NUR A lecture 4

In this lecture:

Discretization
(Extended) trapezoid rule
(Extended) Simpson's rule
Higher orders
Open (extended) formulas/midpoint
Romberg integration and errors
Richardson extrapolation
Handling improper integrals
Integrating in log vs linear space
Basics of Gaussian quadrature
Multidimensional integration

Numerical Recipes for Astrophysics A

Lecture 4

Questions about last week?

Integrating functions

NR Ch 4

Integrating functions

x_i $f(x_i) \equiv f_i$

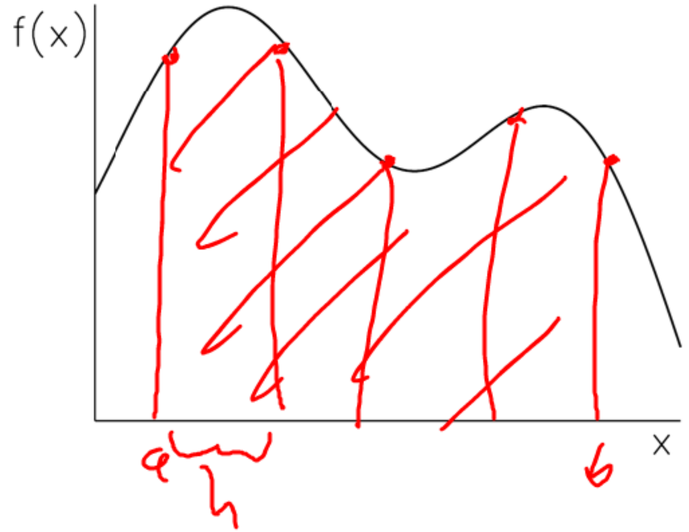
- Discretize and evaluate

  N intervals $\Rightarrow$ $N+1$ points (x_i, f_i)

  If equally spaced: step $h = \dfrac{b-a}{N}$

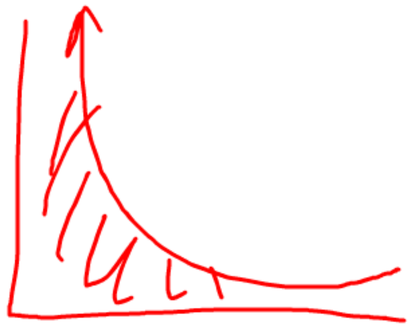
  $f(x)$ may be expensive: keep N low

  Higher N: more accurate, unless N is so high/h so small that round-off dominates



- Undefined $f(x)$/divergence

  Able to not evaluate $f(x)$ at certain x

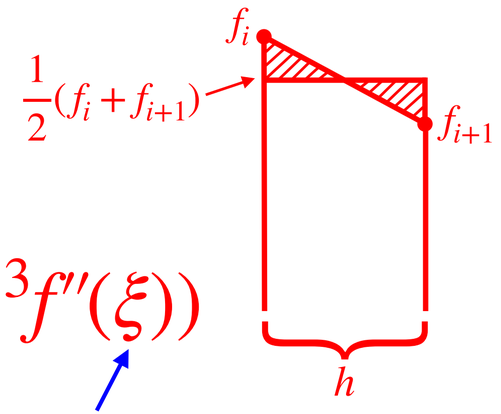


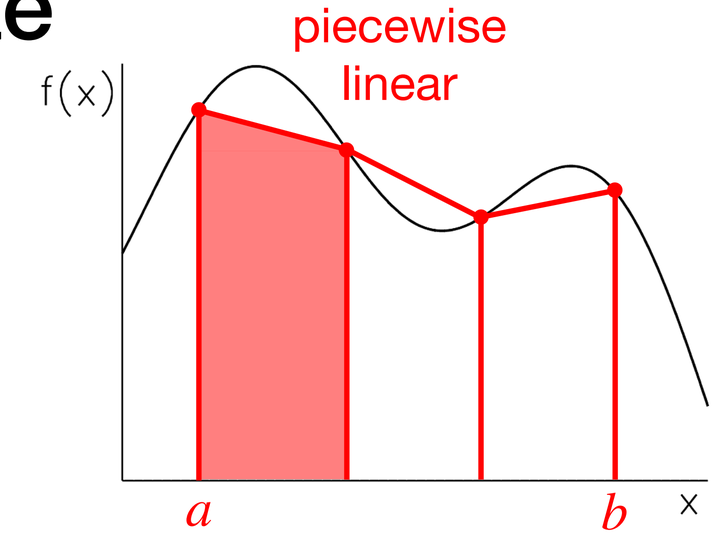
Equally spaced abscissae

- Trapezoid rule and extension

$$\int_{x_i}^{x_{i+1}} f(x)\,\mathrm{d}x = \frac{h}{2}(f_i + f_{i+1}) + \mathcal{O}(h^3 f''(\xi))$$

some unknown point in interval



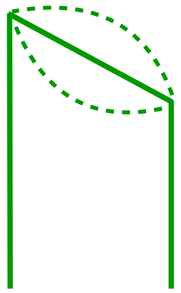


Extended trapezoid:

$$\int_a^b f(x)\,\mathrm{d}x = \sum_{i=0}^{N-1} \int_{x_i}^{x_{i+1}} f(x)\,\mathrm{d}x = h\left(\frac{f_0}{2} + \sum_{i=1}^{N-1} f_i + \frac{f_N}{2}\right) + \mathcal{O}\left(\frac{(b-a)^3}{N^2} f''\right)$$

$$\left(\frac{f_0}{2} + \frac{f_1}{2}\right) + \left(\frac{f_1}{2} + \frac{f_2}{2}\right) + \ldots$$

