

Método de diferencias finitas para ecuaciones diferenciales parciales elípticas

(Parte II)

Métodos numéricos para sistemas lineales

- Solución numérica de EDPs requiere resolver sistemas de ecuaciones lineales
- Matrices pueden ser grandes y sparse (5 o 6 elementos distintos de cero por fila)
- Qué métodos existen?
 - Exactos
 - Iterativos

Métodos numéricos para sistemas lineales

- Métodos exactos (Ejemplo: Gauss):
 - Resuelve el sistema en forma “exacta” (al usar punto flotante siempre hay errores)
 - Hay métodos (y software) especiales para manejar matrices sparse de manera eficiente. A pesar de esto requieren mucho espacio de almacenamiento
- Métodos iterativos:
 - Requieren menos espacio de almacenamiento
 - Guardan solo valores distintos de cero de la matriz
 - Requieren de una solución inicial del sistema para comenzar
 - Criterio de detención es complicado de manejar para asegurar precisión en la solución aproximada obtenida
- (capítulo 3, Mathews-Fink, apuntes MC Rivara)

Métodos directos

- Método de Gauss: La matriz ampliada de $AX=B$ es la siguiente:

$$[A|B] = \left[\begin{array}{cccc|c} a_{11} & a_{12} & \cdots & a_{1N} & b_1 \\ a_{21} & a_{22} & \cdots & a_{2N} & b_2 \\ \vdots & \vdots & & \vdots & \vdots \\ a_{N1} & a_{N2} & \cdots & a_{NN} & b_N \end{array} \right].$$

- Esta se resuelve usando operaciones elementales sobre sus filas (las variables x_k sirven para marcar el sitio de los coeficientes y pueden ser omitidas hasta el final de los cálculos)

Métodos directos: método de Gauss

- Operaciones elementales con las filas, pivotes y multiplicadores

Teorema 3.8 (Operaciones elementales con las filas). Cualquiera de las siguientes operaciones aplicada a la matriz ampliada (7) produce un sistema lineal equivalente.

- (8) Intercambio: El orden de las filas puede cambiarse.
- (9) Escalado: Multiplicar una fila por una constante no nula.
- (10) Sustitución: Una fila puede ser reemplazada por la suma de esa fila más un múltiplo de cualquier otra fila; o sea,
$$\text{fila}_r = \text{fila}_r - m_{rq} \times \text{fila}_q.$$

Definición 3.3 (Pivotes y multiplicadores). El elemento a_{qq} de la matriz de los coeficientes en el paso $q + 1$ que se usará en la eliminación de a_{rq} , para $r = q + 1, q + 2, \dots, N$, se llama q -ésimo **pivote** y la fila q -ésima se llama **fila pivote**. Los números $m_{rq} = a_{rq}/a_{qq}$ ($r = q + 1, q + 2, \dots, N$) por los que se multiplica la fila pivote para restarla de las correspondientes filas posteriores se llaman **multiplicadores** de la eliminación. ▲

Métodos directos: método de Gauss

Ejemplo

- Sistema de ecuaciones

$$\begin{aligned}x_1 + 2x_2 + x_3 + 4x_4 &= 13 \\2x_1 + 0x_2 + 4x_3 + 3x_4 &= 28 \\4x_1 + 2x_2 + 2x_3 + x_4 &= 20 \\-3x_1 + x_2 + 3x_3 + 2x_4 &= 6.\end{aligned}$$

- Matriz ampliada

$$\begin{array}{l} \text{pivote} \rightarrow \\ m_{21} = 2 \\ m_{31} = 4 \\ m_{41} = -3 \end{array} \left[\begin{array}{cccc|c} \underline{1} & 2 & 1 & 4 & 13 \\ 2 & 0 & 4 & 3 & 28 \\ 4 & 2 & 2 & 1 & 20 \\ -3 & 1 & 3 & 2 & 6 \end{array} \right].$$

Métodos directos: método de Gauss

Ejemplo

- Elementos primera columna eliminados

$$\begin{array}{l} \text{pivote} \rightarrow \\ m_{32} = 1.5 \\ m_{42} = -1.75 \end{array} \left[\begin{array}{cccc|c} 1 & 2 & 1 & 4 & 13 \\ 0 & \underline{-4} & 2 & -5 & 2 \\ 0 & -6 & -2 & -15 & -32 \\ 0 & 7 & 6 & 14 & 45 \end{array} \right]$$

