# M O D U L
# PRAKTIKUM
# METODE NUMERIK

Penyusun :
Priyo Sidik Sasongko,S.Si,M.Kom


**DEPARTEMEN ILMU KOMPUTER /I NFORMATIKA**
**FAKULTAS SAINS DAN MATEMATIKA**
**UNIVERSITAS DIPONEGORO**
**2020**

# KATA PENGANTAR

Alhamdulillah, Berkat Rahmat Allah SWT, buku petunjuk praktikum metode numerik berhasil kami selesaikan. Semoga bisa sebagai panduan dalam praktikum di laboratorium komputer. Rujukan utama buku ini bersumber pada buku teks standar yang sangat populer di dunia komputasi, yaitu buku yang ditulis oleh Richard L. Burden dan J. Douglas Faires dengan judul *Numerical Analysis* edisi ke-9, diterbitkan oleh Penerbit Brooks/Cole, Thomson Learning Academic Resource Center.

Ibarat pohon yang akan terus tumbuh semakin besar, buku panduan ini pun memiliki tabiat pertumbuhan sebagaimana pohon itu.
Kepada rekan-rekan mahasiswa yang akan *ngambil* mata kuliah tersebut, saya sampaikan permohonan maaf jika rencana ini akan membuat anda kurang tidur karena *bakal* semakin lama berada di depan komputer, menyelesaikan tugas dan *report*. Namun tujuannya untuk memberikan bekal yang lebih pada penguasaan pemrogramannya. Pada edisi ini kita mulai mencoba membiasakan diri menulis *script* dalam lingkungan **Python dan** Jupyter notebook.

Buku panduan ini masih jauh dari sempurna, karena tidak semua materi perkuliahan disajikan di sini dn juga penyajiannya masih perlu perbaikan. Walaupun buku ini masih jauh dari sempurna, namun semoga ini dapat menyumbangkan kontribusi yang berarti bagi terciptanya pemahaman ilmu pengetahuan di bidang metode numerik. Kami menerima masukan, kritikan dan saran, silakan dikirimkan ke email:
**priyoss1234@gmail.com**

# PYTHON DAN JUPYTER NOTEBOOK

TUJUAN :

1. Mengenalkan Bahasa Python
2. Mengenal Jupyter Notebook

## Mengenalkan Bahasa Python

Python merupakan bahasa pemrograman yang bersifat *object-oriented*. Bahasa python dibuat pada tahun 1980an yang namanya diambil dari salah satu tayangan televisi di Inggris yaitu *Monty Python's Flying Circus*. Meskipun python tidak terlalu terkenal dibandingkan dengan bahasa pemrograman lainnya, python dapat diandalkan untuk membuat software aplikasi di bidang sains dan teknik dengan efisien dan elegan. *Script* python tidak perlu di*compile* ke dalam suatu kode mesin apapun, karena ia dapat dijalankan cukup dengan bantuan *interpreter*, seperti halnya Matlab. Keuntungan dari *script* yang bisa dijalankan dengan interpreter adalah ia dapat diuji dan di*debug* dengan cepat, sehingga programmer bisa lebih berkonsentrasi pada prinsip dibalik *script* yang sedang dibangunnya.

Keuntungan ini juga menjadikan aplikasi python lebih cepat dibangun dibandingkan dengan aplikasi yang sama jika dibangun dengan bahasa C maupun Fortran. Di sisi lain, kekurangan *script* python selaku *interpreted program* adalah ia tidak dapat di*compile* sehingga menjadi program aplikasi yang bersifat *stand-alone*. Sehingga suatu *script* python hanya bisa dieksekusi jika pada komputer tersebut sudah terinstall program python.

Python memiliki kelebihan lain yang sangat penting dibanding bahasa pemrograman yang terdahulu:

- Python merupakan *open-source software*, yang artinya ia dapat diperoleh secara gratis. Bahkan python sudah otomatis terinstall di Linux.
- Python tersedia pada semua *operating systems* (OS) terkenal seperti Linux, Unix, Windows,dan MacOS. Suatu script python yang ditulis pada OS tertentu, dapat dijalankan di OS lain tapa ada modifikasi sedikitpun.
- Python lebih mudah dipelajari sekaligus lebih mudah "'dibaca"' dibandingkan dengan bahasa pemrograman lainnya. Python dan program ekstensinya mudah diinstall. Python berdiri di atas landasan pondasi Java and C++. Hal-hal seperti *classes*, *methods*,

*inheritance*, yang kerapkali diimplementasikan pada bahasa yang bersifat *object-oriented*, juga dapat diimplementasikan di python.

**Mengenal Jupyter notebook**

Jupyter notebook adalah software sangat sangat populer beberapa tahun terakhir. ***Jupyter*** (https://jupyter.org/) adalah organisasi non-profit untuk mengembangkan software interaktif dalam berbagai bahasa pemrograman. ***Notebook*** adalah satu software buatan Jupyter, adalah aplikasi web open-source yang memungkinkan Anda membuat dan berbagi dokumen interaktif yang berisi kode *live*, persamaan, visualisasi, dan teks naratif yang kaya. Jupyter Notebook menyatukan baik itu teks/narasi, kode hidup, persamaan, tampilan hasil, gambar statis, dan visualisasi grafis, dalam satu file interaktif. Dan, kelebihan lainnya, notebook dapat dijalankan ulang oleh siapapun yang membukanya, untuk mereproduksi eksekusi kode di dalamnya. Banyak jenis-jenis media lain yang bisa ditampilkan secara hidup, misalnya Markdown, HTML, audio, video, Javascript, dan sebagainya.

Petunjuk Instalasi Jupyter Notebook

Sebelumnya, semestinya Anda harus mempunyai instalasi Python. Kalau belum, maka Anda perlu menginstalasinya terlebih dahulu dari https://www.python.org/downloads/. Pilih Python versi 3 terakhir.

Ada beberapa cara untuk menginstall Jupyter, tapi menurut saya yang paling praktis adalah menggunakan pip. Cukup jalankan perintah ini di Terminal (Mac/Linux) atau Command Prompt (Windows):

```
pip3 install jupyter
```

Alternatifnya, kalau Anda mengalami kesulitan menginstall dengan cara di atas, mungkin bisa dicoba cara ini:

```
python3 -m pip install --upgrade pip
python3 -m pip install jupyter
```

Setelah terinstall, maka siap dijalankan.

