

TRABAJO ESPECIAL DE GRADO

**VALIDACIÓN DE UN ALGORITMO SIMULADOR DE DATOS
BIDIMENSIONALMENTE GEOREFERENCIADOS**

Presentado ante la Ilustre
Universidad Central de Venezuela
Por el Br. Natera O., Armando J
Para optar al Título
De Ingeniero Geofísico

Caracas, 2013

TRABAJO ESPECIAL DE GRADO

**VALIDACIÓN DE UN ALGORITMO SIMULADOR DE DATOS
BIDIMENSIONALMENTE GEOREFERENCIADOS**

TUTOR ACADÉMICO: Prof. Ricardo Ambrosio

Presentado ante la Ilustre
Universidad Central de Venezuela
Por el Br. Natera O., Armando J
Para optar al Título
De Ingeniero Geofísico

Caracas, 2013

Caracas, mayo de 2013

Los abajo firmantes, miembros del Jurado designado por el Consejo de Escuela de Geología, Minas y Geofísica, para evaluar el Trabajo Especial de Grado presentado por el Bachiller **ARMANDO JESÚS NATERA ORTEGA**, titulado:

**“VALIDACIÓN DE UN ALGORITMO SIMULADOR DE DATOS
BIDIMENSIONALMENTE GEOREFERENCIADOS”**

Consideran que el mismo cumple con los requisitos exigidos por el plan de estudios conducente al Título de Ingeniero Geofísico, y sin que ello signifique que se hacen solidarios con las ideas expuestas por el autor, lo declaran APROBADO.

Prof. Mauricio Bermudez
Jurado

Prof. Ronny Meza
Jurado

Prof. Ricardo Ambrosio
Tutor Académico

DEDICATORIA

A mis padres... por haberme enseñado los valores y principios fundamentales de la vida y estar allí brindando apoyo cada vez que fue necesario, así que este trabajo representa una pequeña retribución a su esfuerzo y dedicación.

Para ustedes.

AGRADECIMIENTOS

En primer lugar, a mis padres Dinorah Ortega y Armando Natera, por haberme apoyado durante todo este tiempo facilitando enormemente las calamidades y obstáculos que se han presentado y permitiendo de manera inequívoca que hoy haya llegado a alcanzar esta meta.

Mamá, en un principio no estuviste de acuerdo con que estudiara esta carrera en la Ilustre Universidad Central de Venezuela, sin embargo, permitiste que lo hiciera y gracias a eso hoy me encuentro subiendo este peldaño de la escalera de la vida, gracias por haber confiado.

A Ricardo Ambrosio por ofrecerme este trabajo y haber prestado apoyo incondicional cada vez que fue necesario sin importar las circunstancias, ha sido toda una experiencia y por lo tanto siempre estaré agradecido.

A Miguel Gascón por haber sido el primero en enseñarme los fundamentos básicos de la programación en lenguaje R y haberme ayudado a resolver infinidad de problemas referentes a ese rubro de manera desinteresada y amable.

A Matías Corrales por prestar apoyo desde el punto de vista de sintaxis de programación cada vez que me encontraba con dilemas filosóficos. Además por proporcionar opiniones productivas cada vez que estas fueron solicitadas.

A Daybelis Mendonca por haberme hecho reír diariamente con sus ocurrencias y acciones, amenizando la rutina y por lo tanto haciendo más agradable el trabajo.

A Yaneth Garzón por sentarse conmigo a revisar la redacción del tomo todas las veces que se solicitó, brindarme su amistad y por fastidiar a Ricardo las veces que se olvidaba de hacer las correcciones.

A Giuseppe Loconte y Francisco Sánchez por presionarme con la redacción del tomo las veces que se percataban de mi inconsecuencia y además por prestar apoyo cada vez que se necesitó.

A todos, muchas gracias...

Natera O., Armando J

**VALIDACIÓN DE UN ALGORITMO SIMULADOR DE DATOS
BIDIMENSIONALMENTE GEOREFERENCIADOS.**

Tutor Académico: Prof. Ricardo Ambrosio.

**Tesis. Caracas, U.C.V. Facultad de Ingeniería. Escuela de Geología, Minas y
Geofísica. Departamento de Geofísica. Año 2012, 80 pág.**

Palabras Claves: *geoestadística, variogramas, georeferenciados, simulación,
algoritmo.*

El objetivo general del presente trabajo es validar, mediante una serie de pruebas, el desempeño de un algoritmo simulador de datos sintéticos bidimensionalmente georeferenciados que cumplen con características estadísticas, geoestadísticas y de visualización establecidas por el usuario. Las pruebas se enfocan en comprobar la convergencia de los modelos y establecer combinaciones de las cinco variables geoestadísticas ajustables de manera que se fijen cuatro de ellas mientras se varía la restante a lo largo del dominio y así comprobar su funcionamiento por separado. Una vez validada la variable, se le asigna cualquier valor perteneciente a su dominio para la siguiente prueba. Dicha validación se ejecuta a través del análisis de los mapas de salida y de la construcción de los variogramas experimentales de dichos datos, comprobando su ajuste al modelo teórico inicial. Los resultados permiten sustentar el buen desempeño del algoritmo dando paso a una nueva línea de investigación dentro del campo de las geociencias basada en estos datos simulados según requerimientos específicos.

TABLA DE CONTENIDO

ÍNDICE DE FIGURAS.....	ix
INTRODUCCIÓN	1
1.1 PLANTEAMIENTO DEL PROBLEMA.....	1
1.2 OBJETIVOS.....	3
1.1.1 Objetivo general	3
1.1.2 Objetivos específicos	3
1.3 JUSTIFICACIÓN.....	3
MARCO TEÓRICO	5
2.1 ENFOQUE ESTADÍSTICO Y GEOESTADÍSTICO.....	5
2.1.1 Estacionariedad y no-estacionariedad.....	5
2.1.2 Variograma o semivariograma.....	5
2.1.3 Parámetros intrínsecos del modelo teórico de correlación espacial.....	5
2.1.4 Variogramas teórico.....	6
2.1.5 Variogramas experimentales.....	7
2.1.6 Mallado.....	8
2.1.7 Espaciamiento del mallado.....	8
2.1.8 Técnicas de análisis estadístico y geoestadístico.....	8
2.1.9 Técnicas frecuentemente usadas de interpolación.....	8
2.1.10 Kriging.....	9
2.1.11 Simulación secuencial gaussiana (S.G.S.).....	10
2.1.12 Simulación condicional o estocástica.....	10
2.2 ENFOQUE A PROGRAMACIÓN.....	12
2.2.1 Ambiente de una función.....	12
2.2.2 Archivo fuente.....	12
MARCO METODOLÓGICO	13
3.1 DISEÑO	13
3.1.1 Características y precauciones sobre los insumos.....	14

3.1.2	Funcionamiento del algoritmo	15
3.2	VALIDACIÓN	17
3.3	ENFOQUE A PROGRAMACIÓN.....	18
3.4	DESCRIPCIÓN DE LA SECUENCIA DE PROCESAMIENTO	20
RESULTADOS Y ANÁLISIS.....		25
4.1	VALIDACIÓN DE LAS VARIABLES GEOESTADÍSTICAS.....	25
4.1.1	Prueba N°1.	27
4.1.2	Prueba N°2.	29
4.1.3	Prueba N°3.	31
4.1.4	Prueba N°4.	32
4.1.5	Prueba N°5.	34
4.2	COMPROBACIÓN DE LA CONVERGENCIA DE LOS MODELOS.	35
4.2.1	Variogramas de las 30 simulaciones	35
4.2.2	Diagramas de caja y bigote de las 30 simulaciones	40
4.2.3	Ajuste del variograma promedio experimental al modelo teórico	41
4.3	DESCRIPCIÓN DE LOS POST-PROCESOS.....	41
CONCLUSIONES Y RECOMENDACIONES.....		51
REFERENCIAS BIBLIOGRÁFICAS		52

ÍNDICE DE FIGURAS

Figura 3.1 Secuencia general del diseño del algoritmo.....	13
Figura 3.2 Diagrama de flujo del funcionamiento interno del simulador.	16
Figura 4.1 Mapas referentes a la prueba N°1.....	28
Figura 4.2 Mapas referentes a la prueba N°2, variación de rangos de menor a mayor de izquierda a derecha.	30
Figura 4.3 Mapas referentes a la prueba N°3, variación de la tasa de anisotropía de menor a mayor de izquierda a derecha.....	31
Figura 4.4 Mapas referentes a la prueba N°4, variación de la dirección de anisotropía.	33
Figura 4.5 Mapas referentes a la prueba N°5, variación del nugget de mínimo a máximo de izquierda a derecha.....	34
Figura 4.6 Variogramas experimentales principales de las 30 realizaciones.....	37
Figura 4.7 Diagrama de caja y bigote de los variogramas secundarios de las 30 realizaciones, variograma promedio y mediano.....	38
Figura 4.8 Variogramas experimentales secundarios de las 30 realizaciones.	39
Figura 4.9 Diagrama de caja y bigote de los variogramas secundarios de las 30 realizaciones, variograma promedio y mediano.....	40
Figura 4.10 Variogramas promedio de máxima y mínima continuidad espacial de las 30 realizaciones.	41
Figura 4.11 Residuales originales o datos estacionarios simulados.....	32
Figura 4.12 Residuales transformados.....	32
Figura 4.13 Superficie de tendencia.....	33
Figura 4.14 Mapa residual mas regional.....	34
Figura 4.15 Crossplot de la variable simulada con la correlacionada.....	35
Figura 4.16 Mapa de cuartiles de la muestra con sesgo alto hacia valores mínimos.....	36
Figura 4.17 Mapa de cuartiles de la muestra con sesgo bajo hacia valores máximos.....	36

Figura 4.18 Mapa de cuartiles de la muestra con patrón aleatorio uniforme.....37

ÍNDICE DE TABLAS

Tabla 1 Descripción de los parámetros empleados para las 5 pruebas de validación de las variables geoestadísticas.....	26
Tabla 2 Descripción de los parámetros de ejecución de la corrida de las 30 realizaciones.....	36
Tabla 3 Descripción de los parámetros empleados para la nueva simulación referente a los post-procesos.	42

CAPÍTULO I

INTRODUCCIÓN

1.1 PLANTEAMIENTO DEL PROBLEMA.

En una publicación del año 2008, Jimba, señala que los algoritmos geoestadísticos son complicados de programar y corregir, considerando todas las posibles combinaciones de herramientas y sistemas operativos. Sin embargo, a través de los últimos 40 años, se ha observado el desarrollo de las técnicas geoestadísticas, lo cual combinado con los estudios de ciencias de la tierra, ha permitido un mejor entendimiento y mejor integración en el flujograma de trabajo en las disciplinas de geología, geofísica e ingeniería de yacimientos. Lo anterior representa importantes avances metodológicos en dichas disciplinas, y a pesar de que hoy en día aún existen muchos geocientíficos que conciben las geoestadísticas como una caja negra de generación de números aleatorios que no guarda ninguna relación con las restricciones geológicas, muchos progresos aún pueden ser realizados (Dubrule, 2003).

Según Chiles y Delfiner (1999), en diversas áreas de las geociencias, como por ejemplo, el sector minero, ambiental, petrolero, geotecnista, entre otros, se llevan a cabo proyectos que se valen de la utilización de variadas herramientas estadísticas descriptivas que permiten realizar controles de calidad de datos, determinar tendencias y observar comportamientos atípicos que pudieran resultar de interés en los mismos. Por otro lado, Deutsch en el 2002 afirma que también se emplean herramientas geoestadísticas que facilitan la estimación del valor de propiedades en las localizaciones no observadas, evaluar la distribución de ciertas características de un medio y ayudar a reducir riesgos que sustenten la toma de decisiones. Utilizar dichas herramientas es posible siempre y cuando se disponga de un conjunto de datos adquiridos en campo y que sirven de insumo para la aplicación de las metodologías nombradas anteriormente.

No obstante, a nivel académico en el Departamento de Geofísica de la Facultad de Ingeniería de la Universidad Central de Venezuela (U.C.V), tiende a evidenciarse la poca disponibilidad de datos georeferenciados para la realización de investigaciones de distinta índole, entre las cuales destacan trabajos especiales de grado, así como, desarrollo de clases de pre y post grado, debido a que los mismos son de difícil acceso por su confidencialidad motivado, entre otras cosas, al alto costo en que se incurre para adquirir dichos datos, lo cual afecta la producción científica del departamento.

En función de lo anterior y con la finalidad de ofrecer una solución viable a la problemática mencionada, este trabajo se enfatizó en validar un algoritmo de simulación de propiedades bidimensionalmente distribuidas que cumpliera con características y patrones introducidos por el usuario. Para el desarrollo del mismo, se emplearon herramientas de diversas disciplinas entre las cuales resaltan la matemática, estadística, geoestadística y programación

1.2 OBJETIVOS

1.1.1 Objetivo general

Validar una herramienta computacional de simulación de propiedades bidimensionalmente distribuidas que cumplan con patrones introducidos por el usuario.

1.1.2 Objetivos específicos

- Analizar metodologías geoestadísticas relacionadas con el algoritmo de simulación de propiedades bidimensionalmente distribuidas.
- Diseñar el algoritmo en función de los resultados obtenidos en las pruebas realizadas.
- Generar una herramienta muestreadora de datos observados.
- Establecer criterios de validación de la herramienta muestreadora.

1.3 JUSTIFICACIÓN

La importancia del desarrollo de este proyecto radica en fomentar la investigación en el ámbito de las geociencias, principalmente en el sector académico de la Universidad Central de Venezuela, Facultad de Ingeniería, Escuela de Geología, Minas y Geofísica, Departamento de Geofísica; actualmente dicho departamento presenta un déficit de disponibilidad de datos vigentes y fidedignos pese a las dificultades que acarrearán los otorgamientos de permisos asociadas a los mismos, por parte de las empresas patrocinantes, los cuales retrasan considerablemente los tiempos pautados para el desarrollo de dichos trabajos especiales de grado. Por otra parte, también se estima alcanzar la meta de implementar un muestreador de datos observados para realizar experimentos que sirvan para hacer comparaciones e interpretaciones de interés que conlleven a un mejor entendimiento del medio.

Desde el punto de vista metodológico, dicha herramienta, ofrece una nueva línea de investigación en cuanto a procedimientos de estimación, predicción y caracterización, mediante la utilización de un conjunto de datos simulados, georeferenciados y amoldados a las necesidades del usuario. Esto trae beneficios tanto para la comunidad

estudiantil como a la profesoral e investigadora, implicando entonces una mejor gestión universitaria.

CAPÍTULO II

MARCO TEÓRICO

2.1 ENFOQUE ESTADÍSTICO Y GEOESTADÍSTICO.

2.1.1 Estacionariedad y no-estacionariedad.

La estacionariedad es la propiedad de los datos de no exhibir con claridad una tendencia, es decir, homogeneidad estadística espacial. Esto implica que una ventana de promedio en movimiento muestra homogeneidad en la media y varianza sobre toda el área de estudio. Esto puede verificarse de varias maneras tales como la construcción del variograma ajustado a los datos el cual, de presentar una clara tendencia, no alcanzará el techo o la varianza sino que siempre crece. De manera análoga, los datos son no-estacionarios cuando presentan una clara tendencia descriptible y estimable (Dubrule, 2003).

2.1.2 Variograma o semivariograma.

Es una medida de dependencia espacial de una variable regionalizada sobre cierta distancia; un gráfico de disimilitud entre puntos como una función de distancia entre los mismos. Este puede ser calculado con preferencia direccional. En otras palabras, es una medida geoestadística usada para caracterizar la variabilidad espacial de un atributo (Jimba, 2008). Desde el punto de vista netamente matemático se define como la esperanza de la diferencia al cuadrado entre todos los pares de puntos ubicados espacialmente a una distancia “h” a lo largo de una dirección y pertenecientes a un mismo universo tal como lo señala la ecuación 1.

$$2\gamma(h) = E\{[Z(u+h) - Z(u)]^2\}; \quad \forall u, u+h \in A \dots \dots \dots (1)$$

La diferencia principal entre el variograma y el semivariograma es que el primero es el doble del otro respectivamente (Deutsch, 2002).

2.1.3 Parámetros intrínsecos del modelo teórico de correlación espacial.

Los variogramas se caracterizan, independientemente del modelo matemático, por ser definidos mediante tres parámetros principales: el sill, el rango y el nugget.

- **Sill:** es el nivel superior de la varianza, donde el variograma alcanza su rango de correlación. La media de la muestra de la población es el sill teórico del modelo de correlación espacial.

- **Nugget:** El “efecto nugget” representa el comportamiento a distancias menores que el mínimo espaciado de puntos considerado para los cálculos del variograma experimental o también conocido como “lag” y se denota como una discontinuidad aparente en el origen del gráfico. (Deutsch, 2002). El término “efecto nugget” debe su origen a la primera utilización del kriging para la estimación de la proporción de mineral de oro y reservas en Suráfrica, haciendo referencia a las “pepitas de oro”.

- **Rango:** es la distancia sobre una dirección específica a la cual el variograma alcanza el sill, con la particularidad de que los puntos ubicados a distancias inferiores o iguales al mismo están espacialmente correlacionados y volviéndose no correlacionados todos aquellos superiores a él. Éste suele ser un buen indicador de anisotropía (Jimba, 2008).

2.1.4 Variogramas teórico.

Existen diversos tipos de variogramas teóricos que son funciones reales de variable real de la forma $\gamma = f(h)$, donde “h” es la variable independiente referente a la distancia o lags del variograma; que reciben de insumo los valores de rango (r), sill (s) y nugget (n) clasificados por su regla de correspondencia matemática entre los cuales resaltan tres modelos principales:

- **Variograma Exponencial:** Es el modelo que presenta mayor heterogeneidad intrínseca consecuencia de su regla de correspondencia exponencial, es decir, que alcanza el sill con mayor rapidez tal como lo muestra la ecuación 2.

$$\gamma(h) = (s - n) \left(1 - e^{-\left(\frac{h}{r*a}\right)} \right) + n \dots \dots \dots (2)$$

- **Variograma Gaussiano:** Es el modelo que presenta mayor homogeneidad a distancias cercanas al origen dado a su regla de correspondencia gaussiana, es decir, presenta poca variabilidad espacial a distancias cortas tal como lo señala la ecuación 3.

$$\gamma(h) = (s - n) \left(1 - e^{-\left(\frac{h}{r\sqrt{a}}\right)^2} \right) + n \dots \dots \dots (3)$$

- **Variograma Esférico:** Es el modelo semi-lineal que presenta una heterogeneidad media respecto a la de los dos modelos descritos anteriormente y que alcanza el sill de manera definitiva y no asintótica a diferencia de los otros dos tal como lo muestra su regla de correspondencia en la ecuación 4.

$$\gamma(h) = (s - n) \left(\frac{3h}{2r} - \frac{h^3}{2r^3} \right) + n ; \text{para } h < r ; \gamma(h) = s ; \text{si } h \geq r \dots \dots \dots (4)$$

El valor “a” es una constante de equivalencia entre los modelos matemáticos el cual puede ser encontrado con pequeñas variaciones en distintas referencias bibliográficas dado la ambigüedad en la definición del rango, para Chiles y Delfiner en su publicación del año 1999, a=1/3. Una manera simple de entender el concepto de la constante de equivalencia es mediante una analogía, por ejemplo, para el cálculo de integrales múltiples siempre que se realiza un cambio de sistemas de coordenadas se genera además un valor de proporcionalidad conocido como jacobiano el cual multiplica al integrando, sin embargo, calcular la integral en uno u otro marco de referencia conlleva siempre al mismo resultado numérico por lo que dicho valor de proporción pudiera ser considerado a su vez como una constante de equivalencia desde un punto de vista abstracto.

