# Parallel computing for efficient calculation of multidimensional Bernstein copulas in modeling nonlinear dependence between random variables

# Cómputo paralelo para el cálculo eficiente de cópulas de Bernstein multidimensionales en el modelado de dependencia no lineal entre variables aleatorias

Hernández-Maldonado, Victor Miguel*[a], Erdely, Arturo [b] and Diaz-Viera, Martin Alberto [c]

[a] ROR Dirección Adjunta de Innovación y Conocimiento (DAIC) • 0000-0002-9306-8535 • 174514
[b] ROR Universidad Nacional Autónoma de México • 0000-0003-1653-8342 • 1011993
[c] ROR Instituto Mexicano del Petróleo (IMP) Ciudad de México. CDMX • 0000-0001-5811-6186 • 464993

**CONAHCYT classification:**

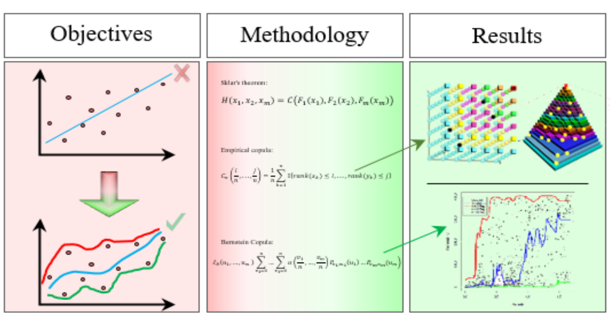
Area: Physics-Mathematics and Earth Sciences
Field: Mathematics
Discipline: Computer sciences
Subdiscipline: Code and coding systems

https://doi.org/10.35429/EJRC.2024.10.18.1.11
**History of the article:**
Received: January 22, 2024
Accepted: December 10, 2024

* ✉ [vmhernann@gmail.com]

**Abstract**

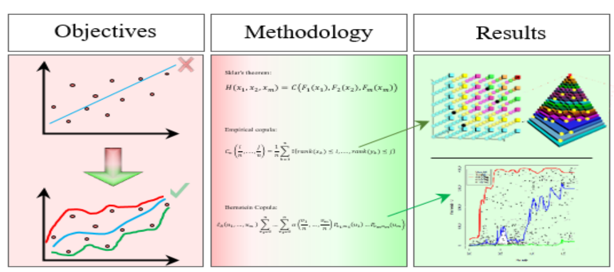
Copulas are a versatile tool for modeling the dependence structure between random variables. By defining marginal distributions, copulas can capture complex joint distributions that are often beyond the scope of traditional statistical methods, which typically rely on linearity and normality assumptions. Unlike these methods, copulas are marginal-free and can effectively model nonlinear dependencies. A Bernstein copula is an empirical, data-driven model capable of reproducing intricate relationships between variables. While highly effective for real-world data, computing Bernstein copulas becomes computationally demanding in higher dimensions. In an $m-dimensional$ case with a sample size of $(n)$, the computation requires evaluating an $n^m$ grid of points, which leads to significant resource demands in terms of processing time and memory as $(m)$ and $(n)$ increase. In this paper, we propose an efficient method for implementing multidimensional Bernstein copulas. We introduce both an optimized algorithm for calculating a multidimensional empirical copula and a parallelized approach for computing the Bernstein copula.




**Empirical and Bernstein copula; Multidimensional dependence; parallel computing**

**Resumen**

Las cópulas son una herramienta versátil para modelar la estructura de dependencia entre variables aleatorias. Al definir distribuciones marginales, las cópulas pueden capturar distribuciones conjuntas complejas que a menudo están fuera del alcance de los métodos estadísticos tradicionales, los cuales suelen basarse en suposiciones de linealidad y normalidad. A diferencia de estos métodos, las cópulas no dependen de las marginales y pueden modelar eficazmente dependencias no lineales. Una cópula de Bernstein es un modelo empírico, basado en datos, capaz de reproducir relaciones intrincadas entre variables. Aunque son muy eficaces para datos del mundo real, el cálculo de las cópulas de Bernstein se vuelve computacionalmente exigente en dimensiones más altas. En un caso de $m-dimensiones$ con un tamaño de muestra de $(n)$, el cálculo requiere evaluar una cuadrícula de puntos de $n^m$, lo que genera una demanda significativa de recursos en términos de tiempo de procesamiento y memoria a medida que $(m)$ y $(n)$ aumentan. En este artículo, proponemos un método eficiente para implementar cópulas de Bernstein multidimensionales. Introducimos tanto un algoritmo optimizado para calcular una cópula empírica multidimensional como un enfoque paralelizado para calcular la cópula de Bernstein.




**Cópula empírica y de Bernstein; Dependencia multidimensional; computación paralela**

**Citation:** Hernández-Maldonado, Victor Miguel, Erdely, Arturo and Diaz-Viera, Martin Alberto. [2024]. Parallel computing for efficient calculation of multidimensional Bernstein copulas in modeling nonlinear dependence between random variables. ECORFAN-Journal Republic of Cameroon. 10[18]1-11: e61018111.

**Introduction to multivariate copulas**

Linear regression-based dependence models often fail to capture complex dependence structures, which can result in the underestimation of variance and standard deviation.

This limitation compromises their ability to reproduce the inherent variability of the data, which is critical to understanding the nature of many problems. While these models are appropriate when the joint behaviour of variables adheres to linearity assumptions, nonlinear dependencies among random variables are frequently encountered.

As an alternative, copula functions offer a robust method for modelling the joint distribution of random variables. The core principle of the copula approach lies in expressing the joint distribution of random variables as a function of their marginal distributions. This allows copulas to efficiently capture and model complex dependencies among variables [I, II].

According to Sklar's theorem [III], the underlying copula associated to a multivariate random vector $(X_1, X_2, \ldots, X_m)$ represents a functional link between the joint probability distribution and the univariate marginal distributions $(F_1, F_2, \ldots, F_m)$ respectively, Eq.(1):

$$H(x_1, x_2, x_m) = C\left(F_1(x_1), F_2(x_2), F_m(x_m)\right) \quad (1)$$

For all $(X_1, X_2, \ldots, X_m)$ in the extended real numbers system, where $C: [0,1]^m \to [0,1]$ the underlying copula is unique whenever $(X_1, X_2, \ldots X_m)$ are continuous random variables. Therefore, all the information about the dependence between continuous random variables is contained in their corresponding copula. Several properties may be derived for copulas [II] and among them we have an immediate corollary from Sklar's theorem: $(X_1, X_2, \ldots X_m)$ are independent continuous random variables if and only if their underlying copula is $C(u_1, \ldots, u_n) = (u_1, \ldots, u_n)$.

