Numerical Analysis and Computing

Lecture Notes #9
— Numerical Integration and Differentiation —
Gaussian Quadrature; Multiple Integrals; Improper Integrals

Peter Blomgren,
⟨blomgren.peter@gmail.com⟩

Department of Mathematics and Statistics
Dynamical Systems Group
Computational Sciences Research Center
San Diego State University
San Diego, CA 92182-7720

http://terminus.sdsu.edu/

Fall 2014

Outline

Gaussian Quadrature

Idea: Evaluate the function at a set of **optimally chosen** points in the interval.

We will choose $\{x_0, x_1, \ldots, x_n\} \in [a, b]$ and coefficients c_i, so that the approximation

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

is exact for the largest class of polynomials possible.

We have already seen that the open Newton-Cotes formulas sometimes give us better "bang-for-buck" than the closed formulas (*e.g.* the mid-point formula uses only 1 point and is as accurate as the two-point trapezoidal rule). — Gaussian quadrature takes this one step further.

Quadrature Types — A Comparison

	Newton-Cotes		Gaussian
	Open	Closed	
Quadrature Points	Degree of Accuracy	Degree of Accuracy	Degree of Accuracy
1	**1***	—	1
2	1	**1**†	3
3	3	**3**#	5
4	3	3	7
5	5	5	9

* — The mid-point rule.
† — Trapezoidal rule.
# — Simpson's rule.

The mid-point rule is the only optimal scheme we have see so far.

Gaussian Quadrature — Example

2-Point Formula

Suppose we want to find an optimal two-point formula:

$$\int_{-1}^{1} f(x)\,dx = c_1 f(x_1) + c_2 f(x_2).$$

Since we have 4 parameters to play with, we can generate a formula that is **exact up to polynomials of degree 3**. We get the following 4 equations:

$$\begin{array}{rcccl} \int_{-1}^{1} 1\,dx &=& 2 &=& c_1 + c_2 \\ \int_{-1}^{1} x\,dx &=& 0 &=& c_1x_1 + c_2x_2 \\ \int_{-1}^{1} x^2\,dx &=& \frac{2}{3} &=& c_1x_1^2 + c_2x_2^2 \\ \int_{-1}^{1} x^3\,dx &=& 0 &=& c_1x_1^3 + c_2x_2^3 \end{array} \quad \Bigg\| \quad \begin{array}{rcl} c_1 &=& 1 \\ c_2 &=& 1 \\ x_1 &=& -\frac{\sqrt{3}}{3} \\ x_2 &=& \frac{\sqrt{3}}{3} \end{array}$$

Higher Order Gaussian Quadrature Formulas

We could obtain higher order formulas by adding more points, computing the integrals, and solving the resulting non-linear system of equations... but it gets very painful, very fast.

The **Legendre Polynomials** come to our rescue!

The Legendre polynomials $P_n(x)$ are orthogonal on $[-1, 1]$ with respect to the weight function $w(x) = 1$, *i.e.*

$$\int_{-1}^{1} P_n(x)P_m(x)\,dx = \alpha_n \delta_{n,m} = \begin{cases} 0 & m \neq n \\ \alpha_n & m = n. \end{cases}$$

If $P(x)$ is a polynomial of degree less than n, then

$$\int_{-1}^{1} P_n(x)P(x)\,dx = 0.$$

A Quick Note on Legendre Polynomials

We will see Legendre polynomials in **more detail later**. For now, all we need to know is that they satisfy the property

$$\int_{-1}^{1} P_n(x)P_m(x)\,dx = \alpha_n \delta_{n,m}.$$

and the first few Legendre polynomials are

$$\begin{array}{rcl} P_0(x) &=& 1 \\ P_1(x) &=& x \\ P_2(x) &=& x^2 - 1/3 \\ P_3(x) &=& x^3 - 3x/5 \\ P_4(x) &=& x^4 - 6x^2/7 + 3/35 \\ P_5(x) &=& x^5 - 10x^3/9 + 5x/21. \end{array}$$

It turns out that the **roots** of the Legendre polynomials are the nodes in Gaussian quadrature.

