

Lecture 11: Even more quadrature

Outline

- 1) Quick Review of single-interval quadrature
- 2) Extended Newton-Cotes formulas
 - A) Extended Trapezoidal Rule
 - B) Extended Simpson's Rule
 - C) Higher order rules
- 3) Toward's high accuracy: Two approaches
 - A) Adaptive Quadrature: matlab's quad, quadl (and quadtx)
 - B) Romberg Integration and Richardson's extrapolation
- 4) Numerical differentiation (?)

Review: Basic Newton Cotes Formulas

In general, all quadrature schemes can be written as

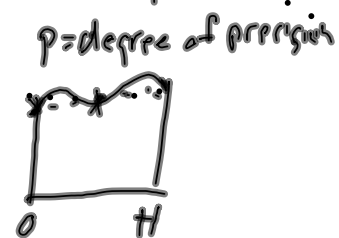
$$I[f] = \int_a^b f(x) dx \sim H \sum_{i=1}^N w_i f(x_i) + O(f^{(p+1)} H^{(p+1)})$$

$H = b - a$

Basic schemes: *Closed Newton Cotes*

Mid-Point: $M = H f\left(\frac{H}{2}\right) + O(f'' H^3)$

$p=1$



Trapezoidal: $T = H \left(\frac{f(0)}{2} + \frac{f(H)}{2} \right) + O(f'' H^3)$ $p=1$

Simpson's Rule: $S = H \left[\frac{f(0)}{6} + \frac{4f\left(\frac{H}{2}\right)}{6} + \frac{f(H)}{6} \right] + O(f^{(4)} H^5)$

$S = \frac{2}{3}M + \frac{1}{3}T$ $p=3$

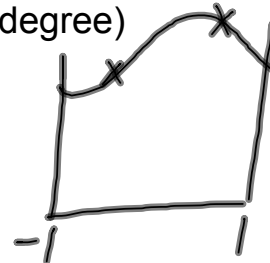
Review: Gauss-Legendre Quadrature

(choose both x_i and w_i to maximize polynomial degree)

2-Point rule: $I = \sum w_i f(x_i)$

$$P_2 = (3x^2 - 1) \quad x_i = \left[-\frac{1}{\sqrt{3}} \quad \frac{1}{\sqrt{3}}\right] \quad w_i = [1 \quad 1]$$

$$P_k \quad P_{2k-1} \quad Q_2 = f\left(-\frac{1}{\sqrt{3}}\right) + f\left(\frac{1}{\sqrt{3}}\right) + O(f^{(4)} H^4)$$



3-Point rule: x_i are roots of $P_3(x) = x/2 (5x^2 - 3)$, $w_i = [5 \ 8 \ 5]/9$

$$x_i = \left[-\sqrt{\frac{3}{5}} \quad 0 \quad \sqrt{\frac{3}{5}}\right]$$

$$P = 5$$

Affine Transformation of the Interval [a,b]

a	x	b	$\frac{(x-a)}{b-a} = \frac{(t+1)}{2} \quad x(t) = a + \frac{H}{2}(t+1)$
-1	t	1	
0	s	1	

$$\int_a^b f(x) dx = \frac{H}{2} \int_{-1}^1 f(x(t)) dt$$

$$I = \frac{H}{2} \sum w_i f(x(t_i))$$

Note: for Gauss-Legendre quadrature...a bit of matlab shorthand

given, a,b,f(x) and assume e.g. 3 point GL

$$H = b - a;$$

$$x = @(t) a + H/2*(t+1)$$

$$ti = [-1 \ 0 \ 1]*\text{sqrt}(3/5);$$

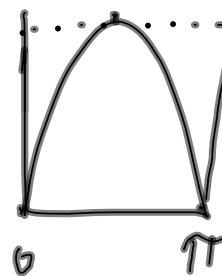
$$wi = [5 \ 8 \ 5]/9;$$

$$Q = H/2*wi*f(x(ti));$$

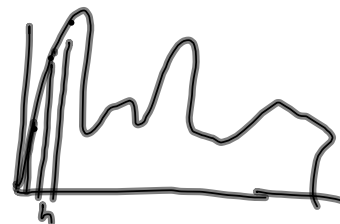
Review Example

Let $I = \int \sin(x) dx$ for x in $[0, \pi]$ (exact $I=2$)

