

PENERAPAN OPTIMASI CHAOS DAN METODE BFGS (BROYDEN, BROYDEN, FLETCHER, GOLDFARB, AND SHANNO) PADA PENYELESAIAN PERMASALAHAN SISTEM PERSAMAAN NONLINIER

Rully Soelaiman, Nur Chasiani

Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Kampus Keputih, Sukolilo, Surabaya 60111, Indonesia
Email : rully@is.its.ac.id

Abstrak

Penyelesaian permasalahan sistem persamaan nonlinier merupakan salah satu hal yang paling sulit dalam kasus komputasi numerik dan metode Newton merupakan salah satu metode yang banyak digunakan untuk menyelesaikannya. Konvergensi dari metode Newton ini sangat sensitif terhadap perkiraan awal solusi. Tetapi sangatlah sulit memilih perkiraan awal solusi yang bagus untuk sebagian besar sistem persamaan nonlinier. Selain itu, penggunaan matriks Jacobian membuat biaya komputasi untuk metode Newton menjadi besar. Metode Quasi-Newton merupakan perbaikan dari metode Newton. Dengan memanfaatkan kemampuan pencarian global dari optimasi chaos dan nilai konvergensi yang tinggi dari metode Quasi-Newton, maka pendekatan gabungan dari kedua metode tersebut diusulkan. Pada paper ini digunakan algoritma optimasi chaos yang hasilnya akan dijadikan input sebagai perkiraan awal solusi pada algoritma Quasi-Newton. Dari hasil pengujian dan evaluasi, dengan pendekatan gabungan ini diperoleh tingkat konvergensi yang tinggi dan tidak diperlukan lagi perkiraan awal solusi untuk dapat menyelesaikan permasalahan sistem persamaan nonlinier.

Kata kunci : sistem persamaan nonlinier, optimasi chaos, metode Quasi-Newton.

1. Pendahuluan

Penyelesaian sistem persamaan non linier merupakan salah satu hal yang paling sulit dalam kasus komputasi numerik. Diperlukan usaha yang besar untuk bisa menyelesaikan sistem persamaan non linier ini dan sampai sekarang terdapat banyak macam teori dan algoritma yang digunakan untuk menyelesaikan kasus ini. Para ahli berlomba mencari metode yang paling benar, paling efektif dan paling efisien. Tetapi pada kenyataannya masih ditemui rintangan dalam penyelesaian sistem

persamaan non linier. Setiap metode dan algoritma yang ada mempunyai kelebihan dan kekurangan masing-masing.

Chaos, sifat kacau atau tidak beraturan yang sebenarnya bersifat deterministik atau bisa ditentukan, merupakan fenomena yang banyak terjadi di banyak sistem dalam bidang ilmu pengetahuan. Chaos Optimization Algorithm (COA) merupakan algoritma optimasi yang berdasarkan *ergodicity*, *stochastic properties*, dan *regularity* dari chaos itu sendiri.

Metode Newton merupakan metode yang paling banyak digunakan untuk menyelesaikan sistem persamaan non linier. Tetapi metode ini mempunyai kelemahan, antara lain tingginya waktu yang digunakan karena pada metode ini terdapat iterasi yang di dalamnya mengandung matriks Jacobian. Untuk mengatasi masalah tersebut, muncullah metode Quasi-Newton yang merupakan perbaikan dari metode Newton. Quasi-Newton mengganti komputasi yang bersifat derivatif dengan fungsi komputasi langsung. Quasi-Newton merupakan salah satu unconstrained Optimization untuk permasalahan non linier.

Di dalam paper ini akan dicari penyelesaian sistem persamaan non linier dengan menggunakan suatu pendekatan yang merupakan penggabungan antara dua metode di atas yaitu Chaos Optimization Algorithm (COA) dengan metode Quasi-Newton. Diharapkan kelebihan dari kedua metode tersebut akan menghasilkan penyelesaian yang efisien yaitu dengan mengambil kemampuan *global search* dari COA dan rata-rata konvergensi lokal yang tinggi yang dimiliki oleh metode Quasi-Newton.

2. Sistem Persamaan Nonlinier

Penyelesaian persamaan non linier adalah penentuan akar-akar persamaan nonlinier, dimana akar sebuah persamaan $f(x) = 0$ adalah nilai-nilai x yang menyebabkan nilai $f(x)$ sama dengan nol.

Sistem persamaan nonlinier adalah kumpulan dari beberapa persamaan nonlinier yang dicari

penyelesaiannya. Bentuk persamaannya adalah sebagai berikut

$$\begin{aligned} f_1(x_1, x_2, \dots, x_n) &= 0 \\ f_2(x_1, x_2, \dots, x_n) &= 0 \\ &\vdots \\ f_n(x_1, x_2, \dots, x_n) &= 0 \end{aligned} \quad (1)$$

Hasil penyelesaiannya adalah $x = (x_1, x_2, \dots, x_n)^T$.

Untuk menyelesaikan sistem persamaan non linier ekuivalen dengan meminimalkan fungsi utama yang di jabarkan sebagai berikut:

$$\begin{aligned} \text{Dicari} &: x = (x_1, x_2, \dots, x_n)^T, x \in \Phi \\ \text{Min} &: F(x) = \sum_{i=1}^n f_i^2(x) \end{aligned} \quad (2)$$

dimana Φ adalah ruang penyelesaian. Fungsi ini definit positif dan memiliki global minimum di setiap akar pemecahannya. Bila minimisasi dari $F(x)$ adalah 0, maka x adalah solusi yang tepat [1].