Referensi : https://indoml.com/2019/09/29/pengenalan-dan-panduan-jupyter-notebook-untuk-pemula/

# MODUL I

# SOLUSI PERSAMAAN NONLINEAR

Tujuan :

1. Dapat menghitung akar persamaan nonlinear dengan metode Biseksi, metode Newton Raphson dan metode Secant
2. Mencari besarnya kesalahan dari suatu perhitungan akar persamaan nonlinear dengan metode Biseksi, metode Newton Raphson, dan metode Secant

Petunjuk Praktikum :

1. Buatlah laporan praktikum. Adapun isi laporan meliputi :
2. Dasar teori untuk menentukan penyelesaian model regresi secara numerik
3. Program dan Pembahasan Program

Pendahuluan

Pencarian akar(penyelesaian) suatu persamaan non-linear, $y = f(x)$, adalah mencari suatu harga $x^*$ , yang apabila disubstitusikan ke dalam persamaan itu, akan memberikan harga fungsi nol. Secara matematika

$$f(x^*) = 0$$

Sebagai contoh, akar dari persamaan $f(x) = x - e^{1/x}$ adalah 1.763223 karena, apabila harga $x^* = 1.763223$ disubstitusikan ke dalam persamaan itu, harga fungsi itu menjadi nol.

Metode-metode numeric yang banyak digunakan untuk menyelesaikan persamaan non-linear adalah metode biseksi, metode regula falsi, secant, newton –Raphson dan titik tetap. Tiga metode akan dibicarakan di sini adalah metode biseksi , metode Newton-Raphson dan metode Secant.

## A. Metode Biseksi

Dalam metode Biseksi, interval yang mengandung akar dibagi menjadi dua secara berurutan hinggga ukuran interval mengecil dan akhirnya mencapai harga toleransi kesalahan yang diinginkan. Dalam interval [a,b] terdapat sebuah akar (yang akan dicari), apabila dipenuhi :

$$f(a) * f(b) \leq 0$$

Algoritma :

Masukan :

Batas kiri dan kanan interval, $a$ dan $b$

Toleransi $tol$, Maksimum iterasi $maxit$

Fungsi, dinyatakan sebagai$fx(x)$

Keluaran :

Akar pendekatan, $x_m$

Proses :

1. iter=0
2. $x_l = a, x_r = b$
3. Jika $fx(x_l) * fx(x_m) < 0$, kerjakan langkah 4-langkah 7
4. $x_m = 0.5 * (x_l + x_r)$
5. $iter = iter + 1$
6. $delta = |x_m - x_l|$
7. Selagi $delta > tol \& iter < maxit$kerjakan :

   7.1. jika $fx(x_l) * fx(x_m) < 0$, maka

   $$x_r = x_m$$

   Jika tidak, $x_l = x_m$

   7.2. baharui harga $x_m$: $x_m = 0.5 * (x_l + x_r)$
8. Akar pendekatan = $x_m$
9. Selesai

Tugas01 :

Diberikan fungsi $f(x) = e^x + x^2 - 3x - 2 = 0$terdapat sebuah akar riil dalam selang [-1.0, 1.0]. Carilah akar tersebut dengan metode Biseksi dengan toleransi kesalahan 1e-5.

Format Luarannya:

Pencarian Akar dari $f(x) = e^x + x^2 - 3x - 2 = 0$

Dengan Metode Biseksi

| iter | $x_l$ | $x_r$ | $x_m$ | $delta$ | $f(x_l)$ | $f(x_m)$ | $f(x_l) * f(x_m)$ |
|---|---|---|---|---|---|---|---|
| | | | | | | | |

Akar pendekatannya : ……..

Dengan Toleransi : …….

## B. Metode NEWTON-RAPHSON

Metode Newton Raphson, dalam mencari akar suatu fungsi nonlinear $y = f(x)$memerlukan evaluasi harga fungsi dan turunannya pada sembarang titik x yang merupakan harga awal tebakan akar fungsi tersebut. Metode ini didasarkan atas perluasan deret Taylor di sekitar suatu titik. Karenanya, apabila harga awal tebakan jauh dari akar sebenarnya, konvergensi akan lambat atau mungkin tidak dicapai sama sekali.

Algoritma :

Masukan : Fungsi, dinyatakan sebagai $fx(x)$

Turunan fungsi, dinyatakan sebagai $dfx(x)$

Harga Tebakan awal $x_c$

Toleransi $eps$, maksimum iterasi $maxit$

Keluaran : Akar pendekatan, $x_{c+1}$

Proses :

1. iter=0
2. Hitung $fx(x_c)$ dan $dfx(x_c)$
3. iter=iter+1
4. $x_{c+1} = x_c - \frac{fx(x_c)}{dfx(x_c)}$
5. $delta = |x_{c+1} - x_c|$
6. Jika$delta \leq eps$& iter>maxit maka Akar pendekatan = $x_{c+1}$, Selesai
7. $x_c = x_{c+1}$, kembali ke langkah 2

Tugas02:

Diberikan fungsi $f(x) = e^x + x^2 - 3x - 2 = 0$mempunyai akar riil dalam selang [-1.0, 1.0]. Carilah akar tersebut dengan metode Newton Raphson dengan toleransi kesalahan 1e-5.

Format Luarannya:

Pencarian Akar Fungsi $f(x) = e^x + x^2 - 3x - 2 = 0$

Dengan Metode Newton Raphson

| iter | $x_c$ | $f(x_c)$ | $delta$ |
|---|---|---|---|
| | | | |

Akar pendekatannya : ……..

Dengan Toleransi : …….

