

カスプ面解析用 Web アプリケーションの構築

久米靖文*, 森田裕三**

Construction of Web Application for Cusp Surface Analysis

Yasufumi KUME and Yuzou MORITA

This paper describes about construction of Web application for cusp surface analysis. Client accesses the Web server to analyze the data for cusp surface analysis, and a user's browser downloads Web page for data input automatically. Client can be offered through Web browsers by using HTML generated dynamically by Java Server Pages (JSP) technology, or Java applets. Tomcat is operated as a plug-in of Apache to start the Java servlet on the server. Tomcat is the servlet container that is used to implement cusp for Java. Cusp for Java is applied as Java servlet. It is a java class that extends a J2EE-compatible Web server. Cusp for Java receives HTTP (Hypertext Transfer Protocol) request from the browser, and provides views of result data as HTML. In put of data must be to in put by handling operation. In this system, the result of the case of stand-alone can be similar to this system. Apache sever receive requirement from client and this requirement transfer to Tomcat perform to restart cusp for JAVA as servlet. The result of analysis is responded to client. Therefore, Cusp for JAVA is able to opened by Web.

Key Words: Web application, Web server, Java Server Pages (JSP), Java servlet, Tomcat, Cusp for JAVA.

1. まえがき

従来技術計算は複雑で計算能力と時間を必要とするため、通常スタンドアローンのコンピュータで計算を行う。このシステムでは開発された計算プログラムの使いやすさ、すなわちユーザビリティを調査することが困難である。まずグループウェアであるロータスノートにこの計算プログラムを搭載したシステムを構築することを検討したが不可能であった。インターネットは種々の情報システムで利用されており、とくに不特定多数のコミュニケーションには非常にすぐれた機能をもっている。開発された解析プログラムを World Wide Web 上に Web サーバを用いて公開するシステムを検討している。そのためにはシステムのセキュリティを考慮し、安全に公開することが重要である。上級のシステムエンジニアの参加を要請して 5 回の研究会を開催し、Linux を Web サーバの OS とし、サーバソフトの Apache と Java サブレットコンテナの Tomcat を連携させることで、解析プログラムを Java サブレットとして用いる。Java サブレットはクライアントからの入力に対し Web サーバ上の Tomcat 等のサブレットコンテナ内で Java プログラムが実行され、処理を実行し、その

結果を出力するサーバサイドプログラムのことである。サブレットを追加することにより、Web サーバの機能を拡張することができる。このサブレットを用いて、クライアントからの入力をサーバで処理し、結果をクライアントに返信するサーバクライアント方式の技術計算を行うシステムを開発した。また Linux, Apache, Tomcat, Java を連携した技術計算を行うシステムはほとんど見当たらない。創造過程のようなヒューマン・ファクターを含む場合の不連続現象を解析する手段として、Cobb 等によって、統計的カストロフィモデルを用いた 2 次元カスプ面解析が提案されている¹⁾。久米等は 3 次元表示カスプ面解析システムを提案している²⁾。このカスプ面解析は Java 言語でコーディングされ、jar ファイル化された Cusp for JAVA という Java アプリケーションを用いている。この Cusp for JAVA をより多くの研究者が利用できる環境を作るため、Cusp for JAVA を Web サーバ上で安全かつ円滑に動作させることを目的とする。

2. Web サーバの環境

この章では、構築したサーバの環境や、Web アプリケーションが動作する実行環境などについて述べている。

2.1 Linux³⁾

1991 年にフィンランドのヘルシンキ大学の大学院生(当時)Linus Torvalds が卒業研究の題材として開発したオペレーティングシステムを基礎として開発された UNIX

平成 18 年 6 月 24 日

* 機械工学科

** 総合理工学研究科 メカニクス系工学専攻

日本経営工学会関西支部修士論文発表会にて発表 (2005.3)

互換の OS である。その後、オープンソースのフリーソフトウェアとして公開され、全世界のボランティアの開発者によって改良が重ねられた。Linux は既存の OS のコードを流用せず、全く新たに書き起こされた。

