Computational Statistics
M.Sc. Mathematical Engineering
Prof. A. Manzoni

A.A. 2020 – 2021
Politecnico di Milano

Hands-On 4
Monte Carlo Methods

Problem 4.1 (Computation of integrals with Monte Carlo).
We want to evaluate numerically the integral in dimension d

$$I = \int_{\mathbb{R}^d} f(\mathbf{x})p(\mathbf{x})d\mathbf{x}$$

where $p(x)$ is a positive function such that $\int_{\mathbb{R}^d} p(\mathbf{x})d\mathbf{x} = 1$ and $f \in L^2_p(\mathbb{R}^d)$, that is, such that $\int_{\mathbb{R}^d} f^2(\mathbf{x})p(\mathbf{x})d\mathbf{x} < +\infty$. Denote by $\mathbf{X}$ the random vector having joint probability density p and by Y the random variable $Y = f(\mathbf{X})$. The value of the integral is then equal to

$$I = \int_{\mathbb{R}^d} f(\mathbf{x})p(\mathbf{x})d\mathbf{x} = \mathbb{E}\left[f(\mathbf{X})\right] = \mathbb{E}\left[Y\right].$$

Assuming to be able to effectively sample the vector $\mathbf{X}$, the integral I can be approximated by the Monte Carlo method using the sample mean estimator for the random variable $Y = f(\mathbf{X})$:

$$I \approx I_n = \frac{1}{n}\sum_{i=1}^{n} Y^{(i)} = \frac{1}{n}\sum_{i=1}^{n} f(\mathbf{X}^{(i)}).$$

1. Calculate the integral in dimension 1:

$$I = \int_{-\infty}^{+\infty} x^2 e^{x-\frac{x^2}{2}} dx,$$

using the Monte Carlo method. Take a standard normal variable as variable X and use $n = 10^4$ samples. Compare the result obtained with the exact value $I = \sqrt{8\pi e}$.

2. Using the Central Limit Theorem, provide a confidence interval of level $1 - \alpha$ for the estimate obtained in the previous point. Take $\alpha = 0.05$. Verify the correctness of the result by repeating the calculation 20 times.

3. Now consider an increasing number of samples: $n = 2^5, 2^6, \ldots, 2^{20}$ and, for each choice of n, calculate the value of the integral I_n and the relative error committed. Display on a graph, in logarithmic scale, the trend of the relative error $|I_n - I|/I$ as a function of n. Whick kind of trend do you observe?

4. Resume the confidence interval estimate provided at point b). If we want to obtain a relative error $|I_n - I|/I \leq \varepsilon$, with $\varepsilon = 0.01$, how many samples should we need to use?

5. Propose an adaptive algorithm for choosing the number of samples n in order to satisfy an accuracy requirement $|I_n - I|/I \leq \varepsilon$ with confidence level $(1 - \alpha)$.

6. The `gauss.m` function supplies the n nodes and corresponding weights of the Gaussian quadrature formula with weight e^{-x^2} (Gauss-Hermite). Use this formula to calculate numerical- the integral I. Use $n = 3, 4, 5, 6$ points and compare the accuracy of the result obtained with the method of the previous point.

7. Consider now the integral in dimension d

$$I = \int_{\mathbb{R}^d} \sqrt{\sum_{i=1}^{d} x_i^4 e^{2x_i}} e^{-\frac{\sum_{i=1}^{d} x_i^2}{2}} d\mathbf{x}.$$

As the dimension d increases, how many samples are expected to have to be used to approximate the integral with Monte Carlo ensuring the same accuracy as in point b)? And how many Gauss-Hermite nodes, if you want to use the Gaussian quadrature formula instead? Numerical tests have to be carried out.

best quadrature formula

In this case we can check what happens if we tensorize it (we increase the dimension of the space): we encounter the curse of dimensionality → for growing dimensions Monte Carlo is the way to go