# MODUL II
# INTERPOLASI

*TUJUAN:* Mahasiswa dapat membuat program interpolasi dengan metode polinom Newton dan metode polinom Lagrange

Petunjuk Praktikum :.
Buatlah laporan praktikum. Adapun isi laporan meliputi :

a. Dasar teori untuk menentukan penyelesaian interpolasi Newton dan interpolasi Lagrange tersebut di atas
b. Program dan Pembahasan Program
c. Pembahasan hasil/keluaran

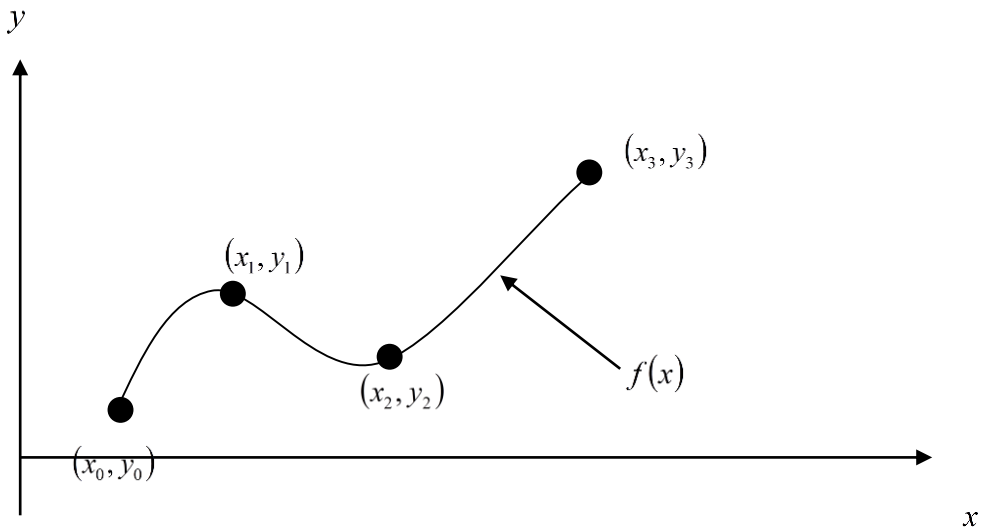
## Pendahuluan

Diketahui pasangan data $(x_0, y_0)$, $(x_1, y_1)$, ....., $(x_{n-1}, y_{n-1})$, $(x_n, y_n)$. Bagaimana mencari $y$ untuk nilai x lain yang dikehendaki? Fungsi kontinu $f(x)$ direpresentasikan $n+1$ data (Gambar 1).

Interpolasi Polynomial meliputi pencarian polynomial derajad $n$ yang melalui $n+1$ titik. Metode yang sering dipakai adalah metode interpolasi Newton dann metode interpolasi Lagrange.

## Metode Interpolasi Newton

Ilustrasi gambar 1.




**Gambar 1** Interpolasi dari data diskrit.

## Bentuk Umum metode polinom Newton

Diberikan $n+1$ titik data, $(x_0, y_0), (x_1, y_1), ....., (x_{n-1}, y_{n-1}), (x_n, y_n)$, jarak titik x dengan selang yang sama, sebagai

$$P_n(x) = b_0 + b_1(x - x_0) + .... + b_n(x - x_0)(x - x_1)...(x - x_{n-1})$$

Atau dapat ditulis sebagai

$$P_n(x) = P(x_0 + hu) = y_0 + u\Delta^1 y_0 + \frac{u(u-1)}{2!}\Delta^2 y_0 + .... + \frac{u(u-1)...(u-n+1)}{n!}\Delta^n y_0$$

Dengan

$$\frac{x - x_0}{h} = u$$

Algortima Interpolasi Newton ke depan :

1. Baca data masukan pasangan x dan y
2. Berikan nilai x yang dicari y nya.
3. Hitung h
4. Lakukan inisialisasi :
   Sum = y0
   R = (x-x0)/h
5. Lakukan iterasi proses berikut untuk i=1 sampai n-1
   a. Product =1
   b. Untuk j=0 sampai i-1 lakukan perhitungan
   Product = product*(u-j)/(j+1)
   c. Lakukan penjumlahan
   Sum = sum +product * $\Delta^i y_0$
6. Kembalikan nilai sum sebagai hassil perhitungan

## Bentuk Umum metode polinom Lagrange

Diberikan $n+1$ titik data, $(x_0, y_0), (x_1, y_1), ....., (x_{n-1}, y_{n-1}), (x_n, y_n)$, jarak titik x dengan selang yang tidak sama. Dengan $f(x_i) = y_i$.
Polinom Lagrange diberikan oleh

$$P_n(x) = \sum_{i=0}^{n} L_i(x) f(x_i)$$

dengan $n$ dalam $f_n(x)$ merupakan derajad order $n$ yang mengaproksimasi fungsi $y = f(x)$ diberikan $n+1$ titik data sebagai. $(x_0, y_0), (x_1, y_1), ....., (x_{n-1}, y_{n-1}), (x_n, y_n)$, dan

$$L_i(x) = \prod_{\substack{j=0 \\ j \neq i}}^{n} \frac{x - x_j}{x_i - x_j}$$

Algortima Interpolasi Lagrange :

1. Baca data masukan pasangan x dan y
2. Lakukan inisialisasi :
   Sum = $y_0$
   R = (x-$x_0$)/h
3. Lakukan iterasi proses berikut untuk i=1 sampai n-1
   a. Product =$y_i$
   b. Untuk j=0 sampai i-1 lakukan perhitungan
      Product = Product*(x-$x_j$)/($x_i - x_j$)
   c. Lakukan penjumlahan
   Sum = sum +product * $\Delta^i y_0$
4. Kembalikan nilai sum sebagai hasil perhitungan

**Tugas 04a. Gunakan Program Interpolasi Newton untuk menghitung x = 2.5**

| **x** | **0.0** | **1.0** | **2.0** | **3.0** | **4.0** |
|---|---|---|---|---|---|
| **f(x)** | **1.0000** | **0.5403** | **-0.4161** | **-0.9900** | **-0.6536** |

**Tugas 04b. Gunakan Program Interpolasi Lagrange untuk menghitung x = 323.5**

| **x** | **321.0** | **322.8** | **324.2** | **325.0** |
|---|---|---|---|---|
| **f(x)** | **2.50651** | **2.50893** | **2.51081** | **2.51188** |
| | | | | |

# MODUL III

# PENCOCOKAN KURVA

Tujuan :

1. Dapat menentukan penyelesaian model regresi secara Numerik
2. Mencari besarnya kesalahan dari suatu perhitungan solusi model regresi secara numerik

Petunjuk Praktikum :.

