有限要素法可視化対話型シミュレータにおける適応的シミュレーション進度調整機能

Adaptive Speed Counter of The Visual/Interactive Numerical Simulation System Using Finite Element Method

竹下明良[1)]，梅谷征雄[2)]
Akira TAKESHITA and Yukio UMETANI

1)静岡大学　情報学研究科情報学専攻　所属（〒432-8011 浜松市城北3-5-1,cs6055@cs.inf.shizuoka.ac.jp）
2) 工博　静岡大学教授　情報学部情報科学科（〒432-8011 浜松市城北3-5-1,umetani@cs.inf.shizuoka.ac.jp）

概要

物理シミュレーション環境においては,応答性と対話性に優れたシミュレータが必要である.そこで本研究では物理モデルに平板の板振動を取り上げ,可視化と対話的進度調整機能を持つ有限要素法シミュレータを作成した.シミュレータの特徴的な機能に実時間比例処理がある.メッシュ分割数やマシン性能とは独立にシミュレーション進度を指定できる機能だが,性能の異なるマシン上で指定された進度を保つためには工夫が必要となる.そこで,シミュレーション実行中に進度を計測し,その結果からフィードバックを掛ける方法を検討した.

1　はじめに

物理シミュレーションにおいてそのモデルの忠実度を上げるためにはパラメータの感度解析や実測値との対比が重要な役割を占める．このような操作を容易にするためには，応答性と対話性に優れたシミュレーション環境が必要と考える．[1]

本研究は物理モデルとして平板の板振動を取り上げ，そのモデルの空間形状，分割刻み数，パラメータ値などの変更を即座に計算して表示することにより，そのモデルの調整を効率よく行うための物理シミュレータを試作し，試作したシミュレータの応答性についての評価と改良を行う．

既存の差分法シミュレータに対し境界条件の自由度を増すために有限要素法による解法機能を付加した．有限要素法においては剛性マトリックスを使った線形計算を解く必要があるので計算負荷が大きい．したがって差分法シミュレータと比べ可視化と計算のバランスに注意を払う必要がある．本シミュレータの特徴的な機能の一つに実時間比例処理が挙げられる.ユーザが1ステップの実行の時間を指定できるようにしてシミュレーションの進度調整する機能である.シミュレーションの進度はメッシュ分割数やマシン性能により異なるので，シミュレーション実行中に進度を計測し,その結果から表示間引きと待ち時間を使ってユーザが指定したステップあたりの実行進度を保っていく.さらに実行中にユーザの指定時間と実際のステップ時間との誤差の修正を行っていく.試作のためのプログラミング言語としてVisual C++を用いた.

2　物理モデル

(1)　板の曲げ振動の物理モデル

2次元の板の曲げ振動の方程式は次のようになる[2].

$$D\left(\frac{\partial^4 w}{\partial x^4}+2\frac{\partial^4 w}{\partial x^2\partial y^2}+\frac{\partial^4 w}{\partial y^4}\right)+m\frac{\partial^2 w}{\partial t^2}-q_z(x,y,z)=0$$

D：板の曲げ剛性，w：板の変位 μ：板の単位面積あたりの質量

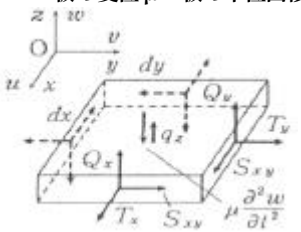


図 1　板の曲げ振動モデル

3　シミュレーション環境

(1)　シミュレータ画面

作成した有限要素法可視化対話シミュレータのシミュレーション画面を図2に示す.各ボタンでパラメータ設定とシミュレーションの進行を操作し,表示画面ではシミュレーション結果,パラメータ設定値,現在の実行ステップ数を表示する.

(2)　制御方法

表示速度ボタンを押すと図3のダイアログが現れる.「指定する」を選択すると図4のダイアログが現れ1ステップの実行時間を指定できる.「指定しない」を選択した場合はマシン速度で処理を行う.

この他に「速く」「遅く」のボタンで指定時間を1msec単位で実行中に動的に変更することができる.

