

数値シミュレーション過程の可視化トレース

Visual Trace for Numerical Simulation Process

後藤知一¹⁾, 梅谷征雄²⁾

Tomokazu GOTO and Yukio UMETANI

1) 静岡大学 情報学研究科情報学専攻 (〒432-8011 浜松市城北3-5-1, cs8028@cs.inf.shizuoka.ac.jp)

2) 工博 静岡大学教授 情報学部情報化学科 (〒432-8011 浜松市城北3-5-1, umetani@cs.inf.shizuoka.ac.jp)

We have developed the function of “Visual Trace” which enables us to visually grasp the process and the result of the numerical calculation using finite element method for the piano soundboard simulator. First, we visualized the variable array obtained as a result of calculation and the matrix values which have big influences on calculation results. Next, we visualized the calculation process. Finally, we have visualized the difference of two calculation results for two different parameter values.

Key Words : visual trace, finite element method, process of calculation

1. はじめに

シミュレータを使用するときそれぞれのパラメータ値の移り変わりが計算結果に与える影響、さらにはその伝播の過程がわかると非常に便利である。

そこで本研究ではピアノの響板振動をシミュレートする有限要素法のシミュレータ(図1)を対象として、その計算過程や計算結果に影響を与える重要な値の移り変わり、さらには異なる2つのパラメータにおける計算結果の違いとその原因を視覚的に把握できるような“Visual Trace”の機能を開発する。

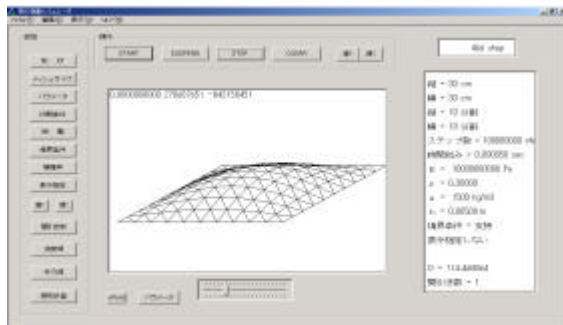


図1 有限要素法シミュレータの画面

Fig.1 GUI of Finite Element Simulator

2. 有限要素法シミュレータのプログラム

(1) 有限要素法

今回対象とした有限要素法シミュレータはピアノの響板の振動をシミュレートしている。つまり、この有限要素法シミュレータは響板の振動の方程式を解いている。響板振動の方程式を以下に示す。

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

上の式を

$$u = \frac{\partial^2 w}{\partial x^2}, v = \frac{\partial^2 w}{\partial y^2}$$

なる変数を導入し、有限要素法を用いて解いている。

(2) 有限要素法とプログラムの対比

有限要素法とプログラム動作を対応させると図2のようになる。なお、図2において u^t と f_u 、 v^t と f_v がそれぞれ対応している。また、 f_w については左側のフローチャートの四角で囲まれた部分と対応している。また、式1はSOR法による連立一次方程式の反復解法の基本演算で ω は加速係数を表している。今回は $\omega = 1$ としている。

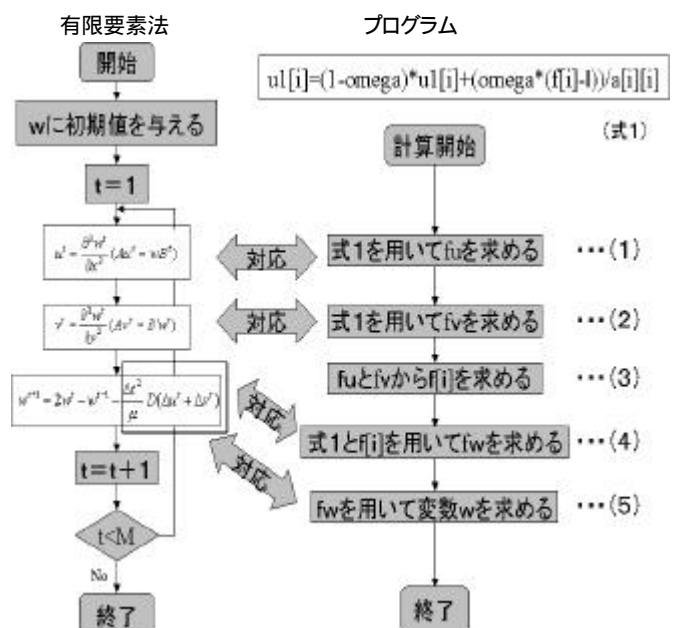


図2 有限要素法の流れとプログラムとの対応
Fig.2 Computation Flow And Program Chart

図2からわかるように u^t と v^t を求めそれらを用いて w^{t+1} を求めている。プログラム上ではそれら3つの変数は fu 、 fv 、 fw に対応している。また、式1において $u_1[i]$ は収束して値が決まるのだが、そのとき変数 l の値が非常に重要となってくる。その変数 l （エル）の値は以下の式で決まる。

$$l = \sum_{j=1}^m a_{ij} u_j$$

3. Visual Traceの機能

この研究で作成したVisual Traceは以下4つの機能を果たす。

- 計算結果である変数 w の可視化
- w の計算に必要な配列 am の値の可視化
- 計算過程と値の移り変わりの可視化
- 2つの異なるパラメータによる計算結果の違いの可視化

