Sistemas de ecuaciones lineales compatibles

Recordemos…

*Fases en la **resolución de un problema***

1. Definición del problema
2. Análisis del problema

Análisis de Requerimientos:
¿**Qué** es lo que hay que hacer?

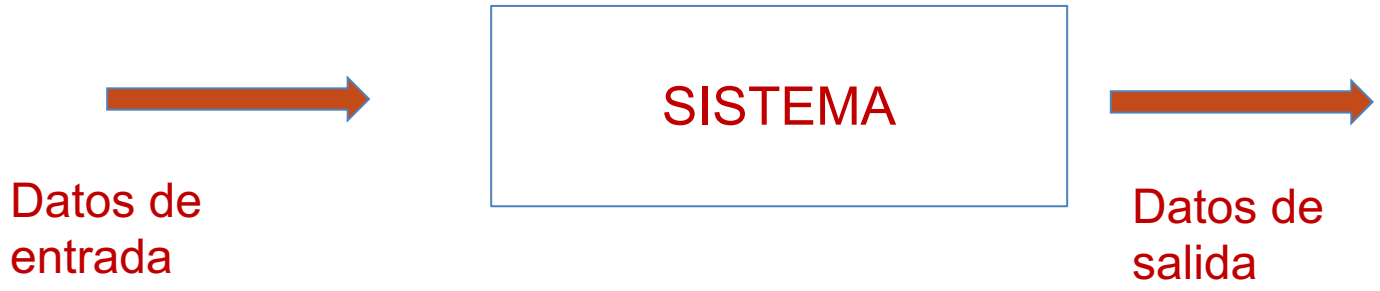
3. Diseño del algoritmo
4. Prueba del algoritmo

Diseño:
¿**Cómo** hacer lo especificado en la etapa de Análisis?

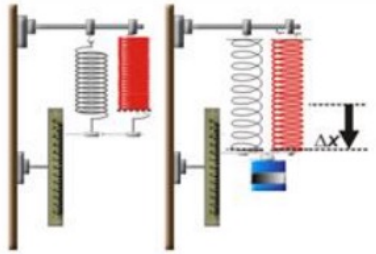
Algoritmo

5. Codificación del algoritmo en un lenguaje
6. Prueba y puesta a punto del programa
7. Documentación del programa

Programa



Problema: Sistema masa-resorte



Definición del problema

Se quiere caracterizar un sistema de una masa con dos resortes. Las fuerzas [N] actuantes se miden con un dinamómetro de 4 cifras significativas de precisión. Los desplazamientos [mm] se midieron con una regla común. El sistema de ecuaciones queda determinado por:

$$\begin{cases} 18.55k_0 + 5.86k_1 = 150.9 \\ 204.88k_0 + 104.31k_1 = 2000.0 \end{cases}$$

Problema: Sistema masa-resorte

Análisis del problema

Cifras significativas: El enunciado especifica que se requieren 4 cifras significativas, por lo tanto:

$$e_{s=0,5\ x\ 10^{2-4}=0.5\ x\ 10^{-2}}$$

$$e_{s=0.005}$$

Problema: Sistema masa-resorte

Análisis del problema

$$A * k = c$$

En forma matricial para un sistema de 2 ecuaciones por 2 incógnitas:

$$\begin{bmatrix} a_{00} & a_{01} \\ a_{10} & a_{11} \end{bmatrix} * \begin{bmatrix} k_0 \\ k_1 \end{bmatrix} = \begin{bmatrix} c_0 \\ c_1 \end{bmatrix}$$

A = Matriz Coeficientes; k = Arreglo Incógnitas; C = Arreglo constantes

$$\begin{bmatrix} 18.55 & 5.86 \\ 204.88 & 104.31 \end{bmatrix} * \begin{bmatrix} k_0 \\ k_1 \end{bmatrix} = \begin{bmatrix} 150.9 \\ 2000.0 \end{bmatrix}$$