このフリーソフトウェアとは無料のソフトの意味ではなく、自由であるという意味であり、GPL というライセンス体系に基づいて、誰でも自由に改変・再配布することができる。Linux は他の OS に比べ、低い性能のコンピュータでも軽快に動作する。また、ネットワーク機能やセキュリティに優れ、また非常に安定しているという特長を持つ。Linux は学術機関を中心に広く普及しており、企業だけでなく個人のインターネットサーバとしても多く採用されている。Linux は Sackware 系、Red Hat 系、Debian 系の 3 つ系統があり、本研究では Red Hat Linux 9 を使用している。

2.2 Web サーバ

WWW システムは、情報送信を行うコンピュータと WWW による情報送信機能を持ったソフトウェアからなる。Web サーバは、HTML 文書や画像などの情報を蓄積しておき、Web ブラウザなどのクライアントソフトウェアの要求に応じて、インターネットなどのネットワークを通じて、これらの情報を送信する役割を果たす。要求に応じてプログラムを実行し、結果をクライアントに送信する動的ページ生成の機能などを持つ。Java 言語を利用した Java サーブレットや JSP、Microsoft 社独自の技術である ASP などを利用した Web サイトも増加している。

2.3 Apache⁴⁾

NCSA httpd 1.3 をベースに、1995 年に開発が始まった Web サーバである。Apache はフリーソフトウェアとして無償で公開され、世界中のボランティアプログラマたちの手によって開発された。誰でも修正・再配布することができる。Apache はもともと、NCSA(米国イリノイ大学スーパーコンピュータセンター)が開発した NCSA httpd の細かいバグに修正を加え、新しい機能を追加するためのパッチ(patch)の寄せ集めとして開発されていたため、この名称がついたとされている。現在 Apache は単独で動作する Web サーバとなっており、世界で最も使用されている Web サーバとなっている。

2.4 Tomcat^{3,5)}

Tomcat とは The Apache Software Foundation が運営している Jakarta プロジェクトの生み出した Java サーブレットコンテナ兼 Web サーバの名前である。この Tomcat には Java サーブレットと Java Server Pages(JSP)という 2 つのテクノロジーを実装しており、静的な HTML ページにのみ用いられるだけでなく、ユーザーからのリクエストに

応じてプログラムを実行し、その実行結果をブラウザに返すという作業に適している。Tomcat は単独の Web サーバとしても使用することができるが、他の Web サーバに対するアドオンのサーブレットコンテナとしても使用することができ、どちらの使用形態でも状況に応じて使い分けることができる。Apache Software License というライセンスに基づいて公開されており、誰でも自由かつ無償で利用・改変・再配布できる。もともと Tomcat は Sun Microsystems 社の Java サーブレットコンテナである Java Web Server(JWS)から始まった。その後、Apache の Jserv や CERN/W3C の Jigsaw などがオープンソースの Java サーブレットコンテナとして発表され、徐々に浸透した後に、Sun Microsystems 社から JSP をサポートした Java Server Development Kit(JSDK)が 1997 年にリリースされた。翌年、Sun Microsystems 社から JSDK リリース 2.1 が発表された後、Sun Microsystems 社の James Duncan Davidson によって JSDK サーバのコアが Tomcat というまったく新しい Java サーブレットエンジンに書き直され、JSDK2.1 の後継として Tomcat3.0 がリリースされた。その後 Sun Microsystems 社と開発者である James の意向で Tomcat はオープンソースプロジェクトとして The Apache Software Foundation に寄贈され、現在 Tomcat は Jakarta プロジェクトとして The Apache Software Foundation によって運営されている。

2.5 Apache と Tomcat の連携^{3,4)}

Apache ではサーブレットの処理を行うことができず、Tomcat を Web サーバとしても使用できるが、レスポンス性、セキュリティ性に欠けるため、Tomcat を Apache のプラグインとして動作させるのが一般的である。その際、サーバ上に Apache、Java2 SDK、Tomcat 連携モジュールの 4 つをインストールしなければならない。Apache の公式サイト (<http://www.apache.org>) から最新版の Apache2.0 (<http://2.0.52.tar.gz>) をダウンロードし、インストールを行う。次に Sun Microsystems 社 (<http://java.sun.com>) より Java (TM) 2 SDK, SE (j2sdk-1_4_2_06-linux-i586-rpm.bin) をダウンロードし、Java をインストールする。最後に Apache の公式サイトにある Jakarta Project (<http://jakarta.apache.org>) より Tomcat4.1.31 (jakarta-tomcat-4.1.31.tar.gz) をダウンロードしインストールを行う。3 つのインストール終われば Apache と Tomcat を連携に必要なモジュール (mod_jk2.so) をインストールすることで Apache と Tomcat 間の連携が可能となる。

