Studi Pencarian Akar Solusi Persamaan Nirlanjar Dengan Menggunakan Metode Brent

Tommy Gunardi / 13507109[1]
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
[1]*tommy_gunardi@hotmail.com*

***Abstract*—Dewasa ini, banyak sekali masalah-masalah matematis yang sangat kompleks dalam berbagai bidang kerja. Masalah-masalah ini membutuhkan jawaban yang tepat dalam waktu yang sesingkat-singkatnya. Persoalan matematis tersebut seringkali tidak dapat diselesaikan secara analitik karena terlalu rumit. Oleh karena itu dibuatlah metoda-metoda numerik yang dapat menyelesaikan masalah-masalah matematis tersebut. Masalah matematis tersebut mencakup beberapa permasalahan seperti pencarian solusi persamaan lanjar, pencarian solusi persamaaan nirlanjar, persoalan interpolasi, persoalan integrasi dan persoalan diferensiasi. Salah satu permasalahan yang akan dibahas pada makalah ini adalah pencarian akar solusi persamaan nirlanjar dengan menggunakan metode Brent. Metode ini menggunakan beberapa metode lain dalam pencarian akar. Untuk lebih jelasnya akan dijelaskan pada bagian isi makalah ini.**

***Index Terms*—numerik, metode Brent, persamaan nirlanjar, penggunaan beberapa metode lain.**

I. INTRODUCTION

Perusahaan - perusahaan menggunakan grafik untuk menggambarkan data-data seperti data penjualan dan data pembelian agar terlihat lebih jelas. Bahkan grafik dapat digunakan untuk memperlihatkan apakah perusahaan itu mengalami keuntungan atau kerugian. Grafik-grafik tersebut dapat berupa garis lurus, atau kurva. Grafik yang berbentuk kurva biasanya dapat direpresentasikan dengan persamaan nirlanjar. Persoalan mencari solusi persamaan nirlanjar dapat dirumuskan secara singkat sebagai berikut

$$f(x) = 0$$

yaitu bilangan x = s sedemikian rupa sehingga f(s) sama dengan nol. Di sini f adalah persamaan nirlanjar. Dalam metode numerik, ada kalanya dimana persamaan terlalu rumit sehingga kita tidak dapat menggunakan rumus-rumus analitik untuk mencari solusi persamaan nirlanjar tersebut. Bila metode analitik tidak dapat menyelesaikan persamaan, maka alternatif pencarian solusi yang ditawarkan adalah secara numerik[RIN1997].

Dalam metode numerik, pencarian akar fungsi persamaan nirlanjar dilakukan secara lelaran. Metode-metode yang digunakan dalam pencarian akar secara lelaran ini dapat dikelompokkan menjadi dua yaitu:

1. **Metode tertutup**
   Metode ini sering disebut metode pengurung (*bracketing method*). Metode yang termasuk dalam golongan ini mencari akar dari persamaannya dalam selang [a, b]. Selang [a, b] dikondisikan harus mengandung akar. Sehingga lelaran terhadap selan ini selalu konvergen menuju akar. Beberapa contoh metode ini antara lain Metode Bagidua dan Metode Regula-Falsi.

2. **Metode terbuka**
   Metode ini tidak memerlukan selang [a, b] yang mengandung akar. Yang dibutuhkan metode ini hanyalah tebakan awal akar, yang digunakan untuk menghitung hampiran akar yang baru dengan menggunakan prosedur lelaran. Pada setiap lelaran, hampiran akar yang lama digunakan sebagai masukan untuk mendapatkan hampiran akar yang baru. Hampiran akar yang baru tidak selalu konvergen ke akar, melainkan divergen. Beberapa contoh metode ini adalah Metode Lelaran Titik-Tetap, Metode Newton-Raphson dan Metode Secant.

Metode-metode di atas memiliki kelebihan dan kelemahannya masing-masing. Contohnya metode Bagidua pasti akan menemukan akar, namun menggunakan banyak sekali lelaran. Metode-metode tertutup dapat tidak menemukan akar apabila tebakan awal akar tidak valid. Oleh karena itu pada makalah ini akan dibahas mengenai Metode Brent, salah satu metode pencarian akar yang menggabungkan beberapa metode sekaligus antara lain metode Bagidua, Secant dan *Inverse Quadratic Interpolation*.

