UQ for high dimensional problems

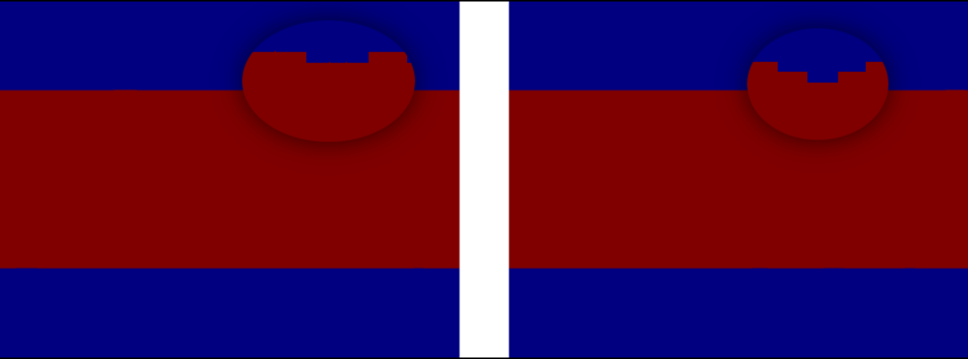
with a view towards hyperbolic conservation laws

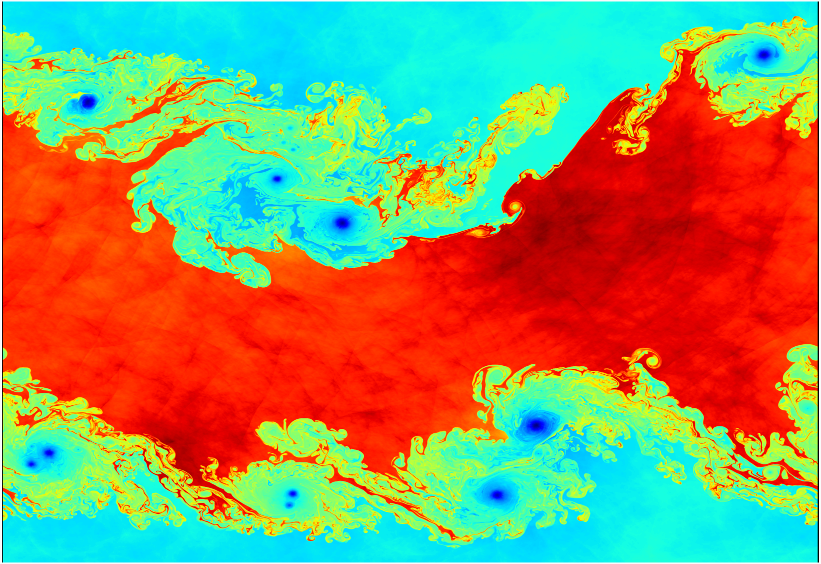
Kjetil Olsen Lye

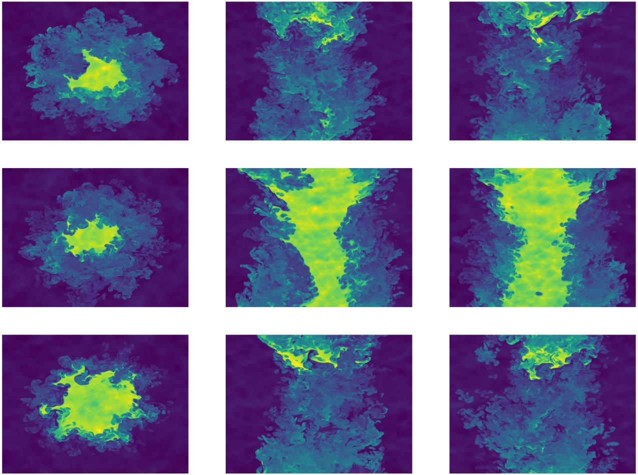
ETH Zurich

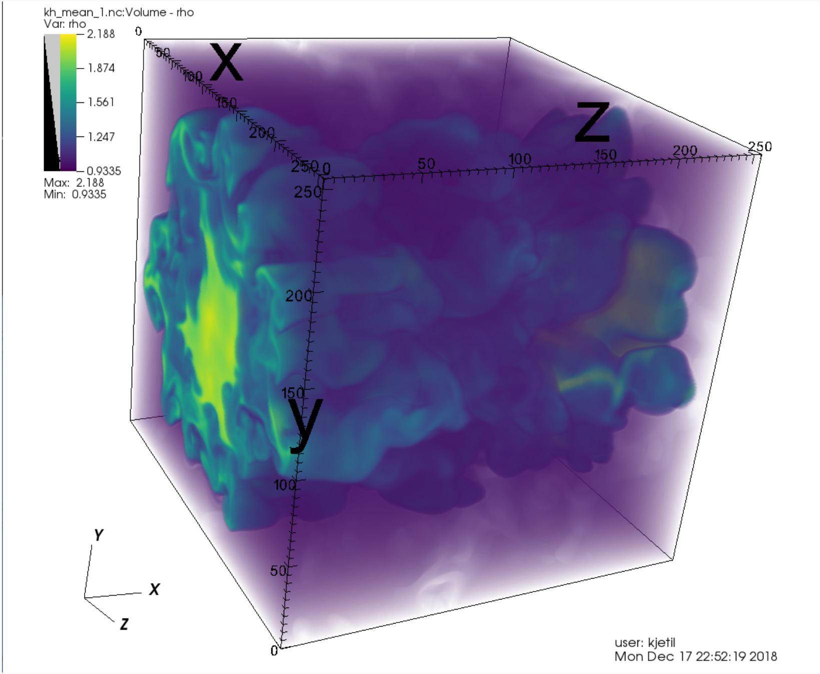
About me

- PhD student at ETH Zürich
- Research interests:
    - Non-linear hyperbolic equations
    - Uncertainty quantifications
    - (Machine learning)








kh_mean_1.nc:Volume - rho
Var: rho
2.188
1.874
1.561
1.247
0.9335
Max: 2.188
Min: 0.9335
x
y
z
0
50
100
150
200
250
Y
X
Z
user: kjetil
Mon Dec 17 22:52:19 2018

Introduction

Uncertainty quantifications

- $\mathbf{y} \in [0,1]^d$ uncertain parameters

Uncertainty quantifications

- $\mathbf{y} \in [0,1]^d$ uncertain parameters
- $Q(\mathbf{y})$ quantity of interest

Uncertainty quantifications

- $\mathbf{y} \in [0,1]^d$ uncertain parameters
- $Q(\mathbf{y})$ quantity of interest

Questions:

- Expectation of Q $(= \mathbb{E}(Q))$?

Uncertainty quantifications

- $\mathbf{y} \in [0,1]^d$ uncertain parameters
- $Q(\mathbf{y})$ quantity of interest

Questions:

- Expectation of Q $(= \mathbb{E}(Q))$?
- Variance of Q $(= \mathrm{Var}(Q))$?

Uncertainty quantifications

- $\mathbf{y} \in [0,1]^d$ uncertain parameters
- $Q(\mathbf{y})$ quantity of interest

Questions:

- Expectation of Q $(= \mathbb{E}(Q))$?
- Variance of Q $(= \mathrm{Var}(Q))$?
- PDF, CDF, ...