2.6 Java サーブレット

Java サーブレットとはクライアントからの入力に対し Web サーバ上の Tomcat 等のサーブレットコンテナ内で

Java プログラムを実行し、処理した結果を出力する Java プログラムのことである。サーブレットを追加することにより、Web サーバの機能を拡張することができる。サーブレットは Java 言語で記述されているため、特定の OS やハードウェアに依存することがなく、サーブレット API を実装したあらゆる Web サーバで稼働させることができる。CGI などの他のサーバサイドプログラムと異なり、一度呼び出されるとそのままメモリに常駐するため、高速な処理が可能である。また、データを永続的に扱うことができるため、複数のユーザー間で情報を共有することもできる。当初は Sun Microsystems 社の Java Web Server の機能の一つであった、仕様が Java servlet API として独立し、現在では様々な Web サーバ向けにサーブレットの実行環境が用意されている。

2.7 TomcatにCusp for JAVAを実装

Java サーブレットを Tomcat に実装するには、デフォルトでインストールした場合、Tomcat は /usr/local/tomcat に格納される。その Tomcat フォルダの下層にある webapps フォルダの中に、そのサーブレットを配置することで Tomcat から、そのサーブレットを起動することができる。

3. Web サーバの構築

この章では、サーバの構築方法について述べている。本研究で用いた Linux のインストールにはじまり、サーバのインストール、Java サーブレットコンテナの実装などを詳細に示した。

3.1 Linux のインストール

以下に述べているのは Linux のインストールについてである。

3.1.1 Linux の入手方法

Linux は雑誌や書籍などの付録として添付されていることもあるが、それらの書籍が入手困難である場合は直接サイトから FTP でインストールファイルをダウンロードしてくる方法もある。ダウンロードしたファイルは、CD-ROM 等へ書き込むことによって、インストール CD-ROM として使用することができる。

3.1.2 Linux のインストール方法^{3,4,6)}

インストール方法は以下の方法がある。

3.1.2.1 CD-ROM

CD-ROM ドライブと Red Hat Linux CD-ROM がある場合、この方法を使用できる。ブートディスク、またはブート可能な CD-ROM が必要である。

3.1.2.2 ハードドライブ

Red Hat Linux の ISO イメージをローカルハードドライブにコピーしている場合は、この方法が使える。ブートディスクが必要である。

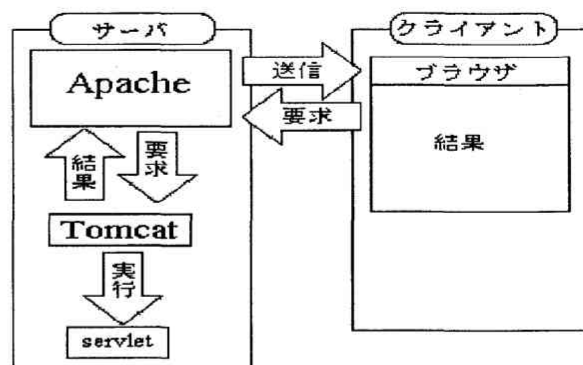


Fig.1 Connection of Apache and Tomcat.

3.1.2.3 NFS イメージ

Red Hat Linux の ISO イメージまたはミラーイメージを使用して NFS サーバからインストールする場合は、この方法を使用することができる。ネットワークドライブディスクが必要である。

3.1.2.4 FTP

FTP サーバから直接インストールする場合は、この方法を使用する。ネットワークドライブディスクが必要である。

3.1.2.5 HTTP

HTTP (Web)サーバから直接インストールする場合は、この方法を使用する。ネットワークドライブディスクが必要である。

4. Web アプリケーションの動作確認

この章ではクライアントからのサーブレットの起動を行い、サーバ上で動作させたサーブレットの Cusp for JAVA の動作確認と、あらかじめ別クライアントにインストールされているスタンドアローンの Cusp for JAVA とを同じデータを用いて動作の比較を行い、その考察について述べている。

