Quadrature

Fabio Cannizzo

Definition of Integral

Definition

$$I = \int_a^b f(x)dx = \lim_{n\to\infty}\sum_{i=0}^{n-1} f(\tilde{x}_i)\,\Delta x, \quad \tilde{x}_i \in [a + i\,\Delta x, a + (i+1)\,\Delta x], \quad \Delta x = \frac{b-a}{n}$$

normally computed via the Fundamental Theorem of Calculus

$$\int_a^b f(x)dx = F(b) - F(a)$$

where $F(x)$ is some function such that :

$$\frac{dF(x)}{dx} = f(x)$$

Math Review: Advanced Derivatives Rules

Liebniz rule

$$\frac{d}{dx}\int_{a(x)}^{b(x)} f(x,y)dy = \int_{a(x)}^{b(x)} \frac{d}{dx} f(x,y)dy + f(x,b(x))\frac{d}{dx}b(x) - f(x,a(x))\frac{d}{dx}a(x)$$

As a special case, the **fundamental theorem of calculus**

$$\frac{d}{dx}\int_{a}^{x} f(y)dy = f(x)$$

Math Review: Mean Value Theorem

- **Mean Value Theorem**

$$f(b)-f(a)=(b-a)f'(\xi), \quad \xi \in [a,b]$$

- **2nd Mean Value Theorem for Definite Integrals**

If g(x) has constant sign in [a,b]

$$\int_a^b f(x)g(x)dx = f(\xi)\int_a^b g(x)dx, \quad \xi \in [a,b]$$

Math Review: Properties of Integrals

$$\int_a^b f(x) + g(x) dx = \int_a^b f(x) dx + \int_a^b g(x) dx$$

$$\int_a^b af(x) dx = a \int_a^b f(x) dx$$

$$\int_a^b f(x) dx = -\int_b^a f(x) dx$$

$$\int_a^b f(x) dx = \int_a^c f(x) dx + \int_c^b f(x) dx$$

Why Numerical

- It may just be more convenient
- We may not know the function f(x) in closed form
- In some cases a primitive F(x) in closed form may not exist.

$$\int e^{-x^2} dx$$

Polynomial Integration

- A simple approach is to replace the function being integrated with the approximating polynomial of degree n

$$\int_a^b f(x)dx \approx \int_a^b p_n(x)dx$$

- For greater accuracy, we could use high order polynomials, but they oscillate, so the typical approach is to break the integral (**composite**) and use low order polynomial. If we have n+1 points:

$$\int_a^b f(x)dx \approx \sum_{i=0}^{n-1} \int_{x_i}^{x_{i+1}} p_n(x)dx$$

Rectangle Rule

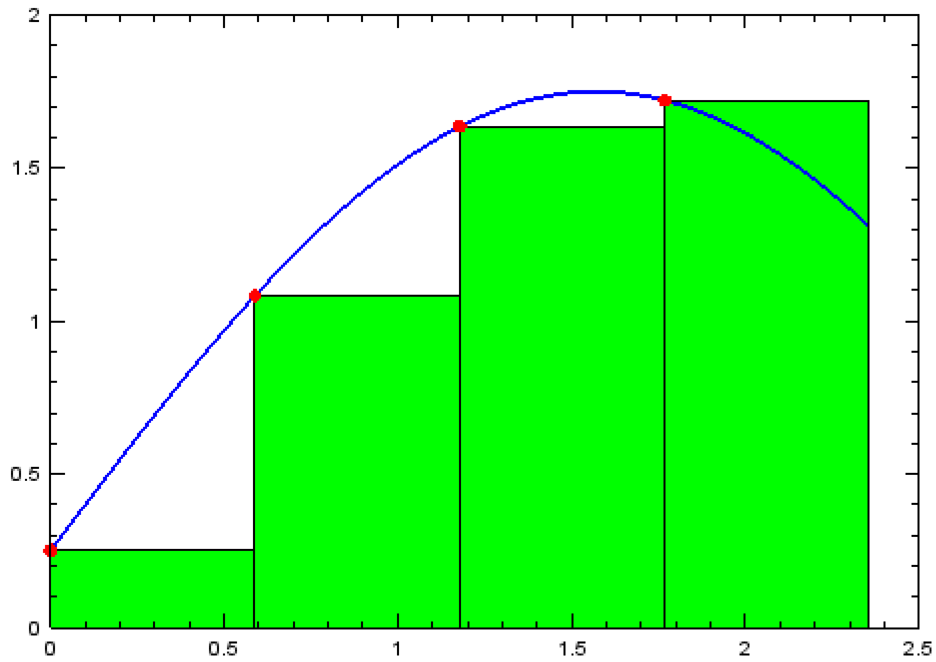
- We can simply use the definition
- We can choose the position of the x in the intervals $[x_i, x_{i+1}]$ as $x=x_i$

$$I = f(a)(b-a) \qquad \text{(SINGLE INTERVAL)}$$

$$I = \int_a^b f(x)dx \approx \sum_{i=0}^{m-1} f(a+i\,h)h, \quad h = \frac{b-a}{m} \qquad \text{(COMPOSITE)}$$

- Note that this is the same as constructing an interpolation **piecewise constant**, then compute its integral