Let $S = \{(x_{11}, x_{21}, \ldots, x_{m1}), \ldots, (x_{1n}, x_{2n}, x_{mn})$ be $n$ observations of a random vector $(X_1, X_2, \ldots, X_m)$. We may obtain empirical estimates for the marginal distributions $(X_1, X_2, \ldots X_m)$ by means of, [IV]:

$$\widehat{F}_j(x) = \frac{1}{n}\sum_{k=1}^{n} \mathbb{I}_{[-\infty, x_i]}(X_{j,i}) \quad (2)$$

Where $\mathbb{I}$ stands for an indicator function which takes value 1 whenever its argument is true, and 0 otherwise. It is well-known [V] that the empirical distribution $\widehat{F}_j$ is a consistent estimator of $\widehat{F}_j$ that is, $\widehat{F}_j$ converges almost surely to $\widehat{F}_j$ as $n \to \infty$ for all $t$.

We address the issue of examining or characterizing the dependence characteristics of multivariate distributions using a series of observed data points. The multivariate empirical copula is formally defined in Equation (3) as referenced by [VI].

$$C_n(\boldsymbol{u}) = \frac{1}{n}\sum_{k=1}^{n}\prod_{j=1}^{d} \mathbb{I}\left\{\frac{R_{i,j}{}^{n}}{n} \leq u_j\right\}, u = (u_1, u_d)\epsilon[0,1]^d \quad (3)$$

where $n$ is the size of the sample, and $\mathbb{I}$ stands for an indicator function which takes value 1 whenever its argument is true, and 0 otherwise.

The empirical copula is a function $C_n$ with domain $\left\{\frac{1}{n}: i = 0,1, \ldots, n\right\}^m$ and its convergence to the true copula $C$ has also been proved by [VII] The empirical copula is not a copula, since it is only defined on a finite grid, not in the whole unit hypercube $[0,1]^m$ but by Sklar's Theorem [III] it may be extended to a copula.

Sklar's theorem is completely general and a joint distribution function can be constructed using a copula function. The copula separates the marginal distributions from correlation, and the copula itself can capture the dependence structure. This is an essential property of copulas.

From Sklar's theorem (1) each random variable $X_m$ is modeled as an absolutely continuous random variable with unknown marginal distribution function $F_m$.

For simulation of continuous random variables, the use of the empirical distribution function (2) is not appropriate since $\widehat{F}_j$ is a step function, and therefore discontinuous, so a smoothing technique is needed.

ISSN: 2414-4959.
RENIECYT-CONAHCYT: 1702902


Hernández-Maldonado, Victor Miguel, Erdely, Arturo and Diaz-Viera, Martin Alberto. [2024]. Parallel computing for efficient calculation of multidimensional Bernstein copulas in modeling nonlinear dependence between random variables. ECORFAN-Journal Republic of Cameroon. 10[18]1-11: e61018111.
https://doi.org/10.35429/EJRC.2024.10.18.1.11

Since the objective of using copulas is to simulate a primary variable using one or more descriptive variables, it is necessary to have a smooth estimation of marginal quantile function $Q(u) = inf\{x: F(x) \geq u\}, 0 \leq u \leq 1$ which is possible by means of Bernstein polynomials as in Muñoz-Pérez and Fernández-Palacín (1987).

$$\hat{Q}_n(u) = \sum_{k=1}^{n} \frac{1}{2}(x_k + x_{k+1}) \binom{n}{k} u^k (1-u)^{n-k} \quad (4)$$

For a smooth estimation of the underlying copula we make use of the Bernstein copula Eq. (5) [VIII], [IX]:

$$\hat{c}_B(u_1, \dots, u_m) \sum_{v_1=0}^{n} \dots \sum_{v_m=0}^{n} \alpha\left(\frac{v_1}{n}, \dots, \frac{v_m}{n}\right) P_{v_1,m_1}(u_1) \dots P_{v_m,n_m}(u_m) \quad (5)$$

where:

$$P_{v_1,m_1}(u_1) = \binom{k_j}{v_j} u_j^{v_j} (1-u_j)^{k_j - v_j} \quad (6)$$

For every $(u_1, \dots, u_m)$ in the unit hypercube $[0,1]^m$ and $\alpha\left(\frac{v_1}{n}, \dots, \frac{v_m}{n}\right)$ is the empirical copula, defined in (3) [VIII].

**Bivariate and trivariate sampling algorithms**

For a pair of random variables $(X_1, X_2)$ with joint distribution function $H$ and underlying copula $C$ we need to generate an observation of uniform $(0,1)$ random variables $(U, V)$ whose joint distribution function is $C$ and then transform those uniform variables as in step 3 of the sampling bivariate algorithm. For generating such pair $(u, v)$ it is used a conditional distribution method, this method needs the conditional distribution function for $V$ given $U = u$, which we denote as $C_u(v)$

$$C_u(v) = \frac{\partial \tilde{C}_B(u,v)}{\partial u} \quad (7)$$

where $\tilde{C}_B$ is the bivariate Bernstein copula model, obtained by (5).

To simulate replications from the random vector $(X_1, X_2)$ with the dependence structure estimated from the observed data, $S := \{(x_{11}, x_{21}), \dots, (x_{1n}, x_{2n})\}$ it is applied the following algorithm:

**Sampling bivariate algorithm**

Generate two independent and continuous Uniform $(0, 1)$ random variables $u$ and $t$.

1. Set $v = C_u^{-1}(t)$ where $C_u$ is defined in *(7)*

2. The desired pair is $(x_1, x_2) = \tilde{Q}_n(u), \tilde{R}_n(u)$ , where $\tilde{Q}_n$and $\tilde{R}_n$, according to (4), are the estimated and smoothed quantile functions of and , respectively.

For the multivariate case we must solve equations that represent conditional distribution functions for $W$ given $U = u,\ V = v$

To simulate replications from the random vector $X_1, X_2, X_3$ with dependence structure estimated from data $S := \{(x_{11}, x_{21}, x_{31}), \dots, (x_{1n}, x_{2n}, x_{3n})\}$ it is applied the next algorithm.

**Sampling trivariate algorithm**

1. Generate three independent and continuous Uniform $(0, 1)$ random variables $u$, $t_1$ and $t_2$.

2. Set $v = C_u^{-1}(t_1)$ where $C_u$ is defined in *(7)*.

3. Set $w = C_{uv}^{-1}(t_2)$ where $w = c\_uv(W)$

$$C_{uv}(w) = \frac{\frac{\partial \tilde{C}_B(u,v,w)}{\partial u \partial v}}{\frac{\partial \tilde{C}_B(u,v,1)}{\partial u \partial v}} \quad (8)$$

Where $\tilde{C}_B$ is the trivariate Bernstein copula model *(5)*.