$\propto Nh^3$

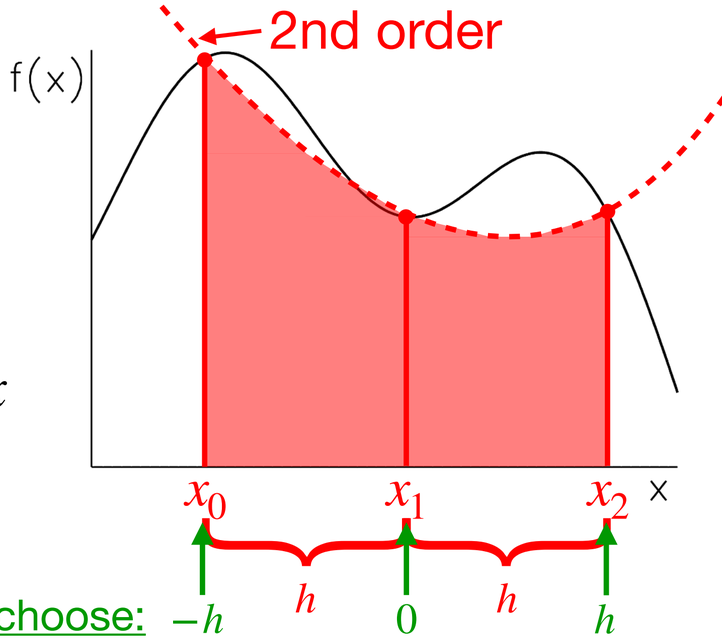


(Note that trapezoid always underestimates concave-down functions, and always overestimates concave-up ones)

all error terms are even powers of N

$$\left(\frac{1}{N^2}, \frac{1}{N^4}, \frac{1}{N^6}, \cdots\right)$$

Equally spaced abscissae



- Higher order: Simpson's rule

$$g(x) = ax^2 + bx + c \;\Rightarrow\; G(x) = \frac{1}{3}ax^3 + \frac{1}{2}bx^2 + cx$$

$$\int_{x_0}^{x_2} f(x)\,\mathrm{d}x \approx G(x_2) - G(x_0) = \frac{2}{3}ah^3 + 2ch$$

$$(1)\quad g(x_0) = f_0 = ah^2 - bh + c$$

$$(2)\quad g(x_1) = f_1 = c$$

$$(3)\quad g(x_2) = f_2 = ah^2 + bh + c$$

$$(1) + (3) \;\Rightarrow\; a = \frac{f_0 - 2f_1 + f_2}{2h^2}$$

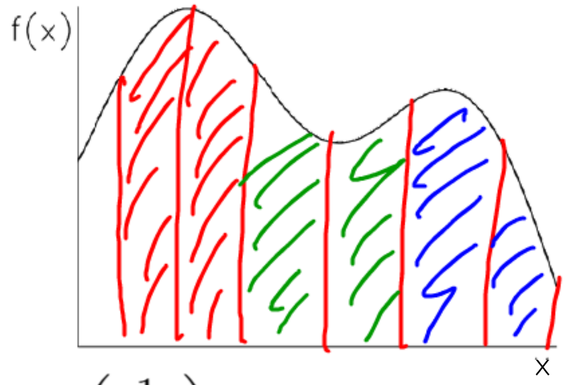
$$(3) - (1) \;\Rightarrow\; b = \frac{f_2 - f_0}{2h}$$

$$\int_{x_0}^{x_2} f(x)\,\mathrm{d}x = \frac{h}{3}(f_0 + 4f_1 + f_2) + \mathcal{O}(h^5 f^{(4)}) \;\Rightarrow$$

($h^4, h^6, \ldots$ cancel)

accurate to 3rd order! (symmetry)

Equally spaced abscissae



- Extended Simpson's rule

$$\int_{x_0}^{x_\mathrm{N}} f(x)\,\mathrm{d}x = \frac{h}{3}\left(f_0 + 4f_1 + 2f_2 + 4f_3 + \cdots + 2f_{\mathrm{N}-2} + 4f_{\mathrm{N}-1} + f_\mathrm{N}\right) + \mathcal{O}\left(\frac{1}{N^4}\right)$$

$$1 + 4 + 1$$

$$1 + 4 + 1$$

$$\left(+ \sim \frac{1}{N^6} + \sim \frac{1}{N^8} + \ldots\right)$$

- Higher order still

(still) 3rd order

no cancellation

Simpson's $\frac{3}{8}$ rule:

$$\int_{x_0}^{x_3} f(x)\,\mathrm{d}x = \frac{3h}{8}(f_0 + 3f_1 + 3f_2 + f_3) + \mathcal{O}(h^5 f^{(4)})$$