- Elementos segunda columna eliminados

$$\begin{array}{l} \text{pivote} \rightarrow \\ m_{43} = -1.9 \end{array} \left[\begin{array}{cccc|c} 1 & 2 & 1 & 4 & 13 \\ 0 & -4 & 2 & -5 & 2 \\ 0 & 0 & \underline{-5} & -7.5 & -35 \\ 0 & 0 & 9.5 & 5.25 & 48.5 \end{array} \right]$$

Métodos directos: método de Gauss

Ejemplo

- Elementos tercera columna eliminados

$$\left[\begin{array}{cccc|c} 1 & 2 & 1 & 4 & 13 \\ 0 & -4 & 2 & -5 & 2 \\ 0 & 0 & -5 & -7.5 & -35 \\ 0 & 0 & 0 & -9 & -18 \end{array} \right].$$

- Resultado: sistema triangular superior
 - Se resuelve de abajo hacia arriba

- $$-9x_4 = -18$$

Costo computacional del método de Gauss

- Para una matriz de $N \times N$ es de $O(N^3)$ (este es el orden del número de operaciones divisiones, multiplicaciones y restas que realiza el algoritmo)
 - Propuesto: contarlas!
- Qué problemas trae esto?
 - Muy lento para grillas no muy grandes y usa harto espacio
 - Por ejemplo: un problema de Dirichlet requiere la resolución de un sistema de $(n-2)(m-2)$ ecuaciones
 - Si la grilla de R es de 10×10 cuadrados, cuántas ecuaciones e incógnitas son?
Si R es de 100×100 ?
 - De qué orden es el número de operaciones que realiza el algoritmo de Gauss?
 - Cuanto se demora sobre un computador que tiene 2GHz (suponiendo que tiene la memoria suficiente)? 2GHz significa $2 \cdot 10^9$ tics por segundo. Supongamos que 1 operación de punto flotante necesita 4 tics \Rightarrow 500 megaflops por segundo (operaciones de punto flotante).

Métodos iterativos: Jacobi

- Sistema de ecuaciones
- Solución: (2,4,3)

$$\begin{aligned}4x - y + z &= 7 \\4x - 8y + z &= -21 \\-2x + y + 5z &= 15.\end{aligned}$$

- Estas ecuaciones se pueden escribir como:

$$\begin{aligned}x &= \frac{7 + y - z}{4} \\y &= \frac{21 + 4x + z}{8} \\z &= \frac{15 + 2x - y}{5},\end{aligned}$$

- Qué sugiere esto?

Métodos iterativos: Jacobi

- Sugiere un método iterativo:

$$\begin{aligned}x_{k+1} &= \frac{7 + y_k - z_k}{4} \\y_{k+1} &= \frac{21 + 4x_k + z_k}{8} \\z_{k+1} &= \frac{15 + 2x_k - y_k}{5}.\end{aligned}$$

- Qué necesitamos? Valores iniciales (x_0, y_0, z_0) para calcular (x_1, y_1, z_1)

- Comencemos con $P_0 = (x_0, y_0, z_0) = (1, 2, 2)$

- Parece converger a la solución

$$\begin{aligned}x_1 &= \frac{7 + 2 - 2}{4} = 1.75 \\y_1 &= \frac{21 + 4 + 2}{8} = 3.375 \\z_1 &= \frac{15 + 2 - 2}{5} = 3.00.\end{aligned}$$

Métodos iterativos: Jacobi

Este proceso se conoce como: Método de iteración de Jacobi

- No sirve para resolver todos los sistemas lineales
- En el ejemplo: 19 iteraciones

Tabla 3.2 Convergencia del método iterativo de Jacobi para el sistema (1).

k	x_k	y_k	z_k
0	1.0	2.0	2.0
1	1.75	3.375	3.0
2	1.84375	3.875	3.025
3	1.9625	3.925	2.9625
4	1.99062500	3.97656250	3.00000000
5	1.99414063	3.99531250	3.00093750
⋮	⋮	⋮	⋮
15	1.99999993	3.99999985	2.99999993
⋮	⋮	⋮	⋮
19	2.00000000	4.00000000	3.00000000