1. The integral can be rewritten as

$$I = \int_{-\infty}^{\infty} \underbrace{\sqrt{2\pi}x^2 e^x}_{f(x)} \underbrace{\frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}}_{p(x)} dx$$

that is, isolating as weight p the PDF of a standard Gaussian random variable. Evaluating the integral using the Monte Carlo method can then be done using the following commands:

```
n=1e4;
f= @(x)  sqrt(2*pi)*x.^2.*exp(x)
Iex=sqrt(8*pi*exp(1));
X=randn(1,n);
Y=f(X);
In=mean(Y)
   In = 8.6135
err=abs(In-Iex)/Iex
   err = 0.0421
```

2. Let us denote by $\bar{Y}_n$ the sample mean,

$$\bar{Y}_n = \frac{1}{n}\sum_{i=1}^{n} Y^{(i)}$$

and by $\sigma_Y = \sqrt{\mathrm{Var}[Y]}$ the standard deviation of Y. The central limit theorem states that

$$\frac{\bar{Y}_n - \mathbb{E}[Y]}{\sigma_Y/\sqrt{n}} \rightharpoonup N(0,1) \qquad \text{as } n \to \infty;$$

hence, denoting by $\Phi(x)$ the CDF of a standard normal random variable, and by $z_\alpha = \Phi^{-1}(\alpha)$, we have

$$P\left(\frac{\bar{Y}_n - \mathbb{E}[Y]}{\sigma_Y/\sqrt{n}} \le z_{1-\alpha/2}\right) \to 1-\alpha \text{as } n \to \infty$$

or, in other words, that with probability $1-\alpha$ we have

$$\mathbb{E}[Y] \in \left[\bar{Y}_n - z_{1-\alpha/2}\frac{\sigma_Y}{\sqrt{n}}, \bar{Y}_n + z_{1-\alpha/2}\frac{\sigma_Y}{\sqrt{n}}\right].$$

We can replace σ_Y with the sample variance

$$\hat{\sigma}_Y = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(Y_i - \bar{Y}_n)^2}$$

To compute the quantile $z_{1-\alpha/2}$ in `Matlab`, the Statistical Toolbox provides the function `norminv` that evaluates the function Φ^{-1} at a given point (or vector of points). For instance, with $\alpha = 0.05$,

```
alpha=0.05;
za=norminv(1-alpha/2)
   za = 1.9600
```

Alternatively, the function `erfinv` (built-in in `Matlab`) returns the inverse of the error function

$$\mathrm{erf}(x) = \frac{2}{\pi}\int_0^x e^{-t^2} dt.$$

Indeed, recall that

$$\Phi(x) = \frac{1}{2}\left[1 + \mathrm{erf}\left(\frac{x}{\sqrt{2}}\right)\right], \qquad \Phi^{-1}(y) = \sqrt{2}\mathrm{erf}^{-1}(2y-1).$$

Hence, the quantile $z_{1-\alpha/2}$ can be computed as

$$z_{1-\alpha/2} = \sqrt{2}\mathrm{erf}^{-1}(1-\alpha)$$

```
za=sqrt(2)*erfinv(1-alpha)
   za = 1.9600
```

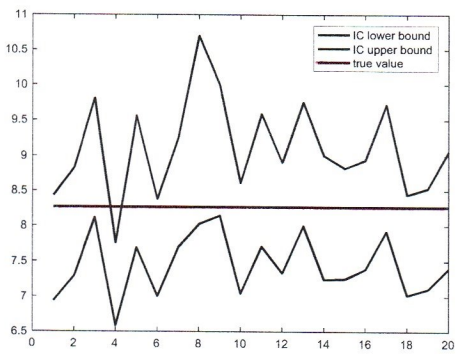
We can now compute the confidence interval for the sample mean estimator in `Matlab` as follows:

```
alpha=0.05;
za=sqrt(2)*erfinv(1-alpha);
sY=std(Y);
IC=[ In-za*sY/sqrt(n), In+za*sY/sqrt(n) ]
   IC = 7.822210219671316   9.404747742234935
```

We remark that the confidence interval is quite large ($\approx I_n \pm 10\%$) coherently with the result of the previous point. If we repeat the evaluation 20 times, we get the plot in Figure ??, left; only in a single case out of 20 we obtain a CI which does not cover the exact value.

