PERBANDINGAN METODE GAUSS-JACOBI DAN METODE GAUSS-SEIDEL DALAM PENYELESAIAN PERSAMAAN LINEAR

Haeni Budiati

Ilmu Komputer, FMIPA, UKRIM

Abstrak

Metode-metode tersebut adalah metode Gauss-Jacobi dan metode Gauss-Seidel dengan cara aproksimasi Keuntungan penting dari metode iterasi adalah kesederhanaan dan keseragaman dari operasi yang dilakukan.

Kata Kunci : perbandingan metode Gauss-Jacobi, metode Gauss-Seidel.

1. PENDAHULUAN

Banyak metode yang dapat digunakan untuk mencari akar-akar suatu persamaan, yaitu menentukan nilai-nilai x yang memenuhi $f(x) = 0$. sering dijumpai pula masalah yang mirip dengan masalah mencari akar tersebut, tetapi tidak hanya berkaitan dengan sebuah fungsi dan sebuah variabel tetapi beberapa fungsi yang terdiri dari beberapa variabel membentuk sebuah sistem persamaan.

Pada tugas akhir ini akan dibandingkan 2 metode untuk menyelesaikan persamaan linear. Metode-metode tersebut adalah metode Gauss-Jacobi dan metode Gauss-Seidel dengan cara aproksimasi.

2. RUMUSAN MASALAH

Metode iteratif atau metode tidak langsung dimulai dengan aproksimasi awal untuk sebuah pemecahan dan kemudian membangun sebuah aproksimasi terbaik terhadap pemecahan eksak. Metode ini diterapkan dalam penyelesaian sistem linear dari n persamaan n bilangan tidak diketahui. Disamping mengatasi dilema pembulatan metode iteratif mempunyai beberapa keuntungan lain yang secara khas membuatnya menarik dalam konteks masalah-masalah rekayasa tertentu. Dari beberapa metode iteratif masing-masing memiliki ciri khas yang berbeda untuk itulah diperlukan suatu perbandingan, untuk mengetahui ciri khas masing-masing.

3. TUJUAN

Tujuan penulisan ini adalah untuk membandingkan dua metode yang mempunyai tujuan sama yaitu menyelesaikan persamaan suatu fungsi linear dengan bantuan suatu program pembanding. Hal-hal yang akan dibandingkan adalah: Hasil bilangan riil yang dihasilkan, Jumlah iterasi pengerjaan suatu fungsi, Bilangan Relaksasi, Waktu pengerjaan dalam program komputer.

4. LANDASAN TEORI

4.1 Sistem Persamaan Linear

Masalah persamaan linear aljabar pada dasarnya serupa dengan akar-akar persamaan dalam arti bahwa disini lebih menelaah nilai-nilai yang memenuhi persamaan. Tetapi berlawanan dengan pemenuhan sebuah persamaan tunggal, disini yang dicari adalah suatu himpunan nilai yang secara simultan memenuhi suatu himpunan persamaan aljabar. Terdapat 2 metode untuk menyelesaikan persamaan linear simultan. Cara pertama disebut metode iterasi atau metode tak langsung atau aproksimasi, cara kedua melalui proses eliminasi dan cara determinan yang disebut metode langsung atau analitik. Pada bagian ini hanya dibahas metode iteratif yaitu : metode Gauss-Jacobi dan Gauss-Seidel .

4.2 Metode Iterasi

Metode iterasi dimulai dengan pendekatan awal dan dengan penggunaan algoritma yang sesuai ke pendekatan-pendekatan yang berturut-turut lebih baik. Bahkan jika proses tersebut konvergen, dapat mencapai suatu penyelesaian yang mendekati saja. Keuntungan penting dari metode iterasi adalah kesederhanaan dan keseragaman dari operasi yang dilakukan

4.2.1 Metode Iterasi Gauss-Jacobi

Metode Gauss-Jacobi memerlukan rekaan awal dan fungsi iterasi untuk mendekati penyelesaiannya. Persamaan (2.1) diselesaikan ke elemen diagonal menghasilkan bentuk (2.2). Iterasi Gauss-Jacobi menggunakan rumusan rekursif untuk menghitung nilai pendekatan solusi persamaan. Proses iterasi dilakukan sampai suatu nilai konvergen dengan toleransi yang diberikan.

