線形陰的な微分代数方程式で表されるシステムのための多倍長シミュレーションパッケージ

A Multiple Precision Simulation Package for Systems with Linearly Implicit Differential-Algebraic Equation

九州工業大学　○ 林 裕之，古賀 雅伸，野口 剛史
福岡工業大学短期大学部　矢野 健太郎
H.Hayashi, M.Koga and G.Noguchi
Kyushu Institute of Technology
K.Yano
Fukuoka Institute of Technology, Junior College

Abstract In this paper, we introduce a multiple precision simulation package for complex systems represented by differential-algebraic equation. By using this package, it is possible to compare and review the numerical solution of lineary implicit differential-algebraic equation, because switching the solver according to format and index of equation, or changing calculation accuracy.

1 はじめに

機構系や流体系，電気系などの多くのシステムは，ダイナミクスを表す常微分方程式と制約条件を与える代数方程式を連立した微分代数方程式 (DAE:Differential-Algebraic Equation) で表現できる．そして，Simulink や MapleSim，Mathematica などのモデリング・シミュレーションツールから微分代数方程式に対するソルバを利用できる．

また，近年のシステムの高度化・複雑化に伴い，微小な丸め誤差が予想もしないような大きさに拡大されてしまう悪条件な問題や整数型 int の最大数あたりで起こる奇妙な振る舞いの抑制，数値計算を十分な精度で行えているかの検証などで多倍長演算の必要性が高まってきている．実際に，様々なプログラミング言語において多倍長数値計算ライブラリは充実してきており，MapleSim や Mathematica などのモデリング・シミュレーションツールも多倍長演算に対応している．

本研究では，モデリング・シミュレーションツール Jamox[1] への導入を視野に入れた，線形陰的な DAE で表されるシステムのための多倍長シミュレーションパッケージを開発した．本パッケージを用いることにより，方程式の形式や指数に応じてソルバを切り替えたり，演算精度を変更したりすることで，線形陰的な DAE の数値解を比較・検討することができる．

パッケージ内のソルバーに関しては，Fotran で記述されたソルバと DAE 問題を提供している Test Set for Initial Value Problem Solvers[2] を参考に Java 言語で実装を行った．また，例題による性能評価として Fortran コードとの計算結果の比較を行った．

2 DAE

DAE とは，代数方程式の制約条件の影響を受ける常微分方程式である．DAE 問題は一般的に陰的常微分方程式

$$F(t, y, y') = 0 \tag{1}$$

で与えられる．本研究では，式 1 が

$$My' = f(t, y)$$

のような線形陰的な形式に変換できる場合について考える．ただし，M は非正則定数行列である．

2.1 指数

DAE を理解する上で重要なものとして指数が挙げられる．指数とは，y' を y と t に関して解くときに必要な連立方程式に対する最小の微分回数のことである [3]．例えば，連立方程式

$$y_1 = q(t)$$
$$y_2 = {y_1}'$$

に対して，y_2 の導関数を得るために連立方程式を微分すると

$${y_1}' = q'(t)$$
$${y_2}' = {y_1}''$$

となる．y_1'' を得るために，もう一度微分すると

$$y_2' = y_1'' = q''(t) \tag{2}$$

となり，$q(t)$ の二回微分が必要だったので指数は 2 となる．

3　シミュレーションパッケージ

本章では，DAE のシミュレーションを行うために作成したパッケージの説明を行う．

3.1　Test Set for Initial Value Problem Solvers

開発では，Test Set for Initial Value Problem Solver で紹介されている Fortran で提供された DAE ソルバを参考に実装を行う．提供されているソルバと問題は，DAE の他に常微分方程式 (ODE) や陰的微分方程式 (IDE) のものも存在する．ここで，表 1 に提供されているソルバの一覧を示す．

表 1: DAE ソルバ一覧

ソルバ名	指数	問題の型
VODE	0	ODE
BIMD	3 以下	ODE,DAE
GAMD	3 以下	ODE,DAE
MEBDFDAE	3 以下	ODE,DAE
RADAU	3 以下	ODE,DAE
RADAU5	3 以下	ODE,DAE
DASSL	1 以下	ODE,DAE,IDE
MEBDFI	3 以下	ODE,DAE,IDE
PSIDE	3 以下	ODE,DAE,IDE

