



Modul Praktikum FISIKA KOMPUTASI

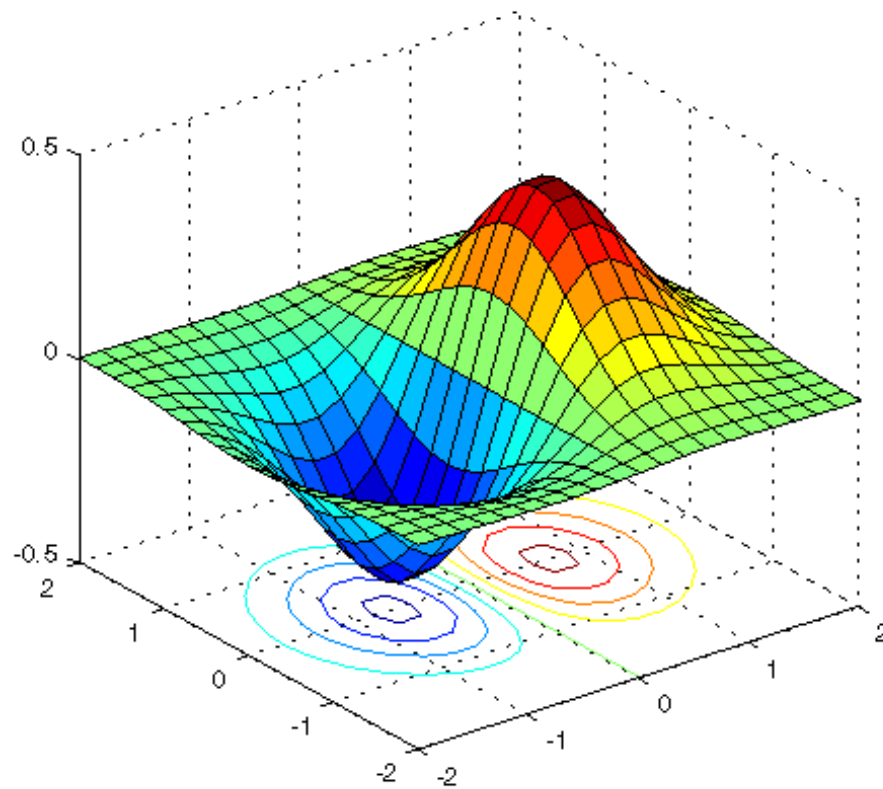
Rustan



Program Studi Fisika
Fakultas Sains dan Teknologi
Universitas Jambi
Tahun 2019

Modul Praktikum
FISIKA KOMPUTASI

Rustan, M.Si



Program Studi Fisika
Fakultas Sains dan Teknologi
Universitas Jambi

LEMBAR PENGESAHAN

MODUL PRAKTIKUM FISIKA KOMPUTASI

Disusun Oleh :
Rustan, S.Pd., M.Si
NIP. 199009172019031011

Modul ini disusun sebagai pedoman dan acuan dalam
Pelaksanaan Praktikum Fisika Komputasi
Program Studi Fisika
Fakultas Sains dan Teknologi
Universitas Jambi

Dinyatakan dapat Digunakan

Disahkan pada Desember 2019

Wakil Dekan BAKSI
Fakultas Sains dan Teknologi
Universitas Jambi

Disetujui:
Sekretaris Program Studi Fisika

Dr. Tedjo Sukmono, S.Si., M.Si
NIP. 197207052000031003

Nurhidayah, S.Pd., M.Sc
NIP. 198708012015042002

KATA PENGANTAR

Syukur kami ucapkan kehadiran Tuhan Yang Maha Kuasa atas selesainya penyusunan modul praktikum fisika komputasi ini. Modul ini diharapkan dapat membantu mahasiswa dalam pelaksanaan praktikum ataupun dalam memahami materi kuliah dalam kelas.

Materi yang disajikan dalam modul ini merupakan kumpulan dari berbagai macam sumber seperti jurnal, modul praktikum beberapa universitas lain, dan buku referensi. Dengan demikian penjelasan ataupun pemaparan yang diberikan dalam modul ini tidak begitu detail sehingga mahasiswa diharapkan tetap mencari dan membaca buku ataupun referensi-referensi lain untuk menambah pemahaman mahasiswa.

Kekurangan pada modul praktikum ini masih ada dan modul ini masih jauh dari kata sempurna sehingga penyempurnaan modul ini berupa perbaikan, penambahan materi atau penjelasan akan terus dilakukan. Kritik dan saran sangat kami harapkan agar modul praktikum ini menjadi lebih baik kedepannya.

Selamat menggunakan modul praktikum ini dan semoga bermanfaat bagi yang membaca.

Jambi, Desember 2019

Penulis

DAFTAR ISI

LEMBAR PENGESAHAN.....	i
KATA PENGANTAR.....	ii
DAFTAR ISI	iii
Modul 1. Pengenalan Program Matlab.....	1
Modul 2. Penyelesaian Turunan menggunakan MATLAB.....	11
Modul 3. Penyelesaian Integral dengan Metode Trapezoida	14
Modul 4. Penyelesaian Integral dengan Metode Simpson	16
Modul 5. Diferensiasi Numerik dengan Metode Euler	18
Modul 6. Diferensiasi Numerik dengan Metode Heun	20
Modul 7. Diferensiasi Numerik dengan Metode Runge Kutta.....	23
Modul 8. Metode Beda Hingga (<i>Finite Differences</i>) untuk PDP Eleptik.....	27
Modul 9. Metode Beda Hingga (<i>Finite Differences</i>) untuk PDP Parabolik.....	32
Modul 10. Metode Beda Hingga (<i>Finite Differences</i>) untuk PDP Hiperbolik.....	36
Daftar Pustaka	38

Modul 1. Pengenalan Program Matlab

Tujuan Praktikum

1. Mahasiswa dapat memahami fungsi-fungsi dasar di program Matlab
2. Mahasiswa dapat menggunakan fungsi-fungsi dasar di Matlab untuk menyelesaikan aljabar sederhana
3. Mahasiswa dapat membuat grafik suatu fungsi di Matlab

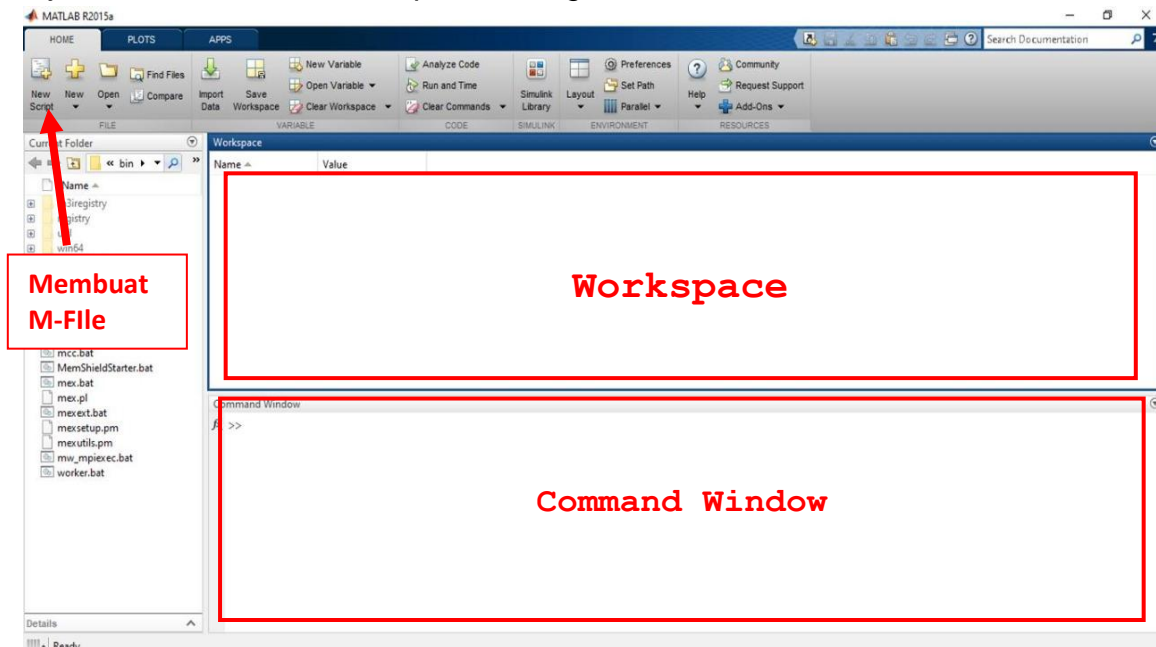
Dasar Teori

Matlab (*Matrix Laboratory*) merupakan bahasa pemrograman komputer berbasis windows dengan orientasi dasarnya adalah matriks, digunakan untuk analisis numerik, analisis aljabar linier, dan teori matriks. Saat ini dengan fitur yang semakin lengkap, Matlab juga digunakan untuk visualisasi dan analisa data, simulasi dan pemodelan, serta pengembangan aplikasi dengan memanfaatkan GUI (*Graphical User Interface*).

Agar semua kode program yang ditulis di modul ini dapat berjalan dengan baik, maka perlu dilakukan pengenalan awal terhadap program Matlab dan perintah-perintah dasar yang ada di Matlab.

Desktop Matlab

Ketika pertama kali menjalankan Matlab, maka yang tampil adalah Desktop Matlab. Di dalam dekstop Matlab, terdapat berbagai tool yang berfungsi untuk manajemen file, variabel, dan aplikasi dengan Matlab.



Gambar 1.1 Tampilan Awal Matlab 2015a

Salah satu jendela yang ada pada dekstop Matlab adalah *Command Windows* (Jendela Perintah). Dari jendela ini kita dapat memasukkan perintah Matlab. Disamping itu kita juga dapat menjalankan atau mengeksekusi program yang sudah kita buat di *editor window* dan disimpan di direktori aktif. Perintah atau eksekusi program dapat dilakukan setelah prompt atau tanda “ >> “. Sebagai contoh:

```
>> a=4;
>> b=5;
>> c=a+b;
c=
    9
```

Operator Dasar Aritmatik

Operator dasar aritmatik antara lain adalah + (tambah), - (kurang), * (kali), / (bagi), dan ^ (pangkat).

Dalam mengeksekusi operasi matematika, Matlab mengikuti aturan-aturan berikut:

- 1) Matlab mendahulukan operasi dalam kurung
- 2) Operasi yang melibatkan operator * dan / (dapat * / atau / *) bekerja dari kiri ke kanan
- 3) Operasi matematika yang melibatkan operator + dan - (dapat + - atau - +) juga bekerja dari kiri ke kanan

Sebagai contoh:

```
>> 1+2*4/3
ans =
    3.6667
>> 1+2/4*3
ans =
    2.5000
```

Untuk bentuk yang lebih jelas operasi $1+2*4/3$ dapat dituliskan sebagai:

$$\begin{aligned} 1+((2*4)/3) &= 1+8/3 \\ &= 1+ 2.667 \\ &= 3.667 \end{aligned}$$

Sedangkan operasi $1+2/4*3$ dapat dituliskan sebagai:

$$\begin{aligned} 1+2/4*3 &= 1+(2/4)*3 \\ &= 1 + 0.5 *3 \\ &= 2.5000 \end{aligned}$$

Membuat Variabel

Variabel pada Matlab berifat **case sensitif**, artinya antara huruf besar dengan huruf kecil dibedakan. Dibawah ini diberikan beberapa aturan penulisan variabel dalam Matlab:

- 1) Variable tidak boleh diawali dengan angka, misalnya 2abc, 45y, 3ok43
- 2) Variabel dapat merupakan kombinasi antara huruf dan angka, misalnya ok45, ok45ok, abc432 dsb
- 3) Variabel tidak boleh menggunakan karakter khusus Matlab, misalnya %, #, - , +, = . Misalnya %ok, net-cost, %x, @sign dan sebagainya.
- 4) Hindari memberikan nama variabel dengan nama-nama khusus yang ada di Matlab, misalnya hindari memberikan nama variabel dengan nama pi, eps, i, j, karena $pi=22/7$, $eps=2^{-54}$, i dan j memiliki harga $\sqrt{-1}$.

