

海外出張報告書

高度計算科学に関する米国の研究動向調査

1996年1月

動力炉・核燃料開発事業団

大洗工学センター

複製又はこの資料の入手については、下記にお問い合わせ下さい。

〒311-13 茨城県東茨城郡大洗町成田町4002

動力炉・核燃料開発事業団 大洗工学センター

システム開発推進部 技術管理室

Inquiries about copyright and reproduction should be addressed to:
Technology Management Section, O-arai Engineering Center, Power Reactor
and Nuclear Fuel Development Corporation 4002, Narita O-arai-machi Higashi-
Ibaraki-gun, Ibaraki, 311-14, Japan

動力炉・核燃料開発事業団 (Power Reactor and Nuclear Fuel Development
Corporation)1994

海外出張報告書

高度計算科学に関する米国の研究動向調査

大島 宏之*

要 旨

原子力基盤技術クロスオーバー研究「計算科学」分野において、「計算科学的手法による原子力分野の複雑現象の解明」を実施している。本出張は、国家的に情報インフラストラクチャの整備、高度計算科学研究を強力に推進している米国の現状・動向・研究（技術）情報を把握・分析し、研究に反映することを目的として実施したものである。

訪問先および日程は次のとおり。

1995 年 12 月 4 日～8 日： スーパーコンピューティング'95 会議（サンディエゴ）

12 月 11 日： マサチューセッツ工科大学メディアラボラトリー（ボストン）

12 月 12 日： クレイ社（ミネアポリス）

12 月 13 日： ミシシッピ大学エンジニアリングリサーチセンター（コロンプス）

本報告書は、これらの調査内容および入手資料をまとめたものである。

* 大洗工学センター基盤技術開発部熱流体技術開発室

目 次

1. 目的	1
2. 背景	1
3. 訪問先と調査項目	1
4. 調査報告	3
4. 1 スーパーコンピューティング'95	3
4. 2 マサチューセッツ工科大学メディア・ラボラトリー	7
4. 3 クレイリサーチ社	8
4. 4 ミシシッピ大学エンジニアリング研究センター	9
5. まとめ	11
Appendix 1 : 米国 HPCC 計画および NII 構想の概要	A-1
Appendix 2 : スーパーコンピューティング'95 会議 論文集の目次および技術論文各セッションの要約	A-7
Appendix 3 : スーパーコンピューティング'95 会議入手資料リスト	A-98
Appendix 4 : マサチューセッツ工科大学 メディア・ラボラトリ入手資料	A-100
Appendix 5 : クレイリサーチ社での入手資料	A-135
Appendix 6 : ミシシッピ大学エンジニアリング研究センター 入手資料	A-190

高度計算科学に関する米国の研究動向調査

1. 目 的

高度計算科学研究分野の先進国である米国において展開されているグランドチャレンジ（Appendix 1を参照）の現状を調査し、原子力基盤技術クロスオーバー研究に反映する。そのため、米国の高度計算科学に関する最新情報が集まるスーパーコンピューティング会議に参加するとともに、実際に研究を推進している大学や計算機メーカーを訪問し、討議する。

2. 背 景

原子力委員会が進める原子力基盤技術クロスオーバー研究において、平成6年度より新たに「原子力用計算科学」分野が設定された。動燃・原研・電総研の3機関が共同で提案した「計算科学的手法による原子力分野の複雑現象の解明」はここで承認され研究を開始した。動燃事業団は「流体-構造系の統合シミュレーションの研究」として、計算科学的手法の原子力分野への応用を図るためのソフトウェアの開発を実施するとともに、幹事機関として研究のとりまとめや報告も担当している。

この原子力用計算科学は、最近の飛躍的な計算機の性能向上にともない、従来多大な費用と時間のかかる実験に依存していた、あるいは実験不可能であった複合現象を、高性能コンピュータを駆使して解明しようとするもので、3機関は我が国におけるその先導的役割を果たすことが1つの使命とされている。5カ年という研究期間の中で研究を遂行するには、国家的に情報インフラストラクチャーの整備、高度計算科学研究を強力に推進している米国の現状・動向・研究（技術）情報を把握・分析し、タイムリーに3機関の研究に反映する必要がある。また、クロスオーバー研究交流委員会等で主査（東大矢川教授）並びに各委員からも、技術の進展がきわめて速い計算科学研究では最新情報を得つつ、国内外の各研究機関と討議しつつ研究を推進することが必要であると指摘されている。

3. 訪問先と調査項目

3.1 スーパーコンピューティング'95（12月4日～8日、サンディエゴ）

米国電気電子工学学会（IEEE）を中心にいくつかの学会の共催によって開催される米国最大の高度計算科学に関する会議で、米国あるいは世界におけるハードウェア／ソフトウェア技術の最先端の動向に触れることができる。ここではアプリケーション

