July 20, 2013

Chapter 15: Numerical Integration

Uri M. Ascher and Chen Greif
Department of Computer Science
The University of British Columbia
{ascher,greif}@cs.ubc.ca

Slides for the book
A First Course in Numerical Methods (published by SIAM, 2011)
`http://www.ec-securehost.com/SIAM/CS07.html`

Goals of this chapter

- To develop some classical methods for approximating definite integrals;
- to concentrate on issues that arise also in more involved and complicated circumstances, such as high order method development and adaptivity;
- * to consider some challenging problems in multidimensional integration.

Outline

- Basic rules
- Composite numerical integration
- Gaussian quadrature
- Adaptive quadrature
- *Multi-dimensional integration

*advanced

Numerical integration (quadrature)

- The need to integrate arises very frequently in numerical computations. Instance: finite element methods for differential equations use basis functions in the spirit of Chapter 11, in combination with integrals computed over tiny pieces of the computational domain.
- The need to know how to integrate numerically can be more immediate than in the case of differentiation because we often do not know how to integrate even simple-looking functions.
- As opposed to differentiation, which is local in nature, integration is a *global* operation.
- Note that while the derivative of $f(x)$ is typically rougher than f, the integral of $f(x)$ is smoother. Consequently, no special roundoff error difficulties are expected here, unlike in Chapter 14.
- Many-dimensional integration often arises in statistical applications.

Basic rules

Consider only definite integrals; quadrature $\equiv$ numerical integration in one dimension. Seek approximation formulas of the form

$$I_f = \int_a^b f(x)dx \approx \sum_{j=0}^{n} a_j f(x_j).$$

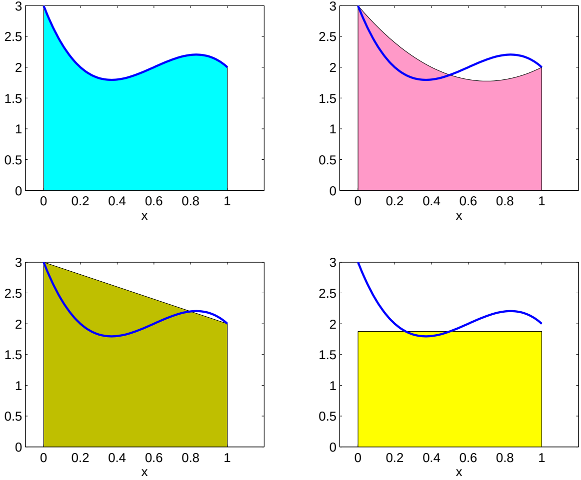
- Derive basic quadrature rules by interpolating the integrand $f(x)$ using Lagrange form and integrating the resulting polynomial.
- The *weights* a_j are then given by

$$a_j = \int_a^b L_j(x)dx, \quad L_j(x) = \prod_{\substack{k=0 \\ k\neq j}}^{n} \frac{(x-x_k)}{(x_j-x_k)}.$$

- These weights are independent of f and can be found in advance!

Basic trapezoidal, midpoint and Simpson rules

A picture is better than 2048 bytes. Upper left: exact; upper right: Simpson; lower left: trapezoidal; lower right: midpoint.



Error in basic rules

- The error satisfies

$$
\begin{aligned}
E(f) &= \int_a^b f(x)dx - \sum_{j=0}^{n} a_j f(x_j) \\
&= \int_a^b f[x_0, x_1, \ldots, x_n, x](x - x_0)(x - x_1)\cdots(x - x_n)dx.
\end{aligned}
$$

- To estimate this further can be delicate. Results:

Method	Formula	Error
Midpoint	$(b-a)f(\frac{a+b}{2})$	$\frac{f''(\xi_1)}{24}(b-a)^3$
Trapezoidal	$\frac{b-a}{2}[f(a)+f(b)]$	$-\frac{f''(\xi_2)}{12}(b-a)^3$
Simpson	$\frac{b-a}{6}[f(a)+4f(\frac{b+a}{2})+f(b)]$	$-\frac{f''''(\xi_3)}{90}(\frac{b-a}{2})^5$

- These are all Newton-Cotes formulas: Trapezoidal and Simpson are *closed*, midpoint is *open*.

