

PERBANDINGAN TIGA METODE UMUM UNTUK MEMECAHKAN PERSAMAAN LINEAR JARANG (*SPARSE*)

Riyad Mubarak Abdullah, Kais Ismail, dan Totok Suprawoto

STMIK AKAKOM

Jl. Janti, Ringroad Timur, Karangjambe, Yogyakarta

E-mail: riyad_m@yahoo.com

E-mail: kaisismail@yahoo.com

E-mail: totok@akakom.ac.id

INTISARI

Penelitian ini menyajikan sebuah perbandingan menarik di antara tiga metode untuk memecahkan sistem linear $Ax = b$. Metode tersebut adalah metode *LU*, *conjugate gradient* dan *wavelet*. Kami menyimpulkan bahwa metode *wavelet* lebih efisien daripada metode lain, karena metode ini memerlukan waktu yang lebih singkat untuk menyelesaikan semua jenis matriks, padat, tridiagonal dan jarang. Dari penelitian juga ditemukan bahwa metode *wavelet* bisa digunakan untuk menyelesaikan sistem persamaan linear jarang yang berukuran besar dengan waktu yang lebih singkat dibandingkan dengan metode yang lain.

ABSTRACT

This paper presents an interesting comparison among three methods for solving linear system $Ax = b$. These methods are the *LU* method, conjugate gradient method, and the wavelet method. We concluded that the wavelet method is more efficient than the others, because its need short time in computation for all kind of matrices, dense, tridiagonal and sparse. This comparison is explained by many numerical examples. In this paper we conclude that wavelet method can be used to solve a large linear sparse system equation in a shorter time than other methods.

Latar Belakang

Persamaan linear jarang $Ax = b$, A adalah matriks persegi empat dimensi n , adalah model umum dari banyak sistem rekayasa kontemporer, dan usaha telah banyak dilakukan untuk mendesain skema efisien dalam penyelesaian persamaan tersebut.

Untuk sistem linear besar yang mengandung ribuan persamaan, metode iterasi sering memiliki kelebihan yang menentukan jika dibandingkan dengan metode lain dalam hal kecepatan dan kapasitas memori komputer yang dibutuhkan. Jika suatu solusi tidak membutuhkan ketelitian tinggi, maka penggunaan cacah iterasi yang sering muncul akan memberikan hasil yang baik. Untuk sistem jarang (dimana perbandingan besar unsur dalam matriks $A = 0$, metode iterasi sangatlah cocok dalam masalah jaranganya unsur yang tidak bernilai nol yang terkadang disimpan dalam suatu format penyimpanan jarang, pastilah tidak efisien untuk menyimpan seluruh unsur matriks A dalam memori komputer. Keadaan ini biasanya terjadi pada solusi numeris dalam persamaan diferensial parsial. Dimana tiap baris dalam matriks A bisa dihasilkan sesuai dengan keinginan kita (Kincaid dan Chency, 1996).

Gelombang-singkat (*Wavelet*) dikembangkan secara luas di bidang matematika, fisika kuantum, teknik elektro, dan geologi seismik. Bidang ilmu lain yang menggunakan gelombang-singkat diantaranya astronomi, akustika, teknik nuklir, pengkodean subbidang, pengolahan sinyal dan citra, neurofisiologi, musik, citra resonans magnetik, diskriminasi pembicaraan, optika fraktal/pemecahan, turbulens, peramalan gempa bumi, radar, penglihatan manusia, dan penerapan matematika murni seperti pemecahan persamaan diferensial parsial (Graps A., 1995).

Tujuan Penelitian

Ada tiga tujuan dari penelitian ini yaitu:

- Di dalam penelitian ini lebih difokuskan pada tiga metode LU , metode *conjugate gradient* dan metode gelombang-singkat (*wavelet*), dengan suatu perbandingan menarik di antara tiga metode untuk memecahkan suatu sistem linear jarang guna mengetahui metode yang terbaik.
- Dalam penelitian ini akan diperkenalkan suatu metode baru untuk menyelesaikan persamaan linear jarang yaitu metode gelombang-singkat.
- Metode gelombang-singkat ini juga kami gunakan untuk memecahkan sistem persamaan linier jarang yang berukuran besar.

Manfaat Penelitian

Penelitian ini bisa mempersingkat waktu penyelesaian suatu sistem persamaan linear jarang yang mempunyai ukuran cukup besar dengan memanfaatkan metode gelombang-singkat.

Landasan Teori

Dalam bagian ini akan dijelaskan metode-metode yang akan digunakan dalam penelitian ini untuk memecahkan sistem persamaan liner jarang sebagai berikut:

3.1 Metode LU

Eliminasi Gaussian adalah bagian yang paling dikenal dari metode dekomposisi LU langsung. Metode langsung yang lain juga digunakan secara luas. Akan dibahas metode ini dan kemudian melihat pada cara bagaimana metode LU Gaussian bisa digunakan secara efisien untuk memecahkan soal-soal yang melibatkan sisi kanan ganda.