Rectangle Rule: Example



$$\int_0^{\frac{3}{4}\pi} \frac{1}{4} + \frac{3}{2}\sin(x)dx = \frac{3}{16}\pi + \left(\frac{\sqrt{2}}{2} + 1\right)\frac{3}{2} \approx 3.1497$$

```
>> RectangleDemo(@myfun, 0, 0.75*pi(), 4)
ans =  2.7628
```

Rectangle Rule: Source Code

```
function I = Rectangle( f, a, b, m )

   h = (b-a)/m;

   yi = f( a+[0:m-1]*h );

   I=h*sum(yi);

endfunction
```

Rectangle Rule: Convergence

On one single interval [a,b]

$$\left|E\right| \le \frac{1}{2}\left\|f^{'}\left(x\right)\right\|_{\infty}^{[a,b]}\left(b - a\right)^{2}$$

Composite interval:
if we break [a,b] in equally spaced sub-intervals of size h=(b-a)/m

$$\left|E\right| \le \left(\sum_{i=0}^{m-1}\left\|f^{(1)}\left(x\right)\right\|_{\infty}^{[a+ih,a+(i+1)h]}\right)\frac{h^{2}}{2} \le m\left\|f^{(1)}\left(x\right)\right\|_{\infty}^{[a,b]}\frac{h^{2}}{2} = O\left(h\right) = O\left(m^{-1}\right)$$

Rectangle Rule Summary

- Very simple to implement
- Robust
- Computation cost: n evaluations of f(x)
- Composite convergence: $O(m^{-1})$
- If f(x) is increasing, the rectangle rule underestimates, otherwise it overestimates

Mid-Point Rule

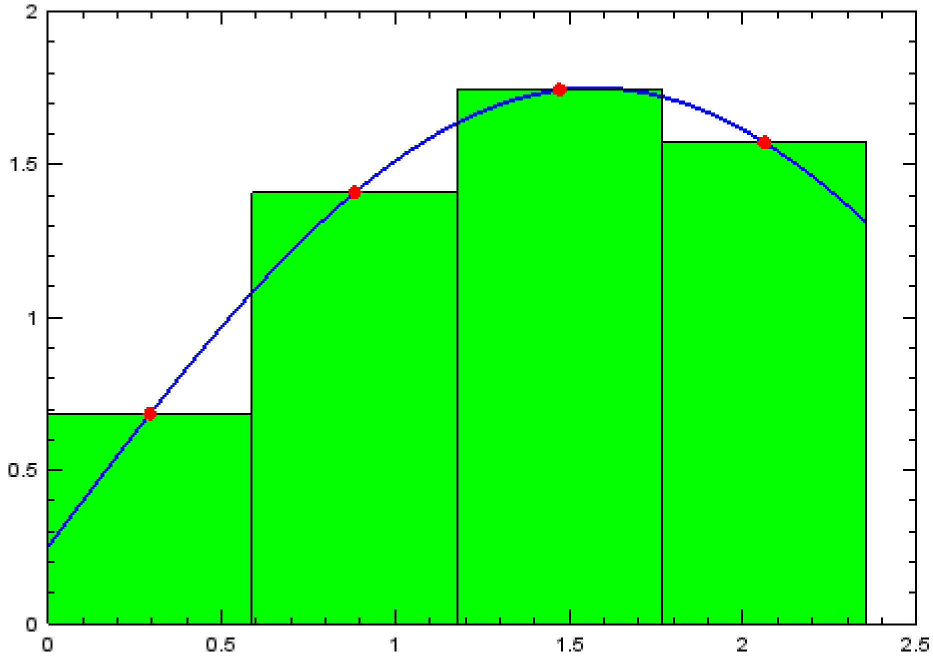
- Similar to the rectangle rule, but we take the mid point of the interval $[x_i, x_{i+1}]$

$$I = \int_a^b f(x)dx \approx f\left(\frac{b+a}{2}\right)(b-a) \qquad \text{(SINGLE INTERVAL)}$$

$$I = \int_a^b f(x)dx \approx \sum_{i=0}^{m-1} f(\tilde{x}_i)h, \quad \tilde{x}_i = a + \left(i + \frac{1}{2}\right)h, \quad h = \frac{b-a}{m} \qquad \text{(COMPOSITE)}$$

- Note that this is the same as constructing an interpolation **piecewise constant**, then compute its integral

Mid-Point Rule: Example



$$\int_0^{\frac{3}{4}\pi} \frac{1}{4} + \frac{3}{2}\sin(x)dx = \frac{3}{16}\pi + \left(\frac{\sqrt{2}}{2} + 1\right)\frac{3}{2} \approx 3.1497$$

```
>> RectangleDemo(@myfun, 0, 0.75*pi(), 4)
ans =  3.1871
```

Mid-Point Rule: Source Code

```
function I = MidPoint( f, a, b, m )

   h = (b-a)/m;

   yi = f( a+([0:m-1]+0.5)*h );

   I=h*sum(yi);

endfunction
```

Mid-Point Rule: Convergence

On one single interval [a,b]

$$\left|E\right| \le \frac{1}{24}\left\|f^{(2)}(\xi)\right\|_{\infty}^{[a,b]}(b - a)^3$$