3. Metode Quasi-Newton dengan BFGS

Metode ini merupakan perbaikan dari metode Newton. Metode Newton bergerak berdasarkan informasi derivatif dan berasal dari analisis deret Taylor [1]. Format iteratif untuk metode Newton dapat ditulis sebagai berikut

$$x^{(k+1)} = x^{(k)} - J(x^{(k)})^{-1} f(x^{(k)}) \quad (3)$$

dimana $J(x)$ adalah matriks Jacobian

$$J(x) = \begin{bmatrix} \frac{\partial f_1(x)}{\partial x_1} & \frac{\partial f_1(x)}{\partial x_2} & \dots & \frac{\partial f_1(x)}{\partial x_n} \\ \frac{\partial f_2(x)}{\partial x_1} & \frac{\partial f_2(x)}{\partial x_2} & \dots & \frac{\partial f_2(x)}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_n(x)}{\partial x_1} & \frac{\partial f_n(x)}{\partial x_2} & \dots & \frac{\partial f_n(x)}{\partial x_n} \end{bmatrix} \quad (4)$$

Untuk kasus optimasi, metode Newton ini diterapkan pada kondisi $\nabla f(x) = 0$ [6]. Jacobian dari $\nabla f(x)$ adalah $\nabla^2 f(x)$ dan biasa disebut matriks Hessian, sehingga

$$x^{(k+1)} = x^{(k)} - H(x^{(k)})^{-1} J(x^{(k)}) \quad (5)$$

Metode Quasi-Newton mengganti komputasi yang bersifat derivatif dengan fungsi komputasi langsung. Matriks Hessian H diganti dengan aproksimasi atau perkiraan matriks Hessian A yang merupakan matriks definit positif yang mempunyai sifat seperti matriks Hessian [2]. Format iteratif dari metode ini adalah sebagai berikut:

$$x^{(k+1)} = x^{(k)} + \alpha^{(k)} S^{(k)} \quad (6)$$

dimana $\alpha^{(k)}$ adalah *step length* yang dapat meminimumkan fungsi $f(\alpha) = f(x^{(k)} + \alpha^{(k)} S^{(k)})$ dan

$S^{(k)}$ adalah *search direction*. *Search direction* ini didefinisikan sebagai berikut:

$$S^{(k)} = -(A^{(k)})^{-1} \nabla f(x^{(k)}) \quad (7)$$

Metode Quasi-Newton berbeda dalam bagaimana perkiraan matriks Hessian dibentuk dan diupdate. Metode Quasi-Newton yang paling sederhana menetapkan perkiraan matriks Hessian sebagai matriks identitas.

Cara update yang paling terkenal adalah dengan metode *Broyden, Fletcher, Goldfarb, and Shanno* (BFGS). Metode ini dikenal akan ketahanannya (*robustness*) dan dapat mencapai konvergensi superlinear dengan baik [2]. Rumus yang digunakan untuk update matriks Hessian adalah

$$A^{(k+1)} = A^{(k)} + \frac{q^{(k)}(q^{(k)})^T}{(q^{(k)})^T s^{(k)}} + \frac{g^{(k)}(g^{(k)})^T}{(\delta^{(k)})^T g^{(k)}} \quad (8)$$

dimana

$$q^{(k)} = \nabla f(x^{(k+1)}) - \nabla f(x^{(k)}), \quad g^{(k)} = \nabla f(x^{(k)}),$$

$$\delta^{(k)} = S^{(k)}, \quad s^{(k)} = \Delta x = x^{(k+1)} - x^{(k)}$$

4. Algoritma Optimasi Chaos (COA)

Optimasi chaos merupakan teknik pembangkitan bilangan random dengan menggunakan fungsi chaos, bisa berbentuk fungsi polinomial atau eksponensial seperti persamaan logistik di dalam ekologi yang digunakan untuk menghitung pertumbuhan populasi suatu spesies [10].

Salah satu fungsi chaos sederhana adalah persamaan logistik (*logistic map*). Persamaan logistik dinyatakan sebagai berikut [1]

$$t_{k+1} = \lambda t_k (1 - t_k) \quad (9)$$

dimana konstanta λ menyatakan laju pertumbuhan fungsi, yang dalam hal ini $0 \leq \lambda \leq 4$, $0 \leq t_0 \leq 1$, $k = 0, 1, 2, \dots$, Ketika $\lambda \in [3.56, 4]$ menyebabkan persamaan (9) menjadi chaotic.

Chaos Optimization Algorithm (COA) merupakan algoritma optimasi yang berdasarkan *ergodicity, stochastic properties, dan regularity* dari chaos itu sendiri.

