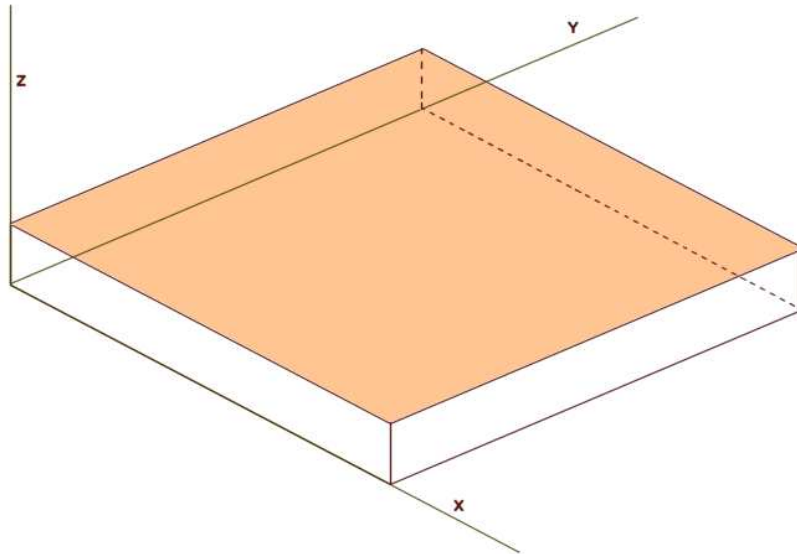


## PAISAJE FRACTAL. PROGRAMACIÓN EN PHP PDF

### PROCESO DE CONSTRUCCIÓN

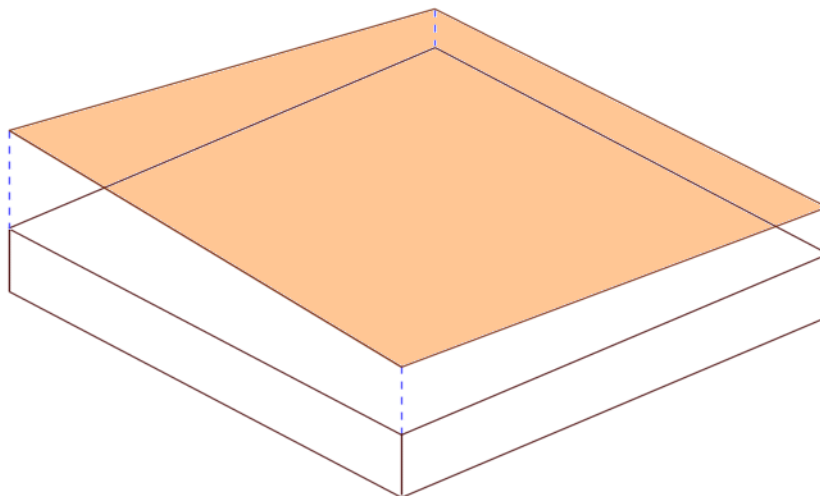
#### **ITERACIÓN 0**

Comenzaremos por trazar un cuadrado sobre el plano XY. Este cuadrado viene determinado por sus cuatro vértices.