ソフトウェア（解析コード等）の開発・利用の動向を中心に、超並列計算機を含む高速計算機および周辺技術の動向、高速計算から共同利用に至るネットワーク環境の動向などを調査した。

3.2 研究機関の調査・研究討議

出張者はマサチューセッツ工科大学、ミシシッピ大学、クレイリサーチ社を訪問し、画像処理、並列計算機、熱流動解析技術の動向を調査するとともに研究者と情報交換を行った。

(1) マサチューセッツ工科大学（12月11日、ボストン）

大学と民間の共同で設立運営されているメディア・ラボラトリーを訪問した。計算科学の前提である大規模解析は、今後益々大容量のデータを取り扱うことになるため、マンマシンインターフェースとしての高速プリ・ポストプロセッサシステムの開発は不可欠である。このラボラトリーでは、特に従来とは比較にならない情報記録容量を処理する技術を研究しているので（グラフィックやバーチャルリアリティを含む）、プリ・ポストプロセッサシステムの在り方・方向性に関する知見を得ることを目的に、これらの処理技術等を調査した。

(2) クレイリサーチ社（12月12日、ミネアポリス）

スーパーコンピュータの開発を先行してきた会社を訪問した。CPU性能向上の鈍化を「CPU並列化」で打破しようとする動きが顕著であるが、これを利用するにはアプリケーションもハードの特性を考慮したプログラミング構造が要求される（並列利用技術の確立はこれからであり、標準化技術もまだ存在しない）。従って、解析プログラムの開発は今後のハードや周辺技術の開発動向を勘案しながら進める必要がある。ここでは、動燃に導入された並列計算機T3Dをはじめとしたハードの動向、熱流動解析を含むアプリケーション類を調査するとともに、動燃が進める研究内容を紹介し議論を行った。また、超高速計算機技術を開発してきた同社の今後の計算機技術の実用化課題も併せて調査した。

(3) ミシシッピ大学（12月13日、コロンバス）

同大学のComputational Field Simulation Centerを訪問する。ここは学部の規模でComputational Fluid Dynamics（CFD）を行っているという世界でも類を見ない計算科学研究の場である。ここでは、現象解明の研究現場としての環境（システム効率性、機能性を含む）や、各種アプリケーションソフトについて調査した。

4. 調査報告

4.1 スーパーコンピューティング' 95

(1) 会議全体概要

今年で8回目を迎えた本会議は、サンディエゴのコンベンションセンターで開催された。この間、米国政府のHigh Performance Computing and Communications (HPCC) 計画、National Information Infrastructure (NII) 計画とリンクされ、発表や展示等の規模が飛躍的に増大するとともに、従来の「会議」という枠組みを超えた様々な趣向が凝らされるようになった。今回の参加者は登録者だけでも約5千人であり、各領域の研究者のみならず銀行関係者といった調査団も見受けられた。会議は基調講演、招待講演、受賞講演、チュートリアル（講習会）、技術論文、ワークショップ、パネル、ラウンドテーブル、Education program（青少年育成報告）、展示（大学、研究所、メーカー）、出展者フォーラムなど盛りだくさんの企画で構成されており、出張者1名ではとても網羅できるものではなかった。

会議の最初に行われた基調講演（William A. Wulf教授：バージニア大学）では、スーパーコンピュータはもはや一部の学者の専有物ではなくなり、一般人のレベルまで到達していること、学問・研究そして人々のコミュニケーションまでが進歩した情報技術により大きく変化してきた状況に触れた後、スーパーコンピュータを取り巻く環境および米国の今後のあるべき姿勢が述べられた。ここでは、科学技術分野において計算科学が浸透し、シミュレーションの重要性・必要性が広く認識され、もはやスーパーコンピュータは「特殊なもの」ではなくなったが、予算削減など厳しい現実に対して新しい展望を切り開く努力が必要であること、この会議が10年先の社会、コンピュータ業界の反映したものであり、方向性を定めリーダーシップを維持していくものでなければならないこと、科学技術を育てる責任、大学・企業の柔軟な協力関係の必要性などが強調された。（予算削減は大統領選挙の影響が大きいとのこと）

会議のトーンとしては、テラ（ 10^{12} ）フロップス（註1）へ向けてのマシン開発、国家的ネットワーク、その両者による応用面の拡大（科学、工学、医療など）、そして特に次世代へ向けての強力な人材育成（青少年教育）といったものが感じられた。マシンについては、当初の目的であったテラフロップス到達は時間の問題であり、次世代マシンの目標としてペタ（ 10^{15} ）フロップスというキーワードが登場した。一方、ハードウェアの進歩に対して並列処理技術の立ち遅れは否めず、各分野の発表においても新規性に乏しく既存手法の並列化がほとんどであった。但し、全般的には（HPCC計画によるフェーズ1としての）高速化したコンピュータおよびその環境が整い、「計算科学の実用化」に近づいたという印象を受けた。スーパーコンピューティングはマシン性能の追求「如何に速く処理