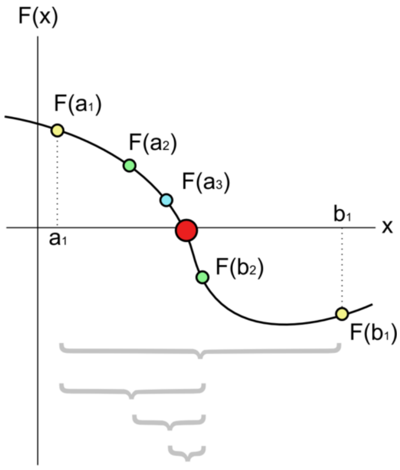
II. METODE PENCARIAN AKAR

Pada bagian ini akan dibahas beberapa metode pencarian akar persamaan nirlanjar yang sudah lama digunakan dan berkaitan dengan metode Brent. Metode-metode tersebut adalah metode Bagidua, Secant dan

Inverse Quadratic Interpolation. Berikut penjelasan masing-masing metode yang ada :

1. **Metode Bagidua**

   Metode ini merupakan salah satu metode tertutup klasik [RIN1997]. Misalkan ditentukan selang [*a*, *b*] sehingga *f(a)f(b)* < 0. Pada setiap kali lelaran, selang [*a*, *b*] dibagi dua pada *x* = *c*, sehingga terdapat dua buah upaselang yang berukuran sama, yaitu selang [*a*, *c*] dan [*c*, *b*]. Selang yang diambil untuk lelaran berikutnya adalah selang yang memuat akar, bergantung pada *f(a)f(c)* < 0 atau *f(c)f(b)* < 0.



Gambar 1 Metode Bagidua

Selang yang baru dibagi dua lagi dengan cara yang sama. Begitu seterusnya sampai ukuran selang yang baru sangat kecil. Lelaran berhenti jika |*a* - *b*| < *e*, dimana e adalah lebar selang sempit yang diinginkan (epsilon). Algoritma Bagidua adalah sebagai berikut :

```
procedure bagidua(a, b: real; var c:
real);
const
  epsilon=0.000001;
begin
  repeat
     c:= (a+b)/2; /* midpoint [a, b] */
     if f(a)*f(c) < 0 then
         b:= c;
     else
         a:= c;
  until ABS(a-b) < epsilon;
end;
```

Beberapa keuntungan dari metode ini adalah :

a. Selalu konvergen
b. Batas kesalahannya jelas dan semakin menurun setiap lelaran berhasil

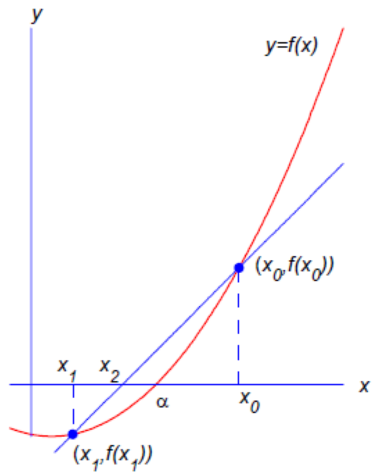
Beberapa kerugian dari metode Bagidua adalah :

a. Lambat apabila dibandingkan dengan metode pencarian akar yang lain.
b. Algoritma tidak memeriksa apakah epsilon terlalu kecil dibandingkan dengan komputer aritmatik yang digunakan.

2. **Metode Secant**

   Metode ini merupakan modifikasi dari metode Newton-Raphson yang dianggap kurang efektif karena harus menggunakan turunan fungsi, sedangkan tidak semua fungsi mudah dicari turunannya. Maka dengan metode ini, turunan fungsi dapat dihilangkan dan diganti dengan bentuk lain yang ekivalen.