Higher Order Gaussian Quadrature Formulas

Theorem

Suppose that $\{x_1, x_2, \ldots, x_n\}$ are the roots of the n^{th} Legendre polynomial $P_n(x)$ and that for each $i = 1, 2, \ldots, n$, the coefficients c_i are defined by

$$c_i = \int_{-1}^{1} \prod_{\substack{j=1 \\ j \neq i}}^{n} \frac{x - x_j}{x_i - x_j}\,dx.$$

If $P(x)$ is any polynomial of degree less than $2n$, then

$$\int_{-1}^{1} P(x)\,dx = \sum_{i=1}^{n} c_i P(x_i).$$

Proof of the Theorem

Let us first consider a polynomial, $P(x)$ with degree less than n. $P(x)$ can be rewritten as an $(n-1)$-st Lagrange polynomial with nodes at the roots of the n^{th} Legendre polynomial $P_n(x)$. This representation is exact since the error term involves the n^{th} derivative of $P(x)$, which is zero. Hence,

$$\int_{-1}^{1} P(x)\,dx = \int_{-1}^{1} \left[\sum_{i=1}^{n} \prod_{\substack{j=1 \\ j\neq i}}^{n} \frac{x-x_j}{x_i-x_j} P(x_i) \right] dx$$

$$= \sum_{i=1}^{n} \left[\int_{-1}^{1} \prod_{\substack{j=1 \\ j\neq i}}^{n} \frac{\mathbf{x}-\mathbf{x_j}}{\mathbf{x_i}-\mathbf{x_j}}\,\mathbf{dx} \right] P(x_i) = \sum_{i=1}^{n} \mathbf{c_i} P(x_i),$$

which verifies the result for polynomials of degree less than n.

Proof of the Theorem

If the polynomial $P(x)$ of degree $[n, 2n)$ is divided by the n^{th} Legendre polynomial $P_n(x)$, we get:

$$P(x) = Q(x)P_n(x) + R(x)$$

where both $Q(x)$ and $R(x)$ are of degree less than n.

[1] Since $\deg(Q(x)) < n$

$$\int_{-1}^{1} Q(x)P_n(x)\,dx = 0.$$

[2] Further, since x_i is a root of $P_n(x)$:

$$P(x_i) = Q(x_i)P_n(x_i) + R(x_i) = R(x_i).$$

Proof of the Theorem

[3] Now, since $\deg(R(x)) < n$, the first part of the proof implies

$$\int_{-1}^{1} R(x)\,dx = \sum_{i=1}^{n} c_i R(x_i).$$

Putting [1], [2] and [3] together we arrive at

$$\int_{-1}^{1} P(x)\,dx = \int_{-1}^{1} \left[Q(x)P_n(x) + R(x)\right] dx$$

$$= \int_{-1}^{1} R(x)\,dx = \sum_{i=1}^{n} c_i R(x_i)$$

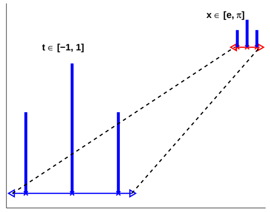
$$= \sum_{i=1}^{n} c_i P(x_i),$$

which shows that the formula is exact for all polynomials $P(x)$ of degree less than $2n$. □

Gaussian Quadrature beyond the interval $[-1, 1]$

By a simple linear transformation,

$$t = \frac{2x-a-b}{b-a} \Leftrightarrow x = \frac{(b-a)t+(b+a)}{2},$$



we can apply the Gaussian Quadrature formulas to any interval

$$\int_{a}^{b} f(x)\,dx = \int_{-1}^{1} f\left(\frac{(b-a)\mathbf{t}+(b+a)}{2}\right) \underbrace{\frac{(b-a)}{2}}\,dt.$$

Rescale summation weights by this factor.

