


PEN

 PERPUSTAKAAN	MILIK PERPUSTAKAAN UNIVERSITAS PADJADJARAN YOGYAKARTA
Diterima	- 8 MAR 2011
Inventarisasi	: 72 / 71F / Hd 3 / Pen / 2011
Klasifikasi	: 005. 711 / Pra / m
Subyek	: Programming

Perpustakaan UAJY



0700002065

# LAPORAN PENELITIAN

**METODE BEDA HINGGA KOMPAK ORDE TINGGI UNTUK  
DISKRETISASI METODE LATTICE BOLTZMANN**



**DISUSUN OLEH:**

**Dr. Pranowo, ST., MT.  
Ag. Gatot Bintoro, ST., MT.**

**FAKULTAS TEKNOLOGI INDUSTRI  
UNIVERSITAS ATMA JAYA YOGYAKARTA  
2010**

## HALAMAN PENGESAHAN

1. a. Judul Penelitian : Metode Beda Hingga Kompak Orde Tinggi untuk Diskretisasi Metode Lattice Boltzmann
- b. Macam penelitian : Studi Literatur
2. Personalia Ketua Penelitian
  - a. Nama : Dr. Pranowo, ST., MT.
  - b. Tempat, tanggal lahir : Magelang, 03 November 1971
  - c. Jenis Kelamin : Laki-laki
  - d. NPP : 09.96.596
  - e. Jabatan : Lektor / III d  
    akademik/Golongan  
    Fakultas/Prodi : Fakultas Teknologi Industri/Teknik Informatika
3. Jumlah Anggota : 1 orang  
    Nama Anggota : Ag. Gatot Bintoro, ST., MT. NPP: 03.97.617
4. Lokasi Penelitian : Laboratorium Komputasi  
    Fakultas Teknologi Industri  
    Universitas Atma Jaya Yogyakarta
5. Jangka waktu penelitian : 6 bulan
6. Biaya yang diperlukan : Rp. 3.740.000,00

Yogyakarta, 22 Desember 2010

Mengetahui:

Dekan FTI UAJY



Ir. B. Kristyanto, M.Eng., Ph.D.  
FAKULTAS  
TEKNOLOGI INDUSTRI

Ketua Peneliti

Dr. Pranowo, ST., MT.

Menyetujui:

Ketua LPPM UAJY



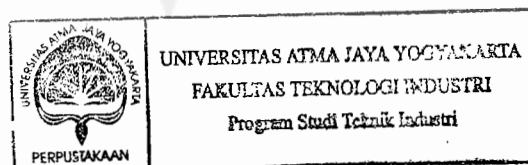
Dr. M. E. Shellyana Junaedi, M.Si.  
L P P M

## ABSTRAK

### Metode Beda Hingga Kompak Orde Tinggi untuk Diskretisasi Metode Lattice Boltzmann

Dewasa ini peran metode komputasi yang disebut *Computational Fluid Dynamic* (CFD) di bidang rekayasa semakin berkembang. CFD konvensional menggunakan persamaan Navier Stokes sebagai persamaan model untuk fluida dengan variabel makroskopis seperti kecepatan tekanan sebagai variabel yang dihitung, persamaan ini bersifat nonlinear dan kompleks sehingga relatif susah untuk dipecahkan. Alternatif lain adalah menggunakan metode Lattice Boltzmann (LB) sebagai alat komputasi untuk simulasi dinamika fluida. Metode ini berbasis pada persamaan kinetik yang diturunkan dari Teori Kinetik Gas Boltzmann yang mempunyai suku adveksi linear dan suku tumbukan (*collision*) yang bersifat nonlinear. Persamaan kinetik dari metode LB lebih sederhana dibanding dengan persamaan Navier Stokes, oleh karena itu metode LB sekarang juga banyak dikembangkan sebagai bagian dari CFD. Upaya peningkatan akurasi diskretisasi metode LB terus dilakukan, penelitian ini mengusulkan penggunaan diskretisasi ruang metode LB menggunakan skema beda hingga kompak orde tinggi dan penggunaan skema eksplisit Runge-Kutta orde-4. Analisis dispersi dan kestabilan dilakukan untuk mempelajari properti skema beda hingga kompak orde tinggi untuk suku konveksi. Berdasarkan dari kedua analisis tersebut dan pengujian dengan memecahkan permasalahan Taylor Green Vortex dapat disimpulkan bahwa: 1) skema Runge-Kutta orde-4 dapat meningkatkan kestabilan perhitungan numerik, tetapi stabilitas ini menurun seiring dengan meningkatnya orde skema beda hingga, 2) kenaikan orde akurasi perhitungan numerik seiring dengan kenaikan orde skema beda hingga

**Kata Kunci:** Beda Hingga, orde tinggi, Lattice Boltzmann



## KATA PENGANTAR

Puji dan syukur kepada Tuhan Yang Maha Esa atas berkah dan rahmat -Nya yang telah memberi kemudahan dan petunjuk kepada penulis dalam menyelesaikan penelitian ini. Pada kesempatan ini, penulis ingin menyampaikan terima kasih kepada pihak-pihak yang telah banyak memberikan bantuan dalam menyelesaikan penelitian ini dengan baik terutama kepada :

1. Dekan Fakultas Teknologi Industri, Ketua LPPM, Ketua Program Studi Teknik Industri, Kepala Laboratorium Pemodelan dan Optimasi yang telah memberikan kesempatan penelitian.
2. Rekan-rekan komunitas Laboratorium Komputasi.
3. Rekan-rekan dan patner kerja atas pengertian motivasi dan peluang serta bantuannya.
4. Rekan-rekan Staf Pengajar dan Karyawan Jurusan Teknik Informatika, Fakultas Teknologi Industri, Universitas Atma Jaya Yogyakarta.
5. Semua pihak yang telah banyak membantu, yang namanya tidak dapat disebutkan satu per satu.

Akhir kata, penulis mohon maaf atas segala kekurangan dalam penyusunan penelitian ini dan semoga penelitian ini dapat memberikan manfaat kepada semua pihak dan kepada penelitian-penelitian selanjutnya.

Yogyakarta, Desember 2010

Pranowo

## DAFTAR ISI

<b>HALAMAN PENGESAHAN</b> .....	ii
<b>ABSTRAK</b> .....	iii
<b>KATA PENGANTAR</b> .....	iv
<b>DAFTAR ISI</b> .....	v
<b>DAFTAR GAMBAR</b> .....	vi
<b>DAFTAR TABEL</b> .....	vii
<b>DAFTAR SIMBOL</b> .....	viii
<b>Bab I Pendahuluan</b> .....	1
<b>I.1. Latar Belakang</b> .....	1
<b>I.2. Perumusan Masalah</b> .....	1
<b>I.3. Tujuan Penelitian</b> .....	2
<b>Bab II Studi Literatur</b> .....	3
<b>II.1. Persamaan Model</b> .....	3
<b>II.2. Skema Beda Hingga Kompak Orde Tinggi</b> .....	4
<b>II.3. 2.3. Integrasi Waktu</b> .....	5
<b>II.4. 2.4. Diagram Alir Integrasi Waktu Runge-Kutta</b> .....	6
<b>Bab III Analisis Dispersi dan Stabilitas</b> .....	8
<b>III.1. Analisis Dispersi</b> .....	8
<b>III.2. Studi Pustaka dan Observasi</b> .....	11
<b>Bab IV Hasil Perhitungan dan Pembahasan</b> .....	14
<b>IV.1. Pengujian Akurasi</b> .....	14
<b>IV.2. Simulasi Vorteks Taylor Green</b> .....	18
<b>Bab V Kesimpulan dan Saran</b> .....	29
<b>V.1. Kesimpulan</b> .....	29
<b>DAFTAR PUSTAKA</b> .....	30
<b>LAMPIRAN A</b> .....	32
<b>LAMPIRAN B</b> .....	47

## DAFTAR GAMBAR

Gambar II.1	Kecepatan Diskret dalam model Lattice Boltzmann D2Q9 .....	4
Gambar II.2	<i>Stencil</i> Skema Beda Hingga .....	5
Gambar II.3	Diagram alir integrasi waktu Runge-Kutta.....	7
Gambar III.1	Hubungan Dispersi Numerik .....	9
Gambar III.2	Daerah batas kestabilan skema Runge-Kutta orde-4 .....	12
Gambar IV.1	Penyelesaian numerik dan eksak untuk $t=0,3$ .....	14
Gambar IV.2	Penyelesaian numerik dan eksak untuk $t=0,6$ .....	15
Gambar IV.3	Penyelesaian numerik dan eksak untuk $t=0,9$ .....	15
Gambar IV.4	Penyelesaian numerik dan eksak untuk $t=1,2$ .....	15
Gambar IV.5	Kesalahan perhitungan saat $t=0,628$ untuk CFL = 0,1.....	16
Gambar IV.6	Kesalahan perhitungan saat $t=0,628$ untuk CFL = 0,15.....	16
Gambar IV.7	Kesalahan perhitungan saat $t=0,628$ untuk CFL = 0,2.....	17
Gambar IV.8	Konvergensi Persamaan Adveksi .....	17
Gambar IV.9	Orde Akurasi berbagai metode numerik.....	18
Gambar IV.10	Vektor kecepatan saat $t = 150$ .....	20
Gambar IV.11	Kecepatan vertikal untuk seluruh domain saat $t = 10$ .....	21
Gambar IV.12	Kecepatan vertikal untuk seluruh domain saat $t = 50$ .....	21
Gambar IV.13	Kecepatan vertikal untuk seluruh domain saat $t = 100$ .....	22
Gambar IV.14	Kecepatan vertikal untuk seluruh domain saat $t = 150$ .....	22
Gambar IV.15	Kecepatan vertikal untuk seluruh domain saat $t = 200$ .....	23
Gambar IV.16	Kecepatan vertikal untuk seluruh domain saat $t = 250$ .....	23
Gambar IV.17	Distribusi kecepatan pada potongan $y = 3,14$ saat $t = 10$ .....	24
Gambar IV.18	Distribusi kecepatan pada potongan $y = 3,14$ saat $t = 50$ .....	24
Gambar IV.19	Distribusi kecepatan pada potongan $y = 3,14$ saat $t = 100$ .....	25
Gambar IV.20	Distribusi kecepatan pada potongan $y = 3,14$ saat $t = 150$ .....	25
Gambar IV.21	Logaritma galat pada potongan $y = 3,14$ saat $t = 8$ .....	26
Gambar IV.22	Logaritma galat pada potongan $y = 3,14$ saat $t = 24$ .....	26
Gambar IV.23	Logaritma galat pada potongan $y = 3,14$ saat $t = 42$ .....	27
Gambar IV.24	Konvergensi permasalahan Taylor Green Vortex.....	27

## DAFTAR TABEL

Tabel III.1 Nilai $k^* \Delta x$ maksimum .....	10
Tabel III.2 Nilai $k^* \Delta x$ dan PPW untuk toleransi galat 1 % .....	10





## DAFTAR SIMBOL

$a$	:	kecepatan perambatan adveksi
$c_s$	:	kecepatan perambatan suara
$e_i$	:	kecepatan partikel diskret
$f_i$	:	fungsi distribusi kecepatan
$f_i^{eq}$	:	fungsi distribusi keadaan seimbang
$\Delta x, \Delta y, \Delta z$	:	interval/lebar grid pada arah-x, y dan z
$\Delta t$	:	interval diskritisasi waktu (lebar grid waktu)
$IM, JM$	:	jumlah grid pada sumbu-x, sumbu-y dan sumbu-z
$i, j, k, n$	:	menyatakan grid dengan arah x,y,z dan t
$i$	:	Bilangan imajiner
$k\Delta x$	:	bilangan gelombang
$n$	:	langkah waktu
$\rho$	:	massa jenis
$p$	:	tekanan
$t$	:	waktu
$\tau$	:	faktor relaksasi
$u_j$	:	kecepatan makroskopis
$\omega$	:	frekuensi
$x, y$	:	sumbu koordinat
$\nu$	:	viskositas dinamis

## Bab I Pendahuluan

### I.1. Latar Belakang

Dewasa ini peran metode komputasi yang disebut *Computational Fluid Dynamic* (CFD) di bidang rekayasa semakin menguat. Di bidang *Aero-acoustics*, persamaan ini digunakan untuk memodelkan kebisingan (*noise*) yang diakibatkan mesin pesawat terbang. Di bidang Arsitektur, CFD digunakan untuk memodelkan aliran udara dalam ruangan yang berhubungan erat dengan masalah kenyamanan tempat tinggal. Sedangkan di bidang Teknik Mesin persamaan ini digunakan untuk rancang bangun piranti yang berhubungan dengan fluida seperti pompa, kompressor dan alat penukar panas. Karena CFD banyak memainkan peranan yang penting di bidang rekayasa, upaya pengembangan metode ini terus dikembangkan. CFD konvensional menggunakan persamaan Navier Stokes sebagai persamaan model untuk fluida dengan variabel makroskopis seperti kecepatan tekanan sebagai variabel yang dihitung, persamaan ini bersifat nonlinear dan kompleks sehingga relatif susah untuk dipecahkan. Alternatif lain adalah menggunakan metode Lattice Boltzmann (LB) sebagai alat komputasi untuk simulasi dinamika fluida. Metode ini berbasis pada persamaan kinetik yang diturunkan dari Teori Kinetik Gas Boltzmann yang mempunyai suku adveksi linear dan suku tumbukan (*collision*) yang bersifat nonlinear (Mei and Shyy, 1999). Persamaan kinetik dari metode LB lebih sederhana dibanding dengan persamaan Navier Stokes, oleh karena itu metode LB sekarang juga banyak dikembangkan sebagai bagian dari CFD.

### I.2. Perumusan Masalah

Metode LB menggunakan *mesh* yang seragam dan berbentuk bujur sangkar untuk kasus 2-dimensi dan kubus untuk kasus 3-dimensi. Dengan bentuk mesh tersebut suku adveksi dapat dihitung secara Lagrangian menggunakan mesh sehingga hasilnya eksak dan suku tumbukan dihitung dengan pendekatan BGK

(Bhatnagar, Gross and Krook). Karena menggunakan mesh seperti itu maka metode LB dianggap kurang fleksibel untuk pemecahan kasus di dunia nyata. Untuk itu usaha perbaikan agar metode LB dapat menggunakan mesh yang tidak seragam terus dikembangkan. Metode-metode yang diusulkan adalah berbasis metode Eulerian, yaitu: metode Beda Hingga (Mei and Shyy, 1999; Shi, Zhao and Guo, 2003), Volume Hingga (Peng dkk, 1999; Ubertini and Succi, 2006; Patil dan Lakshmish, 2009) dan metode Elemen Hingga (Lee and Lin, 2001; 2003). Karena termasuk metode Eulerian dan menggunakan diskretisasi berorde rendah maka metode-metode tersebut mempunyai galat dispersi yang cukup tinggi. Oleh karena itu, penelitian ini mengusulkan pengembangan diskretisasi metode Beda Hingga kompak orde tinggi (Svard and Jan Nordstrom, 2008; Laizet and Lamballais, 2009; Shah dkk, 2010) untuk diskretisasi suku adveksi metode lattice Boltzmann.

### **1.3. Tujuan Penelitian**

Tujuan penelitian ini adalah mengupayakan peningkatan akurasi dengan menggunakan skema beda hingga kompak orde tinggi pada suku adveksi metode Lattice Boltzmann.

## Bab II Studi Literatur

### II.1. Persamaan Model

Persamaan model adalah persamaan Boltzmann dengan pendekatan suku tumbukan BGK (Guo and Zhao, 2003):

$$\frac{\partial f_i}{\partial t} + \mathbf{e}_i \cdot \nabla f_i = -\frac{(f_i - f_i^{eq})}{\tau} \quad (\text{II.1})$$

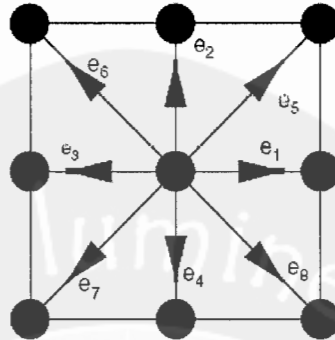
Variabel  $\tau$  adalah faktor relaksasi,  $f_i$  adalah fungsi distribusi,  $f_i^{eq}$  adalah fungsi distribusi dalam keadaan seimbang,  $i = 0, 1, \dots, 8$ , dan  $\mathbf{e}_i$  adalah kecepatan diskret yang mempunyai nilai (Gambar II.1):

$$\mathbf{e}_i = \begin{cases} (0, 0) & , i = 0 \\ (\cos \theta_i, \sin \theta_i) & , \theta_i = (i-1)\frac{\pi}{4}, i = 1, 2, 3, 4 \\ \sqrt{2}(\cos \theta_i, \sin \theta_i) & , \theta_i = (i-1)\frac{\pi}{4}, i = 5, 6, 7, 8 \end{cases} \quad (\text{II.2})$$

$$\rho = \sum_{i=0}^8 f_i \ ; \ \rho u_j = \sum_{i=0}^8 f_i e_{ij} \ ; \ f_i^{eq} = \omega_i \left( \frac{e_i \cdot u}{c_s^2} + \frac{(e_i \cdot u)^2}{2c_s^4} - \frac{|u|^2}{2} \right) \quad (\text{II.3})$$

$\rho$  adalah densitas,  $u_j$  kecepatan makroskopis,  $\omega_i$  adalah koefisien bobot dan  $c_s = 1/\sqrt{3}$ , hubungan antara faktor relaksasi dan viskositas dinamis:

$$\nu = \frac{1}{3}(\tau - 0.5) \frac{\Delta x^2}{\Delta t} .$$



Gambar II.1 Kecepatan Diskret dalam model Lattice Boltzmann D2Q9

## II.2. Skema Beda Hingga Kompak Orde Tinggi

Penggunaan orde-2 untuk diskritisasi ruang pada suku adveksi persamaan (II.1) menyebabkan akurasi berkurang untuk menganalisis fungsi yang tidak *smooth*. Untuk itu skema berorde tinggi digunakan, skema beda hingga orde 4 konvensional melibatkan grid/titik perhitungan (*stencil*) yang panjang sehingga menyebabkan kesulitan pada syarat batas. Untuk itu skema beda hingga kompak orde tinggi diusulkan untuk digunakan, skema ini pada dasarnya adalah skema Pade (Hirsch, 1988). Skema beda hingga orde 4 kompak pada untuk turunan pertama  $\left(\frac{\partial f}{\partial x}\right)$  yang hanya melibatkan 3 titik perhitungan (Gambar II.2) adalah seperti berikut (Ghrist, 2000):

$$\frac{1}{6}\left(\frac{\partial f}{\partial x}\right)_{i+1,j} + \frac{2}{3}\left(\frac{\partial f}{\partial x}\right)_{i,j} + \frac{1}{6}\left(\frac{\partial f}{\partial x}\right)_{i-1,j} = \frac{f_{i+1,j} - f_{i-1,j}}{2\Delta x} \quad (\text{II.4})$$

