

Numerical Methods for Astrophysics:

NUMERICAL INTEGRATION

Michela Mapelli

Integrals. Concept

NUMERICALLY integrate a function over a range

Useful

- if the function has no exact analytic solution**
- or if the analytic solution is too complex**
- or whenever we do not want to calculate analytic solution**

Many possible algorithms to integrate numerically, e.g.:

- trapezoidal rule**
- Simpson's rule**
- Romberg integration**
- Gauss quadrature**
- Monte Carlo methods**

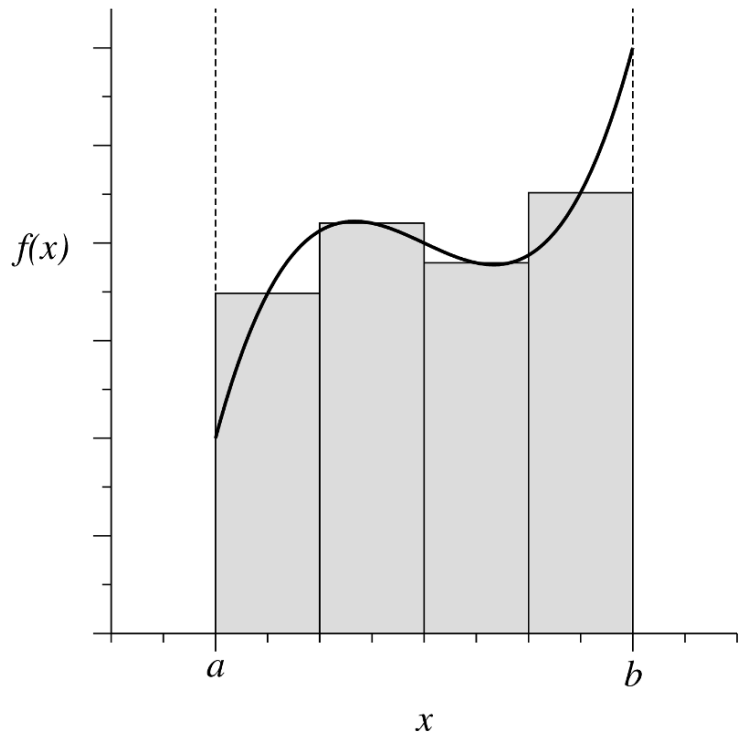
**Here just two examples: trapezoidal rule and Monte Carlo
but Mark Newman's book contains many more**

+ at the end of the lecture, we will see the built-in functions of scipy

Integrals. Trapezoidal rule

$$I = \int_a^b f(x) dx$$

is equivalent to calculate the area under the curve $f(x)$ from a to b .

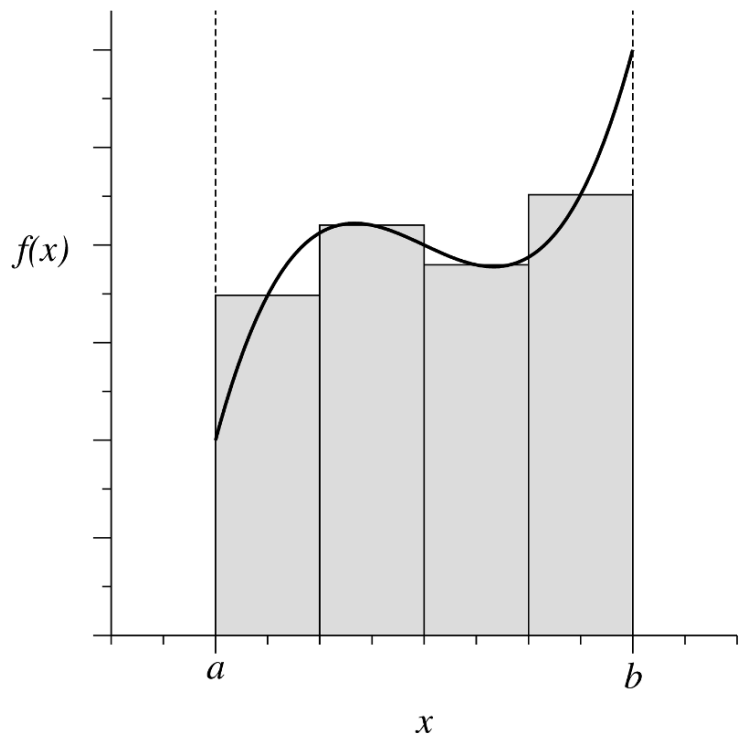


Integrals. Trapezoidal rule

$$I = \int_a^b f(x) dx$$

is equivalent to calculate the area under the curve $f(x)$ from a to b .

Simplest way to proceed: divide this area into rectangular slices, calculate the area of each rectangle and then sum them up.



