3 Akar Persamaan Tak Linier $f(x) = 0$

Salah satu masalah yang paling umum ditemui di dalam matematika dan teknik adalah mencari akar suatu persamaan; yakni jika diketahui fungsi $f(x)$, akan dicari nilai-nilai x yang memenuhi $f(x) = 0$. Termasuk dalam masalah ini adalah menentukan titik-titik potong dua buah kurva. Apabila kurva-kurva tersebut dinyatakan oleh fungsi $f(x)$ dan $g(x)$, maka absis titik-titik potong kedua kurva tersebut merupakan akar-akar persamaan $f(x) - g(x) = 0$.

Definisi 3.1 (Akar suatu persamaan, Pembuat nol fungsi).
Misalkan $f(x)$ adalah suatu fungsi kontinyu. Setiap bilangan r pada domain f yang memenuhi $f(r) = 0$ disebut **akar persamaan** *$f(x) = 0$, atau juga disebut* **pembuat nol fungsi** *$f(x)$. Secara singkat, r sering disebut* **akar fungsi** *$f(x)$.*

Akar persamaan, pembuat nol suatu fungsi

Sebagai contoh, persamaan $2x^2 + 5x - 3 = 0$ mempunyai dua buah akar nyata $r_1 = 0.5$ dan $r_2 = -3$. Kita tahu bahwa $f(x) = 2x^2 + 5x - 3 = (2x - 1)(x + 3)$ sehingga $r_1 = 0.5$ dan $r_2 = -3$ merupakan pembuat nol $f(x)$.

Khusus untuk fungsi linier berbentuk $f(x) = ax + b$ dengan $a \neq 0$ dan b adalah konstanta-konstanta riil yang diketahui, kita tahu pembuat nol fungsi tersebut adalah $x = -b/a$.

Demikian juga, untuk fungsi kuadrat berbentuk $f(x) = ax^2 + bx + c$ dengan $a \neq 0$, b, dan c adalah konstanta-konstanta riil yang diketahui, kita mempunyai rumus yang sangat terkenal, yakni rumus "abc" untuk mencari akar persamaan kuadrat, yakni

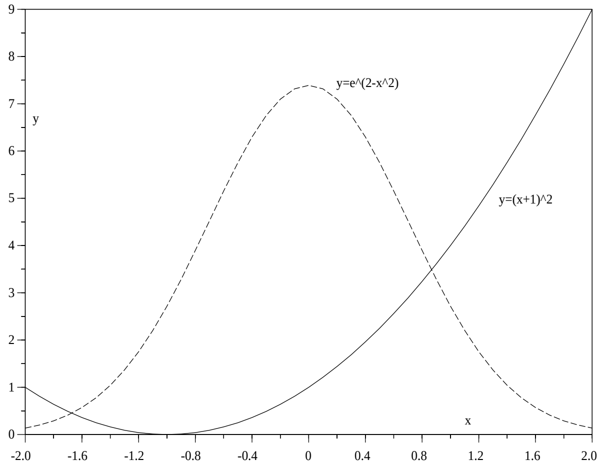
$$x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}. \tag{3.1}$$

Dalam beberapa kasus kita mempunyai fungsi di mana kita tidak dapat menentukan secara eksplisit pembuat nol fungsi tersebut.
Sebagai contoh, kita tidak mempunyai rumus eksak untuk mencari akar

persamaan

$$(x+1)^2 e^{x^2-2} - 1 = 0. \tag{3.2}$$

Di sinilah peranan metode numerik di dalam memberikan hampiran suatu penyelesaian permasalahan matematis yang tidak dapat diselesaikan secara analitik, atau jika mungkinpun diperlukan perhitungan yang sangat rumit.



Gambar 3.1: Akar suatu persamaan merupakan absis titik potong dua buah kurva yang diperoleh dari persamaan yang bersangkutan

Kalau kita amati lebih jauh persamaan (3.2) dapat ditulis ke dalam bentuk

$$(x+1)^2 = e^{2-x^2}. \tag{3.3}$$

Gambar 3.1 menunjukkan grafik kedua kurva $y = (x+1)^2$ dan $y = e^{2-x^2}$ pada interval $[-2,\ 2]$. Ternyata kedua kurva tersebut berpotongan di dua titik, sehingga persamaan di atas memiliki dua buah akar. Bagaimanakah cara menentukan akar-akar tersebut?

DEFINISI 3.2.
Misalkan r adalah akar persamaan $f(x) = 0$. Jika terdapat bilangan asli m dan

$0.8668945 - 0.8668457 = 0.0000488$ *(masih lebih besar daripada nilai toleransi yang diberikan, 0.00001.* □

Tabel 3.1: Hasil iterasi dengan metode Pengapitan Akar untuk $f(x) = (x+1)^2 e^{x^2-2} - 1$

Iterasi	a	b	x	$(x+1)^2 e^{x^2-2} - 1$
1	0.4	1.2	0.8	- 0.1684191
2	0.8	1.2	1.	0.4715178
3	0.8	1.	0.9	0.0982388
4	0.8	0.9	0.85	- 0.0460354
5	0.85	0.9	0.875	0.0231052
6	0.85	0.875	0.8625	- 0.0121796
7	0.8625	0.875	0.86875	0.0052800
8	0.8625	0.86875	0.865625	- 0.0034950
9	0.865625	0.86875	0.8671875	0.0008811
10	0.865625	0.8671875	0.8664063	- 0.0013098
11	0.8664063	0.8671875	0.8667969	- 0.0002150
12	0.8667969	0.8671875	0.8669922	0.0003329
13	0.8667969	0.8669922	0.8668945	0.0000589
14	0.8667969	0.8668945	0.8668457	- 0.0000781
15	0.8668457	0.8668945	0.8668701	- 0.0000096

SOAL 3.1.

1. *Ubah kode MATLAB di atas untuk mendapatkan hampiran dengan tingkat kesalahan tidak lebih dari 0.0000001.*
2. *Ubah kode MATLAB di atas untuk mendapatkan hampiran akar negatif dari persamaan* (3.2).

3.1.1 Analisis Kekonvergenan Metode Bagi Dua

Hampiran pertama terhadap akar persamaan yang diberikan oleh Metode Bagi Dua adalah $x_0 = (a_0 + b_0)/2$; dengan $a_0 = a$ dan $b_0 = b$. Galat mutlak di dalam hampiran pertama ini adalah $|r - x_0| \le (b_0 - a_0)/2$. Jika x_1, x_2, x_3, Jadi, jika batas galatnya sudah ditentukan sebelumnya kita dapat menentukan banyaknya iterasi yang diperlukan di dalam metode Bagi Dua, asalkan interval awal yang diberikan benar-benar memuat pembuat nol.

CONTOH 3.3.
Jika $a = 0.1$ dan $b = 1.0$, berapa banyak iterasi yang diperlukan dalam metode Bagi Dua untuk mencari akar suatu persamaan agar galatnya paling besar $10^{-8}/2$?
Penyelesaian:
Misalkan n banyak iterasi yang diperlukan. Maka $|r - x_n| \leq 10^{-8}/2$. Dari (3.7), $(b_0 - a_0)/2^{n+1} \leq 10^{-8}/2$, berarti $(1.0 - 0.1)/2^{n+1} \leq 10^{-8}/2$, atau $0.9/2^{n+1} \leq 10^{-8}/2$, atau $0.9 * 10^8 \leq 2^n$. Dengan mengambil logaritma kedua ruas kita peroleh $n \log(2) \geq \log(0.9) + 8$. Jadi, banyak iterasi $n \geq (\log(0.9) + 8)/\log(2) = 26.4$. Dengan demikian sedikitnya diperlukan 27 iterasi untuk mendapatkan akar persamaan tersebut dengan derajad keakuratan yang ditentukan. □

Metode **Bagi Dua** *hanya memperhatikan tanda nilai-nilai suatu fungsi, bukan nilai-nilai fungsi itu sendiri.*

Salah satu kekurangan metode Bagi Dua adalah metode ini mencari nilai-nilai hampiran akar x_n hanya dengan memperhatikan tanda $f(a)$ dan $f(b)$, bukan pada nilai-nilai fungsi. Padahal secara intuitif, misalkan kita tahu $f(a) = 500$ dan $f(b) = -0.1$, sudah tentu kita akan memilih hampiran x_n yang dekat ke b. Metode **posisi palsu**, yang akan dijelaskan pada bagian belakang dalam bab ini, memberikan cara untuk melakukan hal tersebut.

Implementasi Metode Bagi Dua dengan MATLAB

Implementasi metode **Bagi Dua** *pada MATLAB untuk sebarang fungsi*

Pada contoh sebelumnya sudah diberikan sebuah kode MATLAB untuk mendapatkan hampiran akar suatu persamaan dengan metode Bagi Dua. Akan tetapi, kode tersebut masih bersifat khusus, dengan fungsi dan interval yang sudah diberikan. Kita menginginkan sebuah kode umum yang dapat digunakan untuk menghitung hampiran akar suatu persamaan untuk sembarang fungsi dan interval. Di dalam MATLAB hal itu dapat dilakukan dengan membuat `M-file` (atau fungsi), misalkan kita namakan `bagidua`. Fungsi ini kita simpan ke dalam file bernama `bagidua.m`. Berikut adalah kodenya.

```
function [iterasi A B X FX] = bagidua(f,a,b,tol)
%-----------------------------------------------
```

```
% BAGIDUA Metode Bagi Dua
% untuk menghampiri akar persamaan f(x)=0
% Contoh pemakaian
%   [i,a,b,fx] = bagidua('f',a,b,tol)
%   [i,a,b,x,fx] = bagidua('f',a,b,tol)
% Input:
%   f       nama fungsi, yakni file ''f.m''
%           yang mendefinisikan f(x)
%   a       ujung kiri
%   b       ujung kanan
%   tol     toleransi kekonvergenan
% Hasil (output):
%   iterasi vektor penghitung iterasi
%   A    vektor ujung-ujung kiri
%   B    vektor ujung-ujung kanan
%   X    vektor titik-titik tengah (hampiran akar)
%   FX   nilai-nilai f(x) selama iterasi berlangsung
%----------------------------------------------------
if b < a, break;end
% salah memasukkan batas-batas interval!
A=[a];B=[b];X=[];FX=[];iterasi=[];
fa = feval(f,a);
fb = feval(f,b);
if fa*fb > 0, break, end            %tak ada akar!
N = 1 + round((log(b-a)-log(tol))/log(2));
%maksimum iterasi
for k=1:N,
  iterasi=[iterasi;k];
  x  = (a+b)/2;
  fx = feval(f,x);
  X=[X;x];FX=[FX;fx];
  if  (fx == 0)|  ((b-a) < tol), break, end
  if  fa*fx < 0,
    b = x;
  else
  a = x;
  end
  A=[A;a];B=[B;b];
```

5. Gunakan metode Bagi Dua untuk memperoleh hampiran akar-akar persamaan $f(x) = 0$ untuk fungsi-fungsi di atas dengan toleransi 10^{-6}. Jika Anda menggunakan MATLAB, berikan tampilan nilai-nilai penghitung iterasi, a, b, titik-titik tengah x, dan $f(x)$.