Outline

- Basic rules
- Composite numerical integration
- Gaussian quadrature
- Adaptive quadrature
- *Multi-dimensional integration

Composite methods

- The basic rules are good for small intervals. So, use them on subintervals.
- Similar to piecewise polynomial interpolation, but easier because no need here to worry about global smoothness.
- $$\int_a^b f(x)dx = \sum_{i=1}^{r} \int_{t_{i-1}}^{t_i} f(x)dx, \quad \text{e.g. } t_i = a + ih.$$
- If error in basic rule is $E(f) = \tilde{K}(b-a)^{q+1}$ then error in composite method is

$$E(f) = K(b-a)h^q.$$

Composite trapezoidal method

- Composite method

$$\int_a^b f(x)dx \quad \approx \quad \frac{h}{2}[f(a) + 2f(t_1) + 2f(t_2) + \cdots + 2f(t_{r-1}) + f(b)].$$

- Error estimate

$$E(f) = \sum_{i=1}^{r} \left(-\frac{f''(\eta_i)}{12} h^3 \right) = -\frac{f''(\eta)}{12}(b-a)h^2.$$

- Special case: $f(b) = f(a)$. Then the method

$$\int_a^b f(x)dx \quad \approx \quad h \sum_{i=0}^{r-1} f(a+ih)$$

is particularly effective (related to best ℓ_2-approximations and trigonometric polynomials, Chapter 13).

Composite trapezoidal method

- Composite method

$$\int_a^b f(x)dx \approx \frac{h}{2}[f(a) + 2f(t_1) + 2f(t_2) + \cdots + 2f(t_{r-1}) + f(b)].$$

- Error estimate

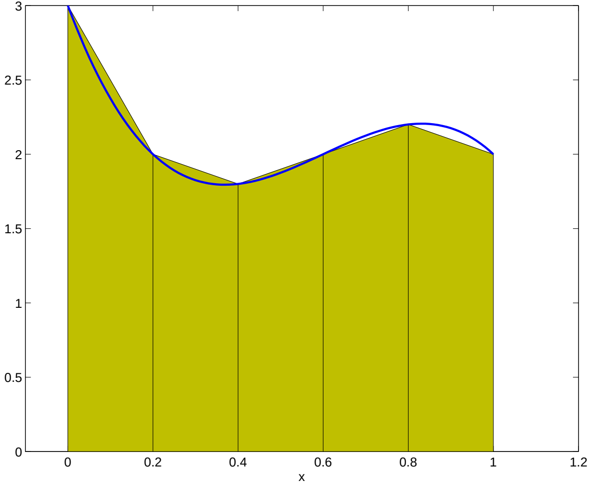
$$E(f) = \sum_{i=1}^{r} \left(-\frac{f''(\eta_i)}{12} h^3 \right) = -\frac{f''(\eta)}{12}(b-a)h^2.$$

- Special case: $f(b) = f(a)$. Then the method

$$\int_a^b f(x)dx \approx h \sum_{i=0}^{r-1} f(a+ih)$$

is particularly effective (related to best ℓ_2-approximations and trigonometric polynomials, Chapter 13).

