Romanian Journal of Mathematics and Computer Science
available online at https://rjm-cs.utcb.ro
Issue x, Vol. xx (20xx)

SOLVING LINEAR SYSTEMS USING MONTE CARLO METHODS

DANIEL CIUIU

Abstract. In this paper we solve linear systems with positive definite matrices using Monte Carlo methods. We first estimate the minimum and maximum eigenvalues of the matrix of the system to determine the region of the solution. Then, we simulate a large number of vectors in this region and estimate the minimum of the quadratic function involved in relaxation methods.

Mathematics Subject Classification (2010): 65C05, 91G60, 65F10, 65F15
Key words: Positive definite matrices, relaxation methods, Monte Carlo methods.

Article history:
Received: April 26, 2024
Received in revised form: June 12, 2025
Accepted: June 20, 2025

1. Introduction

In the relaxation methods a key result is the following theorem [9, 11].

Theorem 1.1. *Solving the linear system $Ax = b$, where the matrix A is symmetric and positive definite is equivalent to finding the minimum of $F(x) = \frac{1}{2}\langle x, Ax\rangle - \langle x, b\rangle$.*

The minimum/ maximum of a function Ψ on the domain D is obtained as follows [12].

(1) Simulate $nrsim$ uniform random variables on D or on another domain D_1 such that there exists a one-to-one continuous function $\phi : D_1 \to D$.
(2) If we generate Y_i uniformly in D_1 we compute $X_i = \phi(Y_i)$.
(3) Compute $\Psi(X_i)$ and determine the minimum if we solve a minimization problem, respectively maximum if we solve a maximization problem.

In our paper we solve a minimization problem, hence we compute the minimum.

Suppose that the theoretical minimum of Ψ is m^*, and $m = m^* + \varepsilon^2$ is the minimum obtained by Monte Carlo methods after $nrsim$ simulations. Then the probabilistic error is

$$P\left(\Psi(X) < m^* + \varepsilon^2\right) = G\left(m^* + \varepsilon^2\right), \tag{1.1}$$

where G is the cumulative distribution function of $\Psi(X)$.

Because G is given by a geometrical probability, we obtain

$$G\left(m^* + \varepsilon^2\right) = \frac{V_\varepsilon}{V}, \tag{1.2}$$

where V_ε is the volume of sub-domain of D given by $\Psi(X) \leq m^* + \varepsilon^2$ and V is the volume of D. While V is a constant of the Monte Carlo model, V_ε tends to zero when $\varepsilon \to 0$.

In [4] it is shown how the minimum eigenvalue of a symmetric positive definite matrix can be computed using the Monte Carlo methods. We have simulated normal vectors with zero expectation and the matrix as variance-covariance matrix, and points on the unit sphere as orthogonal regression coefficients. The minimum eigenvalue is the minimum variance of errors.

A special case is when matrix A is a Toeplitz matrix. This is the case of Yule—Walker algorithm for determining the coefficients of $AR(p)$ time series [3, 7].

If the linear system arrises from linear regression the multicolinearity can be detected [6]. It means that the we have a strong dependence between explanatory variables and the determinant of matrix is close to zero. One of the method to detect multicolinearity is the BKW=Belsley—Kuh—Welsch test [1], which detects the ill conditioned linear system. We compute the square root of the condition number, the condition index

$$CI = \sqrt{\frac{\lambda_n}{\lambda_1}}. \tag{1.3}$$

The above condition number is defined in [8] as

$$\kappa(A) = \|A\| \cdot \|A^{-1}\|, \tag{1.4}$$

where A is the matrix of linear system.

If CI is between 10 and 30 we have moderate multicolinearity, while for $CI > 30$ we have severe multicolinearity.

Let X_t, $t = \overline{1,n}$ be a time series. This time series is stationary if the expectation $E(X_t)$ and the variance $Var(X_t)$ do not depend on t, and the covariance $\gamma_k = Cov(X_t, X_{t-k})$ does not depend on t.

Definition 1.2. The function $\gamma : \mathbb{N}^* \to \mathbb{R}$, $\gamma(k) = \gamma_k$ defined above is called the autocovariance function.

The autocorrelation function is the Pearson correlation between X_t and X_{t-k}.

In the case of stationary time series the autocorrelation function is defined by:

$$\rho_k = \frac{\gamma_k}{\gamma_0}, \tag{1.5}$$

where $\gamma_0 = Var(X_t)$ is the variance of time series.

A test for the stationarity of a time series is the Dickey–Fuller unit root test [5]. We determine the linear regressions

$$\Delta X_t = \Phi X_{t-1} + a_t \text{ for model 1,} \tag{1.6}$$

$$\Delta X_t = \beta + \Phi X_{t-1} + a_t \text{ for model 2,} \tag{1.6'}$$

$$\Delta X_t = \beta + \Phi X_{t-1} + \gamma t + a_t \text{ for model 3,} \tag{1.6"}$$

where a_t is the white noise (error). Therefore a_t is a zero expectation time series that does not depend on X_t.

The three above models check the dependence of X_t (current value) on the previous value, X_{t-1}. Φ non significant in model 1 means that the difference between the current and the previous value of X is the error, hence X_t is random walk. This model is tested when we know that X_t has constant zero expectation. For instance, if we want to test if a time series has the

same variance (volatility in finance) we determine the (eventually not constant) expectation of X_t and we subtract it. If the new time series is stationary the variance is constant.

Two methods to stationarize time series are presented in [3, 7, 10]: the differentiation method and the moving average method. The first one consists in computing $\Delta X_t = X_t - X_{t-1}$ first for X_t, next for ΔX_t until we obtain stationarity.

The second method consists in computing the moving average of order q

$$M_t = \frac{\sum_{i=t-q}^{t+q} X_i}{2q+1}, \tag{1.7}$$

and $X_t - M_t$ is stationary.