$$\left.\begin{aligned} x_1 &= \frac{y_1 - (a_{12}x_2 + a_{13}x_3 + \ldots + a_{1n}x_n)}{a_{11}} \\ x_2 &= \frac{y_2 - (a_{21}x_1 + a_{23}x_3 + \ldots + a_{2n}x_n)}{a_{22}} \\ &\ldots \\ x_n &= \frac{y_n - (a_{n1}x_1 + a_{n2}x_2 + \ldots + a_{nn-1}x_{n-1})}{a_{nn}} \end{aligned}\right\} \qquad (2.2)$$

Persamaan (2.2) tersebut dapat digunakan sebagai fungsi iterasi yang menghubungkan terkaan awal dengan terkaan selanjutnya. Untuk terkaan awal biasanya diambil $x_i^{(0)} = 0$, untuk I = 1, 2, … , n kemudian dihitung untuk k = 1, 2, … sampai cukup:

$$x_i^{(k)} = \frac{y_i - \sum_{(j=1, j\neq i)}^{n} a_{ij} x_j^{(k-1)}}{a_{ii}} \qquad , untuk \quad i = 1, 2, \ldots, n \qquad (2.3)$$

Metode iterasi ini memperoleh deret:

$$x_i^{(0)}, x_i^{(1)}, x_i^{(2)}, x_i^{(3)}, \ldots \quad (i = 1,2,\ldots,n)$$

yang bila berkonvergen menghasilkan nilai-nilai yang dicari. Konvergensi hasil diperoleh jika hasil pendekatan memberikan nilai konstan atau simpangan antara dua hasil lebih kecil dari toleransi kesalahan yang ditetapkan.

4.2.2 Algoritma Gauss-Jacobi

a) Masukan nilai koefisien persamaan sebagai nilai matrik [A] dan hasilnya sebagai nilai matrik {y}yang membentuk sistem persamaan linear simultan.

b) Set waktu perhitungan = 0, Perhit = 0, dan $\lambda = 1$.

c) λ ditentukan komputer atau ditentukan user. Jika komputer $\lambda = 1 - \left(\frac{Perhit}{10}\right)$.

d) Inisialisasi nilai $x^{(k)}$. Cek apakah $\lambda = 0$, jika ya: Hentikan perhitungan dan tampilkan pesan. Jika tidak: ke langkah selanjutnya.

e) Cek iter = 35.000, jika ya: kembali ke langkah c. jika tidak: kelangkah selanjutnya.

f) Hitung harga $x^{(k+1)}$ dengan rumusan iterasi jacobi dan perbaiki dengan rumus relaksasi.

g) Periksa hasil perhitungan; jika telah memenuhi galat yang diberikan cetak nilai $x^{(k+1)}$sebagai hasil akhir perhitungan dan hentikan perhitungan. Jika tidak, lanjutkan kelangkah berikutnya.

h) Gantikan nilai $x^{(k)}$ dengan $x^{(k+1)}$, dan kembali ke langkah (d).

4.2.3. Metode Iterasi Gauss-Seidel

Berbeda dengan metode Jacobi, metode ini tidak mengiterasi x_i untuk i = 1, 2, … , n secara serempak, tetapi setiap mendapatkan nilai baru tersebut digunakan untuk menghitung nilai berikutnya. Dengan demikian iterasi Gauss-Seidel sebagai cara penyelesaian persamaan linear simultan tidak jauh beda dengan iterasi Gauss-Jacobi.