また、表示の切り替えのダイアログボックスがあり、以上4つの表示を切り替えることができる。

4. 実装方法

Visual Traceの機能の～の順序で作成していく。この有限要素シミュレータはプログラム上、Documentクラスで計算し、Viewクラスで表示している。（図3）そのため、必要な値をDocumentクラスで格納しておきViewクラスに渡し表示する。

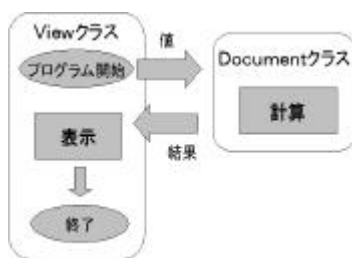


図3 プログラムの流れ
Fig.3 Process Flow

また、においては異なる2つのパラメータで同時に2回計算を行う必要がある。そこでDocumentクラスをもうひとつ追加し双方の結果をViewクラスで表示する。

5. 変数 w の可視化

(1) 実装結果

変数 w の値の移り変わりを可視化した画面を図4に示す。

(2) 仕様

- 丸はそれぞれの接点を示している。
- 色の指定はその時間ステップ内での最大値を赤、最小値を青として表示している。その中間の色は以下の式を用いて比率を導き出し色に反映している。

$$\text{比率} = (\text{値} - \text{最小値}) / (\text{最大値} - \text{最小値}) \dots(\text{式2})$$

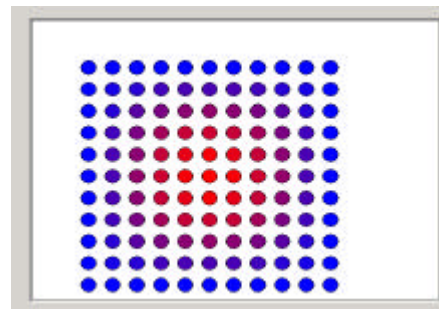


図4 変数 w の表示
Fig.4 Visualization of Value w

(3) 考察

パラメータ値を基本振動の初期設定とした響板の振動を示した画面と比較してみると、板が上に振動しているとき中心部分が赤で表示され境界線が青く表示される。逆に下に振動しているときは中心部分が青く表示され境界線が赤く表示される。また、色の変化は5000step毎にしか見られなかった。これは、振動の上下の移り変わりに対応していて、時間ステップ内での最大値と最小値を基準としたためだと考えられる。

また、パラメータ値をいろいろと変えてみると激しい色の変化が確認でき、変数 w の変化を視覚的に把握できたと考えられる。

6. 配列 am の可視化

(1) 実装結果

配列 am の要素は接点と周りの接点の関連を示すものなので接点間を結ぶ線で表示する。接点自身も表示する必要があるため、変数 w の可視化画面と一緒に表示する。また、配列 am は一つの接点から全ての接点へとつながっているが、そのほとんどが0の値をとっているためそれは表示しないこととする。その実装結果を図5に示す。

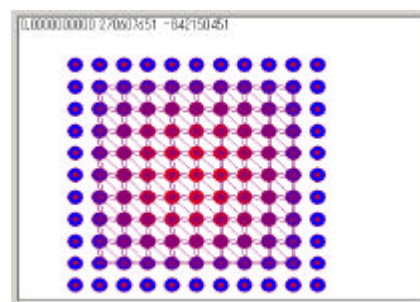


図5 配列 am の表示
Fig.5 Visualization of Matrix am

(2) 仕様

- 色の指定はその時間ステップ内での最大値を赤、最小値を青として表示。また、その中間の値は式2を用いて比率を出し色に反映する。
- 自分自身へ向かう配列 am の値があるので接点を表す丸の中の小さな丸で表示している。
- 自分から相手への am と相手から自分への am が存在するため、その2本の線が一緒になってしまわないようにずらして表示する。丸の中心を出発点とし相手へとつながっている。