Focus of talk

Compute

$$\int_{[0,1]^d} f(\mathbf{y})\, d\mathbf{y}$$

Focus of talk

Compute

$$\int_{[0,1]^d} f(\mathbf{y})\ d\mathbf{y}$$

where

- f is expensive to compute

Focus of talk

Compute

$$\int_{[0,1]^d} f(\mathbf{y}) \, d\mathbf{y}$$

where

- f is expensive to compute
- d is large

Focus of talk

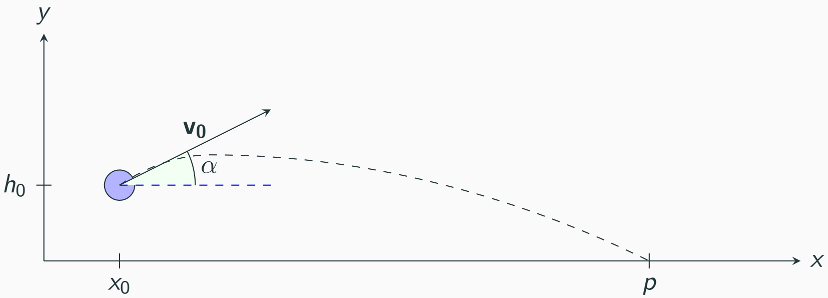
Compute

$$\int_{[0,1]^d} f(\mathbf{y})\, d\mathbf{y}$$

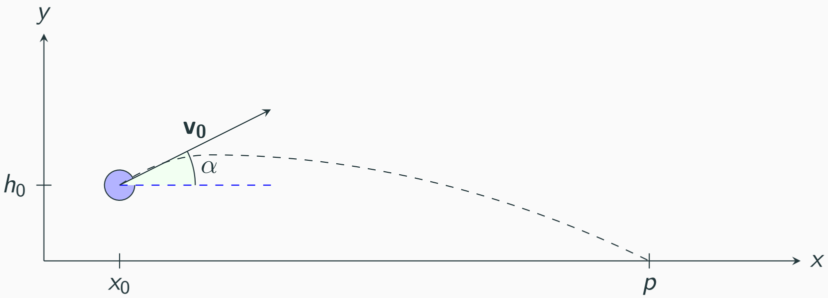
where

- f is expensive to compute
- d is large
- f could be ill-behaved (eg. discontinuous)

Example



Example



$$p(h_0) = v_0 \cos(\alpha) \frac{v_0 \sin(\alpha) + \sqrt{(v_0 \sin(\alpha))^2 + 2gh_0}}{g} + x_0$$

Interesting quantities

Assume $h_0 \sim \mathcal{U}[0, 1]$

Interesting quantities

Assume $h_0 \sim \mathcal{U}[0, 1]$

- Expectation $\mathbb{E}(p) = \int_0^1 p(h_0)\, dh_0$

Interesting quantities

Assume $h_0 \sim \mathcal{U}[0, 1]$

- Expectation $\mathbb{E}(p) = \int_0^1 p(h_0)\ dh_0$
- Variance $\mathrm{Var}(p) = \mathbb{E}((p - \mathbb{E}(p))^2) = \int_0^1 (p(h_0) - \mathbb{E}(p(h_0)))^2\ dh_0$

Interesting quantities

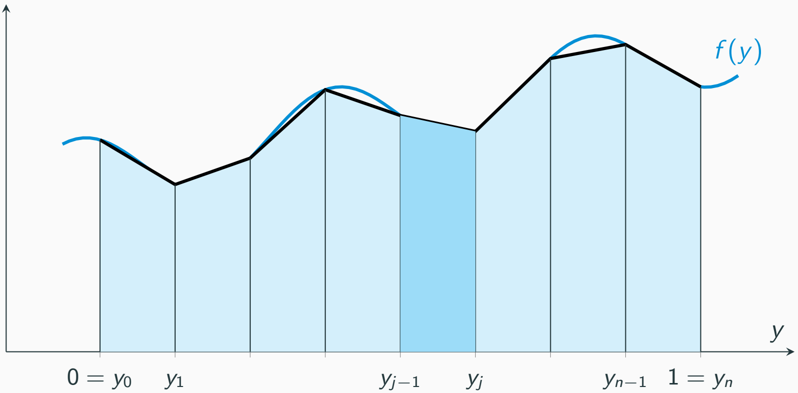
Assume $h_0 \sim \mathcal{U}[0, 1]$

- Expectation $\mathbb{E}(p) = \int_0^1 p(h_0)\ dh_0$
- Variance $\mathrm{Var}(p) = \mathbb{E}((p - \mathbb{E}(p))^2) = \int_0^1 (p(h_0) - \mathbb{E}(p(h_0)))^2\ dh_0$

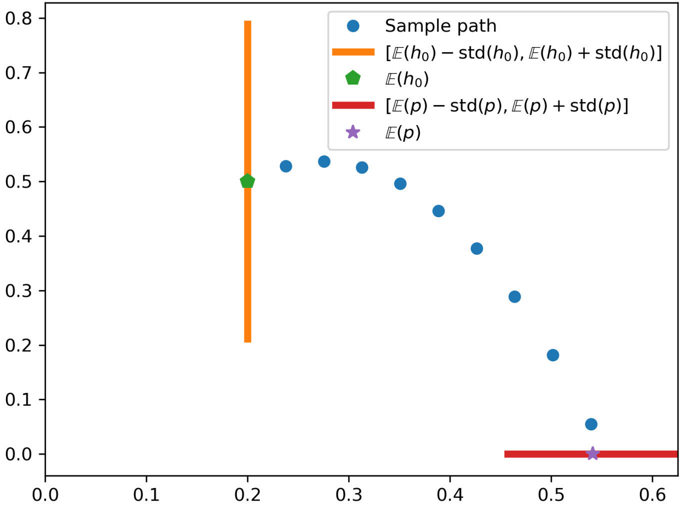
Both quantities involving the integral!

Approximate by Trapezoidal rule

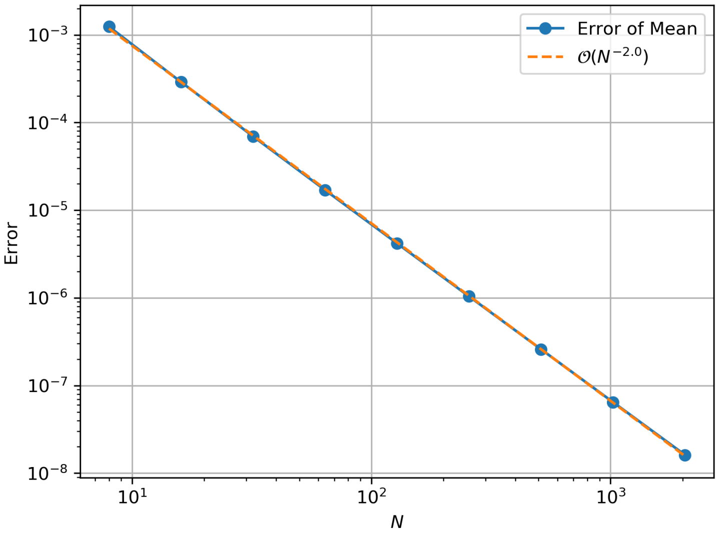
$$\int_0^1 f(y)\,dy \approx \frac{\Delta y}{2}\left(f(y_0) + 2\sum_{i=1}^{N-1} f(y_{i-1}) \cdot + f(y_N)\right)$$



Results varying initial height



Error versus number of evaluations



Making the problem more complicated

- Keep $h_0 \sim \mathcal{U}[0, 1]$

Making the problem more complicated

- Keep $h_0 \sim \mathcal{U}[0, 1]$
- Let the angle $\alpha \sim \mathcal{U}[0, 3\pi/6]$

Making the problem more complicated

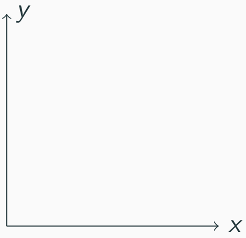
- Keep $h_0 \sim \mathcal{U}[0, 1]$
- Let the angle $\alpha \sim \mathcal{U}[0, 3\pi/6]$
- Now two parameters: h_0 and α

Making the problem more complicated

- Keep $h_0 \sim \mathcal{U}[0, 1]$
- Let the angle $\alpha \sim \mathcal{U}[0, 3\pi/6]$
- Now two parameters: h_0 and α
- Integration in 2D?