Untuk terkaan awal biasanya diambil $x_i^{(0)} = 0$, untuk I = 1, 2, … , n.

kemudian dihitung untuk k = 1, 2, … sampai cukup:

$$x_1^{(k)} = \frac{y_1 - \sum_{(j=2)}^{n} a_{1j} x_j^{(k-1)}}{a_{11}} \quad , untuk \quad i = 1, 2, \ldots, n \qquad (2.4a)$$

$$x_i^{(k)} = \frac{y_i - \sum_{(j=1)}^{i-1} a_{ij} x_i^{(k)} - \sum_{(j=i+1)}^{n} a_{ij} x_i^{(k-1)}}{a_{ii}} \quad , untuk \quad i = 1, 2, \ldots, n \qquad (2.4b)$$

bentuk iteratif persamaan iterasi adalah sebagai berikut:

$$\left.\begin{aligned} x_1 &= \frac{y_1 - (a_{12}x_2 + a_{13}x_3 + \ldots + a_{1n}x_n)}{a_{11}} \\ x_2 &= \frac{y_2 - (a_{21}x_1 + a_{23}x_3 + \ldots + a_{2n}x_n)}{a_{22}} \\ &\ldots \\ x_n &= \frac{y_n - (a_{n1}x_1 + a_{n2}x_2 + \ldots + a_{nn-1}x_{n-1})}{a_{nn}} \end{aligned}\right\} \qquad (2.5)$$

4.2.4 Algoritma Gauss-Seidel:

a. Masukan nilai koefisien persamaan sebagai nilai matrik [A] dan hasilnya sebagai nilai matrik {y}yang membentuk sistem persamaan linear simultan.
b. Set waktu perhitungan = 0.
c. λ ditentukan komputer atau ditentukan user. Jika komputer $\lambda = 1 - \left(\frac{Perhit}{10}\right)$.

d. Inisialisasi nilai $x^{(k)}$ dan True = 1. Cek apakah $\lambda = 0$, jika ya: Hentikan perhitungan dan tampilkan pesan. Jika tidak: ke langkah selanjutnya.
e. Cek iter = 35.000, jika ya: kembali ke langkah c. jika tidak: kelangkah selanjutnya.
f. Untuk setiap baris persamaan (i = 1,2,…,n):
   - Hitung harga x_i^{k+1} dengan rumusan iterasi Gauss-Seidel dan perbaiki dengan relaksasi..
   - Periksa apakah hasil yang didapat sudah memenuhi galat relatif yangditentukan. Jika belum inisialisasi True = 0 sebagai kode bahwa Galat relatif belum terpenuhi.
   - Gantikan x_i^k dengan x_i^{k+1}.
g. Jika True = 0, Ulangi langkah (d).
h. Jika True ≠ 0, tulis nilai $x^{(k)}$ sebagai hasil perhitungan akhir dan hentikan perhitungan.

4.3 Macam-macam Matriks Awal

1) Matriks Simetri Bilangan Real (*Real Symetric Matrices*)

Matriks ini dinyatakan dalam bentuk : $A_{ji} = A_{ij}^{+}$, dimana semua A_{ij} adalah real.

2) Matriks Diagonal (Diagonal Matrices)

Semua elemen dalam matriks ini adalah 0 (nol) kecuali elemen diagonal. Matriks dari nilai eigen memiliki bentuk seperti ini, yang dinyatakan: $A_{ij} = 0$, *hanya jika* $i \neq j$

3) Matriks Tridiagonal (Tridiagonal Matrices)

Semua elemen matriks ini adalah 0 (nol) kecuali diagonal utama dan satu baris elemen diagonal diatas dan dibawah diagonal utama, yang dinyatakan dengan: $A_{i,i} \neq 0$, $A_{i,i\pm1} \neq 0$

4) Matriks Segitiga Atas dan Bawah (Upper and Lower Triangular Matrices)

Dalam matriks segitiga atas semua elemen dibawah diagonal adalah 0 (nol). $A_{ij} = 0$, *Untuk* $i > j$

5) **Matriks Jarang (Sparse Matrices)**

Matriks yang banyak memiliki elemen 0 (nol) yang membentuk suatu pola tertentu.

