Optimasi Tanpa Kendala: Satu Peubah

Pertemuan ke-4

STA1373- Optimisasi Statistika

Akbar Rizki, S.Stat., M.Si

Outline

Elimination Method

- Search with fixed step size
- Exhaustive Search
- Dichotomous/ Binary Search
- **Fibonacci Search**
- Golden Section Search
- Simplex Search

Interpolation Method

- Requiring no derrivatives: **Quadratic interpolation**
- Requiring drrivatives: Cubic, Direct root (Newton, Quasi-Newton, Secant)

Pengantar: Optimasi Tak Linier

Optimasi tak linier adalah pencarian dan penentuan nilai optimum (maksimum dan atau minimum) dari suatu persamaan nonlinier, yang secara umum dapat dituliskan sebagai berikut:

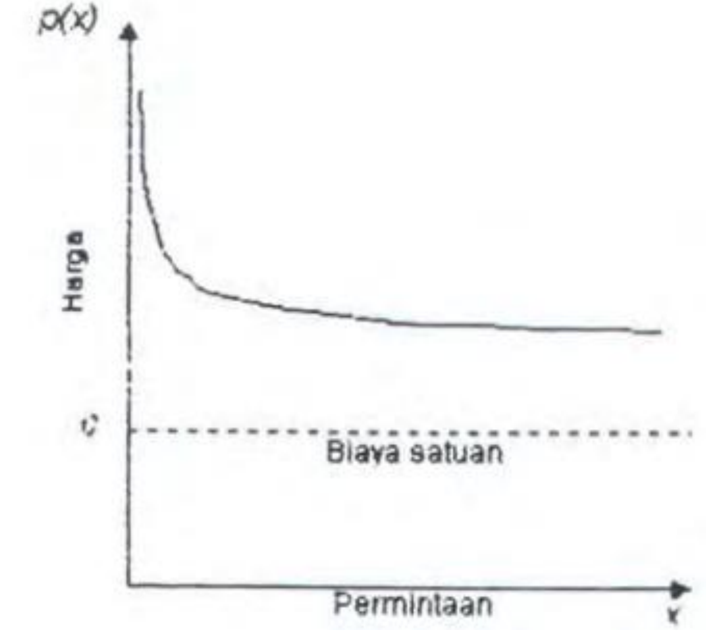
$$f(x^*) = \min_x \; f(x), x \in \mathbb{R}$$

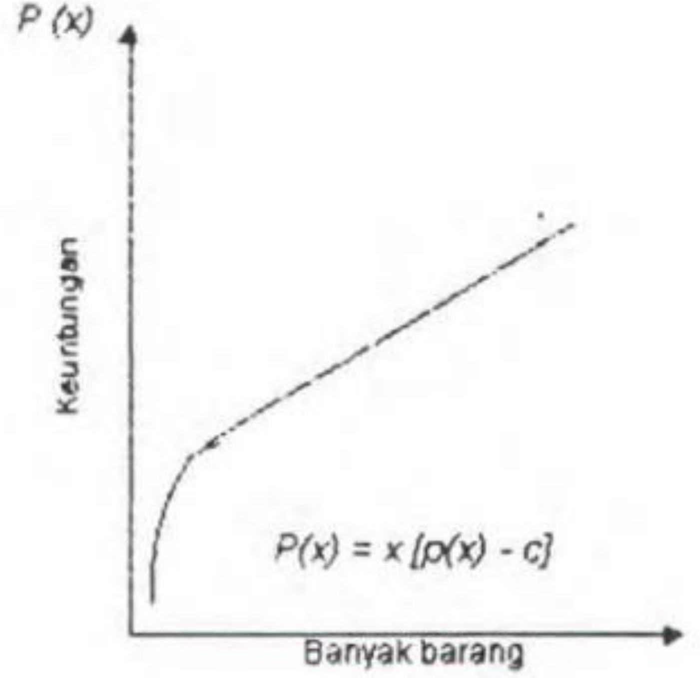
Dimana $f(x)$ merupakan fungsi nonlinier yang dapat melibatkan n buah peubah.

