PROYECTO FINAL DE ANÁLISIS NÚMERICO II

RAFAEL ANÍBAL CALDERÓN* & ESTELY DOLORES MÉNDEZ [1]

18 de noviembre de 2016

ÍNDICE

ABSTRACT

En este articulo se presenta el código del método de diferencias finitas para la ecuación de Poisson, el algoritmo esta programado en el lenguaje de computación técnica MATLAB, se toma de base el algoritmo presentado en el libro de Burden, sección 12.1, algoritmo 12.1, para ver detalles del libro revisar referencias al final del articulo. Vale aclarar que el resultado del código es una matriz donde se presentan los valores para las aproximaciones de la solución $u(x, y)$ y el número de iteraciones necesarias para alcanzar la tolerancia deseada.

* *Escuela de matemática, Universidad de El Salvador, Ciudad Universitaria, San Salvador*
[1] *Análisis Númerico II, Docente: Msc. Carlos Gámez*

1 APROXIMACIÓN DE LA ECUACIÓN DE POISSON

Consideramos la ecuación diferencial de segundo orden elíptica conocida como la ecuación de Poisson

$$\frac{\partial^2 u}{\partial x^2}(x,y) + \frac{\partial^2 u}{\partial y^2}(x,y) = f(x,y) \tag{1.1}$$

Para $R = \{(x,y) \mid a < x < b,\ c < y < d\}$, con la condición que $u(x,y) = g(x,y)$ para $(x,y) \in S$, donde S denota los limites de R.

La idea es utilizar el Método de diferencias finitas que consiste en

$$2\left[\left(\frac{h}{k}\right)^2 + 1\right]\omega_{ij} - (\omega_{i+1j} + \omega_{i-1j}) - \left(\frac{h}{k}\right)^2(\omega_{i,j+1} + \omega_{ij-1}) = -h^2 f(x_i, y_j) \tag{1.2}$$

Para cada $i = 1, 2, \ldots, n-1$ y para $j = 1, 2, \ldots, m-1$ y con las condiciones

$$\omega_{oj} = g(x_0, y_j) \quad y \quad \omega_{nj} = g(x_n, y_j) \quad \text{para cada } j = 0, 1, \ldots m \tag{1.3}$$

$$\omega_{i0} = g(x_i, y_0) \quad y \quad \omega_{im} = g(x_i, y_m) \quad \text{para cada } i = 1, \ldots n-1 \tag{1.4}$$

En la aplicación de este método hay que resolver un sistema que es siempre de dimensión mayor que 3, y es por ello que el algoritmo utiliza el método de Gauss-Sidel.

1.1 Ejemplo de ejecución

Utilizando el Método de Diferencias Finitas de Poisson con $n = 6$, $m = 5$ y una tolerancia de 10^{-10} y un máximo de 100 iteraciones, para aproximar la solución de:

$$\frac{\partial^2 u}{\partial x^2}(x,y) + \frac{\partial^2 u}{\partial y^2}(x,y) = xe^y, \quad 0 < x < 2, \quad 0 < y < 1$$

con las condiciones iniciales

$$u(0,y) = 0, \quad u(2,y) = 2e^y, \quad 0 \leqslant y \leqslant 1$$

$$u(x,0) = x, \quad u(x,1) = ex, \quad 0 \leqslant x \leqslant 2$$

Implementando el programa creado para resolver las ecuaciones de Poisson con el método de diferencias finitas, se creo un programa llamado fung.m que hace las veces de la función g que se utiliza en el método, este scrip tiene los datos iniciales específicamente para las condiciones iniciales del ejemplo 2, y por tanto para la ejecución en MATLAB se tiene lo siguiente:

INPUT:
- $f = xe^y$
- $a = 0$ y $b = 2$
- $c = 0$ y $b = 1$
- $m = 5$ y $n = 6$
- $tol = 1e - 10$
- $N = 100$

OUTPUT:
- Tabla con las aproximaciones para u(x,y) y los puntos x & y.
- l: número de iteraciones requeridas para obtener la solución.

```
>> f = @(x,y)x*exp(y);
>> a=0; b=2; c=0; d=1; m=5; n=6; tol=1e-10; N=100;
>> PEFD(f,a,b,c,d,m,n,tol,N)
```

Los resultados obtenidos se presentan a continuación:

```
   i          j          xi         xj        w(i,j)
1.0000     1.0000     0.3333     0.2000     0.4073
1.0000     2.0000     0.3333     0.4000     0.4975
1.0000     3.0000     0.3333     0.6000     0.6076
1.0000     4.0000     0.3333     0.8000     0.7420

2.0000     1.0000     0.6667     0.2000     0.8145
2.0000     2.0000     0.6667     0.4000     0.9950
2.0000     3.0000     0.6667     0.6000     1.2152
2.0000     4.0000     0.6667     0.8000     1.4840

3.0000     1.0000     1.0000     0.2000     1.2218
3.0000     2.0000     1.0000     0.4000     1.4924
3.0000     3.0000     1.0000     0.6000     1.8227
3.0000     4.0000     1.0000     0.8000     2.2260

4.0000     1.0000     1.3333     0.2000     1.6290
4.0000     2.0000     1.3333     0.4000     1.9898
4.0000     3.0000     1.3333     0.6000     2.4302
4.0000     4.0000     1.3333     0.8000     2.9679

5.0000     1.0000     1.6667     0.2000     2.0360
5.0000     2.0000     1.6667     0.4000     2.4870
5.0000     3.0000     1.6667     0.6000     3.0375
5.0000     4.0000     1.6667     0.8000     3.7097


Y el numero de iteraciones fue: 61
```

En la última colunma de los resultados, se puede encontrar la aproximación w para $u(x, y)$.
Y al pie de la tabla de resultados, se encuentra el número de iteraciones necesarias.

REFERENCIAS

[1] Richard L. Burden, J. Douglas Faires, *Numerical Analysis*, NINTH EDITION, Brooks/Cole, Cengage Learning, 2011.