6. Gunakan metode Bagi Dua untuk mencari hampiran akar-akar yang dikehendaki pada persamaan-persamaan di bawah ini. Gunakan batas toleransi $\epsilon = 10^{-5}$.

   (a) Akar nyata $x^3 - x^2 - x - 1 = 0$.

   (b) Akar persamaan $x = 1 + 0.3\cos(x)$.

   (c) Akar positif terkecil persamaan $\cos(x) = \frac{1}{2} + \sin(x)$.

   (d) Akar persamaan $x = e^{-x}$.

   (e) Akar positif terkecil $e^{-x} = \sin(x)$.

7. Apa yang akan terjadi jika metode Bagi Dua digunakan untuk mencari hampiran akar persamaan $f(x) = 1/(x-2)$ pada interval $[3,\ 7]$ dan $[1,\ 7]$?

8. Misalkan interval-interval yang dihasilkan di dalam metode Bagi Dua adalah $[a_0,\ b_0]$, $[a_1,\ b_1]$, $[a_2,\ b_2]$, $[a_3,\ b_3]$,...

   (a) Tunjukkan bahwa $a_0 \le a_1 \le a_2 \le \ldots \le a_n \le a_{n+1}$ dan $b_0 \ge b_1 \ge b_2 \ge \ldots \ge b_n \ge b_{n+1}$.

   (b) Tunjukkan bahwa $b_n - a_n = (b_0 - a_0)/2^n$.

   (c) Jika titik-titik tengah interval adalah $x_n = (a_n + b_n)/2$, tunjukkan
   $$\lim_{n\to\infty} a_n = \lim_{n\to\infty} b_n = \lim_{n\to\infty} x_n.$$

9. Tunjukkan bahwa banyaknya iterasi, N, yang diperlukan oleh metode Bagi Dua dengan interval awal $[a,\ b]$ untuk mendapatkan hampiran akar dengan tolerasi hampiran sebesar ϵ harus memenuhi
$$N \ge \frac{\ln(b-a) - \ln(2\epsilon)}{\ln 2}.$$

10. Gambarlah grafik fungsi di bawah ini dan gunakan metode Bagi Dua untuk menghitung hampiran pembuat nol
$$f(x) = 32x^6 - 48x^4 + 18x^2 - 1.$$

Persamaan garis yang melalui titik-titik $(a_n,\ f(a_n))$ dan $(b_n,\ f(b_n))$ adalah

$$y = f(a_n) + \frac{f(b_n) - f(a_n)}{b_n - a_n}(x - a_n) = \frac{(f(b_n) - f(a_n))x + (b_n f(a_n) - a_n f(b_n))}{b_n - a_n}.$$

Garis tersebut akan memotong sumbu–x apabila nilai $y = 0$. Maka absis titik potongnya adalah

$$x_{n+1} = \frac{a_n f(b_n) - b_n f(a_n)}{f(b_n) - f(a_n)} = a_n - \frac{b_n - a_n}{f(b_n) - f(a_n)} f(a_n). \tag{3.8}$$

Setelah x dihitung, proses dilanjutkan dengan interval $[a_{n+1},\ b_{n+1}]$ dengan

$$\begin{aligned} a_{n+1} = a_n \text{ dan } b_{n+1} = x_{n+1} \text{ jika } f(a_n) \times f(x_{n+1}) < 0, \\ a_{n+1} = x_{n+1} \text{ dan } b_{n+1} = b_n \text{ jika } f(b_n) \times f(x_{n+1}) < 0. \end{aligned}$$

Metode ini secara visual ditunjukkan pada Gambar 3.3.

CONTOH 3.4.
Misalkan kita ingin menyelesaikan persamaan (3.2) *dengan metode Posisi Palsu, dengan interval awal* $[0,\ 1.2]$. *Tabel 3.2 menyajikan hasil-hasil perhitungan selama 15 iterasi.* □

Sekalipun metode Posisi Palsu sangat mirip dengan metode Bagi Dua, namun keduanya memiliki perbedaan yang sangat berarti. Khususnya, contoh di atas menunjukkan bahwa interval pengapit akar tidak harus menyempit ke nol. Untuk contoh di atas, batas kanan interval pengapit selalu tetap, yakni $b = 1.2$ dan batas kiri semakin menuju akar $r = 0.86687$ (sampai lima angka di belakang koma).

Suatu kriteria penghentian yang menjamin diperolehnya hampiran akar yang memiliki keakuratan sampai sejumlah angka desimal yang ditentukan sulit dirumuskan. Metode ini mungkin dapat dihentikan apabila nilai fungsi $f(x)$ atau selisih nilai-nilai fungsi pada dua iterasi berurutan cukup kecil. Meskipun demikian perlu ditegaskan bahwa tak satu dari kedua kriteria inipun yang menjamin keakuratan hampiran sampai sejumlah angka desimal yang ditentukan. Contoh di atas menunjukkan bahwa metode Posisi Palsu tidak lebih cepat konvergen ke akar dibandingkan dengan metode Bagi Dua. Akan tetapi hal ini tidaklah selalu benar, sangat tergantung pada pemilihan interval awal yang mengapit akar.

(b) $x^2 = \ln x = 0$

(c) $xe^x - 1 = 0$

5. Hitung hampiran akar persamaan

$$f(x) = xe^x - 1 = 0$$

dengan menggunakan metode Posisi Palsu dengan interval awal (i) $[0,\ 2]$, dan (ii) $[0,\ 3]$, dan berhenti setelah $|f(x)| < 5 \times 10^{-2}$.

6. Misalkan metode Posisi Palsu digunakan pada interval $[a,\ b]$ di mana $f(a) \times f(b) < 0$. Jika c adalah nilai yang diperoleh dengan metode tersebut, tunjukkan bahwa panjang subinterval yang memuat akar adalah

$$\frac{f(a)(a-b)}{f(b)-f(a)} \quad \text{jika} \quad f(a)f(c) < 0,$$

dan

$$\frac{f(b)(b-a)}{f(b)-f(a)} \quad \text{jika} \quad f(c)f(b) < 0.$$

7. Gunakan metode Posisi Palsu pada fungsi $f(x) = x^3 - 2x - 5$ dengan interval awal $[1,\ 3]$.

3.3 Metode Titik Tetap

Teknik mencari hampiran akar persamaan yang dijelaskan pada bagian sebelumnya tergantung pada kemampuan menemukan interval pengapit akar awal. Hal ini terkadang sulit dilakukan untuk beberapa persamaan tertentu dan oleh karenanya diperlukan suatu metode yang tidak memerlukan informasi awal. Salah satu metode demikian adalah metode Titik Tetap. Misalkan persamaan $f(x) = 0$ dapat dituliskan dalam bentuk

$$x = g(x). \tag{3.9}$$

Setiap penyelesaian persamaan (3.9) disebut **titik tetap** g. Dengan memilih nilai awal sebarang x_0, maka kita dapat melakukan iterasi perhitungan hampiran titik-titik tetap fungsi g, sebagai berikut

$$x_{n+1} = g(x_n), \qquad n = 0,\ 1,\ 2,\ 3,\ \ldots. \tag{3.10}$$

Tabel 3.3: Hasil ketiga rumus iterasi untuk menghampiri akar persamaan $x^2 - 2x - 8 = 0$

n	$x_{n+1} = \sqrt{2x_n + 8}$	$x_{n+1} = \frac{2x_n+8}{x_n}$	$x_{n+1} = \frac{{x_n}^2-8}{2}$
0	5.	5.	5.
1	4.2426407	3.6	8.5
2	4.0602071	4.2222222	32.125
3	4.0150236	3.8947368	512.00781
4	4.0037541	4.0540541	
5	4.0009384	3.9733333	
6	4.0002346	4.0134228	
7	4.0000586	3.993311	
8	4.0000147	4.0033501	
9	4.0000037	3.9983264	
10	4.0000009	4.0008372	
11	4.0000002	3.9995815	

CONTOH 3.5.
Perhatikan persamaan kuadrat

$$x^2 - 2x - 8 = 0$$

yang mempunyai akar-akar $x = -2$ *dan* $x = 4$*. Kita dapat menyatakan persamaan tersebut dalam tiga bentuk*

$$(a)\ x = \sqrt{2x+8} \qquad (b)\ x = \frac{2x+8}{x} \qquad (c)\ x = \frac{x^2-8}{2}.$$

Hasil-hasil iterasi dengan ketiga rumus di atas dan titik awal $x_0 = 5$ *disajikan pada Tabel 3.3.*

Iterasi mengasilkan barisan yang konvergen ke akar $r = 4$ *untuk rumus (a) dan (b) tetapi divergen untuk rumus (c).* □