3. We can generate an error plot for an increasing number of Monte Carlo samples through the following commands:

```
err=[]; nn=2.^[5:20];
for n=nn
   X=randn(1,n);
   Y=f(X);
   In=mean(Y); err=[err,abs(In-Iex)/Iex];
end
loglog(nn,err,'b-'); hold on
loglog(nn,1./sqrt(nn),'k--') % reference rate 1/sqrt(n)
grid on; hold off
```



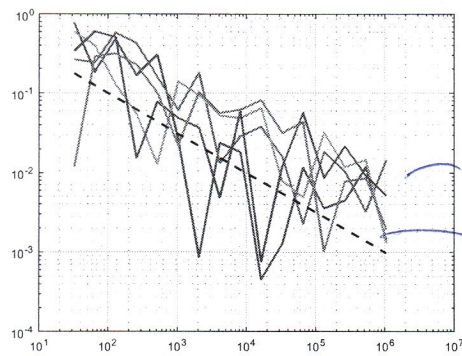


Figura 1: Left: 20 replicas of the evaluation of the 95% confidence interval. Right: relative error (five replicas) and theoretical rate $1/\sqrt{n}$.

Figure 1 shows the obtained result. The error has been computed five times, and compared against the theoretical rate $1/\sqrt{n}$. On average, the error decreases as $1/\sqrt{n}$, consistently with the estimate of the previous point, which states that, at a confidence level $1-\alpha$,

$$\frac{I_n - I}{I} = \frac{\bar{Y}_n - \mathbb{E}[Y]}{\mathbb{E}[Y]} \leq \frac{z_{1-\alpha/2}\sigma_Y}{\mathbb{E}[Y]\sqrt{n}}.$$

4. It is sufficient to impose that

$$\frac{z_{1-\alpha/2}\sigma_Y}{\mathbb{E}[Y]\sqrt{n}} \leq \varepsilon \qquad \Rightarrow \qquad n \geq \left(\frac{z_{1-\alpha/2}\sigma_Y}{\mathbb{E}[Y]\,\varepsilon}\right)^2$$

To make the previous estimate computable we can replace σ_Y and $\mathbb{E}[Y]$ with the corresponding sample. In `Matlab`:

```
alpha=0.05; tol=0.01;
za=sqrt(2)*erfinv(1-alpha);
X=randn(1,n);
Y=f(X);
In=mean(Y)
   In = 8.239150998819404
sY=std(Y);
n_opt=ceil( (za*sY/(In*tol))^2 )
   n_opt = 1051862
```

Note that the estimated number of samples required to fulfill the accuracy constraint is higher than a million!

5. The adaptive algorithm can be structured in the following way:

   0. Choose an initial number n of samples.
   1. Compute the approximation with the Monte Carlo method.
   2. Estimate the variation coefficient $\gamma_Y = \sigma_Y / \bar{Y}_n$.
   3. Use the estimate at point d) to determine the number $\tilde{n}(\varepsilon, \alpha)$ of required samples to get a relative error less than ε with confidence level $1 - \alpha$.
   4. If $\tilde{n}(\varepsilon, \alpha) > n$, set $n = \tilde{n}(\varepsilon, \alpha)$ and go back to point 1.

   In `Matlab` we have:

```
tol=0.01; alpha=0.05;
za=sqrt(2)*erfinv(1-alpha);
n=100; n_opt=n;
while n_opt>=n
   n=n_opt;
   X=randn(1,n);
   Y=f(X);
   In=mean(Y);  sY=std(Y);
   n_opt=ceil( (za*sY/(In*tol))^2 );
   fprintf('n. of samples used%d, optim estimated number %d, rel error%f\n',...
         n,n_opt,abs(In-Iex)/Iex)
end
```

```
n. of samples used 100, optim estimated number 364248, rel error 0.535642
n. of samples used 364248, optim estimated number 1010206, rel error 0.000052
n. of samples used 1010206, optim estimated number 1012518, rel error 0.006672
n. of samples used 1012518, optim estimated number 1058349, rel error 0.000766
n. of samples used 1058349, optim estimated number 1034045, rel error 0.003359
```