(3) 考察

時間ステップが進んでも色の変化は見られなかった。これは、配列 am の値はメッシュサイズによって決まるので計算が進んでも変化しないためだと考えられる。

多くの画像処理をしているため動作が不安定で表示が遅くなってしまったためプログラムの改善が必要である。

7. 計算過程の可視化

(1) 実装結果とその仕様

計算過程の表示ではまず、図4の変数 w の表示画面が表示される。ここで計算過程を見たい接点のうえでダブルクリックするとその計算過程が表示される。計算過程は図2における(1)~(5)の5段階に分け、式1に沿った表示をした。

さらに、式1の $u_1[i]$ は収束して値が決まっているので式1を使用する計算過程においては収束過程も表示できるようにした。また、収束の段階で変数 l (エル) の値も変わっていくのでその計算過程も表示した。

値の移り変わりをアニメーションでわかりやすくし、その色の指定は状況に応じて1番良い方法を試した。その実装結果を図6、7に示す。

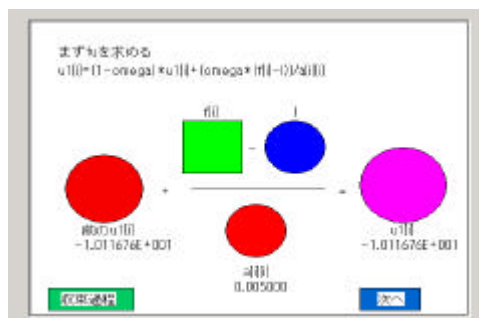


図6 5段階の計算過程の表示1

Fig.6 Visualization of Calculation Processes(Step1)

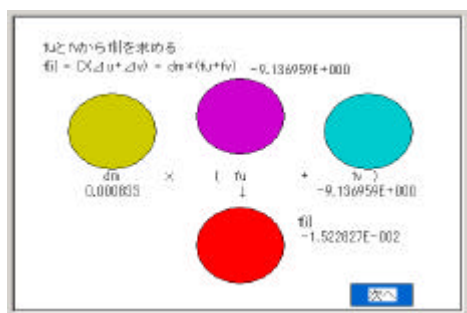


図7 5段階の計算過程の表示2

Fig.7 Visualization of Calculation Processes(Step2)

図6は図2における式1, 2, 4の場合、つまり式1を用いるときの計算過程の表示である。図7はそれ以外の場合の計算過程の表示である。画面上には以下のものが表示される。

- ・現在どのような計算をしているかの表示
- ・変数の名前と実際の値

画面右下の次へボタンをダブルクリックすると計算が進み画面が変わる。5段階の表示が終了したら図5の変数 w の表示画面に戻る。

また図6において画面の左下の収束過程ボタンをダブルクリックすると式1における $u_1[i]$ の収束過程が表示される。その画面を図8に示す。

図8は値が順番に決定し収束していく様子を緑の大きな丸が左上から動いていく

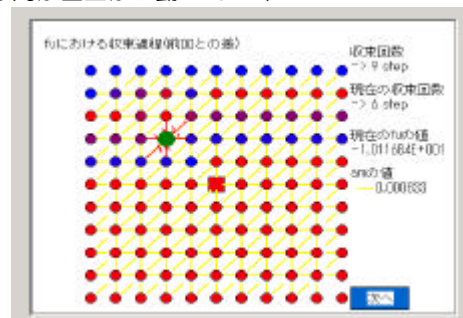


図8 収束過程の表示

Fig.8 Visualization of Convergent Steps

ことにより表示する。赤い矢印は直接関係している配列 am を表す。色の指定は収束の結果、前回の値との差の大きいものを赤、小さいものを青とする。つまり、収束が終了すると画面上の丸は全て青色となる。四角い接点は最初に指定した接点を表す。

また、画面上にはさらに以下のことを表示する。

- fu , fv , fw のどの値を計算しているのかの表示。
- 全体で何回収束するのか。また、現在の収束回数。
- 現在の $u_1[i]$ の値と配列 am の値。

画面右下の次へボタンをダブルクリックすることにより収束を進め画面を切り替える。収束が終了すると図7の画面に戻る。

また、図8の収束過程の表示画面でそれぞれの接点をダブルクリックすることでその接点の周りとの関係を細かく表示できるようにした。周りとの関係を表示している画面を図9に示す。