4. The desired vector is $(x_1, x_2, x_3) = \tilde{Q}_n(u), \tilde{R}_n(v), \tilde{H}_n(w)$, where $\tilde{Q}_n(u), \tilde{R}_n(v)$ and $\tilde{H}_n(w)$, according to (4), are the estimated and smoothed quantile functions of $X_1$ , $X_2$ and $X_3$, respectively.

**Method**

The Bernstein copula is a function based on empirical distributions that can reproduce the underlying dependence structure between random variables in a data-driven way.

ISSN: 2414-4959.
RENIECYT-CONAHCYT: 1702902


Hernández-Maldonado, Victor Miguel, Erdely, Arturo and Diaz-Viera, Martin Alberto. [2024]. Parallel computing for efficient calculation of multidimensional Bernstein copulas in modeling nonlinear dependence between random variables. ECORFAN-Journal Republic of Cameroon. 10[18]1-11: e61018111.
https://doi.org/10.35429/EJRC.2024.10.18.1.11

However its computational performance can be very demanding in terms of processing time and storage capacity. In this proposal it is presented a 2-stage algorithm to improve the implementation of its use.

In the first step it is computed a multidimensional empirical copula using an efficient procedure and in the second step they are generated replications of the Bernstein copula using high performance computational techniques.

**An efficient procedure to compute a multidimensional empirical copula**

Standard calculations of the empirical copula based on equation (3) can end up in an extremely low performance.

In the $m - dimensional$ case with a sample size $n$, this calculation implies dealing with an m-dimensional grid with a total of $n^m$ points, which for moderate values of $m$ and $n$ it demands an important amount of resources in terms of processing time and storage capacity.

For example, building a *3-dimensional copula* ($m = 3$) with a sample size $n = 1000$, it would require an efficient management of a data structure which represents a discrete hypercube with $10^9$ elements.

For the bivariate case, the compute of the empirical copula $C_n$ basically consist into count the number of bivariate points $(x, y)$ observed in a unitary grid. In Figure 1 (Left), it is observed a scatter plot of 15 observations of two random variables.

The graphical representation of the distribution of these points into an empirical copula is observed in Figure 2 (Right).

To generate this distribution, the observations are sorted by primary variable and then each one is mapped into the unitary grid. It is important to note that in this distribution it is observed one and only one point for each vertical and horizontal line into the grid.

**Box 1**

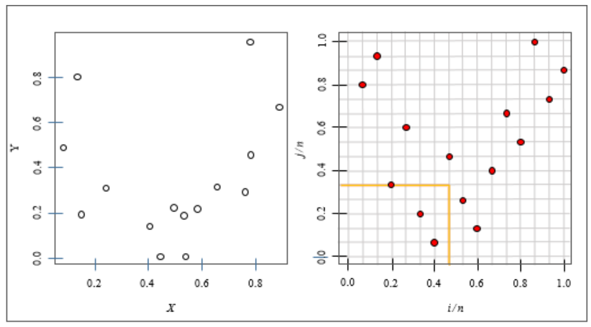


**Figure 1**

Scatterplot and unitary grid
(Left) Scatter plot of bivariate empirical observations. (Right) Unitary grid of the same observations.

Computing the entire empirical copula consists in calculate the value of each node of the unitary grid of the Figure 1 (Right), using equation (9).

$$C_n\left(\frac{i}{n},\ldots,\frac{j}{n}\right) = \frac{1}{n}\sum_{k=1}^{n} \mathbb{I}\{rank(x_k) \le i,\ldots, rank(y_k) \le j\} \quad (9)$$

Despite (9) represents a systematic calculation, it implies some restrictions, for example, each node always increases its values from 0 to $(observations/n)$ from down to up, from left to right and only it is taken into account the observation that are enclosed into given 'region' of the grid (orange rectangle, Figure 1. (Right).

We are going to take a set of 5 data values of three random variables which are shown in Table 1.

**Box 2**

**Table 1**

A set of tree variables and 5 data values. Two secondary variables (SV1, SV2) and one primary variable (PV)

| $SV_1$ | $SV_2$ | PV |
|---|---|---|
| 1.1 | 2.11 | 1.111 |
| 2.1 | 1.11 | 1.111 |
| 3.1 | 4.11 | 5.111 |
| 4.1 | 3.11 | 2.111 |
| 5.1 | 5.11 | 4.111 |

*Source: Microsoft Word.*

To construct the Empirical Copula and the Bernstein Copula, we implemented the procedure proposed in [X, XI], which provides a quick and effective method for constructing these copulas, particularly in two dimensions.

ISSN: 2414-4959.
RENIECYT-CONAHCYT: 1702902

However, in this study, we extend that approach to a multivariate context, specifically to three dimensions. Let's construct the 2D empirical copula, which is exhaustively explained in fragments of code written in ***C++*** which will be found in the section: *"Appendix A"* of this document.

**Steps of the first stage of this proposal**

1. The computational function that calculates the empirical copula in 2D has the $C{+}{+}$ form shown in (10). It transforms the double data set matrix into integer values, sort the values of the matrix by the main variable and mainly maintain its dependence structure.

The variable 'Rows' is the number of rows of the data set. 'Cols' is the number of variables, in this example we have 3 variables, but it is necessary to say that, for the 2D case only they are considered 2 variables, let us say, 'SV1' and 'SV2'. 'SV1' as secondary variable and SV2 as primary variable, 'VarToSort' is the variable which the sort process will be based on.

2. This matrix is sorted by the $SV_2$ variable, we do this because we are solving a 2D copula, since the idea that secondary variable is $SV_1$ and primary variable is *'SV2'*. In fact, it can be any variable but is necessary to think that in the $2D$ case we have only one secondary variable and one primary variable.

As proposed here, the sort function always sorts variables by the primary variable ($VarToSort = SV_2$), see (11). After those process, (IntMaker and SORT) the arrData matrix end up having the form presented in Table 2. Note the array shown in Table 1 now is sorted by $\boldsymbol{SV_2}$ and converted to integer values.

**Box 3**

**Table 2**

It is shown the **arrData** Matrix after **IntMakerandSORT** process. Note the array shown in 1 now is sorted by 'SV2' and converted to integer values

| $SV_1$ | $SV_2$ | PV |
|---|---|---|
| 2 | 1 | 1 |
| 1 | 2 | 3 |
| 4 | 3 | 2 |
| 3 | 4 | 5 |
| 5 | 5 | 4 |

*Source: Microsoft Word*

3. Next, in (12) it is created the matrix that will contain the 2D empirical copula ("*copemp2D*"). Let us carefully consider the following: equation (10) presents the prototype of the function that calculates the 2D empirical copula, while equation (12) only presents the declaration of the matrix that contains the 2D empirical copula, which is already included within the function in (10). In equation (13), *copemp2D* matrix is populated with zero values.