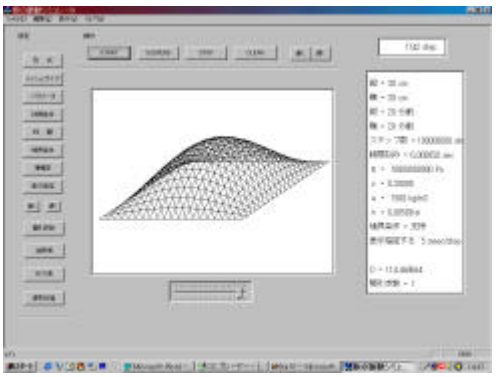


図 2　シミュレータ画面

図 3　表示方法ダイアログ

図 4　表示速度指定ダイアログ

（3）　制御アルゴリズム

実時間比例処理ではメッシュ分割数に影響せずに，ユーザが指定した1ステップ時間で処理を行う必要がある．つまり，実時間比例処理を行うには1ステップの時間を長くしたり短くしたりする必要がある．ユーザに1ステップの実行時間を指定させ，指定時間が実際の処理時間より長い場合は待ち時間を設け時間調整を行う．逆に指定時間が実際の処理時間より短い場合は，表示回数を間引くことで処理時間の短縮を行う．

各ステップ毎に計算,表示および1ステップの実行時間を計測しその値を使って表示の間引き係数と待ち時間を計算する.さらに一定の間隔で指定された1ステップ時間と実際の1ステップ時間との誤差を修正するために目標時間の変更を行って新しい表示間引き係数および待ち時間を決定する.制御処理のフローチャート図7を示す.

4　SLEEP関数

指定された時間だけループを繰り返すことで処理を中断し待ち時間を実現する関数（SLEEP）を作成した.

（1）　方法

あらかじめ一定時間ループを繰り返させる.一定時間のループ回数から指定された時間に対するループ回数を比例的に決定する.今回は1秒間のループ回数を計測し,そこからループ1回にかかる時間を計算する.その結果から指定時間におけるループ回数を決定する.

（2）　結果

指定待ち時間に対するループ回数とSLEEP関数を1000回繰り返した時の実行時間を表1に示す.

表 1　待ち時間関数の実行結果とループ回数

指定された時間(msec)	1000ステップ分の実行時間(sec)	ループ数(回)
0.5	0.50	17
1.0	0.99	35
1.5	1.49	53
2.0	2.03	71
2.5	2.58	89
3.0	3.08	107
3.5	3.62	125
4.0	4.12	143
4.5	4.67	161
5.0	5.16	179

5　間引き数と待ち時間の決定

実時間比例処理を行うには1ステップの時間を長くしたり短くしたりする必要がある．ユーザに1ステップの時間を指定させ，指定時間が実際の処理時間より長い場合は待ち時間を設け時間調整を行う．逆に指定時間が実際の処理時間より短い場合は，表示回数を間引くことで処理時間の短縮を行う．表示間引き係数および待ち時間は計算,表示,ユーザの1ステップ指定時間の3要素から決定する.

（1）　待ち時間の計算

ユーザの指定時間が実際の処理時間より長い場合は，1ステップ毎に計算と表示にかかった時間を計測する．そして，指定時間から表示と計算にかかった時間を引いたものを待ち時間とする.

待ち時間を利用した処理の流れを図5に示す.

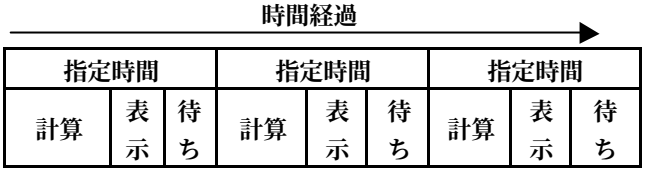


図 5　待ち時間を利用した処理の流れ

（2）　表示間引き係数の計算

ユーザの指定時間が実際の処理時間より短い場合は,計算,表示,1ステップ指定時間から間引き係数と待ち時間を決定する.間引き係数の計算方法は次のようになる.

- 指定された1ステップ時間を100倍し，100ステップの処理時間を計算する
- 100ステップの処理時間から100ステップの計算時間を引き，100ステップ中で表示に与えられる時間を計算する
- 表示に与えられた時間から指定時間内に可能な表示回数を計算する
- 可能な表示率から間引き係数を計算する

間引き係数とは何ステップに1回表示を行うかをあらわしたもの.図6に間引き係数3の処理の流れを示す.この場合3ステップに1回の割合で表示を行っていく.

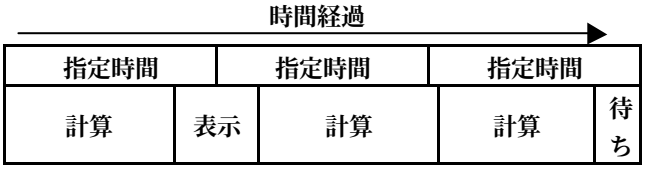
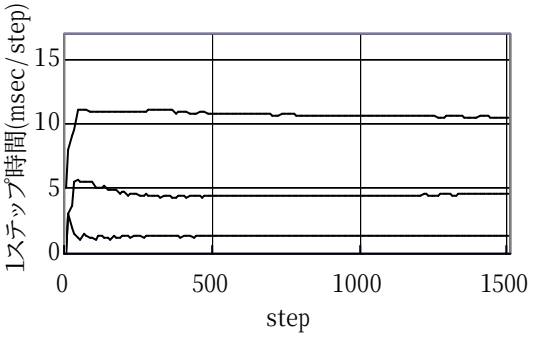