Untuk masalah optimasi seperti dibawah ini

$$\min f(x_1, x_2, \dots, x_n), x_i \in [a_i, b_i], \quad i = 1, 2, \dots, n \quad (10)$$

diperkirakan

$$x^k = (x_1^k, x_2^k, \dots, x_n^k), \quad t^k = (t_1^k, t_2^k, \dots, t_n^k).$$

Secara umum garis besar algoritma optimasi chaos dapat diberikan sebagai berikut:

- Step 1. Inialisasi

$k = 0, r = 0, k$ adalah variabel tanda untuk setiap iterasi chaos, r adalah variabel tanda untuk

pencarian. Tetapkan jumlah iterasi maksimum untuk variabel chaos yaitu K_{\max} dan jumlah iterasi maksimum untuk pencarian (*fine search*) yaitu r_{\max} . Membuat nilai t^0 dengan random nilai antara $[0,1]$ dan kemudian memastikan bahwa $t_i^0 \neq (0, 0.25, 0.5, 0.75, 1.0)$. Tetapkan bahwa nilai $t^k = t^0, t^* = t^0, a_i^r = a_i, b^r = b_i$, dimana t^* adalah variabel chaos terbaik pada saat ini, a dan b adalah batas ruang pencarian (*search space*), kemudian tentukan nilai x^* dengan mengacak (*random*) dengan batas ruang pencarian $[a,b]$ dan hitung $f^* = f(x^*)$.

▪ Step 2. Mapping variabel yaitu

Memetakan t_i^k ke daerah optimasi, sehingga

$$x_i^k = a_i^r + t_i^k (b_i^r - a_i^r)$$

▪ Step 3

Hitung $f(x^k)$, bandingkan hasilnya dengan f^* .

Jika $f(x^k) < f^*$ maka $f^* = f(x^k)$, $x^* = x^k$, $t^* = t^k$

▪ Step 4. Iterasi variabel chaos

$$k = k + 1; t_i^k = 4t_i^k (1.0 - t_i^k)$$

▪ Step 5

Jika $k < K_{\max}$ maka ulangi step 2. Jika tidak, maka $r = r + 1$ dan lanjutkan ke step 6.

▪ Step 6. Ubah batas pencarian

$$a_i^{r+1} = x_i^* - \gamma(b_i^r - a_i^r); b_i^{r+1} = x_i^* + \gamma(b_i^r - a_i^r)$$

Jika $a_i^{r+1} < a_i^r$ maka $a_i^{r+1} = a_i^r$. Jika

$$b_i^{r+1} > b_i^r \text{ maka } b_i^{r+1} = b_i^r \text{ dimana } \gamma \in (0,0.5)$$

▪ Step 7

Jika $r < r_{\max}$, maka buat t^0 melalui random,

tentukan $k = 0$, dan $t^k = t^0$, kemudian kembali ke step 2. Jika tidak, maka COA diakhiri dan x^* adalah solusinya.

5. Pendekatan gabungan

Ide utama dalam paper ini adalah menggabungkan dua metode optimasi untuk menyelesaikan permasalahan sistem persamaan nonlinier. Optimasi chaos (COA) digunakan untuk menyelesaikan sistem persamaan nonlinier dengan mengoptimasi persamaan (2), sedangkan metode Quasi-Newton digunakan untuk menyelesaikan sistem persamaan nonlinier berdasarkan persamaan (1) dengan menggunakan solusi yang dihasilkan dari optimasi chaos (COA) sebagai perkiraan awal solusi. Proses ini akan diulang sampai didapatkan solusi yang akurat [1].

Gabungan antara 2 metode ini bisa dijabarkan seperti berikut :

• Step I.

Menentukan *solution error* ε_0 dan menjalankan step 1 dari metode COA.

• Step II.

Menjalankan step 2-6 dari metode COA dan fungsi objektifnya dideskripsikan sebagai persamaan (2)

• Step III.

Jadikan solusi x^* yang didapatkan dari COA sebagai inisialisasi awal pada metode quasi Newton, kemudian selesaikan sistem persamaan non linear yang dideskripsikan di persamaan (1) dan dapatkan solusi x^{**} ; jika $F(x^{**}) < F(x^*)$, buat $x^* = x^{**}$.

• Step IV

Jika $\varepsilon = \max(f_i(x^*))$ ($i = 1, 2, \dots, n$) $< \varepsilon_0$, algoritma hybrid dihentikan; jika tidak $r > r_{\max}$, hentikan algoritma hybrid, jika tidak buat x^* sebagai perkiraan awal dari pencarian chaos. Hasilkan t^0 dengan acak, $k = 0$, $t^k = t^0$, $r := r + 1$, kemudian kembali ke step 2.

6 Uji Coba dan Analisis

Uji coba dilakukan pada sebuah PC dengan prosesor Intel(R) Core(TM) 2 Duo CPU T5750 @2.00GHz (2 CPUs) dengan memori sebesar 1526 MB RAM. Sistem operasi yang digunakan adalah Windows Vista Home Basic dan bahasa komputasi yang digunakan untuk implementasi metode adalah Matlab Versi 7.0.

Untuk membuktikan kapabilitas metode yang ditawarkan, uji coba dilakukan terhadap 3 sistem persamaan nonlinier yang berbeda.

6.1 Sistem persamaan nonlinier 1

Bentuk dari sistem persamaan nonlinier 1 adalah

$$f_1(x) = x_1 + \frac{x_1^2 x_4 x_6}{4} + 0.75$$

$$f_2(x) = x_2 + 0.405 e^{1+x_1 x_2} - 1.405$$

$$f_3(x) = x_3 - \frac{x_4 x_6}{2} + 1.5$$

$$f_4(x) = x_4 - 0.605 e^{1-x_3^2} - 0.395$$

$$f_5(x) = x_5 - \frac{x_2 x_6}{2} + 1.5$$

$$f_6(x) = x_6 - x_1 x_5$$

Solusi untuk sistem persamaan nonlinier 1 ini [1] adalah $x_1^* = (-1, 1, -1, 1, -1, 1)^T$, $x_2^* = (-1.5544, 1.2470, -0.4625, 1.7289, -0.7638, 1.1917)^T$ dan $x_3^* = (-1.0432, -0.5509, 0.4319, 1.7597, -2.1049, 2.1958)^T$.

