



Escola Tècnica Superior d'Enginyers
de Camins, Canals i Ports de Barcelona

UNIVERSITAT POLITÈCNICA DE CATALUNYA

PROJECTE O TESINA D'ESPECIALITAT

Títol

**APLICACIÓN DE LA TEORÍA DE GRAFOS PLANOS A
LA PERCOLACIÓN EN MEDIO POROSO**

Autor/a

ESTHER BLANCO MACIÀ

Tutor/a

F. XAVIER SÁNCHEZ VILA

M. CAMINO TEÓFILA BALBUENA MARTÍNEZ

Departament

ETMC - MAIII

Intensificació

HIDROLOGIA SUBTERRÀNIA

Data

OCTUBRE 2009

AGRADECIMIENTOS

Als meus tutors Xavi i Camino, per haver estat sempre tan propers, per la vostra dedicació i compromís, per regalar-me esforç, optimisme i fer-me gaudir de la feina. Gracias por ser ejemplo de brillantes maestros y mejores personas. Sou el meu model.

A mis amigos, por querer serlo.

Por vuestro brillo y autenticidad. A los nuevos por vuestro voto de confianza, a los de siempre, por vuestra inagotable paciencia, por ser amiguísimos, gracias por hacerme crecer. También a los que sí agoté su paciencia, por retarme a mejorar.

A toda mi familia, por estar, comprender y nunca fallar. Porque si no me hubierais tocado, os elegiría entre un millón.

A Berni, por entrar en la película haciendo un cameo y acabar firmando por el papel protagonista. Por tu singularidad, inteligencia, humor y encanto. Por compartir y sumar. Porque eres el UNO, muchacho. Y eres mi regalo, 38.

A ti, porque mi vida empezó aquel día.

Y los últimos, son sin duda los primeros.

Mi batería, mi causa, mi latir y mi fuerza. El amor que jamás perderé, el único que viene de fábrica. Por vuestro amor desmesurado e incluso, por suerte, exagerado. Por comprender lo incomprensible, por los ánimos, por tanta complicidad, por no gastaros nunca, por mejorarme cada día, por quererme infinito, ¿Cómo lo hacéis? Papá, Mamá, gracias por existir, porque sin vosotros, la nada.

RESUMEN

La presente tesina tiene por objetivo realizar una simulación del modelo de percolación de enlaces en un medio poroso combinando los conocimientos hidrológicos con la teoría de grafos planos.

La percolación se define como el movimiento o flujo de un líquido a través de un medio. La teoría de percolación se utiliza en estudios de contaminación de acuíferos, en la obtención de petróleo y gas y en la resolución de problemas de redes, entre otros.

La teoría de grafos, considerada una parte de la matemática discreta ha presenciado un crecimiento sin precedentes en el siglo veinte. Una de las principales razones para este crecimiento es que los grafos sirven como modelos matemáticos para analizar con éxito muchos problemas concretos del mundo real. Ciertos problemas en física, química, ciencias de las comunicaciones, tecnología de los computadores, genética, psicología, sociología y lingüística, pueden ser formulados como teoría de grafos.

En un primer capítulo se describe la estructura del trabajo, las motivaciones y los objetivos del mismo.

El segundo y tercer capítulo versan en torno a los conceptos básicos que componen las teorías de percolación y de grafos, a partir de los cuales se señalan algunas interacciones entre ambas y se describen algunas de sus aplicaciones. A continuación, se define el camino o cluster percolante como un camino formado por aristas conectadas mediante vértices.

El siguiente capítulo contiene los elementos del programa *BINGO_PERCOLATION*, desarrollado para los fines de este trabajo. Mediante la simulación de un modelo de percolación de enlaces en una red bidimensional, se asimila una superficie porosa como un grafo plano y se consideran los nodos y los enlaces de percolación como vértices y aristas de un grafo plano. La conectividad entre aristas será el elemento clave para la obtención de los parámetros que describen el fenómeno de percolación. El umbral de percolación es uno de estos parámetros, del cual se conoce su valor teórico para una malla bidimensional, y podrá ser comparado con el obtenido en las simulaciones. Se analizarán también el número de caminos no percolantes y la longitud del camino percolante.

La simulación se realiza para un caso isótropo, donde todas las aristas pueden ser generadas por el programa con la misma probabilidad; el caso anisótropo con preferencia vertical, donde las aristas en la dirección Y tienen una mayor probabilidad de activarse; y finalmente el caso anisótropo con preferencia horizontal, donde las aristas en dirección X poseen mayor probabilidad de activación.

Para comprobar la validez del modelo simulado, se realiza un análisis de Monte Carlo para una muestra de cincuenta elementos de cada caso. A continuación se lleva a cabo un estudio estadístico con el fin de hallar una relación entre el umbral de percolación, la longitud del cluster percolante y el número de caminos no percolantes.

Finalmente, para cada uno de los tres ejemplos considerados se compara la variación de los valores obtenidos en función de la anisotropía, del factor de forma y del tamaño de la malla.

Se concluye el trabajo verificando la fiabilidad del modelo y el valor de los parámetros percolantes. Las futuras líneas de trabajo se presentan en torno a la ampliación del modelo y sus aplicaciones.

ÍNDICE

AGRADECIMIENTOS	i
RESUMEN	iii
ÍNDICE	v
LISTADO DE FIGURAS	vii
LISTADO DE TABLAS	xi
GLOSARIO HIDROLÓGICO	xiii
GLOSARIO MATEMÁTICO	xv
1. INTRODUCCIÓN	1
1.1. MOTIVACIÓN	3
1.2. OBJETIVOS	4
1.3. ESTRUCTURA DE LA TESINA	5
2. TEORÍA DE PERCOLACIÓN	7
2.1. FENÓMENO Y MODELOS DE PERCOLACIÓN	9
2.1.1. FENÓMENO DE PERCOLACIÓN	10
2.1.2. MODELOS TEÓRICOS	16
2.2. PERCOLACIÓN EN UN MEDIO POROSO	21
2.3. APLICACIONES	23
3. TEORÍA DE GRAFOS	25
3.1 CONCEPTOS BÁSICOS DE LA TEORÍA DE GRAFOS	27
3.1.1. CONCEPTOS BÁSICOS	28
3.1.2. TIPOS DE GRAFOS	28
3.1.3. TEOREMAS BÁSICOS EN GRAFOS	29
3.2 GRAFOS PLANOS	31
3.2.1. CARACTERIZACIÓN DE LOS GRAFOS PLANOS	32
3.2.2. COLORACIÓN DE GRAFOS	36
3.2.3. REDES DE PERCOLACIÓN Y GRAFOS PLANOS	37
3.3. APLICACIONES	38
4. PROGRAMACIÓN	41
4.1. OBJETIVOS	43
4.2. EL PROGRAMA BINGO-PERCOLATION	44
4.2.1. PARÁMETROS Y METODOLOGÍA	44

4.2.2. MODELOS UTILIZADOS	50
5. EJEMPLO ISÓTROPO	53
5.1. ANÁLISIS DE UNA SIMULACIÓN	55
5.1.1. PARÁMETROS PERCOLANTES	55
5.1.2. ETAPAS DE LA SIMULACIÓN	58
5.2. ANÁLISIS DE MONTECARLO	61
5.2.1. PARÁMETROS PERCOLANTES	61
5.2.2. ESTUDIO ESTADÍSTICO	64
5.3. INFLUENCIA DEL FACTOR DE FORMA Y DEL TAMAÑO	70
5.3.1. EFECTO DEL FACTOR DE FORMA (FF)	70
5.3.2. EFECTO DEL TAMAÑO	75
6. EJEMPLOS ANISÓTROPOS	77
6.1. ANÁLISIS DE UNA SIMULACIÓN	79
6.1.1. PARÁMETROS PERCOLANTES	79
6.2. ANÁLISIS DE MONTECARLO	84
6.2.1. ESTUDIO ESTADÍSTICO	85
CASO ANISÓTROPO CON PREFERENCIA VERTICAL	85
CASO ANISÓTROPO CON PREFERENCIA HORIZONTAL	87
6.3. INFLUENCIA DEL FACTOR DE FORMA Y DEL TAMAÑO	91
6.3.1. INFLUENCIA DEL FACTOR DE FORMA EN EL CASO VERTICAL	91
6.3.2. INFLUENCIA DEL TAMAÑO EN EL CASO VERTICAL	94
6.4. COMPARACIÓN DE LOS EJEMPLOS ISÓTROPO Y ANISÓTROPOS	97
7. CONCLUSIONES	99
8. FUTURAS LÍNEAS DE TRABAJO	103
9. REFERENCIAS	107
ANEJOS	111
ANEJO1. TEORÍA DEL CAOS Y FRACTALES	113
ANEJO 2. ALGORITMO Y PROGRAMA BINGO_PERCOLATION	117
ANEJO3. SIMULACIONES	118
ANEJO4. VALORES DE LOS PARÁMETROS PERCOLANTES EN EL ANÁLISIS MONTE CARLO	121
ANEJO5. ESTUDIO ESTADÍSTICO	123

LISTADO DE FIGURAS

Figura 1. Esquema estructural de la tesina.	5
Figura 2. Ciclo hidrológico del agua [1].	9
Figura 3. Percolación de enlaces en una red cuadrada.	11
Figura 4. Mallas bidimensionales: cuadrada (a), triangular (b) y panal de abeja (c).	13
Figura 5. Red con enlaces (i) y nodos (ii) ocupados.	14
Figura 6. Percolación de nodos con un valor de $p=0,375$	17
Figura 7. Percolación de aristas en una malla (10x10).	18
Figura 8. Esquema de percolación de agua y petróleo en medio poroso.	22
Figura 9: Primer grafo, Puentes de Königsberg [11].	27
Figura 10. Representación de dos grafos isomorfos.	29
Figura 11. Matriz de adyacencia A_{ij}	30
Figura 12. Representación del mismo grafo.	31
Figura 13. Representación del grafo $K_{3,3}$	34
Figura 14. Dos regiones del grafo $K_{3,3}$	35
Figura 15. Grafo K_5	35
Figura 16. Distribución del Rey Sol y del Rey Luna.	36
Figura 17. Grafo del Rey Sol y del Rey Luna.	36
Figura 18. Redes de percolación representadas con mallas planas.	37
Figura 19: Grafo G correspondiente a la representación del butano C_4H_{10}	38
Figura 20. Archivo Input_data.	45
Figura 21. Relación de aristas vecinas en una malla (4x4).	47
Figura 22. Simulación del BINGO_PERCOLATION en una malla (6x6).	50
Figura 23. Modelos utilizados en la simulación.	51
Figura 24. Evolución de la tortuosidad.	56
Figura 25. Evolución de nonperc_paths.	57
Figura 26. Evolución de length_percpath.	57
Figura 27. 40% de aristas generadas.	58
Figura 28. 50% de aristas generadas, percolación.	59
Figura 29. 60% de aristas generadas caminos no percolantes y percolante.	59
Figura 30. 76,7% presencia única del cluster percolante.	60
Figura 31. Valores de alfacrit para las 50 iteraciones.	63
Figura 32. Valores de nonperc_path para las 50 iteraciones.	63
Figura 33. Valores de <i>length_perc</i> para las 50 iteraciones.	64
Figura 34. Función de distribución de frecuencia de la variable <i>alfacrit</i>	65
Figura 35. Diagrama de dispersión entre alfacrit y non_percpaths.	66

Figura 36. (a) Histograma de residuos y (b) Diagrama P-P de la variable dependiente.....	68
Figura 37. (a) Dispersión entre el residuo tipificado y <i>nonperc_paths</i> (b) dispersión entre la variable pronosticada y la distancia de Cook.	69
Figura 38. Diagrama de dispersión entre la variable real y la predicha.	69
Figura 39. Red 60x10 en el momento de la percolación (Escala deformada).	71
Figura 40. Evolución de <i>alfacrit</i> respecto al FF para Y= 5.....	71
Figura 41. Evolución de <i>alfacrit</i> respecto al FF para Y= 10.....	72
Figura 42. Evolución de <i>length_perc</i> respecto al FF para Y= 5.....	73
Figura 43. Evolución de <i>length_perc</i> respecto al FF para Y= 10.....	73
Figura 44. Red 50x5 con un 57% de aristas generadas y 6 <i>percpaths_num</i>	74
Figura 45. Evolución de <i>percpaths_num</i> respecto al FF para Y= 5.....	74
Figura 46. <i>Percpaths_num</i> respecto a FF para Y= 10.	75
Figura 47. <i>Alfacrit</i> en función de la dimensión de la malla.....	75
Figura 48. <i>Length_perc</i> en función de la dimensión de la malla.....	76
Figura 49. Tiempo CPU en función de la dimensión de la malla.	76
Figura 50. Evolución de la tortuosidad en el 80	80
Figura 51. Evolución de la tortuosidad en el 80	80
Figura 52. Representación del camino percolante en el caso horizontal.	81
Figura 53. Evolución del nº caminos no percolantes en el caso vertical.	81
Figura 54. Evolución del nº caminos no percolantes en el caso horizontal.....	81
Figura 55. Representación del máximo número de caminos no percolantes.....	82
Figura 56. Coexistencia de dos caminos percolantes en el caso vertical.....	83
Figura 57. Evolución de <i>length_percpaths</i> en el caso anisótropo horizontal.	83
Figura 58. Histograma de <i>length</i> percolantes en el caso vertical.....	84
Figura 59. Histograma de <i>alfacrit</i> en el caso horizontal.....	84
Figura 60. (a) Histograma de residuos y (b) Diagrama P-P de la variable dependiente.....	86
Figura 61. Plano correspondiente a la regresión lineal de las tres variables.	87
Figura 62. Distribución de frecuencia acumulada de <i>alfacrit</i>	88
Figura 63. Histograma de los residuos.....	89
Figura 64. Diagrama p-p.	89
Figura 65. Gráfico de dispersión del residuo tipificado <i>vs non_perpaths</i>	90
Figura 66. Gráfico de dispersión del valor pronosticado <i>vs non_perpaths</i>	90
Figura 67. Evolución de <i>alfacrit</i> (caso vertical) respecto al FF para Y= 5.....	91
Figura 68. Evolución de <i>alfacrit</i> (caso horizontal) respecto al FF para Y= 5.	92
Figura 69. Evolución de <i>length_perpath</i> (caso vertical) respecto al FF para Y= 10.....	92
Figura 70. Evolución de <i>length_perpath</i> (caso horizontal) respecto al FF para Y= 10.	93
Figura 71. Evolución de <i>perpath_num</i> (caso vertical) respecto al FF para Y= 5.....	93
Figura 72. Evolución de <i>length_perpath</i> (caso horizontal) respecto al FF para Y= 10.	94
Figura 73. Evolución de <i>alfacrit</i> (caso vertical) respecto al tamaño.....	94
Figura 74. Evolución de <i>alfacrit</i> (caso horizontal) respecto al tamaño.	95

Figura 75. Evolución de <i>length_perc</i> (caso vertical) respecto al tamaño.	95
Figura 76. Evolución de <i>length_perc</i> (caso horizontal) respecto al tamaño.	95
Figura 77. Tiempo CPU (caso vertical) en función de la dimensión de la malla.	96
Figura 78. Imágenes ambiguas: La vieja y la joven/ el indio esquimal [20].	113
Figura 79. Triángulo de Sierpinski [21].	114
Figura 80. Copo de nieve de Koch [22].	115
Figura 81. Simulación de una percolación por nodos [23].	115
Figura 82. Simulación de una malla (15x50).	118
Figura 83. Simulación de una malla (40x15).	119
Figura 84. Diagrama de dispersión de <i>length</i> vs <i>nonperc_paths</i>	123
Figura 85. Histograma de los residuos.	125
Figura 86. Diagrama p-p.	125
Figura 87. Gráfico de dispersión de residuo vs la variable dependiente.	126
Figura 88. Distancia de Cook vs la variable pronostica.	126
Figura 89. Dispersión entre la variable dependiente y su valor pronosticado.	127
Figura 90. Diagrama de dispersión.	127
Figura 91. Histograma de residuos.	128
Figura 92. Diagrama p-p.	128
Figura 93. Diagrama de dispersión de <i>length</i> vs <i>non_percpaths</i>	129
Figura 94. Diagrama de dispersión de <i>alfacrit</i> vs <i>nonperc_paths</i>	130
Figura 95. Diagrama de dispersión de <i>alfacrit</i> vs <i>nonperc_paths</i>	131
Figura 96. Diagrama de dispersión de <i>alfacrit</i> vs <i>nonperc_paths</i>	132
Figura 97. Diagrama de dispersión de <i>nonperc_paths</i> vs <i>length_perc</i> sin considerar tirada 19.	133
Figura 98. Plano de las tres variables (isótropo).	134
Figura 99. Plano de las tres variables (vertical).	134
Figura 100. Plano de las tres variables (horizontal).	135

LISTADO DE TABLAS

Tabla 1: Coordenadas nodales	47
Tabla 2: Nodos extremos de cada arista	48
Tabla 3: Matriz de conectividades	49
Tabla 4: Valores de los parámetros percolantes para las cincuenta iteraciones	62
Tabla 5: Valores descriptivos de las variables	64
Tabla 6: Matriz de correlaciones entre las variables.....	66
Tabla 7: Resumen del modelo.....	66
Tabla 8: ANOVA.....	67
Tabla 9: Coeficientes	67
Tabla 10: Prueba de Kolmogorov-Smirnov para una muestra	68
Tabla 11: Valores de alfacrit	80
Tabla 12: Estadísticos descriptivos.....	85
Tabla 13: Correlaciones.....	85
Tabla 14: Resumen del modelo.....	86
Tabla 15: Coeficientes	86
Tabla 16: Estadísticos descriptivos.....	87
Tabla 17: Estadísticos descriptivos.....	88
Tabla 18: Correlaciones.....	88
Tabla 19: Resumen del modelo.....	89
Tabla 20: Valores de los parámetros percolantes para el caso anisótropo vertical.....	121
Tabla 21: Valores de los parámetros percolantes para el caso anisótropo horizontal	122
Tabla 22: Estadísticos descriptivos.....	123
Tabla 23: Resumen del modelo.....	124
Tabla 24: ANOVA.....	124
Tabla 25: Coeficientes	124
Tabla 26: Prueba de Kolmogorov-Smirnov para una muestra	125
Tabla 27: Coeficientes	128
Tabla 28: Estadísticos descriptivos.....	129
Tabla 29: Coeficientes	129
Tabla 30: Estadísticos descriptivos.....	130
Tabla 31: Correlaciones.....	130
Tabla 32: Resumen del modelo.....	130
Tabla 33: ANOVA.....	131
Tabla 34: Estadísticos descriptivos.....	131
Tabla 35: Coeficientes	132
Tabla 36: Estadísticos descriptivos.....	132
Tabla 37: Estadísticos descriptivos.....	133

GLOSARIO HIDROLÓGICO

Acuífero: formación geológica que contiene agua de manera que puede ser aprovechada. Los acuíferos pueden ser confinados, en cuyo caso están rodeados por una formación geológica generalmente impermeable; pueden ser libres, donde el agua se transmite y circula libremente por el terreno. Existen también los llamados acuíferos semiconfinados, los cuales son una combinación de los dos anteriores. Un acuífero es homogéneo si presenta las mismas condiciones en toda su extensión. Es heterogéneo, si las propiedades hidráulicas varían espacialmente.

Conductividad hidráulica: representa la mayor o menor facilidad con que el medio deja pasar el agua a través de él por unidad de área transversal a la dirección del flujo. Tiene las dimensiones de una velocidad $[L \ T^{-1}]$. El valor de este parámetro está en función del material geológico, cuanto mayor es la permeabilidad, mayor conductividad.

Hidrogeología: ciencia que estudia las aguas dulces del planeta, en especial las subterráneas.

Isotropía (con respecto a la conductividad hidráulica): cuando K en la dirección X es igual que en la dirección Y . En caso contrario se llama *anisotropía*.

Ley de Darcy (1856): la cantidad de flujo en un medio poroso es proporcional a la pérdida de carga e inversamente proporcional al largo de la trayectoria de fluido.

Medio saturado: aquel en el que todos sus poros están llenos de agua, por tanto existen sólo dos fases, la líquida y la sólida.

Permeabilidad: propiedad de poder penetrar el agua o dejar salir el agua de una determinada formación geológica.

Porosidad: cociente entre el volumen de poros y el volumen total considerado. El volumen de poros es la suma del volumen de aire y el volumen de agua. En un medio saturado, el volumen de poros será igual al volumen de agua.

Tortuosidad: razón de la longitud del camino sinuoso del fluido y su distancia en línea recta, o camino más corto.

Transmisividad: capacidad de fluidez del agua entre dos puntos. También definida como el flujo de agua que atraviesa una superficie lateral vertical de ancho unitario, cuando el gradiente hidráulico es igual a 1. La unidad de medida es $[L^3/L \ T]$.

GLOSARIO MATEMÁTICO

Árbol de un grafo: cuando el grafo G es conexo y no tiene ciclos.

Camino: Estado en el que todos los vértices del recorrido son distintos.

Caras de un grafo: partes conexas bidimensionales, que diseccionan el plano. Las caras de un grafo son finitas, con una única excepción, la denominada cara C_∞ o cara no finita.

Ciclo: camino cuyo vértice inicial y final son el mismo. Es decir, un ciclo será un circuito en el que no se repiten vértices. La notación r -ciclo responde al hecho de que el ciclo consta del mismo número r de vértices que de aristas.

Circuito: dado un recorrido, se denomina circuito cuando el vértice inicial y final coinciden.

Curva de Jordán: cuando una curva C es continua y no tiene autointersecciones.

Grado o valencia: de un vértice $u \in V$, será el número de aristas incidentes con el vértice u .

Grafo: estructura combinatoria $G = (V(G), A(G))$ formada por un conjunto de puntos $V(G) = V = \{v_1, v_2, \dots, v_p, \dots\}$ llamados vértices o nodos y por un conjunto $A(G) = A = \{a_1, a_2, \dots, a_p, \dots\}$ cuyos elementos son llamados aristas o enlaces, tales que $a_j = (v_h, v_k) \in V \times V$.

Grafo conexo o conectado: para todo par de vértices u y v existe un camino que los conecta. Una componente conexa será el subgrafo G_i de mayor tamaño que cumpla la propiedad anterior.

Longitud: número de aristas que forman un camino.

Recorrido de un grafo G : sucesión de vértices v_1, v_2, \dots, v_n tales que v_i y v_{i+1} ($1 \leq i \leq n-1$) son adyacentes. El recorrido se denomina circuito cuando el vértice inicial y final coinciden. G a la sucesión de vértices son adyacentes.

Región de G : es una porción maximal del plano tal que dos puntos cualesquiera de la región, pueden unirse mediante una curva C , de tal forma que ningún punto de C corresponde a un vértice de G ni pertenece a ninguna de las curvas correspondientes a las aristas de G . Toda realización plana de un grafo planar G da lugar a una región no acotada llamada región exterior de G .

Vértice incidente: si $a_i = (u, v) \in A$ diremos que la arista a_i es incidente con el vértice u y el vértice v . Asimismo, los vértices u y v serán adyacentes o vecinos.

1. INTRODUCCIÓN

1.1. MOTIVACIÓN

Cuando llegó el momento de elegir un tema para mi tesina final de carrera, pensé en cuáles eran las materias que más interés habían despertado en mí. La hidrología subterránea fue mi primer pensamiento. Acudí al Dr. Xavier Sánchez Vila, quien me propuso un estudio del fenómeno de percolación, convirtiéndose así en mi tutor.

La percolación era un término nuevo para mí, las dudas frente al hecho de partir de cero se convirtieron en el interés necesario para poder tirar adelante el proyecto.

Observé que numerosos estudios sobre la percolación se apoyaban en las bases de la teoría de grafos. Por ello me dirigí a la Dra. M. Camino Balbuena, cuya investigación se desarrolla en teoría de grafos. Entró en el proyecto convirtiéndose en mi tutora y me animó a estudiar los grafos planos para poder aplicarlos en mi trabajo.

La primera etapa del camino sería conocer los aspectos básicos de las teorías de percolación y de grafos, campos que no había estudiado anteriormente, y encontrar sus relaciones.

La idea principal será: dado un grafo plano finito, simular una percolación de aristas en dos dimensiones (problema local), a partir de la cual se puedan obtener los parámetros percolantes de un sistema global, como podría ser un medio poroso. El modelo es una fracción a pequeña escala y debe ser válido, ya que la percolación describe propiedades asociadas con la conectividad de elementos microscópicos para poder entender la conectividad en un sistema macroscópico desordenado.

1.2. OBJETIVOS

Esta tesina tiene como objetivo primordial estudiar el fenómeno de la percolación y sus valores críticos para resolver un problema real de movimiento de flujo en una red. Para ello se utilizarán elementos de la teoría de grafos. En particular los modelos matemáticos de la teoría de grafos pueden ser muy útiles a la hora de estudiar la percolación en una malla plana.

Por tanto, el primer objetivo de esta tesina consiste en familiarizarse con los conceptos básicos de grafos y de percolación.

El segundo objetivo es desarrollar un programa que simule el movimiento de un flujo en un medio poroso en una red plana. Se considerará la percolación por aristas como modelo básico.

El objetivo principal del programa es el de obtener una serie de valores de los parámetros percolantes tales como el umbral de percolación, del que se conoce el valor teórico para una malla bidimensional, así como la obtención del cluster percolante y de los no percolantes.

Otros objetivos del trabajo son:

- Contrastar los resultados proporcionados por el programa, comparándolos con el valor teórico de la probabilidad crítica, así como verificar las hipótesis iniciales.
- Estudiar la variabilidad de los parámetros percolantes en función del tamaño de la malla, del factor de forma, del umbral de percolación y de la anisotropía en cuanto a la preferencia de dirección del flujo se refiera.
- Simular mediante el método de MonteCarlo tres muestras de cincuenta valores, con el fin de comparar el comportamiento de los parámetros en los casos isótropo, anisótropo con preferencia de flujo vertical y anisótropo con preferencia de flujo horizontal.
- Realizar un estudio estadístico de los valores obtenidos para cada uno de los parámetros en la simulación.
- Establecer modelos estadísticos lineales que permiten relacionar los diferentes parámetros percolantes y cuantificar la incertidumbre sobre sus valores medios.

El objetivo final es comprobar la validez de los modelos obtenidos.

1.3. ESTRUCTURA DE LA TESINA

Se describe en la **Figura 1** la estructura del trabajo. Se considera oportuno hacer hincapié en la unión entre los primeros capítulos hidrológico y matemático de carácter teórico, ya que a partir de ellos es posible desarrollar la simulación de un modelo de percolación mediante la teoría de grafos.

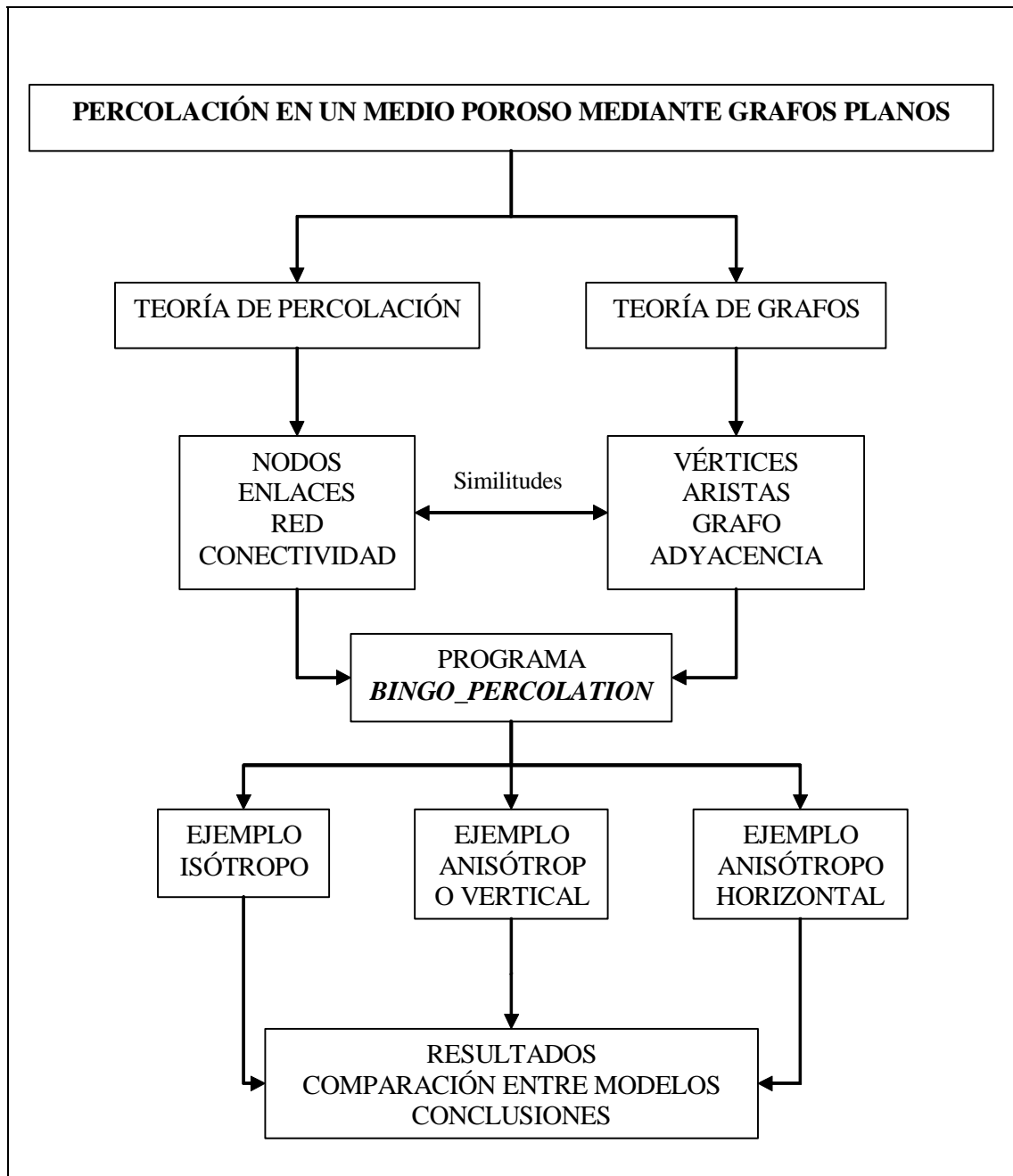


Figura 1. Esquema estructural de la tesina.

2. TEORÍA DE PERCOLACIÓN

2.1. FENÓMENO Y MODELOS DE PERCOLACIÓN

El ciclo hidrológico está compuesto por una circulación continua de las reservas de agua en la Tierra, a través de los océanos, el terreno y el aire. La cantidad de agua a lo largo de la historia del planeta es prácticamente constante gracias al ciclo representado en la **Figura 2**, que está formado por la precipitación, escorrentía, percolación, evaporación, transpiración y condensación del agua. Las únicas excepciones en este ciclo son el agua perteneciente a depósitos termales o aguas sulfuradas, las cuales nunca han estado en contacto con el exterior, y las aguas fósiles, formadas hace millones de años al quedar atrapadas durante la formación de un sedimento.

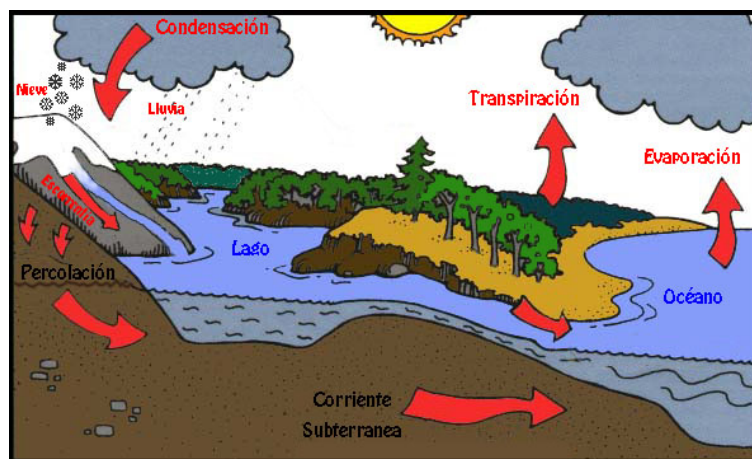


Figura 2. Ciclo hidrológico del agua [1].

Se denomina **percolación** al movimiento o flujo de un líquido a través de un medio poroso, por ejemplo, el agua en el terreno bajo la acción de la gravedad.

La percolación es un proceso aleatorio desarrollado a partir de la observación de la conectividad entre elementos microscópicos y sus efectos en las propiedades macroscópicas.

El modelo de percolación fue introducido por Broadbent y Hammersley en 1957 como un modelo de comportamiento de un fluido cuando atraviesa un medio poroso desordenado, siendo el ejemplo del agua atravesando el café su símil más gráfico.

Se toma una cafetera como elemento percolador y se aprieta más o menos el filtro del aparato. En el momento en que aumenta la presión del filtro, su densidad aumenta como consecuencia de la aglomeración de las finas partículas que lo componen. El tiempo necesario para que el agua atravesase la masa porosa de café, es decir, el tiempo en el que el

agua está en contacto con el café, depende de esta densidad. Por tanto, existirá una densidad por debajo de la cual el agua no puede atravesar el filtro. A esta densidad se le denomina umbral de percolación [2]. Cuando la densidad alcanza este valor crítico, tiene lugar el fenómeno de percolación y el agua atraviesa toda la superficie porosa dando lugar al café.

Otro ejemplo gráfico de percolación sería el paso de un archipiélago a un continente a partir de un descenso del nivel del mar [2]. Se parte de un conjunto de islas independientes en las que no es posible recorrerlas a pie. Al bajar el nivel del mar, el tamaño de las islas aumenta y algunas de ellas se unen. El espacio para el peatón es mayor pero sigue limitado al uso de una embarcación hasta el momento en que el nivel del mar alcanza un valor crítico. En ese momento emerge un continente rodeado de lagos que permite al peatón ir de un lado al otro sin necesidad de atravesar el mar.

La transición que se produce entre un archipiélago y un continente es denominada transición de percolación y el valor crítico del nivel del mar para que se produzca la aparición del continente corresponde al citado umbral de percolación.

El fenómeno de percolación responde a un fenómeno global analizado a partir de la observación de un fragmento. Al tomar una fracción de elementos de un sistema, se observa si están conectados o no. Las propiedades de dichas conexiones, creadas de forma aleatoria, definen un fenómeno global determinístico (momento de la percolación) y por tanto, no aleatorio.

El contenido de este capítulo está dedicado a caracterizar el fenómeno de percolación, definir los modelos posibles y describir su aplicación en un fluido atravesando un medio poroso.

2.1.1. FENÓMENO DE PERCOLACIÓN

La percolación describe propiedades asociadas con la conectividad de un gran número de objetos los cuales, individualmente tienen un alcance espacial y cuya correlación es aplicable y preestablecida estadísticamente [3]. La percolación sirve para entender la conectividad en un sistema desordenado.

Para desarrollar el modelo de percolación se asimila el medio natural a una malla o red cuadrada. Cada uno de los nodos que la forman está ocupado (lleno de fluido) o no, en

función de una probabilidad $p \in [0,1]$ dada. Por tanto, con una probabilidad $p=0$ no hay nodos ocupados. El mismo tratamiento reciben los enlaces, si el estudio de la percolación se realiza en referencia a la conexión entre enlaces.

La **Figura 3** muestra una malla cuadrada de dimensiones (30x30) en la que se observan de color azul, los enlaces conectados con probabilidad p igual a 1. A este modelo de percolación se le conoce como Percolación por enlaces (Bond percolation). En el siguiente apartado se describe junto con otros tipos de percolación.

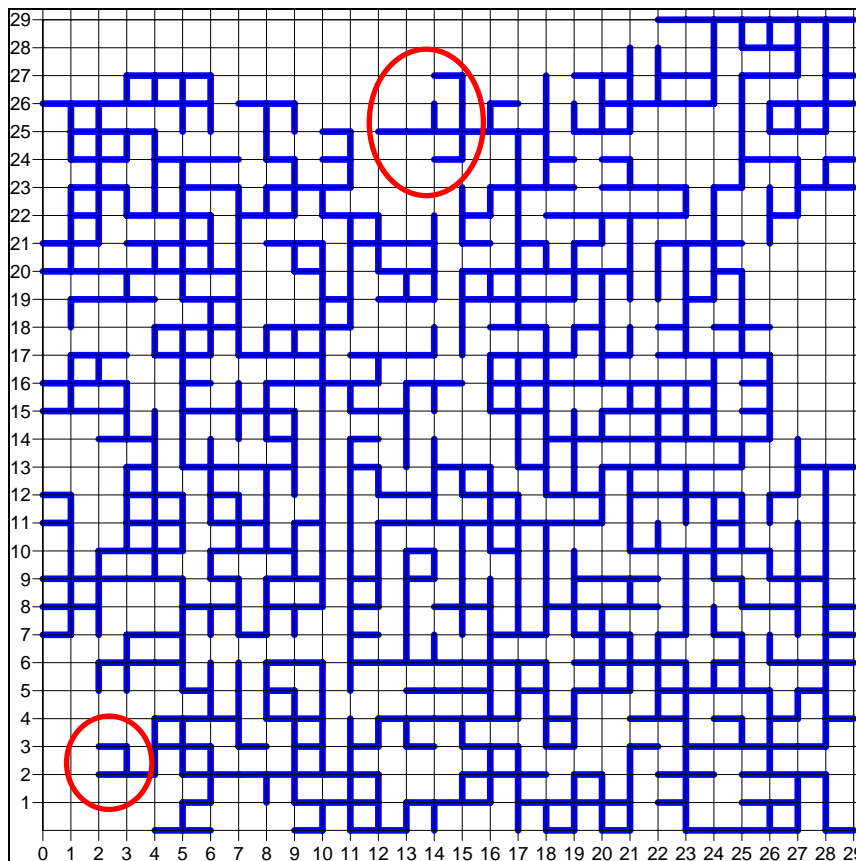


Figura 3. Percolación de enlaces en una red cuadrada.

La malla queda definida por la conexión o no de sus elementos vecinos. Sean nodos o enlaces, estos elementos son activados de forma aleatoria. A cada uno de los elementos se le asocia un estado ocupado (con probabilidad p) o libre (con probabilidad $q = 1-p$).

Para valores pequeños de p , únicamente se obtienen parejas de elementos conectados al unirse los vecinos más próximos. A medida que aumenta el valor de p , emergen mayores áreas que aglutinan elementos activados de menor tamaño.

Cuando la citada p alcanza un determinado valor de probabilidad (p_c), se llega al denominado *umbral de percolación*. En este momento, el aglomerado o racimo resultante es un objeto fractal (ver el *Anejo 1*) llamado *cluster infinito de percolación*, ya que el tamaño diverge cuando las dimensiones de la malla aumentan indefinidamente, quedando las regiones no conectadas en un área finita.

Como estructura fractal, si se toma una fracción del objeto, por pequeña que sea, se observa que es una réplica de su estructura principal. Para etiquetar un objeto como fractal, éste debe cumplir una de las siguientes características:

- Poseer detalle en cualquier escala de observación.
- Tener autosemejanza exacta o estadística.
- Ser definido recursivamente, o que su dimensión sea fraccionaria.

Se parte de la idea que una fracción del cluster tiene el mismo comportamiento que el original a nivel estadístico.

Cuando una estructura sufre una modificación, pasando de un conjunto de partes desconectadas a formar un cluster, se dice que tiene lugar el *Fenómeno de Percolación*. En ese caso se diferencian tres casos en función del valor de la probabilidad p respecto a la probabilidad crítica p_c :

- $p < p_c$: los nodos ocupados forman pequeñas áreas formadas con nodos vecinos ocupados.
- $p = p_c$: formación del cluster infinito de percolación, conectando los lados opuestos de la malla o red.
- $p > p_c$: el espesor del cluster es mayor.

En esta tesina se denomina *alfacrit* a la probabilidad crítica.

La **Figura 3** señala en círculos rojos algunos de los llamados *Dead ends* (finales muertos), conectados al cluster por un único punto. Cuando dichos elementos son eliminados, el cluster se denomina *Backbone* [3], pudiendo traducirse como *espina dorsal*.

En nuestro análisis posterior (*Capítulos 5 y 6*) se considera el cluster de percolación formado por todos sus elementos, incluidos los *dead ends*.

Valores teóricos de *alfacrit* (p_c)

Desde que fue introducido el término *percolación* en el lenguaje científico de mediados del s. XIX, han sido numerosos los estudios dedicados a buscar un valor teórico del umbral de percolación. Este valor es el atribuido a la concentración p a la cual un agregado de tamaño infinito aparece en una red de dimensión infinita.

En las primeras etapas se analizó este fenómeno en mallas finitas regulares bidimensionales, obteniendo un valor exacto de p_c para unas determinadas mallas representadas en la

Figura 4.

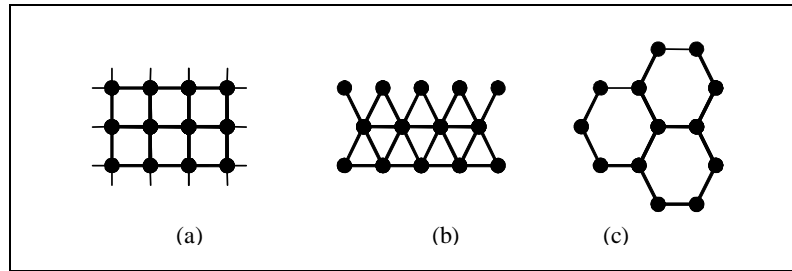


Figura 4. Mallas bidimensionales: cuadrada (a), triangular (b) y panal de abeja (c).

Las mallas triangular y de panal de abeja (*honeycomb*) tienen la misma estructura pero la posición de los nodos difiere en el mallado. Son útiles cuando el tamaño de la malla es considerable ya que aumenta su legibilidad.

La probabilidad crítica es un valor determinista del que se tienen sus valores para determinadas formas de mallas bidimensionales:

- Malla cuadrada: $p_c(\text{nodos}) = 0.592746$ $p_c(\text{enlaces}) = 0.5$
- Malla triangular: $p_c(\text{nodos}) = 0.5$ $p_c(\text{enlaces}) = 2\text{sen}(\pi/18) = 0.3473$
- Malla honeycomb: $p_c(\text{nodos}) = 0.6962$ $p_c(\text{enlaces}) = 1 - 2\text{sen}(\pi/18) = 0.6527$

En el momento en que la probabilidad p alcanza el valor crítico, el agregado percolante va creciendo. A medida que aumenta esta probabilidad disminuye el número de nodos o enlaces que se encuentran alejados del cluster.

Los valores de *alfacrit* (p_c) son menores cuando se trata de enlaces debido a que por ejemplo en una malla cuadrada, como la presentada en la **Figura 5**, tiene seis posibles vecinos si hablamos de conexión entre enlaces o cuatro vecinos si la conexión hace referencia a los nodos.

La red (i) tiene a las enlaces como elementos que contienen la información sobre la probabilidad de estar ocupado o libre. Cada enlace tiene en sus nodos extremos, seis enlaces vecinos.

La red (ii) muestra el caso en el que son los nodos los portadores de la información. Cada uno de estos nodos tiene cuatro nodos vecinos que pueden agregarlo a su área ocupada mediante de enlaces.

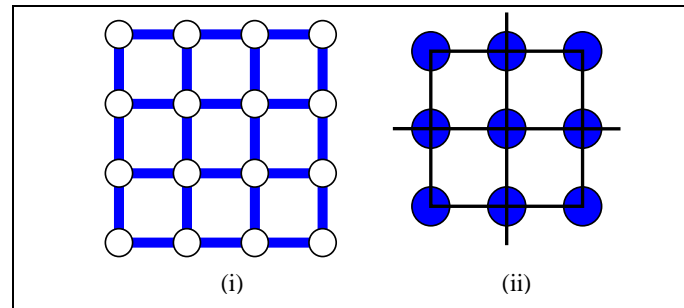


Figura 5. Red con enlaces (i) y nodos (ii) ocupados.

La percolación es un proceso de carácter aleatorio. Las redes de estudio pueden tener diferentes formas y medidas. Para poder discutir sus propiedades promedio, se estudian los estadísticos de los clusters.

La presencia del cluster representa un cambio cualitativo en la estructura de la red, pasando de un estado desconectado a uno conectado. En el caso de tomar una red infinita, el cluster de percolación será también de tamaño infinito.

En este trabajo se estudian los parámetros percolantes para una dimensión igual a 2, pero el fenómeno de percolación puede extenderse a mayores dimensiones, como la malla cúbica de dimensión $d=3$.

Aunque el valor más importante de la probabilidad p es el umbral de percolación (p_c), uno de los aspectos más importantes de la teoría de percolación es que la longitud de correlación diverge en el límite $p \rightarrow p_c$ [2]. Por ese motivo, el instante cercano al umbral de percolación es de gran interés también y viene representado por los siguientes parámetros.

Exponentes críticos

Se denominan Exponentes críticos a aquellos valores que corresponden a ciertas propiedades del fenómeno en un momento cercano al valor crítico.

Al tomar una fracción finita del modelo de percolación, las simulaciones del fenómeno no dan un valor exacto del *alfacrit*. Este tema será abordado en los capítulos 4, 5 y 6, en los que la simulación de una percolación de aristas nos dará distintos valores del umbral de percolación para una malla cuadrada bidimensional.

En el fenómeno de percolación, dichos exponentes se representan con las siglas β , ν y γ y se obtienen a partir de las siguientes definiciones:

Probabilidad (p_∞): es la probabilidad de que un nodo o enlace pertenezca al cluster infinito de percolación.

$$\text{Para una dimensión (d=1)} \begin{cases} p_\infty = 0 \Leftrightarrow p < p_c \\ p_\infty = 1 \Leftrightarrow p > p_c \end{cases}$$

Para una $d > 1$: $p_\infty = 0 \Leftrightarrow p < p_c$ y para valores superiores al umbral de percolación cumplirá la siguiente ley potencial:

$$p > p_c \rightarrow p_\infty(p) \sim (p - p_c)^\beta \quad (1)$$

donde $\beta = 5/36$ para mallas planas ($d=2$).

Longitud de correlación (ξ): se define como la distancia media entre dos puntos nodales del mismo agregado y caracteriza el tamaño de los agregados finitos. También se llama Tamaño promedio del cluster. Para los intervalos $p > p_c$ y $p < p_c$ sigue la siguiente ley potencial:

$$\xi(p) \sim |p - p_c|^{-\nu} \quad (2)$$

siendo $\nu = 4/3$ para $d=2$.

Masa de un agregado finito (S): se evalúa como el número de nodos que lo forman y promediada para todos los cluster finitos. Sigue la siguiente ley potencial:

$$S(p) \sim |p - p_c|^{-\gamma} \quad (3)$$

donde $\gamma = 43/18$ para malla planas.

β , ν , y γ son los encargados de caracterizar el comportamiento de p_∞ , ξ y S .