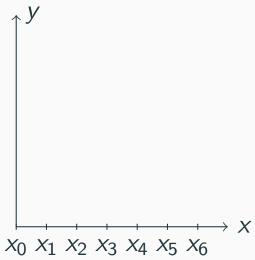
Trapezoidal rule for two dimensions

$$\int_0^1 \int_0^1 f(x, y)\, dx\, dy$$



Trapezoidal rule for two dimensions

$$\int_0^1 \int_0^1 f(x,y)\,dx\,dy \approx \int_0^1 \left(f(x_0,y) + 2\sum_{i=1}^{N-1} f(x_i,y) + f(x_N,y) \right) \Delta x\,dy$$

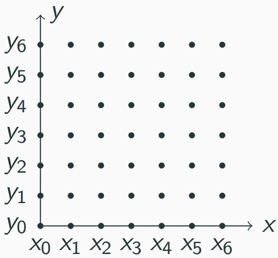


Trapezoidal rule for two dimensions

$$\int_0^1 \int_0^1 f(x,y)\,dx\,dy \approx \int_0^1 \left(f(x_0,y) + 2\sum_{i=1}^{N-1} f(x_i,y) + f(x_N,y) \right) \Delta x\,dy$$

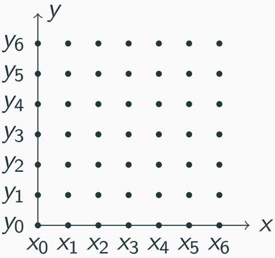
$$\approx \Big(f(0,0) + f(1,0) + f(0,1) + f(1,1) + 4\sum_{i=1}^{N-1}\sum_{i=1}^{N-1} f(x_i,y_i)$$

$$+ 2\sum_{i=1}^{N-1} f(x_i,0) + 2\sum_{i=1}^{N-1} f(x_i,1) + 2\sum_{i=1}^{N-1} f(1,y_i) + 2\sum_{i=1}^{N-1} f(0,y_i) \Big)$$

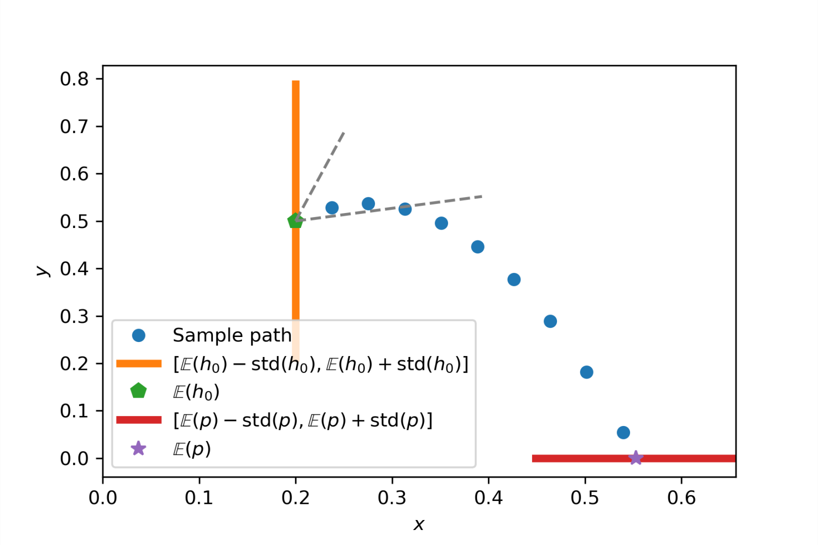


Trapezoidal rule for two dimensions

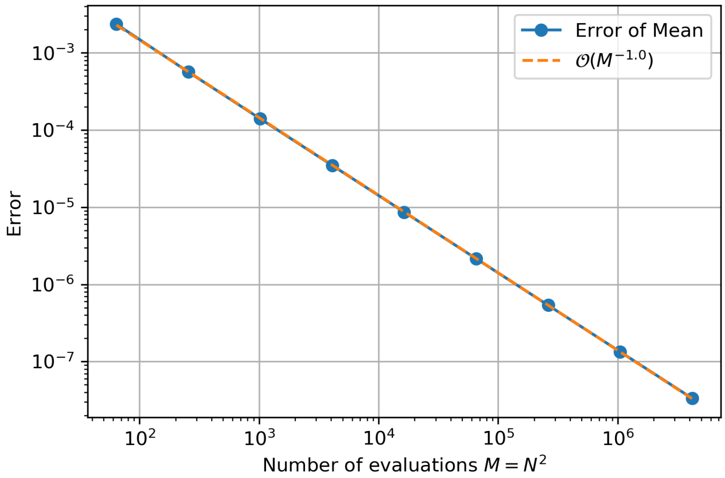
$$\int_0^1 \int_0^1 f(x,y)\,dx\,dy \approx \sum_{i=0}^{N} \sum_{j=0}^{N} w_{i,j} f(x_i, y_j)$$



Results varying initial height



Error versus number of evaluations



Even more complications

- Keep $h_0 \sim \mathcal{U}[0, 1]$, $\alpha \sim \mathcal{U}[0, 5\pi/12]$

Even more complications

- Keep $h_0 \sim \mathcal{U}[0, 1]$, $\alpha \sim \mathcal{U}[0, 5\pi/12]$
- Initial velocity, $v_0 \sim \mathcal{U}[20, 40]$

Even more complications

- Keep $h_0 \sim \mathcal{U}[0, 1]$, $\alpha \sim \mathcal{U}[0, 5\pi/12]$
- Initial velocity, $v_0 \sim \mathcal{U}[20, 40]$
- Air resistance $F_D = \frac{1}{2}\rho v^2 C_D \pi r^2$, where

Even more complications

- Keep $h_0 \sim \mathcal{U}[0, 1]$, $\alpha \sim \mathcal{U}[0, 5\pi/12]$
- Initial velocity, $v_0 \sim \mathcal{U}[20, 40]$
- Air resistance $F_D = \frac{1}{2}\rho v^2 C_D \pi r^2$, where
    - $C_D \sim \mathcal{U}[0.09, 0.11]$ (Drag coefficient)

Even more complications

- Keep $h_0 \sim \mathcal{U}[0, 1]$, $\alpha \sim \mathcal{U}[0, 5\pi/12]$
- Initial velocity, $v_0 \sim \mathcal{U}[20, 40]$
- Air resistance $F_D = \frac{1}{2}\rho v^2 C_D \pi r^2$, where
    - $C_D \sim \mathcal{U}[0.09, 0.11]$ (Drag coefficient)
    - $r \sim \mathcal{U}[0.22, 0.235]$ (Radius of ball)

Even more complications

- Keep $h_0 \sim \mathcal{U}[0, 1]$, $\alpha \sim \mathcal{U}[0, 5\pi/12]$
- Initial velocity, $v_0 \sim \mathcal{U}[20, 40]$
- Air resistance $F_D = \frac{1}{2}\rho v^2 C_D \pi r^2$, where
    - $C_D \sim \mathcal{U}[0.09, 0.11]$ (Drag coefficient)
    - $r \sim \mathcal{U}[0.22, 0.235]$ (Radius of ball)
    - $\rho \sim \mathcal{U}[1.1, 1.4]$ (Air density)

Even more complications

- Keep $h_0 \sim \mathcal{U}[0, 1]$, $\alpha \sim \mathcal{U}[0, 5\pi/12]$
- Initial velocity, $v_0 \sim \mathcal{U}[20, 40]$
- Air resistance $F_D = \frac{1}{2}\rho v^2 C_D \pi r^2$, where
  - $C_D \sim \mathcal{U}[0.09, 0.11]$ (Drag coefficient)
  - $r \sim \mathcal{U}[0.22, 0.235]$ (Radius of ball)
  - $\rho \sim \mathcal{U}[1.1, 1.4]$ (Air density)
- Mass $m \sim \mathcal{U}[0.142, 0.149]$

Even more complications

- Keep $h_0 \sim \mathcal{U}[0, 1]$, $\alpha \sim \mathcal{U}[0, 5\pi/12]$
- Initial velocity, $v_0 \sim \mathcal{U}[20, 40]$
- Air resistance $F_D = \frac{1}{2}\rho v^2 C_D \pi r^2$, where
    - $C_D \sim \mathcal{U}[0.09, 0.11]$ (Drag coefficient)
    - $r \sim \mathcal{U}[0.22, 0.235]$ (Radius of ball)
    - $\rho \sim \mathcal{U}[1.1, 1.4]$ (Air density)
- Mass $m \sim \mathcal{U}[0.142, 0.149]$
- **Seven** parameters

Landing position depends on 7 parameters

$$p(h_0, \alpha, v_0, C_D, r, \rho, m) = \ldots$$

Landing position depends on 7 parameters

$$p(h_0, \alpha, v_0, C_D, r, \rho, m) = \dots$$

$$\mathbb{E}(p) = \int_0^1 \int_0^{5\pi/12} \int_{20}^{40} \int_{0.09}^{0.11} \int_{0.22}^{0.235} \int_{1.1}^{1.4} \int_{0.142}^{0.149} \dots$$

$$p(h_0, \alpha, v_0, C_D, r, \rho, m) \frac{dm\ dr\ dC_D\ dv_0\ d\alpha\ dh_0}{20 \cdot 5\pi/12 \cdot 0.2 \cdot 0.015 \cdot 0.3 \cdot 0.07}$$

How many evaluations of p do we need?