Contoh permasalahan

Masalah produk campuran dan elastisitas harga

Penentuan fungsi keuntungan pada perusahaan besar tidak hanya dipengaruhi oleh jenis produk saja, melainkan dapat juga dipengaruhi oleh elastisitas harga di mana banyaknya barang yang dapat dijual berbanding terbalik dengan harganya. Hal ini menjadikan kurva harga permintaan menjadi tidak linear.





NUMERICAL EVALUATION OF ROOTS OF EQUATIONS

- Metode yang sebelumnya dipelajari adalah metode berbasis iteratif, Dimana algoritma pencarian titik optimum adalah sbb:

Remarks. For all iterative procedures, care should be taken, especially, of the following:

(i) In choosing the initial value.

(ii) In choosing an algorithm that converges.

(iii) Rate of convergence should be known.

(iv) Stopping criterion should be given in advance.

We may use one of the following stopping criteria:

(a) Stop when the gradient vector g_i is small, i.e., $|g_i| < \epsilon$ for a given ϵ.

(b) Components of iterates do not change. That is,

$$\max_j |j\text{th component of } g_i| < \epsilon.$$

(c) The function values do not change much. That is, for a given ϵ,

$$|f(\boldsymbol{x}_n) - f(\boldsymbol{x}_{n+1})| < \epsilon$$

at the nth stage.

DIRECT SEARCH METHOD

When the objective function to be optimized is either too complicated or is unavailable for direct computation in a closed mathematical form, optimum-seeking procedures using search strategies can be used. Sometimes, the functional relationship can be estimated from available data for the problem and the estimated relationship can then be used for the purpose of optimization. **However, such procedures may not always be feasible, and direct search procedures are used.**

Pengantar: Optimasi Tanpa Kendala Satu Peubah

- Fungsi yang akan dioptimumkan merupakan fungsi $f(x)$ unimodal.
- Fungsi unimodal adalah fungsi yang hanya memiliki satu puncak (maksimum) atau lembah (minimum)dalam interval tertentu.
- Oleh karena itu syarat perlu dan cukup agar penyelesaian $x = x^*$ menjadi optimal (maksimum global) adalah:

$$\frac{df}{dx} = 0 \; pada \; x = x^*$$

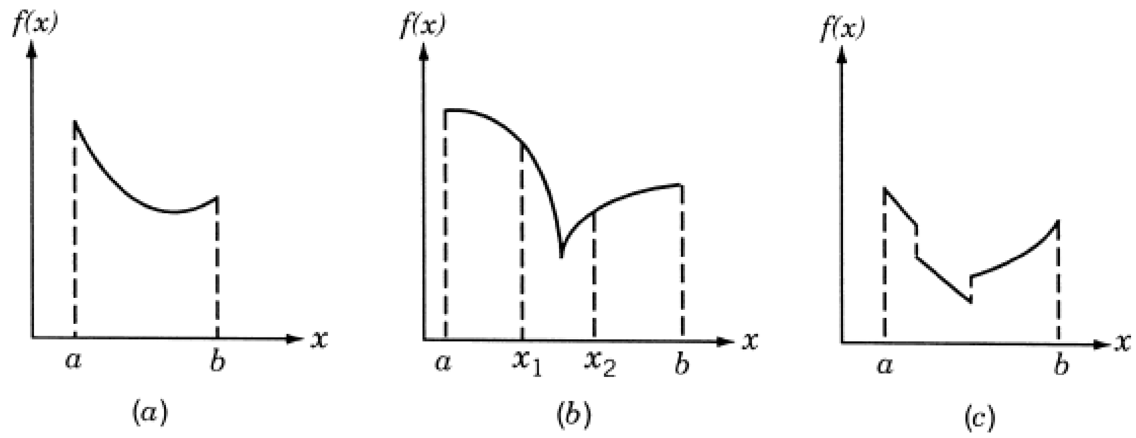


Figure 5.4 Unimodal function.