Metode Newton berbasis *line tangent* terhadap kurva y = f(x) dengan titik (x0, f (x0)). When x0 ≈ α, grafik dari garis tangent dihampiri seperti grafik y = f (x) sekitar x = α. Kemudian digunakan akar dari garis tangent untuk menghampiri α. Grafik dapat dilihat di bawah ini:



Gambar 2 Metode Secant

Diasumsikan penggunaan garis hampiran berdasarkan interpolasi. Diambil perkiraan terhadap akar α, misalnya x0 and x1. Kemudian dihasilkan fungsi lanjar:

$$q(x) = a_0 + a_1 x \tag{1}$$

dengan

$$q(x_0) = f(x_0), \qquad q(x_1) = f(x_1) \tag{2}$$

Garis ini kadang disebut *secant line*. Equationnya diberikan sebagai berikut :

$$q(x) = \frac{(x_1 - x)\, f(x_0) + (x - x_0)\, f(x_1)}{x_1 - x_0} \tag{3}$$

fungsi lanjar pada x dan dengan evaluasi arah, ini memenuhi kondisi interpolasi dari (2). Selanjutnya diselesaikan permasalahan *q(x)* = 0, denoting the root by x^2. Dihasilkan

$$x_2 = x_1 - f(x_1) \div \frac{f(x_1) - f(x_0)}{x_1 - x_0} \quad (4)$$

Lalu dilakukan lelaran terhadap prose. Digunakan x1 dan x2 untuk menghasilkan secant line yang baru, kemudian akarnya digunakan untuk menghampiri α. Dihasilkan rumus lelaran :

$$x_{n+1} = x_n - f(x_n) \div \frac{f(x_n) - f(x_{n-1})}{x_n - x_{n-1}}, \quad (5)$$

untuk nilai n = 1, 2, 3,…

Metode ini juga dapat dihampiri dari Metode Newton-Raphson :

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \quad (6)$$

Dengan menggunakan hampiran :

$$f'(x_n) \approx \frac{f(x_n) - f(x_{n-1})}{x_n - x_{n-1}} \quad (7)$$

Berikut dapat kita lihat pseudocode algoritma metode Secant:

```
procedure secant(x0,x1:real; var x =
 real);
const
  epsilon = 0.000001;
var
  x_sebelumnya : real;
begin
  repeat
      x_sebelumnya:= x1;
      x:=x-(f(x1)*(x1 - x0)/(f(x1)-
      f(x0)));
      x0:= x1;
      x1:= x;
  until(ABS(x-x_sebelumnya)< epsilon);
end;
```

Metode ini merupakan modifikasi metode Newton-Raphson. Metode ini memiliki beberapa keuntungan sebagai berikut :

a. Konvergen dengan cepat. Konvergensinya jauh lebih cepat dibandingkan metode Bagidua.
b. Tidak membutuhkan turunan dari fungsi, karena kadang sulit mendapatnkan turunan dari fungsi.
c. Hanya membutuhkan sebuah fungsi evaluasi per lelaran, dibandingkan dengan metode Newton yang membutuhkan dua.

Metode ini juga memiliki beberapa kelemahan. Beberapa kelemahan metode Secant adalah sebagai berikut:

a. Ada kemungkinan tidak konvergen
b. Tidak ada jaminan batas error untuk setiap lelaran
c. Metode Newton secara umum lebih mudah untuk menyelesaikan sistem yang simultan untuk persamaan nonlanjar.

3. Metode *Inverse Quadratic Interpolation*

Metowde ini merupakan salah satu algoritma pencari akar, suatu algoritma untuk memecahkan formulasi f(x) = 0. Ide untuk menggunakan interpolasi kuadratik untuk menghampiri invers dari fungsi. Algoritma ini jarang digunakan sendiri, namun penting karena merupakan bagian dari metode Brent.

Metode ini didefinisikan sebagai berikut :

$$x_{n+1} = \frac{f_{n-1}f_n}{(f_{n-2}-f_{n-1})(f_{n-2}-f_n)}x_{n-2} + \frac{f_{n-2}f_n}{(f_{n-1}-f_{n-2})(f_{n-1}-f_n)}x_{n-1} + \frac{f_{n-2}f_{n-1}}{(f_n-f_{n-2})(f_n-f_{n-1})}x_n \quad (8)$$

Dimana $f_k = f(x_k)$. Seperti yang dapat dilihat dari relasi di atas, metode ini membutuhkan tiga nilai awal, x_0, x_1 dan x_2.