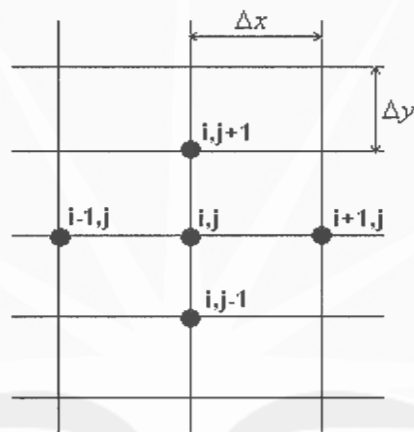
Turunan pertama  $\left(\frac{\partial f}{\partial x}\right)$  di atas tidak bisa langsung diketahui nilainya di setiap grid, untuk mengetahuinya harus menyelesaikan sistem persamaan linear tridiagonal yang terbentuk dari persamaan (II.4). Turunan pertama  $\left(\frac{\partial f}{\partial y}\right)$  dihitung dengan cara yang sama:

$$\frac{1}{6}\left(\frac{\partial f}{\partial y}\right)_{i,j+1} + \frac{2}{3}\left(\frac{\partial f}{\partial y}\right)_{i,j} + \frac{1}{6}\left(\frac{\partial f}{\partial y}\right)_{i,j-1} = \frac{f_{i,j+1} - f_{i,j-1}}{2\Delta y} \quad (\text{II.5})$$

Peningkatan orde lebih dari 4 pada skema Beda Hingga kompak akan dapat menekan galat numerik dengan cepat, tetapi stencil yang digunakan semakin panjang sehingga penambahan orde menyebabkan kesulitan pada kondisi batas. Contoh skema Beda Hingga kompak orde-6 dan 8 dapat dilihat pada rumus berikut:

$$\frac{1}{5} \left( \frac{\partial f}{\partial x} \right)_{i+1,j} + \frac{3}{5} \left( \frac{\partial f}{\partial x} \right)_{i,j} + \frac{1}{5} \left( \frac{\partial f}{\partial x} \right)_{i-1,j} = \frac{7f_{i+1,j} - 7f_{i-1,j}}{15\Delta x} + \frac{f_{i+2,j} - f_{i-2,j}}{60\Delta x}$$

$$\frac{3}{14} \left( \frac{\partial f}{\partial x} \right)_{i+1,j} + \frac{8}{14} \left( \frac{\partial f}{\partial x} \right)_{i,j} + \frac{3}{14} \left( \frac{\partial f}{\partial x} \right)_{i-1,j} = \frac{25f_{i+1,j} - 25f_{i-1,j}}{56\Delta x} + \frac{f_{i+2,j} - f_{i-2,j}}{35\Delta x} + \frac{f_{i+3,j} - f_{i-3,j}}{840\Delta x} \quad (II.6)$$



Gambar II.2 Stencil Skema Beda Hingga

### II.3. 2.3. Integrasi Waktu

Setelah skema Beda Hingga digunakan untuk diskritisasi suku adveksi persamaan (II.1), maka persamaan (II.1) akan berbentuk semi diskret. Persamaan semi diskret tersebut perlu diintegrasikan menurut waktu. Metode eksplisit biasanya digunakan untuk integrasi waktu, tetapi penelitian ini menggunakan integrasi waktu metode eksplisit Runge-Kutta (RK) orde-4 karena lebih akurat dan batas kestabilan lebih luas. Untuk itu persamaan (II.1) sedikit dimodifikasi menjadi:

$$\begin{aligned}\frac{\partial f_i}{\partial t} &= L(f_i) \\ &= -\mathbf{e}_i \cdot \nabla f_i - \frac{(f_i - f_i^{eq})}{\tau}\end{aligned}\tag{II.7}$$

Integrasi waktu persamaan di atas dengan metode eksplisit Euler adalah seperti berikut:

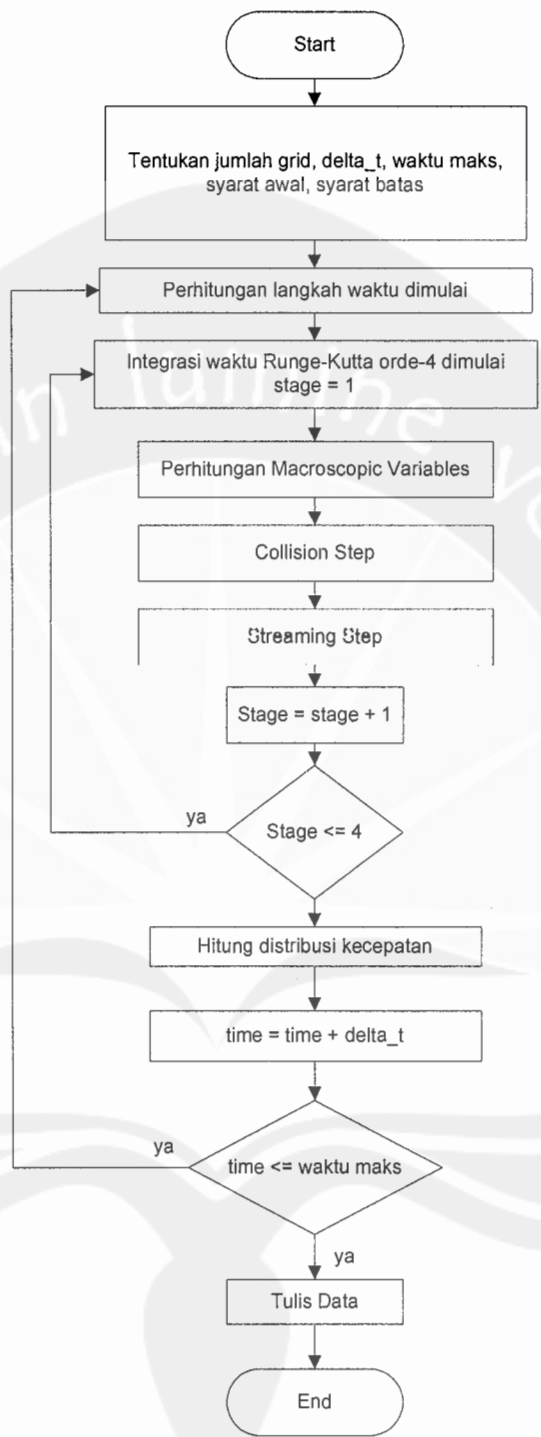
$$\begin{aligned}f_i^{n+1} &= f_i^n - \Delta t L(f_i^n) \\ &= f_i^{n+1} - \Delta t \left( \mathbf{e}_i \cdot \nabla f_i^n - \frac{(f_i - f_i^{eq})^n}{\tau} \right)\end{aligned}\tag{II.8}$$

Notasi  $n$  menunjukkan urutan tahap waktu, sedangkan  $\Delta t$  menunjukkan lebar langkah waktu (*time step*). Integrasi waktu menggunakan metode eksplisit Runge-Kutta (RK) orde-4 mempunyai 4 tahap, yaitu:

$$\begin{aligned}k_1 &= L(f_i^n, t^n) \\ k_2 &= L\left(f_i^n + \frac{1}{2} \Delta t k_1, t^n + \frac{1}{2} \Delta t\right) \\ k_3 &= L\left(f_i^n + \frac{1}{2} \Delta t k_2, t^n + \frac{1}{2} \Delta t\right) \\ k_4 &= L(f_i^n + \Delta t k_3, t^n + \Delta t) \\ f_i^{n+1} &= f_i^n + \frac{1}{6} \Delta t (k_1 + 2k_2 + 2k_3 + k_4)\end{aligned}\tag{II.9}$$

#### II.4. 2.4. Diagram Alir Integrasi Waktu Runge-Kutta

Algoritma skema numeris yang dijabarkan di atas kemudian dituangkan ke dalam program komputer menggunakan Matlab. Adapun diagram alirnya adalah seperti Gambar II.3.



Gambar II.3 Diagram alir integrasi waktu Runge-Kutta



## Bab III Analisis Dispersi dan Stabilitas

Analisis dispersi dan stabilitas merupakan analisis yang penting untuk menentukan parameter numeris perambatan gelombang. Bab ini memberikan analisis tersebut untuk metode Beda Hingga implisit orde tinggi dengan integrasi waktu Runge-Kutta orde-4.

### III.1. Analisis Dispersi

Persamaan adveksi linear 1-dimensi (III.1) dipakai sebagai contoh persamaan model gelombang:

$$\begin{aligned} \frac{\partial u}{\partial t} + a \frac{\partial u}{\partial x} &= 0, \quad x \in [0, 2\pi] \\ u(x, 0) &= e^{ikx} \end{aligned} \quad (\text{III.1})$$

Jawaban eksak dari persamaan di atas adalah :  $u(x, t) = e^{i(kx - \omega t)}$  dimana  $k$  adalah bilangan gelombang eksak dan adalah frekuensi. Dengan menyisipkan jawaban eksak ke dalam persamaan (III.1) diperoleh hubungan dispersi eksak  $\omega = ak$ . Jawaban numerik pada setiap grid diasumsikan adalah  $u^i(x, t) = U^i e^{i(k^* x^i - \omega t)}$ , dimana  $k^*$  adalah bilangan gelombang numerik, dengan menyisipkan jawaban numerik ini ke dalam persamaan (III.1) diperoleh hubungan dispersi numerik untuk berbagai macam metode numerik seperti berikut:

1. Metode FD eksplisit orde-2 :

$$ik^* \Delta x = i \sin(k \Delta x) \quad (\text{III.2})$$

2. Metode FD kompak orde-4 :

$$ik^* \Delta x = i \left( \frac{1.5 \sin(k \Delta x)}{1 + \frac{2}{4} \cos(k \Delta x)} \right) \quad (\text{III.3})$$

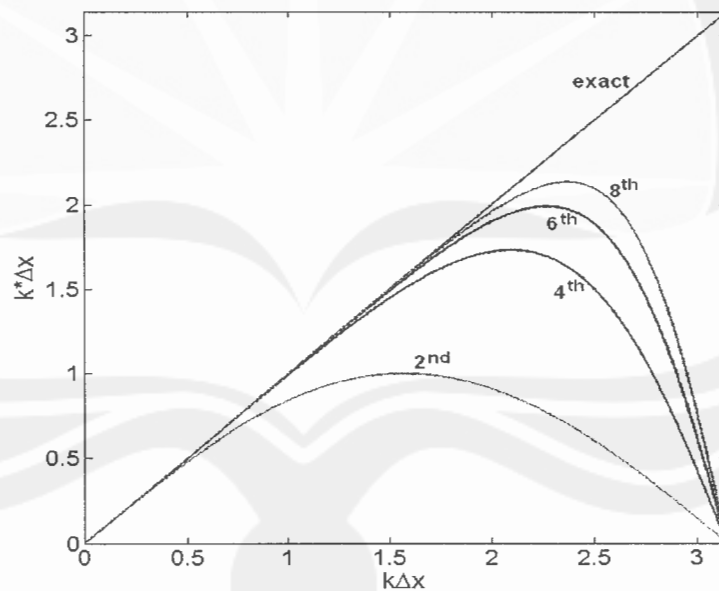
3. Metode FD kompak orde-6 :

$$ik^* \Delta x = i \left( \frac{\left(\frac{14}{9}\right) \sin(k\Delta x) + \left(\frac{1}{18}\right) \sin(2k\Delta x)}{1 + \frac{2}{3} \cos(k\Delta x)} \right) \quad (\text{III.4})$$

4. Metode FD kompak orde-8 :

$$ik^* \Delta x = i \left( \frac{\left(\frac{175}{112}\right) \sin(k\Delta x) + \left(\frac{1}{10}\right) \sin(2k\Delta x) + \left(\frac{1}{240}\right) \sin(2k\Delta x)}{1 + \frac{3}{4} \cos(k\Delta x)} \right) \quad (\text{III.5})$$

Berdasarkan persamaan (III.2) - (III.5) di atas maka dapat diperoleh gambar hubungan dispersi numerik seperti berikut:



Gambar III.1 Hubungan Dispersi Numerik

Hubungan dispersi di atas tidak menunjukkan tidak adanya bagian yang bernilai imajiner, sehingga dengan demikian diskretisasi numeriknya bersifat konservatif

tanpa disertai galat dispansi. Nilai maksimum untuk  $k^* \Delta x$  untuk berbagai metode numerik adalah:

Tabel III.1. Nilai  $k^* \Delta x$  maksimum

No	Metode Numerik	Max( $k^* \Delta x$ )
1	Explicit 2 <sup>nd</sup>	1,000
2	Compact 4 <sup>th</sup>	1,732
3	Compact 6 <sup>th</sup>	1,989
4	Compact 8 <sup>th</sup>	2,133

Galat numerik dapat diperoleh dari selisih dari dispersi eksak dikurangi numerik, seperti berikut:

$$e = |k^* \Delta x - k \Delta x| \quad (\text{III.6})$$

Untuk galat numerik yang dapat diterima sebesar 1 % , maka kita peroleh nilai  $k^* \Delta x$  dan jumlah grid setiap panjang gelombang (*point per wavelength-PPW*) untuk berbagai metode numerik seperti berikut:

Tabel III.2 Nilai  $k^* \Delta x$  dan PPW untuk toleransi galat 1 %

No	Metode Numerik	$k^* \Delta x$	$PPW = 2\pi/k^* \Delta x$
1	Explicit 2 <sup>nd</sup>	0,382	16,419
2	Compact 4 <sup>th</sup>	1,089	5,768
3	Compact 6 <sup>th</sup>	1,482	4,240
4	Compact 8 <sup>th</sup>	1,718	3,658

Dengan melihat Tabel III.2 dapat diketahui bahwa semakin tinggi orde metode numerik maka jumlah grid untuk setiap panjang gelombang (*PPW*) semakin sedikit. Hal ini menunjukkan bahwa metode numerik orde tinggi memerlukan jumlah grid yang lebih sedikit dibanding metode orde rendah untuk toleransi galat dispersi yang sama.

### III.2. Studi Pustaka dan Observasi

Kestabilan perhitungan numerik dianalisis menggunakan metode Von Neumann (keterangan rinci dapat dilihat di Hirsch, 2007). Persamaan adveksi dipakai sebagai model persamaan:

$$\frac{\partial f_i}{\partial t} = L(f_i) \quad (III.7)$$

dimana  $L$  adalah suku sisa yang mengandung turunan ruang persamaan atur. Dengan menyisipkan  $f_i = \bar{f}_i(t)e^{kx}$  ke dalam persamaan (III.4), maka kita peroleh”

$$\frac{\partial \bar{f}_i}{\partial t} = \lambda \bar{f}_i \quad (III.8)$$

dimana  $\bar{f}$  adalah koefisien Fourier,  $i = \sqrt{-1}$ , dan  $\lambda$  bilangan kompleks. Suku bagian kiri persamaan (III.5) diekspansi menggunakan skema eksplisit Runge-Kutta orde-4 untuk memperoleh faktor amplifikasi:

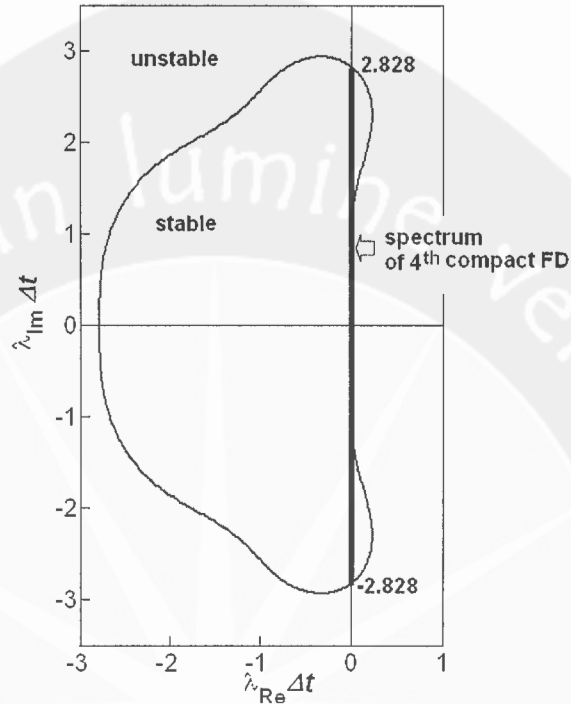
$$G = \frac{\bar{f}_i^{n+1}}{\bar{f}_i^n} = \left( 1 + \lambda \Delta t + \frac{1}{2}(\lambda \Delta t)^2 + \frac{1}{6}(\lambda \Delta t)^3 + \frac{1}{24}(\lambda \Delta t)^4 \right) \quad (III.9)$$

Syarat kestabilan numerik mengharuskan faktor amplifikasi harus terbatas nilainya.

$$|G| \leq 1$$

(III.10)

Daerah batas ketabilan dapat dilihat pada Gambar III.2:



Gambar III.2 Daerah batas kestabilan skema Runge-Kutta orde-4

Berdasar Gambar III.2 diperoleh bahwa nilai batas ketabilan skema Runge-Kutta orde-4 adalah pada nilai eigenvalue = 2,828.

