Computational methods for Medical Physics

Lecture 1
Monte Carlo Basics

Dr. George DEDES

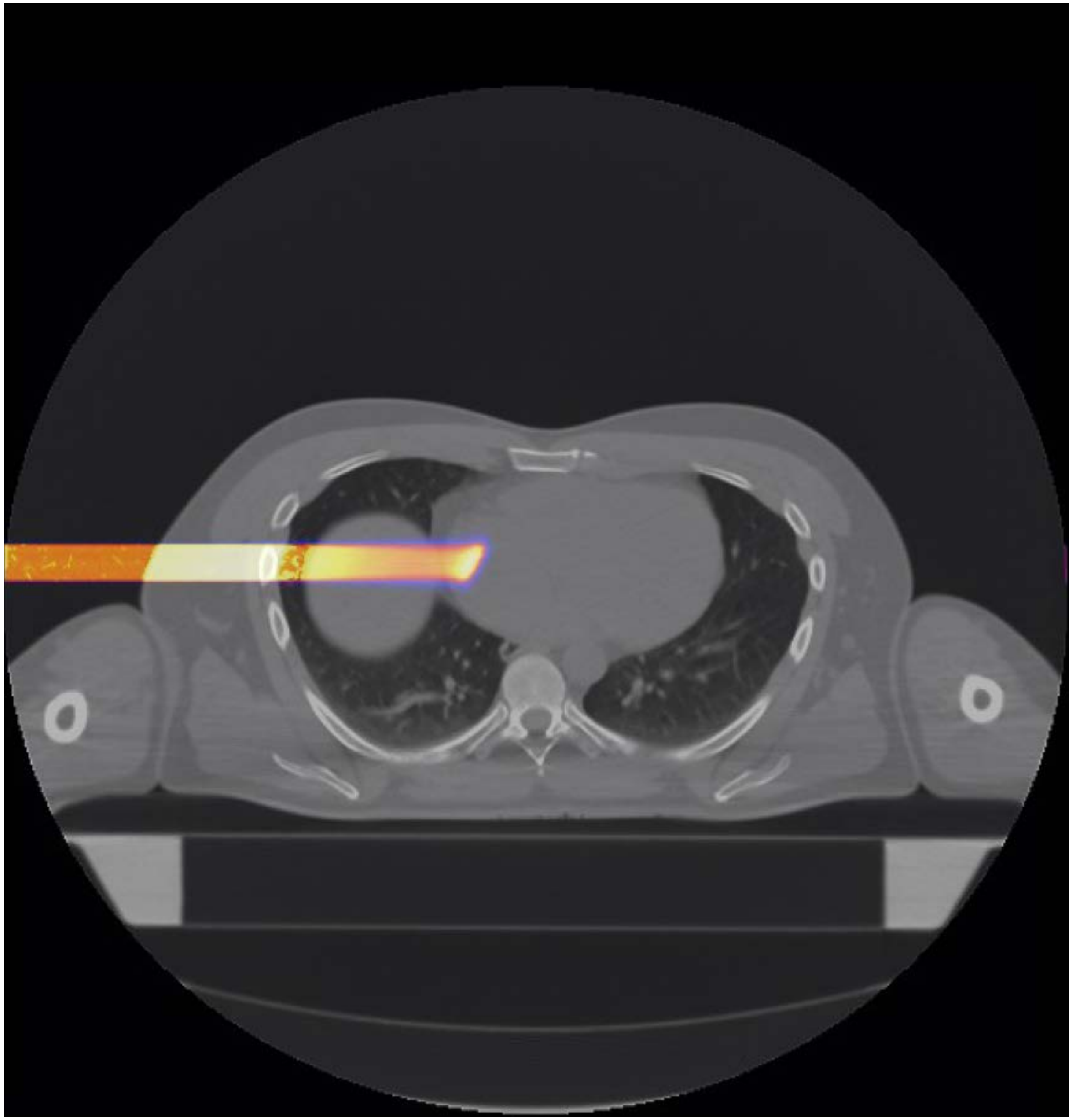
WS 2016-2017

LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

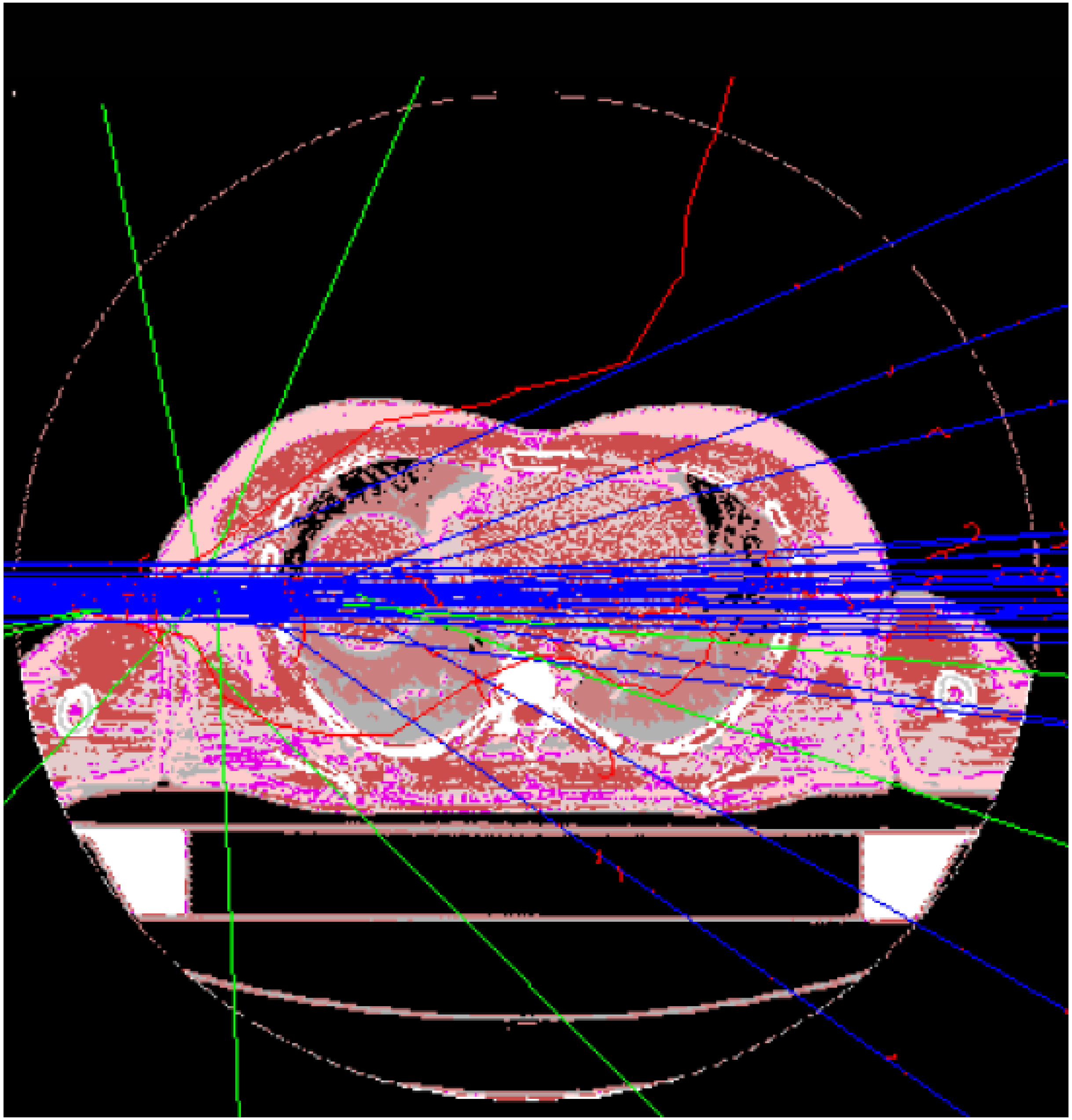
Past and next lectures’ content

- Next few lectures:
  - Monte Carlo technique basics
  - Simulation of particle transport using Monte Carlo

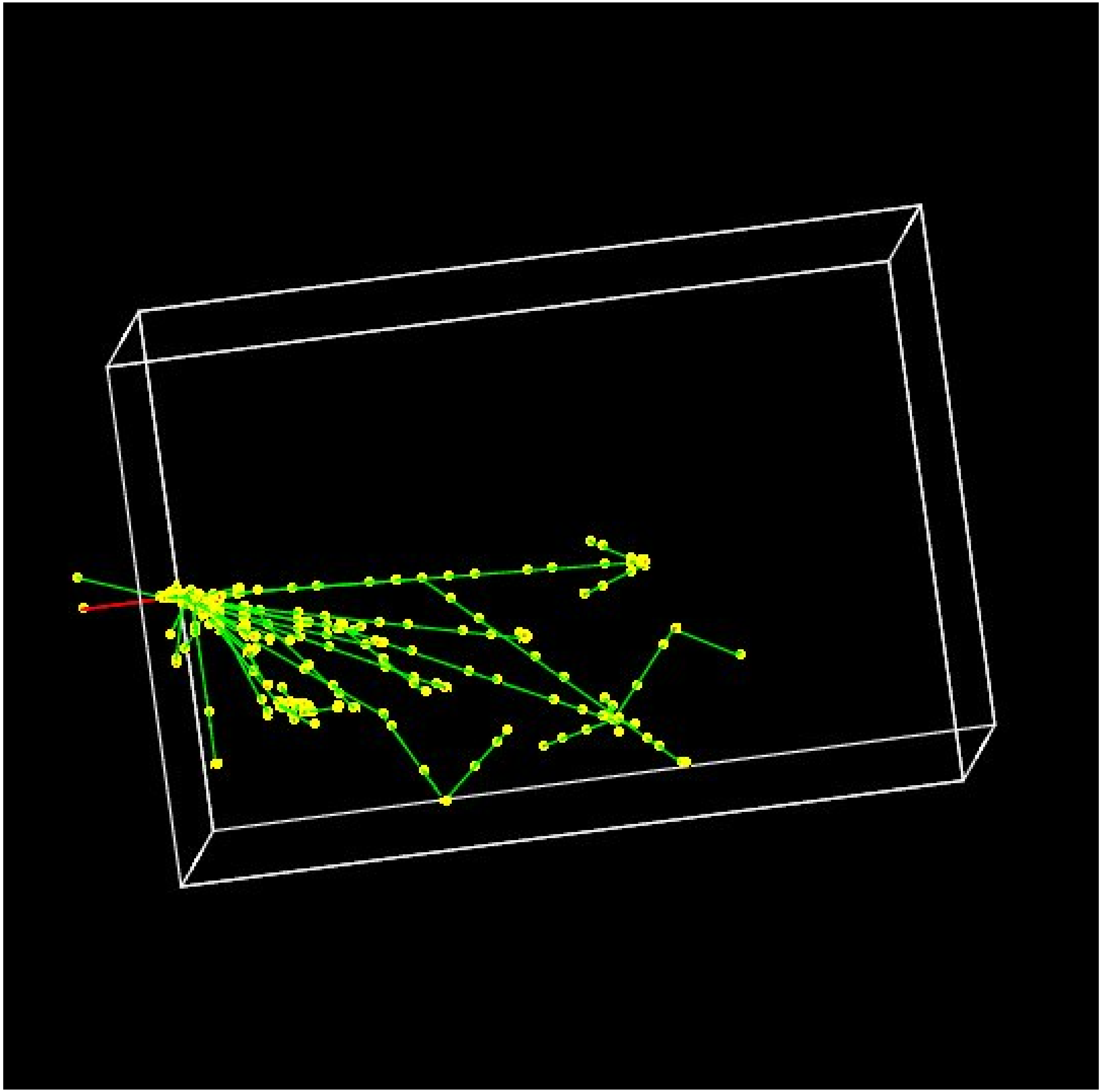
Motivation



Motivation



Motivation

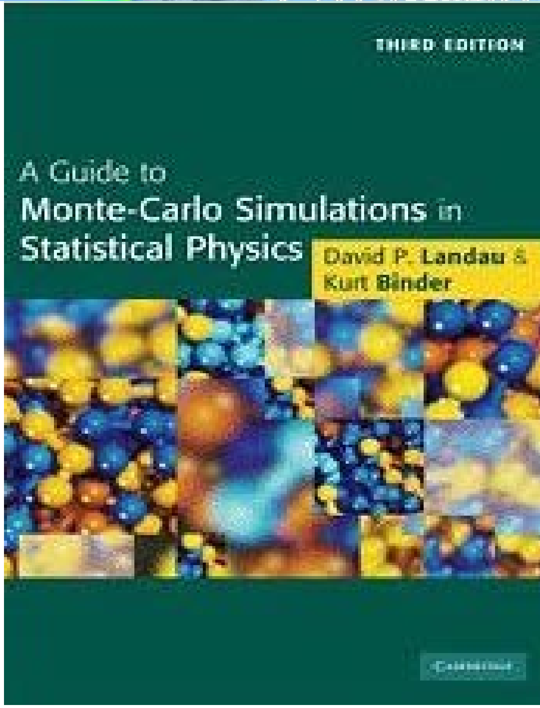


External material

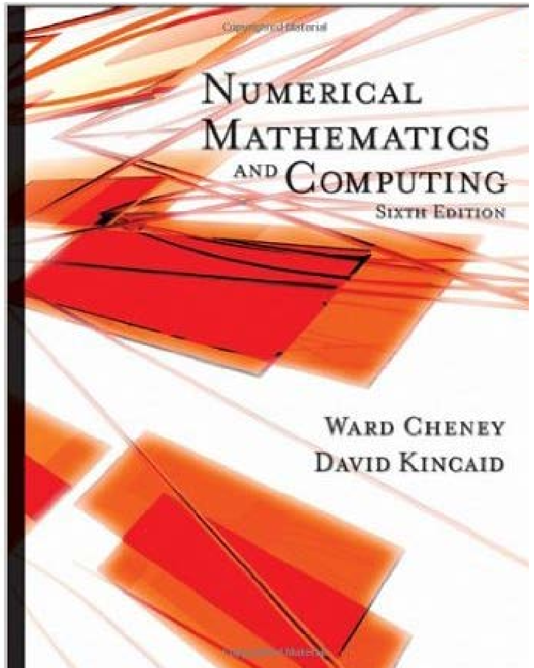
- Material form this lecture was taken from the following lectures:
  - Aalto University School of Science
    Department of Biomedical Engineering and Computational Science
    Lecture on Computational Science
    Dr. Ricu Linna
    Dr. Laura Juvonen
  - University of Helsinki
    Department of Physics
    Basics of Monte Carlo Simulations Lecture
    Priv. Doz. Dr. Flyura Djurabekova

External material

- Books:
  - A Guide to Monte Carlo Simulations in Statistical Physics
    David P. Landau, Kurt Binder



  - Numerical Mathematics and Computing
    Ward Cheney, David Kincaid



- Monte Carlo is a(n interesting) city

- It is also a (hopefully interesting) technique

- It is also a (hopefully interesting) technique
- The question for this lecture is: MC is an interesting technique for what?