4.1 Cusp for JAVA の起動

クライアントからサーバに指定のページにアクセスすると Fig.2 のような Cusp for JAVA のトップページがブラウザ上にあらわれる。Java サーブレットを起動すると、Fig.3 のような入力画面が別のウィンドウにあらわれ、その入力フォームに数値を入力し Start ボタンを押すと解析が開始され、結果が表示される。この動作確認では以前用いたことのある Table 1 のサンプルデータを入力した。

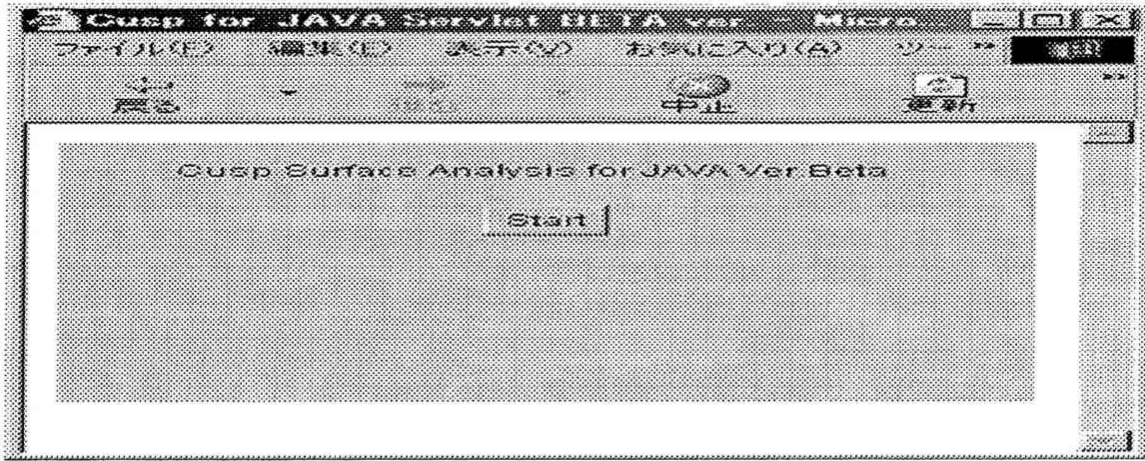


Fig. 2 Top page of Cusp for JAVA.

Table.1 Experimental data.

variable1	variable2	variable3
56.8	56.8	1.81
65.8	69.5	-0.18
71.6	64.2	0.00
61.9	61.1	-2.64
76.8	56.8	-3.23
64.5	53.7	1.29
65.8	69.5	0.53
65.2	60.0	-1.00
58.7	53.7	-3.10
72.3	62.1	1.78
48.4	51.6	4.07
65.2	58.9	0.49
60.0	61.1	-0.10
56.1	64.2	-3.62
61.3	35.8	-0.15
69.0	48.4	-0.97
78.1	61.1	-3.20
78.1	62.1	-0.84
69.0	60.0	-2.11
65.8	60.0	0.43
65.8	60.0	-3.26
69.0	61.1	1.28
65.8	61.1	2.07
71.6	60.0	-0.02
69.0	58.9	-0.60
71.6	58.9	0.17
69.0	61.1	3.36
65.8	60.0	0.42
69.0	61.1	3.50
69.0	60.0	3.81