6. The Gauss-Hermite quadrature formula with n points allows to approximate the integral

$$I = \int_{-\infty}^{\infty} g(x) e^{-x^2} dx$$

as

$$I \approx \sum_{i=1}^{n} g(x_i) \omega_i$$

being x_i the quadrature nodes and ω_i the corresponding quadrature weights. In our example, in order to isolate the factor e^{-x^2} in the integral, we can write it as

$$I = \int_{-\infty}^{\infty} x^2 e^x - \frac{x^2}{2} dx = \left\{ \text{ setting } y = x/\sqrt{2} \right\} = \int_{-\infty}^{\infty} 2\sqrt{2} y^2 e^{\sqrt{2}y} e^{-y^2} dy.$$

Hence, we can use the quadrature formula to integrate the function $g(x) = 2\sqrt{2} x^2 e^{\sqrt{2}x}$. In `Matlab`:

```
f = @(x) 2*sqrt(2)*x.^2.*exp(sqrt(2)*x);
I_ex = sqrt(8*pi*exp(1));
I = []; err=[];
for n = 3 : 6
    [x_g, w_g] = gauss(n);
    Ic = w_g * f(x_g) ';
    I = [I, Ic];
    err = [err, abs(Ic-I_ex)/Ic];
end
   I = 7.305762220146633   8.144355806837478
       8.254960778889657   8.264775046636778
 err = 0.131362130217138   0.014870040587597
       0.001272196154122   0.000083203911096
```

We see how the error decreases very rapidly (exponentially) with n. Already for $n = 5$ the error is below 0.01.

7. The error estimate provided at point a) does not depend on the dimension d. Therefore, it will be expected that the number of samples necessary to satisfy the accuracy requirements in the Monte Carlo method will not vary greatly as the spatial dimension increases. It is numerically verified that in dimension 10 it is necessary to use about 160000 points (even less than for $d = 1$). Conversely, if we use a Gaussian quadrature formula with n points in dimension 1, we will have to use n^d points in dimension d. We have seen at point f) that in dimension 1 $n = 5$ points are sufficient to satisfy the accuracy requirements. The same formula, in dimension 10, will however require $5^10 \approx 10^7$ points, a number much higher than that required by the Monte Carlo method.

Problem 4.2 (Calculation of volumes with Monte Carlo).
We want to calculate the volume of a domain $\Omega \subset \mathbb{R}^d$ numerically. Let us consider a hypercube $\Sigma = \Pi_{i=1}^{d}[a_i, b_i]$, containing Ω. Moreover, let $\mathbf{X}$ be a randomly chosen point in Σ with uniform distribution and $Y = Y(\mathbf{X})$ the Bernoulli random variable

$$Y(\mathbf{X}) = \begin{cases} 1 & \text{if} X \in \Omega \\ 0 & \text{if} X \in \Sigma \setminus \Omega \end{cases}$$

The volume V of Ω can therefore be calculated as

$$V = \mathbb{E}[Y] \cdot |\Sigma|.$$

An approximation V_n of V can be obtained using the Monte Carlo method which consists in sampling the random variable Y and calculating the sample mean of the values obtained:

$$V \approx V_n = \frac{|\Sigma|}{n} \sum_{i=1}^{n} Y^{(i)}$$

being $Y^{(i)}$, $i = 1, \ldots, n$ the sampled values.

1. Let Ω be the sphere of unit radius in dimension d and $\Sigma = [-1, 1]^d$. Calculate the volume of the sphere $V = |\Omega|$, in dimension $d = 3$ and $d = 8$, by the Monte Carlo method, using $n = 2^5, 2^6, \ldots, 2^{20}$ samples. Visualize on a graph, in logarithmic scale, the trend of the relative error $|V_n - V|/V$ as a function of n, knowing that the exact value is $V = 4/3\,\pi R^3$ in dimension 3 and $V = 1/24\,\pi^4 R^8$ in dimension 8.
2. Using the Central Limit Theorem (and a sample estimate of variance), estimate the number of samples n necessary to have a relative error less than 0.05 with confidence level $1 - \alpha = 0.95$ in calculating the volume of the sphere in dimension 3 and 8.