Reduksi Crout mentransformasikan koefisien matriks A , menjadi hasil dari dua matriks, L dan U , di mana U memiliki satu pada diagonal utamanya. Teknik ini berbeda dari metode bagian sebelumnya di mana L memiliki satu pada diagonalnya. Sebelumnya kami telah melihat bahwa sebuah matriks yang telah mengalami triangularisasi dan dikombinasikan dengan matriks segitiga bawah yang terbentuk dari rasio tersebut dan digunakan di dalam reduksi membentuk sebuah pasangan LU . Tetapi pasangan LU mengambil banyak bentuk lain. Pada kenyataannya, matriks tertentu yang memiliki semua elemen diagonal *nonzero* bisa ditulis sebagai sebuah hasil dari matriks segitiga bawah dan matriks segitiga atas dengan cara tak terhingga.

Dari keseluruhan susunan LU yang hasilnya sama dengan matriks A , pada metode Crout kami memilih pasangan di mana U hanya memiliki satu pada diagonalnya, seperti pada pasangan pertama di atas. Kami mendapatkan aturan untuk dekomposisi LU semacam itu dari hubungan tertentu sehingga $LU = A$. Pada kasus matriks 4×4 :

$$\begin{bmatrix} l_{11} & 0 & 0 & 0 \\ l_{21} & l_{22} & 0 & 0 \\ l_{31} & l_{32} & l_{33} & 0 \\ l_{41} & l_{42} & l_{43} & l_{44} \end{bmatrix} \begin{bmatrix} 1 & u_{12} & u_{13} & u_{14} \\ 0 & 1 & u_{23} & u_{24} \\ 0 & 0 & 1 & u_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix}$$

Dengan mengalikan baris L pada kolom pertama U , kita dapatkan

$$l_{11} = a_{11},$$

$$l_{21} = a_{21},$$

$$l_{31} = a_{31},$$

$$l_{41} = a_{41},$$

kolom pertama L adalah sama seperti kolom pertama matriks A . Sekarang kita mengalikan baris pertama L pada kolom U :

$$l_{11}u_{12} = a_{12}, \quad l_{11}u_{13} = a_{13}, \quad l_{11}u_{14} = a_{14} \quad (1.1)$$

dari mana

$$u_{12} = \frac{a_{12}}{l_{11}}, \quad u_{13} = \frac{a_{13}}{l_{11}}, \quad u_{14} = \frac{a_{14}}{l_{11}} \quad (1.2)$$

Oleh karena itu baris pertama U ditentukan. Di dalam metode ini kita bergantian di antara mendapatkan satu kolom L dan sebuah baris U , sehingga selanjutnya kita mendapatkan persamaan untuk kolom kedua L dengan mengalikan baris-baris L pada kolom kedua U :

$$l_{21}u_{12} + l_{22} = a_{22}$$

$$l_{31}u_{12} + l_{32} = a_{32} \quad (1.3)$$

$$l_{41}u_{12} + l_{42} = a_{42}$$

yang menghasilkan

$$l_{22} = a_{22} - l_{21}u_{12}$$

$$l_{32} = a_{32} - l_{31}u_{12} \quad (1.4)$$

$$l_{42} = a_{42} - l_{41}u_{12}$$

Memulai dengan cara yang sama, persamaan yang kita butuhkan adalah:

$$u_{23} = \frac{a_{23} - l_{21}u_{13}}{l_{22}}, u_{24} = \frac{a_{24} - l_{21}u_{14}}{l_{22}},$$

$$l_{23} = a_{33} - l_{31}u_{13} - l_{32}u_{23}, l_{43} = a_{43} - l_{41}u_{13} - l_{42}u_{23},$$

$$u_{34} = \frac{a_{34} - l_{31}u_{14} - l_{32}u_{24}}{l_{33}},$$

$$l_{44} = a_{44} - l_{41}u_{14} - l_{42}u_{24} - l_{43}u_{34}.$$

Rumus umum untuk mendapatkan elemen-elemen L dan U yang berhubungan dengan matriks koefisien untuk persamaan simultan n bisa ditulis.

$$l_{ij} = a_{ij} - \sum_{k=1}^{j-1} l_{ik}u_{kj}, \quad j \leq i, \quad i = 1, 2, \dots, n,$$

$$u_{ij} = \frac{a_{ij} - \sum_{k=1}^{i-1} l_{ik}u_{kj}}{l_{ii}}, \quad i \leq j, \quad j = 2, 3, \dots, n.$$

untuk $j=1$, aturan untuk l dikurangi menjadi

$$l_{i1} = a_{i1}.$$

untuk $i=1$, aturan untuk u dikurangi menjadi

$$u_{1j} = \frac{a_{1j}}{l_{11}} = \frac{a_{1j}}{a_{11}}$$

Alasan mengapa metode ini populer di dalam program tertentu adalah bahwa ruang penyimpanan bisa menjadi lebih ekonomis. Tidak butuh menyimpan nol baik pada L maupun U , Dengan kata lain, barisan A bisa ditransformasikan oleh persamaan-persamaan di atas dan menjadi:

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} \rightarrow \begin{bmatrix} l_{11} & u_{12} & u_{13} & u_{14} \\ l_{21} & l_{22} & u_{23} & u_{24} \\ l_{31} & l_{32} & l_{33} & u_{34} \\ l_{41} & l_{42} & l_{43} & l_{44} \end{bmatrix}$$

Kemudian algoritma yang digunakan untuk dekomposisi LU adalah sebagai berikut: Untuk mentransformasikan sebuah matriks A berukuran $n \times n$ menjadi $L*U$:

for $i = 1$ to n

$$L[i,1]=a[i,1]$$

```

end
for j = 1 to n
    U[1,j] = a[1,j] / L[1,1]
end
for j = 2 to n
    for i = j to n
        sum = 0.0
        for k = 1 to j-1
            sum = sum + L[i,k]*U[k,j]
        end
        U[i,j] = a[i,j]-sum
    end
    U[j,j]=1
    for i = j+1 to n
        sum = 0.0
        for k = 1 to j-1
            sum = sum + L[j,k]*U[k,i]
        end
        U[j,i] = (A[j, i]-sum)/L[j,j]
    end
end
end

```

Pemecahan susunan persamaan $A\underline{x} = \underline{b}$ bisa diperoleh dengan matriks L dan U . Matriks L sungguh merupakan catatan dari operasi yang diperlukan untuk membuat matriks A masuk ke dalam matriks segitiga atas U . Kami menggunakan transformasi yang sama untuk vektor sisi kanan b , dengan merubahnya menjadi vektor baru b' . Jika kita menambahkan b' pada U dan kembali mensubstitusikannya, kita dapatkan pemecahan tersebut. Persamaan umum untuk reduksi b :

$$b'_1 = \frac{b_1}{l_{11}}$$

$$b'_i = \frac{b_i - \sum_{k=1}^{i-1} l_{ik} b'_k}{l_{ii}}, \quad i = 2, 3, \dots, n$$

Persamaan untuk substitusi kembali adalah:

$$x_n = b'_n,$$

$$x_j = b'_j - \sum_{k=j+1}^n u_{jk} x_k, \quad j = n-1, n-2, \dots, 1$$

(Gerald dan Wheatley, 1994)

Metode Conjugate Gradient

Metode gradian konjugasi (*conjugate gradient*) berusaha mendapatkan solusi untuk persamaan matriks melalui proses iteratif. Pemecahan ini bisa

diturunkan dengan memperhitungkan sebuah fungsi yang nilai minimumnya berhubungan dengan pemecahan persamaan matriks tersebut.

Metode gradien konjugasi mencari solusi persamaan matriks melalui proses iterasi, yang bisa diturunkan dari sebuah fungsi yang memiliki tingkat kecocokan minimum terhadap solusi persamaan matriks. Misalnya dua vektor

$$f = [f_1, f_2, f_3, \dots, f_n]$$

dan $g = [g_1, g_2, g_3, \dots, g_n]$, ingat bahwa produk dalam f dan g didefinisikan sebagai:

$$\langle f, g \rangle = \sum_{i=1}^n f_i g_i^* \tag{1.5}$$

dan adjoint A^a dari sebarang matriks A didefinisikan oleh:

$$\langle Af, g \rangle = \langle f, A^a g \rangle \tag{1.6}$$

dari hal ini bisa diamati bahwa:

$$A^a = (A^*)^T$$

dimana asterisk menandakan konjugasi kompleks dan T menandakan transposisi. Jelaslah, jika $B = A^a A$, kemudian B merupakan adjoint sendiri atau Hermitian (saat $B^a = B$), maka akan bernilai definit positif.

Untuk menyelesaikan $Ax = b$, akan dikalikan dengan A^a , menjadi $A^a Ax = A^a b$. Jika $B = A^a A$ dan $h = A^a b$, diperoleh:

$$Bx = b. \tag{1.7}$$

Kemudian, bentuk suatu fungsi $F(x)$ yang memiliki kecocokan minimum terhadap solusi pers.(1.7):

$$F(x) = \frac{1}{2} \langle Bx, x \rangle - \frac{1}{2} \langle h, x \rangle - \frac{1}{2} \langle h, x \rangle \tag{1.8}$$

Persamaan ini bisa diperiksa diman minimalisasi fungsional dengan minimalisasi $\langle r, r \rangle$, dimana $r = b - Ax$.