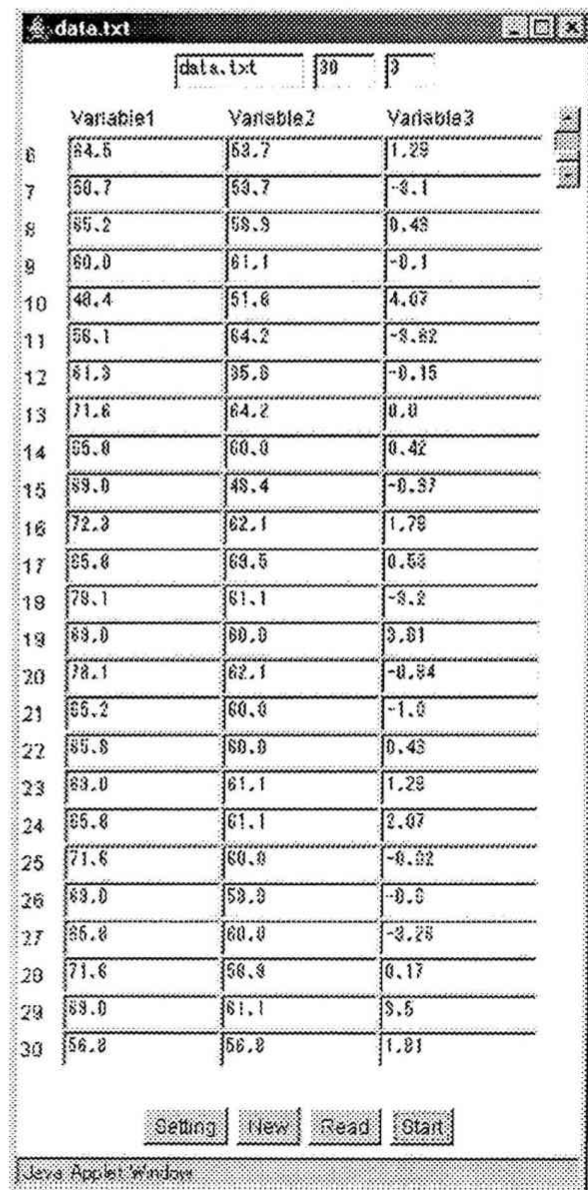


Fig. 3 Input form of Cusp for JAVA.

Fig. 3の入力フォームではスタンドアロン版では、ウインドウ情報に data.txt と入力されているボックスに解

析するデータのファイル名を入力し、ウィンドウ下方にある Read ボタンを押せば、そのファイルを読み込み数値行列データがウィンドウ上に読み込まれる。今回の動作確認では、そのファイル読み込み機能が使えなかったのでハン

ドリングによって直接入力を行った。ハンドリング作業は、まずウィンドウ上方のボックスに、対象データの行列の数を入力し、New ボタンを押し、入力フィールドを決定する。次に各 Variable の欄に対象の数値データを入力する。

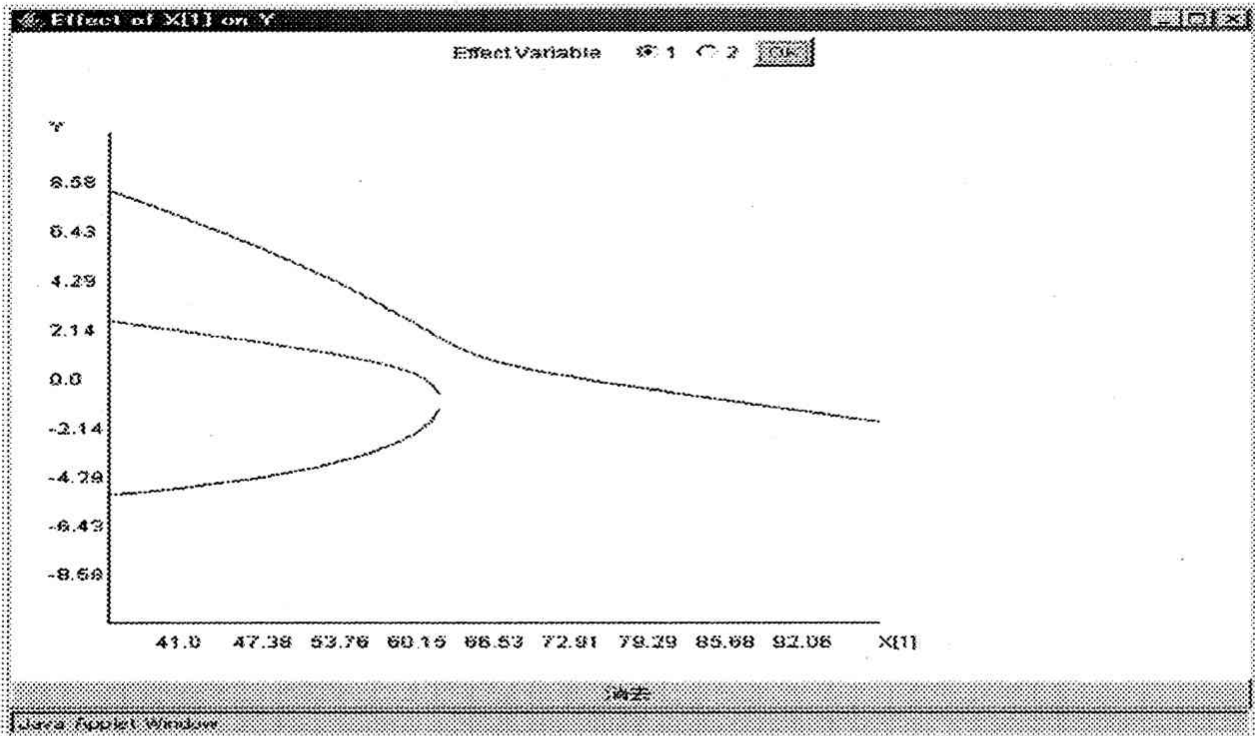


Fig. 4 Relation of Y-X[1] at X[2]=59.093.