Composite interval:
If we break [a,b] in equally spaced sub-intervals of size h=(b-a)/m

$$\left|E\right| \le \left(\sum_{i=0}^{m-1}\left\|f^{(2)}(x)\right\|_{\infty}^{[a+ih,a+(i+1)h]}\right)\frac{h^3}{24} \le m\left\|f^{(2)}(x)\right\|_{\infty}^{[a,b]}\frac{h^3}{24} = O\left(h^2\right) = O\left(\frac{1}{m^2}\right)$$

Mid-Point Rule Summary

- Very simple to implement
- Robust
- Computation cost: n evaluations of f(x)
- Composite convergence: $O(m^{-2})$
- Superior to the rectangle rule at roughly the same cost

Trapezoid Rule

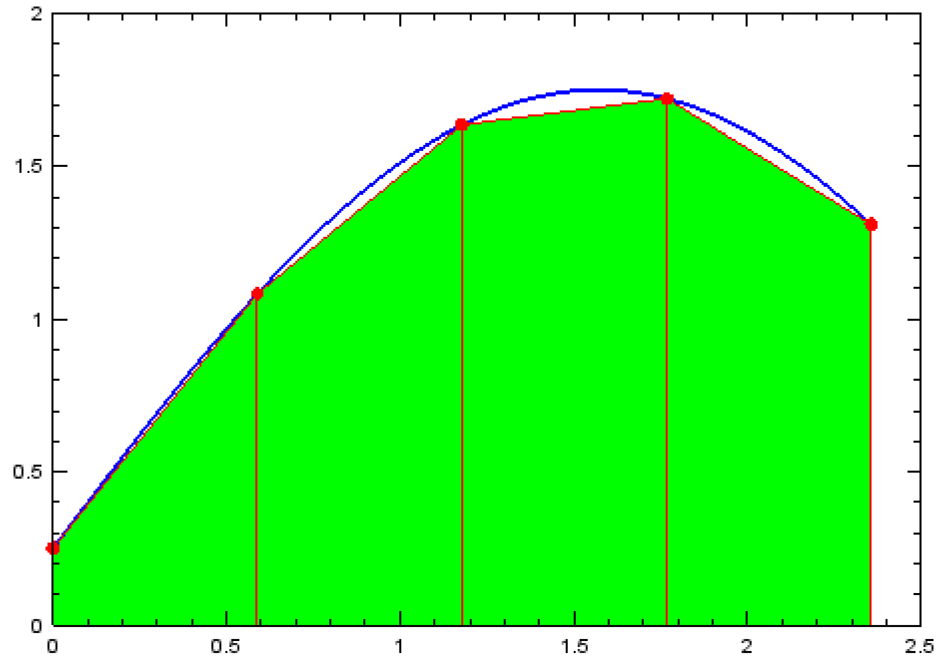
- Instead of choosing one point in $[x_i,x_{i+1}]$, we can use the average of $f(x_i)$ and $f(x_{i+1})$

$$I = \int_a^b f(x)dx \approx \frac{f(a)+f(b)}{2}(b-a), \qquad \text{(SINGLE INTERVAL)}$$

$$I = \int_a^b f(x)dx \approx \sum_{i=0}^{m-1} \frac{f(x_i)+f(x_{i+1})}{2} h, \quad h = \frac{b-a}{n} \qquad \text{(COMPOSITE)}$$

- Note that this is the same as constructing an interpolation **piecewise linear**, then compute its integral
- In other words, we are summing the areas of the trapezoids delimited by $f(x_i)$ and $f(x_{i+1})$

Trapezoid Rule: Example



$$\int_0^{\frac{3}{4}\pi} \frac{1}{4} + \frac{3}{2}\sin(x)dx = \frac{3}{16}\pi + \left(\frac{\sqrt{2}}{2}+1\right)\frac{3}{2} \approx 3.1497$$

```
>> TrapezoidDemo(@myfun, 0, 0.75*pi(), 4)
ans =  3.0752
```

Trapezoid Rule: Source Code

$$I_m = \sum_{i=0}^{m-1} \frac{f(x_i)+f(x_{i+1})}{2} h = \left[\frac{f(x_0)+f(x_m)}{2} + \sum_{i=1}^{m-1} f(x_i) \right] h$$

```
function I = Trapezoid( f, a, b, m )

   h = (b-a)/m;

   yi = f( a+[0:m]*h );

   I = h * ( 0.5 *(yi(1)+yi(m+1)) + sum(yi(2:m)) );

endfunction
```

Trapezoid Rule: Convergence

On 1 single interval [a,b]

$$\left|E\right| \leq \frac{1}{12}\left\|f^{(2)}(x)\right\|_{\infty}^{[a,b]}(b-a)^{3}$$

Composite interval:
If we break [a,b] in equally spaced sub-intervals of size h=(b-a)/m

$$\left|E\right| \leq \left(\sum_{i=0}^{m-1}\left\|f^{(2)}(x)\right\|_{\infty}^{[a+ih,a+(i+1)h]}\right)\frac{h^{3}}{12} \leq m\left\|f^{(2)}(x)\right\|_{\infty}^{[a,b]}\frac{h^{3}}{12} = O\left(h^{2}\right) = O\left(m^{-2}\right)$$

