

Práctica 4: diagramas de Voronoi

Raul L.

8 de marzo de 2022

1. Introducción

El tema de esta cuarta práctica tiene su importancia en las matemáticas puras y en las ciencias aplicadas como por ejemplo, en la ciencia de materiales. Se toma de un espacio bidimensional una zona con medidas conocidas que contiene k puntos semillas p_i representados por sus coordenadas (x_i, y_i) lo que se busca es dividir esa zona en regiones llamadas celdas de Voronoi de tal forma que todos los puntos que pertenecen a la región de p_i estén más cerca de esa semilla que de cualquier otra.

El modelo matemático en si es continuo, es decir, las coordenadas son números reales, pero nosotros lo vamos a discretizar en esta práctica [1].

2. Objetivo

Examinar el efecto de la tasa n versus k (por lo menos tres niveles de la densidad de semillas), en la probabilidad de que una segunda grieta llegue a tocar una primera grieta (es decir, fracturando la pieza dos veces con posiciones iniciales generadas independientemente al azar, sobre varias réplicas), visualizando los resultados y aplicando métodos estadísticos [1].

3. Código

En el siguiente código se realizó un estudio cuantificando la probabilidad que hay para que dos grietas se intercepten, estas se muestran de dos diferentes colores (negro y blanco), para ello, el pixel de color rojo muestra el punto exacto de interferencia. Para realizar un concreto estudio, se varió el numero de semillas en 3 diferentes números de semillas dejando fijo el tamaño de la celda, esto garantiza realizar un estudio sistemático eficiente, el cual se representa en forma de gráfica de barras horizontales y verticales, así como también, se muestran las imágenes donde las grietas hicieron contacto con los diferentes números de semilla.

Código en Python

<https://github.com/satuelisa/Simulation/blob/master/VoronoiDiagrams/fracture.py>

****Código creado en Python****

https://github.com/Raullr28/Resultados/blob/main/P4/codigo_celdas.py

```

1  sem=8, 40, 120
2  porcentaje=[]
3  for k in sem:
4      print(" semillas:",k,)
5      n, semillas = 80, []
6      for s in range(k):
7          while True:
8              x, y = randint(0, n - 1), randint(0, n - 1)
9              if (x, y) not in semillas:
10                 semillas.append((x, y))
11                 break
12
13  celdas = [celda(i) for i in range(n * n)]
14  voronoi = Image.new('RGB', (n, n))
15  vor = voronoi.load()
16  c = sns.color_palette("Set3", k).as_hex()
17  for i in range(n * n):
18      vor[i % n, i // n] = ImageColor.getrgb(c[celdas.pop(0)])
19  limite, vecinos = 10, [] # se modifiko para que produzca mas grietas
20  for dx in range(-1, 2):
21      for dy in range(-1, 2):
22          if dx != 0 or dy != 0:
23              vecinos.append((dx, dy))

```

Código 1: Representa la automatización para variar el número de semillas que aparecen.

```

1  def propaga(replica):
2      prob, difcil = 0.9, 0.8
3      grieta = voronoi.copy()
4      g = grieta.load()
5      (x, y) = inicio()
6      largo = 0
7      negro = (0, 0, 0)
8      while True:
9          g[x, y] = negro
10         largo += 1
11         frontera, interior = [], []
12         for v in vecinos:
13             (dx, dy) = v
14             vx, vy = x + dx, y + dy
15             if vx >= 0 and vx < n and vy >= 0 and vy < n: # existe
16                 if g[vx, vy] != negro: # no tiene grieta por el momento
17                     if vor[vx, vy] == vor[x, y]: # misma celda
18                         interior.append(v)
19                     else:
20                         frontera.append(v)
21         elegido = None
22         if len(frontera) > 0:
23             elegido = choice(frontera)
24             prob = 1
25         elif len(interior) > 0:
26             elegido = choice(interior)
27             prob *= difcil
28         if elegido is not None:
29             (dx, dy) = elegido
30             x, y = x + dx, y + dy
31         else:
32             break # ya no se propaga
33     if largo >= limite: # aqui decide que imprima las mayores a 80 el limite
34         visual = grieta.resize((10 * n, 10 * n))
35     return (largo, grieta)