Quadrature Rule	Q	relative error r
Mid Point	π	0.57
Trapezoidal	0	1.00
Simpsons	$2\pi/3=2.0944$	0.047
2-Point GL	1.9358	0.032
3-Point GL	2.0014	0.0007



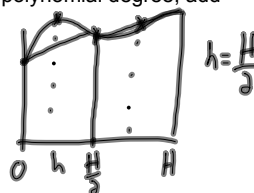
p refinement
h refinement



h-Refinement and Extended Newton-Cotes formulas

Big Idea: rather than add points to increase polynomial degree, add points to reduce "grid-spacing" h .

Example: Extended Trapezoidal rule



$$T_1 = H \left[\frac{f(0)}{2} + \frac{f(H)}{2} \right] + O(f'' H^3)$$

$$T_2 = \frac{H}{2} \left[\frac{f(0)}{2} + \frac{f(H/2)}{2} \right] + O(f'' (H/2)^3) + \frac{H}{2} \left[\frac{f(H/2)}{2} + \frac{f(H)}{2} \right] + O(f'' (H/2)^3)$$

$$\frac{H}{h} \left[\frac{f(0)}{2} + \frac{f(H/2)}{2} + \frac{f(H)}{2} \right] + O\left(2 \left(\frac{H}{2}\right)^3\right) = O(Hh^2)$$

Extended Trapezoidal rule

$$T_N = h \left[\frac{f_0}{2} + \frac{f_N}{2} + \sum_{i=1}^{N-1} f(x_i) \right] + O(h^2)$$

trapz(x, f(x))

progressive T_1, T_2, T_4, T_8

$$T_{2N} = \frac{T_N}{2} + \sum_{i=1}^N f(x_i)$$

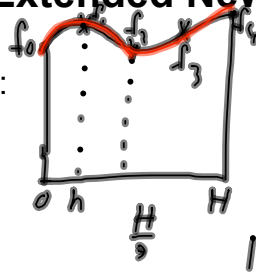
x_i = midpoint (a_i, b_i, h)

h-Refinement and Extended Newton-Cotes formulas

Extended Simpson's Rule:

$$S_1 = H \left[\frac{f_0}{6} + \frac{4}{6} f_m + \frac{1}{6} f_n \right]$$

$$= h \left[\frac{f_0}{3} + \frac{4}{3} f_m + \frac{1}{3} f_n \right]$$



$$S_2 = h \left[\frac{f_0}{3} + \frac{4}{3} f_1 + \frac{f_2}{3} \right]$$

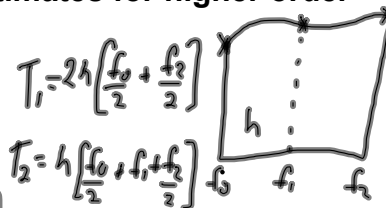
$$h = \frac{H}{4} \Rightarrow h \left[\frac{f_2}{3} + \frac{4}{3} f_3 + \frac{f_4}{3} \right]$$

$$= h \left[\frac{f_0}{3} + \frac{4}{3} \frac{2}{3} \frac{4}{3} \frac{1}{3} \right] + O(h^4)$$

