Modul I
Metode Beda Hingga Turunan Pertama

1. Tujuan Praktikum

a. Mahasiswa mampu menyusun program komputer dalam bahasa FORTRAN untuk menghitung nilai turunan pertama dari suatu fungsi atau berdasarkan data variabel medan.
b. Mahasiswa mampu menyusun program komputer dalam bahasa FORTRAN untuk menyelesaikan suatu persamaan model adveksi.
c. Mahasiswa mampu menyusun program komputer dalam bahasa FORTRAN untuk menghitung hasil integrasi numerik dari suatu persamaan gelombang sederhana berdasarkan skema Euler, Backward dan Leap Frog.
d. Mahasiswa mampu memahami permasalahan kriteria kestabilan.

2. Kajian Pustaka

2.1 Persamaan Beda Hingga untuk Turunan Pertama

Ekspansi Taylor untuk $u(x)$ di sekitar Δx :

$$u(x+\Delta x) = u(x) + \frac{d}{dx}u(x)\Delta x + \frac{d^2}{dx^2}u(x)\frac{(\Delta x)^2}{2!} + \frac{d^3}{dx^3}u(x)\frac{(\Delta x)^3}{3!} + O[(\Delta x)^4] \qquad (1)$$

$$u(x-\Delta x) = u(x) - \frac{d}{dx}u(x)\Delta x + \frac{d^2}{dx^2}u(x)\frac{(\Delta x)^2}{2!} - \frac{d^3}{dx^3}u(x)\frac{(\Delta x)^3}{3!} + O[(\Delta x)^4] \qquad (2)$$

x menunjukkan variabel jarak/ruang, $u(x)$ menunjukkan u adalah fungsi dari x, $\frac{d}{dx}u(x)$ menunjukkan turunan pertama orde pertama, Δx menunjukkan perubahan pada variabel x dan $O[(\Delta x)^4]$ menunjukkan tingkat ketelitian orde keempat.

First order

Dari persamaan (1) dapat diperoleh:

$$\frac{du}{dx}(x) = \frac{u(x+\Delta x)-u(x)}{\Delta x} + O[(\Delta x)] \qquad (3)$$

Apabila $u(x) = u_i, u(x+\Delta x) = u_{i+1}$ maka,

$$\frac{du}{dx}(x) = \frac{u_{i+1} - u_i}{\Delta x} + O[(\Delta x)$$

Persamaan (3) disebut juga dengan pendekatan beda maju (Euler) dengan tingkat ketelitian orde pertama.

Dari persamaan (2) dapat diperoleh:

$$\frac{\mathrm{du}}{\mathrm{dx}}(\mathrm{x}) = \frac{\mathrm{u(x)-u(x-\Delta x)}}{\Delta \mathrm{x}} + \mathrm{O}[(\Delta \mathrm{x})] \qquad (4)$$

Apabila $u(x) = u_i, u(x - \Delta x) = u_{i-1}$ maka,

$$\frac{du}{dx}(x) = \frac{u_i - u_{i-1}}{\Delta x} + O[(\Delta x)]$$

Persamaan (4) disebut juga dengan pendekatan beda mundur dengan tingkat ketelitian orde pertama.

Second Order

Dengan mengurangkan persamaan (1) dan (2) atau menjumlahkan persamaan (3) dan (4), diperoleh:

$$\frac{du}{dx}(x) = \frac{u(x+\Delta x)-u(x-\Delta x)}{2\Delta x} + O[(\Delta x)^2] \qquad (5)$$

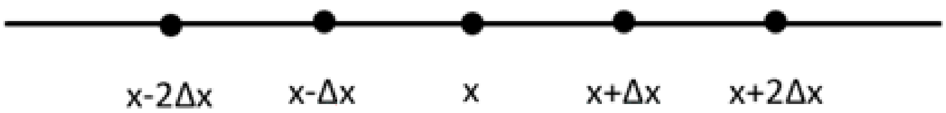
Apabila $u(x + \Delta x) = u_{i+1}, u(x - \Delta x) = u_{i-1}$ maka,

$$\frac{du}{dx}(x) = \frac{u_{i+1} - u_{i-1}}{2\Delta x} + O[(\Delta x)^2]$$

Persamaan (5) disebut juga dengan pendekatan beda tengah dengan tingkat ketelitian orde kedua.

Fourth Order

Ketelitian lebih tinggi dapat dicapai dengan menggunakan beberapa rumus ekspansi Taylor untuk titik $u(x + \Delta x)$,$u(x - \Delta x)$,$u(x + 2\Delta x)$ dan $u(x - 2\Delta x)$ yang terletak seperti pada gambar dibawah.