Método iterativo de Jacobi: Ejemplo de divergencia

- El mismo sistema de ecuaciones reordenado:

$$\begin{aligned} -2x + y + 5z &= 15 \\ 4x - 8y + z &= -21 \\ 4x - y + z &= 7. \end{aligned}$$

- Si escribimos estas ecuaciones como:

$$\begin{aligned} x &= \frac{-15 + y + 5z}{3} \\ y &= \frac{21 + 4x + z}{8} \\ z &= 7 - 4x + y, \end{aligned}$$

Método iterativo de Jacobi: Ejemplo de divergencia

- Así, el método iterativo de Jacobi queda:

$$x_{k+1} = \frac{-15 + y_k + 5z_k}{3}$$

$$y_{k+1} = \frac{21 + 4x_k + z_k}{8}$$

$$z_{k+1} = 7 - 4x_k + y_k.$$

- Y partimos con el mismo punto inicial

$$P_0 = (x_0, y_0, z_0) = (1, 2, 2)$$

- El punto (1.5, 3.375, 5) está más alejado de la solución que el punto inicial de la solución.

$$x_1 = \frac{-15 + 2 + 10}{2} = -1.5$$

$$y_1 = \frac{21 + 4 + 2}{8} = 3.375$$

$$z_1 = 7 - 4 + 2 = 5.00.$$

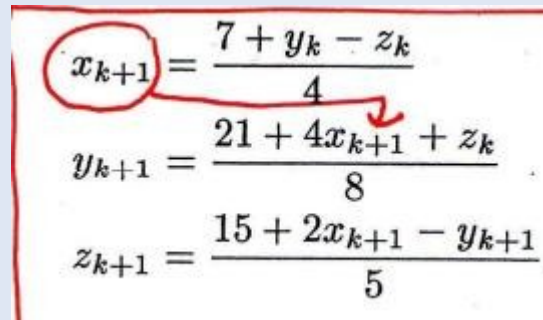
Método iterativo de Jacobi: Ejemplo de divergencia

- En este ejemplo, el método diverge!

k	x_k	y_k	z_k
0	1.0	2.0	2.0
1	-1.5	3.375	5.0
2	6.6875	2.5	16.375
3	34.6875	8.015625	-17.25
4	-46.617188	17.8125	-123.73438
5	-307.929688	-36.150391	211.28125
6	502.62793	-124.929688	1202.56836
\vdots	\vdots	\vdots	\vdots

Método iterativo de Gauss-Seidel

- Podemos algunas veces acelerar la convergencia?
 - Basado en el método de Jacobi, la idea es que cada vez que tenemos un nuevo valor, usamos ese para obtener el valor de la siguiente ecuación
- Lo aplicamos al ejemplo anterior:


$$\begin{aligned}x_{k+1} &= \frac{7 + y_k - z_k}{4} \\y_{k+1} &= \frac{21 + 4x_{k+1} + z_k}{8} \\z_{k+1} &= \frac{15 + 2x_{k+1} - y_{k+1}}{5}\end{aligned}$$

- El nuevo punto está más cerca de la solución que el punto obtenido con Jacobi

$$x_1 = \frac{7 + 2 - 2}{4} = 1.75.$$

$$y_1 = \frac{21 + 4(1.75) + 2}{8} = 3.75.$$

$$z_1 = \frac{15 + 2(1.75) - 3.75}{5} = 2.95.$$

Método iterativo Gauss-Seidel

- Y se necesitaron 10 iteraciones!

Tabla 3.4 Convergencia del método iterativo de Gauss-Seidel para el sistema (1).

k	x_k	y_k	z_k
0	1.0	2.0	2.0
1	1.75	3.75	2.95
2	1.95	3.96875	2.98625
3	1.995625	3.99609375	2.99903125
\vdots	\vdots	\vdots	\vdots
8	1.99999983	3.99999988	2.99999996
9	1.99999998	3.99999999	3.00000000
10	2.00000000	4.00000000	3.00000000

Cómo saber cuando convergen?

- Definición: Se dice que una matriz A de $N \times N$ es de **diagonal estrictamente dominante** cuando:

$$|a_{kk}| > \sum_{j=1, j \neq k}^N |a_{kj}| \quad \text{para } k=1,2,\dots,N$$

- Teorema (Método iterativo de Jacobi). Supongamos que A es una matriz diagonal estrictamente dominante. Entonces el sistema de ecuaciones lineales $AX = B$ tiene solución única $X=P$. Además, el proceso iterativo de Jacobi produce una sucesión de vectores que converge a P cualquiera sea el vector de partida.
- Dem: Puede encontrarse en textos análisis numérico avanzado.
- Nota: Esta es una condición suficiente pero no necesaria...