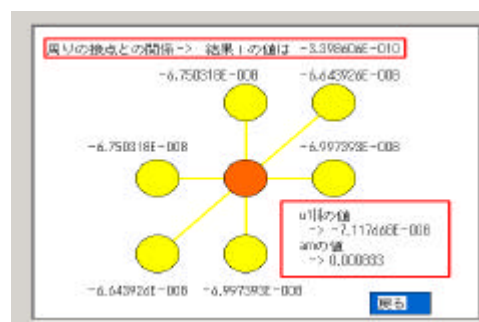


図9 周りの接点との関係表示

Fig.9 Zoomed View of a Convergent Step

真中の丸が指定した接点を表し図8と対応している。周りの丸が中心に向かって動いていくことにより値が決まる様子を示す。また、周りの接点と配列 am から変数 l が決まるため変数 l の値も表示している。さらに以下のことも表示する。

- $u_1[i]$ の値と周りの接点の値
- 配列 am の値

画面右下の戻るボタンをダブルクリックすることにより図8の収束過程表示画面に戻る。

(2) 考察

5段階の計算過程の表示

計算過程を5段階にわけ表示する、さらに実際の値を表示することにより計算の流れや値の移り変わりがわかりやすくなった。また、接点ごとに計算過程を見ることができるのでそれぞれの接点の差も把握できる。

色の表示も値の大小で一応変化するようにしたのだが一つの画面上では違った変数同士を表示しているのそれほど効果的ではなかったと感じた。そのため、違った接点の差の表示や時間ステップに沿った差の表示などが必要であると考えられる。

収束過程の表示

値の決まる様子をアニメーションで表示し、色の指定に前回との差を反映させることにより収束の様子が非常によく把握できた。また、全体で何回収束するのか、現在何回目の収束なのかを表示することで収束の流れをつかむことができた。

実際に収束過程を表示してみた結果、時間ステップが進むと収束回数が減ることがわかった。また、収束の画面がだんだんと青くなっていくわけではなく、赤くなったり青くなったりすることから一定の割合で収束していくわけではないことがわかった。

周りの接点との関係表示

収束の段階で回りの接点との関係を細かく表示することによりどのように値が決定するのかがよくわかった。

接点を選択することができるので収束の仕方の違いがどの接点の値に影響しているかなど細かいことまで把握できた。

8. 2つの異なるパラメータによる計算結果の違いの可視化

(1) 実装結果

異なる2つのパラメータ値による計算結果を重ねて表示することによりその差を確認できるようにした。その差の表示画面を図10に示す。

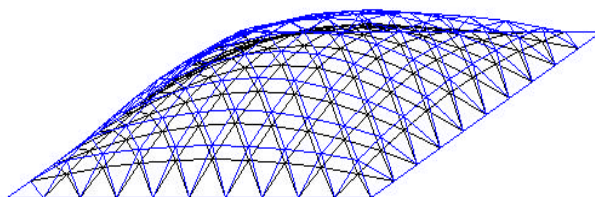


図 10 差の表示

Fig.10 Visualization of Value Differences

(2) 考察

2つを重ねて表示することによりその差を把握しやすくなった。次はその差がどの計算過程において生じるのかを発見できるようにする必要がある。そのため、時間ステップに即した計算過程の可視化が必要になってくると考えられる。

9. まとめ

全体的に計算過程とパラメータ値の移り変わりを可視化することにより計算の流れを視覚的につかむことが容易になった。特に、2つの異なるパラメータ値による計算結果の表示においては、その差を確認するのが非常にわかりやすくなった。また、この Visual Trace は状況に応じて見たい画面を出すことができるので非常に使いやすいと思われる。

しかし、この Visual Trace を作成するにあたって様々な問題点や改善すべき点があることに気づいた。それを以下に述べる。

- ・配列 am の表示のときなど、せっかく色の指定をしても色の変化が見られなかったり、その色の指定があまり効果的ではない場合があった。そのため、より良い表示方法の発見とその実装が必要であると考えられる。
- ・状況に応じて表示方法や色の指定を変えているので複雑になってしまった。そのため表示方法の程度の統一化、また表示方法の整理が必要である。
- ・多くの画像処理を行っているため動作が不安定なところがあった。そのため、プログラムの安定化と高速化が必要である。
- ・今回は時間ステップにはあまり即していなかったため時間ステップに即した表示が必要である。

今回使用したのは有限要素法シミュレータだが、差分法などの異なる計算方法での計算過程の可視化にも同様な手法が適用できる。

謝辞：

本研究を行うに当たり、日頃親切にご協力頂いた梅谷研究室の皆様へ厚く御礼を申し上げます

参考文献

- 1) 菊地文雄：有限要素法概説、株式会社 サイエンス社、1999
- 2) 竹下明良、塩沢徳人、梅谷征雄：「有限要素法可視化対話シミュレータの試作」、日本計算工学会講演会論文集、5巻2号、pp、543-546、2000
- 3) 山本信雄：Visual C++、株式会社 翔泳社、1999