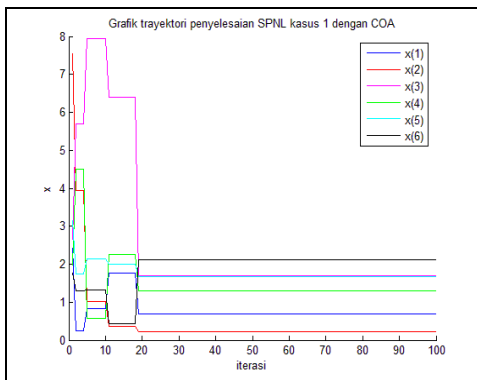
Ujicoba dilakukan dengan menggunakan k_{\max} 100, r_{\max} 100 dan batas ruang pencarian a dan b antara $[0, 10]$.

Hasil perhitungan dari percobaan ini adalah sebagai berikut :

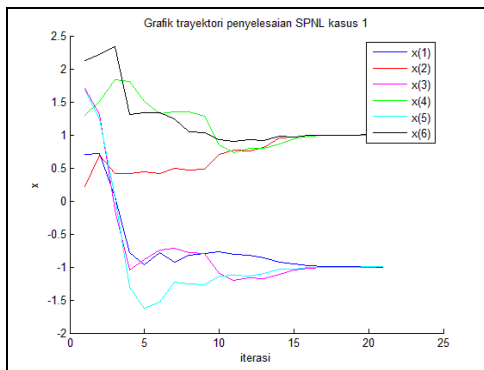
- Solusi COA : (0.6991, 0.22209, 1.6985, 0.2968, 1.6762, 2.1192)
- Solusi Akhir : (-1, 1, -1, 1, -1, 1)
- Jumlah iterasi : 20

Pada tahap uji coba, setiap sistem persamaan nonlinier memiliki grafik trayektori yang merepresentasikan pergerakan nilai aproksimasi solusi yang dihasilkan oleh metode penyelesaian pada tiap iterasi.

Gambar 1 merupakan grafik trayektori penyelesaian sistem persamaan nonlinier 1 dengan menggunakan COA, sedangkan gambar 2 merupakan grafik trayektori penyelesaian sistem persamaan nonlinier 1 dengan menggunakan metode gabungan. Dari Gambar 2 dapat dilihat bahwa metode gabungan (COA dan quasi-Newton) dapat menunjukkan performa kestabilan dan konvergensi pada satu titik kesetimbangan yang menjadi solusi optimal dari sistem persamaan nonlinier 1 ini.



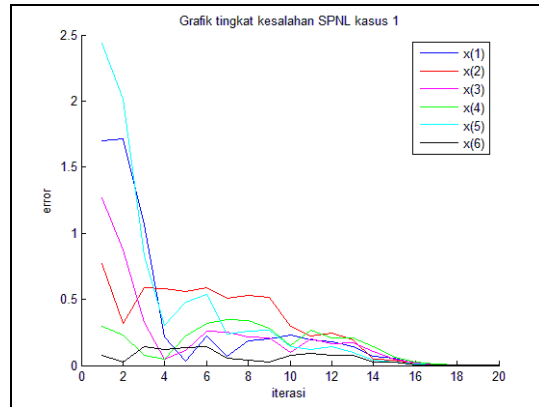
Gambar 1 Grafik trayektori penyelesaian SPNL 1 dengan COA



Gambar 2 Grafik trayektori penyelesaian SPNL 1 dengan metode gabungan

Untuk mengetahui kebenaran dari solusi yang dihasilkan dari metode gabungan ini, maka solusi yang sudah didapat dibandingkan dengan solusi sebenarnya yang sudah tersedia pada [1] dan dari Gambar 3 dapat terlihat bahwa tingkat kesalahan

dari solusi yang didapat untuk sistem persamaan nonlinier 1 ini mencapai angka 0. Selisih terakhir antara solusi yang dihasilkan dari metode gabungan dengan solusi sebenarnya adalah (0, 0, 0, 0, 0, 0) dan solusi ini sama dengan x_1^* .



Gambar 3 Grafik tingkat kesalahan SPNL 1

6.2 Sistem persamaan nonlinier 2

Bentuk dari sistem persamaan nonlinier 2 adalah sebagai berikut

$$A = bh - (b - 2t)(h - 2t),$$

$$I_y = \frac{bh^3}{12} - \frac{(b - 2t)(h - 2t)^3}{12},$$

$$I_n = \frac{2(h - t)^2(b - t)^2 t}{h + b - 2t}$$

Sistem persamaan nonlinier 2 ini merupakan pencarian solusi untuk ukuran geometri dari bagian balok penyangga (*grider section*) rectangular pada suatu bangunan [1]. b adalah lebar dari bagian *grider*, h adalah tinggi dari bagian *grider* dan t adalah ketebalan dari bagian *grider*.

Diketahui bahwa $A=165$, $I_y = 9369$, $I_n = 6835$.