If in the model 3 Φ is not significant we determine the linear regression without Φ: ΔX_t in terms of t.

If Φ and γ are significant the time series has a trend, which is removed by moving average method or by exponential smooth method [10]. If both are not significant we study the other two models.

If Φ is not significant the time series is not stationary and we stationarize it by differentiation. Otherwise the time series is stationary with zero/ nonzero expectation if β from model 2 is non significant/ significant.

For a stationary time series one model is the autoregressive model $AR(p)$ [2, 3, 10]

$$X_t = \sum_{i=1}^{p} \phi_i X_{t-i} + a_t, \tag{1.8}$$

where a_t is independent identically distributed time series (noise). White noise means zero expectation for X and a.

If L is the lag operator ($LX_t = X_{t-1}$), we denote by

$$\phi(L) = 1 - \sum_{i=1}^{p} \phi_i L^i, \tag{1.9}$$

and formula (1.8) becomes

$$\phi(L) X_t = a_t. \tag{1.8'}$$

Another model for stationary time series is [3, 10] the moving average model $MA(q)$

$$X_t = \theta(L) a_t = a_t - \sum_{i=1}^{q} \theta_i a_{t-i}. \tag{1.10}$$

A mixture between the two above models is the $ARMA(p,q)$ time series model

$$X_t = \sum_{i=1}^{p} \phi_i X_{t-i} + a_t - \sum_{i=1}^{q} \theta_i a_{t-i}, \text{ or} \tag{1.11}$$

$$\phi(L) X_t = \theta(L) a_t. \tag{1.11'}$$

In the stationary case $\phi(L)$ and $\theta(L)$ have roots greater than one in absolute values.

The $AR(p)$ model is solved by the Yule—Walker algorithm [3, 7, 10]. We solve a linear system with the symmetric matrix given by the variance on diagonal and $A_{ij} = \gamma_{|i-j|}$ for $i \neq j$ (autocovariance function). The right side is given by the values of the autocovariance function

with lags between one and p. We have also the version of autocorrelation function, dividing by the variance γ_0. The key of this algorithm is to write a_t from (1.8′) and to take into account that the covariance between a_t and X_{t-i}, $i = \overline{1,p}$ is zero (current error does not depend on past values of time series).

For the $ARMA(p,q)$ model we apply the Hannan—Rissanen algorithm [3, 7, 10]. We solve the $AR(m)$ model with $m > p$ and $m > q$. After we compute the white noise b_t of the $AR(m)$ model, we perform the linear regression of X_t in terms of X_{t-1},..., X_{t-p} and b_{t-1}, ..., b_{t-q} without intercept, or zero intercept. The coefficients are ϕ_i and $-\theta_j$.

2. Methodology

If the matrix of the linear system, A, is symmetric and positive definite, then it is diagonalizable and the eigenvalues are positive. Given U the orthogonal matrix of the corresponding eigenvectors as columns, we have

$$D = U'AU, \tag{2.1}$$

hence

$$A = UDU', \tag{2.1'}$$

where $D = diag(\lambda_1, \lambda_2, ..., \lambda_n)$, and U' is the transposed matrix of U.

The linear system $Ax = b$ can be written as

$$UDU'x = b, \ or \tag{2.2}$$

$$Dy = c, \tag{2.2'}$$

where $y = U'x$ and $c = U'b$.

Because U is a rotation, the Euclidean norm is the same:

$$\begin{cases} \|y\| = \|x\| \\ \|c\| = \|b\|. \end{cases} \tag{2.3}$$

Solving the linear system (2.2′) we obtain

$$y_i = \frac{c_i}{\lambda_i}, \tag{2.4}$$

where λ_i are the eigenvalues of A. It follows that

$$\|y\| \leq \frac{\|c\|}{\lambda_1}, \tag{2.5}$$

where λ_1 is the minimum eigenvalue.

Therefore we generate feasible solutions in the cube that contains the sphere with center zero and radius $\frac{\|c\|}{\lambda_1}$, and we compute the minimum of the function from Theorem 1.1.

Remark 2.1. We can substitute the above radius $\frac{\|c\|}{\lambda_1}$ by any value $\frac{\|c\|}{\widetilde{\lambda}}$ such that $\widetilde{\lambda} \leq \lambda_1$.

In the same manner we conclude that

$$\|y\| \geq \frac{\|c\|}{\lambda_n}, \tag{2.5'}$$

where λ_n is the maximum eigenvalue. In this way we can simulate the feasible solution on the spherical shell

$$\frac{\|b\|}{\lambda_n} \leq \|x\| \leq \frac{\|b\|}{\lambda_1}. \tag{2.5"}$$

In [4] the minimum eigenvalue is computed as the minimum of sum of squares of distances between the n points and the hyper-plane through the center of gravity (the origin in the mentioned paper).

We can simulate feasible solutions inside the sphere of radius $R = \frac{\|b\|}{\lambda_1}$ according to (2.5), or in the spherical shell of radii $r = \frac{\|b\|}{\lambda_n}$ and $R = \frac{\|b\|}{\lambda_1}$ according (2.5"). To simplify the simulation of the feasible solutions we can replace the sphere of radius $\frac{\|b\|}{\lambda_1}$ by its circumscribed cube, and the sphere of radius $\frac{\|b\|}{\lambda_n}$ by its inscribed cube.