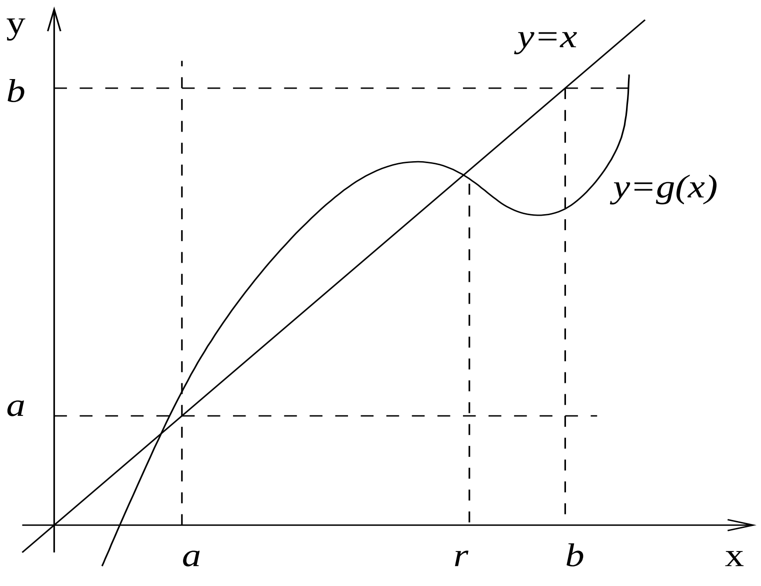
Contoh di atas memberikan pelajaran kepada kita bahwa metode Titik Tetap memerlukan analisis matematika untuk melakukan proses iterasi yang konvergen ke suatu akar. Syarat cukup kekonvergenan metode Titik Tetap dinyatakan dalam Teorema 3.2. Kita mulai dengan sebuah jutnya, $f(a) \leq 0 \leq f(b)$, karena $a \leq g(a)$ dan $b \geq g(b)$. Menurut Teorema Nilai Antara, terdapat titik $r \in [a,\ b]$ sedemikian hingga $f(r) = 0$. Jadi r merupakan titik tetap g.

2. Andaikan terdapat dua titik tetap, misalnya r_1 dan r_2 yang memenuhi $r_1 = g(r_1)$ dan $r_2 = g(r_2)$. Menurut Teorema Nilai Rata-rata terdapat titik c dengan $a < c < b$ sedemikian hingga,

$$g'(c) = \frac{g(r_2) - g(r_1)}{r_2 - r_1} = \frac{r_2 - r_1}{r_2 - r_1} = 1.$$

Hasil terakhir kontradiksi dengan hipotesis bahwa $|g'(x)| < 1$ untuk semua $x \in (a,\ b)$. Jadi, pengandaian harus diingkar, berarti hanya ada tepat sebuah titik tetap $r \in [a,\ b]$.

■



Gambar 3.5: Keberadaan akar persamaan $x = g(x)$

Gambar 3.5 menunjukkan keberadaan penyelesaian $x = g(x)$, yang secara grafis adalah titik potong antara garis $y = x$ dan $y = g(x)$.

$$r - x_{n+1} \approx g'(r)(r - x_n) \tag{3.13}$$

untuk n yang cukup besar. Fakta ini menunjukkan bahwa, jika $g'(r) \neq 0$, galat hampiran x_{n+1} pada akhirnya proporsional terhadap galat hampiran sebelumnya, x_n. Oleh karena faktor proporsinya $g'(r)$ maka dapat disimpulkan bahwa laju kekonvergenan metode Titik Tetap tergantung pada nilai $|g'(r)|$; semakin kecil gradien g di r, semakin cepat konvergen. Dalam hal ini, barisan hampiran titik tetap $\{x_n\}_{n=0}^{\infty}$ dikatakan konvergen secara linier, dan metode Titik Tetap dikatakan berorde satu. Secara umum, kita mempunyai definisi berikut ini.

Jika $0 < |g'(r)| < 1$, maka iterasi $x_{n+1} = g(x_n)$ konvergen ke titik tetap r secara linier.

DEFINISI 3.3 (DERAJAD KEKONVERGENAN).
Misalkan $x_0,\ x_1,\ x_2,\ \ldots$ suatu barisan yang konvergen ke r dan misalkan $e_n = r - x_n$. Apabila terdapat sebuah bilangan m dan sebuah konstanta $C \neq 0$, sedemikian hingga

$$\lim_{n\to\infty} \frac{|e_{n+1}|}{|e_n|^m} = C$$

maka m disebut **derajad kekonvergenan** *barisan tersebut dan C disebut* **konstanta galat asimptotik**.

Pengertian derajad kekonvergenan suatu iterasi untuk menghampiri akar persamaan

Untuk $m = 1,\ 2,\ 3$, kekonvergenannya berturut-turut disebut *linier*, *kuadratik*, dan *kubik*.

Dari hubungan (3.13) dapat diketahui, sebagaimana telah disebutkan di atas, bahwa apabila $g'(r) \neq 0$ maka iterasi titik tetap konvergen secara linier. Bagaimanakah jika $g'(r) = 0$? Untuk menjawab pertanyaan ini, kita gunakan ekspansi Taylor $g(x_n)$ di sekitar r:

$$g(x_n) = g(r) + (x_n - r)g'(r) + \frac{1}{2}(x_n - r)^2 g''(c_n),$$

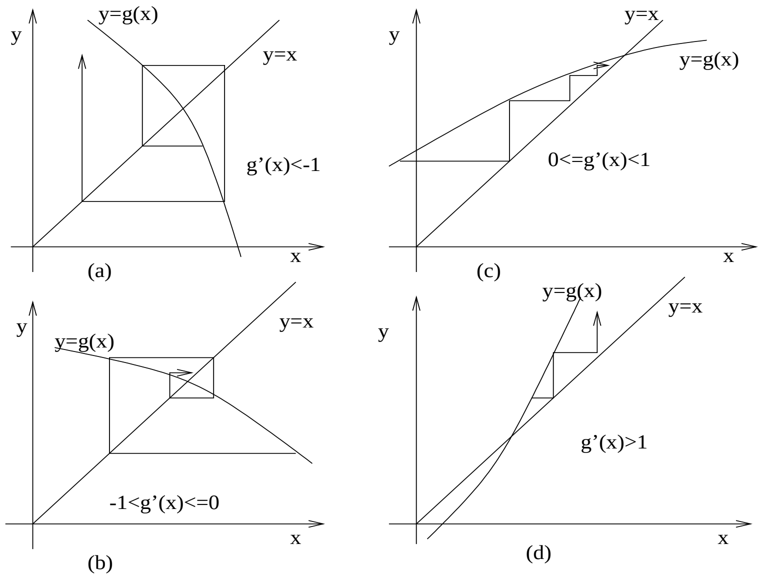
dengan c_n antara r dan x_n. Dengan mengingat $x_{n+1} = g(x_n)$, $r = g(r)$, dan $g'(r) = 0$, maka diperoleh

$$x_{n+1} = r + \frac{1}{2}(x_n - r)^2 g''(c_n),$$

atau

$$r - x_{n+1} = -\frac{1}{2}(x_n - r)^2 g''(c_n),$$

Contoh 3.5 konvergen sedangkan pada kasus (c) divergen.



Gambar 3.6: Berbagai kemungkinan kekonvergenan metode Titik Tetap

(a) Jika $g(x) = \sqrt{2x+8}$, maka $g'(x) = (2x+8)^{-1/2}$. Oleh karena $g'(x) < 1$ pada interval $[3,\ 5]$ yang memuat hampiran awal $x_0 = 5$ dan titik tetap sebagai titik tengahnya, $r = 4$, maka Teorema 3.2 menjamin bahwa iterasi titik tetap konvergen, sebagaimana sudah terlihat kenyataannya.

(b) Pada rumus $g(x) = \frac{2x+8}{x}$, berlaku $-1 < g'(x) = -\frac{8}{x^2} < 0$ untuk $x \in [3,\ 5]$, sehingga lagi, Teorema 3.2 menjamin kekonvergenan proses iterasi.

(c) Pada rumus $g(x) = \frac{x^2-8}{2}$, $g'(x) = x$ dan $g'(4) = 4 > 1$, sehingga Teorema 3.2 tidak menjamin kekonvergenan metode Titik Tetap. Selanjutnya, dari persamaan (3.11) kita peroleh

$$r - x_{n+1} = g'(\xi_n)(r - x_n) \geq 4(r - x_n) \geq ... \geq 4^{n+1}(r - x_0),$$

karena ξ_n terletak antara r dan x_n, untuk $n = 0,\ 1,\ 2,\ \ldots$. Ini berarti, Tentukan persamaan-persamaan yang memiliki akar yang dapat dihampiri oleh rumus-rumus iterasi di atas.

3. Gunakan metode Titik Tetap untuk menghitung hampiran akar persamaan

   (a) $x - \cos x = 0$
   (b) $x^2 + \ln x = 0$

   dengan hampiran awal $x_0 = 1$. Hentikan iterasi apabila $|x_{n+1} - x_n| < 5 \times 10^{-6}$.

4. Hitunglah sembilan suku pertama barisan yang dihasilkan oleh

$$x_{n+1} = e^{-x_n}$$

   dengan titik awal $x_0 = 1$.

5. Dengan menganggap barisan-barisan yang dihasilkan oleh

   (a) $x_{n+1} = \frac{x_n 62+2}{3}$
   (b) $x_{n+1} = 3 - \frac{2}{x_n}$

   keduanya konvergen, tunjukkan keduanya juga konvergen ke akar-akar yang berbeda dari persamaan yang sama.

6. Buatlah sketsa kurva-kurva yang bersesuaian dengan persamaan $5x - e^x = 0$ untuk menunjukkan bahwa persamaan tersebut memiliki dua buah akar. Dengan menggunakan metode Titik Tetap hitunglah hampiran akar-akar persamaan tersebut dengan hampiran awal $x_0 = 2$. Hentikan iterasi setelah $|x_{n+1} - x_n| < 5 \times 10^{-6}$.