```

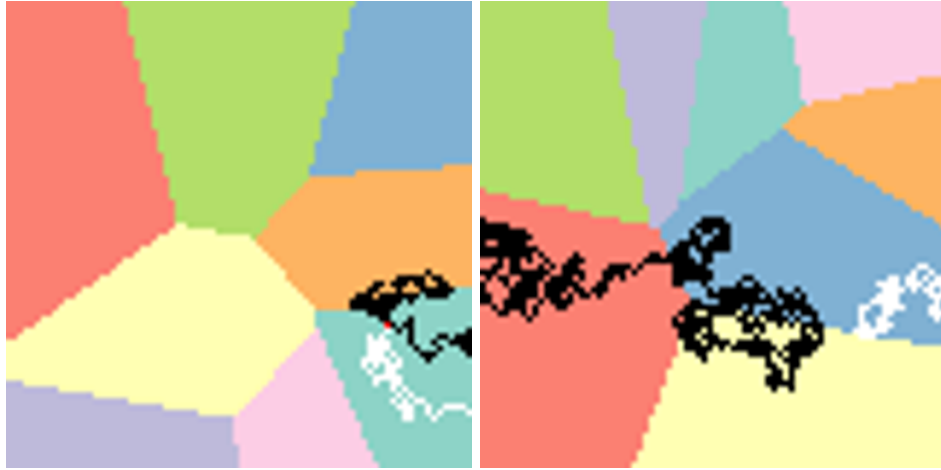
Código 2: Representa un comando para generar la grieta.

```

1  rep=400
2  contacto=[]#guarda cuantas veces chocaron
3  for r in range(rep): # pruebas sin paralelismo
4      largo, nueva_grieta =propaga(r)# para grieta 1 color negro
5      if largo > limite:
6          largo2 =propaga2(r,nueva_grieta,contacto)#para grieta 2 color blanca
7  print(contacto)
8  probabilidad= ((len(contacto))/rep)*100#convierte a porcentaje
9  porcentaje.append(probabilidad)
10 print("Probabilidad de contacto entre grietas es:",probabilidad, "%")

```

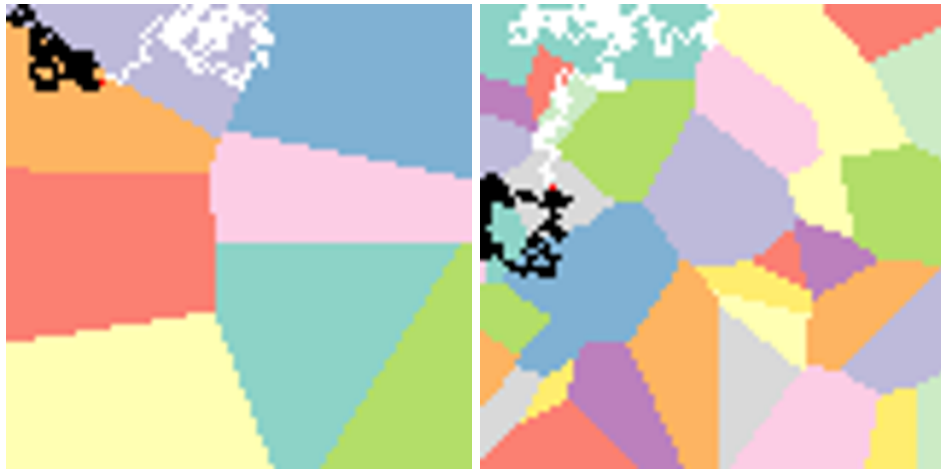
Código 3: Representa el lugar donde hacen contacto las grietas.



(a) Matriz 80x80

(b) Matriz 80x80

Figura 1: imagen de unión de grietas con igual número de semilla.