Solusi x ditulis sebagai:

$$\begin{aligned} x &= x_1 + \alpha_1 p_1 + \alpha_2 p_2 + \alpha_3 p_3 + \dots + \alpha_n p_n \\ &= x_1 + \sum_{k=1}^n \alpha_k p_k \end{aligned} \tag{1.9}$$

dimana x_1 merupakan initial guess, p_k adalah vektor yang belum diketahui arahnya pada tahap k , dan α_k merupakan koefisien yang belum diketahui. Dengan mensubstitusi, kita mendapatkan fungsi F sebagai suatu *quadratic* α_k , dan untuk meminimalkannya, turunan parsialnya diatur menurut bagian imajiner dan real α_s menjadi nol, dihasilkan:

$$\sum_{k=1}^n \langle Bp_k, p_s \rangle \alpha_k = \langle h, p_s \rangle - \langle Bx_1, p_s \rangle, \quad s = 1, 2, 3, \dots, n \quad (1.10)$$

kemudian, jika p_k dimasukkan ke dalam kondisi:

$$\langle Bp_k, p_s \rangle = 0, \quad \text{untuk } s \neq k \quad (1.11)$$

dan α_k bisa diperoleh dari pers. (1.11) sebagai:

$$\alpha_k = \frac{\langle h - Bx_1, p_k \rangle}{\langle Bp_k, p_k \rangle} \quad (1.12)$$

vektor p_k memenuhi pers. (1.11) yang disebut vektor *B-orthogonal* atau *B-conjugate*.

Sekarang, tinggal menentukan bagaimana deret vektor p_k dihasilkan. Ingat bahwa $x_{k+1} = x_k + \alpha_k p_k$ dan vektor *residual* $r_{k+1} = h - Bx_{k+1}$ bisa ditulis sebagai:

$$r_{k+1} = r_k - \alpha_k Bp_k. \quad (1.13)$$

jika

$$\langle h, p_k \rangle = \langle r_k + Bx_1, p_k \rangle = \langle r_k, p_k \rangle + \langle Bx_1, p_k \rangle,$$

akan diperoleh

$$\alpha_k = \frac{\langle r_k, p_k \rangle}{\langle Bp_k, p_k \rangle} \quad (1.14)$$

dan menghasilkan suatu yang sangat penting:

$$\begin{aligned} F(x_{k+1}) &= F(x_k) + \frac{1}{2} \alpha_k \alpha_k^* \langle Bp_k, p_k \rangle - \frac{1}{2} \alpha_k^* \langle r_k, p_k \rangle - \frac{1}{2} \alpha_k \langle p_k, r_k \rangle \\ &= F(x_k) - \frac{1}{2} \frac{|\langle r_k, p_k \rangle|^2}{\langle Bp_k, p_k \rangle} \end{aligned} \quad (1.15)$$

ini menandakan bahwa fungsional menurun pada tiap iterasi dan akhirnya mencapai nilai minimum, solusi terhitung $A\underline{x} = \underline{b}$, setelah n langkah.

Jika vektor B -orthogonal p_k bisa dihasilkan melalui proses ortogonalisasi Gram-Schmidt, akan diperoleh:

$$p_{k+1} = r_{k+1} + \frac{\langle r_{k+1}, r_{k+1} \rangle}{\langle r_k, r_k \rangle} p_k \quad (1.16)$$

Sebuah algoritma untuk metode gradian konjugasi untuk memecahkan $A\underline{x} = \underline{b}$ diberikan di bawah ini (Due to Jin 1993). Matriks tentu positif B tidak perlu diperhitungkan:

Ambil x_1 sebagai perkiraan awal dan hitunglah

$$r_1 = b - Ax_1$$

$$p_1 = \frac{A^a r_1}{\langle A^a r_1, A^a r_1 \rangle}$$

sekarang, iterasikan untuk $k = 1, 2, 3, \dots, n$

$$\alpha_k = \frac{1}{\langle Ap_k, Ap_k \rangle}$$

$$x_{k+1} = x_k + \alpha_k p_k$$

$$r_{k+1} = r_k - \alpha_k Ap_k$$

$$\beta_k = \frac{1}{\langle A^a r_{k+1}, A^a r_{k+1} \rangle}$$

$$p_{k+1} = p_k + \beta A^a r_{k+1}$$

dan hentikan ketika

$$\frac{\|r_{k+1}\|}{\|b\|} < \varepsilon.$$

Metode Alihragam Gelombang-singkat

Gagasan mendasar di belakang gelombang-singkat adalah untuk melakukan analisa sesuai dengan skala. Dengan menggunakan gelombang-singkat, seseorang menggunakan keseluruhan pemikiran baru atau pespektif pemrosesan data baru.

