Ministry Of high Education
and Scientific Research
University of Babylon
Dep of Mathematics

Stiff differential equations

بحث مقدم الى كلية التربية للعلوم الصرفة قسم الرياضيات وهو جزء من متطلبات نيل درجة البكالوريوس في علوم الرياضيات

Preparing by

Haider Talib

Supervisor

Dr. Bushra H.Aliwi

الإهــداء

إلى من أُفضِّلها على نفسي، ولِمَ لا؛ فلقد ضحَّت من أجلي

ولم تدَّخر جُهدًا في سبيل إسعادي على الدَّوام

(أُمِّي الحبيبة).

نسير في دروب الحياة، ويبقى من يُسيطر على أذهاننا في كل مسلك نسلكه

صاحب الوجه الطيب، والأفعال الحسنة.

فلم يبخل عليَّ طيلة حياته

(والدي العزيز).

إلى أصدقائي، وجميع من وقفوا بجواري وساعدوني بكل ما يملكون، وفي أصعدة كثيرة

أُقدِّم لكم هذا البحث، وأتمنَّى أن يحوز على رضاكم.

الشكر والتقدير

لا يمكن للكلمات أن تعبر عن امتناني لأستاذي ورئيس لجنتي لما أبداه من صبر وتعليقات لا تقدر بثمن. كما أنني لم أتمكن من القيام بهذه الرحلة بدون لجنة الدفاع الخاصة بي ، التي قدمت بسخاء المعرفة والخبرة. بالإضافة إلى ذلك ، لم يكن هذا المسعى ممكنًا لولا الدعم السخي من مؤسسة...، التي مولت بحثي. (نموذج شكر وتقدير)

كما أنني ممتن لزملائي وأعضاء المجموعة ، وخاصة زملائي في المكتب ، لمساعدتهم في التحرير ، وجلسات النقاش في وقت متأخر من الليل ، والدعم المعنوي. لذا يجب أن نتوجه بالشكر أيضًا إلى أمناء المكتبات ومساعدي الأبحاث والمشاركين في الدراسة من الجامعة ، الذين أثروا فيّ وألهموني.

أخيرًا ، سأكون مقصراً في عدم ذكر عائلتي ، وخاصة والديّ وزوجتي وأطفالي. لقد أبقى إيمانهم بي معنوياتي ودوافعي عالية خلال هذه العملية. أود أيضًا أن أشكر قطتي على كل الدعم الترفيهي والعاطفي.

بِسْمِ اللَّــهِ الرَّحْمَـٰنِ الرَّحِيمِ

(هُوَ الَّذِي جَعَلَ الشَّمْسَ ضِيَاءً وَالْقَمَرَ نُورًا وَقَدَّرَهُ مَنَازِلَ لِتَعْلَمُوا عَدَدَ السِّنِينَ وَالْحِسَابَ مَا خَلَقَ اللَّهُ ذَٰلِكَ إِلَّا بِالْحَقِّ يُفَصِّلُ الْآيَاتِ لِقَوْمٍ يَعْلَمُونَ). سورة يونس- الآية ٥.

Abstract :

Stiff differential equations refer to a type of ordinary differential equations where the solution exhibits widely varying time scales. These equations often arise in systems with reactions or processes characterized by significantly different rates. Stiffness poses challenges for numerical solvers as it requires small step sizes to capture the fast and slow dynamics accurately. Specialized methods, such as implicit schemes or stiff solvers like Gear's method, are employed to efficiently solve stiff ODEs. Understanding and effectively handling stiff differential equations are crucial in various fields, including chemical .kinetics, biological systems, and physical simulations

1- Introduction

In common engineering applications such as fast equilibrium reactions, catalysis, adsorption and evaporation the solution of systems of stiff ordinary differential equations is of key importance. In an attempt to solve such systems of equations encountered in CFD applications
a standalone software module, Stiff ODE Suite, was developed and coupled to the commercial CFD code FLUENT v12.1, Ansys inc. The functionality of the hybrid software package was validated for usage in common engineering applications.