Método sobrerelajación sucesiva

- **Cómo se deduce?** Recordemos la ecuación de diferencias de Laplace

$$u_{i+1j} + u_{i-1j} + u_{ij+1} + u_{ij-1} - 4u_{ij} = 0$$

- y supongamos que conocemos los valores de $u(x,y)$ en el contorno

$$u(x_1, y_j) = u_{1,j} \quad \text{para } 2 \leq j \leq m-1 \quad (\text{a la izquierda})$$

$$u(x_i, y_1) = u_{i,1} \quad \text{para } 2 \leq i \leq n-1 \quad (\text{abajo})$$

$$u(x_n, y_j) = u_{n,j} \quad \text{para } 2 \leq j \leq m-1 \quad (\text{a la derecha})$$

$$u(x_i, y_m) = u_{i,m} \quad \text{para } 2 \leq i \leq n-1 \quad (\text{arriba})$$

Métodos sobrerelajación sucesiva

- La ecuación escrita de manera adecuada para iterar es:

$$u_{ij} = u_{ij} + r_{ij}$$

- siendo

$$r_{ij} = \frac{u_{i+1j} + u_{i-1j} + u_{ij+1} + u_{ij-1} - 4u_{ij}}{4}$$

- con $2 \leq i \leq n-1$ y $2 \leq j \leq m-1$
-
- Qué información adicional necesitamos?

Métodos sobrerelajación sucesiva

- Disponer de valores iniciales en los puntos interiores de la malla (grilla)
 - Usar por ejemplo la constante K definida como la media de los $2n + 2m - 4$ valores de contorno
- En que consiste cada iteración?
 - Hacer un barrido de todos los puntos interiores de la malla con la fórmula recursiva hasta que r_{ij} se “reduzca” a cero ($|r_{ij}| < \epsilon$, siendo ϵ una tolerancia prefijada)
- Se puede aumentar la velocidad de la convergencia a cero?
 - Si: usando método de sobrerelajación sucesiva

Métodos sobrerrelajación sucesiva

- Método de relajación sucesiva:

$$(22) \quad \begin{aligned} u_{i,j} &= u_{i,j} + \omega \left(\frac{u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1} - 4u_{i,j}}{4} \right) \\ &= u_{i,j} + \omega r_{i,j}, \end{aligned}$$

en la que el parámetro ω verifica $1 \leq \omega < 2$. En el método de sobrerrelajación sucesiva, cada paso de la iteración consiste en hacer un barrido de la malla con la fórmula recursiva (22) hasta que se tenga $|r_{i,j}| < \varepsilon$. Para elegir el valor óptimo del parámetro ω hay que estudiar los autovalores de la matriz que caracteriza el método iterativo que estamos usando para resolver un sistema lineal; en nuestro caso, dicho valor óptimo viene dado por la fórmula

$$(23) \quad \omega = \frac{4}{2 + \sqrt{4 - \left(\cos\left(\frac{\pi}{n-1}\right) + \cos\left(\frac{\pi}{m-1}\right) \right)^2}}$$

Ejemplo: Laplace- cond. Dirichlet

- Problema: determinar la solución aproximada de la ecuación de Laplace en el rectángulo donde $u(x,y)$ denota la temperatura en un punto (x,y) , los valores de frontera son:
 - $u(x,0) = 20 \quad 0 < x < 4$
 - $u(x,4) = 180 \quad 0 < x < 4$
 - $u(0,y) = 80 \quad 0 < y < 4$
 - $u(4,y) = 0 \quad 0 < y < 4$

Solución: dividimos el cuadrado en 64 cuadrados, usamos $h = 0.5$ y valor inicial puntos interiores = 70, para $i=2,\dots,8$ y $j = 2,\dots,8$. $\omega = 1.44646$, $n=9$, $m=9$

Resultados

Tabla 10.6 Solución aproximada de la ecuación de Laplace con condiciones de Dirichlet.