La siguiente relación indica la no independencia entre ellos:

$$2\beta + \gamma = \nu d \quad (4)$$

Los efectos del corte para el paso de una superficie infinita a una finita pueden ser producto del efecto de las fronteras creadas, por ejemplo. Para calcular la influencia del corte finito se utilizan los exponentes críticos, que dependen únicamente del modelo en estudio y de la dimensión euclídea del espacio en el que se implanta dicho modelo. Son independientes de la forma de la malla. Por tanto, hablamos de universalidad. En modelos aparentemente no relacionados se observan exponentes críticos comunes.

2.1.2. MODELOS TEÓRICOS

El modelo teórico de percolación considera que el sistema se genera de forma aleatoria, admitiendo que si un nodo o arista está ocupado, depende únicamente del azar. En función del elemento al que se le asigna esta probabilidad de estar o no ocupado tenemos tres tipos de percolación:

- Percolación de nodos (*Site percolation*).
- Percolación de enlaces (*Bond percolation*).
- Percolación mixta (*Site-bond percolation*/ *Continuum percolation*).

PERCOLACIÓN DE NODOS

En una malla formada por un conjunto de nodos (*sites*) y aristas o enlaces, se dice que dos nodos son los vecinos más cercanos si están unidos por una arista. Limitamos nuestro estudio a mallas o grafos infinitamente simétricos, por tanto, compuestos por un número infinito de nodos y aristas, en las cuales dos nodos cualesquiera son unidos a lo sumo por una arista no orientada. Desde este grafo se puede elaborar un laberinto aleatorio en función de dos posibles estados: ocupado (1 con probabilidad p) o libre (0 con probabilidad $q = 1-p$).

En los casos de mallas bidimensionales, la representación es sencilla y se le asigna un color diferente a cada estado del nodo (ocupado o libre). De ese modo se indica de forma clara si una serie de nodos están conectados o no en función del color asignado. Si la probabilidad p es superior a 0.592746 (valor crítico teórico de la percolación de nodos en una malla

cuadrada), la malla percola y el color representado por los nodos ocupados será el predominante del gráfico.

La **Figura 6** presenta una malla cuadrada con un 37,5% de nodos ocupados (color negro). El valor de la probabilidad p_c para mallas cuadradas en una percolación de nodos era igual a 0.592746, por tanto la malla representada aún no ha alcanzado el umbral de percolación.

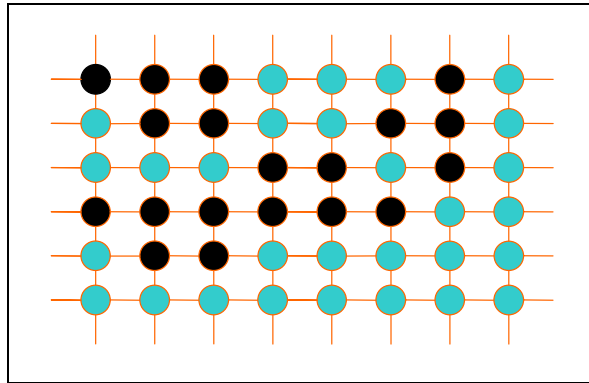


Figura 6. Percolación de nodos con un valor de $p=0,375$.

El modelo de percolación de nodos considera que todas las aristas o enlaces son activos, pero que los contactos entre enlaces están controlados por los nodos.

El valor de p_c es sensible a la conexión (enlace) de la red y a la dimensión del espacio. Así, cuanto más aumenta el número de nodos y de vecinos más próximos, mayor será el valor crítico necesario para la formación de un cluster infinito y débil, que es lo que se observa en numerosos tipos de redes.

Una percolación de nodos puede ser representada tomando como ejemplo un conjunto de bolas metálicas y de plástico situadas en una gran área.

Si dos bolas metálicas son las vecinas más cercanas, la corriente eléctrica pasa de una a otra. Si el número de bolas metálicas es suficiente para alcanzar una densidad crítica, puede establecerse un camino continuo conectado a través del metal. Este camino dará conductividad eléctrica. El conjunto formado por las bolas metálicas será la mejor conexión del camino y su longitud la conductividad eléctrica [3].

La teoría de percolación permite obtener conductividad eléctrica en función de un fragmento de bolas hechas de metal. Del mismo modo, si dos poros están llenos de agua y unidos por un enlace activo, la conductividad hidráulica será mayor.

PERCOLACIÓN DE ENLACES

Se toma una malla análoga a la descrita en la percolación de nodos y se considera al enlace el elemento que rige el proceso. Ahora los nodos serán simplemente los elementos que unirán a los enlaces y se consideran todos activos.

Sea el enlace o arista $a_j = (v_h, v_k)$, donde v_h y v_k son los nodos que ocupan sus extremos. Se habla de percolación por aristas al fenómeno que contiene un camino de nodos conectados mediante enlaces ocupados. En consecuencia, para una red de dimensiones $(v \times v)$, habrá un número de enlaces igual a $2v(v-1)$.

La **Figura 7** muestra una percolación de enlaces en el que se representa (color azul) el camino que une los nodos extremos mediante enlaces conectados, por tanto, el cluster percolante. Los enlaces negros responden a pequeñas áreas formadas por enlaces vecinos conectados que no forman un camino entre nodos extremos a la malla cuadrada $(v \times v)$.

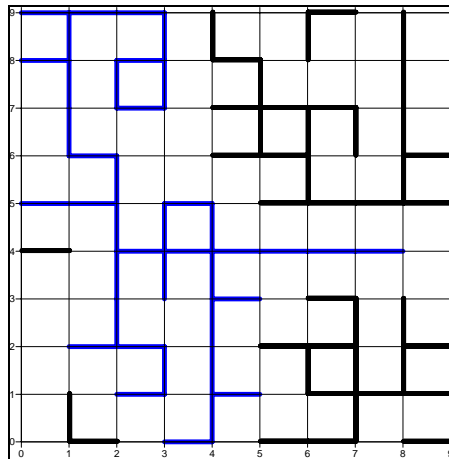


Figura 7. Percolación de aristas en una malla (10x10).

La malla cuadrada es una de las más utilizadas en percolación de enlaces.

El modelo aleatorio queda pues asociado a uno de los dos casos posibles: enlace ocupado ($p=1$) o libre ($q=1-p$).

Ya se ha comentado que el valor de *alfacrit* es menor en el caso de una percolación de enlaces, debido a que cada enlace tiene seis vecinos posibles y cada nodo, cuatro vecinos. Este valor es el más importante en la fase de transición. Kesten (1982) probó que el valor crítico en dimensión Z2 de la percolación de enlaces era igual a 0.5.

Cualquier modelo de *bond percolation* puede reformularse como un modelo de *site percolation*, pero la biyección es falsa. Fue M. E. Fisher (1961) quien demostró que sustituyendo un nodo por cada enlace de la red inicial y trazando líneas entre dichos nodos vecinos, se puede transformar siempre un problema de enlaces en uno de nodos. Ese hecho nos indica que la percolación de nodos es un modelo más general en el sentido que la existencia de un camino (*path*) percolante en una percolación de enlaces, es equivalente siempre a la existencia de un camino de la percolación de nodos [5].

PERCOLACIÓN MIXTA

Tanto la percolación de nodos como la de enlaces pueden ser consideradas casos particulares de la percolación mixta.

La probabilidad de encontrar un elemento ocupado o libre dependerá de dos factores en vez de uno, como en los casos anteriores.

Se toma una red formada por nodos unidos mediante enlaces. Cada nodo tiene una probabilidad p^s de estar ocupado y una probabilidad $1 - p^s$ de estar libre. Para cada enlace ocurre exactamente lo mismo: un enlace estará ocupado con una probabilidad p^b o libre con una probabilidad $1 - p^b$.

En la percolación mixta, la probabilidad es función de p^s y p^b . Cuando un nodo se encuentra activado pero su enlace no lo está, el conjunto se considerado libre. Deben estar los dos elementos activados para que la probabilidad del elemento se considere activada.

El valor de las probabilidades p^s y p^b es independiente, por ese motivo se considera un modelo más completo.

Una aplicación del modelo de percolación mixta es el estudio del posible contagio de una enfermedad en una determinada población. En este caso, el estado activado o libre de un nodo es atribuido a la posibilidad o no de transmitir la enfermedad por parte de un individuo concreto. Las probabilidades p^s y p^b pueden hacer referencia a la vulnerabilidad de las personas y a la virulencia de la enfermedad [2].

La ventaja de los modelos de percolación de nodos y de enlaces será, pues, el estudio de un solo parámetro en cuanto a la probabilidad se refiere.

El cálculo del umbral de percolación, en este caso, será más complejo porque intervienen dos probabilidades. Es evidente que si las dos aumentan conjuntamente, se llegará a la percolación de forma clara. En el caso en que sea un único valor (de nodos o enlaces) el que aumente, el cluster también puede llegar a formarse.

Con el objetivo de observar la evolución de cada grupo de elementos (nodos y enlaces) se compara el modelo aumentando la activación de uno, de otro y de los dos de forma constante.

2.2. PERCOLACIÓN EN UN MEDIO POROSO

Como se cita en la introducción del capítulo, fueron Broadbent y Hammersley (1957) los primeros en introducir el término percolación en el contexto de la modelización del flujo a través de un medio poroso. El experimento consistía en sumergir una roca porosa en un tanque de agua, a continuación se hicieron la siguiente pregunta: ¿Cuál era la probabilidad de que el centro de la roca se mojara? Vieron que la probabilidad p sería función de la porosidad de la roca [6].

En sus inicios, la teoría de percolación aplicada a un medio poroso se utilizó para modelar la propagación del petróleo en agua. En la actualidad se usa para modelizar, por ejemplo, los posibles caminos de poros conectados en una roca porosa para buscar la forma de extraer el petróleo que ocupa dichos poros [7].

En el campo de la hidrología subterránea, la teoría de percolación se aplica asimilando el medio a una red susceptible de ser atravesada por un fluido. Numerosos estudios toman la teoría de percolación para localizar la fuente de contaminación de un acuífero, para analizar la capilaridad o para describir los fenómenos de movimiento de un fluido a gran escala partiendo de propiedades conocidas a escala de poro.

Se parte con la obtención de una serie de estadísticos y se aplican a un caso real. Sabemos que dado un medio poroso con presencia de un fluido, existe una realidad (fenómeno de percolación) que desconocemos (en que momento percola el sistema) y buscamos los parámetros que la definen.

Imaginemos un medio poroso saturado. La circulación del fluido, por ejemplo de agua o petróleo, a través de los poros del terreno dependerá de diversos parámetros hidrológicos, como la permeabilidad, la porosidad de la roca y la conductividad hidráulica, a su vez ligados a la geología.

Existen estudios muy recientes [8]-[11] en los que se describen las propiedades de las areniscas a partir de la teoría de percolación. En cambio, a finales del S. XX, se realizaba su estudio, analizando el estado del material poro a poro.

La teoría de percolación permite tratar el medio poroso como un fractal

La **Figura 8** presenta dos casos de caminos percolantes y considera dos tipos de fluidos: camino percolante descendente con agua (A) y ascendente con petróleo (B).

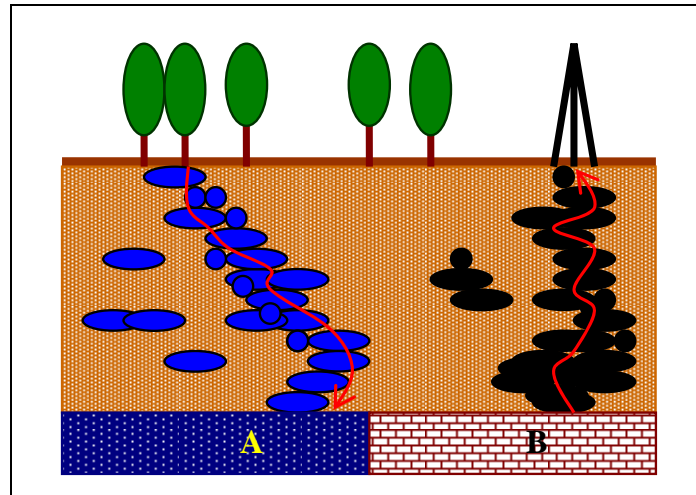


Figura 8. Esquema de percolación de agua y petróleo en medio poroso.

Se ha demostrado que a partir del estudio de un fragmento finito de un medio poroso se obtienen resultados de los parámetros percolantes satisfactorios. Por tanto, el modelo de percolación se considera un modelo intuitivo y transparente para explicar los desórdenes geométricos en el seno de numerosos medios heterogéneos como es el subsuelo.

Los parámetros percolantes en un medio poroso que serán objeto de estudio en este trabajo, se describen a continuación:

Camino percolante (perc_path): equivalente al cluster del modelo general. En este campo, cuando las fronteras consideradas del medio se unen mediante un camino de nodos o enlaces conectados, se llega al umbral de percolación.

Para mallas bidimensionales se ha demostrado que existe un único camino percolante.

Camino no percolante (non_percpath): equivalente a las áreas formadas por nodos o enlaces vecinos que tenían una probabilidad inferior a la crítica.

Su estudio tiene gran interés ya que un gran número de caminos no percolantes pueden tener más peso en las propiedades del suelo que el propio camino percolante.

Tortuosidad: es la relación entre la longitud del camino percolante más corta, eliminando los “backbones” (costillas), y la longitud del camino más corto teóricamente posible, es decir, la línea recta. Este parámetro es igual a 1 cuando la longitud del perc_path es una línea recta y, por tanto, el camino no es sinuoso. Este parámetro es importante, ya que tiene mucho peso a nivel de optimización de caminos, como la salida más directa de petróleo o de agua contaminada.

2.3. APLICACIONES

La teoría de percolación tiene numerosas aplicaciones: el estudio de un fluido en medio poroso (nuestro caso), la propagación de enfermedades en la población, la formación estocástica de las estrellas o la formación de glaciares. Esta polivalencia se atribuye a la originalidad y potencia del modelo de percolación. A continuación se describen algunas de las aplicaciones más gráficas e intuitivas.

Propagación de incendios

Imaginemos un bosque representado por una malla cuyos nodos están ocupados por un árbol (p) o no ($q = 1 - p$). Cada nodo puede estar en uno de los siguientes estados: árbol no quemado, quemándose, o no hay árbol. La propagación puede simularse así: el punto de ignición se escoge de forma aleatoria, como es el caso de la ignición de fuego debido a un rayo. Una vez el árbol está en llamas, existe una probabilidad (p) de que el fuego se extienda al árbol vecino y así sucesivamente. Podría resultar de interés determinar la probabilidad que cierto punto sea alcanzado por el grupo ardiente, o el recorrido que tendrá el fuego atravesando el bosque.

Polimerización

En este caso asumimos que pequeñas moléculas ramificadas forman macromoléculas más y más grandes creando enlaces entre ellas. Un ejemplo de este proceso, transición líquido-gel, es el que tiene lugar cuando se hierve un huevo.

Resistencia eléctrica en una mezcla de dos medios

Se mezclan dos materiales, A y B, presentados en forma de partículas diminutas, donde A es un material conductor y B es un aislante perfecto. Se aplica una diferencia de potencial a la mezcla desordenada, esperando que cada partícula diminuta en la mezcla sea escogida de forma aleatoria para ser del tipo A o B. Las partículas vecinas de tipo A forman grupos que conducen electricidad. Si el cluster de percolación abarca las dos caras opuestas (electrodos), la resistencia eléctrica es limitada o ilimitada en función de la probabilidad. Para $p < p_c$ la resistencia es ilimitada, mientras que para $p > p_c$ la resistencia es limitada.

Medicina

Al estudiar una determinada enfermedad, tratamos la población expuesta como un conjunto de nodos conectados entre sí. Podremos predecir el riesgo de contagio o el porcentaje de afectados a partir del cual, dicha población sufre una epidemia.

Campos Petrolíferos

La percolación se puede usar como un simple modelo idealizado para predecir la distribución del petróleo o del gas dentro de las cuencas sedimentarias.

La malla considerada en este caso está compuesta por poros llenos de petróleo que pueden estar conectados, correspondiendo a nodos abiertos en el modelo de percolación, o pueden estar aislados.

La probabilidad con la que un lugar está ocupado será función de la porosidad o el promedio de concentración de petróleo en la roca. Para obtener una buena producción de petróleo en un pozo, es deseable situarlo en un área con alta porosidad, es decir, cuanto mayor sea el *alfacrit*, más rentable será la explotación.

Para predecir la cantidad de petróleo que se producirá necesitamos estimar la porosidad de la roca en el área donde la reserva de petróleo estará. La dificultad viene del hecho que las muestras son normalmente testigos de roca con diámetros del orden de Cm. o Km., así que para obtener el resultado final necesitamos extrapolar la medida a la escala de la reserva, que por lo menos está ordenada en Km.

La teoría de percolación predice que este método es válido cuando la probabilidad en el problema de la percolación es sensiblemente más alta que el umbral de la percolación. Por otro lado, si la probabilidad está cerca del umbral y existe un grupo extendido que puede estar bastante ramificado, es posible que la muestra contenga otros grupos (camino no percolante) que no puedan ser alcanzados. En este caso la decisión tomada podría no ser provechosa.

La aplicación de la teoría de percolación es muy común en el mundo del oro negro, ya que pretende utilizarse como herramienta para una mayor rentabilidad de la exploración y producción de hidrocarburos.

3. TEORÍA DE GRAFOS

3.1 CONCEPTOS BÁSICOS DE LA TEORÍA DE GRAFOS

Los inicios de la teoría de grafos datan del s. XVIII, época en que el matemático suizo Leonhard Euler obtuvo la resolución del llamado “Problema de los puentes de la ciudad de Königsberg”. En la **Figura 9** observamos que la ciudad queda dividida en cuatro zonas de tierra (A, B, C y D) debido al paso del río Pregel. Se planteó la cuestión de si era posible, empezando en una de las zonas de tierra, atravesar los siete puentes una sola vez y volver al punto inicial. Viendo que era imposible, demostró que el grado de cada vértice tiene que ser par para que exista un camino que recorra las aristas una sola vez y termine en el punto de inicio. Fue bautizado como *Camino Euleriano*.

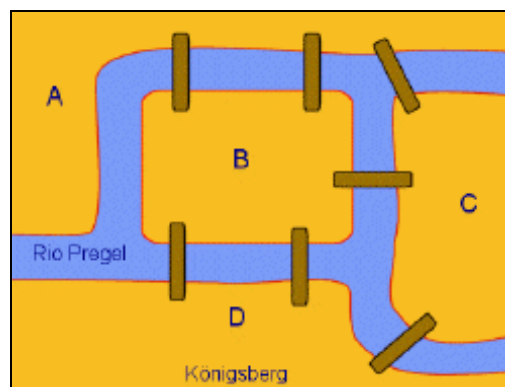


Figura 9: Primer grafo, Puentes de Königsberg [11].

La teoría de grafos tuvo un nuevo impulso gracias a Gustav Kirchhoff, quién en 1847 utilizó los grafos como modelos teóricos de redes eléctricas. Más adelante llegarían nombres célebres como William Rowan Hamilton y Casimir Kuratowski, que ahondarían en el estudio de los grafos regulares y sus ciclos (caso de Hamilton) y en la caracterización de los grafos planos (caso de Kuratowski).

Enmarcada en la Matemática discreta, la teoría de grafos proporciona una herramienta muy útil, potente e intuitiva a la hora de diseñar modelos teóricos de distintas clases de redes de interconexión. El gran avance experimentado por la teoría de grafos se debe al interés suscitado por su gran utilidad en el mundo de la informática. Muchas propiedades teóricas así como algorítmicas de los grafos resultan de gran aplicación cuando se desea estudiar estas mismas propiedades en problemas prácticos, tales como la búsqueda del camino más corto en una red dada, o el diseño de redes que permitan el envío de mensajes u otra clase de elementos de la manera más eficaz posible.

La elaboración de este capítulo ha sido extraída de los libros [12] - [15].

3.1.1. CONCEPTOS BÁSICOS

Definición de Grafo: se llama grafo a la estructura combinatoria $G = (V(G), A(G))$ formada por un conjunto de puntos $V(G) = V = \{v_1, v_2, \dots, v_p, \dots\}$ llamados *vértices* o *nodos* y por un conjunto $A(G) = A = \{a_1, a_2, \dots, a_p, \dots\}$ cuyos elementos son llamados *aristas* o *enlaces*, tales que $a_j = \{v_b, v_k\}$ donde $v_b, v_k \in V$.

Se dice que G es un grafo cuando las aristas no están orientadas y se escribe: $a_i = uv$.

Si las aristas están orientadas, se dice que se trata de *arcos* y escribiremos (v_b, v_k) , entonces el grafo será dirigido y se le denomina *digrafo*. Una de las aplicaciones más importantes de los digrafos es la de hallar el camino más corto desde un nodo origen hacia un destino en una red de transporte.

En particular, un grafo da lugar al denominado digrafo simétrico en el sentido de que su conjunto de arcos contiene las aristas $(v_b, v_k) = a_i \in A$ y $(v_k, v_b) = a_i' \in A$.

3.1.2. TIPOS DE GRAFOS

Grafo regular: Es aquel grafo en el que todos sus vértices tienen el mismo grado. Si dicho grado es k , se le denomina *k-regular*.

Grafo completo: Grafo que contiene una arista entre cada par de vértices. Todo grafo completo es también regular, ya que cada vértice tiene grado $v-1$, al estar conectado con el resto de vértices. Un grafo completo con v vértices se denota K_v . El número de aristas es el siguiente:

$$\frac{v(v-1)}{2} = \binom{v}{2} = a \quad (5)$$

Grafo bipartito: Grafo en el que su conjunto de vértices puede partirse en dos conjuntos disjuntos denominados “clases”, de tal forma que no haya adyacencias entre vértices pertenecientes a la misma clase.

Grafo bipartito completo: Se denota $K_{m,n}$, y consiste en dos clases de vértices de cardinales m y n respectivamente, de manera que cada vértice en la m -clase tiene grado n , y cada vértice de la n -clase tiene grado m . El número de aristas es $a = m \cdot n$.

Grafos isomorfos: Los grafos G_1 y G_2 son isomorfos y f es un *isomorfismo* cuando dos grafos (G_1 y G_2) tienen exactamente la misma forma, es decir, existe una biyección f entre el conjunto de vértices de G_1 y el conjunto de vértices de G_2 , tal que $\{f(x), f(y)\}$ es una arista de G_2 si, y sólo si, $\{x, y\}$ es una arista de G_1 .

En la **Figura 10** se muestran dos grafos isomorfos G_1 y G_2 . El isomorfismo es el siguiente: $f(a)=1, f(b)=3, f(c)=4$ y $f(d)=2$. Puede comprobarse que cada arista de G_1 corresponde de forma única a una arista de G_2 .

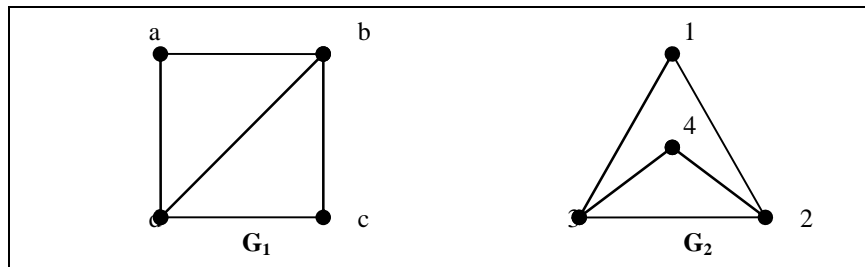


Figura 10. Representación de dos grafos isomorfos.

3.1.3. TEOREMAS BÁSICOS EN GRAFOS

Uno de los teoremas iniciales de la teoría de grafos afirma que la suma de los grados es igual a dos veces el número de aristas.

Teorema 1: Sea $G = (V, E)$ un grafo no dirigido con a aristas, entonces:

$$2a = \sum_{v \in V} \delta(v) \quad (6)$$

Este resultado tiene una aplicación útil denominado *el Lema de los apretones de manos*: “en cualquier grupo de gente, el número de personas que dan la mano a un número impar de personas es par”.

Teorema 2: (Lema de los apretones de manos) todo grafo no dirigido tiene un número par de vértices de grado impar.

Demostración: Por el Teorema 1 sabemos que:

$$2a = \sum_{v \in V_p} \delta(v) + \sum_{v \in V_i} \delta(v) \quad (7)$$

Por tanto deducimos que:

$$2a - \sum_{v \in V_p} \delta(v) = \sum_{v \in V_i} \delta(v) \quad (8)$$

Como la primera parte de la igualdad es la diferencia de dos números pares, se obtiene que la suma de los grados de los vértices impares es par.

Si el grafo es dirigido, el número máximo de arcos es $v(v-1)$ donde v representa el número de vértices. Y el número exacto de aristas de un grafo dirigido con v vértices se calcula en el teorema siguiente:

Teorema 3: Sea $G = (V, E)$ un grafo dirigido, entonces:

$$\sum_{v \in V} \delta^-(v) + \sum_{v \in V} \delta^+(v) = |E| \quad (9)$$

Matriz de adyacencia de un grafo G

Llamamos *matriz de adyacencia* de G a la matriz $A(i,j)$ de orden $v \times v$ donde:

$$a_{ij} = 1 \quad \text{si } \{i,j\} \in A$$

$$a_{ij} = 0 \quad \text{en caso contrario}$$

La matriz de adyacencia siempre es simétrica porque: $a_{ij} = a_{ji}$

La **Figura 11** nos muestra el ejemplo de un grafo y su matriz de adyacencia.

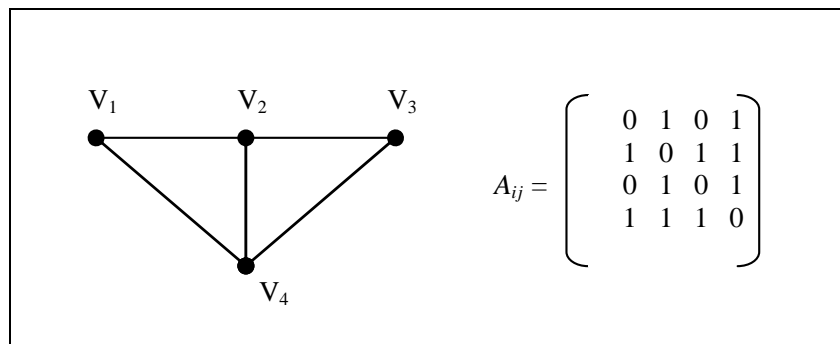


Figura 11. Matriz de adyacencia A_{ij} .

3.2 GRAFOS PLANOS

Un grafo G es *plano* si admite una representación en el plano de tal forma que las aristas no se cortan. La **Figura 12** muestra la llamada *Representación plana del grafo*. Una realización plana de G divide el plano en un cierto número de regiones.

Dentro de la planaridad de los grafos, entra en escena el concepto de *cara* de un grafo. Las aristas de los grafos planos, representadas por curvas de Jordan, tienen la característica de que no intersecan.

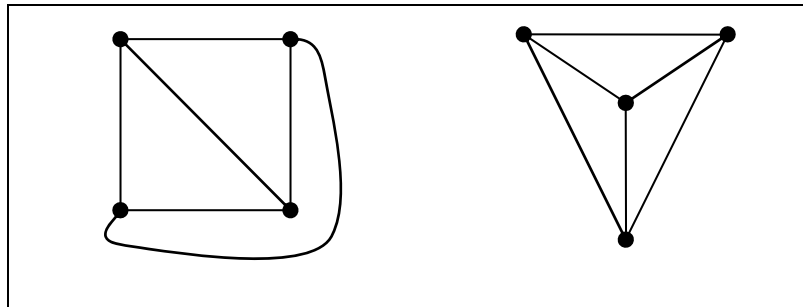


Figura 12. Representación del mismo grafo.

Teorema de la curva de Jordan: definimos C como una curva de Jordan cerrada, tal que $\mathbb{R}^2 - C$ se descompone en dos partes disjuntas E (*exterior*) e I (*interior*), de forma que toda curva de Jordan C' de extremos a y b corta a la curva C en al menos un punto.

Los llamados grafos planos admiten pues una representación en el plano, donde las aristas, dibujadas como curvas de Jordan, no se cortan. En cambio, no todo grafo puede ser representado de forma plana (se demostrará más adelante la no-planaridad de los conocidos los grafos $K_{3,3}$ y K_5).

Revisamos a continuación el método de demostración por inducción, ya que es una técnica empleada en la demostración de los teoremas relativos a grafos planos y en general, de todos los resultados relacionados con números naturales.

Método de inducción en matemática discreta

El método de inducción en matemática es un proceso clásico para probar relaciones entre números enteros. Da lugar a las llamadas “demostraciones por inducción o inductivas”.

Tomamos como ejemplo la identidad aritmética que proporciona la suma de los primeros números naturales:

$$1 + 2 + \dots + n = \sum_{k=1}^n k = \frac{n(n+1)}{2} \quad (10)$$

Sea n_0 un número natural y $P(n)$ una proposición sobre n , para cada número natural $n \geq n_0$:

1. Se muestra que la proposición $p(n_0)$ es verdadera, es decir, se verifica que el primer valor cumple la proposición.

En nuestro ejemplo, la igualdad se cumple trivialmente para $n_0 = 1$.

2. *Hipótesis de inducción*: se supone que la proposición es verdadera hasta un cierto valor k , por ejemplo, $n-1$.

En nuestro ejemplo supondremos que:

$$1 + 2 + \dots + n - 1 = \frac{(n-1)n}{2} \quad (11)$$

3. Demostrar que si $p(n-1)$ con $n-1 \geq n_0$ es cierta, también lo es $p(n)$.

En nuestro ejemplo tendremos:

$$1 + 2 + \dots + n = (1 + 2 + \dots + (n-1)) + n \xrightarrow{\text{hip. inducción}} \frac{(n-1)n}{2} + n = \frac{(n-1)n + 2n}{2} = \frac{n(n+1)}{2} \quad (12)$$

3.2.1. CARACTERIZACIÓN DE LOS GRAFOS PLANOS

La *Fórmula de Euler* es un resultado para grafos planos conexos.

Sea G un grafo plano, conexo, con v vértices, a aristas y r regiones (o caras), incluyendo la región no finita R_∞ , que descomponen el plano, entonces se cumple:

$$v - a + r = 2 \quad (13)$$

Demostración inductiva de la fórmula de Euler, respecto al número de aristas a .

Tomando el cero como valor mínimo de a en el paso inicial.

1º paso: Para $a = 0$ tenemos que $v + r = 2$, por tanto, $v = 1$ y $r = 1$.

2º paso: Suponemos que el resultado es cierto para todos los grafos planos y conexos con $a-1$ aristas, donde $a \geq 1$.

Sea G un grafo plano y conexo con e aristas, si G es un árbol, entonces $v = a + 1$, lo cual lleva a la conclusión que $r = 1$ (la cara infinita). Por tanto, se cumple que: $v - a + r = 2$

Si G no es un árbol, existirá alguna arista a de un ciclo C de G para formar $G - \{e\}$ plano, conexo, con v vértices, $a - 1$ aristas y $r - 1$ regiones o caras, al cual se le puede aplicar la hipótesis de inducción y en consecuencia, se puede afirmar la fórmula de Euler, ya que:

$$v - (a - 1) + (r - 1) = 2 \Rightarrow v - a + r = 2 \quad (14)$$

Como consecuencia del teorema anterior se obtienen los siguientes corolarios:

Corolario 1: Si $v \geq 3$ y G no contiene circuitos de longitud igual a 3, entonces

$$a \leq 3v - 6 \quad \text{y} \quad r \leq 2v - 4 \quad (15)$$

Una de las consecuencias será que el grado de cualquier región debe ser al menos 4. La fórmula de Euler indica que el número de r regiones es un número intrínseco al grafo, no depende de la representación plana considerada.

Corolario 2: Sea G un grafo plano, simple y conexo. Entonces G tiene un vértice de grado menor o igual que 5.

Demostración:

- i) Si G tiene grado uno o dos, se cumple la propiedad.
- ii) Si G tiene como mínimo 3 vértices, por el *corolario 1*: $a \leq 3v - 6$.

De modo que $2a \leq 6v - 12$. Si el grado de todos los vértices fuese mayor o igual que seis, entonces, dado que $2a = \sum_{v \in V} \delta(v)$ (por el teorema de los apretones de manos), tendríamos que $2a \geq 6v$.

Pero esto contradice la desigualdad $2a \leq 6v - 12$. Por tanto, tiene que haber un vértice de grado menor o igual que cinco.

Grafo dual: Sea G un grafo plano, conexo. La construcción de un grafo dual G^* se realizará convirtiendo las caras del grafo inicial en los vértices del nuevo grafo G^* , y dos vértices serán adyacentes si, y sólo si, las caras correspondientes compartían una arista en el grafo inicial G .

Subdivisión elemental de un grafo $G(V, E)$: se produce cuando se suprime una arista $e = uv$, añadiendo un vértice u , para finalmente añadir dos nuevas aristas uv y uw .

Sean las aristas uv , uw y su vértice incidente u de grado 2 de un grafo G , obtenemos el grafo $G' = G - u + uv$. Diremos entonces que el grafo G' se ha formado por la *fusión* de las aristas uv y uw .

Grafos homeomorfos: dos grafos G_1 y G_2 se consideran homeomorfos si existen dos grafos G_1' y G_2' isomorfos, que hayan sido obtenidos mediante una sucesión de subdivisiones elementales y/o fusiones desde G_1 y G_2 , respectivamente.

GRAFOS $K_{3,3}$ Y K_5

Cualquier intento de dibujar los grafos $K_{3,3}$ y K_5 en el plano sin que se corten ninguna de sus aristas está condenado al fracaso. Veamos a continuación el porqué.

La **Figura 13** muestra la representación del grafo $K_{3,3}$.

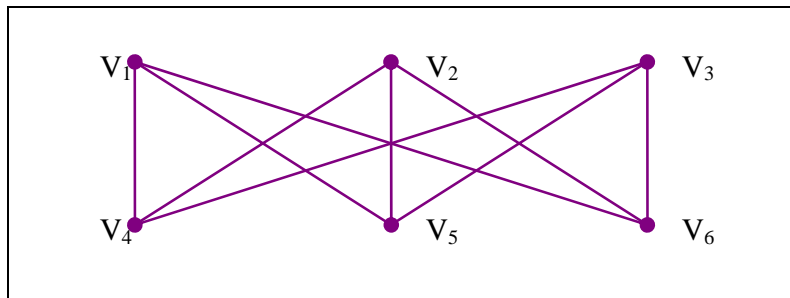


Figura 13. Representación del grafo $K_{3,3}$.

Los vértices v_1 y v_2 tienen que estar conectados tanto con v_4 como con v_5 . Las cuatro aristas de unión forman una curva cerrada que divide el plano en dos regiones, R_1 y R_2 . El vértice v_3 estará en R_1 o R_2 . Si v_3 está en R_2 , el interior de la curva cerrada, las dos aristas que conectan v_3 con v_4 y v_3 con v_5 dividen R_2 en dos subregiones, R_{21} y R_{22} . Es entonces cuando se observa que no existe probabilidad de colocar el vértice v_6 sin producir un corte (ver **Figura 14**), es decir:

- Si v_6 está en R_1 , entonces no se puede dibujar la arista entre v_6 y v_3 sin que ésta sea cortada.
- Si v_6 está en R_2 , observamos que no se puede dibujar la arista entre v_2 y v_6 sin producir corte.
- Si v_6 está en R_{21} , no se puede dibujar la arista entre v_1 y v_6 sin producir corte.

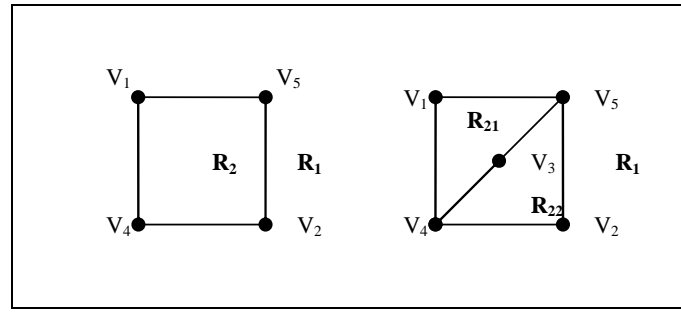


Figura 14. Dos regiones del grafo $K_{3,3}$.

Una forma rápida de demostrar la no-planaridad del grafo $K_{3,3}$ es mediante el corolario de planaridad en que $a \leq 2v - 4$. Se trata de un grafo conexo pero no contiene circuitos de longitud 3, ya que es bipartito. Por tanto, tenemos 6 vértices y 9 aristas:

$$\text{con } a \leq 2v - 4 \Rightarrow 9 \leq 8 \Rightarrow K_{3,3} \text{ NO ES PLANO}$$

En la **Figura 15** se representa el grafo K_5 , que tiene $\binom{5}{2} = 10$ aristas, es decir, que cada conjunto con 2 elementos es una arista.

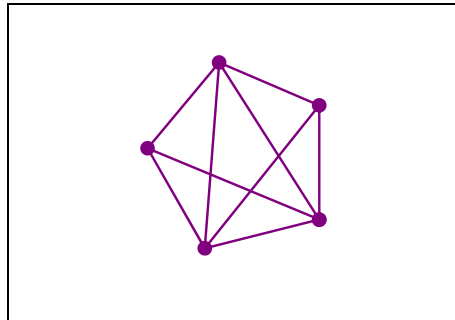


Figura 15. Grafo K_5 .

Se contabilizan 5 vértices y 10 aristas, que aplicadas al Corolario 1 de planaridad de grafos, obteniendo:

$$\text{con } a \leq 3v - 6 \Rightarrow 10 \leq 9 \Rightarrow K_5 \text{ NO ES PLANO}$$

Teorema de Kuratowski

Un grafo $G(V, E)$ es planar \Leftrightarrow no contiene ningún subgrafo homeomorfo a K_5 y $K_{3,3}$.

La afirmación de que un grafo plano no puede contener un subgrafo homeomorfo a K_5 y $K_{3,3}$ es sencilla. En cambio, la demostración recíproca, en que todo grafo que no contiene un subgrafo homeomorfo a K_5 y $K_{3,3}$ es plano, resulta más complicada y no se detallará en esta tesina.

3.2.2. COLORACIÓN DE GRAFOS

Cuando se intenta colorear un mapa, se suele asignar distintos colores a las regiones que tienen una frontera en común. Esta norma se aplica para facilitar la visualización del mismo. ¿Cómo podemos garantizar que no repetiremos color? Asignándole tantos colores como número de regiones. Pero esta solución no suele ser muy eficiente ya que, en ocasiones, necesitaríamos de una completa gama de colores que resultaría poco útil.

Por tanto, debemos utilizar el menor número de colores siempre que nos sea posible. Supongamos un continente hipotético llamado *Atlantis* presentado en la **Figura 16**, distribuido por dos reyes en distintas regiones.

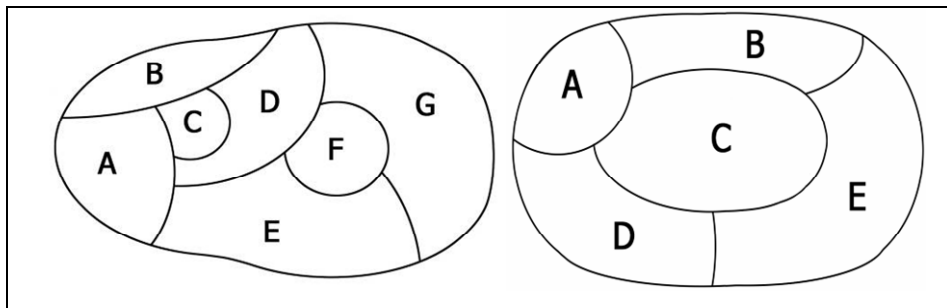


Figura 16. Distribución del Rey Sol y del Rey Luna.

A partir de la **Figura 17** se observa como resolver este problema asociándole a cada región un vértice y una arista a cada par de regiones fronterizas. De ese modo asimilamos la acción de “pintar un mapa” a la *coloración de un grafo*.

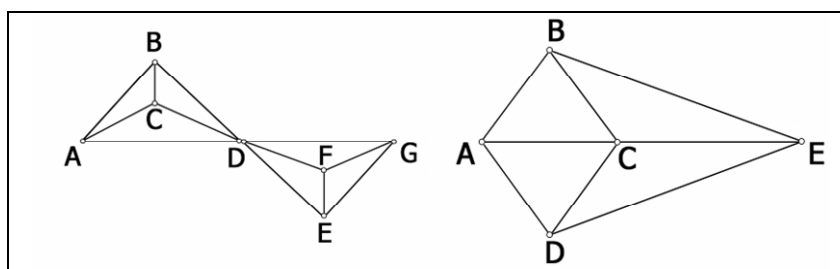


Figura 17. Grafo del Rey Sol y del Rey Luna.

Una **coloración** de un grafo simple consiste en asignarle un color a cada vértice del grafo, de manera que a cada dos vértices adyacentes se les asignen colores distintos. Debemos buscar una coloración que utilice menos colores que el número de vértices del grafo. En consecuencia, definimos el **número cromático** de un grafo como el número mínimo de colores que se requieren para una coloración del grafo.

Teorema de los cuatro colores

En 1976 los matemáticos estadounidenses Kenneth Appel y Wolfgang Haken demostraron que el número cromático de un grafo plano es menor o igual que cuatro. Esto se denominó Teorema de los cuatro colores.

3.2.3. REDES DE PERCOLACIÓN Y GRAFOS PLANOS

Una vez descritos los conceptos básicos de la teoría de grafos, se observa que los vértices de un grafo plano equivalen a los nodos de una malla de percolación. Del mismo modo, las aristas que unen los vértices son iguales a los enlaces entre nodos. En consecuencia, el modelo teórico y la representación gráfica de una red de percolación son la de un grafo plano.

La conexión entre los nodos y enlaces de la percolación puede ser descrita a través de una matriz de adyacencia de grafos. A partir de dicha matriz se plantea la idea de simular un modelo en el que, introduciendo las posibles conexiones de cada enlace (en función del lugar ocupado en la red), pueda describirse el fenómeno de percolación.

Los grafos duales han sido utilizados para estudiar el fenómeno de percolación. Habitualmente se identifica a cada región con un nodo de la red de percolación. Dos nodos son adyacentes cuando las regiones correspondientes en la red tienen frontera común.

Dado un grafo G particular se construye su grafo G^* dual, por ejemplo, en la **Figura 18** se muestran dos tipos de redes de percolación formadas por una malla y su correspondiente dual.

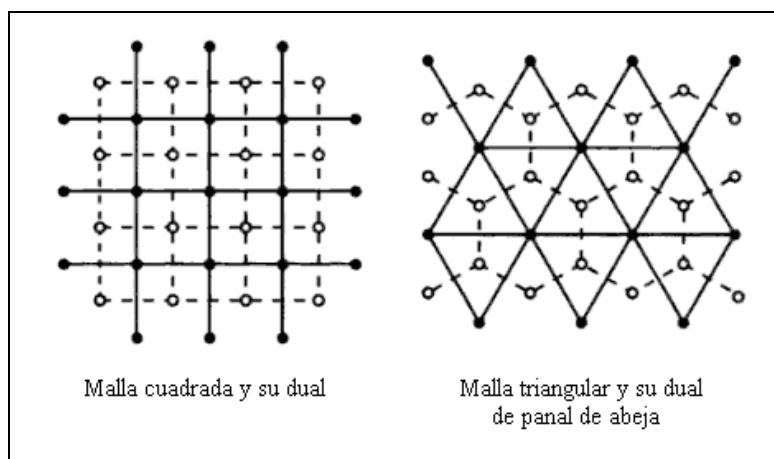


Figura 18. Redes de percolación representadas con mallas planas y sus correspondientes mallas duales [5].

3.3. APLICACIONES

En ocasiones hacemos uso de los grafos sin darnos cuenta, en cambio si realizamos una suma somos conscientes de usar las matemáticas. Se utiliza un símil futbolístico y se asimila un grafo a un centrocampista, éstos juegan un papel menos vistoso pero principal. En los partidos recordamos más al goleador que al organizador, el cual tiene una visión global y distribuye el juego. A continuación se describen algunas de sus aplicaciones.

Redes eléctricas

Gustav Kirchhoff fue el primero en aplicar los grafos con el objetivo de desarrollar un método efectivo para el análisis de redes eléctricas [16].

Isómeros químicos

Cayley estudió el problema de la enumeración de los isómeros de los hidrocarburos saturados C_nH_{2n+2} (ver **Figura 19**). Se trataba de determinar el número de compuestos químicos con idéntica composición (fórmula) pero distinta estructura molecular (disposición distinta de los enlaces). Para ello representó cada hidrocarburo mediante un árbol donde los vértices representaban los átomos (de grado uno los de hidrógeno y cuatro los de carbono) y las aristas los enlaces químicos [16].

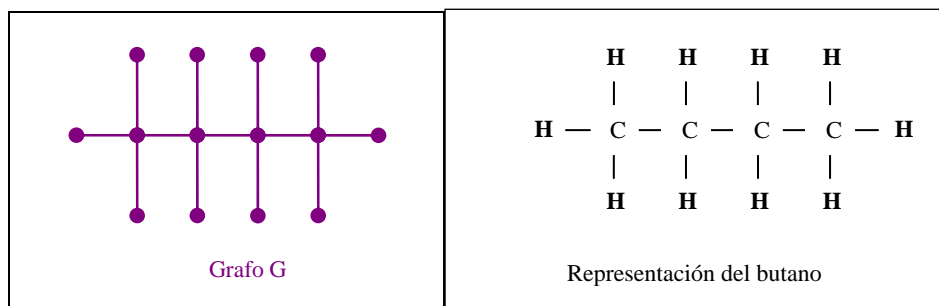


Figura 19: Grafo G correspondiente a la representación del butano C_4H_{10} .

Redes de transporte

El mejor modelo para optimizar el flujo de elementos desde un origen a un destino es un digrafo. Por ejemplo, una red urbana se modela representando cada intersección por un nodo, siendo las aristas las calles propiamente dichas. Una vez establecido el modelo teórico, puede procederse a encontrar el algoritmo adecuado de generación de árboles de caminos más cortos. Dado un punto de salida y otro de llegada, podemos analizar mediante los grafos cuál será el camino más rentable en términos temporales, así como la ruta que

conecte un mayor número de puntos de interés (el plano del metro no es más que la representación de un grafo).

Internet

Las redes del pequeño mundo están recibiendo mucha atención actualmente. Son grafos densos en el sentido de que tienen gran cantidad de nodos y lo que se pretende es estudiar su crecimiento en términos de incertidumbre. Uno de los ejemplos más emblemáticos es tal vez la red de internet.

Representación de un mapa

El anteriormente citado Teorema de los cuatro colores se utiliza para colorear un mapa. De ese modo podremos asegurar que a zonas con límites en común no les corresponderá nunca el mismo color.