Syarat kestabilan metode numerik beda hingga dengan integrasi waktu skema Runge-Kutta orde-4 dapat diperoleh dengan menggunakan batas amplifikasi faktor dan nilai  $k^* \Delta x$  maksimum seperti berikut:

1. Metode FD eksplisit orde-2

$$CFL = \frac{a\Delta t}{\Delta x} \leq 2,828/1,000$$

$$CFL \leq 2,828$$

(III.11)

2. Metode FD kompak orde-4

$$CFL = \frac{a\Delta t}{\Delta x} \leq 2,828/1,732 \quad (\text{III.12})$$
$$CFL \leq 1,6328$$

3. Metode FD kompak orde-6

$$CFL = \frac{a\Delta t}{\Delta x} \leq 2,828/1,989 \quad (\text{III.13})$$
$$CFL \leq 1,422$$

4. Metode FD kompak orde-8

$$CFL = \frac{a\Delta t}{\Delta x} \leq 2,828/2,133 \quad (\text{III.14})$$
$$CFL \leq 1,326$$

*CFL* adalah bilangan Courant Friedrichs Lewy. Dari persamaan (III.11) – (III.14) dapat diketahui bahwa semakin tinggi orde metode numerik maka batas kestabilannya semakin lebih rendah.

## Bab IV Hasil Perhitungan dan Pembahasan

### IV.1. Pengujian Akurasi

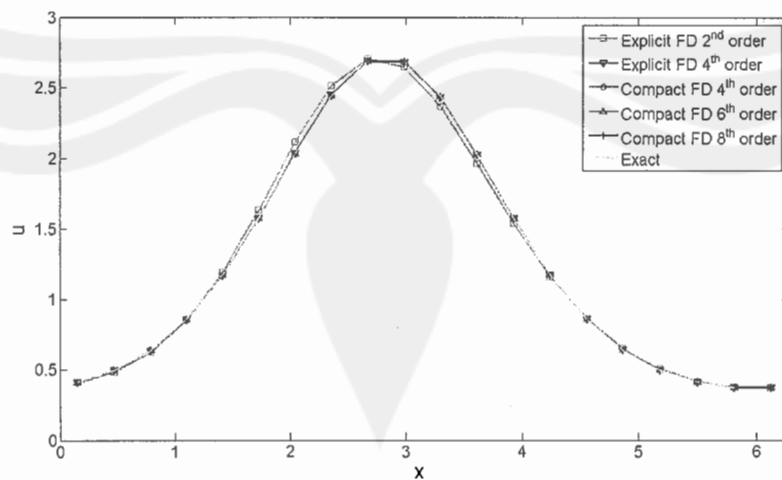
Sebelum diterapkan untuk diskretisasi LBM metode Bada Hingga kompak orde tinggi dipakai untuk diskretisasi ruang untuk persamaan adveksi linear satu dimensi untuk menguji akurasi metode FD implisit beda hinga orde tinggi:

$$\begin{aligned} \frac{\partial u}{\partial t} + 2\pi \frac{\partial u}{\partial x} &= 0, \quad x \in [0, 2\pi] \\ u(x, 0) &= e^{\sin(x)} \end{aligned} \quad (\text{IV.1})$$

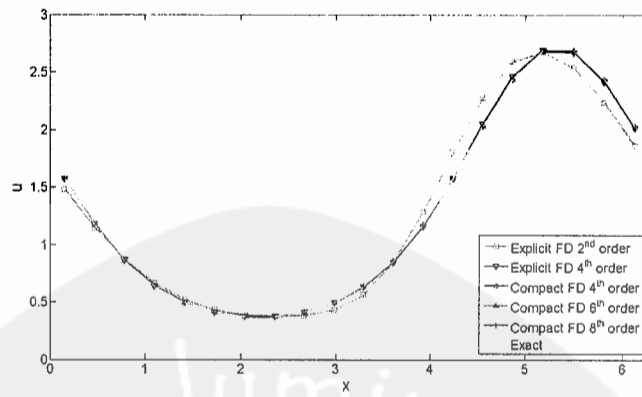
Syarat batas adalah periodis dan jawaban eksaknya adalah:

$$u(x, t) = e^{\sin(x-2\pi t)} \quad (\text{IV.2})$$

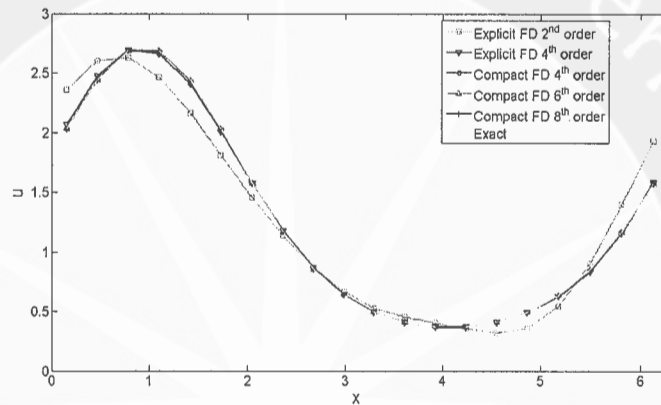
Seluruh perhitungan menggunakan langkah waktu konstan  $\Delta t = 5e-3$ , Gambar IV.1 - IV.4 menunjukkan perambatan gelombang hasil perhitungan berbagai metode beda hinga berikut jawaban eksak untuk  $CFL = 0,1$ .



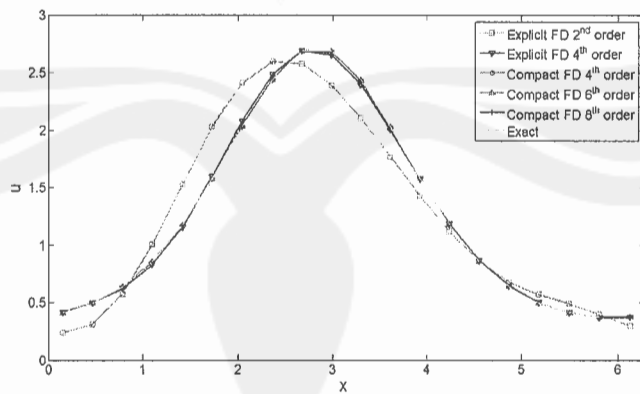
Gambar IV.1 Penyelesaian numerik dan eksak untuk  $t=0,3$



Gambar IV.2 Penyelesaian numerik dan eksak untuk  $t=0,6$



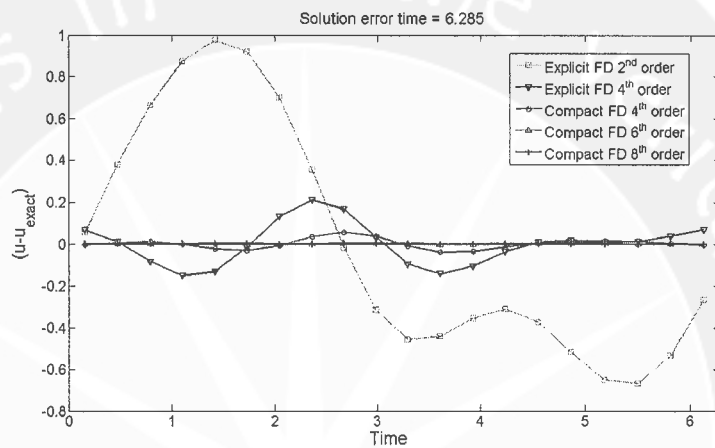
Gambar IV.3 Penyelesaian numerik dan eksak untuk  $t=0,9$



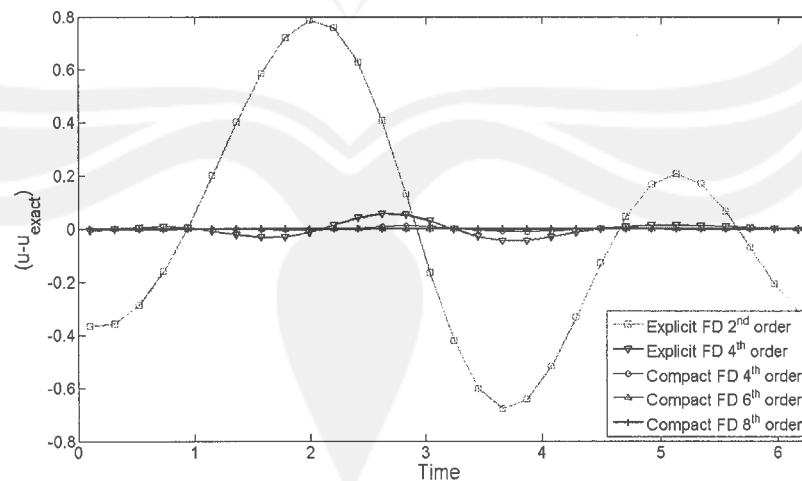
Gambar IV.4 Penyelesaian numerik dan eksak untuk  $t=1,2$



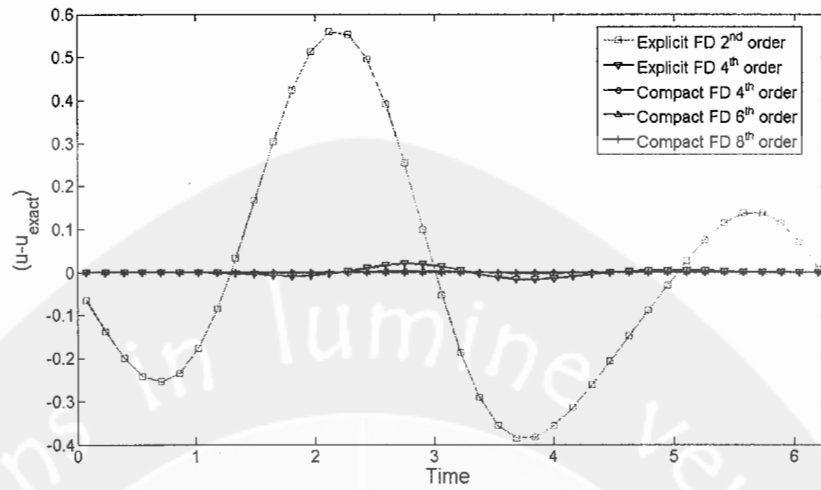
Berdasar gambar Gambar IV.1 - IV.4 terlihat perambatan sinyal dari kiri ke kanan secara periodik, selain hal itu juga dapat dilihat bahwa kesalahan perhitungan paling besar adalah perhitungan metode eksplisit Benda Hingga orde-2. Hal ini diperkuat oleh gambar-gambar IV.5 - IV.7 yang menunjukkan kesalahan perhitungan untuk berbagai bilangan  $CFL$  ( $\Delta x = 0.0982, \Delta x = 0.0491, \Delta x = 0.024$ ).



Gambar IV.5 Kesalahan perhitungan saat  $t=0,628$  untuk  $CFL = 0,1$

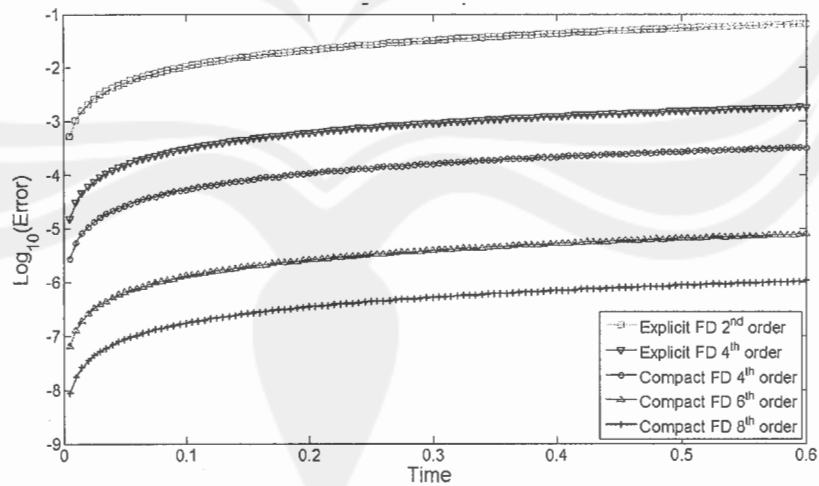


Gambar IV.6 Kesalahan perhitungan saat  $t=0,628$  untuk  $CFL = 0,15$



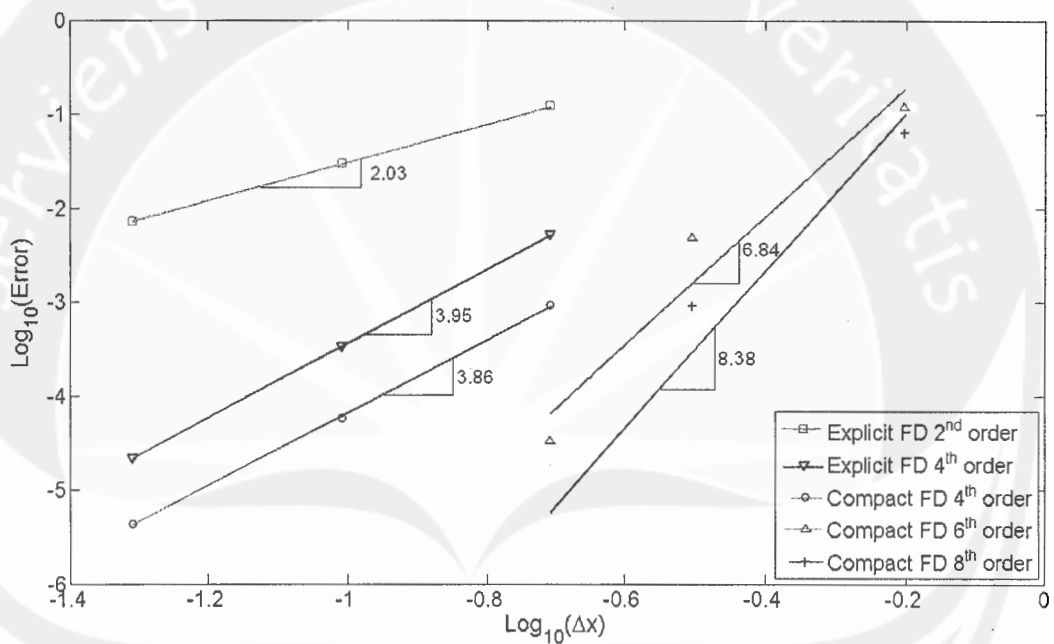
Gambar IV.7 Kesalahan perhitungan saat  $t=0,628$  untuk  $CFL = 0,2$

Gambar IV.8 menunjukkan sejarah konvergensi perhitungan numerik untuk metode Beda Hingga untuk berbagai orde. Urutan tingkat akurasi berdasar gambar-gambar adalah metode Beda Hingga eksplisit orde 2, eksplisit orde 4, compact orde 4, compact orde 6 dan compact orde 8, hal ini sudah sesuai dengan teori.



Gambar IV.8 Konvergensi Persamaan Adveksi

Ekperimen numerik juga menunjukkan orde akurasi untuk metode Benda Hingga sudah mendekati teorinya, hal ini dapat dilihat pada Gambar IV.9. Perhitungan orde akurasi untuk metode numerik orde tinggi perlu dilakukan dengan hati-hati, pengaruh galat pembulatan (*round-off error*) mempunyai pengaruh besar. Semakin tinggi orde akurasi metode numerik dan semakin kecil  $\Delta t$  atau  $\Delta x$  maka perhitungan orde akurasi semakin rentan terhadap galat pembulatan, orde akurasi bernilai lebih rendah dibanding teorinya.



Gambar IV.9 Orde Akurasi berbagai metode numerik

## IV.2. Simulasi Vorteks Taylor Green

Setelah orde akurasi metode numerik Benda Hingga untuk berbagai orde telah dicek melalui ekperimen numerik maka langkah selanjutnya adalah menguji metode numerik tersebut untuk menyelesaikan persamaan Boltzmann dalam bentuk diskret. Dalam hal ini simulasi peluruhan vorteks Taylor-Green diambil sebagai contoh kasus.

