Numerical Methods

Monte Carlo Quadrature

Course 2022-2023

[DΣIM]

Monte Carlo Quadrature

- Consider the random sample $x_1, x_2, \ldots, x_n$ taken from a random variable distributed with a probability density $f(x)$ in a real domain D. Then an estimator of the mean value of the function $g(x)$ defined on this domain:

$$\langle G_N \rangle = E[g(x)] = \int_{-\infty}^{\infty} g(x) f(x) dx$$

- Is the value

$$G_N = \frac{1}{N} \sum_{i=1}^{N} g(x_i)$$

Monte Carlo Quadrature

- The variance of this estimator is obtained as:

- Where $$Var(G_N) = \frac{Var(g)}{N}$$

$$Var(g) = E\left[g^2(x)\right] - E\left[g(x)\right]^2$$

- If this variance exists, this means that the possible values of the estimator of the mean, G_N are distributed around the mean value approximately as $N^{-1/2}$

Monte Carlo Quadrature

- These relations are also valid if we use discrete random variables.
- If the values $x_1, x_2, \ldots, x_N$ form a random sample of a discrete random variable, then we define the mean of a function $g(x)$ defined on this discrete domain as:

$$E\left[g(x)\right] = \left\langle G \right\rangle = \sum_i p_i g(x_i)$$

- An estimator of this mean is:

$$G_N = \frac{1}{N}\sum_{i=1}^{N} g(x_i)$$

Monte Carlo Quadrature

- Consider the integral:

$$\int_0^1 \sqrt{1-x^2}\,dx = \frac{\pi}{4}$$

- We could compute this value using the uniform probability distribution defined on the interval [0,1]

$$f_U(x) = 1, \quad 0 \le x \le 1$$

- And rewrite the integral as:

$$\int_0^1 \sqrt{1-x^2}\, f_U(x)\,dx = \frac{\pi}{4}$$

[DEIM]

Monte Carlo Quadrature

- Taking a random sample $x_1,x_2,\ldots,x_n$ uniformly distributed in the interval $[0,1]$ we can compute an approximate value of this integral using the function:

$$g(x) = \sqrt{1-x^2}$$

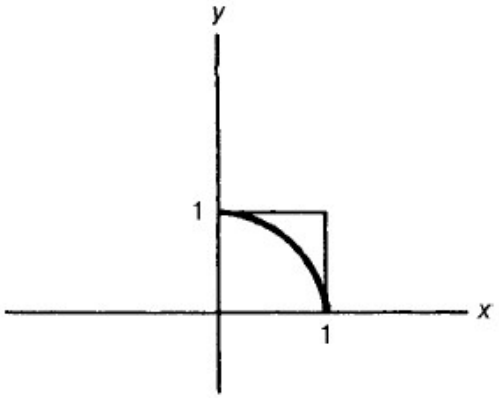
- To obtain the estimation:

$$I = \int_0^1 g(x)dx \approx \frac{1}{N}\sum_{i=1}^{N} g(x_i) = \frac{1}{N}\sum_{i=1}^{N}\sqrt{1-x_i^2}$$

[DΣIM]

Monte Carlo Quadrature

- There are also different approaches to the problem of computing this integral using random numbers.
- Consider the unit square on the xy plane containing the first quadrant of the circle of unit radius:



[DΣIM]

Monte Carlo Quadrature

- We could throw randomly points on this square uniformly distributed. The we could count all the points that fall in the area enclosed by the curve of the unit radius circle
- This procedure would be an estimation of the integral

$$\int_0^1 \int_0^1 g(x,y) f(x,y) dxdy = \frac{\pi}{4}$$

- Where

$$f(x,y) = \begin{cases} 1 & \text{if } (x,y) \in [0,1]^2 \\ 0 & \text{else} \end{cases}$$

Monte Carlo Quadrature

- The function to be integrated in this case is:

$$g(x,y)=\begin{cases}1 & \text{si } x^2+y^2\le 1\\ 0 & \text{si } x^2+y^2>1\end{cases}$$

- As the values of the variables x and y are independent we have:

$$f(x,y)=f_U(x)f_U(y)$$

- This means that the random sample of points (x_i,y_i) can be build using coordinates taken from independent uniform random variables defined on [0,1].