Untuk kasus ini, terdapat kondisi bahwa $h > b > t > 0$. Kondisi ini menjadikan sistem persamaan nonlinier 2 mempunyai 3 batasan (*constraint*) yang berupa pertidaksamaan, batasan tersebut adalah

$$g_1(x) = b - h < 0,$$

$$g_2(x) = t - b < 0,$$

$$g_3(x) = -t < 0$$

Untuk mengatasi permasalahan karena adanya batasan, maka digunakan fungsi *penalty*, sehingga fungsi objektif dirumuskan sebagai berikut:

$$F(x) = \sum_{i=1}^n f_i^2(x) + M \sum_{i=1}^3 \max(0, g_i(x)) \quad (11)$$

Dimana M adalah koefisien *penalty*.

Solusi sebenarnya berdasarkan [1] adalah $x_1^* = (22.8949, 12.5655, 2.7898)^T$, $x_2^* = -x_1^*$, $x_3^* = (23.271482, 8.943089, 12.912774)^T$, $x_4^* = -x_3^*$,

$$x_5^* = (35.756376, -2.363740, 3.015078)^T, \quad x_6^* = -x_5^*.$$

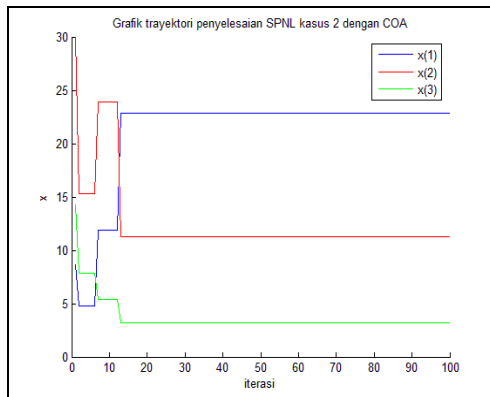
Ujicoba dilakukan dengan menggunakan k_{\max} 100, r_{\max} 100 dan batas ruang pencarian a dan b antara $[0, 30]$.

Dari Gambar 5 dapat dilihat bahwa metode gabungan (COA dan quasi-Newton) dapat menunjukkan performa kestabilan dan konvergensi pada satu titik kesetimbangan yang menjadi solusi optimal dari sistem persamaan nonlinier 2 ini.

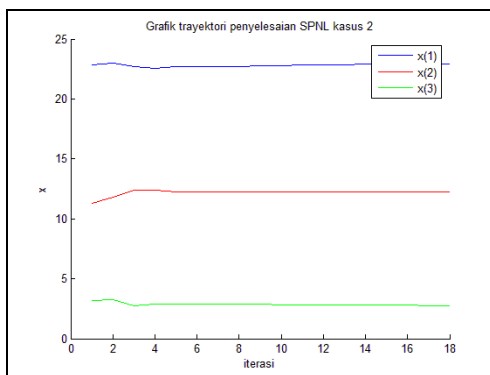
Hasil perhitungan dari uji coba ini adalah sebagai berikut :

- Solusi COA : (22.4878, 11.8816, 2.56314)
- Solusi Akhir : (22.9096, 12.2568, 2.78078)
- Jumlah iterasi : 26

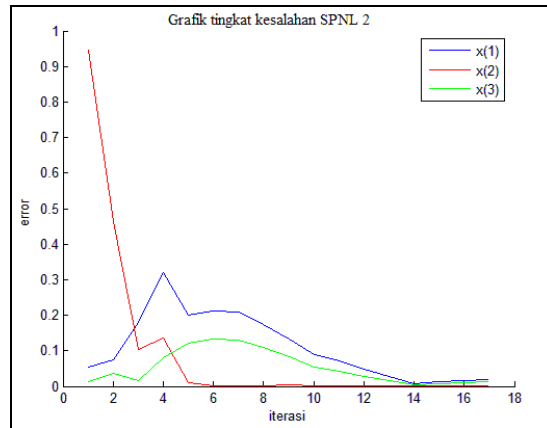
Dan dari Gambar 6 dapat dilihat juga bahwa tingkat kesalahan dari solusi yang didapat untuk sistem persamaan nonlinier 2 ini mendekati angka 0, hal ini menunjukkan bahwa solusi yang dihasilkan merupakan solusi yang tepat. Selisih terakhir antara solusi yang dihasilkan dari metode gabungan dengan solusi sebenarnya adalah (0.0147, 0.3087, 0.00902) dan solusi ini mengarah pada x_1^* .