- We have 7 parameters ($d = 7$)
- Trapezoidal rule error

$$\text{Error} \leqslant Ch^2$$

Require Error $< 1/25 \approx 4\%$

- $\Rightarrow h \leqslant \sqrt{1/25} = 1/5$

How many evaluations of p do we need?

- We have 7 parameters ($d = 7$)
- Trapezoidal rule error

$$\text{Error} \leqslant Ch^2$$

Require Error $< 1/25 \approx 4\%$

- $\Rightarrow h \leqslant \sqrt{1/25} = 1/5$
- $\Rightarrow N \geqslant 5$

How many evaluations of p do we need?

- We have 7 parameters ($d = 7$)
- Trapezoidal rule error

$$\text{Error} \leqslant Ch^2$$

Require Error $< 1/25 \approx 4\%$

- $\Rightarrow h \leqslant \sqrt{1/25} = 1/5$
- $\Rightarrow N \geqslant 5$
- $\Rightarrow N^d = N^7 = 5^7 = 78125$ evaluations

Finding p

Governing ODE

$$\begin{cases} \dfrac{d^2}{dt^2}\mathbf{x}(t) = -\mathbf{F}_D(C_D, r, \rho, \dfrac{d}{dt}\mathbf{x}(t))/m - g\mathbf{e}_2 \\ \mathbf{x}(0) = \begin{pmatrix} x_0 \\ y_0 \end{pmatrix} \\ \mathbf{x}'(0) = \begin{pmatrix} v_0\cos(\alpha) \\ v_0\sin(\alpha) \end{pmatrix} \end{cases}$$

Finding p

$\mathbf{x}(t)$ depends on the parameters

$$\begin{cases} \dfrac{d^2}{dt^2}\mathbf{x}(h_0, \alpha, v_0, C_D, r, \rho, m; t) = -\mathbf{F}_D(C_D, r, \rho, \dfrac{d}{dt}\mathbf{x}(\ldots; t))/m - g\mathbf{e}_2 \\ \mathbf{x}(h_0, \alpha, v_0, C_D, r, \rho, m; 0) = \begin{pmatrix} 0 \\ y_0 \end{pmatrix} \\ \dfrac{d}{dt}\mathbf{x}(h_0, \alpha, v_0, C_D, r, \rho, m; 0) = \begin{pmatrix} v_0 \cos(\alpha) \\ v_0 \sin(\alpha) \end{pmatrix} \end{cases}$$

Finding p

Assuming underlying probability space Ω

$$\begin{cases} \dfrac{d^2}{dt^2}\mathbf{x}(\omega; t) = -\mathbf{F}_D(C_D(\omega), r(\omega), \rho(\omega), \dfrac{d}{dt}\mathbf{x}(\omega; t))/m(\omega) - g\mathbf{e}_2 \\ \mathbf{x}(\omega; 0) = \begin{pmatrix} 0 \\ y_0(\omega) \end{pmatrix} \\ \dfrac{d}{dt}\mathbf{x}(\omega; 0) = \begin{pmatrix} v_0(\omega)\cos(\alpha(\omega)) \\ v_0(\omega)\sin(\alpha(\omega)) \end{pmatrix} \end{cases}$$

for $\omega \in \Omega$.

Finding p

Assuming underlying probability space Ω

$$\begin{cases} \dfrac{d^2}{dt^2}\mathbf{x}(\omega; t) = -\mathbf{F}_D(C_D(\omega), r(\omega), \rho(\omega), \dfrac{d}{dt}\mathbf{x}(\omega; t))/m(\omega) - g\mathbf{e}_2 \\ \mathbf{x}(\omega; 0) = \begin{pmatrix} 0 \\ y_0(\omega) \end{pmatrix} \\ \dfrac{d}{dt}\mathbf{x}(\omega; 0) = \begin{pmatrix} v_0(\omega)\cos(\alpha(\omega)) \\ v_0(\omega)\sin(\alpha(\omega)) \end{pmatrix} \end{cases}$$

for $\omega \in \Omega$. Solve using eg. Forward-Euler:

$$\mathbf{x}^{n,\Delta t}(\omega) = \mathbf{x}^{n-1,\Delta t}(\omega) + \Delta t \mathbf{v}^{n-1,\Delta t}(\omega)$$
$$\mathbf{v}^{n,\Delta t}(\omega) = \mathbf{v}^{n-1,\Delta t}(\omega) + \Delta t \left(-\mathbf{F}_D(C_D(\omega), r(\omega), \rho(\omega), \mathbf{v}^{n-1,\Delta t})/m(\omega) - g\mathbf{e}_2\right)$$

Finding p

Assuming underlying probability space Ω

$$\begin{cases} \dfrac{d^2}{dt^2}\mathbf{x}(\omega; t) = -\mathbf{F}_D(C_D(\omega), r(\omega), \rho(\omega), \dfrac{d}{dt}\mathbf{x}(\omega; t))/m(\omega) - g\mathbf{e}_2 \\ \mathbf{x}(\omega; 0) = \begin{pmatrix} 0 \\ y_0(\omega) \end{pmatrix} \\ \dfrac{d}{dt}\mathbf{x}(\omega; 0) = \begin{pmatrix} v_0(\omega)\cos(\alpha(\omega)) \\ v_0(\omega)\sin(\alpha(\omega)) \end{pmatrix} \end{cases}$$

for $\omega \in \Omega$. Solve using eg. Forward-Euler:
Until $y^{n,\Delta t}(\omega) < 0$

$$\mathbf{x}^{n,\Delta t}(\omega) = \mathbf{x}^{n-1,\Delta t}(\omega) + \Delta t \mathbf{v}^{n-1,\Delta t}(\omega)$$

$$\mathbf{v}^{n,\Delta t}(\omega) = \mathbf{v}^{n-1,\Delta t}(\omega) + \Delta t \left(-\mathbf{F}_D(C_D(\omega), r(\omega), \rho(\omega), \mathbf{v}^{n-1,\Delta t})/m(\omega) - g\mathbf{e}_2\right)$$

Finding p

Assuming underlying probability space Ω

$$\begin{cases} \frac{d^2}{dt^2}\mathbf{x}(\omega; t) = -\mathbf{F}_D(C_D(\omega), r(\omega), \rho(\omega), \frac{d}{dt}\mathbf{x}(\omega; t))/m(\omega) - g\mathbf{e}_2 \\ \mathbf{x}(\omega; 0) = \begin{pmatrix} 0 \\ y_0(\omega) \end{pmatrix} \\ \frac{d}{dt}\mathbf{x}(\omega; 0) = \begin{pmatrix} v_0(\omega)\cos(\alpha(\omega)) \\ v_0(\omega)\sin(\alpha(\omega)) \end{pmatrix} \end{cases}$$

for $\omega \in \Omega$. Solve using eg. Forward-Euler:

Until $y^{n,\Delta t}(\omega) < 0$

$$\mathbf{x}^{n,\Delta t}(\omega) = \mathbf{x}^{n-1,\Delta t}(\omega) + \Delta t \mathbf{v}^{n-1,\Delta t}(\omega)$$

$$\mathbf{v}^{n,\Delta t}(\omega) = \mathbf{v}^{n-1,\Delta t}(\omega) + \Delta t \left(-\mathbf{F}_D(C_D(\omega), r(\omega), \rho(\omega), \mathbf{v}^{n-1,\Delta t})/m(\omega) - g\mathbf{e}_2\right)$$

Set $p^{\Delta t}(\omega) = y^{N,\Delta t}(\omega)$.