2.1.5 Variogramas experimentales.

Es el variograma calculado a partir de un conjunto de datos estacionarios mediante la ecuación 1. Por lo general se definen dos direcciones horizontales mutuamente ortogonles como patrón de búsqueda espacial y además considerando un valor de tolerancia para tomar en cuenta las distancias aproximadas. Un patrón de búsqueda omnidireccional es diseñado mediante la selección de una dirección cualquiera y permitiéndole una tolerancia de 90 grados transformando la elipse de anisotropía en una circunferencia. Este tipo de modelos es conveniente para el variograma inicial el cual subsecuentemente será redefinido por un modelo direccional (Jimba, 2008).

2.1.6 Mallado.

Llevado a los extremos, cada mapa contiene un número infinito de puntos en su área. Dado a que resulta poco eficiente muestrear o estimar el valor de una variable a la totalidad infinita de puntos que forman su superficie, se define un mallado regular para describir locaciones en las que las estimaciones serán calculadas y posteriormente utilizadas para el proceso de realización de dicho mapa. Un mallado es construido mediante la colocación de valores dentro de un arreglo regularmente espaciado, comúnmente cuadrado o rectangular, aunque otras formas podrían ser también usadas (Jimba, 2008).

2.1.7 Espaciamiento del mallado.

Este controla el nivel de detalle o resolución con que puede ser visto en el mapa. Las características que sean más pequeñas que el intervalo de espaciamiento del mallado no podrán ser visualizadas. Por lo tanto, espaciamientos grandes entre nodos producen mapas a tiempos cortos y a baja resolución, mientras que a espacios cortos se incrementan los tiempos de cómputo, usualmente a razón exponencial, generando mejores resoluciones y por tanto mejor apariencia (Jimba, 2008).

2.1.8 Técnicas de análisis estadístico y geoestadístico.

La estadística es una rama de la matemática que se encarga de estudiar los procesos de recolección, análisis e interpretación de conjuntos de datos, exhibir sus funciones de distribución probabilística, realizar controles de calidad de datos, entre otras cosas (Davis, 1986). Por otro lado, la geoestadística se encarga de analizar datos correlacionados espacialmente, así como también estimar los valores de dichas variables que se distribuyen en espacio, en localizaciones donde éstas no han sido observadas (Olea, 1991). Entre las herramientas más empleadas en las estadísticas y geoestadísticas resaltan los histogramas y variogramas (Deutsch, 2002).

2.1.9 Técnicas frecuentemente usadas de interpolación.

Desde que se comenzó a aplicarse la computadora como herramienta versátil dentro del área de la geoestadística, particularmente en el sector de la interpolación, han

tenido auge diversas metodologías de estimación de propiedades espacialmente referenciadas entre las cuales destacan el kriging y la simulación secuencial gaussiana o S.G.S. por sus siglas en inglés (Deutsch, 2002). El kriging por su lado es un método para calcular estimaciones de una variable regionalizada usando una combinación lineal de pesos obtenida a partir de un modelo espacial de correlación. El mismo asigna pesos a las muestras para minimizar la estimación de la varianza. Existen distintos tipos de kriging clasificados por nombre según los parámetros que incluye en sus sistemas de ecuaciones tales como el kriging ordinario, simple, universal, co-kriging, kriging con deriva externa, entre otros (Jimba, 2008). Sin embargo, el kriging en general presenta un detalle, tiende a suavizar los resultados en algunos casos de manera indeseable, opacando los mínimos y máximos (Jimba, 2008). Por su parte la S.G.S. es un método eficiente que resalta los puntos extremos y sigue un carácter de aleatoriedad por cada simulación, conllevando algunas veces a una imagen caótica pero que sigue con rigurosidad las tendencias de los datos de entrada y ofreciendo entonces una excelente opción para aplicaciones como el modelado de yacimientos en la industria petrolera (Deutsch, 2002). Esta supone que cada simulación presenta una media y varianza particular con los cuales simula una observación de su distribución específica, además, recorre el mallado regular de localizaciones no observadas en forma aleatoria e incorpora cada punto simulado como un nuevo punto observado el cual puede ser utilizado para la siguiente interpolación y así evitar que se genere correlación secundaria intrínseca entre los vecinos consecutivos en caso de que dichos nodos se recorrieran en forma ordenada bajo un patrón específico lo cual se conoce, dentro de la geoestadística, como un artefacto (Deutsch, 2002).

2.1.10 Kriging.

Según Deutsch, es un método de interpolación desarrollado en los años 60 por el ingeniero de minas Daniel Krige quien empleó los estudios previamente realizados por Georges Matheron en 1950. Este método tiene la finalidad última de generar un conjunto de datos que puedan ser representados mediante un mapa de contornos

suavizado y continuo en toda una región a partir de una serie de datos discretizados de la variable regionalizada bajo la premisa de un modelo de variograma teórico y unos pesos que indican el aporte de cada dato observado para la estimación de uno no observado según sea la distancia entre los mismos. El sistema de ecuaciones da como resultado la media y la varianza de cada valor interpolado. La ecuación matricial que se emplea para el kriging de “n” muestras se observa en las ecuaciones 5 y 6:

$$C\lambda = c \dots \dots \dots (5)$$

La cual se expande de la siguiente forma (ecuación 6):

$$\begin{bmatrix} C(u_1 - u_1) & \dots & C(u_1 - u_n) \\ \vdots & \ddots & \vdots \\ C(u_n - u_1) & \dots & C(u_n - u_n) \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \vdots \\ \lambda_n \end{bmatrix} = \begin{bmatrix} C(u - u_1) \\ \vdots \\ C(u - u_n) \end{bmatrix} \dots \dots \dots (6)$$

Donde, $C(u_i - u_j)$ es el valor de variograma de puntos observados, λ_i los pesos relativos a las distancias de los vecinos al punto que se desea estimar y $C(u - u_i)$ son los valores de variograma relativos a la posición no observada que se desea estimar.

2.1.11 Simulación secuencial gaussiana (S.G.S.).

Es un método de simulación estocástica de datos bajo un patrón aleatorio de recorrido del mallado o mediante una matriz de covarianza que usa como método de interpolación el kriging para obtener la media y la varianza de cada realización y así construir una distribución gaussiana de probabilidades y tomar una muestra al azar que no necesariamente corresponde con la media para cada simulación por lo que suele generar una imagen caótica o de elevada entropía. Cada punto simulado se considera como un dato observado para las simulaciones consecuentes (Deutsch, 2002).

2.1.12 Simulación condicional o estocástica.

Es un método geoestadístico para crear múltiples e igualmente probables realizaciones de una variable regionalizada basado en un modelo espacial. Esta es condicional sólo cuando los datos de control son realmente honrados, es decir, bien representados por los simulados (Jimba, 2008).

- **Ventajas de la simulación condicional.**

Según Jimba, en su publicación del 2008, se exhiben las siguientes ventajas:

- Mediante la distribución del histograma de los datos y la estructura de correlación espacial, la simulación condicional provee imágenes de yacimientos más realistas que el kriging.
- Son de gran utilidad para simular volúmenes y tasas de producción de hidrocarburos dado a que éstas pueden reproducir los valores extremos y sus patrones de conectividad.
- La simulación condicional provee modelos alternativos que son consistentes con los datos.
- Las simulaciones generan diferentes pero igualmente probables escenarios geológicos para ser usados en valoración de riesgo.

- **Limitaciones de la simulación condicional.**

Según Jimba, en su publicación del 2008, se exhiben las siguientes limitaciones:

- Requiere de un ordenador potente para las simulaciones.
- Enormes números de simulaciones podrían crear conflictos de manejo de datos.
- Se deben interpretar los intervalos de confianza con precaución ya que la incertidumbre en los datos condicionales podría ser extensa.
- Las simulaciones son muy sensibles a los parámetros del modelo de covarianza, como el sill y el nugget, o al coeficiente de correlación si se emplea la co-simulación co-ubicada.
- Dispersar los datos condicionales generalmente produce un amplio rango de variabilidad entre las simulaciones.
- De las simulaciones igualmente probables estadísticamente resultan algunas realizaciones geológicamente improbables o poco realistas.

2.2 ENFOQUE A PROGRAMACIÓN

2.2.1 Ambiente de una función.

Es el universo en donde interactúan únicamente variables internas de una función, las cuales sirven para ejecutar el proceso o los procesos específicos de la misma y que se pierden una vez que estos culminan y devuelven, al ambiente (de un nivel superior en jerarquía) donde se ejecutó la función, su parámetro de salida. El objetivo de emplear distintos ambientes sigue la filosofía de ocupar menos memoria en disco y así optimizar el funcionamiento de los procesos. Obtenido de: <http://cran.r-project.org/>

2.2.2 Archivo fuente.

Existen muchos tipos de archivos fuente todos destinados a propósitos específicos según la naturaleza de su aplicación, herramienta computacional en el cual interactúa, orientación del programador, entre otras. Para el caso particular de los archivos fuente orientados a R, son archivos de formato “.R” como cualquier otro código pero que cumplen la función de ser leídos por la consola sin necesidad de ser ejecutados manualmente en la misma, sino que se corre un proceso que llama a tal archivo y lo ejecuta automáticamente en mucho menos tiempo. Esto se aplica principalmente para códigos largos con la finalidad de ahorrar tiempos de cómputo y amenizar el proceso. Obtenido de: <http://cran.r-project.org/>

CAPÍTULO III

MARCO METODOLÓGICO

3.1 DISEÑO

La secuencia lógica empleada para el diseño del algoritmo se describe seguidamente en la figura 3.1.



Figura 3.1 Secuencia general del diseño del algoritmo

Los requerimientos del usuario son los siguientes:

- Las coordenadas del área de estudio.
- Resolución de los datos sintéticos la cual corresponde con la raíz cuadrática del número de puntos que se desea tenga el mallado de salida.
- Los parámetros descriptivos del modelo de correlación espacial.
 - Tipo de variograma: exponencial, esférico o gaussiano.
 - Dirección de anisotropía: ángulo azimutal en grados medido desde el norte en sentido horario.

- Tasa de anisotropía: relación entre el rango de menor y mayor continuidad espacial. Entre 0 y 1.
- Rango de mayor continuidad espacial.
- Nugget.
- Los coeficientes de la superficie de tendencia de los regionales.
 - Polinomio de grado 1 ó 2.
- Parámetros necesarios para la toma de muestras.
 - Porcentaje de la muestra a tomar.
 - Intensidad del sesgado.
 - Sentido del sesgado: ascendente o descendente.
- Parámetros de la variable correlacionada.
 - Coeficiente de correlación.
 - Media y varianza de la variable a partir de la cual se genera la correlación.
- Parámetros para la transformación de la propiedad simulada.
 - La media: coeficiente independiente de la superficie de tendencia
 - Desviación estándar de la misma.

3.1.1 Características y precauciones sobre los insumos

Algunas de las consideraciones importantes que se deben tomar en cuenta para la selección de parámetros de entrada son:

- El tiempo de cómputo que toma la simulación de datos de alta resolución dado a que el mismo se comporta como una curva exponencial y varía según las especificaciones del software y hardware en el cual se ejecute la corrida.
- Verificar que los rangos de los variogramas no excedan el 50% de la longitud de la diagonal principal de la zona de estudio.

3.1.2 Funcionamiento del algoritmo

La herramienta computacional creada opera como una gran función que recibe de insumo los requerimientos del usuario y arroja de salida un conjunto de datos sintéticos simulados. Dicho simulador toma los parámetros de entrada para construir principalmente tres objetos: un modelo de correlación espacial, un conjunto de datos independientes que para este trabajo llamaremos “datos iniciales simulados” y un mallado de salida. Estos objetos se construyen con la finalidad de ejecutar una simulación secuencial gaussiana, la cual utiliza: el modelo de correlación espacial como criterio de interpolación direccional, los datos iniciales simulados como puntos de referencia y el mallado de salida para almacenar los datos simulados. Una vez culminado este proceso se tiene el conjunto de datos sintéticos simulados estacionarios. Es importante resaltar que dichos datos pasan a ser no estacionarios una vez aplicada la superficie de tendencia mediante un post-proceso. La secuencia de procesos descrita anteriormente se visualiza en la figura 3.2.

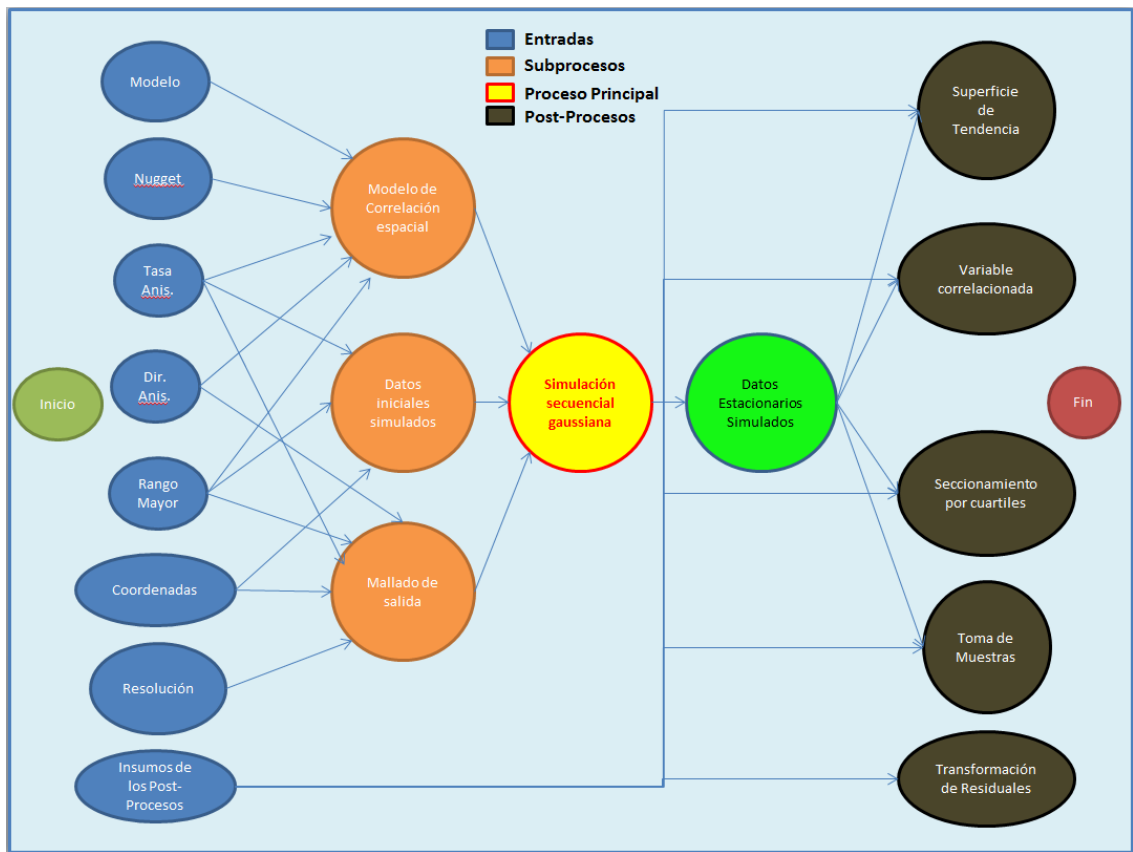


Figura 3.1 Diagrama de flujo del funcionamiento interno del simulador.

Modelo de correlación espacial: Está compuesto por los variogramas teóricos, los cuales pudieran ser omnidireccionales en caso de isotropía espacial o anisotrópicos en el caso contrario, de manera que se construye un modelo en la dirección principal y otro en su dirección ortogonal según las especificaciones del usuario.

Datos iniciales simulados: Se simula un mallado de puntos independientes e idénticamente distribuidos, cuya configuración espacial de coordenadas depende del modelo de correlación indicado por el usuario (distribución uniforme bajo escenario isotrópico) y cuyo valor de la propiedad inicial simulada es una distribución normal tipificada. Dicho mallado posteriormente es rotado en la dirección de anisotropía asignada para así garantizar continuidad espacial en dichos ángulos. Estos datos son removidos durante la simulación una vez que ya no son necesarios, cuando ha sido cubierto el 98% del área con datos simulados, para así evitar añadir artefactos a los mismos.

Mallado de salida: Se construye una matriz de posiciones circunscritas al área de estudio a una resolución similar a la indicada en la sección de parámetros de visualización, y se va llenando de manera aleatoria a medida que se ejecuta la S.G.S.

S.G.S: Se hace uso de los tres objetos descritos previamente para realizar la simulación secuencial gaussiana y entonces generar el conjunto de datos sintéticos residuales solicitados por el usuario.

Post-Procesos: En esta sección se encuentran todos los procesos que hacen uso de los datos simulados estacionarios, es decir, la superficie de tendencia, la variable correlacionada, la transformación de los residuales, el seccionamiento por cuartiles y la función de tomas de muestras. Dicho criterio para tomar muestras se basa en sesgado alto o bajo hacia valores extremos de los datos, ya sean mínimos o máximos, según sea requerido. Otro criterio implementado no emplea el sesgado sino la toma de la muestra de manera uniforme a lo largo de toda el área de estudio.

3.2 VALIDACIÓN

En esta sección se describen los diversos controles de calidad llevados a cabo para evaluar el desempeño del algoritmo.

Se generó una serie de pruebas de simulación estocástica de datos sintéticos con el propósito de construir todos los variogramas en un mismo marco de referencia para inspeccionar visualmente la varianza de los resultados, elaborando además el variograma experimental promedio y comprobando su ajuste al modelo teórico inicial, verificando así su convergencia.