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9
y_9	130.000	180.000	180.000	180.000	180.000	180.000	180.000	180.000	90.0000
y_8	80.000	124.821	141.172	145.414	144.005	137.478	122.642	88.6070	0.0000
y_7	80.000	102.112	113.453	116.479	113.126	103.266	84.4844	51.7856	0.0000
y_6	80.000	89.1736	94.0499	93.9210	88.7553	77.9737	60.2439	34.0510	0.0000
y_5	80.000	80.5319	79.6515	76.3999	70.0003	59.6301	44.4667	24.1744	0.0000
y_4	80.000	73.3023	67.6241	62.0267	55.2159	46.0796	33.8184	18.1798	0.0000
y_3	80.000	65.0528	55.5159	48.8671	42.7568	35.6543	26.5473	14.7266	0.0000
y_2	80.000	51.3931	40.5195	35.1691	31.2899	27.2335	21.9900	14.1791	0.0000
y_1	50.000	20.0000	20.0000	20.0000	20.0000	20.0000	20.0000	20.0000	10.0000

- Obtenido despues de 19 iteraciones
- $|r_{ij}| \leq 0.000606 < 0.001$
- Uso de promedio en esquinas

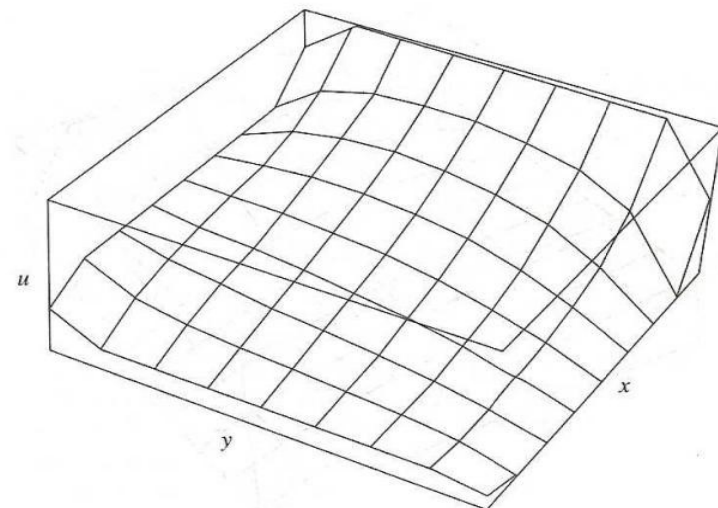


Figura 10.20 Solución $u = u(x, y)$ de un problema de Dirichlet.

Métodos iterativos ...

- Método de relajación sucesiva: Condiciones de Neumann

$$(24) \quad u_{i,1} = u_{i,1} + \omega \left(\frac{2u_{i,2} + u_{i-1,1} + u_{i+1,1} - 4u_{i,1}}{4} \right) \quad (\text{lado inferior}),$$

$$(25) \quad u_{i,m} = u_{i,m} + \omega \left(\frac{2u_{i,m-1} + u_{i-1,m} + u_{i+1,m} - 4u_{i,m}}{4} \right) \quad (\text{lado superior}),$$

$$(26) \quad u_{i,j} = u_{i,j} + \omega \left(\frac{2u_{2,j} + u_{1,j-1} + u_{1,j+1} - 4u_{1,j}}{4} \right) \quad (\text{lado izquierdo}),$$

$$(27) \quad u_{n,j} = u_{n,j} + \omega \left(\frac{2u_{n-1,j} + u_{n,j-1} + u_{n,j+1} - 4u_{n,j}}{4} \right) \quad (\text{lado derecho}).$$

Ejemplo: Laplace- Cond. Mixtas

- Problema: determinar la solución aproximada de la ecuación de Laplace en el rectángulo donde $u(x,y)$ denota la temperatura en un punto (x,y) , los valores de frontera son:
 - $u(x,4) = 180 \quad 0 < x < 4$ (Dirichlet)
 - $u_y(x,0) = 0 \quad 0 < x < 4$ (Neumann)
 - $u(0,y) = 80 \quad 0 < y < 4$ (Dirichlet)
 - $u(4,y) = 0 \quad 0 < y < 4$ (Dirichlet)

Solución: dividimos el cuadrado en 64 cuadrados, usamos $h = 0.5$, valor inicial puntos interiores 70, para $i=2,\dots,8$ y $j = 2,\dots,8$. $\omega = 1.44646$, $n=9$, $m=9$, valor inicial para $y = y_1 = 0$, interpolación lineal,

Resultados

Tabla 10.7 Solución aproximada de la ecuación de Laplace con condiciones de contorno mixtas.