The Stiff ODE Suite software package is based on an adaptive time step maximum fifth order Backwards Differential Formulation. Said algorithm introduces a validated and versatile stiff and non-stiff ODE solver potentially outperforming MATLABs ODE15s. Functioning as a
standalone module the Stiff ODE Suite is however best applied in a hybrid scheme with FLUENT v12.1 or other compatible CFD codes. Such hybrid couplings vastly increase the applicability of otherwise limited CFD codes.

The Stiff ODE Suite was used to with satisfactory accuracy predict solution pH in neutralization of hydrochloric acid using a sodium carbonate solution. The water auto proteolysis is an extremely fast equilibrium reaction posing a significant challenge to any
chemical reaction engineering solver. The Stiff ODE Suite also enabled the simulation of rapid humidification of dry heated air using a water droplet jet in a pipe segment.

In this simulation the Stiff ODE Suite proved capable of coupling fast mass and heat transfer between phases in multiphase applications.

By introduction of the Stiff ODE Suite the CFD engineer is able to use much larger time steps in transient simulations than the time scale of the stiff system of ODEs governing the solution.

In addition the software module largely extends the applicability of FLUENT v12.1 since the user is no longer limited to what can be described using the graphical user inter phase

2- Stiff ODE Suite mathematical *formulation*

The system of equations solvable in the Stiff ODE Suite framework is presented on a general form in equation.

$$\dot{y} = f(y,t), \quad y(t_0) = y_0, \quad y \in \vec{R}^N$$

However not restricted to autonomous problem formulations the Stiff ODE Suite gains additional numerical precision by assuming time independency resulting in the code itself being optimized according to equation. Note the differences in the formulation of the derivatives. As most conceivable real physical systems are autonomous this limitation results in numerical gains with no obvious drawbacks.

$$\dot{y} = f(y), \quad y(t_0) = y_0, \quad y \in \vec{R}^N$$

In order to solve the system of equations in equation 2.50 a general algorithm is needed. The Stiff ODE Solver is based on a variable order Backwards Differential Formulation (BDF) similar to the algorithm used in ODE15s in MATLAB. Equation 2.51 outlines the BDF formulation.

$$\sum_{i=0}^{K_1} \alpha_{n,i} y_{n-i} + h_n \sum_{i=0}^{K_2} \beta_{n,i} \dot{y}_{n-i} = 0$$
$$h_n = t_n - t_{n-1}$$

For stiff systems of equations, which essentially is why Stiff ODE Suite was constructed to begin with, *K* = *q* 1 and 0 2 *K* = . q denotes the order of the BDF formulation and varies between 1 and 5 depending on system characteristics. In each time step the non linear system of equations

$$G(y_n) = y_n - h_n \beta_{n,0} f(t_n, y_n) - a_n$$
$$a_n = \sum_{i>0} \left(\alpha_{n,i} y_{n-i} + h_n \beta_{n,i} \dot{y}_{n-i} \right)$$

In solving this system of equations the Stiff ODE Suite uses a Newton iteration scheme outlined.

$$M(y_{n(m+1)} - y_{n(m)}) + G(y_{n(m)}) = 0$$
$$M = \mathrm{I} - \gamma J$$
$$J = \frac{\delta f}{\delta y}$$
$$\gamma = h_n \beta_{n,0}$$

The algorithm itself, originally proposed by Bill Gear, is advanced and regarded as one of the

most versatile solution strategies to stiff systems of ordinary differential equations available.