Composite trapezoidal



Composite Simpson method

- Composite method (for convenience, pose basic rule on subinterval of length $2h$):

$$\int_{t_{2k-2}}^{t_{2k}} f(x)dx \approx \frac{2h}{6}[f(t_{2k-2}) + 4f(t_{2k-1}) + f(t_{2k})].$$

Then sum up contributions (using even r), obtaining the famous formula

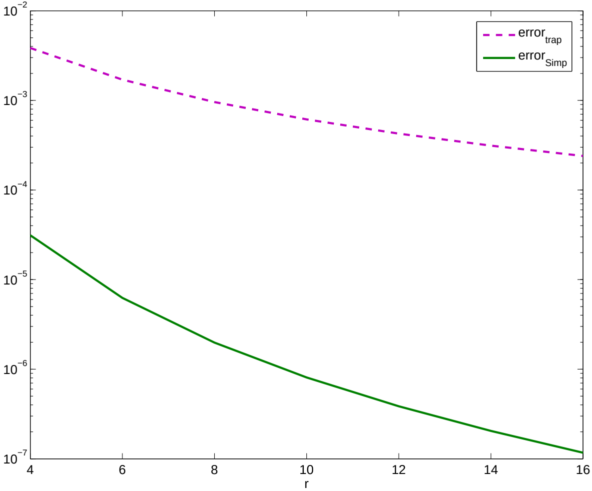
$$\int_a^b f(x)dx \quad \approx \quad \frac{h}{3}[f(a) + 2\sum_{k=1}^{r/2-1} f(t_{2k}) + 4\sum_{k=1}^{r/2} f(t_{2k-1}) + f(b)].$$

Error estimate

$$E(f) = -\frac{f''''(\zeta)}{180}(b-a)h^4.$$

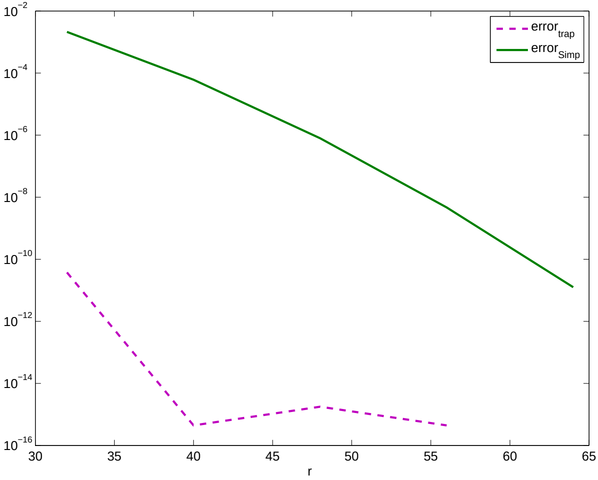
Example: errors for trapezoidal and Simpson

Integrate $I = \int_0^1 e^{-x^2} dx$.
Plot errors for $h = 1/r$: evidently the 4th order Simpson is much more accurate.



Example: errors for trapezoidal and Simpson

Integrate $I = \int_{-10}^{10} e^{-x^2} dx$.
Plot errors for $h = 1/r$: here trap is great because integrand is "almost periodic".



Composite method and abscissae

- The basic rule involves points $x_0, x_1, \ldots, x_n$ on $[-1, 1]$, say. These are mapped into each panel $[t_{i-1}, t_i]$.
- So, the integrand $f(x)$ is sampled at points

$$t_{i,k} = \frac{t_{i-1} + t_i}{2} + \frac{t_i - t_{i-1}}{2} x_k, \quad k = 0, 1, \ldots, n, \quad i = 1, \ldots, r.$$

- If $x_0 = -1,\ x_n = 1$ then $t_{i-1,n} = t_{i,1}$. Thus, in the composite trapezoidal method, we sample the integrand at $r + 1$ points, not $2r$. Likewise in the composite Simpson method.
- In composite midpoint, sample integrand at r points (no special deal). So, comparable to composite trapezoidal in both cost and accuracy. It shines if the integrand $f(x)$ has jump discontinuities at mesh points.