Therefore we have defined four domains D_i on which we simulate the possible solutions. D_1 is the hypercube $[-R, R]^n$, D_2 is the hypersphere of radius R, D_3 is the region between the hypercubes $[-R, R]^n$ and $\left[-\frac{r}{\sqrt{n}}, \frac{r}{\sqrt{n}}\right]^n$ and D_4 is the spherical shell of radii r and R. For each $i = \overline{1,4}$, we denote by V_i the volume of the domain D_i. We have

$$V_1 = 2^n R^n, \tag{2.6}$$

$$V_2 = \frac{2\pi^{\frac{n}{2}} R^n}{n\Gamma\left(\frac{n}{2}\right)}, \tag{2.7}$$

$$V_3 = 2^n \left(R^n - \frac{r^n}{n^{\frac{n}{2}}}\right), \tag{2.8}$$

$$V_4 = \frac{2\pi^{\frac{n}{2}}}{n\Gamma\left(\frac{n}{2}\right)} \left(R^n - r^n\right). \tag{2.9}$$

We compute now the volume V_ε of the region $F(x) \leq F(x^*) + \frac{\varepsilon^2}{2}$.

The frontier of this domain is

$$\langle x, Ax\rangle - 2\langle x, b\rangle = -\langle x^*, b\rangle + \varepsilon^2. \tag{2.10}$$

Passing the right side to left, we obtain the hyper-quadric given by the matrix of the system A for the first n rows and columns, b_i on row i column $n+1$, and $\langle x^*, b\rangle + \varepsilon^2$ on row and column $n+1$. Because if $\varepsilon = 0$ the quadric is degenerated into the solution of the system, x^*, the invariants are the Viète sums of the characteristic polynomial of A, including $\delta = \det(A) > 0$. The last invariant is $\Delta = -\delta\varepsilon^2$. Therefore the hyper-quadric is a hyper-ellipsoid of canonical form

$$\sum_{i=1}^{n} \lambda_i \left(X_i'\right)^2 = \varepsilon^2, \tag{2.11}$$

where λ_i are the eiganvalues of A. The semi-axes are

$$a_i = \frac{\varepsilon}{\sqrt{\lambda_i}}. \tag{2.12}$$

Of course, from the above formula a_1 is the longest semi-axis and a_n the shortest one.

The volume V_ε is the volume of hyper-ellipsoid:

$$V_\varepsilon = \frac{2\pi^{\frac{n}{2}}}{n\Gamma\left(\frac{n}{2}\right)} \cdot \frac{\varepsilon^n}{\sqrt{\prod_{i=1}^{n} \lambda_i}} = \frac{2\pi^{\frac{n}{2}}}{n\Gamma\left(\frac{n}{2}\right)} \cdot \frac{\varepsilon^n}{\sqrt{\det A}}, \tag{2.13}$$

i.e. the volume of unit hyper-sphere $V_s = \frac{2\pi^{\frac{n}{2}}}{n\Gamma\left(\frac{n}{2}\right)}$ multiplied by the semi-axes of the hyper-ellipsoid.

We conclude that the probability of error is

$$P\left(F(X) \leq \min F + \frac{\varepsilon^2}{2}\right) = \frac{V_\varepsilon}{V_i} = \begin{cases} \frac{V_s}{2^n R^n} \cdot \frac{\varepsilon^n}{\sqrt{\det A}} \text{ for hyper-cube} \\ \frac{1}{R^n} \cdot \frac{\varepsilon^n}{\sqrt{\det A}} \text{ for hyper-sphere} \\ \frac{V_s}{2^n\left(R^n - \frac{r^n}{n^{\frac{n}{2}}}\right)} \cdot \frac{\varepsilon^n}{\sqrt{\det A}} \text{ for region between cubes} \\ \frac{1}{R^n - r^n} \cdot \frac{\varepsilon^n}{\sqrt{\det A}} \text{ for spherical shell.} \end{cases} \tag{2.14}$$

If A is diagonally dominant, we can consider $\widetilde{\lambda}_1$ as the minimum difference between the absolute value of the diagonal element A_{ii} and the sum of absolute values of the other elements in the corresponding line:

$$\widetilde{\lambda}_1 = \min_{i=\overline{1,n}} \left\{ |A_{ii}| - \sum_{j \neq i} |A_{ij}| \right\}. \tag{2.15}$$

The maximum eigenvalue is less than

$$\widetilde{\lambda}_n = Tr(A) - (n-1)\,\widetilde{\lambda}_1, \tag{2.15'}$$

where $Tr(A)$ is the trace of matrix A.

We compute the characteristic polynomial

$$P(\lambda) = \lambda^n - S_{n-1}\lambda^{n-1} + S_{n-2}\lambda^{n-2} + ... + (-1)^{n-1} S_1 \lambda + (-1)^n S_0, \tag{2.16}$$

where S_i are the sums of Viète. These sums are computed as in classical manner:

(1) S_{n-1} is the trace of matrix.
(2) S_0 is the determinant of the matrix.
(3) The other sums, S_{n-j} for $j = \overline{2, n-1}$, are computed as sums of principal minors of order j.

We obtain

$$\widetilde{\lambda_1} = \frac{S_0}{S_1}, \tag{2.17}$$

and

$$\widetilde{\lambda_n} = S_{n-1} - (n-1)\,\widetilde{\lambda_1}. \tag{2.18}$$

If we simulate the feasible solutions in the spherical shell or in the region between the inscribed cube in the small sphere, with radius $\frac{\|b\|}{\lambda_n}$ and the circumscribed cube of big sphere, with radius $\frac{\|b\|}{\lambda_1}$, the ratio between the volume of this region and the entire sphere/ cube is

$$ratio = 1 - \left(\frac{\lambda_1}{\lambda_n}\right)^n. \tag{2.19}$$

Therefore if the multicolinearity is found by the BKW test, we have no improvement from cube/ sphere case to "between cubes"/ spherical shell: we subtract a very small number from one. Due to the exponent n it follows that for the same ratio of eigenvalues (hence the same condition number) the above improvement decreases with the dimension of the system n.