The solution methodology is to initially decompose the integration time [0,T] into non equidistant time steps shown in equation. On each time step equation inserted into essentially forms an independently solvable predictor equation shown in.

$$y_n^p = \sum_{i=1}^{k} \gamma_i y_{n-i}$$

Using the approximate solution *p*

n y at time *n t* as an initial guess in iterating equation results in the real intermediate solution $n\ y$ at time $n\ t$. With both solutions available a local truncated error can be constructed and compared to a predefined user specified value. In the case of too large an error the algorithm will as a first measure if possible increase the order of the BDF formulation. Secondly it will decrease the internal time step and the solution procedure is repeated. The interested reader is referred to the thorough work of Stabrowski for a full description of the BDF based algorithm.

Summing up the algorithm the Stiff ODE Suite supplies a versatile solution strategy via automatically adapted time steps, an economical solution methodology via problem specific BDF order and a most importantly reliable performance comparable with or better than ODE15s.

It should be noted that considerable effort has been put into validating the functionality of the standalone solver as introduced by the Stiff ODE Suite. No discrepancies in the algorithm or code have yet been found. [4]

1. General Formulation: The system of ODEs is represented as:

$$dy/dt = f(y, t), y(0) = y0$$

where **y** is a vector of dependent variables, **t** is the independent variable (time), **f** is the vector-valued function representing the right-hand side of the ODEs, and **y0** is the initial condition vector.

2. Time Independency Assumption: To improve numerical precision, the Stiff ODE Suite assumes time independency, resulting in the code optimization. This assumption does not restrict the suite to autonomous problems.

3. Backwards Differential Formulation (BDF): The Stiff ODE Solver uses a variable order BDF algorithm, similar to MATLAB's ODE15s. The BDF formulation is given by:

$$\sum [\alpha i\ yn + hi\ f(yn)] = yn+1$$

where **αi** and **hi** are coefficients, **yn** and **yn+1** represent the solutions at time **tn** and **tn+1**, respectively, and **f(yn)** is the evaluation of the derivative function at **yn**.

4. Nonlinear System of Equations: In each time step, a nonlinear system of equations is constructed and solved using a Newton iteration scheme. The system of equations is given by:

$$\sum [\gamma i\ (yn+1 - yn - hi\ f(yn+1))] = 0$$

where **γi** are coefficients and **f(yn+1)** is the evaluation of the derivative function at **yn+1**.

5. Newton Iteration Scheme: The Newton iteration scheme is used to solve the nonlinear system of equations and obtain the solution **yn+1**. It involves computing the Jacobian matrix **J** and performing iterations until convergence.
6. Predictor Equation: For each time step, a predictor equation is formed using the BDF formulation, which can be solved independently to obtain an initial guess for the solution at **tn+1**.
7. Error Control: After obtaining both the predictor solution and the actual solution, a local truncated error is calculated and compared to a predefined user-specified value. If the error is too large, the algorithm may increase the BDF order or decrease the internal time step to improve accuracy.

Applications The Stiff ODE Suite aims to provide a versatile and efficient solution strategy for stiff systems of ODEs by adapting time steps, selecting problem-specific BDF orders, and ensuring reliable performance comparable to or better than MATLAB's ODE15s solver.

For a more detailed description of the BDF-based algorithm, you can refer to the work of Stabrowski, as mentioned in the excerpt.

Further details are given in section 5.1.

With the time step or the lack thereof determined the actual theory behind the FLUENT – Stiff ODE Suite coupling can be outlined. Theoretically a control volume in the computational domain can be regarded as a small semi-continuous batch reactor governed by equation 2.60.

$$\int_V \frac{\partial \rho \phi}{\partial t} dV + \oint \rho \phi \vec{v} \cdot d\vec{A} = \oint \Gamma_\Phi \nabla \phi \cdot d\vec{A} + \int_V S_\phi dV \qquad (2.60)$$

Additional information on the terms in the equation outlined above is to be found in section 2.2. In this case the term of interest is the source term F S as it describes generation and decay of a species Φ due to chemical reactions or any other physical event.

The proposed method behind the Stiff ODE Suite is both versatile and stable and uses the large benefits of linearizing the source term on the time step at hand. In figure 2.1 an arbitrary system reaching equilibrium on a very small time scale is illustrated.