4.4 Perbaikan Kekonvergenan Dengan Memakai Relaksasi

Ada beberapa golongan dari sistem persamaan linear simultan yang matriks koefisien [A] pada [A][X] = [y] yang dominan diagonal cenderung konvergen tetapi kemungkinan juga tidak konvergen, yaitu:

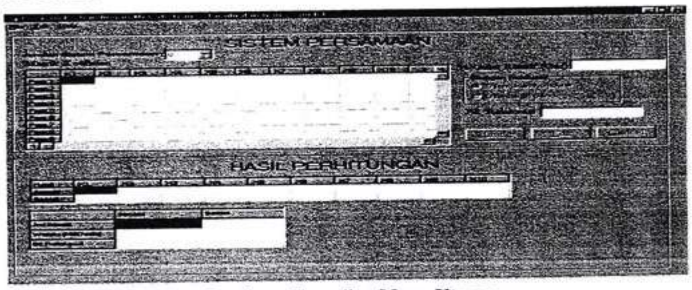
$$|a_{ii}| \geq \sum_{\substack{j=1 \\ j \neq i}}^{n} |a_{ij}| \qquad , \textit{untuk semua 'i'} \qquad (2.7)$$

$$|a_{ii}| > \sum_{\substack{j=1 \\ j \neq i}}^{n} |a_{ij}| \qquad , \textit{untuk} \text{ min } \textit{satu 'i'} \qquad (2.8)$$

5. PENGUJIAN / HASIL ANALISA

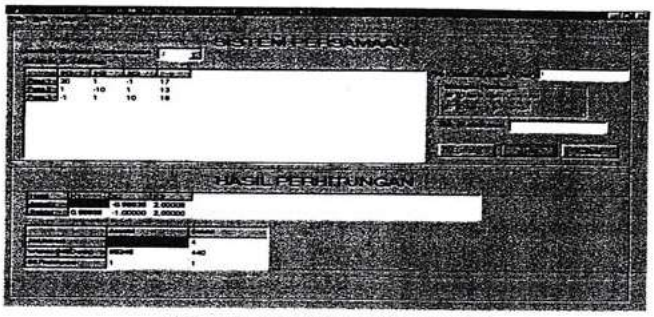
A. Interface program

Menu Utama



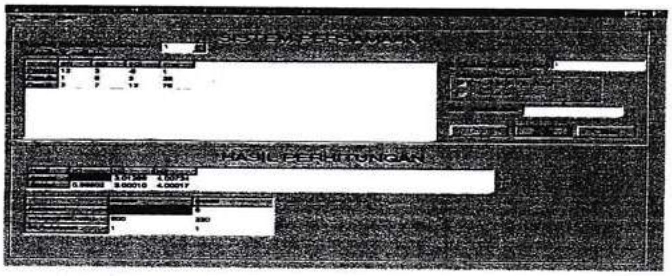
Gambar : Tampilan Menu Utama

Hasil Perhitungan Sistem Persamaan Menggunakan Contoh 2.7



Gambar : Tampilan Hasil Contoh 2.7

Hasil Perhitungan Sistem Persamaan Menggunakan Contoh 2.8



Gambar : Tampilan Hasil Contoh 2.8

B. Hasil Perbandingan

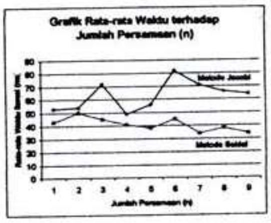
a. Matriks Simetri Bilangan Real

Dari 10 kali hasil percobaan dengan matriks simetri bilangan real , dan rlat toleransi yang sama untuk masing-masing n didapatkan data hasil perhitungan seperti pada table :

Tabel : Hasil Perbandingan Rata-Rata Waktu Matriks Simetri Bilangan Real

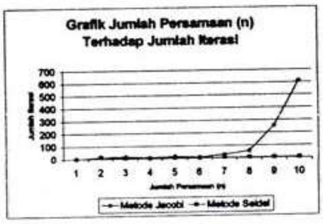
n	Jacobi		Seidel	
	Rata-rata Waktu (ms)	Iterasi	Rata-rata Waktu (ms)	Iterasi
1	0	0	0	0
2	22	14	23	15
3	50	18	28	5
4	49	8	30	4