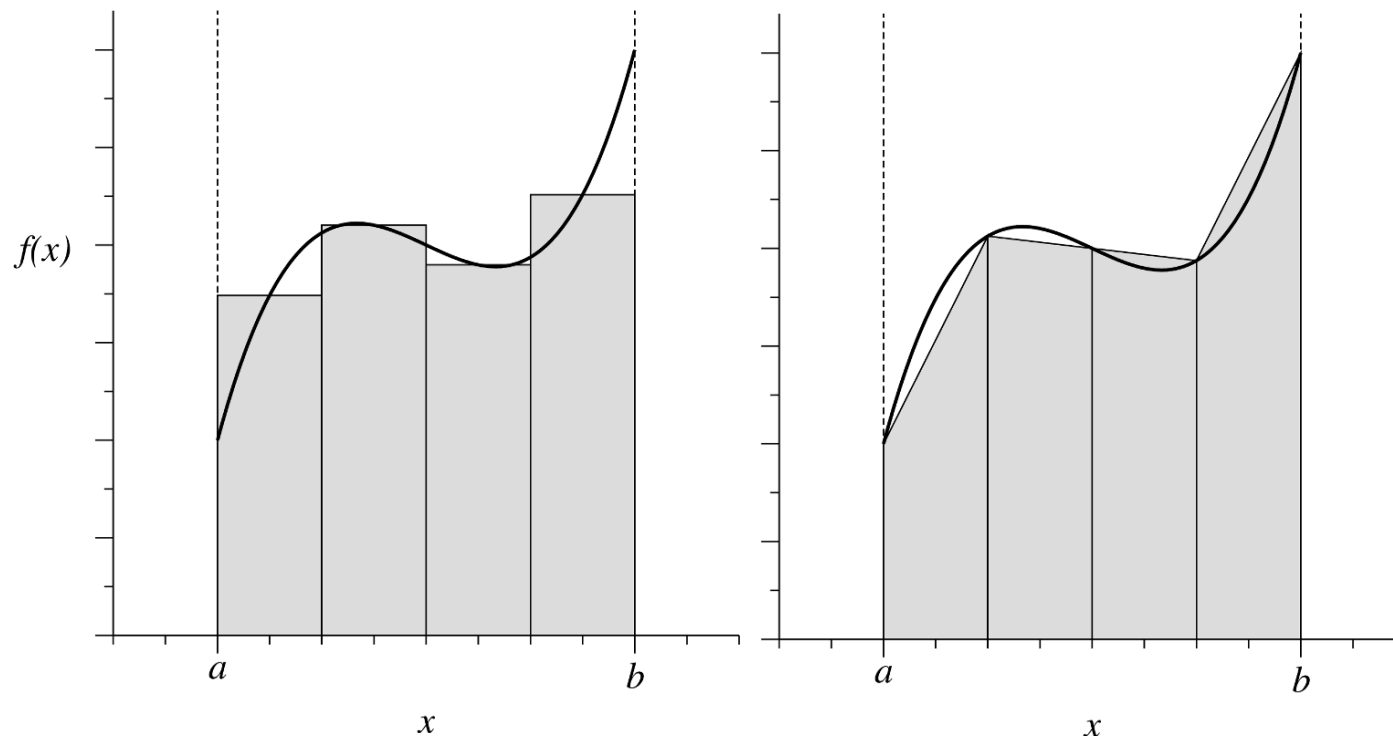
Integrals. Trapezoidal rule

$$I = \int_a^b f(x) dx$$

is equivalent to calculate the area under the curve $f(x)$ from a to b .

Simplest way to proceed: divide this area into rectangular slices, calculate the area of each rectangle and then sum them up.

Next simplest way to proceed: use TRAPEZOIDS instead of rectangles



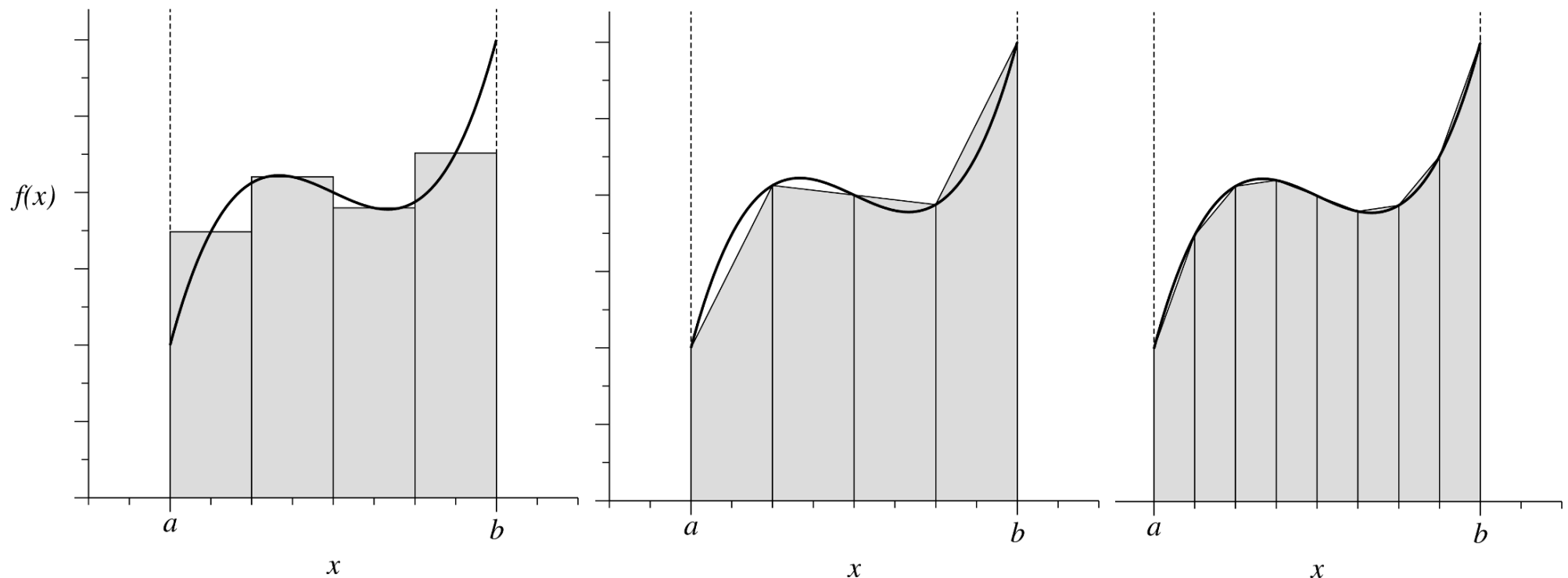
Integrals. Trapezoidal rule

$$I = \int_a^b f(x) dx$$

is equivalent to calculate the area under the curve $f(x)$ from a to b .

Simplest way to proceed: divide this area into rectangular slices, calculate the area of each rectangle and then sum them up.

Next simplest way to proceed: use TRAPEZOIDS instead of rectangles
Things improve if we make the trapezoids' height smaller



Integrals. Trapezoidal rule

In practice,

1. divide the interval from a to b into N slices (or steps)
so that each slice has a height $h = (b - a) / N$

2. if k is an integer number $k = 1, 2, \dots, N$

the k -th trapezoid lies between $a + (k - 1) * h$ and $a + k * h$

It has height = h and its two bases measure $f(a + (k-1)h)$ and $f(a + kh)$

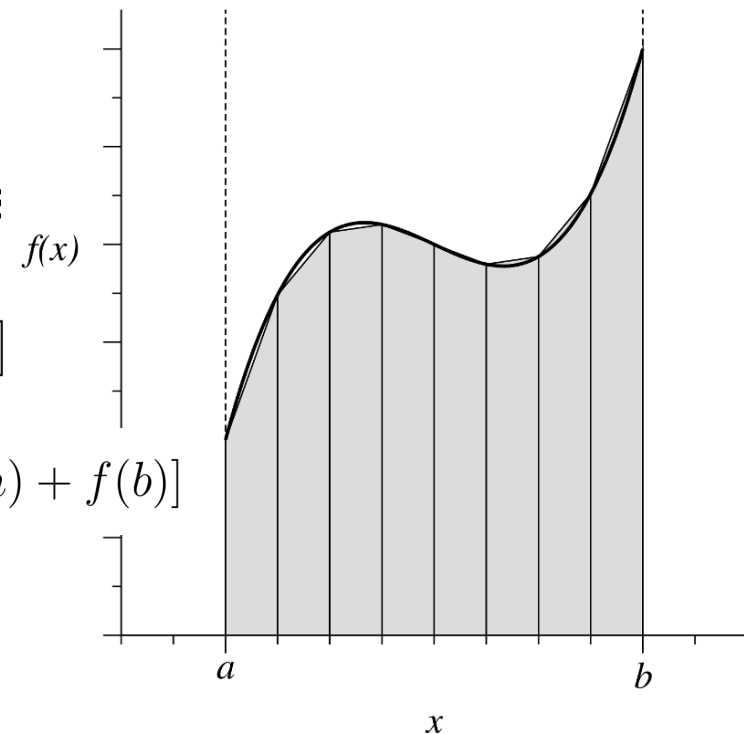
Hence, the area of the k -th trapezoid is