Por otro lado, se elaboró el diagrama de caja y bigote de los variogramas a manera de ponderar con mayor precisión su varianza así como también verificar la similitud del variograma medio y el mediano experimental.

Una última verificación o control de calidad se realiza al visualizar el mapa de los datos sintéticos para comprobar que se honren las especificaciones solicitadas por el usuario.

3.3 ENFOQUE A PROGRAMACIÓN

Esta sección está dedicada única y exclusivamente a dar una breve descripción acerca de los paradigmas de programación empleados para la realización de este trabajo y que están vinculados directamente con el código. Además, especificaciones acerca de las versiones de las plataformas utilizadas, librerías, funciones nativas y creadas.

El algoritmo principal fue diseñado en lenguaje R, versión 2.13.2, el cual es un paradigma de programación funcional interpretativo que dispone de diversas funciones nativas orientadas principalmente a estadística y geoestadística a través del uso de librerías especializadas en tales tópicos. En términos detallados, se emplearon las librerías “lattice”, “compiler” y “gstat” las cuales son las referentes a herramientas versátiles pre-programadas y de sencilla ejecución tales como histogramas de frecuencias, compilador de funciones, variogramas teóricos y experimentales, simulación de variables aleatorias uniformes, normales e interpolación mediante kriging ordinario.

En cuanto a la estructura del código fuente, se diseñaron funciones que ocupan un papel específico dentro de la secuencia de procesos y una que se encarga de llamar a los subprocesos de manera ordenada, proporcionándole los parámetros de entrada que cada una necesita, mostrando en ventanas de la consola los resultados simulados y respaldando toda la información en una carpeta especificada en archivos de texto e imágenes de formato pdf. Estos subprocesos se programaron inicialmente de manera aislada por dos razones, una que se pudiera realizar control de calidad y pruebas de cada uno sin afectar a las demás y simplificando las correcciones y dos para hacer uso de distintos ambientes dónde las variables que coexisten sólo tienen validez y ocupan memoria mientras se ejecuta el subproceso y una vez que culmina y genera su salida se pierde la información de las variables auxiliares, permitiendo así un considerable ahorro de memoria y por lo tanto de tiempos de cómputo.

Los parámetros de entrada son recibidos por esta función matriz por medio de un archivo plano de formato txt que debe ser llamado “entrada1.txt” que contiene únicamente un encabezado compuesto por la letra “v” y en orden subsiguiente los 23 valores numéricos de las variables que se utilizan para cada simulación. Para

simplicidad de la construcción de este archivo de entradas se programó una interfaz gráfica en Java, que es otro paradigma de programación orientado a objetos en la cual el usuario dispone de botones de ayuda con una descripción de los parámetros que se solicitan y que finalmente se guardan automáticamente en la carpeta donde se encuentre el ejecutable de la aplicación gráfica que es la ruta en la cual se guardarán posteriormente todos los archivos producto de la simulación. Una vez guardado el archivo plano, se acciona el botón de aceptar para recibir la instrucción mediante un mensaje que el usuario debe escribir “ejecutar(ruta)” en la consola el cual es el llamado de la función matriz que ejecuta la secuencia de procesamiento en el código de R. Al accionar nuevamente el botón de aceptar, la interfaz abre directamente la consola de R la cual realiza a su vez la declaración de todo el código de subprocesos y función matriz que se conoce como archivo fuente, carga las 3 librerías que se necesitan y asigna la ruta en la cual se leerá el archivo plano de entrada. Todo esto se realiza sin que el usuario tenga que hacer nada adicional, simplemente escribir en la consola lo descrito anteriormente para dar paso a la ejecución de la simulación.

Las funciones nativas propias de las librerías nombradas anteriormente en lenguaje R empleadas en este trabajo son las siguientes:

- Krige: para la interpolación mediante kriging ordinario. Requiere la librería “gstat”.
- Variogram: para construir los variogramas experimentales. Requiere la librería “gstat”.
- Vgm: para construir los variogramas teóricos. Requiere la librería “gstat”.
- Persp: para realizar los gráficos en 3D. Requiere la librería “graphics” que se carga automáticamente al cargar la librería “gstat”.
- Cmpfun: para compilar funciones directamente y ahorrar tiempos de cómputo. Requiere la librería “compiler”.
- La librería “lattice” es utilizada principalmente para construir los objetos que tienen coordenadas.

Las funciones listadas arriba son las nativas principales empleadas en este trabajo, para mayor información consultar la ayuda de R.

Por otro lado, las funciones programadas es lo que llamamos subprocesos y que a su vez hacen uso de algunas de las funciones nativas nombradas anteriormente.

Por último, hay que tomar en cuenta que las versiones y librerías de R son constantemente actualizadas por los administradores de los servidores, trayendo en algunos casos, cambios en cuanto a funciones nativas o incluso discontinuación de las mismas por lo que si se desea emplear una versión más reciente o más antigua que la empleada para diseñar el algoritmo pudiera esperarse conflictos entre variables, librerías o funciones nativas.

3.4 DESCRIPCIÓN DE LA SECUENCIA DE PROCESAMIENTO

En esta sección se da una breve descripción de la secuencia de procesamiento, los parámetros de entrada que recibe cada función, una explicación del proceso que realiza a partir de ellos y cual es el resultado que arroja una vez finalizado el procesamiento.

La función “model” recibe el parámetro “m” el cual es una variable discreta que puede tomar los valores 1, 2 ó 3 que son etiquetas referidas a las cadenas de caracteres “Sph”, “Exp” y “Gau” respectivamente, los cuales son requeridos posteriormente por otro proceso para elaborar el modelo de correlación espacial, en consecuencia, la salida es dicha cadena de caracteres. Seguidamente se declara la función “i.data” que recibe las coordenadas, el rango máximo del modelo de correlación espacial y el coeficiente de anisotropía para elaborar el mallado de los datos iniciales desde una esquina a la otra y asignarles el valor de la propiedad inicial que es una distribución normal tipificada y finalmente arroja un objeto de R que contiene el valor de las coordenadas y la propiedad inicial. La siguiente función en ser declarada es “malla.c” la cual recibe las coordenadas y la resolución del mallado de salida para construirlo tomando en cuenta un número de puntos igual a la resolución al cuadrado y devuelve la tabla con el valor de las coordenadas del mallado de salida. Luego se declara el subproceso “r.grid.c” que recibe de entradas

las coordenadas y la resolución para tomar el mallado de salida y desordenar las posiciones siguiendo un patrón aleatorio uniforme de manera de generar el recorrido aleatorio que necesita hacer la simulación secuencial gaussiana, por lo tanto devuelve el mallado de salida con las posiciones desordenadas. El siguiente proceso es “rot.grid” el cual recibe las coordenadas, el rango máximo, el coeficiente de anisotropía y el ángulo de rotación para construir los datos iniciales haciendo el llamado de la función previamente declarada “i.data” y rotarla en el ángulo asignado mediante una matriz de rotación de manera que se simplifique el cálculo de los variogramas en las direcciones de interés una vez simulados los datos y es por ello que su salida es el mallado de puntos iniciales rotados a partir del norte en sentido horario. En orden subsiguiente se declara la función “t.coord” que recibe las coordenadas y la tabla de los datos simulados y realiza la transformación de las coordenadas genéricas que emplea el programa a las coordenadas que desea observar el usuario en los resultados finales y por ende devuelve la tabla de coordenadas transformadas a la zona de estudio definida por el usuario. A continuación se declara la función “rd3” el cual recibe rango en porcentaje y lo transforma a distancia para la realización de los cálculos y estimaciones internas de todos los procesos que lo requieren y por lo tanto se declara esta distancia en el ambiente de la función matriz para que pueda ser usado por los demás procesos. Luego se declara la función “tablas” la cual necesita de entradas las coordenadas, la resolución y las cinco variables geoestadísticas para construir el modelo de correlación espacial, hacer el llamado de los puntos iniciales rotados y el mallado de salida para así ejecutar la simulación secuencial gaussiana y a su vez mostrar en la consola de R el número de puntos que ha simulado, una etiqueta booleana que indica en que momento remueve los datos iniciales y mostrar el gráfico de la interpolación que permite observar toda la simulación en tiempo real, en consecuencia, arroja la tabla que contiene los datos residuales originales o estacionarios cuya propiedad se distribuye normal tipificada. Inmediatamente después se llama la librería “compiler” y se declara la función “tablas1” la cual compila la función tablas directamente para así disminuir los tiempos de cómputo en aproximadamente un 30% y por lo tanto optimizar las

ejecuciones y permitir alcanzar más altas resoluciones. Posteriormente se declara la función “plot.kresults”, tomada y modificada de una publicación de Rossiter del 2012 la cual recibe las coordenadas, el valor de la propiedad y una muestra de datos que se deseen graficar en un mapa secundario y devuelve un gráfico que contiene dos mapas con distintas paletas de colores. En orden subsecuente se declara “plot.map” la cual recibe de entradas las coordenadas, la resolución, los 5 parámetros geoestadísticos y las coordenadas genéricas para hacer el llamado de la función “tablas1” y “plot.kresults” para graficar así el mapa de los residuales originales y el mallado de puntos empleado para tal fin lo cual es la salida que arroja al culminar. A continuación se empiezan a declarar los post-procesos, empezando por la función “trend1” la cual recibe la tabla de los datos estacionarios originales y los coeficientes de la superficie de tendencia para hacer uso de la regla de correspondencia del campo escalar general de segundo orden para así construir los datos regionales que es lo que devuelve. Luego se declara la función “reescalar” que necesita de entrada la tabla de los datos estacionarios originales y la desviación estándar de los residuales transformados que desea el usuario para realizar dicha transformación de una normal tipificada a una que se distribuya con media y desviación estándar definidas por el usuario, por lo tanto arroja dicha propiedad transformada. En orden subsiguiente se declara el proceso “muestra.u” que recibe la tabla de los datos no estacionarios y el porcentaje de la muestra que desea obtener el usuario para realizar un muestreo aleatorio uniforme en toda la zona y devolver este subconjunto de datos como salida. A continuación, las siguientes dos funciones en ser declaradas son “muestra.l” y “muestra.f” las cuales son análogas debido a que realizan el mismo proceso con la salvedad de que emplean distintos porcentajes de muestreo por cuartil de manera que se obtenga un efecto de muestreo con sesgo bajo y alto respectivamente, motivo por el cual necesita la tabla de los datos no estacionarios, el porcentaje total de la muestra final y una variable que solo puede valer 1 ó 0 que significa verdadero o falso y a su vez realiza el ordenamiento de los datos “decreciente = verdadero” para sortearlos en orden decreciente y así obtener el efecto de sesgo hacia valores mínimos de la propiedad o “decreciente = falso” para sortearlos en orden creciente y por lo tanto

generar un efecto de sesgo hacia valores máximos de la propiedad y finalmente devuelven el subconjunto de la muestra tomada sesgada. En orden subsiguiente se declara la función “covariable” la cual necesita la tabla de los datos no estacionarios, el coeficiente de correlación, la media y la desviación estándar de la variable auxiliar a ser utilizada en el cálculo de la variable correlacionada mediante la ecuación de correlación de variables y arrojando como salida está nueva propiedad. Luego se declara el subproceso “polar.plot” el cual necesita principalmente los valores de radio y de ángulos a ser graficados en un marco de referencia polar en el cual las cónicas pueden tener el centro en el polo y no necesariamente un foco como se suele utilizar para aplicaciones en cálculo multivariable de variable real y devuelve como resultado dicho gráfico en un marco de referencia polar. En consecuencia se declara el proceso “anis.diag” el cual necesita el rango mayor, el coeficiente de anisotropía y el ángulo de rotación para hacer uso de la función “polar.plot” y de la ecuación cartesiana de una elipse genérica horizontal con centro en el origen llevada posteriormente a coordenadas polares y así finalmente graficar la elipse de anisotropía con la rotación respectiva respecto al norte en sentido horario que es lo que arroja como salida. En orden subsecuente se declara la función “trid.plot” que recibe la tabla de los datos estacionarios, los coeficientes de la superficie de tendencia y dos ángulos adicionales no editables por el usuario que sirven para establecer con cual rotación respecto a los ejes espaciales se desea observar el gráfico del campo escalar que es lo que arroja finalmente este proceso. A continuación se declara la función “cuartiles” que únicamente necesita la tabla de los datos no estacionarios para realizar el seccionamiento por cuartiles y la asignación de una etiqueta a cada valor de la propiedad alusivo al cuartil al cual pertenece de manera que se pueda realizar posteriormente un mapa de cuartiles de la propiedad simulada, dato que es devuelto como salida. Las siguientes dos funciones en ser declaradas son “validar1” y “validar2” que reciben la tabla de los residuales, las coordenadas, la resolución y las variables geoestadísticas para calcular el variograma experimental principal y el secundario respectivamente. En último lugar se declara la función matriz “ejecutar” que recibe la ruta y se encarga de construir la secuencia de procesamiento del

algoritmo de manera ordenada mediante la realización de los llamados de las funciones previamente declaradas. Los procesos se ejecutan en el siguiente orden:

- Lectura de los 23 parámetros de entrada definidos por el usuario.
- Ejecución de la simulación secuencial gaussiana para generar los datos residuales originales.
- Construcción de los variogramas experimentales de mínima y máxima continuidad espacial.
- Generación de los datos residuales.
- Transformación de los residuales.
- Adición de los datos residuales con los regionales para generar los datos no-estacionarios.
- Muestreo de los datos mediante los tres criterios programados.
- Construcción de la variable correlacionada.
- Seccionamiento de los datos no-estacionarios por cuartiles.
- Despliegue del gráfico de la elipse de anisotropía.
- Graficación y escritura de los datos en la ruta indicada.

CAPÍTULO IV

RESULTADOS Y ANÁLISIS

4.1 VALIDACIÓN DE LAS VARIABLES GEOESTADÍSTICAS.

El objetivo general de esta sección consta en validar, mediante una serie de pruebas, el desempeño del algoritmo simulador de datos sintéticos estacionarios bidimensionalmente georeferenciados que cumplen con características estadísticas, geoestadísticas y de visualización establecidas por el usuario. Las pruebas se enfocan en establecer combinaciones de las cinco variables geoestadísticas ajustables de manera que se fijen cuatro de ellas mientras se varía la restante a lo largo del dominio y así comprobar su funcionamiento por separado. Una vez validada la variable, se le asigna cualquier valor perteneciente a su dominio para la siguiente prueba.

A continuación en la tabla 1 se presenta una síntesis de las cinco pruebas mediante una matriz cuadrada cuya diagonal principal representa las variables a validar (en color rojo), los valores superiores a la misma son los fijos iniciales y los inferiores los pertenecientes al dominio de cada variable validada; donde los diminutivos “Exp”, “Gau” y “Sph” son alusivos a los modelos de variograma: exponencial, gaussiano y esférico, respectivamente. Los rangos de los variogramas no pueden exceder el 50% de la longitud de la diagonal principal de la zona de estudio por restricciones matemáticas del sistema de ecuaciones del kriging, razón por la cual este parámetro se expresa en porcentaje. Para las pruebas realizadas, esta longitud de la diagonal principal a un 100% es de 5122.5 metros.

Tabla 1 Descripción de los parámetros empleados para las 5 pruebas de validación de las variables geoestadísticas.

N°	Modelo	Rango (%)	Prop. Anisotropía	Direc. Anisotropía	Nugget
1	Exp - Sph - Gau	27	1	90°	0
2	Exp	16 - 32 - 48	1	90°	0
3	Gau	25	0.3 - 0.5 - 0.7	90°	0
4	Sph	30	0.35	45° - 120° - 160°	0
5	Exp	33	0.4	135°	0 - 0.5 - 0.9

4.1.1 Prueba N°1.

Objetivo: Comparar la continuidad espacial asociada a los modelos de variograma empleados, es decir: Exponencial, esférico, gaussiano.

En primer lugar, se pretende resaltar el grado de variabilidad intrínseca de los modelos matemáticos, asume isotropía, ya que el coeficiente de anisotropía es igual a uno y por lo tanto el rango máximo es equivalente al mínimo. En consecuencia, los mapas muestran una propiedad de formas semicirculares siendo el gaussiano el de mayor continuidad espacial, el exponencial el de mayor variabilidad (dado que alcanza el sill siguiendo la regla de correspondencia más pronunciada) y el esférico el modelo intermedio de los dos anteriores (producto de la linealidad de los puntos contenidos entre el origen y el rango del mismo). Dicho efecto se aprecia en la figura 4.1, donde se muestran tres mapas: exponencial, esférico y gaussiano respectivamente, el primero de los cuales experimenta cambios abruptos de la propiedad, el segundo cambios moderados y el último cambios suaves y transicionales. Es importante destacar que cuando se está bajo escenario isotrópico, como es el caso, la dirección de anisotropía es una variable independiente ya que cualquier valor que tome, mantiene el resultado sin cambios desde el punto de vista teórico, ya que sería el equivalente de rotar un círculo respecto a su centro.