5	56	17	55	5
6	51	10	32	4
7	49	29	35	5
8	39	57	37	5
9	50	255	38	5
10	50	611	27	5



Gambar : Grafik Rata-rata Waktu Iterasi terhadap Jumlah Persamaan (n) pada Matriks diagonal

Gambar : Grafik Rata-rata Waktu Iterasi terhadap Jumlah Persamaan (n) pada matriks Simetri Bilangan Real



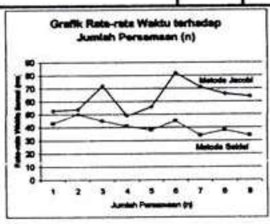
Gambar : Grafik Jumlah Iterasi terhadap Jumlah Persamaan (n) pada Matriks Simetri Bilangan Real.

b. Matriks Diagonal

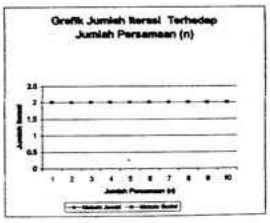
Dari 10 kali hasil percobaan dengan matriks diagonal dan ralat toleransi yang sama untuk masing-masing n didapatkan data hasil perhitungan seperti pada tabel dibawah ini :

Tabel : Hasil Perbandingan Rata-rata Waktu Matriks Diagonal

N	Jacobi		Seidel	
	Iterasi	Rata-rata Waktu (ms)	Iterasi	Rata-rata Waktu (ms)
	2	0	2	0
2	2	53	2	43
3	2	54	2	50
4	2	72	2	45
5	2	49	2	41
6	2	56	2	38
7	2	82	2	45
8	2	71	2	34
9	2	66	2	38
10	2	64	2	34



Gambar : Grafik Rata-rata Waktu Iterasi terhadap Jumlah Persamaan (n) pada Matriks Diagonal



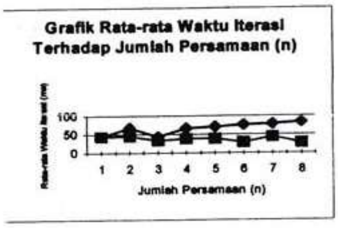
Gambar : Grafik Jumlah Iterasi terhadap Jumlah Persamaan (n) pada Matriks Diagonal

c. Matriks TriDiagonal

Dari 10 kali hasil percobaan dengan matriks tridiagonal dan galat toleransi yang sama untuk masing-masing n didapatkan data hasil perhitungan seperti pada tabel :

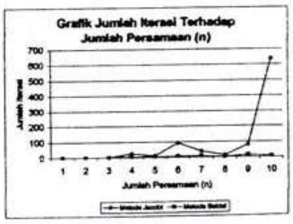
Tabel : Hasil Perbandingan Rata-rata Waktu Matriks Tridiagonal

N	Jacobi		Seidel	
	Iterasi	Rata-rata Waktu (ms)	Iterasi	Rata-rata Waktu (ms)
	0	0	0	0
2	0	0	0	0
3	2	44	2	43
4	26	68	6	45
5	11	43	6	33
6	93	67	9	39
7	39	71	9	38
8	13	76	3	28
9	80	78	14	43
10	637	83	7	27



3 4 5 6 7 8 9 1

Gambar : Grafik Rata-rata Waktu Iterasi Terhadap Jumlah Persamaan (n) pada Matriks Tridiagonal



Gambar: Grafik Jumlah Iterasi terhadap Jumlah Persamaan (n) pada Matriks Tridiagonal

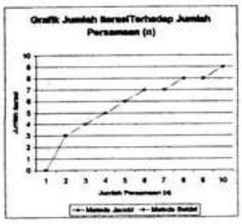
d. Matriks Diagonal Atas

Dari 10 kali hasil percobaan dengan matriks diagonal atas dan galat toleransi yang sama untuk masing-masing n didapatkan data hasil perhitungan seperti pada table :