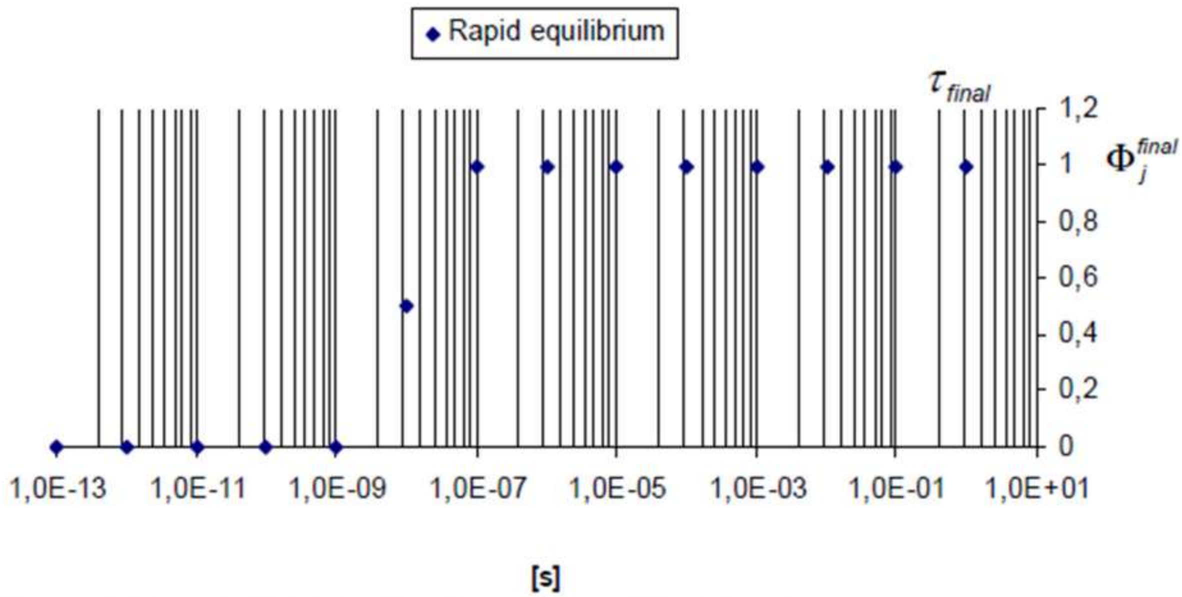

Figure 2.1 Characteristic behavior of a rapidly equilibrated system

As seen in figure 2.1 the equilibrium is reached on a scale of nano seconds. Without using the Stiff ODE Suite this means that FLUENT needs to progress with a time step of equivalent size in order to resolve the extremely rapid change occurring on this time scale. In addition

experience shows that even at a time step of this magnitude the solution will tend to be unstable. Using the Stiff ODE Suite makes it possible to resolve the source term SΦ and the simulation can hence progress with a time step determined by the flow instead of the fast reactions in the system. Characteristically the flow time step is in the vicinity of 1ms rather than 1ns giving a hint of the computational benefits of using the Stiff ODE Suite.

Example 1:

A kinetics problem: The following kinetics problem, given by Robertson, is frequently used as an illustrative example. It involves the following three nonlinear rate equations:

$$y_1' = -.04 y_1 + 10^4 y_2 y_3 \tag{8}$$

$$y_2' = .04 y_1 - 10^4 y_2 y_3 - 3.10^7 y_2^2 \tag{9}$$

$$y_3' = 3.10^7 y_2^2 \tag{10}$$

The initial values at $t = 0$ are

$$y_1(0) = 1, \quad y_2(0) = y_3(0) = 0 \tag{11}$$

Since $\sum y_i' = 0,$ the solution must satisfy $\sum y_i = 1,$

dentically. This identity can be used as an error check

Here we intend to solve this problem with the BDF method and use the chord of iteration method with the user-supplied Jacobian (MITER=1). $J = \partial f / \partial y$ Suppose a local error bound of EPS = 10^{-6}, and control absolute error (IERROR=1). We choose an initial step size of H0=10^{-8} . The use of MITER=1 requires that the Jacobian be calculated and programmed. This is given by