4. Then in the 2D empirical copula matrix is filled with zero values, see (14). Up to this point, the copemp2D matrix has the appearance of Figure 3. where blue cells indicate a value of *0.0* within them.

**Box 4**

| | | | | | | |
|---|---|---|---|---|---|---|
| 5 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 4 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 3 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 2 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 1 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| | 0 | 1 | 2 | 3 | 4 | 5 |

**Figure 3**

Empirical copula matrix with zero entries

Current state of the 2D empirical copula matrix

5. The calculation of the discrete values of the empirical copula is performed from this point. We know that the primary variable is sorted in ascending mode, so we can take its values and those of the secondary variable given by the current value of 'j' variable of the first 'for' loop (15). Knowing this, we take the value of the secondary variable to locate where the point of the current propagation is (16).

Given this scenario we can set the values of the empirical copula into the copem2D matrix, (17). A propagation must reach the final point of the copem2D Matrix (i.e. When its subscripts are equal to Rows 'copemp2D[Rows][Rows]'), then we have to propagate the influence of the presence of a point in both directions, in 'x' (17) and 'y' direction (18). Schematically all this process be shown in Figure 4.




Hernández-Maldonado, Victor Miguel, Erdely, Arturo and Diaz-Viera, Martin Alberto. [2024]. Parallel computing for efficient calculation of multidimensional Bernstein copulas in modeling nonlinear dependence between random variables. ECORFAN-Journal Republic of Cameroon. 10[18]1-11: e61018111.
https://doi.org/10.35429/EJRC.2024.10.18.1.11

## Box 5

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | j | 5 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 5 | 0.0 | 0.0 | 0.2 | 0.2 | 0.2 | 0.2 |
| | | 4 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 4 | 0.0 | 0.0 | 0.2 | 0.2 | 0.2 | 0.2 |
| | | 3 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 3 | 0.0 | 0.0 | 0.2 | 0.2 | 0.2 | 0.2 |
| | | 2 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 2 | 0.0 | 0.0 | 0.2 | 0.2 | 0.2 | 0.2 |
| | | 1 | 0.0 | 0.0 | 0.2 | 0.2 | 0.2 | 0.2 | 1 | 0.0 | 0.0 | 0.2 | 0.2 | 0.2 | 0.2 |
| | | 0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 2 | j | 5 | 0.0 | 0.0 | 0.2 | 0.2 | 0.2 | 0.2 | 5 | 0.0 | 0.2 | 0.4 | 0.4 | 0.4 | 0.4 |
| | | 4 | 0.0 | 0.0 | 0.2 | 0.2 | 0.2 | 0.2 | 4 | 0.0 | 0.2 | 0.4 | 0.4 | 0.4 | 0.4 |
| | | 3 | 0.0 | 0.0 | 0.2 | 0.2 | 0.2 | 0.2 | 3 | 0.0 | 0.2 | 0.4 | 0.4 | 0.4 | 0.4 |
| | | 2 | 0.0 | 0.2 | 0.4 | 0.4 | 0.4 | 0.4 | 2 | 0.0 | 0.2 | 0.4 | 0.4 | 0.4 | 0.4 |
| | | 1 | 0.0 | 0.0 | 0.2 | 0.2 | 0.2 | 0.2 | 1 | 0.0 | 0.0 | 0.2 | 0.2 | 0.2 | 0.2 |
| | | 0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 3 | j | 5 | 0.0 | 0.2 | 0.4 | 0.4 | 0.4 | 0.4 | 5 | 0.0 | 0.2 | 0.4 | 0.4 | 0.6 | 0.6 |
| | | 4 | 0.0 | 0.2 | 0.4 | 0.4 | 0.4 | 0.4 | 4 | 0.0 | 0.2 | 0.4 | 0.4 | 0.6 | 0.6 |
| | | 3 | 0.0 | 0.2 | 0.4 | 0.4 | 0.6 | 0.6 | 3 | 0.0 | 0.2 | 0.4 | 0.4 | 0.6 | 0.6 |
| | | 2 | 0.0 | 0.2 | 0.4 | 0.4 | 0.4 | 0.4 | 2 | 0.0 | 0.2 | 0.4 | 0.4 | 0.4 | 0.4 |
| | | 1 | 0.0 | 0.0 | 0.2 | 0.2 | 0.2 | 0.2 | 1 | 0.0 | 0.0 | 0.2 | 0.2 | 0.2 | 0.2 |
| | | 0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 4 | j | 5 | 0.0 | 0.2 | 0.4 | 0.4 | 0.6 | 0.6 | 5 | 0.0 | 0.2 | 0.4 | 0.4 | 0.8 | 0.8 |
| | | 4 | 0.0 | 0.2 | 0.4 | 0.6 | 0.8 | 0.8 | 4 | 0.0 | 0.2 | 0.4 | 0.6 | 0.8 | 0.8 |
| | | 3 | 0.0 | 0.2 | 0.4 | 0.4 | 0.6 | 0.6 | 3 | 0.0 | 0.2 | 0.4 | 0.4 | 0.6 | 0.6 |
| | | 2 | 0.0 | 0.2 | 0.4 | 0.4 | 0.4 | 0.4 | 2 | 0.0 | 0.2 | 0.4 | 0.4 | 0.4 | 0.4 |
| | | 1 | 0.0 | 0.0 | 0.2 | 0.2 | 0.2 | 0.2 | 1 | 0.0 | 0.0 | 0.2 | 0.2 | 0.2 | 0.2 |
| | | 0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 5 | j | 5 | | | | 0.0 | 0.2 | 0.4 | 5 | 0.4 | 0.8 | 1.0 | | | |
| | | 4 | | | | 0.0 | 0.2 | 0.4 | 4 | 0.6 | 0.8 | 0.8 | | | |
| | | 3 | | | | 0.0 | 0.2 | 0.4 | 3 | 0.4 | 0.6 | 0.6 | | | |
| | | 2 | | | | 0.0 | 0.2 | 0.4 | 2 | 0.4 | 0.4 | 0.4 | | | |
| | | 1 | | | | 0.0 | 0.0 | 0.2 | 1 | 0.2 | 0.2 | 0.2 | | | |
| | | 0 | | | | 0.0 | 0.0 | 0.0 | 0 | 0.0 | 0.0 | 0.0 | | | |
| | | | 0 | 1 | 2 | 3 | 4 | 5 | | 0 | 1 | 2 | 3 | 4 | 5 |
| | | | i | | | | | | | i | | | | | |

**Figure 4**

Empirical copula, complete process

Graphical representation of the complete process of the generation of the 2D Empirical Copula proposed in [X]. The propagation points are found in white font colour and black background

For the three-dimensional Empirical Copula generation, it is used the same process, but is augmented one more subscript ($k$) to the *copemp*3*d*[*k*][*j*][*i*] matrix, in order to include the compute of the third dimension, or fourth, fifth, etc.

