# ANALISIS KINERJA INTEGRATOR SDIRK DALAM MENYELESAIKAN SISTEM PERSAMAAN DIFERENSIAL UNTUK PERSOALAN YANG KAKU (*STIFF PROBLEM*)

Alhadi B.* dan T. Basaruddin**


## ABSTRAK

**ANALISIS KINERJA INTEGRATOR SDIRK DALAM MENYELESAIKAN SISTEM PERSAMAAN DIFERENSIAL UNTUK PERSOALAN YANG KAKU (*STIFF PROBLEM*).** Pada tulisan ini akan dibahas salah satu metode Runge-Kutta implisit 4-tahap indeks-singly diagonally implicit runge-kutta (SDIRK) untuk menyelesaikan sistem persamaan diferensial biasa (SODE). Persoalan yang kaku (*stiff problem*) membutuhkan metode yang ukuran langkahnya tidak dibatasi oleh stabilitas metode. Kita mengharapkan metode ini bersifat A-stable karena tidak memiliki pembatasan stabilitas dalam menyelesaikan $y'=\lambda y$ dimana $\mathrm{Re}\lambda>0$ dan $h>0$, sehingga dengan memilih fungsi stabilitas yang sesuai kita akan memperoleh konstansta yang cocok ($\gamma$) dalam membentuk integrator SDIRK untuk menyelesaikan SODE. Untuk prediksi kesalahan digunakan metode dengan order yang lebih rendah, yaitu metode embedded 2-tahap yang diambil dari 2 tahapan internal SDIRK. Strategi pemilihan ukuran langkah diadopsi dari strategi yang diperkenalkan oleh Hall [1996:6]. Algoritma yang disusun dalam tulisan ini diimplementasikan menggunakan perangkat lunak MATLAB 5.3 yang berjalan di atas sistem operasi Windows95. Ukuran-ukuran uji kinerja yang digunakan adalah akurasi dari kesalahan pemotongan lokal; dan evaluasi efisiensi metode berdasarkan hasil-hasil statistik berupa: jumlah langkah (diterima/ditolak), jumlah pemanggilan fungsi, rata-rata iterasi Newton, dan waktu yang digunakan. Hasil dari eksperimen numerik menunjukkan bahwa SDIRK bersifat stabil tanpa syarat. Dengan menggunakan strategi pengontrolan langkah dari Hall dapat dilihat bahwa metode ini dapat diimplementasikan secara efisien, pada waktu menggunakan parameter yang sesuai.



## ABSTRACT

**INTEGRATOR PERFORMANCE ANALYSIS IN SOLVING STIFF DIFFERENTIAL EQUATION SYSTEM**. In this paper we discuss the four-stage index-2 singly diagonally implicit Runge-Kutta method, which is used to solve stiff ordinary differential equations (SODE). Stiff problems require a method where step size is not restricted by the method's stability. We desire SDIRK to be A-stable that has no stability restrictions when solving $y'= \lambda y$ with $\mathrm{Re}\lambda>0$ and $h>0$, so by choosing suitable stability function we can determine appropriate constant ($\gamma$) to formulate SDIRK integrator to solve SODE. We select the second stage of the internal stage as embedded method to perform low order estimate for error predictor. The strategy for choosing the step size is adopted from the strategy proposed by Hall(1996:6). And the algorithm that is developed in this paper is implemented using MATLAB 5.3, which is running on Windows95 environment. Our performance measurement's local truncation error accuracy, and efficiency were evaluated by statistical results of sum of steps, sum of calling functions, average of Newton iterations and elapsed times.As the results, our numerical experiment show that SDIRK is unconditionally stable. By using Hall's step size strategy, the method can be implemented efficiently, provided that suitable parameters are used.


* Jurusan Matematika, FMIPA – Universitas Indonesia
** Fakultas Ilmu Komputer – Universitas Indonesia

## PENDAHULUAN

Ada banyak permasalahan dalam dunia nyata yang dapat diformulasikan dalam bentuk persamaan diferensial. Bentuk persamaan diferensial tersebut pada umumnya dapat diselesaikan secara analitis ataupun pendekatan numerik, tetapi penyelesaian secara analitis tersebut biasanya cukup sulit sehingga untuk lebih memudahkan digunakan pendekatan solusi secara numerik. Akibatnya banyak sekali berkembang variasi metode dan integrator untuk solusi pendekatan numerik itu, termasuk di antaranya metode implisit Runge-Kutta dengan integratornya berupa *singly diagonally implicit Runge-Kutta* (SDIRK) [Burrage, K., 1999] yang akan akan dibahas dalam paper ini.