[DΣIM]

Monte Carlo Quadrature

- Both methods will show a similar performance to estimate the numerical value of the integral. In any case, the error goes as $N^{-1/2}$ with the size of the sample N, and these methods does not seem to be a good idea to compute one-dimensional integrals.
- In any case, note that we have been able to reconfigure a one-dimensional integral to a two-dimensional integral without loosing any performance in the computation.

Monte Carlo Quadrature

- In fact, all these ideas can be generalized with no effort to compute muldi-dimensional integrals of multi-variable functions defined within an hypercube of the form:

$$\int \ldots \int g(x_1, x_2, \ldots, x_L) dx_1 dx_2 \ldots dx_L$$

- We just need to create a random sample of values of the form $(x_1, x_2, \ldots, x_L)$, using L independent uniform random variables.

[DIM]

Monte Carlo Quadrature

- Moreover, if we could be able to generate random samples from any random distribution, this method will not be restricted to the integration domains based on hypercubes of the from $[0,1]^L$.
- Then, we should be also able to generate random values according the shape of the function to be integrated, with more random points where the function takes higher values and is "more important".

Classical Quadrature

- We want to compute integrals of the form:

$$\int_A f(\mathbf{x}) d\mathbf{x}$$

- Where the domain of definition A is a subset of $\mathbb{R}^n$ and f is a multi-variable function defined on the domain A onto the real set of numbers:

$$f : A \subset \mathbb{R}^n \rightarrow \mathbb{R}$$

Classical Quadrature

- If the function f is bounded and we can write the definition domain A in the form:

$$A = \prod_{i=1}^{k} [a_i, b_i]$$

- Where a_i and b_i take real values, we have $A \subset \mathbb{R}^n$, i.e. A is an hypercube, we are dealing with the Reimann integral on the region A.

Classical Quadrature

- If this integral exists, it is defined as the limit of a sequence of sums of the form:

$$\sum_{i=1}^{N} f(\mathbf{x}_i)\mathrm{vol}(\mathbf{R}_i)$$

- Where the hypercube A is partitioned in smaller hypercubes $\boldsymbol{R}_i$ of $\mathbb{R}^n$ in a way that $\boldsymbol{x}_i$ is a point within $\mathbf{R}_\mathrm{i}$

Classical Quadrature

- Then we have:

$$\int_A f(\mathbf{x})d\mathbf{x} = \lim_{\max\{diam(\mathbf{R}_1),\ldots,diam(\mathbf{R}_N)\}\to 0} \sum_{i=1}^{N} f(\mathbf{x}_i)vol(\mathbf{R}_i)$$

- The classical numerical methods of quadrature are built inspired in this definition. In the case of one dimension, if we want to compute the integral:

$$I = \int_0^1 g(x)dx$$

[DΣIM]

Classical Quadrature

- We create numerical schema where we evaluate the function on a certain number of points, say m, in the domain: $r_1, \ldots, r_m$ and then we compute a numerical approximation of the form:

$$I \approx \sum_{i=1}^{m} w_i g(r_i)$$

- Where we can use some values w_i as a set of weights that give shape to a particular Integration method.

Classical Quadrature

- When we have defined our numerical schema, we can increase the number of function evaluations to $n = km$, dividing the original domain in n subintervals of the form $[ik, (i+1)k]$ with $0 \leq i < k$, in such a way that we approximate:

$$I_i = \int_{i/k}^{(i+1)/k} g(x)dx$$

- by

$$I_i \approx \frac{1}{k}\sum_{j=1}^{m} w_j g\left(\frac{i}{k} + \frac{r_j}{k}\right)$$

[DΣIM]

Classical Quadrature

- The approximate value of the integral results from the addition of all these approximations:

$$I \approx \frac{1}{k}\sum_{j=1}^{m}\sum_{i=0}^{k-1} w_j g(i/k + r_j/k)$$

- Each subinterval has a length of $h = 1/k$. If this value is small enough, the function to be integrated g, can be approximated reasonable by a polynomial of certain degree p within this region.