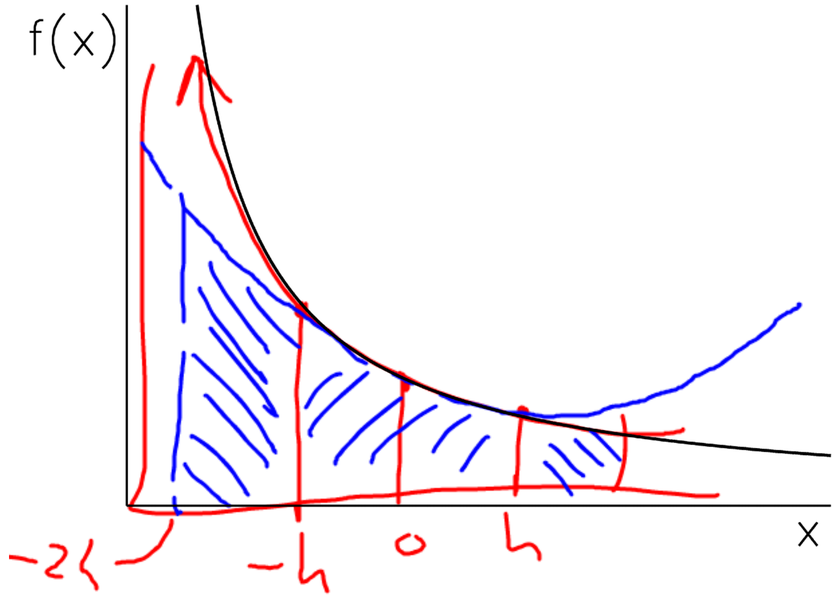
cancellation again

Boole's rule:

$$\int_{x_0}^{x_4} f(x)\,\mathrm{d}x = \frac{2h}{45}(7f_0 + 32f_1 + 12f_2 + 32f_3 + 7f_4) + \mathcal{O}(h^7 f^{(6)})$$

(not "Bode's"!)

Open formulas



- Deriving open formulas

f not evaluated at endpoints

$$\int_{-2h}^{2h} f(x)\,\mathrm{d}x \approx \frac{4}{3}h(2f_1 - f_2 + 2f_3)$$: Milne's rule

Only extrapolate to the left: $$\int_{-2h}^{h} f(x)\,\mathrm{d}x \approx \frac{h}{12}(23f_1 - 16f_2 + 5f_3)$$

- Open extended formulas

only open at endpoints, closed in between

- Semi-open

only open at a or b

Open formulas

- Second-order extended (Simpson's)

$$\int_{x_0}^{x_N} f(x)\,\mathrm{d}x = \frac{h}{3}\left(\frac{27}{4}f_1 + \frac{13}{4}f_3 + 4f_4 + 2f_5 + 4f_6 + \cdots + 2f_{N-5} + 4f_{N-4} + \frac{13}{4}f_{N-3} + \frac{27}{4}f_{N-1}\right) + \mathcal{O}\left(\frac{1}{N^4}\right)$$

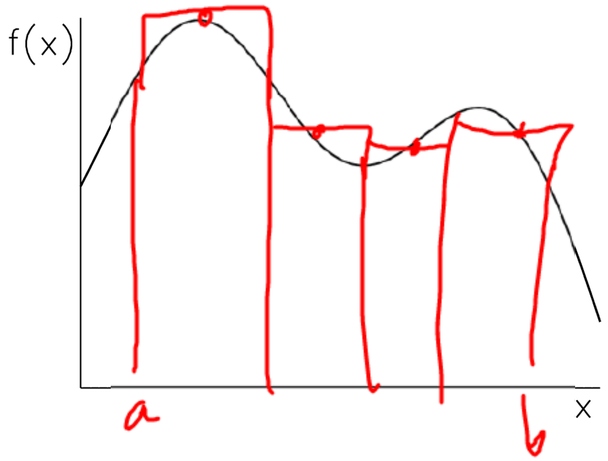
✓ no f_0 — no f_2! — "Simpson's pattern" — no f_{N-2}, no f_N

- Extended midpoint rule = middle Riemann sum

$$\int_{x_0}^{x_N} f(x)\,\mathrm{d}x = h\left(f_{1/2} + f_{3/2} + f_{5/2} + \cdots + f_{N-3/2} + f_{N-1/2}\right) + \mathcal{O}\left(\frac{1}{N^2}\right)$$

open equivalent of the trapezoid rule

(also: only even terms $\sim \frac{1}{N^{2k}}$)



Romberg integration

- Combining trapezoids N intervals: $\int_a^b f(x)\,\mathrm{d}x \approx S_0$ (using $h_0 = h = (b-a)/N$)

  ($= N+1$ evaluations; ext. trapezoid result for $h = h_0$)

Now with $N_1 = 2N$ intervals: $\int_a^b f(x)\,\mathrm{d}x \approx S_1$ (using $h_1 = h_0/2$; h_i uses $h_0/2^i$)

Error for S_0: $\mathcal{O}\left(\frac{1}{N^2}\right)$; error for S_1: $\mathcal{O}\left(\frac{1}{N_1^2}\right) = \frac{1}{4}\mathcal{O}\left(\frac{1}{N^2}\right)$

$$S = \frac{4}{3}S_1 - \frac{1}{3}S_0$$

$$= \frac{4}{3}\frac{h}{2}\left(\frac{f_0}{2} + \sum_{i=1}^{2N-1} f_i + \frac{f_{2N}}{2}\right) + \frac{4}{3}\mathcal{O}\left(\frac{1}{4N^2}\right) - \frac{1}{3}h\left(\frac{f_0}{2} + \sum_{i=1}^{N-1} f_{2i} + \frac{f_{2N}}{2}\right) - \frac{1}{3}\mathcal{O}\left(\frac{1}{N^2}\right)$$