するか」から、それらを応用して「如何に現象を理解するか」へ焦点が移りつつある。それゆえ大規模シミュレーションの結果発生する大規模データの取り扱い（保存・アクセス）、大規模データによる現象解明を行う手段としての多次元グラフィックス（画像・映像による表現技術）の開発が今後益々重要となってくる。

註1：FLOPS = Floating-point Operations Per Second (1秒間に処理する浮動小数点演算数)

参考：PNCに導入されている富士通VP2600（ベクトルマシン）の公称最大性能は5ギガ（10⁹）フロップス(GFLOPS)である。ベクトル並列マシンである富士通VPP500は、1プロセッサ(PE)あたり1.6GFLOPSのものを128個使用することにより、行列演算ベンチマーク問題で150GFLOPSを達成した実績を持つ。

(2) チュートリアル（講習会）

会議初日と2日目に半日コース・1日コース合わせて17のセッションが開催された（参加には高額な別料金が必要）。並列計算機システム（ネットワーク含む）、並列・高速プログラミング技法（各種プログラム言語含む）といった解析手段／手法の分野に加えて、高性能計算機から得られる大規模なデータを処理する科学計算ビジュアル化手法、仮想現実（バーチャルリアリティ）システムなども含まれていた。ここでも、大規模高速計算が実際に動き出すにつれて、得られた膨大な結果をどう処理してどう理解するかといったところに関心が集まっている。

(3) 技術論文発表

計算処理能力向上の恩恵が様々な分野に浸透してきているのを反映して、今年は技術論文においても、計算機アーキテクチャ・ツール・コンパイラなどに加えて、生化学、工学、流体力学、海洋・気象学、物理学といった応用分野のセッションも設けられた（論文は厳選されており、その平均受理率は29%である）。しかしながら、これらの応用分野セッションは全体的に新規性は乏しく、既存手法の並列化や処理速度向上に興味の対象が向いている印象を受けた。適用対象とする問題は分野によって様々であるが、中身は類似しており既存の手法をスケーラビリティ（註2）を重視して並列化することに主眼が置かれている発表が多かった。「スーパーコンピューティング」に関する会議であるためある程度は仕方がないことだが、並列計算機が登場した当初期待された「プロセッサ数に比例して処理速度が向上」の達成は容易ではないということがあらためて浮き彫りにされた感じである。

流体解析関連では、マルチグリッド法による非圧縮性流体解析の並列処理、乱流直接シミュレーション、並列スケーラビリティの高い圧縮性流体解析手法などが報告された。全体的に大規模な計算を狙ったものであったが、「ハイパフォーマンス」の意味として、如

何に高速に膨大な問題を解くかというよりも、ネットワークにより各地に分散する計算資源・周辺技術を有機的に結合し、如何に容易に且つ低コストで要求を満たせるかを力点とした発表が多かった。

なお、会議論文集は従来の製本版は廃止され、CD-ROM 1枚のみの配布となった。これは持ち運びを容易にするだけでなく、CD-ROMという特色を生かして、解析結果のグラフィック表示やアニメーションまでも収録するものであり、購入者は自前のパソコンで再生が可能である。

論文集の目次および各セッションの要約をAppendix 2に示す。

註2：並列処理において、プロセッサ数の増加に対する処理速度向上の割合を表す。

(4) High Performance Computing Challenge および GII Testbed

High Performance Computing Challengeは最大級のプロセッサ数を用いて（単純化されたベンチマークではなく）具体的なアプリケーションプログラムでテラフロップスの処理速度を狙おうとする試みである。参加グループが使用マシンとアプリケーションプログラムを決め、短期間で並列化等のチューニングを行い計算した結果が示された。また、GII Testbedはグランドチャレンジ（Appendix 1を参照）で様々な研究所が取り組んでいる問題について、実際に各研究所とコンベンションセンターを高速ネットワークで結ぶことによって、遠隔でリアルタイムで解析したりその結果をインタラクティブに表示する試みがなされた。このTestbedの目的はハード、ソフト、情報そして研究グループ（人材）を国家レベルで共有できるシステムを開発することであり、マルチ-サイト共同研究を促進し研究開発効率を向上させることが狙いである。CAVE（註3）という3次元可視化手法やバーチャルリアリティを体験できるコーナーとしても人気を博していた。

註3：立方体空間内に映像を再生できる高画質3次元ビデオ・オーディオシステムで、数人が同時に仮想現実を体験できる。電中研でも利用を検討しているとのこと。

(5) 展示会場

約15,000m²という広大な面積に、計算機、ソフトウェア、コミュニケーションといったメーカーブース（industry booth）が100以上、50のポスター、そして大学・研究所などの研究機関ブース（research exhibits）が45程度出展されていた。