Tabel 4.4 : Hasil Perbandingan Rata-rata Waktu Matriks Diagonal Atas

N	Jacobi		Seidel	
	Iterasi	Rata-rata Waktu (ms)	Iterasi	Rata-rata Waktu (ms)
	0	0	0	0
2	3	44	3	44
3	4	37	4	43
4	5	41	5	32
5	6	49	6	38
6	7	70	7	17
7	7	78	7	46
8	8	60	8	32
9	8	53	8	44
10	9	38	9	45

Gambar: Grafik Rata-rata Waktu Iterasi terhadap Jumlah Persamaan (n) pada Matriks Diagonal Atas



Gambar: Grafik Jumlah Iterasi terhadap Jumlah Persamaan (n) pada Matriks Diagonal Atas

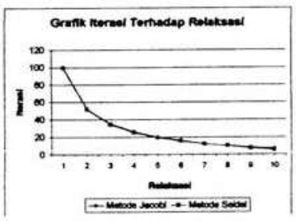
C. Hasil Analisa

Iterasi Terhadap Relaksasi

Dengan menggunakan persamaan pada Contoh 2.7 untuk menguji pengaruh Iterasi terhadap Relaksasi, didapatkan data seperti pada Tabel dibawah ini :

Tabel Data Perhitungan Iterasi dari Bilangan relaksasi Berbeda dengan menggunakan contoh 2.7

Relaksasi	Metode Jacobi Iterasi	Metode Seidel Iterasi
1	7	5
0.9	8	7
0.8	10	10
0.7	12	12
0.6	16	15
0.5	20	19
0.4	26	25
0.3	35	34
0.2	52	51
0.1	100	99



0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0

Gambar 4.23 : Grafik Iterasi terhadap Bilangan Relaksasi Menggunakan Contoh 2.7.

Kesimpulan

Berdasarkan beberapa hasil percobaan yang telah dilakukan dapat diambil kesimpulkan bahwa :

1. Hasil bilangan riil yang di hasilkan oleh kedua metode dapat dicapai yang terbaik jika galat masing-masing variable lebih kecil dari toleransi galat aproksimasi yang ditentukan. Semakin kecil galat yang diberikan maka akan diperoleh hasil yang semakin baik.
2. Dari kedua metode yang digunakan untuk penyelesaian sistem persamaan linear diperoleh bahwa iterasi dengan menggunakan metode Gauss-Seidel lebih pendek daripada metode Gauss-Jacobi.
3. Perubahan bilangan relaksasi tidak mempengaruhi hasil tetapi mempengaruhi jumlah iterasi perhitungan metode Gauss-Jacobi dan metode Gauss-Seidel. Memberikan bilangan relaksasi yang kecil pada persamaan yang sudah konvergen akan memperbesar % kesalahan hasil.
4. Waktu iterasi pada metode Gauss-Seidel lebih pendek dari pada metode Gauss-Jacobi. Tetapi tidak selalu demikian karena pada kasus tertentu Gauss-Jacobi akan lebih cepat.
5. Dari hasil kesimpulan diatas dapat dibuktikan bahwa pada kasus sistem persamaan linear biasa metode Gauss-Seidel lebih baik, dan lebih efisien daripada metose Gauss-Jacobi.

Daftar Pustaka

Anton, Howard,1992, *Aljabar Linear Elementer*, Penerbit Erlangga, Jakarta.

Bambang Triadmojo,1992, *Metode Numerik*, Beta Offset, Yogyakarta.

Chapra,S.C and Canale,Raymond P,1985,*Numerical Methods For Enginers*, Singapore: Macgrow-Hill.

Conte. S. D.Carl de Boor, 1986, *Elementary Numerical Analysis*, Singapore: MacGrow-Hill, 3rd edition.

Djoko Susilo, 2002, *Komponen Visual: Perancangan dan Implemensasinya pada Delphi 6*, J&J Learning, Yogyakarta.

Kaw,Autar. K,2002, *Introduction to Matrix Algebra*, University of South Florida.

Nasution, Amrinsyah dan Hasballah Zakaria,2001, *Metode Numerik dalam Ilmu Rekayasa Sipil*, Penerbit ITB, Bandung.