Gambar 4 Grafik trayektori penyelesaian SPNL 2 dengan COA



Gambar 5 Grafik trayektori penyelesaian SPNL 2 dengan metode gabungan



Gambar 6 Grafik tingkat kesalahan SPNL 2

6.3 Sistem persamaan nonlinier 3

Bentuk dari sistem persamaan nonlinier 3 adalah sebagai berikut

$$f_1 = x_1 + x_4 - 3 = 0,$$

$$f_2 = 2x_1 + x_2 + x_4 + x_7 + x_8 + x_9 + 2x_{10} - R = 0,$$

$$f_3 = 2x_2 + 2x_5 + x_6 + x_7 - 8 = 0,$$

$$f_4 = 2x_3 + x_9 - 4R = 0,$$

$$f_5 = K_5 x_2 x_4 - x_1 x_5 = 0,$$

$$f_6 = K_6 x_2^{1/2} x_4^{1/2} - x_1^{1/2} x_6 (p / x_{11})^{1/2} = 0,$$

$$f_7 = K_7 x_1^{1/2} x_2^{1/2} - x_4^{1/2} x_7 (p / x_{11})^{1/2} = 0,$$

$$f_8 = K_8 x_1 - x_4 x_8 (p / x_{11}) = 0,$$

$$f_9 = K_9 x_1 x_3^{1/2} - x_4 x_9 (p / x_{11})^{1/2} = 0,$$

$$f_{10} = K_{10} x_1^2 - x_4^2 x_{10} (p / x_{11}) = 0,$$

$$f_{11} = x_{11} - \sum_{j=1}^{10} x_j = 0$$

Sistem persamaan nonlinier 3 digunakan untuk mencari solusi dari permasalahan “*Combustion of Propane*” [1]. Masalah kesetimbangan kimia ini menggambarkan pembakaran propane di udara. Terdapat 11 variabel, dimana x_i ($i = 1, 2, 3, \dots, 10$) adalah jumlah mol dari setiap produk yang terbentuk untuk setiap pembakaran propane. Variabel ke-11 digunakan untuk menyederhanakan persamaan dan merupakan jumlah dari 10 variabel yang lain. Terdapat 10 produk dari pembakaran yang ditunjukkan dengan f_i ($i = 1, 2, 3, \dots, 10$). Persamaan f_{11} merupakan syarat bahwa jumlah dari 10 variabel x_i ($i = 1, 2, 3, \dots, 10$) adalah sama dengan variabel ke-11 x_{11} .

Parameter yang sudah pasti adalah p (tekanan pada atmosfer) dan R (perbandingan udara terhadap bahan bakar). $K_5, K_6, K_7, K_8, K_9,$ dan K_{10} adalah ukuran data. Idealnya, diharapkan semua f_i ($i = 1, 2, 3, \dots, 11$) menjadi nol.

Parameter-parameter yang digunakan untuk sistem persamaan nonlinier ini adalah : $R = 10$, $p = 40$, $K_5 = 1.0$, $K_6 = 1.0$, $K_7 = 1.0$, $K_8 = 0.1$, $K_9 = 1.0$, $K_{10} = 0.1$.

Karena sistem persamaan nonlinier 3 menggunakan operasi " $\sqrt{\quad}$ ", maka x_1, x_2, x_3, x_4 , dan x_{11} harus lebih besar dari nol. Metode quasi-newton tidak bisa menjamin hal ini. Berdasarkan percobaan yang sudah dilakukan, jika program diimplementasi berdasarkan persamaan maka akan selalu terjadi keanehan. Sehingga, operasi " $\sqrt{\quad}$ " dirubah menjadi " $\sqrt{|\quad|}$ ".

Berdasarkan [1], solusi sebenarnya untuk sistem persamaan nonlinier 3 adalah

$$x_1^* = (1.378887, 1.372909, 18.426810, 1.621113, 1.614085, 1.094798, 0.931214, 0.063162, 3.146380, 0.053724, 29.703082)^T$$

$$x_2^* = (6.611605, 4.794332, 23.537556, -3.611605, -2.618915, 1.289160, 2.360007, -0.116171, -7.075112, 0.212670, 25.383526)^T$$

Ujicoba dilakukan dengan menggunakan k_{\max} 100, r_{\max} 100 dan batas ruang pencarian a dan b antara $[0,30]$.

Dari Gambar 7 dapat dilihat bahwa metode gabungan (COA dan quasi-Newton) dapat menunjukkan performa kestabilan dan konvergensi pada satu titik kesetimbangan yang menjadi solusi optimal dari sistem persamaan nonlinier 3 ini

Hasil perhitungan dari uji coba adalah sebagai berikut :

- Solusi COA : (5.81332, 8.54405, 4.91173, 0.898297, 10.7663, 17.7761, 15.6784, 0.45548, 13.6943, 8.14768, 21.3486)

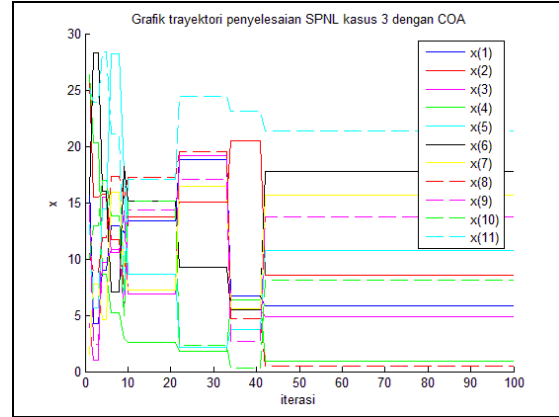
- Solusi akhir : (1.37888, 1.37291, 18.4268, 1.6211, 1.61408, 1.09479, 0.931209, 0.0631636, 3.14641, 0.0537237, 29.7031)