Problema: Sistema masa-resorte

Análisis del problema

Ya tenemos el sistema de ecuaciones

Objetivo: Hallar valores de k_0 y k_1

Analizar la situación
- es lineal?
- de qué tipo es el sistema?
- tiene solución? (Teorema Rouché-Frobenius)

Problema: Sistema masa-resorte

Análisis del problema

Teorema Rouché-Frobenius:

Menciona que si el rango de la matriz de coeficientes (Mc) = rango de la matriz ampliada (Ma) = numero de incógnitas (Ni), entonces el sistema es compatible determinado, es decir existe una única solución, que gráficamente significa que existe un punto en el que se cruzan las rectas.

$$M_c = M_a = N_i$$
$$2 = 2 = 2$$

[link] https://matrix.reshish.com/es/rank.php

Problema: Sistema masa-resorte

Condicionamiento del sistema: Notar que el sistema esta mal condicionado. Por lo tanto, debemos utilizar todas las cifras significativas que podamos y además debemos escalar y pivotear para obtener una diagonal demandante

Dividir por el máximo de la fila (escalar fila):

$$\begin{bmatrix} 18.55/\mathbf{18.55} & 5.86/\mathbf{18.55} \\ 204.88/\mathbf{204.88} & 104.31/\mathbf{204.88} \end{bmatrix} * \begin{bmatrix} k_0 \\ k_1 \end{bmatrix} = \begin{bmatrix} 150.9/\mathbf{18.55} \\ 2000.0/\mathbf{204.88} \end{bmatrix}$$

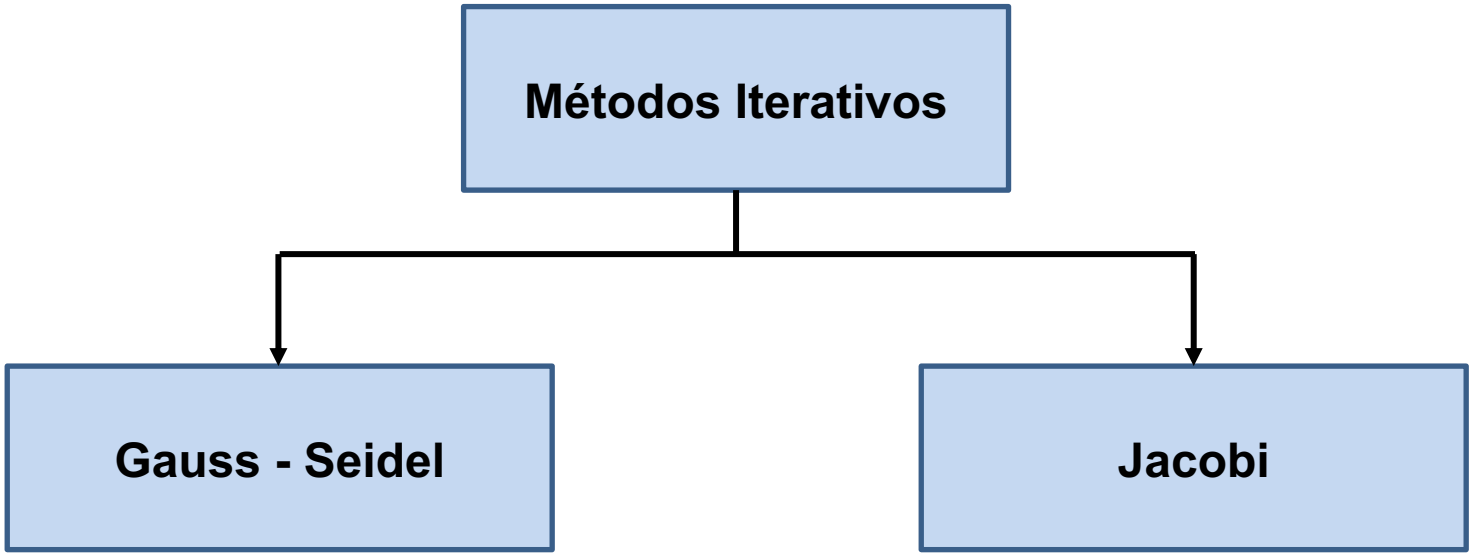
$$\begin{bmatrix} 1 & 0.3159 \\ 1 & 0.5091 \end{bmatrix} * \begin{bmatrix} k_0 \\ k_1 \end{bmatrix} = \begin{bmatrix} 8.1347 \\ 9.7618 \end{bmatrix}$$

En este sistema no podemos llegar a una diagonal dominante.