1. Buatlah program Pencocokan kurva
2. Buatlah laporan praktikum. Adapun isi laporan meliputi :
    a. Dasar teori untuk menentukan penyelesaian model regresi secara numerik
    b. Program dan Pembahasan Program
    c. Pembahasan hasil/keluaran

## PENDAHULUAN

Dalam penerapan model regresi, satu tujuan adalah untuk mendapatkan persamaan $y=f(x)$ yang paling menggambarkan *n* titik-titik data respon $(x_1, y_1), (x_2, y_2), ....., (x_n, y_n)$. Akibatnya, kita dihadapkan dengan jawaban dua pertanyaan dasar.

1. Apakah model y = f (x) menggambarkan data secara memadai, yaitu apakah ada kecocokan yang memadai?

2. Seberapa baik model memprediksi variabel respons (prediktabilitas)?

Untuk menjawab pertanyaan di atas, mari kita mulai dari pemeriksaan beberapa ukuran ketidaksesuaian antara seluruh data dan beberapa kecenderungan sentral kunci. Lihatlah dua persamaan yang diberikan di bawah ini.

$$S_r = \sum_{i=1}^{n} \left(y_i - \hat{y}_i\right)^2 = \sum_{i=1}^{n} \left(y_i - f(x_i)\right)^2 \qquad (1)$$

$$S_t = \sum_{i=1}^{n} \left(y_i - \bar{y}\right)^2 \qquad (2)$$

dimana $S_r$ adalah jumlahan kuadrat dari residu (residu adalah perbedaan antara nilai observasi, $y_i$ dan nilai prediksi, $\hat{y}_i$), dan $S_t$ adalah jumlahan kuadrat dari perbedaan antara nilai obervasi dan nilai rata-rata.

**METODE KUADRAT TERKECIL**

Kami ingin menyesuaikan kurva "terbaik" ke kumpulan titik data yang berpasangan : $(x_1,Y_1)$, $(x_2,Y_2)$, …,$(x_N,Y_N)$. ekspresi matematika untuk nilai terhitung adalah :

$$y_i = a_1 f_1(x_i) + a_2 f_2(x_i) + \ldots + a_n f_n(x_i) \qquad (5.2\text{-}1)$$

dimana $y_i$ adalah nilai terhitung yang mengaproksimasi nilai eksperimental $Y_i$. Ekspresi di atas adalah model kuadrat terkecil linear umum dengan parameter tidak diketahui $a_1$, $a_2$, …, $a_n$ adalah kombinasi linear dari fungsi yang diketahui $f_1(x_i)$, $f_2(x_i)$, …, $f_n(x_i)$. kesalahan model , atau residu, $e_i$ dapat disajikan sebagai

$$e_i = Y_i - y_i = Y_i - [a_1 f_1(x_i) + a_2 f_2(x_i) + \ldots + a_n f_n(x_i)] \qquad (5.2\text{-}2)$$

dimana $e_i$ adalah selisih antara nilai terukur $Y_i$ dan nilai pendekatan $y_i$ sebagai prediksi persamaan model.

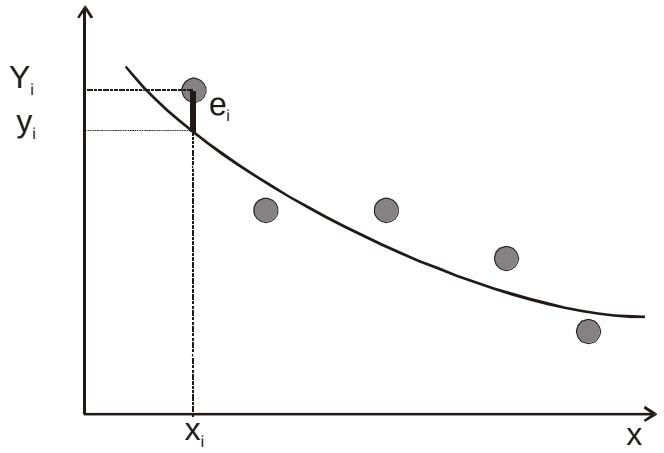



**Figure 5.2-1**. Relasi antara persamaan model dan data

Kami ingin mencari nilai-nilai untuk parameter $a_1$ ke $a_n$ untuk memberikan kecocokan "terbaik" untuk semua data. Analisis regresi digunakan untuk menentukan konstanta dalam hubungan antar fungsi. Kriteria kuadrat-terkecil mensyaratkan bahwa S didefinisikan oleh Persamaan. (5.2-3) menjadi minimum

$$S = e_1^2 + e_2^2 + ..... + e_N^2 = \sum_{i=1}^{N} e_i^2 \qquad (5.2\text{-}3)$$

atau

$$S = \sum_{i=1}^{N} \{ Y_i - [a_1 f_1(x_i) + a_2 f_2(x_i) + \ldots + a_n f_n(x_i)]\}^2 \qquad (5.2\text{-}4)$$

Setting the derivative of this sum with respect to each coefficient equal to zero will result in a minimum for the sum. Thus the coefficients $a_1$, $a_2$, and $a_3$ must satisfy the conditions

Mengatur turunan dari jumlah ini dengan memperhatikan setiap koefisien sama dengan nol akan menghasilkan minimum untuk penjumlahan. Jadi koefisien $a_1$, $a_2$, dan $a_3$ harus memenuhi persyaratan

$$\frac{\partial S}{\partial a_1} = \sum_{i=1}^{N} \{-2\}\{ Y_i - [a_1 f_1(x_i) + a_2 f_2(x_i) + \ldots + a_n f_n(x_i)]\}( - f_1(x_i)) = 0 \qquad (5.2\text{-}5.a)$$

$$\frac{\partial S}{\partial a_2} = \sum_{i=1}^{N} \{-2\}\{ Y_i - [a_1 f_1(x_i) + a_2 f_2(x_i) + \ldots + a_n f_n(x_i)]\}( - f_2(x_i)) = 0 \qquad (5.2\text{-}5.b)$$

$$\ldots = \ldots$$

$$\frac{\partial S}{\partial a_n} = \sum_{i=1}^{N} \{-2\}\{ Y_i - [a_1 f_1(x_i) + a_2 f_2(x_i) + \ldots + a_n f_n(x_i)]\}( - f_n(x_i)) = 0 \qquad (5.2\text{-}5.n)$$