$$J = \begin{pmatrix} -.04 & 10^4 y_3 & 10^4 y_2 \\ .04 & -10^4 y_3 - 6.10^7 y_2 & -10^4 y_2 \\ 0 & 6.10^7 y_2 & 0 \end{pmatrix}$$

The final value of t is 40. So we consider taking output at $t = 4 \text{ x } 10^k$,

where k = -1, 0, 1, 2 , …. These will be the values of the argument TOUT.

The following coding, together with the EPISODE package, can be used to solve this problems with the options described above.

The output of the above program in tabular form is as follows

Table 2 : MF=21, EPS=10^{-6}

T	H	Y_1	Y_2	Y_3	SUM(Y)-
0.4E+00	0.16E+00	0.98517E+00	0.33864E-04	0.14794E-01	-0.4E-15
0.4E+01	0.56E+00	0.90552E+00	0.22405E-04	0.94462E-01	-0.5E-15
0.4E+02	0.23E+01	0.71582E+00	0.91851E-05	0.28417E+00	-0.3E-16
0.4E+03	0.20E+02	0.45051E+00	0.32228E-05	0.54949E+00	-0.5E-15
0.4E+04	0.24E+03	0.18320E+00	0.89423E-06	0.81680E+00	-0.2E-15
0.4E+05	0.33E+04	0.38986E-01	0.16219E-06	0.96101E+00	0.3E-15
0.4E+07	0.38E+06	0.50319E-03	0.20660E-08	0.99948E+00	0.3E-15
0.4E+10	0.12E+10	0.54561E-06	0.37316E-11	0.10000E+01	-0.1E-13

We see that the equilibrium values are

$y1 = y2 = 0$, $y3 = 1$

and that the approach

to equilibrium is quite slow. Here we note that the time step, H, rises steadily with time,

T. We also observe that the code generated negative and thus physically incorrect answers

during the last decade. This reflects instability, or a high sensitivity of the problem to

numerical errors at late t, and will, if the integration is continued, lead to answers

diverging to $\pm\infty$.

The accuracy of the above result can be verified in the usual way by Re running the program with a smaller value of EPS=10^{-9} and nothing else changed, the output in the tabular form is as follows

Table 3 : MF=21, EPS=10^{-9}

T	H	Y_1	Y_2	Y_3	SUM(Y)-
0.4E+00	0.34E-01	0.985172E+00	0.338641E-04	0.147940E-01	0.2E-15
0.4E+01	0.14E+00	0.905519E+00	0.224048E-04	0.944589E-01	0.5E-15
0.4E+02	0.13E+01	0.715827E+00	0.918552E-05	0.284164E+00	0.6E-15
0.4E+03	0.82E+01	0.450519E+00	0.322290E-05	0.549478E+00	0.8E-15
0.4E+04	0.76E+02	0.183202E+00	0.894237E-06	0.816797E+00	0.1E-14
0.4E+05	0.88E+03	0.389834E-01	0.162177E-06	0.961016E+00	0.9E-15
0.4E+07	0.20E+06	0.516813E-03	0.206835E-08	0.999483E+00	0.1E-14
0.4E+10	0.67E+09	0.522363E-06	0.208942E-11	0.999999E+00	0.1E-14