Perintah Tambahan

Berikut beberapa perintah tambahan yang berguna untuk pemrograman:

- 1) **clc** : menghapus layar di command window
- 2) **close all** : menghapus semua gambar yang tampil sebelumnya.
- 3) **clear** : perintah untuk menghapus data di memori Matlab
- 4) **cd** : perintah untuk mengubah direktori
- 5) **pwd** : perintah untuk mengetahui kita berada di direktori mana pada saat ini.
- 6) **dir** : perintah untuk mengetahui file apa saja yang ada di current directory
- 7) **mkdir** : perintah untuk membuat direktori dibawah current direktori
- 8) **delete** : perintah untuk menghapus file
- 9) **who** : menampilkan semua variabel saat ini.
- 10) **whos** : menampilkan semua variabel saat ini bersama dengan informasi tentang ukuran, bytes, class dll
- 11) **what** : menampilkan semua file dengan ekstensi .M (M-File)
- 12) **lookfor** : `perintah untuk mencari file dengan katakunci

Fungsi Dasar Matlab

Disamping fungsi trigonometri, fungsi-fungsi dasar juga penting. Beberapa fungsi dasar tersebut antara **abs()**, **sqrt()**, **exp()**, **log()**, **log10()**, **log2()**. Untuk lebih jelasnya, lihat tabel dibawah ini:

Tabel 1.1 Fungsi dasar Matlab

No	Nama variabel	Keterangan
1	abs()	Menyatakan harga mutlak, misal $ x $
2	sqrt()	Menyatakan akar pangkat dua, misal \sqrt{x}
3	exp()	Menyatakan harga eksponensial, misal e^x
4	log()	Menyatakan harga ln, misal $\ln(x)$
5	log10()	Menyatakan harga logaritma basis 10, misal $\log(x)$
6	log2()	Menyatakan harga logaritma basis 2, misal ${}^2\log(x)$

Konstanta Khusus Matlab

Beberapa konstanta khusus sebaiknya dihindari untuk didefinisikan kembali sebagai konstanta. Beberapa konstanta khusus tersebut antara lain:

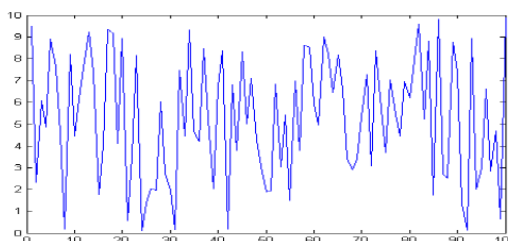
Tabel 1.2 Konstanta dalam Matlab

No	Konstanta	Keterangan
1	pi	3.14159265...
2	i	Unit imajiner, $\sqrt{-1}$
3	j	Sama dengan i
4	eps	Ketelitian relatif floating-point
5	realmin	Bilangan floating-point terkecil
6	realmax	Bilangan floating-point terbesar
7	inf	Bilangan tak hingga
8	NaN	Not-a-Number

Plot Grafik

Untuk membuat sebuah grafik garis, fungsi yang kita gunakan adalah *plot*. Sebagai contoh, misalnya kita memiliki data dalam bentuk array dan kita simpan dalam vektor *y*, maka *plot(y)* akan ditampilkan grafik elemen-elemen *y* terhadap indeks elemen-elemen tersebut.

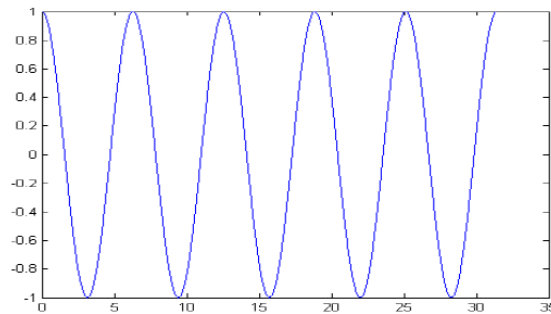
```
y=10*rand(100,1);
plot(y)
```



Gambar 1.2. Plot grafik garis suatu fungsi y

Jika kita memiliki dua buah argumen x dan y , dimana $0 \leq x \leq 10\pi$ dan $y = \cos x$ maka grafik y vs x dapat dilihat pada gambar berikut:

```
x=0:pi/200:10*pi;  
y=cos(x);  
plot(x,y)
```



Gambar 1.3 Tampilan grafik y vs x

Fungsi `plot` dapat menerima argumen bewujud karakter maupun string yang menyatakan warna, jenis garis dan penanda. Secara umum, bentuk umum

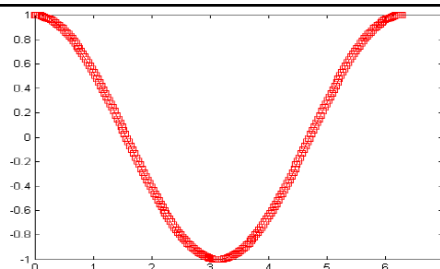
`plot(x,y,'JenisGaris_Penanda_Warna')`

Argumen `JenisGaris_Penanda_Warna` berbentuk string dan diapit oleh tanda petik tunggal.

- 1) **Jenis garis** dapat berupa garis putus-putus (dash), titik-titik (dot) dan lain-lain.
- 2) **Penanda** dapat berupa tanda bintang (*), kotak (square), bulatan (o), diamond, tanda silang (x) dan lain-lain.
- 3) **Warna** dapat berupa merah (r), kuning (y), hijau (g), cian (C), hitam (b) dan lain-lain.

Sebagai contoh perintah `plot(x,y,'-squares')` akan menampilkan grafik vs x dengan jenis grafik dash (-), penanda kotak (square) dan warna merah.

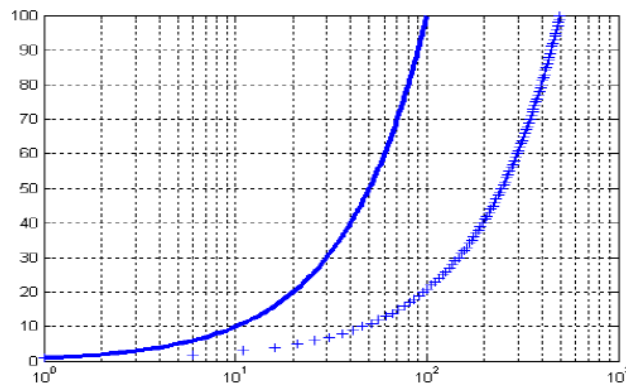
```
x=linspace(0,2*pi,200);  
y=cos(x);  
plot(x,y,'-squares')
```



Gambar 1.4 Tampilan grafik dengan style garis diikuti kotak berwarna merah

Kita dapat menambahkan grafik baru pada grafik sebelumnya dengan perintah *hold*. Perintah ini akan aktif saat di *on*-kan atau *hold on* dan tidak aktif saat diberi perintah *hold off*. Contoh:

```
clear; close all;
x=1:100;
semilogx(x, '-', 'LineWidth', 3);
hold on;
plot(1:5:500, 1:100, '+');
hold off
```

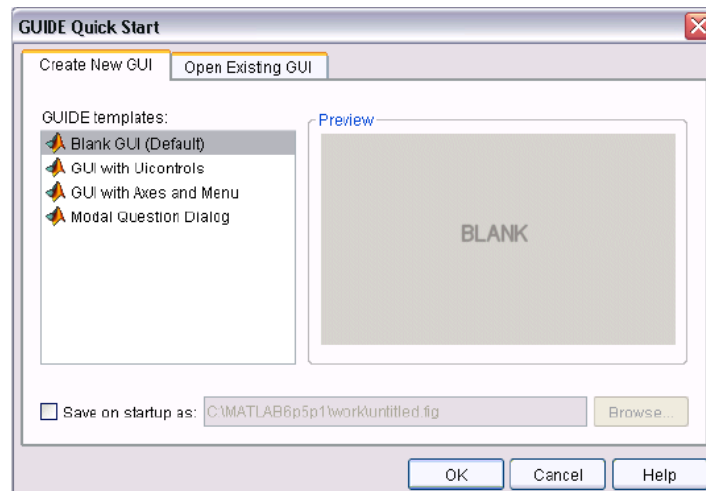


Gambar 1.5 Tampilan dua grafis dalam satu figure

Membuat GUI (*Graphical User Interface*)

Memulai GUI Matlab

- Panggil program MATLAB : Start → program → Matlab
- Melalui editor window Matlab : ketiklah >> **guide**
- Akan ditampilkan kotak dialog seperti gambar 1.6 berikut:



Gambar 1.6 Guide Quick Start

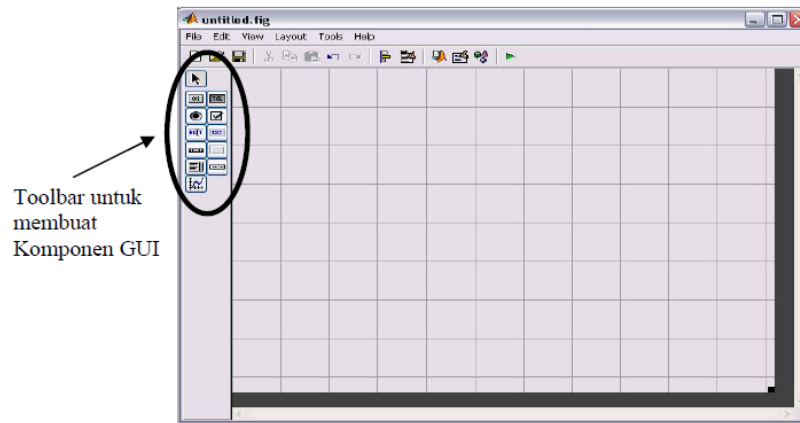
Membuat GUI menggunakan grafik

Berikut adalah contoh membuat GUI untuk menggambar grafik fungsi dengan perintah **ezplot**.

Ezplot (f) akan menggambar grafik fungsi f pada selang interval $(-2\pi, 2\pi)$.