7. Misalkan iterasi $x_{n+1} = g(x_n)$ menghasilkan barisan bilangan yang konvergen ke titik tetap r. Jika g''' ada, gunakan teorema Taylor untuk menunjukkan bahwa

$$g(x_n) = g(r) + (x_n - r)g'(r) + \frac{(x_n - r)^2}{2} g''(r) + \frac{1}{6} g'''(\xi_n)$$

   untuk suatu bilangan ξ_n antara x_n dan r. Turunkan bahwa iterasi titik tetap konvergen secara kubik jika $g'(r) = g''(r) = 0$ dan $g'''(r) \neq 0$.

berapakah (jika ada) kekonvergenan iterasi tersebut kuadratik?

13. Perhatikan persamaan $x = g_c(x) \equiv cx(1-x)$, dengan c adalah konstanta taknol. Persamaan ini memiliki dua penyelesaian, dan misalkan r_c adalah penyelesaian taknolnya. Berapakah r_c? Untuk nilai c yang manakah iterasi $x_{n+1} = cx_n(1-x_n)$ konvergen ke r_c (asalkan x_0 dipilih cukup dekat dengan r_c)? Persamaan $x = g_c(x)$ disebut *persamaan logistik*, dan bersama iterasi $x_{n+1} = g_c(x_n)$ merupakan topik yang menjadi perhatian dalam **teori kekacauan** (*mathematical theory of chaos*). Selidiki perilaku iterasi tersebut dengan menambah nilai-nilai c secara pelan pada interval yang ditemukan sebelumnya, dengan menjaga nilai $c < 4$. Perhatikan hasil iterasi untuk n yang cukup besar.

14. Gunakan rumus ekstrapolasi Aitken untuk menaksir galat $r - x_2$ pada iterasi-iterasi di bawah ini.

    (a) $x_{n+1} = e^{-x_n}, \quad x_0 = 0.57$
    (b) $x_{n+1} = \frac{0.5}{1+x_n^4}, \quad x_0 = 0.48$
    (c) $x_{n+1} = 1 + 0.5\sin(x_n), \quad x_0 = 1.5$

15. Rumus ekstrapolasi Aitken dapat digunakan untuk mempercepat kekonvergenan iterasi yang lambat. Salah satu contoh algoritma gabungan Titik Tetap dan Ekstrapolasi Aitken adalah sebagai berikut:

$$\begin{aligned} x_n &= g(x_{n-1}), \quad \text{jika} \quad 3 \nmid n \\ x_n &= \text{ekstrapolasi Aitken} \quad x_{n-1}, x_{n-2}, x_{n-3}, \ \text{jika} \quad 3|n \end{aligned}$$

    Gunakan algoritma tersebut pada iterasi-iterasi di bawah ini.

    (a) $x_{n+1} = 2e^{-x_n}, \quad x_0 = 0.8$
    (b) $x_{n+1} = \frac{0.9}{1+x_n^4}, \quad x_0 = 0.75$
    (c) $x_{n+1} = 6.28 + \sin(-x_n), \quad x_0 = 6$

16. Tunjukkan bahwa rumus ekstrapolasi (3.17) dapat ditulis ulang sebagai

$$r \approx x_n - \frac{(x_n - x_{n-1})^2}{(x_n - x_{n-1}) - (x_{n-1} - x_{n-2})}.$$

linier. Misalkan $f(x)$ suatu fungsi diferensiabel pada $[a,\ b]$. Maka $f(x)$ mempunyai kemiringan tertentu dan garis singgung tunggal pada setiap titik di dalam $(a,\ b)$. Garis singgung di titik $(x_0,\ f(x_0))$ merupakan pendekatan grafik $f(x)$ di dekat titik $(x_0,\ f(x_0))$. Jadi pembuat nilai nol garis singung tersebut merupakan hampiran pembuat nol $f(x)$.

Kita dapat menyusun prosedur sebagai berikut untuk mencari akar persamaan $f(x) = 0$. Mula-mula kita pilih absis titik awal, x_0. Hampiran pertama akar, x_1, merupakan absis titik potong garis singgung di titik $(x_0,\ f(x_0))$ dengan sumbu–x. Hampiran kedua, x_2, merupakan absis titik potong garis singgung di titik $(x_1,\ f(x_1))$ dengan sumbu–x. Demikian seterusnya, sampai diperoleh hampiran terbaik untuk akar persamaan $f(x) = 0$. Gambar 3.7 melukiskan proses tersebut.

3.4.1 Penurunan Rumus Newton–Raphson

Misalkan x_0 adalah absis titik awal yang diberikan. Gradien garis singgung kurva $y = f(x)$ di titik $(x_0,\ f(x_0))$ adalah $f'(x_0)$. Persamaan garis singungnya adalah

$$y - f(x_0) = f'(x_0)(x - x_0) \tag{3.19}$$

Hampiran pertama akar, x_1, diperoleh dari (3.19) jika $y = 0$. Artinya, titik $(x_1,\ 0)$ memenuhi persamaan (3.19), yakni

$$\begin{aligned} 0 - f(x_0) &= f'(x_0)(x_1 - x_0) \\ x_1 - x_0 &= -\frac{f(x_0)}{f'(x_0)} \\ x_1 &= x_0 - \frac{f(x_0)}{f'(x_0)} \end{aligned}$$

Dengan cara yang sama, akhirnya diperoleh secara umum,

Rumus iterasi Newton – Raphson untuk menyelesaikan $f(x) = 0$.

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}, \quad n = 0,\ 1,\ 2,\ ... \tag{3.20}$$

Catatan:
Fungsi $g(x)$ yang didefinisikan sebagai

$$g(x) = x - \frac{f(x)}{f'(x)} \tag{3.21}$$

akar tersebut.

```
>>x0=-5;y0=5;
>>h=[0 x0 y0];
>>for k=1:15,
>>x=(exp(x0)*(x0-1)+2)/(exp(x0)-1);
>>y=(exp(y0)*(y0-1)+2)/(exp(y0)-1);
>>h=[h;k x y];
>>if (abs(x-x0)<0.000001)&(abs(y-y0)<0.000001),
>>break;end
>>x0=x;y0=y;
>>end
>>h
 h  =
   0.  - 5.           5.
   1.  - 1.9728654    4.0407019
   2.  - 1.8428645    3.13093
   3.  - 1.8414059    2.3195977
   4.  - 1.8414057    1.6815416
   5.  - 1.8414057    1.2946288
   6.  - 1.8414057    1.1606438
   7.  - 1.8414057    1.1463445
   8.  - 1.8414057    1.1461932
   9.  - 1.8414057    1.1461932
```

Terlihat bahwa metode Newton–Raphson konvergen ke akar $r_1 = -1.8414057$ *setelah iterasi ke-4 dan konvergen ke akar* $r_2 = 1.1461932$ *setelah iterasi ke-8.* □

CONTOH 3.7.

Sekarang akan ditunjukkan suatu prosedur yang digunakan oleh komputer untuk menghitung operasi pembagian a/b. *Hardware komputer biasanya hanya memiliki aritmetika untuk penjumlahan, pengurangan, dan perkalian, sedangkan operasi pembagian diimplementasikan dengan menggunakan software (program). Pembagian tersebut dapat dilakukan dengan cara mengalikan* a *dan* $1/b$, *sedangkan nilai* $1/b$ *dihampiri dengan metode Newton – Raphson.*

Nilai $1/b$ *merupakan penyelesaian persamaan*

$$f(x) = b - \frac{1}{x} = 0,$$

dengan $b > 0$. *Turunan* f *adalah*

$$f'(x) = \frac{1}{x^2},$$

sehingga rumus iterasi Newton – Raphson untuk hampiran akar $f(x) = 0$ diberikan oleh rumus

Rumus iterasi Newton – Raphson untuk menghitung hampiran $1/b$

$$x_{n+1} = x_n - \frac{b - 1/x_n}{1/x_n^2} = x_n(2 - b * x_n), \quad n \geq 0. \tag{3.22}$$

Dalam rumus iterasi di atas tidak terdapat operasi pembagian, hanya memuat perkalian dan pengurangan. Hampiran awal tentunya dipilih $x_0 > 0$.

Misalnya galat relatif x_n sebagai hampiran akar $r = 1/b$ dituliskan sebagai

$$Rel(x_n) = \frac{r - x_n}{r}.$$

Dengan menggunakan rumus (3.22) *selanjutnya, dengan mengingat $b = 1/r$, diperoleh*

$$\begin{aligned} Rel(x_{n+1}) &= \frac{r - x_{n+1}}{r} = \frac{r - x_n(2 - x_n/r)}{r} \\ &= \frac{(r - x_n)^2}{r^2} \\ &= [Rel(x_n)]^2, \quad n \geq 0. \end{aligned} \tag{3.23}$$

Dari relasi (3.23) *diperoleh*

$$Rel(x_n) = [Rel(x_0)]^{2n},$$

sehingga agar galat hampiran x_n semakin mengecil ke nol, syaratnya haruslah

$$|Rel(x_0)| < 1,$$

atau

$$-1 < \frac{1/b - x_0}{1/b} < 1,$$

sehingga diperoleh

$$0 < x_0 < \frac{2}{b}. \tag{3.24}$$

Jadi, iterasi (3.22) *konvergen jika dan hanya jika hampiran awal x_0 memenuhi* (3.24). *Dari hasil* (3.23), *terlihat bahwa kekonvergenan iterasi* (3.22) *sangat-*

$\sqrt{A}$, *yakni* $\lim_{n\to\infty} x_n = \sqrt{A}$.