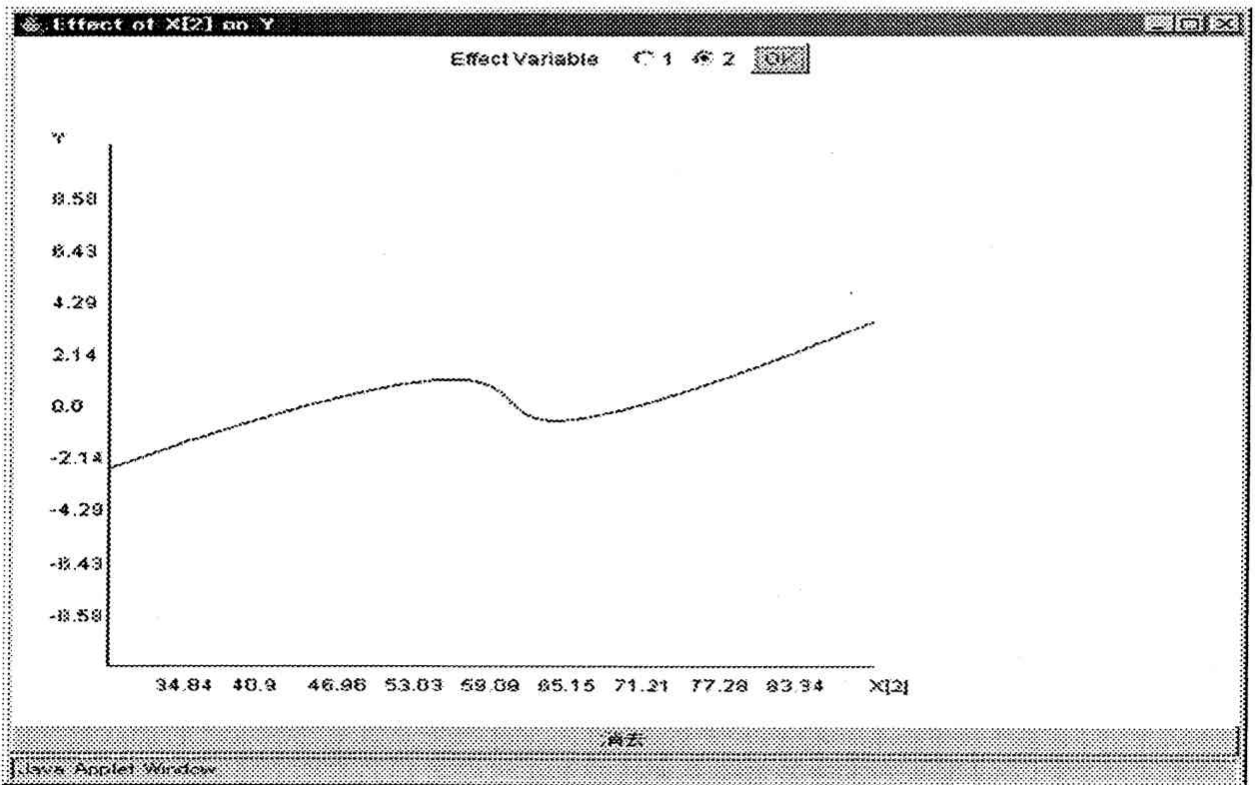


Fig. 5 Relation of Y-X[2] at X[1]=66.533.