Problema: Sistema masa-resorte

Análisis del problema

Métodos Iterativos para resolver grandes Sistemas de ecuaciones lineales Compatibles



Los métodos Gauss-Seidel y Jacobi garantizan convergencia si la matriz es diagonal dominante.

Problema: Sistema masa-resorte

Análisis del problema

Jacobi

Cicla encontrando en cada iteración una solución al sistema (para esto usa en cada iteración los valores de las incógnitas calculados en la anterior iteración) hasta que se converja y obtener una solución aproximada a la verdadera que cumpla con la tolerancia de error previamente especificada

¿Hasta cuando se itera?

Gauss - Seidel

Gauss-Seidel a diferencia de Jacobi usa los valores recientemente obtenidos de las incógnitas para calcular los de la iteración actual

Problema: Sistema masa-resorte

Análisis del problema

¿Hasta cuando se itera?

$$\varepsilon_{a,i} = \frac{\left\| \mathbf{x}^{(i)} - \mathbf{x}^{(i-1)} \right\|}{\left\| \mathbf{x}^{(i)} \right\|} 100\% < \varepsilon_s$$

i: iteración actual.

i-1: iteración anterior.

$\left\| \mathbf{x}^{(i)} \right\|$: *norma 2 o Euclidea del vector x en iteración actual i.*

$\left\| \mathbf{x}^{(i)} - \mathbf{x}^{(i-1)} \right\|$: *norma 2 o Euclidea de la diferencia entre los vectores aproximados en las iteraciones actual y anterior.*

Problema: Sistema masa-resorte

Análisis del problema

Datos de entrada?	Datos del sistema de ecuaciones
	Error esperado
Datos de salida?	Variables incógnitas
	Iteraciones
	Error
Proceso?	Método JACOBI o Gauss Seidel

Problema: Sistema masa-resorte

$$A * k = c$$

En forma matricial para un sistema de de ecuaciones por dos incógnitas:

$$\begin{bmatrix} a_{00} & a_{01} \\ a_{10} & a_{11} \end{bmatrix} * \begin{bmatrix} k_0 \\ k_1 \end{bmatrix} = \begin{bmatrix} c_0 \\ c_1 \end{bmatrix}$$