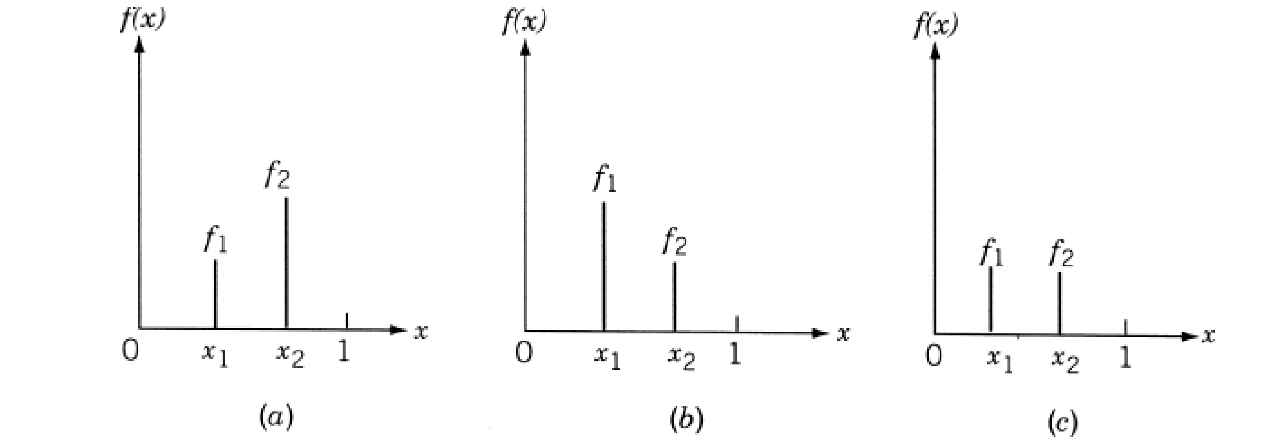


Figure 5.5 Outcome of first two experiments: (a) $f_1 < f_2$; (b) $f_1 > f_2$; (c) $f_1 = f_2$.

SEARCH WITH FIXED STEP SIZE

- The most elementary approach for such a problem is to use a fixed step size and move from an initial guess point in a favorable direction (positive or negative).
- Algoritma:

1. Start with an initial guess point, say, x_1.

2. Find $f_1 = f(x_1)$.

3. Assuming a step size s, find $x_2 = x_1 + s$.

4. Find $f_2 = f(x_2)$.

5. If $f_2 < f_1$, and if the problem is one of minimization, the assumption of unimodality indicates that the desired minimum cannot lie at $x < x_1$. Hence the search can be continued further along points $x_3, x_4, \ldots$ using the unimodality assumption while testing each pair of experiments. This procedure is continued until a point, $x_i = x_1 + (i-1)s$, shows an increase in the function value.

6. The search is terminated at x_i, and either x_{i-1} or x_i can be taken as the optimum point.

7. Originally, if $f_2 > f_1$, the search should be carried in the reverse direction at points $x_{-2}, x_{-3}, \ldots$, where $x_{-j} = x_1 - (j-1)s$.

8. If $f_2 = f_1$, the desired minimum lies in between x_1 and x_2, and the minimum point can be taken as either x_1 or x_2.

9. If it happens that both f_2 and f_{-2} are greater than f_1, it implies that the desired minimum will lie in the double interval $x_{-2} < x < x_2$.

EXHAUSTIVE SEARCH

The exhaustive search method can be used to solve problems where the interval in which the optimum is known to lie is finite. Let xs and xf denote, respectively, the starting and final points of the interval of uncertainty. The exhaustive search method consists of evaluating the objective function at a predetermined number of equally spaced points in the interval (xs, xf), and reducing the interval of uncertainty using the assumption of unimodality.