For $ARMA(p,q)$ time series, after we have solved the $AR(m)$ model with $m > p,q$ we compute the white noise. Next we solve the linear regression of X_t in terms of $X_{t-i},\ i=\overline{1,p}$ and $a_{t-j},\ j=\overline{1,q}$, with intercept zero. We obtain a linear regression with $p+q$ coefficients. The involved linear system has $p+q$ lines and $p+q$ columns. The first p lines/ columns are given by autocovariance function with lags between 0 and $p-1$. When we compute the covariance between X_{t-i} and a_{t-j} we have value zero for $i>j$ (in Yule—Walker algorithm these values are set to zero). For $i<j$ we compute the covariance between X_t and $a_{t+i-j}=a_{t-k}$, where $k=j-i$. We obtain

$$Cov(X_t, a_{t-k}) = \gamma_{t-k} - \sum_{i=1}^{m} \phi_i \gamma_{t-k-i} = \phi(L)\gamma_{t-k}. \tag{2.20}$$

Dividing by γ_0 (variance) we obtain

$$\frac{Cov(X_t, a_{t-k})}{\gamma_0} = \rho_{t-k} - \sum_{i=1}^{m} \phi_i \rho_{t-k-i} = \phi(L)\rho_{t-k}, \tag{2.20'}$$

i.e. we replace in (2.20) the autocovariance function by autocorrelation function.

Therefore in the above matrix of $ARMA(p,q)$ time series in the last q rows and first p columns we have an L matrix: only the covariance between current X and past errors are not zero. Symmetrically, in the first p rows and last q columns we have the transpose of L.

We notice that (2.20) is true also for $k=0$, as the variance of a_t is the covariance between a_t and $\phi(L)X_t$, and from the last one we maintain X_t. The right side in (2.20) is the variance of errors in $AR(m)$ model [3, 6, 10]. Dividing by the variance of X we obtain

$$\frac{Var(a_t)}{Var(X_t)} = 1 - \sum_{i=1}^{m} \phi_i \rho_i = 1 - \langle \phi, \rho \rangle \tag{2.21}$$

When we compute the last q lines/ columns of the matrix we have the variance of errors on the diagonal if we use (2.20), and $\frac{Var(a_t)}{Var(X_t)}$ if we use (2.20′).

3. Applications

Example 3.1. Solve the linear system

$$\begin{cases} 9x+3y-z=2 \\ 3x+7y+2z=1 \\ -x+2y+8z=3 \end{cases}.$$

The solution of above system obtained with Excel is $x=0.31034$, $y=-0.11671$ and $z=0.44297$. With simple relaxation method for given maximal error $\max|r_i|=0.001$ we obtain after 18 steps $x=0.31025$, $y=-0.11664$ and $z=0.44287$. The real error is 0.000587.

The matrix is symmetric, diagonally dominant, and has positive values on the main diagonal, hence it is positive definite.

Using the over-relaxation method with $\omega=1.07977$ we obtain in the same conditions as above $x=0.31028$, $y=-0.11665$ and $z=0.44287$

The eigenvalues of A are $\lambda_1 = 3.70017$, $\lambda_2 = 9.0852$ and $\lambda_3 = 11.21463$. The condition index is $CI = 1.74093$.

With our $C++$ program we obtain the results presented in Table 1

Table 1. Results for the system in Example 3.1 by Monte Carlo methods

Values method	Monte Carlo	Diagonal dominance	Viète
λ_1	3.63408	2	2.12994
λ_3	11.29358	20	19.74011
Max radius	1.0296	1.87083	1.7567
Min radius	0.3313	0.18708	0.18955
Solution on cube	$\begin{pmatrix} 0.304 \\ -0.10175 \\ 0.43433 \end{pmatrix}$	$\begin{pmatrix} 0.31302 \\ -0.10637 \\ 0.46313 \end{pmatrix}$	$\begin{pmatrix} 0.29159 \\ -0.10104 \\ 0.41714 \end{pmatrix}$
Solution on sphere	$\begin{pmatrix} 0.31338 \\ -0.11431 \\ 0.45005 \end{pmatrix}$	$\begin{pmatrix} 0.32619 \\ -0.10968 \\ 0.43419 \end{pmatrix}$	$\begin{pmatrix} 0.29627 \\ -0.11431 \\ 0.45005 \end{pmatrix}$
Solution between cubes	$\begin{pmatrix} 0.32674 \\ -0.16568 \\ 0.45521 \end{pmatrix}$	$\begin{pmatrix} 0.30922 \\ -0.12265 \\ 0.45948 \end{pmatrix}$	$\begin{pmatrix} 0.34273 \\ -0.19388 \\ 0.44636 \end{pmatrix}$
Solution on spherical shell	$\begin{pmatrix} 0.30255 \\ -0.1154 \\ 0.44722 \end{pmatrix}$	$\begin{pmatrix} 0.30056 \\ -0.13112 \\ 0.43377 \end{pmatrix}$	$\begin{pmatrix} 0.32217 \\ -0.12088 \\ 0.46151 \end{pmatrix}$
Minimum F on cube	-0.91578	-0.91397	-0.91351
Minimum F on sphere	-0.91615	-0.91307	-0.91562
Minimum F between cubes	-0.91005	-0.91538	-0.89897
Minimum F on spherical shell	-0.91608	-0.91435	-0.91498

We have simulated 10000 normal vectors with variance-covariance matrix A, 10000 points on unit sphere and 1 million feasible solutions. The possible solutions are simulated inside the following 4 regions: sphere with radius $\frac{\sqrt{14}}{\lambda_1}$; the circumscribed cube of this sphere; between the spheres of radius $\frac{\sqrt{14}}{\lambda_3}$ and $\frac{\sqrt{14}}{\lambda_1}$ (spherical shell); between the inscribed cube in the sphere of radius $\frac{\sqrt{14}}{\lambda_3}$ and the circumscribed cube of the sphere of radius $\frac{\sqrt{14}}{\lambda_1}$.

Because $F_{\min} = -0.9164456233$ we obtain the following values of ε and of probabilistic errors.