$$I_k = \frac{1}{2} [f(a + kh) + f(a + (k - 1)h)] h$$

3. The total area summing over the trapezoids is

$$\begin{aligned} I(a, b) &\simeq \sum_{k=1}^N I_k = \frac{1}{2} h \sum_{k=1}^N [f(a + (k - 1)h) + f(a + kh)] \\ &= \frac{1}{2} h [f(a) + 2f(a + h) + 2f(a + 2h) + \dots + 2f(a + (N - 1)h) + f(b)] \end{aligned}$$

$$= h \left[\frac{1}{2} f(a) + \frac{1}{2} f(b) + \sum_{k=1}^{N-1} f(a + kh) \right]$$



Integrals. Trapezoidal rule

TO SUMMARIZE: best equation to implement the trapezoidal rule

$$I = h \left[\frac{1}{2} f(a) + \frac{1}{2} f(b) + \sum_{k=1}^{N-1} f(a + k h) \right]$$

Equation to write in your scripts

Integrals. Estimate of the errors

The numerical solution of an integral is an approximation.

How can we know how large the error is?

1. do the Taylor expansion of $f(x)$ about x_{k-1} and about x_k

$$f(x) = f(x_{k-1}) + (x - x_{k-1}) f'(x_{k-1}) + \frac{1}{2}(x - x_{k-1})^2 f''(x_{k-1}) + \dots$$

$$f(x) = f(x_k) + (x - x_k) f'(x_k) + \frac{1}{2}(x - x_k)^2 f''(x_k) + \dots$$

2. rewrite the integral of the function between x_k and x_{k-1}

for its Taylor expansion

$$\int_{x_{k-1}}^{x_k} f(x) dx = f(x_{k-1}) \int_{x_{k-1}}^{x_k} dx + f'(x_{k-1}) \int_{x_{k-1}}^{x_k} (x - x_{k-1}) dx + \frac{1}{2} f''(x_{k-1}) \int_{x_{k-1}}^{x_k} (x - x_{k-1})^2 dx + \dots$$

$$\int_{x_{k-1}}^{x_k} f(x) dx = f(x_k) \int_{x_{k-1}}^{x_k} dx + f'(x_k) \int_{x_{k-1}}^{x_k} (x - x_{k-1} - h) dx + \frac{1}{2} f''(x_k) \int_{x_{k-1}}^{x_k} (x - x_{k-1} - h)^2 dx + \dots$$

Integrals. Estimate of the errors

3. After some math tricks (see lecture notes), the solution of the integrals is

$$\int_{x_{k-1}}^{x_k} f(x) dx = h f(x_{k-1}) + \frac{1}{2} h^2 f'(x_{k-1}) + \frac{1}{6} h^3 f''(x_{k-1}) + \mathcal{O}(h^4)$$

$$\int_{x_{k-1}}^{x_k} f(x) dx = h f(x_k) - \frac{1}{2} h^2 f'(x_k) + \frac{1}{6} h^3 f''(x_k) + \mathcal{O}(h^4)$$

4. take the average of the two solutions

$$\int_{x_{k-1}}^{x_k} f(x) dx = \frac{1}{2} h [f(x_{k-1}) + f(x_k)] + \frac{1}{4} h^2 [f'(x_{k-1}) - f'(x_k)] + \frac{1}{12} h^3 [f''(x_{k-1}) + f''(x_k)] + \mathcal{O}(h^4)$$

5. sum over all the slices between a and b

$$\begin{aligned} \int_a^b f(x) dx &= \sum_{k=1}^N \int_{x_{k-1}}^{x_k} f(x) dx \\ &= \frac{1}{2} h \sum_{k=1}^N [f(x_{k-1}) + f(x_k)] + \frac{1}{4} h^2 [f'(a) - f'(b)] + \frac{1}{12} h^3 \sum_{k=1}^N [f''(x_{k-1}) + f''(x_k)] + \mathcal{O}(h^4) \end{aligned}$$

Integrals. Estimate of the errors

6. Now let's compare what we have got with the trapezoid method:

TAYLOR EXPANSION TO THE THIRD ORDER IN h :

$$I = \underbrace{\frac{1}{2} h \sum_{k=1}^N [f(x_{k-1}) + f(x_k)]}_{\text{TRAPEZOID RULE}} + \frac{1}{4} h^2 [f'(a) - f'(b)] + \frac{1}{12} h^3 \sum_{k=1}^N [f''(x_{k-1}) + f''(x_k)] + \mathcal{O}(h^4)$$

$$\text{TRAPEZOID RULE: } I = h \underbrace{\left[\frac{1}{2} f(a) + \frac{1}{2} f(b) + \sum_{k=1}^{N-1} f(a + k h) \right]}_{\text{TRAPEZOID RULE}}$$

The two terms highlighted in red (depending on h) are the same
All the terms in h^2 or higher order are neglected in the trapezoid

1* the trapezoid rule is a first-order method

2* the **APPROXIMATION ERROR** depends on h

→ by reducing h we can reduce the approximation error

When h becomes comparable to the **ROUNDING ERROR**

(rounding error := the amount by which the finite
computer's representation of the number is wrong),

reducing h does not improve the calculation anymore

Integrals. Exercise: Mass of astrophysical systems

We can obtain the mass of a spherical astrophysical system (galaxy, cluster of galaxies or stellar cluster) by computing the following integral

$$M = \int_0^{r_{\max}} 4 \pi r^2 \rho(r) dr$$

with $\rho(r)$ = density distribution function (DDF)

1. SINGULAR ISOTHERMAL SPHERE:

$$\rho(r) = \frac{\sigma^2}{2 \pi G r^2}$$

where σ is the central velocity dispersion of the system and G is the gravity constant.

Calculate the mass integral for a singular isothermal sphere with the trapezoidal rule, assuming $\sigma = 10 \text{ km s}^{-1}$ (typical of a star cluster) and $r_{\max} = 10 \text{ pc}$.

The isothermal sphere is very simple but has two problems: the central divergence and the fact that mass never goes to zero at infinite distance from the centre (i.e. we must truncate it drastically).