Trapezoid Rule Summary

- Very simple to implement
- Robust
- Computation cost: n+1 evaluations of f(x)
- Composite convergence: $O(m^{-2})$
  - Same as the Mid-Point rule. This is not because the trapezoid rule is poor, it is because the Mid-Point rule, thanks to symmetry, does very well. The intuition is that the mid point is not that different from the average of the two points
- Error
  - if the integrand is concave up, the trapezoidal rule overestimates the true value.
  - Similarly, a concave-down function yields an underestimate.
  - If the interval of the integral being approximated includes an inflection point, the error is harder to identify

Simpson Rule

- Given an even number of intervals, on every pair of intervals we approximate the function with a parabola
- I.e. we are defining an interpolation scheme piecewise quadratic

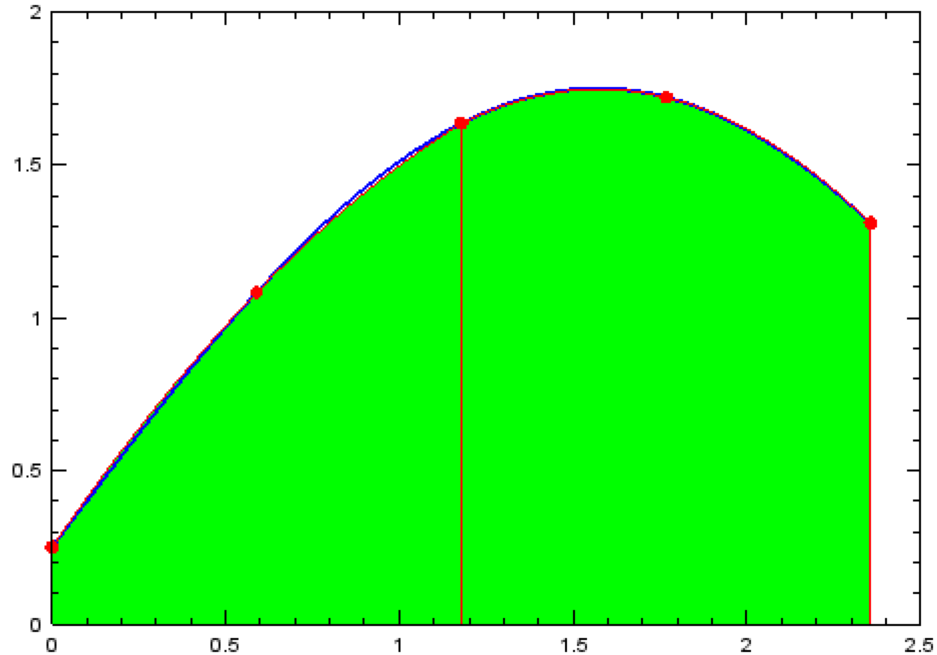
$$\text{(SINGLEINTERVAL)}$$

$$I = \int_a^b f(x)dx \approx \left[f(a) + 4f\left(\frac{a+b}{2}\right) + f(b) \right] \frac{b-a}{6}$$

$$\text{(COMPOSITE)}$$

$$I = \int_a^b f(x)dx \approx \sum_{i=0}^{m/2-1} \frac{f(x_{2i}) + 4f(x_{2i+1}) + f(x_{2i+2})}{3} h, \quad h = \frac{b-a}{m}$$

Simpson Rule: Example



$$\int_0^{\frac{3}{4}\pi} \frac{1}{4} + \frac{3}{2}\sin(x)dx = \frac{3}{16}\pi + \left(\frac{\sqrt{2}}{2} + 1\right)\frac{3}{2} \approx 3.1497$$

```
>> SimpsonDemo(@myfun, 0, 0.75*pi(), 4)
ans =  3.1515
```

Simpson Rule: Source Code

$$I_m = \frac{h}{3}\left(f(x_0) + f(x_m) + 4\sum_{i=0}^{m/2-1} f(x_{2i+1}) + 2\sum_{i=1}^{m/2-1} f(x_{2i}) \right)$$

```
function I = Simpson( f, a, b, m )

   assert(mod(m,2)==0); # requires an even number of points

   h = (b-a)/m;

   yi = f( a+[0:m]*h );

   I=h*( yi(1)+yi(m+1) + 4*sum(yi(2:2:m)) + 2*sum(yi(3:2:m-1)) )/3;

endfunction
```

Simpson Rule: Convergence

On one single interval [a,b]:

$$\left|E\right| \leq \frac{1}{2880}\left\| f^{(4)}(\xi)\right\|_{\infty}^{[a,b]}(b-a)^5$$

Composite interval:
If we break [a,b] in equally spaced sub-intervals of size h=(b-a)/m, and apply the rule to pairs of intervals

$$\left|E\right| \leq \left(\sum_{i=0}^{m/2-1}\left\| f^{(4)}(x)\right\|_{\infty}^{[a+2ih,a+2(i+1)h]}\right)\frac{(2h)^5}{2880} \leq m\left\| f^{(4)}(x)\right\|_{\infty}^{[a,b]}\frac{h^5}{180} = O\left(h^4\right) = O\left(\frac{1}{m^4}\right)$$

