

Método de diferencias finitas en el lenguaje Julia (La ecuación de Onda)

Yair Antonio Castillo Castillo

Manuel Irán Torres González

Asesor: Dr. Jonás De Basabe Delgado

28 de julio de 2017

1 INTRODUCCIÓN

La mayoría de las ecuaciones diferenciales que describen el comportamiento de la naturaleza, por más sencillas que sean, no tienen soluciones analíticas, por lo que se han desarrollado muchos métodos que aproximan numéricamente estas soluciones. En este trabajo nos centramos en la ecuación de onda acústica en dos dimensiones.

$$\frac{\partial^2 u}{\partial t^2} = c^2 \Delta u$$

Y en dos dimensiones

$$\frac{\partial^2 u}{\partial t^2} = c^2 \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial z^2} \right)$$

Hay 2 tipos de programación en que estamos interesados: **Programación con memoria compartida:** Todos los procesadores se encuentran igualmente comunicados con la memoria principal y pueden acceder por medio de un ducto común. **Programación para computadoras paralelas:** La programación paralela es diferente para cada arquitectura. La programación con hilos se aplica principalmente a las arquitecturas de multiprocesadores con memoria compartida y la programación con paso de mensajes a las arquitecturas de multicomputadoras.

En lenguaje de programación Julia provee un entorno de multiprocesamiento basado en pasar mensajes para permitir a los programas correr en múltiples procesos en dominios de la memoria separa a la vez. La programación paralela en Julia se basa en dos primitivas: referencias remotas y

llamadas remotas. Una referencia remota es un objeto que puede utilizarse desde cualquier proceso para referirse a un objeto almacenado en un proceso en particular. Una llamada remota es una petición de un proceso para llamar a una determinada función en ciertos argumentos en otro proceso (posiblemente el mismo).

2 METODOLOGÍA

El método de diferencias finitas (FDM) consiste en reemplazar las derivadas en las ecuaciones diferenciales por aproximaciones dadas por fórmulas de diferencias finitas basadas en valores de la función en algunos puntos discretos. Para hacer esto, podemos partir de la definición de derivada:

$$f'(x) = \lim_{\Delta x \rightarrow 0} \frac{f(x + \Delta x) - f(x)}{\Delta x}$$

Entonces, si queremos obtener una aproximación D_+ de la derivada de una función $w(x)$ en el punto \bar{x} podemos hacer:

$$D_+w(x) = \frac{w(\bar{x} + h) - w(\bar{x})}{h}$$

Para algún valor chico de h . Notemos que necesitamos saber el valor de la función w en algún valor de $x > \bar{x}$. Podemos obtener otra aproximación D_- para $w(x)$ muy parecido al anterior

$$D_-w(x) = \frac{w(\bar{x}) - w(\bar{x} - h)}{h}$$

Para algún valor chico de h . Notemos que necesitamos saber el valor de la función w en algún valor de $x < \bar{x}$. Decimos que D_+ y D_- son aproximaciones por *un lado*. Si promediamos estas dos, tenemos otra aproximación D_0w para $w'(\bar{x})$ a la que llamaremos *centrada*:

$$D_0w(\bar{x}) = \frac{1}{2}(D_+w(x) + D_-w(x)) = \frac{w(\bar{x} + h) - w(\bar{x} - h)}{2h}$$

Con Con ayuda de series de potencias, podemos ver que D_+ y D_- tienen un error proporcional a h mientras que el error de D_0 es proporcional a h^2 . Expandimos en series de potencias $w(\bar{x} + h)$ y $w(\bar{x} - h)$:

$$w(\bar{x} + h) = w(\bar{x}) + hw'(\bar{x}) + \frac{1}{2}h^2w''(\bar{x}) + \frac{1}{6}h^3w'''(\bar{x}) + O(h^4)$$

$$w(\bar{x} - h) = w(\bar{x}) - hw'(\bar{x}) + \frac{1}{2}h^2w''(\bar{x}) - \frac{1}{6}h^3w'''(\bar{x}) + O(h^4)$$

Si usamos estas expresiones para obtener D_+ y D_- obtenemos:

$$D_+w(x) = \frac{w(\bar{x} + h) - w(\bar{x})}{h} = w'(\bar{x}) + \frac{1}{2}hw''(\bar{x}) + \frac{1}{6}h^2w'''(\bar{x}) + O(h^3)$$

$$D_-w(x) = \frac{w(\bar{x}) - w(\bar{x} - h)}{h} = w'(\bar{x}) - \frac{1}{2}hw''(\bar{x}) + \frac{1}{6}h^2w'''(\bar{x}) + O(h^3)$$