- Jumlah iterasi : 31

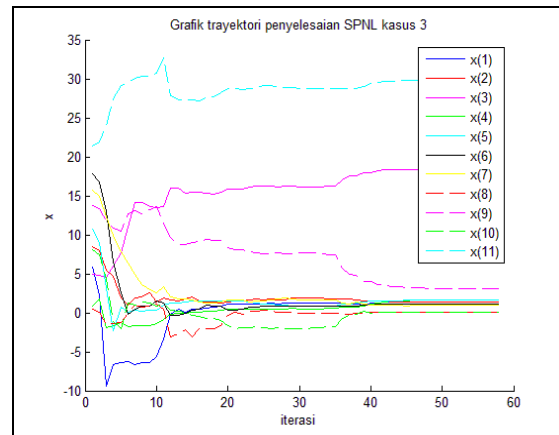
Gambar 8 menunjukkan tingkat kesalahan dari solusi yang dihasilkan metode penyelesaian terhadap solusi sebenarnya. Dari gambar tersebut terlihat bahwa tingkat kesalahan mendekati angka 0. Selisih hasil berdasarkan perhitungan adalah (0.000007, 0.000001, 0.00001, 0.000013, 0.000005, 0.000002, 0.000001, 0.0000006, 0.00002, 0.0000007, 0.000018). Dengan demikian solusi yang dihasilkan bisa dikatakan benar dan solusi ini sesuai dengan x_1^* .

Untuk mengetahui pengaruh dari variabel-variabel yang ada, maka dilakukan ujicoba untuk masing-masing sistem persamaan nonlinier dengan

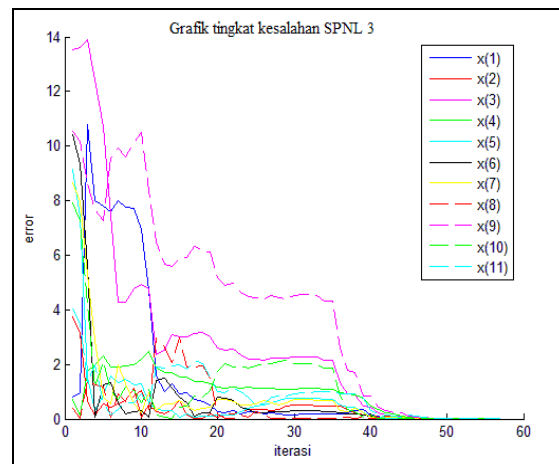
merubah nilai variabel-variabel tersebut. Variabel-variabel yang diuji adalah k_{\max} , r_{\max} , dan batas ruang pencarian a dan b . Khusus untuk sistem persamaan nonlinier 2, perubahan nilai M juga diperhitungkan.



Gambar 7 Grafik trayektori penyelesaian SPNL 3 dengan COA



Gambar 8 Grafik trayektori penyelesaian SPNL 3 dengan metode gabungan



Gambar 9 Grafik tingkat kesalahan SPNL 3

Dari beberapa uji coba yang dilakukan didapatkan hasil seperti pada Tabel 1, Tabel 2, Tabel 3, dan Tabel 4.

| Skenario 1 | | Skenario 2 | | Skenario 3 | |
|------------|------|------------|------|------------|------|
| Solusi | iter | Solusi | iter | Solusi | iter |
| -0.99997 | 56 | -1 | 43 | -1 | 20 |
| 0.99998 | | 1 | | | |
| -1 | | -1 | | | |
| 0.99998 | | 1 | | | |
| -1 | | -1 | | | |
| 0.99999 | | 1 | | | |

Tabel 1 Pengaruh kmax, rmax dan batas ruang pencarian terhadap solusi SPNL 1

Keterangan Tabel 1:

Skenario 1 = kmax 100, rmax 100, a = 0, b = 50

Skenario 2 = kmax 200, rmax 200, a = 0, b = 50

Skenario 3 = kmax 100, rmax 100, a = 0, b = 10

| Skenario 1 | | Skenario 2 | | Skenario 3 | |
|------------|------|------------|------|------------|------|
| Solusi | iter | Solusi | iter | Solusi | iter |
| 22.9195 | 89 | 22.9014 | 73 | 22.9096 | 26 |
| 12.2566 | | 12.2567 | | | |
| 2.77534 | | 2.78564 | | | |

Tabel 2 Pengaruh kmax, rmax dan batas ruang pencarian terhadap solusi SPNL 2

Keterangan Tabel 2:

Skenario 1 = kmax 100, rmax 100, a = 0, b = 100

Skenario 2 = kmax 200, rmax 200, a = 0, b = 100

Skenario 3 = kmax 100, rmax 100, a = 0, b = 30

| | Solusi | | | iter |
|----------|---------|---------|---------|------|
| M = 10 | 35.1752 | -2.0430 | 3.2220 | 39 |
| M = 50 | 32.1740 | 0.0709 | 4.7393 | 21 |
| M = 100 | 31.8380 | 0.3726 | 4.9731 | 56 |
| M = 500 | 25.7320 | 6.5049 | 10.4794 | 40 |
| M = 1000 | 25.3254 | 6.8690 | 10.8409 | 26 |

Tabel 3 Pengaruh M terhadap solusi SPNL 2

| Skenario 1 | | Skenario 2 | | Skenario 3 | |
|------------|------|------------|------|------------|------|
| Solusi | iter | Solusi | iter | Solusi | iter |
| 1.37888 | 75 | 1.37888 | 49 | 1.37888 | 31 |
| 1.37291 | | 1.37292 | | | |
| 18.4268 | | 18.4268 | | | |
| 1.6211 | | 1.62107 | | | |
| 1.61408 | | 1.61407 | | | |
| 1.0948 | | 1.09479 | | | |
| 0.931213 | | 0.931206 | | | |
| 0.0631626 | | 0.0631608 | | | |
| 3.1464 | | 3.14643 | | | |
| 0.0537247 | | 0.0537257 | | | |
| 29.7031 | | 29.7031 | | | |