Simpson Rule Summary

- Very simple to implement
- Robust
- Computation cost: n+1 evaluations of f(x)
- Composite convergence: $O(m^{-4})$
- Simpson is exact for polynomial of order 3 or lower (error is $O(h^4)$)

Richardson Extrapolation

- As an example, we apply it to the Simpson rule

$$I = S(h) + a\,h^4 + O(h^5)$$

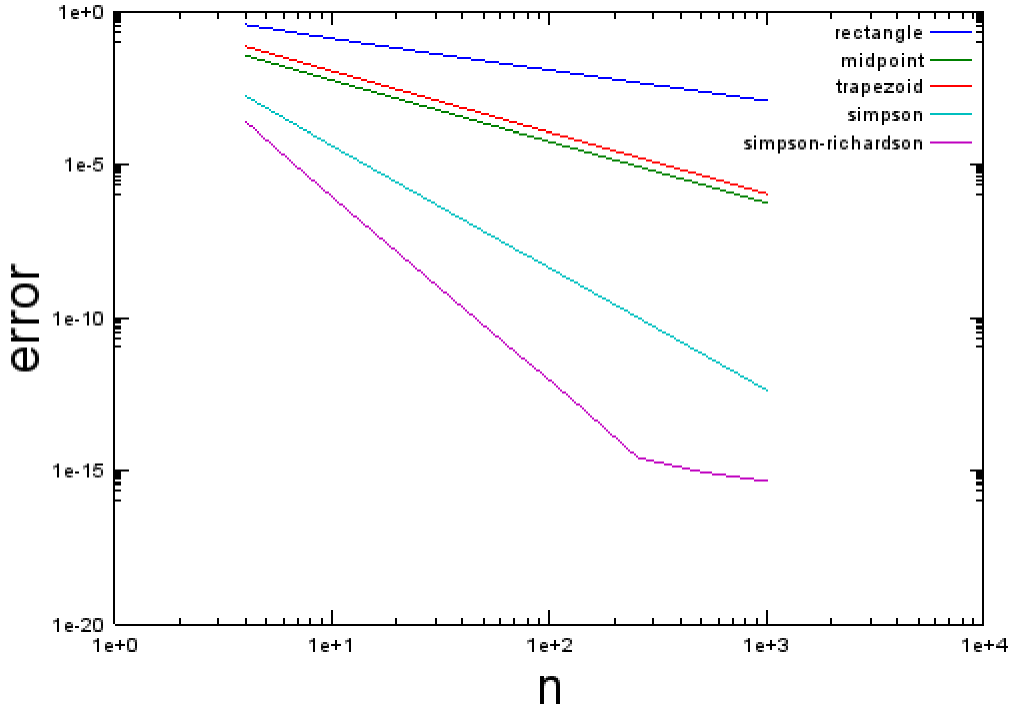
$$I = S\left(\frac{h}{2}\right) + a\left(\frac{h}{2}\right)^4 + O(h^5)$$

$$15I = 16\,S\left(\frac{h}{2}\right) - S(h) + O(h^5)$$

$$I = \frac{16\,S\left(\frac{h}{2}\right) - S(h)}{15} + O(h^5)$$

```
>> s2=Simpson(@myfun, 0, 0.75*pi(), 2)
s2 =  3.1824
>> s4=Simpson(@myfun, 0, 0.75*pi(), 4)
s4 =  3.1515
>> (16*s4-s2)/15
ans =  3.1494
```

Convergence Comparison



The weird scarlet segment connects just two points, the second of which is at machine precision

$$\int_0^{\frac{3}{4}\pi} \frac{1}{4} + \frac{3}{2}\sin(x)dx = \frac{3}{16}\pi + \left(\frac{\sqrt{2}}{2} + 1\right)\frac{3}{2} \approx 3.1497$$

```
>> ConvergenceTest( @myfun, 0, 0.75*pi(), I )
```

Newton Cotes

- Mid-Point, Trapezoid and Simpson belong to the family of Newton Cotes formula, which use equally spaced points in [a,b]
- We distinguish between OPEN and CLOSED type, depending on if *a* and *b* are included (semi-closed formulas also exists)
- Let n be the number of sub-intervals (do not confuse with m, in the composite formulas)

$$\int_a^b f(x)dx \approx (b-a)\sum_{i=0}^{n} A_i f(x_i), \quad \text{where} \quad A_i = \frac{\int_0^1 \prod_{j=0,\, j\neq i}^{n} (ny-j)\,dy}{\prod_{j=0,\, j\neq i}^{n} (i-j)} \quad \text{(CLOSED)}$$

$$\int_a^b f(x)dx \approx (b-a)\sum_{i=1}^{n-1} B_i f(x_i), \quad \text{where} \quad B_i = \frac{\int_0^1 \prod_{j=1,\, j\neq i}^{n-1} (ny-j)\,dy}{\prod_{j=1,\, j\neq i}^{n-1} (i-j)} \quad \text{(OPEN)}$$

Newton Cotes Coefficients

- Newton Cotes integration means approximating the function f(x) in [a,b] with a polynomial p(x)

$$\int_a^b f(x)dx \approx \int_a^b p(x)dx$$

- passing by a number of points equally spaced in [a,b]

$$let \quad \Delta = \frac{b-a}{n}, \quad x = \{a, a+\Delta, a+2\Delta, ..., (n-1)\Delta, b\}$$