Existe un largo etcétera de aplicaciones de la teoría de grafos, algunos tan ilustrativos como la realización de un **calendario de exámenes**, donde debemos asegurar que un mismo alumno no tenga dos exámenes el mismo día. Con los grafos podemos optimizar los días de exámenes sin perjudicar a ningún alumno. Del mismo modo podemos usar la teoría de grafos para diseñar un **Mundial de fútbol** donde los equipos puedan ser emparejados sin importar el resultado y llegar cualquiera de ellos a la final. También en la realización de un **árbol genealógico** se usan los grafos.

4. PROGRAMACIÓN

4.1. OBJETIVOS

En el capítulo 3 se ha comentado que una red de percolación puede ser modelada mediante un grafo plano. Entre los estudios de percolación las similitudes entre las teorías de percolación y de grafos. Muchos estudios de percolación asimilan su red a un grafo [17] y [18]. Este programa se desarrolla en lenguaje FORTRAN.

BINGO_PERCOLATION nace con la finalidad de simular el flujo en medio poroso mediante un modelo de percolación de aristas (*Bond Percolation*). En este trabajo se asimilan los nodos de percolación como vértices de un grafo plano. De la misma forma se tratan las aristas o enlaces. La conectividad entre nodos y aristas es uno de los elementos clave en la teoría de percolación. Es más, se puede considerar que es uno de los principales puntos donde las teorías de percolación y de grafos convergen. El objetivo del programa *BINGO_PERCOLATION* es el de obtener una serie de estadísticos sobre la percolación a partir de una simulación de la misma en un medio poroso.

4.2. EL PROGRAMA BINGO-PERCOLATION

El programa *BINGO_PERCOLATION* realiza la simulación de la percolación en un medio poroso por el que circula un fluido. Los caminos que van formándose a medida que se activan las aristas serán llamados *paths*, los cuales formarán parte del conjunto de caminos “no percolantes” hasta el momento en que se produzca la percolación (cuando obtengamos nuestro primer “camino percolante”). El objetivo principal del programa *BINGO_PERCOLATION* es el de observar la evolución de dicho fluido desde su estado inicial (o de no saturación) hasta que el 100% de la red está ocupada por el fluido (estado de saturación), con especial atención al momento de la formación del primer camino percolante.

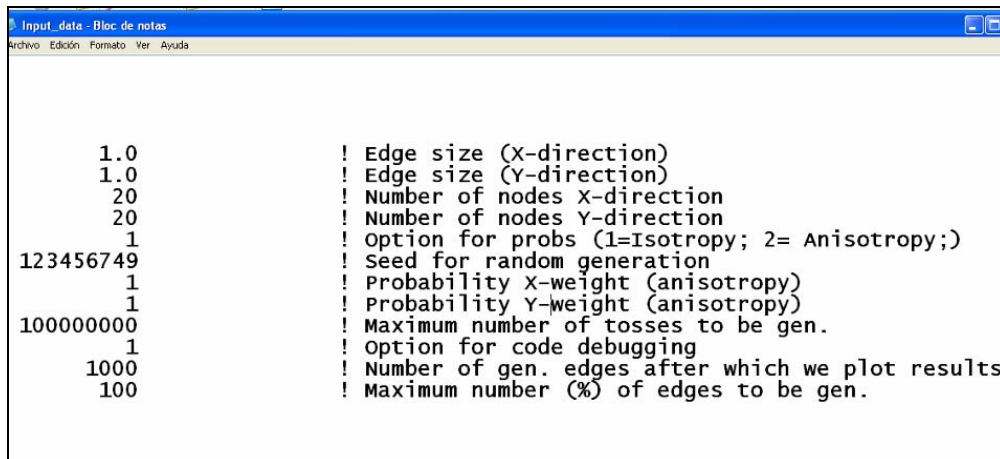
Se consideran grafos planos particulares constituidos por v vértices o nodos y a aristas o enlaces, distribuidos en una red rectangular de magnitud vertical X y magnitud horizontal Y . El programa genera números de forma aleatoria (con restricciones como la no repetición o la preferencia del flujo en una dirección), almacenando la información de los caminos candidatos a percolar. En el momento de la percolación (cuando el fluido ha atravesado la red de un extremo al otro mediante aristas conectadas), el programa nos indica que se ha producido la percolación, dándonos el valor de *alfacrit*. No obstante, el programa continúa generando aristas y almacenando los caminos no percolantes y los percolantes, hasta llegar al 100%, es decir, hasta tener toda la red ocupada.

4.2.1. PARÁMETROS Y METODOLOGÍA

Los programas informáticos pueden resultar realmente farragosos de explicar, incluso cuando la misión del mismo es realizar una simple suma. Con el fin de explicar el programa de la manera más sencilla posible, se asemeja el *BINGO_PERCOLATION* a una receta de cocina. Se describe en primer lugar la entrada de datos (ingredientes), la metodología empleada (elaboración) y finalmente los parámetros percolantes que se obtienen.

Entrada de datos

El archivo *input-data.dat* (ver **Figura 20**) contiene los valores de los datos iniciales y su descripción.



```

1.0      ! Edge size (X-direction)
1.0      ! Edge size (Y-direction)
20       ! Number of nodes X-direction
20       ! Number of nodes Y-direction
1        ! Option for probs (1=Isotropy; 2= Anisotropy;)
123456749 ! Seed for random generation
1        ! Probability X-weight (anisotropy)
1        ! Probability Y-weight (anisotropy)
100000000 ! Maximum number of tosses to be gen.
1        ! option for code debugging
1000     ! Number of gen. edges after which we plot results
100      ! Maximum number (%) of edges to be gen.

```

Figura 20. Archivo Input_data.

- ✓ *Edge Size*: tamaño de la arista en dirección X e Y. Con ellos podemos estrechar o ensanchar una dimensión determinada de forma que el número de nodos en ambas direcciones continúe siendo el mismo.
- ✓ *Number of nodes*: número de nodos en dirección X e Y que forman el grafo.
- ✓ *Option of prob*: tipo de probabilidad utilizada. El caso isótropo corresponde a la opción 1 y el anisótropo a la opción 2.
- ✓ *Iseed*: número o semilla de la generación de números aleatorios. Se recomienda utilizar un número elevado e impar.
- ✓ *Probability X(Y)-weight*: la opción anisótropa nos hace dar un peso determinado a cada dirección. Por ejemplo, si el flujo circula con el doble de facilidad en sentido horizontal, colocaremos un 2 en el peso X y un 1 en el Y. En el caso isótropo no es necesario rellenar estos campos.
- ✓ *Maximum number of tosses*: número máximo de tiradas aleatorias que se generarán durante la ejecución. Debemos asegurarnos que dicho número sea lo suficientemente alto para llegar a activar el 100% de las aristas.
- ✓ *Maximum number of edges to be gen(%)*: porcentaje de aristas generadas.

Parámetros percolantes

Los resultados corresponden a los parámetros percolantes que interesan en el estudio, los cuales son almacenados en distintos archivos, en los que destacamos:

- *ALFACRIT* (p_c): el llamado umbral de percolación se presenta en unidades de porcentaje de aristas generadas en el momento de la percolación. Así, un *alfacrit* igual a 50, será un umbral de percolación correspondiente a una probabilidad $p_c = 0,5$.
- *NON_PERCPATH*: grupo correspondiente a cada uno de los caminos no percolantes que se forman a medida que se generan las aristas de forma aleatoria. El archivo también incluye la longitud del camino no percolante (*LENGTH_NONPERC*).
- *PERC_PATH*: grupo correspondiente al camino o caminos percolantes que se forman en el momento de la percolación. El archivo también incluye su longitud (*LENGTH_PERC*).
- El archivo de salida *OUTPUT* nos indica el número de caminos percolantes y no percolantes que se van formando paulatinamente, así como el número de arista en el que se van produciendo.

Metodología

Como vemos, los ingredientes del plato son sencillos e intuitivos.

La **Figura 21** indica la distribución de los vértices y de las aristas y su relación de vecindad. En este caso se presenta como ejemplo una red de dimensión (4x4) observando que las relaciones entre aristas dependen del lugar que ocupan.

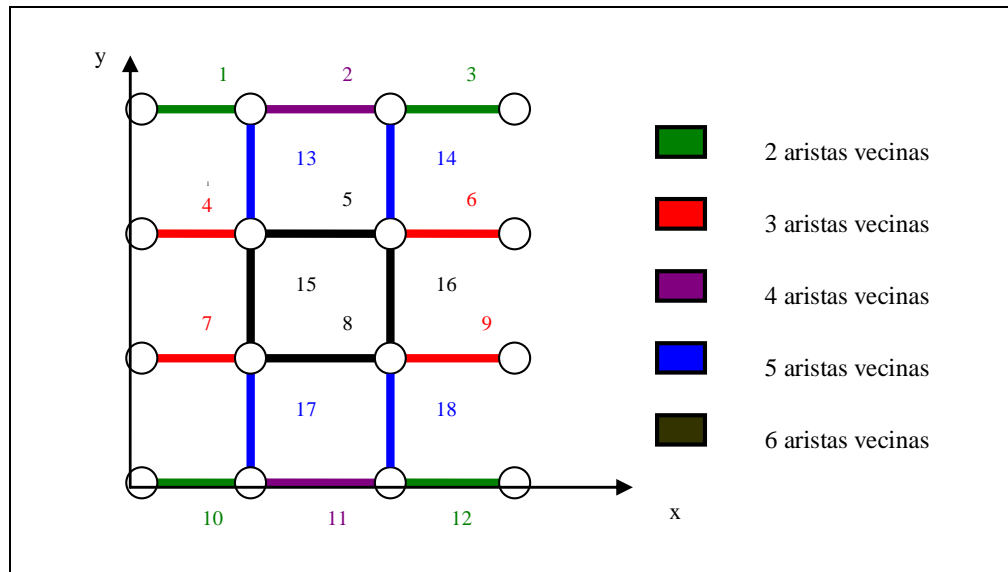


Figura 21. Relación de aristas vecinas en una malla (4x4).

Se define el grafo comenzando por el número de nodos en la dirección X e Y, creándose la tabla de coordenadas nodales (**Tabla 1**). A continuación se etiquetan las aristas indicando los nodos extremos que la forman (**Tabla 2**). Las fronteras en dirección Y se consideran abiertas, es decir, sin aristas que la limiten como en el caso de la dirección X.

Nodo	Coordenada X	Coordenada Y
1	0.000	3.000
2	1.000	3.000
3	2.000	3.000
4	3.000	3.000
5	0.000	2.000
6	1.000	2.000
7	2.000	2.000
8	3.000	2.000
9	0.000	1.000
10	1.000	1.000
11	2.000	1.000
12	3.000	1.000
13	0.000	0.000
14	1.000	0.000
15	2.000	0.000
16	3.000	0.000

Tabla 1. Coordenadas nodales.

Arista	Nodo inicial	Nodo final
1	1	2
2	2	3
3	3	4
4	5	6
5	6	7
6	7	8
7	9	10
8	10	11
9	11	12
10	13	14
11	14	15
12	15	16
13	2	6
14	3	7
15	6	10
16	7	11
17	10	14
18	11	15

Tabla 2. Nodos extremos de cada arista.

De ese modo se obtiene una red definida y etiquetada para poder trabajar en ella. Cabe destacar que la introducción de los datos contenidos en estas tablas se lleva a cabo una sola vez al inicio de la ejecución del programa. Esto es un aspecto a tener en cuenta pues, si la malla de percolación es muy fina (muchos nodos), puede llegar a consumir una cantidad considerable de esfuerzo computacional

Partiendo de la diferenciación entre aristas se diseña una matriz formada por tantas filas como aristas tenga en total el grafo y por seis columnas correspondientes a las posibles aristas vecinas que pueda tener cada una de ellas. La dimensión de la matriz será por tanto (*número de aristas* \times 6), ya que sólo hay seis grados posibles que son, 2, 3, 4, 5 y 6, como indica la **Figura 21**. Cada arista tendrá una fila correspondiente a sus aristas vecinas donde, en caso de ser inferior a 6, será completada con ceros. Así definimos nuestra **Matriz de Conectividades**.

La **Tabla 3** corresponde a la matriz de conectividades del ejemplo 4x4, donde se observa la conectividad entre las aristas de la red. La matriz de localizaciones adoptada aquí se basa en el modelo de la matriz de adyacencia de la teoría de grafos, donde cada vértice formaba una fila y columna con valores 0 o 1 en función de si los nodos estaban o no conectados mediante una arista.

Arista	A ₁	A ₂	A ₃	A ₄	A ₅	A ₆
1	2	0	13	0	0	0
2	3	1	0	0	13	14
3	2	0	14	0	0	0
4	5	13	15	0	0	0
5	6	4	13	14	15	16
6	5	14	16	0	0	0
7	8	15	17	0	0	0
8	9	7	15	16	17	18
9	8	16	18	0	0	0
10	11	17	0	0	0	0
11	12	10	17	18	0	0
12	11	18	0	0	0	0
13	0	15	1	2	4	5
14	0	16	2	3	5	6
15	13	17	4	5	7	8
16	14	18	5	6	8	9
17	15	0	7	8	10	11
18	16	0	8	9	11	12

Tabla 3. Matriz de conectividades.

La matriz de conectividades tiene una gran importancia ya que, una vez se genera la arista aleatoria (*random generation*), el programa revisa en dicha matriz si la arista es vecina de alguna arista que haya sido generada anteriormente. De ese modo, *BINGO_PERCOLATION* incorpora la nueva arista a un camino ya existente (si tienen nodos en común) o en caso contrario, etiqueta un nuevo camino con la arista como único elemento. Los caminos (*paths*) son fácilmente almacenados y así puede conocerse el estado de la conexión de la red en cualquier momento.

El programa comienza almacenando los caminos no percolantes (representados en rojo) y comprobando si existe o no percolación. Ésta se detecta cuando un camino almacenado tiene como extremos las aristas con $y=0$ e $y=\text{máximo}$, unidas mediante aristas conectadas. Será el momento en que aparece el cluster infinito de percolación o el camino percolante.

Una vez tenemos “*Bingo*”, el programa considera que en la red se ha producido la percolación y almacena el camino percolante, que será representado con aristas de color verde. A medida que aumentamos la generación de aristas, se alcanza un porcentaje en el que los caminos no percolantes se conectan con el camino percolante (de mayor longitud). Sería la llamada influencia del “agregado gigante”. Es evidente que llegará un momento en que únicamente tendremos camino percolante, hasta que se generen el 100% de las aristas, donde la red estará totalmente representada en verde.

El proceso queda reflejado en la evolución de una red de dimensión 6x6 representada en la **Figura 22**. En la parte superior izquierda se observa un conjunto de siete caminos no

percolantes. A su derecha se indica el momento en que aparece el cluster percolante, quedando un único camino no percolante en la red. En la parte inferior izquierda se observa el momento en que desaparecen los caminos no percolantes y el cluster percolante es mayor. Finalmente se muestra toda la malla ocupada.

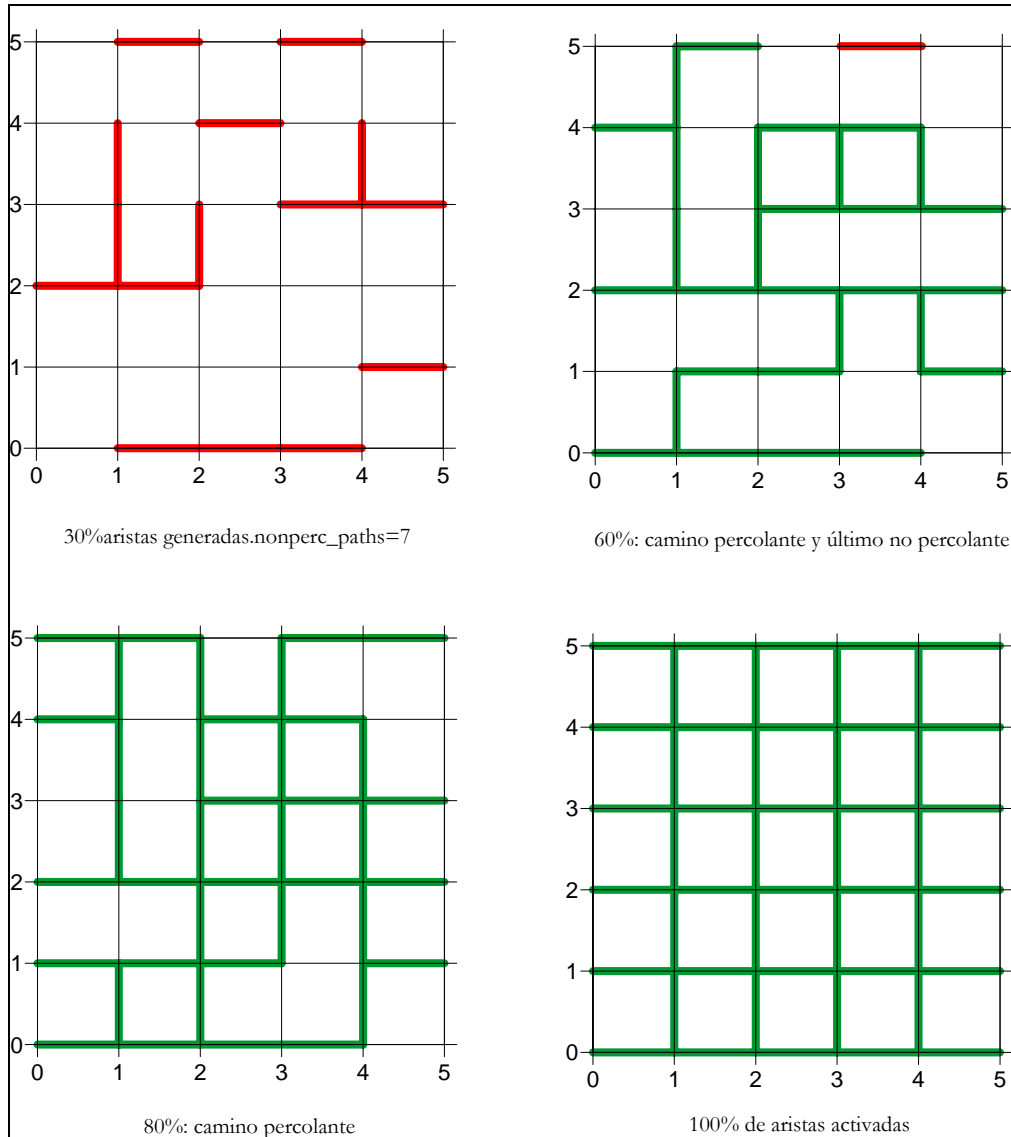


Figura 22. Simulación del BINGO_PERCOLATION en una malla (6x6).

4.2.2. MODELOS UTILIZADOS

La simulación del fenómeno de percolación en un medio poroso mediante el programa *BINGO_PERCOLATION* ha sido tratada en tres casos distintos (ver **Figura 23**):

1) Caso isótropo

Se considera un modelo en el que el terreno tiene condiciones de circulación de flujo isótropas, es decir, las aristas pueden ser generadas en cualquier punto y dirección con la misma probabilidad. El fluido no tiene una dirección preferente de movimiento y, por tanto, el medio podría representar un terreno homogéneo, formado por sedimentos de similar mineralogía y sin fracturación.

2) Caso anisótropo con preferencia vertical

El modelo considera que las aristas verticales tienen más probabilidad de ser generadas que las horizontales. El peso de la probabilidad depende del valor introducido en los parámetros iniciales. En un medio natural podría representarse esta situación imaginando un terreno en el que el material se encuentre fuertemente fracturado y, por tanto, favorezca el flujo vertical a través de las diaclasas. Existen afloramientos en los que es posible ver fracturación subvertical apreciable, por ejemplo, El Cabril Córdoba.

3) Caso anisótropo con preferencia horizontal

En este caso, el terreno presenta mayor facilidad para la circulación horizontal del fluido. Las aristas horizontales tendrán más peso a la hora de generarse. Un ejemplo gráfico sería suponer un terreno formado por un material permeable intercalado con lentes de arcillas, material impermeable. En ese caso, el fluido no podrá atravesar el cuerpo impermeable y circulará horizontalmente hasta encontrar una zona en la que poder descender.

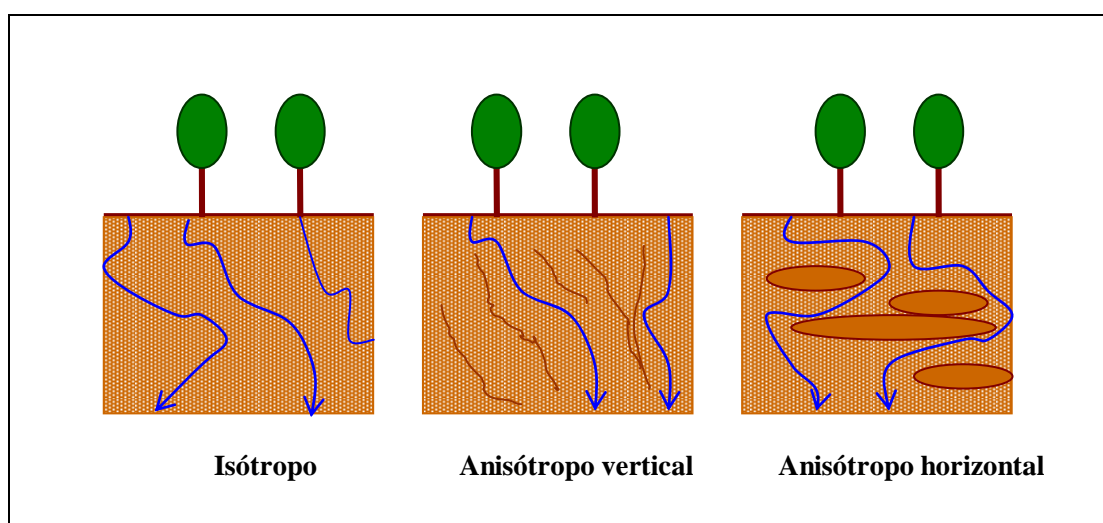


Figura 23. Modelos utilizados en la simulación.

5. EJEMPLO ISÓTROPO

5.1. ANÁLISIS DE UNA SIMULACIÓN

En el caso isótropo se supone un terreno homogéneo en el que no existe una dirección preferente del movimiento del flujo. La permeabilidad es la misma en todas las aristas.

La red elegida para analizar una simulación ha sido la de (20x20), ya que sus dimensiones proporcionan un resultado satisfactorio y cómodo para comentar sus resultados. En este caso la red está formada por los siguientes elementos:

- 400 nodos o vértices (sites).
- 722 aristas o enlaces (bonds): 380 horizontales y 342 verticales (se recuerda que las fronteras verticales se consideran abiertas).
- 100% de aristas generadas.
- Tamaño de la arista en dirección X e Y (Edge Size) igual a 1.

5.1.1. PARÁMETROS PERCOLANTES

En medio poroso se considera relevante realizar el análisis de los siguientes datos:

- Umbral de percolación (Alfacrit).
- Tortuosidad.
- Número de caminos no percolantes (nonperc_paths).
- Longitud del camino percolante (length_perc).

La percolación tiene como parámetro fundamental el umbral de percolación (p_c). En este estudio se presenta en porcentajes siguiendo la siguiente ecuación:

$$ALFACRIT(p_c) = \frac{\text{número_aristas_generadas}}{\text{número_aristas_total}} \times 100 \quad (16)$$

El valor obtenido de *alfacrit* es igual al 50%.

Se toma como semilla el momento de la percolación y se obtienen otros datos de interés analizados a continuación.

La **tortuosidad** se define como la relación entre la longitud del camino percolante más corta (*length_perc_mínimo*), eliminando los “backbones” (costillas), y la longitud del camino más corto teóricamente posible, es decir, la línea recta.

$$\text{Tortuosidad} = \frac{\text{length_perc_mínimo}}{\text{dim}Y - 1} \quad (17)$$

Dada la malla (20x20), el camino percolante más corto estará formado por 19 aristas (dim Y- 1). Así definido, el valor mínimo de la tortuosidad es la unidad.

La **Figura 24** se obtiene graficando el porcentaje de aristas generadas frente a la tortuosidad correspondiente, lo que da una idea intuitiva de la sinuosidad del camino percolante para cada % representado en abcisas.

Los datos de tortuosidad empiezan a registrarse una vez se ha producido la percolación, ya que antes no tiene sentido hablar de tortuosidad si no existe camino percolante. A partir del 80% de aristas generadas, existe el camino en línea recta que atraviesa la malla, es por ese motivo por el que la tortuosidad alcanza su valor mínimo hasta llegar a 1. Antes del 80% podríamos considerar sencillamente que “el fluido se va por las ramas”. Cuanto mayor es el valor de la tortuosidad, más tiempo tarda en atravesar el terreno.

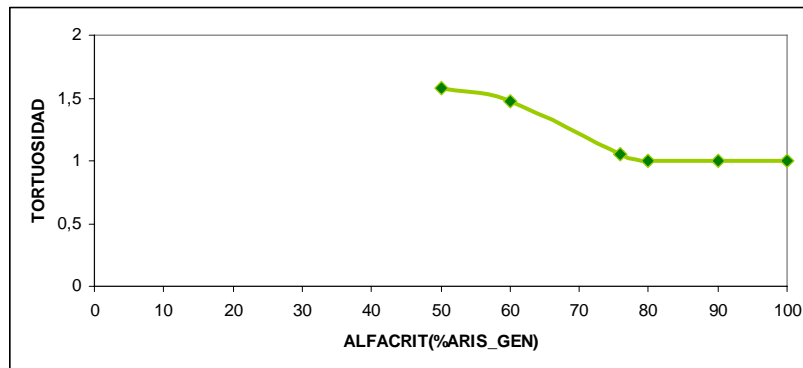


Figura 24. Evolución de la tortuosidad.

El número de caminos no percolantes, llamados *nonperc_paths*, forman pequeñas áreas llenas de fluido que no conectan las fronteras y por tanto, no provoca la percolación.

La **Figura 25** muestra la evolución del número de caminos no percolantes o *nonperc_paths* en función del porcentaje de aristas generadas. Éstos crecen a medida que el porcentaje de aristas generadas aumenta, hasta llegar a un valor en que el efecto de la conexión reagrupa a los distintos caminos no percolantes. El máximo se registra en el 18,42%, con 75 caminos

no percolantes presentes. En el momento de la percolación (50%) quedan aún 19 caminos no percolantes, hasta desaparecer al 76,7%.

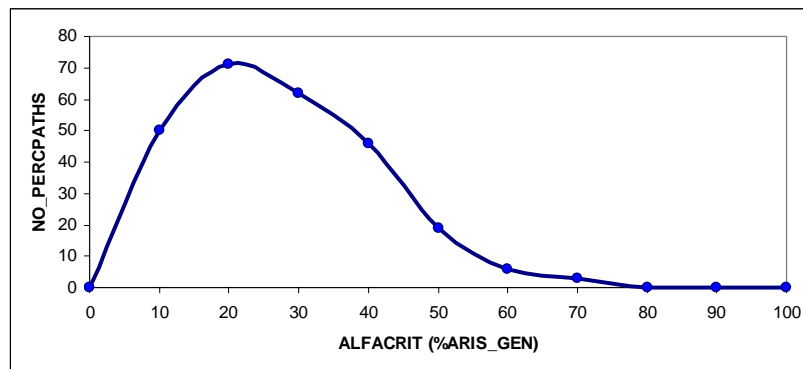


Figura 25. Evolución de nonperc_paths.

El valor del camino percolante, también llamado *perc_path*, aparece evidentemente en el momento en que “cantamos Bingo” y hay percolación. La **Figura 26** presenta el aumento de la longitud del camino percolante, llamada *length_percpath* en función del porcentaje de aristas generadas.

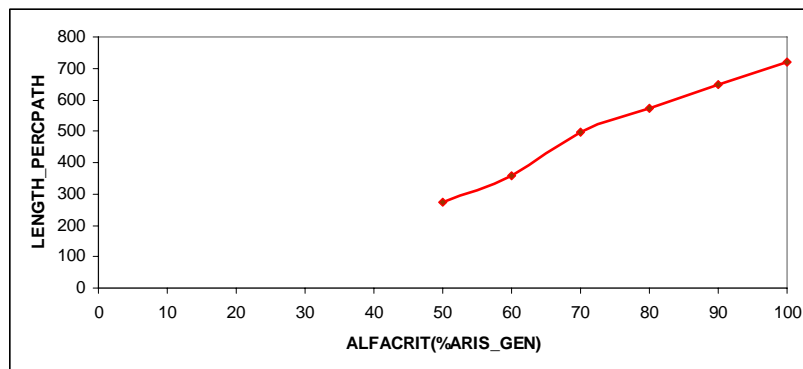


Figura 26. Evolución de length_percpath.

Los valores de la longitud siguen una recta ascendente a partir del 70% de aristas generadas. Otras dimensiones de malla (véase *Anejo A3*) verifican este porcentaje.

Debe hacerse una observación respecto a la existencia de un único camino percolante. Tratándose del caso isótropo en una malla cuadrada se obtiene un camino, hecho que no ocurrirá en casos anisótropos, ni cuando la malla tenga un número de aristas X muy diferente del número de aristas Y tal como reflejamos en capítulos posteriores.

5.1.2. ETAPAS DE LA SIMULACIÓN

El modelo de percolación por aristas de un fluido en un medio poroso isotrópico ha sido concebido a partir de una serie de hipótesis iniciales: cualquier arista tiene la misma probabilidad de ser activada (sea cual sea su dirección). Las aristas correspondientes a las fronteras verticales quedan libres, no pudiendo ser ocupadas. La percolación se produce cuando un camino conecta los límites horizontales de la malla ($y=0$ e $y=\text{máximo}$), siendo vertical el sentido del camino.

Ejecutando el *BINGO_PERCOLATION* (14'54" en tiempo CPU) se obtienen todos los parámetros percolantes citados anteriormente. Su evolución queda reflejada a partir de la siguiente serie de figuras.

La **Figura 27** representa el momento en que el 40% de las aristas están activadas antes de llegar al umbral de percolación. Se puede observar la presencia únicamente de caminos no percolantes (color rojo).

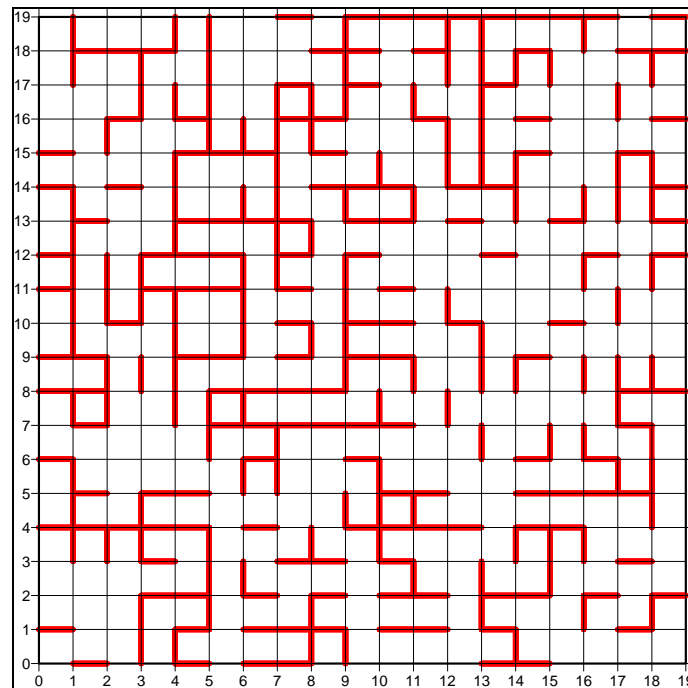


Figura 27. 40% de aristas generadas.

El camino percolante (color verde) aparece en el momento en que se han generado el 50% de las aristas generadas, representado en la **Figura 28**. La longitud del *perc_path* es de 274 unidades, es decir, el cluster percolante ocupa ya un 40% de la red en el momento de la percolación. De todos modos, los caminos no percolantes siguen ocupando un 13% de la red.

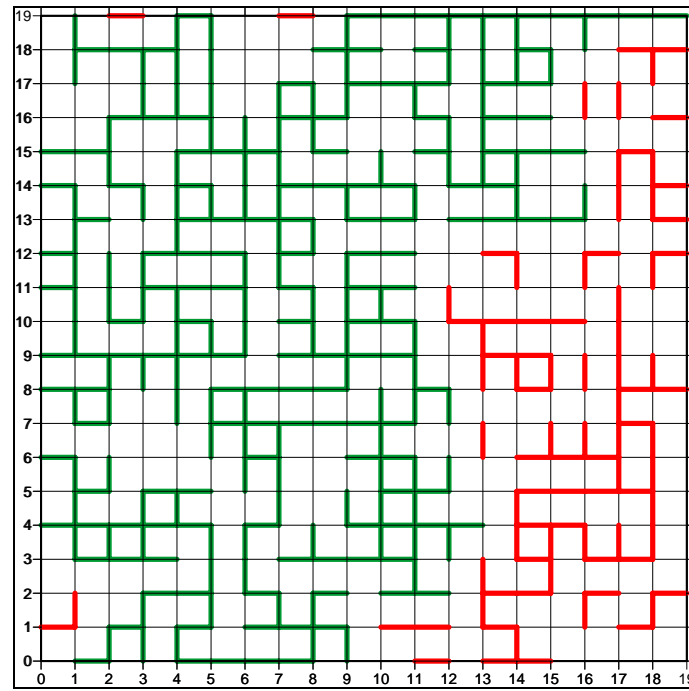


Figura 28. 50% de aristas generadas, percolación.

Como se apuntaba, los caminos no percolantes van siendo engullidos por el cluster de percolación a medida que el porcentaje de aristas generadas aumenta. La **Figura 29** representa el estado de la malla al 60% de la generación de aristas y refleja ya un descenso del número de *nonperc_paths* en favor del camino percolante.

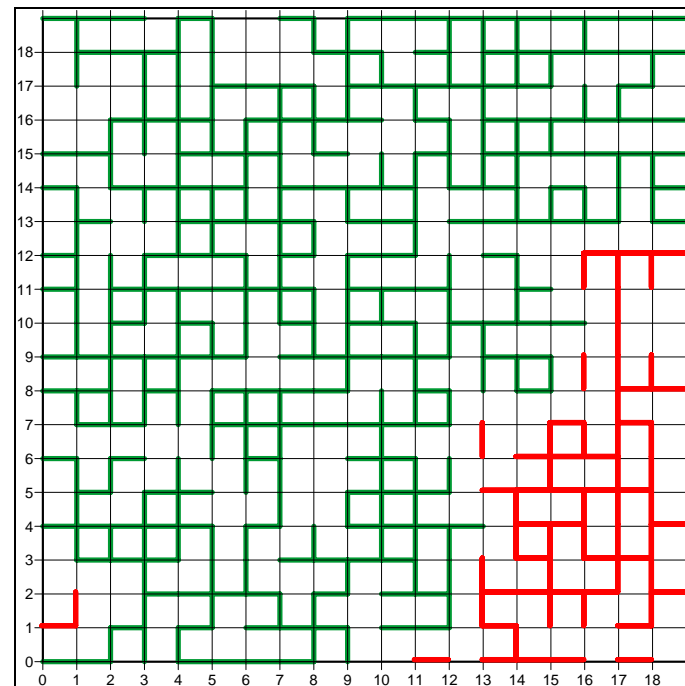


Figura 29. 60% de aristas generadas caminos no percolantes y percolante.

El momento en que los caminos no percolantes pasan a formar parte del cluster de percolación, se produce en el 76,7% (ver la **Figura 30**).

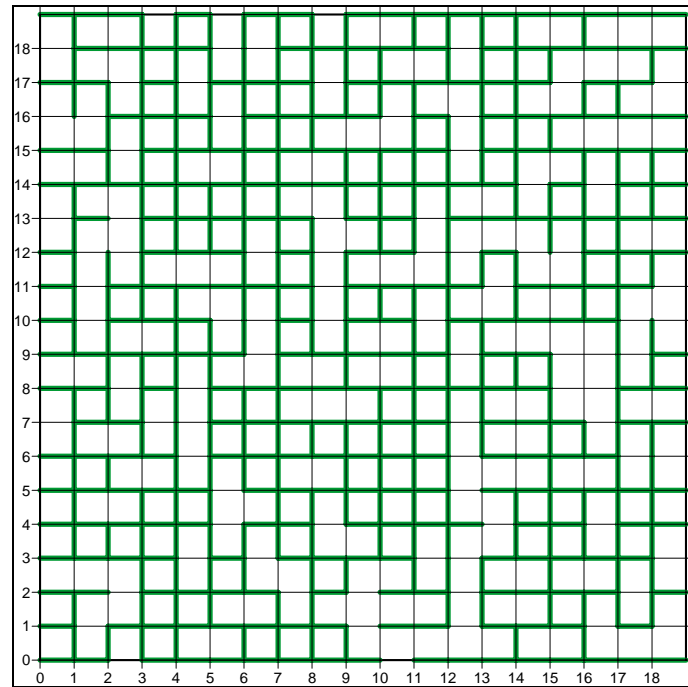


Figura 30. 76,7% presencia única del cluster percolante.

La simulación ha sido realizada tomando un tamaño de aristas igual a la unidad, tanto en la dirección X como en la Y.

5.2. ANÁLISIS DE MONTECARLO

Después de haber realizado un estudio sobre una única simulación, debe plantearse si el modelo isótropo de percolación por aristas es una buena aproximación al caso real de una percolación de un fluido en un medio poroso homogéneo. Con ese fin, se deben utilizar más herramientas para dar credibilidad a los resultados. Por ese motivo se realiza el análisis de MonteCarlo.

A la hora de diseñar el modelo, se definirán las variables aleatorias a estudiar y se iterará las veces que se crean necesarias para obtener una muestra representativa y finalmente discutir los resultados obtenidos.

5.2.1. PARÁMETROS PERCOLANTES

El modelo de percolación por aristas de un fluido en un medio poroso isótropo ha sido concebido a partir de una serie de hipótesis iniciales:

- Toda arista tiene la misma probabilidad de ser activada, sea cual sea su dirección.
- Las aristas correspondientes a las fronteras verticales no se consideran.
- La percolación se produce cuando un camino conecta los límites horizontales de la malla ($y=0$ e $y=\text{máximo}$), siendo vertical el sentido del camino.

Dada una red de dimensiones (20x20), se repite el proceso hasta obtener una muestra representativa. Este estudio ha considerado suficiente una cincuentena de iteraciones. Por tanto, se modifica la semilla de generación aleatoria, obteniendo 50 valores para cada uno de los parámetros percolantes estudiados.

Las variables aleatorias que serán objeto de estudio son las siguientes:

1. *ALFACRIT*: umbral de percolación o porcentaje de aristas generadas en el momento de la percolación.
2. *NONPERC_PATHS*: número de caminos no percolantes en el momento de la percolación.
3. *LENGTH_PERC*: longitud del camino percolante (número de aristas que forman el cluster percolante).

Se obtienen los valores de la **Tabla 4** a partir de las cincuenta iteraciones.

TIRADAS	ALFACRIT	LENGTH PERC	NONPERC-PATHS	TIRADAS	ALFACRI	LENGTH PERC	NONPERC-PATHS
1	51,66	326	16	26	58,59	392	8
2	55,96	380	8	27	54,29	321	19
3	48,75	255	24	28	46,4	205	31
4	46,95	250	21	29	53,88	300	15
5	51,52	249	20	30	49,31	153	20
6	50,42	239	22	31	46,68	187	30
7	53,05	318	19	32	49,45	220	22
8	54,02	326	9	33	50	186	20
9	49,58	237	23	34	50,97	289	23
10	45,98	115	31	35	51,25	221	25
11	51,8	274	21	36	53,88	235	18
12	51,52	263	17	37	48,89	215	24
13	50,28	260	24	38	53,46	303	20
14	48,61	155	25	39	47,51	174	29
15	47,37	177	22	40	46,12	157	29
16	57,2	366	12	41	48,89	244	18
17	50,97	199	19	42	51,52	218	17
18	55,68	342	10	43	49,72	253	28
19	40,86	127	48	44	49,03	195	26
20	52,49	288	20	45	56,37	338	14
21	51,11	292	13	46	43,21	113	33
22	52,77	232	25	47	46,4	163	37
23	48,61	148	21	48	51,94	276	19
24	50,69	169	23	49	50	274	19
25	53,88	278	15	50	50,28	280	19

Tabla 4. Valores de los parámetros percolantes para las cincuenta iteraciones.

A priori, puede anticiparse que las tres variables aleatorias definidas anteriormente, están correlacionadas de la siguiente forma:

- Si $alfacrit \downarrow$ entonces $length_perc \downarrow$
- Si $alfacrit \downarrow$ entonces $nonperc_paths \uparrow$

En consecuencia, si $length_perc \downarrow$ entonces $nonperc_paths \uparrow$

El objetivo es estudiar el tipo de relación (lineal, no-lineal) entre las tres variables. De ese modo se podrá estimar el valor medio aproximado de una de ellas, conociendo valores de las otras dos variables.

La **Figura 31** muestra unos valores de *alfacrit*, en función del número de aristas generadas, comprendidos entre el 40 y 60%. Se puede considerar que los resultados de la variable no presentan una gran variabilidad de resultados.

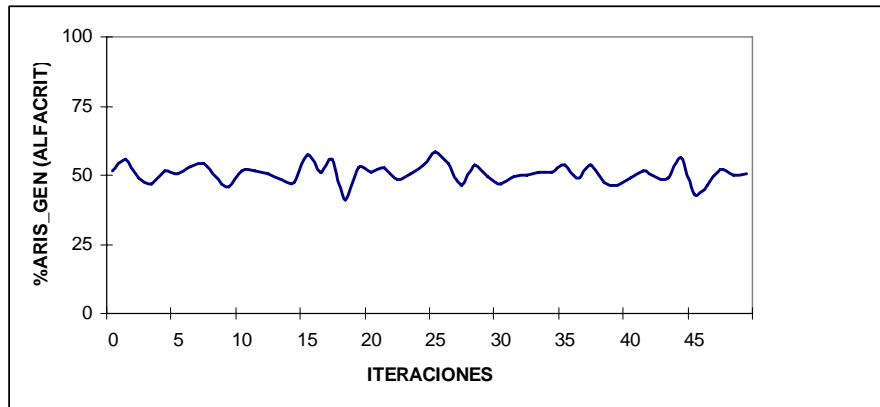


Figura 31. Valores de *alfacrit* para las 50 iteraciones.

En cambio, las **Figuras 32** y **33** muestran que los valores referidos al número de caminos no percolantes y la longitud del camino percolante, no son consistentes en las distintas iteraciones, de modo que existe una fuerte variabilidad de unas simulaciones a otras. Esto indica una alta varianza de dicha variable aleatoria.

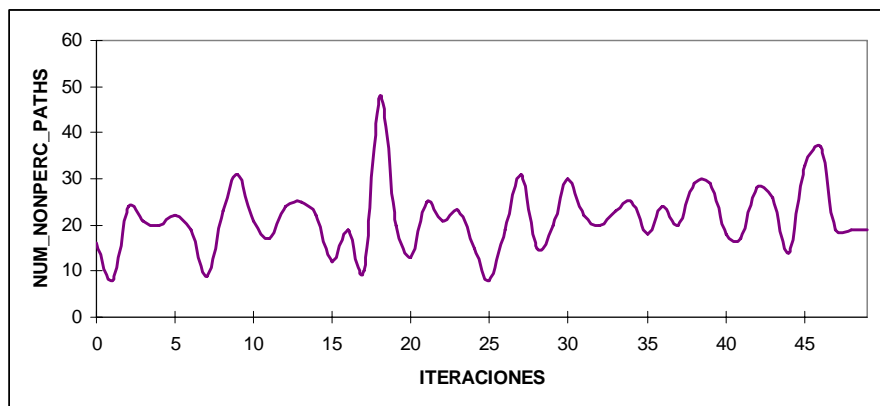


Figura 32. Valores de *nonperc_path* para las 50 iteraciones.

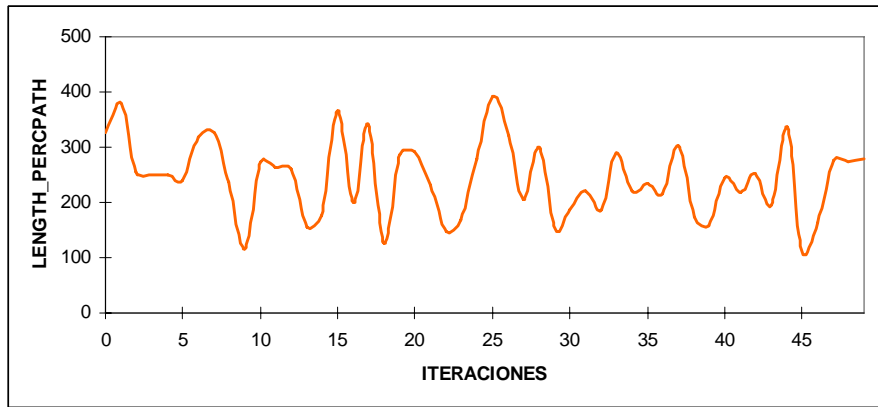


Figura 33. Valores de *length_perc* para las 50 iteraciones.

5.2.2. ESTUDIO ESTADÍSTICO

El objetivo de esta sección es realizar un tratamiento estadístico de los datos recogidos, con el fin de obtener una serie de conclusiones que afianzarán o mermarán las hipótesis iniciales acerca de las variables estudiadas, tales como la relación lineal entre ellas.

Mediante la técnica de regresión lineal y haciendo uso del programa SPSS, se pueden predecir los valores medios de la variable Y (dependiente) a partir de los datos de la variable X (independiente). La diferencia entre el valor predicho por el modelo obtenido y el valor real observado se denomina Residuo.

Dadas las tres variables, *alfacrit*, *nonperc_paths* y *length_perc*, jugarán el papel de variable dependiente o independiente para poder hallar la relación entre ellas.

En el documento se muestran los resultados que dan mejores aproximaciones con el método de regresión lineal. El resto de combinaciones se pueden consultar en el Anejo X.

MÉTODO DE REGRESIÓN LINEAL

Dada una muestra de valor $N=50$, se obtienen los estadísticos descriptivos que se presentan en la **Tabla 5**.

	Mínimo	Máximo	Media	Desv. típ.
ALFACRIT	40,86	58,59	50,5954	3,45053
LENGTH-PERC	113	392	243,54	69,273
NONPERC-PATHS	8	48	21,42	7,376

Tabla 5. Valores descriptivos de las variables.

Para comprobar la variación del parámetro *alfacrit* se recurre a la función de frecuencia. De ese modo se observa en la **Figura 34** que los cincuenta valores de *alfacrit* siguen una distribución normal, con un valor de la media igual a 50,5954. Por tanto, conocido el valor teórico (50), se supone una buena muestra del umbral de percolación.