Table 2. Errors for the Monte Carlo methods in Example 3.1

Model\ method for λ_i	Monte Carlo	Diagonal dominance	Viète
Cube	0.0586 $1.34012 \cdot 10^{-6}$	0.30181 $2.34542 \cdot 10^{-7}$	0.13203 $1.48068 \cdot 10^{-5}$
Sphere	0.04443 $1.43568 \cdot 10^{-6}$	0.30492 $7.92164 \cdot 10^{-7}$	0.04993 $9.82342 \cdot 10^{-7}$
Between cubes	0.16393 $8.21156 \cdot 10^{-5}$	0.48284 $1.53622 \cdot 10^{-6}$	0.13074 $1.42365 \cdot 10^{-5}$
Spherical shell	0.05323 $2.97918 \cdot 10^{-6}$	0.29667 $7.09898 \cdot 10^{-7}$	0.06827 $3.43854 \cdot 10^{-6}$

Example 3.2. Determine the linear regression of the dynamics of bank deposits Y in terms of the dynamics of income X_1 and the dynamics of passive interest rate X_2 [6].

The matrix of linear system is
$A = \begin{pmatrix} 1 & 1.068 & 3.984 \\ 1.068 & 2.1636 & 4.2696 \\ 3.984 & 4.2696 & 15.8984 \end{pmatrix}$,
and the right side is
$b = \begin{pmatrix} 0.452 \\ 1.27 \\ 1.8344 \end{pmatrix}$.
The linear regression obtained in [6] is:

$$Y = A_0 + A_1X_1 + A_2X_2 = -3.7869 + 0.7572X_1 + 0.861X_2.$$

The matrix is not diagonally dominant, hence we can not apply the over-relaxation method. We can neither apply (2.15) for estimating λ_1.

The eigenvalues of A are $\lambda_1 = 0.00154$, $\lambda_2 = 0.94901$ and $\lambda_3 = 18.11144$. The condition index is $CI = 108.32511$, i.e. severe multicolinearity.

If we use in the same conditions as in Example 3.1 the simple relaxation method (maximum error 0.001) we obtain after 2243 steps $A_0 = -3.17723$, $A_1 = 0.75911$ and $A_2 = 0.70771$. The real error is 0.000999.

With successive relaxation method we obtain after 3684 steps $A_0 = -3.21965$, $A_1 = 0.75928$ and $A_2 = 0.71829$. We have the same real error, 0.000999.

If we apply the Monte Carlo method to obtain the minimum/ maximum eigenvalue simulating 10000 normal vectors and 10000 sets of coefficients, we obtain $\lambda_1 = 0.00189$ and $\lambda_3 = 19.0873$. They are close to the above real λ_1 and λ_3 obtained with Scilab. But if we simulate 1 million feasible solutions we obtain strange results: $A_0 = 53.58246$, $A_1 = 1.44435$ and $A_2 = -13.63316$. The minimum of F is 2.490955 instead of -0.4147.

Analogously, if we estimate the values λ_1 and λ_3 using Viète sums we obtain $\lambda_1 = 0.00154$ and $\lambda_3 = 57.186$. Even if the estimation of λ_1 is quite accurate, we still obtain strange results: $A_0 = -48.92698$, $A_1 = 2.52771$ and $A_2 = 13.80325$. The minimum of F is 38.266.

If we simulate 1 billion feasible solutions we obtain $A_0 = -18.16524$, $A_1 = -0.06271$ and $A_2 = 4.6121$. The minimum of F is 0.11586.

Example 3.3. Consider [13] the CPI in the period June 2019 - March 2024 quarterly data. For the stationarized time series determine the $AR(4)$ coefficients. Using the $AR(4)$ model apply the Hannan—Rissanen algorithm to obtain the $ARMA(2,1)$ coefficients.

According to Dickey—Fuller unit root test we obtain
$\Delta X_t = 1.037 - 0.05667X_{t-1} - 0.04647t$
with Student statistics 1.25116, -0.72172 and -1.0193 in the model 3,
$\Delta X_t = 0.46814 - 0.07207X_{t-1}$
with Student statistics 1.25116 and -0.72172 in the model 2, respectively
$\Delta X_t = -0.02284X_{t-1}$
with Student statistics -0.54584 in the model 1. Because Φ is not significant, in all three models the series is not stationary.

For $Y_t = \Delta X_t$ we have the following results of Dickey—Fuller test:

$\Delta Y_t = 0.43992 - 0.4586 Y_{t-1} - 0.03094t$
with Student statistics 0.66745, -2.57806 and -0.73412 in the model 3,
$\Delta Y_t = 0.00715 - 0.42611 Y_{t-1}$
with Student statistics 0.02453 and -2.49835 in the model 2, respectively
$\Delta Y_t = -0.42614 Y_{t-1}$
with Student statistics -2.55224 in the model 1. In this case Φ is only 10% significant, but not 5% significant, since the quantiles for 10%, 5% and 1% for the three models and $n = 26$ are -3.2367 for 10%, -3.6027 for 5% and -4.3738 for 1% in the case of model 3, -2.6318 for 10%, -2.985 for 5% and -3.7204 for 1% in the case of model 2, and -1.6228 for 10%, -1.9552 for 5% and -2.6603 for 1% in the case of model 1. We notice that only for model 1 we have 5% significance for Φ.

For $Z_t = \Delta^2 X_t$ we have the following results of Dickey—Fuller test:
$\Delta Z_t = -0.06517 - 1.3641 Z_{t-1} + 0.06572$
with Student statistics -0.09168, -6.76196 and 0.0657 in the model 3,
$\Delta Z_t = -0.02359 - 1.36445 Z_{t-1}$
with Student statistics -0.07447 and -6.92436 in the model 2, respectively
$\Delta Z_t = -7.0794 Z_{t-1}$
with Student statistics -2.55224 in the model 1. In this case Φ is significant 1%. Therefore the time series is $I(2)$.