Mientras que $D_0w(\bar{x})$ se ve como:

$$D_0w(\bar{x}) = \frac{w(\bar{x} + h) - w(\bar{x} - h)}{2h} = w'(\bar{x}) + \frac{1}{6}h^2w'''(\bar{x}) + O(h^4)$$

Nos interesa saber aproximaciones de w'' . Para ello, vamos a usar una aproximación a w' muy parecida a la centrada.

$$\hat{D}_0 w(\bar{x}) = \frac{1}{h} \left(w \left(x + \frac{h}{2} \right) - w \left(x - \frac{h}{2} \right) \right)$$

Luego aplicamos esta aproximación a $w(x + \frac{h}{2})$ y a $w(x - \frac{h}{2})$

$$\begin{aligned} \hat{D}_0 w(\bar{x}) &= \frac{1}{h} \left(w \left(x + \frac{h}{2} \right) - w \left(x - \frac{h}{2} \right) \right) \\ &= \frac{1}{h} \left(\frac{1}{h} (w(x+h) - w(x)) - \frac{1}{h} (w(x) - w(x-h)) \right) \\ &= \frac{1}{h^2} (w(x+h) - w(x) - w(x) - w(x-h)) \\ &= \frac{1}{h^2} (w(x+h) - 2w(x) - w(x-h)) \end{aligned}$$

Volvemos a escribir la ecuación de onda:

$$\frac{\partial^2 u}{\partial t^2} = c^2 \Delta u + f$$

Donde c es la velocidad de propagación de la onda y f se define como:

$$f(x, z, t) = \delta(x - x_s) \delta(z - z_s) \omega(t)$$

Con δ y ω definidas como lo siguiente:

$$\lambda(i) = \begin{cases} 0 & \text{si } i = 0 \\ 1 & \text{si } i \neq 0 \end{cases}$$

$$\omega(t) = e^{-\pi^2 f_{pk}^2 (t-\hat{t})^2}$$

Supongamos que queremos modelar la onda en el rectángulo $D = [-500, 500] \times [-500, 500]$ en el plano xy , con las siguientes condiciones de frontera:

$$u(x, z, t) = 0 \quad \forall (x, z) \in \partial D \quad (2.1)$$

Y con las condiciones iniciales

$$u(x, z, 0) = 0 \quad \forall (x, z) \in \partial D \quad (2.2)$$

Para comenzar con el FDM, vamos a discretizar tanto el espacio (D), como el tiempo. Para discretizar D, vamos a dividir el eje x y el eje z en partes de longitud dx y dz respectivamente. Lo mismo haremos con el tiempo, en parte de longitud dt . Esto se puede apreciar en la Figura 4.2

Adoptamos la siguiente notación:

$$u_{i,j}^k = u(x_i, z_j, t_k) \quad (2.3)$$

Entonces la ecuación de onda se puede reescribir como lo siguiente

$$\frac{u_{i,j}^{k-1} - 2u_{i,j}^k + u_{i,j}^{k+1}}{(dt)^2} = c_{i,j} \left(\frac{u_{i-1,j}^k - 2u_{i,j}^k + u_{i+1,j}^k}{(dx)^2} + \frac{u_{i,j-1}^k - 2u_{i,j}^k + u_{i,j+1}^k}{(dz)^2} \right) + f(x, z, t) \quad (2.4)$$

Donde c ahora es una función $c(x, z)$, por lo que ahora la velocidad depende del espacio. Y finalmente despejamos $u_{i,j}^{k+1}$

$$u_{i,j}^{k+1} = (dt)^2 \left(c_{i,j} \left(\frac{u_{i-1,j}^k - 2u_{i,j}^k + u_{i+1,j}^k}{(dx)^2} + \frac{u_{i,j-1}^k - 2u_{i,j}^k + u_{i,j+1}^k}{(dz)^2} \right) + f(x, z, t) \right) - u_{i,j}^{k-1} + 2u_{i,j}^k \quad (2.5)$$

Usando que $u_{i,j}^0 = u_{i,j}^1 = 0 \quad \forall (i, j)$ tal que $(x_i, z_j) \in D$

