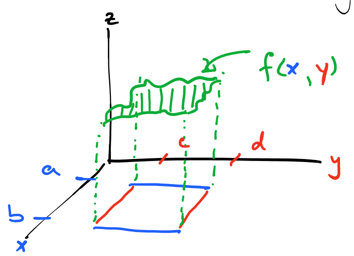
Multiple Integrals (2D, 3D, etc.)

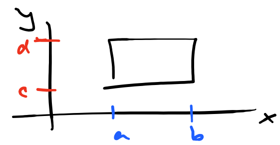
Thursday, October 15, 2020 4:54 PM

Goal: extend our 1D techniques for $\int_a^b f(x)\,dx$

to higher dimensions, like $\int_a^b \int_c^d f(x,y)\,dy\,dx$ or more generally $\int_{R \subseteq \mathbb{R}^d} f(\vec{x})\,d\vec{x}$



If we draw...

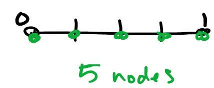


... this is just the domain. We're not finding its area

The extension is reasonably straightforward conceptually, but requires more computation. As the dimension d increases, the problem becomes more and more computationally difficult.

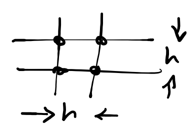
Why? Suppose I want a grid spacing of h in 1D

Ex. $h = 1/4$



and let my domain be $[0,1]$. Then I choose $n = \frac{1}{h}$ meaning I have $n+1$ gridpoints (i.e., $O(1/h)$ gridpoints)

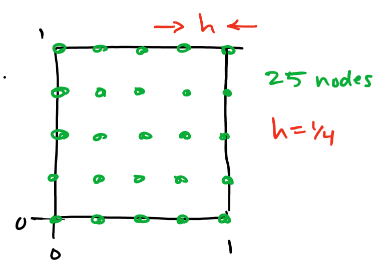
In 2D, if I still want a spacing of h,



and my domain is $\underbrace{[0,1] \times [0,1]}_{\text{aka } [0,1]^2}$

then I need $n = O(1/h^2)$ gridpoints

Ex.



In d-dimensions w/ domain $[0,1]^d$ I need $n = O(1/h^d)$ gridpoints

BAD!

1D, on $[0,1]$, $h = 1/9$ $\Rightarrow$ $n = 10$

10 D, on $[0,1]^{10}$, $h = 1/9$ $\Rightarrow$ $n = 10^{10}$ = 10 billion

Sometimes called the "curse of dimensionality" *

* many phenomenon are called this

Basic Technique

$$I = \int_a^b \underbrace{\int_c^d f(x,y)\,dy}_{g(x) := \int_c^d f(x,y)\,dy}\,dx \quad \text{(iterated integral)}$$

Just apply your preferred quadrature rule to $\int_a^b g(x)\,dx$

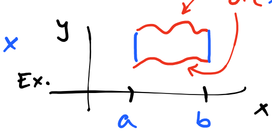
i.e., composite Simpson's, choose an even n, so $n+1$ gridpoints with spacing $h = \frac{b-a}{m}$, nodes $\{x_0, x_1, \dots, x_n\}$ $\quad x_i = x_0 + ih$

$$I = \frac{h}{3}\left(g(x_0) + 2\sum_{i=1}^{n/2-1} g(x_{2i}) + 4\sum_{i=1}^{n/2} g(x_{2i-1}) + g(x_n) \right)$$
$$- \frac{(b-a)h^4}{180}\,\frac{\partial^4}{\partial x^4} g(\xi)$$

Now the trick is that $g(x)$ is itself an integral, so use composite Simpson to evaluate it.

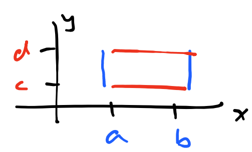
In practice (programming), this is <u>easy</u> because we just call our quadrature function. Writing it out by hand is very tedious

$g(x) = \int_c^d f(x,y)\,dy$ ... or it's even possible to let $c(x)$, $d(x)$ be functions of x



in software, this is easy, and making the # of y nodes $m+1$ depend on the x value is also easy.

But for writing it out by hand (so we can see <u>error terms</u>), we'll assume c, d are constants



Let's just do a bit of it

$$I = \frac{h}{3}\Big(\underbrace{\overbrace{g(x_0)}^{A}} + 2\Sigma \dots + 4\Sigma \dots + \dots \Big) - \frac{(b-a)h^4}{180}\,\frac{\partial^4}{\partial x^4} g(\xi)$$

Choose an even m

form nodes $y_0, y_1, \dots, y_m$ with spacing $k = \frac{d-c}{m}$

$$= \frac{h}{3}\frac{k}{3}\left(\Big[\overbrace{f(x_0,y_0) + 2\sum_{j=1}^{m/2-1} f(x_0,y_{2j}) + 4\sum_{j=1}^{m/2} f(x_0,y_{2j-1}) + f(x_0,y_m)}^{A^{(1)}} \Big] \right.$$

$$+ 2\cdot{\textstyle\sum}'[\ldots] + 4\cdot{\textstyle\sum}'[\ldots]\Big) - \underbrace{\frac{h}{3}\frac{(d-c)k^4}{180}\frac{\partial^4}{\partial y^4}f(x_0,\eta_0)}_{A^{(2)}} - \frac{(b-a)h^4}{180}\frac{\partial^4}{\partial x^4}g(\xi)$$

$A = A^{(1)} + A^{(2)}$

error terms get fairly messy.

If $f \in C^4$ can show via IVT and weighted MVT that error looks like

$$E = -\frac{(d-c)(b-a)}{180}\left(h^4 \frac{\partial^4 f}{\partial x^4}(\xi,\eta) + k^4 \frac{\partial^4 f}{\partial y^4}(\hat{\xi},\hat{\eta}) \right)$$

Can you do this in 3D?

yes, just repeat ... get 3 nested "for" loops

Can you do this with Gaussian quadrature?

yes, just think of it with $g(x) = \int_c^d f(x,y)\,dy$

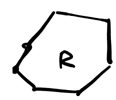
Can you do this for non-rectangular regions?

yes, see above.

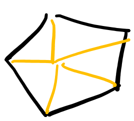
What are alternatives?

(1) To integrate $\iint_R f(x,y)\,dA$ where R is a convex polygon

you can break region into triangles



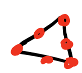
via triangulation

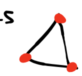


(Ex. "Triangle", winner of 2003 Wilkinson prize)

Used extensively for finite-element PDE simulations

Then on a triangle we can interpolate w/ a 2D polynomial

(e.g., add more internal nodes  and use a higher-degree polynomial,

or stick to vertices  and use a linear 2D polynomial)

3 nodes

$\underbrace{ax + by + c}_{3\text{ parameters}}$

(2) Monte Carlo "MC"

Statistical technique that generates a $\underbrace{\text{random variable}}_{\text{r.v.}}$ I_N

such that the r.v.'s mean is the correct integral,

$$\mathbb{E}[I_N] = I, \quad \text{and} \quad \mathrm{Var}[I_N] \to 0 \text{ as } N \to \infty$$

In small dimensions, this is much much less accurate than quadrature. But in large dimensions, quadrature is infeasible, so MC is the only game in town.

(well, MC and its generalizations, like quasi-Monte Carlo)

Summary

① See **demo code** !

② Integration in high-dimensions is much harder than in low-dimensions