Berikut ini adalah langkah-langkahnya:

1. Dari dialog gambar xx, pilihlah **Blank GUI (default)**, lalu klik **OK**. Akan ditampilkan figure kosong dengan nama `untitled.fig`. simpan blank gui tersebut, save as **File** → **Save As** . Beri nama **grafik.fig**



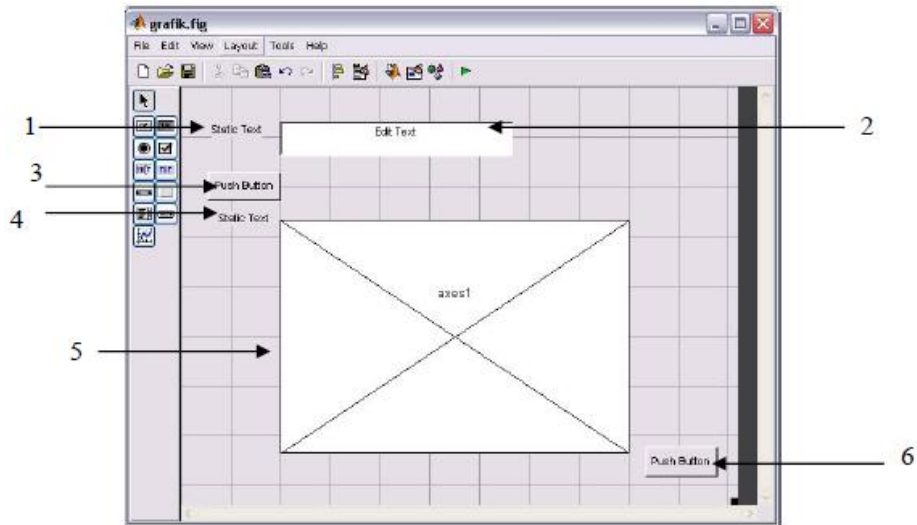
Gambar 1.7 Blank GUI

Dalam Blank GUI tersebut tambahkan komponen-komponen GUI dengan menggunakan toolbar yang tersedia di sisi kiri. Berikut adalah komponen-komponen GUI beserta fungsinya:

no	gambar	nama	fungsi
1		select	Memilih beberapa obyek secara bersama-sama
2		Push button	Tombol eksekusi, jika di-klik akan mengeksekusi sebuah perintah dan menampilkan hasilnya
3		Radio button	Untuk memilih satu pilihan dari beberapa pilihan yang disediakan
4		Edit teks	Untuk memasukkan input dan menampilkan hasil teks
5		slider	Memasukkan input berupa range angka dengan cara menggeser control pada slider
6		List box	Memilih item dari semua daftar item yang ditampilkan.
7		axes	Menampilkan grafik atau gambar
8		Toggle button	Berfungsi sama dengan Push button. Bedanya, toggle button diklik tidak akan kembali ke semula
9		Check box	Memilih satu atau lebih pilihan dari semua daftar pilihan yang ditampilkan.
10		Static Text	Membuat teks label
11		frame	Untuk membuat frame
12		Pop up menu	Membuka daftar pilihan, dan memilih pilihan

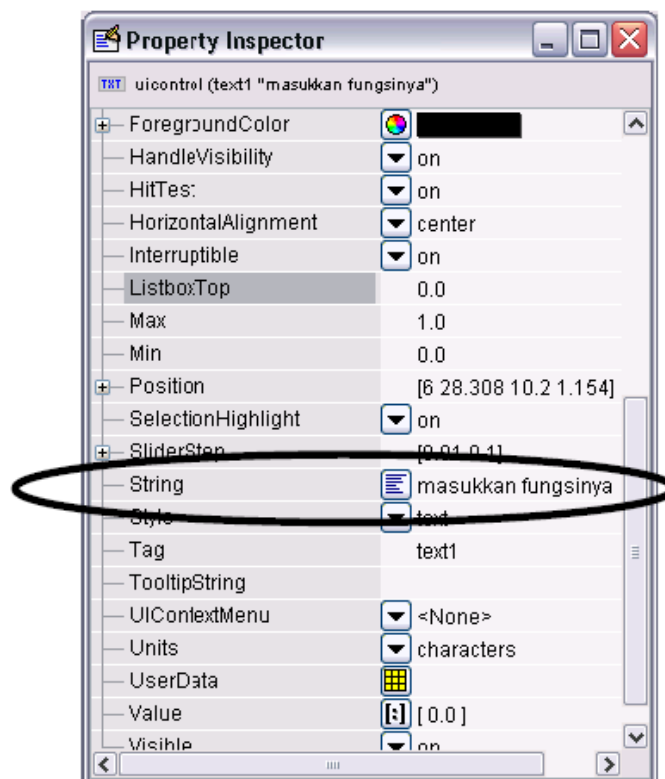
Gambar 1.8 Komponen GUI dan Fungsinya

2. Tambahkan komponen-komponen berikut pada blank GUI sehingga tampilan menjadi sebagai berikut:



Gambar 1.9 Tampilan Blank GUI

3. Klik kanan komponen **static text** nomor 1, pilih **property inspector**. Ganti isi property string 'static text' dengan 'masukkan fungsinya', seperti pada gambar:



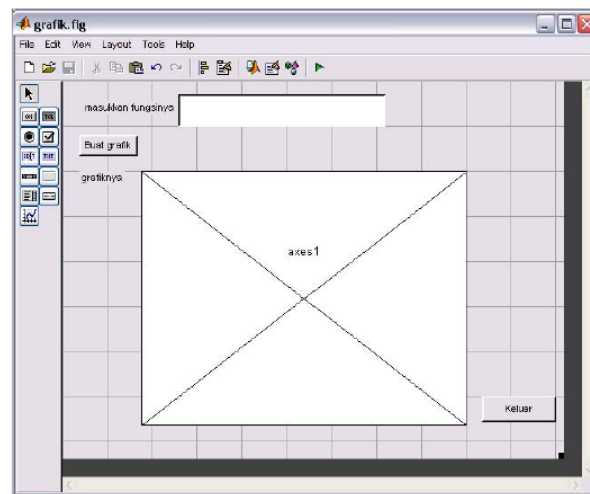
Gambar 1.10 Property Inspector

Dengan cara yang sama, klik kanan komponen-komponen lain dalam **grafik.fig**. Atur propertinya seperti dalam tabel berikut:

Tabel 1.3 Komponen GUI

No	komponen	Isi	
		Tag	String/Title/Name
1	Static Text	Text1	Masukkan fungsinya
2	Edit Text	efungsi	-(kosongkan, tekan del)
3	Push Button	Pushbutton1	Buat grafik
4	Static Text	Text2	Grafiknya
5	axes	Axes1	-
6	Push Button	tkluar	Keluar

Hasilnya sebagai berikut:



Gambar 1.11 Tampilan Akhir GUI

4. Berikutnya agar tombol buat grafik dapat berfungsi, kode harus dimasukkan ke dalam komponen tersebut. caranya, klik kanan tombol buat grafik, pilih **View Callback**

Ketikkan kode-kode seperti terlihat pada baris nomor 132 sampai 137 dalam gambar berikut:

```

125
126 % --- Executes on button press in pushbutton1.
127 function pushbutton1_Callback(hObject, eventdata, handles)
128 % hObject handle to pushbutton1 (see GCBO)
129 % eventdata reserved - to be defined in a future version of MATLAB
130 % handles structure with handles and user data (see GUIDATA)
131
132 - proyek=guidata(gcbo);
133 - f=get(proyek.efungsi,'String');
134 - axes(proyek.axes1)
135 - ezplot(f)
136 - set(proyek.axes1,'XMinorTick','on')
137 - grid on
138

```

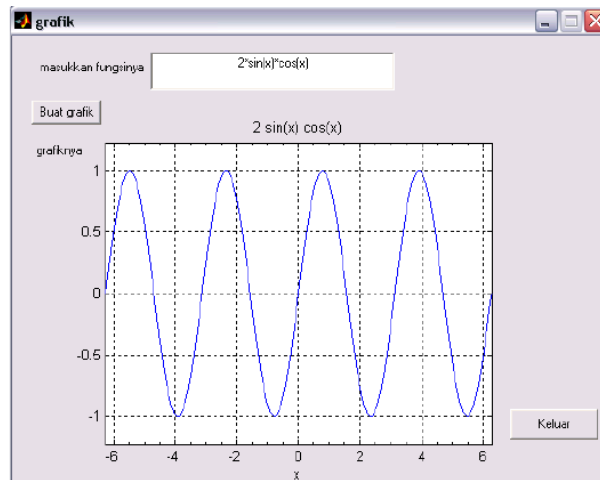
- Lakukan hal yang sama pada tombol **keluar**. Klik kanan, **View Callbacks** → **Callback**. Ketikkan kode 'close' seperti pada gambar berikut:

```

140 % --- Executes on button press in tkluar.
141 function tkluar_Callback(hObject, eventdata, handles)
142 % hObject    handle to tkluar (see GCBO)
143 % eventdata  reserved - to be defined in a future version of MATLAB
144 % handles    structure with handles and user data (see GUIDATA)
145
146
147 - close

```

- Simpan kembali grafik.fig
- Untuk menjalankan GUI yang telah dibuat, pilih menu **Tools** → **Run**.
Misalkan: isikan fungsi **2*sin(x)*cos(x)**, kemudian klik tombol **Buat Grafik**, maka akan ditampilkan grafiknya.



Tugas

- Buatlah program Matlab operasi perkalian berikut:
Panjang = 7
Lebar = 5
Tinggi = 2
Volume = Panjang * Lebar * Tinggi
- Buatlah program Matlab operasi penjumlahan berikut:
Panjang = 7
Lebar = 5
Tinggi = 2
Volume = Panjang * Lebar * Tinggi
- Buatlah program Matlab untuk menampilkan grafik dari fungsi $y = \sin(2t)\cos(2t)t$ menggunakan fungsi *plot3*.
- Manakah diantara nama-nama variabel berikut yang valid?
Luas, kel_1, 2_data, diff:3, time, time_from_start, 10_hasil_terakhir, nilai-awal.

Modul 2. Penyelesaian Turunan menggunakan MATLAB

Tujuan Praktikum

1. Mahasiswa dapat memahami berbagai perhitungan fisika yang melibatkan turunan
2. Mahasiswa dapat membuat script sederhana yang berkaitan dengan penyelesaian turunan berdasarkan rumus-rumus turunan

Dasar Teori

Misalkan suatu fungsi $y = ax^n$ dengan a dan n adalah konstanta, maka turunan fungsi tersebut adalah $\frac{dy}{dx} = a \times n \times x^{n-1}$. Contoh: $y = 3x^2$ jika diturunkan hasilnya adalah

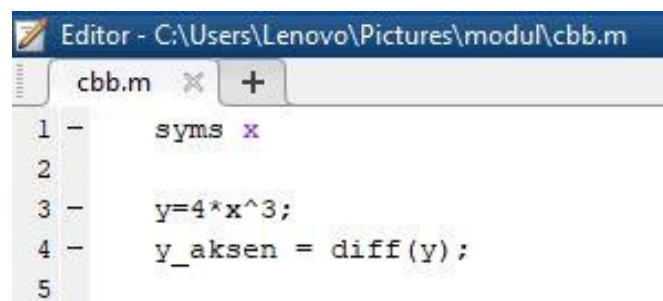
$$\frac{dy}{dx} = 3 \times 2 \times x^{2-1} = 6x.$$

Untuk mencari turunan suatu fungsi di Matlab, ada beberapa perintah yang harus dimasukkan ke dalam program agar hasilnya bisa ditampilkan, yaitu:

- Perintah **syms**. Perintah ini dituliskan di awal script program dan bertujuan untuk mendefinisikan variabel bebas suatu fungsi. Misalkan suatu fungsi yang ingin diturunkan adalah $y = 2x + 5$. Variabel bebas fungsi tersebut adalah x , sehingga di awal script dituliskan perintah **syms x**. Jika perintah **syms** tidak dituliskan, maka program tidak akan dapat menampilkan hasil turunan fungsi tersebut.
- Perintah **diff**. Merupakan perintah untuk mendefinerensialkan suatu fungsi yang bersifat numerik.