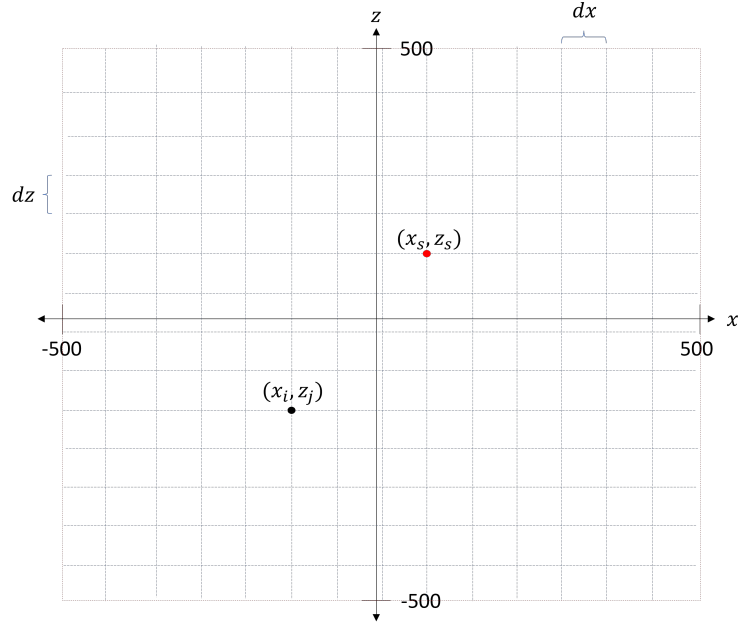


Figure 2.1: Donde $x_{i+1} - x_i = dx$, $z_{i+1} - z_i = dz$ y $t_{i+1} - t_i = dt$

3 IMPLEMENTACION

Nuestra computadora cuenta con 4 procesadores, la computadora del investigador cuenta con 20 procesadores, y la supercomputadora Lamb tiene aproximadamente 34 nodos y cada uno con 20 procesadores. Pero para implementar en una supercomputadora se necesita usar Memoria Distribuida y no memoria compartida como lo que hemos hecho. FDMParallel es para la simulación de una hora y FDMParallel2 es para la simulación de dos ondas.

```

1 function FDMParallel(nx,nz,nt,t,C,x0,xm,z0,zm,fpk,that,xs,zs)
2   dx=abs(xm-x0)/(nx-1)
3   dz=abs(zm-z0)/(nz-1)
4   dt=t/(nt-1)
5   P0=SharedArray(zeros(nx,nz))
6   P1=SharedArray(zeros(nx,nz))
7   PAUX=SharedArray(zeros(nx,nz))
8   jx=floor((xs-x0)/dx)
9   jz=floor((zs-z0)/dz)
10  for it=2*dt:dt:t
11    @parallel for iz=2:(nz-1)
12      for ix=2:(nx-1)
13        H=(P1[ix-1,iz]-2*P1[ix,iz]+P1[ix+1,iz])/(dx^2)+(P1[ix,iz-1]-2*P1[ix,iz]+P1[ix,iz+1])/(dz^2)
14        if ix==jx && iz==jz
15          P0[ix,iz]=2*P1[ix,iz]-P0[ix,iz]+(C[ix,iz]^2)*(dt^2)*H+exp(-(fpk^2)*(pi^2)*(it-that)^2)*(dt^2)
16        else
17          P0[ix,iz]=2*P1[ix,iz]-P0[ix,iz]+(C[ix,iz]^2)*(dt^2)*H
18        end
19      end
20    end
21    PAUX=P1
22    P1=P0
23    P0=PAUX
24  end
25  return P0
26 end