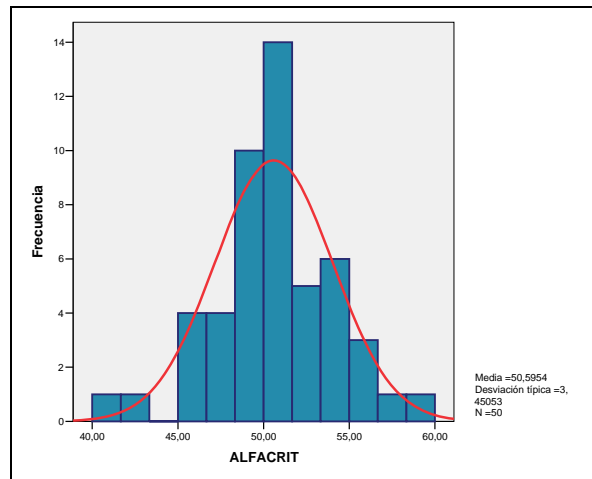


Figura 34. Función de distribución de frecuencia de la variable *alfacrit*.

La **Tabla 6** muestra la matriz de correlaciones entre las variables, la linealidad entre las variables *nonperc_paths* y *alfacrit* es mayor, con una correlación igual a 0,866. Asimismo, el valor de 0,855 correspondiente a la correlación entre *alfacrit* y *length_perc* se considera suficientemente alto (cercano a 1, es decir, linealmente dependientes), lo que implica que habrá problemas de colinealidad si ambas variables son consideradas independientes en una regresión lineal. Por ello se descarta esta opción.

En la parte inferior de la matriz aparecen los valores de significación sobre la hipótesis de ausencia de relación lineal entre las parejas de variables. Se puede aceptar la existencia de linealidad entre las variables ya que el valor de significación es nulo.

		NONPERC-PATHS	ALFACRIT	LENGTH-PERC
Correlación de Pearson	NONPERC-PATHS	1,000	-0,866	-0,778
	ALFACRIT	-0,866	1,000	0,855
	LENGTH-PERC	-0,778	0,855	1,000
Sig. (unilateral)	NONPERC-PATHS	.	0,000	0,000
	ALFACRIT	0,000	.	0,000
	LENGTH-PERC	0,000	0,000	.

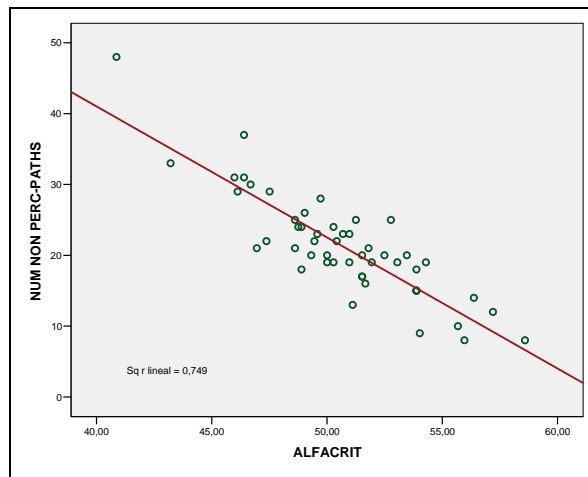
Tabla 6. Matriz de correlaciones entre las variables.

La regresión lineal en el caso isótropo se realiza con *nonperc_paths* (dependiente) y *alfacrit* (dependiente). Por consiguiente:

$$Y' = f(X) \Rightarrow Y - Y' = \text{Residuo} \quad (18)$$

El residuo obtenido debe ser el menor posible. Finalmente habrá que verificar que los residuos sigan una distribución normal, sean independientes y homocedásticos.

El diagrama de dispersión y la línea de ajuste representados en la **Figura 35** dan una idea gráfica sobre la linealidad negativa existente entre ambas.

**Figura 35.** Diagrama de dispersión entre alfacrit y non_percpaths.

La **Tabla 7** indica los valores del *coeficiente de determinación de la regresión* (R), que son más próximos a 1 cuanto menor es el residuo. Es el llamado *coeficiente de correlación lineal* entre la variable predicha por el modelo Y' y la variable observada Y .

Modelo	R	R cuadrado	R cuadrado corregida	Error típ. de la estimación
1	0,866	0,749	0,744	3,732

Tabla 7. Resumen del modelo.

El contraste *ANOVA* descrito en la **Tabla 8** tiene un objeto doble. Por un lado, descomponer la variabilidad total en un primer término que representa la variabilidad explicada por el modelo que en nuestro caso es un 75%, y en un segundo término o variabilidad no explicado por el modelo, que es la variabilidad residual. Por otro lado, verificar si la hipótesis de ausencia de relación lineal puede ser aceptada o rechazada.

Modelo		Suma de cuadrados	gl	Media cuadrática	F	Sig.
1	Regresión	1997,549	1	1997,549	143,401	,000
	Residual	668,631	48	13,930		
	Total	2666,180	49			

Tabla 8. ANOVA.

En la última columna de la tabla anterior aparece una significación nula, por tanto podemos aceptar el modelo lineal que aparece en la **Tabla 9**, donde se estiman los coeficientes. En este modelo se obtiene una variable (Y') que será la variable pronosticada y proporciona estimaciones medias de los valores de la variable *nonperc_paths* (variable Y) a partir de la recta siguiente:

$$Y' = 115,042 - 1,85 \cdot \text{alfacrit} \quad (19)$$

El modelo se resume en la **Tabla 9**. Las significaciones nulas permiten rechazar la hipótesis de que los coeficientes de la recta no son cero, y por ello se aceptan las estimaciones de estos coeficientes.

Modelo		Coeficientes no estandarizados		Coeficientes estandarizados	t	Sig.	Intervalo de confianza para B al 95%	
		B	Error típ.	Beta			Límite inferior	Límite superior
1	(Constante)	115,042	7,836		14,681	,000	99,287	130,797
	ALFACRIT	-1,850	,155	-,866	-11,97	,000	-2,161	-1,540

Tabla 9. Coeficientes.

Análisis de los residuos

Para finalizar veremos si se verifican las hipótesis básicas de la regresión:

- Normalidad de los residuos.
- Independencia en los errores cometidos en las observaciones.
- Varianza constante en los errores, u homocedasticidad.

En las **Figuras 36(a)** y **36(b)** puede apreciarse el histograma de los residuos con la curva normal superpuesta así como el diagrama p-p, en el que podemos ver como los datos se agrupan en torno a la recta.

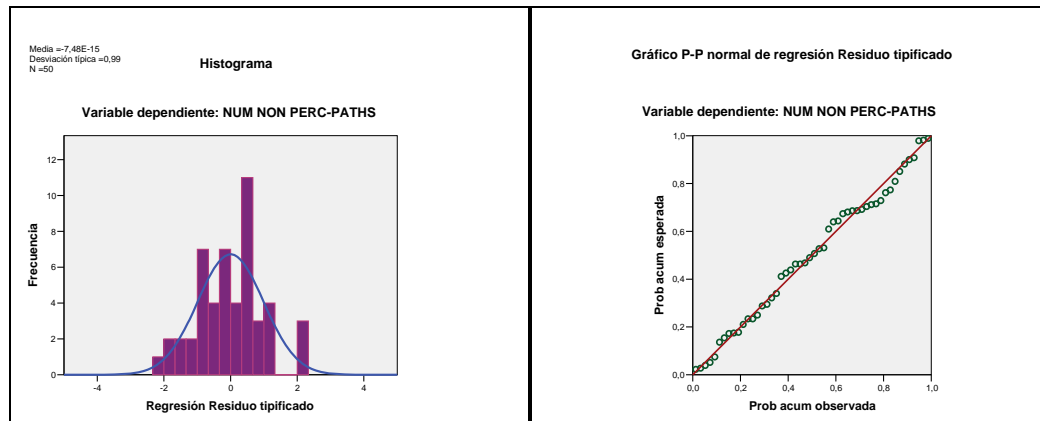


Figura 36. (a) Histograma de residuos y (b) Diagrama P-P de la variable dependiente.

A continuación, la **Tabla 10** contiene el contraste no paramétrico de Kolmogorov-Smirnov. En este caso la hipótesis que se desea contrastar es que los residuos también siguen una distribución normal. Como el nivel de significación obtenido es de 0,999, puede aceptarse que los residuos siguen una distribución normal.

		ALFACRIT	NONPERC-PATHS	PRED_1
Parámetros normales	Media	50,5954	21,42	21,4200000
	Desviación típica	3,45053	7,376	5,73782779
Diferencias más extremas	Absoluta	,083	,114	,053
	Positiva	,068	,114	,050
	Negativa	-,083	-,091	-,053
Z de Kolmogorov-Smirnov		,583	,804	,373
Sig. asintót. (bilateral)		,885	,537	,999

Tabla 10. Prueba de Kolmogorov-Smirnov para una muestra.

En la **Figura 37(a)** los datos se presentan en una pelota sin mostrar ningún patrón, a excepción de la tirada 19. Se acepta tanto la homocedasticidad como la independencia en los errores cometidos en las observaciones, pero parece conveniente repetir el estudio prescindiendo del caso 19 (ver *Anejo A4*). Para ver si realmente este caso influye en gran manera en el modelo observamos la **Figura 37(b)**, donde se representan los valores de influencia y se aprecia claramente que el caso 19 puede gobernar de manera muy importante el modelo.

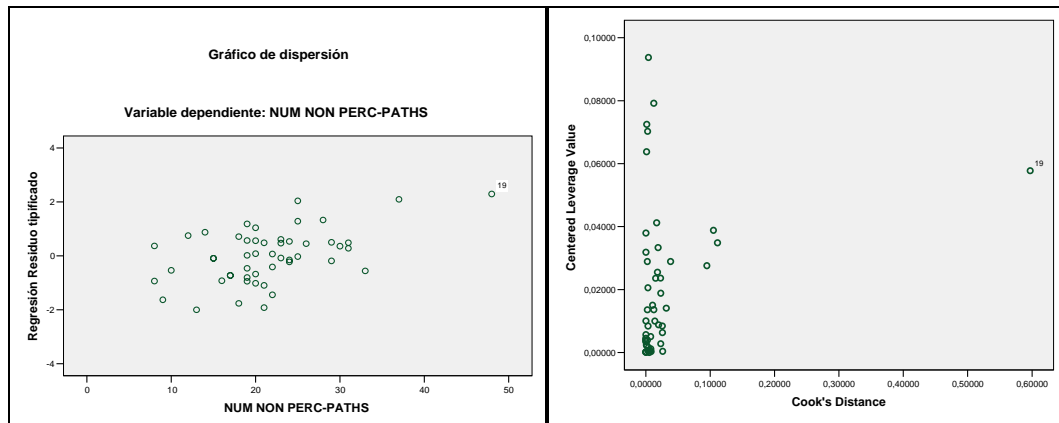


Figura 37. (a) Dispersión entre el residuo tipificado y nonperc_paths (b) dispersión entre la variable pronosticada y la distancia de Cook.

El gráfico de dispersión correspondiente a la **Figura 38** representa la variable dependiente frente a la variable dependiente pronosticada. Los puntos se agrupan en torno a la bisectriz aunque aparentemente parece haber una variabilidad mayor entorno al 20. Además, la tirada 19 parece separarse del resto.

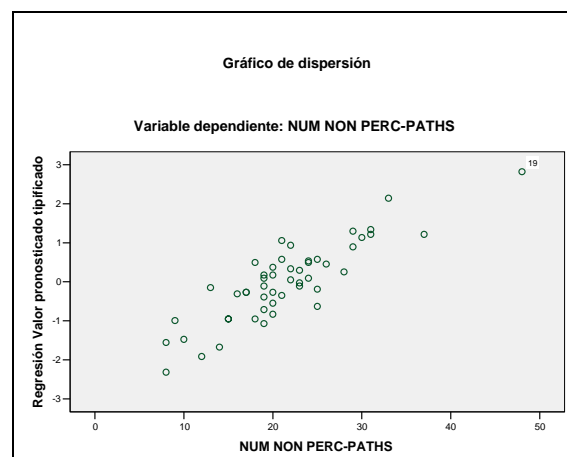


Figura 38. Diagrama de dispersión entre la variable real y la predicha.

Por tanto, se considera válida la hipótesis inicial dado que la linealidad entre la variable *nonperc_paths* y la variable *alfacrit* queda patente.

En un caso isótopo en el que se conozca el umbral de percolación " p_c " (*alfacrit*), se podrá predecir el número medio de caminos no percolantes.

5.3. INFLUENCIA DEL FACTOR DE FORMA Y DEL TAMAÑO

Las mallas tratadas hasta ahora han tenido la característica de ser de dimensión cuadrada ($n \times n$). Como se observa en el programa, puede asignarse a la dirección X un número de nodos distinto al de la dirección Y.

En el presente capítulo se aborda la influencia del factor de forma y del tamaño que le es adjudicado a la malla objeto de estudio. Los parámetros percolantes tratados a lo largo del capítulo pueden sufrir modificaciones en función de las dimensiones de la malla, obteniendo un valor muy dispar de *alfacrit* entre una red de dimensiones (10x10) y otra de dimensiones (100x5), por ejemplo. De igual modo, el tamaño también puede influir en los resultados.

5.3.1. EFECTO DEL FACTOR DE FORMA (FF)

Se define el *factor de forma* como la relación que existe entre el tamaño en la dirección X y el de la dirección Y:

$$\text{Factor de forma}(FF) = \frac{\text{dim } X}{\text{dim } Y} \quad (20)$$

Los objetivos serán analizar la influencia del factor forma sobre los parámetros percolantes. Estos parámetros son los ya conocidos *alfacrit* y *length_perc*, y un nuevo parámetro que aparece debido al factor de forma, el número de caminos percolantes *percpaths_num*. Se recuerda que en una red cuadrada, este valor era igual a 1. La aparición del factor de forma convierte la red en un objeto de dimensiones superiores en la dirección X respecto a la dirección Y (o viceversa) y por ese motivo aparecen a lo largo de la generación de aristas un mayor número de *percpaths_num*.

El número de caminos no percolantes no será objeto de estudio en este apartado, ya que su evolución es semejante al de una red cuadrada y el FF no influye en su valor.

Para analizar la influencia de FF sobre los parámetros elegidos, se escoge una dirección en la que el número de nodos va aumentando y la otra dirección queda constante.

Como ejemplo se representa en la **Figura 39** una red de dimensiones (60x10) en el momento de la percolación ($p_c = 0,33$, *alfacrit*=33%). Su factor de forma es igual a 10

unidades, habiendo elegido la dirección Y con un valor constante de 10 aristas y la dirección X con un mayor número, en este caso, seis veces mayor al de la dirección vertical.

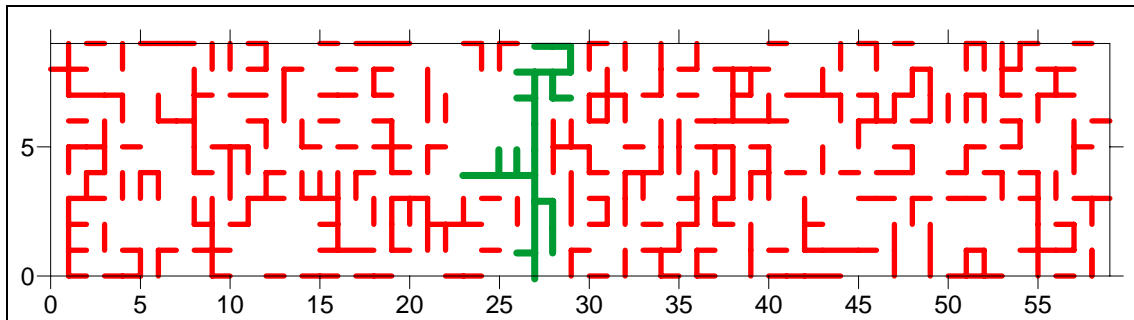


Figura 39. Red 60x10 en el momento de la percolación (Escala deformada).

Estudio de *alfacrit* (umbral de percolación)

En primer lugar se observa el descenso del valor de *alfacrit* cuando se le aplica un factor de forma elevado. En el caso anterior el valor de *alfacrit* es igual al 33% habiéndole asignado un $FF=6$. El resultado es lógico ya que, al haber alargado la red en sentido horizontal, la distancia en la dirección Y que debe salvar el camino percolante es 6 unidades menor a la distancia en la dirección X.

Las siguientes figuras muestran respectivamente la evolución del *alfacrit* en función del factor de forma. Se toma en la **Figura 40** un valor de aristas en sentido vertical igual a 5. Los valores en el sentido horizontal se toman desde el 5 ($FF=1$) hasta alcanzar el valor de 100 ($FF=20$), siendo la dimensión de la malla de 100x5.

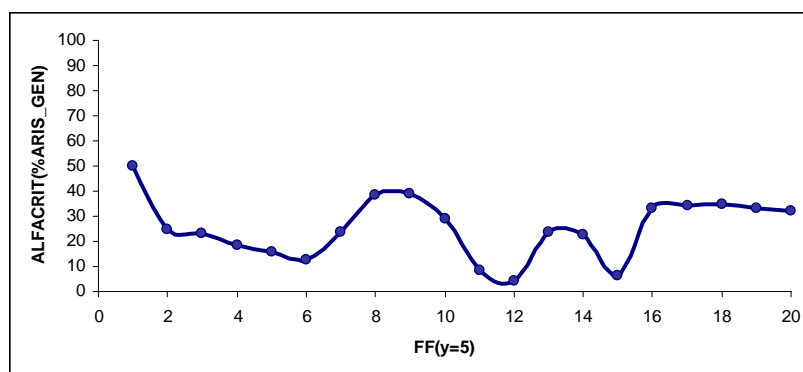


Figura 40. Evolución de alfacrit respecto al FF para Y= 5.

La **Figura 41** muestra la misma evolución, tomando un valor de aristas en la dirección horizontal igual a 10. De ese modo, los valores de los nodos en X empezarán siendo igual a

10 ($FF=1$) hasta llegar a un valor de 100 ($FF=10$), correspondiente a una red de dimensiones (100x10).

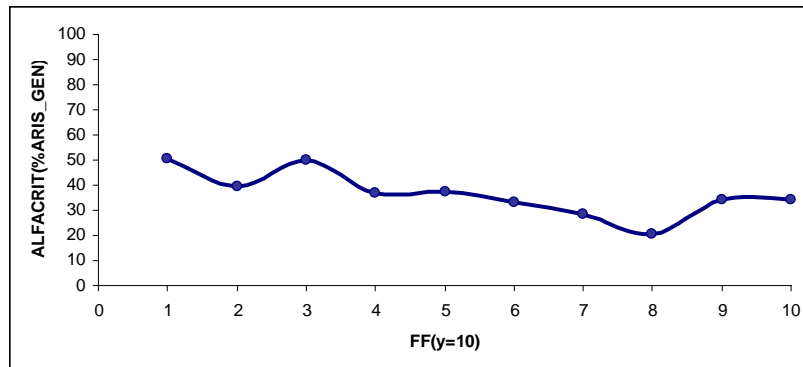


Figura 41. Evolución de *alfacrit* respecto al FF para Y= 10.

Se observa que los valores correspondientes a la red con 5 nodos en dirección Y, muestra unos resultados más dispersos e inferiores al de una red cuadrada, pero sin poderse considerar representativos. En cambio, cuando los nodos en la dirección Y son igual a 10, los valores muestran un descenso a medida que aumenta el factor de forma. Intuitivamente, el resultado obtenido es lógico, ya que al aumentar la dimensión horizontal *versus* la vertical, la probabilidad de hallar un camino percolante entre cualquier arista con $y=0$ y cualquier otra con $y=10$ (máximo) ha de crecer.

En consecuencia, el valor de *alfacrit* parece depender ligeramente del factor de forma (disminuye suavemente), por ello se considera que su valor permanece constante a medida que FF crece.

Estudio de *length perc* (longitud del camino percolante)

El siguiente parámetro percolante a analizar corresponde a la suma de todas las aristas que forman el camino percolante, teniendo en cuenta además los *backbones* (costillas) que forman la totalidad del camino. Cabe esperar un resultado más o menos constante.

Tomando 5 nodos en la dirección Y, se observa su evolución en función del FF (ver la **Figura 42**). Los valores están comprendidos entre 4 y 20 unidades de longitud o aristas que forman el camino percolante.

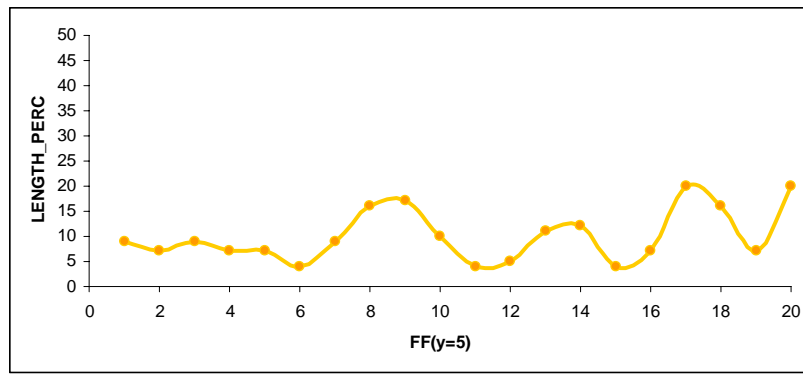


Figura 42. Evolución de *length_perc* respecto al FF para Y= 5.

La **Figura 43** representa la misma evolución con un valor de nodos en el sentido vertical igual a 10. Los resultados son mayores en cuanto a unidades, oscilando entre 15 y 69 unidades. Sin embargo, su comportamiento no resulta ser tan constante como el anterior.

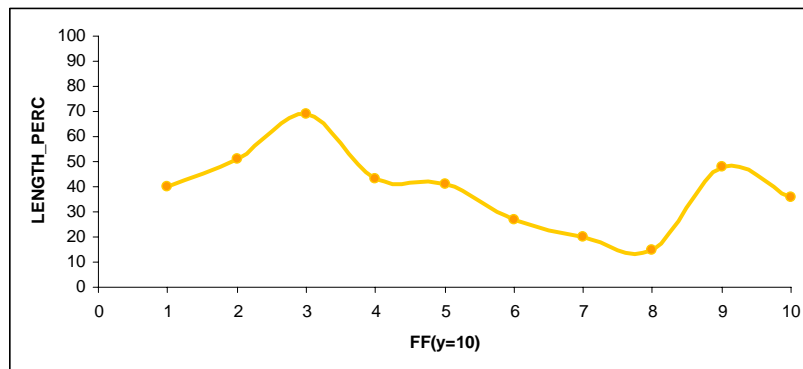


Figura 43. Evolución de *length_perc* respecto al FF para Y= 10.

En vista de los resultados obtenidos, parece que la longitud del camino percolante no depende fuertemente del factor de forma y por ello se considerará constante.

Estudio de *percpaths_num* (número de caminos percolantes)

Finalmente se estudia el valor del número de caminos percolantes que se forman a lo largo de toda la generación de aristas, hasta llegar al 100% de la malla activada. *A priori* se intuye que a medida que el factor de forma aumenta, siendo la diferencia de X respecto de Y mayor, los *percpaths_num* también aumentarán. La **Figura 44** muestra el estado de una red de dimensiones (50x5), con FF= 10 para valores con Y= 5, donde el número de caminos percolantes alcanza un valor máximo de 6 cuando se han generado un 57% de las aristas. En ese momento, los caminos no percolantes (7 en total) aún no han sido del todo absorbidos por el cluster de percolación.

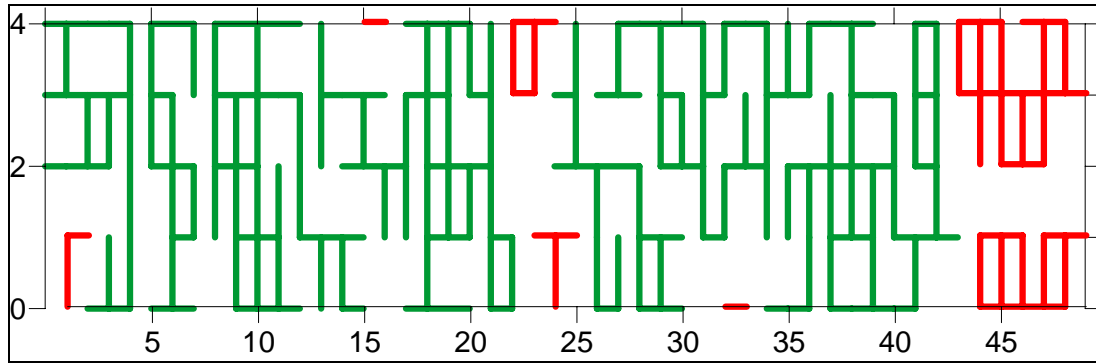


Figura 44. Red 50x5 con un 57% de aristas generadas y 6 *percpaths_num*.

Una vez realizado el estudio se comprueba que la intuición inicial respecto al aumento del número de caminos percolantes queda confirmada. A medida que aumenta la dimensión en el eje perpendicular a la dirección del flujo, existe un área mayor en la que pueden formarse caminos percolantes. El ancho implica que las aristas puedan formar caminos sin llegar a unirse con el resto de caminos, como se observaba en la **Figura 45**. En la **Figura 45** se indica la evolución del parámetro *percpaths_num* en función del factor de forma para $Y=5$.

En consecuencia, el valor de *percpath_num*, parece depender del factor de forma, aumentando su valor a medida que FF crece.

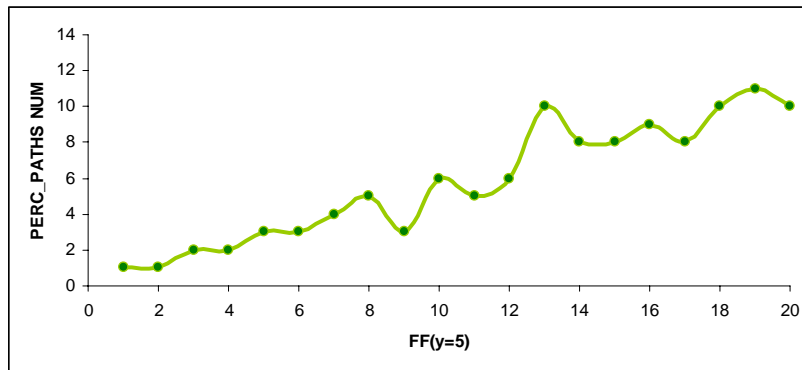


Figura 45. Evolución de *percpaths_num* respecto al FF para $Y=5$.

La **Figura 46** indica del mismo modo, e incluso de forma más clara, el aumento del parámetro cuando la coordenada vertical elegida es igual a 10.

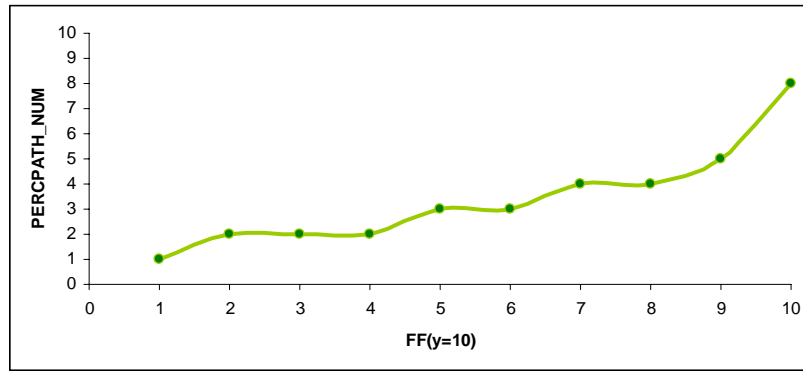


Figura 46. Percpaths_num respecto a FF para Y= 10.

En resumen: si $FF \uparrow$ entonces $\left\{ \begin{array}{l} \text{alfacrit} \approx \text{cte} \\ \text{length_perc} \approx \text{cte} \\ \text{non_percpaths} \uparrow \end{array} \right\}$

5.3.2. EFECTO DEL TAMAÑO

Se finaliza el análisis del ejemplo isótropo del fenómeno de percolación señalando la influencia que ejerce el tamaño de la red sobre el resto de parámetros.

Dada una serie de mallas cuadradas ($v \times v$), con un intervalo de v comprendido entre 5y 30, se evalúan a continuación las posibles diferencias en los resultados si se trata por ejemplo, con una red (5x5) formada por 32 aristas o si la red está formada por 1682 aristas (30x30).

La **Figura 47** nos muestra los valores del *alfacrit* a lo largo de todas las dimensiones tratadas. Se considera que su valor es estable ya que está en todo momento comprendido entre el 40 y el 60%.

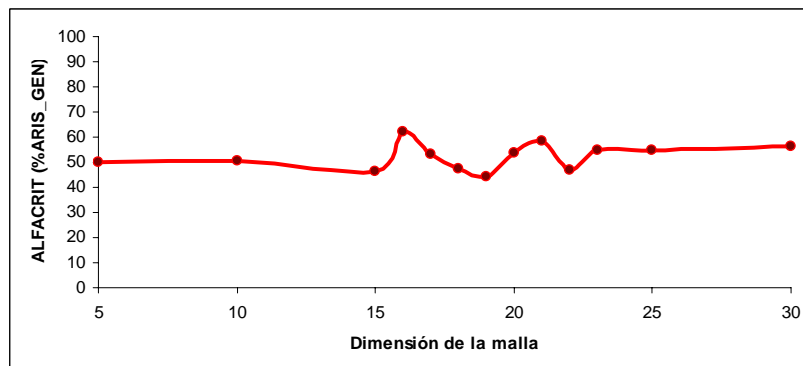


Figura 47. Alfacrit en función de la dimensión de la malla.

Con la **Figura 48** se verifica que el valor de *length_perc* aumenta a medida que las dimensiones de la malla aumentan.

En esta ocasión se verifica la relación ascendente intuita previamente. Parece comprensible que al aumentar el tamaño de la red, el parámetro *length_perc* también aumenta. La forma más rápida de verificarlo sería pensar que es más largo el camino de una malla (40x40) que el de una malla (5x5). Pero puede suceder que en mallas de dimensiones mayores, ocurra un camino percolante más corto por el mero hecho de haberse generado de forma aleatoria aristas que formen el camino más rápidamente.

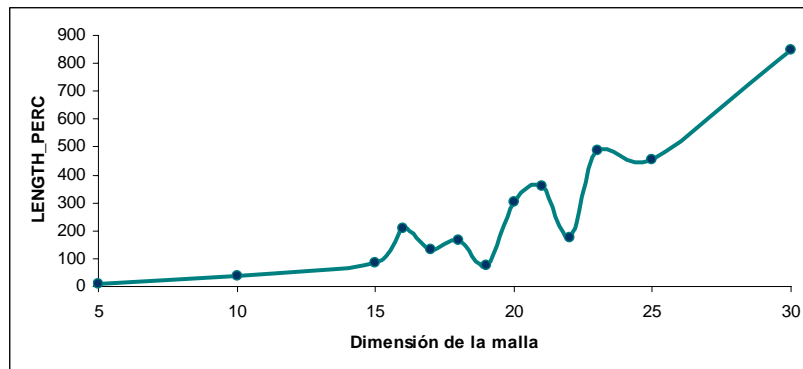


Figura 48. Length_perc en función de la dimensión de la malla.

Al realizar simulaciones del *BINGO_PERCOLATION*, se ha planteado la búsqueda de la relación entre la dimensión de la red y el tiempo de CPU, es decir, ¿a partir de qué dimensión de malla, el tiempo de ejecución del programa se dispara? La **Figura 49** indica que a partir de una red (30x30), el tiempo es superior a las 10 horas.

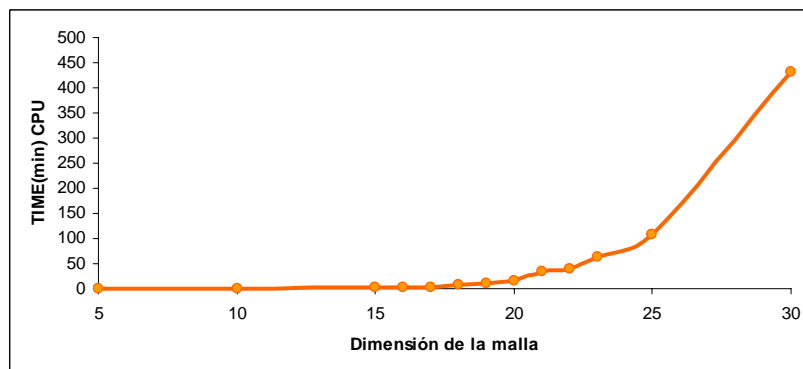


Figura 49. Tiempo CPU en función de la dimensión de la malla.

6. EJEMPLOS ANISÓTROPOS

6.1. ANÁLISIS DE UNA SIMULACIÓN

Los ejemplos tratados en este capítulo corresponden al caso anisótropo, en los que existe una preferencia con respecto a la probabilidad de activarse una arista u otra en función de su dirección. El flujo que atraviesa la malla se considera vertical.

Caso anisótropo con preferencia de flujo vertical: Un medio poroso en el que existe un material muy fracturado en la misma dirección del flujo favorece el movimiento, es decir, el peso en la probabilidad de activar una arista vertical es mayor que el de la probabilidad de activar una arista horizontal.

Caso anisótropo con preferencia de flujo horizontal: Un medio poroso en el que existe un material impermeable (arcilla, por ejemplo) formando lentejas horizontales en dirección perpendicular al flujo, dificulta el movimiento, es decir, el peso en la probabilidad de activar una arista horizontal es mayor que el de la probabilidad de activar una arista vertical.

La malla elegida para analizar una simulación ha sido de nuevo la de (20x20), considerando que sus dimensiones proporcionan un resultado satisfactorio.

Para realizar la simulación se supone un peso cinco veces mayor en la dirección preferente del flujo (dirección Y para el caso vertical y dirección X para el horizontal).

El objetivo de este apartado será hallar los parámetros percolantes para ambos casos y realizar posteriormente una comparación de resultados.

6.1.1. PARÁMETROS PERCOLANTES

Dadas dos simulaciones del programa *BINGO_PERCOLATION* para un modelo de percolación en medio poroso, se obtienen de nuevo los mismos parámetros que en el caso isotrópico.

- *Alfacrit*
- *Tortuosidad.*
- *Length_percpath*
- *Non_percpaths*

La **Tabla 11** indica los valores obtenidos de *alfacrit*:

Caso anisótropo vertical	Caso anisótropo horizontal
42,38%	57,89

Tabla 11. Valores de *alfacrit*.

Se comprueba que el valor del caso vertical es inferior al del caso isotrópico, ya que la dirección preferencial en cuanto a la generación de aristas coincide con la dirección del flujo y favorece la formación del camino percolante.

En cambio, el *alfacrit* del caso horizontal es superior a los demás casos, puesto que la probabilidad cinco veces superior de activar una arista horizontal, dificulta la formación del cluster percolante o camino percolante.

Con respecto a la **tortuosidad**, los valores obtenidos se muestran en las **Figuras 50 y 51**. Se comprueba que en el caso vertical, el resultado es más cercano a 1 (tortuosidad de 1,23) en el momento de la percolación. Si la anisotropía vertical favorece la generación del camino percolante, es de suponer que el valor obtenido sea menor al caso isotrópico, como se indica en la **Figura 50**.

Se recuerda que en una malla (20x20) la longitud mínima posible será 19.

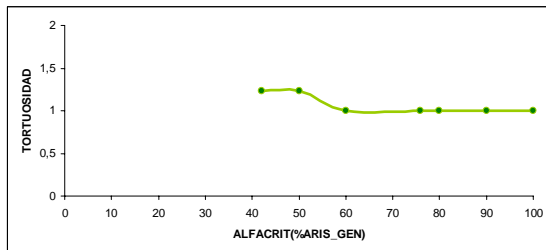


Figura 50. Evolución de la tortuosidad en el caso vertical.

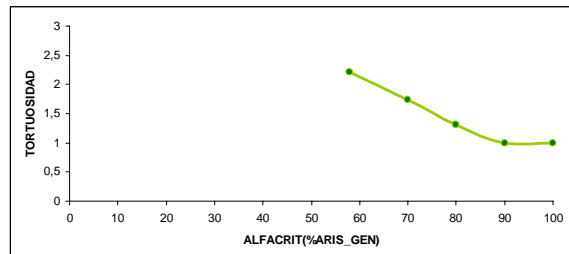


Figura 51. Evolución de la tortuosidad en el caso horizontal.

La **Figura 51** muestra que el valor de la tortuosidad es el más elevado de los casos expuestos (2,21), descendiendo de forma constante hasta alcanzar el 90% de la malla activada, donde existe un camino en línea recta que conecta los extremos de la malla.

La tortuosidad ofrece su valor más elevado en el caso anisótropo horizontal y en consecuencia, muestra un camino percolante más sinuoso.

La **Figura 52** muestra el camino percolante del caso anisótropo con preferencia horizontal. Se observa claramente la dificultad para seguir un camino en línea recta, ya que el peso de la dirección horizontal es cinco veces mayor.

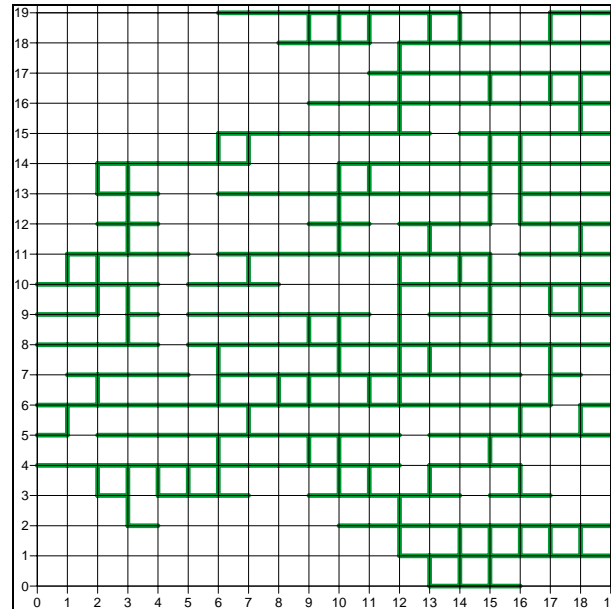


Figura 52. Representación del camino percolante en el caso horizontal.

El número de caminos no percolantes viene representado en las **Figuras 53 y 54**. En el caso anisótropo vertical se produce el máximo en el 25,6%, alcanzando 88 caminos no percolantes en la malla.

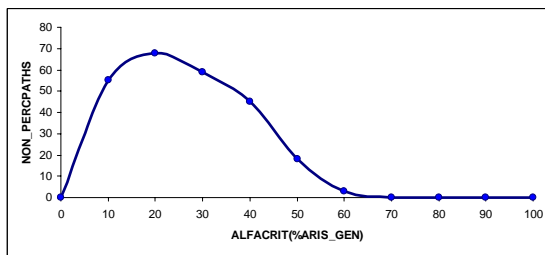


Figura 53. Evolución del n° caminos no percolantes en el caso vertical.

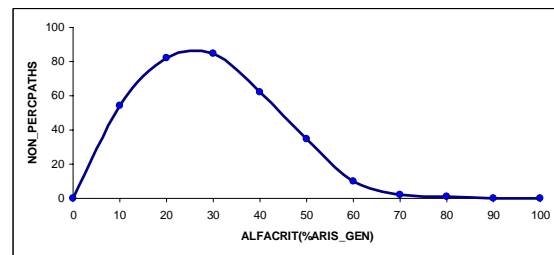


Figura 54. Evolución del n° caminos no percolantes en el caso horizontal.

En el caso de anisotropía vertical, alcanza un máximo de 72 caminos no percolantes en el 16,62% de las aristas generadas (ver **Figura 55**).

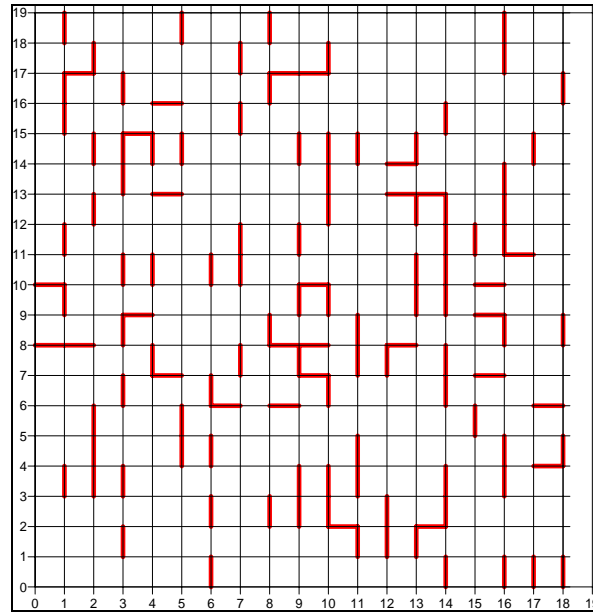


Figura 55. Representación del máximo número de caminos no percolantes (16,62% de aristas generadas).

La evolución del número de caminos no percolantes es similar al caso isótropo, el pico que se producía entre el 20 y el 30% de aristas generadas, vuelve a darse en los dos casos anisótropos. Existe pues un momento previo a la percolación, donde los caminos no percolantes alcanzan su valor máximo, para empezar luego a descender y llegar a desaparecer. En caso vertical son engullidos por el camino percolante en el 70%, a diferencia del caso horizontal, donde existen hasta el 90%.

Finalmente se analiza el camino percolante. Cabe señalar que en el caso anisótropo vertical se ha observado la presencia de dos caminos percolantes simultáneos. La razón es porque favoreciendo el flujo, el camino es menos sinuoso (tortuosidad cercana a 1) y se puede dar el caso de que dos caminos crucen la malla sin tener ninguna arista en común.

La **Figura 56** representa el momento en que coexisten dos caminos en la malla anisótropa vertical. Concretamente vendrán dados para unos valores de generación de aristas comprendidos entre el 52,6% y el 54,6%.

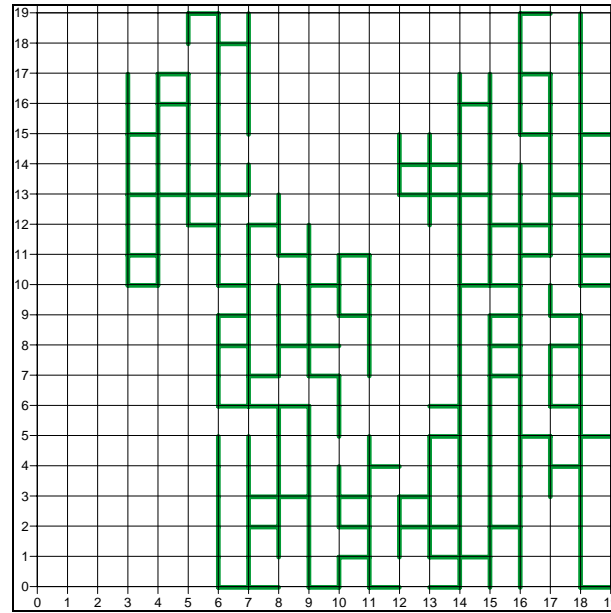


Figura 56. Coexistencia de dos caminos percolantes en el caso vertical.

Length_percpath es un parámetro percolante que aumenta de forma constante a medida que las aristas van generándose. En el caso isótropo se veía que a partir del 70% de aristas activadas, el aumento era constante. En el caso anisótropo se corrobora el resultado del capítulo anterior, tanto en el caso anisótropo vertical como en el horizontal. Se indica en la **Figura 57** la evolución de la longitud del camino percolante en el caso anisótropo horizontal.

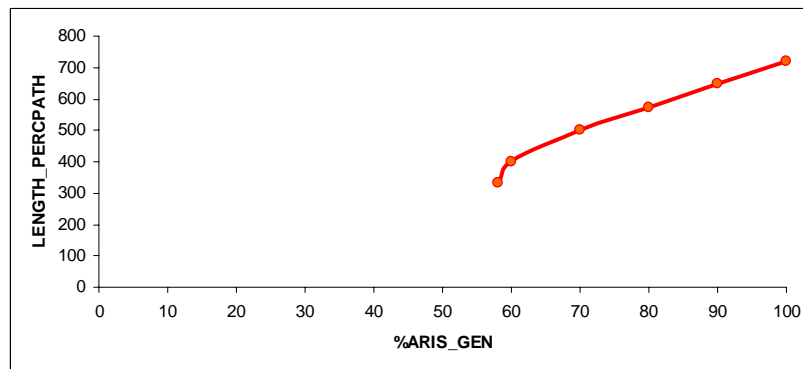


Figura 57. Evolución de *length_percpaths* en el caso anisótropo horizontal.

6.2. ANÁLISIS DE MONTECARLO

Las variables aleatorias objeto de estudio son de nuevo *alfacrit*, *nonperc_paths* y *length_perc*. Los valores resultantes de las cincuenta iteraciones se pueden consultar en el *Anejo A4*.

De nuevo se busca el tipo de relación (lineal, no-lineal) entre las tres variables.

En el caso vertical se observa únicamente que la variable *length* tiene unos valores que pueden considerarse normales, como indica la **Figura 58**.

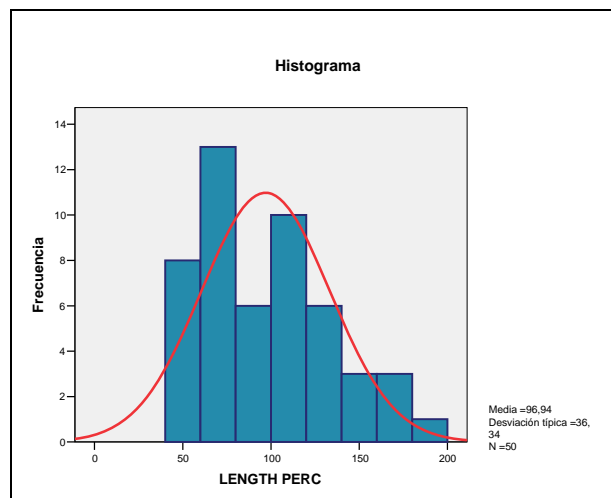


Figura 58. Histograma de *length* percolantes en el caso vertical.

En el caso horizontal, la única variable que presenta un valor constante en un rango de valores permitido, es el *alfacrit* (**Figura 59**). En este caso coincide con los resultados obtenidos en el caso anisótropo.

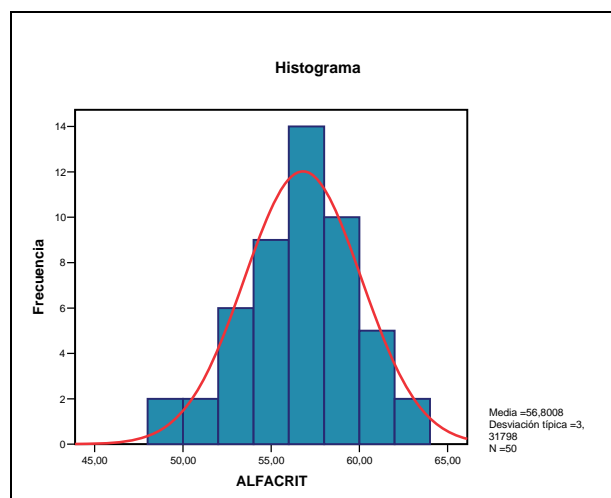


Figura 59. Histograma de *alfacrit* en el caso horizontal.