Persoalan peluruhan vorteks Taylor-Green mempunyai jawaban eksak seperti berikut:

$$u_x(x, y, t) = -U_0 \cos(k_x x) \sin(k_y y) e^{-(k_x + k_y)^2 t}$$

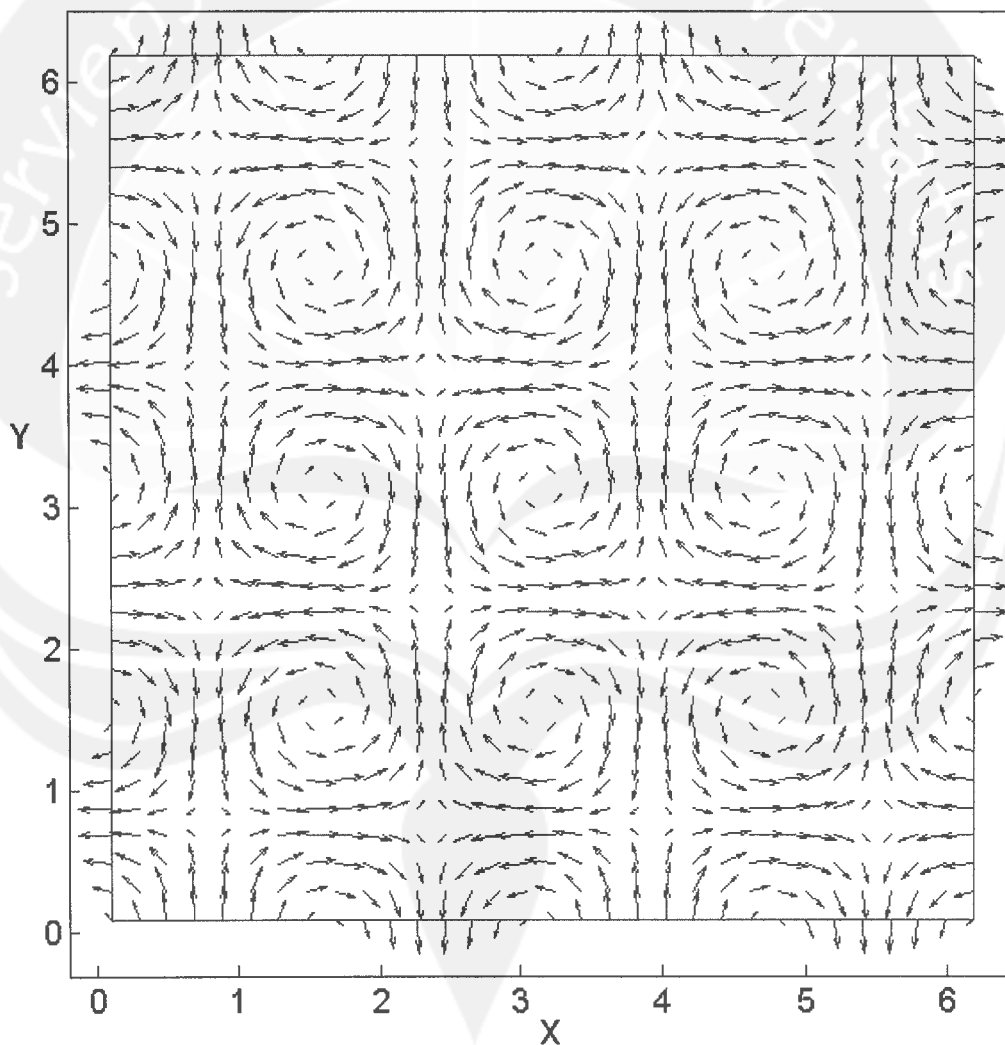
$$u_y(x, y, t) = \frac{k_x}{k_y} U_0 \cos(k_y y) \sin(k_x x) e^{-(k_x + k_y)^2 t}$$

$$p(x, y, t) = -\frac{U_0^2}{4} \left( \cos(2k_x x) + \left( \frac{k_x}{k_y} \right)^2 \sin(2k_y y) \right) e^{-(k_x + k_y)^2 t} - 1$$

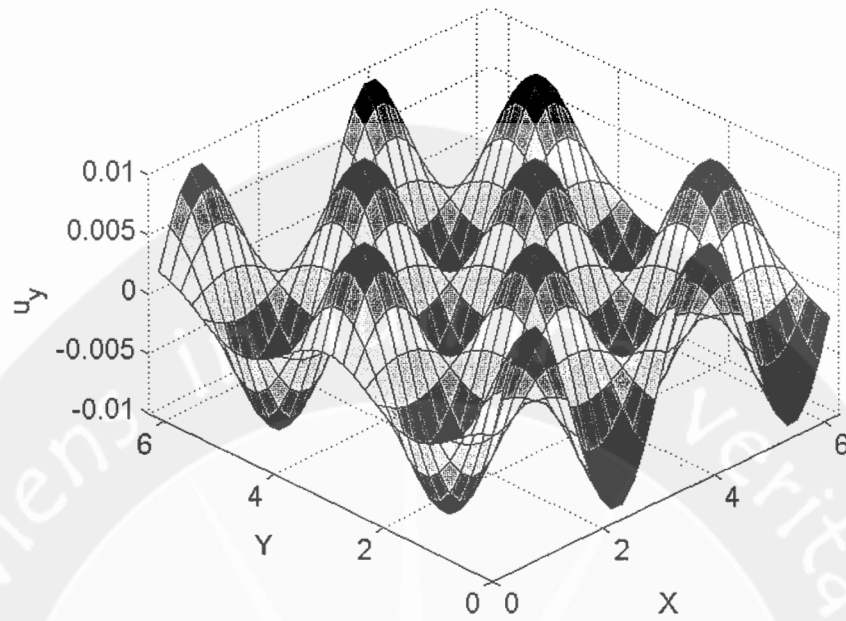
$U_0$  menyatakan amplitudo kecepatan awal,  $k_x$  and  $k_y$  adalah bilangan gelombang pada arah koordinat  $x$  dan  $y$ . Domain dibagi menjadi sejumlah elemen dengan jumlah grid  $32 \times 32$  dan syarat batas periodik digunakan pada semua batas. Parameter simulasi yang digunakan adalah  $U_0 = 0,01$ ,  $k_x = k_y = 2$ ,  $\Delta t = 0,005$ ,  $\tau = 0,0018$  dan  $CFL = 0,101$ . Kondisi awal untuk fungsi distribusi kecepatan  $f_i$  sebenarnya tidak diketahui. Cara menentukan syarat awal  $f_i$  yang konsisten tidak mudah dilakukan dan sampai sekarang penelitian tentang hal tersebut masih terus dilakukan (Mei dkk, 2006). Pada penelitian, syarat awal  $f_i$  ditentukan dengan cara yang sederhana yaitu menggunakan fungsi distribusi kecepatan pada kondisi keseimbangan  $f_i^{eq}$  sebagai syarat awal.

Fenomena peluruhan vorteks Taylor-Green merupakan peristiwa yang menggambarkan sejumlah pusaran (*vortex*) yang amplitudonya berkurang secara eksponensial terhadap waktu. Gambar vektor kecepatan vektor pusaran tersebut diperlihatkan oleh gambar IV.10, karena arah vektor pusaran adalah tetap dan yang berkurang adalah panjang vektornya maka gambar tersebut cukup jelas untuk menggambarkan kecepatan pusaran. Perubahan amplitudo pusaran dapat dilihat dari urutan gambar IV.11 - IV.16 dan gambar IV.17 - IV.20. Gambar IV.11 - IV.16 menunjukkan perubahan kecepatan vertikal ( $u_y$ ) untuk seluruh domain. Sedangkan gambar IV.17 - IV.20 menunjukkan perubahan kecepatan vertikal ( $u_y$ ) pada potongan  $y = 3,14$  untuk perhitungan numerik metode

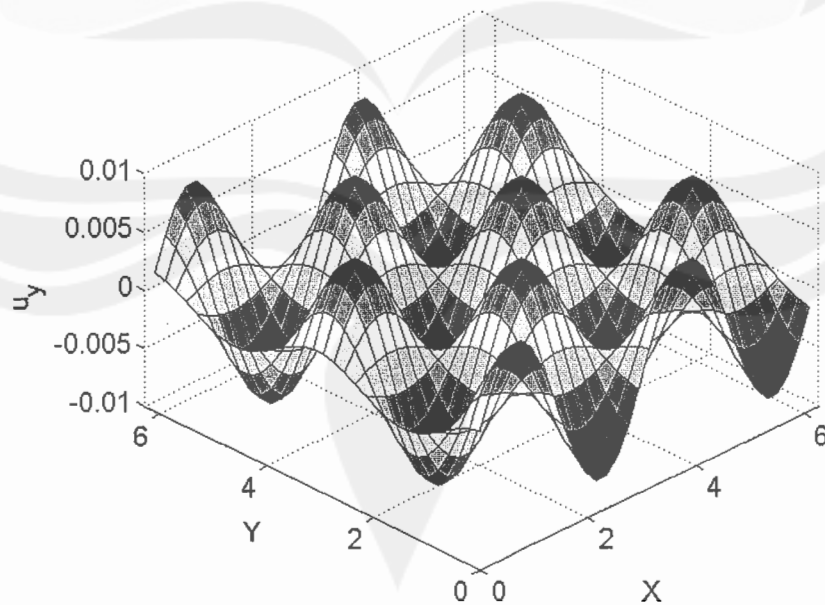
implisit orde-4 dan jawaban eksak beserta perbandingan dengan jawaban eksak, hasil perhitungan numerik sangat dekat dengan jawaban eksak sehingga grafiknya berhimpitan. Perbandingan galat perhitungan numerik untuk berbagai metode numerik dapat dilihat pada Gambar IV.21 - IV.23, berdasarkan gambar-gambar tersebut dapat diketahui bahwa metode eksplisit orde-2 mempunyai akurasi paling rendah sedangkan metode kompak orde-8 mempunyai akurasi paling tinggi. Sehingga dapat disimpulkan bahwa semakin tinggi orde metode hingga maka akurasinya semakin tinggi juga.



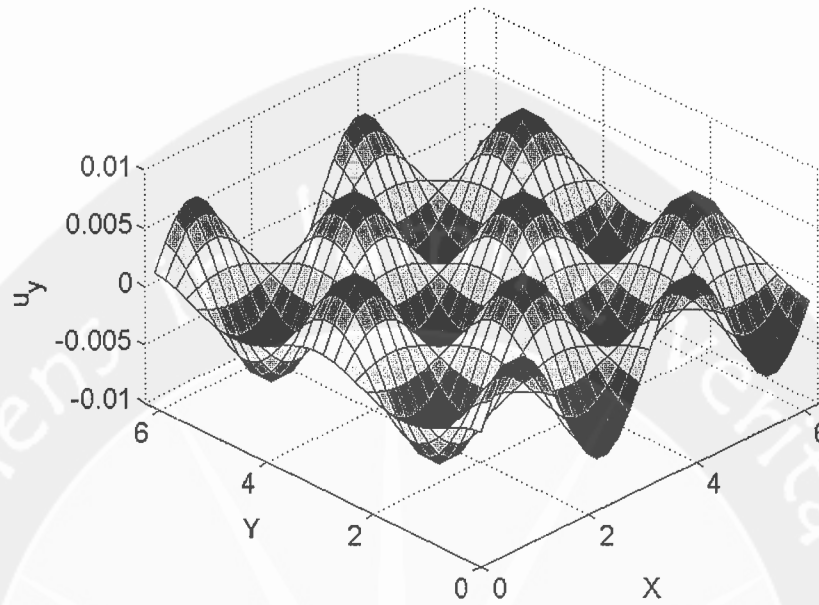
Gambar IV.10 Vektor kecepatan saat  $t = 150$



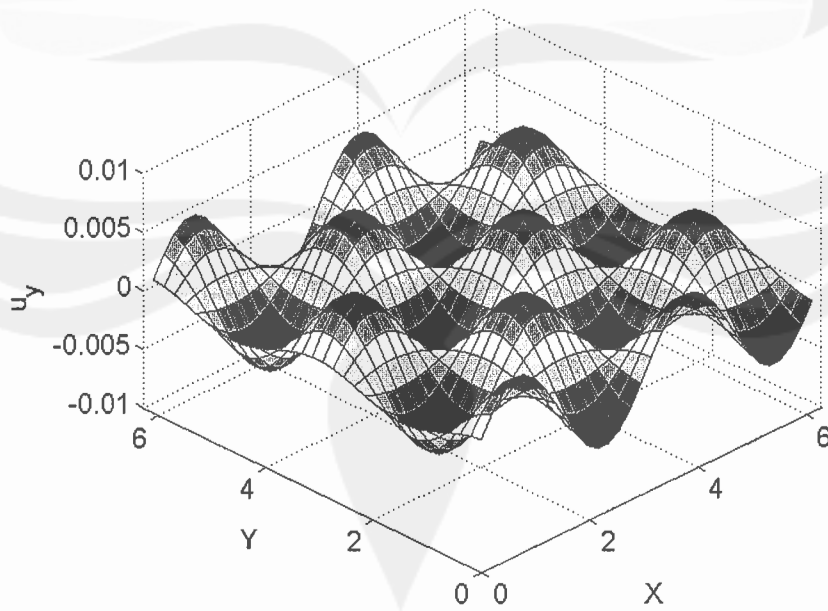
Gambar IV.11 Kecepatan vertikal untuk seluruh domain saat  $t = 10$



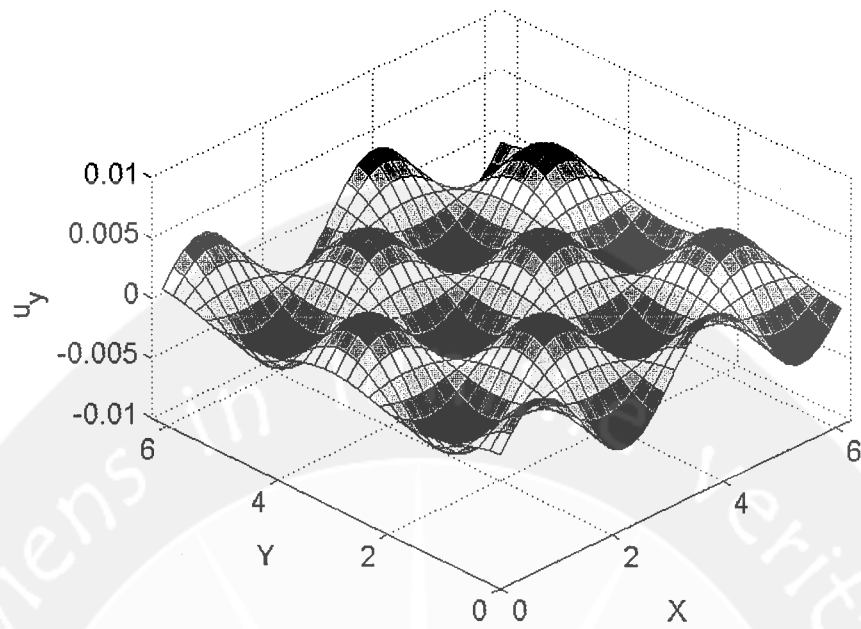
Gambar IV.12 Kecepatan vertikal untuk seluruh domain saat  $t = 50$



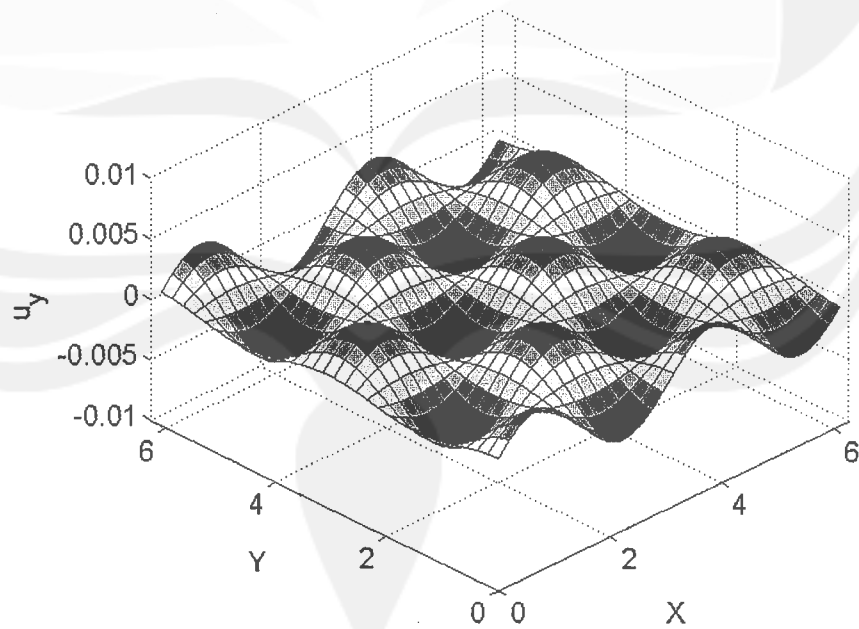
Gambar IV.13 Kecepatan vertikal untuk seluruh domain saat  $t = 100$



Gambar IV.14. Kecepatan vertikal untuk seluruh domain saat  $t = 150$

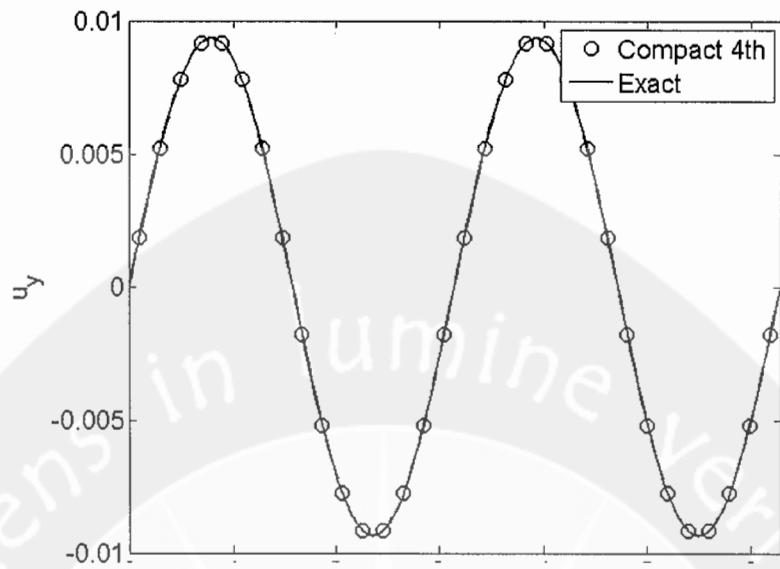


Gambar IV.15. Kecepatan vertikal untuk seluruh domain saat  $t = 200$

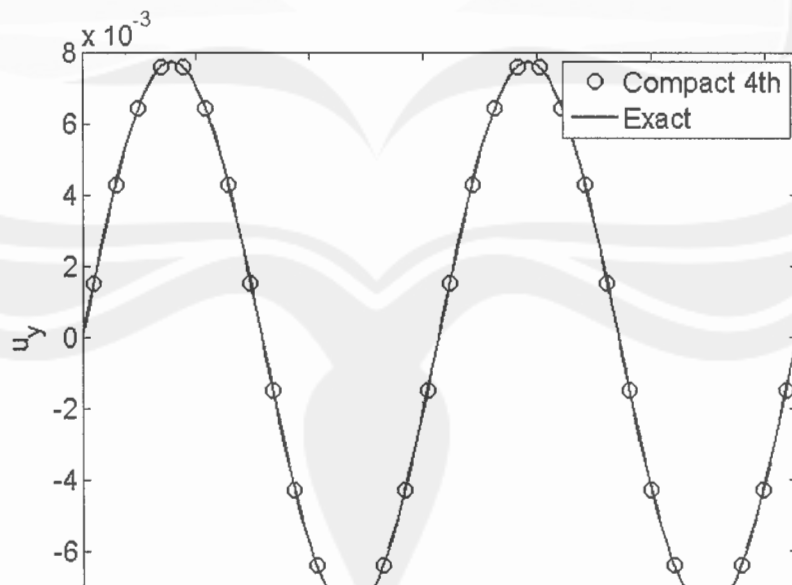


Gambar IV.16. Kecepatan vertikal untuk seluruh domain saat  $t = 250$

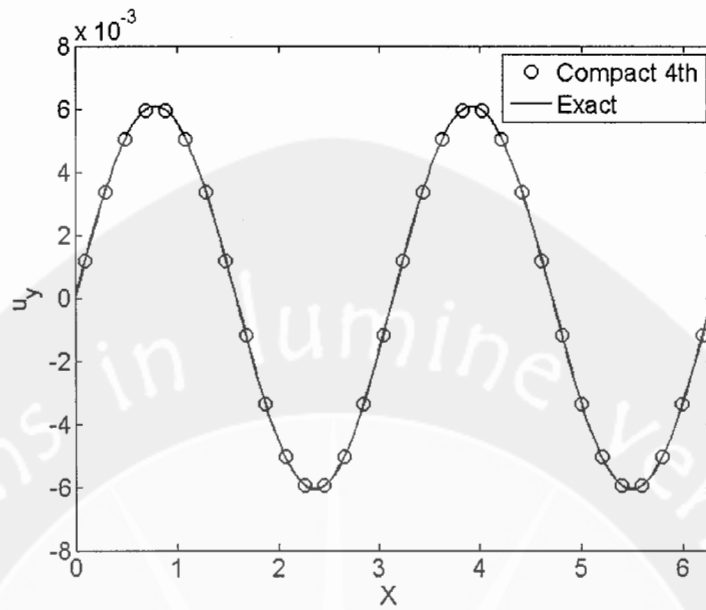




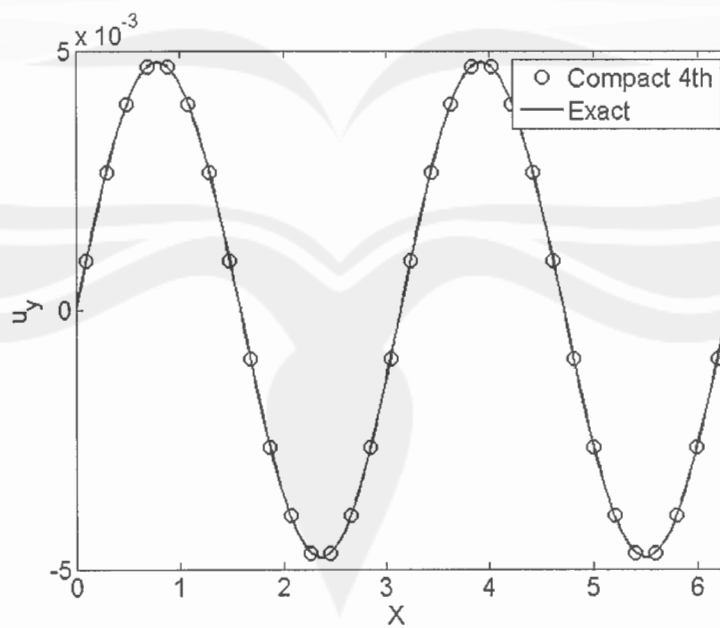
Gambar IV.17. Distribusi kecepatan pada potongan  $y = 3,14$  saat  $t = 10$