For Z_t we have the first four values of autocorrelation function -0.3642514, 0.207902, 0.0875773 and -0.5423844. Therefore the matrix of the linear system involved in Yule—Walker algorithm is

$$\begin{pmatrix} 1 & 0.36425 & 0.2079 & 0.08758 \\ 0.36425 & 1 & 0.36425 & 0.2079 \\ 0.2079 & 0.36425 & 1 & 0.36425 \\ 0.08758 & 0.2079 & 0.36425 & 1 \end{pmatrix},$$

and the right side is

$$\begin{pmatrix} 0.36425 \\ 0.2079 \\ 0.08758 \\ -0.54238 \end{pmatrix}.$$

Using Cholesky method, the $AR(4)$ model is

$$Z_t = -0.22749 Z_{t-1} + 0.25002 Z_{t-2} + 0.01926 Z_{t-3} - 0.56743 Z_{t-4} + a_t.$$

If we apply the simple relaxation method we obtain after 13 steps $\phi_1 = -0.22712$, $\phi_2 = 0.24996$, $\phi_3 = 0.01895$ and $\phi_4 = -0.56799$. The real error is 0.000431.

With over-relaxation method we obtain after 21 steps $\phi_1 = -0.22716$, $\phi_2 = 0.25067$, $\phi_3 = 0.0186$ and $\phi_4 = -0.5679$. The real error is 0.000672.

The eigenvalues obtained with Scilab are 0.5232, 0.58692, 1.1364 and 1.75348. The condition index is $CI = 1.8307$, hence we do not have multicolinearity.

Using our $C++$ program we obtain the results presented in Table 3.

Table 3. Results for the system in Example 3.3 by the Monte Carlo methods

Values method	Monte Carlo	Diagonal dominance	Viète
λ_1	0.50642	0.0636	0.4098
λ_4	1.77132	3.80921	2.12207
Max radius	1.36486	10.8678	1.88668
Min radius	0.39022	0.18179	0.32572
Solution on cube	$\begin{pmatrix} -0.22759 \\ 0.25296 \\ -0.02341 \\ -0.60106 \end{pmatrix}$	$\begin{pmatrix} -0.25584 \\ 0.52044 \\ 0.20577 \\ -0.62388 \end{pmatrix}$	$\begin{pmatrix} -0.25584 \\ 0.20675 \\ 0.03205 \\ -0.5677 \end{pmatrix}$
Solution between cubes	$\begin{pmatrix} -0.25818 \\ 0.29728 \\ 0.20044 \\ -0.55125 \end{pmatrix}$	$\begin{pmatrix} -0.10517 \\ 0.77946 \\ 0.23969 \\ -0.71702 \end{pmatrix}$	$\begin{pmatrix} -0.28582 \\ 0.24033 \\ -0.02917 \\ -0.53114 \end{pmatrix}$
Solution on spherical shell	$\begin{pmatrix} -0.24708 \\ 0.21128 \\ 0.05147 \\ -0.52949 \end{pmatrix}$	$\begin{pmatrix} -0.23266 \\ 0.18107 \\ 0.30162 \\ -0.51323 \end{pmatrix}$	$\begin{pmatrix} -0.28508 \\ 0.18478 \\ -0.02124 \\ -0.54977 \end{pmatrix}$
Roots of $P(\rho)$ on cube	$\rho_{1,2}$=−0.73155±0.54688i $\rho_{3,4}$=0.61776±0.58211i $\lvert\rho_{1,2}\rvert$=0.91337 $\lvert\rho_{3,4}\rvert$=0.84881	$\rho_{1,2}$=−0.73163±0.57918i $\rho_{3,4}$=0.7202±0.44476i $\lvert\rho_{1,2}\rvert$=0.93313 $\lvert\rho_{3,4}\rvert$=0.84646	$\rho_{1,2}$=−0.74279±0.56251i $\rho_{3,4}$=0.58458±0.5505i $\lvert\rho_{1,2}\rvert$=0.93175 $\lvert\rho_{3,4}\rvert$=0.80298
Roots of $P(\rho)$ on sphere	$\rho_{1,2}$=−0.73436±0.56096i $\rho_{3,4}$=0.60429±0.54664i $\lvert\rho_{1,2}\rvert$=0.9241 $\lvert\rho_{3,4}\rvert$=0.81485	$\rho_{1,2}$=−0.75836±0.46737i $\rho_{3,4}$=0.65373±0.44718i $\lvert\rho_{1,2}\rvert$=0.8908 $\lvert\rho_{3,4}\rvert$=0.79204	$\rho_{1,2}$=−0.70728±0.56645i $\rho_{3,4}$=0.60322±0.54094i $\lvert\rho_{1,2}\rvert$=0.90615 $\lvert\rho_{3,4}\rvert$=0.81024
Roots of $P(\rho)$ between cubes	$\rho_{1,2}$=−0.74046±0.59571i $\rho_{3,4}$=0.61137±0.4864i $\lvert\rho_{1,2}\rvert$=0.95035 $\lvert\rho_{3,4}\rvert$=0.78126	$\rho_{1,2}$=−0.81506±0.54155i $\rho_{3,4}$=0.76247±0.40916i $\lvert\rho_{1,2}\rvert$=0.97857 $\lvert\rho_{3,4}\rvert$=0.86532	$\rho_{1,2}$=−0.73899±0.54513i $\rho_{3,4}$=0.62953±0.58056i $\lvert\rho_{1,2}\rvert$=0.9183 $\lvert\rho_{3,4}\rvert$=0.85637
Roots of $P(\rho)$ on spherical shell	$\rho_{1,2}$=−0.71231±0.56137i $\rho_{3,4}$=0.58877±0.54506i $\lvert\rho_{1,2}\rvert$=0.90693 $\lvert\rho_{3,4}\rvert$=0.80233	$\rho_{1,2}$=−0.70497±0.65231i $\rho_{3,4}$=0.58864±0.45809i $\lvert\rho_{1,2}\rvert$=0.96047 $\lvert\rho_{3,4}\rvert$=0.74589	$\rho_{1,2}$=−0.73113±0.53253 $\rho_{3,4}$=0.67478±0.59428 $\lvert\rho_{1,2}\rvert$=0.90451 $\lvert\rho_{3,4}\rvert$=0.89916
Minimum F on cube	−0.22116	−0.1766	−0.22043
Minimum F on sphere	−0.22116	−0.17566	−0.2209
Minimum F between cubes	−0.20871	−0.10558	−0.2736
Minimum F on spherical shell	−0.22073	−0.17814	−0.21982