Figure 5 provides a three-dimensional depiction, where discrete and incrementally ascending steps within the empirical copula are visible for a given dataset, reaching the final value of 1.00.

## Box 6

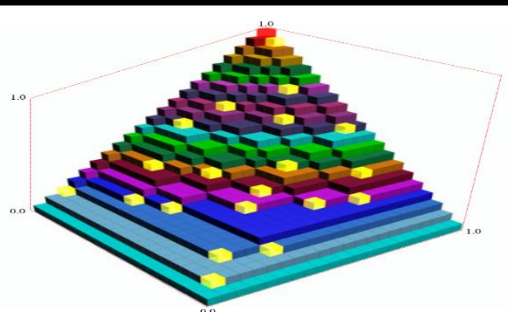



**Figure 5**

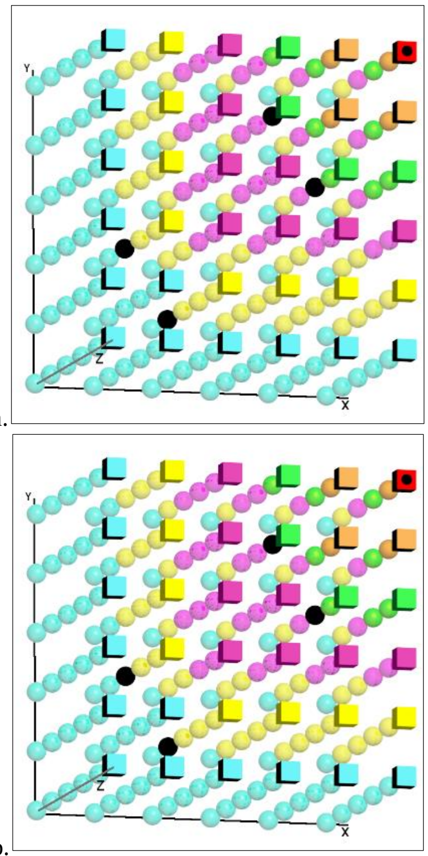
A 3D empirical copula

A three-dimensional depiction of the empirical copula

In Figure 6, the 3D Empirical Copula is displayed in a perspective view, as generated by the proposed process.

The propagation points are represented by black cubes, while the propagated points are shown in cubes of various colors, reflecting different values (observations/n).

## Box 7



a.

b.

**Figure 6**

A 3D empirical copula

3D Empirical Copula generated using the proposed process "*a. Perspective view with spheres*"; "*b. Perspective view with cubes.*" In both views, propagation points are represented by black spheres in (a) and black cubes in (b), while propagated points are shown in spheres and cubes of various colors (a/b)

### Parallel computing of a 3D Bernstein copula

Regression and Simulation processes are made in this proposal, a brief explanation of them will be discussed.

It is created a matrix (19) and (20) where results of regression process will be collected. It is a two-sub indexed array where original data set and quantile regressions are collocated.

ISSN: 2414-4959.
RENIECYT-CONAHCYT: 1702902


Hernández-Maldonado, Victor Miguel, Erdely, Arturo and Diaz-Viera, Martin Alberto. [2024]. Parallel computing for efficient calculation of multidimensional Bernstein copulas in modeling nonlinear dependence between random variables. ECORFAN-Journal Republic of Cameroon. 10[18]1-11: e61018111.
https://doi.org/10.35429/EJRC.2024.10.18.1.11

It is decided the next calculations will be done in parallel mode, see (21). Immediately it is called a main method to perform regressions of a given quantile, where are passed as argument, the data set, the current value of the quantile regression, the sorted data set, the secondary variable 1, the secondary variable 2, the primary variable, the *2D* empirical copula, the *3D* Empiric Copula, the number of rows and finally the results precision are passed as argument, see (22). Inside this function the Bivariate and trivariate sampling algorithms are solved. Note that (22) must be inside (21).

Finally, it is computed and written an output message to indicate to the user in which step of the process it is, see (23).

**Results**

In Petrophysics, assessment of formation permeability is a complex and challenging problem that plays a key role in reservoir forecasts and optimal reservoir management[XI]. In heterogeneous carbonate reservoirs, permeability evaluation is commonly performed using permeability-porosity relationships, which often seem to be nonlinear and complex.

Copulas are marginal-free dependence functions that may capture such nonlinear relationships. In the present work we make use of a nonparametric copula approach for bivariate and trivariate modelling of permeability, porosity, and VP Meas real data. A *2D* and *3D* copula function can reproduce complex joint distributions that others statistical techniques cannot, because in many cases these techniques are usually based on linear assumptions [XII].

Next it is presented a set of Petrophysical variables modeled by a *2D* copula model. In Figure 7 are plotted dataset values, *3* quantile Regressions (*0.1*, *0.5* and *0.9*) and *380* simulations. Note that Dataset values and simulated ones are enclosed in the regression bands (*quantile=0.1* and *0.9*), which, preliminary, it is a symptom of good estimation of the *2D* Bernstein Copula.

However, there are some points that are outside of these bands, the question is: How can we improve the estimation bands here?

Answers may appear like change the quantile estimation values or introducing more descriptive variables or take more values in the data set. In this work we will explore the use of more descriptive values and discuss about its convenience and inconvenience of its use.