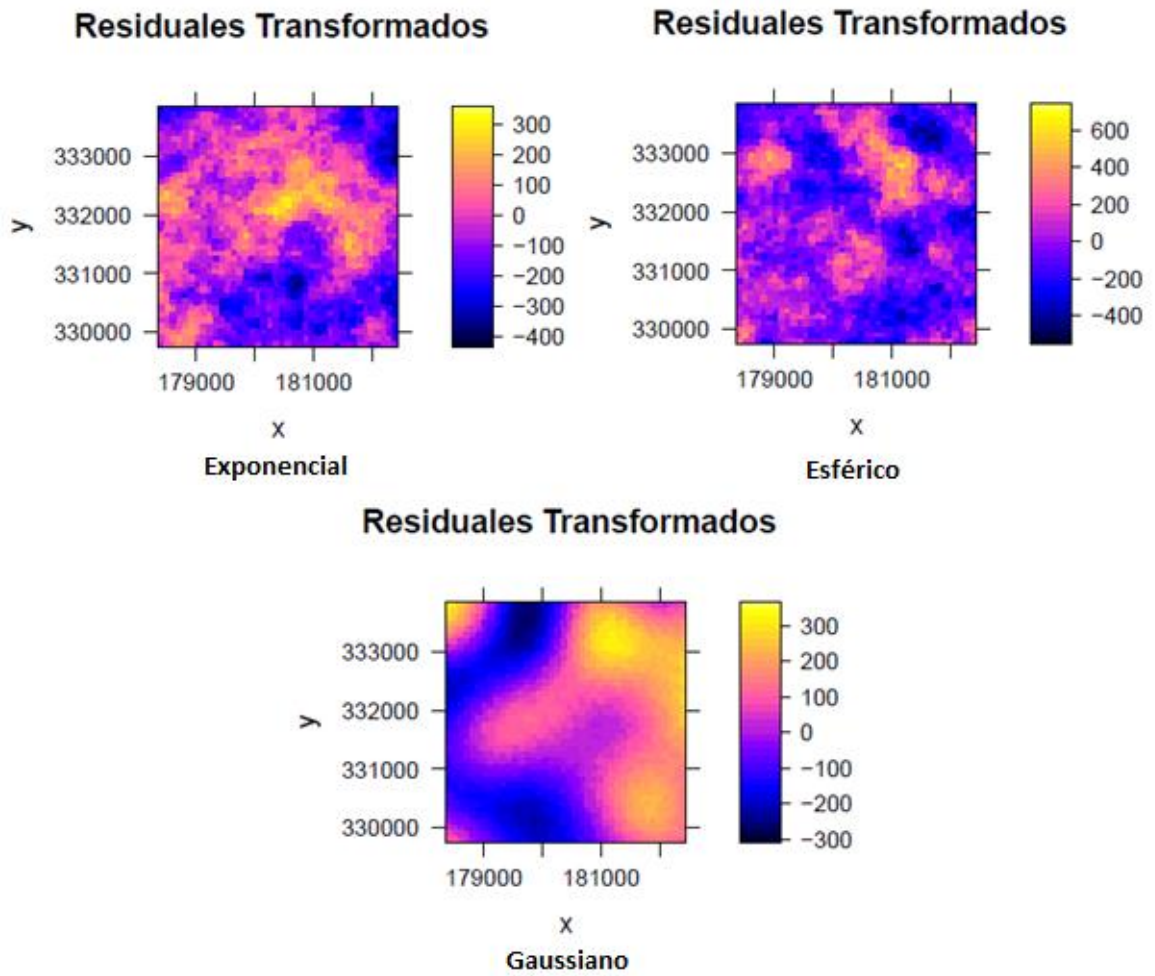


Figura 4.1 Mapas referentes a la prueba N°1.

4.1.2 Prueba N°2.

Objetivo: Comparar el efecto que impone el rango en los mapas bajo escenario isotrópico.

Seguidamente la figura 4.2, se enfoca en el tamaño de los contornos de la propiedad simulada dejando fijo el modelo de variograma para los tres mapas, observándose tamaños de las formas de los cuerpos en proporción directa al valor en porcentaje del rango, aún bajo el escenario isotrópico, ordenados en forma creciente de izquierda a derecha y mostrando la misma tendencia circular apreciada en la prueba anterior debido a que el coeficiente de anisotropía es igual a la unidad, en consecuencia se observa independencia direccional en cuanto al ángulo de anisotropía.

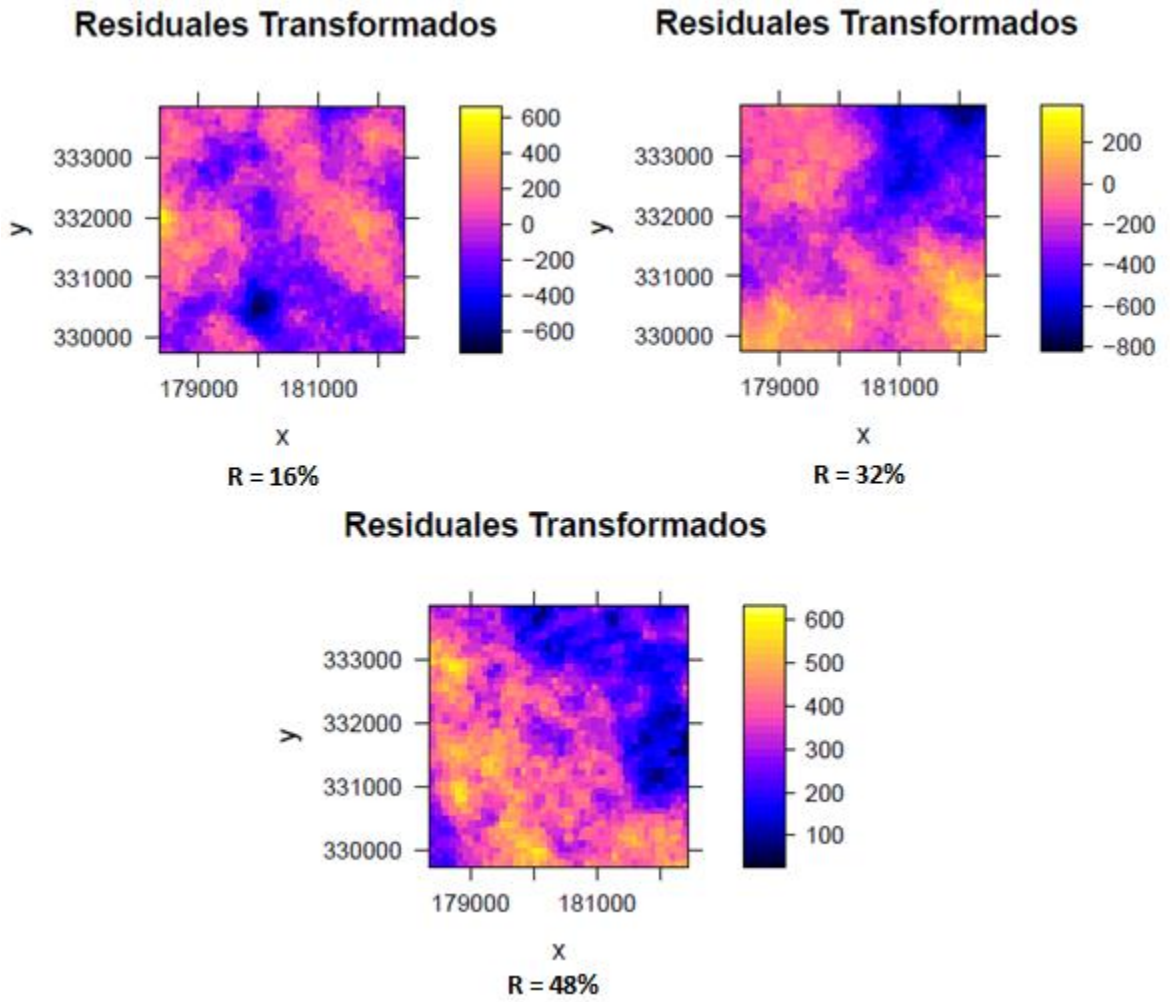


Figura 4.2 Mapas referentes a la prueba N°2, variación de rangos de menor a mayor de izquierda a derecha.

4.1.3 Prueba N°3.

Objetivo: Comparar los mapas a distintas tasas de anisotropía.

En esta prueba se varía la proporción de anisotropía, observando que la dirección de máxima continuidad espacial es la horizontal, que para efectos de la variable “dirección de anisotropía” representa 90° medidos en sentido horario a partir del norte. Dicha prueba aborda el modelo anisotrópico a distintas proporciones, generando formas elípticas en función a las mismas como se enfatiza en la figura 4.3.

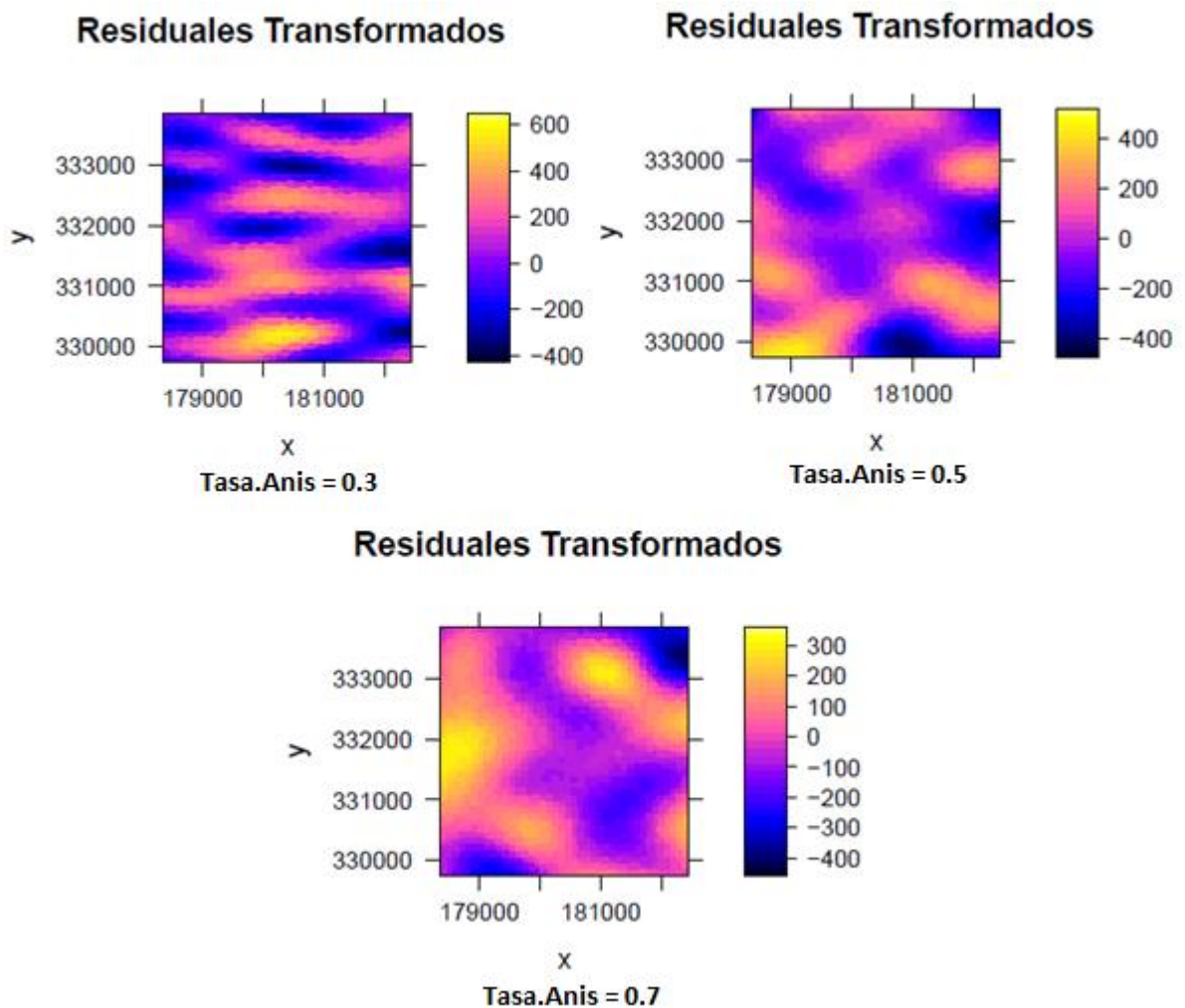


Figura 4.3 Mapas referentes a la prueba N°3, variación de la tasa de anisotropía de menor a mayor de izquierda a derecha.

4.1.4 Prueba N°4.

Objetivo: Comparar la dirección de anisotropía a lo largo de su dominio.

La finalidad de la actual prueba radica en validar la variable “dirección de anisotropía”, fin que se comprueba a simple vista en la figura 4.4, cuyos mapas describen una tendencia de los contornos de la propiedad a ser más alargada en la dirección de mayor correlación espacial y menos alargada en la perpendicular común, siendo fieles representantes de sus elipses de anisotropía. El ángulo de rotación está medido en grados a partir del norte en sentido horario y para este caso en particular la tasa de anisotropía vale 0.35 motivo por el cual se observa la anisotropía de manera enfática. Es importante notar que el dominio eficiente de la dirección de anisotropía es de 0 grados hasta 180 grados ya que ángulos superiores siempre tendrán un equivalente dentro de los dos cuadrantes del dominio, sin embargo, fueron considerados ángulos de hasta 720 grados en la programación del algoritmo para albergar cierto rango de exclusiones.

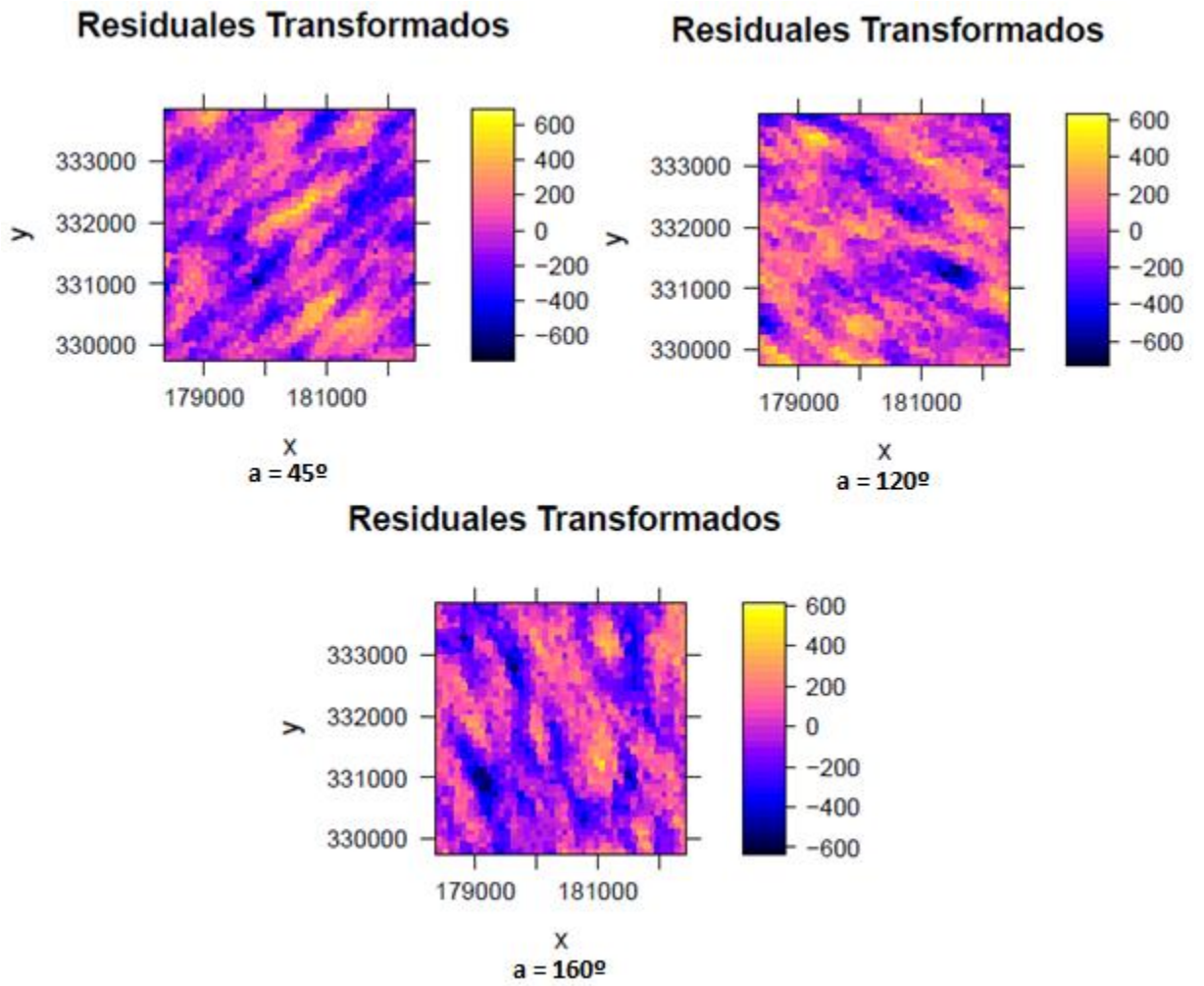


Figura 4.4 Mapas referentes a la prueba N°4, variación de la dirección de anisotropía.

4.1.5 Prueba N°5.

Objetivo: Comparar la intensidad del efecto nugget en los distintos mapas.

Finalmente, el último experimento consistió en comprobar la distorsión de la imagen que impone el nugget del modelo matemático, estableciendo una dirección de máxima continuidad espacial de 135° la cual se observa en los tres mapas a pesar de su visualización ruidosa. La figura 4.5 señala como el efecto nugget“ pasa de mínimo a máximo, de izquierda a derecha, respetando los restantes cuatro parámetros geoestadísticos.

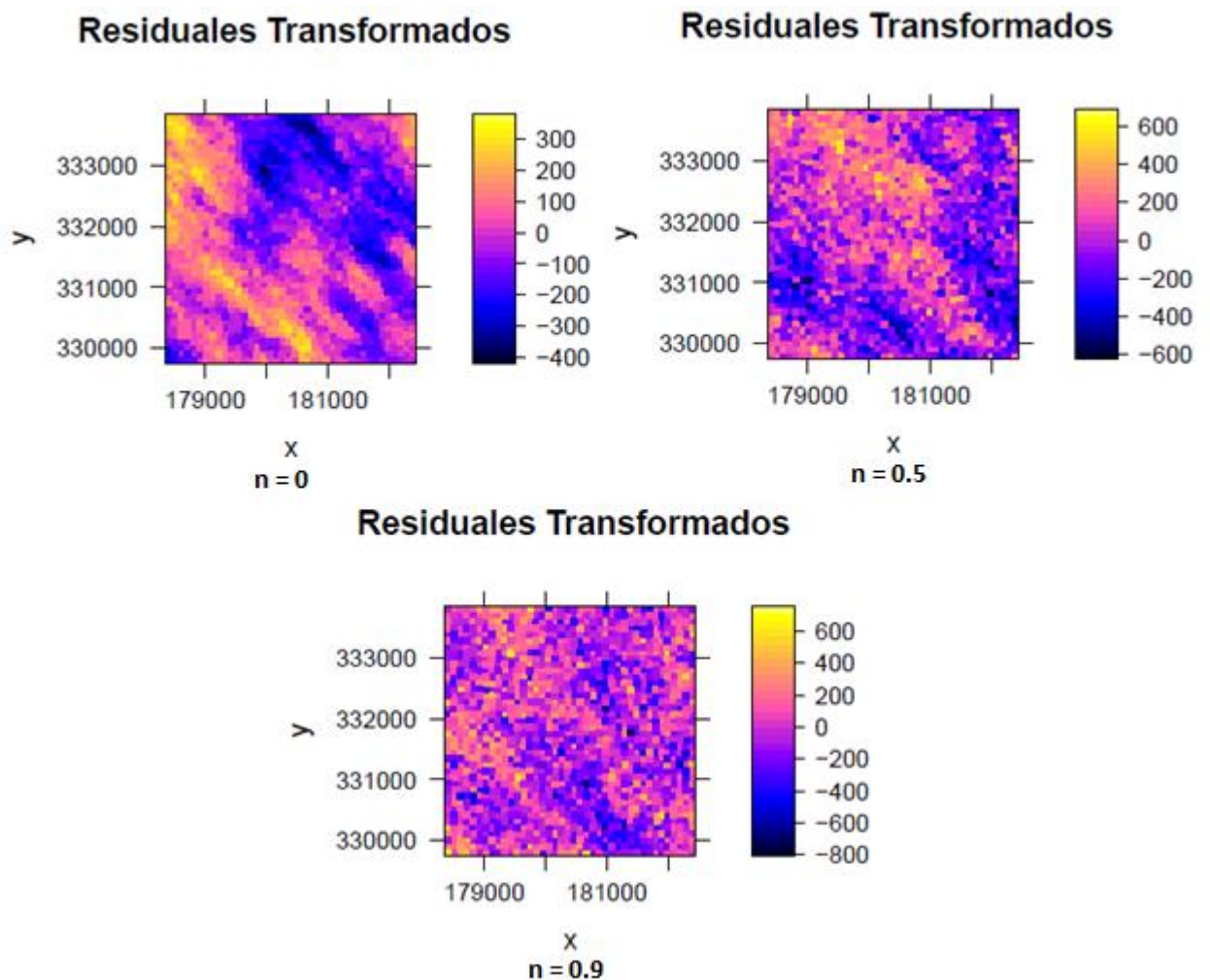


Figura 4.5 Mapas referentes a la prueba N°5, variación del nugget de mínimo a máximo de izquierda a derecha.