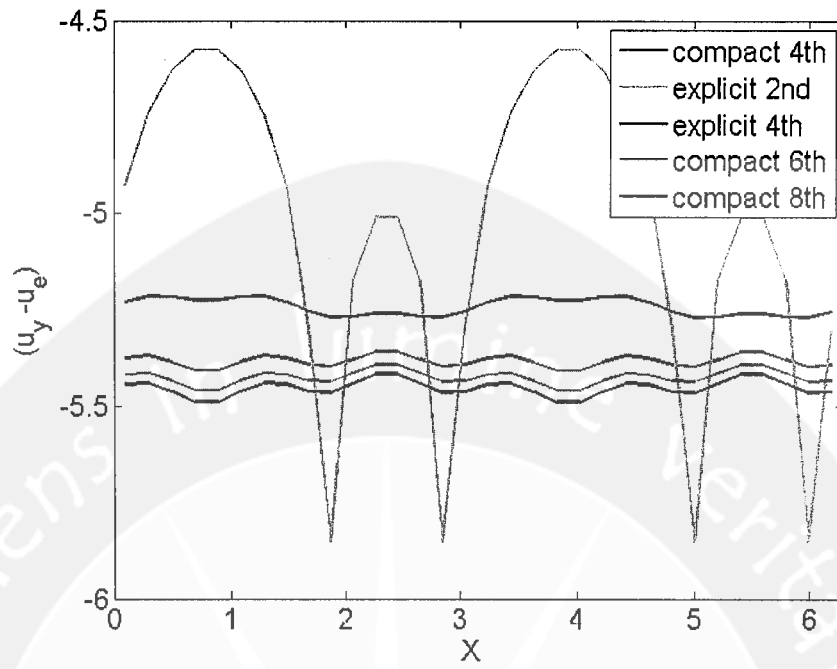
Gambar IV.18. Distribusi kecepatan pada potongan  $y = 3,14$  saat  $t = 50$



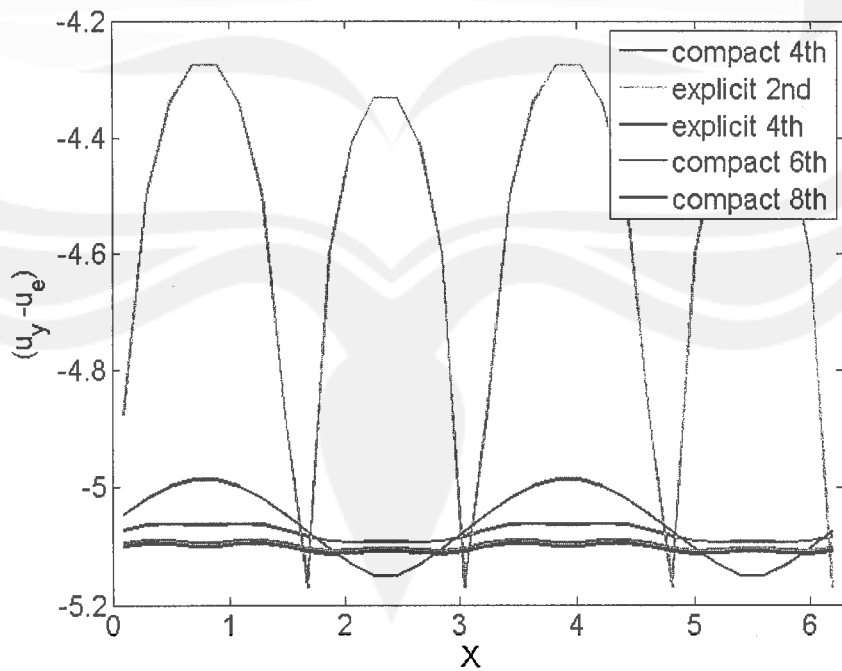
Gambar IV.19. Distribusi kecepatan pada potongan  $y = 3,14$  saat  $t = 100$



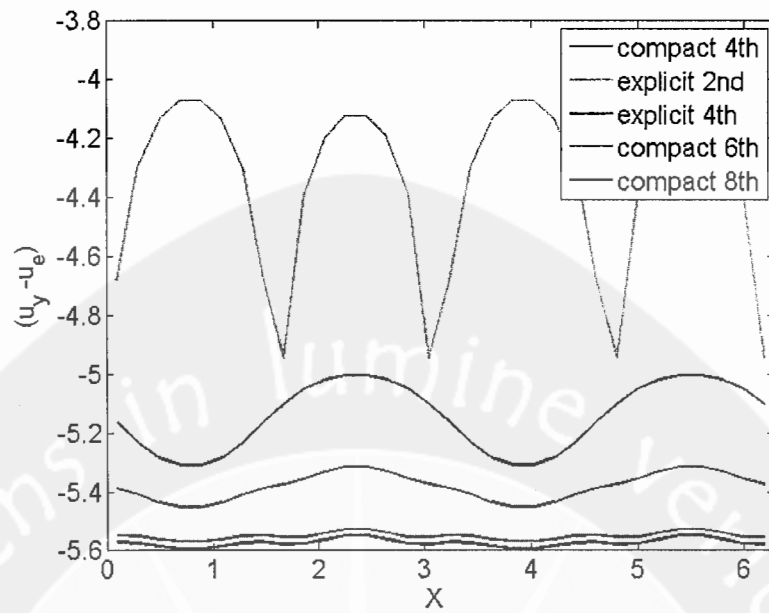
Gambar IV.20. Distribusi kecepatan pada potongan  $y = 3,14$  saat  $t = 150$



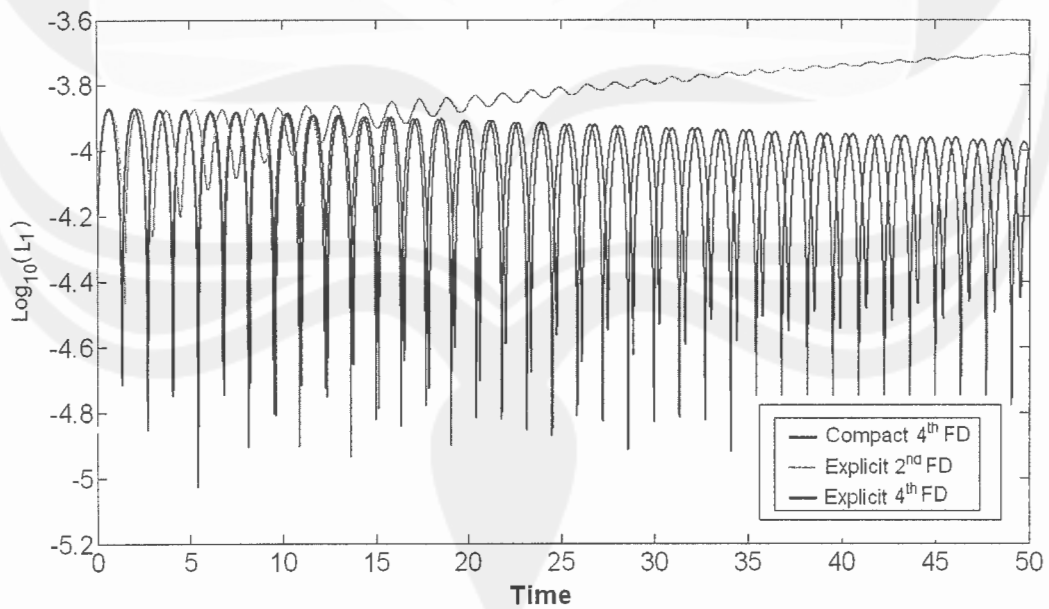
Gambar IV.21. Logaritma galat pada potongan  $y = 3,14$  saat  $t = 8$



Gambar IV.22. Logaritma galat pada potongan  $y = 3,14$  saat  $t = 24$



Gambar IV.23. Logaritma galat pada potongan  $y = 3,14$  saat  $t = 42$



Gambar IV.24. Konvergensi permasalahan Taylor Green Vortex

Gambar IV.24 menunjukkan perubahan galat rata-rata terhadap waktu (konvergensi), dari gambar tersebut dapat diketahui bahwa metode eksplisit orde 2 mempunyai galat yang lebih tinggi dibanding metode lainnya. Konvergensi metode orde tinggi menunjukkan galat rata-rata yang menurun, hal ini disebabkan karena amplitudo kecepatan dan tekanan berkurang menuju nol seiring dengan bertambahnya waktu.



## Bab V Kesimpulan dan Saran

### V.1. Kesimpulan

Kesimpulan yang dapat ditarik dari penelitian ini adalah:

1. Penggunaan skema beda hingga kompak orde tinggi pada suku adveksi metode Lattice Boltzmann untuk penyelesaian permasalahan Taylor Green Vortex menunjukkan akurasi yang tinggi, kenaikan orde akurasi seiring dengan kenaikan orde skema beda hingga
2. Penggunaan skema Runge-Kutta orde-4 dapat meningkatkan kestabilan perhitungan numerik, tetapi stabilitas ini menurun seiring dengan meningkatnya orde skema beda hingga.

### 6.1 Saran Penelitian Lanjutan

Kajian lebih lanjut masih diperlukan untuk mempelajari permasalahan dengan syarat yang tidak periodik. Untuk itu perlu dikaji penggunaan metode beda hingga kompak orde tinggi yang mampu mengatasi syarat batas tersebut tanpa mengalami penurunan akurasi.

## DAFTAR PUSTAKA

- Ghrist, Michelle, 2000, *Finite Difference Method for Wave Equations*, PhD Thesis, The University of Colorado.
- Hirsch, Charles, 2007, *Numerical Computation of Internal and External Flows*, Volume I, John Wiley & Sons, Chicester.
- Laizet, S. and Lamballais, E., 2009, *High-order compact schemes for incompressible flows: A simple and efficient method with quasi-spectral accuracy*, Journal of Computational Physics 228 (2009) 5989–6015.
- Lee, T. and Lin, C. L., 2001, *A characteristic Galerkin Method for Discrete Boltzmann Equations*, Journal of Computational Physics 171 (2001) 336–356.
- Lee, T. and Lin, C. L., 2003, *An Eulerian description of the streaming process in the lattice Boltzmann equation*, Journal of Computational Physics 185 (2003) 445–471.
- Mey, R. W. and Shyy, W., 1999, *On the finite difference-based lattice Boltzmann method in curvilinear coordinates*, Journal of Computational Physics 143 (1998) 426–448.
- Patil, D. V. and Lakshmisha, 2009, *Finite volume TVD formulation of lattice Boltzmann simulation on unstructured mesh*, Journal of Computational Physics 228 (2009) 5262–5279.
- Peng, G. et al., 1999, *Finite volume scheme for the lattice Boltzmann method on unstructured meshes*, Phys. Rev. E 59 (4) 4675.
- Shah, A., Yuan, L., and Khan, A., 2010, *Upwind compact finite difference scheme for time-accurate solution of the Incompressible Navier–Stokes equations*, Applied Mathematics and Computation 215 (2010) 3201–3213.
- Shi, Y., Zhao, T. S., and Guo, Z. L., 2006, *Finite difference-based lattice Boltzmann simulation of natural convection heat transfer in a horizontal concentric annulus*, Computers & Fluids 35 (2006) 1–15.

Svard, M. and Nordstromb, J., 2008, *A stable high-order finite difference scheme for the compressible Navier–Stokes equations No-slip wall boundary conditions*, Journal of Computational Physics 227 (2008) 4805–4824.

Ubertini, S., Bella, G. and Succi., S., 2006, *Unstructured lattice Boltzmann equation with memory*, Math. Comput. Simul. 72 (2006) 237–241.





## LAMPIRAN A

Lampiran A berisi skrip Matlab yang digunakan pada penelitian.

### Taylor Vortex 4 metode.m

```
% Perhitungan Taylor vortex dengan 4-th, 6-th & 8th compact , explicit 2nd & 4th
FDLBM
% 4th Runge-Kutta time integration

clear all;
close all;
clc;
clear

% GENERAL FLOW CONSTANTS
lx = 32;
ly = 1*lx;
lm = ly/2;
tau=0.0018;
%nu = 1e-2; %uLid *lx / Re; % kinematic viscosity
nu=tau/3;
omega = 1. / (3*nu); %+0./2.); % relaxation parameter

tPlot = 400; % cycles for graphical output
k1=2;
k2=2;
Uo=0.01;
dx=2*pi/lx;
dy=2*pi/ly;
dt=5.0e-3;
time=0.0;
CFL=dt/dx;

U1=Uo*(k1/k2);
time_max=50; %max time
maxT = time_max/dt ;% total number of iterations
alpha =1.0; % (0 <= alpha <=1; alpha = 0 --> upwind, alpha = 1 --> central
difference);
% D2Q9 LATTICE CONSTANTS
t = [4/9, 1/9,1/9,1/9,1/9, 1/36,1/36,1/36,1/36];
cx = [ 0, 1, 0, -1, 0, 1, -1, -1, 1];
cy = [ 0, 0, 1, 0, -1, 1, 1, -1, -1];
opp = [ 1, 4, 5, 2, 3, 8, 9, 6, 7];
lid = [2: (lx-1)];

[y,x] = meshgrid(1:ly,1:lx);

x = (x-0.5)*dx;
y = (y-0.5)*dy;
```

```

xe=linspace(0,2*pi,201);
ye=x(1,lm);

% INITIAL CONDITION: (rho=0, u=0) ==> fIn(i) = t(i)
fIn = reshape( t' * ones(1,lx*ly), 9, lx, ly);
ux(1, :, :) = -Uo*cos(k1*x).*sin(k2*y);
uy(1, :, :) = Uo*(k1/k2)*sin(k1*x).*cos(k2*y);
rho(1, :, :)=1.0-0.25*Uo^2*(cos(2.0*k1*x)+(k1/k2)^2*cos(2.0*k2*y));

for i=1:9
    cu = 3*(cx(i)*ux+cy(i)*uy);
    fIn(i, :, :) = rho .* t(i) .* ...
        ( 1 + cu + 1/2*(cu.*cu) - 3/2*(ux.^2+uy.^2) );
end

ux2nd=ux; uy2nd=uy; rho2nd=rho; fIn2nd=fIn;
ux4th=ux; uy4th=uy; rho4th=rho; fIn4th=fIn;
ux6th=ux; uy6th=uy; rho6th=rho; fIn6th=fIn;
ux8th=ux; uy8th=uy; rho8th=rho; fIn8th=fIn;

fname = sprintf('Taylor_compact_%f.txt', dx);
fid = fopen(fname, 'w');
% MAIN LOOP (TIME CYCLES)
for cycle = 1:maxT
    time=time+dt;

    % compact 4-th FD
    [rhs1]=lbm_compact_rhs(fIn,cx,cy,t,omega,dt,lx,ly,dx,dy);
    f1=fIn+0.5*rhs1;
    [rhs2]=lbm_compact_rhs(f1,cx,cy,t,omega,dt,lx,ly,dx,dy);
    f2=fIn+0.5*rhs1;
    [rhs3]=lbm_compact_rhs(f2,cx,cy,t,omega,dt,lx,ly,dx,dy);
    f3=fIn+rhs3;
    [rhs4]=lbm_compact_rhs(f3,cx,cy,t,omega,dt,lx,ly,dx,dy);
    fIn=fIn+(rhs1+2*rhs2+2*rhs3+rhs4)/6;

    % compact 6-th FD (under construction !!!)
    [rhs1]=lbm_compact_6th_rhs(fIn6th,cx,cy,t,omega,dt,lx,ly,dx,dy);
    f1=fIn6th+0.5*rhs1;
    [rhs2]=lbm_compact_6th_rhs(f1,cx,cy,t,omega,dt,lx,ly,dx,dy);
    f2=fIn6th+0.5*rhs1;
    [rhs3]=lbm_compact_6th_rhs(f2,cx,cy,t,omega,dt,lx,ly,dx,dy);
    f3=fIn6th+rhs3;
    [rhs4]=lbm_compact_6th_rhs(f3,cx,cy,t,omega,dt,lx,ly,dx,dy);
    fIn6th=fIn6th+(rhs1+2*rhs2+2*rhs3+rhs4)/6;

    % compact 8-th FD (under construction !!!)
    [rhs1]=lbm_compact_8th_rhs(fIn6th,cx,cy,t,omega,dt,lx,ly,dx,dy);