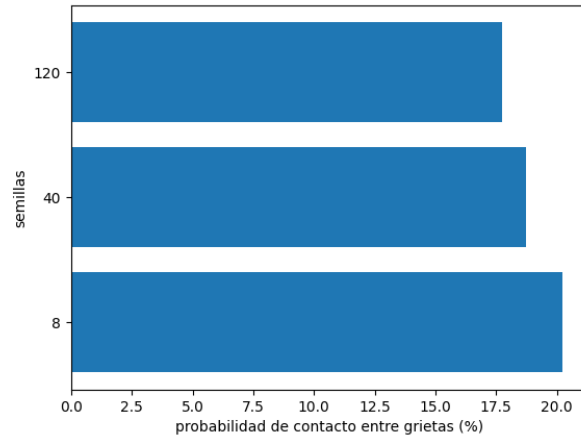
(a) Matriz 80x80

(b) Matriz 80x80

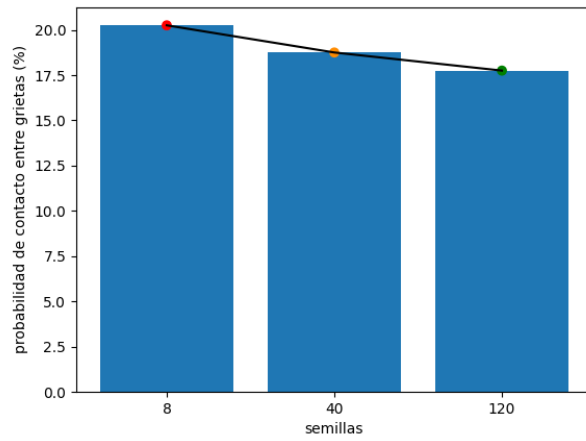
Figura 2: imagen de unión de grietas con diferente número de semilla.

4. Resultados

En las figuras se muestra el porcentaje de intercepción que existió en cada uno de los diferentes números de semillas, para poder dar un resultado estadístico así, se realizó el experimento con un número alto de repeticiones lo cual nos da las gráficas siguientes.



(a)



(b)

Figura 3: Grafica estadística de porcentaje.

5. Reto 1

El primer reto es crecer las celdas dinámicamente alrededor de semillas de tal forma que las semillas aparecen al azar en distintas iteraciones y crecen con una tasa exponencialmente distribuida (variable entre núcleos pero constante para un núcleo específico) hasta toparse con las demás celdas, así como se muestra en la animación. Examina los cambios producidos en el fenómeno de propagación de grietas que esta nueva forma de crear las celdas provoca, ya que las semillas resultan en celdas de tamaños distintos según su edad y su tasa, además del efecto de la posición relativa a las demás semillas.

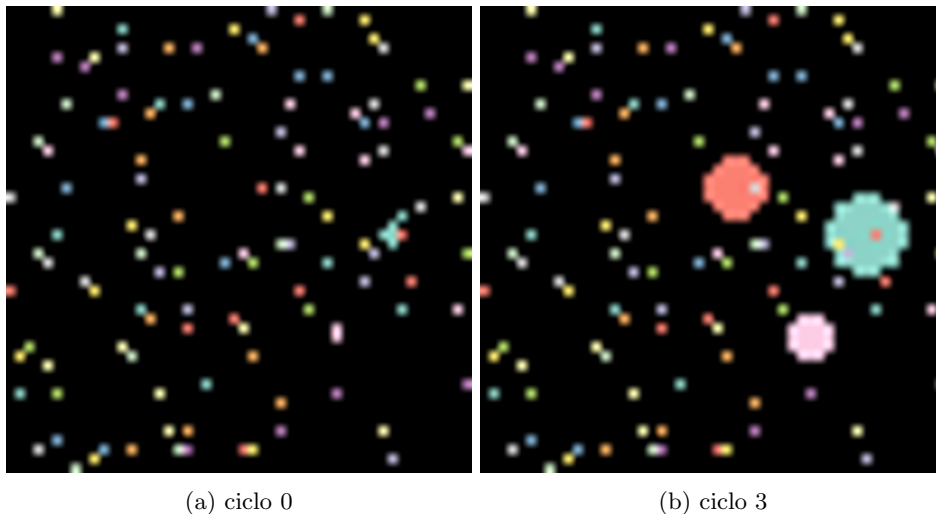


Figura 4: Imagen de crecimiento aleatorio de semillas.