GARIS BESAR BUKTI: Pilih fungsi $f(x) = x^2 - A$, dan perhatikan bahwa $x^2 - A = 0$ memiliki akar-akar $\pm\sqrt{A}$. Di sini kita mempunyai $f'(x) = 2x$, sehingga rumus iterasi Newton–Raphson untuk f adalah

$$x_{n+1} = x_n - \frac{{x_n}^2 - A}{2x_n} = \frac{x_n + A/x_n}{2}.$$

Oleh karena $f''(x) = 2$ (kontinyu di mana-mana) dan $f'(\sqrt{A}) = 2\sqrt{A} > 0$, maka persyaratan Teorema 3.3 dipenuhi, sehingga $\{x_n\}_{n=0}^{\infty}$ konvergen ke $\sqrt{A}$. ■

Keuntungan Akibat 3.2 adalah kita hanya perlu menggunakan operasi penjumlahan, pengurangan, perkalian, dan pembagian untuk menghitung akar suatu bilangan. Apabila kita gunakan fungsi lain di mana rumus iterasi Newton–Raphsonnya memuat perhitungan akar, maka kita terjebak pada sebuah lingkaran, "menghitung akar bilangan menggunakan akar bilangan lain!"

CONTOH 3.8.
Hitunglah hampiran $\sqrt{5}$ sampai enam angka di belakang koma, dengan menggunakan metode Newton–Raphson.
Penyelesaian:
Dengan menggunakan hampiran awal $x_0 = 2$ dan menggunakan rumus (3.26) *kita peroleh tabel hampiran sebagai berikut:*

Iterasi	*Hampiran* $\sqrt{5}$
0.	2.
1.	2.25
2.	2.2361111
3.	2.236068
4.	2.236068

Perhitungan di atas dilakukan dengan menggunakan kode MATLAB sebagai berikut:

```
>>x0=2;hasil=[0 x0];
>>for k=1:15,
```

```
>>x=(x0+5/x0)/2;
>>hasil=[hasil;k x];
>>if abs(x-x0)<0.0000001, break;end
>>x0=x;
>>end
```

Meskipun iterasi direncanakan sampai iterasi ke-15, namun sampai iterasi ke-4, hampiran sudah konvergen sampai 6 angka di belakang koma, yakni $\sqrt{5} \approx 2.236068$ *dengan kesalahan kurang dari* 5×10^{-6}. □

3.4.2 Analisis Kekonvergenan

Misalkan $x_0,\ x_1,\ x_2,\ \ldots,\ x_n,\ x_{n+1}$ merupakan hampiran-hampiran akar yang diperoleh melalui iterasi berturut-turut dalam metode Newton–Raphson. Misalkan r adalah akar yang sesungguhnya. Misalkan juga e_n menyatakan galat pada iterasi ke-n. Maka $e_n = x_n - r$, dan

$$\begin{aligned} e_{n+1} &= x_{n+1} - r = x_n - \frac{f(x_n)}{f'(x_n)} - r \\ &= x_n - r - \frac{f(x_n)}{f'(x_n)} = e_n - \frac{f(x_n)}{f'(x_n)} \\ &= \frac{(e_n f'(x_n) - f(x_n))}{f'(x_n)}. \end{aligned} \tag{3.27}$$

Sekarang, dengan mengekspansi $f(x_n - e_n)$ dalam bentuk deret Taylor kita dapatkan

$$\begin{aligned} f(x_n - e_n) &= f(x_n) - f'(x_n)e_n + \frac{f''(c_n)}{2}e_n^2 \\ f(r) &= f(x_n) - f'(x_n)e_n + \frac{f''(c_n)}{2}e_n^2 \\ 0 &= f(x_n) - f'(x_n)e_n + \frac{f''(c_n)}{2}e_n^2 \\ e_n f'(x_n) - f(x_n) &= \frac{f''(c_n)}{2}e_n^2, \end{aligned} \tag{3.28}$$

dengan c_n adalah suatu bilangan antara r dan x_n. Dari (3.27) dan (3.28) kita peroleh

$$e_{n+1} = \frac{f''(c_n)e_n^2}{2f'(x_n)}$$

linier ke akar ganda r yang berderajad $m > 1$.

Secara formal hasil-hasil di atas dinyatakan dalam teorema sebagai berikut.

TEOREMA 3.4 (LAJU KEKONVERGENAN ITERASI NEWTON – RAPHSON). *Misalkan barisan $\{x_n\}_{n=0}^{\infty}$ yang dihasilkan oleh iterasi Newton – Raphson $x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$ konvergen ke akar r, di mana $f(r) = 0$. Misalkan $e_n = x_n - r$ adalah galat hampiran ke-n.*

1. *Jika r adalah akar sederhana, maka kekonvergenan tersebut kuadratik, dan*

$$\lim_{n\to\infty} \frac{|e_{n+1}|}{e_n^2} = \left| \frac{f''(r)}{2f'(r)} \right|.$$

2. *Jika r adalah akar ganda berderajad $m > 1$, maka kekonvergenannya linier, dan*

$$\lim_{n\to\infty} \left| \frac{e_{n+1}}{e_n} \right| = \frac{m-1}{m}.$$

Kekonvergenan kuadratik (ke akar sederhana) metode Newton–Raphson merupakan alasan utama kepopulerannya dalam praktek. Akan tetapi, metode ini memiliki kekurangan, karena ia memerlukan perhitungan turunan f'. Hal ini dapat merupakan kendala apabila fungsi f sangat rumit dan tidak mungkin menyatakan f' secara eksplisit. Salah satu alternatif untuk mengatasi kelemahan ini adalah dengan menggunakan metode Tali Busur, yang akan dijelaskan pada bagian berikutnya. Pada metode ini, perhitungan $f'(x_n)$ dilakukan dengan menggunakan hampiran gradien tali busur yang menghubungkan titik-titik $(x_{n-1},\ f(x_{n-1}))$ dan $(x_n,\ f(x_n))$.

Berikut adalah beberapa contoh kekonvergenan iterasi Newton – Raphson untuk mencari akar-akar persamaan $(x+2)(x-1)(x-1) = x^3 - 3x + 2 = 0$.

CONTOH 3.9 (KONVERGEN KUADRATIK KE AKAR SEDERHANA). *Gunakan iterasi Newton – Raphson dengan hampiran awal $x_0 = -2.4$ untuk menghampiri akar sederhana $r = -2$ dari persamaan $x^3 - 3x + 2 = 0$.*

Penyelesaian:

```
>>x=(2/3)*(x0^3-1)/(x0^2-1);
>>fx=x^3-3*x+2;
>>en=r-x;c=abs(en)/abs(e0);
>>hasil=[hasil;n x fx en c];
>>x0=x;e0=en;
>>if abs(e0)<0.0000000001, break; end;
>>end

>>hasil
 hasil  =

  1   1.1030303   0.0329394  -0.1030303   0.5151515
  2   1.0523564   0.0083671  -0.0523564   0.5081652
  3   1.0264008   0.0021094  -0.0264008   0.5042517
  4   1.0132577   0.0005296  -0.0132577   0.5021714
  5   1.0066434   0.0001327  -0.0066434   0.5010975
  6   1.0033254   0.0000332  -0.0033254   0.5005518
  7   1.0016636   0.0000083  -0.0016636   0.5002767
  8   1.000832    0.0000021  -0.0008320   0.5001385
  9   1.0004161   5.194E-07  -0.0004161   0.5000693
 10   1.0002081   1.299E-07  -0.0002081   0.5000347
```

Terlihat bahwa iterasinya, sekalipun konvergen ke akar dobel $r = 1$, namun cukup lambat. Nilai-nilai $f(x_n)$ menuju ke nol lebih cepat daripada galat e_n. Dari kolom terakhir terlihat bahwa proporsi galat dua hampiran berturutan konvergen ke 0.5. Hal ini sesuai dengan Teorema 3.4, bahwa nilai-nilai tersebut konvergen ke $(m-1)/m = (2-1)/2 = 1/2$. □

Sebuah pertanyaan yang muncul adalah, dapatkah kekonvergenan linier ke akar ganda dipercepat pada iterasi Newton – Raphson? Teorema berikut memberikan jawaban atas pertanyaan tersebut. (lihat [6]: hal. 83).

Rumus iterasi Newton – Raphson Dipercepat untuk menghampiri akar ganda berderajad $m > 1$

TEOREMA 3.5 (ITERASI NEWTON – RAPHSON DIPERCEPAT). *Misalkan iterasi Newton – Raphson menghasilkan barisan hampiran yang konvergen secara linier ke akar ganda r yang berderajad $m > 1$. Modifikasi rumus Newton – Raphson*

$$x_{n+1} = x_n - \frac{mf(x_n)}{f'(x_n)}, \tag{3.31}$$

dapat didefinisikan fungsi

$$F(x) = f^{(m-1)}(x).$$

Dari hubungan (3.4) dapat dilihat bahwa r merupakan akar sederhana. Jadi metode Newton – Raphson dapat dipakai pada $F(x)$ untuk menghampiri akar ganda dari $f(x) = 0$. Jika iterasinya konvergen, maka akan konvergen kuadratik.

Implementasi Metode Newton – Raphson dengan MATLAB

Seperti biasanya berikut disajikan fungsi MATLAB yang disimpan dalam file `newton.m` untuk mengimplementasikan metode Newton–Raphson, guna mempermudah perhitungan dengan berbagai fungsi.

```
function [iterasi,x,fx,galatx,galaty] = %
newton(f,df,x0,delta,epsilon,N)
%---------------------------------------------------------
% Iterasi  Newton-Raphson untuk menghampiri akar persamaan
%                         f(x)=0
% dengan rumus iterasi x_{n+1} = x_n-f(x_n)/f'(x_n),
%                                         n= 0, 1, 2, ...
% Contoh pemakaian:
%   [iterasi,x] = newton('f','df',x0,delta,epsilon,N)
%   [iterasi,x,fx,galatx,galaty] =
%                      newton('f','df',x0,delta,epsilon,N)
% Input:
%   f         nama fungsi yang mendefinisikan f(x)
%   df        nama fungsi turunan f(x)
%   x0        hampiran awal
%   delta     batas toleransi kekonvergenan hampiran r
%   epsilon   batas toleransi kekonvergenan
%             nilai fungsi f(r)
%   N         maksimum iterasi
% Output:
%   iterasi   vektor penghitung iterasi
%   x         vektor hampiran-hampiran selama iteras
%   fx        vektor nilai-nilai hampiran f(x)
%   galatx    vektor selisih dua hampiran berturut-turut
%   galaty    vektor selisih dua hampiran
```

```
>>[k,x,fx,ex,ey]=newton('f','df',5,0.000001,0.000001,15)
```

Silakan dicoba! Gunakan fungsi `newton` untuk memperoleh hampiran yang lain dari contoh kita di atas.