図 6　表示間引きをした処理の流れ

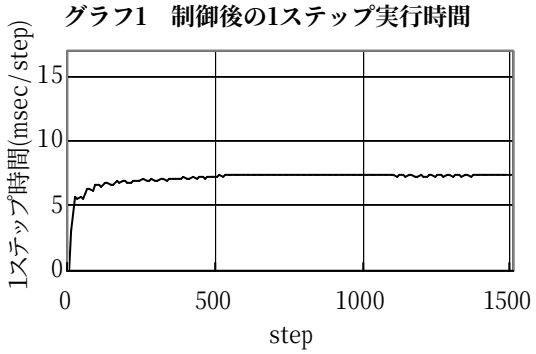
（3）　結果

上記の方法で決定した間引き係数と待ち時間で実行を行った場合の1ステップ実行時間をグラフ1に示す.メッシュ分割数は10×10で1ステップの指定時間は上から順に10msec,5msec,1msecである.また間引き係数と待ち時間の計算は10ステップに1回行い,その時利用する計算,表示時間はそれまでの累積平均を使用した.

時間制御を行わずマシン速度で実行を行った場合の1ステップ時間をグラフ2に示す.メッシュ分割数が10×10の場合を表す.



グラフ1　制御後の1ステップ実行時間



グラフ2　制御なし1ステップ実行時間

6　1ステップ時間の誤差の修正

上記の方法で決定した間引き係数と待ち時間を使用して1ステップ時間の制御を行った場合,グラフ1のように実際に実行される1ステップ時間は目標からずれが生じる.そこでこのずれを改善するために一定間隔で目標時間の修正を行っていく.

（1）　目標時間の修正

間引き係数と待ち時間の決定で使用する目標時間（1ステップ指定時間）を変更して時間修正を行う.

1）　方法

目標時間と実際の1ステップ時間の誤差は一定であると仮定し修正を行う.実際のステップ時間が長い場合は目標時間を誤差の分だけ短くし,逆に実際のステップ時間が短い場合は目標時間を誤差の分だけ長くする.そしてその新しい目標時間を使って新しい間引き係数と待ち時間を決定する.なお1ステップ時間と計算および表示にかかる時間は全ステップの累積平均を使用する.目標時間修正の式を式(1),待ち時間計算の式を式(2)に示す.(2)は間引き数3の場合，目標時間と間引き係数をかけて3ステップ分の実行時間を計算する.そして計算は3ステップ，表示は1ステップ行われるので，計算と間引き係数をかけたもに表示時間を加えたものが3ステップ中の計算と表示に必要な時間となる.これを先ほどの3ステップ分の実行時間から引き間引き係数で割ったものが1ステップ分の待ち時間となる.

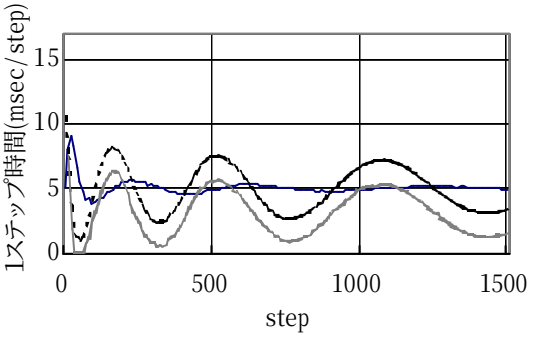
新しい目標時間=前の目標時間+
(ユーザの1ステップ指定時間-実際の1ステップ時間)　(1)

待ち時間=(新しい目標時間*間引き係数-
(計算*間引き係数+表示))　/　間引き係数　(2)

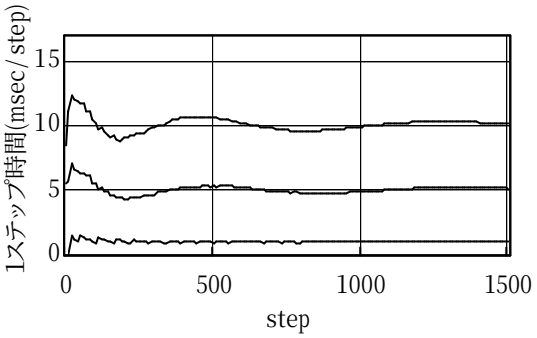
2）　結果

メッシュ分割数10×10で指定時間が5msecの時の待ち時間,目標時間,実際の1ステップ時間の推移をグラフ3に示す.（変化が少ないものが実際の1ステップ時間で，変化の大きい波形で値の大きいものが目標時間，小さいものが待ち時間を表している.）また各指定時間での1ステップ時間をグラフ4に示す.