```

```

1 function FDMParallel2(nx,nz,nt,t,C,x0,xm,z0,zm,fpk,that,xs,zs,xs2,zs2)
2   dx=abs(xm-x0)/(nx-1)
3   dz=abs(zm-z0)/(nz-1)
4   dt=t/(nt-1)
5   P0=zeros(nx,nz)
6   P1=zeros(nx,nz)
7   jx=floor((xs-x0)/dx)
8   jz=floor((zs-z0)/dz)
9   jx2=floor((xs2-x0)/dx)
10  jz2=floor((zs2-z0)/dz)
11  for it=2*dt:dt:t
12    for iz=2:(nz-1)
13      for ix=2:(nx-1)
14        H=(P1[ix-1,iz]-2*P1[ix,iz]+P1[ix+1,iz])/(dx^2)+(P1[ix,iz-1]-2*P1[ix,iz]+P1[ix,iz+1])/(dz^2)
15        if ix==jx && iz==jz
16          P0[ix,iz]=2*P1[ix,iz]-P0[ix,iz]+(C[ix,iz]^2)*(dt^2)*H+exp(-(fpk^2)*(pi^2)*(it-that)^2)
17          *(dt^2)
18        elseif ix==jx2 && iz==jz2
19          P0[ix,iz]=2*P1[ix,iz]-P0[ix,iz]+(C[ix,iz]^2)*(dt^2)*H+exp(-(fpk^2)*(pi^2)*(it-that)^2)
20          *(dt^2)
21        else
22          P0[ix,iz]=2*P1[ix,iz]-P0[ix,iz]+(C[ix,iz]^2)*(dt^2)*H
23        end
24      end
25    end
26    PAUX=P1
27    P1=P0
28    P0=PAUX
29  end

```

donde los parámetros son los siguientes en este orden:

- nx = # de pedazos de la matriz en el eje x
- nz = # de pedazos de la matriz en el eje y
- nt = # de pedazos del tiempo
- t = tiempo
- C = matriz de velocidad del medio
- $x0$ = punto de la izquierda en el eje x
- xm = punto a la derecha del eje x
- $z0$ = punto abajo en el eje y
- zm = punto arriba en el eje y
- fpk = frecuencias pico
- $that$ = tiempo en que la fuente alcanza su máxima amplitud
- xs = coordenada e en el eje x donde se inicia la onda,
- zs = coordenada en el eje y donde inicia la onda
- $xs2$ = coordenada e en el eje x donde se inicia la 2^o onda,
- $zs2$ = coordenada en el eje y donde inicia la 2^o onda

4 EXPERIMENTOS

Se hizo una serie de experimentos para ver el comportamiento de las ondas, en las siguientes imágenes se verán representadas los parámetros de la siguiente manera:

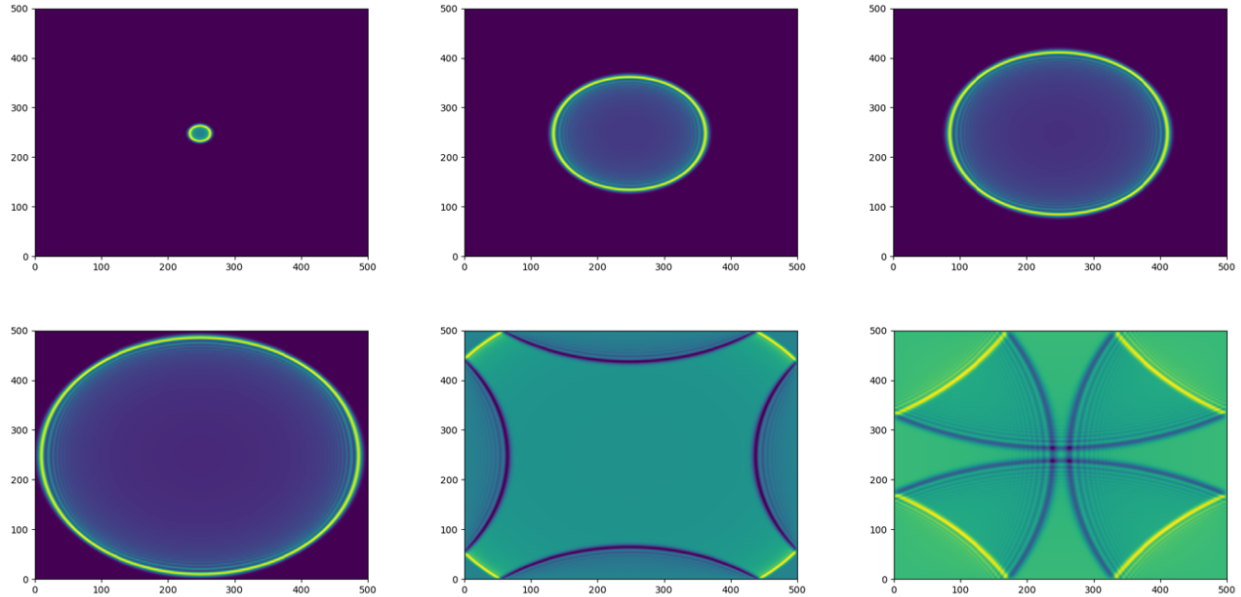


Figure 4.1: Simulación de una onda para diferentes tiempos, los parámetros son los siguientes: **(500, 500, 500, t, 500*ones(500,500) , -500, 500, -500, 500, 30, 1/15, 0, 0)** donde el tiempo t varía de 0 a 2.