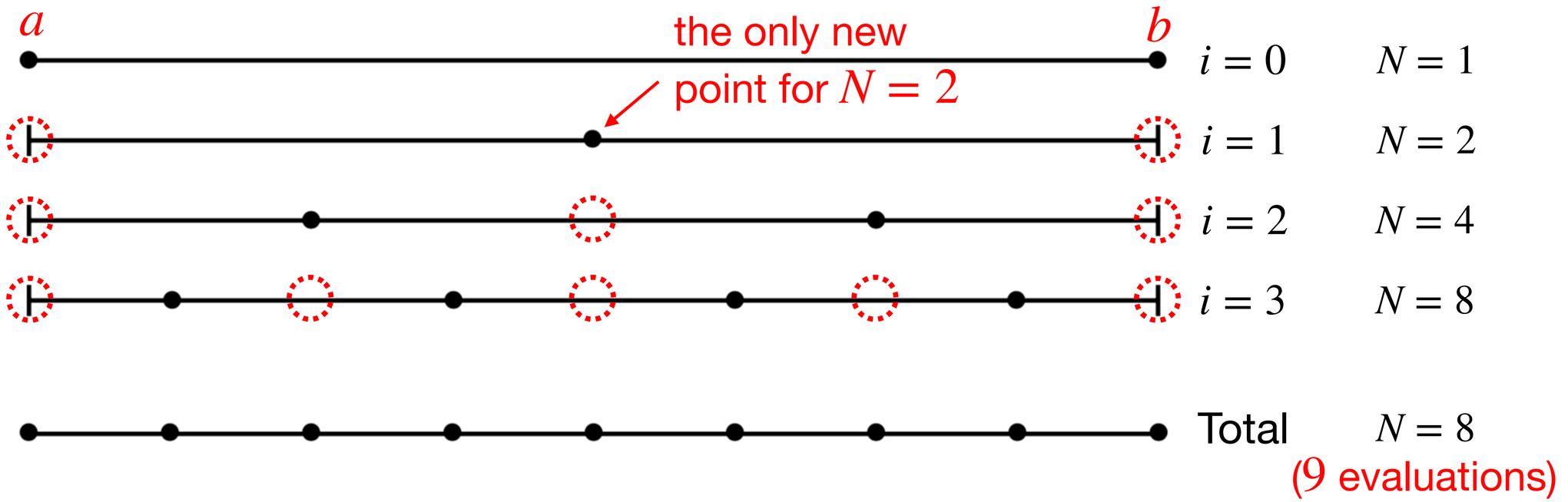
(every other point)

$$= \frac{h_1}{3}\left(f_0 + 4f_1 + 2f_2 + 4f_3 + \ldots + 2f_{N_1-2} + 4f_{N_1-1} + f_{N_1}\right) + \mathcal{O}\left(\frac{1}{N^4}\right)$$

← exactly Simpson's rule! (except more efficient and with less code)

But why stop there?