These problems can be cured by deriving more complex versions of this density distribution function (for example the lowered King model, King 1966)

Integrals. Exercise: Mass of astrophysical systems

We can obtain the mass of an astrophysical system (galaxy, cluster of galaxies or stellar cluster) by computing the following integral

$$M = \int_0^{r_{\max}} 4 \pi r^2 \rho(r) dr$$

with $\rho(r)$ = density distribution function (DDF)

2. NAVARRO FRENK & WHITE (NFW) profile:
$$\rho(r) = \frac{\rho_0}{\frac{r}{r_s} \left(1 + \frac{r}{r_s}\right)^2}$$

where ρ_0 = density normalization, r_s = scale radius

Calculate the integral for a NFW profile with the trapezoidal rule, assuming $\rho_0 = 10^8 \text{ Msun kpc}^{-3}$, $r_s = 10 \text{ kpc}$ and $r_{\max} = 100 r_s$.

Empirically derived (from cosmological simulations) by Navarro, Frenk & White 1997 the NFW profile is a good description of dark matter haloes in Λ CDM

Integrals. Exercise: Mass of astrophysical systems

The integrals you just calculated numerically can be solved analytically (both for the isothermal sphere and for the NFW profile).

Solve the integrals analytically.

Compare the result of the numerical integration with the analytic value.

How large are the errors?

How much do they improve when reducing the step size in the integration?

Results:

Isothermal sphere: $M \sim 4.65 \times 10^5 M_{\text{sun}}$

NFW profile: $M \sim 4.56 \times 10^{12} M_{\text{sun}}$

Integrals. Monte Carlo technique for integration

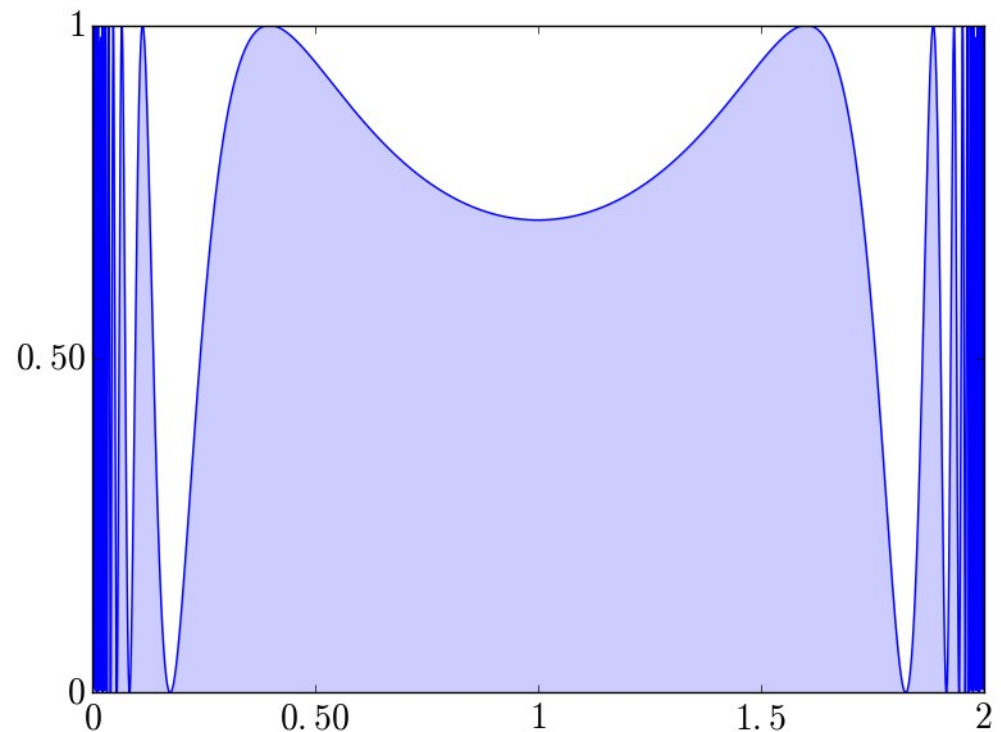
Random numbers can be used to integrate “challenging” functions, which vary too fast or require multi-dimensional integration.

Suppose we want to integrate:
$$I = \int_0^2 \sin^2 \left[\frac{1}{x(2-x)} \right] dx$$

This function varies wildly
But fits inside a rectangle with
area $A = 2 \times 1 = 2$

Thus, the value of the integral I
(= the area below the curve)
is finite and $I < A = 2$

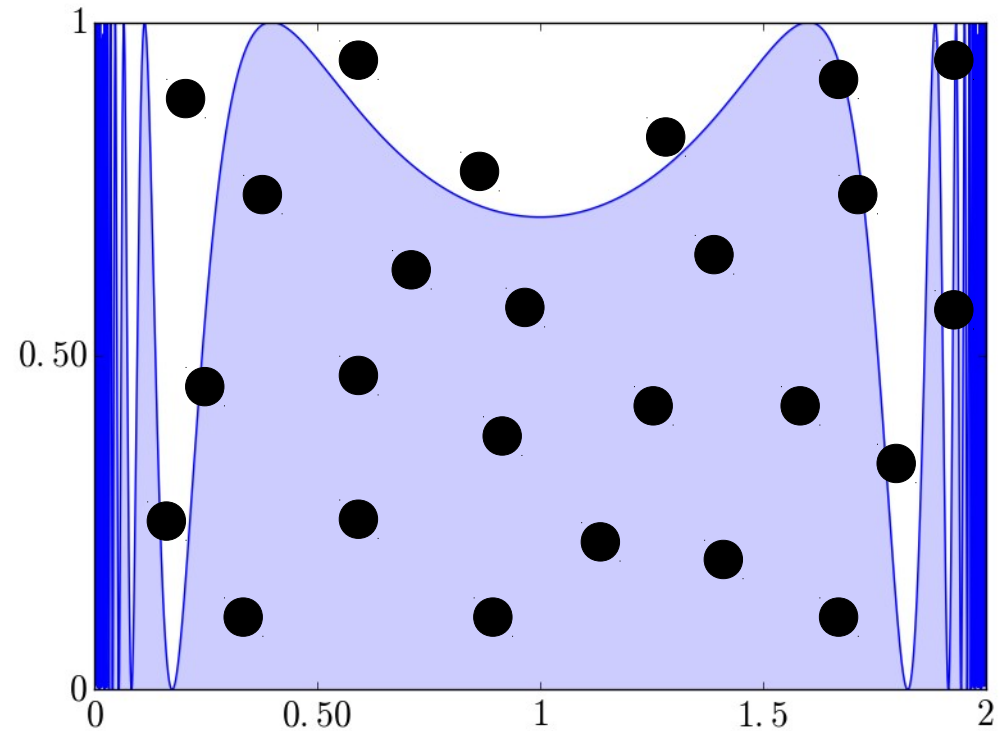
How can we calculate I ?