メッシュ分割数は10×10で1ステップの指定時間は上から10msec，5msec，1 msecである.



グラフ3　目標時間,待ち時間,1ステップ時間



グラフ 4　目標時間修正による1ステップ実行時間

（2）　待ち時間の修正

待ち時間のみ変更して時間修正を行う.

1）　方法

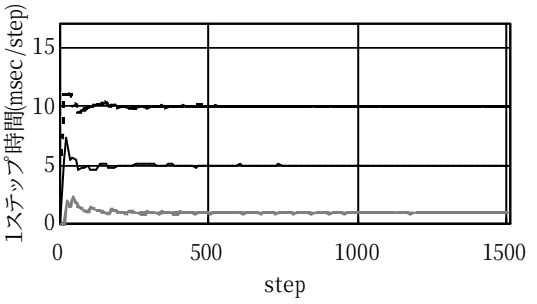
実際のステップ時間が長い場合は待ち時間を誤差の分だけ短くし,逆に実際のステップ時間が短い場合は待ち時間を誤差の分だけ長くする.ただし最初の待ち時間は式(2)を使い,間引き係数が変化した場合も同様の式を利用する.待ち時間修正の式(3)を以下に示す.

新しい待ち時間=前の待ち時間+

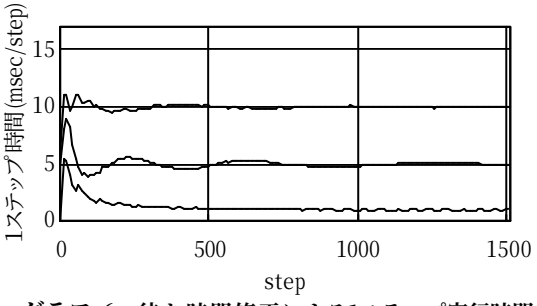
(ユーザの1ステップ指定時間-実際の1ステップ時間)　(3)

2）　結果

メッシュ分割数10×10指定時間が5msecの時の待ち時間,目標時間,実際の1ステップ時間をグラフ5に示す.(値の大きいものから目標時間，実際の1ステップ時間，待ち時間を表している.)また各指定時間での1ステップ時間をグラフ6に示す.メッシュ分割数は10×10で1ステップの指定時間は上から10msec，5msec，1 msecである



グラフ 5　目標時間,待ち時間,1ステップ時間



グラフ 6　待ち時間修正による1ステップ実行時間

7　おわりに

目標時間の修正による方法は制御毎に新しい目標時間と計算と表示時間を使って間引き係数と待ち時間の決定を行う.計算と表示時間はそれまでの累積平均を使用しているが,ステップ毎のばらつきが大きい.そのためこの2つの要素を修正に利用すると目標時間と待ち時間が安定せず1ステップ時間も振動してしまう.改善策で減衰係数をかけて修正幅を小さくした場合安定性は増したが指定時間への収束が遅くなってしまった.

これに対し待ち時間の修正による方法は不安定な計算と表示の要素を利用しないので安定した結果が得られた.しかし時間計測に使用したCLOCK関数の精度が悪く計測結果が安定するまで時間かかる.そのため,開始後まもないステップ数では1ステップ時間が安定せず急激な目標への収束はできなかった.

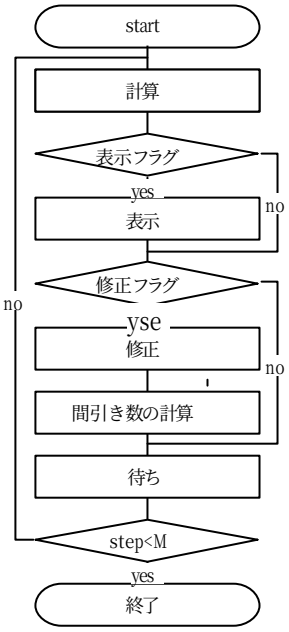


図 7　時間制御フローチャート

謝辞：

本研究を行うに当たり，日頃親切に御協力頂いた梅谷研究室の皆様に厚く御礼を申し上げます

参考文献

1)梅谷征雄：PSE可視化インターフェース，第一回問題解決ワークショップ論文集，pp25-30，1998

2)小林繁夫：振動学，丸善株式会社，1994

3)柳沢猛,野田直剛,他共著:パソコン対応基礎材料力学,日新出版株式会社，1995

4)菊地文雄：有限要素法概説，株式会社　サイエンス社,1999

5)林晴比古：新Visual　C++　6.0入門　ビギナー編　SOFTBANK株式会社，1999

6)林晴比古：新Visual　C++　6.0入門　シニア編　SOFTBANK株式会社，1999