A = matriz coeficientes

k = arreglo incógnitas

C = arreglo constantes

Problema: Sistema masa-resorte

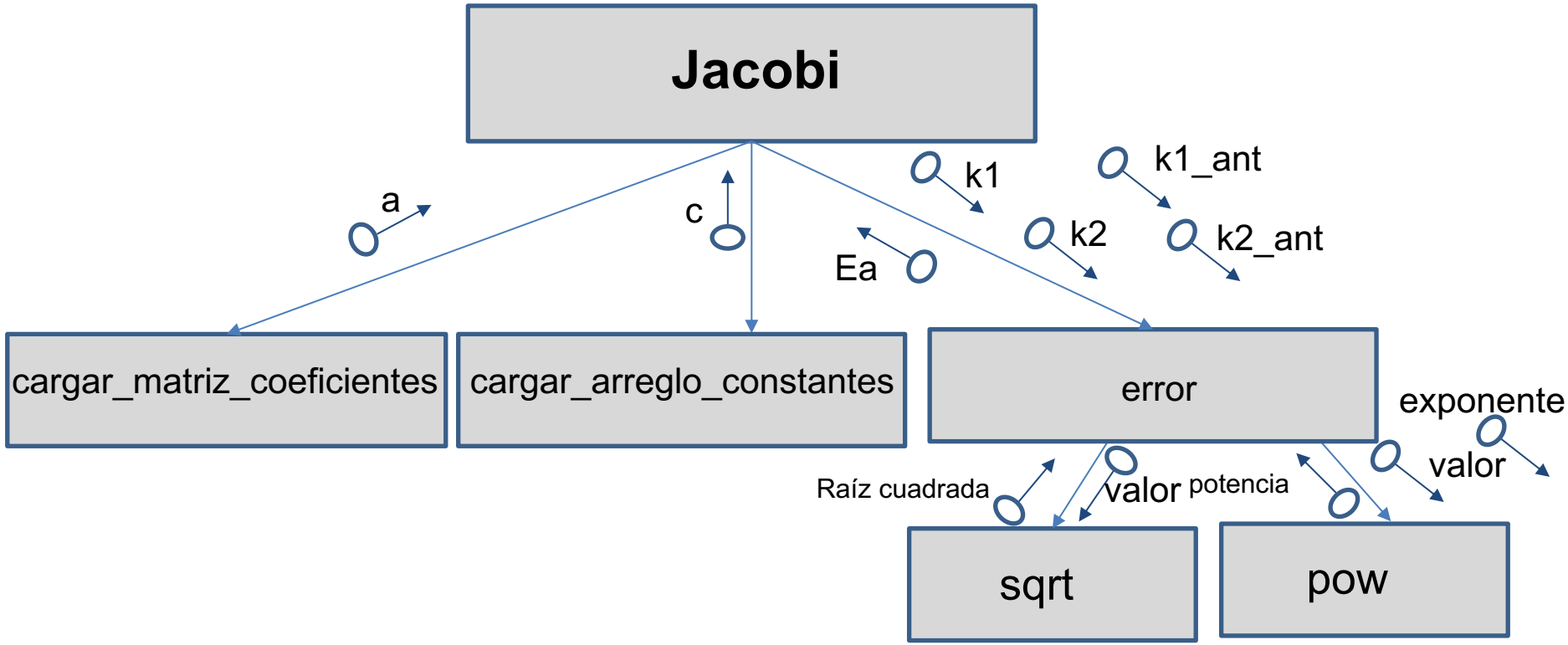
Algoritmo - Jacobi

a = cargar matriz coeficientes dados por el usuario
c = cargar arreglo constantes dados por el usuario
k = inicializar arregloincognitas en cero
es = cargar error brindado por el usuario
ea=es + 1
i = 1
Mientras Es<= Ea
  k1= (c[0] – a[0][1] * k[1])/a[0][0]
  k2= (c[1] – a[1][0] * k[0])/a[1][1]
  i+=1
  Si i>=2
    Ea= sqrt(suma(exp(k1-k[0],2), exp(k2-k[1],2))) / sqrt(suma(exp(k1,2), exp(k2,2))) *100
  k[0]=k1
  k[1]=k2
Mostrar por pantalla k1, k2, i, Ea