メーカーブース

今年は新製品や新しいアイデアといったものは少ないようで、並列計算機で商業的に唯一成功を収めているクレイ社の新製品T3E（動燃に導入しているT3Dの後継機）の人氣が高かった。他の米国計算機メーカーは独自の趣向、特にグラフィックを駆使したデモンストレーション（普通の解析も見せ方によってはこうも迫力が出るものかと思わすべきところもあった）により計算機の性能をアピールしていた。例えば、インテル社の並列マシンであるパラゴンでは、サンディア国立研究所提供の地震解析や爆発事故解析シミュレーションなどである。ちなみに、CMシリーズの開発販売など超並列計算機の先駆的存在であったThinking Machines社はこの分野から撤退し、今やワークステーションを並列に接続したシステムと周辺ソフトのみを扱うだけとなってしまった。並列計算機だけの会社はことごとく撤退し、IBMやクレイなど他の分野で十分収益のある（体力のある）会社のみが生き残ったようである（十分実用に耐える並列計算機の開発はそれだけ困難ということかもしれない）。

日本のメーカーのブースは、参加実績により場所を割り振られることや企画／アピール力で劣ることから、人はまばらであった。富士通は高性能だが高価すぎると評価されたVPP500に代わりVPP300（動燃に今春導入予定）を全面に押し出していた。また、日立も多少遅れた感はあるものの高性能並列計算機を今春発売する予定である（日立関連者は聴講も含めて約30名近くも参加していたようである）。テラフロップスを目指す並列計算機ともなれば、（プロセッサが多くなる分）マシンの信頼性確保は重要な課題であり、バックアップ用プロセッサを備えるなどの対策を模索している。

グラフィックスに関しては、シリコングラフィック社のブースが圧倒的で、マシンのみならず多くの提携ソフトハウスによる様々なグラフィック表示システムのデモ（人体、気象、流体、自動車クラッシュ、スペースシャトル、分子構造など）が展開されていた。動燃においても計算科学の進展に伴い今後益々プリ・ポストプロセッサ（グラフィック表示システム）のニーズが高まるが、この分野での日米間のハード・ソフトのレベルの差および開発効率ということを考慮すると、このようなソフトハウスと提携しシステムを開発することが最も効率よく且つ有効であるという印象を受けた。

研究機関ブース

- ・ Research Exhibitsには、米国の航空関係・軍事関係の研究所、コンピュータセンター、大学に混じって、日本からも東大、東工大、早稲田大、埼玉大などが参加していた。特に東大は世界最高速宇宙物理専用シミュレーションマシンGRAPE-4（884Gflops）でGordon Bell Prizeを受賞している。
- ・ 原子力に携わったアルゴンヌ、ロスアラモスといった国立研究所は軒並み立派なブースをかまえ、その解析能力の高さをアピールしていた。出展のテーマは流体、構造、気象、環境、医療など様々な分野に広がっている。例えば、ローレンスリバモア国立研究所では、「エネルギーサイエンス」の1つとしてCAVEシステムを利用したトカマク乱流解析シミュレーション、医療関係として骨粗鬆症分析や人体解剖バーチャルリアリ

ティを実際に体験できるコーナーなどが設置されていた。また、オレゴン国立研究所では、地下水、気象、材料物性シミュレーションや組織マネジメントシステムといった具合である。いずれも原子力に直接関連するものはなかった。

(ロスアラモスのブースでは来年はPNCブースの出現を待っているとのメッセージを頂いた。また、会場で偶然お会いした東工大高橋教授(クロスオーバー研究副主査)からも「誰もやったことがない大規模計算でもっと動燃をアピールすべきである」とのご意見を頂いた。)

技術論文発表のOHPシートのみならず、ポスターセッション、各機関の展示などすべてにわたって専門家の電子メールアドレスが表示されており、質問等に対する詳細な回答は電子メールで送るあるいは問い合わせしてくれといった応答が多かった。また、ほとんどのブースでは、インターネット(WWW)による各機関のホームページへのアドレスも掲示していた。インターネットをベースとした情報化社会をいかに有効に活用できるかが、技術者・研究者にとって死活問題になることを予見させられた。

展示会場等で入手した資料のリストをAppendix 3に示す。(資料閲覧の希望は直接報告者に連絡されたし。)

(6) 教育・人材育成プログラム

今回の調査の対象ではないが、情報基盤を支える将来の人材の育成に米国はかなり熱心であるとの印象を受けた。教育に関する発表は、ポスターも含めると会議全体の約1/4に達していた。会場には、高校生あるいは中学生によるコンピュータやネットワーク環境を利用した研究のポスター発表がところ狭しとばかりに展示されており、中にはグラフィック表示による気象計算結果など研究所の発表と互角ではないかと思われるようなレベルの高いものまで見受けられた。