ファイル間，サブルーチン間の関係が強い設計となっており，新たな問題に対応させたり，アルゴリズムの修正を行うことが難しく拡張性が低いことが問題点として挙げられる．

3.2　シミュレーションパッケージのアーキテクチャ

本論文で紹介するシミュレーションパッケージのアーキテクチャを図 1 に示す．インタフェース DaejDifferentialAlgebraicEquation の主要なメソッドを実装した抽象クラス AbstractDaejDifferentialAlgebraicEquation を継承することで問題クラスを生成し，インタフェース DaejDifferentialAlgebraicEquationSolver の主要なメソッドを実装した抽象クラス AbstractDaejDifferentialAlgebraicEquationSolver を継承することでソルバの駆動部分を記述したドライバクラスを作成する．そして，ドライバクラスは実際に演算を行うソルバクラスを呼び出し，さらにソルバクラスは演算に必要な様々なルーチンを含んだオグジリアリクラスを利用することで計算を行う．また，多倍長精度浮動小数点数を生成する MPFloat 型に対応した同様のインタフェースおよび抽象クラスを用意しており，それらを実装することで多倍長演算に対応している．

以上のような，Fortran にはない Java ならではのオブジェクト指向な構成となっており，共通部分をインタフェースとして抽出することでファイル間の役割を明確にし，保守性・拡張性の向上を図っている．

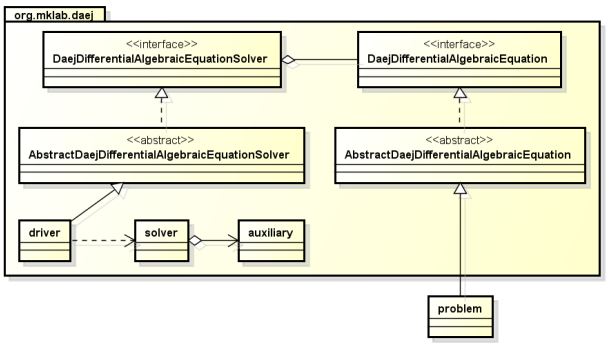

図 1: シミュレーションパッケージのクラス図

3.3　DAE ソルバ

DASSL[4] と RADAU5[5] に加え，BIMD[6] を利用することができる．

3.3.1　DASSL

DASSL とは，L.Petzold 氏によって提供された ODE，DAE，IDE 問題の指数 1 以下に対応したソルバである．数値解放として，微分代数方程式を積分する BDF(後退微分公式) 多段階法を利用している．

3.3.2　RADAU5

RADAU5 とは，E.Hairer 氏と G.Wanner 氏によって提供された ODE，DAE 問題の指数 3 以下に対応したソルバである．これは，5 次の陰的ルンゲクッタ法 (3 段ラダウ IIA 公式) を用いたソルバであり，強いスティッフ問題を数値的に高効率に計算できる．

3.3.3　BIMD

BIMD とは，C.Magherini 氏と L.Brugnano 氏によって提供される，ODE(常微分方程式) と指数 3 以下の

DAE に対応するソルバである．これは，Blended Implicit Methods と呼ばれるものに基づいて開発され，4 次，6 次，8 次，10 次，12 次の高次方程式で定義される L 安定な陰的解法の ODE ソルバ BiM[7] を DAE 問題に対応できるように拡張したものである [8]．

4 性能評価

本章では，DAE 問題を Java の BIMD ソルバを用いて解き，Fortran で解いた結果と比較する．使用する問題を表 2 に示す (以下，問題は指数で表現する)．また，実行環境と実行パラメータはそれぞれ表 3，4 に示す．

表 2: 使用する DAE 問題

問題名	次数	指数
Transistor amplifier	8	1
Water tube system	49	2
Andrews' squeezing mechanism	27	3

表 3: 実行環境

CPU	Intel Core i5 3570
OS	Windows7 professional
メモリ	16GB

表 4: 実行パラメータ

指数	1	2	3
相対許容誤差 $rtol$	10^{-7}	10^{-10}	10^{-10}
絶対許容誤差 $atol$	10^{-7}	10^{-10}	10^{-10}
初期刻み幅 $h0$	10^{-9}	10^{-10}	10^{-10}