6.2.1. ESTUDIO ESTADÍSTICO

El presente apartado se divide en el estudio de los dos ejemplos anisótropos y se muestran las combinaciones de variables dependientes e independientes que mejor relación han dado. El resto de combinaciones puede ser consultado en el *Anejo A5*.

CASO ANISÓTROPO CON PREFERENCIA VERTICAL

En este caso anisótropo vertical se obtienen de nuevo los estadísticos descriptivos contenidos en la **Tabla 12**.

	Mínimo	Máximo	Media	Desv.Típica
<i>Alfacrit</i>	36,57	48,75	43,2926	3,38806
<i>Length_perc</i>	41	181	96,94	36,340
<i>Nonperc_paths</i>	18	58	37,10	10,792

Tabla 12. Estadísticos descriptivos.

La correlación entre las variables queda plasmada en la **Tabla 13** y puede resumirse en los siguientes puntos:

- Las variables *length* y *alfacrit* toman un valor de correlación igual a 0,525 formando un ángulo de 58,33° entre sus ejes. Se demuestra la escasa dependencia entre ellas.
- La variable *non_percpaths* depende en mayor grado de la variable *alfacrit* (relación de 0,909) que de la variable *length* (relación de 0,604).

En conclusión, tienen sentido hallar un modelo de regresión lineal con las tres variables, considerando como variable dependiente *non_percpaths* en función de las dos restantes.

	NONPERC-PATHS	ALFACRIT	LENGTH PERC
NON PERCPATHS	1,000	-0,909	-0,604
ALFACRIT	-0,909	1,000	0,525
LENGTH PERC	-0,604	0,525	1,000

Tabla 13. Correlaciones.

El modelo de regresión lineal indicado en la **Tabla 14**, muestra un valor de $R = 0,921$ que verifica la dependencia de la variable observada *non_percpaths* en función de la variable pronosticada por el modelo.

Modelo	R	R cuadrado	R cuadrado corregida	Error típ. de la estimación
1	0,921	0,849	0,842	4,289

Tabla 14. Resumen del modelo.

En el contraste ANOVA se obtiene un valor de la suma de cuadrados debidos a la regresión de un 85%. Con el valor de significación nulo, puede aceptarse el modelo lineal.

Los coeficientes del modelo están contenidos en la **Tabla 15**, a partir de los cuales se obtiene el siguiente plano de regresión:

$$Y'' = 154,793 - 2,602 \cdot \text{alfacrit} - 0,052 \cdot \text{length} \quad (21)$$

Modelo		Coeficientes no estandarizados		Coeficientes estandarizados	t	Sig.	Intervalo de confianza para B al 95%	
		B	Error típ.	Beta			Límite inferior	Límite superior
1	(Constante)	154,793	8,375		18,482	0,000	137,944	171,641
	ALFACRIT	-2,602	0,213	-0,817	-12,243	0,000	-3,030	-2,174
	LENGTH PERC	-0,052	0,020	-0,175	-2,627	0,012	-0,092	-0,012

Tabla 15. Coeficientes.

Como se realiza en el caso isótropo, las **Figuras 60(a)** y **60(b)** nos muestran la normalidad en el análisis de residuos a través del histograma y el diagrama p-p, donde los valores se sitúan en la bisectriz.

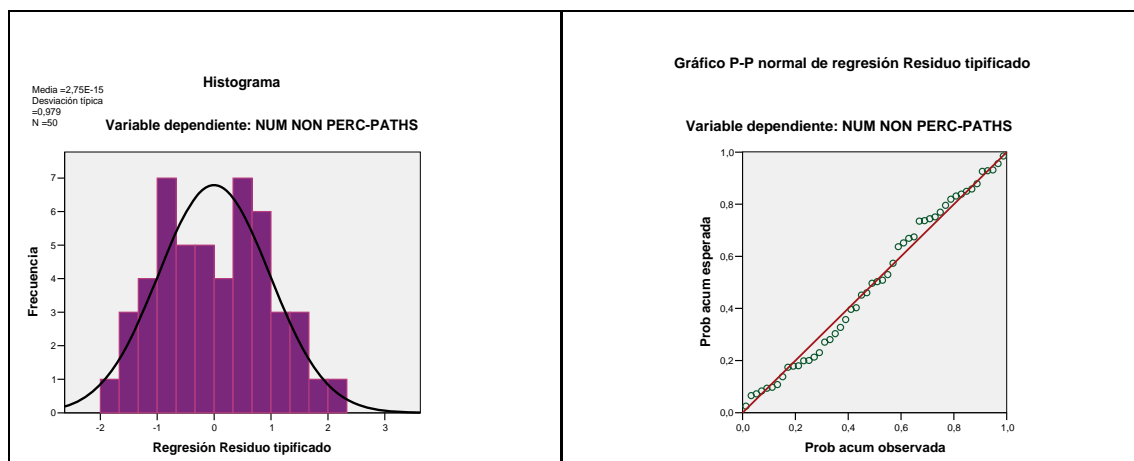


Figura 60. (a) Histograma de residuos y (b) Diagrama P-P de la variable dependiente.

Finalmente, en la **Figura 61** se representa un gráfico de dispersión tridimensional en el que se puede observar cómo los 50 valores se agrupan en torno del plano dibujado en función de *alfacrit* y *length*.

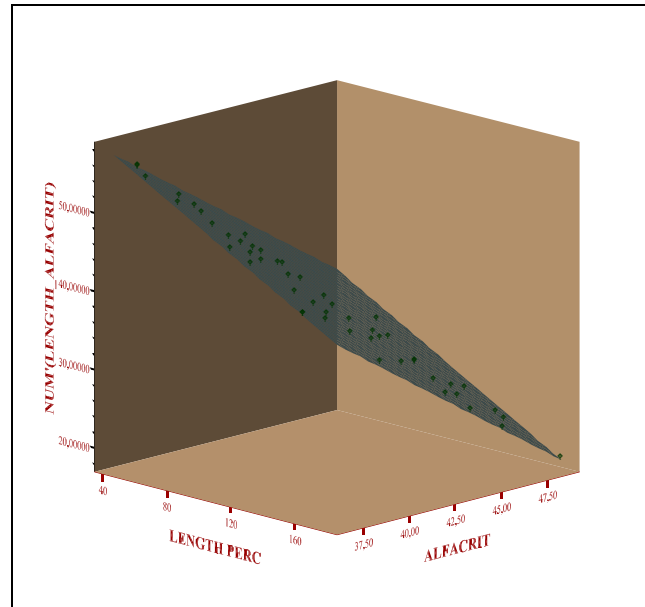


Figura 61. Plano correspondiente a la regresión lineal de las tres variables.

En conclusión, en el caso anisótropo vertical pueden predecirse los valores medios de los caminos no percolantes si se dispone de la longitud del camino percolante (*length*) y del umbral de percolación (*alfacrit*).

CASO ANISÓTROPO CON PREFERENCIA HORIZONTAL

En este último caso, el estudio estadístico de las ya conocidas variables muestra una fuerte correlación entre las independientes *alfacrit* y *length*. En consecuencia, se procede a crear una nueva variable *length* predicha, que será función de la variable independiente *alfacrit*. A partir de esta nueva variable, se podrá tener el valor predicho de *non_perppaths* en función de *alfacrit* y de la nueva *length* pronosticada, acumulando toda la información en el proceso y sus pesos.

La **Tabla 16** muestra los estadísticos descriptivos de la variable *alfacrit*.

Media	Mediana	Moda	Desv. Típ.	Varianza	Mínimo	Máximo
56,8008	56,9950	53,88	3,317	11,009	49,45	63,85

Tabla 16. Estadísticos descriptivos.

La variable *alfacrit* presenta una distribución normal como indica la **Figura 62** en la distribución de frecuencias acumuladas del valor de la variable.

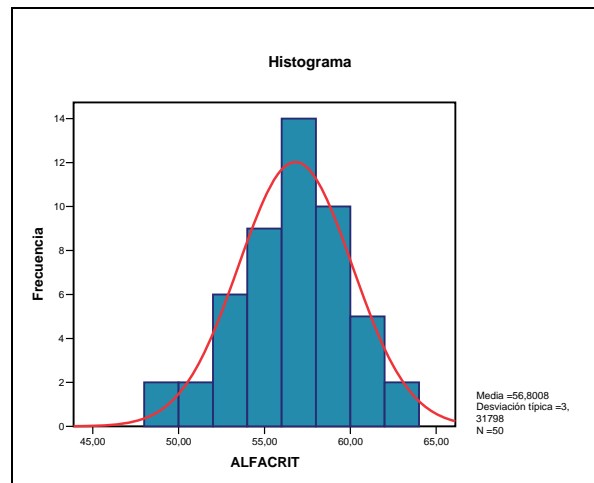


Figura 62. Distribución de frecuencia acumulada de *alfacrit*.

La diferencia con el caso anterior será la obtención de una recta como modelo y no un plano, ya que la información de la variable *alfacrit* estará implícita en la variable creada (*length*). Por tanto, la variable dependiente *non_percpaths* podrá ser pronosticada en función de la nueva *length*.

Los estadísticos descriptivos y las correlaciones existentes entre las tres variables estudiadas se resumen en las **Tablas 17.** y **18.**

	Mínimo	Máximo	Media	Desv. típ.
ALFACRIT	49,45	63,85	56,8008	3,31798
NON PERC-PATHS	2	30	13,18	6,134
LENGTH PERC	231	450	359,70	52,130

Tabla 17. Estadísticos descriptivos.

La matriz de correlaciones entre variables queda descrita en la **Tabla 18**, donde se observa una fuerte dependencia de la variable *length* con respecto a la variable *alfacrit* (0,907).

	NON PERC-PATHS	ALFACRIT	LENGTH PERC
NON PERC-PATHS	1,000	-0,861	-0,848
ALFACRIT	-0,861	1,000	0,907
LENGTH PERC	-0,848	,907	1,000

Tabla 18. Correlaciones.

Como resultado se obtiene de la siguiente recta:

$$length' = -449,927 + 14,254 \cdot alfacrit \quad (22)$$

Entonces, se realiza la regresión lineal tomando como variable dependiente *non_perpaths* en función de la variable creada *length'*. La **Tabla 19** muestra el resumen del modelo:

Modelo	R	R cuadrado	R cuadrado corregida	Error típ. de la estimación
1	0,861(a)	0,741	0,736	3,154

Tabla 19. Resumen del modelo.

La recta resultante de la regresión lineal es la siguiente:

$$Y''' = 53,337 - 0,112 \cdot length' \quad (23)$$

Las **Figuras 63 y 64** muestran el análisis de residuos y corroboran, como en los anteriores casos, que puede aceptarse normalidad en los residuos.

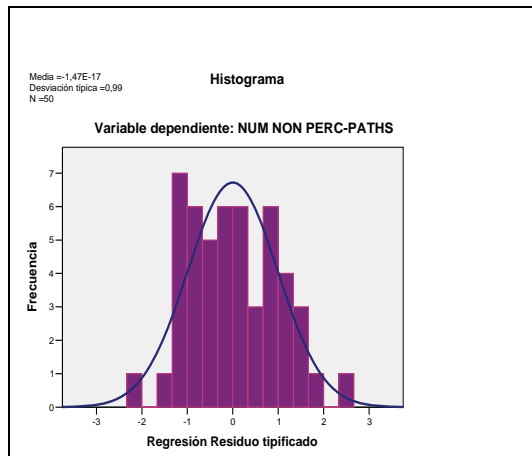


Figura 63. Histograma de los residuos.

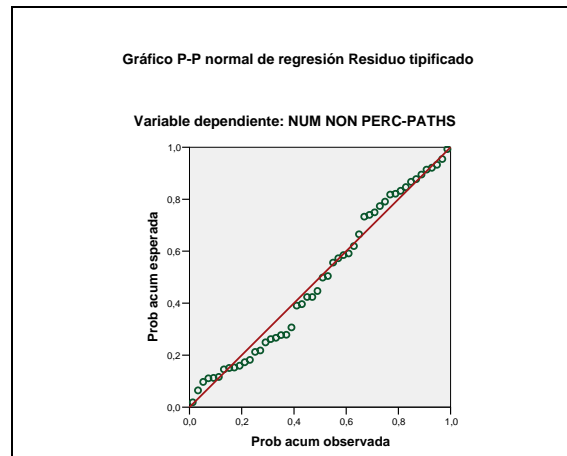


Figura 64. Diagrama p-p.

Finalmente, la **Figuras 65** representan la evolución del residuo tipificado y se acepta tanto la homocedasticidad como la independencia en los errores cometidos en las observaciones. La **Figura 66** que representa el valor pronosticado por el modelo frente a la variable observada muestra que el modelo lineal es aceptable.

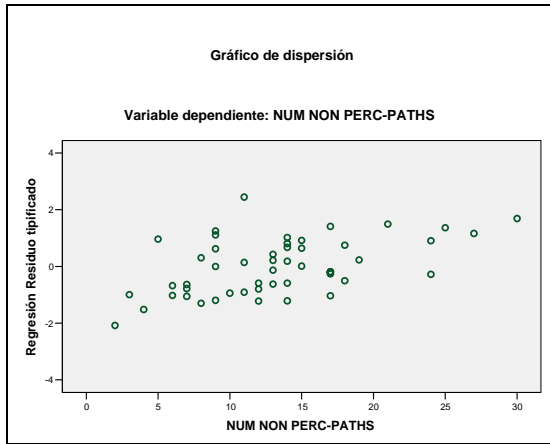


Figura 65. Gráfico de dispersión del residuo tipificado *vs non_perpaths*.

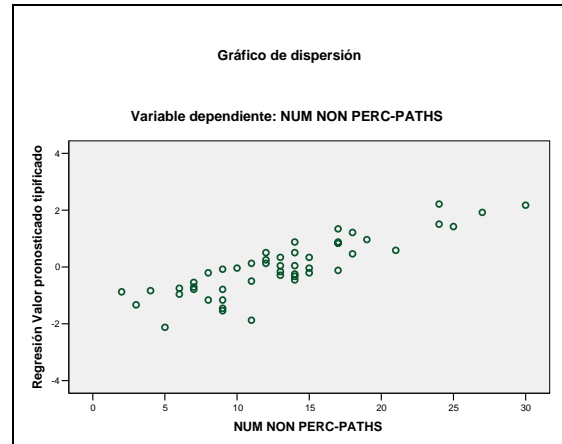


Figura 66. Gráfico de dispersión del valor pronosticado *vs non_perpaths*.

Se puede obtener el valor medio del número de caminos no percolantes a partir del conocimiento de la longitud del camino percolante en función del umbral de percolación.

6.3. INFLUENCIA DEL FACTOR DE FORMA Y DEL TAMAÑO

En el presente capítulo se realiza el mismo análisis que en el capítulo 5.3 para los ejemplos anisótropos con el fin de hallar una posible influencia del FF y del tamaño, así como la comparación de resultados respecto al caso isótropo.

6.3.1. INFLUENCIA DEL FACTOR DE FORMA EN EL CASO VERTICAL

Se toman los mismos valores del factor de forma y se indica la evolución de los parámetros *alfacrit*, *length_perc* y *percpaths_num* en función del factor de forma.

Estudio de *alfacrit*

La **Figura 67** indica la evolución del umbral de percolación en función de un factor de forma en el que el número de nodos en la dirección Y es constante e igual a 5.

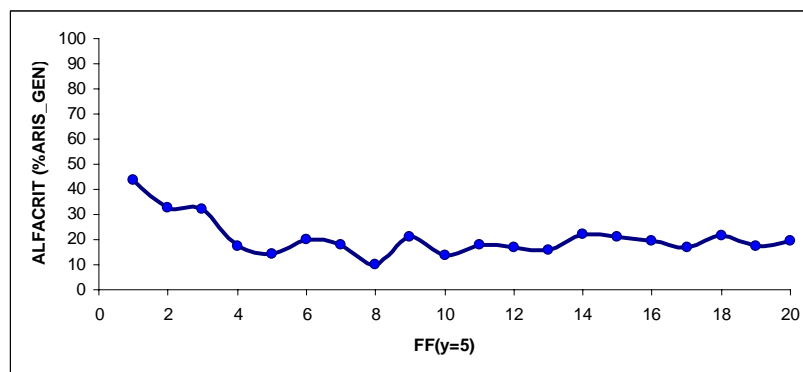


Figura 67. Evolución de *alfacrit* (caso vertical) respecto al FF para Y= 5.

Los valores se encuentran comprendidos entre el 20% y el 40%, siendo inferiores al caso isótropo.

La **Figura 68** indica del mismo modo la evolución del umbral de percolación para el caso anisótropo horizontal, observando que se mantiene más o menos constante.

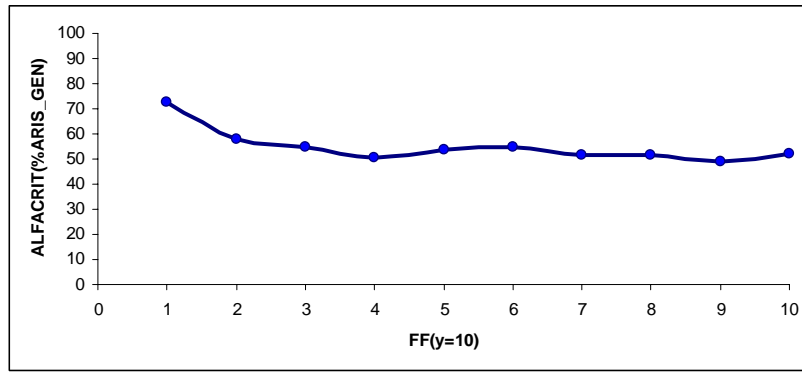


Figura 68. Evolución de *alfacrit* (caso horizontal) respecto al FF para Y= 5.

Por tanto el *alfacrit* tanto vertical como horizontal no se ven afectados por el factor de forma, por lo que podemos suponer que permanecen constantes.

Estudio de *length_percpath*

La **Figura 69** presenta la evolución de la longitud del camino percolante del caso anisótropo vertical en función de un FF con Y=10, con valores comprendidos entre 10 y 21. Para el caso Y=5 sus valores están acotados entre 4 y 12, muy similar al caso isótropo, donde oscilaba entre 4 y 20 unidades de longitud.

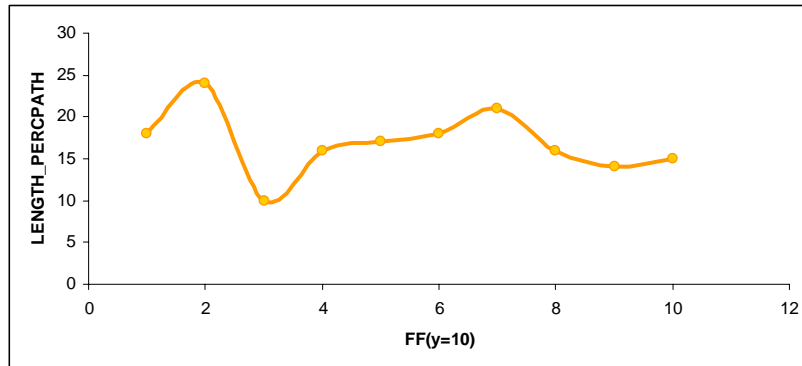


Figura 69. Evolución de *length_percpath* (caso vertical) respecto al FF para Y= 10.

Del mismo modo, se observa en la **Figura 70** que los valores de la longitud del camino percolante para el caso anisótropo horizontal, con respecto al factor de forma guardan una escasa relación. Los valores se consideran independientes del FF, ya que muestran la tendencia de retornar al valor original a pesar de que oscilan entre 118 y 406 unidades de longitud para factores de forma comprendidos entre 4 y 9.

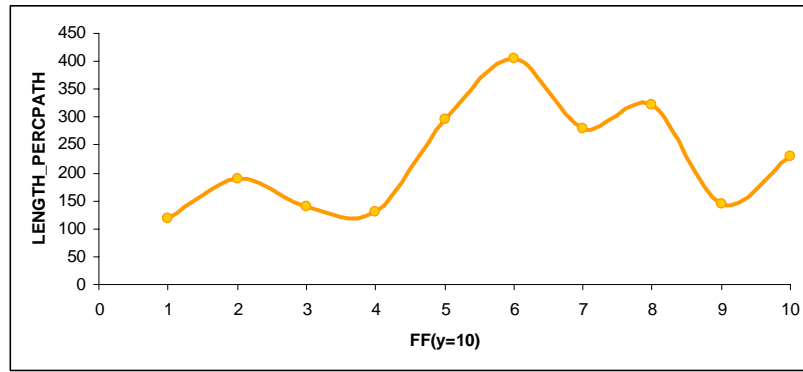


Figura 70. Evolución de *length_percpath* (caso horizontal) respecto al FF para Y= 10.

Estudio de *percpath_num*

El número de caminos percolantes aumenta claramente como en el caso isótropo. La diferencia será que para una malla de igual factor de forma, se obtienen el doble de caminos percolantes a partir del factor de forma igual a 2, es decir, una malla (10x5).

La **Figura 71** indica la evolución ascendente de los valores de caminos percolantes a medida que aumenta el factor de forma.

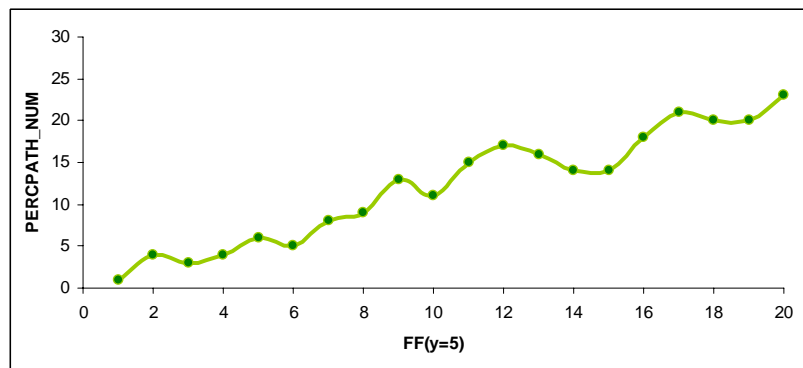


Figura 71. Evolución de *percpath_num* (caso vertical) respecto al FF para Y= 5.

El caso anisótropo horizontal ofrece unos valores totalmente distintos al caso vertical y al isótropo, manteniéndose constante e igual a 1, exceptuando el caso de la malla (90x10), donde existen dos caminos percolantes. Se observa en la **Figura 72** que el FF no influye en el número de caminos no percolantes, es decir, se obtiene el mismo resultado que en el caso de las mallas cuadradas, donde el *percpath* era único.

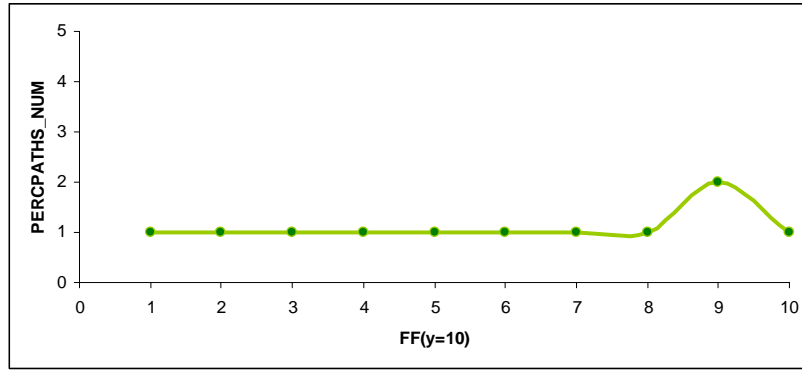


Figura 72. Evolución de *length_percpath* (caso horizontal) respecto al FF para Y= 10.

En resumen si $FF \uparrow$ entonces $\left\{ \begin{array}{l} \text{alfacrit}(\text{vert}) \approx \text{cte}; \text{alfacrit}(\text{horiz}) \approx \text{cte} \\ \text{length_perc}(\text{vert}) \approx \text{cte}; \text{length_perc}(\text{horiz}) \approx \text{cte} \\ \text{non_percpaths}(\text{vert}) \uparrow; \text{non_percpaths}(\text{vert}) \approx \text{cte} \end{array} \right\}$

6.3.2. INFLUENCIA DEL TAMAÑO EN EL CASO VERTICAL

La evolución de *alfacrit* en función de la dimensión de la malla se representa en la **Figura 73**. Los valores se mantienen constantes al igual que en el caso isótropo.

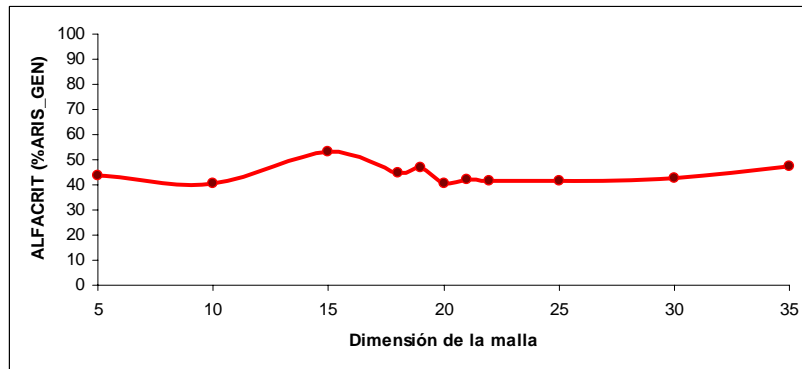


Figura 73. Evolución de *alfacrit* (caso vertical) respecto al tamaño.

Para el caso horizontal se obtiene el mismo resultado, valores constantes. La diferencia será que estos valores son superiores a los del caso anisótropo vertical e isótropo.

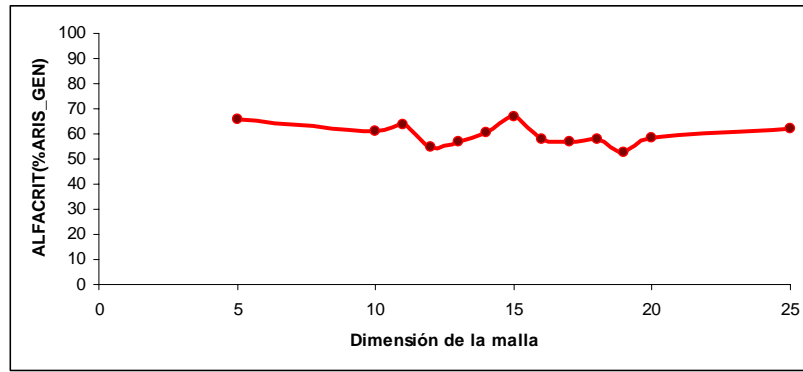


Figura 74. Evolución de *alfacrit* (caso horizontal) respecto al tamaño.

La longitud del camino percolante, como se había intuido en el ejemplo isótropo, aumenta a medida que crece la dimensión de la malla.

Las **Figuras 75 y 76** muestran la tendencia ascendente para los casos de anisotropía vertical y horizontal respectivamente.

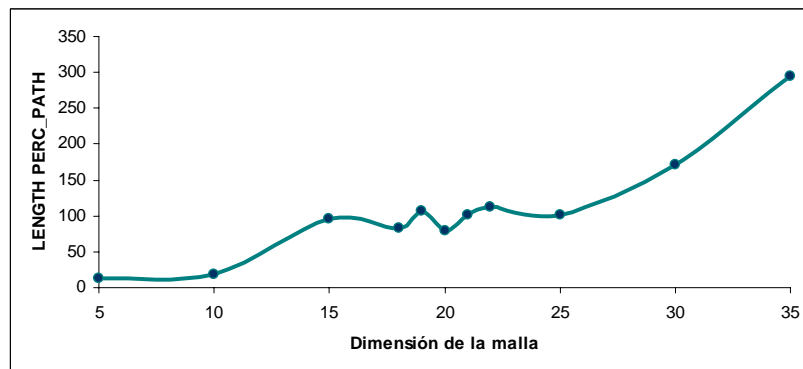


Figura 75. Evolución de *length_perc* (caso vertical) respecto al tamaño.

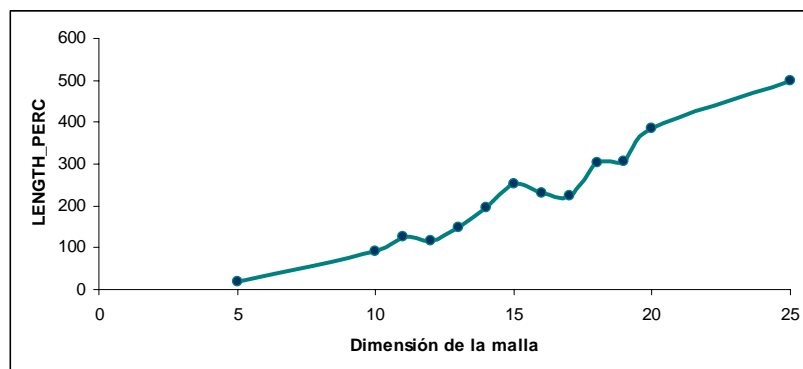


Figura 76. Evolución de *length_perc* (caso horizontal) respecto al tamaño.

Finalmente destacar que el tiempo de CPU para la realización de cada una de las simulaciones, sigue la misma evolución del ejemplo isótropo. La **Figura 77** indica para el ejemplo vertical, que a partir de una malla (30x30) el tiempo es superior a las 10 horas, como ocurría en el ejemplo isótropo.

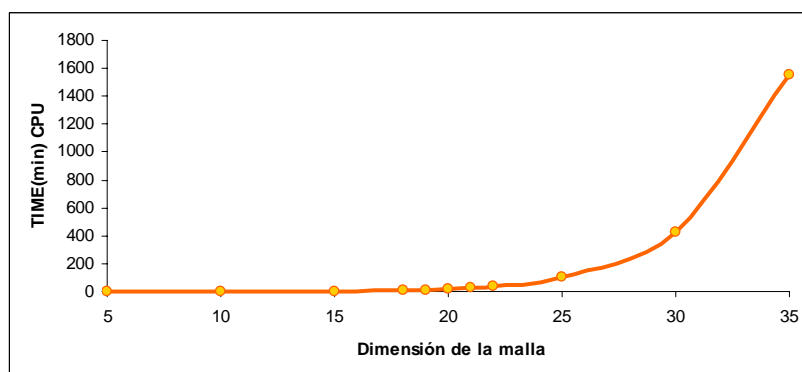


Figura 77. Tiempo CPU (caso vertical) en función de la dimensión de la malla.

6.4. COMPARACIÓN DE LOS EJEMPLOS ISÓTROPO Y ANISÓTROPOS

Dada una simulación particular se observa las siguientes relaciones entre los parámetros percolantes en medio poroso obtenidos:

- [*Alfacrit*: anisótropo vertical < isótropo < anisótropo horizontal]
- [*Tortuosidad*: anisótropo vertical < isótropo < anisótropo horizontal]
- [*Non_percpaths*: isótropo \approx anisótropo vertical \approx anisótropo horizontal]
- [*Length_percpaths*: isótropo \approx anisótropo vertical \approx anisótropo horizontal]

El factor de forma no influye en el valor de *alfacrit* ni en el valor de *length_percpaths*, tanto en el caso isótropo como en los casos anisotropos.

En cambio, el FF sí influye en los valores de *non_percpaths*, ya que se observa cómo en el caso anisótropo horizontal, el valor es constante e igual a 1, análogamente a lo que sucede en una malla cuadrada.

7. CONCLUSIONES

Una vez conocidos algunos aspectos esenciales de las teorías de percolación y de grafos se ha procedido a emplear elementos básicos de los grafos planos para realizar el estudio de una percolación de aristas (*Bond percolation*) en una malla bidimensional.

El programa *BINGO_PERCOLATION* ha sido la herramienta desarrollada para hallar los valores de los siguientes parámetros percolantes en medio poroso: *alfacrit* (umbral de percolación), *length_perc* (longitud del camino percolante) y *non_percpaths* (número de caminos no percolantes). El programa permite simular una percolación de aristas independientemente de los nodos que la conectan.

En el ejemplo isótropo para una malla plana cuadrada, los valores del umbral de percolación oscilan entorno del valor teórico, igual a 0.5, en una banda bastante estrecha. En el caso anisótropo vertical es más fácil generar aristas en la dirección Y, lo que favorece la formación del camino percolante y reduce los valores de *alfacrit*. Estos valores se mantienen en un entorno por debajo del valor teórico en una banda también estrecha. El caso anisótropo horizontal presenta unos valores por encima del isótropo (pero no mucho más grandes), ya que dificulta el flujo vertical y cuesta más crear el camino percolante.

El análisis de una simulación particular ha dado lugar a las siguientes relaciones:

- [*Alfacrit*: anisótropo vertical < isótropo < anisótropo horizontal].
- [*Tortuosidad*: anisótropo vertical < isótropo < anisótropo horizontal].
- [*Non_percpaths*: isótropo \approx anisótropo vertical \approx anisótropo horizontal].
- [*Length_percpaths*: isótropo \approx anisótropo vertical \approx anisótropo horizontal].

El análisis de una simulación de MonteCarlo ha confirmado la estabilidad de los valores del *alfacrit*. En cambio, los parámetros *length_perc* y *non_percpaths* presentan una gran variabilidad. A partir de estas observaciones se ha procedido a realizar un estudio estadístico de los valores de *length_perc*, *non_percpaths* y *alfacrit*.

El estudio estadístico ha proporcionado modelos lineales que permiten predecir el valor medio de la longitud del camino percolante (*length_perc*), así como el número de caminos no percolantes (*non_percpaths*), en función del umbral de percolación (*alfacrit*). Por tanto, parámetros importantes como la tortuosidad en un medio poroso, pueden obtenerse a partir de un *alfacrit* dado. A continuación, se resumen las relaciones lineales que dan un valor medio de caminos no percolantes:

- En un caso isótropo, se obtienen en función del umbral de percolación.
- En un caso anisótropo vertical, se obtienen en función de la longitud del camino percolante y del umbral de percolación.
- En un caso anisótropo horizontal, se obtienen en función de la longitud del camino percolante, que a su vez es función del umbral de percolación.

El factor de forma no influye en los valores de *alfacrit* ni en los de *length_perc*, tanto en el caso isótropo como en los casos anisótropos. En cambio, el FF sí influye en los valores de *non_perpaths*, ya que en el caso anisótropo horizontal, los valores son constantes y con un valor igual a 1 en su mayoría, análogamente a lo que sucede en una malla cuadrada.

8. FUTURAS LÍNEAS DE TRABAJO

Un camino en un grafo es una sucesión de vértices distintos y aristas conectando esos vértices de manera que un vértice inicial queda unido con otro final. Actualmente el programa *BINGO_PERCOLATION* proporciona un camino percolante que en realidad es un cluster. La primera futura línea de trabajo plantea la necesidad de mejorar el programa, con el fin de obtener de forma descriptiva el camino contenido en el cluster percolante que presente la menor tortuosidad de todos los caminos posibles dentro del cluster.

Es intuitivo esperar que si una arista ha sido activada, la probabilidad de que las aristas vecinas sean activadas en las fases inmediatas del proceso aumente. Por ello surge el problema de definir de forma analítica una probabilidad de generar aristas que recoja la información sobre las aristas ya activadas.

Otra línea de trabajo sería estudiar los parámetros percolantes en medio poroso no sólo en redes rectangulares sino en cualquier grafo plano. Podría comenzarse por estudiar mallas sencillas como son las triangulares o cualquier polígono convexo que admita una triangulación.

9. REFERENCIAS

- [1] Disponible en: <http://www.explora.cl/otros/agua/img/ciclo.jpg> [Consulta: Junio 2009].
- [2] PAJOT, S. *Percolation et économie*. Directores: R. Tartalin, D. Bouget. Tesis doctoral de la Universidad de Nantes, 10 octubre, 2001.
- [3] HUNT, A. *Percolation theory for flow in porous media*. Springer: Heidelberg, Alemania, 2005.
- [4] STAUFFER, D.; AHARONY, A. *Introduction to Percolation Theory*. Londres: Taylor Francis, 1985.
- [5] GRIMMETT, G. *Percolation*. Berlín: Springer, 1989.
- [6] Disponible en: <http://www1.coe.neu.edu/~emelas/NetForestFires.htm> [Consulta: Febrero 2009].
- [7] BEFFARA, V.; SIDORAVICIUS V. *Percolation Theory* (arXiv:math.PR/0507220v1 11 Jul 2005) [Consulta: Julio 2009].
- [8] GLASS, R. J.; CONRAD, S. H.; YARRINGTON, L. «Gravity-destabilized nonwetting phase invasion in macroheterogeneous porous media: Near-pore-scale macro modified invasion percolation simulation of experiments». *Water resources research* (2001), vol. 37, nº 5, 1197-1207.
- [9] GLASS, R. J.; CONRAD, S. H.; PEPLINSKI, W. «Gravity-destabilized nonwetting phase invasion in macroheterogeneous porous media: Experimental observations of invasion dynamics and scale analysis». *Water resources research* (2000), vol. 36, nº 36, 3121-3137.
- [10] GLASS, R. J.; NICHOLL, M. J.; YARRINGTON, L. «A modified invasion percolation model for low-capillary number immiscible displacements in horizontal rough-walled fractures: Influence of local in-phase curvature». *Water resources research* (1998), vol. 34, nº 12, 3215-3234.
- [11] Disponible en: <http://www.infovis.net/printMag.php?num=137&lang=1> [Consulta: Abril 2009]
- [12] BIGGS, NORMAN L. *Matemática discreta*. Barcelona: Vicens-Vives, 1994.

- [13] ROSEN, KENNETH H. *Matemática discreta y sus aplicaciones*. Madrid: McGraw-Hill, 2004.
- [14] COMELLAS, F.; FÀBREGA, J.; SÁNCHEZ, A.; SERRA, O. *Matemática discreta*. Barcelona: UPC, 2001.
- [15] BASART I MUÑOZ, J. M. *Grafs: fonaments i algoritmes*. Barcelona: Manuals de la Universitat Autònoma de Barcelona, 1994.
- [16] Disponible en: <http://www.personal.us.es/preyes/Docencia/Introduccion.pdf> [Consulta: Febrero 2009].
- [17] PROCACCI, A.; SCOPPOLA, B. «Infinite graphs with a nontrivial bond percolation threshold: some sufficient conditions». *Journal of Statistical Physics* (2004), vol. 115, nº 3-4, 1113-1127.
- [18] WIERMAN, JOHN C.; NAOR, DORA PASSEN. «Desirable properties of universal formulas for percolation thresholds». *Congressus Numerantium* (2003), vol. 163, 125-142 [05C80](#) (90B15).
- [19] Disponible en: <http://www.iac.es/gabinete/difus/ciencia/silbia/caos.htm> [Consulta: Julio 2009].
- [20] Disponible en: <http://www.ambigrama.galeon.com/aficiones1377938.html> [Consulta: Julio 2009].
- [21] Disponible en: <http://victoriamx.com/fractales/fracweb/Fractais-Semin8.htm> [Consulta: Febrero 2009].
- [22] Disponible en: http://omega.ilce.edu.mx:3000/sites/ciencia/volumen3/ciencia3/147/htm/sec_5.htm [Consulta: Abril 2009].
- [23] Disponible en: <http://fafnir.phyast.pitt.edu/myjava/perc/percTest.html> [Consulta: Septiembre 2009].

ANEJOS

ANEJO 1. TEORÍA DEL CAOS Y FRACTALES

ANEJO1. TEORÍA DEL CAOS Y FRACTALES

En la naturaleza existen innumerables fenómenos que no responden a ningún modelo lineal y sin embargo, no pueden ser tratados como fenómenos totalmente aleatorios. El término **caos** se ha acostumbrado a asociar erróneamente a la aleatoriedad.

A finales del S. XIX, el científico francés Poincaré planteó por primera vez este conflicto y resumió lo aleatorio y el azar en la cita: *El azar no es más que la medida de la ignorancia del hombre*, que desembocaría en la denominada *Teoría del Caos*.

En ella se pueden abordar fenómenos tan cotidianos y difíciles de predecir como son el clima, el cuerpo humano, las epidemias, el crecimiento demográfico, los atascos de tráfico y un largo etcétera. Todos ellos tienen la característica que no pueden ser estudiados con herramientas físico-matemáticas clásicas, ni ser tratados como fenómenos aleatorios [20].

La Teoría del Caos se resume diciendo que el caos no es un fenómeno aleatorio, sino que sigue un determinado orden, es decir, es impredecible, pero a la vez determinable.

Un ejemplo muy ilustrativo en el campo de la psicología son las llamadas imágenes ambiguas mostradas en la **Figura 78**. No se puede predecir por qué imagen de las dos se decantará el observador. Dentro de esta inestabilidad pasamos, en las imágenes llamadas *la vieja y la joven* y *el indio esquimal*, de un estado estable (una de las dos percepciones, como podría ser la vieja y el indio), a otro estado estable (la joven y el esquimal) de forma inestable.

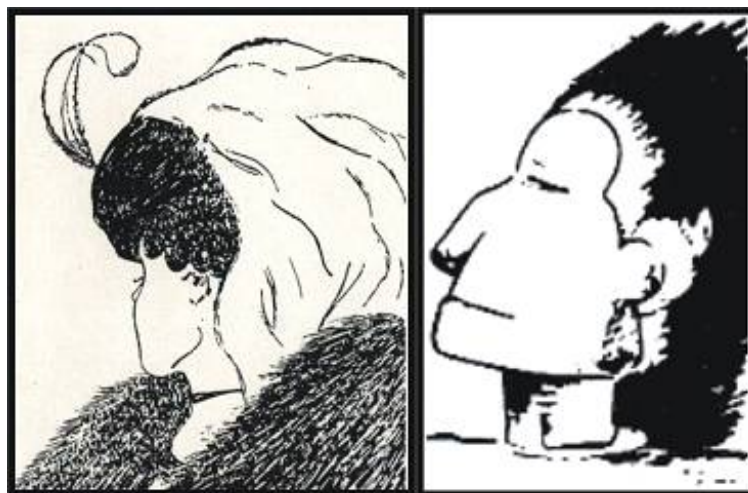


Figura 78. Imágenes ambiguas: La vieja y la joven/ el indio esquimal [20].

La Teoría del Caos es la rama de las Matemáticas que trata acerca del comportamiento cualitativo a largo plazo de un sistema dinámico. No se busca una solución exacta, sino poder saber si el sistema tiende a estabilizarse a largo plazo o si sufrirá grandes variaciones al modificar las condiciones iniciales, entre otras muchas cuestiones.

Aunque suelen confundirse en literatura científica, los términos caos y fractal no son sinónimos. Sí es cierto que muchos fenómenos caóticos tienen una estructura fractal.

Se denominan fractales aquellos objetos geométricos cuya estructura básica se repite en diferentes escalas. El término fue propuesto por Benoît Mandelbrot (1975).

Si se toma una fracción del objeto, por pequeña que sea, se observa que es una réplica de su estructura principal. Para etiquetar un objeto como fractal, éste debe cumplir una de las siguientes características:

- Poseer detalle en cualquier escala de observación.
- Tener autosemejanza exacta o estadística.
- Ser definido recursivamente, o que su dimensión sea fraccionaria.

A continuación, la **Figura.79** nos muestra el triángulo de Sierpinski, en el que se observa cómo se puede dividir el triángulo equilátero inicial en infinitos triángulos idénticos.

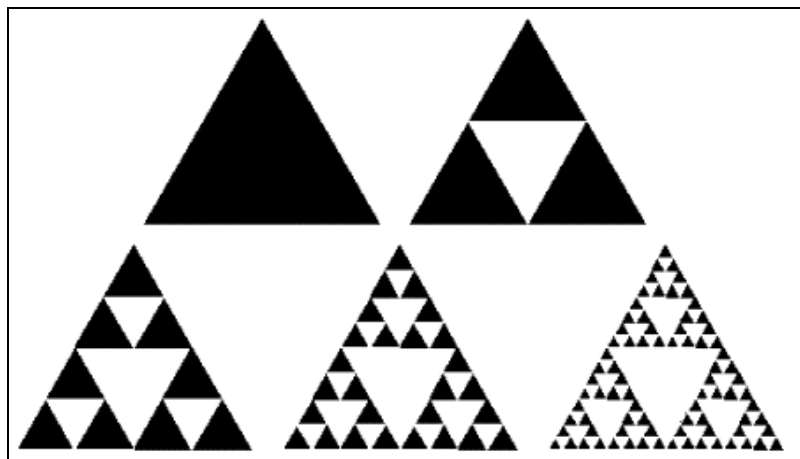


Figura 79. Triángulo de Sierpinski [21].

Otro conocido caso de estructura fractal fue desarrollado por el matemático sueco Hegel Koch. En la **Figura 80** se parte de un triángulo equilátero al que se le van añadiendo triángulos invertidos, formándose así una estrella de David, llegando al llamado copo de nieve de Koch.

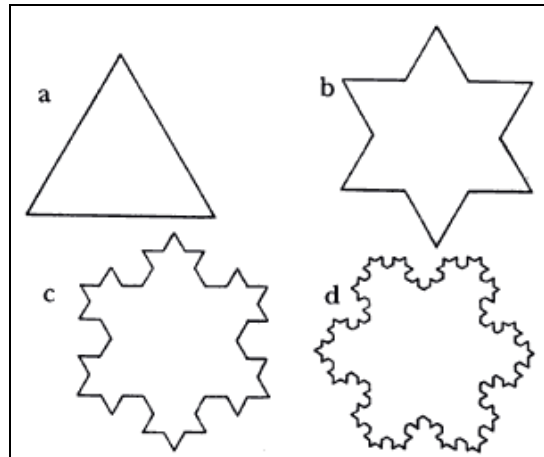


Figura 80. Copo de nieve de Koch [22].

En el último ejemplo que se muestra en la **Figura 81**, se observa un agregado o cluster resultante del fenómeno de la percolación, que no es más que un objeto fractal. He aquí la razón por la que se ha considerado oportuna una introducción al apasionante mundo de los fractales.

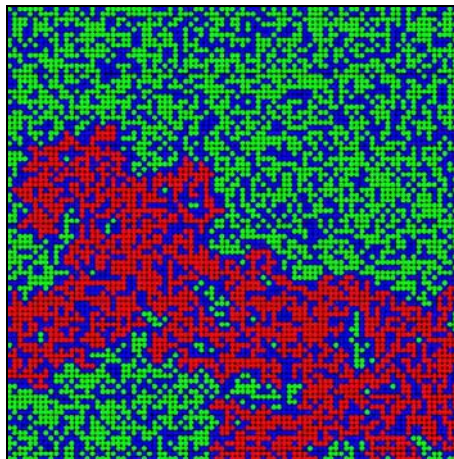


Figura 81. Simulación de una percolación por nodos [23].