Combining extended estimates for higher order accuracy

Example: T_1, T_2 and S_1

$$S_1 = h \left[\frac{1}{3} f_0 + \frac{4}{3} f_1 + \frac{1}{3} f_2 \right]$$



$$T_1 = 2h \left[\frac{f_0}{2} + \frac{f_2}{2} \right]$$

$$T_2 = h \left[\frac{f_0}{2} + f_1 + \frac{f_2}{2} \right]$$

$$S_1 = \frac{4}{3} T_2 - \frac{1}{3} T_1 \quad S_N = \frac{4}{3} T_{2N} - \frac{1}{3} T_N = T_{2N} + \frac{1}{3} (T_{2N} - T_N)$$

$$T_1$$

$$T_2$$

$$T_4$$

$$I = T_1 + E(h^2) + O(h^4) + O(h^6)$$

$$I = T_2 + \frac{E}{4} + O(h^4)$$

$$I = T_N + E(h^2) + O(h^4)$$

$$I = T_{2N} + \frac{E}{4} + O(h^4)$$

$$4I = 4T_{2N} + E + O(h^4)$$

$$I = T_N + E + O(h^4)$$

$$3I = 4T_{2N} - T_N + O(h^4)$$

$$I = \frac{4}{3} T_{2N} - \frac{1}{3} T_N + O(h^4)$$

Extended Simpson's

Combining extended estimates for higher order accuracy

Example: T_N, T_{2N} and S_N

Combining extended estimates for higher order accuracy

Example: Can do the same trick with Extended Simpsons Rule

S_1, S_2, Q_1

$$I = S_N + E(h^4) + O(h^6) \quad h = \frac{H}{N}$$

Extended Boole's Rule

$$I = S_{2N} + \frac{E}{16} + O(h^6) \quad \text{Romberg's Integratrix}$$

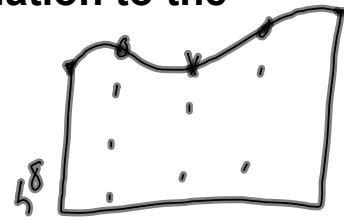
$$15I = 16S_{2N} - S_N + O(h^6)$$

$$Q_1 = \frac{16}{15} S_{2N} - \frac{S_N}{15} + O(h^6)$$

$$= S_{2N} + \frac{1}{15} (S_{2N} - S_N)$$

Higher accuracy: Approach #1 -- Romberg Integration and Richardson's Extrapolation to the limit