```

```

f1=fIn8th+0.5*rhs1;
[rhs2]=lbm_compact_8th_rhs(f1,cx,cy,t,omega,dt,lx,ly,dx,dy);
f2=fIn8th+0.5*rhs1;
[rhs3]=lbm_compact_8th_rhs(f2,cx,cy,t,omega,dt,lx,ly,dx,dy);
f3=fIn8th+rhs3;
[rhs4]=lbm_compact_8th_rhs(f3,cx,cy,t,omega,dt,lx,ly,dx,dy);
fIn8th=fIn8th+(rhs1+2*rhs2+2*rhs3+rhs4)/6;

% explicit 2nd FD
[rhs1]=lbm_fd2nd_rhs(fIn2nd,cx,cy,t,omega,dt,lx,ly,dx,dy);
f1=fIn2nd+0.5*rhs1;
[rhs2]=lbm_fd2nd_rhs(f1,cx,cy,t,omega,dt,lx,ly,dx,dy);
f2=fIn2nd+0.5*rhs1;
[rhs3]=lbm_fd2nd_rhs(f2,cx,cy,t,omega,dt,lx,ly,dx,dy);
f3=fIn2nd+rhs3;
[rhs4]=lbm_fd2nd_rhs(f3,cx,cy,t,omega,dt,lx,ly,dx,dy);
fIn2nd=fIn2nd+(rhs1+2*rhs2+2*rhs3+rhs4)/6;

% explicit 4th FD
[rhs1]=lbm_fd4th_rhs(fIn4th,cx,cy,t,omega,dt,lx,ly,dx,dy);
f1=fIn4th+0.5*rhs1;
[rhs2]=lbm_fd4th_rhs(f1,cx,cy,t,omega,dt,lx,ly,dx,dy);
f2=fIn4th+0.5*rhs1;
[rhs3]=lbm_fd4th_rhs(f2,cx,cy,t,omega,dt,lx,ly,dx,dy);
f3=fIn4th+rhs3;
[rhs4]=lbm_fd4th_rhs(f3,cx,cy,t,omega,dt,lx,ly,dx,dy);
fIn4th=fIn4th+(rhs1+2*rhs2+2*rhs3+rhs4)/6;

% MACROSCOPIC VARIABLES
rho = sum(fIn);
ux = reshape ( (cx * reshape(fIn,9,lx*ly)), 1,lx,ly ) ./rho;
uy = reshape ( (cy * reshape(fIn,9,lx*ly)), 1,lx,ly ) ./rho;

rho2nd = sum(fIn2nd);
ux2nd = reshape ( (cx * reshape(fIn2nd,9,lx*ly)), 1,lx,ly ) ./rho2nd;
uy2nd = reshape ( (cy * reshape(fIn2nd,9,lx*ly)), 1,lx,ly ) ./rho2nd;

rho4th = sum(fIn4th);
ux4th = reshape ( (cx * reshape(fIn4th,9,lx*ly)), 1,lx,ly ) ./rho4th;
uy4th = reshape ( (cy * reshape(fIn4th,9,lx*ly)), 1,lx,ly ) ./rho4th;

rho6th = sum(fIn6th);
ux6th = reshape ( (cx * reshape(fIn6th,9,lx*ly)), 1,lx,ly ) ./rho6th;
uy6th = reshape ( (cy * reshape(fIn6th,9,lx*ly)), 1,lx,ly ) ./rho6th;

rho8th = sum(fIn8th);
ux8th = reshape ( (cx * reshape(fIn8th,9,lx*ly)), 1,lx,ly ) ./rho8th;
uy8th = reshape ( (cy * reshape(fIn8th,9,lx*ly)), 1,lx,ly ) ./rho8th;

```

```

u1 = reshape(ux, lx, ly);
u2 = reshape(uy, lx, ly);

u12nd = reshape(ux2nd, lx, ly);
u22nd = reshape(uy2nd, lx, ly);

u14th = reshape(ux4th, lx, ly);
u24th = reshape(uy4th, lx, ly);

u16th = reshape(ux6th, lx, ly);
u26th = reshape(uy6th, lx, ly);

u18th = reshape(ux8th, lx, ly);
u28th = reshape(uy8th, lx, ly);

u2_exact = Uo*(k1/k2)*exp(-nu*(k1^2+k2^2)*time)*(sin(k1*x).*cos(k2*y));
rho_exact(1, :, :) = 1.0-0.25*Uo^2*exp(-
2.0*nu*(k1^2+k2^2)*time)*(cos(2.0*k1*x)+(k1/k2)^2*cos(2.0*k2*y));

Us1=abs(u2-u2_exact);
Us2=abs(u22nd-u2_exact);
Us3=abs(u24th-u2_exact);
Us4=abs(u26th-u2_exact);
Us5=abs(u28th-u2_exact);

L1=sqrt(sum(sum(Us1)))/lx/ly;
L2=sqrt(sum(sum(Us2)))/lx/ly;
L3=sqrt(sum(sum(Us3)))/lx/ly;
L4=sqrt(sum(sum(Us4)))/lx/ly;
L5=sqrt(sum(sum(Us5)))/lx/ly;

L1m(cycle,1)=time;
L1m(cycle,2)=log10(L1);
L1m(cycle,3)=log10(L2);
L1m(cycle,4)=log10(L3);
L1m(cycle,5)=log10(L4);
L1m(cycle,6)=log10(L5);

% time=time+dt;
% VISUALIZATION
if (mod(cycle,tPlot)==0)
% fprintf(fid, 'tstep = %6.0d ; time = %7.4f ; Linf 4th compact = %7.4f ; Linf
2nd = %7.4f ; Linf 4th = %7.4f; Linf 6th = %7.4f\n', cycle,time, L1m(cycle,2),
L1m(cycle,3), L1m(cycle,4),L1m(cycle,5));
% fprintf(2, 'tstep = %6.0d ; time = %7.4f ; Linf 4th compact = %7.4f ; Linf
2nd = %7.4f ; Linf 4th = %7.4f; Linf 6th = %7.4f\n', cycle,time, L1m(cycle,2),
L1m(cycle,3), L1m(cycle,4),L1m(cycle,5));

```

```

        fprintf(fid, 'tstep = %6.0d ; time = %7.4f ; Linf 4th compact = %7.4f ; Linf
8th compact = %7.4f ; Linf 4th = %7.4f; Linf 6th = %7.4f\n', cycle,time,
L1m(cycle,2), L1m(cycle,6), L1m(cycle,4),L1m(cycle,5));
        fprintf(2, 'tstep = %6.0d ; time = %7.4f ; Linf 4th compact = %7.4f ; Linf
8th compact = %7.4f ; Linf 4th = %7.4f; Linf 6th = %7.4f\n', cycle,time,
L1m(cycle,2), L1m(cycle,6), L1m(cycle,4),L1m(cycle,5));

u_exact = Uo*(k1/k2)*exp(-nu*(k1^2+k2^2)*time)*(sin(k1*x)*cos(k2*y));
figure(gcf);

% plot(x(:,lm),u22nd(:,lm),'bo',xe,u_exact,'r','LineWidth',2);

er1=log10(abs(u2(:,lm)-u2_exact(:,lm)));
er2=log10(abs(u22nd(:,lm)-u2_exact(:,lm)));
er3=log10(abs(u24th(:,lm)-u2_exact(:,lm)));
er4=log10(abs(u26th(:,lm)-u2_exact(:,lm)));
er5=log10(abs(u28th(:,lm)-u2_exact(:,lm)));

plot(x(:,lm),er1,'r',x(:,lm),er2,'g',x(:,lm),er3,'b',x(:,lm),er4,'m',x(:,lm),er5,'
k','LineWidth',2);
legend('compact 4th', 'explicit 2nd', 'explicit 4th', 'compact 6th', 'compact 8th')
xlabel('X ', 'FontSize', 18)
ylabel('(u_y -u_e)', 'FontSize', 18)
set(gca, 'FontSize', 18)

xlim([0 2*pi])
ylim([-U1 U1])
% axis equal;
drawnow;
pause(0.0020)
counte = cycle/tPlot;
if(counte < 10)
    slidename = sprintf('C:/Documents and Settings/Pr/My
Documents/MATLAB/LBM/Taylor_vortex_m0%d.png', counte);
elseif(counte < 100)
    slidename = sprintf('C:/Documents and Settings/Pr/My
Documents/MATLAB/LBM/Taylor_vortex_m0%d.png', counte);
else
    slidename = sprintf('C:/Documents and Settings/Pr/My
Documents/MATLAB/LBM/Taylor_vortex_m%d.png', counte);
end
print('-dpng', '-r90', slidename);
end
end

fclose(fid)
xx=[x(1,1) x(lx,1) x(lx,ly) x(1,ly) x(1,1)];
yy=[y(1,1) y(lx,1) y(lx,ly) y(1,ly) y(1,1)];

```

```

%Ls=max(L1m(:,2));
fprintf(2, 'dt = %6.4d ; dx = %6.4f ; CFL = %6.4f; lx = %4.0d ; ly = %4.0d \n',
dt,dx,CFL,lx,ly);
fprintf(2, 'tstep = %6.0d ; time = %7.4f ; Linf 4th compact = %7.4f ; Linf 2nd =
%7.4f ; Linf 4th = %7.4f; Linf 6th = %7.4f\n', cycle,time, L1m(cycle,2),
L1m(cycle,3), L1m(cycle,4),L1m(cycle,5));

figure(2)
hold on
quiver(x,y,u1,u2); axis equal;
plot(xx,yy);
hold off

figure(3)
surf(x,y,squeeze(rho)); shading interp; view(2) ;

figure(4)
plot(L1m(:,1),L1m(:,2),'r',L1m(:,1),L1m(:,3),'g',L1m(:,1),L1m(:,4),'b');
legend('compact 4th', 'explicit 2nd','explicit 4th', 'compact 6th', 'compact 8th')

figure(5);
plot(x(:,lm),u2(:,lm)-u2_exact(:,lm),'b',x(:,lm),u22nd(:,lm)-
u2_exact(:,lm),'r','LineWidth',2);
legend('compact 4th', 'explicit 2nd','explicit 4th', 'compact 6th', 'compact 8th')

xlabel('X ','FontSize',18)
ylabel('Y ','FontSize',18)
xlim([0 2*pi])
ylim([-U1 U1])
set(gca,'FontSize',16)

```

### lbm\_compact\_rhs

```

function [rhs,ux,uy]=lbm_compact_rhs(f,cx,cy,t,omega,dt,lx,ly,dx,dy);

% MACROSCOPIC VARIABLES
rho = sum(f);
ux = reshape ( (cx * reshape(f,9,lx*ly)), 1,lx,ly ) ./rho;
uy = reshape ( (cy * reshape(f,9,lx*ly)), 1,lx,ly ) ./rho;

% COLLISION STEP
for i=1:9
    cu = 3*(cx(i)*ux+cy(i)*uy);
    fEq(i, :, :) = rho .* t(i) .* ...
    ( 1 + cu + 1/2*(cu.*cu) - 3/2*(ux.^2+uy.^2) );
    fTemp(i, :, :) = omega .* (f(i, :, :)-fEq(i, :, :));
end

% STREAMING STEP
%dufx = diff_x(1,lx,1,ly,cx(2),fOut(2, :, : ),dx);

rhs(1, :, : ) = -dt*fTemp(1, :, : );

dufx = pdiff_x(lx,ly,cx(2),f(2, :, : ),dx);
rhs(2, :, : ) = -dt*(dufx+fTemp(2, :, : ));

dufy = pdiff_y(lx,ly,cy(3),f(3, :, : ),dy);

```

```

rhs(3, :, : ) = -dt*(dufy+fTemp(3, :, : ));

dufx = pdiff_x(lx,ly,cx(4),f(4, :, : ),dx);
rhs(4, :, : ) = -dt*(dufx+fTemp(4, :, : ));

dufy = pdiff_y(lx,ly,cy(5),f(5, :, : ),dy);
rhs(5, :, : ) = -dt*(dufy+fTemp(5, :, : ));

dufx = pdiff_x(lx,ly,cx(6),f(6, :, : ),dx);
dufy = pdiff_y(lx,ly,cy(6),f(6, :, : ),dy);
rhs(6, :, : ) = -dt*(dufx+dufy+fTemp(6, :, : ));

dufx = pdiff_x(lx,ly,cx(7),f(7, :, : ),dx);
dufy = pdiff_y(lx,ly,cy(7),f(7, :, : ),dy);
rhs(7, :, : ) = -dt*(dufx+dufy+fTemp(7, :, : ));

dufx = pdiff_x(lx,ly,cx(8),f(8, :, : ),dx);
dufy = pdiff_y(lx,ly,cy(8),f(8, :, : ),dy);
rhs(8, :, : ) = -dt*(dufx+dufy+fTemp(8, :, : ));

dufx = pdiff_x(lx,ly,cx(9),f(9, :, : ),dx);
dufy = pdiff_y(lx,ly,cy(9),f(9, :, : ),dy);
rhs(9, :, : ) = -dt*(dufx+dufy+fTemp(9, :, : ));
return

```

### pdiff\_x

```

% ICASE, Report 98-13 Wilson, R. V. and Demuren, A. O., NASA
% periodic x-derivative function
% Turunan FD compact 4th
function fx = pdiff_x(lx,ly,cx,f,dx)
for j=1:ly,
for i=1:lx,
if i>1
in = i-1;
else
in = lx;
end

if i<lx
ip = i+1;
else
ip = 1;
end

am(i)=1.d0/4.d0;
bm(i)=4.d0/4.d0;
cm(i)=1.d0/4.d0;
dm(i)=1.5*(f(1,ip,j)-f(1,in,j))/2.d0/dx;
end
ffx=ptridg(am,bm,cm,dm,1,lx);
fx(1, :, j)=cx*ffx(:);
end
return

```

### pdiff\_y

```

%
% periodic y-derivative function
%
function fy=pdiff_y(lx,ly,cy,f,dy)
for i=1:lx,
for j=1:ly,
if j>1

```

```

        jn = j-1;
    else
        jn = ly;
    end
    if j<ly
        jp = j+1;
    else
        jp = 1;
    end

    am(j)=1.d0/4.d0;
    bm(j)=4.d0/4.d0;
    cm(j)=1.d0/4.d0;
    dm(j)=1.5*(f(1,i,jp)-f(1,i,jn))/2.d0/dy;
    end
    ffy=ptridg(am,bm,cm,dm,1,ly);
    fy(1,i,:)=cy*ffy(:);
end
return

```

### **lbm\_compact\_6th\_rhs**

```

function
[rhs,ux,uy]=lbm_compact_6th_rhs(f,cx,cy,t,omega,dt,lx,ly,dx,dy);

% MACROSCOPIC VARIABLES
rho = sum(f);
ux = reshape ( (cx * reshape(f,9,lx*ly)), 1,lx,ly ) ./rho;
uy = reshape ( (cy * reshape(f,9,lx*ly)), 1,lx,ly ) ./rho;

% COLLISION STEP
for i=1:9
    cu = 3*(cx(i)*ux+cy(i)*uy);
    fEq(i,,:) = rho .* t(i) .* ...
        ( 1 + cu + 1/2*(cu.*cu) - 3/2*(ux.^2+uy.^2) );
    fTemp(i,,:) = omega .* (f(i,,:)-fEq(i,,:));
end

% STREAMING STEP
%dufx = diff_x(1,lx,1,ly,cx(2),fOut(2,,:),dx);

rhs(1,,:) = -dt*fTemp(1,,:);

dufx = pdiff_x_6th(lx,ly,cx(2),f(2,,:),dx);
rhs(2,,:) = -dt*(dufx+fTemp(2,,:));

dufy = pdiff_y_6th(lx,ly,cy(3),f(3,,:),dy);
rhs(3,,:) = -dt*(dufy+fTemp(3,,:));

dufx = pdiff_x_6th(lx,ly,cx(4),f(4,,:),dx);
rhs(4,,:) = -dt*(dufx+fTemp(4,,:));

dufy = pdiff_y_6th(lx,ly,cy(5),f(5,,:),dy);
rhs(5,,:) = -dt*(dufy+fTemp(5,,:));

dufx = pdiff_x_6th(lx,ly,cx(6),f(6,,:),dx);
dufy = pdiff_y_6th(lx,ly,cy(6),f(6,,:),dy);
rhs(6,,:) = -dt*(dufx+dufy+fTemp(6,,:));

```



```
dufx = pdiff_x_6th(lx,ly,cx(7),f(7,::, ),dx);
dufy = pdiff_y_6th(lx,ly,cy(7),f(7,::, ),dy);
rhs(7,::, ) = -dt*(dufx+dufy+fTemp(7,::, ));
```

```
dufx = pdiff_x_6th(lx,ly,cx(8),f(8,::, ),dx);
dufy = pdiff_y_6th(lx,ly,cy(8),f(8,::, ),dy);
rhs(8,::, ) = -dt*(dufx+dufy+fTemp(8,::, ));
```

```
dufx = pdiff_x_6th(lx,ly,cx(9),f(9,::, ),dx);
dufy = pdiff_y_6th(lx,ly,cy(9),f(9,::, ),dy);
rhs(9,::, ) = -dt*(dufx+dufy+fTemp(9,::, ));
return
```