1. The Monte Carlo method can be used to compute the volume of the unit sphere, for instance in the case $d = 3$ using 1000 points we can use employ the following `Matlab` commands:

```
d = 3; n = 1000;
u = 2*rand(d,n) - 1;  % generates n points in [-1,1]^d
r = sqrt(sum(u.^2));  % computes the distance of each point from the origin
Y = r < 1;              % Bernoulli: Y=1 if r<1, Y=0 otherwise
Vn = 2^d*mean(Y)      % Monte Carlo estimator
   Vn = 4.1680
V_ex=4*pi/3
   Vn = 4.1887
```

The error for this realization is less than 1%. To analyze the behavior of the relative error as a function of n, we can proceed as follows:

```
d = 3; V_ex = 4*pi/3;
N = 2.^[5:20]; err=[];
for n = N
   u = 2*rand(d,n) - 1;
   r = sqrt(sum(u.^2));
   Y = r < 1;
   Vn= 2^d*mean(Y);
   err = [err, abs(Vn-V_ex)/V_ex];
end
figure(1)
loglog(N,err,'b', N,sqrt(1./N),'r','lineWidth',2)
legend('error', 'y=N^{-1/2}')
xlabel('N'); grid on
```

At the end of the cycle, the vector `err` contains the values of the relative error for $n = 2^5, 2^6, \ldots, 2^{20}$. We can note that the decay of the error, although not monotone, is independent of the dimension d, when compared with the decay rate of $y = n^{-1/2}$.

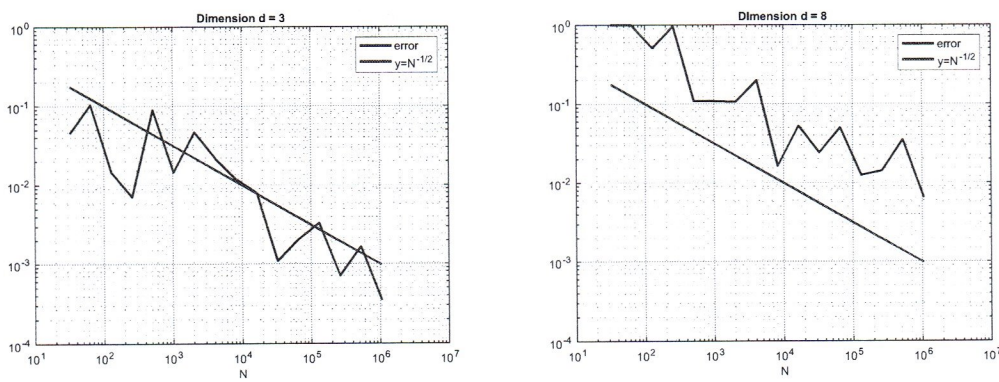


Figura 2: Relative error as a function of n, cases $d = 3$ (left) and $d = 8$ (right).

2. Remark that the relative error on V coincides with the relative error on $\mathbb{E}[Y]$:

$$\frac{|V - V_n|}{V} = \frac{|\mathbb{E}[Y] - \bar{Y}_n|}{\mathbb{E}[Y]}$$

where $\bar{Y}_n$ is the sample mean estimator $\bar{Y}_n = \Sigma_{i=1}^n Y^{(i)}$. With confidence level $1 - \alpha$,

$$\frac{|V - V_n|}{V} = \frac{|\mathbb{E}[Y] - \bar{Y}_n|}{\mathbb{E}[Y]} \leq z_{1-\alpha/2} \frac{\sigma_y}{\sqrt{n}} \cdot \frac{1}{\mathbb{E}[Y]}$$

and the number of samples necessary to fulfill the accuracy constraints can be estimated as

$$n \geq \left(\frac{z_{1-\alpha/2} \sigma_Y}{\mathbb{E}[Y]\, \varepsilon} \right)^2 .$$

Using $n = 10^5$ to estimate the sample variance, we have in Matlab:

```
d=3; tol=0.05; alpha=.05;
za=sqrt(2)*erfinv(1-alpha);
n=1e5;
u=2*rand(d,n)-1;
r=sqrt(sum(u.^2));
Y=r<1;
Vn=2^d*mean(Y)
   Vn = 4.1712

n_opt=ceil( (za*std(Y)/(tol*mean(Y)))^2 )
   n_opt = 1411
```

In the case $d = 8$,

```
   Vn = 4.0627
   n_opt = 95288
```

Note the huge difference in the number of optimal samples in the two cases $d = 3$ and $d = 8$. This is due to the fact that the ratio between the volume of the sphere and the volume of the hypercube Σ in dimension 8 is very small (much smaller than in dimension 3). Therefore, the event *the point is in the sphere* is a rare event and it will be necessary to throw many points in the hypercube so that a significant number of them fall into the sphere. Note that in both cases ($d = 3, 8$) the Monte Carlo convergence is of the type $C/\sqrt{n}$, however the constant is very different in the two cases.