4.2 COMPROBACIÓN DE LA CONVERGENCIA DE LOS MODELOS.

4.2.1 Variogramas de las 30 simulaciones

A continuación se muestran los resultados de la simulación estocástica de 30 conjuntos de datos representados por el variograma de máxima continuidad espacial y el de menor continuidad espacial (en su dirección ortogonal), dado al escenario anisotrópico. Los mismos son graficados en un mismo marco de referencia para inspeccionar visualmente la variación que pueden experimentar dentro de un mismo ambiente de simulaciones estocásticas en las direcciones de máxima y mínima continuidad espacial. Es notorio en las figuras 4.6 y 4.8, que a medida que los puntos se alejan del origen del variograma experimentan mayor varianza, esto es debido a que se emplean para su cálculo menor cantidad de pares de puntos y que además se encuentran a mayores distancias, motivo por el cual presentan menor correlación espacial. Sin embargo, dicha varianza permanece controlada para los lags que se encuentran a longitudes inferiores al rango, que para el modelo principal¹ tiene un valor de 1000 y para el secundario² de 500 metros como se muestra en la tabla 2.

¹ Modelo principal: Se considera el modelo principal el modelo de variograma que se calcula en la dirección principal de anisotropía o de rango máximo de la elipse.

² Modelo secundario: Se considera el modelo secundario aquel modelo de variograma que se calcula en la dirección ortogonal al modelo principal y que a su vez coincide con el rango mínimo de la elipse.

Tabla 2 Descripción de los parámetros de ejecución de la corrida de las 30 realizaciones.

Parámetro	Valor
Realizaciones	30
Modelo	Gaussiano
Rango (m)	1000
Prop.	0,5
Anisotropia	
Direc.	60º
Anisotropia	
Nugget	0,023

En la figura 4.6 se muestra el gráfico de los variogramas de la dirección de máxima continuidad espacial de la simulación estocástica de las 30 realizaciones bajo un mismo marco de referencia para así observar la movilidad que podría presentar una realización particular en diferencia al modelo inicial solicitado.

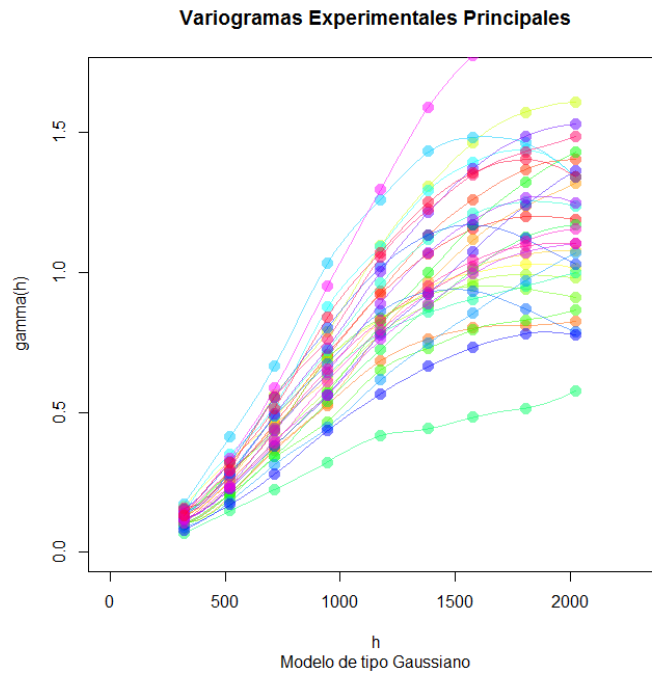


Figura 4.6 Variogramas experimentales principales de las 30 realizaciones.

En la figura 4.7 se observa el diagrama de caja y bigote de los 30 variogramas principales, señalando en color rojo la media de las realizaciones de cada lag del modelo de correlación espacial y en verde su mediana. Lo interesante de esta prueba de validación es que una vez que se supera el valor del rango teórico, que es 1000 metros, se empiezan a separar los puntos rojos de los verdes así como también se alargan los bigotes y las cajas de los diagramas señalando con esto que se ha perdido la correlación espacial entre los puntos ubicados en tal dirección y a distancias superiores al rango de manera que se observa el fenómeno de independencia espacial.

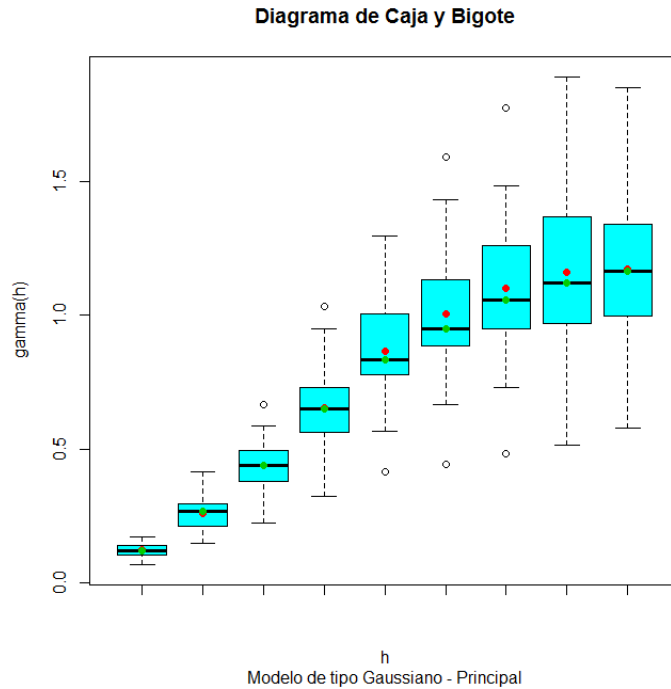


Figura 4.7 Diagrama de caja y bigote de los variogramas secundarios de las 30 realizaciones, variograma promedio y mediano.

Posteriormente, en la figura 4.8 se observan los 30 variogramas experimentales del semieje menor de la elipse de anisotropía, o lo que es lo mismo, la dirección de menor continuidad espacial. Se grafican todos bajo un mismo marco de referencia de manera de observar la movilidad que puede tener el modelo experimental de una realización específica respecto al modelo teórico en la dirección antes mencionada.

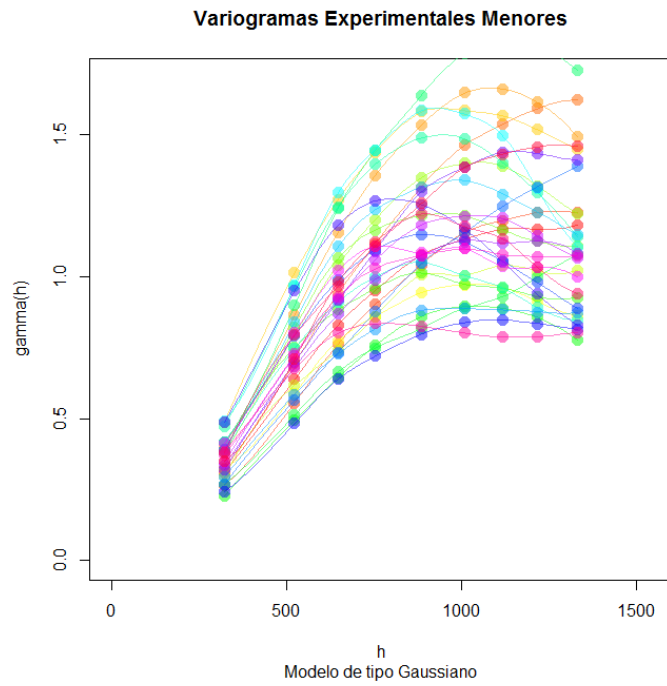


Figura 4.8 Variogramas experimentales secundarios de las 30 realizaciones.

En último lugar se tiene el diagrama de caja y bigote por lag de variograma de la dirección de menor continuidad espacial junto con el variograma medio y mediano en la figura 4.9. Se observa una separación de los puntos rojos y verdes a partir de una distancia de 500 metros y un aumento de la varianza de las distancias superiores al rango tal como se observó a su vez en el modelo de la dirección principal.

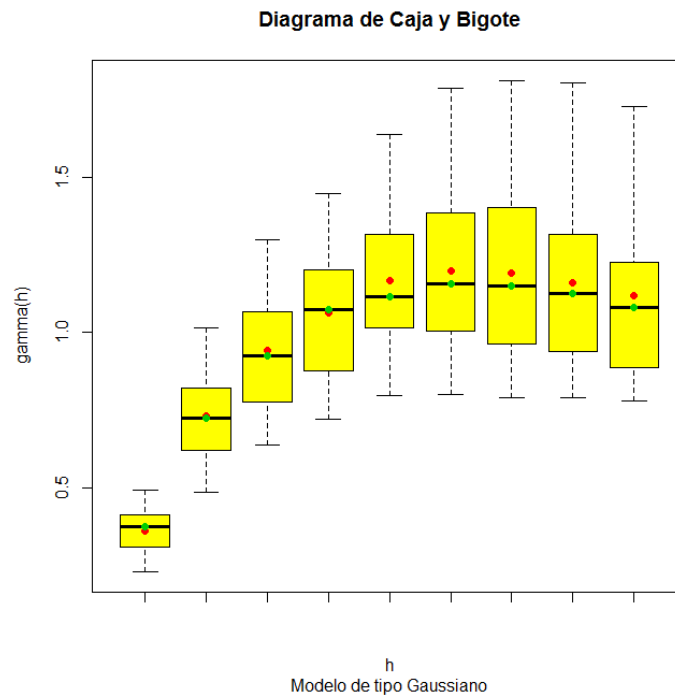


Figura 4. 9 Diagrama de caja y bigote de los variogramas secundarios de las 30 realizaciones, variograma promedio y mediano.

4.2.2 Diagramas de caja y bigote de las 30 simulaciones

Seguidamente en las figuras 4.7 y 4.9, se presentan los mismos 30 variogramas mostrados en las dos figuras anteriores (4.6 y 4.8), esta vez representados mediante los diagramas de caja y bigote de cada lag, su variograma medio y mediano. Se observa una separación de la media y la mediana para distancias que exceden el valor del rango lo cual es efecto de la varianza producto de la pérdida de correlación espacial o independencia de los datos. Además se aprecia la convergencia del variograma promedio experimental al modelo teórico inicial el cual en ambos casos presenta un sill unitario y nugget de 0.023, mientras que el rango principal tiene un valor de 1000 y el secundario de 500 unidades de distancia, parámetros que honran dicho modelo teórico. Aquellos puntos que se observan en la figura 4.7 son los extremos que se salen del intervalo de confianza del diagrama con una probabilidad de ocurrencia aproximadamente de 5%.

4.2.3 Ajuste del variograma promedio experimental al modelo teórico

A continuación, en la figura 4.10, se muestra el variograma experimental promedio de las 30 realizaciones junto con su variograma teórico para comprobar su ajuste visualmente para la dirección de máxima continuidad espacial y su perpendicular común referente al de menor continuidad. En el variograma principal se observa un buen ajuste hasta 995 metros, de ese punto en adelante se aprecia el efecto de la independencia espacial. En cuanto al modelo de menor continuidad espacial se aprecia el mismo efecto que en el anterior con la salvedad de que ocurre a partir de 490 metros aproximadamente. Al observar los valores usados de entrada a las simulaciones en la tabla 2, se verifica que el rango máximo solicitado es equivalente a 1000 metros y el menor de 500 metros, por lo tanto se valida la convergencia del variograma experimental al teórico.

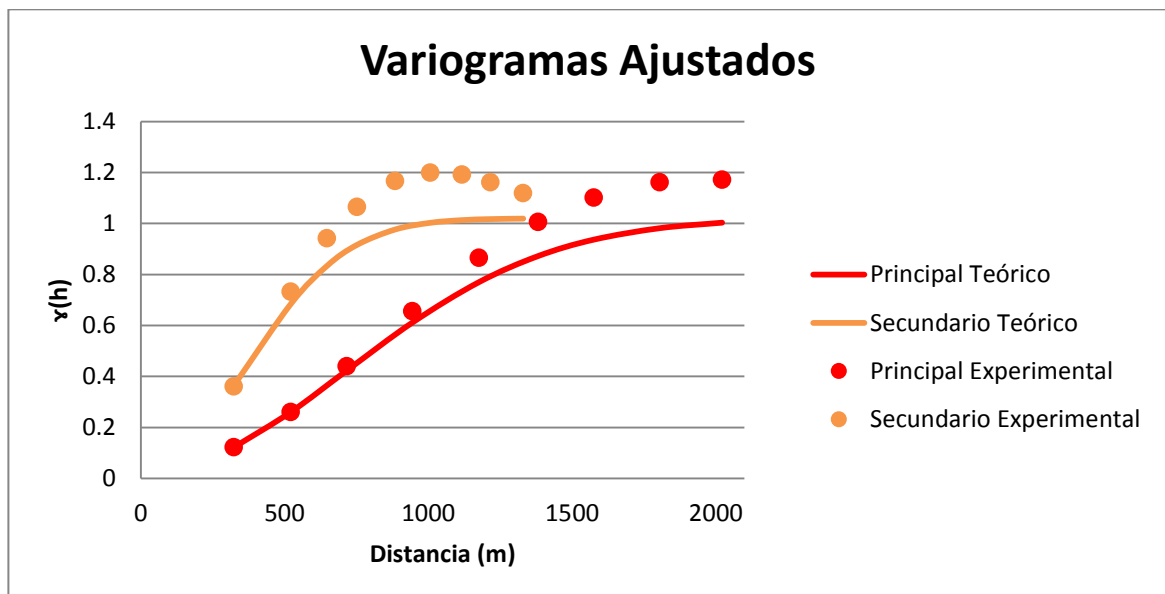


Figura 4.10 Variogramas promedio de máxima y mínima continuidad espacial de las 30 realizaciones.

4.3 DESCRIPCIÓN DE LOS POST-PROCESOS

Esta sección está dedicada única y exclusivamente a describir los post-procesos programados como elementos adicionales a la simulación de los datos estacionarios.

Para ello se generó una nueva ejecución partiendo del conjunto de parámetros de entrada expuesto en la tabla 3, donde la figura 4.11 muestra los residuales originales cuya propiedad se distribuye normal tipificada y que cumple con los parámetros geoestadísticos de entrada. La figura 4.12 señala la transformación de los residuales a una distribución normal definida por el usuario que para este caso tiene media igual a cero y desviación estándar de 220 tal como lo indica la tabla 3.

Tabla 3 Descripción de los parámetros empleados para la nueva simulación referente a los post-procesos.

Sección	Parámetros	Valor
Parámetros para la simulación de los datos estacionarios	x1	177400
	y1	333800
	x2	181600
	y2	329600
	Resolución	50
	Rango Máximo	20
	Tasa de Anisotropía	0.4
	Modelo de Variograma	2
	Ángulo de Rotación	30
	Nugget	0
Superficie de tendencia	a	0.0001
	b	0
	c	0.0001
	d	0
	e	0
Transformación y muestreo	Media Residuales	0
	Desviación Estándar de los Residuales	220
	Porcentaje de Muestreo	15
	Sesgo Bajo	0
	Sesgo Alto	1
	Coefficiente de Correlación	-0.85
	Media Auxiliar Correlacionada	6000
Desviación Estándar Auxiliar Correlacionada	60	

Residuales Originales

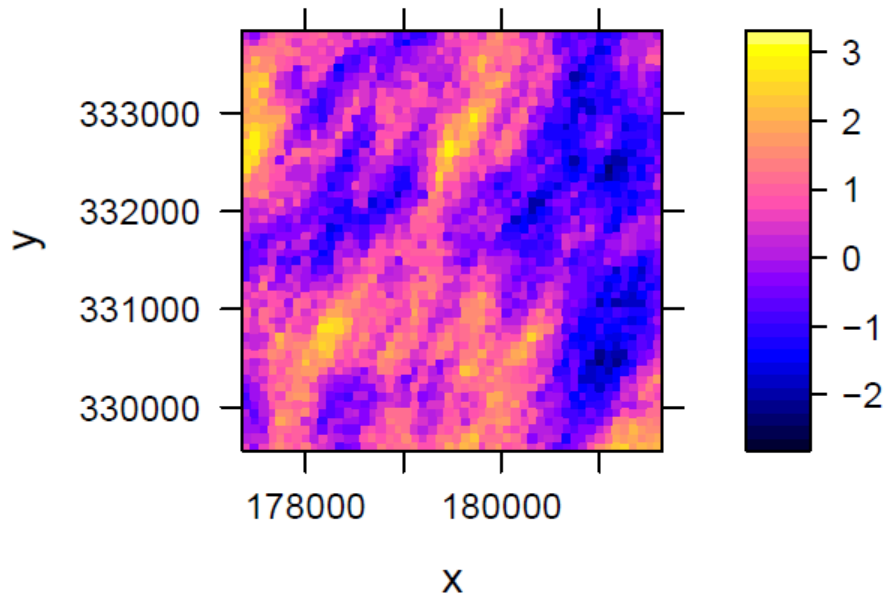


Figura 4.11 Residuales originales o datos estacionarios simulados.