Integració Clàssica

- If we choose appropriately the weights w_i we can obtain an exact approximation for a polynomials of degree p in such a way that the error in the Integration schema is of the order N^{-p} where N is the total number of function evaluations. The additional m free weights can increase the precision to an order of $N^{-(p+m)}$ as in the case of Gaussian quadrature.
- In the case of one-dimensional integrals, we can obtain great precision increasing the divisions an the number of function evaluations.

[DΣIM]

Classical Quadrature

- If we try to apply these ideas in the case of more dimensions, d, to compute an integral of the form:

$$I = \int_{H_d} g(\mathbf{x}) d\mathbf{x}$$

- Where H_d is a domain of d dimensions, of volume L^d and we evaluate the function in N points of this d-dimensional hypercube H_d: $\boldsymbol{r}_1, \ldots, \boldsymbol{r}_N$. If we use additional weights $w_1, \ldots, w_N$ we have:

$$I \approx \sum_{i=1}^{N} w_i g(\mathbf{r}_i)$$

Classical Quadrature

- We can increase the precision if we partition the domain of dimension d, and size L using k sub-hypercubes, each one of volume $(\frac{L}{k})^d$ and build an approximation of the form:

$$I \approx \sum_{i=1}^{k} \sum_{j=1}^{N} w_{ij} g(r_{i_1}, \ldots, r_{i_N})$$

- Again, the weights w_{ij} can be chosen to ensure that the approximation is exact for a monomials of degree lower or equal to m

[DΣIM]

Classical Quadrature

- This schema can bring the approximation to an error of the order n^{-m} where n is now the total number of function evaluations in the domain. However this total number is $n=(kN)^d$ as the number of sub-hypercubes depend on the dimension of the domain.
- The precision can will be of the order of $(kN)^{-m/d}$ and will be rapidly degraded when increasing the dimension of the domain d. This unwanted phenomena is know as the ***"curse of dimensionality"***

Monte Carlo Quadrature

- We can approach the problem of computing the integral of a function $g(\boldsymbol{x})$ in d dimensions as follows:

$$I = \int_{H_d} g(\mathbf{x}) f(\mathbf{x}) d\mathbf{x}$$

- We consider a probability distribution f defined on the hypercube H_d. In the simpler case, we can use the uniform distribution. From this distribution we take a random sample of size m: $\boldsymbol{r}_1, \ldots, \boldsymbol{r}_m$

[DΣIM]

Monte Carlo Quadrature

- Our first approximation to the integral value will be the estimator:

$$I \approx \frac{1}{m}\sum_{i=1}^{m} g(\mathbf{r}_i)$$

- The variance of the estimator can be used as a measure of the precision:

$$\sigma(I) = \frac{\sigma(g)}{\sqrt{m}}$$

- This error will not depend on the dimensionality of the problem, but only on the number of random points used.

Importance Sampling

- The error in Monte Carlo methods will be associated to the variance of the probability distribution. If we reduce this variance, we can increase our precision. The method of "***Importance Sampling***" is devised with this objective.
- Suppose that we want to compute the following integral defined in a domain A included in an space of d dimensions:

$$I = \int_A g(\mathbf{x}) f(\mathbf{x}) d\mathbf{x}$$

Importance Sampling

- The function f do not need to be necessarily a given probability distribution, or simply it is not the best choice.
- Suppose that we want to introduce a new density Distribution of probability, say f^*. We can rewrite our integral as:

$$I = \int_A \frac{g(\mathbf{x}) f(\mathbf{x})}{f^*(\mathbf{x})} f^*(\mathbf{x}) d\mathbf{x}$$

Importance Sampling

- As the function f^* is a probability density distribution, it must verify the following conditions:

$$f^*(\mathbf{x}) \geq 0, \quad \int_{\mathbb{R}^d} f^*(\mathbf{x}) d\mathbf{x} = 1$$