Romberg integration



uses $h_{i+1} = \frac{h_i}{2}$

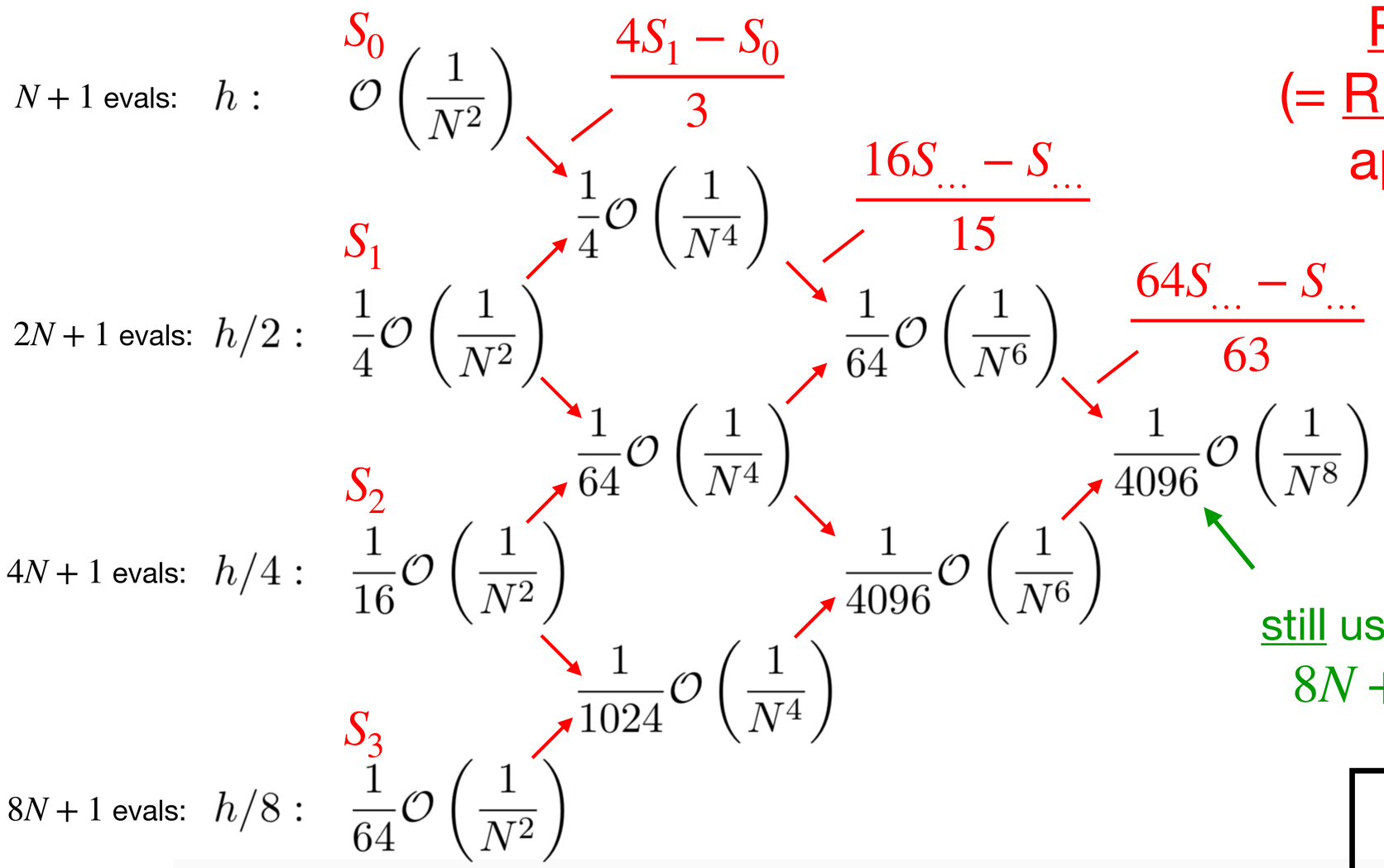
$h_i \to \frac{h_i}{2}$

contributions of new points

space between new points ($= h_i$)

$$S_{i+1} = \frac{1}{2} S_i + \frac{b-a}{N_{i+1}} \sum_j f_j = \frac{1}{2} \left(S_i + \Delta_{i+1} \sum_j f_j \right)$$

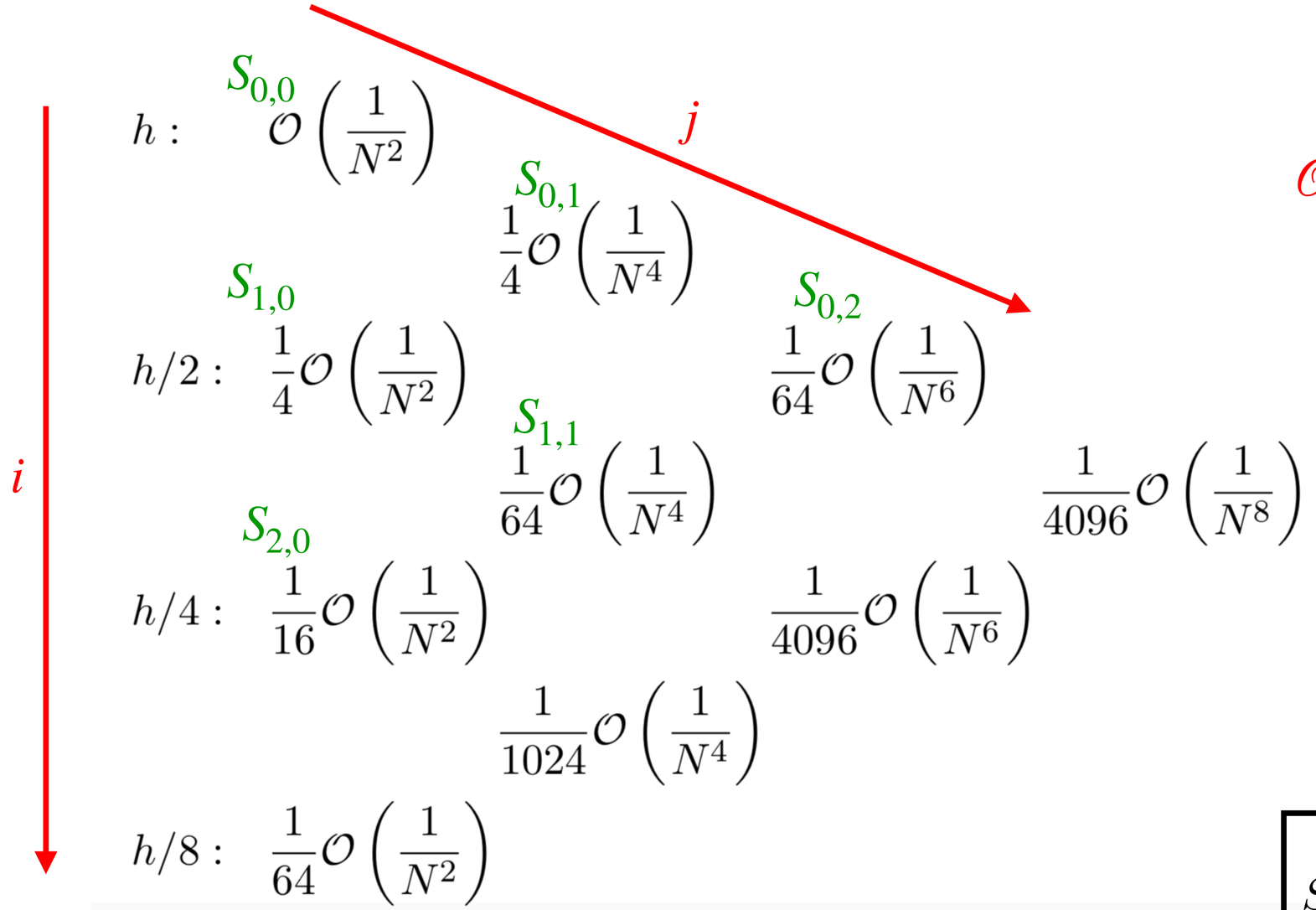
Analogue to Neville's algorithm



Romberg integration
(= Richardson extrapolation applied to integration/ trapezoid)

still uses only the same $8N + 1$ evaluations!