Kami dapat membagi persamaan (5.2-5.a – 5.2-5.n) dengan (–2) dan melakukan pengaturan ulang untuk mendapatkan sekumpulan berikut

$$a_1 \sum_{i=1}^{N} f_1(x_i)\, f_1(x_i) + a_2 \sum_{i=1}^{N} f_1(x_i)\, f_2(x_i) + \ldots + a_n \sum_{i=1}^{N} f_1(x_i) f_n(x_i) = \sum_{i=1}^{N} f_1(x_i)\, Y_i \qquad (5.2\text{-}6.a)$$

$$a_1 \sum_{i=1}^{N} f_2(x_i)\, f_1(x_i) + a_2 \sum_{i=1}^{N} f_2(x_i)\, f_2(x_i) + \ldots + a_n \sum_{i=1}^{N} f_2(x_i) f_n(x_i) = \sum_{i=1}^{N} f_2(x_i)\, Y_i \qquad (5.2\text{-}6.b)$$

$$\ldots \qquad \ldots \qquad \ldots \quad = \ldots$$

$$a_1 \sum_{i=1}^{N} f_n(x_i)\, f_1(x_i) + a_2 \sum_{i=1}^{N} f_n(x_i)\, f_2(x_i) + \ldots + a_n \sum_{i=1}^{N} f_n(x_i) f_n(x_i) = \sum_{i=1}^{N} f_n(x_i)\, Y_i \qquad (5.2\text{-}6.n)$$

sistem dapat diekspresikan dalam notasi matriks

$$\mathbf{P}.\mathbf{a} = \mathbf{B} \qquad (5.2\text{-}7.a)$$

atau

$$\begin{bmatrix} \sum_{i=1}^{N} f_1 f_1 & \sum_{i=1}^{N} f_1 f_2 & \cdots & \sum_{i=1}^{N} f_1 f_n \\ \vdots & \vdots & \vdots & \vdots \\ \sum_{i=1}^{N} f_2 f_1 & \sum_{i=1}^{N} f_2 f_2 & & \sum_{i=1}^{N} f_2 f_n \\ \sum_{i=1}^{N} f_n f_1 & \sum_{i=1}^{N} f_n f_2 & & \sum_{i=1}^{N} f_n f_n \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{N} f_1 Y_i \\ \sum_{i=1}^{N} f_2 Y_i \\ \vdots \\ \sum_{i=1}^{N} f_n Y_i \end{bmatrix} \qquad (5.2\text{-}7.b)$$

Vektor kolom **a** dapat diselesaikan dengan mudah menggunakan kemampuan matriks dari pemrograman tertentu. Misalnya dengan matlab : **a = P\B** (5.1-6)

**ALGORITMA**

Banyak system riil dalam berbagai bidang yang dapat dimodelkan dalam bentuk system persamaan linear, dengan melibatkan banyak variabel yang tidak mungkin diselesaikan secara manual. Untuk menangani system seperti itu, terdapat beberapa metode penyelesaian system persamaan linear tersebut secara efisien dan akurat. Dalam modul ini disajikan metode yang pupuleryaitu : metode eliminasi Gauss. System n persamaan dan n variabel dituliskan sebagai berikut :

$$a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \ldots + a_{1n}x_n = b_1$$
$$a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + \ldots + a_{2n}x_n = b_2$$
$$\vdots \qquad\qquad \vdots$$
$$a_{n1}x_1 + a_{n2}x_2 + a_{n3}x_3 + \ldots + a_{nn}x_n = b_n$$

Atau $Ax = b$

A. Metode eliminasi Gauss

Dalam eliminasi Gauss, system persamaan ditransformasikan ke dalam system ekivalen dalam bentuk segitiga. Jadi matriks koefisien dikonversikan ke matriks segitiga, bentuk baru ini mudah disubstitusikan balik.

Algoritma :

MASUKAN

Dimensi matriks $n$

Matriks koefisien $a$

Larik kanan $b$

KELUARAN : larik penyelesaian $x$

1. Untuk i=1 sampai n
   - 1.1 jika $a_{i,j} = 0$
     - 1.1.1 cari nilai $a_{i+1,j} <> 0$ untuk $i + 1 \leq n$
     - 1.1.2 jika tak ditemukan $a_{i+1,j} <> 0$, keluar, tidak ada solusi
     - 1.1.3 untuk j=i sampai n, tukas $a_{i,j}$ dengan $a_{i+1,j}$
   - 1.2 untuk j=i+1 sampai n
     - 1.2.1 $m_j = a_{j,i}/a_{i,i}$
     - 1.2.2 Untuk $k = i$ sampai n

       $$a_{ij,k} = a_{j,k} - m_j * a_{i,k}$$

     - 1.2.3 $y_j = y_j - m_j * y_i$
2. $x_n = b_n/a_{n,n}$
3. $untuk\ i = n - 1$ sampai i=1
   - 3.1 $ax = 0.0$
   - 3.2 Untuk $j = i + 1$ sampai n

     $ax = ax + a_{i,j} * x_j$
   - 3.3 $x_i = (y_i - ax)/a_{i,i}$
4. selesai

**LATIHAN**

Cocokkan polinomial orde kedua ke data berikut

| $x_i$ | 0.05 | 0.15 | 0.46 | 0.70 | 0.82 | 1.17 |
|---|---|---|---|---|---|---|
| $Y_i$ | 0.956 | 0.832 | 0.571 | 0.378 | 0.306 | 0.104 |

Persamaan model polinomial orde dua $y = a_1 + a_2 x + a_3 x^2$
Misalkan $f_1(x_i) = 1$, $f_2(x_i) = x_i$ and $f_3(t_i) = x_i^2$, $\mathbf{a} = [a_1 \ a_2 \ a_3]^T$, $n = 3$, $N = 6$
Tentukan solusi model regresi polinomial orde dua tersebut!