- On the other hand, as the integral must exist we have the condition:

$$\frac{g(\mathbf{x}) f(\mathbf{x})}{f^*(\mathbf{x})} < \infty$$

- Except maybe, in countable set of points of the domain A

Importance Sampling

- Now, the integral estimator becomes:

$$I = \frac{1}{N}\sum_{i=1}^{N}\frac{g(\mathbf{x}_i)f(\mathbf{x}_i)}{f^*(\mathbf{x}_i)} = \frac{1}{N}\sum_{i=1}^{N} g(\mathbf{x}_i)W(\mathbf{x}_i)$$

- Where the weights come from a weight function $W(\mathbf{x})$ known as the ***importance function***:

$$W(\mathbf{x}) = \frac{f(\mathbf{x})}{f^*(\mathbf{x})}$$

- Note also that in this case the random sample must be taken from the new distribution f^* and not f

Mostreig d'Importància

- The variance of this new estimator can be computed as follows. First the value of the integral is associated to its mathematical expectation:

$$I = E\left[\frac{g(\mathbf{x})f(\mathbf{x})}{f^*(\mathbf{x})}\right] = \int_A \frac{g(\mathbf{x})f(\mathbf{x})}{f^*(\mathbf{x})} f^*(\mathbf{x})d\mathbf{x}$$

- Then, its variance is defined as:

$$Var(I)_{f^*} = E\left[\frac{g^2(\mathbf{x})f^2(\mathbf{x})}{f^{2*}(\mathbf{x})}\right] - E\left[\frac{g(\mathbf{x})f(\mathbf{x})}{f^*(\mathbf{x})}\right]^2 = \int_A \frac{g^2(\mathbf{x})f^2(\mathbf{x})}{f^{2*}(\mathbf{x})} f^*(\mathbf{x})d\mathbf{x} - I^2$$

Importance Sampling

- The second term in this expression is I^2, which is the square of the value to be computed. To reduce the variance we can only work with the first term of this expression.
- The simpler way to reduce the variance is to increase the value of $f^*(\mathbf{x})$, while keeping its shape as a probability density distribution. This means that is integral over $\mathbb{R}^d$ must always be the unity.

$$f^*(\mathbf{x}) \geq 0, \quad \int_{\mathbb{R}^d} f^*(\mathbf{x}) d\mathbf{x} = 1$$

Mostreig d'Importància

- To find the function f^* that minimizes the variance we can use the Lagrange multipliers to find this conditioned minimum:

$$L\{f^*,\lambda\} = \left[\int_A \frac{g^2(\mathbf{x})f^2(\mathbf{x})}{f^{2*}(\mathbf{x})} f^*(\mathbf{x})d\mathbf{x} + \lambda\int_A f^*(\mathbf{x})d\mathbf{x}\right]$$

$$= \left[\int_A \frac{g^2(\mathbf{x})f^2(\mathbf{x})}{f^*(\mathbf{x})} d\mathbf{x} + \lambda\int_A f^*(\mathbf{x})d\mathbf{x}\right]$$

Importance Sampling

- Deriving the last expression and equating to zero we obtain:

$$-\frac{g^2(\mathbf{x})f^2(\mathbf{x})}{f^{2*}(\mathbf{x})}+\lambda=0$$

- And so:

$$f^*(\mathbf{x})=\lambda\left[g(\mathbf{x})f(\mathbf{x})\right]$$

- The probability density distribution must be proportional to the function to be integrated.

[DΣIM]

Importance Sampling

- The value of the multiplier λ can be obtained from the relation:

$$\int_{\mathbb{R}^n} f^*(\mathbf{x})d\mathbf{x} = \int_{\mathbb{R}^n} \lambda\left[g(\mathbf{x})f(\mathbf{x})\right] = 1$$

- Which is simply:

$$\lambda = \frac{1}{I}$$

- But if we want to know the value of λ we should compute first the integral.