Tujuan dari penulisan paper ini adalah untuk membahas perancangan program dan analisa kinerja dari integrator SDIRK untuk mencari penyelesaian numerik dari sistem persamaan diferensial biasa yang kaku (*stiff ordinary differential equations-SODE*).

## METODE SDIRK

Metode implisit digunakan untuk mengatasi keterbatasan metode eksplisit dalam menyelesaikan problem yang kaku. Suatu masalah nilai awal (*initial value problems*-IVPs)

$$y'(x)=f(y(x)),\ y(x_0)=y_0,\ f{:}\mathbf{R}^{\mathbf{m}}\rightarrow\mathbf{R}^{\mathbf{m}}$$

dikatakan kaku bila $(x_f-x_0)L>>1$; $L$=Konsanta Lipschitz. Atau jika nilai ciri (*eigen value*) dari matriks *Jacobi*-nya negatif dan terdapat perbedaan yang besar antara bagian real dari nilai-nilai cirinya, yaitu:

$$S(x)=\frac{Max|\mathrm{Re}\,\lambda\,|}{Min|\mathrm{Re}\,\lambda\,|}>>1 \text{ [Spijker, 1996]}$$

Secara grafis jika untuk suatu masalah grafisnya turun (*decay*) maka akan terjadi penurunan dengan laju penurunan sebesar $|\lambda_i|$ artinya semakin besar $|\lambda_i|$ maka semakin besar penurunan yang terjadi sehingga untuk masalah yang kaku akibat perbedaaan yang besar antara nilai ciri $\lambda_i$ maka terjadi perubahan yang besar dan penurunan yang cepat pada selang tertentu. Untuk menjaga akurasi maka pada selang dengan penurunan yang cepat ini diperlukan strategi pengambilan ukuran langkah (*step size*) yang kecil, tetapi untuk selang tertentu yang penyelesaiannya berjalan dengan lambat atau konstan maka untuk lebih efisien digunakan ukuran langkah yang lebih besar.

Metode yang paling sering digunakan untuk menyelesaikan masalah yang kaku ini adalah metode implisit Runge-Kutta dengan berbagai variasinya termasuk SDIRK yang akan dibahas lebih lanjut dalam paper ini.

## INTEGRATOR SDIRK

Integrator SDIRK yang digunakan sebagai bahan penelitian dan pembahasan dalam paper ini diambil dari hasil penelitian bersama dari William R., Burrage K., Cameron I., and Kerr M., 1999 yaitu *four-stage index-2* SDIRK. Dalam bentuk tabel Butcher dapat dinyatakan sebagai berikut:

$$
\begin{array}{c|cccc}
0 & 0 & & & \\
2\gamma & \gamma & \gamma & & \\
1 & \frac{-6g+1+4g^{2}}{4g} & -\frac{2g-1}{4g} & \gamma & \\
1 & \frac{6g-1}{12g} & -\frac{1}{(24g-1} & -\frac{6g^{2}-6g+}{6g-3} & \gamma \\
\hline
\end{array}
$$

## ANALISA STABILITAS

Karena metode ini mempunyai sifat *A-stable* [Burrage K., 1999, 6] maka metode ini tidak mempunyai pembatasan stabilitas pada penyelesaian $y' = \lambda y$ dengan $\mathrm{Re}\lambda <$ dan $h > 0$. Kondisi ini diperoleh kalau fungsi stabilitas dari metode Runge-Kuttanya memenuhi:

$R(iy) \mid \leq 1$ untuk semua $y$ riil; dan

$R(z)$ adalah *analitic* untuk $\mathrm{Re}z < 0$

Untuk metode ini fungsi stabilitasnya menjadi:

$$
R(z)=\frac{(-g^{3}+3g^{2}-\frac{3}{2}g+\frac{1}{6})z^{3}+(3g^{2}-3g+\frac{1}{2})z^{2}+(-3g+1)z+1}{(1-zg)^{3}}
$$

Sehingga untuk memenuhi sifat *A-stable* kita bisa memilih $\gamma$ sebagai berikut:

$$\gamma \in [1/3, \theta], \text{ dengan } \theta \approx 1.06857902$$

dimana $\theta$ adalah akar nol terbesar dari:

$$\frac{1}{12} - y + 3y^2 - 2y^3$$

Jika kita memilih $\gamma \approx 0.43586652$ maka

$$\lim_{x \to \infty} R(z) = 0$$

maka metode ini akan menjadi L-stable, sehingga untuk permasalahan yang sangat kaku (*highly Stiff Problemsi*) maka metode ini akan dapat menahan kecepatan osilasi dari kesalahan komputasinya.