4.2 マサチューセッツ工科大学メディア・ラボラトリー

コミュニケーションの将来像を探求することを目的として1985年に設立されたメディアラボは研究領域を徐々に拡大し、現在Learning&Common Sense, Perceptual Computing, そしてInformation&Entertainmentの3つの大きなプロジェクトの中で101の研究テーマが進行中である(機器提供を含め資金提供者は、コンピュータ関連、放送、出版、通信事業等100社以上である)。ここではコンピュータの専門家のみならず、デザイナーも参加して研究が進められている。

多くの研究テーマのうち、特に興味深かったのはホログラフィーの活用である。これは

光の干渉や回折といった性質を利用し、物体から反射してきた信号を干渉縞として記録して、参照光を当てることにより立体的に再生するという原理である。その応用として、例えば医療で用いられている核磁気共鳴によるの再生映像に適用すれば、対象部位を自由に立体映像としてとらえることができる。これは高速炉のような複雑形状内解析結果の表示に対しても応用できると考えられる。大規模解析結果の処理方法の1つとして検討に値する技術である。

入手資料をAppendix 4に示す。

4.3 クレイリサーチ社

クレイ社の概要、マシンラインナップの現状と特徴の紹介後、最新の超並列計算機であるT3Eの概要と今後、並列計算機用プログラミング支援環境、アプリケーションとして流動解析コードSTAR-CDの開発現状の説明を受けた。

- ・クレイ社のマシンは、現在34カ国、678システムが世界で運用されており、その中で日本の占める割合は16%で米国内に於ける最大のユーザーである。
- ・研究所のソフト部隊は約80人でそのうち数値流体解析Gr.は8名である（これは他のハード会社に比べて多い）。
- ・T3Dの後継機であるT3Eは、メモリ容量（共有／分散メモリどちらも可）、結合プロセッサ(PE)数の拡大、Giga Ringという名称の新しい通信様式、2次キャッシュ（並列計算機はこの効果が大きい）を取り入れることにより、少なく見積っても同数のPEを有するT3Dに対して実際のアプリケーションベースで数倍以上の速度向上が見込める（十分マーケットを調査した上で性能と価格をバランスさせているとのこと）。いずれ低価格で1プロセッサあたり2ギガバイトまでメモリを拡張できるようにするとのことである。また、並列マシンは多数のPEを使用するためその信頼性を確保することが重要であるが、クレイ社はバックアップパワーを組み込むとともに、PEどうしを結合する方法として従来のワイヤーを廃止し、独自のZipコネクタという方法を開発している。
- ・ソフトウェアの開発方針としては、クレイシステムすべての共通オペレーションシステム（OS）であるUNICOSをさらに進化させ、すべてのPEで別のOSを同時に走らせることを可能としたり、Support PEを設けることにより、PE全体を統括して計算させる形態も可能となるようにする。これは計算対象の拡大、アルゴリズムの変革をもたらすことになるであろう。また、可能な限りユーザーがシングルPEイメージで利用できる環境も提供するとのことである。現在、並列コンパイラの最適化や自動パラレル化モデルの開発に力を入れている。
- ・フォートラン並列プログラミングモデルにおいては、クレイ独自のCRAFTをIBMとの協力によってHPC（ハイパフォーマンスフォートラン）に統合していく方向（移植性の観点）が指向されている。一方、C言語については世界標準的なものが未だ無いため

implicit modelは研究段階である。科学計算ライブラリーについては、T3E用のチューニングを実施している。

- ・流動解析コードSTAR-CDは現在単一CPUで500Mflops (T90上)を発揮できる有限体積法コードであり、主に自動車産業界（ベンツ, BMW, 日産など）で使用されている。現在領域分割の手法を用いて分散メモリ型並列計算機版が作成されている。
- ・東大矢川研出身者で今年クレイに就職した人物に会うことができた。彼は構造解析における自動メッシュ分割で博士号を取得しており、今後クレイがその分野へも積極的に参入していく姿勢が伺われた。
- ・動燃で実施しているT3Dを用いたアプリケーション例を紹介し議論を行った。共通の課題として行列解法の高速化が1つの焦点となったが、共役勾配法（CG）法を利用する場合には高速化のネックとなる前処理部分を、メッセージパッシング方式で組み直すだけでも効率がアップするとのアドバイスを受けた。
- ・最後に「もんじゅ事故」について、スペースシャトルの爆発事故対应当時为例に挙げ、国をあげて徹底的に（計算機による）解析評価を行い、完璧なものを作り上げて欲しいと激励された。

入手資料をAppendix 5に示す。

4. 4 Engineering Reserach Center (ERC) for Computational Field Simulation (ミシシッピ大学)