Composite trapezoidal, midpoint and Simpson methods

With $rh = b - a$, r a positive integer (must be even in the Simpson case), we have the **formulas**

$$\begin{aligned}\int_a^b f(x)dx &\approx \frac{h}{2}[f(a) + 2\sum_{i=1}^{r-1} f(a+ih) + f(b)], \quad \text{trapezoidal} \\ &\approx \frac{h}{3}[f(a) + 2\sum_{k=1}^{r/2-1} f(t_{2k}) + 4\sum_{k=1}^{r/2} f(t_{2k-1}) + f(b)], \quad \text{Simpson} \\ &\approx h\sum_{i=1}^{r} f(a+(i-1/2)h), \quad \text{midpoint.}\end{aligned}$$

Precision and order

- The precision of a basic quadrature rule is ρ if the rule is exact for all poslynomials of degree at most ρ.
- For trapezoidal and midpoint, $\rho = 1$. For Simpson, $\rho = 3$.
- The order of accuracy of the corresponding composite method is $q = \rho + 1$. The error is bounded by

$$|E(f)| \leq Ch^q \|f^{(q)}\|.$$

- Note: *order* relates to h^q, *precision* relates to $f^{(q)}$.

Precision and order

- The precision of a basic quadrature rule is ρ if the rule is exact for all poslynomials of degree at most ρ.
- For trapezoidal and midpoint, $\rho = 1$. For Simpson, $\rho = 3$.
- The order of accuracy of the corresponding composite method is $q = \rho + 1$. The error is bounded by

$$|E(f)| \leq C h^q \|f^{(q)}\|.$$

- Note: *order* relates to h^q, *precision* relates to $f^{(q)}$.

Precision and order

- The precision of a basic quadrature rule is ρ if the rule is exact for all poslynomials of degree at most ρ.
- For trapezoidal and midpoint, $\rho = 1$. For Simpson, $\rho = 3$.
- The order of accuracy of the corresponding composite method is $q = \rho + 1$. The error is bounded by

$$|E(f)| \le Ch^q \|f^{(q)}\|.$$

- Note: *order* relates to h^q, *precision* relates to $f^{(q)}$.

Outline

Gaussian quadrature: maximizing precision

- Back to basic rules on interval $[-1,1]$
$$I_f = \int_{-1}^{1} f(x)dx \approx \sum_{j=0}^{n} a_j f(x_j).$$
- Want to determine $x_0, x_1, \ldots, x_n$ so as to maximize precision.
- For $n=0$ the error is $E(f) = \int_{-1}^{1} f[x_0,x](x-x_0)dx$.
Choose midpoint $x_0 = 0$ because $\int_{-1}^{1}(x-0)dx = 0$. Then writing $f[0,x] = f[0,0] + x\frac{f''(\xi(x))}{2}$ gives $E(f) = \frac{f''(\eta)}{3}$, hence $\rho = 1$.
- Note that $\rho = 0$ if $x_0 \neq 0$: midpoint is the first Gaussian quadrature method.

Gaussian quadrature: maximizing precision

- Back to basic rules on interval $[-1, 1]$

$$I_f = \int_{-1}^{1} f(x)dx \approx \sum_{j=0}^{n} a_j f(x_j).$$

- Want to determine $x_0, x_1, \ldots, x_n$ so as to maximize precision.
- For $n = 0$ the error is $E(f) = \int_{-1}^{1} f[x_0, x](x - x_0)dx$.
  Choose midpoint $x_0 = 0$ because $\int_{-1}^{1}(x - 0)dx = 0$. Then writing $f[0, x] = f[0, 0] + x\frac{f''(\xi(x))}{2}$ gives $E(f) = \frac{f''(\eta)}{3}$, hence $\rho = 1$.
- Note that $\rho = 0$ if $x_0 \neq 0$: midpoint is the first Gaussian quadrature method.

Gauss points

- For general n

$$\begin{aligned} E(f) &= \int_{-1}^{1} f[x_0, x_1, \ldots, x_n, x](x - x_0)(x - x_1)\cdots(x - x_n)dx \\ &\equiv \int_{-1}^{1} g(x)\psi(x)dx. \end{aligned}$$

- If $f(x)$ is a polynomial of degree $\leq 2n+1$ then $g(x)$ is a polynomial of degree $\leq n$.
- Choose $\psi(x)$ to be proportional to the Legendre polynomial $\phi_{n+1}(x)$.
- Achieve this by selecting $\{x_i\}_{i=0}^{n}$ to be Gauss points: roots of ϕ_{n+1}.
- Obtain basic rule of precision $2n+1$, hence composite method of order $2(n+1)$.