ANEJO 2. ALGORITMO Y PROGRAMA
BINGO_PERCOLATION

ANEJO 2. ALGORITMO Y PROGRAMA BINGO_PERCOLATION

El algoritmo del programa es el siguiente:

1. Introducción de datos.
2. Calcula la Matriz de Conectividades.
3. Crea las coordenadas que limitan la malla.
4. Calcula los nodos inicial y final de cada arista.
5. Elige una arista aleatoria: caso isótropo con la misma probabilidad para cada arista y caso anisótropo, con mayor peso de probabilidad en la dirección X o Y.
6. Comprueba si se han generado todas las aristas. Si no es el caso, pasa a 7. si se han generado las aristas, pasa a 13.
7. Añade la arista al camino correspondiente. Si ya existe un camino con un nodo en común, la incorpora al existente. Si no existe, crea un nuevo camino.
8. Agrupa los posibles caminos que tienen nodos en común.
9. Comprueba si el camino percola. Si percola, pasa a 10, si no, pasa a 12.
10. Actualiza el número de caminos percolantes.
11. Escribe los ficheros de los resultados de los caminos percolantes. Vuelve a 5.
12. Escribe los ficheros de los resultados de los caminos no percolantes. Vuelve a 5.
13. Escribe los ficheros de los resultados globales como son la etiqueta de cada arista generada, el tiempo CPU, fecha, etc.

A continuación se recoge el código fuente del programa *BINGO_PERCOLATION*.

```

SUBROUTINE ADD_EDGE2CONNECTED_PATHS
; (NARIS_TOT , NEDGE , NMAX_ARIS_PATH , NMAX_PATHS
; , FLAG_EDGES , CONEC , TOTAL_PATHS)

*****
*
* PURPOSE Adds a given edge NEDGE to paths that connect with it
*
* DESCRIPTION : 1) At this step, NEDGE is a path itself
*               LOOP OVER ALL POSSIBLE EDGES, LOOKING FOR THE PREV. GEN. ONES
*               2) Once it is sure that PREV_EDGE was prev. gen. and is not
*                  NEDGE, checks if NEDGE and PREV_EDGE are connected
*               3) If they are connected, adds NEDGE to all path where
*                  PREV_EDGE is present
*               END LOOP
*
* EXTERNAL VARIABLES: ARRAYS
*
* FLAG_EDGES          Vector containing flags of previously generated paths (NARIS_TOT)
* CONEC               Connectivity matrix (6,NARIS_TOT)
* TOTAL_PATHS         Matrix containing all possible PATHS
*
* EXTERNAL VARIABLES: SCALARS
*
* NARIS_TOT           Total number of edges
* NEDGE               Number of edge being analysed
* NMAX_ARIS_PATH       Maximum number of edges defining a path
* NMAX_PATHS          Maximum number of percol/non-percol paths
*
* INTERNAL VARIABLES: ARRAYS
*
* INTERNAL VARIABLES: SCALARS
*
* FLAG_EDGE           Flag of PREV_EDGE
* IARIS               Dummy counter of edges of a given path
* ICONEC_OK           If 1, NEDGE and PREV_EDGE are connected
* IPATH               Dummy counter of paths
* JARIS               Dummy counter of edges of a given path
* PREV_EDGE           Target edge that may connect with NEDGE
*
* FUNCTIONS AND SUBROUTINES REFERENCED
*
* CHECK_CONECTION     Checks if two edges are connected
*
* HISTORY
*
* AAR                06-2006      First coding
*
*****

IMPLICIT NONE
INTEGER*4 NARIS_TOT,NEDGE,NMAX_ARIS_PATH,NMAX_PATHS
INTEGER*4 FLAG_EDGES(NARIS_TOT),CONEC(6,NARIS_TOT)
; ,TOTAL_PATHS(NMAX_ARIS_PATH,NMAX_PATHS)
INTEGER*4 PREV_EDGE,FLAG_EDGE,ICONEC_OK,IPATH,IARIS,JARIS

C_____ 1) At this step, NEDGE is a path itself

TOTAL_PATHS(1,NEDGE)=NEDGE

C_____ LOOP OVER ALL POSSIBLE EDGES, LOOKING FOR THE PREV. GEN. ONES

DO PREV_EDGE=1,NARIS_TOT

FLAG_EDGE=FLAG_EDGES(PREV_EDGE)

IF (FLAG_EDGE.NE.0 .AND. FLAG_EDGE.NE.NEDGE) THEN ! Previously gen. edge found

C_____ 2) Once it is sure that PREV_EDGE was prev. gen. and is not NEDGE,
C_____ checks if NEDGE and PREV_EDGE are connected

CALL CHECK_CONECTION
; (NEDGE , PREV_EDGE , NARIS_TOT , ICONEC_OK , CONEC)

C_____ 3) If they are connected, adds NEDGE to all path where PREV_EDGE
C_____ is present

IF (ICONEC_OK.NE.0) THEN

DO IPATH=1,NMAX_PATHS ! Loop over paths

DO IARIS=1,NMAX_ARIS_PATH ! Loop over edges of a given path

IF (TOTAL_PATHS(IARIS,IPATH).EQ.PREV_EDGE) THEN ! PREV_EDGE belongs to path IPATH

```

```

DO JARIS=1,NMAX_ARIS_PATH      ! Looks for a void position
    IF (TOTAL_PATHS(JARIS,IPATH).EQ.0) THEN
        TOTAL_PATHS(JARIS,IPATH)=NEDGE      ! And adds NEDGE...
        TOTAL_PATHS(1,NEDGE)=0              ! And deletes single edge as a path
        GOTO 11
    END IF
END DO

END IF

11 CONTINUE

END DO ! IARIS=1,NMAX_ARIS_PATH

END DO ! ILOOP=1,NMAXPATHS

END IF ! ICONEC_OK.NE.0

END IF ! FLAG_EDGE.NE.0 .AND. FLAG_EDGE.NE.NEDGE

END DO ! PREV_EDGE=1,NARIS_TOT

RETURN
END

*****
*****
*****

SUBROUTINE CHECK_CONECTION
; (EDGE1 ,EDGE2 ,NARIS_TOT ,OUTPUT ,CONEC)

*****
*
* PURPOSE Checks if two edges are connected or not
*
* DESCRIPTION : OUTPUT=0 (no connection); OUTPUT=1 (OK !!!)
*
* EXTERNAL VARIABLES: ARRAYS
*
* CONEC                      Connectivity matrix (6,NARIS_TOT)
*
* EXTERNAL VARIABLES: SCALARS
*
* EDGE1                      First edge to be checked
* EDGE2                      Second edge to be checked
* NARIS_TOT                  Total number of edges
* OUTPUT                     0: no connection; 1: connection
*
* INTERNAL VARIABLES: ARRAYS
*
* INTERNAL VARIABLES: SCALARS
*
* IPOS                      Dummy counter (1:6)
*
* FUNCTIONS AND SUBROUTINES REFERENCED
*
* HISTORY
*
* AAR                      06-2006      First coding
*
*****

IMPLICIT NONE
INTEGER*4 NARIS_TOT,EDGE1,EDGE2,OUTPUT
INTEGER*4 CONEC(6,NARIS_TOT)
INTEGER*4 IPOS

OUTPUT=0      ! By now, EDGE1 and EDGE2 are not connected

DO IPOS=1,6
    IF (EDGE1.EQ.CONEC(IPOS,EDGE2)) THEN      ! OK, EDGE1 and EDGE2 connected
        OUTPUT=1
        GOTO 10
    END IF
END DO

10 RETURN
END

```

```

SUBROUTINE CALC_TOTAL LENGHT_PATH
; (LENGHT_PATH ,NARIS_TOT ,NMAX_ARIS_PATH ,NODES_X
; ,NODES_Y ,COORD ,NODES_EDGE ,PATH)

```

```

*****

```

```

*
* PURPOSE Calculates the total lenght of a given path
*
* DESCRIPTION
*
* EXTERNAL VARIABLES: ARRAYS
*
*   COORD          Array containing nodal coordinates (NODES_X*NODES_Y,2)
*   NODES_EDGE     Matrix containing nodes belonging to a given edge (NARIS_TOT,2)
*   PATH           Current path (NMAX_ARIS_PATH)
*
* EXTERNAL VARIABLES: SCALARS
*
*   LENGHT_PATH    Lenght of the path (OUPUT)
*   NARIS_TOT      Total number of edges
*   NMAX_ARIS_PATH Maximum number of edges defining a path
*   NODES_X        Number of nodes X-direction
*   NODES_Y        Number of nodes Y-direction
*
* INTERNAL VARIABLES: ARRAYS
*
* INTERNAL VARIABLES: SCALARS
*
*   IEDGE          Dummy counter of edges
*   LENGHT_EDGE    Lenght of an edge of the given path
*   NEDGE          Number of edge
*   NODE1          First node of NEDGE
*   NODE2          Second node of NEDGE
*   X1             X coord. of NODE1
*   X2             X coord. of NODE2
*   Y1             Y coord. of NODE1
*   Y2             Y coord. of NODE2
*
* FUNCTIONS AND SUBROUTINES REFERENCED
*
* HISTORY
*
*   AAR           06-2006       First coding
*

```

```

*****

```

```

IMPLICIT NONE
INTEGER*4 NMAX_ARIS_PATH,NODES_X,NODES_Y,NARIS_TOT
INTEGER*4 PATH(NMAX_ARIS_PATH),NODES_EDGE(NARIS_TOT,2)
REAL*8 LENGHT_PATH
REAL*8 COORD(NODES_X*NODES_Y,2)
INTEGER*4 IEDGE,NEDGE,NODE1,NODE2
REAL*8 LENGHT_EDGE,X1,X2,Y1,Y2

LENGHT_PATH=0.0D0
DO IEDGE=1,NMAX_ARIS_PATH
  NEDGE=PATH(IEEDGE)
  IF (NEDGE.NE.0) THEN
    NODE1=NODES_EDGE(NEDGE,1)
    NODE2=NODES_EDGE(NEDGE,2)
    X1=COORD(NODE1,1)
    X2=COORD(NODE2,1)
    Y1=COORD(NODE1,2)
    Y2=COORD(NODE2,2)
    LENGHT_EDGE = DSQRT ( (X2-X1)*(X2-X1) + (Y2-Y1)*(Y2-Y1) )
    LENGHT_PATH=LENGHT_PATH + LENGHT_EDGE
  END IF
END DO

RETURN
END

```

```

SUBROUTINE CHECK_PERCOLATION
; (NARIS_TOT ,NMAX_ARIS_PATH ,NODES_X ,NODES_Y ,OUTPUT
; ,CONEC ,COORD ,NODES_EDGE ,PATH)

*****
*
* PURPOSE Checks if a given path is or nor percolating
*
* DESCRIPTION : Simply checks that one of its edges touches the top of the mesh
*               and another one the minimum
*
* EXTERNAL VARIABLES: ARRAYS
*
* CONEC           Connectivity matrix (6,NARIS_TOT)
* COORD           Array containing nodal coordinates (NODES_X*NODES_Y,2)
* NODES_EDGE      Matrix containing nodes belonging to a given edge (NARIS_TOT,2)
* PATH            Path being analyzed (NMAX_ARIS_PATH)
*
* EXTERNAL VARIABLES: SCALARS
*
* NARIS_TOT       Total number of edges
* NMAX_ARIS_PATH  Maximum number of edges defining a path
* NODES_X         Number of nodes X-direction
* NODES_Y         Number of nodes Y-direction
* OUTPUT          1: OK, the path percolates; 0 otherwise
*
* INTERNAL VARIABLES: ARRAYS
*
* INTERNAL VARIABLES: SCALARS
*
* NEDGE           Number of current edge
* IPATH           Number of path being analysed
* IEDGE           Dummy counter of edges
* NODE1           Node 1 of current edge
* NODE2           Node 2 of current edge
* IOK_YMAX        Condition of touching the top (=1)
* IOK_YMIN        Condition of touching the bottom (=1)
*
* FUNCTIONS AND SUBROUTINES REFERENCED
*
* HISTORY
*
*   AAR           06-2006       First coding
*
*****

IMPLICIT NONE
INTEGER*4 NMAX_ARIS_PATH,NARIS_TOT,OUTPUT,NODES_X,NODES_Y
INTEGER*4 PATH(NMAX_ARIS_PATH),NODES_EDGE(NARIS_TOT,2)
; ,CONEC(6,NARIS_TOT)
INTEGER*4 NEDGE,IPATH,IEDGE,NODE1,NODE2,IOK_YMAX,IOK_YMIN
REAL*8 COORD(NODES_X*NODES_Y,2)
REAL*8 YMAX

C_____ Initializes indicator variables

      IOK_YMAX=0
      IOK_YMIN=0
      OUTPUT=0
      YMAX=COORD(1,2)

C_____ Loop over edges of the path

      DO IEDGE=1,NMAX_ARIS_PATH

          IF (PATH(IEDGE).NE.0) THEN

              NEDGE=PATH(IEDGE)      ! Edge number

C_____ Given an edge, seeks NODES

```

```

        NODE1=NODES_EDGE(NEDGE,1)
        NODE2=NODES_EDGE(NEDGE,2)

        IF ( COORD(NODE1,2).EQ.YMAX .OR.
;          COORD(NODE2,2).EQ.YMAX) THEN
            IOK_YMAX=1                ! One of the nodes of this edge is on top
        END IF

        IF ( COORD(NODE1,2).EQ.0.0 .OR.
;          COORD(NODE2,2).EQ.0.0) THEN
            IOK_YMIN=1                ! One of the nodes of this edge is on bottom
        END IF

    END IF

END DO

C_____ Checks both conditions at the same time

    IF (IOK_YMAX.EQ.1 .AND. IOK_YMIN.EQ.1) OUTPUT=1

RETURN

END

```

```

SUBROUTINE CREA_CONEC
; (IDEBUG ,NARIS_TOT ,NARIS_TOT_X ,NARIS_X ,NODES_Y ,CONEC)

*****
*
* PURPOSE Creates connectivity matrix. Connectivity is defined as, for a given
* edge, which edges are connected to.
*
* DESCRIPTION : 1) Calculates number of vertical edges per horiz. "layer"
*               2) Generates connectivity of horizontal edges
*               3) Generates connectivity of vertical edges
*               4) Echoes connectivity matrix
*
* EXTERNAL VARIABLES: ARRAYS
*
* CONEC                      Connectivity matrix (6,NARIS_TOT)
*
* EXTERNAL VARIABLES: SCALARS
*
* IDEBUG                      Option for code debugging
* NARIS_TOT                    Total num. of edges
* NARIS_TOT_X                  Total num. of horiz. edges
* NARIS_X                      Number of edges in a row (given a fixed Y-coordinate)
* NODES_Y                      Number of nodes Y-direction
*
* INTERNAL VARIABLES: ARRAYS
*
* INTERNAL VARIABLES: SCALARS
*
* NODES_Y                      Number of nodes Y-direction
* HEDGE                        Counter for Horizontal edges (1:NARIS_X)
* IFIL                          Counter of rows (1:NODES_Y)
* IPOS                          Dummy counter of Rows at CONEC (1:6)
* JCOL                          Counter of columns at CONEC (1:6)
* NEDGE                        Number of edge (Id)
* NBASE_H                      Auxiliari counter for horiz. edges
* NBASE_V                      Auxiliari counter for vert. edges
* NVERT                        Number of vert. edges connecting a row of the mesh
*
* FUNCTIONS AND SUBROUTINES REFERENCED
*
* HISTORY
*
* AAR          06-2006      First coding
*
*****

IMPLICIT NONE
INTEGER*4 IDEBUG,NODES_Y,NARIS_TOT,NARIS_X,NARIS_TOT_X
; ,CONEC(6,NARIS_TOT)
INTEGER*4 HEDGE,NEDGE,IFIL,JCOL,IPOS,NVERT,NBASE_H,NBASE_V

C_____ Defines some useful variables

NVERT=NARIS_X-1      ! Number of vertical edges per horizontal layer

C_____ Connectivity of HORIZONTAL EDGES

DO IFIL=1,NODES_Y

  NBASE_H=NARIS_X*(IFIL-1)

  DO HEDGE=1,NARIS_X

    NEDGE=NBASE_H+HEDGE

    IF (HEDGE.EQ.1) THEN      ! Adjacent to Left Boundary

      CONEC(1,NEDGE)=NEDGE+1      ! Edge to its right
      CONEC(2,NEDGE)=NVERT*(IFIL-2)+NARIS_TOT_X+1  ! Upper Edge
      CONEC(3,NEDGE)=NVERT*(IFIL-1)+NARIS_TOT_X+1  ! Lower Edge
      IF (IFIL.EQ.1) CONEC(2,NEDGE)=0      ! Upper layer has not "upper" conec.
      IF (IFIL.EQ.NODES_Y) CONEC(3,NEDGE)=0    ! Lower layer has not "lower" conec.

    ELSE IF (HEDGE.EQ.NARIS_X) THEN      ! Adjacent to Right Boundary

      CONEC(1,NEDGE)=NEDGE-1      ! Edge to its left

```



```

CONEC (2,NEDGE)=NVERT*(IFIL-2)+NARIS_TOT_X+NVERT ! Upper Edge
CONEC (3,NEDGE)=NVERT*(IFIL-1)+NARIS_TOT_X+NVERT ! Lower Edge
IF (IFIL.EQ.1) CONEC (2,NEDGE)=0 ! Upper layer has not "upper" conec.
IF (IFIL.EQ.NODES_Y) CONEC (3,NEDGE)=0 ! Lower layer has not "lower" conec.

ELSE ! Intermediate

CONEC (1,NEDGE)=NEDGE+1 ! Edge to its right
CONEC (2,NEDGE)=NEDGE-1 ! Edge to its left
NBASE_V=NVERT*(IFIL-2)+NARIS_TOT_X
CONEC (3,NEDGE)=NBASE_V+NEDGE-1 ! Upper left
CONEC (4,NEDGE)=CONEC (3,NEDGE)+1 ! Upper right
NBASE_V=NVERT*(IFIL-1)+NARIS_TOT_X
CONEC (5,NEDGE)=NBASE_V+NEDGE-1 ! Lower left
CONEC (6,NEDGE)=CONEC (5,NEDGE)+1 ! Lower right
IF (IFIL.EQ.1) THEN
CONEC (3,NEDGE)=0
CONEC (4,NEDGE)=0
ENDIF

IF (IFIL.EQ.NODES_Y) THEN
CONEC (5,NEDGE)=0
CONEC (6,NEDGE)=0
END IF

END IF

END DO

END DO

C_____ Connectivity of VERTICAL EDGES

DO NEDGE=NARIS_TOT_X+1,NARIS_TOT
CONEC (1,NEDGE)=NEDGE-NVERT ! Upper Edge
IF (NEDGE.LE.NARIS_TOT_X+NVERT) CONEC (1,NEDGE)=0
CONEC (2,NEDGE)=NEDGE+NVERT ! Lower edge
IF (NEDGE.GT.NARIS_TOT-NVERT) CONEC (2,NEDGE)=0
IPOS=2
DO IFIL=1,NARIS_TOT_X ! Loop over horizontal connectivity

DO JCOL=1,6
IF (NEDGE.EQ.CONEC (JCOL,IFIL)) THEN
IPOS=IPOS+1
CONEC (IPOS,NEDGE)=IFIL
END IF
END DO

END DO

END DO

C_____ Echoes connectivity matrix

IF (IDEBUG.NE.0) THEN
WRITE (100,100)
100 FORMAT (//,' MATRIX OF LOCATIONS:',/
; , ' ===== == =====',//)
DO NEDGE=1,NARIS_TOT

WRITE (100,200) NEDGE,CONEC (1,NEDGE),CONEC (2,NEDGE)
; ,CONEC (3,NEDGE),CONEC (4,NEDGE)
; ,CONEC (5,NEDGE),CONEC (6,NEDGE)

200 FORMAT ('Edge: ',I5,' is adjacent to edges: ',6I5)

END DO
END IF

RETURN
END

```

```

SUBROUTINE CREA_COORD
; (DELTAX ,DELTAY ,IDEBUG ,NODES_X ,NODES_Y ,COORD)

*****
*
* PURPOSE Generates nodal coordinates
*
* DESCRIPTION : First node is upper left corner. Then cycles first on X,
*               later on Y. Thus, last node is lower right corner
*
* EXTERNAL VARIABLES: ARRAYS
*
*   COORD           Array containing nodal coordinates (NODES_X*NODES_Y,2)
*
* EXTERNAL VARIABLES: SCALARS
*
*   DELTAX          Edge size (X-direction)
*   DELTAY          Edge size (Y-direction)
*   IDEBUG          Option for code debugging
*   NODES_X         Number of nodes X-direction
*   NODES_Y         Number of nodes Y-direction
*
* INTERNAL VARIABLES: ARRAYS
*
* INTERNAL VARIABLES: SCALARS
*
*   IFIL            Dummy counter of mesh (Y) rows
*   INUD            Dummy counter of nodes
*   NUMBER_NODE     Number of node as ordenated in the mesh
*   XCOORD          X coordinate of a given node
*   YCOORD          X coordinate of a given node
*
* FUNCTIONS AND SUBROUTINES REFERENCED
*
* HISTORY
*
*   AAR            06-2006      First coding
*
*****

IMPLICIT NONE
INTEGER*4 IDEBUG,NODES_X,NODES_Y
REAL*8 DELTAX,DELTAY,COORD(NODES_X*NODES_Y,2)
INTEGER*4 IFIL,INUD,NUMBER_NODE
REAL*8 XCOORD,YCOORD

C_____ Generates nodal coordinates

YCOORD=DELTAY*FLOAT (NODES_Y)

DO IFIL=1,NODES_Y

  YCOORD=YCOORD-DELTAY
  XCOORD=-1.0*DELTAX

  DO INUD=1,NODES_X
    NUMBER_NODE=(IFIL-1)*NODES_X+INUD
    XCOORD=XCOORD+DELTAX
    COORD (NUMBER_NODE,1)=XCOORD
    COORD (NUMBER_NODE,2)=YCOORD

  END DO

```

```

        ENDDO

        IF (IDEBUG.NE.0) THEN

            WRITE(100,100)
100      FORMAT(//,' NODAL COORDINATES :',/,',', ' ===== =',//)

            DO INUD=1,NODES_X*NODES_Y
                WRITE(100,200) INUD,COORD(INUD,1),COORD(INUD,2)
            END DO
200      FORMAT(I10,2F10.3)

        END IF

        RETURN
    END

```

```

SUBROUTINE DEPICT_PATH
; (IPATH ,IPERC ,NFILE ,NARIS_GEN ,NARIS_PATH ,NARIS_TOT
; ,NMAX_ARIS_PATH ,NODES_X ,NODES_Y ,PATH_LENGTHT ,COORD
; ,NODES_EDGE ,PATH)

*****
*
* PURPOSE Depicts a given path
*
* DESCRIPTION :
*
* EXTERNAL VARIABLES: ARRAYS
*
* COORD          Array containing nodal coordinates (NODES_X*NODES_Y,2)
* NODES_EDGE     Matrix containing nodes belonging to a given edge (NARIS_TOT,2)
* PATH           Path being depicted (NARIS_TOT)
*
* EXTERNAL VARIABLES: SCALARS
*
* IPATH          Number of path being depicted
* IPERC          =1 Percolating path; 0 otherwise
* NFILE          Output file unit number
* NARIS_GEN      Number of edges generated until this moment
* NARIS_PATH     Path size
* NARIS_TOT      Total number of edges
* NMAX_ARIS_PATH Maximum number of edges defining a path
* NODES_X        Number of nodes X-direction
* NODES_Y        Number of nodes Y-direction
* PATH_LENGTHT   Length of current path
*
* INTERNAL VARIABLES: ARRAYS
*
* INTERNAL VARIABLES: SCALARS
*
* IEDGE          Dummy counter of edges of the path
* NEDGE          Number of current edge
* NODE1          First node of NEDGE
* NODE2          Second node of NEDGE
* J              Dummy counter (1 or 2)
*
* FUNCTIONS AND SUBROUTINES REFERENCED
*
* HISTORY
*
* AAR          06-2006      First coding
*
*****

IMPLICIT NONE
INTEGER*4 NMAX_ARIS_PATH,NODES_X,NODES_Y,NFILE,IPERC,IPATH
; ,NARIS_PATH,NARIS_GEN,NARIS_TOT
INTEGER*4 PATH(NMAX_ARIS_PATH),NODES_EDGE(NARIS_TOT,2)
REAL*8 PATH_LENGTHT
REAL*8 COORD(NODES_X*NODES_Y,2)
INTEGER*4 IEDGE,NEDGE,NODE1,NODE2,J

C_____ Writes header of path

IF (IPERC.EQ.1) THEN
WRITE(NFILE,100) IPATH
; ,DFLOAT(NARIS_GEN)*100.0D0/DFLOAT(NARIS_TOT)
; ,PATH_LENGTHT,NARIS_PATH
100 FORMAT('"PERC. PATH :',I10,
; ' AFTER GEN ',F10.0,'% OF TOTAL EDGES',',',
; '" LENGTH=',F10.0,' NUM. EDGES=',I10,'"','-',2')
ELSE
WRITE(NFILE,200) IPATH
; ,DFLOAT(NARIS_GEN)*100.0D0/DFLOAT(NARIS_TOT)
; ,PATH_LENGTHT,NARIS_PATH
200 FORMAT('"NON PERC. PATH :',I10,
; ' AFTER GEN ',F10.0,'% OF TOTAL EDGES',',',

```

```

;          '" LENGHT=',F10.0,' NUM. EDGES=',I10,'"',''-2')
END IF

C_____ Loop over path edges

DO IEDGE=1,NMAX_ARIS_PATH
  NEDGE=PATH(IEEDGE)
  IF (NEDGE.NE.0) THEN
    NODE1=NODES_EDGE(NEDGE,1)
    NODE2=NODES_EDGE(NEDGE,2)

    WRITE(NFILE,300) NEDGE
300    FORMAT('"EDGE :',I10,'"','"',-2')
    WRITE(NFILE,'(2E10.3)') (COORD(NODE1,J),J=1,2)
    WRITE(NFILE,'(2E10.3)') (COORD(NODE2,J),J=1,2)
  END IF
END DO

RETURN
END

```

```

SUBROUTINE EDGE2NODE
; (IDEBUG ,NARIS_TOT ,NARIS_TOT_X ,NARIS_X ,NODES_X ,NODES_Y
; ,CONEC ,NODES_EDGE)

```

```

*****

```

```

*
* PURPOSE Calculates nodes relationship EDGE-NODE
*
* DESCRIPTION : First, nodes belonging to horizontal edges are calculated.
*               Nodes belonging to vertical edges are calculated from this
*               first calculation and the edge connectivity.
*
* EXTERNAL VARIABLES: ARRAYS
*
*   CONEC           Connectivity matrix (6,NARIS_TOT)
*   NODES_EDGE      Matrix containing nodes belonging to a given edge (NARIS_TOT,2)
*
* EXTERNAL VARIABLES: SCALARS
*
*   IDEBUG          Option for code debugging
*   NARIS_TOT       Total num. of edges
*   NARIS_TOT_X     Total num. of horiz. edges
*   NARIS_X         Number of edges in a row (given a fixed Y-coordinate)
*   NODES_X         Number of nodes X-direction
*   NODES_Y         Number of nodes Y-direction
*
* INTERNAL VARIABLES: ARRAYS
*
* INTERNAL VARIABLES: SCALARS
*
*   HEDGE           Dummy counter of horizontal edges
*   HEDGE1          Horizontal edge connecting the vertical target edge
*   HEDGE2          Horizontal edge connecting the vertical target edge
*   IAUX            Aux. dummy variable (function of the layer)
*   IFIL           Dummy counter of mesh (Y) rows
*   ILAYER          Dummy counter of layers (1= upper, 2=lower)
*   INUD            Dummy counter of nodes (1,2)
*   IPOS           Aux. dummy updatable variable
*   NBASE_H         Auxiliar counter of horiz. edges
*   NEDGE           Number of horizontal edge
*   NNUD_BASE       Auxiliar counter of nodes
*   NNUD_H2         Nodes of HEDGE2
*   NNUD1_H1        First node of HEDGE1
*   NNUD2_H1        Second node of HEDGE1
*   NODE_COMMON     Common node of vertical and horizontal edges
*
* FUNCTIONS AND SUBROUTINES REFERENCED
*
* HISTORY
*
*   AAR           06-2006       First coding
*
*****

```

```

IMPLICIT NONE

```

```

INTEGER*4 NODES_X,NODES_Y,NARIS_X,NARIS_TOT,NARIS_TOT_X,IDEBUG
INTEGER*4 NODES_EDGE(NARIS_TOT,2),CONEC(6,NARIS_TOT)
INTEGER*4 IFIL,NBASE_H,NNUD_BASE,IPOS,HEDGE1,HEDGE2,NEDGE
;           ,NNUD1_H1,NNUD2_H1,NNUD_H2,NODE_COMMON,ILAYER
;           ,HEDGE,INUD,IAUX

```

```

C_____ Defines nodes belonging to horizontal edges

```

```

DO IFIL=1,NODES_Y

```

```

    NBASE_H=(IFIL-1)*NARIS_X
    NNUD_BASE=(IFIL-1)*NODES_X
    IPOS=1

```

```

    DO HEDGE=1,NARIS_X

```

```

        NEDGE=NBASE_H+HEDGE
        NODES_EDGE (NEDGE,1)=NNUD_BASE+IPOS
        IPOS=IPOS+1
        NODES_EDGE (NEDGE,2)=NNUD_BASE+IPOS

    END DO

END DO

C_____ Defines nodes belonging to vertical edges

DO NEDGE=NARIS_TOT_X+1,NARIS_TOT

    DO ILAYER=1,2                                ! Vertical connects with two layers

        IAUX=0
        IF (ILAYER.EQ.2) IAUX=2

        HEDGE1=CONEC (3+IAUX,NEDGE)                ! Horizontal connectivity
        HEDGE2=CONEC (4+IAUX,NEDGE)

        NNUD1_H1=NODES_EDGE (HEDGE1,1)            ! Nodes of HEDGE1
        NNUD2_H1=NODES_EDGE (HEDGE1,2)

        DO INUD=1,2                                ! Loop over nodes of HEDGE2
            NNUD_H2=NODES_EDGE (HEDGE2,INUD)
            IF (NNUD_H2.EQ.NNUD1_H1 .OR. NNUD_H2.EQ.NNUD2_H1)
;                NODE_COMMON=NNUD_H2
            END DO
            NODES_EDGE (NEDGE,ILAYER) =NODE_COMMON

        END DO

    END DO

    IF (IDEBUG.NE.0) THEN

        WRITE (100,100)
100      FORMAT (//, ' NODES BELONGING TO EDGES :',/,
;          ' ===== =',//)
        DO NEDGE=1,NARIS_TOT
            WRITE (100,200) NEDGE, (NODES_EDGE (NEDGE,INUD), INUD=1,2)
        END DO
200      FORMAT (' Edge :',I10,' Nodes :',2I10)
    END IF

    RETURN
END

```

```

SUBROUTINE WRITE_CONEC_ONE_EDGE
; (CONEC,NARIS,NEDGE,NFILE)

IMPLICIT NONE
INTEGER*4 NARIS, I,NEDGE,NFILE,CONEC(NARIS,6)

WRITE (NFILE,100) NEDGE,CONEC(NEDGE,1),CONEC(NEDGE,2)
; ,CONEC(NEDGE,3),CONEC(NEDGE,4)
; ,CONEC(NEDGE,5),CONEC(NEDGE,6)

100 FORMAT('Edge: ',I5,' is adjacent to edges: ',6I5)

RETURN

END

```



```

SUBROUTINE GENER_LONG_PATHS
; (NMAX_ARIS_PATH , NMAX_PATHS , TOTAL_PATHS)

*****
*
* PURPOSE Generates long paths from shorter ones
*
* DESCRIPTION : See below
*
* EXTERNAL VARIABLES: ARRAYS
*
* TOTAL_PATHS          Matrix containing all paths (NMAX_ARIS_PATH,NMAX_PATHS)
*
* EXTERNAL VARIABLES: SCALARS
*
* NMAX_ARIS_PATH       Maximum number of edges defining a path
* NMAX_PATHS           Maximum number of percol/non-percol paths
*
* INTERNAL VARIABLES: ARRAYS
*
* INTERNAL VARIABLES: SCALARS
*
* EDGE1                Dummy counter for erasing zeros at paths
* EDGE2                Dummy counter for erasing zeros at paths
* EDGE_IPATH           Edge number of IPATH
* EDGE_JPATH           Edge number of JPATH
* IEDGE               Dummy counter of edges of IPATH
* IPATH               Path 1 being compared
* IPOS                Dummy counter
* JEDGE               Dummy counter of edges of JPATH
* JPATH               Path 2 being compared
* LASTPOS             Max(LASTPOS_IPATH,LASTPOS_JPATH)
* LASTPOS_IPATH       Last non-null pos. at IPATH
* LASTPOS_JPATH       Last non-null pos. at JPATH
* LONGER_PATH         Longer path among IPATH and JPATH
* NRECORDS_TO_BE_PASTED Number of edges of the shorter path
* PATH_TO_BE_PASTED   Path to be included (shorter) in the longer one
*
* FUNCTIONS AND SUBROUTINES REFERENCED
*
* GET_LAST_POSITION_PATH Gets the last non-null position of a given path
*
* HISTORY
*
* AAR          06-2006      First coding
*
*****

IMPLICIT NONE
INTEGER*4 NMAX_PATHS,NMAX_ARIS_PATH
INTEGER*4 TOTAL_PATHS(NMAX_ARIS_PATH,NMAX_PATHS)
INTEGER*4 IPATH,JPATH,IEDGE,JEDGE,EDGE_IPATH,EDGE_JPATH
; ,LASTPOS_IPATH,LASTPOS_JPATH,LASTPOS,LONGER_PATH
; ,NRECORDS_TO_BE_PASTED,PATH_TO_BE_PASTED,IPOS,EDGE1,EDGE2

C_____ Deletes repeated edges of a given path

DO IPATH=1,NMAX_PATHS
DO IEDGE=1,NMAX_ARIS_PATH-1
EDGE1=TOTAL_PATHS(IEDGE,IPATH)
IF (EDGE1.NE.0) THEN
DO JEDGE=IEDGE+1,NMAX_ARIS_PATH
EDGE2=TOTAL_PATHS(JEDGE,IPATH)
IF (EDGE2.NE.0 .AND. EDGE1.EQ.EDGE2) THEN
TOTAL_PATHS(JEDGE,IPATH)=0
END IF

```

```

        END DO
      END IF
    END DO
  END DO

C_____ Loop over paths

  DO IPATH=1,NMAX_PATHS-1

    DO JPATH=IPATH+1,NMAX_PATHS

C_____ Checks if IPATH and JPATH have a common edge

      DO IEDGE=1,NMAX_ARIS_PATH

        EDGE_IPATH=TOTAL_PATHS(IEEDGE,IPATH)

        IF (EDGE_IPATH.NE.0) THEN

          DO JEDGE=1,NMAX_ARIS_PATH

            EDGE_JPATH=TOTAL_PATHS(JEDGE,JPATH)

            IF (EDGE_JPATH.NE.0 .AND.
;              EDGE_JPATH.EQ.EDGE_IPATH) THEN

C_____ OK. IPATH and JPATH have a common edge

C_____ Calculates last non-null position at IPATH and JPATH

              CALL GET_LAST_POSITION_PATH
; (LASTPOS_IPATH ,NMAX_ARIS_PATH ,TOTAL_PATHS(1,IPATH))

              CALL GET_LAST_POSITION_PATH
; (LASTPOS_JPATH ,NMAX_ARIS_PATH ,TOTAL_PATHS(1,JPATH))

C_____ Gets longer path among IPATH and JPATH

              IF (LASTPOS_IPATH.GE.LASTPOS_JPATH) THEN ! IPATH is longer
                LASTPOS=LASTPOS_IPATH
                LONGER_PATH=IPATH
                NRECORDS_TO_BE_PASTED=LASTPOS_JPATH
                PATH_TO_BE_PASTED=JPATH
              ELSE
                LASTPOS=LASTPOS_JPATH
                LONGER_PATH=JPATH
                NRECORDS_TO_BE_PASTED=LASTPOS_IPATH
                PATH_TO_BE_PASTED=IPATH
              END IF ! LASTPOS_IPATH.GE.LASTPOS_JPATH

C_____ Adds shorter path to the longer one and deletes shorter one

              DO IPOS=1,NRECORDS_TO_BE_PASTED
                TOTAL_PATHS(LASTPOS+IPOS,LONGER_PATH)=
;                  TOTAL_PATHS(IPOS,PATH_TO_BE_PASTED)
                TOTAL_PATHS(IPOS,PATH_TO_BE_PASTED)=0
              END DO

              GOTO 1000 ! Search of two new connected paths

            END IF ! EDGE_JPATH.NE.0.AND.EDGE_JPATH.EQ.EDGE_IPATH

          END DO ! JEDGE=1,NMAX_ARIS_PATH

        END IF ! EDGE_IPATH.NE.0
      
```

```

        END DO ! IEDGE=1,NMAX_ARIS_PATH

1000    CONTINUE

        END DO ! JPATH=IPATH+1,NMAX_PATHS

        END DO ! IPATH=1,NMAX_PATHS-1

C_____ Deletes repeated edges of a given path

        DO IPATH=1,NMAX_PATHS
            DO IEDGE=1,NMAX_ARIS_PATH-1
                EDGE1=TOTAL_PATHS(IEDGE,IPATH)
                IF (EDGE1.NE.0) THEN
                    DO JEDGE=IEDGE+1,NMAX_ARIS_PATH
                        EDGE2=TOTAL_PATHS(JEDGE,IPATH)
                        IF (EDGE2.NE.0 .AND. EDGE1.EQ.EDGE2) THEN
                            TOTAL_PATHS(JEDGE,IPATH)=0
                        END IF
                    END DO
                END IF
            END DO
        END DO

C_____ Erases ZEROS

        DO IPATH=1,NMAX_PATHS
            IPOS=0
            DO IEDGE=1,NMAX_ARIS_PATH ! Loop from left to right
                IF (TOTAL_PATHS(IEDGE,IPATH).EQ.0) THEN ! ZERO found

                    DO JEDGE=IEDGE,NMAX_ARIS_PATH
                        IF (TOTAL_PATHS(JEDGE,IPATH).NE.0) THEN ! NON ZERO TO THE RIGHT
                            TOTAL_PATHS(IEDGE,IPATH)=TOTAL_PATHS(JEDGE,IPATH)
                            TOTAL_PATHS(JEDGE,IPATH)=0
                        END IF
                    END DO

                END IF
            END DO
        END DO

        RETURN
        END

        SUBROUTINE GET_LAST_POSITION_PATH
; (LASTPOSITION ,NMAX_ARIS_PATH ,PATH)

*****
*
* PURPOSE Gets last non-null position of a given path
*
* DESCRIPTION :
*
* EXTERNAL VARIABLES: ARRAYS
*
*   PATH                      Path under analysis
*
* EXTERNAL VARIABLES: SCALARS
*
*   LASTPOSITION              Last non-null position of a given path
*   NMAX_ARIS_PATH            Maximum number of edges defining a path
*
* INTERNAL VARIABLES: ARRAYS
*
* INTERNAL VARIABLES: SCALARS

```

```

*
*   IPOS                      Dummy counter
*
* FUNCTIONS AND SUBROUTINES REFERENCED
*
* HISTORY
*
*   AAR          06-2006      First coding
*
*****

      IMPLICIT NONE
      INTEGER*4 LASTPOSITION,NMAX_ARIS_PATH
      INTEGER*4 PATH(NMAX_ARIS_PATH)
      INTEGER*4 IPOS

      LASTPOSITION=NMAX_ARIS_PATH
      DO IPOS=1,NMAX_ARIS_PATH
         IF (PATH(IPOS).EQ.0) THEN
            LASTPOSITION=IPOS-1
            RETURN
         END IF
      END DO

      RETURN
      END

```

```

SUBROUTINE GENER_PATHS
; (ICRIT      , IDEBUG      , IOPTION_RND , ISEED_RND , IWEIGHTX
; , IWEIGHTY , NARIS_GEN    , NARIS_GEN_MAX, NARIS_TOT , NARIS_TOT_X
; , NARIS_TOT_Y, NTOSS      , NTOSSMAX    , FLAG_EDGES , IPROBABIL
; , ISEED     , NMAX_ARIS_PATH, NMAX_PATHS  , CONEC      , TOTAL_PATHS
; , NODES_X   , NODES_Y     , NODES_EDGE  , NUMPATHS_PERC
; , NUMPATHS_NONPERC      , SIZE_PERC_PATH
; , SIZE_NONPERC_PATH     , LONG_TOT_PERC , LONG_TOT_NONPERC
; , COORD        , NARIS_CRIT , FLAG_PERCOL)

```

```

IMPLICIT NONE
INTEGER*4 NTOSS, NTOSSMAX, ICRIT, NEDGE, NARIS_TOT, NARIS_GEN, ISEED_RND
;      , NARIS_GEN_MAX, RANDOM_GENERATION, IOPTION_RND, IDEBUG
;      , NARIS_TOT_X, NARIS_TOT_Y, IWEIGHTX, IWEIGHTY, NMAX_ARIS_PATH
;      , NMAX_PATHS, NODES_X, NODES_Y, NARIS_CRIT
INTEGER*4 FLAG_EDGES (NARIS_TOT), ISEED (13), IPROBABIL (NARIS_TOT)
;      , CONEC (6, NARIS_TOT), NODES_EDGE (NARIS_TOT, 2)
;      , TOTAL_PATHS (NMAX_ARIS_PATH, NMAX_PATHS)
;      , NUMPATHS_PERC (NARIS_TOT), NUMPATHS_NONPERC (NARIS_TOT)
;      , SIZE_PERC_PATH (NARIS_TOT, NMAX_PATHS)
;      , SIZE_NONPERC_PATH (NARIS_TOT, NMAX_PATHS)
;      , FLAG_PERCOL (NARIS_TOT, NMAX_PATHS)
REAL*8 COORD (NODES_X * NODES_Y, 2)
;      , LONG_TOT_PERC (NARIS_TOT, NMAX_PATHS)
;      , LONG_TOT_NONPERC (NARIS_TOT, NMAX_PATHS)
INTEGER*4 IEDGE, IPATH, DOES_IT_PERC, IS_FIRST_PERC_PATH

```

C _____ Writes main header related to random path

```

      IF (IDEBU.G.NE.0) WRITE (100,100)
100  FORMAT (//, ' GENERATION OF RANDOM EDGES:',/,
;      , ' ===== = ===== = =====',/)

```

C _____ Initializes auxiliar vector containing random seed

```

      DO IEDGE=1,13
         ISEED (IEDGE)=0
      END DO

```

C _____ Initializes some counter variables

```

      NARIS_GEN=0          ! Number of generated edges
      NTOSS=0             ! Number of tosses
      ICRIT=0             ! "Convergence" criteria
      IS_FIRST_PERC_PATH=0 ! Indicates that percolation has been achieved

```

C _____ Loop over maximum number of edges per path.

```

      DO IEDGE=1, NARIS_TOT

```

C _____ 1) Generates a non-repeated edge

```

      CALL GENERATE_EDGE
; (ICRIT      , IDEBUG      , IOPTION_RND , ISEED_RND , IWEIGHTX
; , IWEIGHTY , NARIS_GEN    , NARIS_GEN_MAX, NARIS_TOT , NARIS_TOT_X
; , NARIS_TOT_Y, NEDGE      , NTOSS      , NTOSSMAX    , FLAG_EDGES
; , IPROBABIL , ISEED)

```

C _____ 2) Checks "convergence"

```

      IF (ICRIT.NE.0) RETURN

```

C _____ 3) Checks connection with previously generated edges.
C _____ If it is connected to a prev. gen. edges (PREV_EDGE),
C _____ adds NEDGE to ALL paths where PREV_EDGE is present

```

      CALL ADD_EDGE2CONNECTED_PATHS
; (NARIS_TOT , NEDGE , NMAX_ARIS_PATH , NMAX_PATHS
; , FLAG_EDGES , CONEC , TOTAL_PATHS)

```

C_____ 4) Creates "long" paths from shorter ones.

```
CALL GENER_LONG_PATHS
; (NMAX_ARIS_PATH , NMAX_PATHS , TOTAL_PATHS)
```

C_____ 5) Checks all possible paths, distinguishing among perc /non perc.

```
DO IPATH=1,NMAX_PATHS

  DOES_IT_PERC=0
  CALL CHECK_PERCOLATION
; (NARIS_TOT , NMAX_ARIS_PATH , NODES_X , NODES_Y , DOES_IT_PERC
; , CONEC , COORD , NODES_EDGE , TOTAL_PATHS(1, IPATH))

C_____ Updates counters and calculates statistics

  IF (DOES_IT_PERC.EQ.1) THEN ! Percolating path

    FLAG_PERCOL(IEDGE, IPATH)=1 ! Flags percolating path

    NUMPATHS_PERC(IEDGE)=NUMPATHS_PERC(IEDGE)+1 ! Updates counter

    CALL GET_LAST_POSITION_PATH ! Calculates size
; (SIZE_PERC_PATH(IEDGE, IPATH) , NMAX_ARIS_PATH
; , TOTAL_PATHS(1, IPATH))

    CALL CALC_TOTAL LENGHT_PATH ! Calculates LENGHT
; (LONG_TOT_PERC(IEDGE, IPATH) , NARIS_TOT , NMAX_ARIS_PATH , NODES_X
; , NODES_Y , COORD , NODES_EDGE , TOTAL_PATHS(1, IPATH))

    IF (SIZE_PERC_PATH(IEDGE, IPATH).NE.0) CALL DEPICT_PATH
; (IPATH , 1 , 300 , NARIS_GEN , SIZE_PERC_PATH(IEDGE, IPATH)
; , NARIS_TOT , NMAX_ARIS_PATH , NODES_X , NODES_Y
; , LONG_TOT_PERC(IEDGE, IPATH) , COORD , NODES_EDGE
; , TOTAL_PATHS(1, IPATH))

    IF (IS_FIRST_PERC_PATH.EQ.0) THEN ! First percolating path
      IS_FIRST_PERC_PATH=1
      NARIS_CRIT=NARIS_GEN
    END IF
```