## ESTIMASI KESALAHAN LOKAL

Untuk estimasi kesalahan lokal dalam mengontrol panjang langkah digunakan metode *embedded second-stage* dari integrator SDIRK di atas yaitu:

$$\begin{array}{c|cc} 0 & 0 & \\ 2\gamma & \gamma & \gamma \\ \hline \end{array}$$

Karena metode ini juga bersifat *A-stable* dan *stiffly accurate* sehingga dapat digunakan dengan baik untuk memprediksi kesalahan solusi pada orde lebih rendah.

Kesalahan pemotongan lokal (LTE) yang diperoleh didefiniskan menjadi:

$$\text{LTE} = | y_{\text{four-stage}} - y_{\text{second-stage}} |$$

Jika dipilih $g = \frac{1}{2}$ dan $c_2 = 1$ maka fungsi stabilitasnya menjadi:

$$\hat{R}(z) = \frac{(1 + \frac{z}{2})}{(1 - \frac{z}{2})}$$

dan semua elemen dari integrator dapat dipenuhi sehingga bentuk akhir dari tabel Butcher menjadi:

| | | | | |
|---|---|---|---|---|
| 0 | 0 | | | |
| 1 | 1/2 | 1/2 | | |
| 3/2 | 5/8 | 3/8 | 1/2 | |
| 1 | 7/18 | 1/3 | -2/9 | 1/2 |

## STRATEGI PENGENDALIAN UKURAN LANGKAH

Untuk efisiensi maka ukuran langkah yang digunakan berubah-ubah sesuai berdasarkan nilai estimasi kesalahan lokal yang diperoleh. Jika kesalahan lokal yang diperoleh melebihi nilai toleransi (TOL) maka ukuran langkah dikurangi kemudian iterasi diulang kembali sampai memenuhi  nilai toleransi yang diberikan, sebaliknya jika nilai kesalahan lokal  yang diperoleh terlalu kecil maka untuk iterasi berikutnya ukuran langkahnya diperbesar dengan rumus

$$h_{n+1} = \frac{3.3}{\gamma}\left(\frac{\theta\, TOL}{\|\mathrm{LTE}\|}\right)^{1/p} h_n$$

dengan $2 \le \gamma \le 5$ dan $\theta = 0.5$ (Hall. 1995)

## ALGORITMA

Jika $\mathrm{K} = (K_1{}^T,\ldots,K_s{}^T)^T \in \mathrm{R}^{sm}$ dan $\mathrm{F(K)} =$
$(f(K_1)^T,\ldots f(K_s)^T)^T$

maka metode RK dapat ditulis dalam notasi tensor menjadi

$$K = e \otimes y_n + h(A \otimes I_m)F(K)$$
$$y_{n+1} = y_n + h(b^T \otimes I_m)F(K)$$

Sistem persamaan ini diselesaikan dengan skema modifikasi iterasi Newton. Jika $j = f'(y_n)$ dan pada akhir suatu iterasi, $K_i + \delta_i$, maka K pada setiap iterasi diperoleh dari penyelesaian sistem persamaan linier;

$$(I_s \otimes I_m - hA \otimes J)\mathrm{d} = \mathrm{y}$$

dengan

$$\mathrm{y}_i = -K_i + y_n + h\sum_{j=1}^{s} a_{ij} f(K_j)$$

$$\mathrm{d} = (\mathrm{d}_1^T, \ldots, \mathrm{d}_s^t)^T, \mathrm{y} = (\mathrm{y}_1^T, \ldots, \mathrm{y}_s^T)^T$$