Gauss points

- For general n

$$\begin{aligned} E(f) &= \int_{-1}^{1} f[x_0, x_1, \ldots, x_n, x](x - x_0)(x - x_1) \cdots (x - x_n) dx \\ &\equiv \int_{-1}^{1} g(x)\psi(x) dx. \end{aligned}$$

- If $f(x)$ is a polynomial of degree $\leq 2n + 1$ then $g(x)$ is a polynomial of degree $\leq n$.
- Choose $\psi(x)$ to be proportional to the Legendre polynomial $\phi_{n+1}(x)$.
- Achieve this by selecting $\{x_i\}_{i=0}^{n}$ to be Gauss points: roots of ϕ_{n+1}.
- Obtain basic rule of precision $2n + 1$, hence composite method of order $2(n + 1)$.

Gauss points

- For general n

$$\begin{aligned} E(f) &= \int_{-1}^{1} f[x_0, x_1, \ldots, x_n, x](x - x_0)(x - x_1)\cdots(x - x_n)dx \\ &\equiv \int_{-1}^{1} g(x)\psi(x)dx. \end{aligned}$$

- If $f(x)$ is a polynomial of degree $\le 2n + 1$ then $g(x)$ is a polynomial of degree $\le n$.
- Choose $\psi(x)$ to be proportional to the Legendre polynomial $\phi_{n+1}(x)$.
- Achieve this by selecting $\{x_i\}_{i=0}^{n}$ to be Gauss points: roots of ϕ_{n+1}.
- Obtain basic rule of precision $2n + 1$, hence composite method of order $2(n + 1)$.

Gaussian quadrature formulas

- In general

$$\begin{aligned}\phi_{j+1}(x) &= \frac{2j+1}{j+1}x\phi_j(x) - \frac{j}{j+1}\phi_{j-1}(x), \quad j \ge 1\\ a_j &= \frac{2(1-x_j^2)}{[(n+1)\phi_n(x_j)]^2}, \quad j = 0,1,\ldots,n,\\ E_n(f) &= \frac{(b-a)^{2n+3}((n+1)!)^4}{(2n+3)((2n+2)!)^2}f^{(2n+2)}(\xi).\end{aligned}$$

- $\phi_0(x) = 1,\ \phi_1(x) = x,\ \phi_2(x) = \frac{1}{2}(3x^2-1),\ \phi_3(x) = \frac{1}{2}(5x^3-3x),\ldots.$
- $n = 0:\ x_0 = 0; \qquad n = 1:\ x_i = \pm\sqrt{1/3}; \qquad n = 2:\ x_i = 0, \pm\sqrt{3/5}$.
- etc.

More about what makes this work

$$E(f) = \int_a^b (f(x) - p_n(x))dx = \int_a^b f[x_0, x_1, \ldots, x_n, x] \prod_{i=0}^{n} (x - x_i)dx.$$

- Suppose that $f(x)$ itself is a polynomial of degree m. If $m \leq n$ then

$$f[x_0, x_1, \ldots, x_n, x] = \frac{f^{(n+1)}(\xi)}{(n+1)!} = 0,$$

  no matter what the points $\{x_i\}$ (or ξ) are.
- Thus, any basic rule using $n+1$ points has precision of at least n.
- For instance, the trapezoidal rule has the minimal precision n, with $n = 1$.
- But **if** the $n+1$ abscissae can be chosen, then we have $n+1$ degrees of freedom to play with, so intuition suggests that precision can be inreased by $n+1$, from the basic n to $2n+1$.