Secara umum, lelaran x_n konvergen dengan cepat ke akar ketika sudah mendekati. Namun, performanya biasanya cukup lemah jika tidak dimulai cukup dekat dengan akar. Oleh karena itu *inverse quadratic interpolation* jarang digunakan sebagai algoritma yang berdiri sendiri.

Aturan konvergensinya mendekati 1.8, dan dapat dibuktikan dengan analisis secant.

Dirancanglah suatu metode yang menggabungkan ketiga metode di atas menjadi sebuah metode pencarian akar persamaan nirlanjar yang ditemukan oleh Richard Brent pada tahun 1973 dengan memodifikasi algoritma yang dibuat oleh Theodorus Dekker pada tahun 1969. Berikut detail algortima Dekker dan Brent :

1. Metode Dekker

Metode ini pertama kali mencetuskan ide untuk menggabungkan metode Bagidua dengan metode Secant. Misalkan ingin diselesaikan suatu *equation* $f(x) = 0$. Seperti metode Bagidua, perlu dilakukan inisialisasi selang [a, b] untuk metode Dekker dimana $f(a)$ dan $f(b)$ memiliki tanda yang berlawanan. Jika f kontinu dalam selang [a, b] maka akan ada solusi diantara a dan b. Tiga titik yang dibutuhkan dalam setiap lelaran :

a. b_k adalah lelaran saat ini dalam kata lain tebakan awal untuk akar f.
b. a_k adalah “contrapoint” dalam kata lain suatu

titik dimana $f(a_k)$ dan $f(b_k)$ memiliki tanda yang berlawanan, sehingga interval $[a_k, b_k]$ berisi solusi. Selanjutnya, $|f(b_k)|$ sebaiknya lebih kecil sama dengan $|f(a_k)|$, sehingga b_k menjadi tebakan yang lebih baik sebagai solusi yang tidak diketahui daripada a_k.

c. b_{k-1} adalah lelaran sebelumnya (untuk lelaran pertama, kita isi b_{k-1} = a).

Selanjutnya, dua buah nilai sementara untuk lelaran selanjutnya dikomputasi. Nilai pertama menggunakan metode Secant :

$$s = b_k - \frac{b_k - b_{k-1}}{f(b_k) - f(b_{k-1})} f(b_k), \qquad (9)$$

dan nilai yang kedua diberikan oleh metode Bagidua :

$$m = \frac{a_k + b_k}{2} \qquad (10)$$

Jika hasil dari metode Secant, *s*, berada antara b_k dan *m*, maka ia akan menjadi lelaran selanjutnya ($b_{k+1} = s$) . Jika tidak, titik tengahlah yang digunakan ($b_{k+1} = m$). Selanjutnya, nilai dari *contrapoint* yang baru dipilih sedemikian rupa sehingga $f(a_{k+1})$ dan $f(b_{k+1})$ memiliki tanda yang berlawanan. Jika $f(a_k)$ dan $f(b_{k+1})$ memiliki tanda yang berlawanan, maka *contrapoint* tetap sama : $a_{k+1} = a_k$. Sebaliknya, jika $f(b_{k+1})$ dan $f(b_k)$ memiliki tanda yang berlawanan, contrapoint yang baru menjadi $a_{k+1} = b_k$.

Akhirnya, jika $|f(a_{k+1})| < |f(b_{k+1})|$, maka a_{k+1} kemungkinan adalah tebakan yang lebih baik untuk solusi daripada b_{k+1}, oleh karena itu nilai dari a_{k+1} dan b_{k+1} ditukar.

2. Metode Brent

Metode Brent merupakan modifikasi dari Metode Dekker. Metode Dekker memiliki performa yang baik jika fungsi f cukup *well-behaved.* Bagaimanapun juga, ada kondisi dimana setiap lelaran menggunakan metode Secant, namun proses lelaran b_k berjalan sangat lambat. Bahkan metode Dekker membutuhkan lelaran yang jauh lebih banyak daripada metode Bagidua pada kasus ini.