Algorithm: Compute expected position using Trapezoidal rule

Input: $\Delta t, \Delta h_0, \Delta\alpha, \Delta v_0 \ldots$

1. Initialize mean $E = 0$
2. For each integration point $h_0^i,\ \alpha^j,\ v_0^k, \ldots$:

Algorithm: Compute expected position using Trapezoidal rule

Input: $\Delta t, \Delta h_0, \Delta\alpha, \Delta v_0 \ldots$

1. Initialize mean $E = 0$
2. For each integration point $h_0^i,\ \alpha^j,\ v_0^k, \ldots$:
    2.1 Use forward Euler to compute

$$p^{\Delta t}(h_0^i, \alpha^j, v_0^k, \ldots)$$

Algorithm: Compute expected position using Trapezoidal rule

Input: $\Delta t, \Delta h_0, \Delta\alpha, \Delta v_0 \ldots$

1. Initialize mean $E = 0$
2. For each integration point $h_0^i,\ \alpha^j,\ v_0^k, \ldots$:
   2.1 Use forward Euler to compute

   $$p^{\Delta t}(h_0^i, \alpha^j, v_0^k, \ldots)$$

   2.2 Update mean

   $$E = E + w_{i,j,k,\ldots} \cdot p^{\Delta t}(h_0^i, \alpha^j, v_0^k, \ldots)$$

Algorithm: Compute expected position using Trapezoidal rule

Input: $\Delta t, \Delta h_0, \Delta\alpha, \Delta v_0 \ldots$

1. Initialize mean $E = 0$
2. For each integration point $h_0^i,\ \alpha^j,\ v_0^k, \ldots$:
   2.1 Use forward Euler to compute

   $$p^{\Delta t}(h_0^i, \alpha^j, v_0^k, \ldots)$$

   2.2 Update mean

   $$E = E + w_{i,j,k,\ldots} \cdot p^{\Delta t}(h_0^i, \alpha^j, v_0^k, \ldots)$$

Then

$$E \approx \mathbb{E}(p)$$

Each evaluation is expensive when done 100 000 times!

Runtime to obtain an error ϵ:

Each evaluation is expensive when done 100 000 times!

Runtime to obtain an error ϵ:

- Forward Euler: $\mathcal{O}(\epsilon^{-1})$

Each evaluation is expensive when done 100 000 times!

Runtime to obtain an error ϵ:

- Forward Euler: $\mathcal{O}(\epsilon^{-1})$
- Trapezoidal in 7D: $\mathcal{O}(\epsilon^{-7/2})$

Each evaluation is expensive when done 100 000 times!

Runtime to obtain an error ϵ:

- Forward Euler: $\mathcal{O}(\epsilon^{-1})$
- Trapezoidal in 7D: $\mathcal{O}(\epsilon^{-7/2})$
- Total cost:

$$\overbrace{\mathcal{O}(\epsilon^{-1})}^{\text{Forward Euler}} \cdot \underbrace{\mathcal{O}(\epsilon^{-7/2})}_{\text{Trapezoidal}} = \mathcal{O}(\epsilon^{-4.5}).$$

Higher order quadratures?

- For every dimension k, we could find integration rules such that

$$\text{Error} \leqslant Ch^k$$

Higher order quadratures?

- For every dimension k, we could find integration rules such that

$$\text{Error} \leqslant Ch^k$$

- Then to obtain an error $\mathcal{O}(\epsilon)$, we choose $h = \mathcal{O}(\sqrt[k]{\epsilon})$

Higher order quadratures?

- For every dimension k, we could find integration rules such that

$$\text{Error} \leqslant Ch^k$$

- Then to obtain an error $\mathcal{O}(\epsilon)$, we choose $h = \mathcal{O}(\sqrt[k]{\epsilon})$
- $N = \mathcal{O}(\epsilon^{-1/k}) \Rightarrow M = \mathcal{O}(\epsilon^{-1})$

Higher order quadratures?

- For every dimension k, we could find integration rules such that

$$\text{Error} \leqslant Ch^k$$

- Then to obtain an error $\mathcal{O}(\epsilon)$, we choose $h = \mathcal{O}(\sqrt[k]{\epsilon})$
- $N = \mathcal{O}(\epsilon^{-1/k}) \Rightarrow M = \mathcal{O}(\epsilon^{-1})$
- Good idea?

Higher order quadratures?

- For every dimension k, we could find integration rules such that

$$\text{Error} \leqslant Ch^k$$

- Then to obtain an error $\mathcal{O}(\epsilon)$, we choose $h = \mathcal{O}(\sqrt[k]{\epsilon})$
- $N = \mathcal{O}(\epsilon^{-1/k}) \Rightarrow M = \mathcal{O}(\epsilon^{-1})$
- Good idea?
    - Hard to implement

Higher order quadratures?

- For every dimension k, we could find integration rules such that

$$\text{Error} \leqslant Ch^k$$

- Then to obtain an error $\mathcal{O}(\epsilon)$, we choose $h = \mathcal{O}(\sqrt[k]{\epsilon})$
- $N = \mathcal{O}(\epsilon^{-1/k}) \Rightarrow M = \mathcal{O}(\epsilon^{-1})$
- Good idea?
  - Hard to implement
  - Large constant

Higher order quadratures?

- For every dimension k, we could find integration rules such that

$$\text{Error} \leqslant Ch^k$$

- Then to obtain an error $\mathcal{O}(\epsilon)$, we choose $h = \mathcal{O}(\sqrt[k]{\epsilon})$
- $N = \mathcal{O}(\epsilon^{-1/k}) \Rightarrow M = \mathcal{O}(\epsilon^{-1})$
- Good idea?
    - Hard to implement
    - Large constant
    - Need a stencil of size at least k

Higher order quadratures?

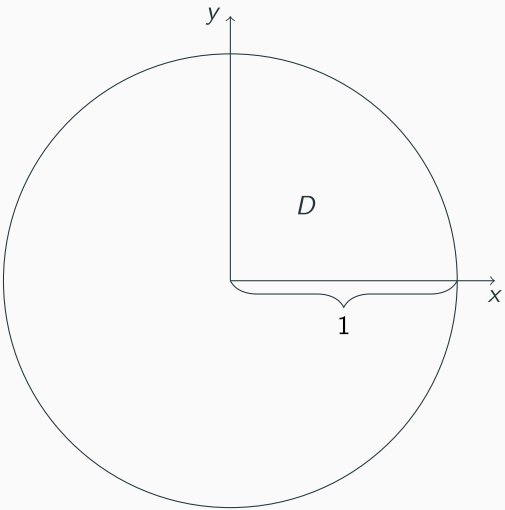
- For every dimension k, we could find integration rules such that

$$\text{Error} \leqslant Ch^k$$

- Then to obtain an error $\mathcal{O}(\epsilon)$, we choose $h = \mathcal{O}(\sqrt[k]{\epsilon})$
- $N = \mathcal{O}(\epsilon^{-1/k}) \Rightarrow M = \mathcal{O}(\epsilon^{-1})$
- Good idea?
  - Hard to implement
  - Large constant
  - Need a stencil of size at least k
  - Oscillation issuess