Integrals. Monte Carlo technique for integration

Refresh your memory on the REJECTION METHOD of random-number technique

If we choose a point uniformly at random in the rectangle A ,
the probability that that point falls in the blue shaded region is $p = I / A$



Integrals. Monte Carlo technique for integration

Refresh your memory on the REJECTION METHOD of random-number technique

If we choose a point uniformly at random in the rectangle A ,
the probability that that point falls in the blue shaded region is $p = I / A$

Here is the simplest scheme:

1. Generate N random points uniform in the rectangle A

This means that you have to uniformly draw both x and y independent of each other

2. Check for each of the random if it is below the curve

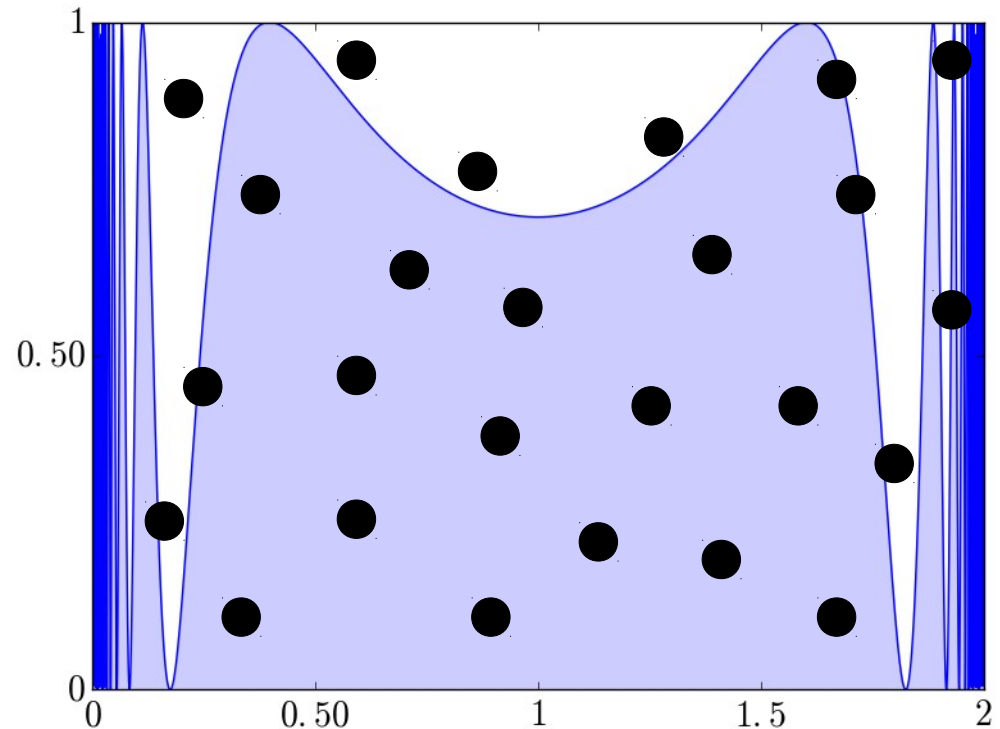
3. Count the number of points below the curve (k):

N random points, k points below the curve

$$\rightarrow k / N \sim p$$

4. Hence the integral is simply

$$I \simeq \frac{k}{N} A$$



Integrals. Exercise: simplest Monte Carlo integral

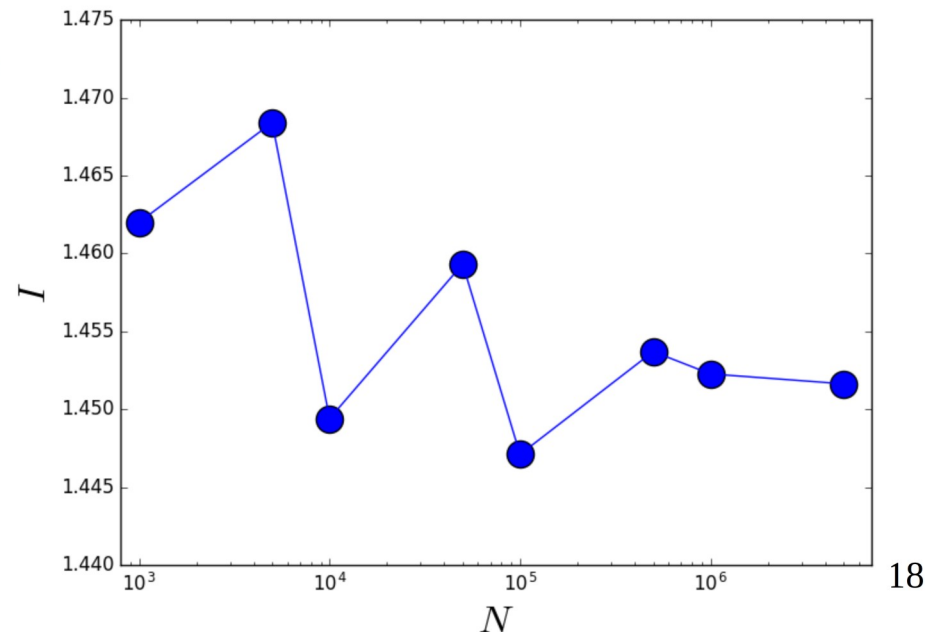
EXERCISE:

Write a python script to perform the integration in equation 114 of the integral in equation 113. Calculate the integral with $N = 10^3, 5 \times 10^3, 10^4, 5 \times 10^4, 10^5, 5 \times 10^5, 10^6, 5 \times 10^6$. Estimate and plot the difference you obtain by repeating this exercise with different values of N (the plot will be N on the x axis and I on the y -axis). The result should look like Figure 37 ($I \sim 1.45$).

Note: we should generate two random numbers per each point: one along the x -axis and one along the y -axis.