Kegiatan Praktikum

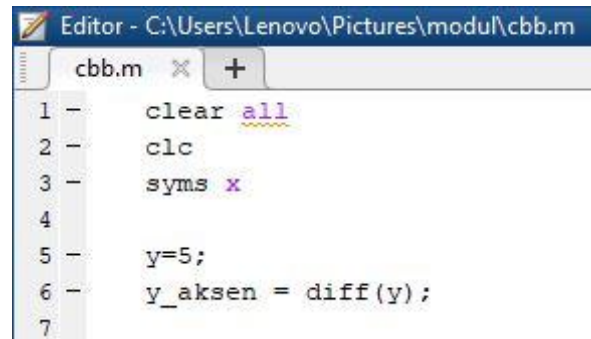
- a) $y = 4x^3$. Dengan program Matlab, dapat dicari turunannya dengan script sebagai berikut:



```
Editor - C:\Users\Lenovo\Pictures\modul\cbb.m
cbb.m x +
1 - syms x
2
3 - y=4*x^3;
4 - y_aksen = diff(y);
5
```

Gambar 2.1 Script turunan fungsi $4x^3$

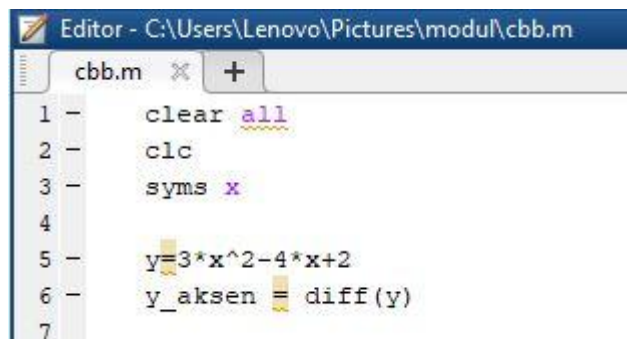
- b) $y = 5$. Dengan program Matlab, dapat dicari turunannya dengan script sebagai berikut:



```
Editor - C:\Users\Lenovo\Pictures\modul\cbb.m
cbb.m x +
1 - clear all
2 - clc
3 - syms x
4
5 - y=5;
6 - y_aksen = diff(y);
7
```

Gambar 2.2 Script turunan dari $y=5$

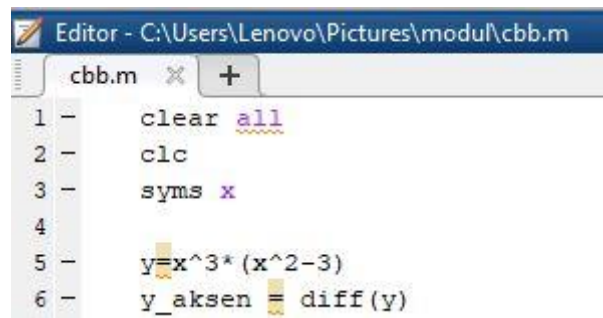
- c) $y = 3x^2 - 4x + 2$. Dengan program Matlab, dapat dicari turunannya dengan script sebagai berikut:



```
Editor - C:\Users\Lenovo\Pictures\modul\cbb.m
cbb.m x +
1 - clear all
2 - clc
3 - syms x
4
5 - y=3*x^2-4*x+2
6 - y_aksen = diff(y)
7
```

Gambar 2.3 Script turunan fungsi $3x^2-4x+2$

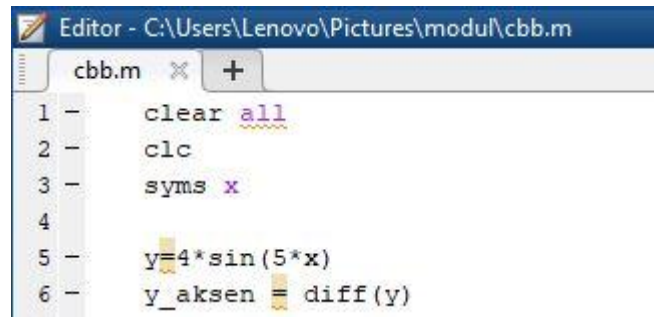
- d) $y = x^3(x^2 - 3)$. Dengan program Matlab, dapat dicari turunannya dengan script sebagai berikut:



```
Editor - C:\Users\Lenovo\Pictures\modul\cbb.m
cbb.m x +
1 - clear all
2 - clc
3 - syms x
4
5 - y=x^3*(x^2-3)
6 - y_aksen = diff(y)
```

Gambar 2.4 Script turunan fungsi $x^3(x^2-3)$

- e) $y = 4\sin(5x)$. Dengan program Matlab, dapat dicari turunannya dengan script sebagai berikut:



```
Editor - C:\Users\Lenovo\Pictures\modul\cbb.m
cbb.m x +
1 - clear all
2 - clc
3 - syms x
4
5 - y=4*sin(5*x)
6 - y_aksen = diff(y)
```

Gambar 2.5 Script turunan fungsi $y=4\sin(5x)$

Tugas

Kerjakan soal berikut di Matlab:

- Jika $y = x^4 \sin(2x)$, maka tentukan turunan pertamanya!
- Jika $y = 2x^{-3} + x^2 - 3x + 5$, maka tentukan nilai dari $y'(2)$!

Modul 3. Penyelesaian Integral dengan Metode Trapezoida

Tujuan Praktikum

1. Mahasiswa dapat memahami berbagai perhitungan dalam fisika yang melibatkan integral
2. Mahasiswa dapat membuat script sederhana berkaitan dengan penyelesaian integrasi secara numerik menggunakan metode Trapezoid

Dasar Teori

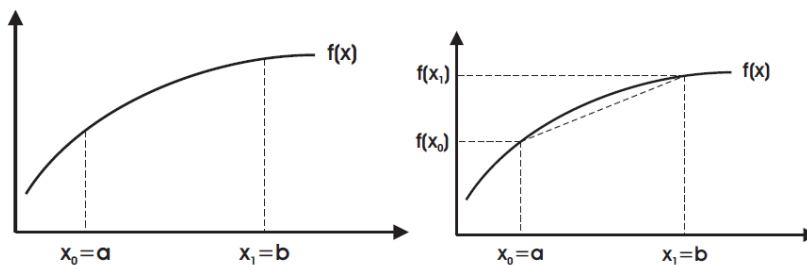
Pendekatan numerik yang paling sederhana dalam memecahkan masalah integral adalah metode Trapezoida, yang dirumuskan sebagai berikut:

$$\int_a^b f(x)dx = \frac{h}{2} [f(x_0) + f(x_1)] - \frac{h^3}{12} f''(\xi)$$

dimana $x_0 = a$, $x_1 = b$ dan $h = b - a$. Akan tetapi, suku terakhir pada ruas kanan dimana terdapat faktor turunan ke-2, f'' , seringkali dapat diabaikan untuk menyederhanakan persamaan:

$$\int_a^b f(x)dx = \frac{h}{2} [f(x_0) + f(x_1)]$$

Pendekatan Trapezoida hanya efektif untuk fungsi-fungsi yang turunan keduanya nol. Gambar berikut memperlihatkan prinsip metode Trapezoida dalam bentuk grafik:



Gambar 3.1 Prinsip metode Trapezoid

Gambar sebelah kiri menunjukkan kurva fungsi $f(x)$ dengan batas bawah integral adalah a dan batas atas b . Gambar sebelah kanan menunjukkan cara metode Trapezoida menghitung integral dengan cara menghitung luas area integrasi, dimana luas area integrasi sama dengan luas trapesium di bawah kurva $f(x)$ dalam batas-batas a dan b . Jika anda perhatikan dengan teliti, ada area kecil dibawah garis kurva dan diatas garis

miring yang berada diluar bidang trapesium. Metode Trapezoida tidak menghitung luas area kecil tersebut. Disinilah letak kelemahan metode trapezoida.

Kegiatan Praktikum

Tentukan hasil integral fungsi $f(x) = x^3$ untuk batas bawah $a = -2$ dan batas atas $b=3$. Bandingkan hasilnya dengan solusi analitik!

- Tentukan nilai h

$$h = b - a = 5$$

- Tentukan nilai $f(a)$ dan $f(b)$

$$f(a) = (-2)^3 = -8$$

$$f(b) = 3^3 = 27$$

- Hitung nilai integral dengan menggunakan formulasi metode Trapezoid

$$\int_{-2}^3 x^3 dx = \frac{h}{2} [f(a) + f(b)] = \frac{5}{2} (-8 + 27) = 47,5$$

- Bandingkan hasilnya dengan solusi analitik

$$\int_{-2}^3 x^3 dx = \frac{1}{4} x^4 \Big|_{-2}^3 = \frac{1}{4} (3^4 - (-2)^4) = 16,25$$

Berikut script untuk mencari integral suatu fungsi menggunakan metode Trapezoid:

```

1 clear all
2 clc
3
4 a = ... %batas bawah integral;
5 b = ... %batas atas integral;
6
7 x0 = a;
8 x1 = b;
9 h = b-a;
10
11 % -- metode trapezoida --
12 Int_trapezoida = h/2*(f(x0)+f(x1))

```

Dengan fungsi eksternal fungsi $f(x)$ adalah

```

1 function y = f(x)
2 y = ... % rumus fungsi yang di-integralkan;

```

Tugas

Tentukan solusi integral fungsi $\sqrt{1+x^2}$ menggunakan metode Trapezoid. Tuliskan script dalam Matlab dan bandingkan hasilnya dengan solusi analitik

Modul 4. Penyelesaian Integral dengan Metode Simpson

Tujuan Praktikum

1. Mahasiswa dapat memahami berbagai perhitungan dalam fisika yang melibatkan integral
2. Mahasiswa dapat membuat script sederhana berkaitan dengan penyelesaian integrasi secara numerik menggunakan metode Simpson

Dasar Teori

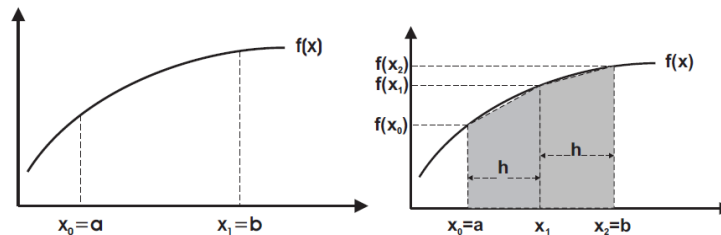
Metode Simpson merupakan salah metode untuk menyelesaikan integral secara numerik, dengan pendekatan yang lebih baik jika dibandingkan dengan Metode Trapezoid. Metode Simpson diformulasikan sebagai berikut:

$$\int_a^b f(x)dx = \frac{h}{3} [f(x_0) + 4f(x_1) + f(x_2)] - \frac{h^5}{90} f^4(\xi)$$

dengan $x_0 = a$, $x_2 = b$, dan $x_1 = a + h$ dimana $h = \frac{b-a}{2}$. Jika suku terakhir diabaikan, maka:

$$\int_a^b f(x)dx = \frac{h}{3} [f(x_0) + 4f(x_1) + f(x_2)]$$

Prinsip metode Simpson ditunjukkan oleh gambar berikut:



Gambar 4.1 Prinsip Metode Simpson

Gambar sebelah kiri menunjukkan kurva fungsi $f(x)$ dengan batas bawah integral adalah a dan batas atas b . Gambar sebelah kanan menunjukkan cara metode Simpson menghitung luas area integrasi, dimana area integrasi di bawah kurva $f(x)$ dibagi 2 dalam batas interval $(a - x_1)$ dan $(x_1 - b)$ dengan lebar masing-masing adalah h .