Langkah pertama adalah mencari turunan pertama beda tengah dari masing-masing $u(x + \Delta x)$,$u(x - \Delta x)$ dan $u(x + 2\Delta x)$, $u(x - 2\Delta x)$. Langkah selanjutnya adalah mencari turunan ketiga dari turunan pertama beda tengah $u(x + \Delta x)$,$u(x - \Delta x)$ dan selanjutnya hasil tersebut disubstitusi pada turunan pertama beda $u(x + 2\Delta x)$, $u(x - 2\Delta x)$. Dari perhitungan tersebut akan didapat turunan pertama seperti berikut:

$$\frac{du}{dx}(x) = \frac{u(x-2\Delta x)-8u(x-\Delta x)+8u(x+\Delta x)-u(x+2\Delta x)}{12\Delta x} + O[(\Delta x)^4] \qquad (6)$$

Apabila $u(x + 2\Delta x) = u_{i+2}, u(x + \Delta x) = u_{i+1}, u(x - \Delta x) = u_{i-1}, u(x - \Delta x) = u_{i-1}$ maka

$$\frac{du}{dx}(x) = \frac{u_{i-2} - 8u_{i-1} + 8u_{i+1} - u_{i+2}}{12\Delta x} + O[(\Delta x)^4]$$

Persamaan (6) disebut juga dengan pendekatan beda tengah dengan tingkat ketelitian orde keempat. Langkah penurunan lengkap lihat di buku Krishnamurti (2006) bab 2.5.

2.2 Skema Eksplisit untuk Model Adveksi Sederhana

Untuk memprediksi nilai u dengan persamaan adveksi sederhana yaitu:

$$\frac{\partial u}{\partial t} = -a\frac{\partial u}{\partial x}$$

Dengan t merupakan variable waktu, x merupakan variable jarak dan a merupakan konstanta. Kita dapat menyelesaikan persamaan di atas menggunakan beberapa metode eksplisit beda hingga seperti:

a. Forward-in-Time and Forward-in-Space (FTFS)/*Euler*

$$U_m^n = U_m^{n-1} - a\frac{\Delta t}{\Delta x}[U_{m+1}^{n-1} - U_m^{n-1}]$$

b. Forward-in-Time and Backward-in-Space (FTBS)

$$U_m^n = U_m^{n-1} - a\frac{\Delta t}{\Delta x}[U_m^{n-1} - U_{m-1}^{n-1}]$$

c. Forward-in-Time and Centered-in-Space (FTCS)

$$U_m^n = U_m^{n-1} - a\frac{\Delta t}{2\Delta x}[U_{m+1}^{n-1} - U_{m-1}^{n-1}]$$

d. Centered-in-Time and Centered-in-Space (FTFS)/*Leapfrog*

$$U_m^{n+1} = U_m^{n-1} - a\frac{\Delta t}{\Delta x}[U_{m+1}^{n} - U_{m-1}]$$

Dengan n merupakan indeks waktu dan m merupakan indeks jarak.

2.3 Konsep Kestabilan Model Adveksi

Model adveksi yang baik memberikan nilai prediksi yang stabil. Model dikatakan tidak stabil jika nilai-nilai prediksi yang dihasilkan menunjukkan amplifikasi yang semakin lama semakin besar dan menjadi tidak realistis. Dalam metode beda hingga konsep kestabilan dapat dibagi dua: kestabilan numerik dan kestabilan fisis.

Kestabilan numerik ditunjukkan oleh rasio c langkah waktu Δt dan langkah jarak Δx (*grid spacing*) yang dikalikan dengan kecepatan adveksi a haruslah kurang dari atau sama dengan 1:

$$c = a\frac{\Delta t}{\Delta x}$$

Nilai c di atas disebut CFL (Courant, Freidrichs dan Levy) *number* atau seringkali Courant *number*.

Meskipun kestabilan numerik terpenuhi, model tetap ada peluang untuk tidak stabil jika kondisi fisis tidak terpenuhi. Kondisi fisis ini bergantung pada arah adveksi (tanda a) dan metode beda hingga yang dipakai. Faktanya metode beda hingga yang ada di 2.2 rentan untuk menjadi tidak stabil. Silakan membaca referensi untuk mengetahui cara kuantifikasi kestabilan dengan beberapa metode seperti metode energi, deret fourier, matriks (e.g., Riddaway, 2001).

Untuk hasil yang didapatkan pasti stabil, dikembangkan metode baru yaitu skema implisit atau skema Crank–Nicolson. Skema implisit lebih bersifat ekonomis daripada skema eksplisit dalam arti bahwa memungkinkan langkah-langkah waktu yang jauh lebih besar daripada yang diperlukan oleh kondisi CFL. Karena skema mengasumsikan secara implisit nilai masa depan yang tidak diketahui (*unknown*), itu disebut implisit. Namun, metode implisit ini tidak dipraktekkan dalam praktikum.