#### **ITERACIÓN 1**

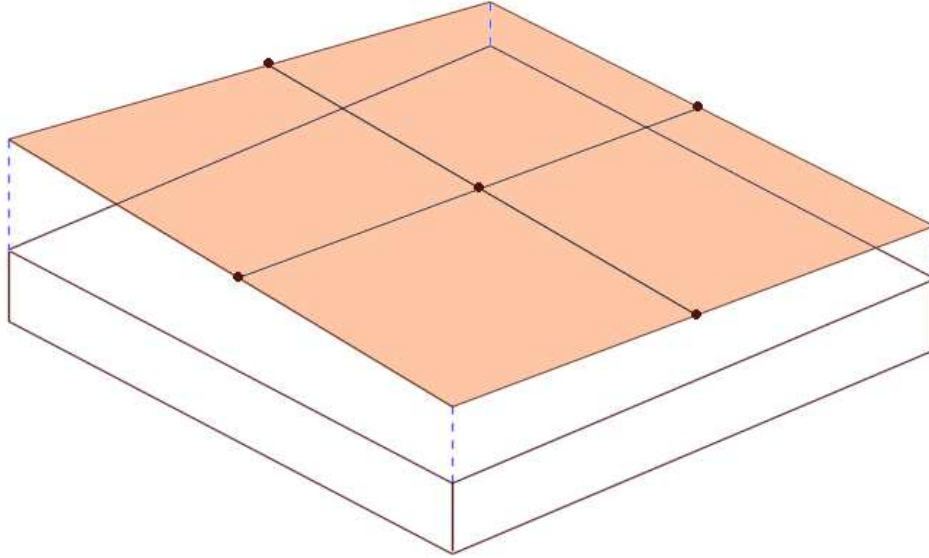
Sobre cada vértice realizamos una perturbación aleatoria vertical, para más tarde unir los nuevos vértices formando un cuadrilátero





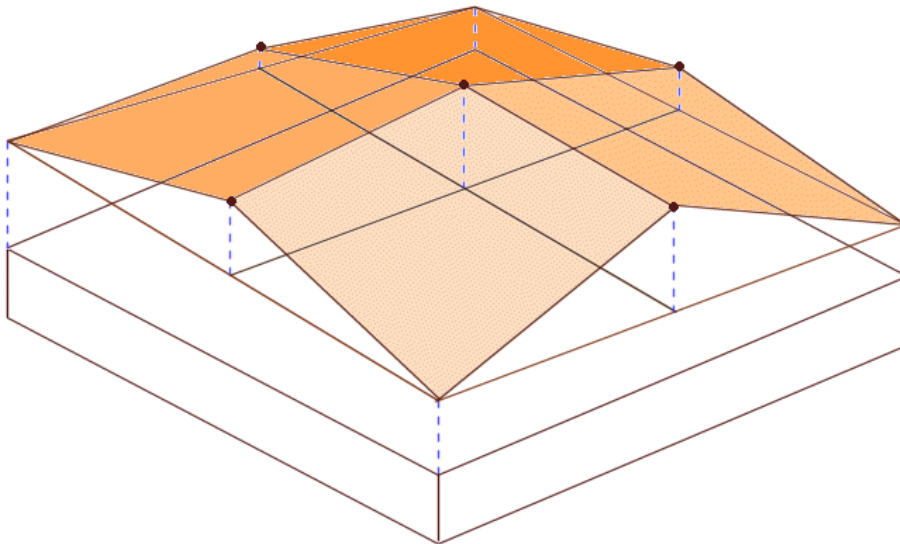
## ITERACIÓN 2

En este cuadrilátero, determinamos los puntos medios de cada lado, dibujamos dos segmentos que unan puntos medios opuestos, y marcamos el punto de intersección de estos dos segmentos.



En cada uno de estos cinco puntos, establecemos una perturbación aleatoria vertical, determinando 5 nuevos puntos.

Uniendo los puntos de 4 en 4 se formarán 4 cuadriláteros.



## ITERACIÓN K

Continuando este proceso, se formará un paisaje tanto más real como mayor sea la iteración generada.

## PERTURBACIONES

Sobre punto medio de un lado:



La base sobre la que efectuar la perturbación será el punto medio con coordenadas calculadas:

$$\left( \frac{x_b + x_a}{2}, \frac{y_b + y_a}{2} \right)$$

Sobre punto central de un cuadrilátero:

En este caso, la base será media de los cuatro puntos determinados por su vértices:

$$\left( \frac{x_b + x_a + x_c + x_d}{4}, \frac{y_b + y_a + y_c + y_d}{4} \right)$$

Sobre estos puntos, generamos una perturbación aleatoria de dos tipos:

- LINEAL

El rango de valores sobre los que obtener el número aleatorio decrece aritméticamente conforme aumentan las iteraciones.

$$\text{valor\_máximo} = 100 - k + L$$

$$p = \text{RAND}(0, \text{valor\_máximo})$$

$$\text{aleatorio} = \text{RAND}(-p, p) \cdot F_z$$

- GEOMÉTRICA

El rango de valores sobre los que obtener el número aleatorio decrece geoméricamente conforme aumentan las iteraciones.

$$p = \frac{100}{2^k} + L$$

$$\text{aleatorio} = \text{RAND}(-p, p) \cdot F_z$$

L es una constante de "laminado". Permite corregir eventuales saltos bruscos en la superficie del terreno

$F_z$  es el factor de altura. Permite establecer una reducción o ampliación del valor aleatorio calculado

Este "valor" aleatorio es el que asignamos como componente del eje "OZ", es decir, la altura del punto.

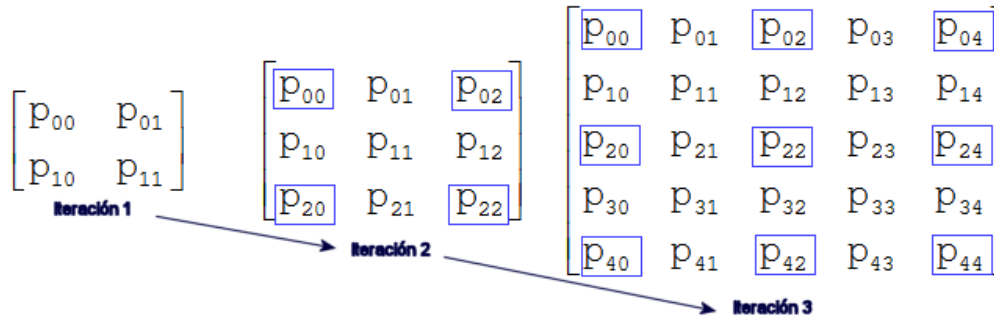
A cada cuadrilátero generado le llamaremos BLOQUE.

Queda claro que en cada iteración un bloque determina 4 bloques más.

El esquema del tratamiento de bloque es el siguiente:

## **TRATAMIENTO DE LOS BLOQUES POR CADA ITERACIÓN**

Veamos la secuencia de puntos en las tres primeras iteraciones:



En cada iteración, los puntos señalados con un recuadro son los puntos determinados en la iteración anterior; los no señalados son los calculados por el método descrito anteriormente en la iteración actual.

Observamos que (considerando el 0 par):

- los puntos fijos tienen por coordenadas en la iteración actual el doble que en la iteración anterior
- los puntos "fijos" calculados en iteraciones anteriores tienen ambas coordenadas PARES
- los puntos medios de segmentos tienen una coordenada PAR y otra IMPAR
- los puntos "centrales" tienen ambas coordenadas IMPARES

Con esto, podemos establecer un algoritmo que permita recalcular los puntos de una iteración basándonos en la iteración anterior.

Denotaremos con  $P$  a los puntos de la iteración actual, y con  $Q$  los de la anterior

- La dimensión de las matrices de puntos según la iteración  $k$  es:  $\text{Dim}_k = 2^{k-1} + 1$

$$i = 0, \dots, 2^{k-1}$$

$$j = 0, \dots, 2^{k-1}$$

- Si  $i$  y  $j$  son pares:  $P_{\frac{i}{2} \frac{j}{2}} = Q_{\frac{i}{2} \frac{j}{2}}$

- Si  $i$  es par y  $j$  impar:  $P_{i \frac{j}{2}} = \frac{Q_{\frac{i}{2} \frac{j-1}{2}} + Q_{\frac{i}{2} \frac{j+1}{2}}}{2}$

- Si  $i$  es impar y  $j$  par:  $P_{\frac{i-1}{2} j} = \frac{Q_{\frac{i-1}{2} \frac{j}{2}} + Q_{\frac{i+1}{2} \frac{j}{2}}}{2}$

- Si  $i$  y  $j$  son impares:  $P_{\frac{i-1}{2} \frac{j-1}{2}} = \frac{Q_{\frac{i-1}{2} \frac{j-1}{2}} + Q_{\frac{i+1}{2} \frac{j-1}{2}} + Q_{\frac{i-1}{2} \frac{j+1}{2}} + Q_{\frac{i+1}{2} \frac{j+1}{2}}}{4}$

Con esto quedaría creada la matriz de puntos y sus alturas base.