More about what makes this work

$$E(f) = \int_a^b (f(x) - p_n(x))dx = \int_a^b f[x_0, x_1, \ldots, x_n, x] \prod_{i=0}^{n} (x - x_i)dx.$$

- Suppose that $f(x)$ itself is a polynomial of degree m. If $m \leq n$ then

$$f[x_0, x_1, \ldots, x_n, x] = \frac{f^{(n+1)}(\xi)}{(n+1)!} = 0,$$

  no matter what the points $\{x_i\}$ (or ξ) are.
- Thus, any basic rule using $n+1$ points has precision of at least n.
- For instance, the trapezoidal rule has the minimal precision n, with $n = 1$.
- But **if** the $n+1$ abscissae can be chosen, then we have $n+1$ degrees of freedom to play with, so intuition suggests that precision can be inreased by $n+1$, from the basic n to $2n+1$.

What makes this work cont.

- For a class of orthogonal polynomials $\phi_0(x), \phi_1(x), \ldots, \phi_{n+1}(x)$ we have

$$\int_a^b g(x)\phi_{n+1}(x)dx = 0$$

  for any polynomial $g(x)$ of degree $\leq n$.

- So, choose the points $x_0, x_1, \ldots, x_n$ as the zeros (roots) of a scaled and shifted Legendre polynomial $\phi_{n+1}(x)$. Then, for any polynomial $f(x)$ of degree $2n+1$ or less, the divided difference $f[x_0, x_1, \ldots, x_n, x]$ is a polynomial of degree n or less.

- Since we can write $\phi_{n+1}(x) = c_{n+1}\prod_{i=0}^{n}(x - x_i)$, by orthogonality the quadrature error $E(f)$ vanishes for any such polynomial f. So, the precision is $2n+1$.

Composite Gaussian quadrature

- Subdivide a given interval $[a, b]$ by a mesh (grid)

$$a = t_0 < t_1 < \cdots < t_r = b.$$

- Map Gauss points into each subinterval $[t_{i-1}, t_i]$, $t_i = t_{i-1} + h_i$.

$$\begin{aligned} I_f &\approx \sum_{i=1}^{r} \sum_{j=0}^{n} b_{i,j} f(t_{i,j}), \\ t_{i,j} &= t_{i-1} + \frac{h_i}{2}(1 + x_j), \\ b_{i,j} &= \frac{h_i}{2} a_j. \end{aligned}$$

- So, obtain a method of order $q = 2n + 2$ at the price of $(n + 1)r$ function evaluations.

Outline

- Basic rules
- Composite numerical integration
- Gaussian quadrature
- Adaptive quadrature
- *Multi-dimensional integration

Adaptive quadrature: writing a software package

- The user is required to supply $f(x),\ a,\ b$, and a tolerance, but certainly not h!
- In MATLAB the call is `Q = quad('f',a,b,tol)`. Our function should then return a value satisfying

$$|Q - I| \le \texttt{tol}.$$

- Want to divide given interval $[a, b]$ to subintervals (panels) $a = t_0 < t_1 < \cdots < t_r = b$ with $h_i = t_i - t_{i-1}$ such that the approximation Q_i for $I_i = \int_{t_{i-1}}^{t_i} f(x)dx$ satisfies

$$|Q_i - I_i| \le \frac{h_i}{b-a}\texttt{tol}, \quad i = 1, 2, \ldots, r.$$

- Use a divide and conquer approach: estimate error for panel i; if tolerance is not satisfied then divide it into two and repeat operation for each.