Tingkat ketelitian metode Simpson dapat ditingkatkan dengan cara membagi interval lebih kecil lagi menjadi banyak interval. Banyaknya pembagian interval

dinyatakan dengan n . Akan tetapi keterbatasan metode Simpson adalah pada jumlah pembagian lebih dari 4, maka solusi penyelesaiannya cukup sulit dilakukan.

Kegiatan Praktikum

Sebuah fungsi $f(x) = \sin x$ pada interval $x=0$ sampa $x=2$. Tentukan solusi fungsi tersebut menggunakan metode Simpson!

- Tentukan x_0, x_1, x_2, h

$$x_0 = 0$$

$$x_2 = 2$$

$$h = \frac{b-a}{2} = 1$$

$$x_1 = a + h = 1$$

- Tentukan $f(x_0), f(x_1), f(x_2)$

$$f(x_0) = \sin 0 = 0$$

$$f(x_1) = \sin(1) = 0,841$$

$$f(x_2) = \sin(2) = 0,909$$

- Tentukan nilai integral fungsi :

$$\int_0^2 \sin x dx = \frac{h}{3} [f(x_0) + 4f(x_1) + f(x_2)] = \frac{1}{3} [0 + 4(0,841) + 0,909] = 1,424$$

- Tentukan nilai integral fungsi secara analitik:

$$\int_0^2 \sin x dx = -\cos x \Big|_0^2 = -\cos 2 + \cos 0 = 1,416$$

Berikut merupakan script dalam Matlab untuk penyelesaian di atas:

```

1  clc
2  clear all
3
4  a = ... %batas bawah integrasi ;
5  b = ... %batas atas integrasi ;
6
7  x0 = a;
8  x2 = b;
9  h = (b-a)/2;
10 x1 = a + h;
11
12 % -- metode simpson --
13 Int_simpson = h/3*(f(x0)+4*f(x1)+f(x2))

```

Tugas

Tentukan solusi integral fungsi $\sqrt{1+x^2}$ menggunakan metode Simpson. Tuliskan script dalam Matlab dan bandingkan hasilnya dengan solusi analitik.

Modul 5. Diferensiasi Numerik dengan Metode Euler

Tujuan Praktikum

1. Mahasiswa dapat mengetahui dan memahami berbagai perhitungan fisika yang melibatkan diferensiasi numerik
2. Mahasiswa dapat membuat script sederhana berkaitan dengan penyelesaian diferensial secara numerik menggunakan metode Euler

Dasar Teori

Metode Euler diturunkan dari deret Taylor. Deret Taylor dirumuskan sebagai:

$$y(t_{i+1}) = y(t_i) + (t_{i+1} - t_i)y'(t_i) + \frac{(t_{i+1} - t_i)^2}{2}y''(\xi_i)$$

dengan memasukkan $h = (t_{i+1} - t_i)$ maka:

$$y(t_{i+1}) = y(t_i) + hy'(t_i) + \frac{h^2}{2}y''(\xi_i)$$

Metode Euler dibangun dengan pendekatan bahwa suku terakhir yang memuat turunan kedua, dapat diabaikan. Sehingga metode Euler diformulasikan sebagai:

$$y_{i+1} = y_i + y'h$$

dimana: $y' = f(x_i, y_i)$

Hal ini dapat berarti nilai baru dari y diprediksi menggunakan slope untuk mengekstrapolasi secara linier dalam selang waktu langkah h . Kelemahan dari metode Euler adalah kesalahan (error) nya sangat tinggi dan makin lama makin membesar.

Kegiatan Praktikum

- Hitung $\frac{dy}{dx} = -2x^3 + 12x^2 - 20x + 8.5$ dari $x=0$ sampai $x=4$ dengan lebar langkah $h=0.5$. syarat awal $y(0)=1$
- Hitung terlebih dahulu solusi analitik dari persamaan tersebut
- Tentukan $f(x) = -2x^3 + 12x^2 - 20x + 8.5$ dan substitusikan seluruh nilai x ke persamaan ini.
- Hitung nilai y menggunakan persamaan :

$$y_{i+1} = y_i + y'h$$

- Buat tabel perhitungan seperti berikut:

x	y'	y
0	8.5	1
0.5
1
...
4

- Buat algoritma dari metode ini dan salin ke dalam bahasa Matlab
- Berikut contoh kode Matlab untuk penyelesaian fungsi di atas:

Berikut adalah script Matlab metode Euler untuk menentukan solusi persamaan di atas:

```
clear all
clc
format long

b=4; batas atas
a=0; batas bawah
h=0.5;
N=(b-a)/h;
y0=1; nilai y awal
x0=0; nilai x awal

%perubahan t per step
for i=1:N
    x(i)=a+(i*h);
end

%solusi y
y(1)=y0+h*(-2*x0^3+12*x0^2-20*x0+8.5);
for i=2:N
    k=i-1
    y(i)=y(k)+h*(-2*x(k)^3+12*x(k)^2-20*x(k)+8.5);
end

%solusi analitik
for i=1:N
    w(i)=-0.5*x(i)^4+4*x(i)^3-10*x(i)^2+8.5*x(i)+1;
end
Plot (x,y,x,w);
```

Tugas

Tentukan solusi PDB berikut menggunakan Metode Euler dan bandingkan hasilnya dengan solusi analitik. Tuliskan kode Matlab untuk menampilkan hasil plot grafiknya.

$$y' = y - t^2 + 1, \quad 0 \leq t \leq 2, \quad y(0) = 0.5$$

Dengan step langkah $h=0,2$. Keterangan: solusi analitik dari PDB di atas adalah:

$$y(t) = (t + 1)^2 - 0.5e^t$$

Modul 6. Diferensiasi Numerik dengan Metode Heun

Tujuan Praktikum

1. Mahasiswa dapat mengetahui dan memahami berbagai perhitungan fisika yang melibatkan diferensiasi numerik
2. Mahasiswa dapat membuat script sederhana berkaitan dengan penyelesaian diferensial secara numerik menggunakan metode Heun

Dasar Teori

Metode Heun merupakan metode yang digunakan untuk memperbaiki estimasi yang dilakukan oleh metode Euler. Jika di metode Euler hanya menggunakan satu perhitungan estimasi slope, maka di metode Heun estimasi slope yang pertama digunakan untuk melakukan estimasi slope kedua yang kemudian diambil nilai rata-ratanya untuk estimasi nilai y berikutnya. Langkah penyelesaian persamaan diferensial biasa menggunakan metode Heun yaitu sebagai berikut:

- a) Tentukan nilai fungsi menggunakan nilai awal x_0 dan y_0 :

$$y' = f(x_i, y_i)$$

- b) Tentukan nilai y_{i+1}

$$y_{i+1} = y_i + y' * h$$

Persamaan disebut sebagai persamaan *predictor*

- c) Tentukan nilai y_{i+1}'

$$y_{i+1}' = f(x_{i+1}, y_{i+1})$$

- d) Tentukan nilai rata-rata \bar{y}

$$\bar{y} = \frac{y' + y_{i+1}'}{2}$$

- e) Tentukan nilai hampiran akhir y_{i+1}

$$y_{i+1} = y_i + \bar{y} * h$$

Yang biasa disebut sebagai persamaan *corrector*

Kegiatan Praktikum

Diberikan persamaan diferensial orde 1:

$$y' = 4e^{0.8x} - 0.5y$$

Dengan batas bawah $x=0$ dan batas atas $x=4$, dengan langkah $h=1$ serta kondisi awal $y(0)=2$. Tentukan solusi PDB tersebut menggunakan metode Heun dengan pendekatan *predictor-corrector*. Solusi analitik nya adalah:

$$y(x) = \frac{4}{1.3} \left(e^{0.8x} - e^{-0.5x} \right) + 2e^{-0.5x}$$

Langkah Kerja:

1. Menentukan nilai fungsi pada $x=0$ dan $y=2$

$$y' = 4e^0 - 0.5(2) = 3$$

2. Menentukan nilai estimasi y pada $x=1$ menggunakan predictor

$$y_1 = 2 + 3(1) = 5$$

3. Untuk memperbaiki nilai estimasi y_{i+1} , digunakan nilai y_1 untuk memprediksi slope di akhir interval

$$y_1' = 4e^{0.8(1)} - 0.5(5) = 6.402164$$

4. Menggabungkan slope awal dan slope akhir untuk menghasilkan slope rata-rata dari interval $x=0$ sampai $x=1$

$$\bar{y} = \frac{3 + 6.402164}{2} = 4.701082$$

5. Solusi (4) disubstitusi ke persamaan korektor untuk memberikan nilai prediksi pada $x=1$

$$y_1 = 2 + 4.701082(1) = 6.701082$$

6. Substitusi balik hasil (5) ke persamaan ruas kanan korektor untuk memperbaiki prediksi y_1

$$y_1 = 2 + \frac{[3 + 4e^{0.8(1)} - 0.5(6.701082)]}{2} = 6.275811$$

7. Lakukan langkah (6) sebanyak iterasi yang diinginkan

$$y_1 = 2 + \frac{[3 + 4e^{0.8(1)} - 0.5(6.275811)]}{2} = 6.382129$$

Kode Matlab Penyelesaian PDB menggunakan Metode Heun:

```

% Program untuk menghitung diffensial orde satu
% y'=(4*exp(0.8*x))-(0.5*y)
% menggunakan metode Heun
% pada x=0 y=2 dari x=0 hingga x=4 step h=1
clc;clear all;
% metoda Heun
y0 = 2;h = 1;x = 0;mx(1) = 0;my(1) = 2;
for ii = 1:4
    f1 = (4*exp(0.8*x))-(0.5*y0);
    y = y0+(f1*h);
    x = x+1;
    f2 = (4*exp(0.8*x))-(0.5*y);
    m = (f1+f2)/2;
    y = y0 + m;
    for jj=1:16
        y1=y;
        y=y0+(((f1+((4*exp(0.8*x))-(0.5*y1))*h)/2);
        y1=y;
    end
    mx(ii+1) = ii;
    my(ii+1) = y;
    y0 = y;
end
% Solusi Analitik
y_analitik = ((4/1.3)*(exp(0.8.*mx)- exp(-0.5.*mx))) + 2*exp(-0.5.*mx);
plot(mx,my,'r',mx,y_analitik,'b');

```

Tugas

Tentukan solusi PDB berikut menggunakan Metode Heun dan bandingkan hasilnya dengan solusi analitik. Tuliskan kode Matlab untuk menampilkan hasil plot grafiknya.

$$y' = y - t^2 + 1, \quad 0 \leq t \leq 2, \quad y(0) = 0,5$$

Dengan step langkah h=0,2.