We present in Table 3 the roots of the polynomial

$$P(\rho) = \rho^p - \sum_{i=1}^{p} \phi_i \rho^{p-i},$$

and their absolute values. In the stationary case, the absolute values must be less than 1.

We note that the roots of $P(\rho)$ obtained for the $AR(p)$ model using the mentioned Yule—Walker algorithm but solving the linear system by Cholesky method are $-0.72327 \pm 0.55261i$ and $0.60953 \pm 0.55979i$, with absolute values 0.91022 and 0.82758.

The minimum of F is -0.22215.

In Table 4, we present the results of Hannan—Rissanen algorithm. We solve the linear systems of Yule—Walker algorithm and that of linear regression of Hannan—Rissanen algorithm by the same method.

Table 4. Linear regression in ϕ_i and $-\theta_j$ in Hannan—Rissanen algorithm

Method	Extended matrix	Simple relaxation	Successive relaxation	Solution Monte Carlo
Monte Carlo, cube	$\begin{pmatrix} 1 & -0.36425 & 0.54055 & -0.36425 \\ -0.36425 & 1 & 0 & 0.2079 \\ 0.54055 & 0 & 0.54055 & -0.09612 \end{pmatrix}$	$\begin{pmatrix} -0.58523 \\ -0.00464 \\ 0.40569 \end{pmatrix}$ 21 steps	$\begin{pmatrix} -0.58537 \\ -0.00471 \\ 0.40589 \end{pmatrix}$ 39 steps	$\begin{pmatrix} -0.59732 \\ -0.01289 \\ 0.45017 \end{pmatrix}$ $F_{min} = -0.08653$
Monte Carlo, sphere	$\begin{pmatrix} 1 & -0.36425 & 0.53927 & -0.36425 \\ -0.36425 & 1 & 0 & 0.2079 \\ 0.53927 & 0 & 0.54055 & -0.06065 \end{pmatrix}$	$\begin{pmatrix} -0.69099 \\ -0.04379 \\ 0.57722 \end{pmatrix}$ 30 steps	$\begin{pmatrix} -0.69197 \\ -0.04368 \\ 0.5782 \end{pmatrix}$ 42 steps	$\begin{pmatrix} -0.64208 \\ -0.07857 \\ 0.42488 \end{pmatrix}$ $F_{min} = -0.09937$
Monte Carlo, between cubes	$\begin{pmatrix} 1 & -0.36425 & 0.52761 & -0.36425 \\ -0.36425 & 1 & 0 & 0.2079 \\ 0.52761 & 0 & 0.52761 & -0.00659 \end{pmatrix}$	$\begin{pmatrix} -0.86495 \\ -0.10716 \\ 0.87604 \end{pmatrix}$ 33 steps	$\begin{pmatrix} -0.8661 \\ -0.10709 \\ 0.87724 \end{pmatrix}$ 42 steps	$\begin{pmatrix} -0.84691 \\ -0.15953 \\ 0.89687 \end{pmatrix}$ $F_{min} = -0.14764$
Monte Carlo, spherical shell	$\begin{pmatrix} 1 & -0.36425 & 0.57438 & -0.36425 \\ -0.36425 & 1 & 0 & 0.2079 \\ 0.57438 & 0 & 0.57438 & -0.07825 \end{pmatrix}$	$\begin{pmatrix} -0.71421 \\ -0.05225 \\ 0.57686 \end{pmatrix}$ 35 steps	$\begin{pmatrix} -0.71385 \\ -0.05152 \\ 0.57598 \end{pmatrix}$ 45 steps	$\begin{pmatrix} -0.7545 \\ -0.47765 \\ 0. \end{pmatrix}$ $F_{min} = -0.1019$
Viète, cube	$\begin{pmatrix} 1 & -0.36425 & 0.52957 & -0.36425 \\ -0.36425 & 1 & 0 & 0.2079 \\ 0.52957 & 0 & 0.52957 & -0.04899 \end{pmatrix}$	$\begin{pmatrix} -0.70576 \\ -0.04917 \\ 0.61199 \end{pmatrix}$ 30 steps	$\begin{pmatrix} -0.70587 \\ -0.04859 \\ 0.61166 \end{pmatrix}$ 39 steps	$\begin{pmatrix} -0.69828 \\ -0.08227 \\ 0.56606 \end{pmatrix}$ $F_{min} = -0.10774$
Viète, sphere	$\begin{pmatrix} 1 & -0.36425 & 0.57879 & -0.36425 \\ -0.36425 & 1 & 0 & 0.2079 \\ 0.57879 & 0 & 0.57879 & -0.07813 \end{pmatrix}$	$\begin{pmatrix} -0.7251 \\ -0.05622 \\ 0.58886 \end{pmatrix}$ 35 steps	$\begin{pmatrix} -0.72604 \\ -0.05609 \\ 0.58977 \end{pmatrix}$ 48 steps	$\begin{pmatrix} -0.75846 \\ -0.09557 \\ 0.56605 \end{pmatrix}$ $F_{min} = -0.10214$
Viète, between cubes	$\begin{pmatrix} 1 & -0.36425 & 0.52768 & -0.36425 \\ -0.36425 & 1 & 0 & 0.2079 \\ 0.52768 & 0 & 0.52768 & -0.09259 \end{pmatrix}$	$\begin{pmatrix} -0.57389 \\ -0.00114 \\ 0.39728 \end{pmatrix}$ 18 steps	$\begin{pmatrix} -0.57424 \\ -0.00078 \\ 0.39743 \end{pmatrix}$ 39 steps	$\begin{pmatrix} -0.58721 \\ -0.02601 \\ 0.48842 \end{pmatrix}$ $F_{min} = -0.08448$
Viète, spherical shell	$\begin{pmatrix} 1 & -0.36425 & 0.53491 & -0.36425 \\ -0.36425 & 1 & 0 & 0.2079 \\ 0.53491 & 0 & 0.53491 & -0.12912 \end{pmatrix}$	$\begin{pmatrix} -0.47628 \\ -0.03442 \\ 0.23371 \end{pmatrix}$ 27 steps	$\begin{pmatrix} -0.47702 \\ -0.0346 \\ 0.23438 \end{pmatrix}$ 39 steps	$\begin{pmatrix} -0.46449 \\ -0.06434 \\ 0.21147 \end{pmatrix}$ $F_{min} = -0.07501$