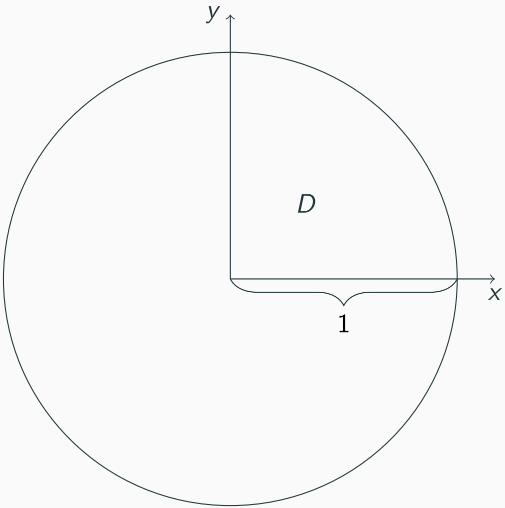
Monte Carlo methods

Obligatory example: Computing π



$$|D| = \frac{\pi}{4}$$

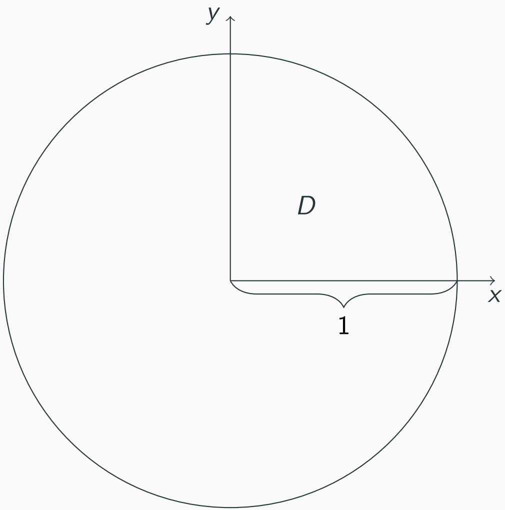
Obligatory example: Computing π



$$|D| = \frac{\pi}{4}$$

$$|D| = \int_0^1 \int_0^1 \mathbf{1}_{B(0,1)}(x, y)\, dx\, dy$$

Obligatory example: Computing π

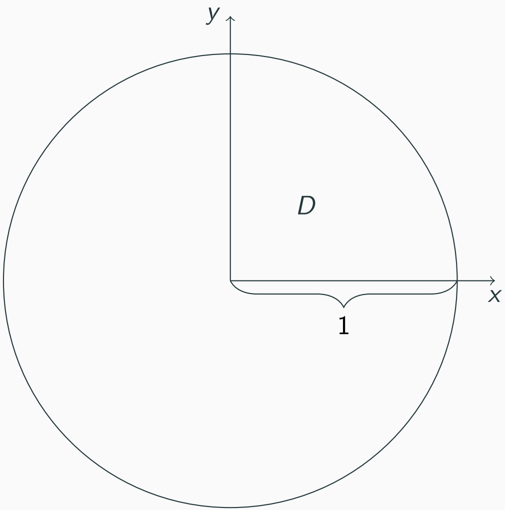


$$|D| = \frac{\pi}{4}$$

$$|D| = \int_0^1 \int_0^1 \mathbf{1}_{B(0,1)}(x, y)\ dx\ dy$$

$$\Rightarrow \pi = 4 \int_0^1 \int_0^1 \mathbf{1}_{B(0,1)}(x, y)\ dx\ dy$$

Obligatory example: Computing π



$$|D| = \frac{\pi}{4}$$

$$|D| = \int_0^1 \int_0^1 \mathbf{1}_{B(0,1)}(x, y)\, dx\, dy$$

$$\Rightarrow \pi = 4 \int_0^1 \int_0^1 \mathbf{1}_{B(0,1)}(x, y)\, dx\, dy$$

$$\approx 4 \frac{1}{M} \sum_{k=1}^{M} \mathbf{1}_{B(0,1)}(X_k, Y_k)$$

$X_k \sim \mathcal{U}[0, 1]$, $Y_k \sim \mathcal{U}[0, 1]$ independent random variables

Convergence of Monte Carlo

$$\left(\frac{1}{M}\sum_{k=1}^{M} X_k - \mathbb{E}(X)\right)^2$$

Convergence of Monte Carlo

$$\left(\frac{1}{M}\sum_{k=1}^{M} X_k - \mathbb{E}(X)\right)^2 = \left(\frac{1}{M}\sum_{k=1}^{M} X_k - \frac{1}{M}\sum_{k=1}^{M}\mathbb{E}(X)\right)^2$$

Convergence of Monte Carlo

$$\left(\frac{1}{M}\sum_{k=1}^{M} X_k - \mathbb{E}(X)\right)^2 = \left(\frac{1}{M}\sum_{k=1}^{M} X_k - \frac{1}{M}\sum_{k=1}^{M}\mathbb{E}(X)\right)^2$$
$$= \frac{1}{M^2}\left(\sum_{k=1}^{M}\left(X_k - \mathbb{E}(X)\right)\right)^2$$

Convergence of Monte Carlo

$$\begin{aligned}
\left(\frac{1}{M}\sum_{k=1}^{M} X_k - \mathbb{E}(X)\right)^2 &= \left(\frac{1}{M}\sum_{k=1}^{M} X_k - \frac{1}{M}\sum_{k=1}^{M}\mathbb{E}(X)\right)^2 \\
&= \frac{1}{M^2}\left(\sum_{k=1}^{M}\left(X_k - \mathbb{E}(X)\right)\right)^2 \\
&= \frac{1}{M^2}\sum_{k=1}^{M}\left(X_k - \mathbb{E}(X)\right)^2 \\
&\quad + \frac{1}{M^2}\sum_{k=1}^{M}\sum_{j\neq k}\left(X_k - \mathbb{E}(X)\right)\left(X_j - \mathbb{E}(X)\right)
\end{aligned}$$

$$\left(\frac{1}{M}\sum_{k=1}^{M} X_k - \mathbb{E}(X)\right)^2 = \frac{1}{M^2}\sum_{k=1}^{M}(X_k - \mathbb{E}(X))^2 + \frac{1}{M^2}\sum_{k=1}^{M}\sum_{j\neq k}(X_k - \mathbb{E}(X))(X_j - \mathbb{E}(X))$$

Now take the expectation

$$\mathbb{E}\left(\left(\frac{1}{M}\sum_{k=1}^{M} X_k - \mathbb{E}(X)\right)^2\right) = \mathbb{E}\left(\frac{1}{M^2}\sum_{k=1}^{M}(X_k - \mathbb{E}(X))^2\right) + \frac{1}{M^2}\sum_{k=1}^{M}\sum_{j\neq k}\underbrace{\mathbb{E}\left((X_k - \mathbb{E}(X))(X_j - \mathbb{E}(X))\right)}_{=0}$$

$$\left(\frac{1}{M}\sum_{k=1}^{M} X_k - \mathbb{E}(X)\right)^2 = \frac{1}{M^2}\sum_{k=1}^{M} (X_k - \mathbb{E}(X))^2 + \frac{1}{M^2}\sum_{k=1}^{M}\sum_{j\neq k} (X_k - \mathbb{E}(X))(X_j - \mathbb{E}(X))$$

Now take the expectation

$$\mathbb{E}\left(\left(\frac{1}{M}\sum_{k=1}^{M} X_k - \mathbb{E}(X)\right)^2\right) = \mathbb{E}\left(\frac{1}{M^2}\sum_{k=1}^{M} (X_k - \mathbb{E}(X))^2\right) + \frac{1}{M^2}\sum_{k=1}^{M}\sum_{j\neq k} \underbrace{\mathbb{E}\left((X_k - \mathbb{E}(X))(X_j - \mathbb{E}(X))\right)}_{=0}$$

$$= \mathbb{E}\left(\frac{1}{M^2}\sum_{k=1}^{M} (X_k - \mathbb{E}(X))^2\right)$$

$$\left(\frac{1}{M}\sum_{k=1}^{M} X_k - \mathbb{E}(X)\right)^2 = \frac{1}{M^2}\sum_{k=1}^{M}\left(X_k - \mathbb{E}(X)\right)^2 + \frac{1}{M^2}\sum_{k=1}^{M}\sum_{j\neq k}\left(X_k - \mathbb{E}(X)\right)\left(X_j - \mathbb{E}(X)\right)$$

Now take the expectation

$$\begin{aligned}
\mathbb{E}\left(\left(\frac{1}{M}\sum_{k=1}^{M} X_k - \mathbb{E}(X)\right)^2\right) &= \mathbb{E}\left(\frac{1}{M^2}\sum_{k=1}^{M}\left(X_k - \mathbb{E}(X)\right)^2\right) \\
&\quad + \frac{1}{M^2}\sum_{k=1}^{M}\sum_{j\neq k}\underbrace{\mathbb{E}\left(\left(X_k - \mathbb{E}(X)\right)\left(X_j - \mathbb{E}(X)\right)\right)}_{=0} \\
&= \mathbb{E}\left(\frac{1}{M^2}\sum_{k=1}^{M}\left(X_k - \mathbb{E}(X)\right)^2\right) \\
&= \frac{1}{M}\mathrm{Var}(X)
\end{aligned}$$

Monte Carlo error

In *expectation* we converge

$$\mathbb{E}\left(\left(\frac{1}{M}\sum_{k=1}^{M} X_k - \mathbb{E}(X)\right)^2\right)^{1/2} = \frac{\mathrm{Var}(X)^{1/2}}{M^{1/2}}$$

Algorithm: Using Monte Carlo for ODE

$$\begin{cases} u'(\omega; t) = F(u(\omega, t)) \\ u(\omega, 0) = u_0(\omega) \end{cases}$$

Approximate: $\mathbb{E}(u(\cdot; T))$

Algorithm: Using Monte Carlo for ODE

$$\begin{cases} u'(\omega; t) = F(u(\omega, t)) \\ u(\omega, 0) = u_0(\omega) \end{cases}$$

Approximate: $\mathbb{E}(u(\cdot; T))$ Input: Δt, M

1. Initialize mean $E = 0$

Algorithm: Using Monte Carlo for ODE

$$\begin{cases} u'(\omega; t) = F(u(\omega, t)) \\ u(\omega, 0) = u_0(\omega) \end{cases}$$