Metode Brent mengajukan modifikasi kecil untuk mengatasi masalah ini. Brent mengajukan pemeriksaan tambahan yang harus memuaskan sebelum hasil metode Secant diterima sebagai lelaran selanjutnya. Dua ketidaksamaan yang secara simultan harus diterima adalah sebagai berikut:

a. Diberikan sebuah toleransi numerik spesifik δ, jika langkah sebelumnya menggunakan metode Bagidua, pertidaksamaan $|\delta| < |b_k - b_{k-1}|$ harus dijaga, jika tidak metode Bagidua dieksekusi dan hasilnya digunakan untuk iterasi selanjutnya. Jika langkah sebelumnya adalah interpolasi, maka pertidaksamaan $|\delta| < |b_{k-1} - b_{k-2}|$ yang digunakan.

b. Jika langkah sebelumnya menggunakan metode Bagidua, pertidaksamaan $|s - b_k| < {}^1/_2 |b_{k-1} - b_{k-2}|$. harus dijaga, jika tidak metode Bagidua dieksekusi dan hasilnya digunakan untuk lelaran selanjutnya. Jika langkah sebelumnya adalah interpolasi, maka yang digunakan adalah pertidaksamaan $|s - b_k| < {}^1/_2 |b_{k-1} - b_{k-2}|$.

Modifikasi ini menjamin pada lelaran ke k^{th}, langkah bagidua akan dieksekusi paling banyak $2\log_2(|b_{k-1} - b_{k-2}| / \delta)$ lelaran tambahan, karena kondisi di atas memaksa ukuran langkah interpolasi yang berturut-turut terbagi setiap dua lelaran, dan setelah paling banyak $2\log_2(|b_{k-1} - b_{k-2}| / \delta)$ lelaran, ukuran langkah akan menjadi lebih kecil dari δ, yang memanggil langkah bagidua. Brent membuktikan bahwa metodenya membutuhkan paling banyak N2 lelaran, dimana N menunjukkan angka dari lelaran untuk metode Bagidua. Jika fungsi f *well-behaved*, maka metode Brent biasanya akan diproses dengan inverse quadratic atau interpolasi lanjar, dalam kasus ini dikatakan *converge superlinearly*.

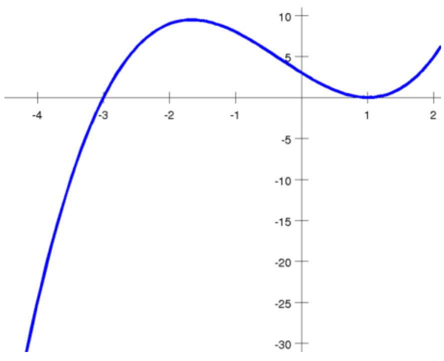
Selanjutnya, metode Brent menggunakan *inverse quadratic interpolation* daripada interpolasi lanjar (yang digunakan pada metode Secant) Jika $f(b_k)$, $f(a_k)$, dan $f(b_{k-1})$ berbeda untuk meningkatkan efisiensi. Sebagai konsekuensinya, kondisi untuk menerima s (nilai yang diajukan interpolasi lanjar atau *inverse quadratic interpolation*) harus dirubah : *s* harus berada antara $(3a_k + b_k) / 4$ and b_k.

III. METODE BRENT

A. Contoh Metode Brent

Pada bagian ini akan dijelaskan mengenai langkah-langkah pada metode Brent. Asumsikan dicari sebuah 0 dari fungsi yang terdefinisi sebagai berikut :

$$f(x) = (x + 3)(x - 1)^2$$



Gambar 3 kurva fungsi f(x)

Diambil $[a_0, b_0] = [-4, 4/3]$ sebagai interval awal. Data lainnya adalah $f(a_0) = -25$ dan $f(b_0) = 0.48148$ (semua angka merupakan bilangan bulat), sehingga kondisi $f(a_0)$ $f(b_0) < 0$ dan $|f(b_0)| \leq |f(a_0)|$ terpenuhi.

1. Pada lelaran pertama, digunakan interpolasi lanjar antara $(b_{-1}, f(b_{-1})) = (a_0, f(a_0)) = (-4, -25)$ dan $(b_0, f(b_0)) = (1.33333, 0.48148)$, yang menghasilkan $s = 1.23256$. Nilai s berada di antara $(3a_0 + b_0) / 4$ dan b_0, maka nilai s diterima. Selanjutnya, $f(1.23256) = 0.22891$, maka diisi $a_1 = a_0$ dan $b_1 = s = 1.23256$.