- ・ERCは米国HPCC計画の一環として、産業界に貢献すべく様々な複雑形状における大規模数値シミュレーションを開発する研究所として、多くのスポンサーのもとに1990年設立された。ここには、計算メッシュ生成手法開発の大御所であるThompson教授をセンター長として、computational engineer, computer engineer, computer scientist, mathematician, application engineer約100名が研究しており、機能的にも知的活動のための居住空間としても非常に快適な環境が提供されている。
- ・研究プロジェクトは、計算メッシュ生成、解法アルゴリズム、科学的ビジュアル化、システムソフトウェア、計算機アーキテクチャ、システムインテグレーションの7つであり、これに教育プログラムとして計算工学、計算機工学が設置されている。
- ・具体的な適用例として、航空機（戦闘機）、潜水艦、ミサイルといった軍事関係、人工心臓内流動など医療関係での流動解析が紹介されたが、そこに含まれる非常に複雑な形状のモデル化や複雑な動きをする移動境界の取り扱いなど、解析技術の高さを感じさせられた。また、ビジュアル化に対してもハードやソフトウェアといったシステム技術だけではなく、表現の仕方（人間の感性や芸術性）そのものを研究している者もいる（東工大高橋教授がかねてから芸術家を雇って研究すべきとの意見を持たれている）。
- ・メッシュ生成については、この研究所の大きな目玉であり、20人程度の研究者がそれぞ

れの課題に対する自動あるいは半自動メッシュ分割手法の開発に取り組んでいた。最も興味深かったのは、複雑形状に対して汎用化をはかるため2次元ではあるが既にハイブリッド法（形状模擬性の高い三角メッシュと解析精度の高い四角メッシュを組合わせた自動メッシュ分割法）が完成していたことであった。

- ・複雑形状メッシュ分割に対して適用される流体解析コードは、ポピュラーな有限要素法ではなく、ほとんどが有限体積法であった。
- ・動燃の概要説明の中で、複雑形状の代表としてワイヤースペーサ型燃料集合体内のイメージ図を提示してみた。彼らのツールでも、3次元空間での螺旋状ワイヤーの存在、隣接燃料ピンとの接触の有無によって、メッシュ分割は困難であろうとの見解であった。お互いの条件が合えば、共同研究という形でメッシュスキームの開発を進めてもよいとの提案を受けた。

入手資料をAppendix 6に示す。

5. まとめ

原子力基盤技術クロスオーバー研究（計算科学）に反映することを目的として、国際会議へ参加し、また大学やメーカーを訪問し研究討議を行うことにより、国家的に情報基盤の整備や高度計算科学研究を推進している米国の現状を調査した。全体を通しての印象を以下にまとめる。

- ・米国のHPCC計画、NII構想の推進政策によって、並列処理による計算機処理能力向上がハード・ソフト両面で思いのほか課題が多く伸び悩んでいるものの、着実に計算能力はテラ・フロップスの世界に到達可能な状況になってきている（4年間で1000倍の処理速度向上を達成し、現在はおおよそ0.5テラフロップスといったレベルである）。今後のR&Dはこのテラフロップスの性能で何がどこまでできるか、あるいは何が解決できるかを十分吟味しその方向性を定める必要がある。注意すべき点は、この性能は単一CPUの性能向上ではなく並列化という手段で達成されるものであるため、余程高性能の自動プログラムチューニングシステムでもできない限り、ユーザーは並列アーキテクチャを意識したプログラミングを行わなければその恩恵を享受することはできない。大規模計算が必然化する動きに乗り遅れないためにも、人材育成など何らかの対処が必要であろう（既存コードの大規模問題への適用に対しては、並列化を外注するというオプションもあるが、日本では並列化の専門家が少ないこと、並列を意識していないプログラムのチューニングは計算効率の改善に限界があることに留意しておかなければならない）。
- ・全米中を結ぶ高速ネットワークの整備は、単なる通信のみならず情報、計算資源、そして人材の共有化までも可能にし、生産性を向上させている。研究者（もしくはビジネスマン）は必要なときに必要な情報を即座に入手することができるだけでなく、ネットワーク上にある様々な計算機さらにはソフトウェアを単独あるいは複数を結合して、ニーズに応じた解析を行うことができる（これは高校生や大学生でも同様である）。言ってみれば1研究室では到底備えることのできない研究環境を、ネットワークに接続するだけで誰もが容易に手に入れることができるのである。高速並列計算機に加えて、ネットワーク上のワークステーションを結合して低コストで大規模計算を行うといった発表が多かったのもこの背景のためであろう。動燃内においてもネットワークを積極的に利用することにより人、情報、資料などに対するアクセスを改善し、研究だけでなく業務全般の効率を上げることは可能なはずである。
- ・大規模解析を可能とする上述のような環境の整備は、計算科学を誕生させるなど従来の研究アプローチを様変わりさせている。解析においては、以前は計算機の実力の制約により実験で決定しなければならないパラメータを含む工学モデルを用いざるを得なかったが、このようなモデルを可能な限り排除し、複雑な現象をそのまま直接解く方向