- It is also a (hopefully interesting) technique
- The question for this lecture is: MC is an interesting technique for what?
- To answer that, we will start from:
  - Some historical information about MC technique
  - Integration (numerical/MC)

- Probability theory started being used in order to understand games of chance
  - Pierre de Fermat and Blaise Pascal were asked by a professional gambler to solve some of his questions on dice games
  - In 1654 they solved the problem by using probabilities calculation
  - In 1657 the first study on probability theory was written by Christiaan Huygens (On Reasoning in Games of Chance)

- Probability theory started being used in order to understand games of chance
  - Pierre de Fermat and Blaise Pascal were asked by a professional gambler to solve some of his questions on dice games
  - In 1654 they solved the problem by using probabilities calculation
  - In 1657 the first study on probability theory was written by Christiaan Huygens (On Reasoning in Games of Chance)

- In 1777 Georges-Louis Leclerc, comte de Buffon stated his needle problem
  - When throwing needles on a floor marked by parallel lines, what is the probability that a needle will land on one of those lines
  - The first problem solved on geometrical probability using integral geometry can be also used to calculate π

- Probability theory started being used in order to understand games of chance
  - Pierre de Fermat and Blaise Pascal were asked by a professional gambler to solve some of his questions on dice games
  - In 1654 they solved the problem by using probabilities calculation
  - In 1657 the first study on probability theory was written by Christiaan Huygens (On Reasoning in Games of Chance)

- In 1777 Georges-Louis Leclerc, comte de Buffon stated his needle problem
  - When throwing needles on a floor marked by parallel lines, what is the probability that a needle will land on one of those lines
  - The first problem solved on geometrical probability using integral geometry can be also used to calculate π

- Work continued by numerous prominent scientists and mathematicians, such as Jacob Bernoulli, Abraham de Moivre, James Stirling, Leonhard Euler, Joseph-Louis Lagrange, Jean-Baptiste le Rond d’Alembert, Pierre-Simon marquis de Laplace

Monte Carlo
Historical review

- Probability theory started being used in order to understand games of chance
  - Pierre de Fermat and Blaise Pascal were asked by a professional gambler to solve some of his questions on dice games
  - In 1654 they solved the problem by using probabilities calculation
  - In 1657 the first study on probability theory was written by Christiaan Huygens (On Reasoning in Games of Chance)

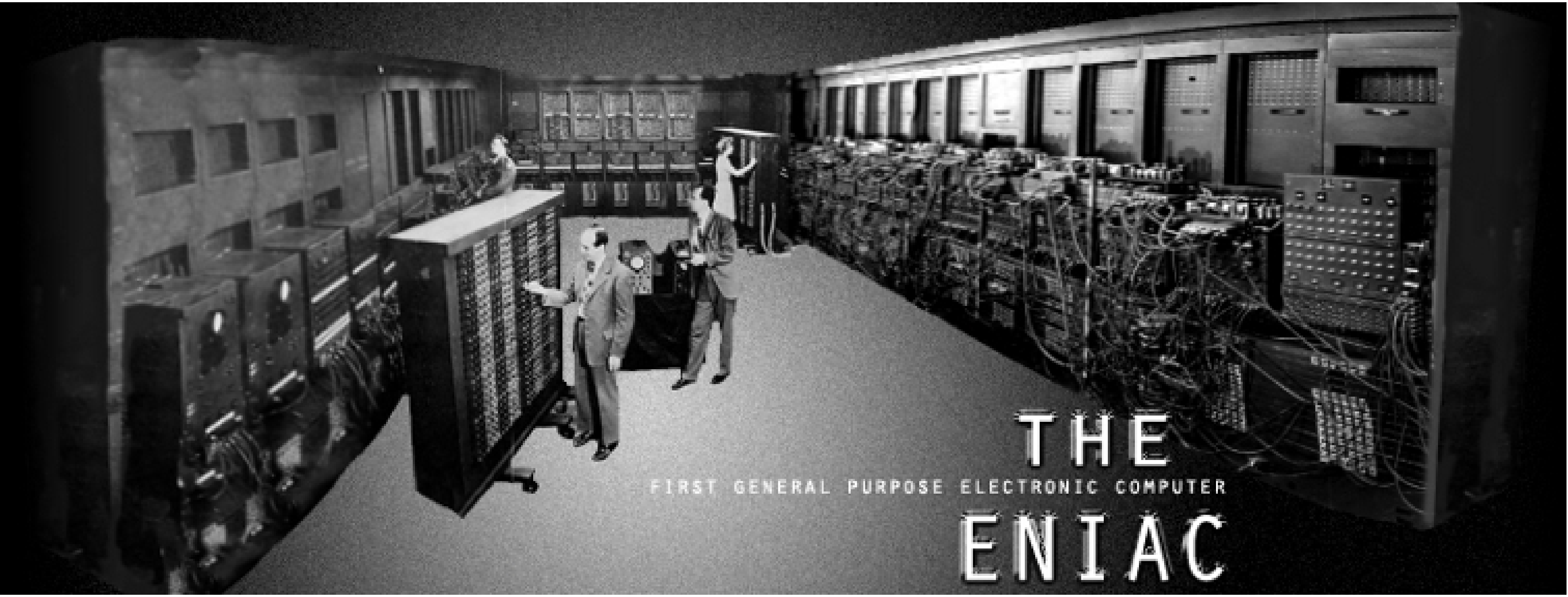
- In 1777 Georges-Louis Leclerc, comte de Buffon stated his needle problem
  - When throwing needles on a floor marked by parallel lines, what is the probability that a needle will land on one of those lines
  - The first problem solved on geometrical probability using integral geometry can be also used to calculate π

- Work continued by numerous prominent scientists and mathematicians, such as Jacob Bernoulli, Abraham de Moivre, James Stirling, Leonhard Euler, Joseph-Louis Lagrange, Jean-Baptiste le Rond d'Alembert, Pierre-Simon marquis de Laplace

- In 1812 Laplace published his "*Théorie analytique des probabilités*"

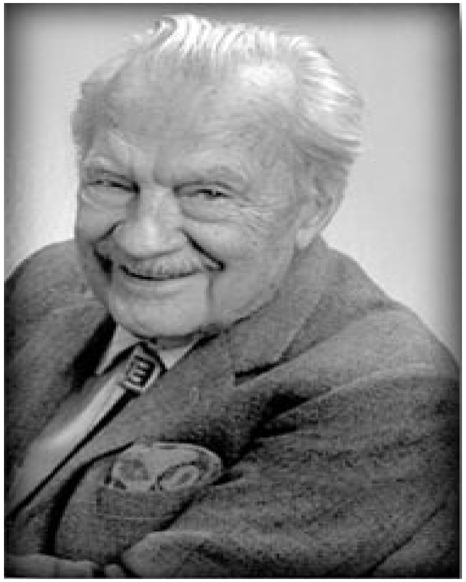
- In 1945, ENIAC the first electronic general-purpose computer was built

- In 1945, ENIAC the first electronic general-purpose computer was built
- 18.000 vacuum tubes and 500.000 shoulder joints!

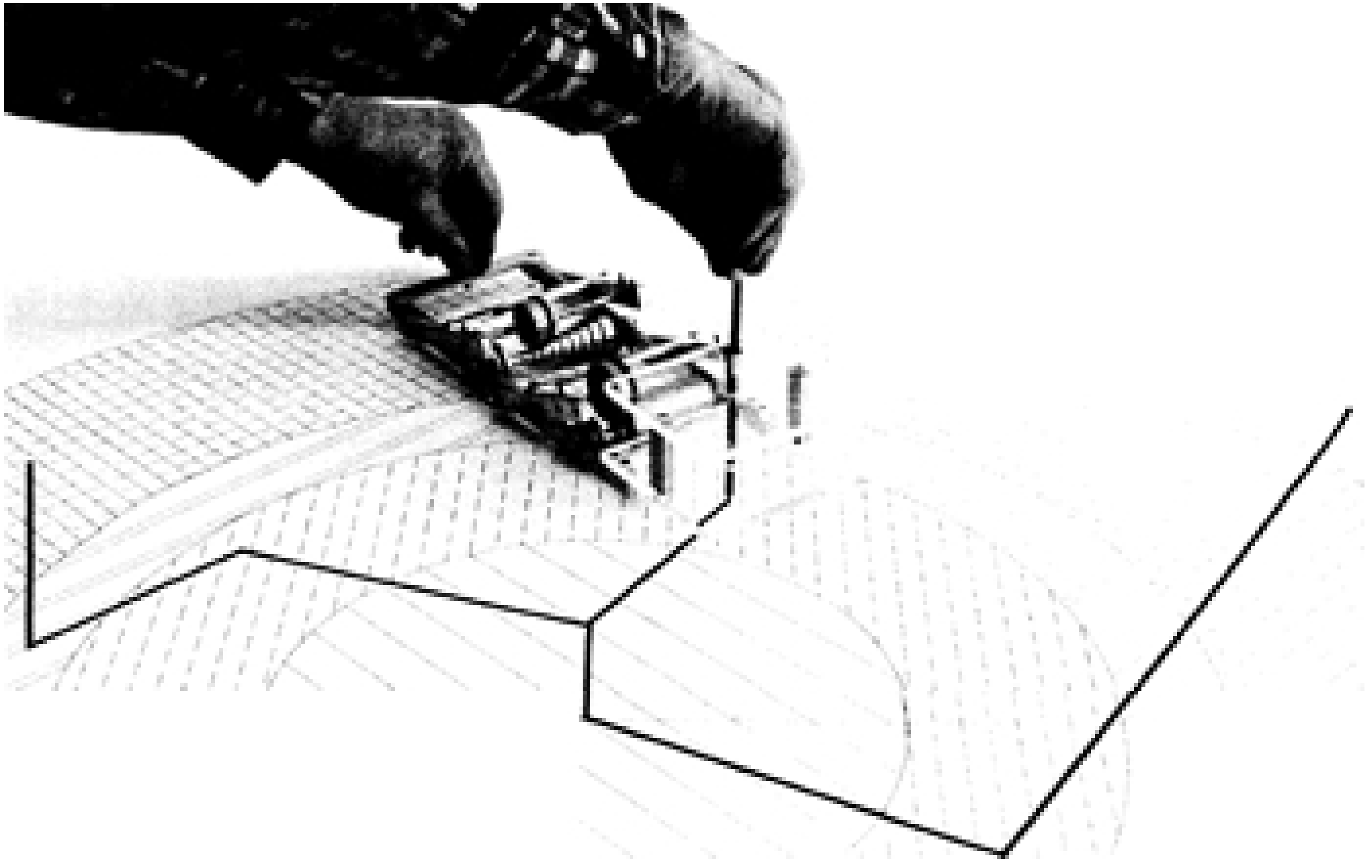


- In 1945, ENIAC the first electronic general-purpose computer was built
- In 1946 there was the idea to solve thermonuclear problems (neutron diffusion in fissile material) by using statistical sampling techniques
- The “Monte Carlo” method was proposed by Nicholas Metropolis, Stanislaw Ulam, John von Neumann (Los Alamos)
- Enrico Fermi had independently discovered the method in the 30s, but never published it (Rome)

- In 1945, ENIAC the first electronic general-purpose computer was built
- In 1946 there was the idea to solve thermonuclear problems (neutron diffusion in fissile material) by using statistical sampling techniques
- The "Monte Carlo" method was proposed by Nicholas Metropolis, Stanislaw Ulam, John von Neumann (Los Alamos)
- Enrico Fermi had independently discovered the method in the 30s, but never published it (Rome)

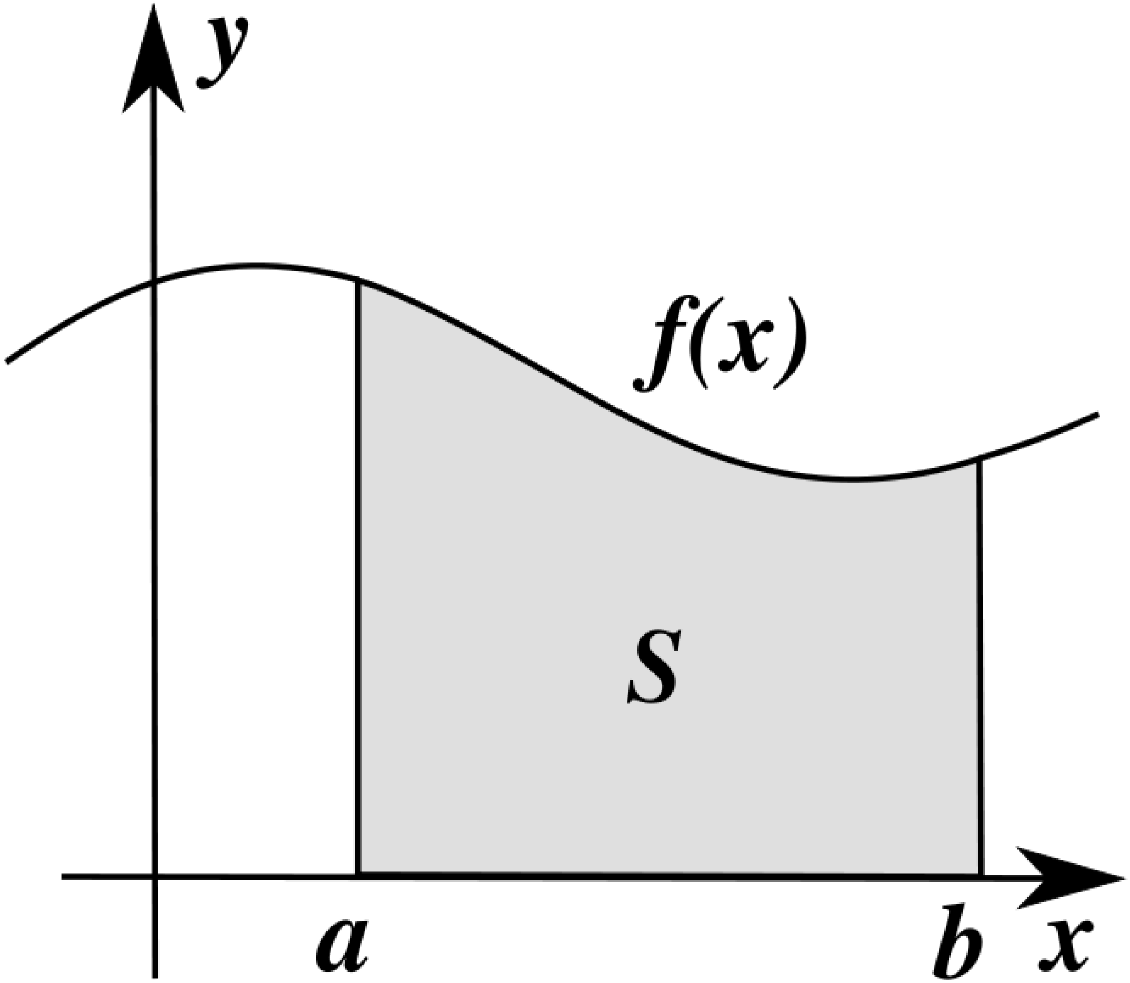
   

- An analog computer based on MC modeling of neutron transport in different materials

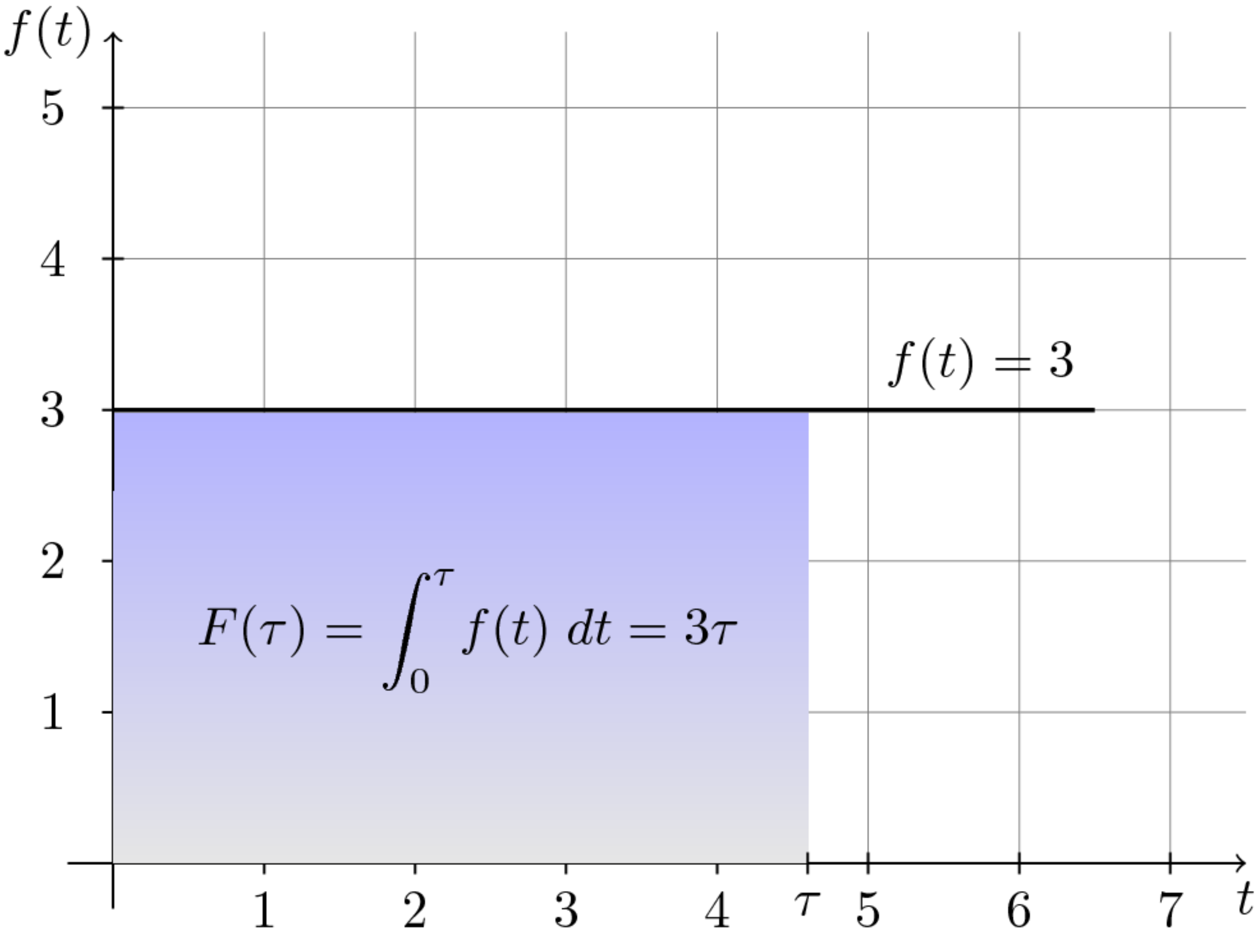


Integrals

- A single integral represents the signed area between the function and the axis where the integration variable is represented