### pdiff\_x\_6th

```
% ICASE, Report 98-13 Wilson, R. V. and Demuren, A. O., NASA
% periodic x-derivative function
```

```
%
function fx = pdiff_x_6th(lx,ly,cx,f,dx)
for j=1:ly,
for i=1:lx,
```

```
if i==1
in=lx;
inl=lx-1;
end
```

```
if i==2
in=1;
inl=lx;
end
```

```
if i>2
in=i-1;
inl=i-2;
end
```

```
if i==lx
ip=1;
ipl=2;
end
```

```
if i==(lx-1)
ip=lx;
ipl=1;
end
```

```
if i<(lx-1)
ip=i+1;
ipl=i+2;
end
```

```
am(i)=1.d0/5.d0;
bm(i)=3.d0/5.d0;
```

```

        cm(i)=1.d0/5.d0;
        dm(i)=(f(1,ip1,j)+28.0*f(1,ip,j)-28.0*f(1,in,j)-f(1,in1,j))/60.d0/dx;
    end
    ffx=ptridg(am,bm,cm,dm,1,lx);
    fx(1,:,j)=cx*ffx(:);
end
return

```

### pdiff\_y\_6th

```

%
% periodic y-derivative function
%
function fy=pdiff_y_6th(lx,ly,cy,f,dy)
for i=1:lx,
    for j=1:ly,
        if j==1
            jn=ly;
            jnl=ly-1;
        end
        if j==2
            jn=1;
            jnl=ly;
        end
        if j>2
            jn=j-1;
            jnl=j-2;
        end
        if j==ly
            jp=1;
            jpl=2;
        end
        if j==(ly-1)
            jp=ly;
            jpl=1;
        end
        if j<(ly-1)
            jp=j+1;
            jpl=j+2;
        end

        am(j)=1.d0/5.d0;
        bm(j)=3.d0/5.d0;
        cm(j)=1.d0/5.d0;
        dm(j)=(f(1,i,jpl)+28.0*f(1,i,jp)-28.0*f(1,i,jn)-f(1,i,jnl))/60.d0/dy;
    end
    ffy=ptridg(am,bm,cm,dm,1,ly);
    fy(1,i,:)=cy*ffy(:);
end
return

```

### lbm\_compact\_8th\_rhs

```

function [rhs,ux,uy]=lbm_compact_8th_rhs(f,cx,cy,t,omega,dt,lx,ly,dx,dy);

% MACROSCOPIC VARIABLES
rho = sum(f);
ux = reshape ( (cx * reshape(f,9,lx*ly)), 1,lx,ly ) ./rho;
uy = reshape ( (cy * reshape(f,9,lx*ly)), 1,lx,ly ) ./rho;

```

```

% COLLISION STEP
for i=1:9
    cu = 3*(cx(i)*ux+cy(i)*uy);
    fEq(i, :, :) = rho .* t(i) .* ...
        ( 1 + cu + 1/2*(cu.*cu) - 3/2*(ux.^2+uy.^2) );
    fTemp(i, :, :) = omega .* (f(i, :, :)-fEq(i, :, :));
end

% STREAMING STEP
%dufx = diff_x(1,lx,1,ly,cx(2),fOut(2, :, :),dx);

rhs(1, :, : ) = -dt*fTemp(1, :, : );

dufx = pdiff_x_8th(lx,ly,cx(2),f(2, :, :),dx);
rhs(2, :, : ) = -dt*(dufx+fTemp(2, :, : ));

dufy = pdiff_y_8th(lx,ly,cy(3),f(3, :, :),dy);
rhs(3, :, : ) = -dt*(dufy+fTemp(3, :, : ));

dufx = pdiff_x_8th(lx,ly,cx(4),f(4, :, :),dx);
rhs(4, :, : ) = -dt*(dufx+fTemp(4, :, : ));

dufy = pdiff_y_8th(lx,ly,cy(5),f(5, :, :),dy);
rhs(5, :, : ) = -dt*(dufy+fTemp(5, :, : ));

dufx = pdiff_x_8th(lx,ly,cx(6),f(6, :, :),dx);
dufy = pdiff_y_8th(lx,ly,cy(6),f(6, :, :),dy);
rhs(6, :, : ) = -dt*(dufx+dufy+fTemp(6, :, : ));

dufx = pdiff_x_8th(lx,ly,cx(7),f(7, :, :),dx);
dufy = pdiff_y_8th(lx,ly,cy(7),f(7, :, :),dy);
rhs(7, :, : ) = -dt*(dufx+dufy+fTemp(7, :, : ));

dufx = pdiff_x_8th(lx,ly,cx(8),f(8, :, :),dx);
dufy = pdiff_y_8th(lx,ly,cy(8),f(8, :, :),dy);
rhs(8, :, : ) = -dt*(dufx+dufy+fTemp(8, :, : ));

dufx = pdiff_x_8th(lx,ly,cx(9),f(9, :, :),dx);
dufy = pdiff_y_8th(lx,ly,cy(9),f(9, :, :),dy);
rhs(9, :, : ) = -dt*(dufx+dufy+fTemp(9, :, : ));
return

```

### pdiff\_x\_8th

```

% ICASE, Report 98-13 Wilson, R. V. and Demuren, A. O., NASA
% periodic x-derivative function
%
function fx = pdiff_x_8th(lx,ly,cx,f,dx)
    for j=1:ly,
        for i=1:lx,
            if i==1
                in=lx;
                in1=lx-1;
                in2=lx-2;
            end
            if i==2
                in=1;
                in1=lx;
                in2=lx-1;
            end
            if i==3
                in=2;
                in1=1;
                in2=lx;
            end

```

```

end
if i>3
    in=i-1;
    in1=i-2;
    in2=i-3;
end
if i==1x
    ip=1;
    ip1=2;
    ip2=3;
end
if i==(1x-1)
    ip=1x;
    ip1=1;
    ip2=2;
end
if i==(1x-2)
    ip=1x-1;
    ip1=1x;
    ip2=1;
end
if i<(1x-2)
    ip=i+1;
    ip1=i+2;
    ip2=i+3;
end
am(i)=3.d0/14.d0;
bm(i)=8.d0/14.d0;
cm(i)=3.d0/14.d0;
dm(i)=(-f(1,ip2,j)+24.0*f(1,ip1,j)+375.0*f(1,ip,j)-375.0*f(1,in,j)-
24.0*f(1,in1,j)+f(1,in2,j))/840.d0/dx;
end
ffx=ptridg(am,bm,cm,dm,1,1x);
fx(1,:,j)=cx*ffx(:);
end
return

```

### pdiff\_y\_8th

```

%
% periodic y-derivative function
%
function fy=pdiff_y_8th(lx,ly,cy,f,dy)
for i=1:lx,
    for j=1:ly,
        if j==1
            jn=ly;
            jn1=ly-1;
            jn2=ly-2;
        end
        if j==2
            jn=1;
            jn1=ly;
            jn2=ly-1;
        end
        if j==3
            jn=2;
            jn1=1;
            jn2=ly;
        end
        if j>3
            jn=j-1;
            jn1=j-2;
            jn2=j-3;
        end
        if j==ly
            jp=1;
            jp1=2;
            jp2=3;

```

```

end
if j==(ly-1)
    jp=ly;
    jp1=1;
    jp2=2;
end
if j==(ly-2)
    jp=ly-1;
    jp1=ly;
    jp2=1;
end
end

if j<(ly-2)
    jp=j+1;
    jp1=j+2;
    jp2=j+3;
end

am(j)=3.d0/14.d0;
bm(j)=8.d0/14.d0;
cm(j)=3.d0/14.d0;
dm(j)=(-f(1,i,jp2)+24.0*f(1,i,jp1)+375.0*f(1,i,jp)-375.0*f(1,i,jn)-
24.0*f(1,i,jn1)+f(1,i,jn2))/840.d0/dy;
end
ffy=ptridg(am,bm,cm,dm,1,ly);
fy(1,i,:)=cy*ffy(:);
end
return

```

### **lbm\_fd2nd\_rhs**

```

function [rhs,ux,uy]=lbm_fd2nd_rhs(f,cx,cy,t,omega,dt,lx,ly,dx,dy);

% MACROSCOPIC VARIABLES
rho = sum(f);
ux = reshape ( (cx * reshape(f,9,lx*ly)), 1,lx,ly ) ./rho;
uy = reshape ( (cy * reshape(f,9,lx*ly)), 1,lx,ly ) ./rho;

% COLLISION STEP
for i=1:9
    cu = 3*(cx(i)*ux+cy(i)*uy);
    fEq(i,,:) = rho .* t(i) .* ...
        ( 1 + cu + 1/2*(cu.*cu) - 3/2*(ux.^2+uy.^2) );
    fTemp(i,,:) = omega .* (f(i,,:)-fEq(i,,:));
end

% STREAMING STEP
%dufx = diff_x(1,lx,1,ly,cx(2),fOut(2,,:),dx);

rhs(1,,:) = -dt*fTemp(1,,:);

dufx = diff_2nd_x(lx,ly,cx(2),f(2,,:),dx);
rhs(2,,:) = -dt*(dufx+fTemp(2,,:));

dufy = diff_2nd_y(lx,ly,cy(3),f(3,,:),dy);
rhs(3,,:) = -dt*(dufy+fTemp(3,,:));

dufx = diff_2nd_x(lx,ly,cx(4),f(4,,:),dx);
rhs(4,,:) = -dt*(dufx+fTemp(4,,:));

dufy = diff_2nd_y(lx,ly,cy(5),f(5,,:),dy);

```

```

rhs(5, :, : ) = -dt*(dufy+fTemp(5, :, : ));

dufx = diff_2nd_x(lx,ly,cx(6),f(6, :, : ),dx);
dufy = diff_2nd_y(lx,ly,cy(6),f(6, :, : ),dy);
rhs(6, :, : ) = -dt*(dufx+dufy+fTemp(6, :, : ));

dufx = diff_2nd_x(lx,ly,cx(7),f(7, :, : ),dx);
dufy = diff_2nd_y(lx,ly,cy(7),f(7, :, : ),dy);
rhs(7, :, : ) = -dt*(dufx+dufy+fTemp(7, :, : ));

dufx = diff_2nd_x(lx,ly,cx(8),f(8, :, : ),dx);
dufy = diff_2nd_y(lx,ly,cy(8),f(8, :, : ),dy);
rhs(8, :, : ) = -dt*(dufx+dufy+fTemp(8, :, : ));

dufx = diff_2nd_x(lx,ly,cx(9),f(9, :, : ),dx);
dufy = diff_2nd_y(lx,ly,cy(9),f(9, :, : ),dy);
rhs(9, :, : ) = -dt*(dufx+dufy+fTemp(9, :, : ));
return

```

### diff\_2nd\_x

```

function fx = diff_2nd_x(lx,ly,cx,f,dx)
for j=1:ly,
    for i=1:lx,
        if i>1
            in = i-1;
        else
            in = lx;
        end

        if i<lx
            ip = i+1;
        else
            ip = 1;
        end
        fx(1,i,j)=(f(1,ip,j)-f(1,in,j))/2.d0/dx;
    end
end
fx=cx*fx;
return

```

### diff\_2nd\_y

```

function fy = diff_2nd_y(lx,ly,cy,f,dy)
for i=1:lx,
    for j=1:ly,
        if j>1
            jn = j-1;
        else
            jn = ly;
        end

        if j<ly
            jp = j+1;
        else
            jp = 1;
        end
        fy(1,i,j)=(f(1,i,jp)-f(1,i,jn))/2.d0/dy;
    end
end
fy=cy*fy;
return

```

### ptring.m

```
% penyelesaian matriks tridiagonal periodik
function [dm] = ptring(am,bm,cm,dm,il,iu)
is=il+1;
ie=iu-1;
dm(il)=dm(il)/bm(il);
cm(il)=cm(il)/bm(il);
am(il)=am(il)/bm(il);
for i=is:ie
bm(i)=bm(i)-am(i)*cm(i-1);
dm(i)=(dm(i)-am(i)*dm(i-1))/bm(i);
dm(iu)=dm(iu)-cm(iu)*dm(i-1);
am(i)=-am(i)*am(i-1)/bm(i);
cm(i)=cm(i)/bm(i);
bm(iu)=bm(iu)-cm(iu)*am(i-1);
cm(iu)=-cm(iu)*cm(i-1);
end
ad=am(iu)+cm(iu);
bm(iu)=bm(iu)-ad*(cm(iu-1)+am(iu-1));
dm(iu)=(dm(iu)-ad*dm(iu-1))/bm(iu);
for ibac=il:ie
i=ie-ibac+1;
du(i)=dm(i)-am(i)*dm(iu)-cm(i)*dm(i+1);
end
```

## LAMPIRAN B

Lampiran ini berisi paper hasil penelitian ini yang diseminarkan di ITB pada International Conference on Industrial and Applied Mathematics pada tanggal 6 - 8 Juli 2010.



### Proceeding of CIAM

*Conference on  
Industrial and Applied Mathematics*

*6<sup>th</sup> – 8<sup>th</sup> July 2010  
Institut Teknologi Bandung*

Editors:

L.H. Wiryanto  
S.R. Pudjaprasetya

Faculty of Mathematics and Natural Sciences  
Institut Teknologi Bandung  
Jalan Ganesha 10 Bandung, Indonesia.  
Phone : +62 22 2502545  
Fax : +62 22 250 6450

ISSN: 977-208-70510-0-8



9 772087 051008



Electronic Proceeding

# CIAM 2010

Conference on Industrial and Applied Mathematics  
6 - 8 July 2010

## The Committees of the conference

### Scientific Committee

Larry Forbes (University of Tasmania, Australia)  
Robert McKibbin (Massey University, New Zealand)  
Susumu Hara (Nagoya University, Japan)  
Edy Soewono (ITB, Indonesia)  
Chan basaruddin (University of Indonesia, Indonesia)  
Robert Saragih (ITB, Indonesia)

### Organizing Committee

L.H. Wiryanto (Chair)  
Sri Redjeki Pudjaprasetya  
Novriana Sumarti  
Andonowati  
Kuntjoro Adji Sidarto

### Technical Committee

Jalina Wijaya	Maulana Wimar Banuardhi
Agus Yodi Gunawan	Indriani Rustomo
Nuning Nuraini	Pritta Etriana
Janson Naiborhu	Adrianus Yosia
Adil Aulia	Rafki Hidayat
Lina Anugerah	Intan Hartri Putri
Ismi Ridha	Yunan Pramesi Haris
Ikha Magdalena	Freddy Susanto

- 26 **ARMA Model Identification using Genetic Algorithm (An Application to Arc Tube Low Power Demand Data)**, *Irhamah, Dedy Dwi Prastyo and M. Nasrul Rohman* 151-155
- 27 **A Particle Swarm Optimization for Employee Placement Problems in the Competency Based Human Resource Management System**, *Joko Siswanto and The Jin Ai* 156-161
- 28 **Measuring Similarity between Wavelet Function and Transient in a Signal with Symmetric Distance Coefficient**, *Nemuel Daniel Pah* 162-166
- 29 **An Implementation of Investment Analysis using Fuzzy Mathematics**, *Novriana Sumarti and Qino Danny* 167-169
- 30 **Simulation of Susceptible Areas to the Impact of Storm Tide Flooding along Northern Coasts of Java**, *Nining Suri Ningsih, Sujwan Hudi, Dwi F. Suputri, Farrah Hanifah, and Amanda P. Rudiawan* 170-178
- 31 **Fuzzy Finite Difference on Calculation of an Individual' Bank Deposits**, *Novriana Sumarti and Siti Mardiah* 179-183
- 32 **An Implementation of Fuzzy Linear System In Economics**, *Novriana Sumarti and Cucu Sukaenah* 184-187
- 33 **Compact Finite Difference Method for Solving Discrete Boltzmann Equation**, *PRANOWO, A. GATOT BINTORO* 188-193
- 34 **Natural convection heat transfer with an Al<sub>2</sub>O<sub>3</sub> nanofluids at low Rayleigh number**, *Zailan Siri, Ishak Hashim and Rozaini Roslan* 194-199
- 35 **Optimization model for estimating productivity growth in Malaysian food manufacturing industry**, *Nordin Hj. Mohamad, and Fatimah Said* 200-206
- 36 **Numerical study of natural convection in a porous cavity with transverse magnetic field and non-uniform internal heating**, *Habibis Saleh, Ishak Hashim and Rozaini Roslan* 207-211
- 37 **THE DISTRIBUTION PATTERN AND ABUNDANCE OF ASTEROID AND ECHINOID AT RINGGUNG WATERS SOUTH LAMPUNG**, *Arwinsyah Arka, Agus Purwoko, Oktavia* 212-215
- 38 **Low biomass of macrobenthic fauna at a tropical mudflat: an effect of latitude?**, *Agus Purwoko and Wim J. Woijf* 216-224
- 39 **Density and biomass of the macrobenthic fauna of the intertidal area in Sembilang national park, South Sumatra, Indonesia**, *Agus Purwoko and Wim J. Woijf* 225-234

## Compact Finite Difference Method for Solving Discrete Boltzmann Equation

PRANOWO<sup>1</sup> & A. GATOT BINTORO<sup>2</sup>

<sup>1</sup> Senior Lecturer of Informatics Engineering, Atma Jaya University, Yogyakarta, Indonesia.  
(Email: pran@mail.uajy.ac.id)

<sup>2</sup> Lecturer of Industrial Engineering Atma Jaya University, Yogyakarta, Indonesia

### ABSTRACT

Fourth compact finite difference (FD) method for solving two dimensional Discrete Boltzmann Equation (DBE) for simulation of fluid flows is proposed in this paper. The solution procedure is carried out in Eulerian framework. BGK (Bhatnagar-Gross-Krook) scheme is adopted to approximate the collision term. The convective terms are discretized using 4<sup>th</sup> compact finite difference method to improve the accuracy and stability. The semidiscrete equations are updated using 4<sup>th</sup> order explicit Runge-Kutta method. Preliminary results of the method applied on the Taylor-Green vortex flows benchmark are presented. We compared the numerical results with other numerical results, i.e. explicit 2<sup>nd</sup> and 4<sup>th</sup> FD, and exact solutions. The comparisons showed excellent agreement.

### KEYWORDS

Compact finite difference; Boltzmann; BGK; ; Taylor vortex

### INTRODUCTION

In the last decade the lattice-Boltzmann method (LBM) has attracted much attention in the simulation of fluid dynamics problems. Unlike conventional computational fluid dynamics methods, which discretize the macroscopic governing equations directly, the LBM method solves the gas kinetic equation at the mesoscopic scale, i.e. the discrete Boltzmann equation with the Bhatnagar-Gross-Krook (BGK) relaxation for the collision operator. The BGK relaxation process allows the recovery of Navier Stokes equations through Chapman Enskog expansion for low Knudsen number.