*** OJO, FALTA CALCULARSE LA LONGITUD MINIMA DEL CAMINO DE PERCOLACION

```
ELSE ! Non percolating path

  NUMPATHS_NONPERC(IEDGE)=NUMPATHS_NONPERC(IEDGE)+1 ! Updates counter
  FLAG_PERCOL(IEDGE, IPATH)=0 ! Flags percolating path

  CALL GET_LAST_POSITION_PATH ! Calculates size
; (SIZE_NONPERC_PATH(IEDGE, IPATH) , NMAX_ARIS_PATH
; , TOTAL_PATHS(1, IPATH))

  CALL CALC_TOTAL LENGHT_PATH ! Calculates LENGHT
; (LONG_TOT_NONPERC(IEDGE, IPATH) , NARIS_TOT , NMAX_ARIS_PATH , NODES_X
; , NODES_Y , COORD , NODES_EDGE , TOTAL_PATHS(1, IPATH))

  IF (SIZE_NONPERC_PATH(IEDGE, IPATH).NE.0) CALL DEPICT_PATH
; (IPATH , 0 , 400 , NARIS_GEN , SIZE_NONPERC_PATH(IEDGE, IPATH)
; , NARIS_TOT , NMAX_ARIS_PATH , NODES_X , NODES_Y
; , LONG_TOT_NONPERC(IEDGE, IPATH) , COORD , NODES_EDGE
; , TOTAL_PATHS(1, IPATH))

  END IF

END DO

END DO ! IEDGE=1,NARISTOT

RETURN
```

END

```

SUBROUTINE GENERATE_EDGE
; (ICRIT      , IDEBUG      , IOPTION_RND , ISEED_RND , IWEIGHTX
; , IWEIGHTY  , NARIS_GEN   , NARIS_GEN_MAX, NARIS_TOT , NARIS_TOT_X
; , NARIS_TOT_Y, NEDGE     , NTOSS      , NTOSSMAX  , FLAG_EDGES
; , IPROBABIL , ISEED)

*****
*
* PURPOSE Generates a NON-REPEATED random edge.
*
* DESCRIPTION : When IOPTION_RND=1 probability is uniform.
*               When IOPTION_RND=2 horizontal edges "deserve" IWEIGHT times more
*               importance than vertical ones. When IOPTION_RND=3, vertical edges
*               in the upper layer have more weight, which is intermediate for the
*               horizontal ones connecting the target edge and finally, the
*               smallest weights are assigned to the vertical edges in the lower
*               layer
*
* EXTERNAL VARIABLES: ARRAYS
*
* FLAG_EDGES           Vector of flags of prev. generated edges (NARIS_TOT)
* IPROBABIL            Auxiliar vector for defining probabilities (NARIS_TOT)
* ISEED                Auxiliar vector for generating probabilities
*
* EXTERNAL VARIABLES: SCALARS
*
* ICRIT                "Convergence" criteria (1=max. tosses; 2= max. edges)
* IDEBUG               Option for code debugging
* IOPTION_RND          Option for random generation
*                      1= Isotropy
*                      2= Anisotropy
*                      3= Programmer's defined weights
* ISEED_RND            Initial seed for random generation
* IWEIGHTX             Probability Weight for horizontal edges
* IWEIGHTY             Probability Weight for vertical edges
* NARIS_GEN            Number of generated edges
* NARIS_GEN_MAX        Max. number of edges to be generated
* NARIS_TOT            Total number of edges
* NARIS_TOT_X          Number of horizontal edges
* NARIS_TOT_Y          Number of vertical edges
* NEDGE               Edge number
* NTOSS               Current toss number
* NTOSSMAX            Maximum number of tosses
*
* INTERNAL VARIABLES: ARRAYS
*
* INTERNAL VARIABLES: SCALARS
*
* FUNCTIONS AND SUBROUTINES REFERENCED
*
* RANDOM_GENERATION    Generates a repeated/non-repeated random edge
*
* HISTORY
*
*   AAR                06-2006        First coding
*
*****

IMPLICIT NONE
INTEGER*4 NTOSS, NTOSSMAX, ICRIT, NEDGE, NARIS_TOT, NARIS_GEN, ISEED_RND
;
; , NARIS_GEN_MAX, RANDOM_GENERATION, IOPTION_RND, IDEBUG
;
; , NARIS_TOT_X, NARIS_TOT_Y, IWEIGHTX, IWEIGHTY
INTEGER*4 FLAG_EDGES(NARIS_TOT), ISEED(13), IPROBABIL(NARIS_TOT)

```



```

C_____ 1) Updates number of tosses

      NTOSS=NTOSS+1
      IF (NTOSS.EQ.NTOSSMAX) THEN
        ICRIT=1      ! Finishes due to max. number of tosses
        RETURN
      END IF

C_____ 2) Random selection of an edge (repeated or not)

1000 NEDGE=RANDOM_GENERATION
      ; (IOPTION_RND , ISEED_RND , NARIS_TOT , NARIS_TOT_X
      ; , NARIS_TOT_Y , IWEIGHTX , IWEIGHTY , IPROBABIL , ISEED)

C_____ 3) Just in case, checks that it does make sense

      IF (NEDGE.EQ.0 .OR. NEDGE.GT.NARIS_TOT) GOTO 1000

C_____ 4) Checks that edge was not previously generated,
C_____ in which case generates a new one

      IF (FLAG_EDGES(NEDGE).NE.0) THEN      ! Repeated
        GOTO 1000
      ELSE                                  ! "Flags" current edge
        FLAG_EDGES(NEDGE)=NEDGE
      END IF

C_____ 5) Writes current edge (debugging purposes)

      IF (IDEBUG.NE.0) WRITE(100,'(I10)') NEDGE

C_____ 6) Updates counter and checks for "convergence"

      NARIS_GEN=NARIS_GEN+1
      IF (NARIS_GEN.EQ.NARIS_GEN_MAX+1) THEN
        ICRIT=2      ! Finishes due to max. number of EDGES
        RETURN
      END IF

      RETURN
      END

```

```

SUBROUTINE INIT_INTEG
; (NARIS_GEN_MAX      ,NARIS_TOT      ,NARIS_TOT_X      ,NARIS_TOT_Y
; ,NARIS_X            ,NARIS_Y            ,NMAX_ARIS_PATH    ,NMAX_PATHS
; ,NODES_X            ,NODES_Y            ,PERCENT_ARIS_GEN)

*****
*
* PURPOSE Calculates some useful variables
*
* DESCRIPTION : Flow is assumed to be vertical (descendent). Vertical edges at
*               left & right boundaries are not considered.
*
* EXTERNAL VARIABLES: ARRAYS
*
* EXTERNAL VARIABLES: SCALARS
*
* NARIS_GEN_MAX      Maximum number of edges to be randomly generated
* NARIS_TOT           Total num. of edges
* NARIS_TOT_X         Total num. of horiz. edges
* NARIS_TOT_Y         Total num. of vert. edges
* NARIS_X             Number of edges in a row (given a fixed Y-coordinate)
* NARIS_Y             Number of edges in a column (given a fixed X-coordinate)
* NMAX_ARIS_PATH      Maximum number of edges per path
* NMAX_PATHS          Maximum number of paths
* NODES_X             Number of nodes X-direction
* NODES_Y             Number of nodes Y-direction
* PERCENT_ARIS_GEN    Maximum number (%) of edges to be gen.

* INTERNAL VARIABLES: ARRAYS
*
* INTERNAL VARIABLES: SCALARS
*
* FUNCTIONS AND SUBROUTINES REFERENCED
*
* HISTORY
*
*   AAR           06-2006      First coding
*
*****

IMPLICIT NONE
INTEGER*4 NARIS_X,NARIS_Y,NARIS_TOT_X,NARIS_TOT_Y,NARIS_GEN_MAX
;           ,NODES_X,NODES_Y,NARIS_TOT,NMAX_ARIS_PATH,NMAX_PATHS
REAL*8 PERCENT_ARIS_GEN

C_____ Calculates variables useful for dimensioning

NARIS_X=NODES_X-1      ! Number of edges in a row (given a fixed Y-coord.)
NARIS_Y=NODES_Y-1      ! Number of edges in a column (given a fixed X-coord.)

C_____ Percolation is assumed to be vertical. Vertical edges at left & right
C_____ boundaries are not considered.

NARIS_TOT=NARIS_X*NODES_Y+NARIS_Y*NODES_X-2*NARIS_Y      ! Total num. of edges
NARIS_TOT_X=NARIS_X*NODES_Y                                ! Total num. of horiz. edges
NARIS_TOT_Y=NARIS_Y*(NODES_X-2)                            ! Total num. of vert. edges
                                           ! Max. number of edges to be rnd. gen.
NARIS_GEN_MAX=(PERCENT_ARIS_GEN*NARIS_TOT/100.0D0)
NMAX_ARIS_PATH=NARIS_TOT
NMAX_PATHS=NARIS_TOT

C_____ Bye

RETURN
END

```

```

        SUBROUTINE INPUT_DATA
; (DELTAX , DELTAY , IDEBUG , IOPTION_RND , ISEED_RND
; , IWEIGHTX , IWEIGHTY , NARIS_PLT , NODES_X , NODES_Y
; , NTOSSMAX , PERCENT_ARIS_GEN)

*****
*
* PURPOSE Reads all input data
*
* DESCRIPTION : 1) Opens input data file
*               2) Reads variables for generating mesh
*               3) Reads dimensioning variables
*               4) Reads probability options
*               5) Reads variables controlling output
*               6) Closes input data file
*               7) Leaves subroutine
*
* EXTERNAL VARIABLES: ARRAYS
*
* EXTERNAL VARIABLES: SCALARS
*
* DELTAX           Edge size (X-direction)
* DELTAY           Edge size (Y-direction)
* IDEBUG           Option for code debugging
* IOPTION_RND      Option for random generation
*                  1= Isotropy
*                  2= Anisotropy
*                  3= Programmer's defined weights
* ISEED_RND        Initial seed for random generation
* IWEIGHTX         Probability Weight for horizontal edges
* IWEIGHTY         Probability Weight for vertical edges
* NARIS_PLT        Number of gen. edges after which we plot results
* NODES_X          Number of nodes X-direction
* NODES_Y          Number of nodes Y-direction
* NTOSSMAX         Maximum number of tosses to be gen.
* PERCENT_ARIS_GEN Maximum number (%) of edges to be gen.
*
* INTERNAL VARIABLES: ARRAYS
*
* INTERNAL VARIABLES: SCALARS
*
* FUNCTIONS AND SUBROUTINES REFERENCED
*
* HISTORY
*
*   AAR           06-2006       First coding
*
*****

        IMPLICIT NONE
        INTEGER*4 IDEBUG,NODES_X,NODES_Y,IOPTION_RND,ISEED_RND,IWEIGHTX
; , NTOSSMAX,NARIS_PLT,IWEIGHTY
        REAL*8 DELTAX,DELTAY,PERCENT_ARIS_GEN

C_____ Opens input data file

        OPEN (UNIT=1,FILE='INPUT_DATA.DAT',STATUS='OLD')

C_____ Mesh generation

        READ (1,'(F10.0)') DELTAX      ! Edge size (X-direction)
        READ (1,'(F10.0)') DELTAY      ! Edge size (Y-direction)
        READ (1,'(I10)')  NODES_X      ! Number of nodes X-direction
        READ (1,'(I10)')  NODES_Y      ! Number of nodes Y-direction

C_____ Probability options

```

```

READ (1, '(I10)') IOPTION_RND    ! Option for random generation
READ (1, '(I10)') ISEED_RND      ! Initial seed for random generation
READ (1, '(I10)') IWEIGHTX       ! HORIZ. Probability Weight (anisotropy)
READ (1, '(I10)') IWEIGHTY       ! VERT. Probability Weight (anisotropy)
IF (IWEIGHTX.NE.1.AND.IWEIGHTY.NE.1.AND.IOPTION_RND.EQ.2) THEN
  WRITE(*,*)
;   'ANISOTROPICAL PROBABILITY AND NONE OF THE WEIGHTS IS ONE.'
  WRITE(*,*) 'PLEASE, CHECK IT AND EXECUTE.'
  STOP ' CRITICAL ERROR. CHECK YOUR INPUT DATA.'
END IF

```

```

READ (1, '(I10)') NTOSSMAX        ! Maximum number of tosses to be gen.

```

C_____ Variables controlling output

```

READ (1, '(I10)') IDEBUG          ! Option for code debugging
READ (1, '(I10)') NARIS_PLT      ! Number of gen. edges after which we plot results
READ (1, '(F10.0)') PERCENT_ARIS_GEN ! Maximum number (%) of edges to be gen.

```

C_____ Closes input data file

```

CLOSE(1)

```

C_____ Bye !

```

RETURN
END

```

```

PROGRAM PERCOLATION

IMPLICIT NONE

INTEGER*4 IIMAX,IRMAX
PARAMETER(IIMAX=25000000,IRMAX=10000000)

C _____ Integers: input data and initialization of variables

INTEGER*4 IDEBUG,NMAX_PATHS,NMAX_ARIS_PATH,NODES_X,NODES_Y
; ,IOPTION_RND,ISEED_RND,IWEIGHTX,IWEIGHTY,NTOSSMAX
; ,NARIS_PLT,NARIS_X,NARIS_Y,NARIS_TOT_X,NARIS_TOT_Y
; ,NARIS_GEN_MAX,NARIS_TOT

C _____ Reals: input data and initialization of variables

REAL*8 DELTAX,DELTAY,PERCENT_ARIS_GEN

C _____ Integers: partition of integers

INTEGER*4 IOLD,IP_CONEC,IP_NODES_EDGE,IP_FLAG_EDGES
; ,IP_TOTAL_PATHS,IP_FLAG_PERCOL,IP_IPROBABIL,IP_NUMPATHS_PERC
; ,IP_NUMPATHS_NONPERC,IP_SIZE_PERC,IP_SIZE_NONPERC,IEND

C _____ Integers: partition of reals

INTEGER*4 ROLD,IP_COORD,IP_LONG_TOT_PERC,IP_LONG_TOT_NONPERC
; ,IP_LONG_MIN_PERC,REND

CHARACTER*14 TIMECPU

C _____ DECLARATION OF MATRICES

INTEGER*4 IRIP_INT,IRIP_REAL
INTEGER*4 AINTEG(IIMAX)
REAL*8 AREAL(IRMAX)

C _____ Reads input data

CALL INPUT_DATA
; (DELTAX ,DELTAY ,IDEBUG ,IOPTION_RND ,ISEED_RND
; ,IWEIGHTX ,IWEIGHTY ,NARIS_PLT ,NODES_X ,NODES_Y
; ,NTOSSMAX ,PERCENT_ARIS_GEN)

C _____ If so desired, writes input data

IF (IDEBUG.NE.0) CALL WRITE_INPUT_DATA
; (DELTAX ,DELTAY ,IDEBUG ,IOPTION_RND ,ISEED_RND
; ,IWEIGHTX ,IWEIGHTY ,NARIS_PLT ,NODES_X ,NODES_Y
; ,NTOSSMAX ,PERCENT_ARIS_GEN)

C _____ Calculates some useful variables

CALL INIT_INTEG
; (NARIS_GEN_MAX ,NARIS_TOT ,NARIS_TOT_X ,NARIS_TOT_Y
; ,NARIS_X ,NARIS_Y ,NMAX_ARIS_PATH ,NMAX_PATHS
; ,NODES_X ,NODES_Y ,PERCENT_ARIS_GEN)

C _____ If so desired, writes useful variables

IF (IDEBUG.NE.0) CALL WRITE_INIT_INTEG
; (NARIS_GEN_MAX ,NARIS_TOT ,NARIS_TOT_X ,NARIS_TOT_Y
; ,NARIS_X ,NARIS_Y)

C _____ Memo partition / pointer definition

C _____ INTEGERS

IOLD=1
IP_CONEC=IOLD ! Connectivity matrix. (6,NARIS_TOT)
IP_NODES_EDGE=IP_CONEC+6*NARIS_TOT ! Matrix of nodes of a given edge (2,NARIS_TOT)
IP_FLAG_EDGES=IP_NODES_EDGE+2*NARIS_TOT ! Flag for previously gen. edges (NARIS_TOT)
IP_TOTAL_PATHS=IP_FLAG_EDGES+NARIS_TOT ! Matrix containing paths (NMAX_ARIS_PATH,NMAX_PATHS)
IP_FLAG_PERCOL=IP_TOTAL_PATHS+NMAX_ARIS_PATH*NMAX_PATHS ! Flag for percol. paths (NMAX_PATHS)
IP_IPROBABIL=IP_FLAG_PERCOL+NMAX_PATHS*NARIS_TOT ! Auxiliar vector for generating probs. (NARIS_TOT)
IP_NUMPATHS_PERC=IP_IPROBABIL+NARIS_TOT ! Num. of perc. paths after gen. an edge (NARIS_TOT)
IP_NUMPATHS_NONPERC=IP_NUMPATHS_PERC+NARIS_TOT ! Num. of non perc. paths after gen. an edge (NARIS_TOT)
IP_SIZE_PERC=IP_NUMPATHS_NONPERC+NARIS_TOT ! Size of perc. paths after gen. an edge (NMAX_PATHS,NARIS_TOT)
IP_SIZE_NONPERC=IP_SIZE_PERC+NMAX_PATHS*NARIS_TOT ! Size of non perc. paths after gen. an edge (NMAX_PATHS,NARIS_TOT)
IEND=IP_SIZE_NONPERC+NMAX_PATHS*NARIS_TOT ! Final position of integers

ROLD=1
IP_COORD=ROLD ! Nodal coordinates (NODES_X*NODES_Y,2)
IP_LONG_TOT_PERC=IP_COORD+NODES_X*NODES_Y*2 ! Lenght of perc. paths (NMAX_PATHS,NARIS_TOT)
IP_LONG_TOT_NONPERC=IP_LONG_TOT_PERC+NMAX_PATHS*NARIS_TOT ! Lenght of non perc. paths (NMAX_PATHS,NARIS_TOT)
IP_LONG_MIN_PERC=IP_LONG_TOT_NONPERC+NMAX_PATHS*NARIS_TOT ! Min. lenght of non perc. paths (NMAX_PATHS,NARIS_TOT)
REND=IP_LONG_MIN_PERC+NMAX_PATHS*NARIS_TOT ! Final position of reals

C _____ Under demand, writes memo partition

IF (IDEBUG.NE.0) CALL WRITE_PART
; (ROLD ,IP_COORD ,IP_LONG_TOT_PERC ,IP_LONG_TOT_NONPERC
; ,IP_LONG_MIN_PERC ,REND ,IOLD
; ,IP_CONEC ,IP_NODES_EDGE ,IP_FLAG_EDGES
; ,IP_TOTAL_PATHS ,IP_FLAG_PERCOL ,IP_IPROBABIL
; ,IP_NUMPATHS_PERC ,IP_NUMPATHS_NONPERC ,IP_SIZE_PERC
; ,IP_SIZE_NONPERC ,IEND)

```

C_____ Physical definition of memory space

```
IF (IEND.GT.IIMAX) THEN
  WRITE(*,*) 'NOT ENOUGH MEMORY FOR INTEGERS.'
  WRITE(*,*) 'IIMAX (MAX. IN CODE) IS',IIMAX
  WRITE(*,*) 'I NEED A MINIMUM OF',IEND
  STOP ' FORCED STOP. CORRECT PERCOL.FOR AND RECOMPILE'
END IF

IF (REND.GT.IRMX) THEN
  WRITE(*,*) 'NOT ENOUGH MEMORY FOR REALS.'
  WRITE(*,*) 'IRMX (MAX. IN CODE) IS',IRMX
  WRITE(*,*) 'I NEED A MINIMUM OF',REND
  STOP ' FORCED STOP. CORRECT PERCOL.FOR AND RECOMPILE'
END IF
```

C_____ Initializes memo arrays

```
CALL ZERO_ARRAY_I (AINTG,IEND)
CALL ZERO_ARRAY_R (AREAL,REND)
```

C_____ Calls main programm

```
CALL PRINCIPAL
; (IDEBUG      ,NMAX_PATHS  ,NMAX_ARIS_PATH  ,NODES_X  ,NODES_Y
; ,IOPTION_RND ,ISEED_RND   ,IWEIGHTX    ,IWEIGHTY  ,NTOSSMAX
; ,NARIS_PLT   ,NARIS_X     ,NARIS_Y     ,NARIS_TOT_X
; ,NARIS_TOT_Y ,NARIS_GEN_MAX,NARIS_TOT    ,DELTAX   ,DELTAY
; ,PERCENT_ARIS_GEN      ,AINTG(IP_CONEC),AINTG(IP_NODES_EDGE)
; ,AINTG(IP_FLAG_EDGES)   ,AINTG(IP_TOTAL_PATHS)
; ,AINTG(IP_FLAG_PERCOL)  ,AINTG(IP_IPROBABIL)
; ,AINTG(IP_NUMPATHS_PERC),AINTG(IP_NUMPATHS_NONPERC)
; ,AINTG(IP_SIZE_PERC)    ,AINTG(IP_SIZE_NONPERC)
; ,AREAL(IP_COORD)        ,AREAL(IP_LONG_TOT_PERC)
; ,AREAL(IP_LONG_TOT_NONPERC),AREAL(IP_LONG_MIN_PERC) )

CALL CPUTIME (100,1,TIMECPU)

CLOSE(100)      ! Closes debug file

STOP ' Finished. Did God play dice?'

END
```

```

SUBROUTINE PRINCIPAL
; (IDEBUG      ,NMAX_PATHS      ,NMAX_ARIS_PATH      ,NODES_X      ,NODES_Y
; ,IOPTION_RND ,ISEED_RND      ,IWEIGHTX          ,IWEIGHTY      ,NTOSSMAX
; ,NARIS_PLT   ,NARIS_X         ,NARIS_Y            ,NARIS_TOT_X
; ,NARIS_TOT_Y ,NARIS_GEN_MAX   ,NARIS_TOT         ,DELTAX        ,DELTAY
; ,PERCENT_ARIS_GEN          ,CONEC                ,NODES_EDGE
; ,FLAG_EDGES ,TOTAL_PATHS     ,FLAG_PERCOL        ,IPROBABIL
; ,NUMPATHS_PERC            ,NUMPATHS_NONPERC    ,SIZE_PERC_PATH
; ,SIZE_NONPERC_PATH        ,COORD                ,LONG_TOT_PERC
; ,LONG_TOT_NONPERC         ,LONG_MIN_PERC)

```

IMPLICIT NONE

C_____ Integers: input data and initialization of variables

```

INTEGER*4 IDEBUG,NMAX_PATHS,NMAX_ARIS_PATH,NODES_X,NODES_Y
;      ,IOPTION_RND,ISEED_RND,IWEIGHTX,IWEIGHTY,NTOSSMAX
;      ,NARIS_PLT,NARIS_X,NARIS_Y,NARIS_TOT_X,NARIS_TOT_Y
;      ,NARIS_GEN_MAX,NARIS_TOT
;      ,NARIS_CRIT

```

C_____ Reals: input data and initialization of variables

```

REAL*8 DELTAX,DELTAY,PERCENT_ARIS_GEN

```

C_____ Integers arrays

```

INTEGER*4 CONEC(6,NARIS_TOT),NODES_EDGE(2,NARIS_TOT)
;      ,FLAG_EDGES(NARIS_TOT),FLAG_PERCOL(NARIS_TOT,NMAX_PATHS)
;      ,TOTAL_PATHS(NMAX_ARIS_PATH,NMAX_PATHS)
;      ,IPROBABIL(NARIS_TOT),NUMPATHS_PERC(NARIS_TOT)
;      ,NUMPATHS_NONPERC(NARIS_TOT)
;      ,SIZE_PERC_PATH(NARIS_TOT,NMAX_PATHS)
;      ,SIZE_NONPERC_PATH(NARIS_TOT,NMAX_PATHS)

```

C_____ Real arrays

```

REAL*8 COORD(NODES_X*NODES_Y,2)
;      ,LONG_TOT_PERC(NMAX_PATHS,NARIS_TOT)
;      ,LONG_TOT_NONPERC(NMAX_PATHS,NARIS_TOT)
;      ,LONG_MIN_PERC(NMAX_PATHS,NARIS_TOT)

```

```

INTEGER*4 ICRIT,NARIS_GEN,NTOSS,ISEED(13)

```

C_____ Generation of connectivity

```

CALL CREA_CONEC
; (IDEBUG      ,NARIS_TOT      ,NARIS_TOT_X      ,NARIS_X      ,NODES_Y      ,CONEC)

```

C_____ Generation of nodal coordinates

```

CALL CREA_COORD
; (DELTAX      ,DELTAY      ,IDEBUG      ,NODES_X      ,NODES_Y      ,COORD)

```

C_____ Generation of nodes belonging to edges

```

CALL EDGE2NODE
; (IDEBUG      ,NARIS_TOT      ,NARIS_TOT_X      ,NARIS_X      ,NODES_X      ,NODES_Y
; ,CONEC      ,NODES_EDGE)

```

C_____ Generation of TOTAL paths. As maximum, one percolation path is found

```

OPEN (UNIT=200,FILE='OUTPUT.OUT',STATUS='UNKNOWN')

```

```
OPEN (UNIT=300, FILE='PERC_PATHS.BNA', STATUS='UNKNOWN')
OPEN (UNIT=400, FILE='NONPERC_PATHS.BNA', STATUS='UNKNOWN')
```

```
CALL GENER_PATHS
; (ICRIT      , IDEBUG      , IOPTION_RND , ISEED_RND , IWEIGHTX
; , IWEIGHTY  , NARIS_GEN   , NARIS_GEN_MAX, NARIS_TOT , NARIS_TOT_X
; , NARIS_TOT_Y      , NTOSS      , NTOSSMAX   , FLAG_EDGES
; , IPROBABIL , ISEED      , NMAX_ARIS_PATH, NMAX_PATHS , CONEC
; , TOTAL_PATHS, NODES_X   , NODES_Y      , NODES_EDGE
; , NUMPATHS_PERC      , NUMPATHS_NONPERC
; , SIZE_PERC_PATH     , SIZE_NONPERC_PATH
; , LONG_TOT_PERC      , LONG_TOT_NONPERC
; , COORD              , NARIS_CRIT
; , FLAG_PERCOL)
```

C_____ Depicts percolation path

```
CALL RESULTS
; (ICRIT      , NARIS_CRIT , NARIS_GEN   , NARIS_TOT , NMAX_ARIS_PATH
; , NMAX_PATHS      , NODES_X      , NODES_Y      , NUMPATHS_PERC
; , NUMPATHS_NONPERC , SIZE_PERC_PATH , SIZE_NONPERC_PATH
; , LONG_TOT_PERC    , LONG_TOT_NONPERC , FLAG_PERCOL)
```

C_____ Closes debug file

```
CLOSE (200)
CLOSE (300)
CLOSE (400)
RETURN
```

```
END
```



```

        INTEGER*4 FUNCTION RANDOM_GENERATION
; (IOPTION_RND      , ISEED_RND      , NARIS_TOT      , NARIS_TOT_X
; , NARIS_TOT_Y      , IWEIGHTX      , IWEIGHTY      , IPROBABIL      , ISEED)

```

```

*****

```

```

*

```

```

* PURPOSE Generates a random edge.

```

```

*

```

```

* DESCRIPTION : When IOPTION_RND=1 probability is uniform.

```

```

*           When IOPTION_RND=2 horizontal edges "deserve" IWEIGHT times more
*           importance than vertical ones. When IOPTION_RND=3, vertical edges
*           in the upper layer have more weight, which is intermediate for the
*           horizontal ones connecting the target edge and finally, the
*           smallest weights are assigned to the vertical edges in the lower
*           layer

```

```

* EXTERNAL VARIABLES: ARRAYS

```

```

*

```

```

*   IPROBABIL           Auxiliar vector for defining probabilities

```

```

*

```

```

* EXTERNAL VARIABLES: SCALARS

```

```

*

```

```

*   IOPTION_RND         Option for random generation

```

```

*                       1= Isotropy

```

```

*                       2= Anisotropy

```

```

*                       3= Programmer's defined weights

```

```

*   IWEIGHTX            Probability Weight for horizontal edges

```

```

*   IWEIGHTY            Probability Weight for vertical edges

```

```

*   NARIS_TOT           Total number of edges

```

```

*   NARIS_TOT_X         Number of horizontal edges

```

```

*   NARIS_TOT_Y         Number of vertical edges

```

```

*

```

```

* INTERNAL VARIABLES: ARRAYS

```

```

*

```

```

* INTERNAL VARIABLES: SCALARS

```

```

*

```

```

*   IARIS               Dummy counter of edges

```

```

*   IPROB_TOTAL         Maximum of iprobabil's components

```

```

*

```

```

* FUNCTIONS AND SUBROUTINES REFERENCED

```

```

*

```

```

*   GEN_PROB_AUX        Generates auxiliar vector for generating probs.

```

```

*   ACORNI              Generates uniform / random number

```

```

*

```

```

* HISTORY

```

```

*

```

```

*   AAR                06-2006        First coding

```

```

*

```

```

*****

```

```

        IMPLICIT NONE

```

```

        INTEGER*4 IOPTION_RND, NARIS_TOT, NARIS_TOT_X, NARIS_TOT_Y

```

```

; , ISEED_RND, IWEIGHTX, IWEIGHTY

```

```

        INTEGER*4 IPROBABIL(NARIS_TOT), ISEED(13)

```

```

        REAL*8 ACORNI

```

```

        INTEGER*4 IARIS, IDUMMY, IPROB_TOTAL, IPROB, JPROB, DISTIPROB, DISTJPROB

```

```

; , LEFT_INT, RIGHT_INT

```

```

        REAL*8 R_DUMMY

```

```

C_____ Changes the seed 1000 times

```

```

        ISEED(1)=ISEED_RND

```

```

        DO IDUMMY=1,1000

```

```

            R_DUMMY=ACORNI( ISEED)

```

```

        END DO

```

C_____ Generates edge according to probability option

```
      IF (IOPTION_RND.EQ.1) THEN                ! Uniform

          RANDOM_GENERATION=INT (ACORNI (ISEED) * (NARIS_TOT+1) )

      ELSE IF (IOPTION_RND.EQ.2) THEN            ! Anisotropy

          CALL GEN_PROB_AUX                      ! Generates auxiliar vector of probabilities
; (IOPTION_RND      ,IPROB_TOTAL      ,IWEIGHTX      ,IWEIGHTY      ,NARIS_TOT
; ,NARIS_TOT_X      ,NARIS_TOT_Y      ,IPROBABIL)

          IPROB_TOTAL=INT (ACORNI (ISEED) *IPROB_TOTAL)
          DO IARIS=1,NARIS_TOT-1
              IPROB=IPROBABIL (IARIS)
              JPROB=IPROBABIL (IARIS+1)
              IF (IPROB_TOTAL.GE.IPROB .AND.
;              IPROB_TOTAL.LE.JPROB) THEN        ! Interval found
                  DISTIPROB=IPROB_TOTAL-IPROB
                  DISTJPROB=JPROB-IPROB_TOTAL
                  LEFT_INT=IARIS
                  RIGHT_INT=IARIS+1
                  GOTO 1000
              END IF
          END DO

1000      RANDOM_GENERATION=RIGHT_INT
          IF (DISTIPROB.LE.DISTJPROB) RANDOM_GENERATION=LEFT_INT

      ELSE IF (IOPTION_RND.EQ.3) THEN            ! Programmer's defined weights

      END IF

      RETURN
      END
```



```
      SUBROUTINE GEN_PROB_AUX
; (IOPTION_RND      ,IPROB_TOTAL      ,IWEIGHTX      ,IWEIGHTY      ,NARIS_TOT
; ,NARIS_TOT_X      ,NARIS_TOT_Y      ,IPROBABIL)
```

*

* PURPOSE Generates auxiliar vector for defining random edges. It only makes sense
* for anisotropic or predefined probability definition (IOPTION_RND=2,3)

*

* DESCRIPTION : When IOPTION_RND=2 Horizontal edges "deserve" IWEIGHT times more
* importance than vertical ones. When IOPTION_RND=3, vertical edges
* in the upper layer have more weight, which is intermediate for the
* horizontal ones connecting the target edge and finally, the
* smallest weights are assigned to the vertical edges in the lower
* layer

*

* EXTERNAL VARIABLES: ARRAYS

*

* IPROBABIL Auxiliary vector for defining probabilities

*

* EXTERNAL VARIABLES: SCALARS

*

* IOPTION_RND Option for random generation

*

1= Isotropy

*

2= Anisotropy

```

*                               3= Programmer's defined weights
*   IPROB_TOTAL                 Maximum of iprobabil's components
*   IWEIGHTX                    Probability Weight for horizontal edges
*   IWEIGHTY                    Probability Weight for vertical edges
*   NARIS_TOT                   Total number of edges
*   NARIS_TOT_X                 Number of horizontal edges
*   NARIS_TOT_Y                 Number of vertical edges
*
*   INTERNAL VARIABLES: ARRAYS
*
*   INTERNAL VARIABLES: SCALARS
*
*   IARIS                        Dummy counter of edges
*
*   FUNCTIONS AND SUBROUTINES REFERENCED
*
*   HISTORY
*
*       AAR           06-2006           First coding
*
*****

IMPLICIT NONE
INTEGER*4 NARIS_TOT, IWEIGHTX, IWEIGHTY, IPROB_TOTAL, NARIS_TOT_X
;           , NARIS_TOT_Y, IOPTION_RND
INTEGER*4 IPROBABIL(NARIS_TOT)
INTEGER*4 IARIS

C_____ Loop over edges, selecting horizontal / vertical and assigning
C_____ the fictitious number

      IPROB_TOTAL=0
      DO IARIS=1, NARIS_TOT

          IF (IWEIGHTX.GE.IWEIGHTY) THEN

              IF (IARIS.LE.NARIS_TOT_X) THEN ! Horizontal

                  IPROBABIL(IARIS)=IARIS*IWEIGHTX
              ELSE ! Vertical

                  IPROBABIL(IARIS)=NARIS_TOT_X*IWEIGHTX+IARIS-NARIS_TOT_X

              END IF ! Horizontal / Vertical

          ELSE ! IWEIGHTY.GE.IWEIGHTX

              IF (IARIS.LE.NARIS_TOT_X) THEN ! Horizontal

                  IPROBABIL(IARIS)=IARIS

              ELSE ! Vertical

                  IPROBABIL(IARIS)=NARIS_TOT_X+(IARIS-NARIS_TOT_X)*IWEIGHTY

              END IF

          END IF ! IWEIGHTX.GE.IWEIGHTY

          IF (IPROBABIL(IARIS).GE.IPROB_TOTAL)
;              IPROB_TOTAL=IPROBABIL(IARIS)

          END DO

      RETURN
      END

```

```

*****
*****
*****

```

```

      REAL*8 function acorni(ixv)

```

```

C-----
C
C Fortran implementation of ACORN random number generator of order less
C than or equal to 12 (higher orders can be obtained by increasing the
C parameter value MAXORD).
C
C
C NOTES: 1. The variable idum is a dummy variable. The common block
C          IACO is used to transfer data into the function.
C
C          2. Before the first call to ACORN the common block IACO must
C              be initialised by the user, as follows. The values of
C              variables in the common block must not subsequently be
C              changed by the user.
C
C              KORDEI - order of generator required ( must be =< MAXORD)
C
C              MAXINT - modulus for generator, must be chosen small
C                      enough that 2*MAXINT does not overflow
C
C              ixv(1) - seed for random number generator
C                      require 0 < ixv(1) < MAXINT
C
C              (ixv(I+1),I=1,KORDEI)
C                  - KORDEI initial values for generator
C                  require 0 =< ixv(I+1) < MAXINT
C
C          3. After initialisation, each call to ACORN generates a single
C              random number between 0 and 1.
C
C          4. An example of suitable values for parameters is
C
C              KORDEI   = 10
C              MAXINT   = 2**30
C              ixv(1)   = an odd integer in the (approximate) range
C                      (0.001 * MAXINT) to (0.999 * MAXINT)
C              ixv(I+1) = 0, I=1,KORDEI
C
C
C
C Author: R.S.Wikramaratna,                      Date: October 1990
C-----

```

```

C_____ Declaration of variables

```

```

      IMPLICIT NONE
      INTEGER*4 KORDEI,MAXINT,MAXOP1,IXV(13)
      INTEGER*4 I

```

```

      KORDEI=12
      MAXOP1=13
      MAXINT=2**30

```

```

      DO I=1,kordei
          IXV(I+1)=(IXV(I+1)+IXV(I))
          IF(IXV(I+1).GE.maxint) IXV(I+1)=IXV(I+1)-maxint
      END DO
      ACORNI=DBLE(IXV(kordei+1))/maxint
      RETURN
      END

```

```

SUBROUTINE RESULTS
; (ICRIT , NARIS_CRIT , NARIS_GEN , NARIS_TOT , NMAXARIS_PATH
; , NMAX_PATHS , NODES_X , NODES_Y , NUMPATHS_PERC
; , NUMPATHS_NONPERC , SIZE_PERC_PATH , SIZE_NONPERC_PATH
; , LONG_TOT_PERC , LONG_TOT_NONPERC , FLAG_PERCOL)

IMPLICIT NONE
INTEGER*4 ICRIT,NARIS_CRIT,NARIS_TOT,NMAXARIS_PATH,NARIS_GEN
; , NMAX_PATHS,NODES_X,NODES_Y
INTEGER*4 NUMPATHS_PERC(NARIS_TOT),NUMPATHS_NONPERC(NARIS_TOT)
; , SIZE_PERC_PATH(NARIS_TOT,NMAX_PATHS)
; , SIZE_NONPERC_PATH(NARIS_TOT,NMAX_PATHS)
; , FLAG_PERCOL(NARIS_TOT,NMAX_PATHS)
REAL*8 LONG_TOT_PERC(NMAX_PATHS,NARIS_TOT)
; , LONG_TOT_NONPERC(NMAX_PATHS,NARIS_TOT)

INTEGER*4 IEDGE,AUX,IPATH
REAL*8 ALFACRIT

```

C_____ Writes "convergence" reason

```

IF (ICRIT.EQ.1) THEN
WRITE(200,100)
100 FORMAT(//,' Very CPU consuming. "Convergence" achieved.',
; , ' Increase NTOSSMAX.')
ELSE IF (ICRIT.EQ.2) THEN
WRITE(200,200)
200 FORMAT(//,' "Convergence" achieved succesfully.')
END IF

```

C_____ Writes ALFACRIT

```

ALFACRIT=DFLOAT(NARIS_CRIT)/DFLOAT(NARIS_TOT)*100.0D0
WRITE(200,300) ALFACRIT
300 FORMAT(//,' AlfaCritical=',F5.2)

OPEN(UNIT=666,FILE='ALFACRIT.OUT',STATUS='UNKNOWN')
WRITE(666,*) NODES_X,NODES_Y,ALFACRIT
CLOSE(666)

```

C_____ Writes history of number of percolating paths vs gen. edges

```

WRITE(200,400)
400 FORMAT(//,' Number of percolating paths vs number of generated'
; , ' edges',/
; , ' ===== == ===== ===== == ===== == ====='
; , ' =====',/)

DO IEDGE=1,NARIS_GEN
WRITE(200,'(2I10)') IEDGE,NUMPATHS_PERC(IEEDGE)
END DO

```

C_____ Writes history of number of non percolating paths vs gen. edges

```

WRITE(200,500)
500 FORMAT(//,' Number of non percolating paths vs number of'
; , ' generated edges',/
; , ' ===== == == ===== ===== == ===== =='
; , ' ===== =====',/)

DO IEDGE=1,NARIS_GEN
AUX=0
DO IPATH=1,NMAX_PATHS
IF (SIZE_NONPERC_PATH(IEEDGE,IPATH).NE.0) AUX=AUX+1
END DO
WRITE(200,'(2I10)') IEDGE,AUX

```

END DO

C _____ Writes a file with lenght and size of perc. / non. perc paths
C _____ vs number of gen. edges

```
OPEN (UNIT=666, FILE='PERCS.OUT', STATUS='UNKNOWN')
DO IEDGE=1, NARIS_TOT
  DO IPATH=1, NMAX_PATHS
    IF (FLAG_PERCOL(IEDGE, IPATH).EQ.1) THEN
      WRITE(666, '(3I10, F10.3, I10, F10.3)') NODES_X, NODES_Y, IPATH
;      , DFLOAT(IEDGE)/DFLOAT(NARIS_TOT)*100.0D0
;      , SIZE_PERC_PATH(IEDGE, IPATH)
;      , LONG_TOT_PERC(IEDGE, IPATH)
    END IF
  END DO
END DO
CLOSE(666)

OPEN (UNIT=666, FILE='NONPERCS.OUT', STATUS='UNKNOWN')
DO IEDGE=1, NARIS_TOT
  DO IPATH=1, NMAX_PATHS
    IF (FLAG_PERCOL(IEDGE, IPATH).EQ.0 .AND.
;      SIZE_NONPERC_PATH(IEDGE, IPATH).NE.0) THEN
      WRITE(666, '(3I10, F10.3, I10, F10.3)') NODES_X, NODES_Y, IPATH
;      , DFLOAT(IEDGE)/DFLOAT(NARIS_TOT)*100.0D0
;      , SIZE_NONPERC_PATH(IEDGE, IPATH)
;      , LONG_TOT_NONPERC(IEDGE, IPATH)
    END IF
  END DO
END DO
CLOSE(666)

RETURN
END
```

```
SUBROUTINE ZERO_ARRAY_I (ARRAY,IDIM)
```

```
*****
*
* PURPOSE Initializes an array of integers
*
* DESCRIPTION :
*
* EXTERNAL VARIABLES: ARRAYS
*
*   ARRAY           Array (matrix / vector) to be initialized
*
* EXTERNAL VARIABLES: SCALARS
*
*   IDIM            Array dimension
*
* INTERNAL VARIABLES: ARRAYS
*
* INTERNAL VARIABLES: SCALARS
*
*   IPOS            Dummy counter
*
* FUNCTIONS AND SUBROUTINES REFERENCED
*
* HISTORY
*
*   AAR            06-2006      First coding
*
*****
```

```
IMPLICIT NONE
INTEGER*4 IDIM
INTEGER*4 ARRAY(IDIM)
INTEGER*4 IPOS
```

```
DO IPOS=1,IDIM
  ARRAY(IPOS)=0
END DO
```

```
RETURN
END
```

```
*****
*****
*****
```

```
SUBROUTINE ZERO_ARRAY_R (ARRAY,IDIM)
```

```
*****
*
* PURPOSE Initializes an array of reals
*
* DESCRIPTION :
*
* EXTERNAL VARIABLES: ARRAYS
*
*   ARRAY           Array (matrix / vector) to be initialized
*
* EXTERNAL VARIABLES: SCALARS
*
*   IDIM            Array dimension
*
* INTERNAL VARIABLES: ARRAYS
*
```

```

* INTERNAL VARIABLES: SCALARS
*
*   IPOS                      Dummy counter
*
* FUNCTIONS AND SUBROUTINES REFERENCED
*
* HISTORY
*
*   AAR          06-2006      First coding
*
*****

      IMPLICIT NONE
      INTEGER*4 IDIM
      REAL*8 ARRAY(IDIM)
      INTEGER*4 IPOS

      DO IPOS=1, IDIM
        ARRAY(IPOS)=0.0D0
      END DO

      RETURN
      END

*****
****  ONLY MACHINE DEPENDENT ROUTINES ARE INCLUDED IN THIS FILE
*****

      SUBROUTINE FECHA (MAINF,TITEL)

*****
***   IT WRITES MODEL'S DATE AND TITLE
*****

      CHARACTER*24 DATIM,FDATE,TITEL*80

      DATIM=FDATE()
      WRITE(MAINF,10) DATIM(1:3)//' ' //DATIM(9:10)//
;          '-'//DATIM(5:7)//'-'//DATIM(21:24)//' ' //DATIM(12:19)
10  FORMAT(//,'D A T E .....: ',A25,/)

      RETURN
      END

*****
*****

      SUBROUTINE CPUTIME (MAINF,ILI,TT)

*****
*** IT WRITES THE TOTAL CPUTIME SINCE THE BEGINNING OF THE PROGRAM
*****

      CHARACTER*14 TT
      REAL*4 DTIME,TARRAY(2),WW,XX

      WW=DTIME(TARRAY)
      XX=TARRAY(1)
      IHORAS=INT(XX/3600)
      XX=XX-IHORAS*3600.
      IMINUT=INT(XX/60)
      XX=XX-IMINUT*60.
      ISEGU=INT(XX+0.5)
      IF (IHORAS.NE.0) THEN

```



```
        WRITE (MAINF,1)  IHORAS,IMINUT,ISEGU
ELSE
        WRITE (MAINF,2)  IMINUT,ISEGU
END IF

1  FORMAT (//,2X,'TOTAL CPUTIME :',I3,' HOURS',I3,
;    ' MINUTES',I3,' SECONDS')
2  FORMAT (//,2X,'TOTAL CPUTIME :',I3,' MINUTES',I3,' SECONDS')

RETURN
END
```

```

SUBROUTINE WRITE_INIT_INTEG
; (NARIS_GEN_MAX ,NARIS_TOT ,NARIS_TOT_X ,NARIS_TOT_Y
; ,NARIS_X ,NARIS_Y)

*****
*
* PURPOSE Writes useful variables calculated by INIT_INTEG
*
* DESCRIPTION : Flow is assumed to be vertical (descendent). Vertical edges at
*               left & right boundaries are not considered.
*
* EXTERNAL VARIABLES: ARRAYS
*
* EXTERNAL VARIABLES: SCALARS
*
* NARIS_GEN_MAX      Maximum number of edges to be randomly generated
* NARIS_TOT           Total num. of edges
* NARIS_TOT_X         Total num. of horiz. edges
* NARIS_TOT_Y         Total num. of vert. edges
* NARIS_X             Number of edges in a row (given a fixed Y-coordinate)
* NARIS_Y             Number of edges in a column (given a fixed X-coordinate)
*
* INTERNAL VARIABLES: ARRAYS
*
* INTERNAL VARIABLES: SCALARS
*
* FUNCTIONS AND SUBROUTINES REFERENCED
*
* HISTORY
*
*   AAR           06-2006      First coding
*
*****

IMPLICIT NONE
INTEGER*4  NARIS_TOT,NARIS_TOT_X,NARIS_TOT_Y,NARIS_X,NARIS_Y
;          ,NARIS_GEN_MAX

WRITE (100,*) 'Total number of edges =',NARIS_TOT
WRITE (100,*) 'Total number of horiz. edges =',NARIS_TOT_X
WRITE (100,*) 'Total number of vert. edges =',NARIS_TOT_Y
WRITE (100,*) 'Edges in a horiz. row =',NARIS_X
WRITE (100,*) 'Edges in a vert. column =',NARIS_Y
WRITE (100,*) 'Max. num. of edges to be gen. =',NARIS_GEN_MAX

RETURN
END

```

```

        SUBROUTINE WRITE_INPUT_DATA
; (DELTAX      ,DELTAY      ,IDEBUG      ,IOPTION_RND      ,ISEED
; ,IWEIGHTX    ,IWEIGHTY    ,NARIS_PLT   ,NODES_X          ,NODES_Y
; ,NTOSSMAX    ,PERCENT_ARIS_GEN)

*****
*
* PURPOSE Writes all input data
*
* DESCRIPTION : 1) Opens debug file
*                2) Writes variables for generating mesh
*                3) Writes dimensioning variables
*                4) Writes probability options
*                5) Writes variables controlling output
*                6) Leaves subroutine
*
* EXTERNAL VARIABLES: ARRAYS
*
* EXTERNAL VARIABLES: SCALARS
*
* DELTAX                Edge size (X-direction)
* DELTAY                Edge size (Y-direction)
* IDEBUG                Option for code debugging
* IOPTION_RND           Option for random generation
*                        1= Isotropy
*                        2= Anisotropy
*                        3= Programmer's defined weights
* ISEED                 Initial seed for random generation
* IWEIGHTX              Probability Weight for horizontal edges
* IWEIGHTY              Probability Weight for vertical edges
* NARIS_PLT             Number of gen. edges after which we plot results
* NODES_X               Number of nodes X-direction
* NODES_Y               Number of nodes Y-direction
* NTOSSMAX              Maximum number of tosses to be gen.
* PERCENT_ARIS_GEN      Maximum number (%) of edges to be gen.
*
* INTERNAL VARIABLES: ARRAYS
*
* INTERNAL VARIABLES: SCALARS
*
* FUNCTIONS AND SUBROUTINES REFERENCED
*
* HISTORY
*
*      AAR            06-2006      First coding
*
*****

        IMPLICIT NONE
        INTEGER*4 IDEBUG,NODES_X,NODES_Y,IOPTION_RND,ISEED,IWEIGHTX
;                ,NTOSSMAX,NARIS_PLT,IWEIGHTY
        REAL*8 DELTAX,DELTAY,PERCENT_ARIS_GEN

C_____ Opens debug file

        OPEN (UNIT=100,FILE='DEBUG.DBG',STATUS='UNKNOWN')
        CALL FECHA (100,'')

C_____ Mesh generation

        WRITE (100,*) 'DeltaX =',DELTAX
        WRITE (100,*) 'DeltaY =',DELTAY
        WRITE (100,*) 'NodesX =',NODES_X

```

```
WRITE (100,*) 'NodesY =',NODES_Y
```

C_____ Probability options

```
WRITE (100,*) 'Option for assigning probabil. =',IOPTION_RND
```

```
WRITE (100,*) 'Seed for random number generation =',ISEED
```

```
WRITE (100,*) 'Weight for horiz. edges (Option=2) =',IWEIGHTX
```

```
WRITE (100,*) 'Weight for vert. edges (Option=2) =',IWEIGHTY
```

```
WRITE (100,*) 'Maximum number of tosses =',NTOSSMAX
```

C_____ Variables controlling output

```
WRITE (100,*) 'Debug Option =',IDEBUG
```

```
WRITE (100,*) 'Plot after =',NARIS_PLT,' edges'
```

```
WRITE (100,*) '% of edges to be gen. =',PERCENT_ARIS_GEN
```

C_____ Bye !

```
RETURN
```

```
END
```

```

SUBROUTINE WRITE_PART
; (ROLD , IP_COORD , IP_LONG_TOT_PERC , IP_LONG_TOT_NONPERC
; , IP_LONG_MIN_PERC , REND , IOLD
; , IP_CONEC , IP_NODES_EDGE , IP_IFLAG_EDGES
; , IP_TOTAL_PATHS , IP_FLAG_PERCOL , IP_IPROBABIL
; , IP_NUMPATHS_PERC , IP_NUMPATHS_NONPERC , IP_SIZE_PERC
; , IP_SIZE_NONPERC , IEND)

```

```

*****
*
* PURPOSE Writes memo partition at DBG file
*
* DESCRIPTION : 1) Partition of array of integers
*               2) Partition of array of reals
*
* EXTERNAL VARIABLES: ARRAYS
*
* EXTERNAL VARIABLES: SCALARS
*
* Just pointers
*
* INTERNAL VARIABLES: ARRAYS
*
* INTERNAL VARIABLES: SCALARS
*
* FUNCTIONS AND SUBROUTINES REFERENCED
*
* HISTORY
*
* AAR      06-2006      First coding
*
*****

```

```

IMPLICIT NONE
INTEGER*4 ROLD, IP_COORD, IP_LONG_TOT_PERC, IP_LONG_TOT_NONPERC
; , IP_LONG_MIN_PERC, REND, IOLD, IP_CONEC, IP_NODES_EDGE
; , IP_IFLAG_EDGES, IP_TOTAL_PATHS, IP_FLAG_PERCOL
; , IP_IPROBABIL, IP_NUMPATHS_PERC, IP_NUMPATHS_NONPERC
; , IP_SIZE_PERC, IP_SIZE_NONPERC, IEND

```

C_____ Partition of integers

```

WRITE(100,100)
100 FORMAT(//, ' POINTERS TO ARRAYS OF INTEGERS', /
; , ' ===== == ===== == =====', //)

WRITE(100, ' (A16, I10) ')
; ' IOLD= ', IOLD
; , ' CONEC= ', IP_CONEC
; , ' NODES_EDGE= ', IP_NODES_EDGE
; , ' IFLAG_EDGES= ', IP_IFLAG_EDGES
; , ' TOTAL_PATHS= ', IP_TOTAL_PATHS
; , ' FLAG_PERCOL= ', IP_FLAG_PERCOL
; , ' IPROBABIL= ', IP_IPROBABIL
; , ' NUMPATHS_PERC= ', IP_NUMPATHS_PERC
; , ' NUMPATHS_NONPERC= ', IP_NUMPATHS_NONPERC
; , ' SIZE_PERC= ', IP_SIZE_PERC
; , ' SIZE_NONPERC= ', IP_SIZE_NONPERC
; , ' IEND= ', IEND

```

C_____ Partition of integers

```

WRITE(100,200)
200 FORMAT(//, ' POINTERS TO ARRAYS OF REALS', /

```

```

;          ' ===== == ===== == =====',//)

WRITE(100,' (A16,I10) ')
; 'ROLD=',          ROLD
; , 'COORD=',       IP_COORD
; , 'LONG_TOT_PERC=', IP_LONG_TOT_PERC
; , 'LONG_TOT_NONPERC=', IP_LONG_TOT_NONPERC
; , 'LONG_MIN_PERC=',  IP_LONG_MIN_PERC
; , 'REND=',          REND

RETURN
END

```

ANEJO 3. SIMULACIONES

ANEJO3. SIMULACIONES

A continuación se muestra una simulación de una malla (15x50) del ejemplo anisótropo. En el primer caso se observan los caminos no percolantes cuando un 30% de las aristas se han generado. Después se representa el momento en que se produce la percolación, al 53% de las aristas. Y finalmente, se indica el estado de la malla en el 80%.