Keterangan: solusi analitik dari PDB di atas adalah:

$$y(t) = (t + 1)^2 - 0,5e^t$$

Modul 7. Diferensiasi Numerik dengan Metode Runge Kutta

Tujuan Praktikum

1. Mahasiswa dapat mengetahui dan memahami metode Runge Kutta
2. Mahasiswa dapat membuat script sederhana berkaitan dengan penyelesaian diferensial biasa secara numerik menggunakan metode Runge Kutta

Dasar Teori

Metode Runge-Kutta merupakan salah satu jenis metode numerik yang digunakan untuk menyelesaikan persamaan-persamaan diferensial biasa (PDB). Kelebihan dari metode Runge-Kutta adalah memiliki akurasi yang tinggi tanpa memerlukan perhitungan turunan rumit. Variasi dari metode ini sangat banyak, akan tetapi secara umum memiliki bentuk:

$$y_{i+1} = y_i + f(x_i, y_i, h)h$$

dengan $f(x_i, y_i, h)$ merupakan fungsi increment. Dalam bentuk umum fungsi ini dapat dituliskan sebagai:

$$f = a_1k_1 + a_2k_2 + a_3k_3 + \dots + a_nk_n$$

dimana a adalah konstanta dan k :

$$k_1 = f(x_i, y_i)$$

$$k_2 = f(x_i + p_1h, y_i + q_{11}k_1h)$$

$$k_3 = f(x_i + p_2h, y_i + q_{21}k_1h + q_{22}k_2h)$$

.

.

.

$$k_n = f(x_i + p_{n-1}h, y_i + q_{n-1,1}k_1h + q_{n-1,2}k_2h + \dots + q_{n-1,n-1}k_{n-1}h)$$

dengan q dan p adalah konstanta. Nilai-nilai k berhubungan secara rekursif. Orde dari metode Runge-Kutta ditentukan dari indeks n yang melekat pada k . Berikut beberapa variasi dari metode Runge-Kutta yang sering digunakan:

a. Metode Runge-Kutta Orde 2

$$y_{i+1} = y_i + \left(\frac{1}{3}k_1 + \frac{2}{3}k_2\right)h$$

dengan

$$k_1 = f(x_i, y_i)$$

$$k_2 = f\left(x_i + \frac{3}{4}h, y_i + \frac{3}{4}k_1h\right)$$

b. Metode Runge-Kutta Orde 3

$$y_{i+1} = y_i + \frac{1}{6}(k_1 + 4k_2 + k_3)h$$

dengan

$$k_1 = f(x_i, y_i)$$

$$k_2 = f(x_i + \frac{1}{2}h, y_i + \frac{1}{2}k_1h)$$

$$k_3 = f(x_i + h, y_i - k_1h + 2k_2h)$$

c. Metode Runge-Kutta Orde 4

$$y_{i+1} = y_i + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)h$$

dengan

$$k_1 = f(x_i, y_i)$$

$$k_2 = f(x_i + \frac{1}{2}h, y_i + \frac{1}{2}k_1h)$$

$$k_3 = f(x_i + \frac{1}{2}h, y_i + \frac{1}{2}k_2h)$$

$$k_4 = f(x_i + h, y_i + k_3h)$$

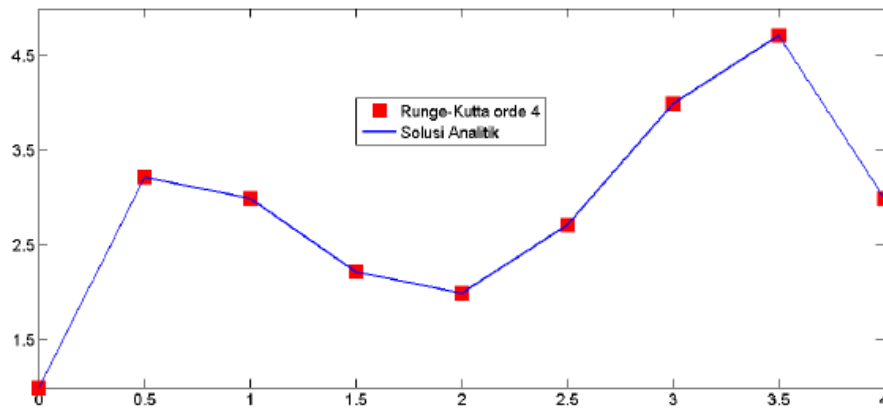
Kegiatan Praktikum

Sebuah persamaan diferensial biasa orde 1: $y' = -2x^3 + 12x^2 - 20x + 8.5$ akan dicari solusi untuk $x=0$ sampai $x=4$ dengan step langkah $h=0.5$ serta kondisi awal $x=0$ nilai $y=1$ menggunakan metode Runge-Kutta Orde 4.

Gunakan bantuan Matlab dengan kode berikut:

```
clc;clear all;
x0=0;y0=1;h=0.5;my(1)=1;mx(1)=0;
for ii = 1:8
    k1 = (-2*(x0^3))+(12*(x0^2))-(20*x0)+8.5;
    x1 = x0+(0.5*h);y1=y0+(0.5*h*k1);
    k2 = (-2*(x1^3))+(12*(x1^2))-(20*x1)+8.5;
    y2 = y0+(0.5*h*k2);
    k3 = (-2*(x1^3))+(12*(x1^2))-(20*x1)+8.5;
    x2 = x0+h; y3=y0+(0.5*h*k3);
    k4 = (-2*(x2^3))+(12*(x2^2))-(20*x2)+8.5;
    y = y0 + ((1/6)*(k1+(2*k2)+(2*k3)+k4)*h);
    x0 = x0 + h;
    y0 = y;
    my(ii+1) = y;
    mx(ii+1) = ii/2;
end
x_analitik=0:0.5:4
y_analitik=(-0.5.*x_analitik.^4)+(4.*x_analitik.^3)-(10.*x_analitik.^2)+(8.5.*x_analitik)+1
plot(mx,my,'or',x_analitik,y_analitik)
```

Tampilan hasilnya sebagai berikut:



Aplikasi Dalam Fisika: Pengisian Muatan pada Kapasitor

Sebuah kapasitor tidak bermuatan dihubungkan secara seri dengan sebuah resistor dan baterai. Diketahui tegangan $V = 12 \text{ volt}$, $C = 5 \mu\text{F}$, dan $R = 8 \times 10^5 \text{ ohm}$. Saat saklar dihubungkan ($t=0$), muatan belum ada ($q=0$).

$$\frac{dq}{dt} = \frac{V}{R} - \frac{q}{RC}$$

Solusi analitik dari persamaan tersebut adalah:

$$q = CV(1 - e^{-t/RC})$$

Untuk menyelesaikan solusi persamaan di atas menggunakan metode runge kutta, kita misalkan:

$$m_1 = \frac{V}{R}$$

$$m_2 = \frac{1}{RC}$$

Sehingga $q = m_1 - qm_2$, maka:

$$q_{i+1} = q_i + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$

Jika $t_0=0$, maka $a=0$, dan pada saat itu (secara fisis) diketahui $q_0 = 0, 0$. Lalu jika ditetapkan $h = 0, 1$ maka $t_1 = 0, 1$ dan kita bisa mulai menghitung k_1 dengan menggunakan $q_0 = 0, 0$, walaupun t_1 tidak dilibatkan dalam perhitungan ini.

Script matlab dari persamaan di atas adalah sebagai berikut:

```

1 clear all
2 clc
3
4 format long
5
6 b=1; % batas akhir interval
7 a=0; % batas awal interval
8 h=0.1; % interval waktu
9 N=(b-a)/h; % nilai step-size
10 q0=0.0; % muatan mula-mula
11 t0=0.0; % waktu awal
12
13 % perubahan t sesuai step-size h adalah:
14 for i=1:N
15     t(i)=a+(i*h);
16 end
17
18 % solusinya:
19 k1=h*futur(q0);
20 k2=h*futur(q0+k1/2);
21 k3=h*futur(q0+k2/2);
22 k4=h*futur(q0+k3);
23 q(1)=q0+1/6*(k1+2*k2+2*k3+k4);
24
25 for i=2:N
26     k=i-1;
27     k1=h*futur(q(k));
28     k2=h*futur(q(k)+k1/2);
29     k3=h*futur(q(k)+k2/2);
30     k4=h*futur(q(k)+k3);
31     q(i)=q(k)+1/6*(k1+2*k2+2*k3+k4);
32 end
33 q

```

Tugas

Dengan menggunakan metode Runge-Kutta Orde 4, tentukan solusi dari persamaan diferensial orde 1: $y' = 4e^{0.8x} - 0.5y$ untuk $x=0$ hingga $x=4$ dengan langka $h=1$ serta kondisi awal $y=2$ pada saat $x=0$. Bandingkan hasilnya dengan solusi analitiknya. Buat kode Matlab untuk menentukan solusi tersebut.

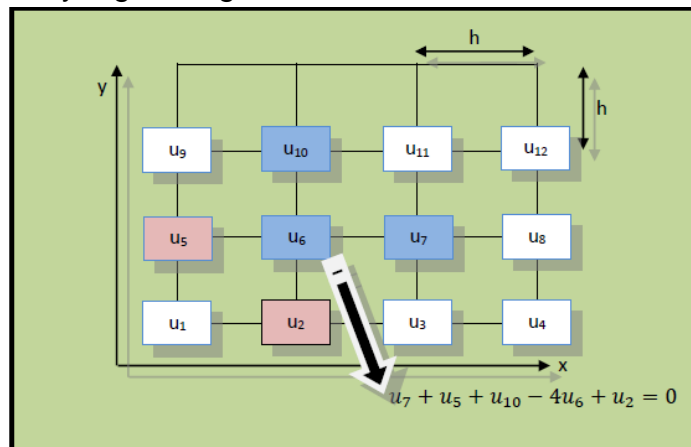
Modul 8. Metode Beda Hingga (*Finite Differences*) untuk PDP Eleptik

Tujuan Praktikum

1. Mahasiswa dapat mengetahui dan memahami metode Beda Hingga
2. Mahasiswa dapat membuat script sederhana berkaitan dengan penyelesaian Persamaan Diferensial Parsial tipe Eleptik secara numerik menggunakan metode Beda Hingga

Dasar Teori

Beda hingga merupakan suatu metode untuk menentukan solusi dari suatu persamaan diferensial parsial dengan menghampiri turunan fungsi dengan deret Taylor untuk kemudian diambil selisih dari hampiran maju dan hampiran mundur yang dilakukan fungsi tersebut. Untuk menggunakan metode beda hingga, pertama harus membagi domain permasalahan menjadi subdomain yang lebih kecil. Setiap variabel pada titik simpul dari subdomain tersebut kemudian akan digunakan untuk membangun sebuah sistem persamaan linier (SPL) (yaitu pada persamaan Laplace, domain ruang pada persamaan difusi dan persamaan gelombang) ataupun sistem persamaan yang bersifat iteratif (yaitu pada domain waktu di persamaan difusi maupun gelombang). Solusi dari persamaan diferensial tersebut adalah solusi dari SPL maupun solusi iteratif dari sistem persamaan yang dibangun.