How does MC perform in the case of rare events? [spoiler: bad]

Problem 4.3 (A simple physical model).
Consider the spring system represented in the figure below, consisting of four nominally identical springs with elastic constants k_i, $i = 1, \dots, 4$ and subject to unit concentrated forces $F_1 = F_2 = F_3 = 1$. Let x_1, x_2 and x_3 be the horizontal displacements of the three nodes indicated in Figure 3. The equilibrium equations of the horizontal component of the force in the nodes are

$$\begin{cases} k_1 x_1 + k_2(x_1 - x_2) = F_1 \\ k_2(x_2 - x_1) + k_3(x_2 - x_3) = F_2 \\ k_3(x_3 - x_2) + k_4 x_3 = F_3 \end{cases} \quad \Rightarrow \quad \begin{bmatrix} k_1 + k_2 & -k_2 & 0 \\ -k_2 & k_2 + k_3 & -k_3 \\ 0 & -k_3 & k_3 + k_4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} F_1 \\ F_2 \\ F_3 \end{bmatrix}$$

the output is the vector $[x_1, x_2, x_3]$ solution of this system for which the matrix is **random** (it depends on k_1, k_2 and k_3 which are random) (we use lognormal because the constant of a spring is > 0)

Now suppose that the four springs are randomly extracted from a population of springs having elasticity constant distributed as a lognormal random variable with parameters $\mu = 1$ and $\sigma = 0.5$, that is, $k = e^X$, with $X \sim N(\mu, \sigma^2)$. We want to calculate the probability that $x_2 > 1.2$, which corresponds to calculating the four-dimensional integral

$$P(x_2 > 1.2) = \int_{\mathbb{R}^4} 1_{x_2 > 1.2} p_k(\mathbf{x}) d\mathbf{x} \qquad (1)$$

$\underline{k} = [k_1, k_2, k_3, k_4]$

where p_k is the joint probability distribution of the elasticity constants k_1, k_2, k_3, k_4.

1. Sample 1000 times the random variable $x_2 = x_2(k_1, k_2, k_3, k_4)$ and represent them in a histogram.
2. Estimate, on the basis of the previous sampling, the number N of samples that must be used if we want to calculate the integral in (1) using the Monte Carlo method to have a relative error less than 0.05 with a confidence level $1 - \alpha = 0.95$. Then calculate the integral with Monte Carlo.
3. Repeat the previous point if you want to calculate the probability that $x_2 > 1.5$. What conclusions can be drawn about the possibility of calculating the probability of rare events using Monte Carlo?

we have to solve the linear system 1000 times

the integral is computed over all the possible realizations of $\underline{k}$ ($\in \mathbb{R}^4$)

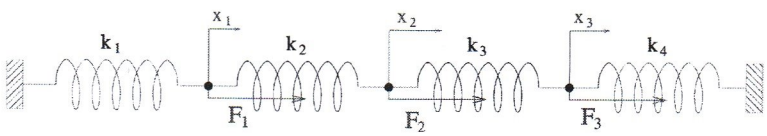


Figura 3: Three masses connected by four springs.

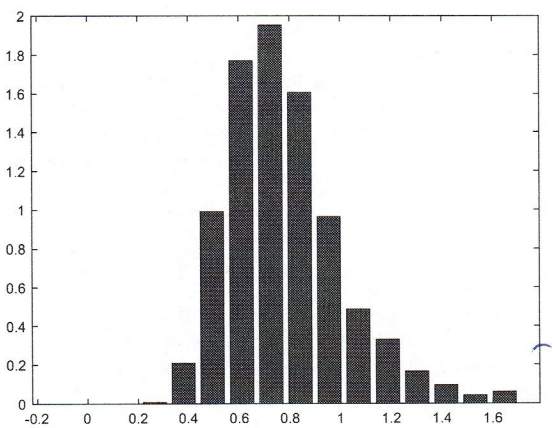
Notice: this problem underlies that the output can depend on the input in different ways. Here we're not sampling $\underline{x}$, we're sampling $\underline{k}$ and we're obtaining $\underline{x}$ from a linear system involving $\underline{k}$

