{h_8}{2}\left[f_2 + f_3\right] + \frac{h_8}{2}\left[f_3 + f_4\right]$$

$$+\frac{h_8}{2}\left[f_4 + f_5\right] + \frac{h_8}{2}\left[f_5 + f_6\right] + \frac{h_8}{2}\left[f_6 + f_7\right] + \frac{h_8}{2}\left[f_7 + f_8\right]$$

Step Halving

$$I_2 = \frac{h_2}{2}\left[f_0 + f_8\right] + \frac{h_2}{2}\left[2f_4\right]$$

$$= \frac{1}{2}I_1 + h_2\left[f_4\right] = \frac{1}{2}\left[I_1 + h_1(f_4)\right]$$

$$I_4 = \frac{1}{2}\left[I_2 + h_2(f_2 + f_6)\right]$$

$$I_8 = \frac{1}{2}\left[I_4 + h_4(f_1 + f_3 + f_5 + f_7)\right]$$

This yields the general refinement recurrence:

$$I_{2n} = \frac{1}{2}\left[I_n + h_n(\text{sum of new } f_i)\right]$$

Example

Approximate via composite Trapezoidal rule

$$\int_0^1 e^x dx = e - 1 = 1.71828182\ldots$$

$$E_{ct} = -(b-a)\frac{h^2}{12} f^{(2)}(\eta) = Ch^2, \quad h = \frac{(b-a)}{n}, \quad r = 2$$

n	I_n	$I - I_n$	error estimate
1	1.85914	-0.140859	
2	1.75393	-0.0356493	-0.0350699
4	1.72722	-0.00894008	-0.00890306
8	1.72052	-0.00223676	-0.00223444
16	1.71884 6	-0.00055930	-0.000559155

Algorithm Sketch

Compute I_c using $\Delta x = h$

while error unacceptable :

    Compute I_f using $\Delta x = h/2$

    $D_h = I_f - I_c; \;\; E_f = D_h/(2^r - 1)$

    if $|E_f| \leq ABSTOL + |I_f| \times RELTOL$

        Error acceptable

        $I = I_f + E_f$

    else

        Error unacceptable

        $h \leftarrow h/2; \;\; I_c \leftarrow I_f$

    end if

end while

return I

Efficiency Improvement

- Given h, split the interval into $[a, c]$ and $[c, b]$ where, e.g., $c = (a + b)/2$
- I_c using $\Delta x = h$ available as $I_c = I_L + I_R$
- compute I_L^f and I_R^f each with n points, i.e., using $\Delta x = h/2$
- $I_f = I_L^f + I_R^f$ can be used to estimate error E_c or E_f
- if acceptable return
- estimate error on left with I_L and I_L^f
- estimate error on right with I_R and I_R^f
- refine one or both with error greater than $\tau/2$

Recursive Adaptive Quadrature

Q(f, a, b, n) is a nonadaptive composite NC formula.

ADAPT(f,a,b,n,τ)

$I_c =$Q(f, a, b, n)

$I_f =$Q$(f, a, (a+b)/2, n)+$ Q$(f, (a+b)/2, b, n)$

$E_f = (I_f - I_c)/(2^r - 1)$

if $|E_f| > \tau$ then

$I_L =$ADAPT$(f, a, (a+b)/2, n, \tau/2)$

$I_R =$ADAPT$(f, (a+b)/2, b, n, \tau/2)$

return $I_L + I_R$

else

return I_f

end if

end

Comments

- need not use midpoint but it is convenient
- need not reevaluate $f(x_i)$ if Newton-Cotes formula used
- split of error into half on each side is very simple
- more information might allow different splitting
- easily specified as a recursive algorithm
- step halving can be replaced by stepsize selection based on error estimate
  - use E_c and/or E_f with h to determine new step size $\tilde{h}$
  - $I_c(\tilde{h})$ must be computed since it is no longer the same as $I_f(h/2)$
  - $I_f(\tilde{h}/2)$ can exploit possible resuse of information from $I_c(\tilde{h})$

General Stepsize Selection

- An error estimate can be used to determine a new stepsize $\tilde{h}$ from h and a tolerance τ.
- Assume we have an estimate E associated with using stepsize h. (This can come from step halving or any other esitmation technique.)
- We have the refinement:

$$E = Ch^r \rightarrow C = \frac{E}{h^r}$$

$$C\tilde{h}^r \leq \tau \rightarrow \tilde{h} \leq \left(\frac{\tau}{C}\right)^{1/r}$$

- Use $\tilde{h}$ to determine the required number of points $\hat{n}$ that yields a uniform $\hat{h}$ given $[a, b]$.
- After evaluating quadrature rule with $\tilde{h}$ the error must be checked again to verify success or need for more refinement.