La diferencia principal entre la figura 4.11 y la figura 4.12 es la escala del valor de la propiedad simulada estacionaria que inicialmente se distribuye normal tipificada y posteriormente con una media y varianza definidos por el usuario, sin embargo, se cumple la estacionariedad de los datos en ambos casos ya que los mismos oscilan alrededor de una media constante y además por tratarse de distribuciones normales satisfacen el principio de intervalo de confianza con 99% de seguridad de estar contenidos en un círculo cuyo radio es igual a la media mas tres veces la desviación estándar de dichos datos.

Residuales Transformados

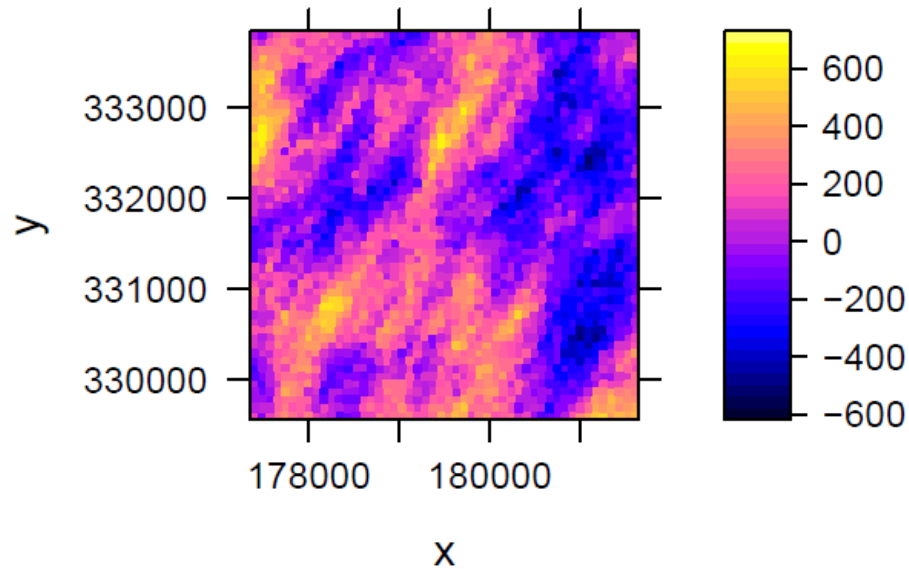


Figura 4.12 Residuales transformados.

Posteriormente se construye la superficie de tendencia a partir de los coeficientes del polinomio de segundo orden obtenidos de la tabla 3 y se le añade a los residuales transformados para obtener el mapa residual mas regional que se muestra en la figura 4.14 el cual presenta las características de los residuales pero sigue la tendencia de aumentar radialmente desde el centro del mapa hacia los bordes debido a la forma de la superficie de tendencia de la figura 4.13

Superficie de Tendencia

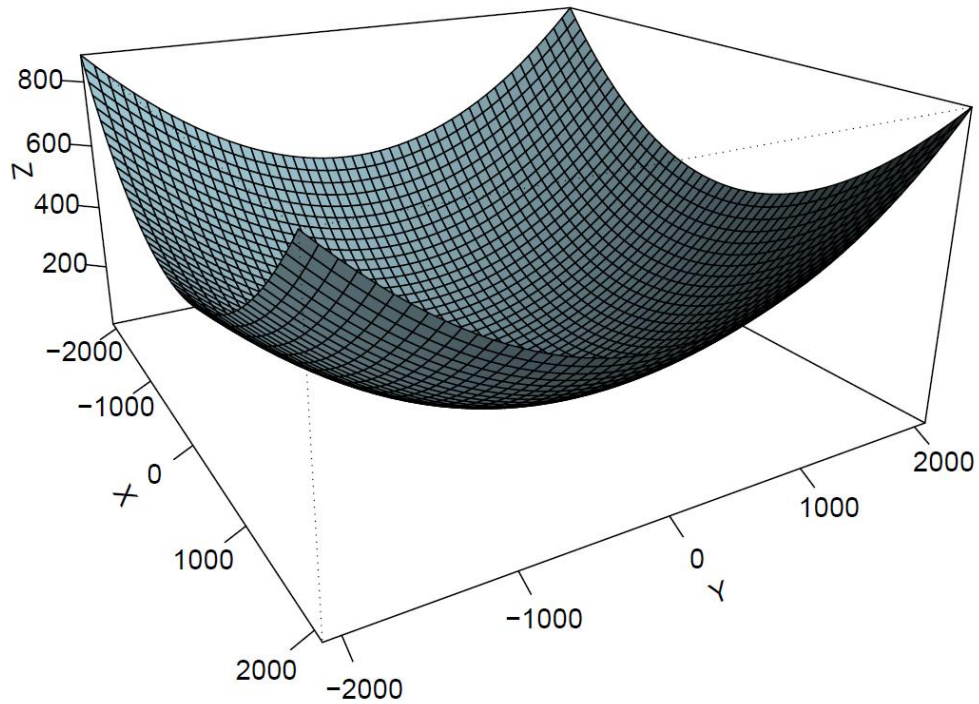


Figura 4.13 Superficie de tendencia.

Una vez que se dispone de los datos estacionarios transformados y de una tendencia definida por su regla de correspondencia y coordenadas, se calculan los datos no-estacionarios como la adición de los residuales y los regionales, obteniendo así los datos que se muestran en la figura 4.14.

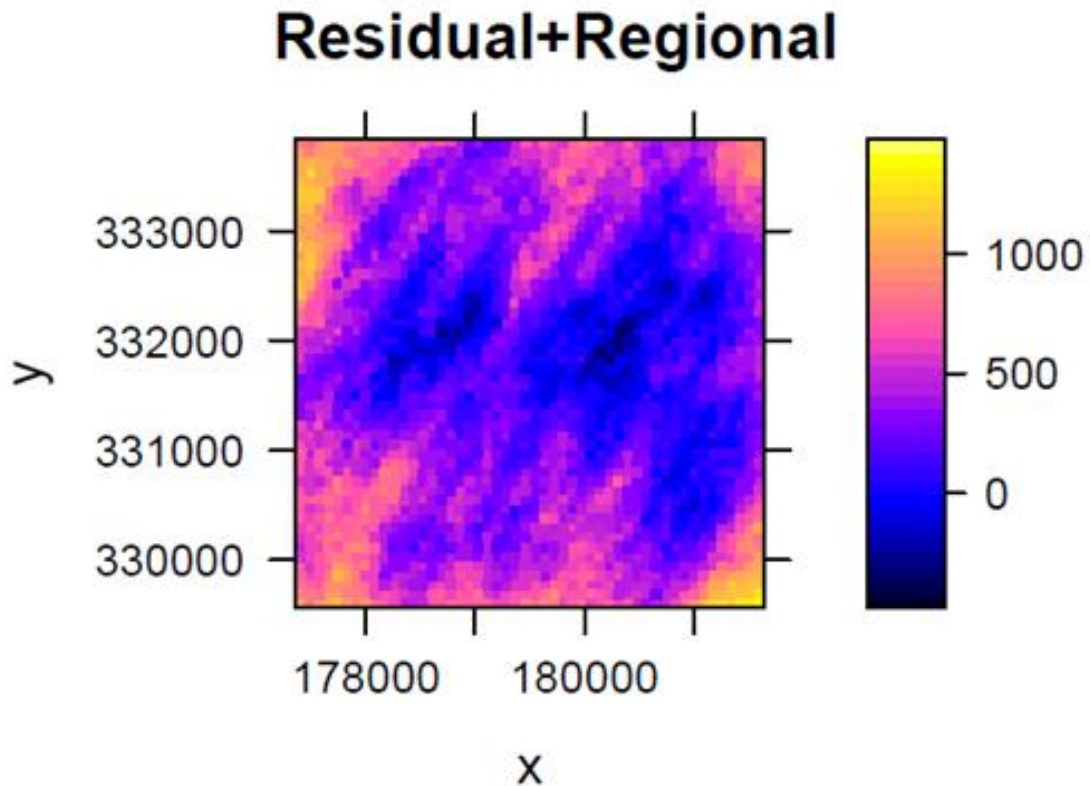


Figura 4.14 Mapa residual mas regional.

A continuación se procede a construir una variable correlacionada con la residual-regional simulada utilizando el coeficiente de correlación indicado por el usuario en la tabla 3, que para este caso tiene un valor de -0.85 , y a partir de una variable auxiliar cuya media y desviación estándar son también indicados por el usuario, particularmente 6000 y 60 respectivamente tal como lo señala dicha tabla. Esta variable correlacionada se grafica en función de la simulada en la figura 4.15 y se observa una marcada relación de proporción inversa señalando a dispersión, producto de la varianza de los puntos y debido al valor del coeficiente de correlación. En otras palabras, si el coeficiente de correlación hubiese sido de -1 todos los puntos se alinearían sobre la recta de tendencia.

Correlación de Variables

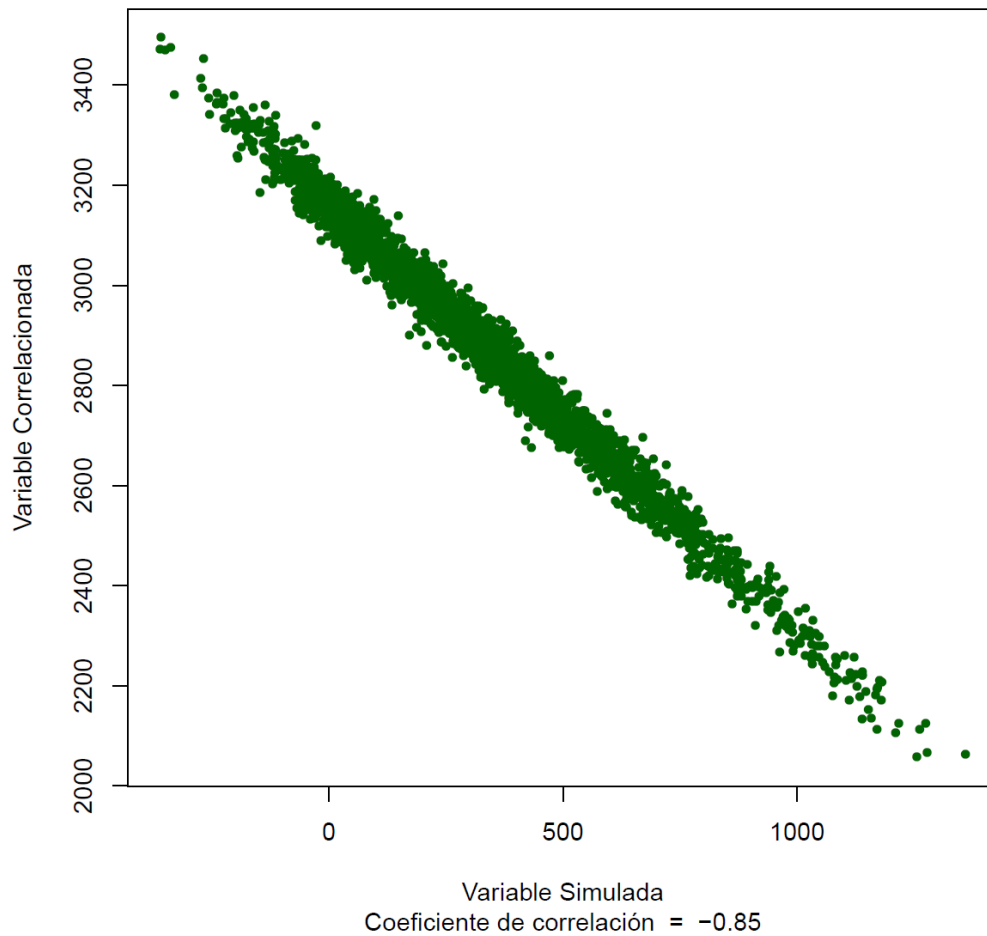


Figura 4.15 Crossplot de la variable simulada con la correlacionada.

Finalmente, se observa el desempeño de la función tomadora de muestras la cual puede utilizar 3 criterios de búsqueda que son: con sesgo alto o bajo hacia valores mínimos o máximos, o mediante un patrón aleatorio uniforme indiferente del valor de la propiedad. Este muestreo se realiza sobre el mapa de cuartiles el cual separa en 4 sectores la propiedad como se observa en las figuras: 4.16, 4.17 y 4.18. En la figura 4.16 se presenta el mapa de cuartiles con una muestra de 15 % con sesgo alto hacia valores mínimos; la figura 4.17 señala una muestra de la misma magnitud con sesgo bajo hacia valores máximos de la propiedad y en la figura 4.18 se observa un muestreo uniforme también de 15%.

Cuartiles - Sesgo Alto

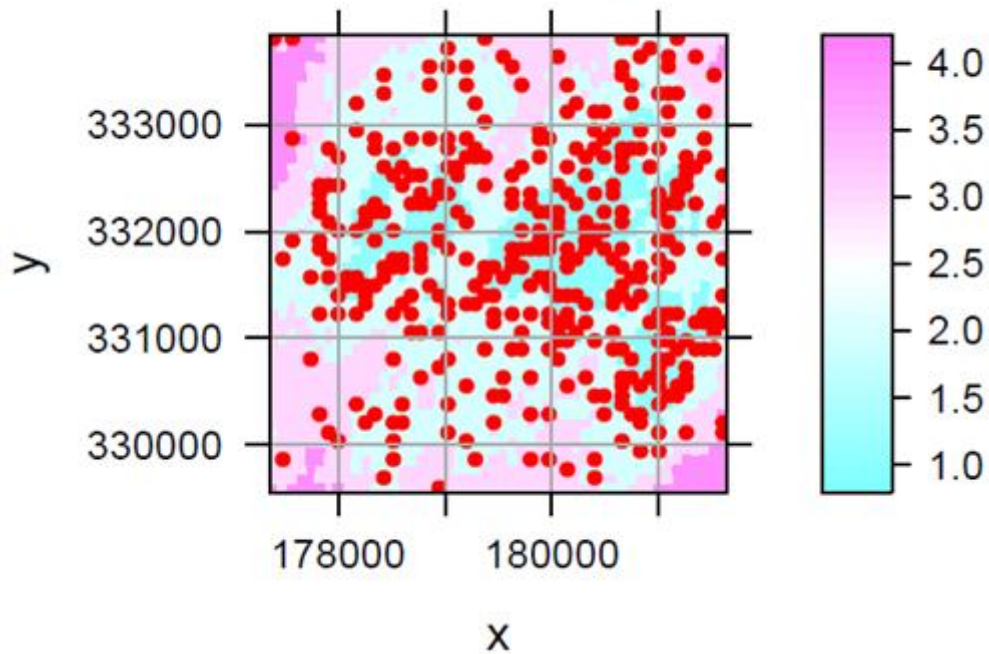


Figura 4.16 Mapa de cuartiles de la muestra con sesgo alto hacia valores mínimos.

Es importante recalcar que el sesgo como tal es definido por el porcentaje de muestreo que se programó para ser tomado de cada cuartil y que la tendencia del sesgo hacia los valores máximos o mínimos de la propiedad depende directamente de si los mismos son ordenados de manera creciente o decreciente respectivamente. Por ejemplo, en la figura 4.16 se observa con claridad una mayor densidad de puntos concentrados hacia las zonas del mapa de colores azules claros que son representantes de los primeros cuartiles y además una menor proporción de puntos distribuidos en los cuartiles superiores de manera que se puede decir que el patrón de muestreo utilizado es de sesgo alto hacia valores mínimos de la propiedad simulada.

Cuartiles – Sesgo Bajo

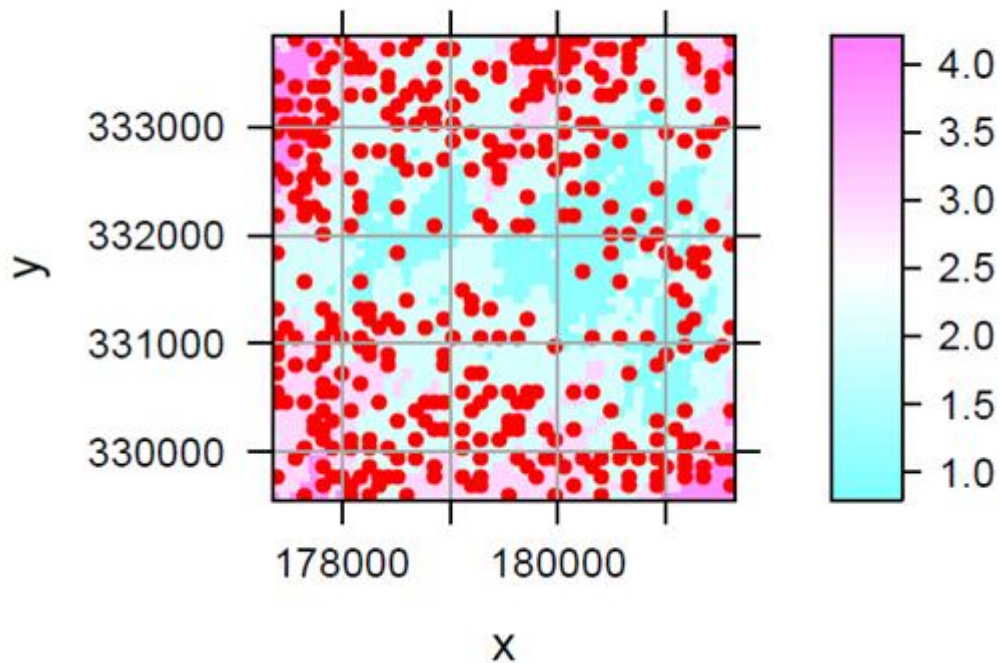


Figura 4.17 Mapa de cuartiles de la muestra con sesgo bajo hacia valores máximos.

De manera análoga se programó una función de toma de muestra con sesgo bajo considerando una diferencia de tan solo 10% por cuartil, es decir: 10, 20, 30 y 40 % para los cuartiles 1, 2, 3 y 4 respectivamente lo que permite un aumento de proporción más suave entre cuartiles y por lo tanto una tendencia de sesgo bajo en cuanto al muestreo. Además presenta la opción de ordenar los datos de manera creciente o decreciente para sesgar hacia valores máximos o mínimos. Por ejemplo, la figura 4.17 evidencia una alta densidad de puntos concentrados en el 4to y 3er cuartil y disminuyendo gradualmente esta densidad hasta llegar al primer cuartil que a pesar de contener la menor cantidad de puntos, lo hace en mayor magnitud que la función de sesgo alto como se aprecia en la figura 4.16, en consecuencia, se puede afirmar que la figura 4.17 presenta sesgo bajo hacia valores máximos de la propiedad.