4. Conclusions

In our $C++$ program to estimate the minimum eigenvalue by Monte Carlo methods we have estimated 10000 random vectors with variance-covariance matrix A and 10000 possible coefficients of orthogonal regression on unit sphere.

In the first example we notice similar results for diagonal dominance and Viète sums cases. This is because the estimated λ_1 and λ_3 are similar: 2 and 20 in diagonal dominance case, and 2.12994 and 19.74011 in the last case. In the Monte Carlo case of estimating λ_1 and λ_3 we have results closer to real values $\lambda_1 = 3.63498$ (versus 3.70017), respectively $\lambda_3 = 11.29358$ (versus 11.21463). That's why ε and V_1 are smaller in Monte Carlo case. But the region where we search the solution is larger in the other cases, and the denominator increases too. For this reason we find no differences in probabilities of errors.

We notice that when we have severe multicolinearity the relaxation methods (if available - for instance if we do not have diagonal dominance as in Example 3.2 we can not apply over-relaxation method) need many steps for a given error. With Monte Carlo methods we obtain strange results due to the large radius of the sphere, 1221.82922 using real λ_1 (with Scilab) and 1107.59869 if we estimate $\lambda_1 = 0.00205$ by Monte Carlo methods, respectively 1477.42737 if we estimate $\lambda_1 = 0.00154$ by Viète sums.

When by some methods we obtain larger CI with some methods to estimate λ_1 and λ_n, we obtain some biased results. Over-relaxation method can require more steps as the simple relaxation method, as in Example 3.3. In this case $CI = 7.7394$ in the case of diagonal dominance, when $\lambda_1 = 0.0636$ and $\lambda_4 = 3.8092$ (see Table 3). Hence $\left(\frac{\lambda_1}{\lambda_4}\right)^4 = 7.7 \cdot 10^{-8}$, so we can see by (2.19) that there is no significant improvement if we use the spherical shell instead of sphere, or the region between cubes instead of cube.

When we simulate one billion feasible solutions we obtain reasonable but not close solution. In the Monte Carlo estimation of $\lambda_1 = 0.00162$ we obtain $F_{min} = 0.11585$, but the run time is over 3 hours.

In the case of diagonal dominance we have used for λ_n the formula (2.15′). An open problem is to estimate λ_n more effective (smaller than using above formula).

References

[1] D.A. Belsley, E. Kuh and R.E. Welsch, *The Condition Number. Regression Diagnostics: Identifying Influential Data and the Sources of Collinearity*, John Wiley & Sons, New York, 1980.

[2] G.E.P. Box, G.M. Jenkins and G.C. Reinsel, *Time Series Analysis: Forecasting and Control* (3rd ed.), Prentice Hall, Englewood Cliffs, 1994.

[3] P.J. Brockwell and R.A. Davis, *Introduction to Time Series and Forecasting* (3rd ed.), Springer, 2016.

[4] C. Costinescu and D. Ciuiu, *The Monte Carlo Methods to find eigenvalues and eigenvectors*, in Proceedings of International Conference Trends and Challenges in Applied Mathematics, June 20-23 2007, TUCB, 157-160.

[5] D. Dickey and W. Fuller, *Likelihood ratio statistics for autoregressive time series with a unit root*, Econometrica **44 (4)** (1981), 1057-1072.

[6] N. Jula and D. Jula, *Economic Modeling: Econometric and Optimization Models* (in Romanian), Mustang, Bucharest, 2010.

[7] N.M. Jula and D. Jula, *Economic Forecast* (in Romanian), Mustang, Bucharest, 2015.

[8] Q. Louveaux, *Introduction to Numerical Analysis*, Université de Liège Publishing, 2015.

[9] G. Păltineanu, P. Matei and G.D. Mateescu, *Numerical Analysis* (in Romanian), Conspress, Bucharest 2010.

[10] T. Popescu, *Time Series: Applications in Systems Analysis* (in Romanian), Technical Publishing House, Bucharest, 2000.

[11] E. Süli and D. Mayers, *An Introduction to Numerical Analysis*, Cambridge University Press, 2003.

[12] I. Văduva, *Simulation Models* (in Romanian), Bucharest University Printing House, 2004.

[13] *Consumer price index (CPI) inflation rate*, Interactive Database of the National Bank of Romania, www.bnr.ro (accessed on April 10, 2024).

Department of Mathematics and Computer Science, Technical University of Civil Engineering Bucharest, Bucharest, Romania

Center for Mathematical Modeling, National Institute for Economic Research "Costin C. Kiriţescu", Romanian academy

Email address: dciuiu@yahoo.com