- The extreme points a and b are used in *closed* formulas, not used in *open* formulas
- Such polynomial can be written in Lagrange form as

$$p(x) = \sum_{i=0}^{n} \underbrace{\frac{\prod_{j=0,j\neq i}^{n}(x-x_j)}{\prod_{j=0,j\neq i}^{n}(x_i-x_j)}}_{l_i(x)} f(x_i) \quad \text{where it is trivial to verify that } l_i(x_j) = \begin{cases} 0, & i \neq j \\ 1, & i = j \end{cases}$$

Newton Cotes Coefficients

- Newton Cotes coefficients can be simply obtained integrating the $l_i(x)$

$$\int_a^b f(x)dx \approx \int_a^b p(x)dx = \int_a^b \sum_{i=0}^n l_i(x) f(x_i)\, dx = \sum_{i=0}^n f(x_i) \int_a^b l_i(x)dx$$

where

$$\int_a^b l_i(x)dx = \int_a^b \frac{\prod_{j=0,j\neq i}^n (x-x_j)}{\prod_{j=0,j\neq i}^n (x_i-x_j)} dx = (b-a)\frac{\int_0^1 \prod_{j=0,j\neq i}^n (ny-j)\, dy}{\prod_{j=0,j\neq i}^n (i-j)} = (b-a)A_i \quad \text{(CLOSED)}$$

Newton Cotes Error

- The error below refers to **Δ=b-a**, i.e. they are for one single interval (not for the composite formula).
- **n** is the number of sub-intervals (e.g. in Simpson or MidPoint n=2, in Trapezoid n=1)
- For the composite formula the order of convergence is expressed with respect to **h=(b-a)/m**, and it decreases by 1
- Note that, even number of sub-interval leads to superior convergence

SINGLE INTERVAL	COMPOSITE FORMULA
CLOSED, n even: $O\left(\Delta^{n+3}\right)$	CLOSED, n even: $O\left(h^{n+2}\right)$
CLOSED, n odd: $O\left(\Delta^{n+2}\right)$	CLOSED, n odd: $O\left(h^{n+1}\right)$
OPEN, n even: $O\left(\Delta^{n+1}\right)$	OPEN, n even: $O\left(h^{n}\right)$
OPEN, n odd: $O\left(\Delta^{n}\right)$	OPEN, n odd: $O\left(h^{n-1}\right)$

Newton Cotes Example: Mid-Point Rule

- Focusing on one single interval [a,b] we have: Δ=b-a
  - there are 3 points x_0=a, x_1=(a+b)/2, x_2=b, i.e. 2 sub-intervals (so n=2)
  - points at the boundary are not used, so the formula is of type *OPEN*
  - Error is $O(\Delta^{n+1})$, e.g. $O(\Delta^3)$
  - Formula is:

$$\int_a^b f(x)dx \approx (b-a)\sum_{i=1}^{1} B_i f(x_i) = (b-a)B_1 f(x_1)$$

$$\text{where } B_1 = \int_0^1 \frac{\prod_{j=1,j\neq 1}^{2-1}(y-j)}{\prod_{j=1,j\neq 1}^{2-1}(1-j)}dy = \int_0^1 dy = 1$$

$$\text{hence } \int_a^b f(x)dx \approx (b-a)f(x_1)$$

Newton Cotes Example: Simpson Rule

- Focusing on one single interval [a,b] we have: Δ=b-a
  - there are 3 points $x_0=a$, $x_1=(a+b)/2$, $x_2=b$, i.e. 2 sub-intervals (so n=2).
  - points at the boundary are used, so the formula is of type *CLOSED*
  - Error is *$O(\Delta^{n+3})$, e.g. $O(\Delta^5)$*

$$\int_a^b f(x)dx \approx (b-a)\sum_{i=1}^{1} \widetilde{A}_i f(x_i) = (b-a)\left[\widetilde{A}_0 f(x_0) + \widetilde{A}_1 f(x_1) + \widetilde{A}_2 f(x_2)\right], \quad \text{where:}$$

$$\widetilde{A}_0 = \frac{\int_0^1 \prod_{j=0,j\neq 0}^{n} (ny-j)\,dy}{\prod_{j=0,j\neq i}^{n} (0-j)} = \frac{\int_0^1 (2y-1)(2y-2)\,dy}{(0-1)(0-2)} = \frac{1}{6}$$

$$\widetilde{A}_1 = \frac{\int_0^1 \prod_{j=0,j\neq 1}^{n} (ny-j)\,dy}{\prod_{j=0,j\neq i}^{n} (1-j)} = \frac{\int_0^1 (2y-0)(2y-2)\,dy}{(1-0)(1-2)} = \frac{4}{6}$$