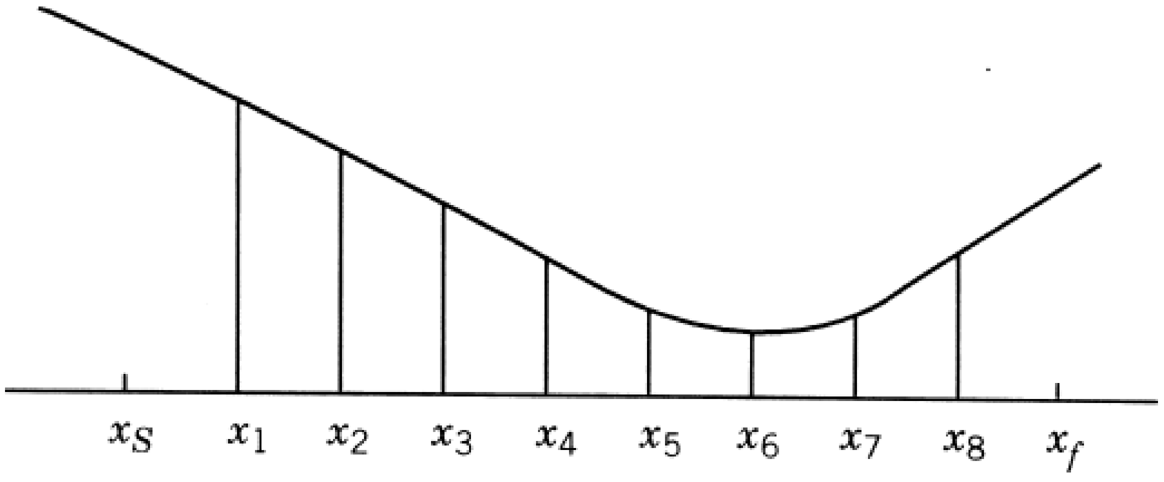


Figure 5.6 Exhaustive search.

DICHOTOMOUS SEARCH

The exhaustive search method can be used to solve problems where the interval in which the optimum is known to lie is finite. Let xs and xf denote, respectively, the starting and final points of the interval of uncertainty. The exhaustive search method consists of evaluating the objective function at a predetermined number of equally spaced points in the interval (xs, xf), and reducing the interval of uncertainty using the assumption of unimodality.

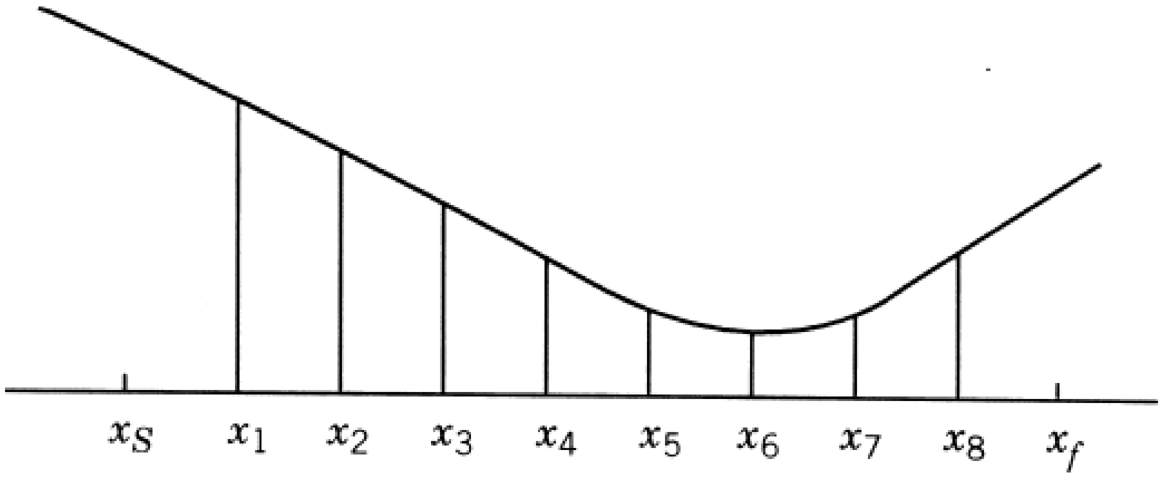


Figure 5.6 Exhaustive search.

FIBONACCI SEARCH METHOD

As stated earlier, the *Fibonacci method* can be used to find the minimum of a function of one variable even if the function is not continuous. This method, like many other elimination methods, has the following limitations:

1. The initial interval of uncertainty, in which the optimum lies, has to be known.

2. The function being optimized has to be unimodal in the initial interval of uncertainty.

3. The exact optimum cannot be located in this method. Only an interval known as the *final interval of uncertainty* will be known. The final interval of uncertainty can be made as small as desired by using more computations.

4. The number of function evaluations to be used in the search or the resolution required has to be specified beforehand.

FIBONACCI SEARCH METHOD

- Pada metode ini kita harus menentukan lebih dulu dua bilangan sebagai interval dimana nilai minimum akan kita cari pada interval tersebut.
- Setiap iterasi pada metode ini memerlukan deret Fibonacci, dimana ditentukan sebagai berikut:

$$F_{v+1} = F_v + F_{v-1}$$

Dimana F_0 dan $F_1 = 1\,;\mathrm{v} = 1, 2, 3, \dots, \mathrm{dst}$

Deret Fibonacci adalah: 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, ….