**Box 8**

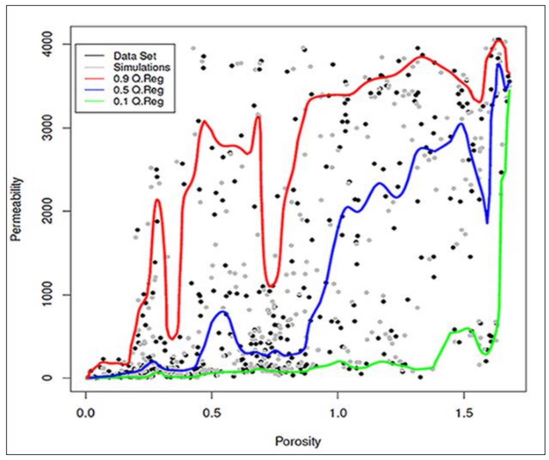



**Figure 7**
Quantile Regressions by 2D Copula model
Dataset, 3 quantile Regressions (0.1, 0.5 and 0.9) and 380 Simulations by Bernstein Copula 2D
*This figure was made in R-project software*

Figures 7 and 8 present distinct datasets. Figure 8 plots dataset values along with three quantile regressions (0.1, 0.5, and 0.9) and 380 simulations of the 3D Bernstein Copula. While both figures display similar data, it is evident that the values in Figure 8 are more effectively contained within the quantile regression bands. This indicates that incorporating an additional variable into the Bernstein Copula enhances its capacity to estimate or simulate values.

**Box 9**

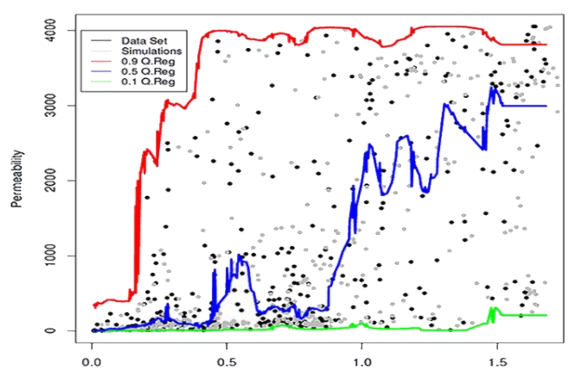



**Figure 8**
Quantile Regressions by 3D Copula model
Dataset, 3 quantile Regressions (0.1, 0.5 and 0.9) and 380 Simulations by 3D Bernstein Copula
*This figure was made in R-project software*




Hernández-Maldonado, Victor Miguel, Erdely, Arturo and Diaz-Viera, Martin Alberto. [2024]. Parallel computing for efficient calculation of multidimensional Bernstein copulas in modeling nonlinear dependence between random variables. ECORFAN-Journal Republic of Cameroon. 10[18]1-11: e61018111.
https://doi.org/10.35429/EJRC.2024.10.18.1.11

Figure 9 displays 3,800 simulations of the same empirical dataset using a 2D Bernstein Copula. It is notable that many of the simulated values fall outside the regression bands, which is an expected outcome. In contrast, Figure 10 presents 3,800 simulations generated with a 3D Bernstein Copula. Here, the simulations are more effectively contained within the regression bands, consistently surrounding the true data values.

**Box 10**

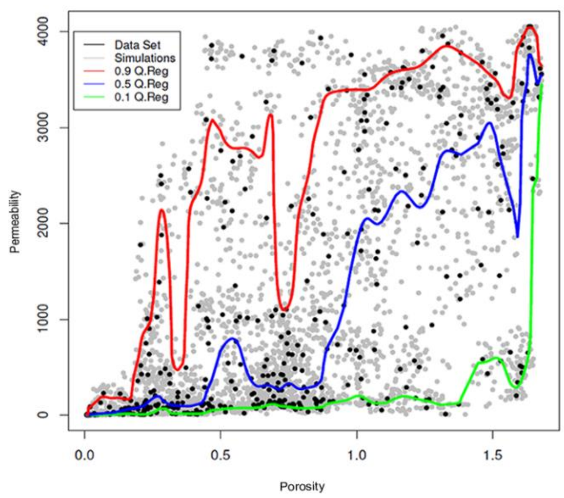



**Figure 9**

Ten 2D simulations scaled in size by a factor of 10.

Dataset, 3 Quantile Regressions and 3800 simulations by 2D Bernstein Copula. This figure was made in R-project software

*This figure was made in R-project software*

**Box 11**

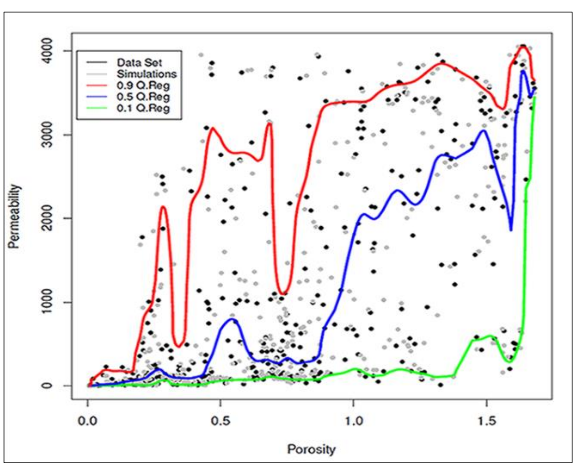



**Figure 10**

10 times simulation in size

Dataset, 3 Quantile Regressions and 3800 simulations by Bernstein Copula 3D

*This figure was made in R-project software*

Computing a multidimensional Bernstein Copula can end up in a very demanding task in computational terms, because the empirical copula has to be visited several times to generate a single result inside of the Bernstein copula, Considering the size of these matrices (the empirical copula and Bernstein copula), it is proposed to use parallel techniques. Here we use these techniques to implement more variables into the calculus.

It is used a current personal computer with Windows 10 Pro OS, with 8 processors in hardware and a speed of 2.7 GHz. It is decided to do so, because these kinds of computers are capable to perform this kind of tasks and they are also available for almost every person. In Figure 11 it is presented the performance of the computer when a multidimensional regression process is not run in parallel mode.

Note that it is not reached the full capacity of the computer, in fact, it is just used the 23\% of its capacity. In Figure 12 and Figure 13, things change drastically, it is performed the copula process in parallel mode and the speed and uses of the computer is used almost in its full capacity.