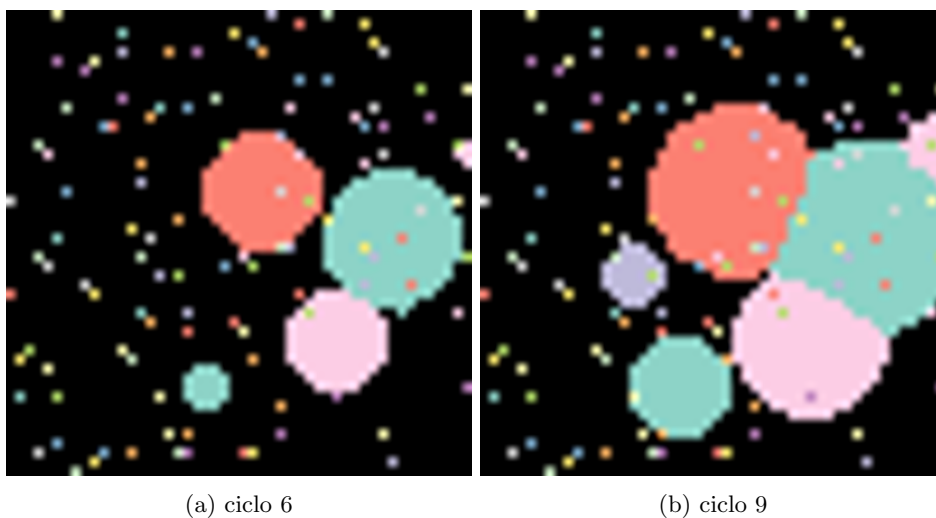


Figura 5: Imagen de crecimiento aleatorio de semillas.

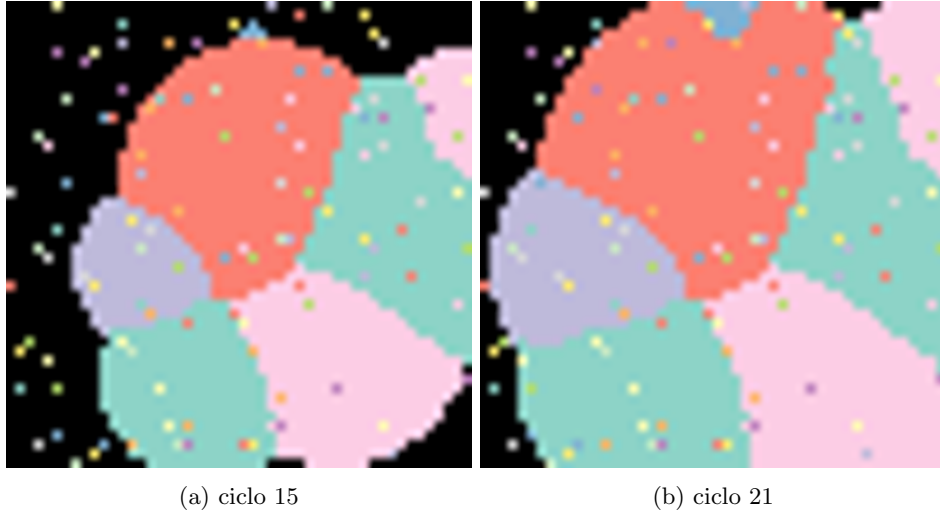


Figura 6: Imagen de crecimiento aleatorio de semillas.

6. Resultados

Se agregó una función donde van creciendo las semillas, cuando crece una semilla se agrega una probabilidad de que otra semilla naciera y generando un análisis de probabilidad donde se determinará el porcentaje que aparezca otra semilla.

7. Conclusión

Se demostró que aumentando el número de repeticiones se puede tener una proporcionalidad en los resultados, demostrando así, un porcentaje alto de intercepción en las grietas con un pequeño número de semillas y por otro lado, muestra un porcentaje más pequeño con un número más pequeño de semillas.

Referencias

- [1] E. Schaeffer. Práctica 4: diagramas de voronoi. 2022. URL <https://satuelisa.github.io/simulation/p4.html>.