Pemakaian fungsi `newton` *untuk sebarang fungsi f*

LATIHAN 3.4

1. Buatlah fungsi MATLAB `akar.m` dengan menggunakan algoritma (3.26) untuk menghitung $\sqrt{A}$. Gunakan fungsi `akar` tersebut untuk menghitung nilai-nilai di bawah ini dengan menggunakan hampiran awal yang diberikan.

   (a) $\sqrt{8}$ dengan $x_0 = 3$

   (b) $\sqrt{50}$ dengan $x_0 = 7$

   (c) $\sqrt{91}$ dengan $x_0 = 10$

   (d) $-\sqrt{8}$ dengan $x_0 = -3$

2. **Algoritma akar kubik**. Tunjukkan bahwa iterasi Newton – Raphson untuk menghitung hampiran $\sqrt[3]{A}$ adalah

$$x_{n+1} = \frac{2x_n + A/x_n^2}{3}, \quad n = 1,\ 2,\ \ldots$$

   Implementasikan algoritma tersebut ke dalam fungsi MATLAB `akarp3.m`. Gunakan fungsi `akarp3` untuk menghitung hampiran nilai-nilai di bawah ini dengan menggunakan hampiran awal yang diberikan.

   (a) $\sqrt[3]{7}$ dengan $x_0 = 2$

   (b) $\sqrt[3]{30}$ dengan $x_0 = 3$

   (c) $\sqrt[3]{200}$ dengan $x_0 = 6$

   (d) $\sqrt[3]{-7}$ dengan $x_0 = -2$

3. Sketsa grafik fungsi $f(x) = x^2 - 2$. Tentukan interval yang memuat hampiran awal x_0 agar metode Newton–Raphson (a) konvergen ke akar positif, (b) konvergen ke akar negatif, dan (c) tidak konvergen. Hitunglah hampiran $\sqrt{2}$ sampai enam angka di belakang koma, dengan menggunakan hampiran awal $x_0 = 1$.

4. Gunakan metode Newton–Raphson untuk menghitung hampiran akar-akar persamaan

   (a) $x - \cos x = 0$
   (b) $x^2 + \ln x = 0$
   (c) $x^3 + 4x^2 + 4x + 3 = 0$

   dengan menggunakan hampiran awal $x_0 = 1$ untuk masing-masing persamaan dan hentikan iterasi jika $|x_{n+1} - x_n| < 10^{-6}$.

5. Tunjukkan bahwa iterasi Newton–Raphson untuk menghitung hampiran akar persamaan $x^k e^x = 0$ adalah

$$x_{n+1} = \frac{(k-1)x_n + x_n^2}{k + x_n}.$$

   Dengan menggunakan hampiran awal $x_0 = 1$, hitung x_5 untuk kasus-kasus (a) $k = 1$, (b) $k = 2$. Jelaskan mengapa salah satu iterasi konvergen lebih cepat daripada yang lain!

6. Salah satu cara menghindari perhitungan $f'(x_n)$ pada setiap iterasi Newton–Raphson adalah dengan menggunakan $f'(x_0)$ sebagai hampiran, sehingga rumus iterasinya menjadi

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_0)}.$$

   Gambarlkan proses iterasi ini secara grafis dan gunakan metode ini untuk menghitung hampiran akar $x - \cos x = 0$ dengan menggunakan hampiran awal $x_0 = 1$, dan hentikan iterasi setelah $|x_{n+1} - x_n| < 10^{-6}$. Bandingkan hasilnya dengan soal 4a.

7. Gunakan metode Newton–Raphson untuk menghitung hampiran akar $f(x) = -x^3 + 3x^2 - 2x$, yang memiliki keakuratan sampai 6 angka di belakang koma, dengan menggunakan hampiran awal (a) $x_0 = 1.4$, (b) $x_0 = 1 + 1/\sqrt{5}$, (c) $x_0 = 1.5$, (d) $x_0 = 1.6$. Jelaskan hasil yang Anda peroleh dengan menggunakan sifat-sifat grafik fungsi f.

8. Turunkan rumus iterasi Newton–Raphson untuk menghitung hampiran $\sqrt[n]{a}$ dengan $a > 0$. Gunakan rumus tersebut untuk nilai $a = 2$ dan $n = 3, 4, 5, 6, 8$ untuk mendapatkan hampiran akurat sampai 8 angka di belakang koma. *Petunjuk:* Selesaikan $f(x) = x^n - a = 0$.

bawah ini dan hitunglah hampiran akarnya dengan menggunakan hampiran awal yang diberikan.

(a) $f(x) = x^2 - 5$, dengan $x_0 = 2$

(b) $f(x) = x^3 - 3x + 2$, dengan $x_0 = -2.4$

3.5 Metode Tali Busur (*Secant*)

Metode Secant (baca: "sékan") merupakan modifikasi metode Newton–Raphson. Pada metode Newton–Raphson kita menggunakan garis singgung pada titik $(x_0, f(x_0))$ sebagai hampiran $f(x)$ di sekitar x_0 dan mencari titik potongnya dengan sumbu–x sebagai hampiran akar. Dengan kata lain, metode Newton–Raphson memerlukan perhitungan nilai dua buah fungsi, yakni $f(x_n)$ dan $f'(x_n)$, pada setiap iterasi. Apabila kedua fungsi tersebut tidak rumit, metode tersebut mungkin sangat baik mengingat tingkat kekonvergenannya. Akan tetapi, sebagaimana sudah disinggung di depan, dalam beberapa kasus mungkin tidak mudah menurunkan $f'(x)$ dari $f(x)$. Oleh karena itu diperlukan suatu metode pengganti yang memiliki tingkat kekonvergenan mendekati tingkat kekonvergenan metode Newton–Raphson. Metode alternatif ini adalah metode Tali Busur (*Secant*).

Pada metode Tali Busur kita gunakan tali busur yang melalui $(x_0, f(x_0))$ dan $(x_1, f(x_1))$ sebagai hampiran $f(x)$ dan mencari titik potongnya dengan sumbu–x sebagai hampiran akar. Misalkan x_0 dan x_1 adalah dua hampiran awal yang diberikan. Gradien garis busur yang melalui titik $(x_0, f(x_0))$ dan $(x_1, f(x_1))$ adalah $\frac{f(x_1)-f(x_0)}{x_1-x_0}$. Persamaan tali busurnya adalah

$$y - f(x_1) = \frac{f(x_1) - f(x_0)}{x_1 - x_0}(x - x1_). \tag{3.32}$$

Hampiran pertama, x_2, diperoleh dengan mencari titik potong kurva (3.32) dengan sumbu–x. Artinya, titik $(x_2, 0)$ memenuhi persamaan (3.32):

$$0 - f(x_1) = \frac{f(x_1) - f(x_0)}{x_1 - x_0}(x_2 - x1_)$$

hitungan fungsi pada setiap iterasi. Metode Tali Busur hanya memerlukan perhitungan nilai-nilai sebuah fungsi f (di dua titik) pada setiap iterasinya.

Metode Posisi Palsu yang sudah dijelaskan di depan merupakan kasus khusus metode Tali Busur, dengan $a = \min(x_0,\ x_1)$ dan $b = \max(x_0,\ x_1)$. Bandingkan rumus iterasi Posisi Palsu (3.8) dan rumus iterasi Tali Busur (3.33).

ALGORITMA 3.4 (METODE TALI BUSUR).
INPUT: *fungsi $f(x)$, tebakan awal x_0 dan x_1, batas toleransi T, dan maksimum iterasi N*
OUTPUT: *p sedemikian $f(p) = 0$ atau pesan "gagal/error"*
LANGKAH-LANGKAH:

1. *Set $i = 2$, $q_0 = f(x_0)$, $q_1 = f(x_1)$.*
2. *WHILE $i \leq N$ DO*
   (a) *Hitung $x = x_1 - \frac{q_1(x_1-x_0)}{q_1-q_0}$*
   (b) *IF $|x - x_1| < T$ THEN set $p = x$; goto STOP.*
   (c) *Set $i = i + 1$*
   (d) *Set $x_0 = x_1$ dan $x_1 = x$, $q_0 = q_1$ dan $q_1 = f(x)$*
3. *Tulis pesan "Metode gagal setelah N iterasi"*
4. *STOP*

CONTOH 3.12.
Sebagai ilustrasi, kita gunakan metode Tali Busur untuk memperoleh hampiran akar-akar persamaan pada Contoh 3.6 dengan menggunakan hampiran-hampiran awal $x_0 = -10$, $x_1 = -9$ dan $x_0 = 10$, $x_1 = 9$. Rumus iterasinya dalam hal ini adalah

$$x_{n+1} = x_n - \frac{(x_n - x_{n-1})(e^{x_n} - x_n - 2)}{e^{x_n} - x_n - e^{x_{n-1}} + x_{n-1}}.$$

Dengan menggunakan kode MATLAB:

```
>>x0=-10;x1=-9;y0=10;y1=9;
>>hasil=[0 x0 y0;1 x1 y1];
>>for k=1:20,
>>x=x1-(exp(x1)-x1-2)*(x1-x0)/(exp(x1)-x1-exp(x0)+x0);
>>y=y1-(exp(y1)-y1-2)*(y1-y0)/(exp(y1)-y1-exp(y0)+y0);
```

```
>>hasil=[hasil;k x y];
>>if (abs(x-x1)<0.000001)&(abs(y-y1)<0.000001),
>>break;end
>>x0=x1;x1=x;y0=y1;y1=y;
>>end
```

kita peroleh hasil perhitungan sebagaimana disajikan pada Tabel 3.4.