FIBONACCI SEARCH METHOD

Algoritma:

- Menentukan interval awal yaitu a_1 dan b_1
- Memilih jumlah iterasi "n"
- Menentukan λ_1 dan μ_1 dan k=1 dan $\varepsilon = 0.01$

nilai bawah $\quad \lambda_k = a_k + \frac{F_{n-k-1}}{F_{n-k+1}}(b_k - a_k) \qquad k = 1,\ldots,n-1.$

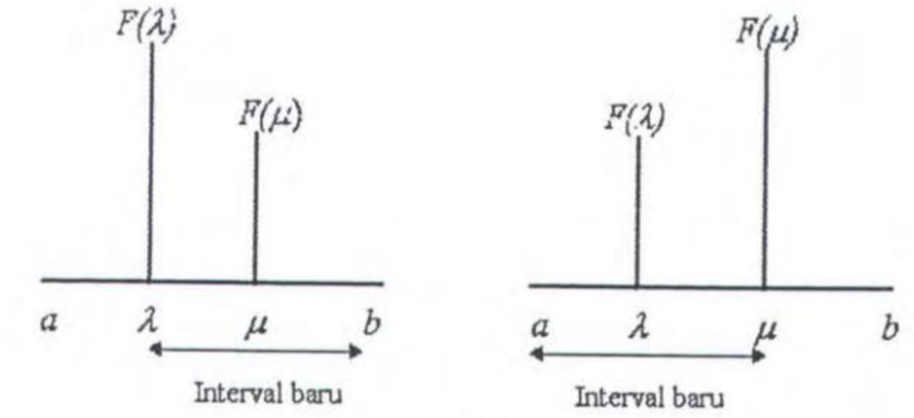
nilai atas $\quad \mu_k = a_k + \frac{F_{n-k}}{F_{n-k+1}}(b_k - a_k) \qquad k = 1,\ldots,n-1.$

- Menghitung $F(\lambda_k)$ dan $\mathrm{F}(\mu_k)$

Jika $F(\lambda_k) > F(\mu_k)$ maka interval yang baru $[\lambda_k, b_k]$

Jika $F(\lambda_k) \le F(\mu_k)$ maka interval yang baru $[a_k, \mu_k]$,

Keterangan:
a_k= batas bawah
b_k= batas atas
n = jumlah evaluasi fungsi



FIBONACCI SEARCH METHOD

Result: Estimate an approximate value of x^*, and determine $f(x^*)$ and R

Define $f(x)$; `// Define the objective function`

Initialize x_l, x_r and n;

Calculate $L_0 \leftarrow x_r - x_l$; `// Initial interval of uncertainty`

Calculate $L_i \leftarrow \frac{F_{n-2}}{F_n} L_0$;

Calculate $R_1 \leftarrow \frac{L_i}{L_0}$; `// Initial reduction ratio`

for $i \leftarrow 2$ **to** n **by** 1 **do**

  if $L_i \geq \frac{L_0}{2}$ `// Compare whether` $x_1 < x_2$

  then

    Assign $x_1 \leftarrow x_r - L_i$;

    Assign $x_2 \leftarrow x_l + L_i$;

  else

    Assign $x_1 \leftarrow x_l + L_i$;

    Assign $x_2 \leftarrow x_r - L_i$;

  end if

  Calculate $f_1 \leftarrow f(x_1)$;

  Calculate $f_2 \leftarrow f(x_2)$;

  if $f_1 < f_2$ **then**

    Assign $x_r \leftarrow x_2$;

    Assign $F_i \leftarrow \frac{F_{n-i}}{F_{n-(i-2)}} L_0$; `// New interval of uncertainty generated`

  else if $f_1 > f_2$ **then**

    Assign $x_l \leftarrow x_1$;

    Assign $F_i \leftarrow \frac{F_{n-i}}{F_{n-(i-2)}} L_0$; `// New interval of uncertainty generated`

  else

    Assign $x_l \leftarrow x_1$;

    Assign $x_r \leftarrow x_2$;