Tabel 4 Pengaruh kmax, rmax dan batas ruang pencarian terhadap solusi SPNL 3

Keterangan Tabel 4:

Skenario 1 = kmax 100, rmax 100, a = 0, b = 100

Skenario 2 = kmax 200, rmax 200, a = 0, b = 100

Skenario 3 = kmax 100, rmax 100, a = 0, b = 30

Dari Tabel 1, Tabel 2, dan Tabel 4 terlihat bahwa jumlah iterasi maksimum untuk algoritma COA cukup berpengaruh terhadap solusi yang dihasilkan. Hal ini dapat dilihat dari jumlah iterasi dan selisih antara solusi yang dihasilkan dengan solusi sebenarnya. Semakin besar iterasi maksimum, maka semakin akurat solusi yang dihasilkan. Semakin sempit batas ruang pencarian, maka semakin mudah solusi ditemukan.

Tabel 3 menunjukkan pengaruh nilai *penalty* M terhadap solusi yang dihasilkan pada sistem persamaan nonlinier 2. Uji coba dilakukan dengan menggunakan nilai kmax = 100, rmax = 100, a = 0, dan b = 30. Nilai perkiraan awal solusi tidak diambil dari algoritma COA, tetapi ditetapkan bernilai (32, 3, 7)^T. Dari tabel tersebut terlihat bahwa hanya uji coba dengan M = 10 yang dapat memberikan hasil yang sesuai dengan solusi sebenarnya, dengan demikian nilai M yang optimal untuk sistem persamaan nonlinier 2 adalah 10.

7. Kesimpulan

Setelah dilakukan uji coba dan analisis terhadap perangkat lunak yang dibuat, maka dapat diambil simpulan sebagai berikut:

1. Metode gabungan dari algoritma optimasi chaos (COA) dan metode quasi-Newton dapat menyelesaikan sistem persamaan nonlinier dengan baik.
2. Algoritma optimasi chaos mampu bekerja dengan baik dalam mendapatkan nilai perkiraan awal solusi yang selanjutnya digunakan sebagai inputan pada metode quasi-Newton. Konvergensi untuk algoritma optimasi chaos ini terjadi pada jumlah iterasi yang besar. Hal ini dapat dilihat dari hasil ujicoba. Semakin besar nilai k_{max} dan r_{max} maka solusi yang dihasilkan semakin mendekati solusi sebenarnya sehingga memudahkan dalam penghitungan menggunakan metode quasi-Newton.
3. Perubahan nilai batas ruang pencarian (*search space*) sangat mempengaruhi keberhasilan metode. Ketika batas ruang pencarian semakin sempit dan semakin mendekati batas ruang solusi, maka solusi sebenarnya akan semakin mudah didapatkan.
4. Khusus untuk sistem persamaan nonlinier 2, nilai koefisien *penalty* juga mempengaruhi keberhasilan metode. Koefisien *penalty* harus

maksimal bernilai 10 untuk bisa menyelesaikan sistem persamaan nonlinier 2 ini.

8. Daftar Pustaka

- [1] Luo, Ya-Zhong, Guo-Jin Tang, Li-Ni Zhou. (2007). **Hybrid approach for solving systems of nonlinear equations using chaos optimization and quasi-Newton method.** ScienceDirect, Applied Soft Computing 8 (2008), 1068-1073.
- [2] Venkataraman, P. (2002). **Applied Optimization with Matlab Programming.** John Wiley & Sons, New York.
- [3] Arora, Jasbir. S. (2000). **Introduction to Optimum Design.**
- [4] Ding, Yang, Enkeleida Lushi, Qingguo Li. **Investigation of Quasi-Newton Methods for Unconstrained Optimization.** Simon Fraser University, Canada.
- [5] Nash, Stephen G., Ariela Sofer. (1996). **Linear and Nonlinear Programming.** McGraw-Hill, USA.
- [6] Rao, S.S. (1985). **Optimization, Theory and Applications (Second Edition).** Wiley Eastern Limited, New Delhi.
- [7] Hilborn, Robert C. (1994). **Chaos and Nonlinear Dynamics.** Oxford University Press, New York.
- [8] Rasband, S. Neil. (2000). **Chaotic Dynamics of Nonlinear Systems.** John Wiley & Sons, New York.
- [9] Jiaqiang, Chunhua Wang, Yaonan Wang, Jinke Gong. (2008). **A New Adaptive Mutative Scale Chaos Optimization Algorithm and its Application.** [J] Control Theory and Application, 2008, 6(2) : 141-145.
- [10] Davendra, Donald, Ivan Zelinka, Godfrey Onwubolu. (2007). **Chaotic Optimization.** Proceedings 21st European Conference on Modelling and Simulation.
- [11] Teng, Hao, Baohua Zhao, Bingru Yang, Bin He. **Study of Quantum Genetic Algorithm Based on Mutative Scale Chaotic Optimization.** China.
- [12] Hui-Juan, Lu, Zhang Huo-ming, Ma Long-hua. (2006). **A new Optimization Algorithm Based on Chaos.** Journal of Zhejiang University Science A, ISSN 1009-3095.