Gelombang-singkat meruapkan fungsi yang memenuhi persyaratan matematis tertentu dan digunakan di dalam merepresentasikan data atau fungsi

lain. Gagasan ini adalah sesuatu yang baru. Perkiraan menggunakan superposisi fungsi telah ada sejak awal tahun 1800-an ketika Joseph Fourier menemukan bahwa ia bisa melakukan superposisi terhadap sinus dan cosinus untuk merepresentasikan fungsi-fungsi lain (Graps, 1995).

Pada bagian ini kami menggunakan gelombang-singkat Haar, basis Haar merupakan basis gelombang-singkat paling sederhana (Stollinz, dkk., 1996). Kami akan mulai dengan membahas bagaimana sebuah fungsi satu dimensi bisa didekomposisikan menggunakan gelombang-singkat Haar.

Sebuah matriks dengan mudah bisa dipahami sebagai penataan baris per baris atau kolom per kolom dari tanda-tanda tertentu, sedemikian rupa sehingga matriks bisa diterima untuk analisis transformasi. Jika operasi semacam itu dilakukan pada sebuah persamaan matriks $Ax = b$, persamaan $Wax = Wb$ yang telah dialihragamkan diperoleh. Dari sini, kita bisa tulis $(WAW^{-1})(Wx) = Wb$. Dengan memilih, sebagai contoh, transformasi ortogonal W , hubungan $(WAW^T)Wx = Wb$. Sifat umum yang menarik dari metode ini adalah bahwa sebuah alihragam gelombang-singkat dari matriks padat menghasilkan matriks jarang (Bond dan Vavasis, 1994).

Terdapat dua cara umum di mana gelombang-singkat bisa digunakan untuk mengalihragamkan nilai di dalam sebuah matriks. Masing-masing dari alihragam ini merupakan genearlisasi dua dimensi dari alihragam gelombang-singkat satu dimensi yang telah dideskripsikan sebelumnya.

Alihragam pertama disebut dekomposisi standar. Untuk mendapatkan dekomposisi standar sebuah matriks, pertama-tama kita menggunakan alihragam gelombang-singkat satu dimensi untuk setiap baris nilai. Operasi ini memberi kita nilai rata-rata dengan koefisien detil untuk setiap baris. Selanjutnya kita memperlakukan baris-baris yang telah dialihragamkan ini seolah-olah merupakan sebuah matriks dan menggunakan alihragam satu dimensi pada setiap kolom. Nilai-nilai yang dihasilkan adalah koefisien detil kecuali untuk koefisien secara keseluruhan tunggal.

Tipe kedua dari alihragam gelombang-singkat dua dimensi disebut dekomposisi nontandar, berpindah-pindah di antara operasi pada baris dan kolom

pada saat yang sama. Pertama, kita melakukan satu tahap penataan berpasangan horisontal dan membedakan nilai pada setiap baris matriks tersebut. Selanjutnya, kita menggunakan pasangan vertikal dengan menentukan rerata dan melakukan perbedaan pada setiap kolom hasil. Untuk menyelesaikan transformasi ini, kita mengulang proses ini secara rekursif hanya pada kuadran yang berisi rata-rata pada kedua arah (Stollintz, dkk., 1996). Lihat juga Burrus (1998) untuk pengenalan mendasar.

Hasil Numeris

Selain matriks jarang kami juga menguji tiga metode tersebut dengan menggunakan jenis matriks yang lain seperti berikut:

Matriks Padat (*Dense Matrices*)

Kami menghasilkan matriks padat A_1, A_2, A_3, A_4, A_5 dan A_6 dengan ukuran 8, 16, 32, 64, 128 dan 256 secara berturut-turut sebagai berikut:

$$A_1 = \begin{bmatrix} 1 & 2 & -30 & 29 & 18 & 17 & 5 & 6 \\ 6 & 26 & 36 & 1 & 2 & 5 & 68 & 10 \\ 13 & 4 & 19 & 18 & 17 & 120 & 13 & 5 \\ 18 & -21 & 60 & 9 & 7 & 11 & 19 & 80 \\ 70 & -6 & 18 & 28 & 2 & 1 & 3 & -2 \\ 4 & 3 & 1 & 1 & 2 & 1 & 1 & -9 \\ 4 & 1 & 18 & 39 & 7 & 1 & 2 & -12 \\ 1 & -34 & -21 & 10 & 16 & 70 & 3 & 6 \end{bmatrix} \text{ matriks berukuran } 8 \times 8$$

$$A_2 = \begin{bmatrix} A_1 & A_1+2 \\ A_1 & 5 * A_1 \end{bmatrix} \text{ matriks berukuran } 16 \times 16$$