    Assign $F_i \leftarrow \frac{F_{n-i}}{F_{n-(i-2)}} (x_r - x_l)$; `// New interval of uncertainty generated`

  end if

  Assign $L_0 \leftarrow x_r - x_l$;

  Assign $R \leftarrow \frac{L_i}{R_1}$; `// New reduction ratio`

end for

if $f_1 \leq f_2$ **then**

  Return $x^* \leftarrow x_1$;

  Return $f(x^*) \leftarrow f(x_1)$;

  Return R;

else

  Return $x^* \leftarrow x_2$;

  Return $f(x^*) \leftarrow f(x_2)$;

  Return R;

end if

FIBONACCI SEARCH METHOD

- Contoh:

Misalkan diketahui sebuah fungsi tak linear sebagai berikut:

$$f(x) = x^5 - 5x^3 - 20x + 5$$

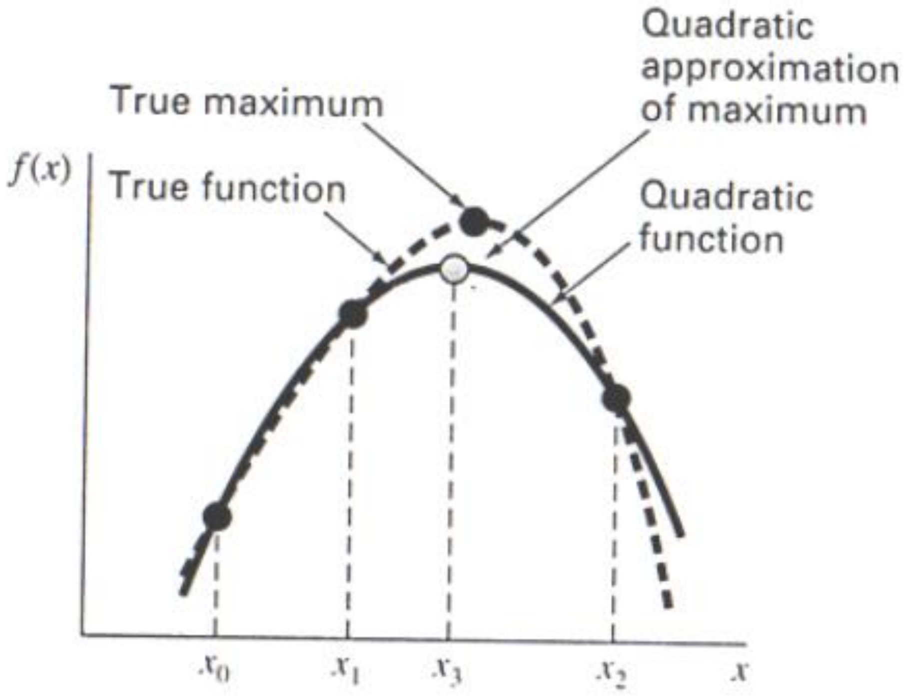
Fibonacci search method akan digunakan untuk mencari nilai x^* minimum. Misalkan n=25 dan interval awal adalah [-2.5; 2.5].

Maka sintaks R untuk menyelesaikan masalah tersebut adalah:

Terlampir pada file: Fibonacci.R

QUADRATIC INTERPOLATION

- Metode interpolasi kuadratik melibatkan tiga titik tebakan awal dalam mencari nilai optimum.
- Penggunaan polinomial orde-dua menghasilkan pendekatan cukup baik terhadap bentuk f (x) di dekat titik optimum, sehingga Metode interpolasi kuadrat dapat digunakan untuk melakukan optimasi secara numerik



QUADRATIC INTERPOLATION

- Algoritma

Result: Estimate an approximate value of x^* and determine $f(x^*)$

Define $f(x)$; // Define the objective function

Define $f_1(x,y)$; // Define the first order forward divided difference $f[x,y]$

Define $f_2(x,y,z)$; // Define the second order forward divided difference $f[x,y,z]$

Define $nt(seq,n)$; // Function to find out the nearest value to a number, n from a list, seq

Define $ft(seq,n)$; // Function to find out the furthest value to a number, n from a list, seq

Define $maxim\_fv(seq)$; // Function to find out the element from the seq that has highest value of $f(x)$