Form a tableau from the sequence T_1, T_2, \dots, T_{2N}

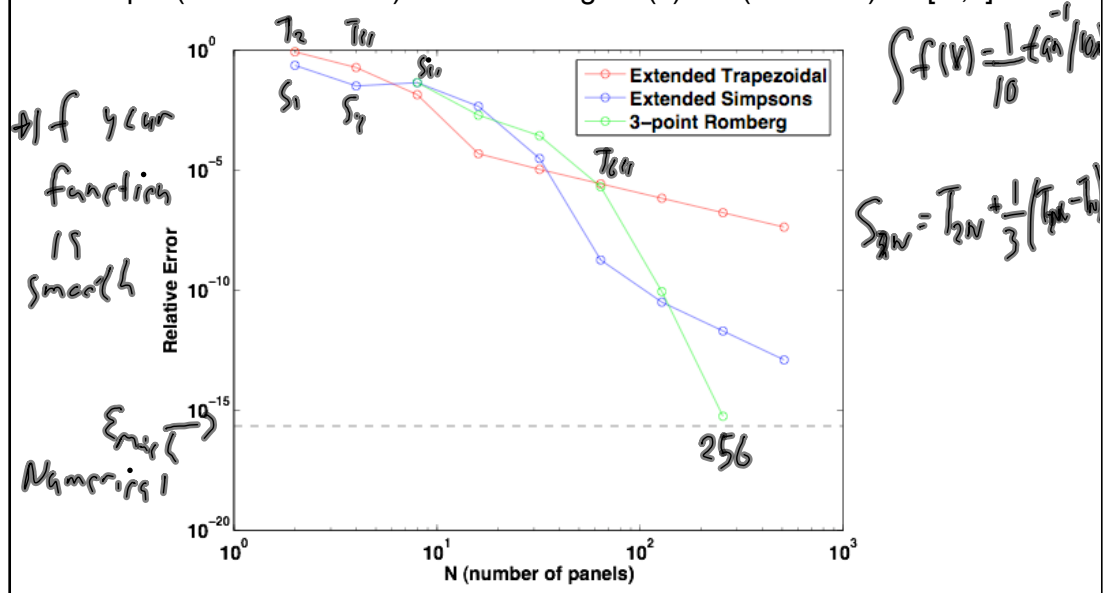


func	h^2	h^4	h^6	h^8
2	T_1			
3	T_2	} $S_1 = T_2 + \frac{1}{3}(T_2 - T_1)$		
5	T_4		} $S_2 = T_4 + \frac{1}{3}(T_4 - T_2)$	G_1
	T_8	S_4		G_2
	\vdots			
$N+1$	T_N	S		
	T_{2N}	S_{2N}		

$\Rightarrow G_N = S_{2N} + \frac{1}{15}(S_{2N} - S_N)$

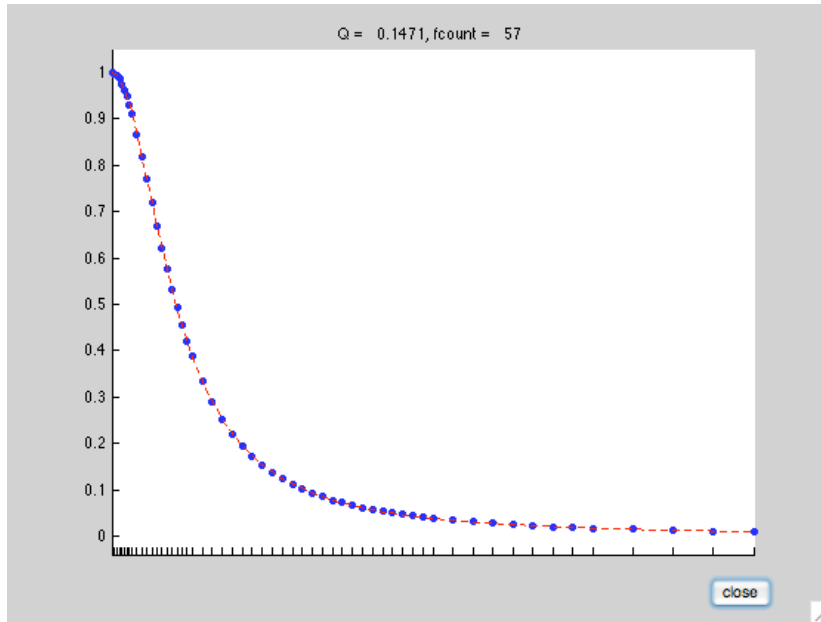
Higher accuracy: Approach #1 -- Romberg Integration and Richardson's Extrapolation to the limit