Gambar 8.1 Prinsip Metode Beda Hingga

Adapun bentuk umum hampiran pada metode beda hingga adalah:

$$\frac{\partial^2 u}{\partial x^2} \cong \frac{u(x+h) - 2u(x) + u(x-h)}{h^2}$$

dengan h merupakan jarak antara satu titik simpul ke titik simpul yang lain.

Untuk persamaan Laplace, formulasi metode beda hingga dengan jarak antar titik simpul adalah h , dituliskan dalam bentuk:

$$\frac{u(x+h, y) - 2u(x, y) + u(x-h, y)}{h^2} + \frac{u(x, y+h) - 2u(x, y) + u(x, y-h)}{h^2} = 0$$

Solusi PDP akan bernilai unik apabila diberikan syarat batas. Dikenal 2 (dua) jenis syarat batas yang digunakan yaitu:

1. Syarat batas Dirichlet, yaitu nilai parameter fisis permasalahan diketahui di dalam maupun pada batas domain.
2. Syarat batas Neumann, yaitu nilai turunan pertama dari parameter fisis yang ditinjau diketahui baik di dalam maupun pada batas domain.

Persamaan linier yang dibangun untuk setiap titik simpul kemudian diselesaikan dengan metode penyelesaian SPL yang pernah dibahas sebelumnya. Solusi dari SPL tersebut adalah solusi bagi PDP Eliptik untuk setiap titik simpul yang terdapat di dalam domain permasalahan.

Syarat batas yang diberikan disubstitusikan pada SPL yang dibangun. Untuk syarat batas Dirichlet, secara langsung telah diaplikasikan pada SPL yang dibangun. Khusus untuk syarat batas Neumann, titik simpul ditambah pada bagian subdomain yang diketahui syarat batas Neumannnya. Turunan pertama dihipotesis dengan perbedaan tengah kemudian diaplikasikan pada titik simpul batas domain. SPL yang telah dibangun sebelumnya kemudian dimodifikasi pada persamaan linier bagian titik simpul.

Dari persamaan umum PDP, jenis eliptik diperoleh apabila nilai D memenuhi hubungan:

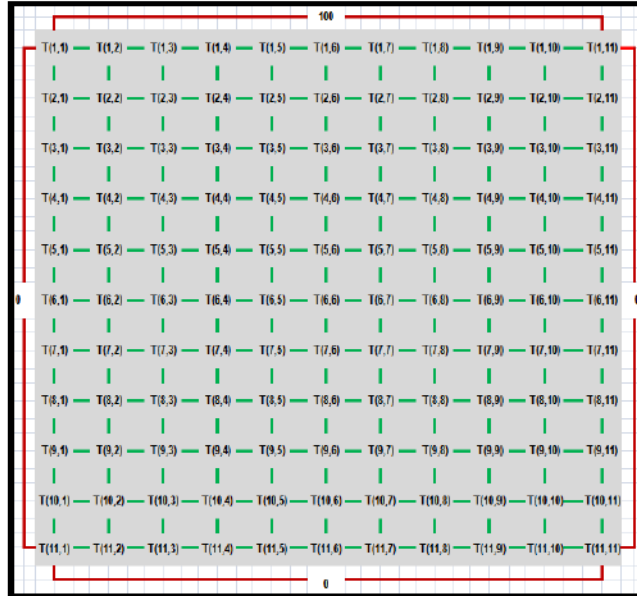
$$D = B^2 - 4AC < 0$$

Kegiatan Praktikum

- Diberikan sebuah persoalan sebagai berikut: Sebuah plat baja berbentuk bujur sangkar memiliki dimensi 10 cm x 10 cm. Plat tersebut dikondisikan memiliki suhu pada 3 (tiga) sisinya 0°C dan satu sisi lainnya 100°C. Tentukan suhu di bagian dalam dari plat baja tersebut pada keadaan tunak dan efek sumber pemanas diabaikan.
- Jika diberikan variabel T adalah temperatur, maka permasalahan tersebut sesuai dengan persamaan Laplace dalam koordinat kartesian:

$$\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} = 0$$

- Diskritisasi domain menjadi 10 bagian sama besar baik dalam arah x maupun y (yaitu $\Delta x = \Delta y = h = 1$ cm)



- Beri tanda setiap titik simpul sesuai kaidah matriks. Inisialisasi syarat batas Dirichlet pada setiap titik simpul yang didefinisikan.

```
% inisialisasi syarat batas Dirichlet
clc;clear all;
nx=10;           % jumlah bagian x
ny=10;           % jumlah bagian y

T(1,1:nx+1)=100;
T(1:ny+1,1)=0;
T(1:ny+1,nx+1)=0;
T(ny+1,1:nx+1)=0;
```

- Buat sistem persamaan linier dari setiap titik simpul yang berada di dalam domain.

$$T(2,2) = \frac{T(3,2) + T(1,2) + T(2,1) + T(2,3)}{4}$$

$$T(10,10) = \frac{T(11,10) + T(9,10) + T(10,9) + T(10,11)}{4}$$

Dalam bahasa Matlab dituliskan dalam bentuk:

```
j_unk = (nx-1)*(ny-1); % jumlah titik simpul yang tidak diketahui
u      = zeros(j_unk,j_unk);
u1=diag(ones(j_unk-1,1),1);
u2=diag(ones(j_unk-1,1),-1);
u3=diag(ones(j_unk-(nx-1),1),nx-1);
u4=diag(ones(j_unk,1)*-4);
u5=diag(ones(j_unk-(nx-1),1),-(nx-1));
u=u1+u2+u3+u4+u5;
for j=1:fix(nx/2)
    u((j*(nx-1))+1,(j*(nx-1)))=0;
    u((j*(nx-1)),(j*(nx-1)+1))=0;
end
A = u;
```

- Susun sistem persamaan linier tersebut dalam format berikut:

$$[A]\{T_{unknown}\} = \{T_{known}\}$$

dengan

$$\{T_{unknown}\} = \{T_{22}T_{23}\dots T_{1010}\}$$

dalam bahasa Matlab dituliskan dalam bentuk:

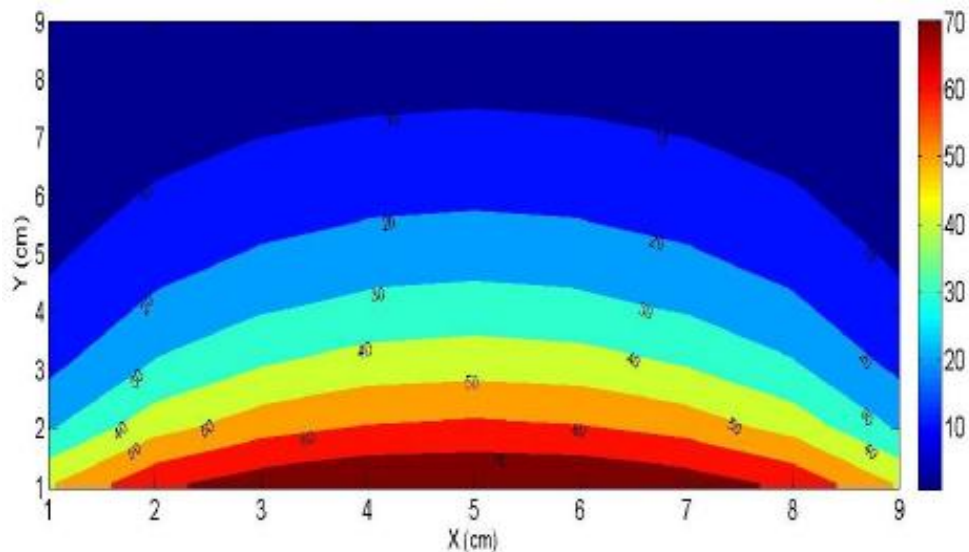
```
M=zeros(j_unk,1);
for i = 1:(nx-1)      % dirichlet atas
    M(i,1)=M(i,1)+T(1,i+1);
end
for i=1:(ny-1)      % dirichlet kanan
    M((nx-1)*i,1)=M((nx-1)*i,1)+T(i+1,nx+1);
end
for i=1:(ny-1)      % dirichlet kiri
    M((i-1)*(nx-1)+1,1)=M((i-1)*(nx-1)+1,1)+T(i+1,1);
end
for i=1:(nx-1)      % dirichlet bawah
    M(2*(nx-1)+i,1)=M(2*(nx-1)+i,1)+T(ny+1,i+1);
end
T_known = -M;
```

- Selesaikan SPL yang sudah dibuat dengan memanfaatkan fungsi invers matriks *built-in* dalam Matlab.

```
T_known = -M;
T_unknown = (A'*A) \ (A'*T_known);

% untuk plot
T_unknown = reshape(T_unknown,nx-1,ny-1)';
image(T_unknown(1:nx-1,1:ny-1));figure(gcf);
```

- Analisis hasilnya dan bandingkan dengan solusi analitiknya:



Tugas

Carilah solusi persamaan diferensial eliptik berikut dengan menggunakan pendekatan beda hingga:

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = (x^2 + y^2)e^{xy}, \quad 0 < x < 2, \quad 0 < y < 1;$$

gunakan $h=0,2$ dan $k=0,1$

$$u(0, y) = 1 \quad u(2, y) = e^{2y} \quad 0 \leq y \leq 1$$

$$u(x, 0) = 1 \quad u(x, 1) = e^x \quad 0 \leq x \leq 2$$

Bandingkan hasilnya dengan solusi analitik $u(x, y) = e^{xy}$

Modul 9. Metode Beda Hingga (*Finite Differences*) untuk PDP Parabolik

Tujuan Praktikum

1. Mahasiswa dapat mengetahui dan memahami metode Beda Hingga
2. Mahasiswa dapat membuat script sederhana berkaitan dengan penyelesaian Persamaan Diferensial Parsial tipe Parabolik secara numerik menggunakan metode Beda Hingga

Dasar Teori

Dari persamaan umum PDP, jenis hiperbolik diperoleh apabila nilai D memenuhi hubungan:

$$D = B^2 - 4AC = 0$$

Persamaan Diferensial Parsial (PDP) jenis parabolik memiliki bentuk:

$$\nabla^2 u = C_p \frac{\partial u}{\partial t}$$

Untuk permasalahan dalam koordinat kartesian 2 dimensi, persamaan tersebut dapat dituliskan:

$$k \left(\frac{\partial^2 u}{dx^2} + \frac{\partial^2 u}{dy^2} \right) = C_p \frac{\partial u}{\partial t}$$

dengan k dan C_p adalah sebuah konstanta. Salah satu contoh Persamaan Diferensial Parsial adalah Persamaan Difusi. Syarat batas yang berlaku pada PDP ini sama seperti syarat batas yang berlaku pada PDP Eliptik, yaitu syarat batas Dirichlet dan syarat batas Neumann. Terdapat sedikit perbedaan hampiran untuk kedua syarat batas ini jika dibandingkan dengan PDP Eliptik.