2. Pada lelaran kedua, digunakan *inverse quadratic interpolation* antara $(a_1, f(a_1)) = (-4, -25)$ dan $(b_0, f(b_0)) = (1.33333, 0.48148)$ dan $(b_1, f(b_1)) = (1.23256, 0.22891)$. Dihasilkan s = 1.14205, berada di antara $(3a_1 + b_1) / 4$ dan b_1. Selanjutnya, pertidaksamaan $|1.14205 - b_1| \leq |b_0 - b_{-1}| / 2$ terpenuhi, maka nilainya diterima. Selanjutnya, $f(1.14205) = 0.083582$, maka diisi $a_2 = a_1$ dan $b_2 = 1.14205$.

3. Pada lelaran ketiga, digunakan *inverse quadratic interpolation* antara $(a_2, f(a_2)) = (-4, -25)$ dan $(b_1, f(b_1)) = (1.23256, 0.22891)$ dan $(b_2, f(b_2)) = (1.14205, 0.083582)$. Dihasilkan s = 1.09032, yang berada di antara $(3a_2 + b_2) / 4$ dan b_2. Namun, kondisi pertidaksamaan $|1.09032 - b_2| \leq |b_1 - b_0| / 2$ tidak terpenuhi, maka nilai s ditolak. Sebagai gantinya, titik tengah $m = -1.42897$ dari interval $[a_2, b_2]$ dikomputasi. Didapatkan $f(m) = 9.26891$, maka diisi $a_3 = a_2$ dan $b_3 = -1.42897$.

4. Pada lelaran keempat, digunakan *inverse quadratic interpolation* antara $(a_3, f(a_3)) = (-4, -25)$ dan $(b_2, f(b_2)) = (1.14205, 0.083582)$ dan $(b_3, f(b_3)) = (-1.42897, 9.26891)$. Dihasilkan s = 1.15448, dimana tidak berada pada interval antara $(3a_3 + b_3) / 4$ and b_3). Oleh karena itu, nilainya diganti dengan titik tengah $m = -2.71449$. Didapatkan $f(m) = 3.93934$, maka diisi $a_4 = a_3$ dan $b_4 = -2.71449$.

5. Pada lelaran kelima, *inverse quadratic interpolation* menghasilkan -3.45500, yang berada pada interval. Namun, lelaran sebelumnya merupakan langkah bagidua, maka pertidaksamaan $|-3.45500 - b_4| \leq |b_4 - b_3| / 2$ harus dipenuhi. pertidaksamaannya tidak terpenuhi, maka digunakan titik tengah $m = -3.35724$. Didapatkan $f(m) = -6.78239$, maka m menjadi *contrapoint* ($a_5 = -3.35724$) dan lelaran tetap sama ($b_5 = b_4$).

6. Pada lelaran keenam, *inverse quadratic interpolation* tidak dapat digunakan karena $b_5 = b_4$. Oleh karena itu, digunakan interpolasi lanjar antara $(a_5, f(a_5)) = (-3.35724, -6.78239)$ dan $(b_5, f(b_5)) = (-2.71449, 3.93934)$. Dihasilkan $s = -2.95064$, yang memenuhi semua kondisi. Namun, karena lelaran tidak berubah dalam langkah sebelumnya, hasil ini ditolak dan kembali menggunakan Bagidua. Diubah nilai s = -3.03587, dan $f(s)$ = -0.58418.

7. Pada iterasi ketujuh, *inverse quadratic interpolation* kembali dapat digunakan. Dihasilkan $s = -3.00219$, yang memenuhi semua kondisi. Dengan $f(s) = -0.03515$, maka diisi $a_7 = b_6$ dan $b_7 = -3.00219$ (a_7 dan b_7 ditukar sehingga kondisi $|f(b_7)| \leq |f(a_7)|$ terpenuhi). (Dibenarkan : interpolasi lanjar *s = -2.99436, f(s) = 0.089961)*

8. Pada lelaran kedelapan, *inverse quadratic interpolation* tidak dapat digunakan karena $a_7 = b_6$. Interpolasi lanjar menghasilkan $s = -2.99994$, yang nilainya diterima. (Dibenarkan *: s = -2.9999, f(s) = 0.0016)*