Cuartiles – Uniforme

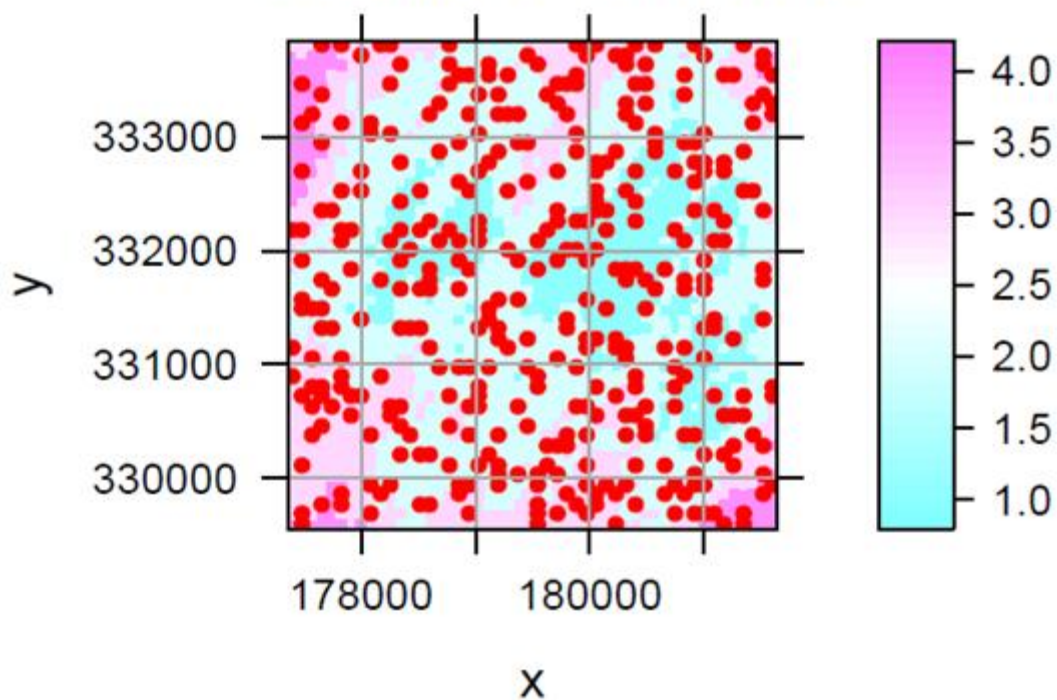


Figura 4.18 Mapa de cuartiles de la muestra con patrón aleatorio uniforme.

Finalmente tomando una muestra equitativamente distribuida por cuartiles se obtiene una distribución uniforme de puntos en todo el mapa con ningún criterio de sesgos condicional en la propiedad simulada tal como lo demuestra la figura 4.18.

CONCLUSIONES Y RECOMENDACIONES.

La herramienta computacional diseñada se validó mediante diversos controles de calidad, permitiendo finalmente simular conjuntos de datos sintéticos bidimensionalmente georeferenciados que cumplen con características y patrones asignados por el usuario, además, esquematizarlos mediante mapas y variogramas.

La función programada para tomar muestras de los datos sintéticos simulados mediante distintos criterios abre la puerta a una línea de investigación en cuanto a reconstrucción de dichos datos mediante variados métodos de interpolación. Del mismo modo, pudiera ser usada para realizar comparaciones y analogías con conjuntos de datos reales que se distribuyan en dos dimensiones.

Algunos post-procesos son adicionales a los objetivos de este trabajo, sin embargo, los complementan y le dan un valor agregado a los resultados desde el punto de vista geoestadístico.

Se recomienda optimizar el código diseñado en este trabajo para expandir sus grados de libertad a tres dimensiones, de manera que se pueda realizar simulaciones de propiedades espaciales tales como la distribución de porosidades de un yacimiento petrolífero, e inclusive abrir la opción de incluir datos de pozos reales para generar la interpolación y en consecuencia un modelo tridimensional del mismo.

Se sugiere complementar la herramienta con variables geológicas que puedan ser consideradas como referencia dentro de los lineamientos del algoritmo y así generar resultados sesgados a restricciones geológicas.

REFERENCIAS BIBLIOGRÁFICAS

Chiles, J.P. & P. Delfiner. (1999). *Geostatistics: Modeling Spatial Uncertainty*. New York: Wiley and Sons, p. 57-79, 128-203.

Davis, J. (1986). *Statistics and Data Analysis in Geology*, Second Edition, Jhon Wiley and Sons, New York, p. 81,83,94,112-119.

Deutsch, C. (2002). *Geostatistical Reservoir Modeling*. Oxford University Press.

Dubrule, O. (2003). *Geostatistics for Seismic Data Integration in Earth Models*. Society of Exploration Geophysics (S.E.G.).

Institute for Statistics and Mathematics of the WU Wien, 2012. Function environments. November 30, 2012. <http://cran.r-project.org/>

Institute for Statistics and Mathematics of the WU Wien, 2012. Source code. November 30, 2012. <http://cran.r-project.org/>

Institute for Statistics and Mathematics of the WU Wien, 2012. Running file. November 30, 2012. <http://cran.r-project.org/>

Jimba, O. (2008). *Basics Geostatistics. Compilation Guide for Students*.

Mendenhall, W. (1971). *Introduction to Probability and Statistics*. 3d. ed., Wadsworth, Belmont, p. 21-36, 64, 68, 100.

Olea, R. A. (1991). *Geostatistical Glossary and Multilingual Dictionary*, Oxford University Press, New York.

Rossiter, D. G. (2012) Co-Kriging with the gstat package of the R environment for statistical computing, University of Twente, Faculty of Geo-Information Science & Earth, Netherland.

ANEXOS

A continuación se muestra el código fuente del algoritmo creado en lenguaje

R:

```
### Version de R recomendada R version 2.13.2
### Nota: Hay que cargar las librerias "lattice" y "gstat"
###Funciones
library(lattice)
library(gstat)

model <- function(m) #modelo de variograma teorico
{
  if (m == 1)
    x <- quote(Sph)
  else if (m == 2)
    x <- quote(Exp)
  else if (m == 3)
    x <- quote(Gau)
  return(as.character(x))
}

i.data <- function(fx,fy,lx,ly,irange,lambda) #Datos de entrada con
anisotropia y sin rotacion
{
  e.zone <- abs(abs(fx-lx) - abs(fy-ly))/2
  if (abs(fx-lx) >= abs(fy-ly))
  {
    fy <- abs(fy + e.zone)
    ly <- abs(ly - e.zone)
  }
  else {
    if (abs(fx-lx) < abs(fy-ly))
    {
      fx <- abs(fx - e.zone)
      lx <- abs(lx + e.zone)
    }
  }
}

x0 <- fx + irange/2
y0 <- fy - irange*(lambda/2)

xf <- lx - irange/2
yf <- ly + irange*(lambda/2)

xdat <- seq(from = x0, by = irange, to = xf)
ydat <- seq(from = y0, by = (-1)*(irange*lambda), to = yf)
xdat1 <- rep(xdat, length(unique(xdat)) )
ydat1 <- rep(ydat, each= length(unique(xdat)))
```

```

i.coord <- as.data.frame(cbind(xdat1, ydat1))
names(i.coord) <- c("x","y")

prop <- rnorm(nrow(i.coord))
xyztable <- cbind(i.coord, prop)
coordinates(xyztable) <- ~x+y
return(xyztable)
}

malla.c <- function(fx,fy,lx,ly,res) #Mallado de salida
{
res <- res -1
if (res > 0)
{
n <- max(abs(fx-lx), abs(fy-ly))/res
e.zone <- abs(abs(fx-lx) - abs(fy-ly))/2
}
else {
print("The number you enter must be higher than zero (0)")
break}

if (abs(fx-lx) >= abs(fy-ly))
{
fy <- abs(fy + e.zone)
ly <- abs(ly - e.zone)
}
else {
if (abs(fx-lx) < abs(fy-ly))
{
fx <- abs(fx - e.zone)
lx <- abs(lx + e.zone)
}}

xdat <- seq(from = fx, by = n, to = lx)
ydat <- seq(from = fy, by = -n, to = ly)

xdat1 <- rep(xdat, res + 1)
ydat1 <- rep(ydat, each = res +1)

cbind(xdat1,ydat1)
}

r.grid.c <- function(fx,fy,lx,ly,res) #Funcion que desordena las posiciones
del mallado para introducir aleatoriedad de la S.G.S.
{
ng1<-malla.c(fx,fy,lx,ly,res)
idx01<-1:nrow(ng1)
idx11<-sample(idx01,length(idx01), FALSE)
ng.g1<- as.data.frame(ng1[idx11,])

ng.g1<-cbind(ng.g1, idx11)
names(ng.g1)<- c("x", "y", "id")
coordinates(ng.g1) <- ~ x + y

```

```

return(ng.g1)
}

rot.grid <- funcion(fx,fy,lx,ly,irange,lambda, rotangle) #Funcion que rota
los datos de entrada generados por i.data
{
a <- (abs(fx-lx))*0.35
b <- (abs(fy-ly))*0.35
fx <- fx - a
fy <- fy + b
lx <- lx + a
ly <- ly - b

i.grid <- i.data(fx,fy,lx,ly,irange,lambda)
aux <- data.frame()
rotcoord <- matrix(nrow = length(i.grid), ncol = 2)

x <- (coordinates(i.grid)[,1] - mean(coordinates(i.grid)[,1]))
y <- (coordinates(i.grid)[,2] - mean(coordinates(i.grid)[,2]))
aux <- as.data.frame(cbind(x, y, i.grid$prop))
names(aux) <- c("x", "y", "prop")
coordinates(aux) <- ~x+y

rotmat <- matrix(nrow= 2, ncol =2)
rotmat[1,1] <- cos(rotangle)
rotmat[1,2] <- sin(rotangle)
rotmat[2,1] <- (-1)*sin(rotangle)
rotmat[2,2] <- cos(rotangle)

for (k in 1:length(i.grid))
{
pxy <- matrix(ncol = 2, nrow = 1)
pxy[1,1] <- coordinates(aux)[k,1]
pxy[1,2] <- coordinates(aux)[k,2]
rotcoord[k,] <- pxy%%rotmat
}

rotcoord <- as.data.frame(cbind(rotcoord,i.grid$prop))
names(rotcoord) <- c("x", "y", "prop")
coordinates(rotcoord) <- ~x+y

x <- (coordinates(rotcoord)[,1] + mean(coordinates(i.grid)[,1]))
y <- (coordinates(rotcoord)[,2] + mean(coordinates(i.grid)[,2]))
rotcoord <- as.data.frame(cbind(x, y, i.grid$prop))
names(rotcoord) <- c("x", "y", "prop")
coordinates(rotcoord) <- ~x+y

return(rotcoord)
}

t.coord <- funcion(x1,y1,x2,y2,tabela) #Transforma las coordenadas
{
x <- coordinates(tabela)[,1]

```

```

y <- coordinates(tabla)[,2]

min.x <- min(x)
max.x <- max(x)
min.y <- min(y)
max.y <- max(y)

x0 <- (x - min.x)*min.x/max.x
y0 <- (y - min.y)*min.y/max.y

x0 <- x0/max(x0)
y0 <- y0 /max(y0)

dx <- x2 -x1
dy <- y1 - y2

xf <- x0*dx + x1
yf <- y0*dy + y2

return(as.data.frame(cbind(xf,yf)))
}

rd3 <- function(perc) #Pasa el rango de % a unidad de distancia
{
  d <- 3700
  irange <- d*perc/100
  return(irange)
}

tablas <- function(fx,fy,lx,ly,res,irange, m, lambda, rotangle, nug)
#S.G.S. - la tabla de residuales original
{
  aexp <- 1.82
  agau <- 1.3
  s <- 1
  nu <- nug + 0.001
  mod <- model(m)
  if (m == 2){m.ltpb.f <- vgm(s ,as.character(mod),irange*aexp, nu, anis
= c(rotangle, lambda))}
  else if (m == 1){m.ltpb.f <- vgm(s ,as.character(mod),irange, nu, anis
= c(rotangle, lambda))}
  else if (m == 3){m.ltpb.f <- vgm(s ,as.character(mod),irange*agau, nu,
anis = c(rotangle, lambda))}
  ng.g1 <- r.grid.c(fx,fy,lx,ly,res)
  salida1 <- list()

  ndis<-as.numeric(vector())
  if (m == 2){
    obs.data2 <- rot.grid(fx,fy,lx,ly,irange*aexp,lambda,7*pi/2 -
rotangle*pi/180)
    obs.data1 <- rot.grid(fx,fy,lx,ly,irange*aexp,lambda,7*pi/2 -
rotangle*pi/180)
    print("Using Exponential Model", sep = "")}
}

```

```

else if (m == 1){obs.data2 <- rot.grid(fx,fy,lx,ly,irange,
lambda,7*pi/2 - rotangle*pi/180)
obs.data1 <- rot.grid(fx,fy,lx,ly,irange,
lambda,7*pi/2 - rotangle*pi/180)
print("Using Spherical Model", sep = "")}
else if (m == 3){obs.data2 <- rot.grid(fx,fy,lx,ly,irange*agau,
lambda,7*pi/2 - rotangle*pi/180)
obs.data1 <- rot.grid(fx,fy,lx,ly,irange*agau,
lambda,7*pi/2 - rotangle*pi/180)
print("Using Gaussian Model", sep = "")}

label <- 0
#####
x11(bg = "Gray")
plot(coordinates(obs.data1), col = "Blue", pch = 19, main =
"S.G.S")
points(coordinates(ng.g1), col = "DarkRed")
#####
for (i in 1:dim(ng.g1)[1])
{
print(i)
print(label)
interp1.2 <- krige(prop ~1, locations=obs.data1,
newdata=ng.g1[i,], model=m.ltpb.f) #Sequential Gaussian Simulation (SGS)
ndis[i]<-
rnorm(1,interp1.2$var1.pred[1],sqrt(interp1.2$var1.var[1]))
Interp1.2 <- data.frame( prop = ndis[i])
Interp1.2 <- SpatialPointsDataFrame(interp1.2@coords,
Interp1.2)
obs.data1 <- rbind(obs.data1,Interp1.2)
####
points(coordinates(Interp1.2), col = "Green", pch = 19)
####
if (((nrow(obs.data1)-nrow(obs.data2)) >=
4*nrow(obs.data2))&(label != 1)){
obs.data3 <- obs.data1[
obs.data1[(nrow(obs.data2)+1):nrow(obs.data1),]
obs.data1 <- obs.data3
label <- 1
}
}
salida1[[1]] <- obs.data1
return(salida1)
}
library(compiler)
tablas1 <- cmpfun (tablas)

plot.kresults <- function(kr.o, var1, samp.pts, subsamp.pts,f=1, title="")
#La graficadora de los mapas
{
to.eval <- paste("plot.kr <- levelplot(",
paste(var1,"pred",sep="."),
" ~ x+y, as.data.frame(kr.o),

```

```

aspect='iso',
col.regions=bpy.colors(64), cut=32,
main=paste(title, ''),
)" );
eval(parse(text=to.eval));
to.eval <- paste("plot.kr.e <- levelplot(",
paste(var1,"var",sep="."),
"~ x+y, as.data.frame(kr.o),
aspect='iso',
col.regions=cm.colors(64), cut=32,
main=paste(title, ''),
panel = function(x, ...) {
panel.levelplot(x, ...);
panel.points(coordinates(samp.pts), col='green', pch=20,
cex=.6);
panel.points(coordinates(subsamp.pts), col='red', pch=20,
cex=.8); # subsample points slightly larger
panel.grid(h=-1, v=-1, col='darkgrey')
})" );
eval(parse(text=to.eval));
print(plot.kr, split = c(1,1,2,1), more=T);
print(plot.kr.e, split = c(2,1,2,1), more=F)
}

plot.map <- function(fx,fy,lx,ly,res,irange, m, lambda,
rotangle,nug,x1,y1,x2,y2) #La que grafica el primer mapa - residuales
original
{
t1 <- tablas1(fx,fy,lx,ly,res,irange,m, lambda, rotangle, nug)
t2 <- data.frame()
t2 <- t1[[1]]
t2 <- as.data.frame(t2)
coordinates(t2) <- ~x+y

tabla <- t.coord(x1,y1,x2,y2,t2)
t3 <- as.data.frame(cbind(tabla,t2$prop))
names(t3) <- c("x","y","prop")
t <- t3
coordinates(t) <- ~x+y

t4 <- cbind(t3, 1:nrow(t3))
coordinates(t4) <- ~x+y

names(t4) <- c("var1.pred","var1.var")
x11()
plot.kresults(kr.o = t4, var1 = "var1" , samp.pts = t4, subsamp.pts = t4,
title = "Residuales Originales")
return(t)
}

trend1 <- function (tabla, a = 1, b = 1, c = 1, d = 1, e = 1, f = 0)
#Superficie de tendencia
{

```

```

n <- 2
x1 <- as.array(coordinates(tabla)[,1] )
y1 <- as.array(coordinates(tabla)[,2] )
x <- x1 - mean(x1)
y <- y1 - mean(y1)
  ifelse((n > 2),
    paste("The order of the polynomial surface must be '0', '1' or '2'",
    sep=" "),
    as.data.frame( a*x^n + b*x^(n-1) + c*y^n + d*y^(n-1) +
e*(x^(n-1))*(y^(n-1)) + f ) )
}

reescalar <- function(desvest, tabla) #Transformacion de los residuales
{
media <- 0
z <- tabla$prop
aux <- z*desvest + media
aux <- as.data.frame(cbind(coordinates(tabla), aux))
coordinates(aux) <- ~x+y
return(aux)
}

muestra.u <- function (tabla, percent) #muestreo uniforme
{
percent <- percent/100
tabla.s <- data.frame()
x1 <- 1:length(tabla)
y1 <- sample(x1,as.integer(percent*length(tabla)), replace = F)
tabla.s <- tabla[y1,]
return(tabla.s)
}

muestra.l <- function(tabla, percent, dec) #ligeramente sesgado
{
nq <- 4
l <- list()
a <- round(length(tabla)*(percent/100))
prob <- seq(from = 0, to = 100, by = 100/nq)/100
aux3 <- c()
aux2 <- list()
aux <- tabla$prop
aux1 <- sort(aux, decreasing = as.logical(dec))
s <- as.integer(length(aux1)*prob)
for (j in 1:(length(s)-1)){
for (i in (s[j]+1):(s[j+1])){
aux3[i] <- aux1[i] }
aux2[[j]] <- aux3
aux3 <- c()}
aux2 <- lapply(aux2,function(x) x[!is.na(x)])
for (i in 1:length(aux2))
l[[i]] <- sample(aux2[[i]], replace = F, a*i/10)
muestra <- unlist(l)
muestra1 <- tabla[1,]