Examples

Degree	$P_n(x)$	Roots / Quadrature points
2	$x^2 - 1/3$	$\{-1/\sqrt{3},\ 1/\sqrt{3}\}$
3	$x^3 - 3x/5$	$\{-\sqrt{3/5},\ 0,\ \sqrt{3/5}\}$
4	$x^4 - 6x^2/7 + 3/35$	$\{-0.86114,\ -0.33998,\ 0.33998,\ 0.86114\}$

Table: Quadrature points on "standard interval.:

$$\int_0^{\pi/4} (\cos(x))^2\, dx = \frac{1}{4} + \frac{\pi}{8} = 0.642699081698724$$

Degree	"Standard" Quadrature points $\in [-1, 1]$	(Unscaled) Weight Coefficients
2	-0.57735, 0.57735	1, 1
3	-0.77459, 0, 0.77459	0.55556, 0.88889, 0.55556
4	-0.86113, -0.33998, 0.33998, 0.86113	0.34785, 0.65215, 0.65215, 0.34785
Degree	**Translated Quadrature points**	**Rescaled Weight Coefficients**
2	0.16597, 0.61942	0.39269, 0.39269
3	0.08851, 0.39270, 0.69688	0.21816, 0.34906, 0.21816
4	0.05453, 0.25919, 0.52621, 0.73087	0.13660, 0.25609, 0.25609, 0.13660

Table: Quadrature points translated to interval of interest; with weight coefficients.

Examples

$$\int_0^{\pi/4} (\cos(x))^2\, dx = \frac{1}{4} + \frac{\pi}{8} = 0.642699081698724$$

Degree	Translated Quadrature points	Rescaled Weight Coefficients
2	0.16597, 0.61942	0.39269, 0.39269
3	0.08851, 0.39270, 0.69688	0.21816, 0.34906, 0.21816
4	0.05453, 0.25919, 0.52621, 0.73087	0.13660, 0.25609, 0.25609, 0.13660

Table: Quadrature points translated to interval of interest; with weight coefficients.

Degree	Integral approximation	Error
2	0.642317235049753	0.0003818466489...
3	0.642701112090729	0.0000020303920...
4	0.642699075999924	0.0000000056988...

Table: Approximation and Error, for GQ.

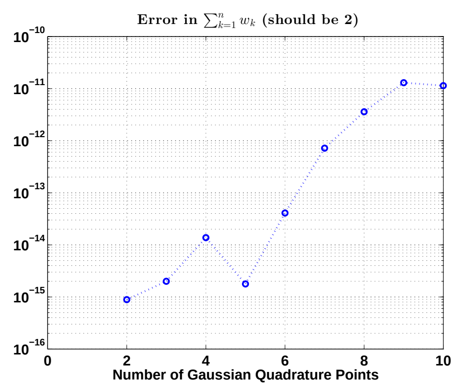
More Quadrature Points?!

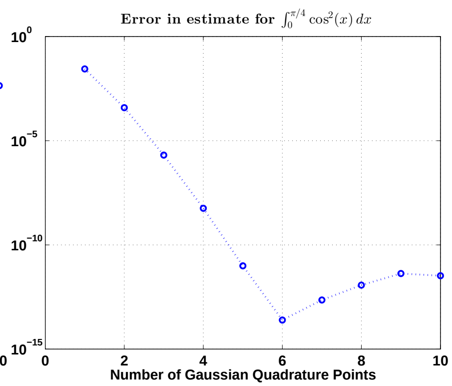
It turns out it is not that difficult to write a piece of (matlab) code which computes the Lagrange polynomials and their roots; however numerical roundoff causes some issues with the coefficients, after some "hand cleaning" we get:

$$\begin{aligned}
L_5(x) &= x^5 - \tfrac{10}{9}x^3 + \tfrac{5}{21}x \\
L_6(x) &= x^6 - \tfrac{15}{11}x^4 + \tfrac{5}{11}x^2 - \tfrac{5}{231} \\
L_7(x) &= x^7 - \tfrac{21}{13}x^5 + \tfrac{105}{143}x^3 - \tfrac{35}{429}x \\
L_8(x) &= x^8 - \tfrac{28}{15}x^6 + \tfrac{14}{13}x^4 - \tfrac{28}{143}x^2 + \tfrac{7}{1287} \\
L_9(x) &= x^9 - \tfrac{36}{17}x^7 + \tfrac{126}{85}x^5 - \tfrac{84}{221}x^3 + \tfrac{17}{656}x \\
L_{10}(x) &= x^{10} - \tfrac{45}{19}x^8 + \tfrac{630}{323}x^6 - \tfrac{210}{323}x^4 + \tfrac{106}{1413}x^2 - \tfrac{1}{733}
\end{aligned}$$

More Quadrature Points?!