**Box 12**

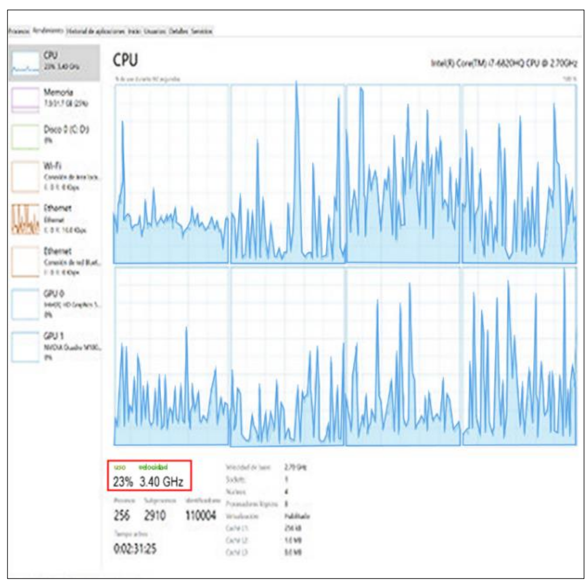



**Figure 11**

10 times simulation in size

Windows 10 Task Manager when Non Parallel process is performed of Bernstein copula either Regressions or Simulations

*This figure is a Windows 10 Pro Task Manager Screenshot*

**Box 13**

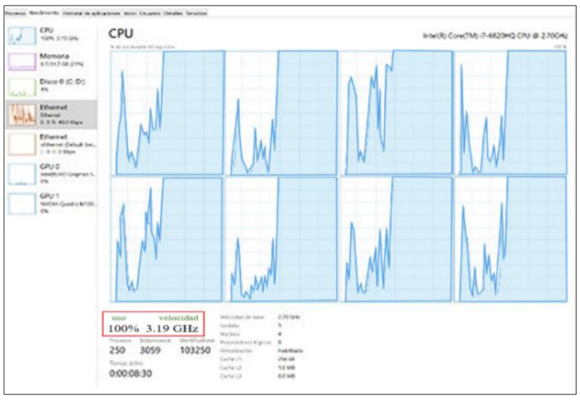


**Figure 12**
10 times simulation in size
Windows 10 Task Manager when <u>Parallel process is performed</u> of Bernstein copula either Regressions or Simulations.
*This figure is a Windows 10 Pro Task Manager Screenshot*

**Box 14**

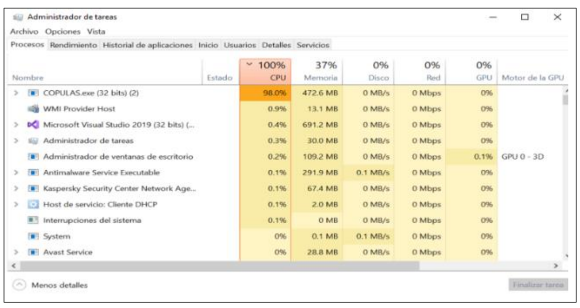


**Figure 13**
10 times simulation in size
Windows 10 Task Manager (List processes) when <u>Parallel process is performed</u> of <u>Bernstein copula either Regressions or Simulations</u>.
*This figure is a Windows 10 Pro Task Manager Screenshot*

It was asked a set of three quantile regressions (*0.1*, *0.5* and k) in nonparallel mode, the first one was performed in *1116.12* seconds (*18.60 minutes*), the second one (*quantile regression = 0.5)* was performed in *1842.80* seconds (*30.71 minutes*) and the third one (*quantile regression = 0.9*) was performed in *1862.98 seconds 31 minutes*.

Things changed substantially when parallel computing took place. As before, the total of calculations made for generate *2D* Empirical Copula was *216,410 calculations*, which they took *0.756 milliseconds*. By other side they were performed *73,087,401 calculations* for the *3D* empirical copula, which they took *5 milliseconds*.

The same sets of three quantile regressions (0.1, *0.5* and *0.9*) were performed. The first one was performed in *120* seconds (*2 minutes*), the second one (*quantile regression = 0.5*) was performed in *191* seconds (*~3 minutes*) and the third one (*quantile regression = 0.9*) was performed in *192* seconds (*~3 minutes*) and a final set of *380* simulations where performed it took *193* seconds (*~3 minutes*). In parallel mode, a total computing time was just of *8* minutes for the three regressions.

When comparing performance, the regression process in non-parallel mode required *4,821.9* seconds (*80 minutes, or 1 hour and 20 minutes*) to complete. In contrast, the same task executed in parallel mode took only *8 minutes—10 times faster*. Both tasks were conducted under identical computational conditions, highlighting the substantial efficiency gain achieved through parallel processing.

The difference between waiting *8 minutes* for results versus waiting *80 minutes* is significant, underscoring the practical advantages of parallelization, particularly in scenarios where timely analysis is critical. This performance enhancement not only saves time but also allows for more iterations and refinements within the same time frame, potentially leading to more accurate and robust outcomes.

A total of *3,800* simulations of the dataset were conducted under both conditions, i.e., in parallel and non-parallel modes. The non-parallel mode required *11,757.7* seconds (approximately three and a half hours) to complete the task, whereas the parallel mode completed the same task in *2,260.9* seconds (less than an hour).

## Conclusions

The proposed method represents a highly versatile tool for modeling the intricate dependence relationships between petrophysical properties, such as porosity, VS Meas and permeability.

Unlike traditional approaches, such as linear regression, this method does not require the assumption of linear dependencies between variables. This flexibility allows for a more accurate and efficient modeling of multivariate dependencies, capturing the underlying complexities in a manner that linear models fail to achieve.

ISSN: 2414-4959.
RENIECYT-CONAHCYT: 1702902


Hernández-Maldonado, Victor Miguel, Erdely, Arturo and Diaz-Viera, Martin Alberto. [2024]. Parallel computing for efficient calculation of multidimensional Bernstein copulas in modeling nonlinear dependence between random variables. ECORFAN-Journal Republic of Cameroon. 10[18]1-11: e61018111.
https://doi.org/10.35429/EJRC.2024.10.18.1.11

In addition to its inherent flexibility, the application of parallel processing techniques in the construction of multidimensional Nonparametric Copulas further enhances the effectiveness of the proposed method. By incorporating empirical data directly into the dependence structure, these parallel techniques not only accelerate computational performance but also enrich the model's ability to capture subtle and complex dependencies among multiple variables.

As a result, the proposed method demonstrates a marked improvement in predictive accuracy and robustness, making it a compelling alternative to conventional modeling techniques in the study of petrophysical properties.

In other scenarios in the petroleum industry, Bernstein copulas have been applied to model nonlinear dependences between fracture direction and length. Precise fracture network modeling is crucial for reservoir characterization, as fractures either obstruct or facilitate flow, making permeability estimation vital. Traditional linear statistical methods are insufficient for capturing these complex dependencies. [XIII].

**Acknowledgement**

The present work was supported by INFOTEC. Also was partially supported by the projects: IN110311 from *Programa de Apoyo a Proyectos de Investigación e Innovación Tecnológica* (PAPIIT) at *Universidad Nacional Autónoma de México FES Acatlan*, and to 116606 (Y.00102) from *Fondo Sectorial CONACYT-SENER-Hidrocarburos.*

**Appendix A**

This appendix provides a selection of the most critical lines of C++ computational code that are essential for replicating the results presented in this study.