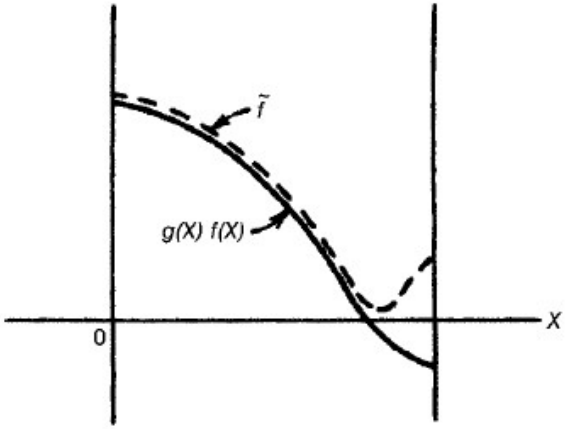
Importance Sampling

- This value of the multiplier, however, will give a zero value for the variance, indeed a minimum value. Knowing exactly the value of I, there will be zero variance.
- The idea is still useful, we can obtain lower values of the variance if we use distribution functions f^*, similar in shape to the function to be integrated.

Importance Sampling

- Using this procedure, we ensure that the random points of the sample come mainly from the "important" zones of the function to be integrated.
- We have only to keep the condition

$$\frac{g(\mathbf{x}_i) f(\mathbf{x}_i)}{f^*(\mathbf{x}_i)} < \infty$$



- Using a bounded function.

[DΣIM]

Importance Sampling

- Consider the integral:

$$\int_0^1 \cos\left(\frac{\pi x}{2}\right) dx = \frac{2}{\pi}$$

- A Monte Carlo quadrature, using an uniform density distribution will give a variance:

$$\int_0^1 \cos^2\left(\frac{\pi x}{2}\right) dx - \left(\int_0^1 \cos\left(\frac{\pi x}{2}\right)\right)^2 = \frac{1}{2} - \left(\frac{2}{\pi}\right)^2 = 0.097$$

Importance Sampling

- But using a Taylor expansion:

$$\cos\left(\frac{\pi x}{2}\right) = 1 - \frac{\pi^2 x^2}{8} + \frac{\pi^4 x^4}{2^4 4!} - \cdots$$

- We observe that we can use a quadratic function to make a first order approximation of the function to be integrated. Thus, we can use:

$$f^*(x) = \frac{3}{2}\left(1 - x^2\right)$$

Importance Sampling

- The estimator becomes:

$$g^*(x) = \frac{g(x)}{f^*(x)} = \frac{2\cos(\pi x/2)}{3(1-x^2)}$$

- While the value of the variance is:

$$\int_0^1 \frac{2\cos^2\left(\frac{\pi x}{2}\right)}{3(1-x^2)} dx - (I)^2 = 0.40628 - \left(\frac{2}{\pi}\right)^2 = 0.000990$$

- 100 times lower.

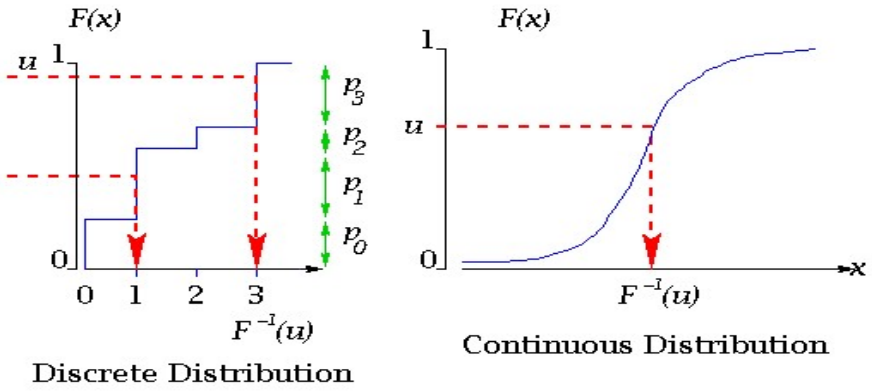
Inversion Algorithm

- The distribution function F of a random variable X is always strictly increasing:

$$x_1 < x_2 \Rightarrow F(x_1) < F(x_2)$$

- If the distribution function is easily invertible we can write:

$$y = F(x) \Leftrightarrow x = F^{-1}(y)$$



[DΣIM]

Inversion Algorithm

- This is true for the uniform distribution $U(0,1)$. If u is a value obtained from de distribution $U(0,1)$ the probability for this value to be lower than x is:

$$P(U \leq x) = F(x) = x$$

- And we can write:

$$u = F(x) = x \Leftrightarrow F^{-1}(u) = F^{-1}F(x) = x = F^{-1}(x)$$

- For the case of the uniform distribution $U(0,1)$ the random value u and its probability $x = P(u) = u$ are the same.