$$I \simeq \frac{k}{N} A$$

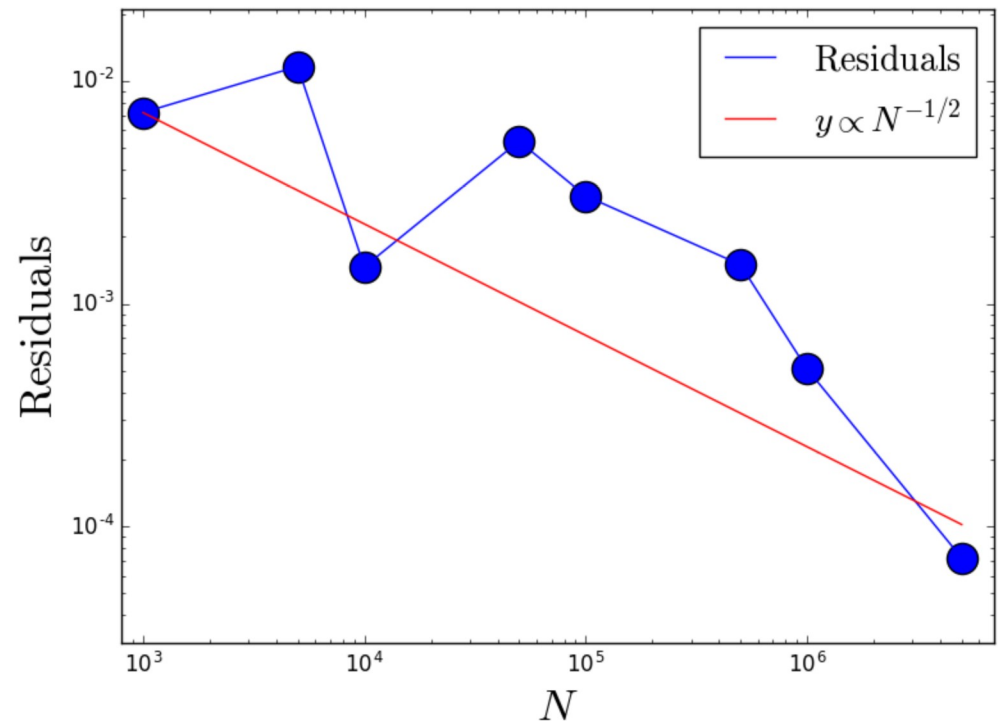
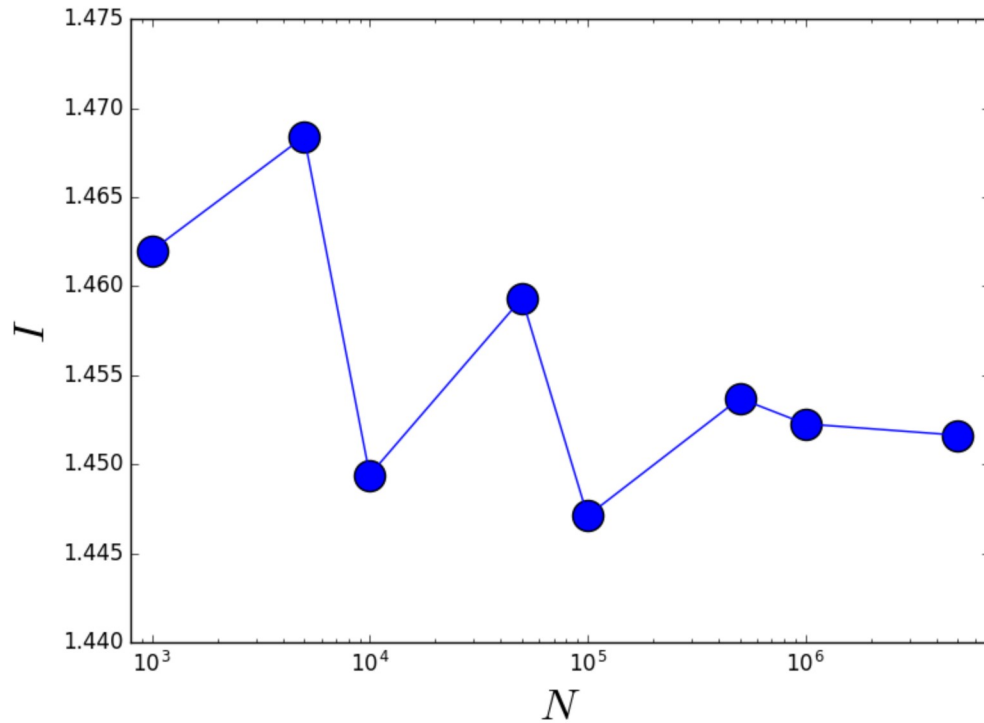
$$I = \int_0^2 \sin^2 \left[\frac{1}{x(2-x)} \right] dx$$



Integrals. Exercise: simplest Monte Carlo integral

The previous exercise teaches us that the error scales with N as $N^{-1/2}$

i.e. the more random numbers we take the better the accuracy



Integrals. The mean value method

Let's consider our general integral $I = \int_a^b f(x) dx$

We can write the average value of $f(x)$ $\langle f \rangle = \frac{1}{b-a} \int_a^b f(x) dx$

Hence the integral can be written as $I = (b-a) \langle f \rangle$

→ If we can evaluate the average of $f(x)$, then we can find I

Let's use the random numbers: calculate $f(x_1), f(x_2), f(x_3), \dots, f(x_N)$

at N points $x_1, x_2, x_3, \dots, x_N$ randomly drawn uniformly between a and b

$$\langle f \rangle = N^{-1} \sum_{i=1}^N f(x_i)$$

→ The formula of the mean value method:

$$I = \frac{(b-a)}{N} \sum_{i=1}^N f(x_i)$$

It can be shown that error scales with $N^{-1/2}$,
but the error is much smaller than previous method

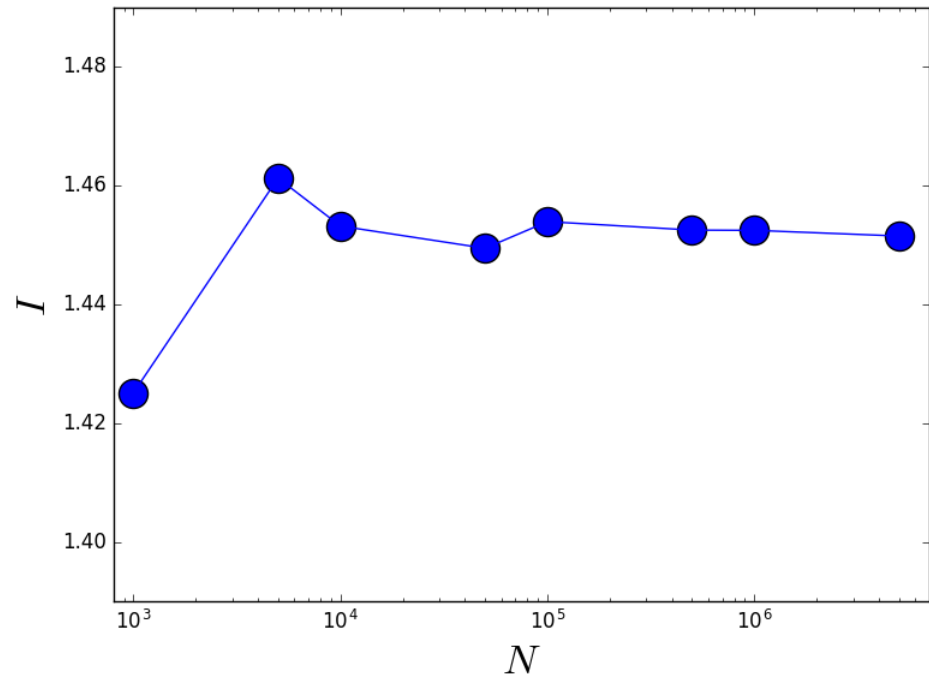
Integrals. Exercise: mean value method

EXERCISE:

Write a python script to perform the integration with the mean value method (equation 118) of the integral in equation 113. Calculate the integral with $N = 10^3, 5 \times 10^3, 10^4, 5 \times 10^4, 10^5, 5 \times 10^5, 10^6, 5 \times 10^6$. Estimate and plot the difference you obtain by repeating this exercise with different values of N (the plot will be N on the x axis and I on the y -axis). The results should look like Figure 38 ($I \sim 1.452$).

$$I = \frac{(b - a)}{N} \sum_{i=1}^N f(x_i)$$

$$I = \int_0^2 \sin^2 \left[\frac{1}{x(2-x)} \right] dx$$



Integrals. Exercise: mean value method for Isothermal sphere and NFW profile

EXERCISE:

Use the scripts you have produced to perform Monte Carlo integration with mean value to integrate the mass of a stellar system according to the singular isothermal sphere and to the NFW density profiles (as you already did for the trapezoidal rule).

Integrals. Integrals in many dimensions

The mean value method can be used for multi-dimensional integrals and is very efficient

Integral of a function $f(x)$ over a volume V in a high-dimensional space:

$$I \simeq \frac{V}{N} \sum_{i=1}^N f(\mathbf{r}_i)$$

where the points \mathbf{r}_i are picked uniformly at random from the volume V

Integrals. Importance Sampling

The main drawback of the mean value method is that it performs rather poorly if the function to integrate is **PATHOLOGICAL** (e.g. contains a divergence). For example

$$I = \int_0^1 \frac{x^{-1/2}}{e^x + 1} dx$$

which diverges for $x = 0$

If you try to solve it with the mean value method you run into troubles because the bins around 0 give a very large contribution to the sum.

You want to find a new Monte Carlo method able to understand where the function becomes pathological and to "get rid" of the pathology

→ **IMPORTANCE SAMPLING basic idea:**
a similar algorithm to the mean value method, but we substitute the uniformly drawn N random points x_i with non-uniformly drawn random points

Integrals. Importance Sampling

For any general function $g(x)$ we can define a weighted average over the interval from a to b as

$$\langle g \rangle_w = \frac{\int_a^b w(x) g(x) dx}{\int_a^b w(x) dx}$$

where $w(x)$ is a generic function (the weighting function)

Integrals. Importance Sampling

For any general function $g(x)$ we can define a weighted average over the interval from a to b as

$$\langle g \rangle_w = \frac{\int_a^b w(x) g(x) dx}{\int_a^b w(x) dx}$$

where $w(x)$ is a generic function (the weighting function)

If we set $g(x) = f(x)/w(x)$

$$\left\langle \frac{f(x)}{w(x)} \right\rangle_w = \frac{\int_a^b f(x) dx}{\int_a^b w(x) dx}$$

Integrals. Importance Sampling

For any general function $g(x)$ we can define a weighted average over the interval from a to b as

$$\langle g \rangle_w = \frac{\int_a^b w(x) g(x) dx}{\int_a^b w(x) dx}$$

where $w(x)$ is a generic function (the weighting function)

If we set $g(x) = f(x)/w(x)$

$$\left\langle \frac{f(x)}{w(x)} \right\rangle_w = \frac{\int_a^b f(x) dx}{\int_a^b w(x) dx}$$

From $I = \int_a^b f(x) dx$ it follows that we can write

$$\left\langle \frac{f(x)}{w(x)} \right\rangle_w = \frac{I}{\int_a^b w(x) dx}$$

Integrals. Importance Sampling

Hence the integral can be written in the form

$$I = \left\langle \frac{f(x)}{w(x)} \right\rangle_w \int_a^b w(x) dx$$

Let's compare it with the formula of the mean value method:

$$I = \frac{(b-a)}{N} \sum_{i=1}^N f(x_i)$$

The two equations are very similar

But the importance sampling formula above **CONTAINS a WEIGHTED MEAN** instead of a simple average

How can we calculate the weighted mean?

Integrals. Importance Sampling

How can we calculate the weighted mean?

Let's define a probability density function $p(x) = \frac{w(x)}{\int_a^b w(x) dx}$

which is the function $w(x)$ normalized so that its integral is 1.

Let us sample N random points x_i non uniformly with this PDF.

The probability of generating a value in the interval between x and $x+dx$ is $p(x) dx$.

If I draw N random numbers, the average number of samples that fall in this interval is $N p(x) dx$ and so for any function $g(x)$

$$\sum_{i=1}^N g(x_i) \simeq \int_a^b N p(x) g(x) dx$$

Integrals. Importance Sampling

Substituting $p(x) = \frac{w(x)}{\int_a^b w(x) dx}$ **into** $\langle g \rangle_w = \frac{\int_a^b w(x) g(x) dx}{\int_a^b w(x) dx}$

we get $\langle g \rangle_w = \int_a^b p(x) g(x) dx$

Substituting $\sum_{i=1}^N g(x_i) \simeq \int_a^b N p(x) g(x) dx$ **into the above eq,**
we get

$$\langle g \rangle_w = \int_a^b p(x) g(x) dx \simeq \frac{1}{N} \sum_{i=1}^N g(x_i)$$

Assuming $g(x) = f(x)/w(x)$ **and substituting the above equation into**

$I = \left\langle \frac{f(x)}{w(x)} \right\rangle_w \int_a^b w(x) dx$ **We finally get**

$$I = \frac{1}{N} \sum_{i=1}^N \frac{f(x_i)}{w(x_i)} \int_a^b w(x) dx$$

Integrals. Importance Sampling

Fundamental equation of importance sampling (IS):

$$I = \frac{1}{N} \sum_{i=1}^N \frac{f(x_i)}{w(x_i)} \int_a^b w(x) dx$$

Fundamental equation of mean value method (MVM):

$$I = \frac{(b-a)}{N} \sum_{i=1}^N f(x_i)$$

Main difference:

* in the MVM, I calculate $f(x)$ for uniformly drawn random numbers

* in the IS, I calculate $f(x)$ for non-uniformly drawn random numbers, distributed according to the PDF

$$p(x) = \frac{w(x)}{\int_a^b w(x) dx}$$

Integrals. Importance Sampling

Importance of the importance sampling (IS):

I can choose a function $w(x)$ so that I get rid of the divergence of $f(x)$

Example:
$$I = \int_0^1 \frac{x^{-1/2}}{e^x + 1} dx$$

Good choice of $w(x)$:
$$w(x) = x^{-1/2}$$

Implications:

1. $f(x)/w(x) = (\exp x + 1)^{-1}$ has no divergence

2. $\int_a^b w(x) dx = \int_0^1 x^{-1/2} dx = 2\sqrt{x}\Big|_0^1 = 2$

3. draw x_i random numbers from $p(x) = \frac{x^{-1/2}}{\int_0^1 x^{-1/2} dx} = \frac{1}{2} x^{-1/2}$