4.1 PROGRAMACIÓN EN PARALELO

Cuando se hace un programa, el objetivo es que realice la tarea correctamente, pero cuando el programa es de uso intensivo, el tiempo es un factor clave. Por lo tanto optimizar se hace fundamental. Existen varios métodos, uno de ellos es la programación paralela; consistente en dividir una tarea en varias tareas mas pequeñas e independientes entre si, y ejecutarlas de forma simultánea. Además de que permite resolver problemas con gran cantidad de datos y reducir el tiempo de la solución

5 REFERENCIAS

LeVeque, J. Randall (2007). Finite Difference Methods for Ordinary and Partial Differential Equations: Steady-State and Time-Dependent Problems (1^a ed.). Washington: University of Washington

Shotts, W. (2016). The Linux Command Line (3^a ed.). Estados Unidos

Geodatos

Scitation

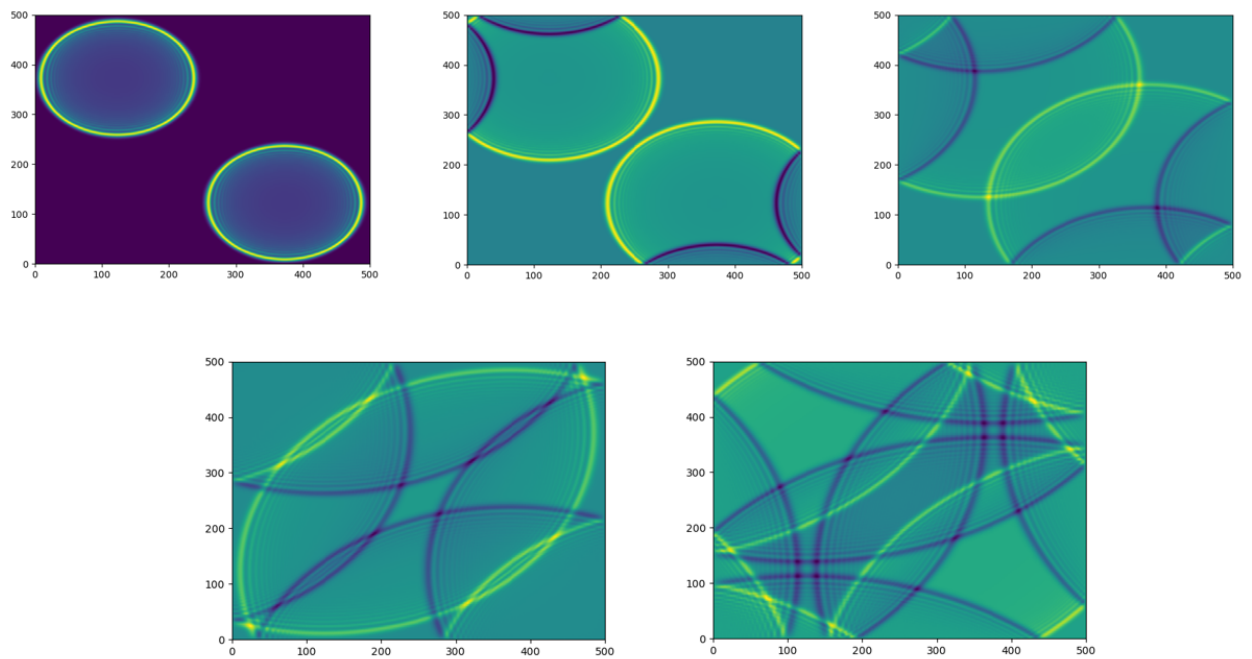


Figure 4.2: Simulación de dos ondas para diferentes tiempos, los parámetros son los siguientes: $(500, 500, 500, t, 500 \cdot \text{ones}(500, 500))$, $-500, 500, -500, 500, 30, 1/15, 0, 0, 1, 1)$ donde el tiempo t varía de 0 a 2.