Inversion Algorithm

- Suppose that the random variable X has an invertible distribution function such that:

$$X = F^{-1}(U)$$

- Where U is distributed as $U(0,1)$. Then:

$$P(X \leq x) = P\left(F^{-1}(U) \leq x\right) = P(U \leq F(x)) = F(x)$$

- The values $F^{-1}(U)$ will be distributed with this probability law.

[DΣIM]

Inversion Algorithm

- The algorithm to generate a random set distributed as the distribution law F is quite simple:
  - ***Inversion Algorithm:***
    - Generate $u \sim U(0,1)$
    - Take the value $x = F^{-1}(u)$

Inversion Algorithm

- Suppose that we want to generate a random sample coming from the random variable X with distribution function:

$$F(x) = \begin{cases} 0 & \text{si } x < 0 \\ x^2 & \text{si } 0 \le x \le 1 \\ 1 & \text{si } x > 1 \end{cases}$$

- In this case we have simply:

$$X = F^{-1}(U) = \sqrt{U}$$

Rejection Sampling

- This algorithm is of general application, even in large dimensions. It is due to von Neumann and it can be quite fast, specially for complicated distributions.
- Suppose that we want to generate a random sample distributed with the distribution probability law F and with density distribution f
- We will need a majorant function t always greater than the density distribution f:

$$t(x) \geq f(x) \quad \forall x$$

[DΣIM]

Rejection Sampling

- This function t will not be a probability density distribution as:

$$c = \int_{-\infty}^{\infty} t(x)dx \geq \int_{-\infty}^{\infty} f(x)dx = 1$$

- If t is bounded, however, the function

$$r(x) = \frac{t(x)}{c}$$

- Will be a probability density.

Rejection Sampling

- This algorithm needs to follow the following steps:
  - ***Rejection Sampling:***
  - 1) Generate a first value y with density distribution r
  - 2) Generate a second value $u \sim U(0,1)$ independent from y
  - 3) If $u \leq f(y)/t(y)$ we return the value $x = y$ otherwise we discard y and go back to the first step.

Rejection Sampling

- We have to prove that these values will be distributed with the density law *f*.

$$P(X \leq x) = \int_{-\infty}^{x} f(y)dy$$

- Let A be the event where we accept the value x. This is a subset of the common definition space of the variables y and u with joint distribution function

$$g(y,u) = f(y)r(u)$$

Rejection Sampling

- On the other hand if the even A is true, we have $y = x$ and so:

$$P(X \leq x) = P(Y \leq y \mid A)$$

- The conditional probability is defined as:

$$P(Y \leq y \mid A) = \frac{P\left(A \cap Y \leq y\right)}{P(A)}$$

Rejection Sampling

- We note also that

$$P(A \mid Y = y) = P\left[U \leq \frac{f(y)}{t(y)}\right] = \frac{f(y)}{t(y)}$$

- As the values of u follow a uniform distribution $U(0,1)$ and are independent of the y values.

Rejection Sampling

- Then we have:

$$P(A \cap Y \le x) = \int_{-\infty}^{x} P(A \cap Y \le x \mid Y = y) r(y) dy$$

$$= \int_{-\infty}^{x} P(A \mid Y = y) \frac{t(y)}{c} dy$$

$$= \frac{1}{c} \int_{-\infty}^{x} f(y) dy$$

Rejection Sampling

- Finally we note:

$$P(A) = \int_{-\infty}^{\infty} P(A \mid Y = y) r(y) dy$$

$$= \int_{-\infty}^{\infty} \frac{f(y)}{t(y)} \frac{t(y)}{c} dy = \frac{1}{c} \int_{-\infty}^{\infty} f(y) dy = \frac{1}{c}$$