Ada 2 jenis cara penyelesaian pada PDP Parabolik, yaitu Metode Eksplisit dan Metode Implisit.

Metode Eksplisit

Pendekatan yang digunakan pada metode ini adalah perbedaan tengah untuk domain ruang, sementara untuk domain waktu digunakan hampiran perbedaan maju. Formulasi yang digunakan pada metode ini jika kasusnya dalam 1 dimensi adalah:

$$u_i^{j+1} = \frac{\Delta t}{C_p(\Delta x)^2} (u_{i+1}^j + u_{i-1}^j) + \left(1 - \frac{2\Delta t}{C_p(\Delta x)^2} \right) u_i^j$$

Hampiran yang umum digunakan untuk menjaga kestabilan perhitungan adalah:

$$\frac{\Delta t}{Cp(\Delta x)^2} = \frac{1}{2}$$

Sedangkan untuk kasus 2 dimensi dan $\Delta x = \Delta y$, digunakan:

$$\frac{\Delta t}{Cp(\Delta x)^2} \leq \frac{1}{4}$$

Metode Implisit (Metode Crank-Nicolson)

Persamaan Diferensial Parsial pada metode Implisit dapat dituliskan:

$$\nabla^2 u = C_p \frac{\partial u}{\partial t} \rightarrow \frac{1}{2} \left(\frac{u_{i+1}^j - 2u_i^j + u_{i-1}^j}{(\Delta x)^2} + \frac{u_{i+1}^{j+1} - 2u_i^{j+1} + u_{i-1}^{j+1}}{(\Delta x)^2} \right) = C_p \left(\frac{u_i^{j+1} - u_i^j}{\Delta t} \right)$$

yang dapat juga dituliskan sebagai:

$$-ru_{i-1}^{j+1} + (2 + 2r)u_i^{j+1} - ru_{i+1}^{j+1} = ru_{i-1}^j + (2 - 2r)u_i^j + ru_{i+1}^j$$

dimana,

$$r = \frac{\Delta t}{Cp(\Delta x)^2}$$

dengan dipilih nilai $r=1$ untuk penyederhaan.

Kelebihan dari Metode Implisit ini adalah adalah kestabilan perhitungan untuk berbagai nilai r , namun lebih disarankan untuk menggunakan nilai r yang cukup kecil.

Kegiatan Praktikum

- Jika diberikan permasalahan sebagai berikut: Sebuah kubus alumunium memiliki ukuran 10cm x 10cm x 10cm ($k=217 \text{ J/sm}^\circ\text{C}$, $c=946 \text{ J/kg}^\circ\text{C}$, $\rho = 2,7.103 \text{ kg/m}^3$). Satu sisi pada kubus tersebut tidak diisolasi dengan baik dan kubus dibuat bersuhu 550°C pada saat awal. Panas yang hilang berasal dari sisi yang tak diisolasi tersebut akibat kontak dengan aliran fluida berdasarkan persamaan (temperatur fluida = 20°C) :

$$\text{laju kehilangan panas} = hA(u-u_0)$$

dengan

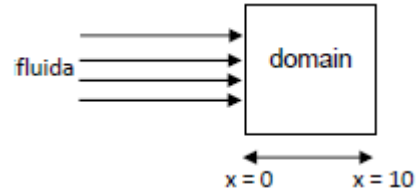
$$h = 98,9117 \text{ J/sm}^2\text{°C}$$

A = luas permukaan setiap sisi aluminium

u = temperatur (°C)

u₀ = temperatur fluida

Tentukan distribusi suhu di dalam kubus setiap saatnya.



- Dari soal dapat ditentukan $u(x,0)=550$ dan pada sisi terbuka berlaku

$$-kA \left. \frac{\partial u}{\partial x} \right|_{x=0} = -hA(u - 20)$$

karena laju aliran panas yang keluar sama dengan laju aliran panas yang masuk sistem. Tanda negatif menunjukkan arah yang berlawanan dengan arah positif gradien. Pada sisi kubus yang lain, laju aliran panas yang keluar adalah

$$\left. \frac{\partial u}{\partial x} \right|_{x=4} = 0$$

- Jika digunakan metode eksplisit dengan $\Delta x = 2\text{cm}$, dan dengan menggunakan hubungan berikut:

$$\frac{k\Delta t}{(\rho\Delta x)^2} = \frac{1}{4}$$

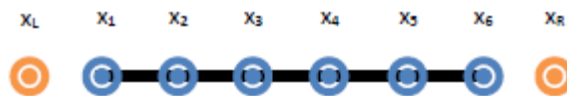
maka akan dihasilkan $\Delta t = 1.17\text{s}$.

Untuk rasio yang dipilih tersebut, persamaan beda hingga untuk kasus ini menjadi:

$$u_i^{j+1} = \frac{1}{4}(u_{i+1}^j + u_{i-1}^j) + \frac{1}{2}u_i^j$$

yang akan diaplikasikan pada titik $x=0$ hingga $x=10$.

- Karena terdapat syarat batas Neuman, ujung domain ditambahkan sebuah node khayal x_L (untuk ujung kiri) dan x_R (untuk ujung kanan)



dan aplikasikan persamaan berikut pada node khayal.

$$u_1^{j+1} = \frac{1}{4}(u_2^j + u_L^j) + \frac{1}{2}u_1^j$$

$$u_6^{j+1} = \frac{1}{4}(u_R^j + u_5^j) + \frac{1}{2}u_6^j$$

- Penggunaan syarat batas Neumann digunakan untuk menghilangkan titik khayal ini. Formulasi beda tengah untuk syarat batas ini memberikan:

$$-k \frac{\partial u}{\partial x} \Big|_{x=0} = -217 \left(\frac{u_2^j - u_L^j}{2(2)} \right) = 98,97(u_1^j - 20)$$

$$-k \frac{\partial u}{\partial x} \Big|_{x=6} = -217 \left(\frac{u_R^j - u_4^j}{2(2)} \right) = 0$$

- Susun ke dalam sistem persamaan linier hasil dari penggunaan syarat batas tersebut di atas. Selesaikan menggunakan salah satu metode pencarian solusi pada sistem persamaan linier.
- Plot hasil yang diperoleh menggunakan fasilitas grafik pada Matlab

Tugas

Carilah solusi persamaan differensial parabolik berikut ini dengan pendekatan numerik menggunakan metode *Finite Difference Backward-Difference*:

$$\frac{\partial u}{\partial t} - \frac{1}{16} \frac{\partial^2 u}{\partial x^2} = 0, \quad 0 < x < 1, \quad 0 < t;$$

$$u(0, t) = u(1, t) = 0, \quad 0 < t;$$

$$u(x, 0) = 2 \sin 2\pi x, \quad 0 \leq x \leq 1;$$

Gunakan $m = 3$, $T = 0,1$, dan $N = 2$. Bandingkan hasilnya dengan solusi analitik:

$$u(x, t) = 2e^{-(\pi^2/4)t} \sin 2\pi x$$

Modul 10. Metode Beda Hingga (*Finite Differences*) untuk PDP Hiperbolik

Tujuan Praktikum

1. Mahasiswa dapat mengetahui dan memahami metode Beda Hingga
2. Mahasiswa dapat membuat script sederhana berkaitan dengan penyelesaian Persamaan Diferensial Parsial tipe Hiperbolik secara numerik menggunakan metode Beda Hingga

Dasar Teori

Jenis Persamaan Diferensial Parsial jenis hiperbolik umumnya ditemukan dalam persamaan gelombang. Dari persamaan umum PDP, jenis hiperbolik diperoleh apabila nilai D memenuhi hubungan:

$$D = B^2 - 4AC > 0$$

Contoh sederhana dari PDP ini adalah persamaan gelombang pada tali yang berbentuk:

$$\frac{\partial^2 y}{\partial t^2} = c \frac{\partial^2 y}{\partial x^2}$$

dengan hampiran deret Taylor diperoleh:

$$\frac{y_i^{j+1} - 2y_i^j + y_i^{j-1}}{\Delta t^2} = c \left(\frac{y_{i+1}^j - 2y_i^j + y_{i-1}^j}{\Delta x^2} \right)$$

Untuk hampiran nilai x yang menuju nol, didapat nilai y yang stabil memenuhi persamaan:

$$y_i^{j+1} = y_{i+1}^j + y_{i-1}^j - y_i^{j-1}$$

Dengan aturan beda tengah maka akan diperoleh nilai untuk y_i^1 yakni sebesar:

$$y_i^1 = \frac{1}{2}(y_{i+1}^0 + y_{i-1}^0) + g(x)\Delta t$$

sehingga solusi persamaan diferensial tersebut dapat ditentukan.

Kegiatan Praktikum

- Jika diberikan kasus berikut: Sebuah alat musik terdiri dari seutas senar dan sebuah sistem resonansi. Senar alat musik tersebut ditarik hingga 0.6 cm pada

titik berjarak 20 cm dari salah satu ujung stasioner, kemudian dilepas dan mengalami osilasi. Jika diketahui $C=40,82$ (untuk sementara satuan C diasumsikan setara), tentukan simpangan tali tiap waktunya menggunakan metode beda hingga eksplisit.

- Ikuti penurunan persamaan di atas kemudian masukkan parameter fisis pada soal. Diskritisasi ruang dan waktu disesuaikan dengan nilai kestabilan metode ini.
- Salin Algoritma berikut ke dalam bahasa Matlab

```

Algoritma

Masukan : T,g,w,l(panjang tali),delta_x
Keluaran : yi
Langkah :
  n=l/delta_x
  C=sqrt((w*delta_x)/(T*g))
  x=0:l/n:l
  [q,p]=size(x)
  Untuk i=1:q
    Jika x(i)<2
      u(i,1)=0.1*x
    lainnya u(i,1)=-0.1*x+0.8
  Untuk j=2:t
    untuk i=2:o-1
      u(i,j)=(1/2)*(u(i-1,j-1)+u(i+1,j-1))
  
```

- Plot menggunakan fasilitas grafik pada Matlab kemudian analisa hasilnya.

Tugas

Suatu fungsi perubahan suhu $\frac{\partial u}{\partial t} = -c \frac{\partial u}{\partial x}$ dengan u menyatakan suhu potensial yang merupakan besaran kekal. Tentukan solusi penyelesaian persamaan tersebut menggunakan metode Beda Hingga dan bandingkan hasilnya dengan solusi analitik.

Daftar Pustaka

- Arman, Y. 2018. Modul Praktikum Fisika Komputasi II. Universitas Tanjungpura Pontianak.
- Indratno, T.K. 2019. Pengembangan Modul Komputasi Fisika untuk Mahasiswa Jurusan Kependidikan. JRKPF UAD Vol.6 No.1
- Rianto, S dan Ubaidillah. 2010. Modul Praktikum Fisika Komputasi dengan Contoh Bahasa Pemrograman Pascal, Matlab, dan Phyton. Universitas Brawijaya: Malang.
- Suparno, S. 2010. Komputasi untuk Sains dan Teknik Menggunakan Matlab Edisi III. Universitas Indonesia: Depok.