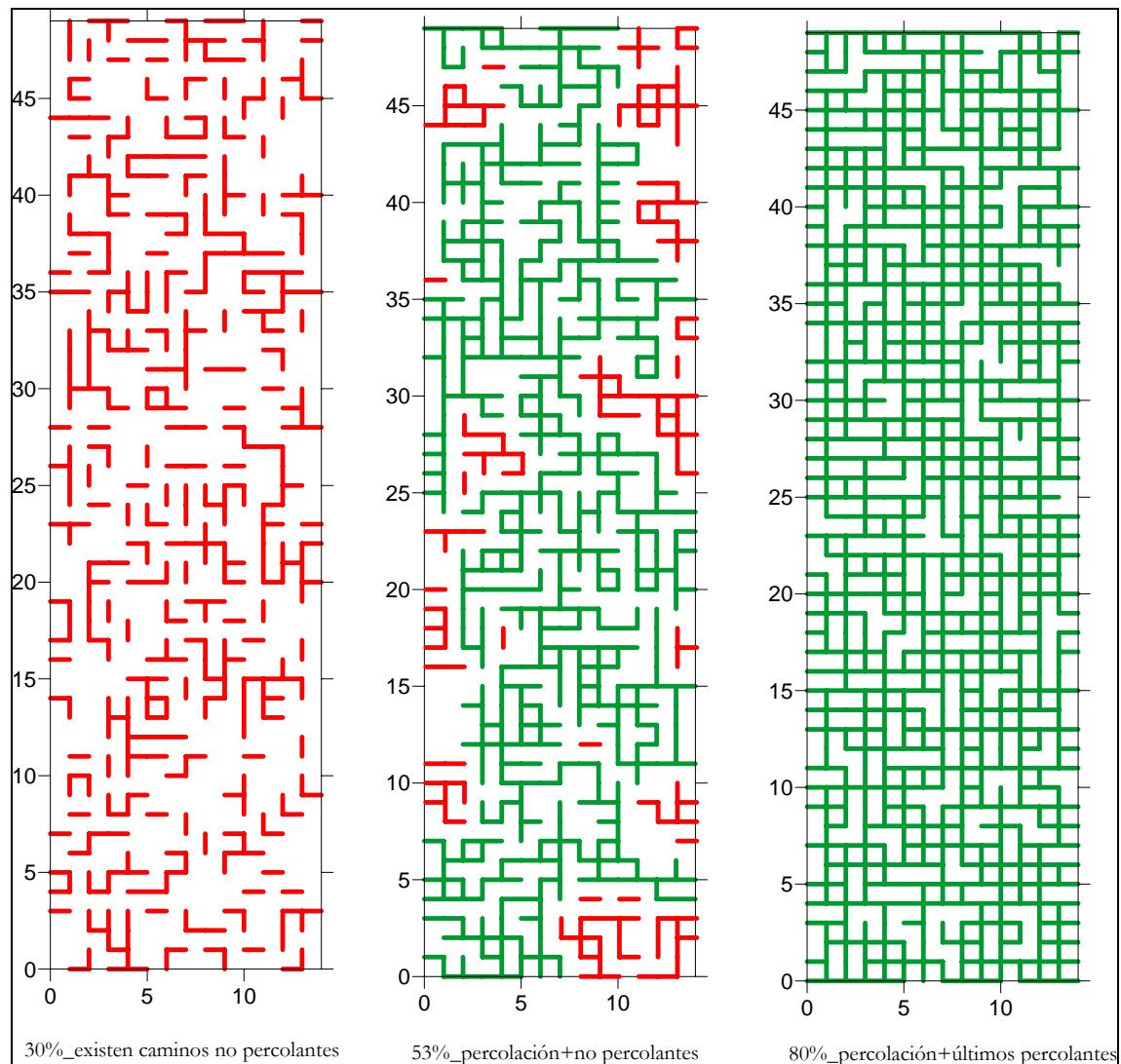


Figura 82. Simulación de una malla (15x50).

La siguiente simulación pertenece a una malla (40x15) en el caso isótropo:

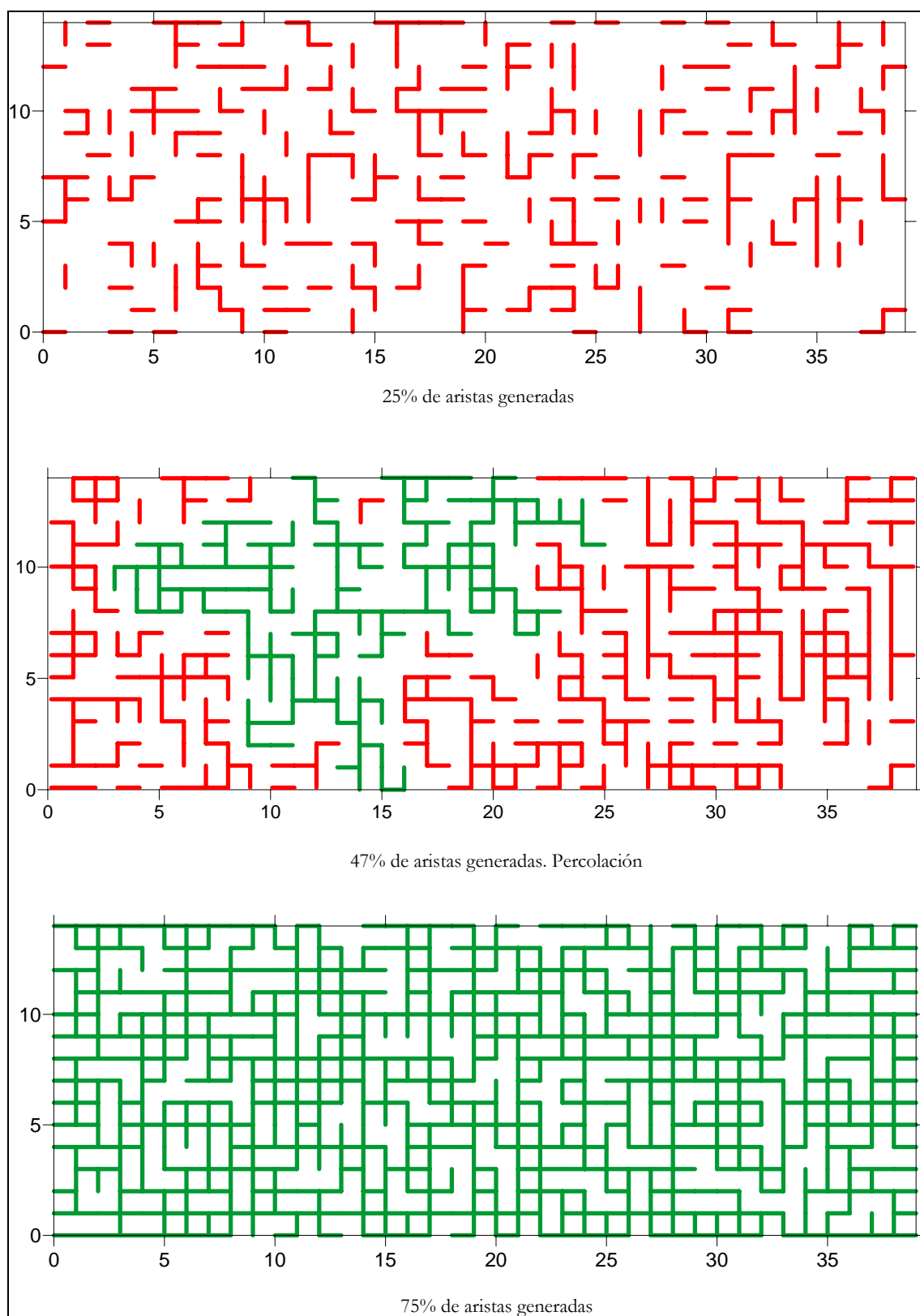


Figura 83. Simulación de una malla (40x15).

**ANEJO 4. VALORES DE LOS PARÁMETROS
PERCOLANTES EN EL ANÁLISIS MONTE CARLO**

ANEJO4. VALORES DE LOS PARÁMETROS PERCOLANTES EN EL ANÁLISIS MONTE CARLO

TIRADAS	ALFACRIT	LENGTH PERC	NONPERC-PATHS	TIRADAS	ALFACRI	LENGTH PERC	NONPERC-PATHS
1	38,37	72	52	26	40,3	86	49
2	45,29	161	27	27	46,12	80	27
3	44,32	78	27	28	44,04	103	38
4	36,57	57	52	29	40,44	77	40
5	46,54	166	24	30	40,72	97	50
6	42,38	48	38	31	40,03	64	50
7	48,75	145	25	32	45,01	78	31
8	39,2	52	53	33	43,07	104	32
9	46,95	41	28	34	40,44	103	42
10	36,84	54	58	35	41,97	89	37
11	48,48	128	27	36	45,01	133	35
12	41,69	75	39	37	47,23	134	21
13	45,98	145	24	38	44,32	119	36
14	47,92	119	24	39	43,35	136	33
15	40,58	68	55	40	37,95	46	48
16	41,55	70	43	41	45,43	101	33
17	47,09	165	24	42	46,54	127	31
18	45,98	66	38	43	37,95	67	57
19	40,17	93	45	44	41,41	103	49
20	43,07	53	40	45	39,06	96	45
21	41,55	149	33	46	42,24	108	33
22	48,75	181	18	47	44,04	69	43
23	46,95	41	32	48	44,04	118	36
24	38,92	70	54	49	46,54	107	27
25	48,2	75	19	50	45,29	130	33

Tabla 20. Valores de los parámetros percolantes para el caso anisótropo vertical.

TIRADAS	ALFACRIT	LENGTH PERC	NONPERC- PATHS	TIRADAS	ALFACRI	LENGTH PERC	NONPERC- PATHS
1	57,48	356	15	26	52,35	231	17
2	56,37	330	12	27	55,12	350	14
3	59,97	421	6	28	61,63	409	9
4	59,7	428	2	29	63,85	450	5
5	57,06	385	9	30	53,88	272	14
6	49,45	262	24	31	55,68	346	15
7	56,93	354	15	32	53,6	324	19
8	52,77	305	18	33	55,26	319	18
9	57,2	372	17	34	57,62	368	14
10	59,42	408	9	35	52,08	306	25
11	55,68	360	13	36	57,89	333	14
12	60,66	420	8	37	49,58	284	30
13	50,42	248	27	38	56,65	356	14
14	61,91	421	9	39	59,14	416	7
15	58,31	315	14	40	51,8	285	24
16	57,48	370	8	41	61,22	413	3
17	57,76	392	13	42	59,56	422	4
18	56,37	373	11	43	54,02	311	17
19	63,02	429	11	44	55,12	357	12
20	57,34	378	13	45	60,66	419	9
21	58,45	385	11	46	56,93	386	10
22	55,96	372	12	47	53,88	346	17
23	59,42	411	7	48	54,85	312	21
24	59,28	405	6	49	58,59	389	7
25	56,65	360	13	50	54,02	321	17

Tabla 21: Valores de los parámetros percolantes para el caso anisótropo horizontal.

ANEJO 5. ESTUDIO ESTADÍSTICO

ANEJO5. ESTUDIO ESTADÍSTICO

LENGTH versus NON_PERCPATHS

CASO ISÓTROPO

En la **Figura 84** se representa el diagrama de dispersión de las longitudes de caminos percolantes frente al número de caminos no percolantes. En ella podemos ya apreciar la estrecha relación lineal que guardan ambas variables. Asimismo aparece en la figura el coeficiente de correlación lineal al cuadrado entre ambas variables con un valor de 0,605 que confirma la relación lineal existente.

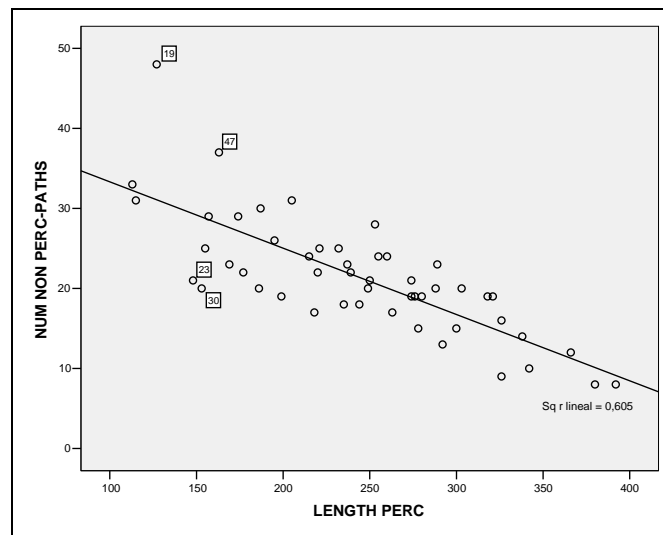


Figura 84. Diagrama de dispersión de *length* vs *nonperc_paths*.

En la **Tabla 22** podemos ver los estadísticos básicos de cada una de las variables.

	Media	Desviación típ.
NON PERC-PATHS	21,42	7,376
LENGTH PERC	243,54	69,273

Tabla 22. Estadísticos descriptivos.

En la **Tabla 23** aparece el coeficiente de correlación lineal con un valor de -0.775 muy próximo a -1 que indica fuerte relación lineal negativa, es decir, a mayor longitud de camino percolante menor es el número de caminos no percolantes. El error típico de estimación es de 4,684, valor que se tiene que comparar con la desviación típica de la variable dependiente *nonperc_paths* de 7,376.

Modelo	R	R cuadrado	R cuadrado corregida	Error típ. de la estimación
1	0,778(a)	0,605	0,597	4,684
a Variables predictoras: (Constante), LENGTH PERC				
b Variable dependiente: NUM NON PERC-PATHS				

Tabla 23. Resumen del modelo.

En la **Tabla 24** se representa el contraste ANOVA que tiene un objeto doble. En la última columna aparece una significación nula, por tanto podemos aceptar el modelo lineal que aparece en la Tabla 4 donde se estiman los coeficientes.

Modelo		Suma de cuadrados	gl	Media cuadrática	F	Sig.
1	Regresión	1613,211	1	1613,211	73,539	0,000(a)
	Residual	1052,969	48	21,937		
	Total	2666,180	49			

Tabla 24. ANOVA.

A partir de la información contenida en la **Tabla 25** concluimos que el modelo estimado es la recta:

$$41,592 - 0,083 \cdot \text{length_perc} = \text{PRE}_1$$

Modelo		Coeficientes no estandarizados		Coeficientes estandarizados	t	Sig.	Intervalo de confianza para B al 95%	
		B	Error típ.	Beta			Límite inferior	Límite superior
1	(Constante)	41,592	2,444		17,020	0,000	36,679	46,506
	LENGTH PERC	-0,083	0,010	-0,778	-8,575	0,000	-0,102	-0,063

Tabla 25. Coeficientes.

Análisis de residuos

En la **Figura 85** puede apreciarse el histograma de los residuos con la curva normal superpuesta así como en la **Figura 86** se observa el diagrama p-p en el que podemos ver como los datos se agrupan en torno a la bisectriz. Además la Tabla 5 confirma que podemos aceptar que los residuos siguen una distribución normal ya que la significación es de 0,765.

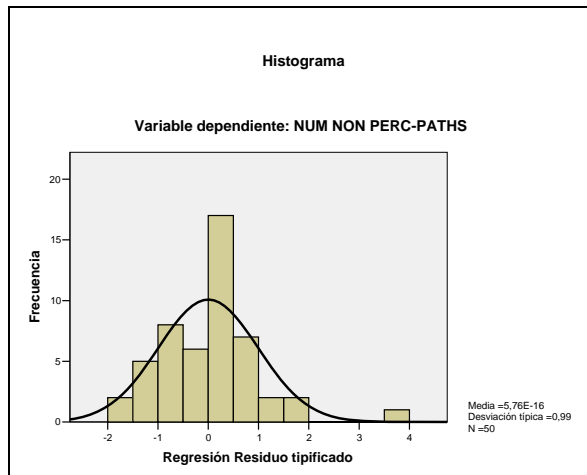


Figura 85. Histograma de los residuos.

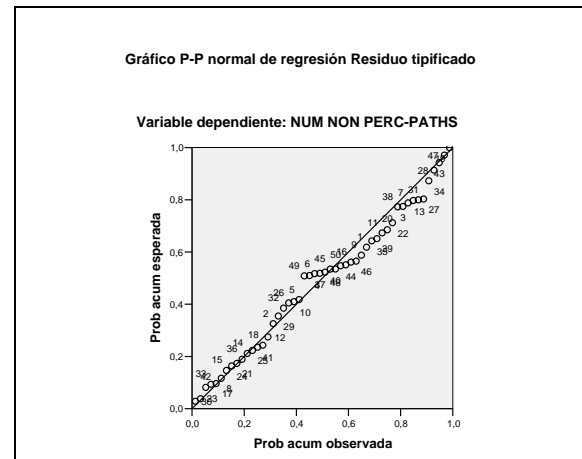


Figura 86. Diagrama p-p.

El contraste no paramétrico de Kolmogorov-Smirnov que se indica en la **Tabla 26** se desea contrastar que los residuos siguen una distribución normal. Como el nivel de significación obtenido es de 0,765, se puede aceptar que los residuos siguen una distribución normal, aunque el valor no sea muy cercano a la unidad.

		Unstandardized Residual
Parámetros normales	Media	0,0000000
	Desviación típica	4,63564116
Diferencias más extremas	Absoluta	0,094
	Positiva	0,094
	Negativa	-0,089
Z de Kolmogorov-Smirnov		0,667
Sig. asintót. (bilateral)		0,765

Tabla 26. Prueba de Kolmogorov-Smirnov para una muestra.

En la **Figura 87**, los datos se presentan en una pelota sin mostrar ningún patrón a excepción de la tirada 19. Es decir, podemos aceptar independencia en los errores cometidos en las observaciones, pero parece conveniente repetir el estudio prescindiendo del caso 19, tal como nos muestra la **Figura 88**, donde hemos representado los valores de influencia que pueden gobernar el modelo.

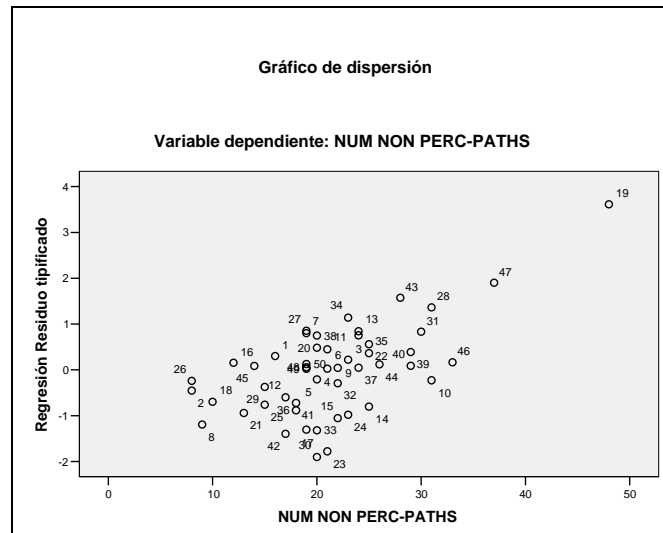


Figura 87. Gráfico de dispersión de residuo vs la variable dependiente.

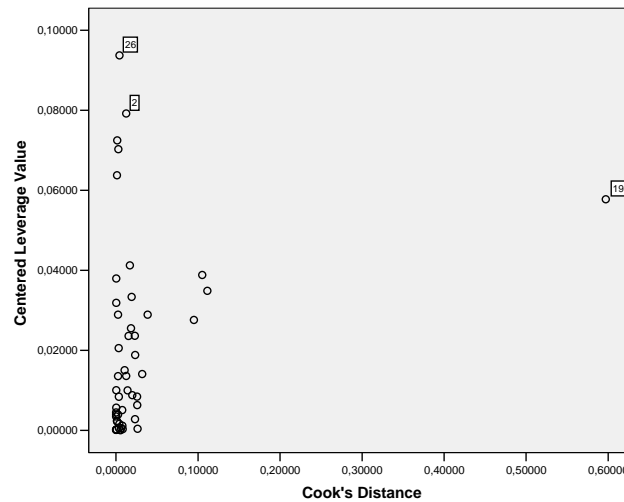
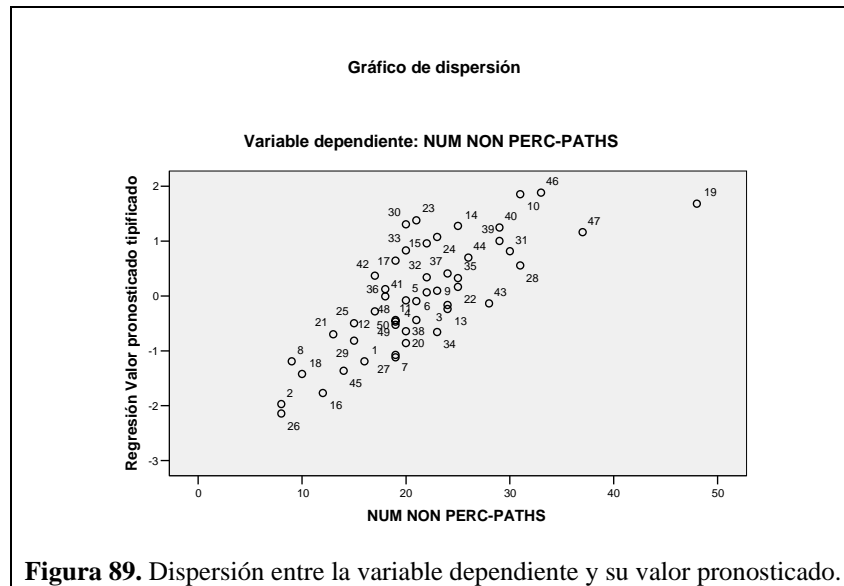


Figura 88. Distancia de Cook vs la variable pronostica.

La Figura 89 representa la variable dependiente frente a la variable dependiente pronosticada. Los puntos se agrupan en torno a la bisectriz aunque aparentemente parece haber una variabilidad mayor en torno del 20. Además la tirada 19 parece separarse del resto.



CASO ANISÓTROPO PREFERENTE VERTICAL

En la **Figura 90** se observa el diagrama de dispersión y la línea de ajuste total.

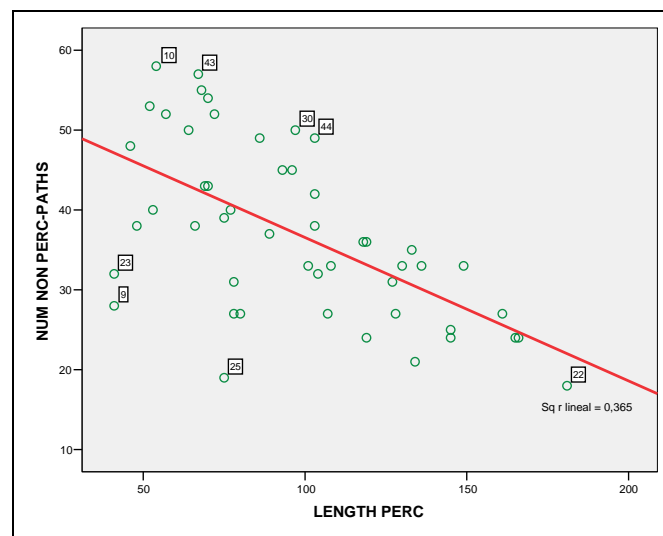


Figura 90. Diagrama de dispersión.

El modelo estimado es la siguiente recta:

$$54,501 - 0,18 \text{ length_perc} = \text{PRE_2}$$

Modelo		Coeficientes no estandarizados		Coeficientes estandarizados	t	Sig.	Intervalo de confianza para B al 95%	
		B	Error típ.	Beta			Límite inferior	Límite superior
1	(Constante)	54,501	3,531		15,437	0,000	47,402	61,600
	LENGTH PERC	-0,180	0,034	-0,604	-5,257	0,000	-0,248	-0,111

Tabla 27. Coeficientes.

Los análisis de residuos se realizan a partir del histograma de residuos y el gráfico P-P que se representan en las **Figuras 91** y **92**.

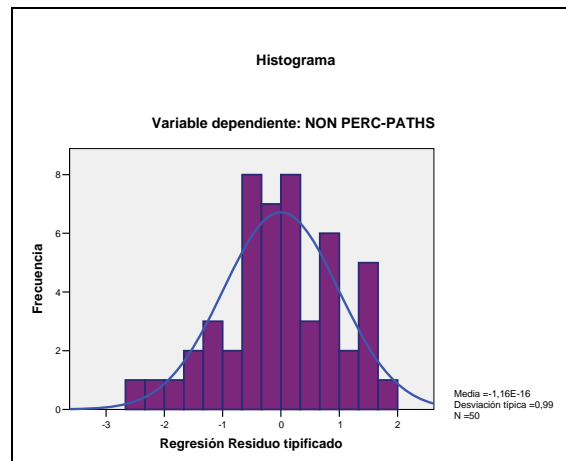


Figura 91. Histograma de residuos.

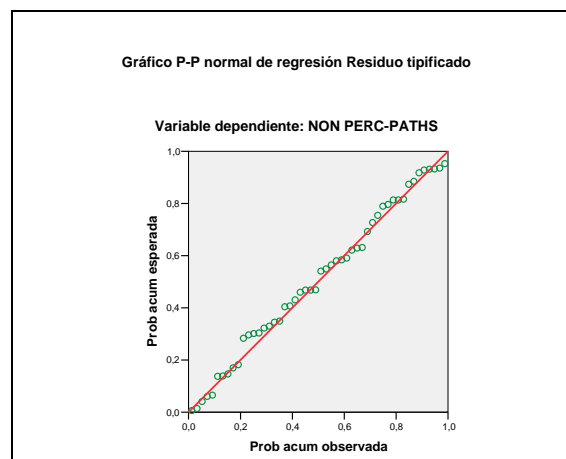


Figura 92. Diagrama p-p.

CASO ANISÓTROPO PREFERENTE HORIZONTAL

Se repite el mismo modelo de regresión lineal. Las figuras y tablas para el caso anisótropo horizontal son las siguientes.

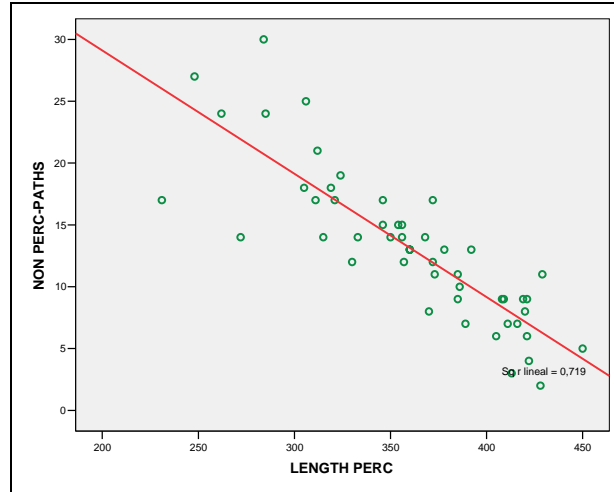


Figura 93. Diagrama de dispersión de *length* vs *non_percpaths*.

	Media	Desviación típ.
NON PERC-PATHS	13,18	6,134
LENGTH PERC	359,70	52,130

Tabla 28. Estadísticos descriptivos.

Se concluye con el modelo estimado a partir de la siguiente recta:

$$49,056 - 0,1 \cdot \text{length_perc} = \text{PRE_3}$$

Modelo		Coeficientes no estandarizados		Coeficientes estandarizados	t	Sig.	Intervalo de confianza para B al 95%	
		B	Error típ.	Beta			Límite inferior	Límite superior
1	(Constante)	49,056	3,274		14,986	0,000	42,474	55,638
	LENGTH PERC	-0,100	0,009	-0,848	-11,072	0,000	-0,118	-0,082

Tabla 29. Coeficientes.

ALFACRIT versus NON PERCPATHS

Las figuras y tablas mostradas a continuación, pertenecen a la regresión lineal de cada uno de los casos y se toma el parámetro *alfacrit* como dependiente y *nonperc_paths* como dependiente.

CASO ISÓTROPO

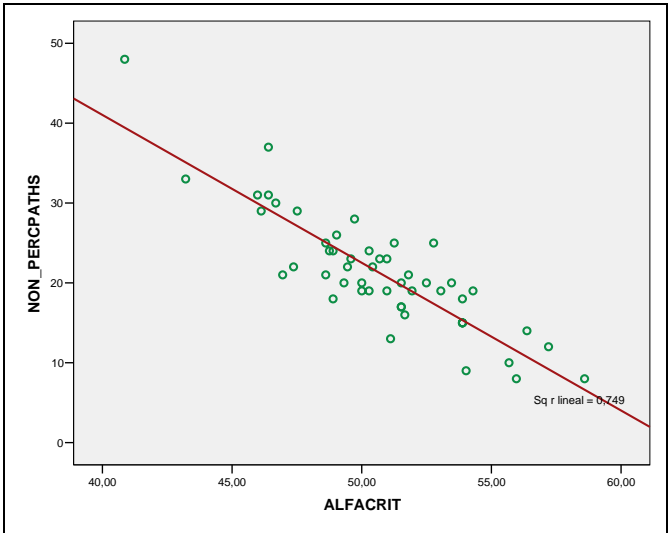


Figura 94. Diagrama de dispersión de alafcrit vs nonperc_paths.

	Media	Desviación típ.	N
NON PERC-PATHS	21,42	7,376	50
ALFACRIT	50,5954	3,45053	50

Tabla 30. Estadísticos descriptivos.

		NUM NON PERC-PATHS	ALFACRIT
Correlación de Pearson	NON PERC-PATHS	1,000	-0,866
	ALFACRIT	-0,866	1,000
Sig. (unilateral)	NON PERC-PATHS	.	0,000
	ALFACRIT	0,000	.

Tabla 31. Correlaciones.

Modelo	R	R cuadrado	R cuadrado corregida	Error típ. de la estimación
1	0,866	0,749	0,744	3,732

Tabla 32. Resumen del modelo.

Modelo		Suma de cuadrados	gl	Media cuadrática	F	Sig.
1	Regresión	1997,549	1	1997,549	143,401	0,000
	Residual	668,631	48	13,930		
	Total	2666,180	49			

Tabla 33. ANOVA.

El modelo estimado es la recta:

$$115,042 - 1,850 \cdot \text{alfacrit} = \text{PRE}_4$$

Coeficientes								
Modelo		Coeficientes no estandarizados		Coeficientes estandarizados	t	Sig.	Intervalo de confianza para B al 95%	
		B	Error típ.	Beta			Límite inferior	Límite superior
1	(Constante)	115,042	7,836		14,681	0,000	99,287	130,797
	ALFACRIT	-1,850	0,155	-0,866	-1,975	0,000	-2,161	-1,540

CASO ANISÓTROPO VERTICAL

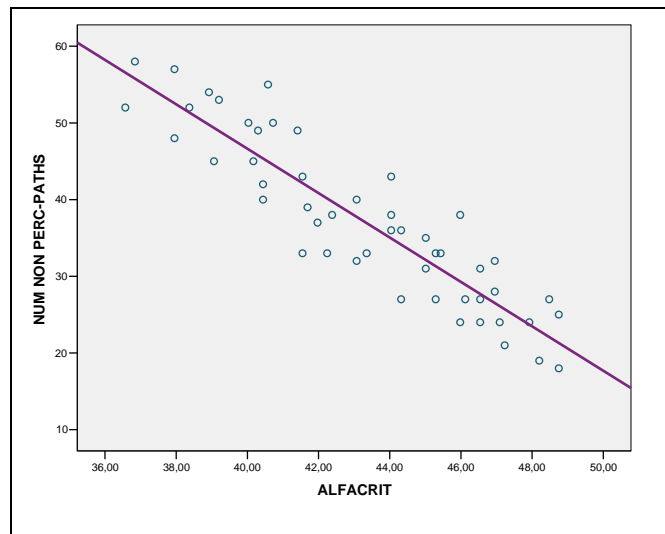


Figura 95. Diagrama de dispersión de alfacrit vs nonperc_paths.

	Mean	Std. Deviation
NON PERC-PATHS	37,10	10,792
ALFACRIT	43,2926	3,38806

Tabla 34. Estadísticos descriptivos.

El modelo estimado es la recta:

$$162,446 - 2,895 \cdot \text{alfacrit} = \text{PRE}_5$$

Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.	95% Confidence Interval for B	
		B	Std. Error	Beta			Lower Bound	Upper Bound
1	(Constant)	162,446	8,321		19,523	0,000	145,715	179,176
	ALFACRIT	-2,895	0,192	-.0909	-5,109	0,000	-3,281	-2,510

Tabla 35. Coeficientes.

CASO ANISÓTROPO HORIZONTAL

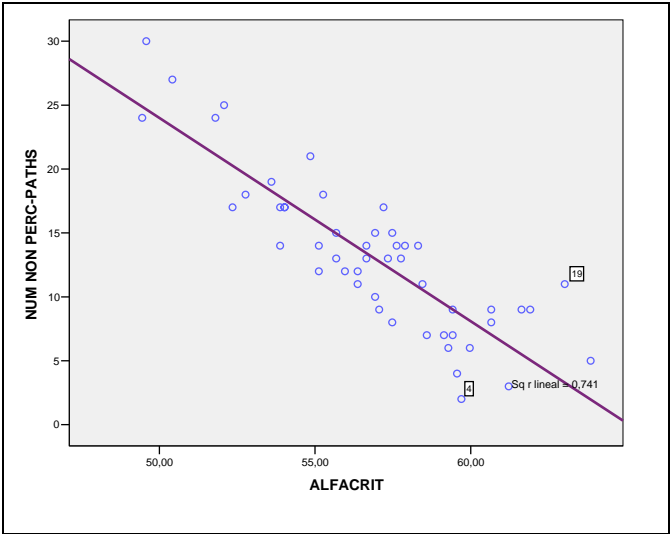


Figura 96. Diagrama de dispersión de alfacrit vs nonperc_paths.

	Mean	Std. Deviation
NON PERC-PATHS	13,18	6,134
ALFACRIT	56,8008	3,31798

Tabla 36. Estadísticos descriptivos.

El modelo estimado es la recta:

$$103,567 - 1,591 \cdot alfacrit = PRE_6$$

ALFACRIT versus NONPERC PATHS versus LENGTH PERC

CASO ISÓTROPO (NO INCLUYE LA TIRADA N° 19)

A continuación se procede a eliminar la tirada número 19. Como se ha comentado anteriormente, dicha tirada responde a un caso particular y aislado. La particularidad de la tirada 19 es la siguiente: le corresponde el valor de *alfacrit* menor, es decir, que con un 40,86% de lanzamientos, es la tirada en la que se produce la percolación más rápidamente. Su longitud será lógicamente menor, ya que el camino percolante tiene menos tortuosidad. La particularidad será pues, en encontrar un alto número de caminos no percolantes

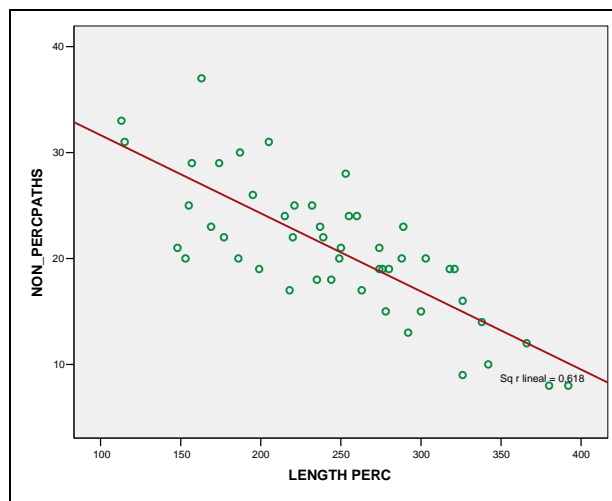


Figura 97. Diagrama de dispersión de *nonperc_paths* vs *length_perc* sin considerar tirada 19.

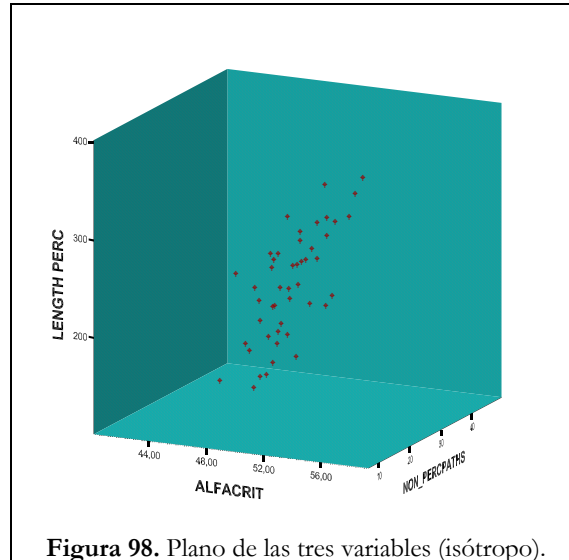
	Mean	Std. Deviation
NON PERC-PATHS	20,88	6,366
ALFACRIT	50,7941	3,18423
LENGTH PERC	245,92	67,897

Tabla 37. Estadísticos descriptivos.

En la matriz de correlaciones, las variables *length* y *alfacrit* tienen una correlación igual a 0.853, por tanto, sus ejes forman un ángulo de 31,46°.

El modelo estimado es el siguiente plano:

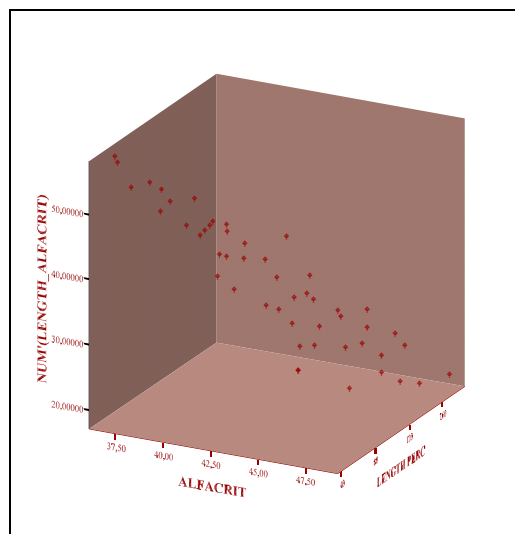
$$89,317 - (1,228 \cdot \text{alfacrit}) - (0,025 \cdot \text{length_perc}) = \text{PRE}_7$$



CASO ANISÓTROPO PREFERENTE VERTICAL

El modelo estimado es el plano:

$$154,793 - (2,602 \cdot \text{alfacrit}) - (0,044 \cdot \text{length_perc}) = \text{PRE}_8$$



CASO ANISÓTROPO PREFERENTE HORIZONTAL

El modelo estimado es el plano:

$$83,598 - (0,959 \cdot \text{alfacrit}) - (0,044 \cdot \text{length_perc}) = \text{PRE}_9$$

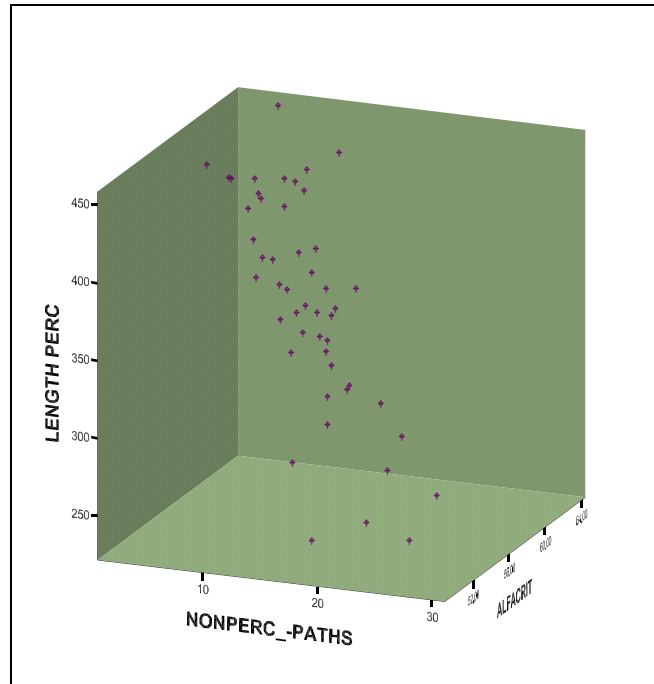


Figura 100. Plano de las tres variables (horizontal).