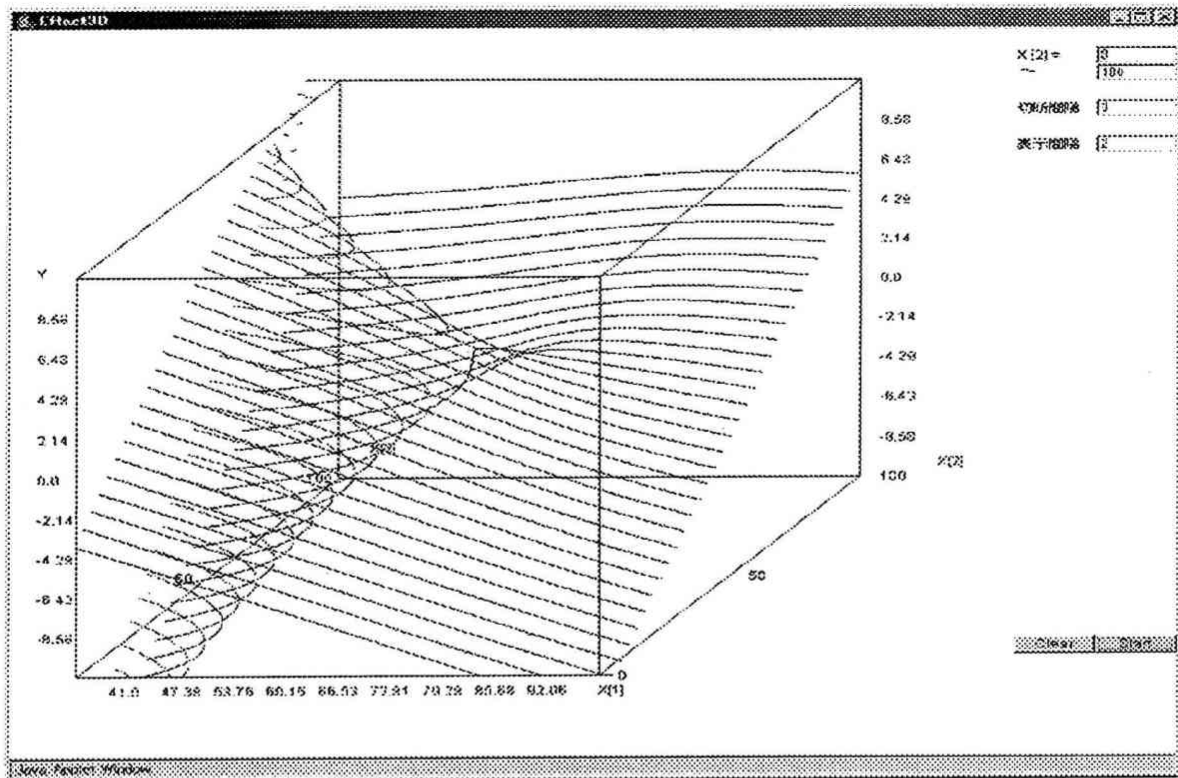


Fig. 6 Effect of X[1], X[2] to Y.