4.1 計算誤差

表 5 は，Fortran の解を基準とした場合に Java の解でどれくらい誤差が発生しているかを示している．表 5 に示すように，誤差は小さく収まっているため実装はほぼ問題ないと言える．

表 5: Fortran に対する誤差

指数	最大絶対誤差	最大相対誤差
1	1.99×10^{-9}	1.23×10^{-9}
2	3.51×10^{-6}	5.38×10^{-6}
3	33.04	8.50×10^{-4}

4.2 計算時間

表 6 は，Fortran と Java の計算時間と Fortran に対して Java が計算に何倍時間がかかったかを示している．表 6 に示すように，Fortran に対して Java の方が遅い．原因としては，Fortran のデータ構造の仕様を満たすために配列操作を行っていることとガーベジ・コレクションの起動が考えられる．今後は，改良を行い 2 ～ 3 倍程度には抑えてたいと考えている．

表 6: 計算時間の比較

指数	言語	計算時間 [ms]	比率
1	Fortran	37.10	9.319 倍
	Java	345.7	
2	Fortran	790.7	4.637 倍
	Java	3666	
3	Fortran	35.50	6.758 倍
	Java	239.9	

4.3 多倍長演算

多倍長演算における性能評価として，表 2 に示す問題を倍精度と四倍精度にそれぞれ設定したソルバを用いて解き，得られた解の比較および考察を行う．ここで，多倍長で使用する実行パラメータを表 7 に示す．

表 7: 実行パラメータ

指数	1	2	3
相対許容誤差 $rtol$	10^{-7}	10^{-7}	10^{-10}
絶対許容誤差 $atol$	10^{-7}	10^{-7}	10^{-10}
初期刻み幅 $h0$	10^{-9}	10^{-7}	10^{-10}

表 8 は，四倍精度で出力された解を基準とした場合に倍精度で出力された解においてどの程度誤差が発生しているか示している．表 8 に示すように，誤差が非常に小さいことがわかる．したがって，今回用いた問題についてはどれも倍精度での解をある程度信頼できる．

表 8: 倍精度と四倍精度の比較

指数	最大絶対誤差	最大相対誤差
1	4.72×10^{-7}	1.57×10^{-7}
2	4.66×10^{-2}	1.12×10^{-3}
3	4.35	1.13×10^{-4}

5　おわりに

本論文では，Jamox への導入を視野に入れた，線形陰的な DAE で表されるシステムのための多倍長シミュレーションパッケージを開発した．今回は特に，新たに実装した BIMD ソルバについて焦点を当て，性能評価により BIMD ソルバが十分に機能することを確認できた．また，演算精度の変更により数値解の比較および考察を行うこともできた．

今後の課題として，計算速度の向上と新しいソルバの調査および実装が挙げられる．また，本来の目的である Jamox との連携を行っていきたいと考えている．

参考文献

[1] Jamox Project. http://jamox.mklab.org/

[2] Francesca Mazzia and Cecilia Magherini. Test Set for Initial Value Problem Solvers release2.4. Technical report, Department of Mathematics, University of Bari, April 2008.

[3] U.M. アッシャー, L.R. ペツォルド. 常微分方程式と微分代数方程式の数値解析, 培風館, 2006.

[4] L.Petzold. DASSL. http://www.netlib.org/ode/ddassl.f

[5] E. Hairer and G. Wanner. RADAU5. http://www.unige.ch/~hairer/prog/stiff/radau5.f

[6] C. Magherini and L. Brugnano. BIMD. http://web.math.unifi.it/users/brugnano/BiM/index.html

[7] C. Magherini, L. Brugnano, F. Mugnai. Blended Implementation of Block Implicit Methods for ODEs. Jour. Comput. Appl. Mathematics, Vol. 42, pp. 29–45, 2002.

[8] C. Magherini and L. Brugnano. Blended Implicit Methods for the Numerical Solution of DAE Problems. Appl. Numer. Math., Vol. 189, pp. 34–50, 2006.