- And so:

$$P(X \le x) = P(Y \le x \mid A) = \int_{-\infty}^{x} f(y) dy$$

[DΣIM]

Numerical Methods

Rejection Sampling

```
% Exemple del Metode de A-R
% f(x)=2x 0<x<1
c = 2;
n = 100;
% Valors guardats
x = zeros(1,n);
xy = zeros(1,n);
% Valors rebutjats
rej = zeros(1,2*n);
rejy = zeros(1,2*n);
% Loop fins tenir tots els n valors
irv = 0;
irej = 0;
while irv < n
    y = rand(1);
    u = rand(1);
    if u <= 2*y/c;
        % Valor acceptat
        irv = irv + 1;
        x(irv)=y;
        xy(irv)=u*c;
    else
        % Valor rebutjat
        irej = irej + 1;
        rej(irej)=y;
        rejy(irej) = u*c;
    end
end
```

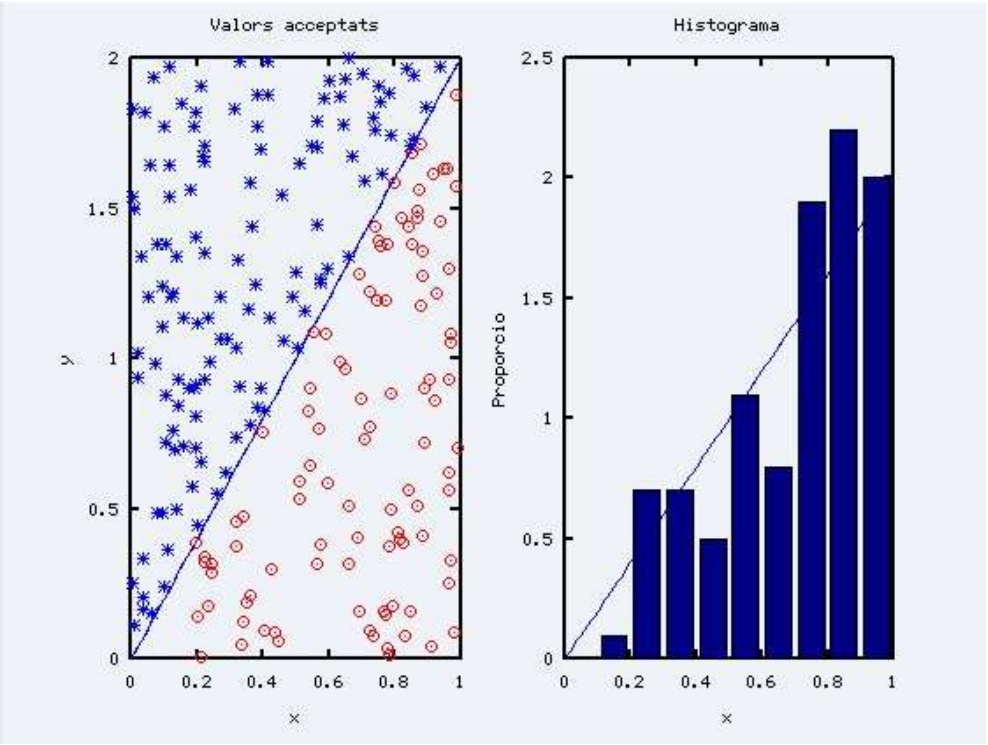
```
% Dibuix dels valors
subplot(1,2,1)
xv = 0:0.01:1;
yv = c*xv;
plot(xv,yv);
xlabel('x')
ylabel('y')
title('Valors acceptats')
hold on
for i=1:irv
    plot(x(i),xy(i),'ro')
end
for i=1:irej
    plot(rej(i),rejy(i),'b*')
end
subplot(1,2,2)
plot(xv,yv);
xlabel('x')
ylabel('Proporcio')
title('Histograma')
hold on
% Histograma dels valors generats
xc = 0.05:0.1:0.95;
N=hist(x,xc);
% Reescalem
N = N /(xc(2)-xc(1))/ n;
% Dibuix histograma
bar(xc,N)
```