Problema: Sistema masa-resorte

Diseño del algoritmo

Diagrama de Estructuras – Jacobi



Problema: Sistema masa-resorte

Algoritmo – Gauss Seidel

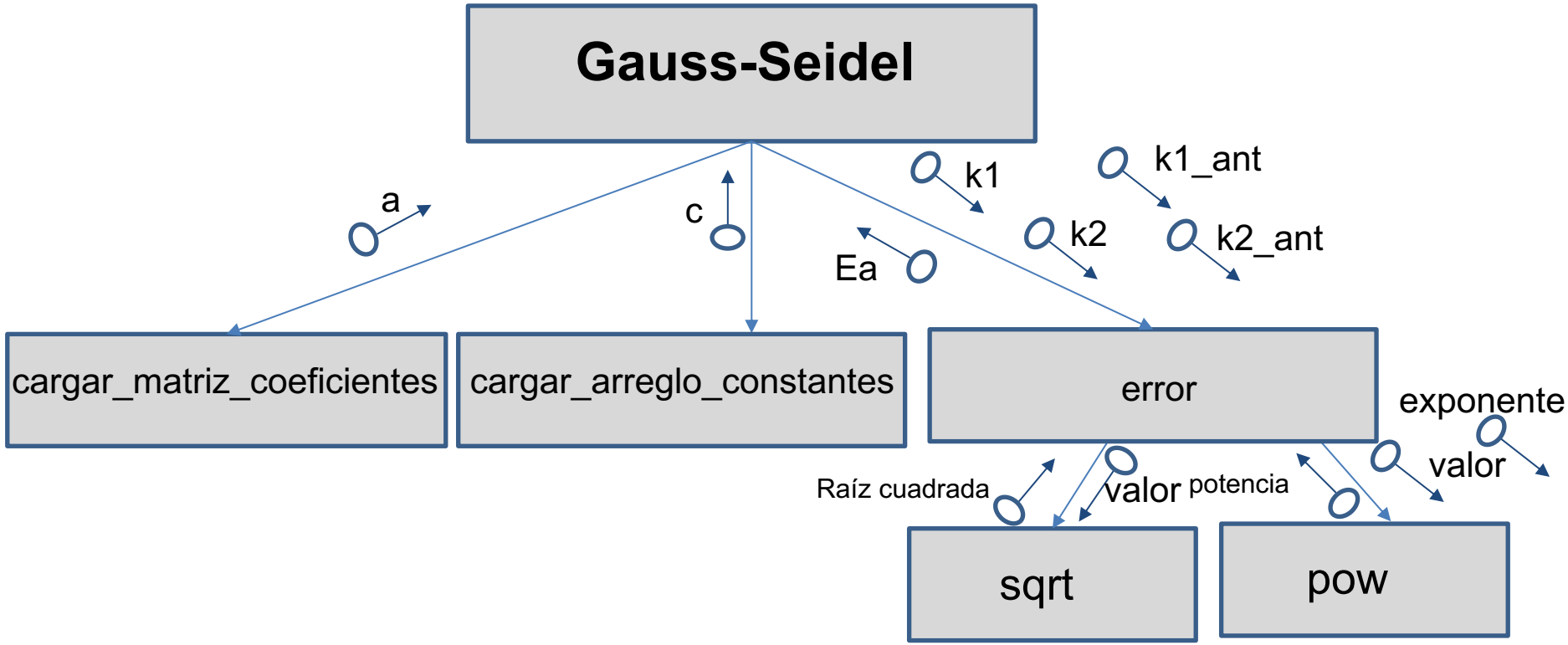
a = cargar matriz coeficientes dados por el usuario
c = cargar arreglo constantes dados por el usuario
k = inicializar arregloincognitas en cero
es = cargar error brindado por el usuario
ea=es + 1
i = 1
Mientras Es<= Ea
  k1= (c[0] – a[0][1] * k[1])/a[0][0]
  k2= (c[1] – a[1][0] * k1)/a[1][1]
  i+=1
  Si i>=2
    Ea= sqrt(suma(exp(k1-k[0],2), exp(k2-k[1],2))) / sqrt(suma(exp(k1,2), exp(k2,2))) *100
  k[0]=k1
  k[1]=k2
Mostrar por pantalla k1, k2, i, Ea

Modificación!

Problema: Sistema masa-resorte

Diseño del algoritmo

Diagrama de Estructuras – Gauss Seidel



Notar que el DE son iguales entre Jacobi y Gauss-Seidel

Codificación del algoritmo en un lenguaje

Función Cargar Matriz Coeficientes

```python
import math as m

def cargar_matriz_coeficientes():
    a = [[0, 0], [0, 0]]
    for f in range(len(a)):
        for c in range(len(a[0])):
            a[f][c] = float(input('Ingrese el coheficiente a' + str(f+1) + str(c+1)))
    #a[0][0] = 1
    #a[0][1] = 0.3159
    #a[1][0] = 1
    #a[1][1] = 0.5091
    return a
```