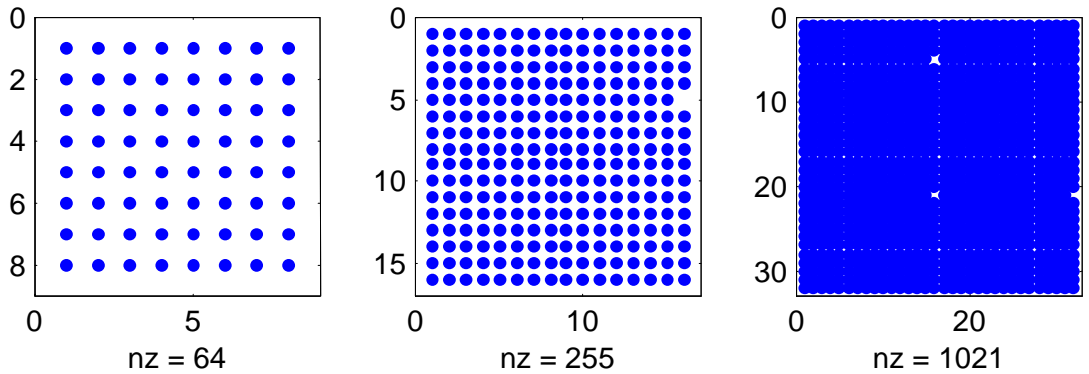
$$A_3 = \begin{bmatrix} A_2 & A_2+3 \\ A_2 & 3 * A_2 \end{bmatrix} \text{ matriks berukuran } 32 \times 32$$

$$A_4 = \begin{bmatrix} A_3 & 2 * A_3 \\ A_3 & A_3+3 \end{bmatrix} \text{ matriks berukuran } 64 \times 64$$

$$A_5 = \begin{bmatrix} A_4 & 3 * A_4 \\ A_4 & A_4+3 \end{bmatrix} \text{ matriks berukuran } 128 \times 128$$

$$A_6 = \begin{bmatrix} A_5 * 4 & A_5 \\ A_5 * 7 & A_5 + 7 \end{bmatrix} \text{ matriks berukuran } 256 \times 256$$

matriks-matriks A_1, A_2 dan A_3 bisa dilihat dengan cara lain seperti pada gambar di bawah ini:



dengan menerapkan metode LU, CG (*Conjugate Gradien*) dan WT (*Haar-based wavelet transform*) pada persamaan linear $Ax = b$ dengan A secara berturut-turut sama dengan A_1, A_2, A_3, A_4, A_5 dan A_6 , kita mencatat jumlah *flops* dan waktu yang diperlukan. Diperoleh hasil-hasil numerik berikut:

Ukuran matriks	LU		WT		CG	
	Flops	Waktu	Flops	Waktu	Flops	Waktu
8 x 8	581	0.000	5093	0.0000	3387	0.4400
16 x 16	3641	0.1100	32486	0.1100	31627	1.2000
32 x 32	25441	0.7200	228329	0.3900	402918	19.4900
64 x 64	189105	4.8300	1699266	1.7600	7147354	344.7200
128 x 128	1455441	50.8600	13087675	5.5500	67527579	2.6161e+003
256 x 256	11414161	279.350	102674720	24.8300	x	Very Long

Tabel berikut menunjukkan hasil dua metode yang telah dibahas sebelumnya untuk metode *wavelet*, sebagai berikut:

Ukuran matriks	WT (Dekomposisi Standar)		WT (Dekomposisi Tak Standar)	
	Flops	Waktu	Flops	Waktu
8 x 8	5760	0.0500	5093	0.0000
16 x 16	35212	0.1600	32486	0.1100
32 x 32	238940	0.7100	228329	0.3900
64 x 64	1742156	2.9200	1699266	1.7600
128 x 128	13257342	8.5100	13087675	5.5500
256 x 256	103358396	44.2200	102674720	24.8300

Matriks Tridiagonal

Matriks tridiagonal A

$$A = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 2 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 2 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 2 \end{bmatrix}$$

merupakan sebuah contoh dari matriks simetrik $A = (a_{ij})$ dengan ukuran n didefinisikan sebagai berikut:

$$a_{11} = 1, a_{ii} = 2 \text{ for } 1 < i \leq n$$

$$a_{ij} = -1 \text{ for } |i-j| = 1,$$

$$= 0 \text{ otherwise.}$$

Inversi $B = (b_{ij})$ dari matriks tersebut juga simetris dengan elemen-elemen triangular lebih rendah $b_{ij} = n-i+1$ untuk semua i ; oleh karena itu pemecahan yang telah diperhitungkan bisa dicek dengan mudah untuk kebenarannya. Dengan melakukan eksperimen sederhana pada nilai n tersebut, diperoleh hasil-hasil berikut:

Ukuran matriks	LU		WT		CG	
	Flops	Waktu	Flops	Waktu	Flops	Waktu
8 x 8	581	0.1600	5163	0.0500	3387	0.4300
16 x 16	3641	0.2200	32438	0.2200	31627	1.5900
32 x 32	25441	1.9300	223517	0.8800	283812	12.9100
64 x 64	189105	9.9400	1639954	3.6800	3191782	119.6300
128 x 128	1455441	73.7700	12455289	13.8400	40947945	1.6408e+003
256 x 256	11414161	371.6300	96673702	59.8200	x	Very Long
512 x 512	90395921	3.9623e+003	760125391	264.7500	x	Very Long
1024x1024	x	Very Long	6.0297e+009	1.4298e+003	x	Very Long

Matriks Jarang

Sebagai contoh pertama matriks jarang umum, kami memperhitungkan matriks-matriks $A_1, A_2, A_3, A_4, A_5, A_6, A_7$ dan A_8 dengan ukuran 8, 16, 32, 64, 128, 256, 512 dan 1024 secara berturut-turut sebagai berikut:

$$a1 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 6 & 0 \\ 10 & 20 & 3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 5 & 0 & 0 & 0 & 0 & 0 \\ 9 & 0 & 7 & 5 & 4 & 9 & 0 & 0 \\ 12 & 0 & 0 & 0 & 7 & 0 & 0 & 5 \\ 8 & 0 & 0 & 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 5 & 9 \\ 0 & 0 & 0 & 0 & 0 & 0 & 6 & 2 \end{bmatrix} \quad a2 = \begin{bmatrix} 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 9 & 0 & 0 & 0 & 6 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 8 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 7 & 0 & 0 & 0 & 0 & 5 \\ 0 & 0 & 9 & 5 & 0 & 0 & 0 & 3 \\ 0 & 7 & 0 & 0 & 3 & 0 & 0 & 0 \end{bmatrix}$$

$$a3 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 8 \\ 0 & 3 & 0 & 0 & 7 & 0 & 0 & 4 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 7 & 9 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 8 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad a4 = \begin{bmatrix} 7 & 0 & 5 & 6 & 0 & 0 & 0 & 8 \\ 0 & 7 & 0 & 0 & 7 & 0 & 0 & 0 \\ 9 & 3 & 6 & 0 & 0 & 0 & 0 & 0 \\ 5 & 0 & 3 & 8 & 0 & 0 & 0 & 0 \\ 0 & 8 & 0 & 0 & 4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 4 & 0 & 0 \\ 0 & 9 & 0 & 0 & 0 & 7 & 4 & 0 \\ 0 & 9 & 0 & 0 & 0 & 0 & 0 & 5 \end{bmatrix}$$

$A1 = a1$ matriks berukuran 8×8

$$A2 = \begin{bmatrix} A1 & a3 \\ a2 & a4 \end{bmatrix} \text{ matriks berukuran } 16 \times 16$$

$$A3 = \begin{bmatrix} 2 * A2 & A2 \\ A2 & A2 + I \end{bmatrix} \text{ matriks berukuran } 32 \times 32$$

$$A4 = \begin{bmatrix} A3 & 2 * A3 \\ A3 & A3 + I \end{bmatrix} \text{ matriks berukuran } 64 \times 64$$

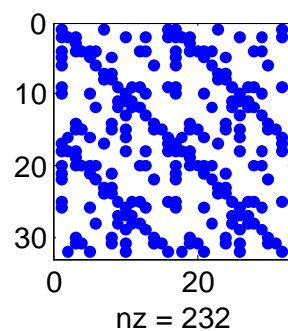
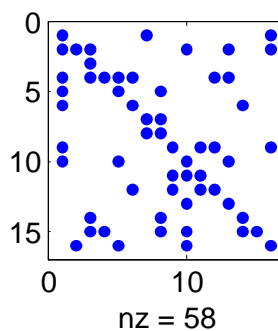
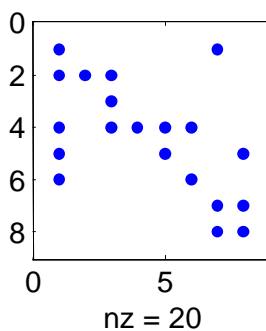
$$A5 = \begin{bmatrix} A4 & 2 * A4 \\ A4 & A4 \end{bmatrix} \text{ matriks berukuran } 128 \times 128$$

$$A6 = \begin{bmatrix} A5 & 2 * A5 \\ A5 + I & A5 + I \end{bmatrix} \text{ matriks berukuran } 256 \times 256$$

$$A7 = \begin{bmatrix} A6 & 2 * A6 \\ A6 + I & A6 + I \end{bmatrix} \text{ matriks berukuran } 512 \times 512$$

$$A8 = \begin{bmatrix} A7 & 2 * A7 \\ A7 + I & A7 + I \end{bmatrix} \text{ matriks berukuran } 1024 \times 1024$$