Analogue to Neville's algorithm



$$\mathcal{O}\left(\frac{1}{N^k}\right) \rightarrow \mathcal{O}\left(\frac{1}{N^{k+2}}\right)$$

with every step

$$S_{i,j} = \frac{4^j S_{i+1,j-1} - S_{i,j-1}}{4^j - 1}$$

Romberg integration

1. Decide on an initial step size h and an “order” m, for example $h = b - a$ and $m = 5$.
2. Make an array r of size m for your estimates, up to step size $h/2^{m-1}$; set these to 0.
3. Calculate the initial estimate r_0: for $h = b - a$, it’s $r_0 = \frac{1}{2}h[f(a) + f(b)]$. (trapezoid for $N = 1$)
4. Set $N_\mathrm{p} = 1$, loop over i from 1 through $m - 1$: (fill out first column)
5. Add in new function evaluations: $r_i = 0$, $\Delta = h$, $h = \frac{1}{2}h$, $x = a + h$, then N_p times do: $r_i = r_i + f(x)$, $x = x + \Delta$.
6. Now combine the new points with the already calculated ones for an estimate for $h/2^i$: $r_i = \frac{1}{2}[r_{i-1} + \Delta r_i]$, then double N_p.
7. Finish the loop over i. We now have the initial “column” of estimations.
8. Reset N_p to 1, start a new loop over i from 1 through $m - 1$: ← columns

   now do weighted combinations
9. Upgrade the previous results by combining them: multiply N_p by 4, then for j from 0 up to $m - i$: $r_j = [N_\mathrm{p} r_{j+1} - r_j]/(N_\mathrm{p} - 1)$.

   rows
10. Finish the loops over j and i; r_0 now contains the best estimate for the integral.

    error $\approx |r_0 - r_1|$

The magic in practice: error function

$$\mathrm{erf}(1) = \frac{2}{\sqrt{\pi}} \int_0^1 e^{-x^2}\,\mathrm{d}x \approx 0.8427007929$$ (correct answer to 10 decimals)

$N = 1,\ h = 1$:

$N = 2,\ h = 1/2$:

$N = 4,\ h = 1/4$:

$N = 8,\ h = 1/8$:

$N = 16,\ h = 1/16$:

The magic in practice: error function

$$\mathrm{erf}(1) = \frac{2}{\sqrt{\pi}} \int_0^1 e^{-x^2}\,\mathrm{d}x \approx 0.8427007929$$

all from trapezoid

$N = 1,\ h = 1:\quad 0.7717433323$

$N = 2,\ h = 1/2:\quad 0.8252629556$

$N = 4,\ h = 1/4:\quad 0.8383677774$

$N = 8,\ h = 1/8:\quad 0.8416192212$

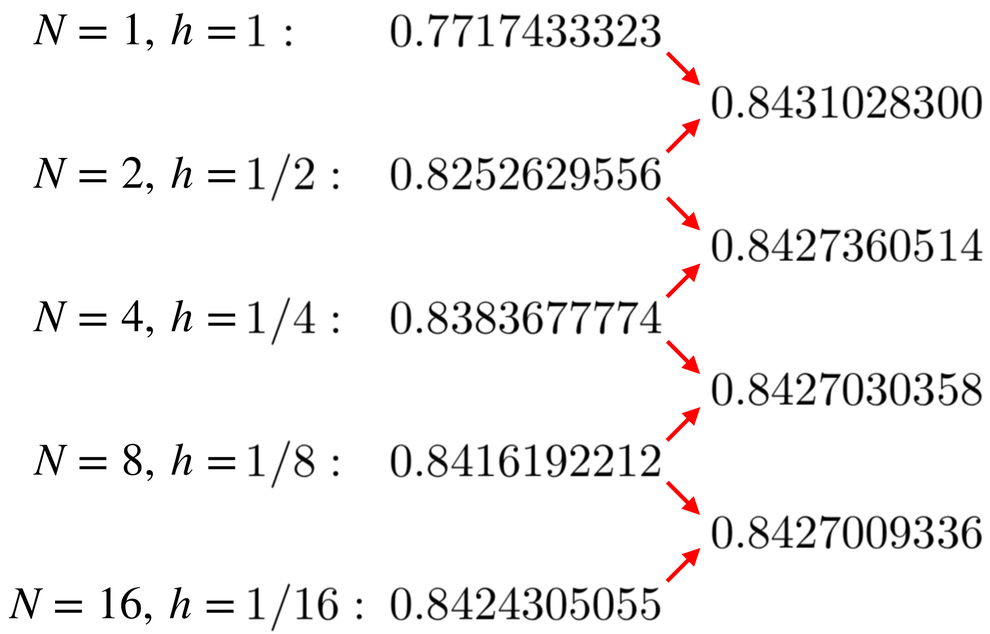
$N = 16,\ h = 1/16:\quad 0.8424305055$

16+1 unique evaluations

Now we combine with worse estimates that use a subset of the same information (f_i) and get out something much better!

The magic in practice: error function

$$\mathrm{erf}(1) = \frac{2}{\sqrt{\pi}} \int_0^1 e^{-x^2}\, \mathrm{d}x \approx 0.8427007929$$



Take these numbers and try it yourself!