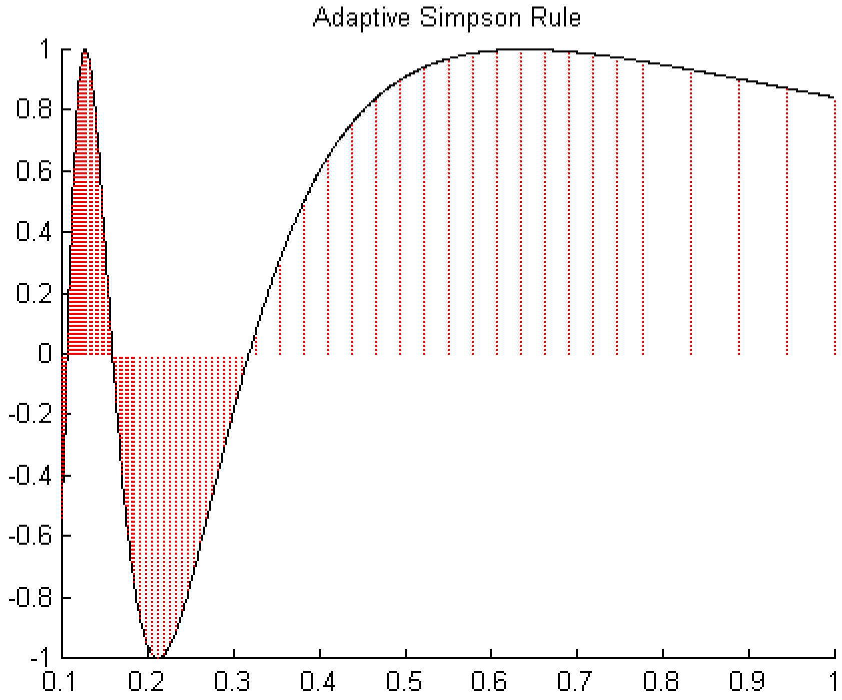
$$\widetilde{A}_2 = \frac{\int_0^1 \prod_{j=0,j\neq 2}^{n} (ny-j)\,dy}{\prod_{j=0,j\neq i}^{n} (2-j)} = \frac{\int_0^1 (2y-0)(2y-1)\,dy}{(2-0)(2-1)} = \frac{1}{6}$$

Adaptive Simpson Rule

- At iteration 1 we start with Simpson with exactly 3 points in [a,b]: {a,c,b}, where c=(b+a)/2
- We compute Simpson in [a,c] and [c,b], adding a central point in each interval
- We check if S(a,b) – [S(a,c)+S(c,b)] is below a certain tolerance. If yes we stop, otherwise we keep bisecting and we now repeat the process individually on [a,c] and [c,b]
- The idea is that we add point only where it is necessary, i.e. where the error is larger

Adaptive Simpson Rule

- We observe that the density of point is not uniform



Adaptive Simpson Rule

- Simpson error terms are of order $O(h^4)$
- Combining the estimates $S_2=[S(a,c)+S(c,b)]$ with the one obtained combining S_2 and $S_1=S[a,b]$ via Richardson extrapolation, we get the stopping criteria:

  $|S_2-[16\ S_2 - S_1]/15| < \varepsilon \Rightarrow |S_2 - S_1| < 15\ \varepsilon$

  where ε is the accuracy criteria for the portion of interval [a,b]
  - if we stop, we return Richardson extrapolation
  - if we continue, we return S_2, as we do not have an estimate of the error and we cannot apply Richardson

```
>> [Ias,n]=AdaptiveSimpson(@myfun,0,0.75*pi(),1e-8)
Ias =  3.14970879432736
n =  121
```

Gaussian Integration

- In Newton Cotes formula, we fix the position of the points and we compute weights for each point
- If we choose also the position of the points arbitrarily, we have more degrees of freedom to minimize the error
- An n-point Gaussian quadrature rule is constructed to yield an exact result for polynomials of degree $2n - 1$ or less by a suitable choice of the points x_i and weights w_i for $i = 1,...,n$
- This is way more accurate than Newton Cotes!
- The domain of integration for such a rule is, without loss of generality, conventionally taken as $[-1, 1]$
- The catch is that high order is not always good. In order to obtain high accuracy, the function must behave like a poynomial (i.e. very smooth)

Gaussian Integration

- The catch is that in order to obtain high accuracy, the function need to behave similarly to a polynomial
- If we can write f(x) as w(x)g(x), where g(x) behaves like a polynomial, then

$$\int_a^b f(x)dx = \int_a^b w(x)g(x)dx \approx \sum_1^n A_i g(x_i)$$

$$\sum_1^n A_i \, x_i^j = \int_a^b w(x) \, x_i^j \, dx, \qquad j = 0...2n-1$$ non-linear system 2n x 2n

- i.e. the weights A_i are chosen so that if g(x) was indeed a polynomial, the formula would be exact
- Newton Cotes are exact if f(x) is a polynomial, here we can achieve exact results also for a polynomial multiplied by some weight function, which can be chosen to improve smoothness of f(x), but if not chosen properly could turn a smooth f(x) into a non-smooth g(x)!