3. Set up MinGW dan Home Directory

3.1 Pastikan MinGW dan software-software pendukung telah terinstall (gfortran, gnuplot, opengrads, Xming, notepad++, dan python).
3.2 Jalankan shortcut “MinGW-Shell” dalam folder C:/MinGW
3.3 Terminal yang terbuka menunjukkan Anda sedang berada di *Home Directory* dengan *path* C:/MinGW/msys/1.0/home/<nama user>/
3.4 Buka Xming masing-masing

4. Tugas Praktikum

4.1 Buat direktori “Modul_1” dalam *Home Directory* MinGW masing-masing. Unduh folder “advection” dan folder “deriv” dari Ms. Teams: files/Praktikum/Modul_1. Buka dan pelajari program DERIV, lalu *compile* dan jalankan. **Keterangan:** Program DERIV (deriv.f) menggunakan *subroutine* eksternal DDX1, DDX2 dan DDX4 (ddx1.f, ddx2.f dan ddx4.f). *Compile*-lah semua *subroutine* menjadi *object file* (*.o) terlebih dahulu dengan perintah

```
gfortran -c ddx1.f
gfortran -c ddx2.f
gfortran -c ddx4.f
```

Kemudian *compile* program utama dengan mengikutsertakan *object file* yang baru dihasilkan

```
gfortran deriv.f ddx1.o ddx2.o ddx4.o -o deriv.exe
```

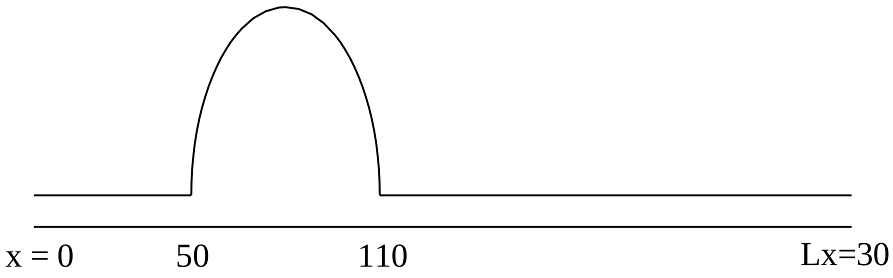
Lalu eksekusi program dengan perintah

```
./deriv.exe
```

Lakukan langkah-langkah berikut dengan *software* pengolahan data yang Anda kuasai:

a. Plot profil tekanan p(z) vs ketinggian z, bagaimanakah bentuk profil tekanan atmosfer yang muncul? (1 gambar)
b. Hitunglah nilai *error* estimasi turunan pertama (dp/dz) menggunakan skema orde pertama, kedua dan keempat. Plot semua nilai turunan (1 gambar) beserta nilai *error*-nya (1 gambar) serta analisis perbandingan antar skema!

4.2 Suatu gangguan menjalar pada ruang 1-dimensi di udara dengan kedua batasnya tertutup. Pada t = 0, gangguan tersebut berupa bentuk setengah sinusoid seperti gambar di bawah.



Persamaan model adveksi untuk gangguan di atas adalah :

$$\frac{\partial u}{\partial t} = -a\frac{\partial u}{\partial x}\text{, dengan a > 0}$$

Nilai a menunjukkan magnitudo dan arah adveksi. Konfigurasi model menggunakan a = 300 m/s, jarak antar grid Δx =5 m, panjang domain Lx=300 m, rentang waktu simulasi Lt = 0.5 s. Selain itu model memakai syarat awal sbb:
saat t = 0 s, maka:

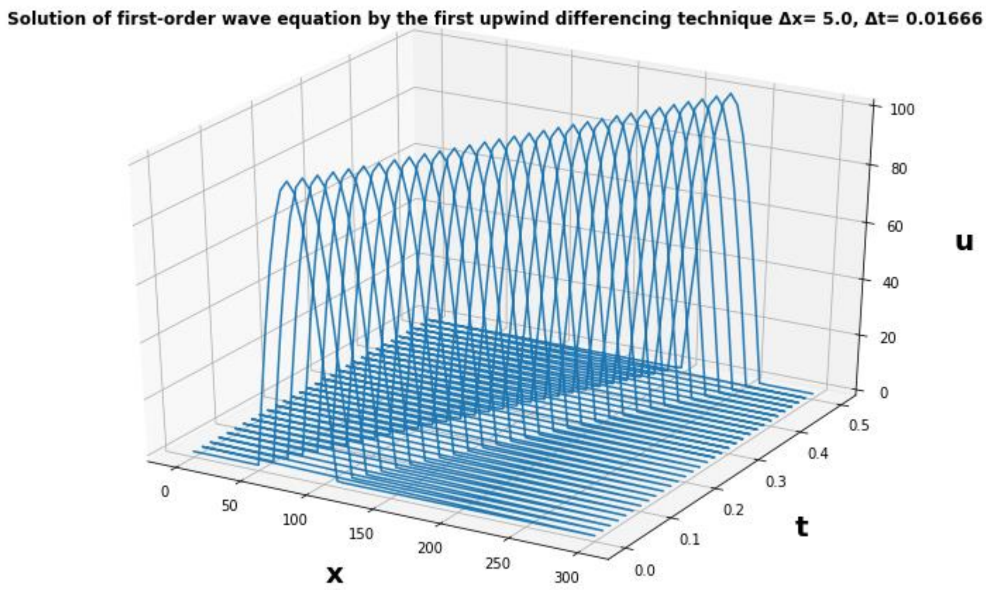
$u(x,0) = 0$ untuk $0 < x < 50$ m,
$u(x,0) = 100\left\{sin\left[\frac{\pi(x-50)}{60}\right]\right\}$ untuk $50 \leq x \leq 110$ m,
$u(x,0) = 0$ untuk $110 < x < 300$ m;

dan syarat batas sebagai berikut :
saat $t \geq 0$ s, maka:

$u(0,t) = 0$ untuk $x = 0$ m
$u(Lx,t) = 0,$ untuk $x =$ Lx

Lakukanlah pemodelan adveksi 1-dimensi dengan langkah-langkah berikut:

a. Program FTBS (ftbs.f90) adalah salah satu program untuk menghitung solusi persamaan model di atas. Program tersebut menggunakan metode *Forward-in-Time and Backward-in-Space (FTBS)*. *Compile*-lah program FTBS, kemudian jalankan. Plot hasilnya menggunakan python:
Referensi tools:
https://matplotlib.org/stable/gallery/mplot3d/index.html
(lihat contoh hasil di bawah, cari di google petunjuk untuk menambah/mengganti attribut gambar python)



b. Ubahlah langkah waktunya (Δt) sebagai berikut :

$$\Delta t = 0.01666 \rightarrow c = 0.9996 \ \approx 1$$

$$\Delta t = 0.0075 \rightarrow c = 0.45$$

$$\Delta t = 0.02 \rightarrow c = 1.2$$

Dimana $c = \frac{\Delta t}{\Delta x}$ analisis perbedaannya!

c. Buatlah program FTFS dengan menduplikasi file ftbs.f90 terlebih dahulu (beri nama ftfs.f90). Jalankan program FTFS dengan Δt = 0.01666 dan plot! Kenapa hasilnya tidak sesuai dengan FTBS? Bandingkan juga jika adveksi mengarah ke kiri dengan mengubah nilai a = -300 m/s!

d. Sama seperti langkah c, namun untuk program FTCS dan CTCS. Bandingkan hasilnya dengan FTBS dan FTFS! Jelaskan kelebihan dan kekurangan masing-masing metode.

4.3 Diberikan sebuah data meteorologi 4-dimensi dalam format grads untuk berbagai ketinggian. Potong / *dump* data angin zonal tersebut pada satu lintang yang berbeda untuk seluruh lingkar bumi pada ketinggian 500 hPa. Tiap kelompok memilih lintang yang berbeda, lalu lakukan perhitungan model adveksi seperti no 2 (hanya FTBS) dengan memakai syarat awal data angin yang telah di-dump. Jalankan model dengan a = 300 m/s dan rentang waktu simulasi Lt diubah menjadi 1 hari. Gunakan Δx yang benar sesuai dengan koordinat asli dengan rumus berikut

$$\Delta x = r\cos(\phi)\Delta\lambda$$

dimana r, ϕ, dan λ menunjukkan radius bumi (6371 km), lintang, dan bujur (dalam radian; 1 rad = 180°/π). Tentukanlah Δt sedemikian rupa sehingga Anda mendapatkan jumlah langkah waktu (nt) yang sedikit mungkin tapi masih memenuhi kestabilan numerik.

Referensi:

Hoffman. 2000. *Computational Fluid Dynamics Volume I.* Bab 2 hal 29-53 dan Bab 3 hal 60-67

Khrisnamurti T.N., dan L. Bounoua. 1996. *An Introduction to Numerical Weather Prediction Techniques.* Bab 2 hal 23-27

Khrisnamurti. 2006. *An Introduction to Global Spectral Modeling Second Revised and Enlarged Edition.* Bab 2 hal 4-11

Riddaway. 2001. *Numerical Methods,* Meteorological Training Course Lecture Series, ECMWF