Initialize the starting experimental point x, the discrete step size s, the maximum step size m and the tolerance ϵ;

Calculate $f_x \leftarrow f(x)$;

Calculate $f_s \leftarrow f(x+s)$;

if $f_x < f_s$ **then**

  Calculate $x_0 \leftarrow x - s$; // Set x_0

  Calculate $x_1 \leftarrow x$; // Set x_1

  Calculate $x_2 \leftarrow x + s$; // Set x_2

else

  Calculate $x_0 \leftarrow x$; // Set x_0

  Calculate $x_1 \leftarrow x + s$; // Set x_1

  Calculate $x_2 \leftarrow x + 2s$; // Set x_2

end if

while *TRUE* **do**

  Calculate $x_t \leftarrow \frac{f_2(x_0,x_1,x_2)(x_0+x_1)-f_1(x_0,x_1)}{2f_2(x_0,x_1,x_2)}$; // Calculate the approximate minimizer

  Calculate $x_n \leftarrow nt((x_0,x_1,x_2),x_t)$; // Picks the point from (x_0,x_1,x_2) which is the nearest to x_t

  Calculate $x_f \leftarrow ft((x_0,x_1,x_2),x_t)$; // Picks the point from (x_0,x_1,x_2) which is the furthest to x_t

  if $(f_2(x_0,x_1,x_2) > 0)$ & $(|x_t - x_n| > m)$ **then**

    Remove x_f from (x_0,x_1,x_2);

    Take a step of size m towards the direction of descent from the point with the lowest value;

  else if $f_2(x_0,x_1,x_2) < 0$ **then**

    Remove x_n from (x_0,x_1,x_2);

    Take a step of size m towards the direction of descent from the point with the lowest value;

  else

    if $|x_t - x_n| < \epsilon$ // Check for terminating condition **then**

      Break;

      Return $x^* \leftarrow \frac{x_t+x_n}{2}$;

      Return $f(x^*)$;

    else

      Replace $maxim\_fv((x_0,x_1,x_2))$ from (x_0,x_1,x_2) with x_t ;

    end if

  end if

end while

QUADRATIC INTERPOLATION

- Contoh:

Misalkan diketahui sebuah fungsi tak linear sebagai berikut:

$$f(x) = x^4 - 2x^2 + \frac{1}{4}$$

Quadratic Interpolation akan digunakan untuk mencari nilai minimum x^* dari fungsi tersebut.

Maka sintaks R untuk menyelesaikan masalah tersebut adalah:

Terlampir pada file: Newton Quadratic.R

NEWTON'S METHOD

- Metode Newton (atau seringkali disebut dengan metode Newton–Raphson) memerlukan fungsi tujuan tanpa kendala dalam interval yang menjadi perhatian dan mempunyai derivasi pertama maupun keduanya. Metode ini banyak pula dikembangkan untuk memecahkan permasalahan optimasi multi variabel. Metode Newton seringkali dipandang sebagai metode untuk mencari akar dari suatu fungsi.
- Pada metode ini pertama kita menentukan nilai awal (x_0), selanjutnya dari nilai awal ini, sebuah garis singgung dapat diperluas dari titik (x_0 ,f(x_0)). Titik dimana garis singgung memotong garis sumbu x biasanya menunjukkan sebuah taksiran perbaikan dari harga x. Metode Newton dapat diturunkan berdasarkan interpretasi geometri k atau sebuah metode alternatif yang didasarkan pada deret Taylor.