9. Pada lelaran selanjutnya, akar $x = -3$ dihampiri dengan sangat cepat: $b_9 = -3 + 6 \cdot 10^{-8}$ dan $b_{10} = -3 - 3 \cdot 10^{-15}$. (Dibenarkan *:* lelaran *9 : f(s) = -1.4E-07*, Lelaran *10 : f(s) = 6.96E-12)*

B. Algoritma Brent

Bagian ini akan menjelaskan algoritma dari Metode Brent. Berikut pseudocode dari metode Brent :

```
Input a, b: real; var c: real
calculate f(a)
calculate f(b)
if f(a) f(b) >= 0 then error-exit end if
if |f(a)| < |f(b)| then swap (a,b) end if
c := a
set mflag
repeat until f(b or s) = 0 or |b − a| is
small enough /* convergence */
  if f(a) ≠ f(c) and f(b) ≠ f(c) then
```

$$s := \frac{af(b)f(c)}{(f(a)-f(b))(f(a)-f(c))} + \frac{bf(a)f(c)}{(f(b)-f(a))(f(b)-f(c))} + \frac{cf(a)f(b)}{(f(c)-f(a))(f(c)-f(b))}$$

```
  else
```

$$s := b - f(b)\frac{b-a}{f(b)-f(a)}$$

```
  end if
  if (con 1) s is not between (3a + b)/4
  and b
   or (con 2) (mflag is set and |s−b| ≥
   |b−c| / 2)
   or (con 3) (mflag is cleared and |s−b| ≥
   |c−d| / 2)
   or (con 4) (mflag is set and |b−c| < |δ|)
   or (con 5) (mflag is cleared and |c−d| <
   |δ|)
   then
      s := (a+b)/2
      set mflag
```

```
  else
    clear mflag
  end if
  calculate f(s)
  d := c /* d is assigned for the first
  time here; it won't be used above on the
   first iteration because mflag is set*/
  c := b
  if f(a) f(s) < 0 then b := s else a := s
  end if
  if |f(a)| < |f(b)| then swap (a,b) end if
end repeat
output b or s /* return the root */
```

IV. Analisis dan Kesimpulan

Pada bab ini akan dilakukan analisis perbandingan hasil metode Brent dan metode Bagidua. Metode Brent di sini menggunakan fungsi *fzero* pada MatLab. Sedangkan metode Bagidua menggunakan program yang dibuat dalam tugas pertama metode numerik. Berikut hasil analisis :

1. $f(x) = e^x – 3x^2$
   Metode Brent : 0.910007572488709
   Metode Bagidua : 0.910007476806640

2. $f(x) = \sin(x) – 0.3e^x$
   Metode Brent : 0.541948113962968
   Metode Bagidua : 0.541949462890625

Berdasarkan hasil analisis terhadap metode Brent dan metode Bagidua di atas, didapat kesimpulan sebagai berikut :

1. Nilai yang diperoleh metode Brent dan metode Bagidua menunjukkan hasil yang hampir sama.

2. Dengan hasil yang sama, metode Brent lebih baik dengan jumlah lelaran yang lebih sedikit.

References

Munir, Rinaldi., "Metode Numerik untuk Teknik Informatika," Makalah Teknik Informatika ITB, 1997. Dibuka pada tanggal 10 Mei 2011.

http://www.cs.uiowa.edu/ftp/atkinson/ENA_Materials/Overheads/sec_3-3.pdf. Dibuka pada tanggal 9 Mei 2011.

http://www.mathworks.com/moler/chapters.html. Dibuka pada tanggal 12 Mei 2011.

http://reference.wolfram.com/mathematica/tutorial/UnconstrainedOptimizationBrentsMethod.html. Dibuka tanggal 12 Mei 2011.

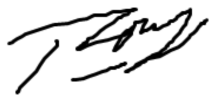
http://pathfinder.scar.utoronto.ca/~dyer/csca57/book_P/node36.html. Dibuka pada tanggal 12 Mei 2011.

Pernyataan

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 13 Mei 2011

ttd



Tommy Gunardi
13507109