```

```

for (k in 1:length(muestra)){
muestra.f <- tabla[tabla$prop == muestra[k],]
muestra1 <- rbind(muestra1, muestra.f)
}
muestra1 <- muestra1[2:length(muestra1),]
return(muestra1)
}

muestra.f <- function(tabla, porcent, dec) #fuertemente sesgado
{
nq <- 4
l <- list()
a <- round(length(tabla)*(porcent/100))
prob <- seq(from = 0, to = 100, by = 100/nq)/100
aux3 <- c()
aux2 <- list()
aux <- tabla$prop
aux1 <- sort(aux, decreasing = as.logical(dec))
s <- as.integer(length(aux1)*prob)
for (j in 1:(length(s)-1)){
for (i in (s[j]+1):(s[j+1])){
aux3[i] <- aux1[i] }
aux2[[j]] <- aux3
aux3 <- c()}
aux2 <- lapply(aux2,function(x) x[!is.na(x)])

l[[1]] <- sample(aux2[[1]], replace = F, a*0.05)
l[[2]] <- sample(aux2[[2]], replace = F, a*0.15)
l[[3]] <- sample(aux2[[3]], replace = F, a*0.3)
l[[4]] <- sample(aux2[[4]], replace = F, a*0.5)
muestra <- unlist(l)
muestra1 <- tabla[1,]
for (k in 1:length(muestra)){
muestra.f <- tabla[tabla$prop == muestra[k],]
muestra1 <- rbind(muestra1, muestra.f)
}
muestra1 <- muestra1[2:length(muestra1),]
return(muestra1)
}

covariable <- function(tabla, coefic, media, s.dev) #Variable
correlacionada
{
a <- coefic
x <- tabla$prop
y <- rnorm(length(x), mean = media, sd = s.dev)
x1 <- a*x + sqrt(1 - (a)^2)*y
plot(x,x1,main= "Correlación de Variables", xlab = "Variable Simulada",
ylab= "Variable Correlacionada"
,sub = paste("Coeficiente de correlación", "=", coefic, sep=" "),
pch=20, col="DarkGreen")
return(x1)
}

```



```

polar.plot <-
function (r, theta, theta.zero = 0, theta.clw = FALSE, method = 1,
        rlabel.axis = 0, dir = 8, rlimits = NULL, grid.circle.pos = NULL,
        grid.lwd = 1, grid.col = "black", points.pch = 20, points.cex = 1,
        lp.col = "black", lines.lwd = 1, lines.lty = 1, polygon.col = NA,
        polygon.bottom = TRUE, overlay = NULL, pi2.lab = TRUE, text.lab = NULL,
        num.lab = NULL, rlabel.method = 1, rlabel.pos = 3, rlabel.cex = 1,
        rlabel.col = "black", tlabel.offset = 0.1, tlabel.cex = 1.5,
        tlabel.col = "black", main = NULL, sub = NULL)      #Permite emplear
marco de referencia polar para la elipse de anisotropía
{
  fit.rad <- function(x, twop = 2 * pi) {
    for (i in 1:length(x)) {
      while (x[i] < 0) x[i] <- x[i] + twop
      while (x[i] >= twop) x[i] <- x[i] - twop
    }
    return(x)
  }
  if (is.null(rlimits))
    rpretty <- pretty(range(abs(r), 0, na.rm = TRUE))
  if (is.numeric(rlimits) & length(rlimits) == 2)
    rpretty <- pretty(range(abs(rlimits[1]), abs(rlimits[2])))
  if (is.numeric(grid.circle.pos) & length(grid.circle.pos) >
      1)
    rpretty <- grid.circle.pos
  lab.dist <- max(rpretty)
  if (!is.null(text.lab) || is.numeric(num.lab) || pi2.lab) {
    lab.dist <- lab.dist * (tlabel.offset + 1)
  }
  if (is.null(overlay)) {
    plot.new()
    ps <- max(lab.dist, max(rpretty))
    plot.window(xlim = c(-ps, ps), ylim = c(-ps, ps), asp = 1)
    title(main = main, sub = sub)
  }
  drawgrid <- function() {
    if (dir > 0) {
      rDir <- seq(0, 2 * pi, length = dir + 1)[-(dir +
        1)]
      segments(0, 0, max(rpretty) * cos(rDir), max(rpretty) *
        sin(rDir), col = grid.col, lwd = grid.lwd)
    }
    grid <- seq(0, 2 * pi, length = 360/4 + 1)
    for (rad in rpretty) {
      if (rad > 0)
        lines(rad * cos(grid), rad * sin(grid), col = grid.col,
          lwd = grid.lwd)
    }
    if (rlabel.method != 0) {
      if (rlabel.method == 1)
        radLabels <- 1:length(rpretty)
      if (rlabel.method == 2)

```

```

        radLabels <- 2:length(rpretty)
    if (rlabel.method == 3)
        radLabels <- 1:(length(rpretty) - 1)
    if (rlabel.method == 4) {
        if (length(rpretty) > 2)
            radLabels <- 2:(length(rpretty) - 1)
        else radLabels <- NULL
    }
    if (!is.null(radLabels)) {
        xpos <- rpretty[radLabels] * cos(rlabel.axis)
        ypos <- rpretty[radLabels] * sin(rlabel.axis)
        text(xpos, ypos, rpretty[radLabels], cex = rlabel.cex,
             pos = rlabel.pos, col = rlabel.col)
    }
}
if (!is.numeric(num.lab)) {
    if (pi2.lab & !is.character(text.lab))
        t.lab <- expression(0, pi/2, pi, 3 * pi/2)
    if (!pi2.lab & is.character(text.lab))
        t.lab <- text.lab
    labDir <- seq(0, 2 * pi, length = length(t.lab) +
                 1)[-(length(t.lab) + 1)]
    labDir <- fit.rad(theta.zero + (!theta.clw) * labDir -
                    (theta.clw) * labDir)
    text(lab.dist * cos(labDir), lab.dist * sin(labDir),
         t.lab, cex = tlabel.cex, col = tlabel.col)
}
if (!pi2.lab & is.null(text.lab) & is.numeric(num.lab)) {
    labDir <- seq(0, 2 * pi, length = num.lab + 1)[-(num.lab +
1)]
    labDir <- fit.rad(theta.zero + (!theta.clw) * labDir -
                    (theta.clw) * labDir)
    text(lab.dist * cos(labDir), lab.dist * sin(labDir),
         paste(num.lab * labDir/(2 * pi)), cex = tlabel.cex,
         col = tlabel.col)
}
if ((is.character(text.lab) & is.numeric(num.lab)) ||
    (is.character(text.lab) & pi2.lab) || (pi2.lab &
is.numeric(num.lab)))
    print("More than one type of angular labels was requested.")
}
theta2 <- fit.rad(theta.zero + (!theta.clw) * theta - (theta.clw) *
theta)
cartesian.rt <- cbind(r * cos(theta2), r * sin(theta2))
if (method == 1) {
    if (is.null(overlay) || overlay == 2)
        drawgrid()
    points(cartesian.rt[, 1], cartesian.rt[, 2], col = lp.col,
          pch = points.pch, cex = points.cex)
}
if (method == 2) {
    if (is.null(overlay) || overlay == 2)
        drawgrid()
}

```

```

        lines(cartesian.rt[, 1], cartesian.rt[, 2], lwd = lines.lwd,
              col = lp.col, lty = lines.lty)
    }
    if ((method == 2 || method == 3) & length(r) <= 1)
        print("More than one data point is needed for line and polygon
methods.")
    if (method == 3) {
        if (!polygon.bottom & (is.null(overlay) || overlay ==
2))
            drawgrid()
        polygon(cartesian.rt, lwd = lines.lwd, col = polygon.col,
border = lp.col, lty = lines.lty)
        if (polygon.bottom & (is.null(overlay) || overlay ==
2))
            drawgrid()
    }
}

anis.diag <- function(irange, lambda, rotangle) #Construye y grafica la
elipse de anisotropia
{
a <- as.numeric(abs(irange))
b <- as.numeric(abs(irange*lambda))
alpha <- 7*pi/2 - rotangle*pi/180
theta <- seq(0,2*pi,pi/64)
    r <- 1/(sqrt((cos(theta - alpha)/a)^2+(sin(theta - alpha)/b)^2))
    polar.plot(r=r, theta=theta, theta.zero=0,
rlabel.col="DarkBlue",grid.col="DarkGreen", main= "Elipse de Anisotropía")
}

trid.plot <- function(tabla, theta, phi, a=1, b=1, c=1, d=1, e=1, f=0)
#Grafica en 3D la superficie de tendencia
{
x1 <- as.array(coordinates(tabla)[,1])
y1 <- as.array(coordinates(tabla)[,2])
x.ord <- sort(x1)
y.ord <- sort(y1)
x <- x.ord - mean(x.ord)
y <- y.ord - mean(y.ord)
x <- unique(x)
y <- unique(y)
    g <- function(x,y) {n =2 ;a=a;b=b;c=c;d=d;e=e;f=f ;r <- a*x^n + b*x^(n-1)
+ c*y^n + d*y^(n-1) + e*(x^(n-1))*(y^(n-1)) + f }
    z <- outer(x, y, g)
    z[is.na(z)] <- 1
    op <- par(bg = "white")

    persp(x, y, z, theta = theta, phi = phi, expand = 0.5, col = "lightblue",
ltheta = 120, shade = 0.75, ticktype = "detailed",
xlab = "X", ylab = "Y", zlab = "Z", main = "Superficie de
Tendencia"
,sub= "Regionales"
)
}

```

```

}

cuartiles <- function(tabla) #Separa la propiedad simulada en cuartiles
{
id <- 1:length(tabla)
b <- as.data.frame(tabla$prop)
a <- cbind(b,id)
names(a) <- c("prop","id")
p <- c(0,0.25,0.5, 0.75,1)
int <- (max(a$prop)-min(a$prop))*p
quart <- min(a$prop)+int

cuart25 <- a[a$prop<=quart[2],]
cuart50 <- a[(a$prop<=quart[3])&(a$prop>quart[2]),]
cuart75 <- a[(a$prop<=quart[4])&(a$prop>quart[3]),]
cuart100 <- a[(a$prop>quart[4]),]

cuart <- list(cuart25$id, cuart50$id, cuart75$id, cuart100$id)
values <- unlist(cuart)

aux1 <- rep(1, nrow(cuart25))
aux2 <- rep(2, nrow(cuart50))
aux3 <- rep(3, nrow(cuart75))
aux4 <- rep(4, nrow(cuart100))

aux <- list(aux1, aux2, aux3, aux4)
id <- unlist(aux)

output <- as.data.frame(cbind(id, values))
salida <- apply(output,2,function(x) x[order(output[,2])])
salida <- salida[,1]
return(salida)
}

validar1 <- function(fx,fy,lx,ly,res,irange,m,nu,rotangle,lambda,ruta)
#Calcula y escribe el variograma mayor
{
setwd(paste(ruta))

s <- 1
mod <- model(m)
nlags <- 9
maindt <- 2
htol <- 10

l<- list()
l[[1]] <- read.table(paste("salida",1,".txt",sep="") , sep="\t"
, dec = ",", header=T)
coordinates(l[[1]]) <- ~ x + y

aux <- list()
i <- 1
l[[i]] <- as.data.frame(l[[i]])

```

```

        aux[[i]] <- l[[i]][,1:2]
        l[[i]] <- l[[i]]$prop/sd(l[[i]]$prop)
        l[[i]] <- cbind(aux[[i]], l[[i]])
        coordinates(l[[i]]) <- ~x+y
        names(l[[i]]) <- "prop"
        cutof <- max(c(abs(fx - lx), abs(fy - ly)))/maindt
        width <- cutof/(nlags)
        l.var<- list()
        v1<- list()
    j<-1
        v1[[j]] <- variogram(prop ~ 1, data=l[[j]], cutoff=cutof, width=width,
alpha = rotangle, tol.hor = htol)
        m1 <- vgm(s,as.character(mod),irange,nu, anis = c(rotangle, lambda))
        variog <- as.data.frame(v1[[j]][,2:3])

return(variog)
}

validar2 <- function(fx,fy,lx,ly,res,irange,m,nu,rotangle,lambda,ruta)
#Calcula y escribe el variograma menor
{
    setwd(paste(ruta))

        s <- 1
        mod <- model(m)
        nlags <- 9
        maindt <- 2
        htol <- 10

        l<- list()
        l[[1]] <- read.table(paste("salida",1,".txt",sep="") , sep="\t"
, dec = ",", header=T)
        coordinates(l[[1]]) <- ~ x + y

        aux <- list()
        i <- 1
            l[[i]] <- as.data.frame(l[[i]])
            aux[[i]] <- l[[i]][,1:2]
            l[[i]] <- l[[i]]$prop/sd(l[[i]]$prop)
            l[[i]] <- cbind(aux[[i]], l[[i]])
            coordinates(l[[i]]) <- ~x+y
            names(l[[i]]) <- "prop"

            cutof <- max(c(abs(fx - lx), abs(fy - ly)))/maindt
            width <- cutof/(nlags)
            l.var<- list()
            v1<- list()
        j<-1
            v1[[j]] <- variogram(prop ~ 1, data=l[[j]], cutoff=cutof, width=width,
alpha = rotangle+90, tol.hor = htol)
            m1 <- vgm(s,as.character(mod),irange,nu, anis = c(rotangle, lambda))
        variog <- as.data.frame(v1[[j]][,2:3])
return(variog)
}

```

```

}

ejecutar <- function(ruta)
{
  setwd(ruta)
  x <- read.table(paste("entrada1.txt", sep="") , sep="\t" , dec = ".",
header = T)

  fx <- 178400
  fy <- 333800
  lx <- 181600
  ly <- 329600

  x1 <- x$v[1]
  y1 <- x$v[2]
  x2 <- x$v[3]
  y2 <- x$v[4]
  res <- x$v[5]
  irange <- rd3(x$v[6])
  lambda <- x$v[7]
  m <- x$v[8]
  rotangle <- x$v[9]
  nug <- x$v[10]
  a <- x$v[11]
  b <- x$v[12]
  c <- x$v[13]
  d <- x$v[14]
  e <- x$v[15]
  f <- x$v[16]
  desvest <- x$v[17]
  percent <- x$v[18]
  dec.l <- x$v[19]
  dec.f <- x$v[20]
  coefic <- x$v[21]
  media.vc <- x$v[22]
  sd.vc <- x$v[23]

  tabla <- plot.map(fx,fy,lx,ly,res,irange, m, lambda, rotangle,
nug,x1,y1,x2,y2)
  savePlot(file = "Mapa1.pdf", type = "pdf")

  write.table(
    tabla,
    paste(ruta,"/salida1",".txt", sep = ""),
    row.names=FALSE, sep="\t", dec=", "
  )

  vgm.M <- validar1(fx,fy,lx,ly,res,irange,m,nug,rotangle,lambda,ruta)
  vgm.m <- validar2(fx,fy,lx,ly,res,irange,m,nug,rotangle,lambda,ruta)

  z <- unlist(trend1(tabla, a = a, b = b, c = c, d = d, e = e, f = f))
  x11()
  trid.plot(tabla,60,20,a = a, b = b, c = c, d = d, e = e, f = f)

```

```

    savePlot(file = "3DMap1.pdf", type = "pdf")
    tabla.resc <- reescalar(desvest, tabla)
    residuales <-
as.data.frame(cbind(as.data.frame(tabla.resc), as.data.frame(1:length(tabla.
resc))))
    names(residuales) <- c("x", "y", "var1.pred", "var1.var")
    coordinates(residuales) <- ~x+y

x11()
plot.kresults(kr.o = residuales, var1 = "var1", samp.pts = residuales
, subsamp.pts = residuales, title = "Residuales Transformados")
savePlot(file = "Residuales.pdf", type = "pdf")

datos <- as.data.frame(tabla.resc$aux + z )
tabla.datos <- SpatialPointsDataFrame(tabla.resc@coords, datos)
names(tabla.datos) <- c("prop")

m.u <- muestra.u(tabla.datos, percent)
m.l <- muestra.l(tabla.datos, percent, dec.l)
m.f <- muestra.f(tabla.datos, percent, dec.f)

x11()
norm.corr <- covariable(tabla.datos, coefic, media.vc, sd.vc)
savePlot(file = "Variable.correlacionada.pdf", type = "pdf")

quant <- cuartiles(tabla.datos)

tabla.final <- cbind(as.data.frame(tabla), z, norm.corr, tabla.resc$aux,
datos, quant)
names(tabla.final) <- c("x", "y", "Residuales
Originales", "Regionales", "Variable
Correlacionada", "Residuales
Transformados", "Residual + Regional", "Cuartiles")
coordinates(tabla.final) <- ~x+y

tabla.final1 <- as.data.frame(tabla.final)
names(tabla.final1) <- c("x", "y", "prop", "z", "norm.corr", "reescalados",
"var1.pred", "var1.var")
coordinates(tabla.final1) <- ~x+y
x11()
plot.kresults(kr.o = tabla.final1, var1 = "var1", samp.pts = m.u,
subsamp.pts = m.u, title = "Residual+Regional - Uniforme")
savePlot(file = "Mapa.u.z1.pdf", type = "pdf")
x11()
plot.kresults(kr.o = tabla.final1, var1 = "var1", samp.pts = m.f,
subsamp.pts = m.f, title = "Residual+Regional - Sesgo Alto")
savePlot(file = "Mapa.f.z1.pdf", type = "pdf")
x11()
plot.kresults(kr.o = tabla.final1, var1 = "var1", samp.pts =
m.l, subsamp.pts = m.l, title = "Residual+Regional - Sesgo Bajo")
savePlot(file = "Mapa.l.z1.pdf", type = "pdf")

x11()
anis.diag(irange, lambda, rotangle)

```

```

savePlot(file = "Elips_Anis.pdf", type = "pdf")
write.table(
  tabla.final,
  paste(ruta,"/tablamapa",1,".txt", sep = ""),
  row.names=FALSE, sep="\t", dec=", "
)

write.table(
  m.u,
  paste(ruta,"/muestra.u",1,".txt", sep = ""),
  row.names=FALSE, sep="\t", dec=", "
)

write.table(
  m.f,
  paste(ruta,"/muestra.f",1,".txt", sep = ""),
  row.names=FALSE, sep="\t", dec=", "
)

write.table(
  m.l,
  paste(ruta,"/muestra.l",1,".txt", sep = ""),
  row.names=FALSE, sep="\t", dec=", "
)

write.table(
  vgm.M,
  paste(ruta,"/Vgm_principal", ".txt", sep = ""),
  row.names=FALSE, sep="\t", dec=", "
)

write.table(
  vgm.m,
  paste(ruta,"/Vgm_menor", ".txt", sep = ""),
  row.names=FALSE, sep="\t", dec=", " )}

```