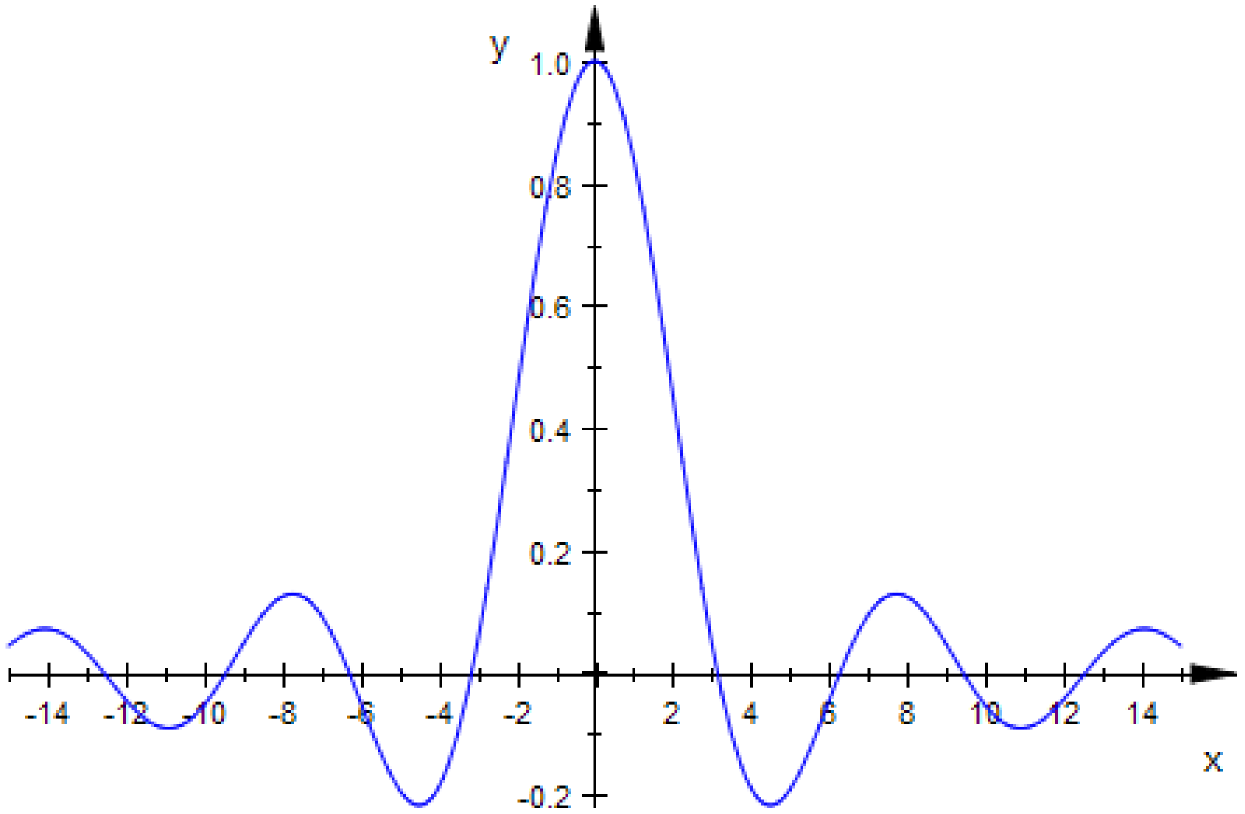


- There is a number of relatively simple functions that can be integrated analytically



Integrals

- Unfortunately, quite often the most interesting integrals cannot be solved analytically



- Such an example from a common physics problem is the magnetic field induced by a current flowing in a circular loop:

$$\boxed{H(x) = \frac{4Ir}{r^2 - x^2}\int_0^{\pi/2}\left[1 - \left(\frac{x}{r}\right)^2 sin^2\,\theta\right]^{1/2} d\theta}$$

- There is no analytical solution to the above integral
- We have to rely on numerical integration

- A non-analytically derivable definite integral can be estimated by means of lower and upper sums

- Assuming a function f and the area between f and the x-axis for xϵ[a,b]

  - We can partition [a,b] to n subintervals

$$\boxed{P = \{a = x_0 < x_1 < \cdots < x_{n-1} < x_n = b\}}$$

- A non-analytically derivable definite integral can be estimated by means of lower and upper sums

- Assuming a function f and the area between f and the x-axis for x϶[a,b]

  - We can partition [a,b] to n subintervals

$$P = \{a = x_0 < x_1 < \cdots < x_{n-1} < x_n = b\}$$

  - With m_i being the greatest lower bound and M_i the least upper bound

$$m_i = inf\{f(x)\colon x_i < x < x_{i+1}\}$$

$$M_i = sup\{f(x)\colon x_i < x < x_{i+1}\}$$

- A non-analytically derivable definite integral can be estimated by means of lower and upper sums
- Assuming a function f and the area between f and the x-axis for x϶[a,b]
  - We can partition [a,b] to n subintervals

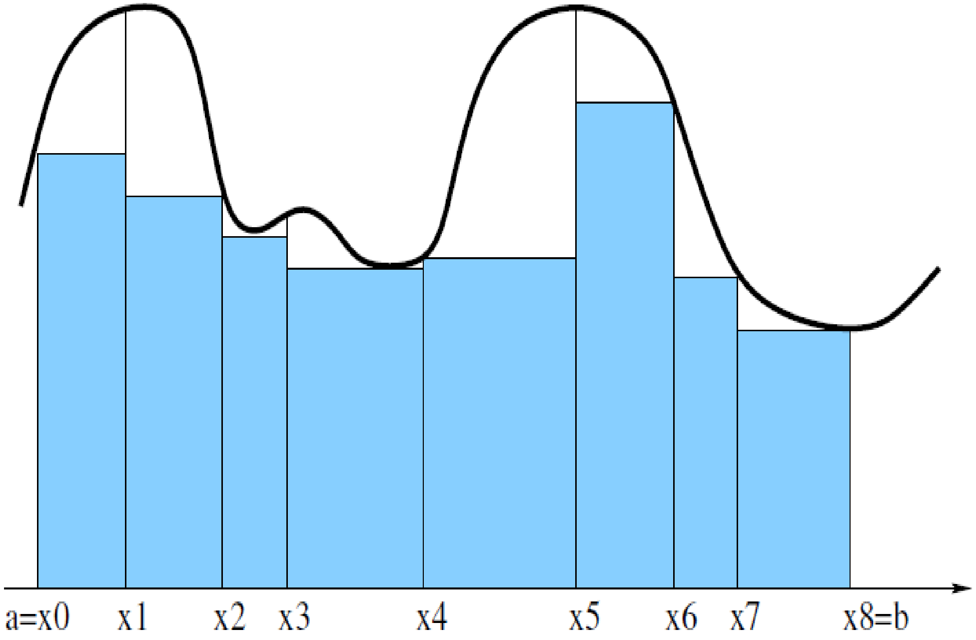
$$\boxed{P = \{a = x_0 < x_1 < \cdots < x_{n-1} < x_n = b\}}$$

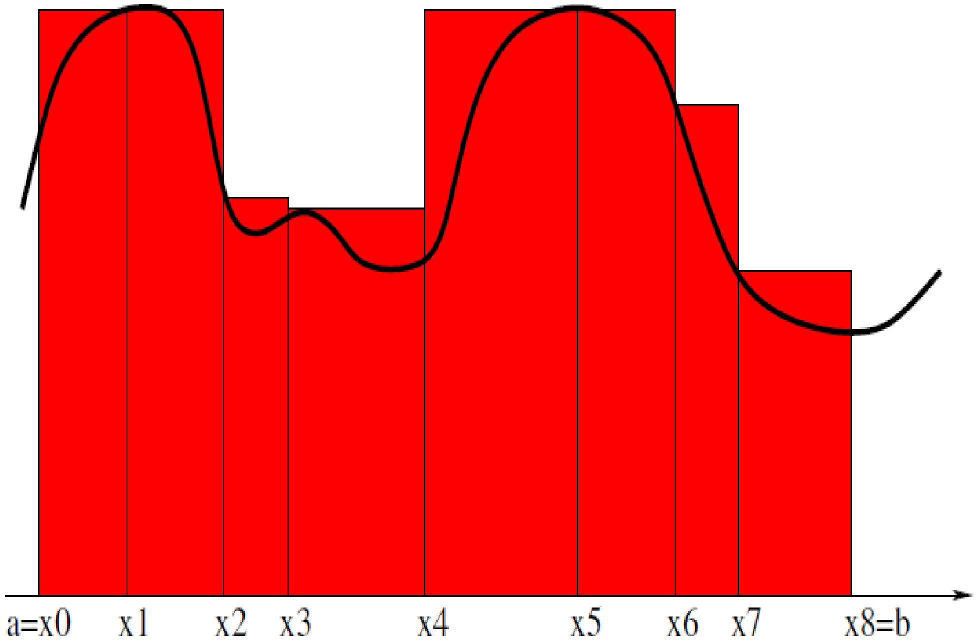
  - The lower sums (*L*) and upper sums (*U*)

$$\boxed{L(f;P) = \sum_{i=0}^{n-1} m_i\,(x_{i+1} - x_i)}$$

$$\boxed{U(f;P) = \sum_{i=0}^{n-1} M_i\,(x_{i+1} - x_i)}$$

- A non-analytically derivable definite integral can be estimated by means of lower and upper sums





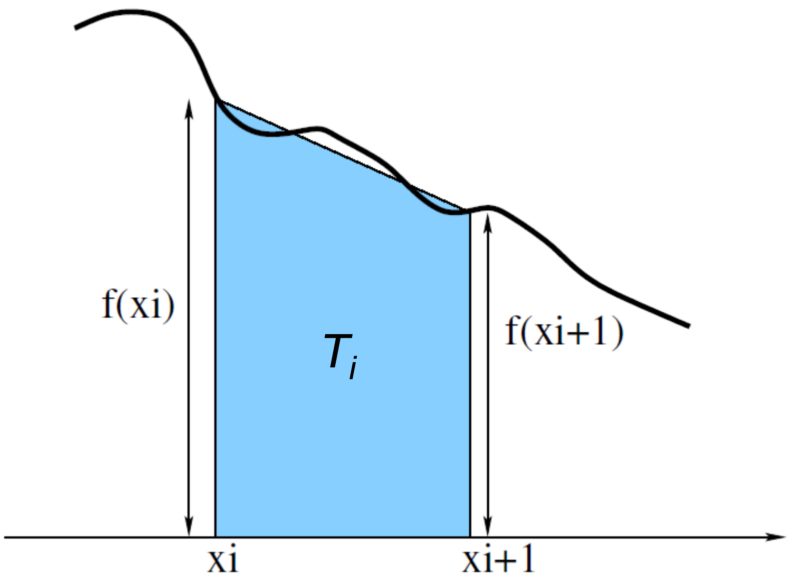
$$L(f;P) = \sum_{i=0}^{n-1} m_i \left(x_{i+1} - x_i\right)$$

$$U(f;P) = \sum_{i=0}^{n-1} M_i \left(x_{i+1} - x_i\right)$$

Numerical integration
Trapezoid rule

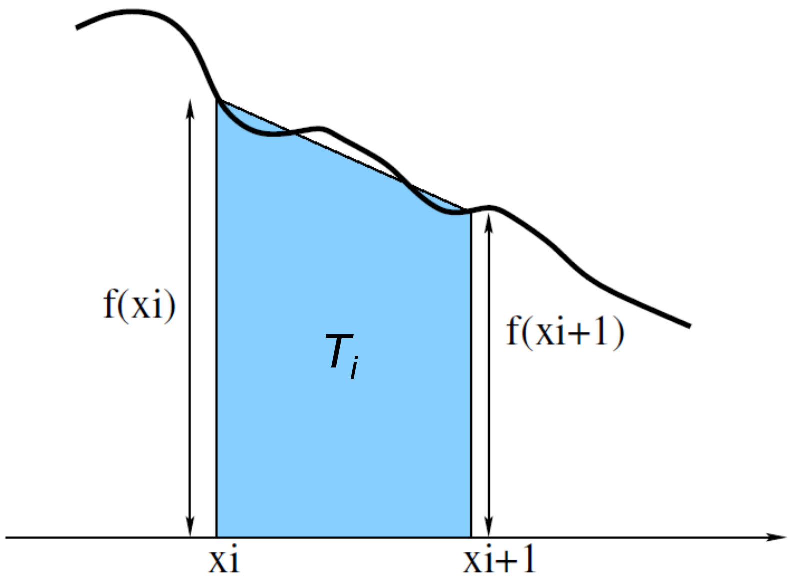
- Using again partitions and sums in a slightly more refined way

- Using again partitions and sums in a slightly more refined way



$$\int_{x_i}^{x_{i+1}} f(x)\,dx \approx T_i = \frac{1}{2}[f(x_i) + f(x_{i+1})](x_{i+1} - x_i)$$

- Using again partitions and sums in a slightly more refined way

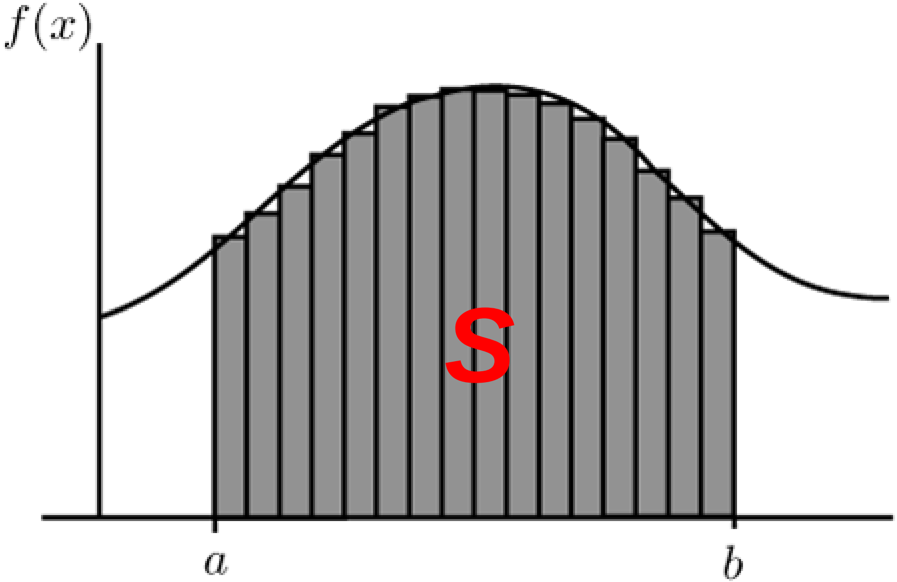


$$\int_{x_i}^{x_{i+1}} f(x)\,dx \approx T_i = \frac{1}{2}[f(x_i) + f(x_{i+1})](x_{i+1} - x_i)$$

- Then the total area below the function A is:

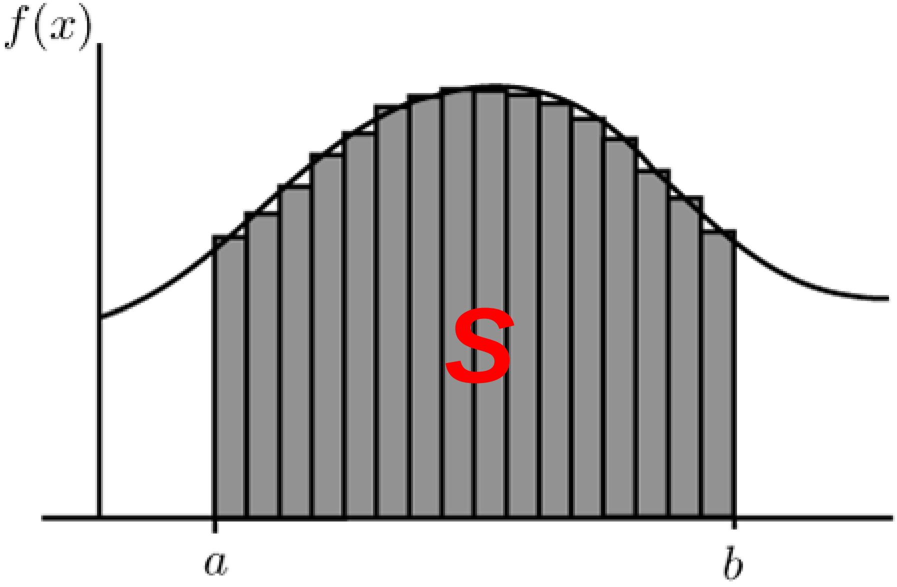
$$\int_a^b f(x)\,dx \approx A(f;P) = \sum_{i=0}^{n-1} T_i = \frac{1}{2}\sum_{i=0}^{n-1}[f(x_i) + f(x_{i+1})](x_{i+1} - x_i)$$

- Let's start again from a simplified 1D numerical integration $\int_a^b f(x)\,dx$



Monte Carlo Integration
Sampling mean method

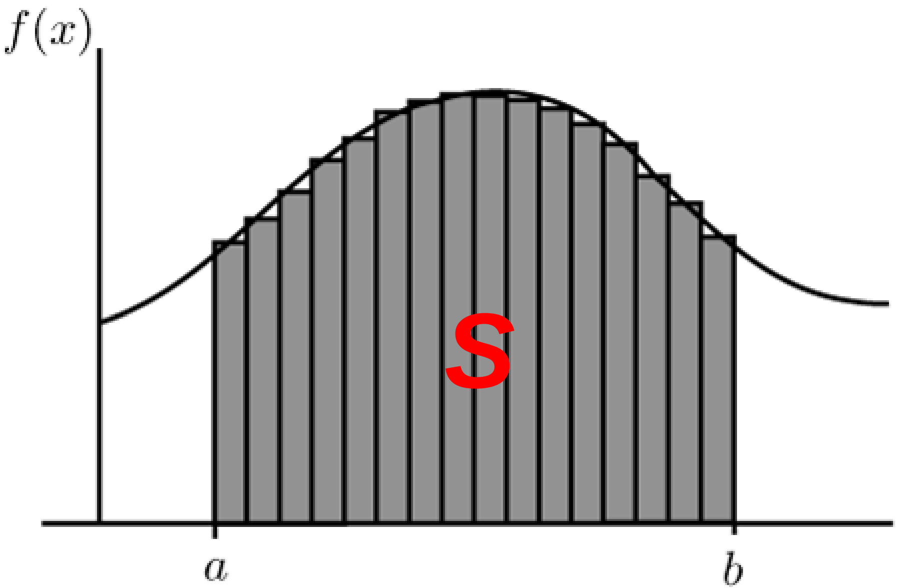
- Let's start again from a simplified 1D numerical integration $\int_a^b f(x)\,dx$



- We partition [a,b] to N equidistant subintervals

Monte Carlo Integration
Sampling mean method

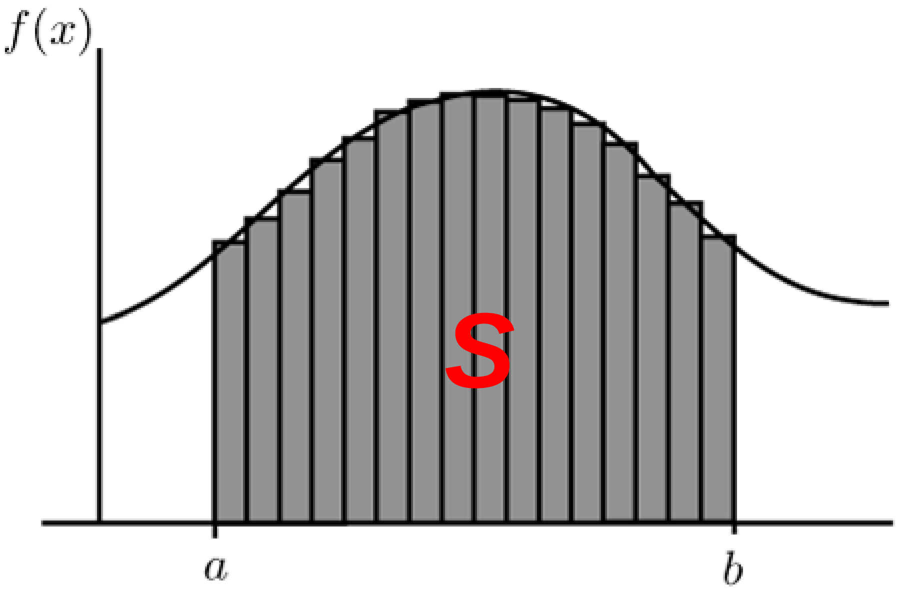
- Let's start again from a simplified 1D numerical integration $\int_a^b f(x)\,dx$



- We partition [a,b] to N equidistant subintervals

- Then $\Delta x = \dfrac{b-a}{N}$

- Let's start again from a simplified 1D numerical integration $\int_a^b f(x)\,dx$



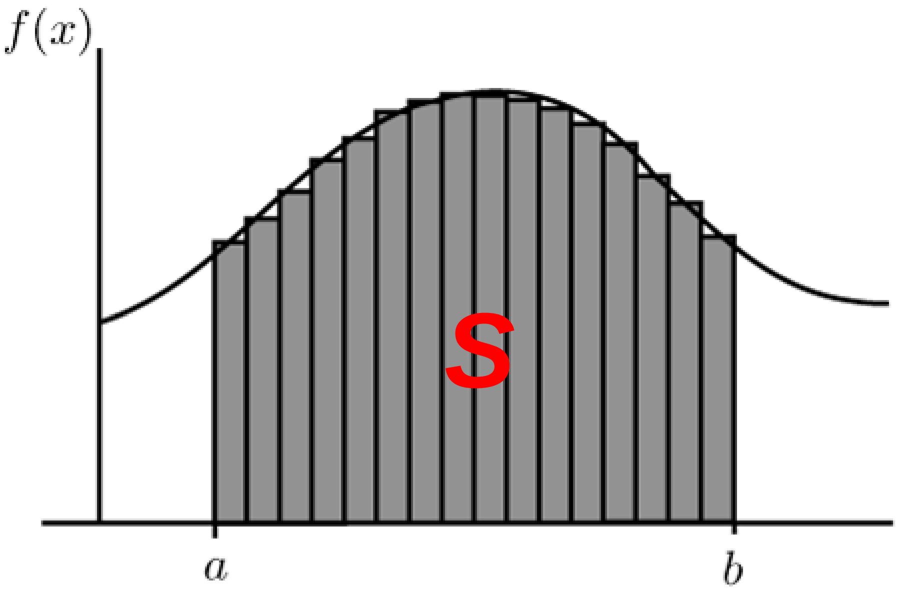
- We partition [a,b] to N equidistant subintervals

- Then $\Delta x = \dfrac{b-a}{N}$

- In that case $S = \dfrac{b-a}{N}\sum_{i=1}^{N} f(x_i)$

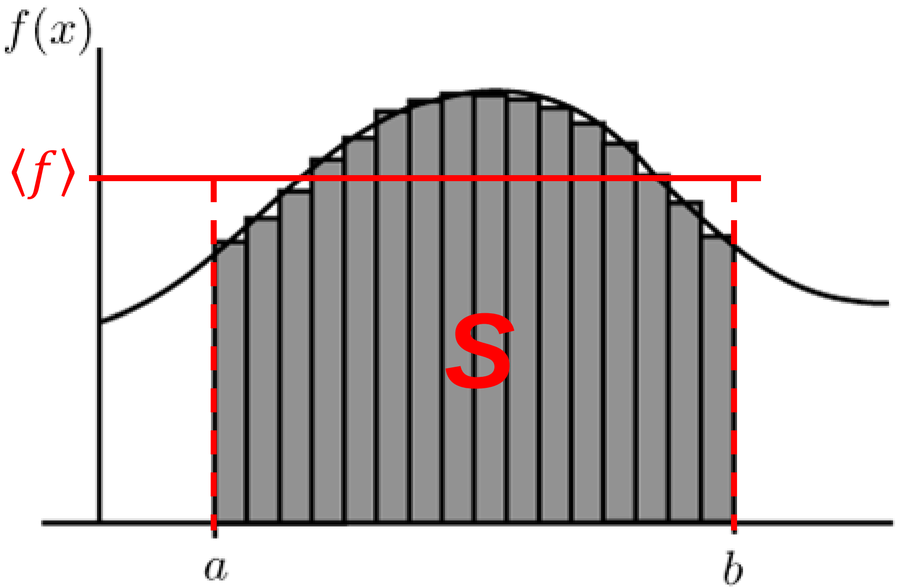
Monte Carlo Integration
Sampling mean method

- Let's start again from a simplified 1D numerical integration $\int_a^b f(x)\,dx$



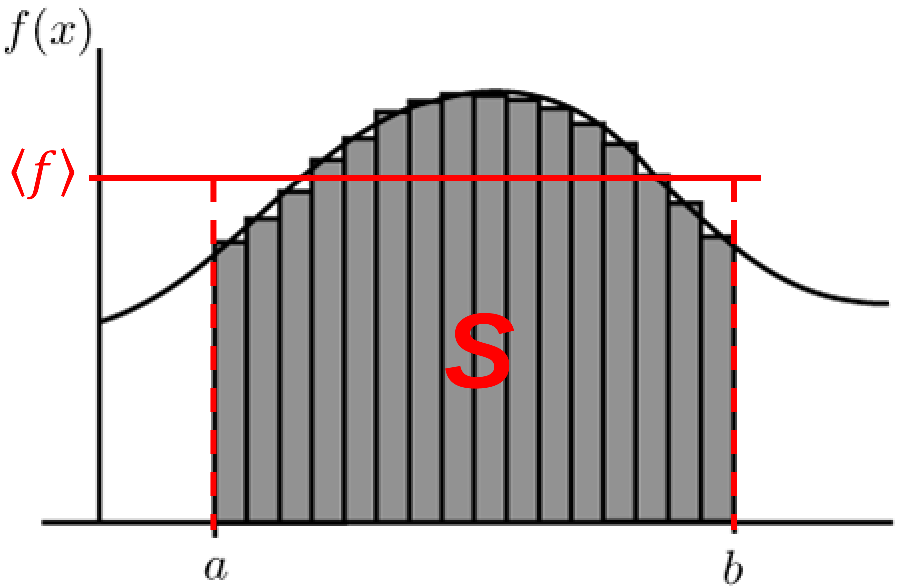
- We partition [a,b] to N equidistant subintervals
- Then $\Delta x = \dfrac{b-a}{N}$
- In that case $S = \dfrac{b-a}{N}\sum_{i=1}^{N} f(x_i)$
- Which can be re-interpreted as $S = (b-a)\dfrac{\sum_{i=1}^{N} f(x_i)}{N} = (b-a)\langle f\rangle$

- Let's start again from a simplified 1D numerical integration $\int_a^b f(x)\,dx$



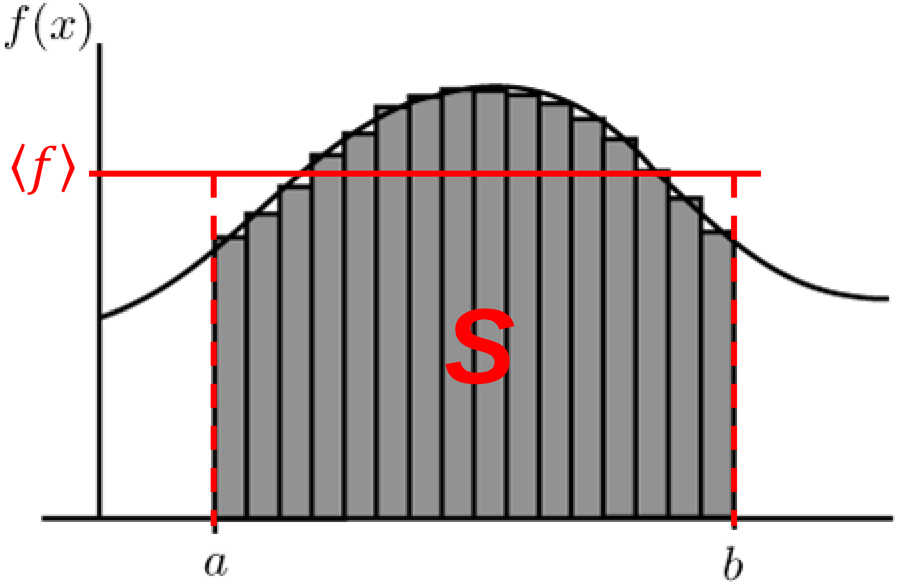
- We partition [a,b] to N equidistant subintervals
- Then $\Delta x = \dfrac{b-a}{N}$
- In that case $S = \dfrac{b-a}{N}\sum_{i=1}^{N} f(x_i)$
- Which can be re-interpreted as $S = (b-a)\dfrac{\sum_{i=1}^{N} f(x_i)}{N} = (b-a)\langle f \rangle$

- Let's start again from a simplified 1D numerical integration $\int_a^b f(x)\,dx$



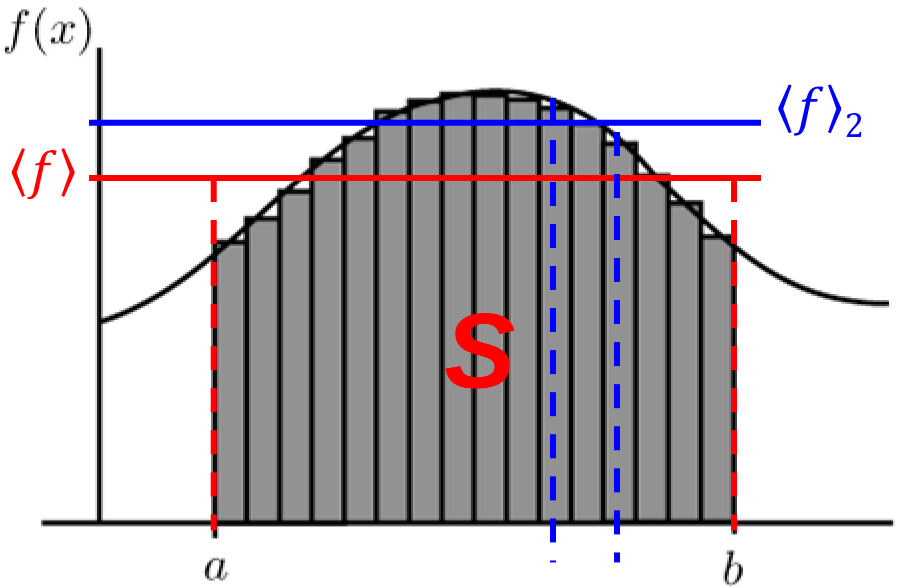
- We partition [a,b] to N equidistant subintervals
- Then $\Delta x = \dfrac{b-a}{N}$
- In that case $S = \dfrac{b-a}{N}\sum_{i=1}^{N} f(x_i)$
- Which can be re-interpreted as $S = (b-a)\dfrac{\sum_{i=1}^{N} f(x_i)}{N} = (b-a)\langle f \rangle$
- Extendable to M-dimensions $S^{M+1} = S^M \langle f \rangle$

- Let's start again from a simplified 1D numerical integration $\int_a^b f(x)\,dx$