4. finally, with the inverse random sampling

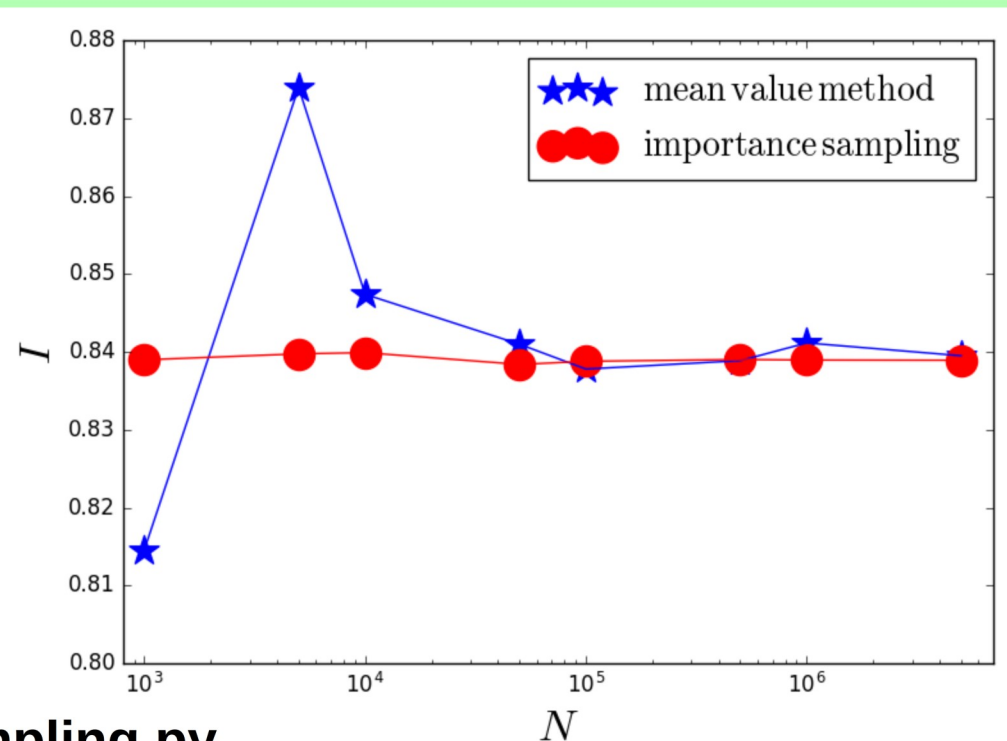
$$y = \int_0^x p(x) dx = \frac{1}{2} \int_0^x x^{-1/2} dx = x^{1/2}$$
$$\rightarrow x = y^2$$

Integrals. Exercise: Importance Sampling

EXERCISE:

Write a python script to perform the integration in equation 120 with the importance sampling ($I \sim 0.839$). Calculate the integral with $N = 10^3, 5 \times 10^3, 10^4, 5 \times 10^4, 10^5, 5 \times 10^5, 10^6, 5 \times 10^6$. Evaluate the same integration, for the same values of N , with your script that makes use of the mean value method. Compare the results. The result should look like Figure 39. The importance sampling performs very well already for a small value of N .

$$I = \frac{1}{N} \sum_{i=1}^N \frac{f(x_i)}{w(x_i)} \int_a^b w(x) dx$$



Integrals. Scipy.integrate functions

1. Methods for integrating functions given fixed samples

The python function takes as argument an array of points $f(x_i)$ (i.e. a fixed sample) and the corresponding array x_i over which it performs the integration. Same as this lecture.

Functions

trapz: trapezoid rule

cumtrapz: cumulative integral with trapezoidal rule

simps: Simpson's rule

romb: Romberg integration

See e.g. Mark Newman book for algorithms we didn't have time to discuss

Integrals. Scipy.integrate functions

Example of trapz:
examples/integral/trapz.py

```
#how to use the trapz function of scipy.integrate
#applied to the NFW profile
#it calculates equation 102 of the lecture notes
#examples/integral/trapz.py

from scipy.integrate import trapz
import numpy as np

G=6.667e-8 #gravity constant in cgs
pc=3.086e18 # 1 pc in cgs
msun=1.989e33 #solar mass in cgs

rs=10.*1e3*pc #10 kpc in cgs
rmax=100.*rs
rho0=1e8*msun/(1e3*pc)**3 #1e8 Msun/kpc^3 in cgs

def NFW(r):
    x=r/rs
    rho=rho0/((1.+x)**2)
    mass=4.*np.pi*rs*r*rho
    return mass

b=rmax
a=0.0
intervallo=(b-a)
N=int(1e6)
h=intervallo/N

trapzx=[a]
trapzy=[NFW(a)]
for i in range(1,N-1,1):
    trapzx.append(a+i*h)
    trapzy.append(NFW(a+i*h))
trapzx.append(b)
trapzy.append(NFW(b))

II=trapz(trapzy,trapzx)
print("with scipy.integrate.trapz I= ",II/msun," Msun")
```

Integrals. Scipy.integrate functions

2. Methods for integrating functions given function object.

The python function takes the function you want to integrate as an argument, plus the integration range, and decides in which points x_i to evaluate the function. This is exactly what we have done for the look-backtime in the example `examples/python/lookback.py`

Functions

quad	General purpose integration (uses a technique from the Fortran library QUADPACK: contains different algorithms to solve integrals)
dblquad	General purpose integration in two dimensions (two variables)
tplquad	General purpose integration in three dimensions (three variables)
fixed_quad	Integrate $\text{func}(x)$ using Gaussian quadrature of order N
quadrature	Integrate with given tolerance using Gaussian quadrature
romberg	Integrate $\text{func}(x)$ using Romberg integration

See e.g. Mark Newman book for algorithms we didn't have time to discuss

Integrals. Scipy.integrate functions

Example of quad: `examples/integral/lookback.py`

```
ltime=integrate.quad(integrand, 0.0, z, epsrel=1.e-13)
```

First argument: `integrand`, function to be integrated

Second and third argument: `0.0, z`, extremes of the interval

`epsrel = 1.e-13` maximum RELATIVE integration error tolerated by quad

The integrand function can be either defined with `def`

```
OmegaM=0.2726 #omega matter, parameter from cosmology
OmegaL= 0.7274 #omega lambda, parameter from cosmology

def integrand(x):
    r=1./((1.+x)*(OmegaM*(1.+x)**3.+OmegaL)**0.5)
    return r
```

Or written in compact form with `lambda`:

```
integrand = lambda x: 1./((1.+x)*(OmegaM*(1.+x)**3.+OmegaL)**0.5)
```