Tabel 3.4: Iterasi untuk hampiran akar persamaan $e^x - x - 2 = 0$ dengan metode Tali Busur

Hampiran ke	*Hampiran r_1*	*Hampiran r_2*
0.	*- 10.*	*10.*
1.	*- 9.*	*9.*
1.	*- 1.9993305*	*8.4187716*
2.	*- 1.8619183*	*7.6829665*
3.	*- 1.841689*	*7.0089621*
4.	*- 1.8414062*	*6.3136547*
5.	*- 1.8414057*	*5.6309112*
6.	*- 1.8414057*	*4.9511115*
7.	*- 1.8414057*	*4.2838424*
8.	*- 1.8414057*	*3.6352896*
9.	*- 1.8414057*	*3.0188162*
10.	*- 1.8414057*	*2.4529329*
11.	*- 1.8414057*	*1.9628761*
12.	*- 1.8414057*	*1.5773211*
13.	*- 1.8414057*	*1.3196437*
14.	*- 1.8414057*	*1.1903941*
15.	*- 1.8414057*	*1.1513625*
16.	*- 1.8414057*	*1.1463574*
17.	*- 1.8414057*	*1.1461938*
18.	*- 1.8414057*	*1.1461932*

Dengan batas keakuratan sampai enam angka di belakang koma, metode Tali Busur konvergen ke akar $r_1 = -1.8414057$ pada iterasi ke–5 dan akar $r_2 = 1.1461932$ pada iterasi ke-17. Bandingkan hasil ini dengan hasil pada contoh 3.6! □

Metode Tali Busur memiliki tingkat kekonvergenan $p = (1+\sqrt{5})/2 \approx 1.618$. Bukti hal ini dapat ditemui pada buku-buku analisis numerik [5, 6].

(a) Akar nyata $x^3 - x^2 - x - 1 = 0$

(b) Akar $x = 1 + 0.3\cos(x)$

(c) Akar positif terkecil $\cos(x) = \frac{1}{2} + \sin(x)$

(d) Akar $x = e^x$

(e) Akar positif terkecil $e^{-x} = \sin(x)$.

4. Selesaikan persamaan di bawah ini dengan metode Tali Busur,

$$f(r) = P[(1+r)^N - 1] - Q[1 - (1+r)^{-M}] = 0$$

jika diketahui $P = 1000$, $Q = 20000$, $N = 30$, dan $M = 20$. Gunakan batas toleransi $\epsilon = 0.0001$.

5. Selesaikan persamaan di bawah ini dengan metode Tali Busur:

$$f(x) = x^3 - 3x^2 + 3x - 1 = 0$$

dengan menggunakan keakuratan $\epsilon = 10^{-6}$. Gunakan beberapa cara untuk menghitung nilai $f(x)$, misalnya (i) dengan menggunakan rumus fungsi tersebut, (ii) dengan membalik urutan suku-suku $f(x)$, dan (iii) dengan perkalian tersarang. Cobalah dengan beberapa hampiran awal, misalnya $x_0 = 0$ dan $x_1 = 1.5$, $x_0 = 0.5$ dan $x_1 = 2.0$, dan $x_0 = 0.5$, dan $x_1 = 1.1$. Catat hasil-hasil tak wajar yang diperoleh, dan jelaskan! Ingat, satu-satunya akar adalah $r = 1$.

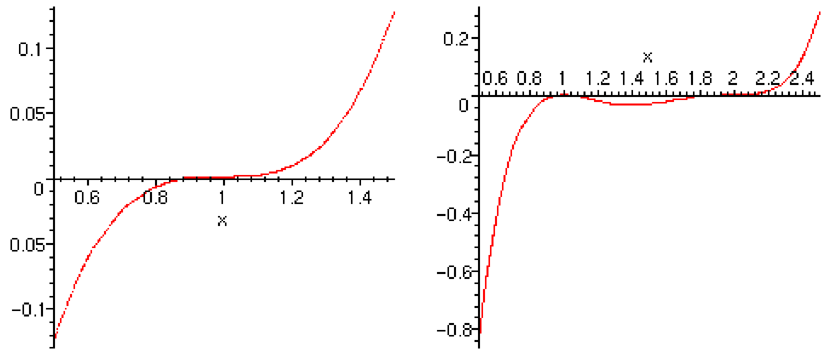
6. Dengan menggunakan metode Tali Busur, carilah semua akar persamaan

$$f(x) = 32x^6 - 48x^4 + 18x^2 - 1 = 0.$$

Akar-akar eksaknya adalah

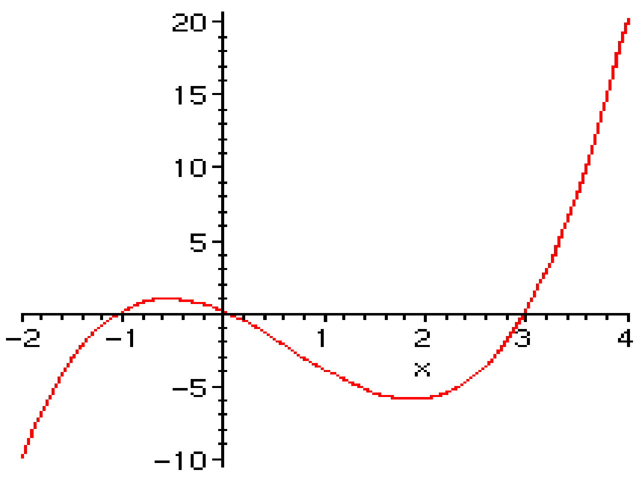
$$\cos[(2k-1)\frac{\pi}{12}], \quad k = 1,\ 2,\ ..., 6.$$

7. Tulislah secara lengkap tentang perbandingan metode Newton – Raphson dan metode Tali Busur.



(a) $y = (x-1)^3$ (b) $y = (x-1)^2(x-2)^3$

Gambar 3.10: Interval Ketidakpastian untuk Akar-akar Ganda



Gambar 3.11: Kurva $y = x(x-2)(x+1)$ dan ketiga akar sederhana

kurva yang memuat akar-akar ganda. Perhatikan bahwa kurva di sekitar akar ganda sangat berdekatan dengan sumbu-x, sehingga letak titik potong yang sebenarnya kurva dengan sumbu-x tidak terlihat dengan jelas. Bandingkan dengan gambar 3.11 di mana titik-titik potong kurva dengan sumbu-x terlihat secara jelas, karena akar-akar yang bersangkutan adalah akar-akar sederhana.

Di depan sudah dijelaskan bahwa kekonvergenan metode Newton–Raphson ke akar ganda bersifat lebih lambat (linier) daripada ke akar sederhana (konvergen secara kuadratik). Satu-satunya cara untuk menghindari adanya akar ganda adalah dengan mendefinisikan fungsi lain yang memiliki akar sama, namun merupakan akar sederhana fungsi baru. Misalnya, jika r adalah akar ganda berderajad m, dari $f(x) = 0$, maka fungsi $F(x) = f^{(m-1)}(x)$ memiliki r sebagai akar sederhana, dan pemakaian metode Newton–Raphson pada $F(x)$ akan konvergen secara kuadratik ke r.

3.7 Pencarian Akar dengan MATLAB

3.7.1 Akar Polinomial

Jika $f(x)$ merupakan suatu polinomial berderajad n, yakni

$$f(x) = a_n x^n + a_{n-1}x^{n-1} + ... + a_2 x^2 + a_1 x + a_0, \tag{3.34}$$

dengan $a_n \neq 0$, maka terdapat n nilai x yang memenuhi $f(x) = 0$. Hal ini dapat diketahui dari pelajaran aljabar. Jika akar-akar polinomial (3.34) adalah r_k, $k = 1,\ 2,\ ..., n$, maka polinomial tersebut dapat difaktorkan menjadi

$$\begin{aligned} f(x) &= a_n x^n + a_{n-1}x^{n-1} + ... + a_2 x^2 + a_1 x + a_0 \\ &= a_n(x - r_n)(x - r_{n-1})...(x - r_2)(x - r_1). \end{aligned} \tag{3.35}$$

Beberapa akar mungkin bernilai sama (merupakan akar ganda) dan beberapa akar lain mungkin berupa bilangan kompleks. Sebagai contoh, polinomial

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

mempunyai sebuah akar kubik di $x = 2$, sebuah akar sederhana di $x = -1$, dan akar-akar kompleks $x = -\frac{1}{2}(1 \pm i\sqrt{3})$.

dengan hampiran `x0`, kemudian bergerak ke kedua arah hingga terjadi perubahan tanda nilai fungsi, selanjutnya mencari akar dengan menggunakan metode bagi dua. Peningkatan algoritma dilakukan dengan menggunakan interpolasi linier, kuadratik, dsb. untuk menghampiri fungsinya, berdasarkan perilaku fungsi di sekitar `x0`.

Perlu dicatat dalam menggunakan metode bagi dua, bahwa suatu fungsi mungkin berganti tanda lebih satu kali pada suatu interval. Dalam hal ini, akar yang diperoleh mungkin bukan akar yang diharapkan. Sebaliknya, suatu fungsi mungkin tidak pernah berganti tanda, namun tetap memiliki akar. Contoh fungsi demikian adalah $f(x) = (x-1)^2$. Baik fungsi MATLAB `fzero` maupun metode bagi dua tidak dapat menghampiri akar ganda berderajad genap.

LATIHAN 3.7

1. **Sifat-sifat Akar Polinomial** Perkirakan akar-akar polinomial-polinomial di bawah ini tanpa menggunakan kalkulator atau komputer. Anda dapat menggunakan aturan Descartes, menghitung nilai fungsi untuk $x = 0$ dan $x = \pm 1$, atau "Dalil Sisa" dari Aljabar.

   (a) Tentukan keempat akar $f(x) = x64 + 6x^3 + 3x^2 - 10x$.

   (b) Tunjukkan bahwa fungsi $g(x) = 8x^3 + 12x^2 + 14x + 9$ mempunyai tepat sebuah akar nyata, dan terletak pada interval $-1 \leq x \leq 0$.