Course 2022-2023

[DΣIM]

Numerical Methods

Rejection Sampling



Course 2022-2023

[DΣIM]

Mostreig d'Importància

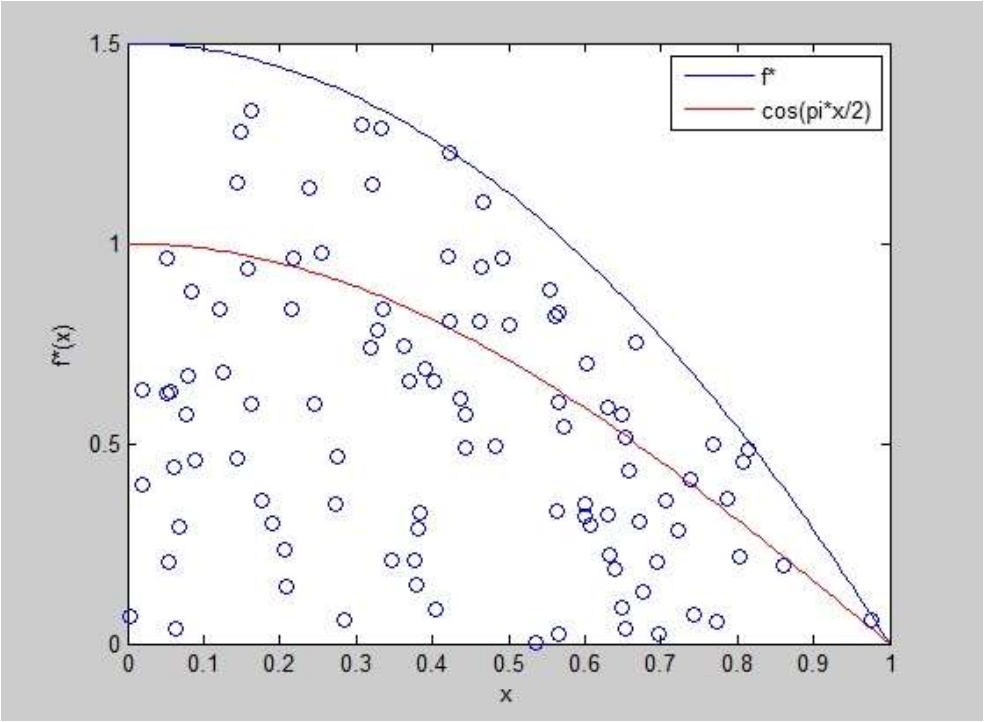
```
%impsam.m
% Calcul aproximat de una integral
% Variables de dibuix
xp = 0:0.01:1;
yc = cos(pi*xp/2);
s = length(xp)
for i = 1:length(xp)
yp(i) = 3.0/2.0*(1-xp(i)*xp(i));
end
plot(xp,yp,'b')
hold on
plot(xp,yc,'r')
xlabel('x')
ylabel('f*(x)')
legend('f*','cos(pi*x/2)','Location','northeast')
hold on
format long
```

```
% Primera aproximacio. Mostra uniforme
N = 100;
x = rand(1,N);
y = cos(pi*x/2);
% Fem ara una mostra de tamany N de la nova
% distribució
% f = 3/2*(1-x^2)
% Fem servir el metode A-R
% El maxim es 3/2
i = 0;
while (i < N)
    xa = rand;
    ya = rand;
    if (ya < (1-xa*xa))
        i = i + 1;
        xr(i) = xa;
        yr(i) = 2*cos(pi*xr(i)/2) / (3*(1-xr(i)^2));
        plot(xa,ya*3/2,'o')
        hold on
    end
end
% Amb els valors xr fem una nova estimacio
I1 = 1/N * sum(y)
I2 = 1/N * sum(yr)
exact = 2/pi
```


[DΣIM]

Numerical Methods

Importance Sampling



Course 2022-2023

[DEIM]

Mostreig d'Importància

- I en una única estimació tenim els valors

```
I1 =

   0.611151731750647

I2 =

   0.634254798514692

exact =

   0.636619772367581
```