## MODUL IV

## INTEGRASI NUMERIK

Tujuan :

1. Dapat menentukan penyelesaian Integrasi Numerik dengan metode Trapesium, dan Metode Simpson 1/3
2. Mencari besarnya kesalahan dari suatu perhitungan solusi Integrasi Numerik dengan Dengan metode Trapesium, dan Metode Simpson 1/3.

Petunjuk Praktikum :

1. Buatlah program integrasi Numerik.
2. Buatlah laporan praktikum. Adapun isi laporan meliputi :
    a. Dasar teori untuk menentukan penyelesaian Integrasi Numerik tersebut di atas
    b. Pembahasan Program
    c. Pembahasan hasil/keluaran

Dalam kalkulus dasar kita belajar cara mengevaluasi integral bermacam-macam fungsi dan kita mengenal teknik-teknik integral. Sayangnya tidak semua fungsi dapat dengan mudah diintegrasikan secara analitik. Dengan bantuan computer, kita dapat mengatasi kesulitan itu dengan memanfaatkan metode-metode numeric yang berkaitan dengan integrasi.

Integrasi numeric dikenal juga sebagai kuadratur; persoalan integrasi numeric ialah menghitung secara numeric integral tertentu

$$I = \int_a^b f(x)dx$$

Yang dalm hal ini $a$ dan $b$ adalah batas-batas integral, $f$ adalah fungsi yang dapat diberikan secara eksplisit dalam bentuk persamaan ataupun secara empiric dalam bentuk tabel nilai.Dalam praktikum modul ini membahas teknik integrasi numeric menurut Kaidah Trapesium dan Kaidah Simpson.

a. Kaidah Trapesium

Pandang sebuah pias bernbentuk trapesium dari $x = x_0$sampai $x = x_l$.

Luas satu trapezium adalah

$$\int_{x_0}^{x_l} f(x)dx = \frac{h}{2}[f(x_0) + f(x_l)]$$

Bila selang [a,b] dibagi atas n buah pias trapezium, kaidah integrasi yang diperoleh adalah

$$\int_a^b f(x)dx = \frac{h}{2}[f(x_0) + 2f(x_1) + 2f(x_2) + \cdots + 2f(x_{n-1}) + f(x_n)]$$

Algoritma:

Masukan :

fungsi yang diintegrasikan , $y = f(x)$

Batas bawah dan batas atas integral, $a, b$

Jumlah panel, $n$

Keluaran :

I = hasil integrasi fungsi.

Proses:

1. Tetapkan lebar panel

   $h = (b - a)/n$
2. Nilai awal total

   $Sum = f(a)$
3. Untuk i=1 sampai n-1 kerjakan :

   Sum = sum + 2f(a+i*h)
4. Hitung hasil integral

   I = h/2*(sum +f(b))
5. Selesai

b. Kaidah Simpson 1/3

Menurut kaidah Simpson, luas bidang di bawah kurva $f(x)$ dalam selang [a,b], dapat didekati dengan

$$\int_{x_0}^{x_l} f(x)dx = \frac{h}{3}\left[f(a) + 4f\left(\frac{a+b}{2}\right) + f(b)\right]$$

Bila selang [a,b] dibagi atas n buah pias, kaidah integrasi yang diperoleh adalah

$$\int_a^b f(x)dx = \frac{h}{3}[f(x_0) + 4f(x_1) + 2f(x_2) + \cdots + 2f(x_{n-2}) + 4f(x_{n-1}) + f(x_n)]$$

Algoritma:

Masukan :

fungsi yang diintegrasikan , $y = f(x)$

Batas bawah dan batas atas integral, $a, b$

Jumlah panel, $n$

Keluaran :

I = hasil integrasi fungsi.

Proses :

1. Tetapkan lebar panel

   $h = (b - a)/n$

   $x = a$
2. Nilai awal total

   $I = f(a) + f(b)$

   $sigma = 0$
3. Untuk i=1 sampai n-1 kerjakan :

   $x = x + h$

   If $i \; mod \; 2 = 1$ then

   $sigma = 4 * f(x)$

   else

   $sigma = 2 * f(x)$

   end

   $I = I + sigma$

4. Hitung hasil integral

   *I = h/3*(I)*

5. Selesai

Tugas 06: Dengan kedua metode tersebut,

a. Tentukan integral dari fungsi $y = \mathrm{x} * \sin(x)$ dengan interval [0.0, pi] dengan n= 128.

b. Tentukan galatnya!

Format Luaran Program adalah :

**********

Hasil Program untuk Menyelesaian Integrasi Numerik

Int $\mathrm{x} * \sin(x)$dx ; syarat x(0) =0.0 , x(1) = pi

dengan Metode Trapesium dan Simpson 1/3

Dibuat oleh :

Nama : .....

NIM : .....

Prog.Studi : .....

| N | H | Int Trapesium | Err IT % | Int Simpson 1/3 | Err IS 1/3 % |
|---|---|---|---|---|---|
| ### | #.### | ########.#### | #####.## | #########.#### | ######.## |

# MODUL V

## Menghitung Penyelesaian Sistem Persamaan Linear Dengan Metode Gauss, dan Metode Gauss Seidel

Tujuan :

1. Dapat menentukan penyelesaian sistem persamaan linear dengan metode Gauss, dan Metode Gauss Seidel
2. Mencari besarnya kesalahan dari suatu perhitungan solusi system persamaan linear dengan metode Gauss dan Metode Gauss Seidel.