Approximate: $\mathbb{E}(u(\cdot; T))$ Input: Δt, M

1. Initialize mean $E = 0$
2. Draw M i.i.d samples $u_0^1, \ldots, u_0^k, \ldots, u_0^M$.

Algorithm: Using Monte Carlo for ODE

$$\begin{cases} u'(\omega; t) = F(u(\omega, t)) \\ u(\omega, 0) = u_0(\omega) \end{cases}$$

Approximate: $\mathbb{E}(u(\cdot; T))$ Input: Δt, M

1. Initialize mean $E = 0$
2. Draw M i.i.d samples $u_0^1, \ldots, u_0^k, \ldots, u_0^M$.
3. For each $k = 1, \ldots, M$:
   3.1 Compute (with eg Forward Euler):

$$u^k_{\Delta T, N} = \mathcal{F}(u_0^k)$$

Algorithm: Using Monte Carlo for ODE

$$\begin{cases} u'(\omega;t) = F(u(\omega,t)) \\ u(\omega,0) = u_0(\omega) \end{cases}$$

Approximate: $\mathbb{E}(u(\cdot;T))$ Input: Δt, M

1. Initialize mean $E = 0$
2. Draw M i.i.d samples $u_0^1, \ldots, u_0^k, \ldots, u_0^M$.
3. For each $k = 1, \ldots, M$:
   3.1 Compute (with eg Forward Euler):
   $$u_{\Delta T,N}^k = \mathcal{F}(u_0^k)$$
   3.2 Update mean
   $$E = E + \frac{1}{M} u_{\Delta T,N}^k$$

Algorithm: Using Monte Carlo for ODE

$$\begin{cases} u'(\omega; t) = F(u(\omega, t)) \\ u(\omega, 0) = u_0(\omega) \end{cases}$$

Approximate: $\mathbb{E}(u(\cdot; T))$ Input: Δt, M

1. Initialize mean $E = 0$
2. Draw M i.i.d samples $u_0^1, \dots, u_0^k, \dots, u_0^M$.
3. For each $k = 1, \dots, M$:
    3.1 Compute (with eg Forward Euler):

    $$u^k_{\Delta T, N} = \mathcal{F}(u^k_0)$$

    3.2 Update mean

    $$E = E + \frac{1}{M} u^k_{\Delta T, N}$$

Then

$$E \approx \mathbb{E}(u(\cdot, T))$$

Example

See notebook

Conclusion

- Monte Carlo

$$\int_{\Omega} f(X(\omega))\, d\mathbb{P}(\omega) \approx \frac{1}{M} \sum_{k=1}^{M} f(X_k(\omega))$$

Conclusion

- Monte Carlo

$$\int_{\Omega} f(X(\omega))\, d\mathbb{P}(\omega) \approx \frac{1}{M} \sum_{k=1}^{M} f(X_k(\omega))$$

- Error of Monte Carlo

$$\mathbb{E}\left(\left(\frac{1}{M}\sum_{k=1}^{M} X_k - \mathbb{E}(X)\right)^2\right)^{1/2} = \frac{\mathrm{Var}(X)^{1/2}}{M^{1/2}}$$

Conclusion

- Monte Carlo

$$\int_\Omega f(X(\omega))\, d\mathbb{P}(\omega) \approx \frac{1}{M} \sum_{k=1}^{M} f(X_k(\omega))$$

- Error of Monte Carlo

$$\mathbb{E}\left(\left(\frac{1}{M}\sum_{k=1}^{M} X_k - \mathbb{E}(X)\right)^2\right)^{1/2} = \frac{\mathrm{Var}(X)^{1/2}}{M^{1/2}}$$

- Motivation for tomorrow: Monte Carlo is *slow*

Conclusion

- Monte Carlo

$$\int_{\Omega} f(X(\omega))\, d\mathbb{P}(\omega) \approx \frac{1}{M} \sum_{k=1}^{M} f(X_k(\omega))$$

- Error of Monte Carlo

$$\mathbb{E}\left(\left(\frac{1}{M}\sum_{k=1}^{M} X_k - \mathbb{E}(X)\right)^2\right)^{1/2} = \frac{\mathrm{Var}(X)^{1/2}}{M^{1/2}}$$

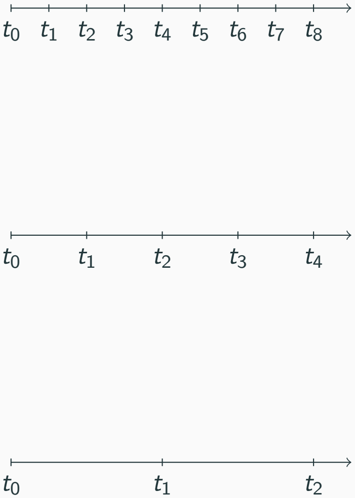
- Motivation for tomorrow: Monte Carlo is *slow*
- ... but sometimes the best we have

Multilevel Monte Carlo

Multilevel telescoping sum

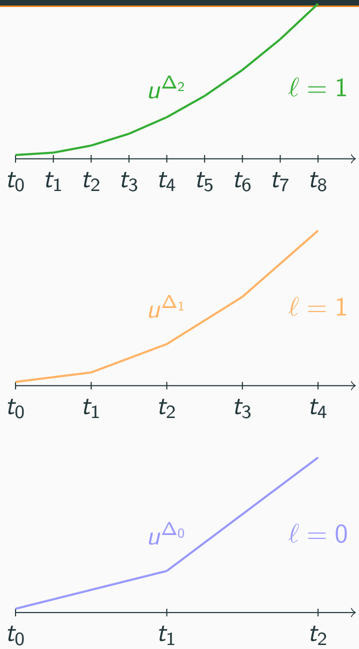
Numerical approximation u^{Δ}

Multilevel telescoping sum



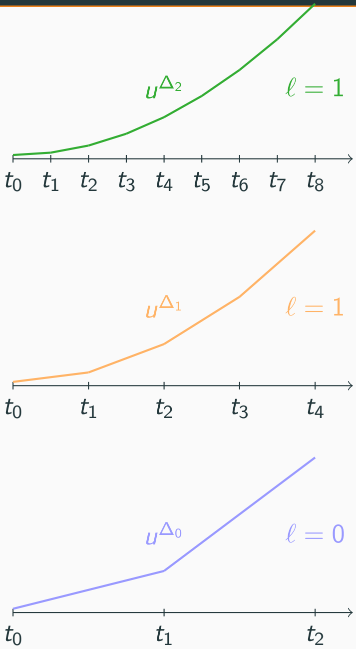
Numerical approximation u^{Δ}

Multilevel telescoping sum



Numerical approximation u^{Δ}

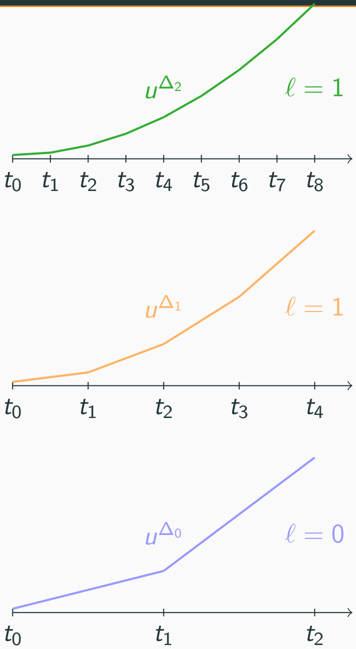
Multilevel telescoping sum



Numerical approximation u^{Δ}

$$u^{\Delta_1} = (u^{\Delta_1} - u^{\Delta_0}) + u^{\Delta_0}$$

Multilevel telescoping sum



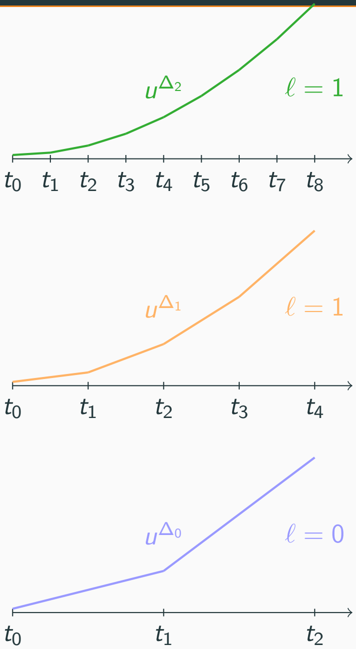
Numerical approximation u^{Δ}

$$u^{\Delta_1} = (u^{\Delta_1} - u^{\Delta_0}) + u^{\Delta_0}$$

Similarly,

$$u^{\Delta_2} = (u^{\Delta_2} - u^{\Delta_1}) + (u^{\Delta_1} - u^{\Delta_0}) + u^{\Delta_0}$$