It is, of course, tempting to use many quadrature points, but the quality of the points has to be considered. Here, using the points given by matlab's `roots` command:





Homework #7 — http://webwork.sdsu.edu

- Will open on 10/22/2014 at 09:30am PDT.
- Will close no earlier than 11/6/2014 at 09:00pm PST.

Problems based on:

Implement Composite Simpson's Rule, and use your code to solve BF-4.4.3-a,b,c,d.

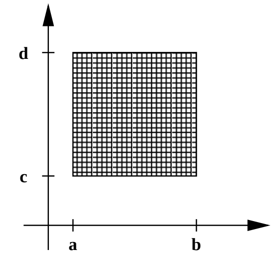
BF-4.7.1-a,b
BF-4.7.2-a,b
BF-4.7.3-a,b
BF-4.7.4-a,b

The World is not One-Dimensional

Very few interesting problems are one-dimensional, so we need integration schemes for multiple integrals, *i.e.*

$$\mathcal{I} = \iint_R f(x,y)\, dx\, dy,$$

where $R = \{(x,y) : x \in [a,b],\ y \in [c,d]\}$.

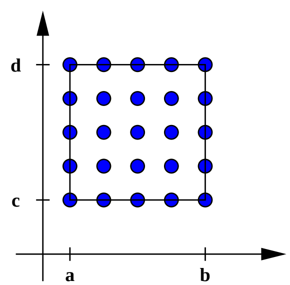


Good News: The integration techniques we have developed previously can be adopted for multi-dimensional integration in a straight-forward way.

Composite Simpson's Rule (CSR) is our favorite integration scheme; we discuss multi-dimensional integration in that context.

Multi-Dimensional Composite Simpson's Rule

We divide the x-range $[a,b]$ into an even number n_x of sub-intervals with nodes spaced $h_x = (b-a)/n_x$ apart, and the y-range $[c,d]$ into an even number n_y of sub-intervals with nodes spaced $h_y = (d-c)/n_y$ apart.



We write

$$\mathcal{I} = \iint_R f(x,y)\, dx\, dy = \int_a^b \left[\int_c^d f(x,y)\, dy \right] dx,$$

and first apply CSR to approximate the integration in y — treating x as a constant.

Composite Simpson's Rule in the y-coordinate

1. Let $y_j = c + jh_y$, $j = 0, 1, \ldots, n_y$, then

$$\int_c^d f(x,y)\, dy = \frac{h_y}{3}\left[f(x,y_0) - f(x,y_n) + \sum_{j=1}^{n_y/2} \left[2f(x,y_{2j}) + 4f(x,y_{2j-1})\right]\right] - \frac{(d-c)h_y^4}{180} \cdot \frac{\partial^4 f(x,\mu_y)}{\partial y^4},$$

for some $\mu_y \in [c,d]$.

2. Then we apply the integral in the $x-$coordinate...

$$\int_a^b \int_c^d f(x,y)\, dy\, dx = \frac{h_y}{3}\left[\int_a^b f(x,y_0)\, dx - \int_a^b f(x,y_n)\, dx + \sum_{j=1}^{n_y/2} \left[2\int_a^b f(x,y_{2j})\, dx + 4\int_a^b f(x,y_{2j-1})\, dx \right]\right] - \frac{(d-c)h_y^4}{180} \int_a^b \frac{\partial^4 f(x,\mu_y)}{\partial y^4}\, dx,$$

Apply Composite Simpson's Rule in the x-coordinate

Now, we "simply" apply CSR in the x-coordinate, for each integral in the expression...