Función Cargar Arreglo Constantes

```
13
14  def cargar_arreglo_constantes():
15      c = [0, 0]
16      for i in range(len(c)):
17          c[i] = float(input('Ingrese la constante c' + str(i+1)))
18      #c[0] = 8.1347
19      #c[1] = 9.7618
20      return c
21
```

Codificación del algoritmo en un lenguaje

Función calcular error

```
22    def error(k1, k1_ant, k2, k2_ant):
23        return m.sqrt(m.pow(k1-k1_ant,2) + m.pow(k2-k2_ant,2)) / m.sqrt(m.pow(k1,2)+m.pow(k2,2))*100
```

Codificación del algoritmo en un lenguaje

Principal: Jacobi

```
25  a = cargar_matriz_coeficientes()
26  print('Matriz coeficientes: ')
27  print(a)
28  c = cargar_arreglo_constantes()
29  print('Arreglo constantes: ')
30  print(c)
31  k = [0, 0]
32  # Error
33  es = float(input('Ingrese el error: '))
34  i = 1
35  ea=es + 1
36  while (es <= ea):
37      k1= (c[0]-a[0][1]* k[1])/a[0][0]
38      k2= (c[1]-a[1][0]* k[0])/a[1][1]
39      i+=1
40      if (i >= 2):
41          ea = error(k1,k[0],k2,k[1])
42      k[0]=k1
43      k[1]=k2
44
45  print ('Contante k1: ' + str(k1))
46  print ('Contante k2: ' + str(k2))
47  print ('Iteraciones: ' + str(i))
48  print('Error: ' + str(ea))
```

Codificación del algoritmo en un lenguaje

Principal: Gauss-Seidel

```
25  a = cargar_matriz_coeficientes()
26  print('Matriz coeficientes: ')
27  print(a)
28  c = cargar_arreglo_constantes()
29  print('Arreglo constantes: ')
30  print(c)
31  k = [0, 0]
32  # Error
33  es = float(input('Ingrese el error: '))
34  i = 1
35  ea=es + 1
36  while (es <= ea):
37      k1= (c[0]-a[0][1]* k[1])/a[0][0]
38      k2= (c[1]-a[1][0]* k1)/a[1][1]
39      i+=1
40      if (i >= 2):
41          ea = error(k1,k[0],k2,k[1])
42      k[0]=k1
43      k[1]=k2
44
45  print ('Contante k1: ' + str(k1))
46  print ('Contante k2: ' + str(k2))
47  print ('Iteraciones: ' + str(i))
48  print('Error: ' + str(ea))
```

Problema: Sistema masa-resorte

Resultados

Para finalizar, recordar mostrar los resultados en forma contextualizada de acuerdo a la definición y análisis del problema.

Se ejecuto en base a un error: **0.001**

Jacobi	Gauss-Seidel
Contante k1: 5.474240907060613 **Contante k2**: 8.42184669272541 **Iteraciones**: 64 **Error**: 7.79601201562504 6e-05	**Contante k1**: 5.474246653868315 **Contante k2**: 8.421829397233715 **Iteraciones**: 29 **Error**: 9.0547796781821 4e-05

$$18.55k_0 + 5.86k_1 = 150.9$$
$$204.88k_0 + 104.31k_1 = 2000.0$$

Problema: Sistema masa-resorte

Evaluación

Jacobi	Gauss-Seidel
Contante k1: 5.474240907060613 **Contante k2**: 8.42184669272541 **Iteraciones**: 64 **Error**: 7.79601201562504 6e-05	**Contante k1**: 5.474246653868315 **Contante k2**: 8.421829397233715 **Iteraciones**: 29 **Error**: 9.0547796781821 4e-05

$$18.55k_0 + 5.86k_1 = 150.9$$
$$204.88k_0 + 104.31k_1 = 2000.0$$

$$18.55*5.474 + 5.86*8.421 = 150.894$$
$$204.88*5.474 + 104.31*8.421 = 1999.999$$