These code snippets are intended to assist the reader in integrating the methods into their own programs, thereby enabling them to achieve similar outcomes and validate the findings discussed in this paper.

$$double ** db\_copEmp2D = Copula\_2Da1D(Sample, Rows, Cols, 1,0) \quad (10)$$

$$arrData = SORT(arrData, Rows, Cols, VarToSort); \quad (11)$$

$$double^{**} copEmp2D = NULL; \quad (12)$$

$$\text{copemp2D[i]} = (\text{double}^*)\text{malloc}((\text{Rows} + 1) * \text{sizeof(double)}); \quad (13)$$

$$for(inti = 0; i <= Rows; i++) for(intj = 0; j <= Rows; j++) copemp2D[i][j] = 0; \quad (14)$$

$$for(int\ j = 1; j <= Rows; j++) \quad (15)$$

$$x = (int)\ arrData[j-1][VarSec]; \quad (16)$$

$$\text{copemp2D[j][i]} = \text{copemp2D[j][i]} + (1.0/(\text{Rows})); \quad (17)$$

$$copemp2D[j+1][i] = copemp2D[j][i]; \quad (18)$$

*REGS*=(*double*∗∗)*malloc*((*intRegNum*)∗*sizeof*(*double*)); (19)

*REGS[i]=(double∗)malloc((Rows)∗*

*sizeof(double));* (20)

*parallelfor*(*size_t*(0),*size*,[&](*size_t m*) (21)

*REGS[i][m]=Regression(Sample[m][VS1-X], Sample[m][VS2-Y], quantil, SortedSamp, SV 1-X, SV 2-Y, PV-Z, dbCopEmp2Da1D, dbCopEmp3D-1D,Rows, 0.0001);* {(22)} (22)

**Declarations**

**Conflict of interest**

The authors hereby declare that there are no conflicts of interest associated with this work. They confirm that they have no known competing financial interests, affiliations, or personal relationships that could have influenced the research, interpretation, or conclusions presented in this article. The integrity and objectivity of the findings remain uncompromised by any external factors.

**Author contribution**

*Erdely Arturo*: Made a significant contribution to this article, particularly in the development and refinement of its theoretical aspects. His expertise and insights were instrumental in shaping the theoretical framework that underpins the research, and his involvement has greatly enhanced the overall quality and depth of the study

*Diaz-Viera, Martin Alberto*: Contribution was pivotal to this work, as he brings over 20 years of experience in the petroleum industry and spatial stochastic simulation. His insights carry significant weight and illuminate the path forward for the effective application of the tools proposed herein.




Hernández-Maldonado, Victor Miguel, Erdely, Arturo and Diaz-Viera, Martin Alberto. [2024]. Parallel computing for efficient calculation of multidimensional Bernstein copulas in modeling nonlinear dependence between random variables. ECORFAN-Journal Republic of Cameroon. 10[18]1-11: e61018111.
https://doi.org/10.35429/EJRC.2024.10.18.1.11

*Hernández-Maldonado Victor*: Made a substantial contribution to this work through his complete implementation of the computational aspects. He was responsible for designing and proposing the entire architecture of the computational models used in this study. His expertise in computational modeling played a crucial role in the successful execution of the research.

**Availability of data and materials**

This article includes a subset of the data used in the study. However, it is important to note that the Petrophysical data referenced are confidential and cannot be fully disclosed.

The available data provided within the article are intended to support the key findings while respecting the confidentiality agreements associated with the Petrophysical data.

**Acknowledgement and Funding:**

The present work was fully funded by INFOTEC.

Also it was partially supported by the projects: IN110311 from *Programa de Apoyo a Proyectos de Investigación e Innovación Tecnológica* (PAPIIT) at *Universidad Nacional Autónoma de México FES Acatlan*, and to 116606 (Y.00102) from *Fondo Sectorial CONACYT-SENER-Hidrocarburos.*

**References**

[I] Dong, H., Huang, S., Fang, W., Leng, G., Wang, H., Ren, K., ... & Ma, C. (2021). Copula-based non-stationarity detection of the precipitation temperature dependency structure dynamics and possible driving mechanism. Atmospheric Research, 249, 105280.

[II] Nelsen, R.B., 2006. An Introduction to Copulas, second ed. Springer, New York.

[III] Sklar, A., 1959. Fonctions de répartition á *n* dimensions et leurs marges, Publ. Inst. Statist. Univ. Paris. 8, 229-231.

[IV] Rüschendorf, L. (2009). On the distributional transform, Sklar's theorem, and the empirical copula process. Journal of statistical planning and inference, 139(11), 3921-3927.

[V] Billingsley, P., 1995. Probability and Measure, third ed. Wiley New York.

[VI] Segers, J., Sibuya, M., Tsukahara, H. (2017). The empirical beta copula. Journal of Multivariate Analysis, 155, 35-51.

[VII] Fermanian, J.D., Radulović, D., Wegcamp, M., 2004. Weak convergence of empirical copula processes. Bernoulli. 10(5), 847-860.

[VIII] Sancetta A, Satchell S. The Bernstein Copula and its Applications to Modeling and Approximations of Multivariate Distributions. *Econometric Theory*. 2004;20(3):535-562.

[IX] Sancetta, A., 2007. Nonparametric estimation of distributions with given marginals via Bernstein-Kantorovic polynomials: *L*1 and pointwise convergence theory. Journal of Multivariate Analysis. 98(7), 1376-1390.

[X] Hernández-Maldonado, V. M., Erdely, A., Díaz-Viera, M., & Rios, L. (2024). *Fast procedure to compute empirical and Bernstein copulas*. Applied Mathematics and Computation, *477*, 128827.

[XI] Hernández-Maldonado, V., Díaz-Viera, M., & Erdely, A. (2012). A joint stochastic simulation method using the Bernstein copula as a flexible tool for modeling nonlinear dependence structures between petrophysical properties. Journal of Petroleum Science and Engineering, 90, 112-123.

[XII] Hernández-Maldonado, V., Díaz-Viera, M., & Erdely, A. (2014). A multivariate Bernstein copula model for permeability stochastic simulation. Geofísica internacional, 53(2), 163-181.

[XIII] Mendoza-Torres, F., Díaz-Viera, M. A., & Erdely, A. (2017). Bernstein copula modeling for 2D discrete fracture network simulations. *Journal of Petroleum Science and Engineering*, *156*, 710-720.

[XIV] Van Huong Le, Díaz-Viera, M. A., Vázquez-Ramírez, D., del Valle-García, R., Erdely, A., & Grana, D. (2020). Bernstein copula-based spatial cosimulation for petrophysical property prediction conditioned to elastic attributes. Journal of Petroleum Science and Engineering, 193, 107382.

ISSN: 2414-4959.
RENIECYT-CONAHCYT: 1702902