Gaussian Integration

- For some special weight functions, the points x_i are the roots of some special polynomials and the values of the weights A_i are tabulated
- Once we get xi and Ai from the tables, we simply apply the formula:

$$\int_a^b f(x)dx = \int_a^b w(x)g(x)dx \approx \sum_1^n A_i g(x_i)$$

Note that g(x) is not the same as f(x)

Gaussian Integration

$W(x)$	$[a,b]$	Polynomial
1	$[-1,1]$	Legendre
$\dfrac{1}{\sqrt{1-x^2}}$	$(-1,1)$	Chebishev 1
$\sqrt{1-x^2}$	$[-1,1]$	Chebishev 2
$\dfrac{1}{\sqrt{x}}$	$[0,1]$	Related to Legendre
$\sqrt{x}$	$[0,1]$	Related to Legendre
$\dfrac{x}{\sqrt{1-x}}$	$[0,1]$	Related to Chebishev 1
$x^{\alpha}e^{-x}$	$[0,\infty]$	Laguerre
e^{-x^2}	$[-\infty,\infty]$	Hermite
$(1-x^{\alpha})(1+x^{\beta})$	$(-1,1)$	Jacobi

Polynomial roots (x_i) and weights (A_i) can be found tabulated

Gauss-Legendre Roots and Weights Table Example

n = 2

i	weight - w_i	abscissa - x_i
1	1.0000000000000000	-0.5773502691896257
2	1.0000000000000000	0.5773502691896257

n = 3

i	weight - w_i	abscissa - x_i
1	0.8888888888888888	0.0000000000000000
2	0.5555555555555556	-0.7745966692414834
3	0.5555555555555556	0.7745966692414834

n = 4

i	weight - w_i	abscissa - x_i
1	0.6521451548625461	-0.3399810435848563
2	0.6521451548625461	0.3399810435848563
3	0.3478548451374538	-0.8611363115940526
4	0.3478548451374538	0.8611363115940526

Gaussian Integration: Example

$$\int_0^{\frac{3}{4}\pi} \frac{1}{4}+\frac{3}{2}\sin(x)dx$$

We use Gauss Legendre with 4 points

(same computational cost as in previous examples)

First we transform into [-1,1], with the transformation $y = ax + b$

$$\begin{cases} y(0) = -1 \\ y\left(\frac{3}{4}\pi\right) = 1 \end{cases} \Rightarrow y = \frac{8}{3\pi}x - 1,\ dx = \frac{3\pi}{8}dy$$

$$\int_0^{\frac{3}{4}\pi} \frac{1}{4}+\frac{3}{2}\sin(x)dx = \int_{-1}^{+1}\left[\frac{1}{4}+\frac{3}{2}\sin\left(\frac{3\pi}{8}(y+1)\right)\right]\frac{3\pi}{8}dy$$

$$\approx \sum_{i=1}^{4} w_i\left[\frac{1}{4}+\frac{3}{2}\sin\left(\frac{3\pi}{8}(y_i+1)\right)\right]\frac{3\pi}{8} = 3.1497 \quad \text{(very accurate!)}$$

```
>> f=@(x)(1/4+3/2*sin(3*pi/8*(x+1)))*3*pi/8;
>> w=[0.6521451548625461, 0.6521451548625461, 0.3478548451374538, 0.3478548451374538];
>> y=[-0.3399810435848563, 0.3399810435848563, -0.8611363115940526, 0.8611363115940526];
>> w*f(y')
ans =  3.1497
```

Comparison with Newton Cotes

- Suppose we use Gauss-Legendre with 3 points
- Convergence on single interval
  - This is expected to be exact for a polynomial of order 3x2-1=5, i.e. $O((b-a)^7)$
  - Simpson also uses 3 points, but it is only exact for a polynomial up to order 3, i.e. $O((b-a)^5)$
- When we transform to **composite**, if we divide [a,b] in m intervals,
  - Simpson requires roughly 2n evaluations, because at the boundary of every interval it uses common points
  - Gauss instead requires 3n evaluations
- Computation cost is O(n) in both cases, but accuracy is $O(h^6)$ vs $O(h^4)$

Warnings

- Same issues discussed with interpolation: high order schemes can give bad surprises with non-smooth functions
- Beware discontinuities and singularities!
  - Non adaptive method will lead to very poor accuracy (they rely on f(x) to ne smooth)
  - Adaptive method will keep bisecting, becoming very expensive, and with poor accuracy (the interval could become smaller than machine precision). There needs to be a guard to avoid bisecting too much.
- Identify discontinuities and break the integral
- Fast and frequent changes of slope can be as bad as discontinuities

Modern Methods

- Most modern methods are adaptive
- They are combination of more basic methods
- A key performance requirement of adaptive method is to be able to reuse previously computed points (e.g. in Simpson, at every bisection, we only need 2 new extra points)

Multiple Integrals

- Very expensive: computational cost grows exponentially with number of dimensions
- If I need n points for a 1-D integral, I may need n^3 for a 3-D integral
- When possible we should reduce the order of the integral
  - e.g. if we start from a double integral and we can solve the inner integral analytically, we are left with a 1-D integral
- A good technique for high dimensionality is Monte Carlo integration

Further Readings

- Prof Amos Rons' lecture notes
  http://pages.cs.wisc.edu/~amos/412/lecture-notes/lecture17.pdf
  http://pages.cs.wisc.edu/~amos/412/lecture-notes/lecture18.pdf
  http://pages.cs.wisc.edu/~amos/412/lecture-notes/lecture19.pdf
- Prof Saltzman's lecture notes
  http://www.dirac.org/numerical/gaussian_quadrature/gaussian.pdf
- Numerical Recipes in C++