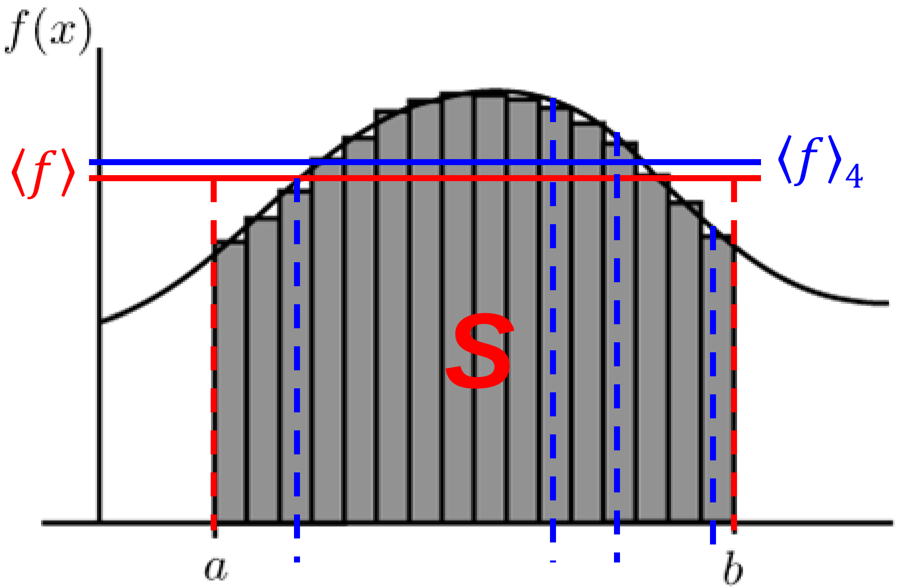
- We partition [a,b] to N equidistant subintervals
- Then $\Delta x = \frac{b-a}{N}$
- In that case $S = \frac{b-a}{N} \sum_{i=1}^{N} f(x_i)$
- Which can be re-interpreted as $S = (b-a) \frac{\sum_{i=1}^{N} f(x_i)}{N} = (b-a)\langle f \rangle$
- Extendable to M-dimensions $S^{M+1} = S^{M} \langle f \rangle$
- With MC integration, we do exactly the same thing, but now sampling by using a uniform random number distribution for each dimension

- Let's start again from a simplified 1D numerical integration $\int_a^b f(x)\,dx$



- We partition [a,b] to N equidistant subintervals
- Then $\Delta x = \dfrac{b-a}{N}$
- In that case $S = \dfrac{b-a}{N}\sum_{i=1}^{N} f(x_i)$
- Which can be re-interpreted as $S = (b-a)\dfrac{\sum_{i=1}^{N} f(x_i)}{N} = (b-a)\langle f\rangle$
- Extendable to M-dimensions $S^{M+1} = S^{M}\langle f\rangle$
- With MC integration, we do exactly the same thing, but now sampling by using a uniform random number distribution for each dimension

- Let's start again from a simplified 1D numerical integration $\int_a^b f(x)\,dx$

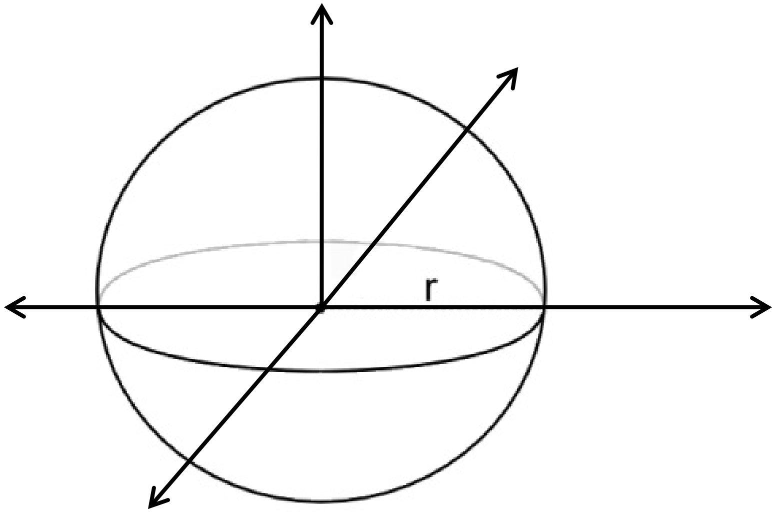


- We partition [a,b] to N equidistant subintervals
- Then $\Delta x = \dfrac{b-a}{N}$
- In that case $S = \dfrac{b-a}{N}\sum_{i=1}^{N} f(x_i)$
- Which can be re-interpreted as $S = (b-a)\dfrac{\sum_{i=1}^{N} f(x_i)}{N} = (b-a)\langle f \rangle$
- Extendable to M-dimensions $S^{M+1} = S^M \langle f \rangle$
- With MC integration, we do exactly the same thing, but now sampling by using a uniform random number distribution for each dimension

- Example estimation of a volume of a sphere (3D space):

$$r^2 = x^2 + y^2 + z^2 \Rightarrow z = \sqrt{r^2 - (x^2 + y^2)}$$

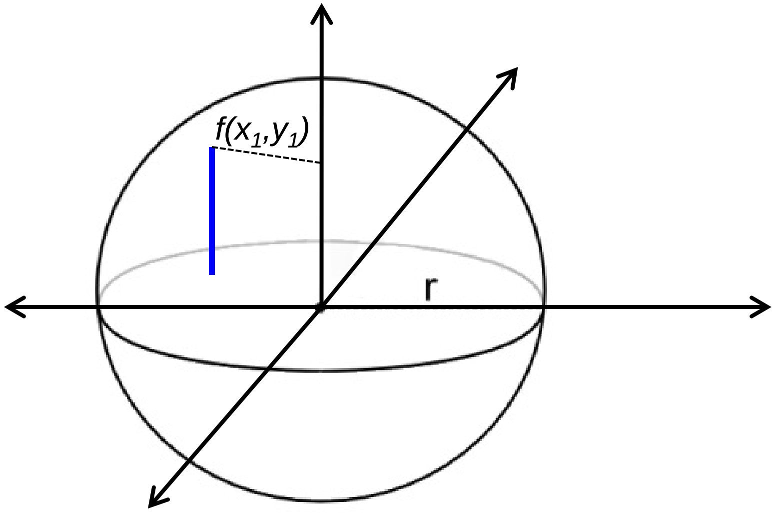
- So $f(x,y) = \sqrt{r^2 - (x^2 + y^2)}$ and we would like to estimate $\int_{-1}^{1} \int_{-1}^{1} f(x,y)\, dx\, dy$



- Example estimation of a volume of a sphere (3D space):

$$r^2 = x^2 + y^2 + z^2 \Rightarrow z = \sqrt{r^2 - (x^2 + y^2)}$$

- So $f(x,y) = \sqrt{r^2 - (x^2 + y^2)}$ and we would like to estimate $\int_{-1}^{1}\int_{-1}^{1} f(x,y)\,dx\,dy$



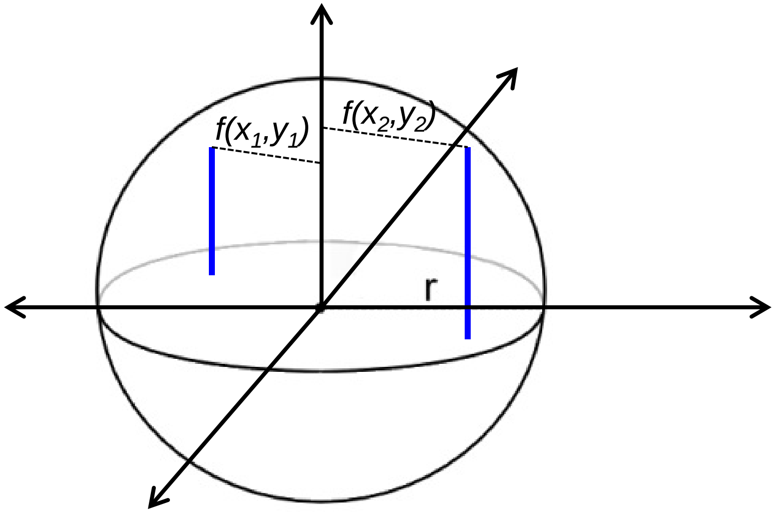
- We sample randomly in two dimensions (x,y)

Monte Carlo Integration
Sampling mean method

- Example estimation of a volume of a sphere (3D space):

$$r^2 = x^2 + y^2 + z^2 \Rightarrow z = \sqrt{r^2 - (x^2 + y^2)}$$

- So $f(x,y) = \sqrt{r^2 - (x^2 + y^2)}$ and we would like to estimate $\int_{-1}^{1}\int_{-1}^{1} f(x,y)\,dx\,dy$



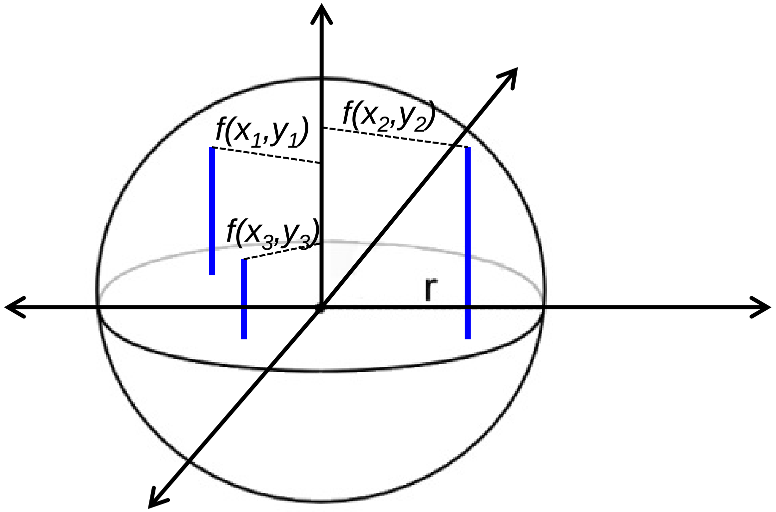
- We sample randomly in two dimensions (x,y)

Monte Carlo Integration
Sampling mean method

- Example estimation of a volume of a sphere (3D space):

$$r^2 = x^2 + y^2 + z^2 \Rightarrow z = \sqrt{r^2 - (x^2 + y^2)}$$

- So $f(x, y) = \sqrt{r^2 - (x^2 + y^2)}$ and we would like to estimate $\int_{-1}^{1}\int_{-1}^{1} f(x, y)\, dx\, dy$



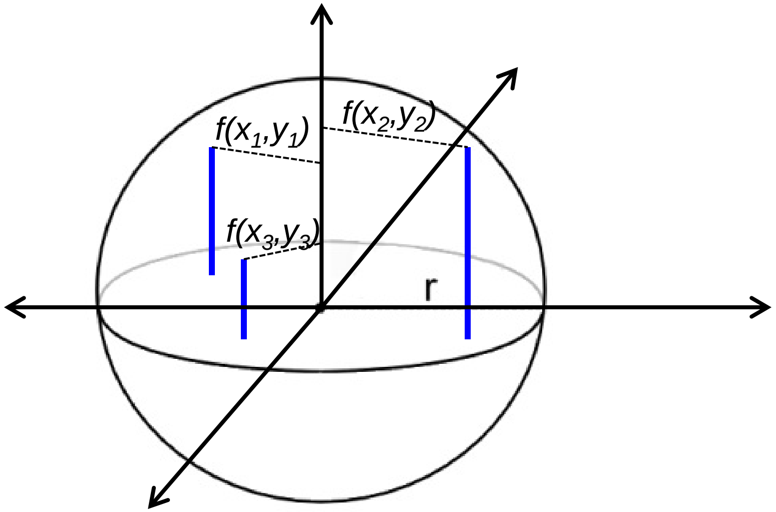
- We sample randomly in two dimensions (x,y)

Monte Carlo Integration
Sampling mean method

- Example estimation of a volume of a sphere (3D space):

$$r^2 = x^2 + y^2 + z^2 \Rightarrow z = \sqrt{r^2 - (x^2 + y^2)}$$

- So $f(x,y) = \sqrt{r^2 - (x^2 + y^2)}$ and we would like to estimate $\int_{-1}^{1}\int_{-1}^{1} f(x,y)\,dx\,dy$

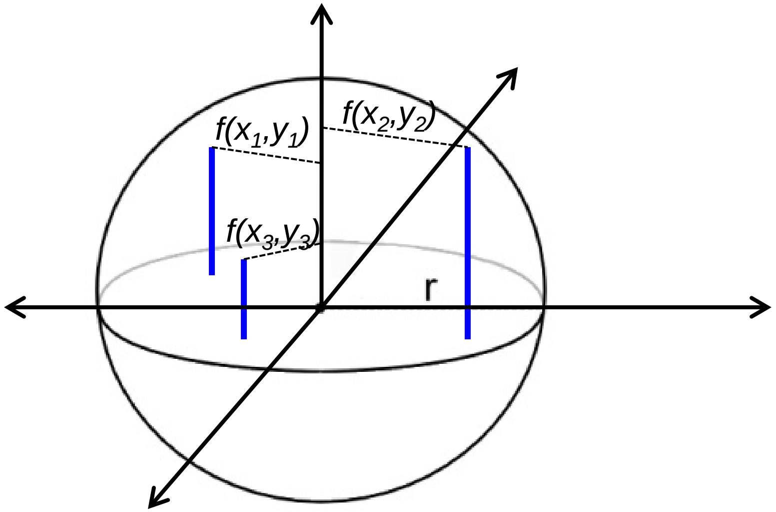


- We sample randomly in two dimensions (x,y)
- We estimate $\langle f \rangle$ from the sampling
- The three-dimensional space (volume of the sphere) will be $S^{M+1} = S^M \langle f \rangle$

- Example estimation of a volume of a sphere (3D space):

$$r^2 = x^2 + y^2 + z^2 \Rightarrow z = \sqrt{r^2 - (x^2 + y^2)}$$

- So $f(x,y) = \sqrt{r^2 - (x^2 + y^2)}$ and we would like to estimate $\int_{-1}^{1}\int_{-1}^{1} f(x,y)\,dx\,dy$



- We sample randomly in two dimensions (x,y)
- We estimate $\langle f \rangle$ from the sampling
- The three-dimensional space (volume of the sphere) will be $S^{M+1} = 2\pi r^2 \langle f \rangle$

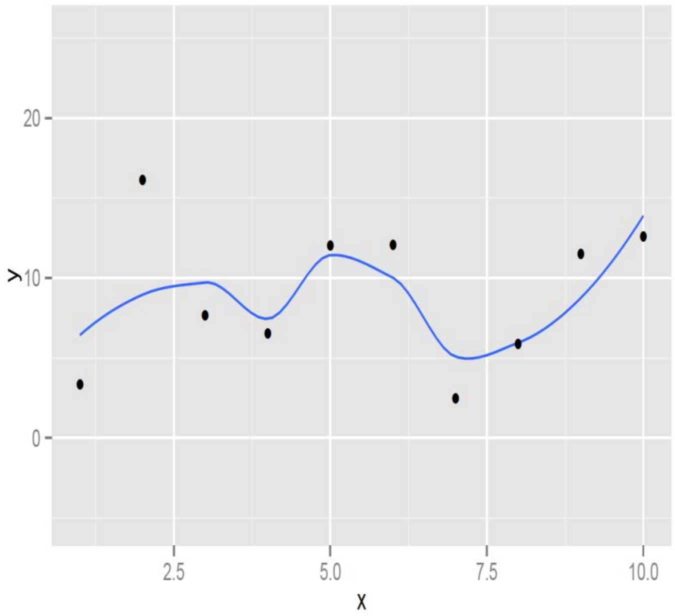
- Summarizing:
  - We want to calculate the integral over an M dimensional space of a function

$$\underbrace{f(x, y, z, \dots, m)}_{\text{M dimensions}}$$

  - The integral will be of M+1 dimension $S^{M+1} = S^{M}\langle f\rangle$
  - The problem is reduced to finding an accurate $\langle f\rangle$ by random sampling in M space
  - Finally we have to multiply the M dimensional sampling space with $\langle f\rangle$

Monte Carlo Integration
Hit or miss method

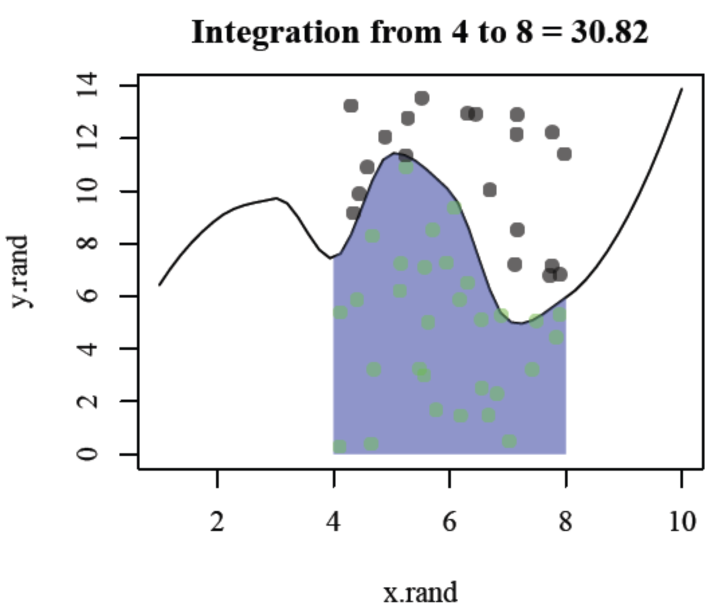
- Another approach of MC integration is the Hit-or-miss method
- Simpler logic than the sampling method
- We have a (complex/not well defined) function we need to integrate