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9
y_9	180.000	180.000	180.000	180.000	180.000	180.000	180.000	180.000	90.0000
y_8	80.000	126.457	142.311	146.837	145.468	138.762	123.583	89.1008	0.0000
y_7	80.000	103.518	115.951	119.568	116.270	105.999	86.4683	52.8201	0.0000
y_6	80.000	91.6621	98.4053	99.2137	94.0461	82.4936	63.4715	35.7113	0.0000
y_5	80.000	84.7247	86.7936	84.8347	78.2063	66.4578	49.2124	26.5538	0.0000
y_4	80.000	80.4424	79.2089	75.1245	67.4860	55.9185	40.3665	21.2915	0.0000
y_3	80.000	77.8354	74.4742	68.9677	60.6944	49.3635	35.0435	18.2459	0.0000
y_2	80.000	76.4244	71.8842	65.5772	56.9600	45.7972	32.1981	16.6485	0.0000
y_1	80.000	75.9774	71.0605	64.4964	55.7707	44.6670	31.3032	16.1500	0.0000

- Obtenido despues de 29 iteraciones
- $|r_{ij}| \leq 0.000998 < 0.001$
- Uso de promedio en esquinas

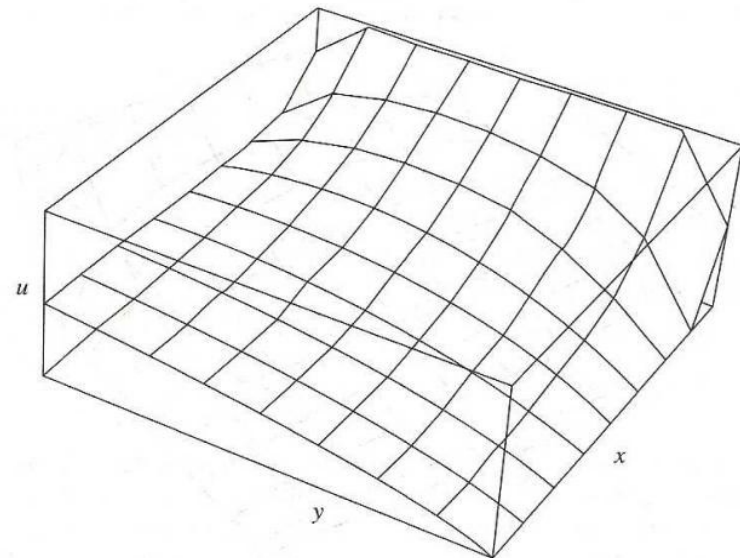


Figura 10.21 Solución $u = u(x, y)$ de un problema mixto.

EDPs Elípticas más generales

- Método iterativo aplicado a ecuación de Poisson

Las ecuaciones de Poisson y Helmholtz

Consideremos la ecuación de Poisson

$$(28) \quad \nabla^2 u = g(x, y).$$

Usando la notación $g_{i,j} = g(x_i, y_j)$, la extensión de la fórmula (20) para resolver la ecuación (28) sobre una malla rectangular es

$$(29) \quad u_{i,j} = u_{i,j} + \frac{u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1} - 4u_{i,j} - h^2 g_{i,j}}{4}.$$

EDPs Elípticas más generales

- Método iterativo aplicado a ecuación de Helmholtz

Consideremos la ecuación de Helmholtz

$$(30) \quad \nabla^2 u + f(x, y)u = g(x, y).$$

Usando la notación $f_{i,j} = f(x_i, y_j)$, la extensión de la fórmula (20) para resolver la ecuación (30) sobre una malla rectangular es

$$(31) \quad u_{i,j} = u_{i,j} + \frac{u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1} - (4 - h^2 f_{i,j})u_{i,j} - h^2 g_{i,j}}{4 - h^2 f_{i,j}}.$$

Estas fórmulas se analizarán con más detalle en los ejercicios.

Resumen

- Métodos para resolver sistemas lineales
 - Método de Gauss (exacto)
 - Método iterativo de Jacobi
 - Método iterativo de Gauss-Seidel
 - Método de sobrerelajación sucesiva (aplicado a la ecuación de Laplace)