$$\int_a^b \int_c^d f(x,y)\,dy\,dx \approx$$

$$\frac{h_x h_y}{9}\Bigg\{\Bigg[f(x_0,y_0) - f(x_n,y_0) + \sum_{i=1}^{n_x/2}\Big(2f(x_{2i},y_0) + 4f(x_{2i-1},y_0)\Big)\Bigg]$$

$$-\Bigg[f(x_0,y_n) - f(x_n,y_n) + \sum_{i=1}^{n_x/2}\Big(2f(x_{2i},y_n) + 4f(x_{2i-1},y_n)\Big)\Bigg]$$

$$+\sum_{j=1}^{n_y/2}\Bigg[2\Bigg[f(x_0,y_{2j}) - f(x_n,y_{2j}) + \sum_{i=1}^{n_x/2}\Big(2f(x_{2i},y_{2j}) + 4f(x_{2i-1},y_{2j})\Big)\Bigg]$$

$$+4\Bigg[f(x_0,y_{2j-1}) - f(x_n,y_{2j-1}) + \sum_{i=1}^{n_x/2}\Big(2f(x_{2i},y_{2j-1}) + 4f(x_{2i-1},y_{2j-1})\Big)\Bigg]\Bigg]\Bigg\}$$

This looks somewhat painful, but do not despair!!! [First, a peek at the error...]

2-Dimensional Composite Simpson's Rule — The Error

The error for the approximation is

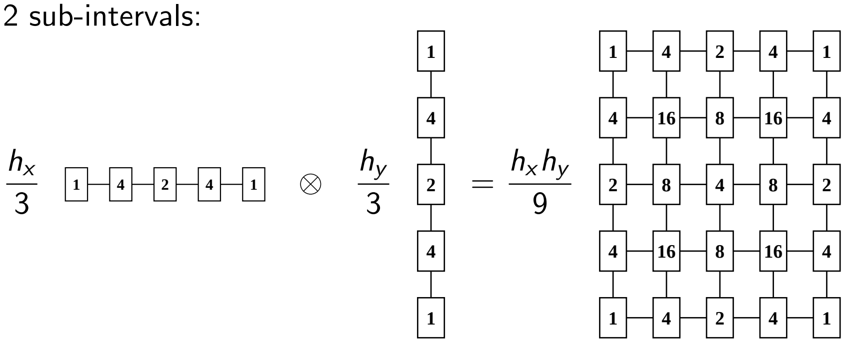
$$E = -\frac{(b-a)(d-c)}{180}\left[h_x^4 \frac{\partial^4 f}{\partial x^4}(\nu_x,\mu_x) + h_y^4 \frac{\partial^4 f}{\partial y^4}(\nu_y,\mu_y)\right]$$

for some $(\nu_x,\mu_x),\ (\nu_y,\mu_y) \in R = [a,b]\times[c,d]$.

"Derivation of the error is left as an exercise for the interested reader..."

Building 2-D CSR in a Comprehensible Way?

Consider the tensor product of the x- and y-stencils for CSR with 2 sub-intervals:



x-locations: $\{x_0,x_1,x_2,x_3,x_4,\ldots\}$, y-locations: $\{y_0,y_1,y_2,y_3,y_4,\ldots\}$

evaluation points: $\{(x_k,y_\ell) : x_k \in \{x_0,x_1,x_2,x_3,x_4,\ldots\},\ y_\ell \in \{y_0,y_1,y_2,y_3,y_4,\ldots\}\}$

Evaluate the function at the corresponding points, multiply by the above weights, and sum $\Rightarrow$ 2-D CSR.

Building 2-D CSR in a Comprehensible Way? — Example

$$\frac{9}{h_x h_y}\int_{x_0}^{x_4}\int_{y_0}^{y_4} f(x,y)\,dx\,dy \approx$$

$$1\Big[f(x_0,y_0) + 4f(x_1,y_0) + 2f(x_2,y_0) + 4f(x_3,y_0) + f(x_4,y_0)\Big] +$$

$$4\Big[f(x_0,y_1) + 4f(x_1,y_1) + 2f(x_2,y_1) + 4f(x_3,y_1) + f(x_4,y_1)\Big] +$$

$$2\Big[f(x_0,y_2) + 4f(x_1,y_2) + 2f(x_2,y_2) + 4f(x_3,y_2) + f(x_4,y_2)\Big] +$$

$$4\Big[f(x_0,y_3) + 4f(x_1,y_3) + 2f(x_2,y_3) + 4f(x_3,y_3) + f(x_4,y_3)\Big] +$$