Note: machine precision/round-off still limits us

The magic in practice: error function

$$\mathrm{erf}(1) = \frac{2}{\sqrt{\pi}} \int_0^1 e^{-x^2}\,\mathrm{d}x \approx 0.8427007929$$

magic

(disclaimer: not actually magic, we're exploiting that $f(x)$ is smooth and has predictable errors)

$N=1,\ h=1:$	0.7717433323			
		0.8431028300		
$N=2,\ h=1/2:$	0.8252629556		0.8427115995	
		0.8427360514		0.8427006639
$N=4,\ h=1/4:$	0.8383677774		0.8427008348	
		0.8427030358		0.8427007928
$N=8,\ h=1/8:$	0.8416192212		0.8427007934	
		0.8427009336		
$N=16,\ h=1/16:$	0.8424305055			

$\varepsilon \sim 3 \times 10^{-4}$

$\varepsilon \sim 4 \times 10^{-10}$

(6 orders of magnitude better for ZERO extra f_i!)

error equivalent to trapezoid with $h \approx 1/16384$

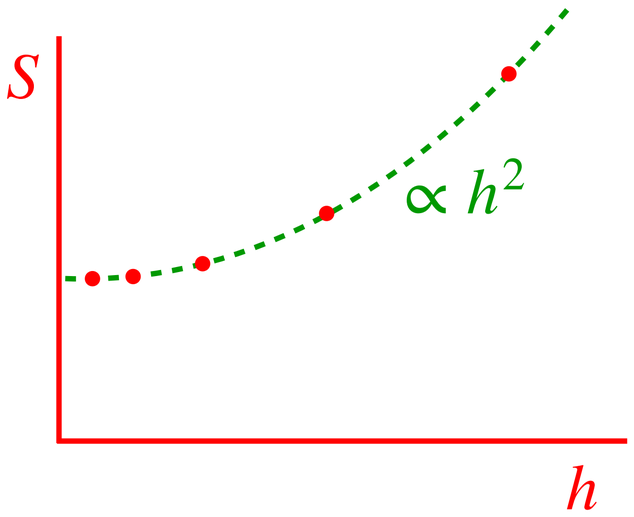
$\Rightarrow \sim 1000 \times$ more evaluations!

(Richardson) Extrapolation

except now explicitly instead of cancelling error terms

- Explicitly extrapolating to $h = 0$

$$(h_i^2, S_i) \rightarrow (0, S)$$ ~Linear in h^2



extrapolate results to $h = 0$ limit based on approximations/estimates for different h_i

- Where does this fail?

Same as Romberg: when f is not smooth/"predictable" $\Rightarrow$ (scaling of) the error is unknown

Improper integrals

- Improper but not infinite or ill-defined:
  - ✓ The integrand is limited and has finite limits but cannot be evaluated at one of the limits (i.e. $\sin x/x$ at $x = 0$). ← open method
  - ? The upper or lower limit is $\pm\infty$.
  - ✓ It has an integrable singularity at either limit (e.g. $x^{-1/2}$ at $x = 0$). ← open method
  - ? It has an integrable singularity at some known place between the limits.
  - ✗ It has an integrable singularity at some unknown place between the limits.

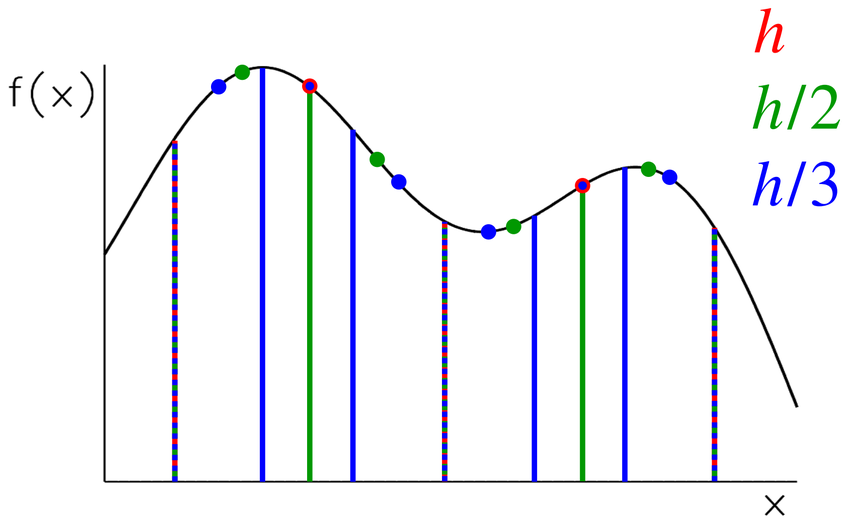
- Extended midpoint Romberg

Just replace trapezoid by midpoint ⇒ open!

But: instead of $N \to 2N \to 4N \to \ldots$

we have to do $N \to 3N \to 9N \to \ldots$

in order to re-use previous evaluations



(so also $h/3$ instead of $h/2$, 9^j instead of 4^j, etc)

Change of variables

= rewrite integral

- Limits to infinity

Example: substitute $x = \frac{1}{t}$ (when $ab > 0$ and $f(x)$ decreases faster than x^{-2} as $x \to \pm\infty$)

$$\Rightarrow \int_a^b f(x)\,\mathrm{d}x = \int_{1/b}^{1/a} \frac{1}{t^2} f\left(\frac{1}{t}\right)\,\mathrm{d}t$$