Monte Carlo Integration
Hit or miss method

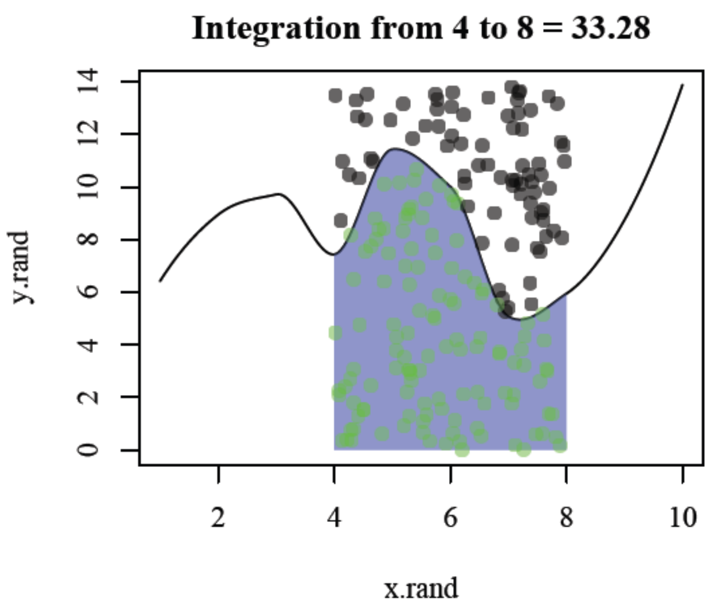
- Another approach of MC integration is the Hit-or-miss method
- Simpler logic than the sampling method
- In the webpage that I’ve found this nice example, it is called as the “Poor man’s” integration for those who have forgotten the “Dark Arts”
(tricks to help you derive analytically a complex integral)

- Another approach of MC integration is the Hit-or-miss method
- Simpler logic than the sampling method
- We have a (complex/not well defined) function we need to integrate



- We define a well known/easy to calculate volume/surface I_{known} that contains the complex volume/surface I_{est} we want to estimate

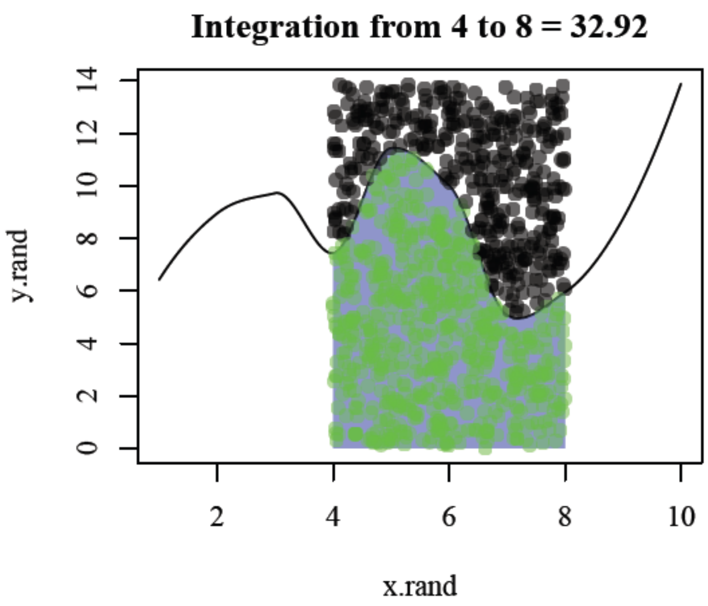
- Another approach of MC integration is the Hit-or-miss method
- Simpler logic than the sampling method
- We have a (complex/not well defined) function we need to integrate



- We define a well known/easy to calculate volume/surface I_{known} that contains the complex volume/surface I_{est} we want to estimate
- We create random sampling of our known volume

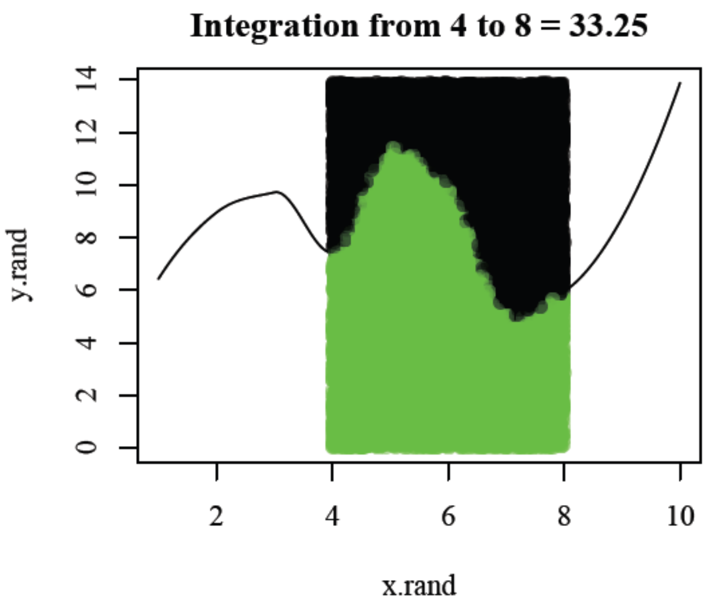
Monte Carlo Integration
Hit or miss method

- Another approach of MC integration is the Hit-or-miss method
- Simpler logic than the sampling method
- We have a (complex/not well defined) function we need to integrate



- We define a well known/easy to calculate volume/surface I_{known} that contains the complex volume/surface I_{est} we want to estimate
- We create random sampling of our known volume (many)

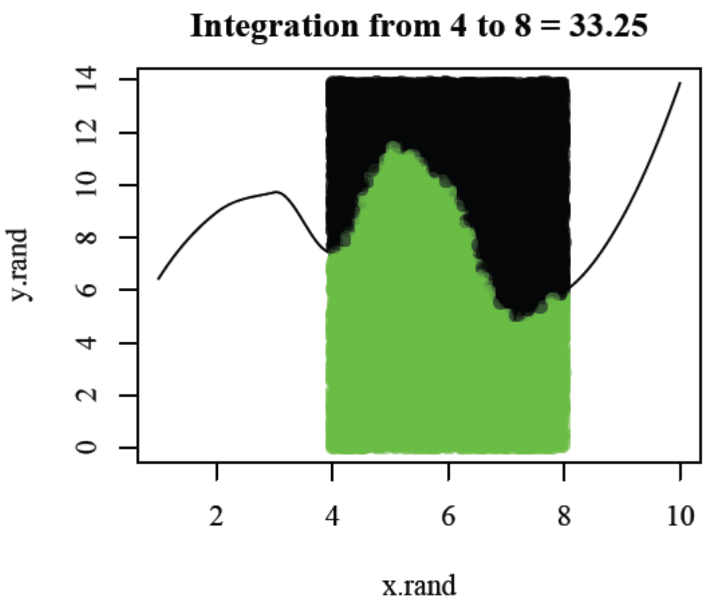
- Another approach of MC integration is the Hit-or-miss method
- Simpler logic than the sampling method
- We have a (complex/not well defined) function we need to integrate



- We define a well known/easy to calculate volume/surface I_{known} that contains the complex volume/surface I_{est} we want to estimate
- We create random sampling of our known volume (as many as possible)

Monte Carlo Integration
Hit or miss method

- Another approach of MC integration is the Hit-or-miss method
- Simpler logic than the sampling method
- We have a (complex/not well defined) function we need to integrate

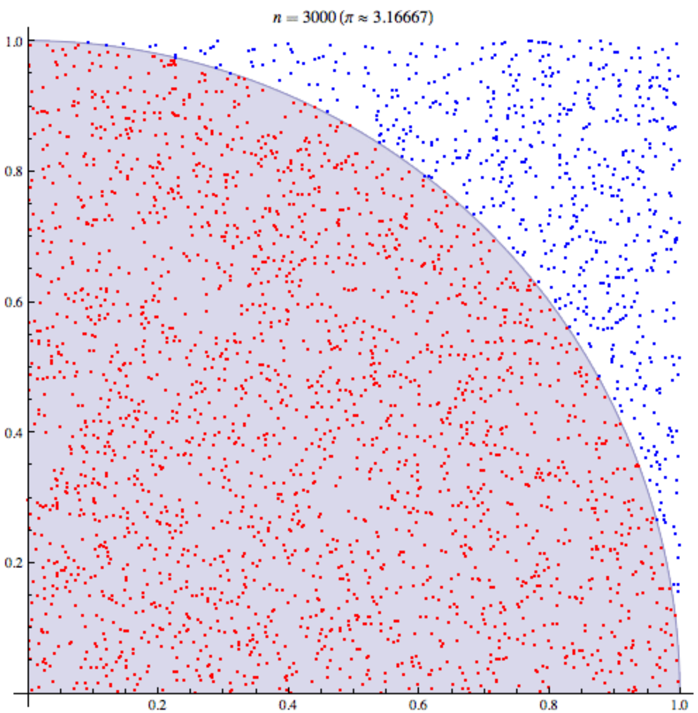


- We define a well known/easy to calculate volume/surface I_{known} that contains the complex volume/surface I_{est} we want to estimate
- We create random sampling of our known volume (as many as possible)
- Then $$\frac{N_{est}}{N_{known}} = \frac{I_{est}}{I_{known}}$$

- The famous π calculation
- Is simply the counting of the number of points in a circle and in a square containing this circle

Monte Carlo Integration
Hit or miss method

- The famous π calculation
- Is simply the counting of the number of points in a circle and in a square containing this circle



$$\frac{N_{circle}}{N_{sq}} = \frac{I_{circle}}{I_{sq}}$$

$$\frac{N_{circle}}{N_{sq}} = \frac{\pi r^2}{4r^2}$$

$$\pi = 4\frac{N_{circle}}{N_{sq}}$$

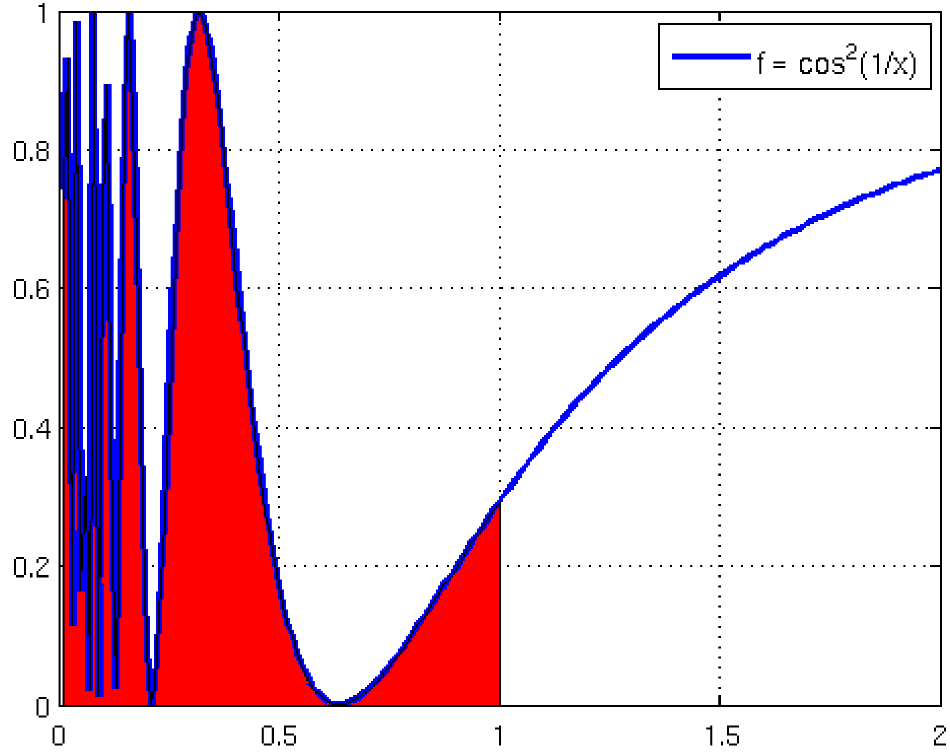
Monte Carlo Integration vs Numerical Integration

- It can be shown that the error of the traditional numerical methods scales as $N^{-1/M}$, where N is the number of intervals/points and M is the number of dimensions
- In the case of Monte Carlo Methods, the error for any number of dimensions M scales with $N^{-1/2}$

Dimensions M	Numerical Methods	Monte Carlo Technique
1	$1/N$	$1/\sqrt{N}$
2	$1/\sqrt{N}$	$1/\sqrt{N}$
>2	$1/\sqrt[M]{N}$	$1/\sqrt{N}$

- Monte Carlo methods become more efficient at higher dimensions

- Programming exercise 1:
  - Write two programs (in any programming language, preferable in C++, C, FORTRAN) that integrate $f(x) = \cos^2 \frac{1}{x}$ on [0,1] using the two MC integration techniques we learned:



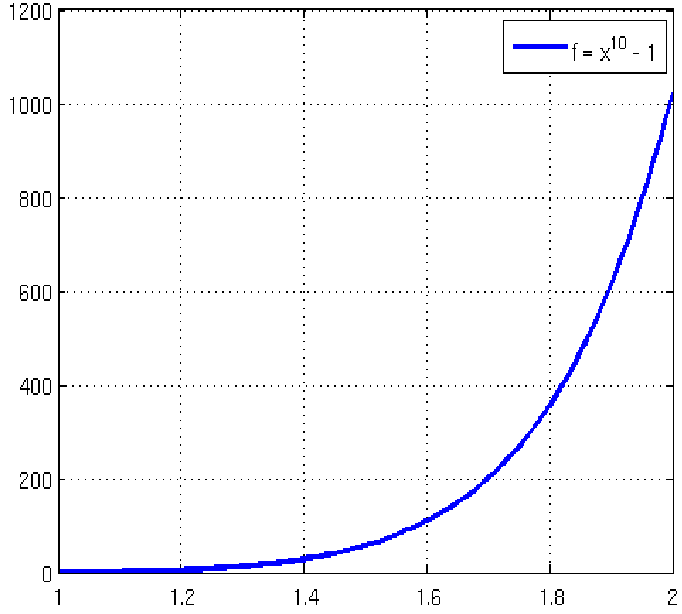
- Programming exercise 1:
  - Write two programs (in any programming language, preferable in C++, C, FORTRAN) that integrate $f(x) = \cos^2 \frac{1}{x}$ on [0,1] using the two MC integration techniques we learned:
    - Sampling (mean) MC method
    - Hit-or-miss method
  - For the two methods test and plot the convergence to the correct answer of 0.326543 (with 6 decimal digits)
  - For the two methods test and plot the variance (the spread of values obtained from a given set of samples)

- Although MC is a powerful technique for integral calculation, in some instances convergence can be slow
- One can use refined/intelligent methods in order to enhance MC performance

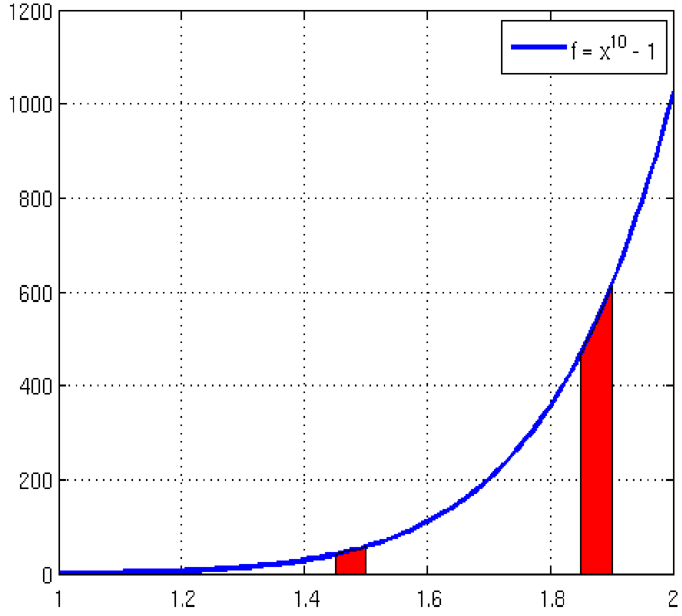
- Although MC is a powerful technique for integral calculation, in some instances convergence can be slow
- One can use refined/intelligent methods in order to enhance MC performance
- We will discuss two advanced MC integration methods:
  - Importance sampling
  - Control variates