Jika kita ingin mengetahui jumlah unsur yang bukan nol dalam matriks-matriks di atas, sebagai contohnya kita ambil tiga matriks pertama yaitu 8×8 , 16×16 dan 32×32 sebagai berikut:



Dengan melakukan eksperimen sederhana pada nilai n tersebut, diperoleh hasil berikut:

Ukuran matriks	LU		WT		CG	
	Flops	Waktu	Flops	Waktu	Flops	Waktu
8 x 8	581	0.0500	5115	0.5500	3768	0.3300
16 x 16	3641	0.1100	32688	0.1100	29089	1.3700
32 x 32	25441	0.7100	228485	0.3900	357108	17.0800
64 x 64	189105	4.8900	1699674	1.7500	7251448	322.200
128 x 128	1455441	50.8100	13089937	6.9200	67527579	2.5132e+003
256 x 256	11414161	281.610	102681612	30.5400	x	Very Long
512 x 512	90395921	2.8171e+003	813365575	213.8200	x	Very Long
1024x1024	x	Very Long	6.4746e+009	663.8400	x	Very Long

Kesimpulan

1. Di dalam penelitian ini, kami membandingkan tiga metode, yakni metode *LU*, metode *Conjugate Gradient* dan metode *wavelet*.
2. Dari penelitian ini kami temukan bahwa metode *wavelet* memerlukan waktu lebih sedikit dibandingkan dua metode lainnya untuk semua jenis matriks. Untuk sistem yang besar, metode *LU* memerlukan waktu yang lebih panjang untuk melakukan perhitungan dan kadang-kadang gagal untuk matriks tridiagonal dan jarang.
3. Dari penelitian tersebut juga ditemukan bahwa metode *wavelet* bisa digunakan untuk menyelesaikan sistem persamaan linear yang berukuran besar dengan waktu yang lebih singkat dibandingkan dengan metode yang lain.

Saran

Kami menyarankan kepada para peneliti lain yang ingin menggunakan karya ini dalam penelitiannya, untuk menggunakan metode *LU* yang telah dimodifikasi dan *wavelet* lain. Sehingga hasilnya bisa dilakukan perbandingan untuk mempertinggi kualitas karya kami. Kami berharap di dalam kondisi semacam itu, hasil yang lebih baik akan diperoleh dan perkembangan akan terjadi.

Daftar Pustaka

- Antonini, M., Barlaud, M., Mathieu, P., dan Daubechies, I., 1992, *Image Coding Using Wavelet Transform*, in Proc. IEEE Trans. Image Processing, Vol. 1, April.
- Bond, D. M. dan S. A. Vavasis, "Fast Wavelet Transforms for Matrices Arising from Boundary Element Methods", Center for Applied Mathematics,

- Engineering and Theory Center, Cornell University, Ithaca, NY 14853, March 24, 1994.
- Burrus, C. S., R. A. Gopinath, dan H. Guo, "Introduction to Wavelet and Wavelet Transforms-A Primer", Prentice-Hall International, Inc., 1998.
- Da Silva, E.A.B. and Ghanbari, M., 1996, *On the Performance of Linear Phase Wavelet Transforms in Low Bit-Rate Image Coding*, IEEE Transactions on Image Processing, Vol. 5, No. 5, May.
- Daubenchies, I., 1988, *Orthonormal Bases of Compactly Supported Wavelets*, Commun. Pure Appl. Math., Vol. 41, November.
- Due to Jin, 1993, "The Finite Element Method in Electromagnetic", John Wley.
- Evangelista, G., 1993, *Pitch-Synchronous Wavelet Representations of Speech and Music Signals*, IEEE Trans. on Signal Processing, Vol. 41, No. 12, December.
- Gerald, C. F. dan Wheatley, P. O., 1994, "Applied Numerical Analysis", Fifth Edition, United States of America.
- Golub, G. E. dan Van Loan, C. F., 1993, "Matrix Computations", Second Edition, The Johns Hopkins University Press, Baltimore and London.
- Graps, A., 1995, "An Introduction to Wavelet", IEEE.
- Kincaid, D. dan Cheney, W., 1996, "Numerical Analysis Mathematics of Scientific Computing", Brooks/Cole Publishing Company, USA.
- Morton, P. dan Petersen A., 1997, "Image Compression Using the Haar Wavelet Transform", Math 45, College of the Redwoods, December 19.
- Rioul, O., dan Vetterli, M., 1991, *Wavelets and Signal Processing*, IEEE Signal Processing, October.
- Stollintz, E. J., T. D. Deroose, dan D. H. Salesin, "Wavelet for Computer Graphics Theory and application", Morgan Kaufmann Publishers, Inc., San Francisco, California, 1996.
- Westerink, P. H., 1987, Subband Coding of Images, Ph.D. dissertation, Delft University of Technology, The Netherlands.