Solamente quedaría realizar la perturbación del modo descrito sobre los puntos que no tengan las dos coordenadas pares, comenzando por la que solamente tienen una coordenada impar y terminando por los que tienen ambas impares.

Con esto quedarían determinadas todas las iteraciones. Ahora debemos dibujarlo.

Así pues, para modelizar la creación de iteraciones en bucle:



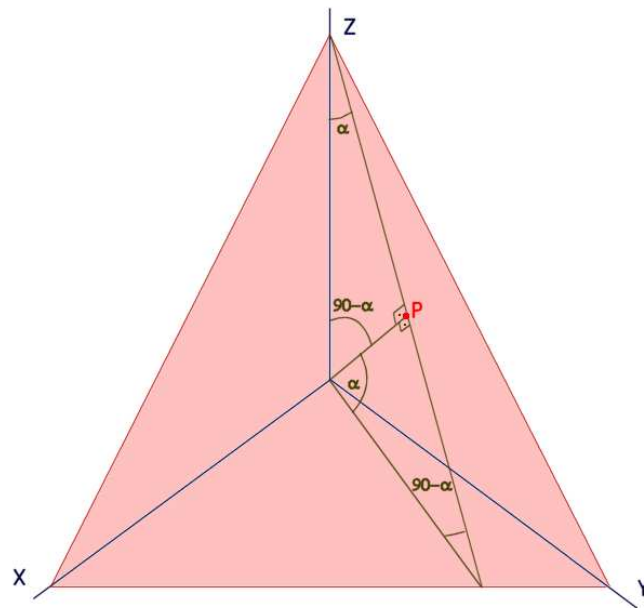
- 1º.- matriz cuadrada de puntos de iteración anterior:  $X_0$  (ya creada)
- 2º.- matriz cuadrada de puntos de iteración actual:  $X_1$  (inicializar)
- 3º.- Formar la matriz  $X_1$  de acuerdo con los criterios expuestos con anterioridad
- 4º.- Borrar la matriz  $X_0$
- 5º.- Crear una nueva matriz  $X_0$  en la que copiamos la matriz  $X_1$
- 6º.- Borrar la matriz  $X_1$
- 7º.- Próxima iteración
- ...

Realizar una proyección isométrica sobre todos los puntos de la matriz  $X_0$

Dibujar cuadrilátero a cuadrilátero, de arriba hacia abajo y de izquierda a derecha

## DIBUJO DEL PAISAJE

Utilizando la transformación isométrica, supongamos que el punto P del espacio lo queremos proyectar sobre el plano (en color salmón):



$$\mathbb{R}^3 \longrightarrow \mathbb{R}^2$$

$$(x, y, z) \rightarrow (X', Y')$$

$$\begin{cases} X' = \frac{\sqrt{2}}{2} \cdot (x - y) \\ Y' = \sqrt{\frac{2}{3}} \cdot z - \frac{1}{\sqrt{6}} \cdot (x + y) \end{cases}$$



## **CODIFICACIÓN A PDF**

No hay problemas importantes a la hora de codificar en PHP-PDF salvo que el sistema de coordenadas tiene el origen en el vértice superior izquierdo de la hoja; el eje X aumenta hacia la derecha, pero el eje Y funciona al revés, aumenta hacia abajo, por lo que los dibujos saldrán invertidos verticalmente salvo que corriamos esta diferencia.