   (c) Gunakan nilai turunan fungsi $h(x) = x^4 - 2x^3 + 2x^2 - 2x + 1$ pada salah satu akarnya untuk menentukan ketiga akar lainnya.

   (d) Tunjukkan bahwa fungsi $g(x) = x^4 + 7.7x^3 + 39.1x^2 + 14.4x - 13$ mempunyai tepat sebuah akar nyata positif dan sebuah akar nyata negatif, dan akar positifnya kurang daripada 1, dan bagian riil akar kompleksnya lebih besar daripada 3.3.

2. **Mencari akar polinomial dengan perintah** `roots`

   (a) Gunakan perintah MATLAB `roots` untuk mencari akar-akar fungsi polinomial pada soal-soal di atas.

   (b) Gunakan perintah MATLAB `compan` dan `eig` untuk menghitung dan mendiagonalkan matriks penyerta setiap polinomial pada soal di atas. Hasilnya harus sama dengan butir a.

(c) Apakah Anda mengira perintah `roots` akan kesulitan mencari akar-akar ganda suatu polinomial? Jelaskan. Polinomial $f(x) = 25x^4 - 135x^3 + 256x^2 - 231x + 121$ mempunyai akar ganda. Carilah akar-akarnya dengan perintah `roots`.

(d) Gunakan perintah MATLAB `conv` untuk mengalikan sebarang dua buah polinomial di atas, kemudian carilah akar-akar polinomial hasil kalinya. Jelaskan bahwa akar-akarnya merupakan akar-akar kedua polinomial yang dikalikan. Gunakan perintah MATLAB `deconv` untuk membagi polinomial hasil kali tersebut dengan berturut-turut $(x - r_k)$, dengan r_k adalah salah satu akar, untuk menjelaskan bahwa sisa pembagiannya sama dengan nol.

3. **Mencari akar dengan `fzero`.** Carilah akar-akar fungsi-fungsi di bawah ini dengan menggunakan perintah MATLAB `fzero`. Sebelumnya gambarlah grafik kurva masing-masing fungsi untuk menentukan hampiran awalnya.

   (a) $xe^{-x^2} - \cos x$

   (b) $\ln x - x/5$

   (c) $(x/\sin x)^2 - \ln x/x^2 - 1.415$

4. **Akar ganda** Tunjukkan bahwa perintah MATLAB `fzero` tidak mampu mencari akar-akar ganda dengan mengambil contoh $f(x) = (x-3)^4(x^2+2x+1)$. Selanjutnya, gunakan metode Newton – Raphson dan Newton – Raphson termodifikasi untuk menghampiri akar ganda $r = -1$, yang berderajad 2, dan $r = 3$, yang berderajad 4.

5. Bandingkan lamanya waktu eksekusi program `secant` (yang Anda buat) dengan fungsi MATLAB `fzero` untuk mencari hampiran akar $xe^{-x^2} - \cos x = 0$. Gunakan kriteria kekonvergenan (batas toleransi) yang sama untuk kedua fungsi tersebut. Gunakan fungsi `tic` dan `toc` untuk menghitung lamanya waktu eksekusi.

3.8 Rangkuman

Akar Persamaan, Pembuat Nol Fungsi Misalkan $f(x)$ adalah suatu fungsi kontinyu. Setiap bilangan r pada domain f yang memenuhi $f(r) = 0$ disebut **akar persamaan** $f(x) = 0$, atau juga disebut **pembuat nol fungsi** $f(x)$. Secara singkat, r sering disebut **akar fungsi** $f(x)$.

Derajad suatu Akar Persamaan Misalkan r adalah akar persamaan $f(x) = 0$. Jika terdapat bilangan asli m dan fungsi kontinyu $h(x)$ dengan $h(r) \neq 0$, sedemikian hingga $f(x)$ dapat dinyatakan sebagai

$$f(x) = (x - r)^m h(x),$$

maka r disebut **akar berderajad** m. Jika r pembuat nol $f(x)$ berderajad m, maka

$$f(r) = f'(r) = ... = f^{(m-1)}(r) = 0, \quad \text{dan} \quad f^m(r) \neq 0.$$

Untuk $m = 1$, r disebut **akar sederhana**, dan untuk $m > 1$, r disebut **akar ganda**. Untuk $m = 2$, r disebut **akar dobel**, dst.

Berikut adalah rangkuman dari beberapa metode yang dapat digunakan untuk mencari hampiran akar persamaan

$$f(x) = 0$$

yang telah dibahas pada bab ini.

Metode Bagi Dua Apabila diketahui dua buah titik awal $x = a$ dan $x = b$ sedemikian hingga $f(a) \times f(b) < 0$, maka terdapat akar persamaan pada interval $[a, b]$. Untuk menghampiri akar tersebut, interval $[a, b]$ dibagi menjadi dua buah subinterval sama panjang dengan titik tengah

$$x_n = \frac{(a+b)}{2}, \qquad n = 1, 2, 3, ...$$

Untuk setiap $n \geq 1$, jika $f(a) \times f(x) < 0$, maka $b = x_n$ (ganti b dengan x_n). Jika kasusnya lain, maka $a = x_n$ (ganti a dengan x_n). Proses diulangi sampai diperoleh subinterval yang memuat akar dengan lebar "sangat kecil", yakni $|b - a| < \epsilon$ untuk nilai ϵ yang diberikan.

Kekonvergenan Metode Bagi Dua Misalkan metode Bagi Dua digunakan pada sebuah fungsi kontinyu $f(x)$ pada interval $[a, b]$ di mana

Syarat-syarat keberadaan dan ketunggalan titik tetap suatu fungsi Misalkan $g(x)$ adalah fungsi kontinyu pada interval $[a,\ b]$.

1. Jika g memenuhi
$$a \le g(x) \le b \qquad \text{untuk semua } x \in (a,\ b),$$
maka persamaan $x = g(x)$ memiliki sedikitnya sebuah penyelesaian r di dalam $[a,\ b]$.
2. Selanjutnya, jika $g'(x)$ terdefinisi pada $(a,\ b)$ dan jika
$$|g'(x)| < 1 \qquad \text{untuk semua } x \in (a,\ b),$$
maka g memiliki titik tetap tunggal pada $[a,\ b]$.

Kekonvergenan Iterasi Titik Tetap Misalkan metode Titik Tetap digunakan untuk menghitung hampiran-hampiran titik tetap $x_{n+1} = g(x_n)$. Misalkan interval $I = [a,\ b]$ memuat titik tetap $r = \frac{a+b}{2}$ dan hampiran awal titik tetap x_0. Apabila terdapat bilangan $0 \le K < 1$ sedemikian hingga
$$|g'(x)| \le K \quad \text{untuk semua} \quad x \in I,$$
maka barisan hampiran titik-titik tetap $\{x_n\}_{n=0}^{\infty}$ konvergen ke r.

Derajad Kekonvergenan Misalkan $x_0,\ x_1,\ x_2,\ \ldots$ suatu barisan yang konvergen ke r dan misalkan $e_n = r - x_n$. Apabila terdapat sebuah bilangan m dan sebuah konstanta $C \neq 0$, sedemikian hingga
$$\lim_{n\to\infty} \frac{|e_{n+1}|}{|e_n|^m} = C$$
maka m disebut **derajad kekonvergenan** barisan tersebut dan C disebut **konstanta galat asimptotik**. Untuk $m = 1,\ 2,\ 3$, kekonvergenannya berturut-turut disebut *linier*, *kuadratik*, dan *kubik*.

Derajad Kekonvergenan Iterasi Titik Tetap Jika $g'(r) = g''(r) = \ldots g^{(m-1)}(r) = 0$ dan $g^{(m)}(r) \neq 0$ dan iterasi $x_{n+1} = g(x_n)$ konvergen ke $r = g(r)$, maka ia konvergen dengan derajad kekonvergenan m.

sedemikian hingga

$$\frac{|f(x)f''(x)|}{[f'(x)]^2} < 1, \quad \forall x \in [r-\delta,\ r+\delta].$$

Laju Kekonvergenan Iterasi Newton – Raphson Misalkan barisan $\{x_n\}_{n=0}^{\infty}$ yang dihasilkan oleh iterasi Newton – Raphson $x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$ konvergen ke akar r, di mana $f(r) = 0$. Misalkan $e_n = x_n - r$ adalah galat hampiran ke-n.

1. Jika r adalah akar sederhana, maka kekonvergenan tersebut kuadratik, dan
$$\lim_{n\to\infty} \frac{|e_{n+1}|}{e_n^2} = \left|\frac{f''(r)}{2f'(r)}\right|.$$

2. Jika r adalah akar ganda berderajad $m > 1$, maka kekonvergenannya linier, dan
$$\lim_{n\to\infty} \left|\frac{e_{n+1}}{e_n}\right| = \frac{m-1}{m}.$$

Iterasi Newton – Raphson Dipercepat Misalkan iterasi Newton – Raphson menghasilkan barisan hampiran yang konvergen secara linier ke akar ganda r yang berderajad $m > 1$. Modifikasi rumus Newton – Raphson
$$x_{n+1} = x_n - \frac{mf(x_n)}{f'(x_n)},$$
akan menghasilkan barisan hampiran $\{x_n\}_{n=0}^{\infty}$ yang konvergen kuadratik ke r. Cara lain untuk mempercepat kekonvergenan iterasi Newton – Raphson ke akar ganda adalah dengan mendefinisikan fungsi baru $F(x)$ dari $f(x)$ sehingga r merupakan akar sederhana $F(x) = 0$. Dalam hal ini dapat didefinisikan fungsi

$$F(x) = f^{(m-1)}(x).$$

Metode Tali Busur Metode ini memerlukan dua buah hampiran awal x_0 dan x_1, hampiran berikutnya diperoleh dengan menghitung absis titik potong garis busur yang melalui titik-titik $(x_0,\ f(x_0))$ dan