Pendahuluan

Banyak system riil dalam berbagai bidang yang dapat dimodelkan dalam bentuk system persamaan linear, dengan melibatkan banyak variabel yang tidak mungkin diselesaikan secara manual. Untuk menangani system seperti itu, terdapat beberapa metode penyelesaian system persamaan linear tersebut secara efisien dan akurat. Dalam modul ini disajikan dua metode yang pupuler: metode eliminasi Gauss dan metode iterasi Gauss Seidel. System n persamaan dan n variabel dituliskan sebagai berikut :

$$a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \ldots + a_{1n}x_n = b_1$$
$$a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + \ldots + a_{2n}x_n = b_2$$
$$\vdots$$
$$a_{n1}x_1 + a_{n2}x_2 + a_{n3}x_3 + \ldots + a_{nn}x_n = b_n$$

Atau $Ax = b$

B. Metode eliminasi Gauss

Dalam eliminasi Gauss, system persamaan ditransformasikan ke dalam system ekivalen dalam bentuk segitiga. Jadi matriks koefisien dikonversikan ke matriks segitiga, bentuk baru ini mudah disubstitusikan balik.

Algoritma :

MASUKAN

Dimensi matriks $n$

Matriks koefisien $a$

Larik kanan $b$

KELUARAN : larik penyelesaian $x$

5. Untuk i=1 sampai n
   - 5.1 jika $a_{i,j} = 0$
     - 5.1.1 cari nilai $a_{i+1,j} <> 0$ untuk $i + 1 \leq n$
     - 5.1.2 jika tak ditemukan $a_{i+1,j} <> 0$, keluar, tidak ada solusi
     - 5.1.3 untuk j=I sampai n, tukas $a_{i,j}$ dengan $a_{i+1,j}$
   - 5.2 untuk j=i+1 sampai n
     - 5.2.1 $m_j = a_{j,i}/a_{i,i}$
     - 5.2.2 Untuk $k = i$ sampai n

$$a_{ij,k} = a_{j,k} - m_j * a_{i,k}$$

5.2.3 $y_j = y_j - m_j * y_i$

6. $x_n = b_n / a_{n,n}$
7. $untuk\ i = n - 1$ sampai i=1
   7.1 $ax = 0.0$
   7.2 Untuk $j = i + 1$ sampai n
   $ax = ax + a_{i,j} * x_j$
   7.3 $x_i = (y_i - ax)/a_{i,i}$
8. selesai

C. Metode Iterasi Gauss Seidel

Merupakan metode iterasi, yang berawal dengan solusi pendekatan. Solusi pendekatan inni, lantas dipakai dalam rumusan berulang untuk memberikan solusi pendekatan lainnya. Dengan memakai runus itu berulang-ulang sederet solusi dihasilkan yang akan menuju solusi eksak di bawah kondisi kondisi yang cocok.

$$x_1 = \frac{b_1 - a_{12}x_2 - a_{13}x_3 \ldots\ldots - a_{1n}x_n}{a_{11}}$$

$$x_2 = \frac{b_2 - a_{21}x_1 - a_{23}x_3 \ldots\ldots - a_{2n}x_n}{a_{22}}$$

$$\vdots$$

$$x_{n-1} = \frac{b_{n-1} - a_{n-1,1}x_1 - a_{n-1,2}x_2 \ldots\ldots - a_{n-1,n-2}x_{n-2} - a_{n-1,n}x_n}{a_{n-1,n-1}}$$

$$x_n = \frac{b_n - a_{n1}x_1 - a_{n2}x_2 - \ldots\ldots - a_{n,n-1}x_{n-1}}{a_{nn}}$$

Sehingga

$$x_1 = \frac{c_1 - \sum_{\substack{j=1 \\ j\neq 1}}^{n} a_{1j}x_j}{a_{11}}$$

$$x_2 = \frac{c_2 - \sum_{\substack{j=1 \\ j\neq 2}}^{n} a_{2j}x_j}{a_{22}}$$

$$x_{n-1} = \frac{b_{n-1} - \sum_{\substack{j=1 \\ j\neq n-1}}^{n} a_{n-1,j}x_j}{a_{n-1,n-1}}$$

$$x_n = \frac{b_n - \sum_{\substack{j=1 \\ j\neq n}}^{n} a_{nj}x_j}{a_{nn}}$$

Selanjutnya utnuk tiap-tiap baris $i$,

$$x_i = \frac{b_i - \sum_{\substack{j=1 \\ j \neq i}}^{n} a_{ij} x_j}{a_{ii}}, i = 1,2,\ldots,n.$$

Algoritma :

1. Definisikan matriks $a_{nn}, b_n, x_n, x0_n$
2. Untuk $i = 1$ sampai n
   2.1 $ax = a_{i,i}, y_n = y_n/ax$
   2.2 Untuk $j = 1$ sampai n
   $a_{i,j} = a_{i,j}/ax$
3. Untuk $i = 1$ sampai n
   $x_i = 1$
4. Untuk $i = 1$sampai n
   4.a sum=0
   4.b. untuk $j = 1$sampai n
   Jika $(j <> i)$ maka $sum = sum + x_j * a_{i,j}$
   4.c. $x0_i = x_i$
   $x_i = y_i - sum$
5. Untuk $i = 1$sampai n
   $dx = \max(abs(x_i - x0_i))$
6. Jika dx>EPS loncat ke langkah 4
7. Solusi adalah matriks x
8. Selesai

Petunjuk Praktikum :

1. Buatlah program serta cetak keluarannya.
2. Buatlah laporan praktikum. Adapun isi laporan meliputi :
   a. Dasar teori untuk menentukan solusi system persamaan linear metode tersebut di atas
   b. Pembahasan Program
   c. Pembahasan hasil/keluaran

Tugas 03:

Diberikan system persamaan linear berikut :

$$3x_1 - 0.1x_2 - 0.2x_3 = 7.85$$
$$0.1x_1 + 7x_2 - 0.3x_3 = -19.3$$
$$0.3x_1 - 0.2x_2 + 10x_3 = 71.4$$

Tentukan solusi dengan kedua metode tersebut.