1. For each sampling of the elastic constants $k_1, \ldots, k_4$, the vector $\mathbf{x} = [x1, x2, x3]$ can be computed solving a 3×3 linear system. The displacement x_2 then provides a sample of the desired distribution. In `Matlab`:

```
F1=1; F2=1; F3=1;    % nodal forces
km=1; ks=0.5;        % parameters, log-normal distribution
n=1000;
X2=zeros(1,n);
for i=1:n
        % sample elastic constants
        k1=exp(ks*randn+km);
        k2=exp(ks*randn+km);
        k3=exp(ks*randn+km);
        k4=exp(ks*randn+km);
        % solve linear system
        A=[k1+k2   -k2     0
            -k2   k2+k3   -k3
             0     -k3   k3+k4];
        xx=A\[F1, F2, F3]';
        X2(i)=xx(2);     % sample of x2
    end
Xm=mean(X2); Xs=std(X2);
dx=Xs/2; x=[Xm-4*Xs+dx/2: dx: Xm+4*Xs-dx/2];
P=hist(X2,x)/(n*dx); bar(x,P)
```

we sample from $N(0,1)$. we multiply by σ and we add μ. Then we do $\exp(\cdot)$ (sampling from a lognormal)

$= [x_1, x_2, x_3]$



with the histogram we see that $x_2 > 1.2$ is not an event extremely rare, however $x_2 > 1.5$ is extremely rare

Figura 4: Sampled values of the random variable $x_2 = x_2(k_1, k_2, k_3, k_4)$

2. Let us now define the Bernoulli random variable

$$Y = \begin{cases} 1 & \text{if } x_2 > 1.2 \\ 0 & \text{if } x_2 \leq 1.2. \end{cases}$$

The desired probability is then $P(x_2 > 1.2) = \mathbb{E}[Y]$. The number of required samples to get a relative error less than $\varepsilon = 0.05$ can be computed as

$$N = \left(\frac{z_{1-\alpha/2}\gamma_Y}{\varepsilon} \right)^2$$

this comes from the fact that: $|\text{error}| \leq z_{1-\frac{\alpha}{2}} \frac{\sigma}{\sqrt{N}} < \varepsilon$

being $\gamma_Y = \sigma_Y / \mathbb{E}[Y]$, estimated as the ratio between the sample standard deviation and the sample mean.

```
Y=X2>1.2;
gammaY=std(Y)/mean(Y);
N=ceil((erfinv(.95)*sqrt(2)*gammaY/.05)^2)
```

`N = 25447`

but we're looking for the **RELATIVE** error:

$$\frac{|\text{error}|}{\mathbb{E}[Y]} \leq \frac{z_{1-\frac{\alpha}{2}} \frac{\sigma}{\sqrt{N}}}{\mathbb{E}[Y]} < \varepsilon$$

the relative error is relative with the size of what is being measured

Repeating now the sampling at point a) with 25547 samples, we obtain

```
Y=X2>1.2;
I=mean(Y)
   I = 0.04826
```

3. Repeating point b) in the case we want to compute the probability that $x_2 > 1.5$, we find

```
Y=X2>1.5;
gammaY=std(Y)/mean(Y);
N=ceil((erfinv(.95)*sqrt(2)*gammaY/.05)^2)
   N = 188109
```

The number of samples required is considerably higher than in the previous case due to the fact that the coefficient of variation γ_Y is almost three times greater (11.064 against 4.0694). Remember that for a Bernoulli random variable Y, the variation coefficient is given by

$$\gamma_Y = \sqrt{\frac{1 - \mathbb{E}[Y]}{\mathbb{E}[Y]}}.$$

' It is clear, therefore, that the more the event whose probability is to be calculated is rare (i.e. $\mathbb{E}[Y]$ small), the greater the coefficient of variation and therefore the number of samples required. The calculation of rare events is therefore very expensive using the Monte Carlo method.