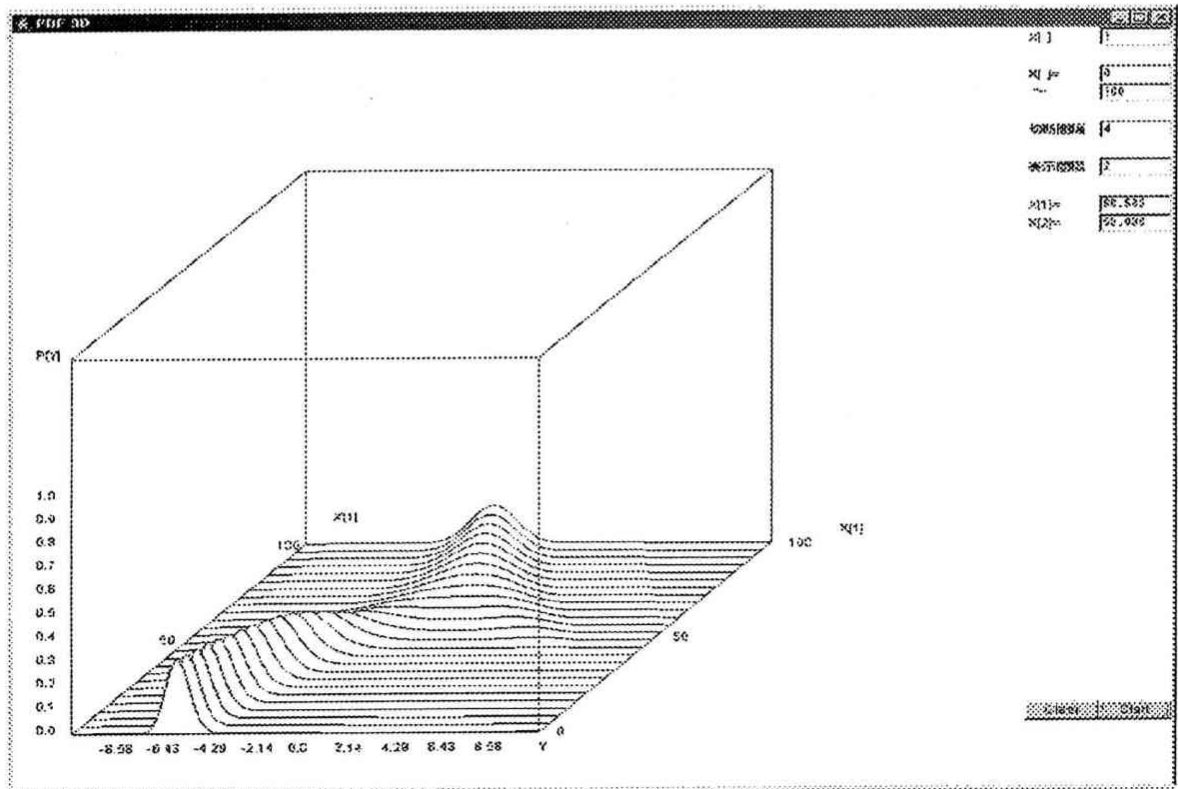


Fig. 7 Transition of probability density function.

5. 考察

動作確認から明らかになったのは、Cusp for JAVAはスタンドアロン状態⁸⁾と同様の計算結果が得られている。クライアントからの要求をサーバ上で処理した結果がクライアントへ返信されていることがわかる。Cusp for JAVAには拡張子.txtなどのテキストファイルを入力フォームに読み込む機能を実装しており、このテキストファイル読み込み機能はCusp for JAVAを用いて解析を行う際、ハンドリング作業を大幅に減らすことができる優秀な機能であるが、その機能を使うことはできなかった。テキストファイルの読み込み機能が実行できなかった理由としてはJavaサーブレットの仕様で、サーバ上で実行されたサーブレットからクライアント側にあるファイル等に干渉できない、というものがあげられる。

6. 結論

解析対象データの入力にハンドリング作業が残っているものの、Cusp for JAVAはスタンドアロン状態と同様の計算結果が得られており、クライアントからの要求をサーバのApacheが受け取り、その要求がTomcatへわたり、Tomcatが実行したCusp for JAVAがサーブレットとして起動し、クライアントへと返信されるというWebアプリケーションシステムが構築できた。これにより目的であるCusp for JAVAのWeb公開が可能となった。また今後Java

で解析ツールなどを作成した場合、本研究で構築したサーバを用いて同じくWeb公開が可能となる。

今後の課題としては、Cusp for JAVAの問題として、入力ファイル読み込み機能の実装、出力結果のサーバへの保存機能があげられる。サーバの問題としては、サーブレット利用者に対するユーザー認証やセキュリティのカスタマイズが必要である。

参考文献

- 1) L. Cobb, Cusp Surface Analysis User's Guide (1988) pp. 1-23.
- 2) Y. Kume, K. Okada and L. Cobb, in: Proc. of 11th International Conference on Human-Computer Interaction (2005) CD Edition.
- 3) 高原利之: Red Hat Linux 9 サーバ入門, ソーテック社 (2003) pp. 77-188
- 4) B. Ball and D. Pitts: 改訂3版標準RedHatLinuxリファレンス, 株式会社インプレス(2001) pp. 11-205
- 5) I. F. Darwin and J. Brittain: Tomcatハンドブック, オライリー社 (2003) pp. 1-282
- 6) SOHO・中小企業向け PC-UNIX サーバ講座: <http://pc-unix.goco.ne.jp/>
- 7) ITNAVI.com: <http://www.itnavi.com>
- 8) Y. Kume, N. Yamamoto and N. Sato, in: Proc. of 2002 7th Pacific Conference on Manufacturing & Management (2002) pp. 23-28.