- It is a refined version of the MC sampling method described before
- In the cases of rapid varying functions, some regions contribute to the integral much less than others

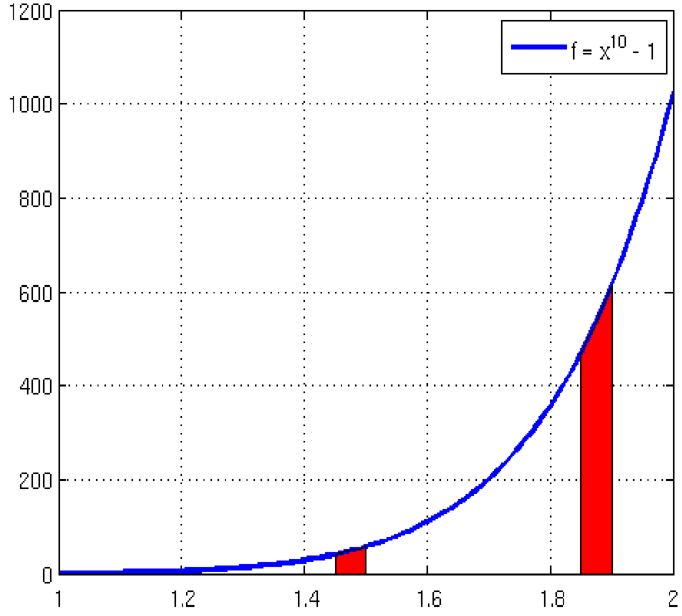
- It is a refined version of the MC sampling method described before
- In the cases of rapid varying functions, some regions contribute to the integral much less than others



- It is a refined version of the MC sampling method described before
- In the cases of rapid varying functions, some regions contribute to the integral much less than others

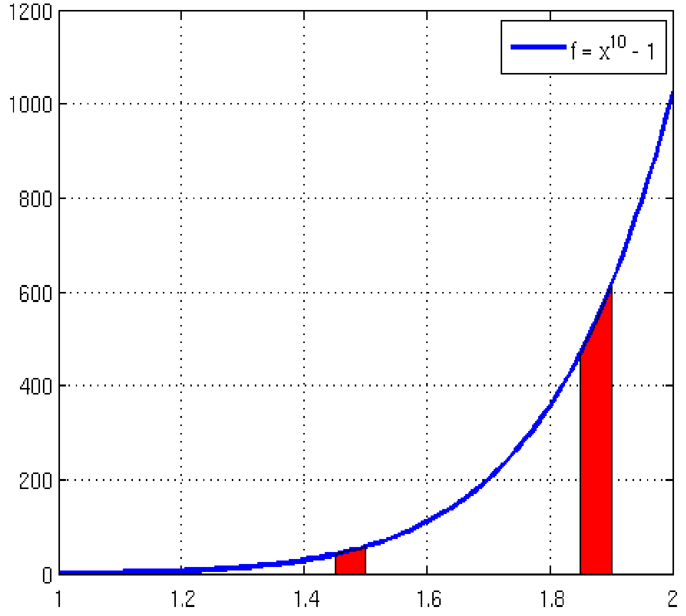


- It is a refined version of the MC sampling method described before
- In the cases of rapid varying functions, some regions contribute to the integral much less than others



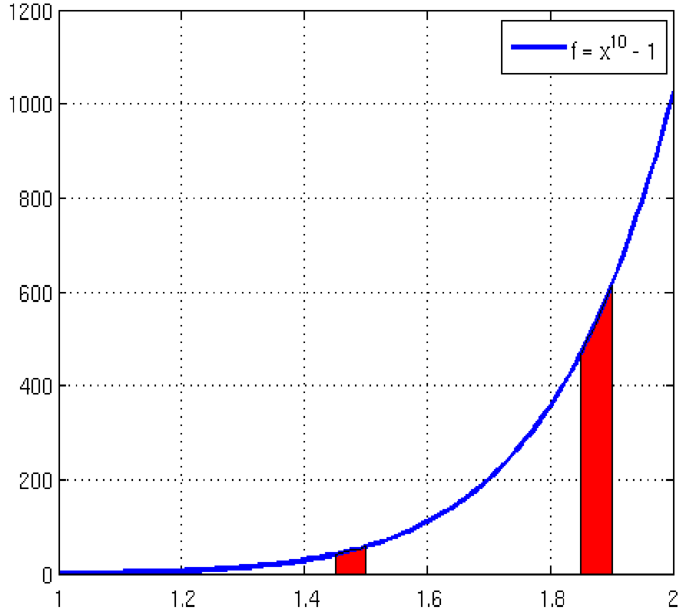
- We would like to “flatten” our function so as to avoid oversampling non-important regions

- It is a refined version of the MC sampling method described before
- In the cases of rapid varying functions, some regions contribute to the integral much less than others



- We would like to “flatten” our function so as to avoid oversampling non-important regions
- For a very steep f(x), we can find a function g(x) that behaves similarly in the integration region [a,b], so that f(x)/g(x) is fairly flat

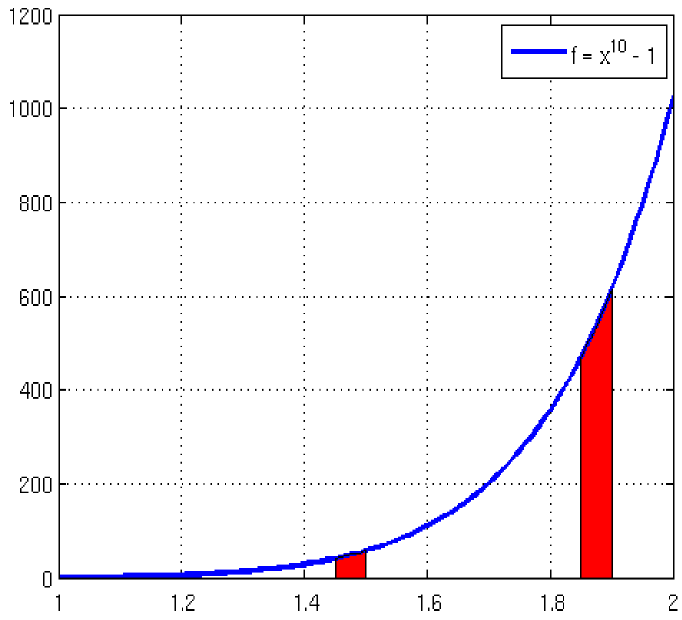
- It is a refined version of the MC sampling method described before
- In the cases of rapid varying functions, some regions contribute to the integral much less than others



- We would like to “flatten” our function so as to avoid oversampling non-important regions
- For a very steep f(x), we can find a function g(x) that behaves similarly in the integration region [a,b], so that f(x)/g(x) is fairly flat
- Then $I = \int_a^b f(x)\,dx$ becomes:

$$I = \int_a^b f(x)\,dx = \int_a^b \frac{f(x)}{g(x)} g(x)\,dx = \int_{G(a)}^{G(b)} \frac{f(x)}{g(x)} dG(x)$$

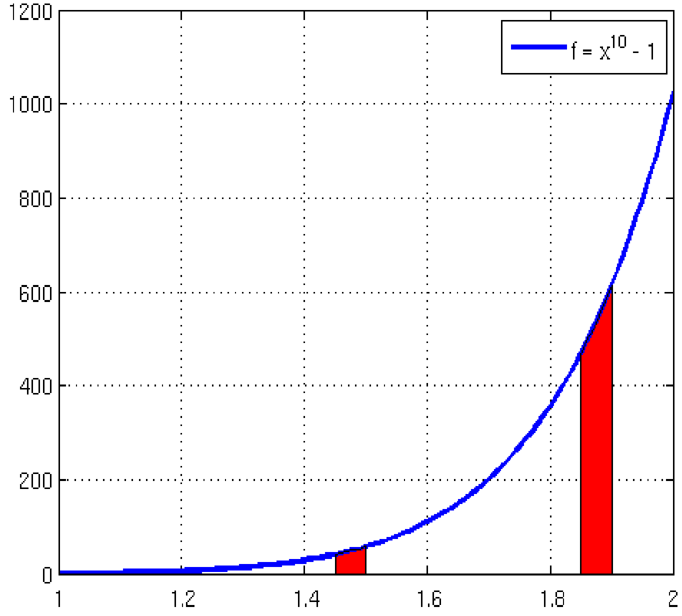
- It is a refined version of the MC sampling method described before
- In the cases of rapid varying functions, some regions contribute to the integral much less than others



- We would like to “flatten” our function so as to avoid oversampling non-important regions
- For a very steep f(x), we can find a function g(x) that behaves similarly in the integration region [a,b], so that f(x)/g(x) is fairly flat
- With $$G(x) = \int_a^x g(x)\,dx$$

- It is a refined version of the MC sampling method described before

- In the cases of rapid varying functions, some regions contribute to the integral much less than others



- We would like to “flatten” our function so as to avoid oversampling non-important regions

- For a very steep f(x), we can find a function g(x) that behaves similarly in the integration region [a,b], so that f(x)/g(x) is fairly flat

- With $$G(x) = \int_a^x g(x)\,dx$$

- And the weighting function g(x) should be normalized in [a,b]: $$\int_a^b g(x)\,dx = 1$$

- Taking $$I = \int_a^b f(x)\,dx = \int_a^b \frac{f(x)}{g(x)} g(x)\,dx = \int_{G(a)}^{G(b)} \frac{f(x)}{g(x)} dG(x)$$

Monte Carlo Integration
Importance sampling

- Taking $I = \int_a^b f(x)\,dx = \int_a^b \frac{f(x)}{g(x)} g(x)\,dx = \int_{G(a)}^{G(b)} \frac{f(x)}{g(x)} dG(x)$
- And doing a variable change $y = G(x)$ so that $x = G^{-1}(y)$

$$I = \int_a^b f(x)\,dx = \int_{G(a)}^{G(b)} \frac{f(G^{-1}(y)\,)}{g(G^{-1}(y)\,)} dy$$

- Taking $I = \int_a^b f(x)\,dx = \int_a^b \frac{f(x)}{g(x)} g(x)\,dx = \int_{G(a)}^{G(b)} \frac{f(x)}{g(x)} dG(x)$

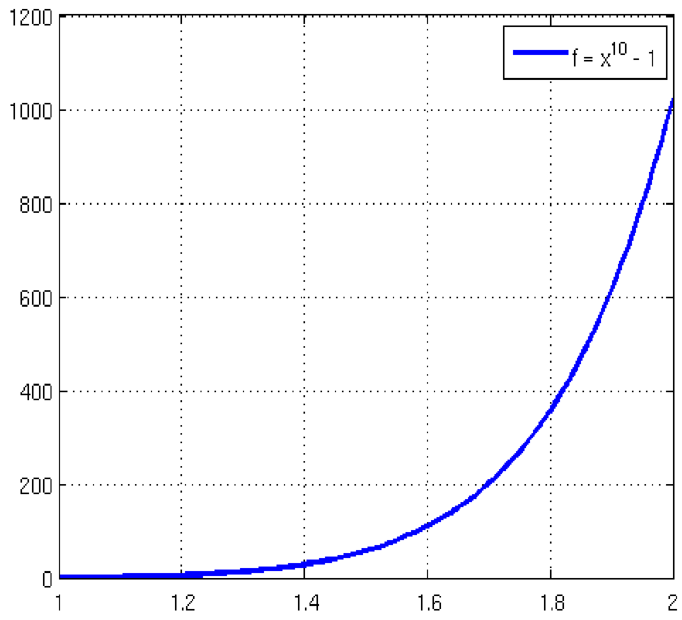
- And doing a variable change $y = G(x)$ so that $x = G^{-1}(y)$

$$I = \int_a^b f(x)\,dx = \int_{G(a)}^{G(b)} \frac{f(G^{-1}(y)\,)}{g(G^{-1}(y)\,)} dy$$

- The integral can be estimated as in the case of the simple sampling method

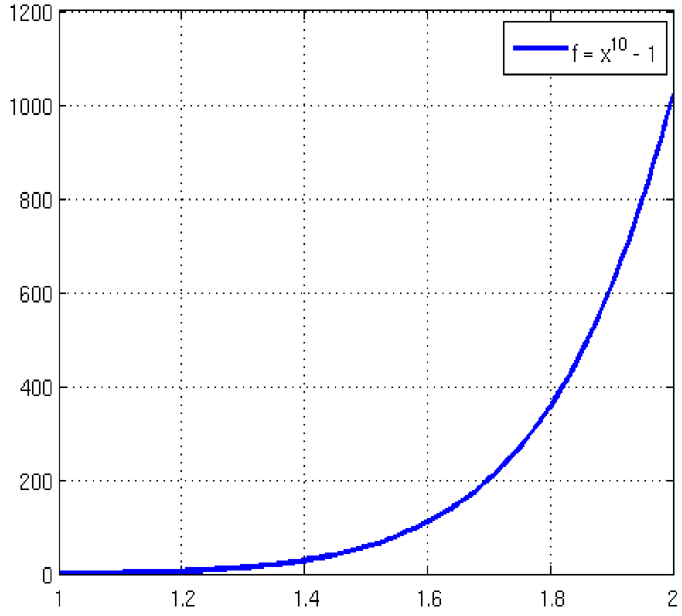
$$I = \frac{G(b) - G(a)}{N} \sum_{i=1}^{N} \frac{f(G^{-1}(y_i)\,)}{g(G^{-1}(y_i)\,)}$$

- Going back to the example of $f(x)=x^{10}-1$ in [1,2]



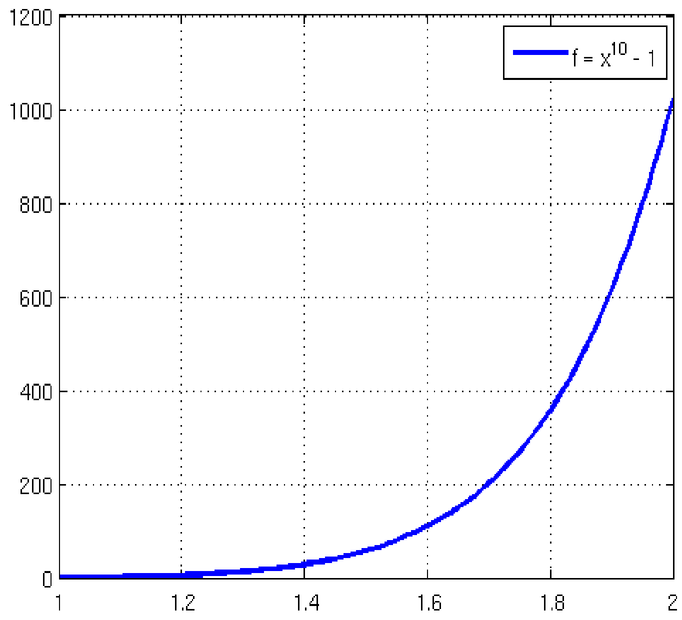
- Going back to the example of $f(x)=x^{10}-1$ in [1,2]

- We select a weighting function $g(x)=Cx^{10}$



- Going back to the example of $f(x)=x^{10}-1$ in [1,2]

- We select a weighting function $g(x)=Cx^{10}$



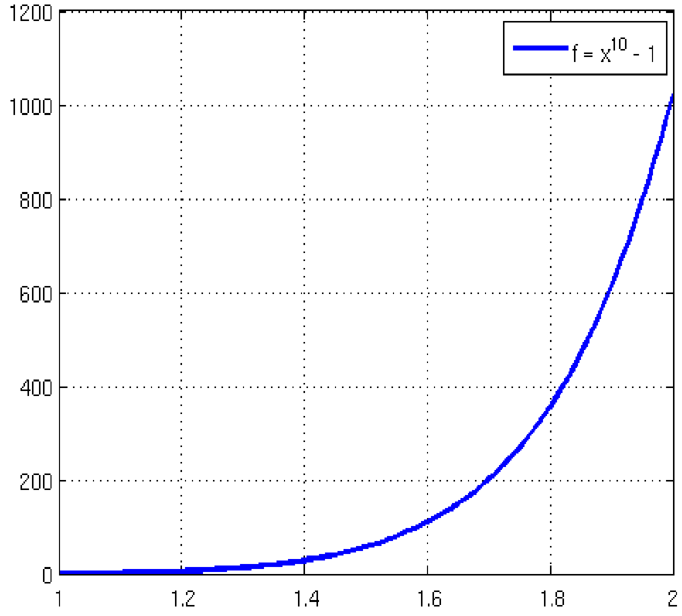
- The g(x) should be normalized in [1,2]

$$\int_1^2 g(x)\,dx = \int_1^2 Cx^{10}\,dx = 1$$

$$C = \frac{11}{2^{11}-1}$$

- Going back to the example of $f(x)=x^{10}-1$ in [1,2]