へ進みつつある（もっとも原子炉における乱流の直接解析などは依然として計算機のパワーが足りないが）。従って、解析対象空間を細かく且つ最適にメッシュ分割する手法と、支配方程式を直接高速に解く手法が今後益々重要となる。また今回の会議でも強調されていたことだが、大規模計算が実際に行われてくるにつれその膨大な出力をどのように処理するか、どのように現象解明に結びつけるかという課題を解決しなければならない。その手段として多次元グラフィックスなど画像・映像による再現技術、さらには「見せ方」を工夫する表現技術の開発は重要である。映像のインパクトは強烈である。これは技術者だけでなく、PAの観点からも1つの研究課題である。

以上の全体の印象に加えて、計算環境に関して動燃の現状を少々振り返ってみたい。

動燃では業務が増加する中、プログラムの作成作業、事務計算、小規模から大規模解析までの多くが、センターマシン上で処理されている。このように様々なスケールが混在する状況では、小規模処理は計算効率が悪く大規模処理は待ち時間が長くなるあるいは不可能となり、今後益々研究開発に様々な支障が出てくる可能性がある。現在、既に高性能で且つ安価な計算機（パソコンも含む）が様々なニーズを満たすべく出現してきている。これらを積極的に取り入れることにより事務的作業を含む業務の効率アップ（人的効率・時間的効率を向上させられるものはたくさんある）を図るとともに、大型計算機は大規模計算・計算科学専用としてその特徴を十分生かせるシステムとすべきであろう。

R&Dにおいては、計算機は実験装置と同等である。実験不可能なものをシミュレーションするだけでなく、可能な限り実験を代替することにより時間的にもコスト的にも、また精度的にも大幅な改善がなされる可能性がある。数人のGrが1つの課題のために大きな試験装置を製作し独占して研究を行うことに疑問を持たないのと同様に、1つの計算機を「実験装置」として独占して活用するといった利用方法も少しずつ取り入れていく必要があると考える。また、これは上から与えていくものではなく（これでは過去の経験からみても最大公約数的なシステムしかつくれず、それぞれのニーズを十分には満たせないために結局利用者がなく宝の持ち腐れ状態に陥る）、R&Dの現場でそのような環境を時代に敏感に反応しながら自ら作り上げようとする雰囲気、またそれを巧くサポートし且つ全体のバランスもとろうとする企画・管理システムといったもの、言ってみれば「文化・土壌」を育てていくことが新しいR&Dのやり方、そしてそれによる革新的な成果を誕生させると考える。計算機に関しては下手な平等主義は弊害である。自ら行おうとする芽を育て文化形成の先導役とし、そうでないところは積極的な情報提供・指導を行い啓蒙していくことが重要である。

次世代の高速炉では、安全性を確保しつつ経済性をさらに向上させることが要求されている。これらを合理的に実現するためには、従来の開発の方法論の延長ではその目標を達成することは困難である（実験物理・理論物理の限界）。一つの有力な新しい手法は、第3の科学と称される「計算科学」によるアプローチである。この方法によって、従来の大規模な試験に依存することなく、設計評価・安全評価作業のほとんどの部分を計算機上で

効率よく且つ高い精度で実行することが可能となる。計算科学を実現すべく、計算機ハードウェア、ネットワーク、あるいはアプリケーションは極めて速い速度で進歩している。原子力分野においても、常にこれらの最新の状況を理解した上で適した技術を活用することが大切である。

本出張により入手した情報・知見を、現在進めている流体と構造物の相互作用などの熱流動現象を解析する手法の開発等に順次活用していくとともに、計算技術を活用した設計最適化・安全裕度適正化に反映していく予定である。

<APPENDIX 1>

米国HPCC計画およびNII構想の概要

米国HPCC計画およびNII構想の概要

1. High Performance Computing and Communication (HPCC) 計画の背景

1980年代より、コンピュータ・ハードウェアやソフトウェアの性能が著しく向上するに伴い、コンピュータによる数値計算の科学／工学への応用研究が進んできた。1980年代初期に行われた高性能コンピューティング政策研究では、米国の一般の研究者が高性能コンピューティング資源に十分にアクセスできないでいること、また、このために米国のコンピュータ産業の強力な地位が脅かされつつあることなどを指摘するとともに、米国政府が率先して、高性能コンピューティング政策に取り組んでいく必要があることを示唆した。