NEWTON'S METHOD

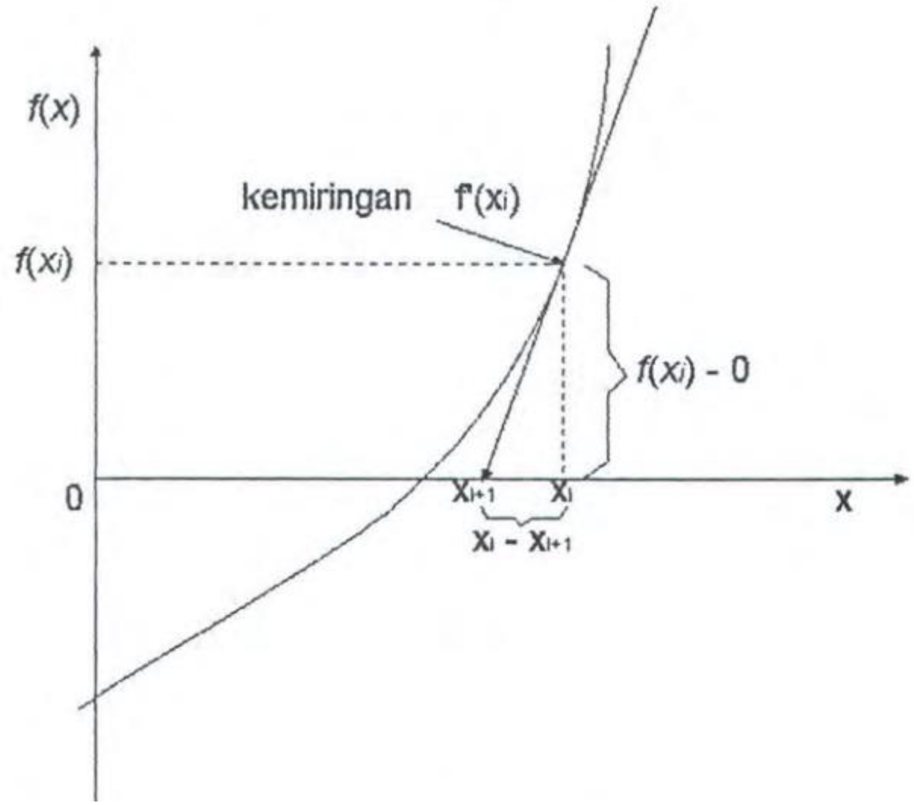
$$F(x) = f(x^{(k)}) + f'(x^{(k)})(x - x^{(k)}) + \frac{1}{2} f''(x^{(k)})(x - x^{(k)})^2$$

$$F'(x) = f'(x^{(k)}) + f''(x^{(k)})(x^* - x^{(k)}) = 0$$

$$x^* = x^{(k)} - \frac{f'(x^{(k)})}{f''(x^{(k)})}$$

$$x^{(k+1)} = x^{(k)} - \frac{f'(x^{(k)})}{f''(x^{(k)})}$$

$$\left| x^{(k+1)} - x^{(k)} \right| < \varepsilon$$



NEWTON'S METHOD

- Algoritma:

Result: Estimate an approximate value of the root x^* and determine $f(x^*)$

Define $f(x)$; // Define the objective function

Define $fprime(x) \leftarrow \frac{df}{dx}(x)$; // Define the first derivative of the objective function

Define $fprime2(x) \leftarrow \frac{d^2f}{dx^2}(x)$; // Define the second derivative of the objective function

Initialize the initial starting point x_j, close to x^*, and the tolerance, ϵ;

while *TRUE* **do**

    Assign $f' \leftarrow fprime(x_j)$; // First derivative of $f(x)$ at x_j

    Assign $f'' \leftarrow fprime2(x_j)$; // Second derivative of $f(x)$ at x_j

    Calculate $x_{j+1} \leftarrow x_j - \frac{f'}{f''}$; // Update the next iterate

    if $|x_{j+1} - x_j| \leq \epsilon$ // Checking the termination condition

    then

        Break;

        Return $x^* \leftarrow x_{j+1}$;

        Return $f(x^*)$;

    else

        Assign $x_j \leftarrow x_{j+1}$;

    end if

end while

NEWTON'S METHOD

- Contoh

Misalkan diketahui sebuah fungsi tak linear sebagai berikut:

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

Newton method akan digunakan untuk mencari nilai akar x^* dari fungsi tersebut.

Maka sintaks R untuk menyelesaikan masalah tersebut adalah:

Terlampir pada file: Newton Rapshon.R

REFERENSI

- B. S. Everitt. 1987. Introduction to Optimization Methods and their Application in Statistics. Chapman and Hall, New York
- Rao, Singiresu SEngineering Optimization Theory and and Practice, 4th ed., pages 273:279
- https://indrag49.github.io/Numerical-Optimization/solving-one-dimensional-optimization-problems.html
- https://rpubs.com/aaronsc32/newton-raphson-method
- https://github.com/brunasqz/NonlinearOpMethods/blob/versionII/R/quadraticinterpolation.R