- We select a weighting function $g(x)=Cx^{10}$



- The g(x) should be normalized in [1,2]

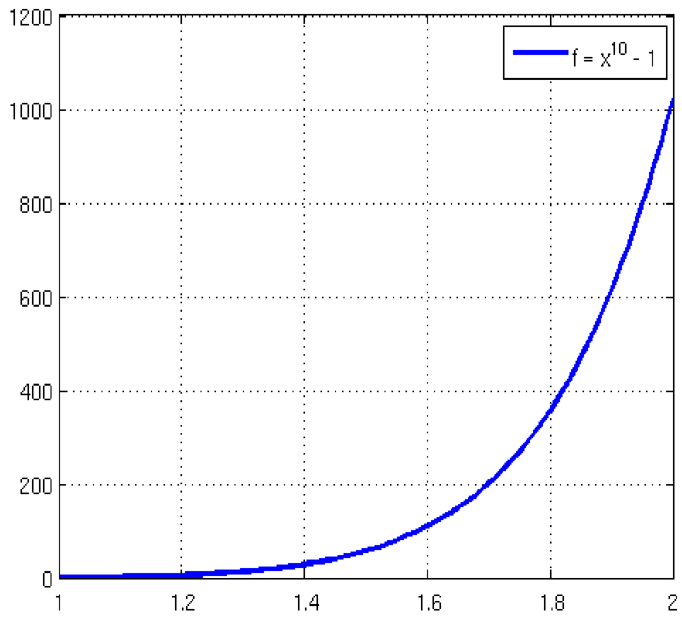
$$\int_1^2 g(x)\,dx = \int_1^2 Cx^{10}\,dx = 1$$

$$C = \frac{11}{2^{11}-1}$$

- Then $G(x) = \int_1^x g(x)\,dx = \frac{11}{2^{11}-1}\int_1^x x^{10}\,dx = y$

- Going back to the example of $f(x)=x^{10}-1$ in [1,2]

- We select a weighting function $g(x)=Cx^{10}$



- The g(x) should be normalized in [1,2]

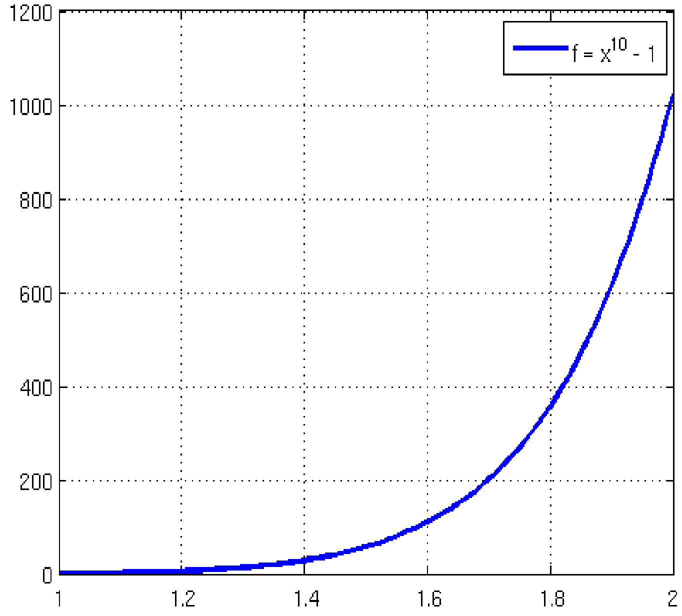
$$\int_1^2 g(x)\,dx = \int_1^2 Cx^{10}\,dx = 1$$

$$C = \frac{11}{2^{11}-1}$$

- Then $G(x) = \int_1^x g(x)\,dx = \frac{11}{2^{11}-1}\int_1^x x^{10}\,dx = y$

- So $G(x) = y = \frac{x^{11}-1}{2^{11}-1}$ and $x = G^{-1}(y) = [(2^{11}-1)y+1]^{1/11}$

- So the estimation of $I = \int_1^2 f(x)\, dx = \int_1^2 \frac{f(x)}{g(x)} g(x)\, dx = \int_{G(1)}^{G(2)} \frac{f(x)}{g(x)} dG(x)$



- Becomes: $\int_{G(1)}^{G(2)} \frac{f(G^{-1}(y)\ )}{g(G^{-1}(y)\ )} dy$

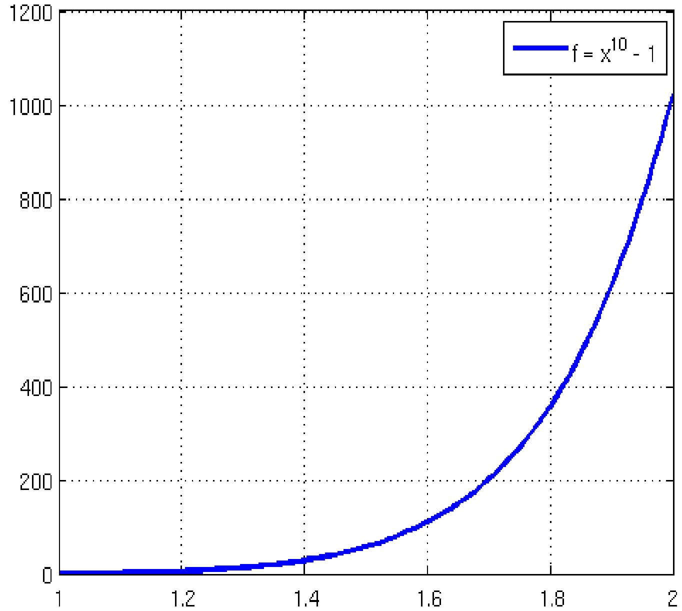
- With $G(1) = 0$ and $G(2) = 1$

- So that

$$I = \frac{1}{N} \sum_{i=1}^{N} \frac{f(G^{-1}(y_i)\ )}{g(G^{-1}(y_i)\ )}$$

- So the estimation of $I = \int_1^2 f(x)\,dx = \int_1^2 \frac{f(x)}{g(x)} g(x)\,dx = \int_{G(1)}^{G(2)} \frac{f(x)}{g(x)} dG(x)$

- Becomes: $\int_{G(1)}^{G(2)} \frac{f(G^{-1}(y)\,)}{g(G^{-1}(y)\,)} dy$



- With $G(1) = 0$ and $G(2) = 1$

- So that

$$I = \frac{1}{N} \sum_{i=1}^{N} \frac{f(G^{-1}(y_i)\,)}{g(G^{-1}(y_i)\,)}$$

- How is this of any help for my calculation?!?!?

- Forget for the moment the whole discussion about “flattening”

- Forget for the moment the whole discussion about “flattening”
- A re-interpretation of what happened is that we started from sampling the function f over the domain of uniformly distributed x

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

- Forget for the moment the whole discussion about “flattening”
- A re-interpretation of what happened is that we started from sampling the function f over the domain of uniformly distributed x

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

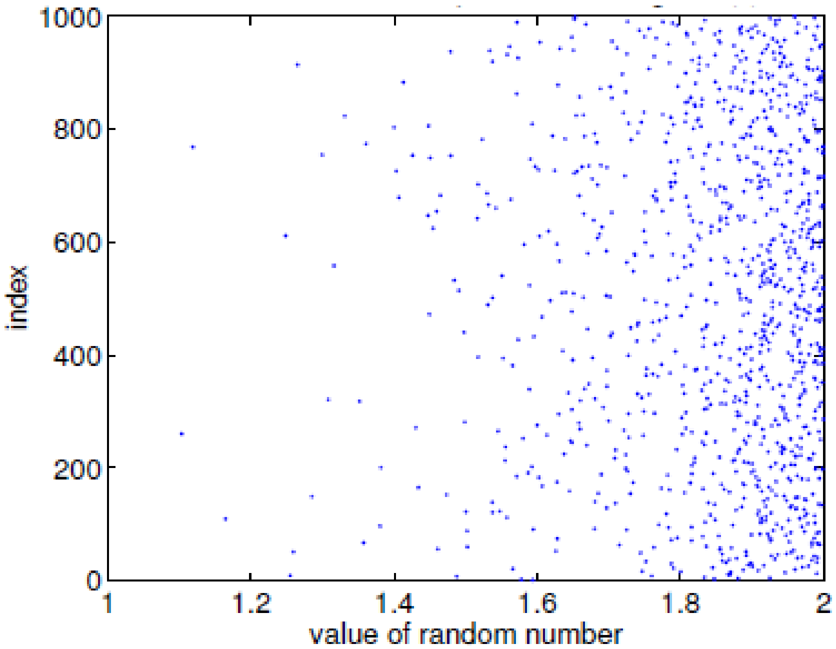
- We reached at the point of sampling f/g again in the domain x but now starting from a uniformly distributed y

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

- With $x = G^{-1}(y)$

- So what we really did is to change from uniform sampling x to non-uniform sampling x because $x = G^{-1}(y)$

- So what we really did is to change from uniform sampling x to non-uniform sampling x because $x = G^{-1}(y)$

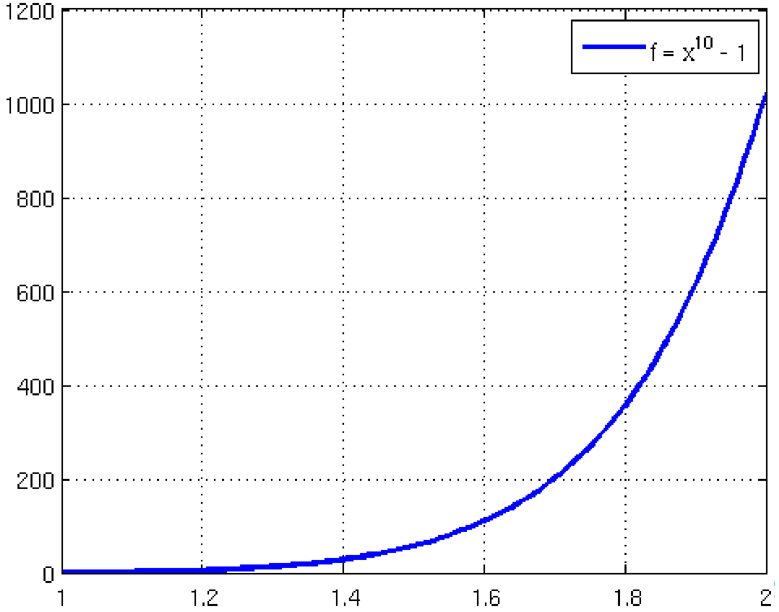


- In our example:

$$x = G^{-1}(y) = [(2^{11}-1)y + 1]^{1/11}$$

- With y uniform in [0,1]

- So what we really did is to change from uniform sampling x to non-uniform sampling x because $x = G^{-1}(y)$

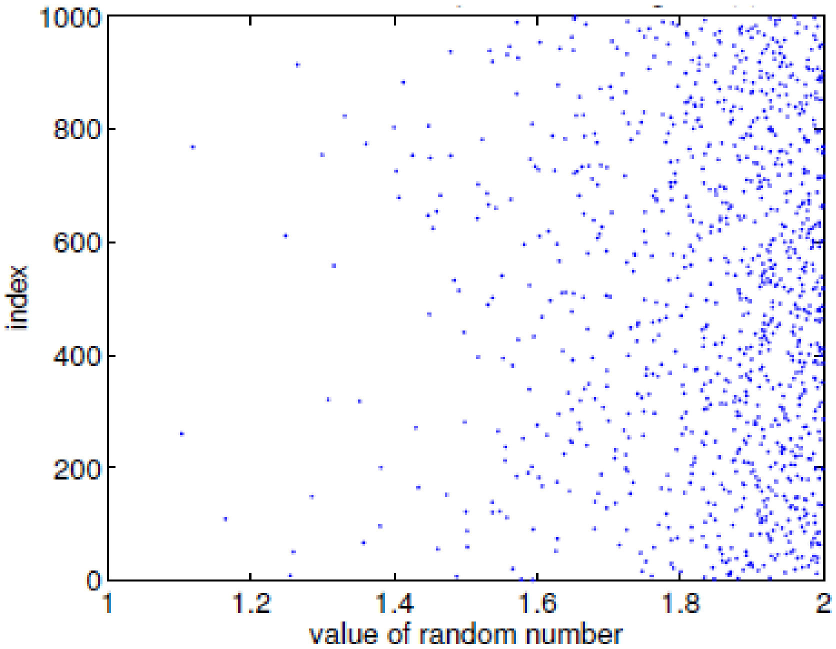


- In our example:

$$x = G^{-1}(y) = [(2^{11} - 1)y + 1]^{1/11}$$

- With y uniform in [0,1]

- So what we really did is to change from uniform sampling x to non-uniform sampling x because $x = G^{-1}(y)$



- In our example:

  $$x = G^{-1}(y) = [(2^{11}-1)y + 1]^{1/11}$$

- With y uniform in [0,1]

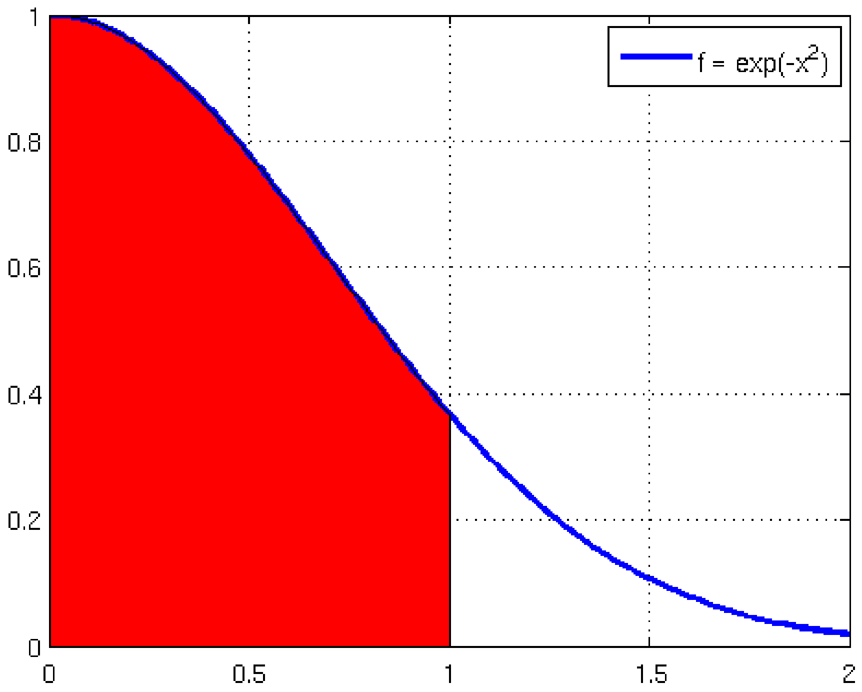
- Instead of using directly the flat random number generator, we transformed it to an engine that gives numbers according to a non-uniform probability density function (pdf) g(x) and a cumulative distribution function G(x)

- Programming exercise 2:
  - Write a program (in any programming language, preferable in C++, C, FORTRAN) that integrates a Gaussian function

$$f(x) = e^{-x^2}$$

- Programming exercise 2:
  - Write a program (in any programming language, preferable in C++, C, FORTRAN) that integrates a Gaussian function

$$f(x) = e^{-x^2}$$



- Programming exercise 2:
  - Write a program (in any programming language, preferable in C++, C, FORTRAN) that integrates a Gaussian function

$$f(x) = e^{-x^2}$$

    in [0,1] using two MC integration techniques:
    - MC Sampling
    - Importance sampling (hint use e^{-x} as your g(x))
  - Compare the two methods in terms of convergence and variance

- We mentioned a second advanced MC integration method, called Control Variates
- Will be discussed in the next lecture alongside with Random Number Generators
- …getting closer to MC particle simulation

- Website of the lecture material (slides):

http://www.physik.uni-muenchen.de/lehre/vorlesungen/wise_15_16/Vorlesung_-Computational-methods-in-medical-physics/index.html

- Exercises to be sent via email (G.Dedes@physik.uni-muenchen.de) as a zip file:

  - Containing all source code and make files
  - Short report on the findings

- The name of the zip file should be:
  - Exercise12_NameLastname.zip