$$1\Big[f(x_0,y_4) + 4f(x_1,y_4) + 2f(x_2,y_4) + 4f(x_3,y_4) + f(x_4,y_4)\Big]$$

$$h_x = \frac{x_4 - x_0}{4}, \quad h_y = \frac{y_4 - y_0}{4}.$$

Building Higher-Dimensional Schemes

Using the same strategy, we can build a 3-D CSR-scheme

$$\text{CSR}_{xyz} = \text{CSR}_x \otimes \text{CSR}_y \otimes \text{CSR}_z.$$

There's nothing unique about the usage of CSR. The same idea can be used to build higher dimensional Gaussian Quadrature schemes. If we have the stencils for the one-dimensional (Composite) Gaussian Quadrature schemes in the x-, y- and z-directions (GQ_x, GQ_y, GQ_z):

$$\text{GQ}_{xyz} = \text{GQ}_x \otimes \text{GQ}_y \otimes \text{GQ}_z.$$

If you're really twisted you could use different schemes in the different coordinate directions, *i.e.*

$$\texttt{NUMINT}_{xyz} = \text{CSR}_x \otimes \text{GQ}_y \otimes \text{Romberg}_z.$$

Needless to say, the error terms would get really *"interesting."*

Integrating Outside the Box

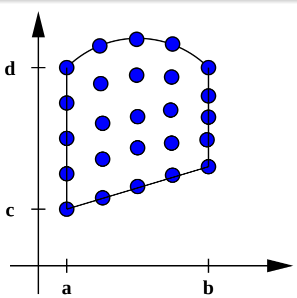
The integration schemes we have discussed so far only works for rectangular regions $[a, b] \times [c, d]$...

In calculus we compute integrals of this form:

$$\int_a^b \int_{c(x)}^{d(x)} f(x, y)\, dy\, dx$$

We can modify our integration schemes to deal with this type of integrals.

Dealing with Variable Integration Limits



In order to numerically compute an integral of this type

$$\int_a^b \int_{c(x)}^{d(x)} f(x, y)\, dy\, dx$$

we are going to use CSR with a fixed step size $h_x = (b-a)/n_x$ in the x-direction, and variable step size $h_y = (d(x) - c(x))/n_y$ in the y-direction.

Variable Integration Limits — Example

For simplicity we apply straight-up one-step SR to

$$\int_a^b \int_{c(x)}^{d(x)} f(x, y)\, dy\, dx,$$

and get

$$\begin{aligned}\int_a^b \int_{c(x)}^{d(x)} f(x, y)\, dy\, dx \approx \\ \frac{h_x}{3}\Bigg\{ & \frac{d(x_0)-c(x_0)}{6}\left[f(x_0, c(x_0)) + 4f(x_0, \tfrac{c(x_0)+d(x_0)}{2}) + f(x_0, d(x_0))\right] + \\ & \frac{4(d(x_1)-c(x_1))}{6}\left[f(x_1, c(x_1)) + 4f(x_1, \tfrac{c(x_1)+d(x_1)}{2}) + f(x_1, d(x_1))\right] + \\ & \frac{d(x_2)-c(x_2)}{6}\left[f(x_2, c(x_2)) + 4f(x_2, \tfrac{c(x_2)+d(x_2)}{2}) + f(x_2, d(x_2))\right]\Bigg\},\end{aligned}$$

where $x_0 = a$, $x_1 = \dfrac{a+b}{2}$, $x_2 = b$.

Variable Integration Limits

We can imagine how to extend to multiple dimensions, *i.e.*

$$\int_a^b \int_{c(x)}^{d(x)} \int_{e(x,y)}^{f(x,y)} g(x,y,z)\,dz\,dy\,dx.$$

Again, there nothing special about Simpson's Rule — we can attack variable integration limits with Gaussian Quadrature, Trapezoidal Rule, or Boole's Rule...

Note that there is nothing stopping us from using adaptive schemes to find the integrals... but the complexity of the code grows!