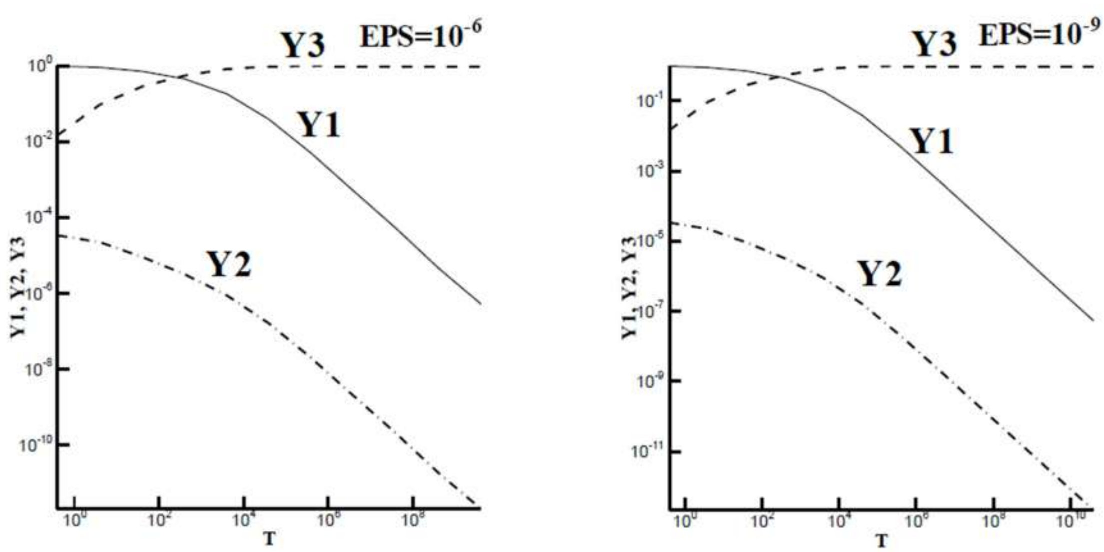

Fig. 1 : The graph of the approximated solution of Example 1 (by using log scale)

Now we consider another example of stiff system of differential equations which can be solved analytically.

Example 2: The system of initial-value problems

$$u_1' = 9u_1 + 24u_2 + 5\cos t - \frac{1}{3}\sin t, \qquad u_1(0) = \frac{4}{3} \tag{12}$$

$$u_2' = -24u_1 - 51u_2 - 95\cos t + \frac{1}{3}\sin t, \qquad u_1(0) = \frac{2}{3} \tag{13}$$

has the unique solution

$$u_1(t) = 2e^{-3t} - e^{-39t} + \frac{1}{3}\cos t. \tag{14}$$

$$u_2(t) = -e^{-3t} + 2e^{-39t} - \frac{1}{3}\cos t \tag{15}$$

The transient term e^{-39t} in the solution causes this system to be stiff.

The results, obtained by EPISODE are summarized in the following table.

Table 4

t	h	Approximated value of $u_1(t)$	Approximated value of $u_2(t)$	Exact value of $u_1(t)$	Exact value of $u_2(t)$
0.0	.40E-02	1.33333333	0.666666666	1.33333333	0.666666666
0 .1	.40E-02	1.79306146	-1.03200020	1.79306300	-1.03200200
0.2	.91E-02	1.42390205	-0.87468033	1.42390200	-0.87468100
0.3	.12E-01	1.13157624	-0.72499799	1.13157700	-0.72499860
0.4	.33E-01	0.90940824	-0.60821345	0.90940860	-0.60821420
0.5	.33E-01	0.73878794	-0.51565752	0.73878780	-0.51565770
0.7	.66E-01	0.49986115	-0.37740429	0.49986030	-0.37740380
1.0	.66E-01	0.27968063	-0.22989065	0.27967490	-0.22988780