Untuk mendapatkan nilai δ, sistem persamaan linier di atas diselesaikan dengan faktorisasi LU dari matrik ruas kiri yang diikuti dengan forward dan back substitusi. Algoritma lebih rinci adalah sebagai berikut:

```
ODESDIRK(ypfun, tspan, y0, gama, pow, tol, trace);
Tentukan integrator SDIRK untuk Y dan Ycap:
     [A,c,A1,c1]=intsdirk(gama);
Tentukan tebakan awal Y0, waktu awal t0 dan waktu akhir
tfinal
While t < tfinal
if t>tfinal-h then h=tfinal-t.
Hitung Ycap dengan integrator SDIRK A1.
[Ycap,fYcap,nws2f,nws2steps]=newtons2(ypfun,t,ycap,h,A1,Y
cap,tol,trace);
Hitung Y dengan integrator SDIRK A.
[Y,fY,nws4f,nws4steps]=newtons4(ypfun,t,y,h,A,Y,tol,trace
);
Hitung LTE
err=norm((ycap-y),inf);
If (5LTE5 < TOL) terima Y, lanjutkan integrasi t ‹  t+h,
Y0=Y;
If (5LTE5 tidak 0) hitung ukuran langkah baru;
hbaru←min(hmax,((3.3/4)*(0.5*tol/err)^(1/5))*hlama)
if (hbaru < hmin) stop
End While
4. IF t < tfinal, "singularity likely", stop
5. Display output: Y,h, statistics
6. END
```

## EKSPERIMEN NUMERIK

### Implementasi

Implementasi dilakukan dengan menggunakan MATLAB 5.2 dilingkungan Windows95. Program dibuat secara modular dengan parameter-parameternya yang memungkinkan untuk melakukan tes dan pemanggilan masing-masing fungsi-fungsi uji secara terpisah dari program utama. Dengan demikian kita dapat menggunakan beberapa fungsi uji dari MATLAB untuk stiffness problems (lihat lampiran).

### Masalah Uji

Uji coba dilakukan untuk mengevaluasi dan menganalisa dua masalah utama

1. Melihat pengaruh variasi nilai *p* dari strategi pengontrolan langkah dan variasi nilai gama dari integrator SDIRK. Berdasarkan (Burrage, 1999) nilai gama yang paling optimal adalah 0.435866, untuk mengevaluasi hasil ini maka dilakukan percobaan numerik dengan nilai gama yang lebih kecil, sama dan yang lebih besar dari 0.435866 kemudian dilakukan analisa terhadap kinerja yang terjadi.
2. Melihat kinerja dari integrator SDIRK dengan menggunakan nilai gama, *p* dan batas tolerasi kesalahan yang standar terhadap beberapa fungsi uji ODEs yang yang kaku. Fungsi-fungsi uji ini diambil langsung dari pustaka fungsi MATLAB 5.2 dalam direktori

   ..`\MATLABR11\toolbox\matlab\demos\`(lihat lampiran),

   selanjutnya dilakukan uji coba dan evaluasi kinerja dari Integrator SDIRK untuk masing-masing fungsi uji tersebut.

## HASIL DAN ANALISA

### Masalah Uji 1

Hasil eksperimen untuk evaluasi kinerja integrator dari masalah 1 dapat dilihat pada tabel 1. Sedangkan hasil plot grafik untuk solusi dan panjang langkah dapat dilihat pada lampiran. Dari eksperimen numerik ini dapat diperoleh beberapa kesimpulan awal bahwa kinerja integrator SDIRK menjadi lebih baik untuk nilai **p=3**, sedangkan nilai **gama = 0.43586652** seperti yang disarankan Burrage,1999 ternyata memberikan hasil yang lebih baik jika dibandingan dengan nilai gama yang lebih kecil atau yang lebih besar dari nilai tersebut.

Dari grafik terlihat bahwa ukuran langkah pada interval [0.5; 1] relatif lebih besar dan terus membesar secara monoton setelah fluktuasi solusi pada interval ini secara perlahan dan teratur berubah menjadi stabil, mendekati konstanta tertentu. Sebaliknya untuk interval [0;0.5] terlihat solusi berfluktuasi dengan cepat dan kaku akibatnya panjang langkah yang digunakan untuk daerah ini relatif lebih kecil.

Dari grafik terlihat juga bahwa perubahan nilai gama yang digunakan untuk integrator SDIRK memberikan efek pada besarnya panjang langkah yang digunakan. Makin besar nilai gama maka ukuran langkah menjadi relatif lebih kecil, rata-rata jumlah iterasi Newtonnya lebih kecil dan akurasi LTE meningkat tetapi berkibat pada menurunnya efisiensi metoda karena jumlah dan waktu iterasi meningkat dengan tajam.

Secara umum integrator ini dapat menangani persoalan ODE yang kaku dengan baik, terbukti dengan sedikitnya jumlah langkah yang ditolak dan nilai maksimum LTE yang cukup kecil pada masing-masing kasus.

