Universidad Autónoma de Querétaro

FACULTAD DE INGENIERÍA

Logouaq.png

Tarea 7: Descomposición de Cholesky y Gauss-Seidel

Análisis numérico

Autor:
J.A. Salinas Sánchez
Marzo2022

Índice general

Capítulo 1

Introducción

Capítulo 2

Metodología

2.1. Códigos

2.1.1. Cholesky

```python
#coding:utf8

import numpy as np

def main():
    uk=int(input('Introduzca el n mero de inc gnitas :'))
    m1=np.zeros((uk,uk+1))
    for i in range(uk):
        for j in range(uk+1):
            if(j==uk):
                m1[i][j]=float(input(f'Introduzca el coeficiente de T.I.: '))
            else:
                m1[i][j]=float(input(f'Introduzca el coeficiente de x{j}: '))

    Cholesky(m1,uk)

def Cholesky(m,uk):
    lower=np.zeros((uk,uk+1))
    lowerT=np.zeros((uk,uk+1))
    temp=np.zeros((uk,1))
    for i in range(uk):
        lower[i][uk]=m[i][uk]
        for j in range(uk):
            a=0
            if(i==j):
                for k in range(j):
                    a+=lower[j][k]**2
                lower[j][j]=(m[j][j]-a)**(1/2)
            else:
                for k in range(j):
                    a+=lower[i][k]*lower[j][k]
                if (lower[j][j]>0):
                    lower[i][j]=(m[i][j]-a)/lower[j][j]

    for i in range(uk):
        for j in range(uk):
            lowerT[i][j]=lower[j][i]

    print(lower)
    temp=Gauss(lower,temp,uk)
```

```
    for j in range(uk):
        lowerT[j][uk]=temp[j]

    print(lowerT)
    temp=Gauss(lowerT,temp,uk)
    print(temp)

def Gauss(m1,tm,uk):
    i=0
    j=0
    k=0

    for i in range(uk):
        for j in range(i+1, uk):
            ratio = m1[j][i]/m1[i][i]
            for k in range(uk+1):
                m1[j][k] = m1[j][k]-ratio*m1[i][k]

    tm[uk-1] = m1[uk-1][uk]/m1[uk-1][uk-1]

    for i in range(uk-2,-1,-1):
        tm[i] = m1[i][uk]
        for j in range(i+1,uk):
            tm[i] = tm[i] - m1[i][j]*tm[j]
        tm[i] = tm[i]/m1[i][i]

    return (tm)


if __name__=='__main__':
    main()
```

2.1.2. NR multivariable

```
#coding:utf8

import numpy as np
import sympy

x=sympy.symbols('x')
y=sympy.symbols('y')
x1=sympy.symbols('x1')
x2=sympy.symbols('x2')

def main():
    func1=str(input('Introduzca su ecuaci n 1: '))
    func2=str(input('Introduzca su ecuaci n 2: '))
    e=float(input('Introduzca su tolerancia :'))
    xi=float(input('Introduzca el valor inicial de x/x1: '))
    yi=float(input('Introduzca el valor inicial de y/x2: '))
    print(NRNL(func1,func2,e,xi,yi))

def NRNL(f1,f2,ea,xi,yi):

    xr=xi
    yr=yi
    u=getfunction(f1)
    v=getfunction(f2)
    dux=sympy.diff(u,x)
    duy=sympy.diff(u,y)
    dvx=sympy.diff(v,x)
    dvy=sympy.diff(v,y)
    fNRx=x-(u*dvy-v*duy)/(dux*dvy-duy*dvx)
```

```
    fNRy=y-(v*dux-u*dvx)/(dux*dvy-duy*dvx)
    e1=100
    e2=100
    respuestas=[]

    while(e1>=ea and e2>=ea):
        temp1=xr
        temp2=yr
        xr=fNRx.evalf(subs={x:temp1,y:temp2})
        yr=fNRy.evalf(subs={x:temp1,y:temp2})
        if(xr!=0):
            e1=abs((xr-temp1)/xr)*100
        if(yr!=0):
            e2=abs((yr-temp2)/yr)*100


    respuestas.append(xr)
    respuestas.append(yr)

    return(respuestas)



def getfunction(f):
    global x
    global y
    return sympy.sympify(f)

if __name__==’__main__’:
    main()
```

2.1.3. Gauss-Seidel

```
#coding:utf8

import numpy as np

def main():
    uk=int(input(’Introduzca el n mero de inc gnitas :’))
    m1=np.zeros((uk,uk+1))
    for i in range(uk):
        for j in range(uk+1):
            if(j==uk):
                m1[i][j]=float(input(f’Introduzca el coeficiente de T.I.: ’))
            else:
                m1[i][j]=float(input(f’Introduzca el coeficiente de x{j}: ’))

    e=float(input(’Introduzca su n mero de iteraciones: ’))
    print(Gseidel(m1,e,uk))

def Gseidel(m,e,uk):
    ea=0
    temp = np.zeros(uk)
    while(ea>=e):
        temp=seidel(m,temp,uk)
        ea+=1

def seidel(m,x,uk):

    for j in range(0, n):
        d = m[j][uk]
```

```
        for i in range(0, n):
            if(j != i):
                d-=m[j][i] * x[i]
        x[j] = d / m[j][j]

    return x

if __name__=='__main__':
    main()
```

2.2. Ejercicios

2.2.1. Chapra

11.3

Determine la matriz inversa del ejemplo 11.1 con base en la descomposición LU y los vectores unitarios:

$$\begin{pmatrix} 0{,}8 & -0{,}4 & 0 \\ -0{,}4 & 0{,}8 & -0{,}4 \\ 0 & -0{,}4 & 0{,}8 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \tag{2.1}$$

$$= \begin{pmatrix} 41 \\ 25 \\ 105 \end{pmatrix} \tag{2.2}$$

11.5

Haga los mismos cálculos que en el ejemplo 11.2, pero para el sistema simétrico que sigue:

$$\begin{pmatrix} 6 & 15 & 55 \\ 15 & 55 & 255 \\ 55 & 255 & 979 \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix} \tag{2.3}$$

$$= \begin{pmatrix} 152{,}6 \\ 585{,}6 \\ 2488{,}8 \end{pmatrix} \tag{2.4}$$

11.7

Calcule la descomposición de Cholesky de:

$$\begin{pmatrix} 9 & 0 & 0 \\ 0 & 25 & 0 \\ 0 & 0 & 4 \end{pmatrix} \tag{2.5}$$

11.13

Use el método de Gauss-Seidel (a) sin relajación y (b) con relajación (λ = 1.2), para resolver el sistema siguiente para una tolerancia de es = 5 %. Si es necesario, reacomode las ecuaciones para lograr convergencia.

$$\begin{pmatrix} 2 & -6 & -1 = -38 \\ -3 & -1 & 7 = -34 \\ -8 & 1 & -2 = -20 \end{pmatrix} \tag{2.6}$$

2.2.2. Kiusalaas

23

Determine the coordinates of the two points where the circles $(x2)^2 + y^2 = 4$ and $x^2 + (y3)^2 = 4$ intersect. Start by estimating the locations of the points from a sketch of the circles, and then use the Newton-Raphson method to compute the coordinates.

24

Las ecuaciones:

$$\sin x + 3\cos x - 2 = 0 \tag{2.7}$$

$$\cos x - \sin y + 0{,}2 = 0 \tag{2.8}$$

Capítulo 3

Conclusión