# MODUL VI

# OPTIMASI TAK BERKENDALA

Tujuan : Dapat menentukan penyelesaian Optimasi tak berkendala multivariabel secara Numerik dengan metode Newton

Petunjuk Praktikum :.
Buatlah laporan praktikum. Adapun isi laporan meliputi :

a. Dasar teori untuk menentukan penyelesaian optimasi tak berkendala multivariabel
b. Program dan Pembahasan Program
c. Pembahasan hasil/keluaran

## PENDAHULUAN

Dalam kehidupan sehari-hari baik disadari maupun tidak, sebenarnya orang selalu melakukan optimasi untuk memenuhi kebutuhannya. Tetapi optimasi yang dilakukan masyarakat awam lebih banyak dilandasi oleh intuisi daripada teori optimasi. Dalam masalah optimasi terdapat dua bentuk optimasi yaitu fungsi optimasi tanpa kendala dan dengan kendala. Banyak aplikasi dari pemodelan matematika yang berbentuk linear atau nonlinear dalam optimasi fungsi yang mensyaratkan beberapa kendala ataupun tanpa kendala untuk diperoleh suatu solusi optimal. Permasalahan ini berfungsi untuk mencari nilai ektrim dari fungsi tujuan. Persoalan dengan model nonlinear yang tidak disertai kendala dinamakan optimasi nonlinear tanpa kendala. Optimasi merupakan masalah yang berhubungan dengan keputusan yang terbaik, maksimum, minimum dan memberikan cara penentuan solusi yang memuaskan. Terdapat beberapa metode dalam mencari optimasi multivariable tanpa kendala. Diantaranya metode Univariate, Steepest Descent, dan Newton. Dalam metode Univariate membutuhkan iterasi yang banyak dalam mencari nilai optimasinya. Begitu pula dengan metode Steepest Descent yang nilai iterasinya tergantung pada pemilihan nilai , semaki kecil semakin banyak iterasi yang dibutuhkan. Dari berbagai masalah itu metode Newton memberikan alternatif yang lebih bagus. Dalam makalah ini akan dijelaskan tentang metode Newton beserta contoh kasusnya.

## METODE NEWTON

Fungsi Multivariable tanpa Kendala Cara analitis yang diterapkan pada permaslahan optimasi satu variable dapat diterapkan pada permasalahan multivariable. Dalam makalah ini akan dibahas masalah optimasi untuk fungsi lebih dari satu variabel. Seperti masalah minimisasi pada satu variabel; $\min z = f(x_1, x_2, \dots, x_n)$
Sedangkan bentuk umum dari fungsi multivariable:

$$\min z = f(x) = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \in R^n$$

Merupakan fungsi dari n variabel $f: R^n \to R$ tranpose dari matriks $x$ ditulis dengan

$$x^t = [x_1, x_2, \dots, x_n]$$

Penggunaan turunan sebagai syarat perlu dan cukup: $\partial f(x) = 0$ dan $\partial^2 f(x)$ definit positif.

**Metode Newton**

Dalam analisis numerik, metode Newton juga dikenal dengan metode Newton Rapshon yang mendapat nama dari Isaac Newton (1669) dan Joseph Rapshon (1690), merupakan suatu metode yang cukup dikenal untuk mencari pendekatan terhadap akar fungsi riil. Metode Newton Rapshon sering konvergen dengan cepat, terutama bila dimulai cukup dekat dengan akar yang diinginkan. Namun, bila iterasi dimulai jauh dari akar yang dicari, metode ini dapat melesat tanpa peringatan. Implementasi metode ini biasanya mendeteksi dan mengatasi kegagalan konvergensi.

Gagasan awal metode Newton Rapshon adalah metode yang digunakan untuk mencari akar dari sebuah fungsi riil. Metode ini dimulai dengan memperkirakan satu titik awal dan mendekatinya dengan memperlihatkan slope atau gradient pada titik tersebut. Diharapkan dari titik awal tersebut akan diperoleh pendekatan terhadap akar fungsi yang dimaksud.

Diberikan fungsi $f: R^n \to R$

$\nabla^2 f(x)$ definit positif untuk semua $x$ , jadi $f$ juga konveks. Misal $x_k$ suatu vektor tertentu (merupakan estimasi untuk ($x^*$). Untuk menemukan estimasi baru $x_{k+1}$ sebagai berikut. Perhatikan pendekatan kuadratis lewat ekspansi taylor dari fungsi $f$ di titik $x_k$.

$P(x) = f(x_k) + (x - x_k).\nabla f(x_k) + \frac{1}{2} \nabla^2 f(x_k)(x - x_k)$

$\nabla P(x) = \nabla f(x_k) + \nabla^2 f(x_k)(x - x_k) \cong 0$

$\nabla^2 P(x) = \nabla^2 f(x_k)$

Sehingga $\nabla^2 P(x)$ definit positif dan $P(x)$ adalah konveks. Estimasi baru untuk $x^*$ dapat dengan mencari minimum dari $P(x)$ diperoleh

**ALGORITMA**

Algoritma Input

Input $f: R^n \to R$; $\nabla f$ , $\nabla^2(f)$ , definit positif, estimasi nilai awal $x_0 \in R^n$, toleransi ($\alpha$), $k = 0$

Step : 1. $x_{k+1} = x_k - [\nabla^2 f(x_k)]^{-1} . \nabla f(x_k)$

2. jika $\|\nabla\text{^}2\, f(x\_k)\| < \alpha$, STOP. $x^* = x_{k+1}$

3. $k = k + 1$, ulangi langkah 1.

**LATIHAN**

1. Diketahui: $A = \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix}, b = \begin{bmatrix} -1 \\ -1 \end{bmatrix}, c = 0, x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$

Carilah minimum dari fungsi kuadratik $f(x)$ yang diperoleh dari matriks $A, b, x,$ dan konstanta 0 di atas, dengan toleransi $\alpha$ = 0.01 dan dimulai dari titik awal $x_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$.

Atau dapat ditulis

Min $f(x_1, x_2) = \frac{1}{2}x^t Ax + b^t x + c$

$= \frac{1}{2}x_1^2 + x_2^2 + x_1 x_2 - x_1 - x_2$

Tentukan nilai minimum dari $f(x_1, x_2)$

2. Tentukan Min $f(x_1, x_2) = (x_1 - 0.5)^2(x_1 + 1)^2 + (x_2 + 1)^2(x_2 - 1)^2$, dengan toleransi $\alpha$ = 0.01 dan dimulai dari titik awal $x_0 = \begin{bmatrix} 0 \\ 0.5 \end{bmatrix}$

**000 SELAMAT BEKERJA 000**