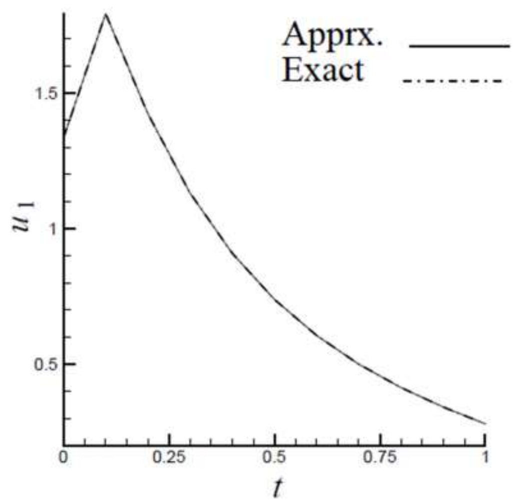

Fig. 2(a) : The graph of the solutions for u_1

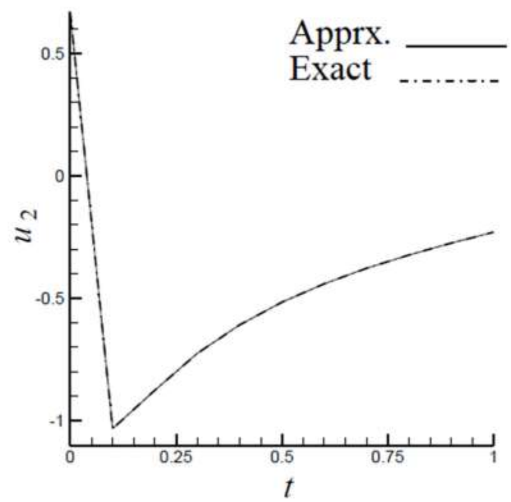

Fig. 2(b) : The graph of the solutions for u_2

3- Stiff differential equations have applications in various scientific and engineering fields. Some common areas where stiff ODEs arise and are studied include:

1. Chemical Kinetics: Stiff ODEs are frequently encountered in modeling chemical reaction networks, where different reactions occur at significantly different rates.
2. Biological Systems: Stiff ODEs are used to model biochemical networks and gene regulatory networks, where multiple reactions or interactions occur simultaneously with different time scales.
3. Physical Simulations: Stiff ODEs are relevant in simulating physical systems, such as fluid dynamics, combustion processes, and plasma physics, where the dynamics involve a wide range of time scales.
4. Circuit Simulation: Stiff ODEs are encountered in electrical circuit simulations involving components with widely varying time constants, such as capacitors, inductors, and transistors.
5. Astrophysics and Cosmology: Stiff ODEs arise in modeling complex systems like stellar evolution, galaxy formation, and the early universe, where different physical processes occur with different time scales.
6. Control Systems: Stiff ODEs are used in the modeling and analysis of control systems, where different components or subsystems exhibit dynamics with varying time scales.

7. Pharmacokinetics: Stiff ODEs are employed in pharmacokinetic modeling, which involves studying the absorption, distribution, metabolism, and excretion of drugs in the body. These processes often occur at different rates and require stiff ODE models for accurate simulations.

8. Environmental Modeling: Stiff ODEs find applications in environmental modeling, such as atmospheric chemistry models, climate models, and pollutant transport models. These systems involve complex interactions with varying time scales.

9. Power Systems: Stiff ODEs are relevant in power system analysis, particularly in transient stability studies, where the dynamics of generators, transmission lines, and loads exhibit different time scales.

10. Population Dynamics: Stiff ODEs are used in studying population dynamics, such as predator-prey models or epidemiological models, where the rates of birth, death, and interactions between species or individuals occur at different rates.

4- References

- Henrik Alfredsson - Solving stiff systems of ODEs in CFD applications Developing and evaluating the Stiff ODE - (2010)

- Sharaban Thohura & Azad Rahman - Numerical Approach for Solving Stiff Differential Equations: A Comparative Study - (2013)

- Hairer, E.Wanner - G. Solving Ordinary Differential Equations II: Stiff and Differential - Algebraic Problems - (1996).

- John Wiley & Sons - Lambert, J. D. Numerical methods for ordinary differential systems - (1991)