In the gas kinetic theory, the evolution of the single-particle density distribution function  $f(t, \mathbf{x}, \mathbf{e})$  which represents the probability density of a particle with unit mass moving with velocity  $\mathbf{e}$  at point  $\mathbf{x}$  at time  $t$ , is governed by the Boltzmann equation:

$$\frac{\partial f}{\partial t} + \mathbf{e} \cdot \nabla f = -\frac{(f - f^{eq})}{\tau} \quad (1)$$

where  $f^{eq}$  is the equilibrium distribution and  $\tau$  is relaxation time. After discretizing the velocity space  $\mathbf{e}$  into various directions, the 2-D Boltzmann equation for the velocity distribution function  $f_i$  may be written as discrete Boltzmann equation.

$$\frac{\partial f_i}{\partial t} + \mathbf{e}_i \cdot \nabla f_i = -\frac{(f_i - f_i^{eq})}{\tau} \quad (2)$$

The discrete velocity  $\mathbf{e}_i$  is expressed as:

$$\mathbf{e}_i = \begin{cases} (0,0) & , i=1 \\ (\cos \theta_i, \sin \theta_i) & , \theta_i = (i-1)\frac{\pi}{4}, i=2,3,4,5 \\ \sqrt{2}(\cos \theta_i, \sin \theta_i) & , \theta_i = (i-1)\frac{\pi}{4}, i=6,7,8,9 \end{cases}$$

$$\rho = \sum_{i=0}^8 f_i \quad (3a)$$

$$\rho u_j = \sum_{i=0}^8 f_i e_{ij} \quad (3b)$$

$$f_i^{eq} = \omega_i \left( \frac{e_i \cdot \mathbf{u}}{c_s^2} + \frac{(e_i \cdot \mathbf{u})^2}{2c_s^4} - \frac{|\mathbf{u}|^2}{2} \right)$$

with  $\omega_1 = 4/9, \omega_2 = \omega_3 = \omega_4 = \omega_5 = 1/9$ ,  
and  $\omega_6 = \omega_7 = \omega_8 = \omega_9 = 1/36$ . The pressure can be calculated from  $p = c_s^3 \rho$  with of sound velocity  $c_s = 1/\sqrt{3}$  in lattice unit

and the kinematic viscosity of fluid is  $\nu = \frac{\tau}{3}$ .

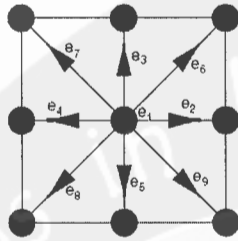


Figure 1. Velocities in 2-D Lattice Boltzmann model (D2Q9)

In Lattice Boltzmann method eq. (2) is solved in the form of

$$f_i(\mathbf{x} + \mathbf{e}_i, t + 1) = f_i(\mathbf{x}, t) - \frac{1}{\tau} (f_i(\mathbf{x}, t) - f_i^{eq}(\mathbf{x}, t)) \quad (4)$$

using  $\Delta x = \Delta y = \Delta t = 1$ . The use of unit square mesh elements is restrictive. Several extension to the LBM have been developed to overcome this restriction. Reference [1] used finite difference method (FDM) with 2<sup>nd</sup> upwind discretization for convective terms. In ref. [1] the FDM is extended to curvilinear coordinates with non-uniform grids. Unfortunately the 2<sup>nd</sup> upwind makes the stencil longer, so it is not easy to handle the boundary condition. Reference [2] used FDM on non-uniform grids. They used implicit temporal discretization to improve the stability. Many modified FDM were proposed to improve the stability and numerical accuracy. Upwind FDM suffers from large dissipation error and standard 2<sup>nd</sup> suffers from large dispersion error. Spectral method [5] offers exact differentiation but suffers from low flexibility in treatment of boundary condition.

In this paper, the 4<sup>th</sup> order compact FDM is proposed to discretize the convective terms of eq. (2). The method is preferred due to high accuracy and flexibility [4]. For improving the stability, the 2<sup>nd</sup> explicit Runge Kutta method is used to integrate the semi-discrete equation.

## II. DISCRETIZATION

The linear convective terms of equations (2) are discretized using 4<sup>th</sup> compact finite difference method:

$$\frac{1}{6} \left( \frac{\partial f_i}{\partial x} \right)_{k+1,j} + \frac{2}{3} \left( \frac{\partial f_i}{\partial x} \right)_{k,j} + \frac{1}{6} \left( \frac{\partial f_i}{\partial x} \right)_{k-1,j} = \frac{f_{k+1,j} - f_{k-1,j}}{2\Delta x} \quad (5a)$$

$$\frac{1}{6} \left( \frac{\partial f_i}{\partial y} \right)_{k,j+1} + \frac{2}{3} \left( \frac{\partial f_i}{\partial y} \right)_{k,j} + \frac{1}{6} \left( \frac{\partial f_i}{\partial y} \right)_{k,j-1} = \frac{(f_i)_{k,j+1} - (f_i)_{k,j-1}}{2\Delta y} \quad (5b)$$

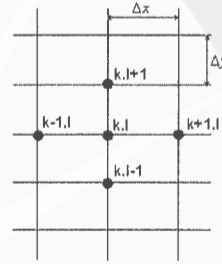


Figure 2. Finite Difference Stencil

After discretizing the convective terms using 4<sup>th</sup> compact FD, we obtain semi discrete equation of (2).

$$\frac{\partial f_i}{\partial t} = \mathbf{e}_i \cdot \nabla f_i - \frac{(f_i - f_i^{eq})}{\tau} = L(f_i) \quad (6)$$

Then the time update is performed using classical 4<sup>th</sup> explicit Runge Kutta method.

## III. ANALYSIS OF DISCRETIZATION

The analysis of spatial and temporal discretization are given in this section. For simplicity, linear advective equation is takes as example.

$$\frac{\partial u}{\partial t} + a \frac{\partial u}{\partial x} = 0, \quad x \in [0, 2\pi] \quad (7)$$

$$u(x, 0) = e^{ikx}$$

where  $a$  is velocity constant.

The exact solution of eq. (7) is easily computed, and we have

$$u(x,t) = e^{i(kx - \omega t)} \quad (8)$$

where  $k$  is the wave number and  $\omega$  is angular frequency. From the exact solution, we obtained the exact dispersion relation:  $\omega = ak$ . By substituting the local solution  $u^i(x,t) = U^i e^{i(k^*x - \omega t)}$  to eq. (6), we obtain the numerical dispersion relation [5]:

$$ik^* \Delta x = i \left( \frac{1.5 \sin(k \Delta x)}{1 + \frac{2}{4} \cos(k \Delta x)} \right) \quad (9)$$

From the eq. (9), it can be seen that numerical dispersion relation of 4<sup>th</sup> compact FD has no imaginary components, so it can be concluded that 4<sup>th</sup> compact FD is conservative and has no dissipation error. For acceptable dispersion error  $|k^* \Delta x - k \Delta x| \leq 10^{-2}$ , we found that  $k^* \Delta x = 1.0893$ . Therefore, the spatial grid points per wavelength (PPW) is  $PPW = 2\pi / 1.0893 = 5.7683$ . From the right term of eq. (9) the eigenvalues of 4<sup>th</sup> compact FD can be calculated and they are purely imaginary, covering  $(-1.732a/\Delta x, +1.732a/\Delta x)$

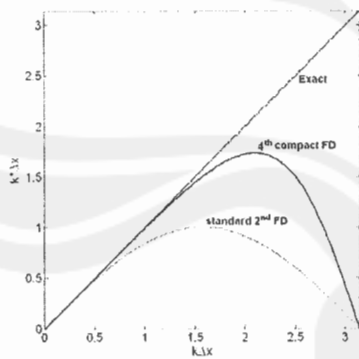


Figure 3. Dispersion error

$$\frac{\partial \bar{f}_i}{\partial t} = L(\bar{f}_i) \quad (10)$$

where  $L$  is the residual terms which contains the spatial terms of the governing equations. By substituting  $f_i = \bar{f}_i(t) e^{ikx}$  into eq. (6), we obtain

$$\frac{\partial \bar{f}_i}{\partial t} = \lambda \bar{f}_i \quad (11)$$

Where  $\bar{f}_i$  is the Fourier coefficient,  $i = \sqrt{-1}$ , and  $\lambda$  is complex number. The left terms of eq. (11) is expanded by using 4<sup>th</sup> explicit Runge-Kutta scheme to obtain the amplification factor:

$$G = \frac{\bar{f}_i^{n+1}}{\bar{f}_i^n} = 1 + (\lambda \Delta t) + \frac{1}{2} (\lambda \Delta t)^2 \quad (12)$$

The stability condition requires that the amplification factor must be bounded,

$$|G| \leq 1 \quad (13)$$

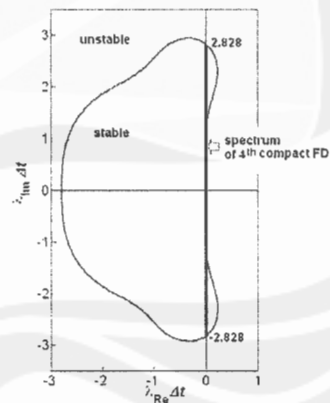


Figure 4. The stability region in complex plane

The stability region in complex plane can be seen in figure 4. The eigenvalues of 4<sup>th</sup> RK in imaginary axis are covering

$(-2.828, +2.828)\Delta t$ , the stability condition of the fully discrete equation is

$$CFL = \frac{a\Delta t}{\Delta x} \leq 2.828/1.732 \quad (14)$$

$$CFL \leq 1.6328$$

CFL is Courant Friedrichs Lewy number.

#### IV. NUMERICAL RESULTS

The performance of numerical is tested by application to 2 benchmark problems. The first problem to be solved is 1-D linear convection problem [3]:

$$\frac{\partial u}{\partial t} + 2\pi \frac{\partial u}{\partial x} = 0, \quad x \in [0, 2\pi]$$

$$u(x, 0) = e^{\sin(x)}$$

with periodic boundary condition

The exact solution to above equation is a right-moving wave of the form:

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

and  $\Delta x = 0.0982, \Delta x = 0.0491, \Delta x = 0.0245$ .

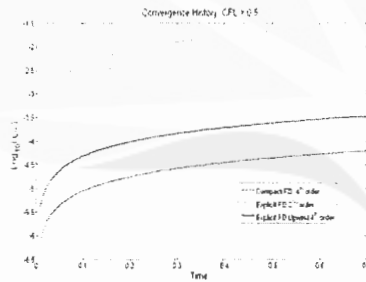


Figure 5. Convergence History for linear convection problem

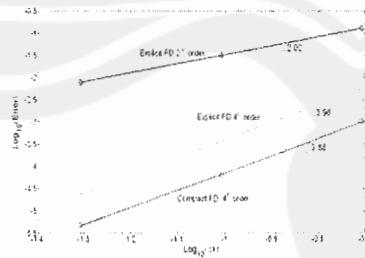


Figure 6. Accuracy Order

We compare the results with explicit 2<sup>nd</sup> and 4<sup>th</sup> order FD. We take constant  $\Delta t = 5e-3$

From figure 5, we can see that 4<sup>th</sup> compact FD has the lowest error among the r methods. The accuracy orders are in a good agreement with theoretical results.

The second problem to be solved is 2-D decaying Taylor Green vortex flow problem [5]. The Taylor-Green vortex flow has the following analytic solutions to incompressible Navier-Stokes equation in 2-D:

$$u_x(x, y, t) = -U_0 \cos(k_x x) \sin(k_y y) e^{-(k_x^2 + k_y^2)t}$$

$$u_y(x, y, t) = \frac{k_x}{k_y} U_0 \cos(k_x x) \sin(k_y y) e^{-(k_x^2 + k_y^2)t}$$

$$u_z(x, y, t) = -\frac{U_0^2}{4} \left( \cos(2k_x x) + \left(\frac{k_x}{k_y}\right)^2 \sin(2k_y y) \right) e^{-(k_x^2 + k_y^2)t} - 1$$

where  $U_0$  is initial velocity amplitude,  $k_x$  and  $k_y$  are the wave number in x dan y direction.

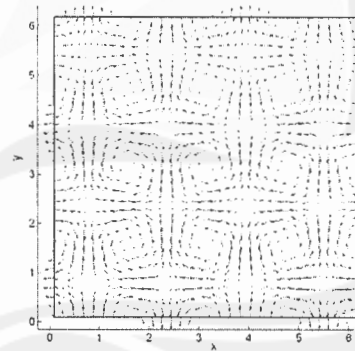


Figure 7. Velocity fields at t=2

We use 2-D system of size  $32 \times 32$  with periodic boundary condition in both directions. The simulation parameters are  $U_0 = 0.01$ ,  $k_x = k_y = 2$ ,  $\Delta t = 0.005$ , and  $\tau = 0.0018$ . The initial condition of velocity distribution

function  $f_i$  actually are unknown, it is not easy how to generate consistent initial condition of  $f_i$ . Research for generating consistent initial condition of  $f_i$  is still in progress [6]. In this paper, we use a simple approach, we use the equilibrium distribution function  $f_i^{eq}$  to initialize  $f_i$ .

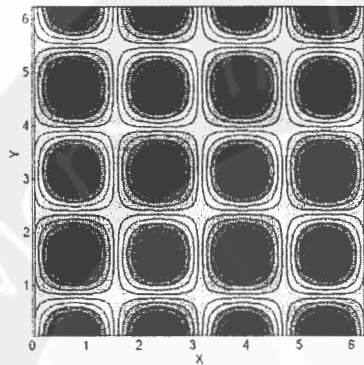


Figure 8. Density distribution at  $t=2$ .

Figure 7 and 8 show the computed results for velocity field and density at  $t = 2$ . Numerical and exact solutions of vertical velocity for  $t = 2$  and  $t = 150$  are compared in figure 8, showing excellent agreement.

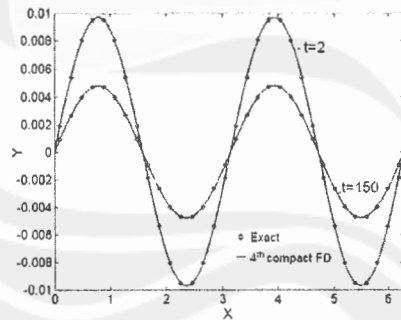


Figure 8. Comparison of vertical velocity at  $t=2$  and  $t=150$

We compared the vertical velocity error of 4<sup>th</sup> compact FD with explicit 2<sup>nd</sup> and 4<sup>th</sup> FD, the comparisons show that 4<sup>th</sup> compact FD much more accurate than 2<sup>nd</sup> and slightly more

accurate than explicit 4<sup>th</sup> FD. Figure 9a and 9b show the comparisons.

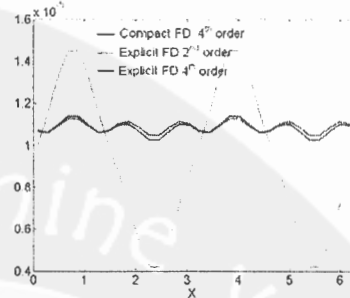


Figure 9a. Vertical velocity error at  $t=2$  and  $y=3.043$

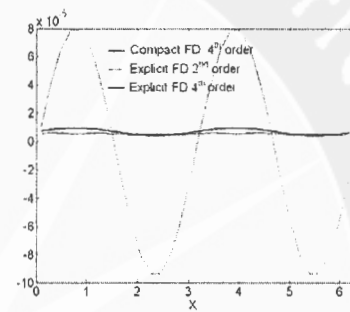


Figure 9b. Vertical velocity error at  $t=150$  and  $y=3.043$

Figure 10 shows the evolution of averaged error for 4<sup>th</sup> compact FD, 2<sup>nd</sup> and explicit 4<sup>th</sup> FD schemes. It can be seen that the averaged error of 4<sup>th</sup> compact FD and explicit 4<sup>th</sup> FD are almost equal and the averaged error of 2<sup>nd</sup> FD scheme is higher than others.

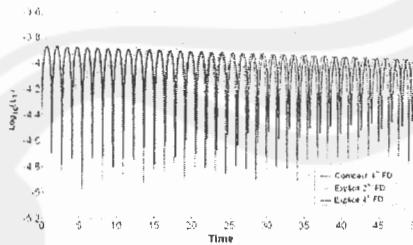


Figure 10. Convergence history of Taylor-Green vortex problem

## V. CONCLUSIONS

In this paper, we have presented a 4<sup>th</sup> compact finite difference method for solving two dimensional Discrete Boltzmann Equation. The proposed method has been verified for the 1-D convective equations and Taylor-Green vortex flows benchmark. The excellent agreement with exact solution and results of 2<sup>nd</sup> and explicit 4<sup>th</sup> FD shows the excellent accuracy and stability of the proposed method.

## ACKNOWLEDGEMENTS

The work described in this paper was fully supported by a grant from LPPM of Atma Jaya Yogyakarta University..

## REFERENCES

- [1]. Mei, R, and Shyy, W. (1998). "On the Finite Difference-Based Lattice Boltzmann Method in Curvilinear Coordinates," *Journal of Computational Physics*, 143, pp 426-448.
- [2]. Toelke, J. et al.(1998). "Implicit discretization and non-uniform mesh refinement approaches for FD discretizations of LBGK models," *Int. Journal of Modern Physics C*, 9(8), pp 1143-1157.
- [3]. Hesthaven, JS, Gottlieb, S, and Gottlieb D (2004). "*Spectral Methods for Time Dependent Problems*," Cambridge University Press, UK.
- [4]. Lele SK (1992). "Compact Finite Difference Schemes with Spectral-like Resolutions," *Journal of Computational Physics*, 103, pp 16-42.
- [5]. Wilson, RV, and Demuren, A O (2004). "*Higher-Order Compact Schemes for Numerical Simulation of Incompressible Flows*," NASA/CR-1998-206922, ICASE Report No. 98-13, Hampton, USA.
- [6]. Mei, R et al. (2006). "Consistent initial conditions for Lattice Boltzmann simulations," *Computers & Fluids*, 35, pp 855-682.