Algorithm: Variable Limits Double Integral using CSR

Algorithm: Variable Limits Double Integral – CSR

```
[1] hx = (b-a)/n, ENDPTS=0, EVENPTS=0, ODDPTS=0
[2] FOR i = 0,1,...,n                                    % CSR in x
      x  = a + i*hx
      k1 = f(x,c(x)) + f(x,d(x))                         % End terms
      k2 = 0                                             % Even terms
      k3 = 0                                             % Odd terms
      hy = (d(x)-c(x))/n
      FOR j = 1,2,...,(m-1)
        y = c(x)+j*hy
        Q = f(x,y)
        IF j EVEN: k2 += Q,  ELSE: k3 += Q
      END-FOR-j
      L = hy*(k1 + 2*k2 + 4*k3)/3;
      IF i is 0 OR n:  ENDPTS  += L
      ELSEIF i EVEN:   EVENPTS += L
      ELSEIF i ODD:    ODDPTS  += L
    END-FOR-i
    INTAPPROX = hx*(ENDPTS+2*EVENPTS+4*ODDPTS)/3
```


Improper Integrals — Introduction

"Improper" integrals:

[1] Integrals over infinite intervals

$$\int_a^\infty f(x)\,dx.$$

[2] Integrals with unbounded functions

$$\int_a^b \frac{f(x)}{(x-a)^p}\,dx.$$

Note: We can always transform [1]→[2]

$$\int_a^\infty f(x)\,dx = \left\{ \begin{array}{l} t = x^{-1} \\ dt = -x^{-2}\,dx \end{array} \right\} = \int_{1/a}^0 -t^{-2} f(t^{-1})\,dt$$

More Forgotten Calculus

The integral

$$\int_a^b \frac{dx}{(x-a)^p}$$

converges if and only if $p \in (-\infty, 1)$, and

$$\int_a^b \frac{dx}{(x-a)^p} = \frac{(b-a)^{1-p}}{1-p}.$$

If $f(x)$ can be written on the form

$$f(x) = \frac{g(x)}{(x-a)^p}, \quad p \in (-\infty, 1), \quad g \in C[a,b]$$

then the improper integral

$$\int_a^b f(x)\,dx, \text{ exists.}$$

Splitting the Integrand using Taylor Expansions

Assuming that $g \in C^{d+1}[a,b]$, for some $d \in \mathbb{Z}^+$, the Taylor polynomial of degree d is

$$P_d(x) = \sum_{k=0}^{d} \frac{g^{(k)}(a)(x-a)^k}{k!}.$$

We can now write

$$\int_a^b f(x)\,dx = \int_a^b \frac{g(x)-P_d(x)}{(x-a)^p}\,dx + \int_a^b \frac{P_d(x)}{(x-a)^p}\,dx,$$

where the last integral is easy to find, since $P_d(x)$ is a polynomial:

$$\sum_{k=0}^{d} \int_a^b \frac{g^{(k)}(a)}{k!}(x-a)^{k-p}\,dx = \sum_{k=0}^{d} \frac{g^{(k)}(a)}{k!(k+1-p)}(b-a)^{k+1-p}$$

Splitting the Integrand using Taylor Expansions

If we let

$$\int_a^b f(x)\,dx \approx \sum_{k=0}^{d} \frac{g^{(k)}(a)}{k!(k+1-p)}(b-a)^{k+1-p},$$

then the approximation error is bounded by:

$$\int_a^b \frac{g(x)-P_d(x)}{(x-a)^p}\,dx = \int_a^b \frac{R_d(x)}{(x-a)^p}\,dx = \int_a^b \frac{g^{(d+1)}(\xi(x))(x-a)^{d+1}}{(d+1)!(x-a)^p}\,dx$$

$$\leq \frac{1}{(d+1)!} \max_{x\in[a,b]} |g^{(d+1)}(x)| \int_a^b (x-a)^{d+1-p}\,dx$$

$$= \frac{g^{(d+1)}(\xi)}{(d+1)!\,(d+2-p)}(b-a)^{d+2-p}.$$

What if we want to do better?

Numerical Approximation of the Remainder Term

To get a more accurate approximation to the integral, we compute the numerical approximation of the remainder term:

$$\int_a^b \frac{g(x)-P_d(x)}{(x-a)^p}\,dx.$$

Define: (Remove the singularity)

$$G(x) = \begin{cases} \dfrac{g(x)-P_d(x)}{(x-a)^p} & x \in (a,b] \\ 0 & x = a. \end{cases}$$

Apply: Composite Simpson's Rule

$$\int_a^b G(x)\,dx \approx \frac{h}{3}\left[G(x_0) - G(x_n) + \sum_{j=1}^{n/2}\left[4G(x_{2j-1}) + 2G(x_{2j})\right]\right].$$

Add the CSR-approximation to $\sum_{k=0}^{d} \frac{g^{(k)}(a)}{k!(k+1-p)}(b-a)^{k+1-p}$.