Multilevel telescoping sum



Numerical approximation u^{Δ}

$$u^{\Delta_1} = (u^{\Delta_1} - u^{\Delta_0}) + u^{\Delta_0}$$

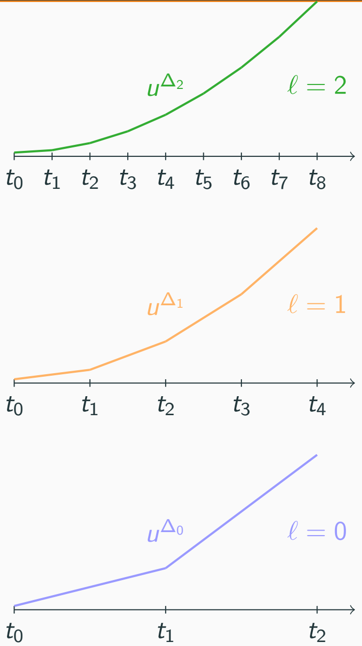
Similarly,

$$u^{\Delta_2} = (u^{\Delta_2} - u^{\Delta_1}) + (u^{\Delta_1} - u^{\Delta_0}) + u^{\Delta_0}$$

In general,

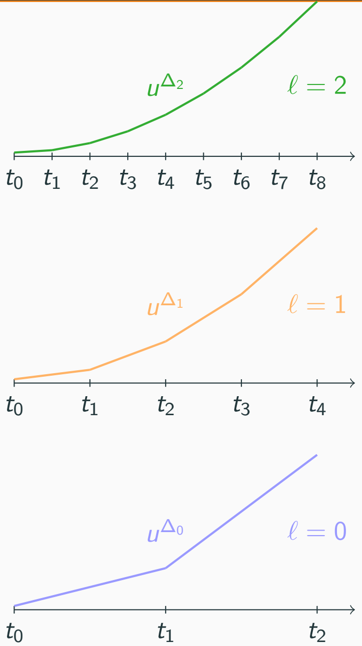
$$u^{\Delta_L} = \sum_{l=1}^{L} (u^{\Delta_\ell} - u^{\Delta_{\ell-1}}) + u^{\Delta_0}$$

Multilevel Monte Carlo [Giles, 2008; Heinrich 2001]



Random variable $u^{\Delta}(\omega)$

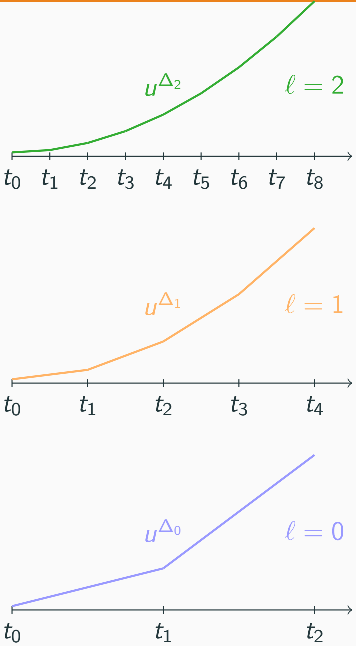
Multilevel Monte Carlo [Giles, 2008; Heinrich 2001]



Random variable $u^{\Delta}(\omega)$

$$\mathbb{E}(u^{\Delta_1}) = \mathbb{E}(u^{\Delta_1} - u^{\Delta_0}) + \mathbb{E}(u^{\Delta_0})$$

Multilevel Monte Carlo [Giles, 2008; Heinrich 2001]



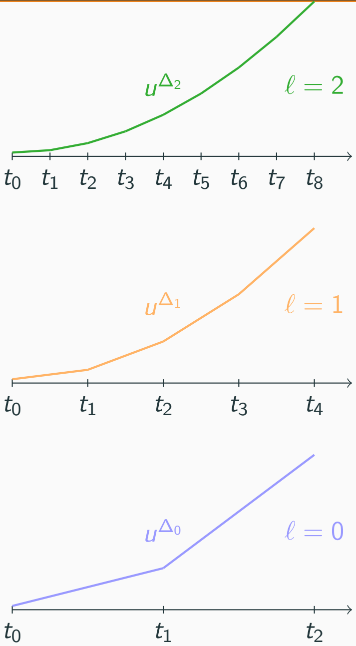
Random variable $u^{\Delta}(\omega)$

$$\mathbb{E}(u^{\Delta_1}) = \mathbb{E}(u^{\Delta_1} - u^{\Delta_0}) + \mathbb{E}(u^{\Delta_0})$$

Similarly,

$$\mathbb{E}(u^{\Delta_2}) = \mathbb{E}(u^{\Delta_2} - u^{\Delta_1}) + \mathbb{E}(u^{\Delta_1} - u^{\Delta_0}) + \mathbb{E}(u^{\Delta_0})$$

Multilevel Monte Carlo [Giles, 2008; Heinrich 2001]



Random variable $u^{\Delta}(\omega)$

$$\mathbb{E}(u^{\Delta_1}) = \mathbb{E}(u^{\Delta_1} - u^{\Delta_0}) + \mathbb{E}(u^{\Delta_0})$$

Similarly,

$$\mathbb{E}(u^{\Delta_2}) = \mathbb{E}(u^{\Delta_2} - u^{\Delta_1}) + \mathbb{E}(u^{\Delta_1} - u^{\Delta_0}) + \mathbb{E}(u^{\Delta_0})$$

In general,

$$\mathbb{E}(u^{\Delta_L}) = \sum_{l=1}^{L} \mathbb{E}(u^{\Delta_\ell} - u^{\Delta_{\ell-1}}) + \mathbb{E}(u^{\Delta_0})$$

Variance reduction

$$\mathbb{E}(u^{\Delta_1}) = \mathbb{E}(u^{\Delta_1} - u^{\Delta_0}) + \mathbb{E}(u^{\Delta_0})$$

Variance reduction

$$\mathbb{E}(u^{\Delta_1}) = \mathbb{E}(u^{\Delta_1} - u^{\Delta_0}) + \mathbb{E}(u^{\Delta_0})$$

$$\approx \underbrace{\frac{1}{M_1} \sum_{k=1}^{M_1} \left(u_k^{\Delta_1} - u_k^{\Delta_0} \right)}_{\approx \mathbb{E}(u^{\Delta_1} - u^{\Delta_0})} + \underbrace{\frac{1}{M_0} \sum_{k=1}^{M_0} u_k^{\Delta_0}}_{\approx \mathbb{E}(u^{\Delta_0})}$$

Variance reduction

$$\begin{aligned}
\mathbb{E}(u^{\Delta_1}) &= \mathbb{E}(u^{\Delta_1} - u^{\Delta_0}) + \mathbb{E}(u^{\Delta_0}) \\
&\approx \underbrace{\frac{1}{M_1}\sum_{k=1}^{M_1}\left(u_k^{\Delta_1} - u_k^{\Delta_0}\right)}_{\approx \mathbb{E}(u^{\Delta_1} - u^{\Delta_0})} + \underbrace{\frac{1}{M_0}\sum_{k=1}^{M_0} u_k^{\Delta_0}}_{\approx \mathbb{E}(u^{\Delta_0})} \\
&=: E_{M_0,M_1}(u)
\end{aligned}$$

Error

$$\begin{aligned}
\mathbb{E}\left(\left(E_{M_0,M_1}(u) - \mathbb{E}(u^{\Delta_1})\right)^2\right)^{1/2} &= \mathbb{E}\left(\left(E_{M_0,M_1}(u) - \left(\mathbb{E}(u^{\Delta_1} - u^{\Delta_0}) + \mathbb{E}(u^{\Delta_0})\right)\right)^2\right) \\
&\quad + \mathbb{E}\left(\left(E_{M_0,M_1}(u^{\Delta_0}, u^{\Delta_1}) - \mathbb{E}(u^{\Delta_1})\right)^2\right)^{1/2}
\end{aligned}$$

$$\frac{\mathrm{Var}(X)^{1/2}}{M^{1/2}}$$