Example (from homework) estimate integral $f(x) = 1/(1+100x^2)$ on $[0,1]$



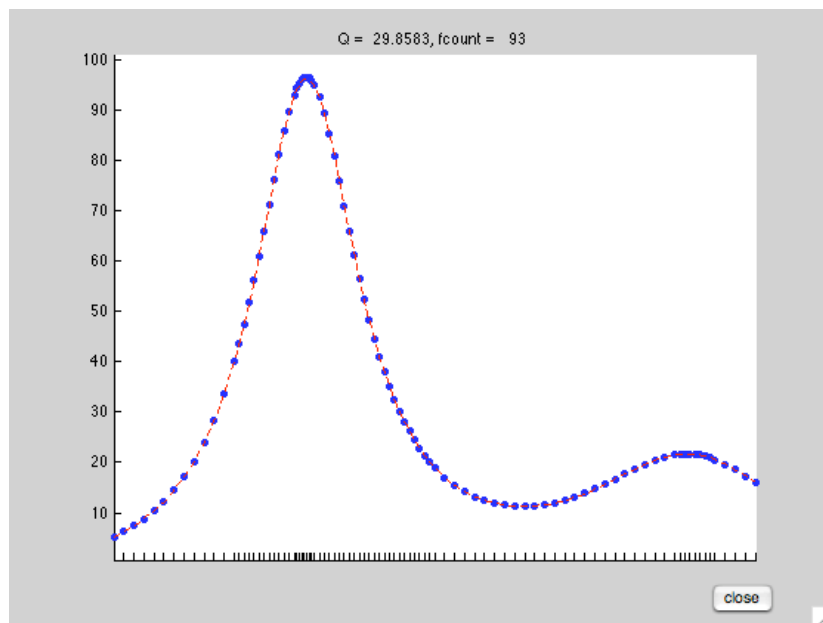
Higher accuracy: Approach #2 -- Adaptive quadrature

$$f(x) = 1/(1+100x^2) \quad I = \tan^{-1}(10x)/10 \text{ on } x = [0,1] \quad (I(1)=0.1471128)$$



Higher accuracy: Approach #2 -- Adaptive quadrature

$$f(x) = \text{humps}(x) \quad \text{on } x = [0,1]$$



Higher accuracy: Approach #2 -- Adaptive quadrature

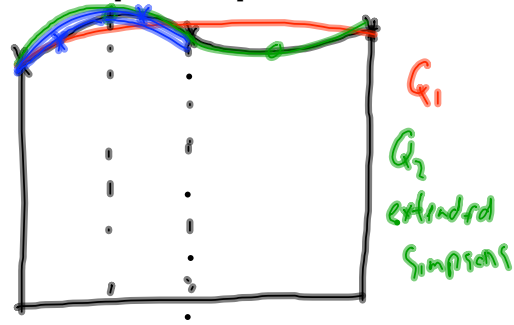
The algorithm: divide and compare

$$\text{if } |Q_2 - Q_1| < \text{tol}$$

$$Q = Q_2 + \frac{1}{15} (Q_2 - Q_1)$$

else

call myself again on left and right piecewise



Higher accuracy: Approach #2 -- Adaptive quadrature

quadtx: a recursive algorithm for adaptive high-order quadrature

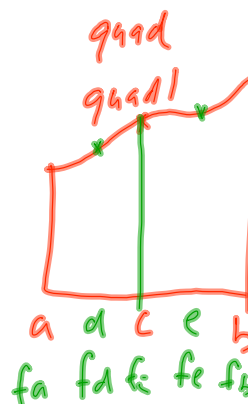
```
function [Q] = quadtx(F,a,b,tol)
% Initialization
c = (a + b)/2;
fa = F(a); fc = F(c); fb = F(b);

% Recursive call
[Q,k] = quadtxstep(F, a, b, tol, fa, fc, fb);
fcount = k + 3;

% -----
function [Q,fcount] = quadtxstep(F,a,b,tol,fa,fc,fb)

% Recursive subfunction used by quadtx.

h = b - a;
c = (a + b)/2; d = (a + c)/2; e = (c + b)/2;
fd = F(d); fe = F(e);
Q1 = h/6 * (fa + 4*fc + fb);
Q2 = h/12 * (fa + 4*fd + 2*fc + 4*fe + fb);
if abs(Q2 - Q1) <= tol
    Q = Q2 + (Q2 - Q1)/15;
    fcount = 2;
else
    [Qa,ka] = quadtxstep(F, a, c, tol, fa, fd, fc);
    [Qb,kb] = quadtxstep(F, c, b, tol, fc, fe, fb);
    Q = Qa + Qb;
    fcount = ka + kb + 2;
end
```



**Higher accuracy: Approach #1 -- Romberg
Integration and Richardson's Extrapolation to the
limit**

An alternative Interpretation of Romberg Integration