Example#1

We want to compute

$$\int_0^1 \frac{e^x}{x^{1/2}}\,dx.$$

The fourth order Taylor polynomial is

$$P_4(x) = 1 + x + \frac{x^2}{2} + \frac{x^3}{6} + \frac{x^4}{24},$$

so

$$\int_0^1 \frac{P_4(x)}{x^{1/2}}\,dx = \int_0^1 x^{-1/2} + x^{1/2} + \frac{x^{3/2}}{2} + \frac{x^{5/2}}{6} + \frac{x^{7/2}}{24}\,dx$$

$$= \frac{2}{1} + \frac{2}{3} + \frac{2}{2\cdot 5} + \frac{2}{6\cdot 7} + \frac{2}{24\cdot 9} \approx 2.923544974$$

Example#1

Next, we apply CSR with $h = 1/4$ to $\int_0^1 G(x)\,dx$, where

$$G(x) = \begin{cases} \dfrac{e^x - P_4(x)}{x^{1/2}} & x \in (0,1] \\ 0 & x = 0. \end{cases}$$

$$\int_0^1 G(x)\,dx \approx \frac{1}{4\cdot 3}\Big[0 + 4\cdot 0.0000170 + 2\cdot 0.00413 + 4\cdot 0.0026026 + 0.0099485\Big] = 0.0017691.$$

Hence,

Result

$$\int_0^1 \frac{e^x}{x^{1/2}}\,dx \approx 2.923544974 + 0.0017691 = 2.9253141$$

Example#1

If $|G^{(k\geq 4)}(x)| < 1$ on $(0,1]$, the error from CSR is bounded by

$$\frac{1}{180}\cdot\frac{1}{4^4} = 0.0000217.$$

The error bound for the Taylor-only approximation is bounded by

$$\frac{1}{5!\cdot 5.5} = 0.00151515$$

If, instead of adding the CSR-approximation of $\int G(x)\,dx$, we used $P_5(x)$, the error bound for that Taylor-only approximation would be

$$\frac{1}{6!\cdot 6.5} = 0.00021044.$$

The $P_6(x)$-only-error is comparable with the $P_4(x)$+CSR-error:

$$\frac{1}{7!\cdot 7.5} = 0.000026455.$$

Example#2

We are going to approximate the integral

$$\int_1^\infty \frac{1}{x^{3/2}}\sin\left(\frac{1}{x}\right)\,dx.$$

A quick change of variables $t = x^{-1}$ gives us

$$\int_0^1 t^{-1/2}\sin(t)\,dt.$$

The sixth Taylor polynomial $P_6(t)$ for $\sin(t)$ about $t = 0$ is

$$P_6(t) = t - \frac{1}{6}t^3 + \frac{1}{120}t^5, \quad |R_6(t)| \leq \frac{1}{7!} = 0.00019841$$

$$\int_0^1 t^{-1/2}P_6(t)\,dt = \int_0^1 t^{1/2} - \frac{1}{6}t^{5/2} + \frac{1}{120}t^{9/2}\,dt$$

$$= \frac{2}{3} - \frac{2}{7\cdot 6} + \frac{2}{11\cdot 120} = 0.62056277$$

Example#2

We define

$$G(t) = \begin{cases} \dfrac{\sin(t) - P_6(t)}{t^{1/2}} & t \in (0,1] \\ 0 & t = 0, \end{cases}$$

and apply CSR with $h = 1/32$ to $\int_0^1 G(t)\,dt$ to get

Result

$$\int_1^\infty \frac{1}{x^{3/2}}\sin\left(\frac{1}{x}\right)\,dx$$

$$\approx 0.62056277 - 0.0000261672790305 = 0.62053660\ldots$$

which is accurate to within $\sim 10^{-8}$.