Error estimation: trapezoidal

- Recall

$$E(f) = E(f;h) = Kh^q + \mathcal{O}(h^{q+1}).$$

- Compute

$$\begin{aligned} R_1 &= \frac{h}{2}[f(a) + 2f(a+h) + \cdots + 2f(b-h) + f(b)], \\ R_2 &= \frac{h}{4}[f(a) + 2f(a+h/2) + 2f(a+h) + \cdots + 2f(b-h/2) + f(b)] \end{aligned}$$

- Then

$$\begin{aligned} I - R_1 &= (I - R_2) + (R_2 - R_1) \\ &\approx \frac{1}{4}(I - R_1) + (R_2 - R_1), \end{aligned}$$

hence

$$\begin{aligned} I - R_1 &\approx \frac{4}{3}(R_2 - R_1), \\ I - R_2 &\approx \frac{1}{3}(R_2 - R_1). \end{aligned}$$

Error estimation: Simpson

Likewise for Simpson's rule,

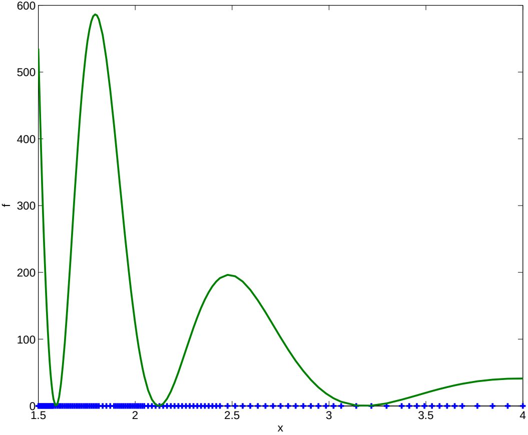
$$
\begin{aligned}
I - S_1 &\approx \frac{16}{15}(S_2 - S_1), \\
I - S_2 &\approx \frac{1}{15}(S_2 - S_1).
\end{aligned}
$$

This is an *a posteriori* error estimate, as opposed to our previous formulas which were *a priori*. So:

- For each subinterval, compute integral approximations for I_i using $h/2$ and h, and estimate error. If too large, then refine; otherwise, be happy and do nothing.
- See code in text (Section 15.4).
- Note that for Trapezoidal and Simpson, no additional computational overhead is entailed; not so for Gaussian quadrature.

Example

The figure shows integrand $f(x) = \frac{200}{2x^3 - x^2}\left(5\sin\left(\frac{20}{x}\right)\right)^2$ with the adaptively obtained quadrature mesh points along the x-axis. More points are concentrated where f varies more rapidly.



Outline

- Basic rules
- Composite numerical integration
- Gaussian quadrature
- Adaptive quadrature
- *Multi-dimensional integration

Multi-dimensional integration

- Consider

$$I = \int_{\Omega} f(\mathbf{x}) d\mathbf{x}$$

where $\Omega \subset \mathcal{R}^d$.

- Note that we can always break the domain Ω into a union of overlapping subdomains C_i and write

$$I = \sum_{i=1}^{N} \int_{C_i} f(\mathbf{x}) d\mathbf{x}.$$

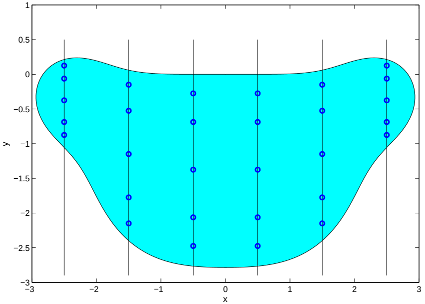
- Consider two types of problems in d dimensions:
    - those where d is small, e.g., $d = 2$ or $d = 3$;
    - those where d is much larger, e.g., $d = 100$.

When d is not large

- Geometric problems arise for non-square domains, but the situation is easier than in Section 11.6.
- If the integral can be written as an iterated integral, e.g., in 2D

$$I = \int_{l_0}^{u_0} \left[\int_{l_1(x)}^{u_1(x)} f(x,y)dy \right] dx,$$

then dimensional splitting can be done using 1D methods: for each x, integrate in y.



When d is large

- If $d = 100$, say, then even on the simplest "hypercube" domain with two points in each direction, obtain 2^{100} function evaluations: out of the question in practice.
- Instead use Monte Carlo methods, sampling the integrand randomly in Ω and averaging the computed values.
- Using N samples, the error is $\mathcal{O}(\sqrt{1/N})$, which is much better than nothing.