- Slow power-law divergence

Example: substitute $x = a + t^{1/(1-\gamma)}$ (when $f(x)$ diverges as $(x-a)^{-\gamma}$ with $0 \le \gamma < 1$)

$$\Rightarrow \int_a^b f(x)\,\mathrm{d}x = \int_0^{(b-a)^{1-\gamma}} t^{\gamma/(1-\gamma)} f\left(a + t^{1/(1-\gamma)}\right)\,\mathrm{d}t$$

- Many implementations exist! E.g. numpy/scipy

Summary: make substitutions not to find the primitive, but to alleviate infinite ranges and/or infinities

Breaking up the integral

- Splitting the limits

don't evaluate at $a < m < b$ (e.g. singularity at $x = m$)

$$\int_a^b f(x)\,\mathrm{d}x = \int_a^m f(x)\,\mathrm{d}x + \int_m^b f(x)\,\mathrm{d}x$$

+ use (semi-)open method

$$\left(= \int_a^{m_0} f(x)\,\mathrm{d}x + \int_{m_0}^{m_1} f(x)\,\mathrm{d}x + \cdots + \int_{m_N}^{b} f(x)\,\mathrm{d}x\right)$$

E.g. $[a, b] = [-5, \infty) \rightarrow [-5, 2],\ [2, \infty)$

Note: we can use a different N and/or method for each sub-integral!

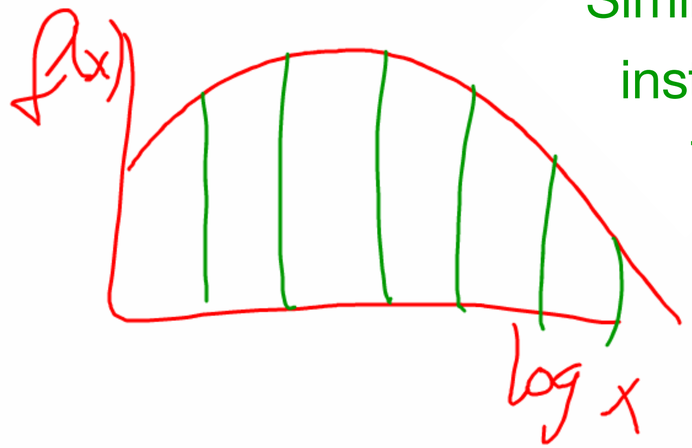
- Basics of adaptive step sizes

Linear vs logarithmic

- Range, space and measure

$$\mathrm{d}x \to \mathrm{d}\ln x$$
$$x \to e^t$$
$$x \to \ln t$$

- Appropriate transformations



Similarly, we could return $\ln f_i$ instead of f_i (as long as we transform back later)

Fewer evaluations for a given accuracy by choosing spaces/measures that match $f(x)$

Examples: $t = \ln x \Rightarrow \dfrac{\mathrm{d}\ln x}{\mathrm{d}x} = \dfrac{1}{x} \Rightarrow \mathrm{d}x = x\,\mathrm{d}\ln x = x\,\mathrm{d}t \Rightarrow \displaystyle\int_a^b f(x)\,\mathrm{d}x = \int_{\ln a}^{\ln b} f(e^t)e^t\,\mathrm{d}t$

or: $t = e^{-x} \Rightarrow \displaystyle\int_a^\infty f(x)\,\mathrm{d}x = \int_0^{e^{-a}} f(-\ln t)\frac{1}{t}\,\mathrm{d}t$

Gaussian quadrature

- Additional freedom: unequal spacing

  e.g. polynomial

  Say $f(x) = g(x)W(x)$ with $g(x)$ well-behaved, $W(x)$ troublesome

  E.g. $f(x) = \dfrac{g(x)}{\sqrt{1-x^2}}, \quad f(x) = g(x)\sin x$

  $\Rightarrow$ Choose x_j and weights w_j such that:

  $$\int f(x)\,\mathrm{d}x \approx \sum w_j g(x_j)$$

  $N + N = 2N$ degrees of freedom (x_j & w_j)

  $\Rightarrow$ Exact if g is a polynomial with degree $< 2N$

- Convergence

  Exponential convergence if x_j and w_j match $W(x)$ (but deriving these is very difficult)

Gaussian quadrature

- Default rule: Gauss-Legendre $W(x) = 1$
  - Gauss-Chebyshev: $W(x) = \left(1 - x^2\right)^{-1/2}$
  - Gauss-Laguerre: $W(x) = \exp(-x)$
  - Gauss-Hermite: $W(x) = \exp\left(-x^2\right)$

- w/x-generation in NumPy

  Uses derived rules (hard!) or hard-coded coefficients (relatively recent alternatives: root-finding algorithms and asymptotic formulas)

  See e.g. scipy.integrate.quadrature

Multidimensional integration e.g. $\int\int f(x,y)\,\mathrm{d}x\mathrm{d}y$

- Reducing the dimensionality Rewrite integral, exploit symmetry, etc

  E.g. surface of sphere is 3D in x, y, z, but 2D in ϕ, θ

  Otherwise a lot of f_i are needed (exponential with dimension!)

- Function evaluation = integral

  E.g. $\int f(x)\,\mathrm{d}x$ with $f_i = \int g(x_i, y)\,\mathrm{d}y$ (etc)

  Note: we're again free to use different methods or N for each (sub)integral

- Pre-calculation and interpolation

  If "inner" integral is expensive, consider pre-calculating it (e.g. on grid (x_i, y_i))
  and interpolate its value to get evaluations for the "outer" integral(s)
  (less accurate, but much faster)

Next week:
Differentiation and random numbers