Kinerja terburuk untuk data percobaan ini diperoleh untuk p=4 dan gama=2*0.43586652, sedangkan kinerja terbaik diperoleh untuk p=3 dan gama=0.43586652.

Tabel 1. Data Statistik Percobaan Numerik untuk Variasi Pow dan Integrator SDIRK

| No | Parameter | | Jml. Iterasi | Jumlah *h* *d*iterima | Jumlah *h* ditolak | Jml Iterasi Newton | Rata-rata Iterasi Newton | Waktu Iterasi (second) | LTE maks |
|---|---|---|---|---|---|---|---|---|---|
| | pow | gama SDIRK | | | | | | | |
| 1 | 1/4 | 0.8*gama | 3287 | 3282 | 5 | 6574 | 1.9724 | 79.75 | 7.7749e-005 |
| 2 | 1/4 | gama | 2503 | 2500 | 3 | 4957 | 1.9341 | 61.51 | 5.6838e-005 |
| 3 | 1/4 | 2*gama | 41573 | 41567 | 6 | 41581 | 0.9994 | 851.95 | 7.6169e-005 |
| 4 | 1/3 | 0.8*gama | 2726 | 2722 | 4 | 5452 | 1.9333 | 61.68 | 3.6464e-005 |
| 5 | 1/3 | gama | 2064 | 2061 | 3 | 4119 | 1.8903 | 47.18 | 1.54759e-004 |
| 6 | 1/3 | 2*gama | 34668 | 34663 | 5 | 34674 | 0.99945 | 679.49 | 2.87998e-005 |

**Default: Toleransi: Tol = 1.e-3; Gama Integrator SDIRK: gama = 0.43586652**
**(** 1/3 < gama < 1.06857902 Burrage, 1999**)**

**Masalah Uji 2**

Hasil eksperimen untuk evaluasi kinerja integrator dari masalah 2 dapat dilihat pada tabel 2. Dari tabel 2 dapat diamati bahwa integrator SDIRK dapat menangani kedelapan fungsi uji ODE yang kaku dengan baik. Hal ini dapat dilihat dari jumlah langkah yang ditolak relatif sangat kecil << 10 dibandingkan dengan jumah iterasinya. Rata rata jumlah iterasi newton juga relatif kecil < 2 dan maksimum LTE yang diperoleh juga relatif kecil dari toleransi default yang diberikan << 1.e-003.

Strategi pengontrolan langkah untuk masalah kekakuan juga dapat berjalan dengan baik. Dari grafik masing-masing solusi dan grafik panjang langkah *h* dapat dilihat bahwa ukuran langkah relatif kecil pada interval yang solusinya menunjukkan gejala kekakuan/berfluktuasi secara tajam. sebaliknya untuk interval yang solusinya relatif stabil/mendekati konstanta tertentu terlihat ukuran langkah yang digunakan relatif lebih besar. Namun demikian, ukuran langkah akan selalu berubah baik dalam kondisi diterim ataupun ditolak.

Untuk persoalan kekakuan pada ODE yang solusinya berulang secara periodik (mis: Van der Poll ODE) dapat dilihat bahwa strategi pengontrolan langkahnya juga berulang secara periodik membesar atau mengecil mengikuti gejala kekakuan pada solusi tersebut.

Secara umum integrator SDIRK ini dapat menangani persoalaan kekakuan pada ODE dengan baik, baik untuk kasus fungsi uji dengan jumlah variabel yang relatif kecil $\leq 2$ (VDPODE, A3ODE,.CHM7ODE), jumlah varibel yang sedang $\leq 6$ (CHM6ODE, B5ODE, CHM9ODE), dan jumlah variabel yang relatif cukup besar > 7 (BRUSSODE, A2ODE)

Penulis telah melakukan perbandingan masing-masing kasus dengan variasi nilai *p* dan gama dari integrator SDIRK, tetapi karena keterbatasan yang ada hasilnya tidak penulis munculkan dalam paper ini, namun demikian dari hasil uji coba sementara yang penulis lakukan ternyata nilai **gama = 0.43586652** memberikan kinerja yang paling baik untuk beberapa kasus yang penulis teliti tersebut.

Tabel 2: Data Statistik Percobaan Numerik untuk Variasi Fungsi ODEs

| No | ODE | Jml. Iterasi | Jumlah *h* diterima | Jumlah *h* ditolak | Jml Iterasi Newton | Rata-rata Iterasi Newton | Waktu Iterasi (second) | LTE maks |
|---|---|---|---|---|---|---|---|---|
| **1** | **PDVODE** | 7501 | 7498 | 3 | 15002 | 1.9857 | 151.76 | 2.28374e-004 |
| **2** | **CHM6ODE** | 11764 | 11759 | 5 | 23528 | 1.9971 | 257.44 | 2.28374e-004 |
| **3** | **B5ODE** | 2064 | 2061 | 3 | 4119 | 1.8903 | 47.18 | 1.54759e-004 |
| **4** | **BRUSSODE** | 5764 | 5760 | 4 | 11530 | 1.9569 | 196.9 | 2.47750e-004 |
| **5** | **CHM7ODE** | 439 | 436 | 3 | 625 | 1.2136 | 6.59 | 2.27340e-004 |
| **6** | **CHM9ODE** | 13655 | 13651 | 4 | 27310 | 1.9934 | 278.58 | 4.96731e-005 |
| **7** | **A2ODE** | 7604 | 7599 | 5 | 9766 | 1.2708 | 165.05 | 2.25335e-004 |
| **8** | **A3ODE** | 39460 | 39453 | 7 | 52108 | 1.3172 | 818.67 | 1.04599e-004 |

**Default: Toleransi: Tol = 1.e-3; pow=1/3; Gama Integrator SDIRK: gama = 0.43586652 (** 1/3 < gama < 1.06857902 Burrage, 1999**)**

## KESIMPULAN

Dari hasil eksperimen dapat disimpulkan hal-hal berikut ini:

1. Integrator SDIRK untuk batasan nilai gama antara 1/3 s.d. 1.06857902 dalam eksperimen ini bersifat stabil tanpa syarat dan memberikan kinerja yang paling baik untuk **gama = 0.43586652.**
2. Ukuran panjang langkah yang digunakan relatif kecil untuk interval daerah solusi yang kaku dan sebaliknnya ukuran panjang langkah relatif besar/membesar pada daerah yang stabil/ tidak kaku.
3. Perbedaan nilai *p* dan nilai gama yang digunakan berdampak langsung pada kinerja, akurasi dan efiseiensi metode. Secara umum nilai *p* yang kecil memberikan kinerja yang lebih baik, tetapi keakuratannya berkurang. Sedangkan nilai gama yang lebih lebih besar menyebabkan panjang langkah yang digunakan relatif lebih kecil sehingga keakuratannya meningkat, tetapi kinerjanya menjadi lambat karena jumlah iterasi meningkat dengan tajam.
4. Secara umum kinerja integrator SDIRK ini baik karena jumlah langkah yang ditolak sangat kecil (<< 10), baik untuk kasus-kasus fungsi uji dengan jumlah variabelnya kecil (≤2), sedang (≤6), dan cukup besar (>7).
5. Ukuran panjang langkah selalu berubah baik untuk kondisi ditolak atau diterima.
6. Ukuran panjang langkah akan berubah mengecil dan membesar secara periodik jika solusi fungsi uji ODE tersebut juga bersifat kaku secara periodik.

## DAFTAR PUSTAKA


1. BURRAGE, K., WILLIAM R., CAMERON I., KERR M.**,** *A-Four Stage Index 2 Diagonally Implicit Runge-Kutta Method*, Laporan riset dari CAPEC (Computer Aided Process Engineering Centre), University of Quensland, Brisbane Australia, (1999)

2. BURRAGE K., *Parallel and Sequential Mtehods for Ordinary Differential Equations*, Clarendon Press., Oxford, (1995)

3. CAVALLO A., SETOLA R., VASCO F., *Using MATLAB, SIMULINK and Control System Toolbox*, Prentice Hall, (1996)

4. D. HANSELMEN, B. LITTLEFIELD**,** *Mastering MATLAB 5, A Comprehensive Tutorial and Reference,* Prentice Hall, (1998)

5. E. HAIRER, S.P. NORSETT, G. WANNER, *Solving Ordinary Differential Equations*, Vol. II, Springer-Verlag, Berlin, (1989)

6. HALL G., *A new step size strategy for Rungge-Kutta Code,* Dalam Numerical Analysis Report No **243** Dept of Mathematics -UMIST

7. L. F. SHAMPINE, *Numerical Solution of Ordinary Differential Equations*, Chaman & Hall, (1994)

8. L. R. PETZOLD, Computer Method for Ordinary Differential Equations and Differential-Algebraic Equations, Siam Publ. (1998)