当時米国では、全米科学財団（National Science Foundation: NSFと略称する）、国防総省（Department of Defence: DOD）、航空宇宙局（National Aeronautics and Space Administration: NASA）、エネルギー省（Department of Energy: DOE）など、いくつかの連邦機関が各々独立に高性能コンピューティングおよびネットワーキングに関するプロジェクトを進めていた。しかし、前述したような高性能コンピューティング政策研究を踏まえて、1980年代半ばからNSFがスーパーコンピューティングセンターの設立およびセンター間を接続するバックボーン・ネットワーク(NSFNET) の創設を開始した。これを機会として、DOE, NASA, DODなども同様の計画を打ち出し、連邦機関間および全国の研究・教育機関を相互につなぐ大規模ネットワーク構想が具体化されるようになってきた。

このような状況の下に、前述の高性能コンピューティング政策研究などを踏まえて、連邦政府による高性能コンピューティング構想研究が進められた。この結果、1989年に大統領府の一期間である科学技術政策局（Office of Science and Technology Policy: OSTP）が、米国議会の要請を受けてHPCC計画構想に関する報告書（The Federal High Performance Computing Program）を提出した。OSTPはこれを踏まえて、1992年度大統領予算教書の補足書（Grand Challenges: High Performance Computing and Communications）にHPCC計画の内容を公表した（1991年）。

2. HPCC計画の流れ

図A-1にHPCC計画の流れを示す。

HPCC計画はブッシュ政権下で発足し、2年間の助走期間を経てクリントン政権に引き継がれた。一方、クリントン政権は全米情報基盤(National Information Infrastructure: NII) 構想を発表したが、これに伴いそれまでのHPCC計画に変更が加えられ、新しいHPCC計画がスタートした。

以下、ブッシュ政権とクリントン政権におけるHPCC計画の流れを示す。

(1) ブッシュ政権下のHPCC計画の流れ

1991年12月にブッシュ政権下で高性能コンピューティング法が成立し、5年間にわたるHPCC計画（1992～1996年度）が正式にスタートした。1992年には引き続いて1993年度HPCC計画が1993年度大統領予算教書の補足書（Grand Challenges 1993: High Performance Computing and Communications）に公表された。当初HPCC計画は「グランド・チャレンジ」と呼ばれる、国家的なニーズである根源的な科学／工学問題の解決を目標としていた。グランド・チャレンジは、航空工学、生物医学、地球環境、エネルギー、宇宙・天体物理、材料科学などの分野における、今後究明あるいは解明すべき重要な問題をあらわすものであり、このような問題を解決するには高性能コンピューティング・通信基盤を必要とする。このためブッシュ政権下のHPCC計画は、高性能コンピューティング・システム（超並列コンピューティング・システムを含む）の開発、高速コンピュータネットワークの開発、先端ソフトウェア／アルゴリズムの開発、および基礎研究の支援と先端コンピューティングにおける人材の育成の4つの要素を軸として構成されていた。

(2) クリントン政権下のHPCC計画の流れ

1992年にクリントン政権は、最優先の科学技術政策としてNOO構想を発表した。クリントン政権のHPCC計画は、NII構想を踏まえて策定されており、1993年には新しいHPCC計画（1994年度HPCC計画）が、1994年大統領予算教書の補足書（High Performance Computing and Communications: Toward a National Information Infrastructure）に公表された。新しいHPCC計画ではNII構想を支援することを目的の1つとしており、グランド・チャレンジに加えてナショナル・チャレンジ（市民生活、健康・医療、教育、ビジネス、製造／生産、環境保全、国家安全保障などの面で、国家の競争力と国民の繁栄にインパクトを与え、HPCC技術の応用により利益を与え得る根源的なアプリケーション）と呼ばれる情報基盤技術の確立が新たな目標として追加された。この一環として前述したそれまでの4つのHPCC計画の構成要素に加えて、NIIの根源的技術の開発（情報基盤技術・アプリケーションの開発）が追加された。

1994年にはHPCC計画の活動状況やNIIを構成するHPCC技術の紹介を目的として、1995年度HPCC計画が、1995年度大統領予算教書の補足書（High Performance Computing and Communications: Technology for the National Information Infrastructure）に公表された。1995年にはHPCC計画の具体的な成果の公表を目的として、1996年度HPCC計画が、1996年度大統領予算教書の補足書（High Performance Computing and Communications: Foundation for American's Information Future）に公表された。なお、クリントン政権から各年度のHPCC実施計画が公表されるようになり、1994年には1995年度実施計画（FY1995 Implementation Plan）が、1995年には1996年度実施計画（FY1996 Implementation Plan）が公表されている。

3. HPCC計画の目標

HPCC計画の目標は以下のとおりである。

- (1) 高性能コンピューティングとコンピュータ通信の分野で、米国の技術的指導力を強化し拡大する。
- (2) 技術革新を加速し、経済競争力、国家安全保障、教育、医療、および地球環境の向上に資する幅広い技術の普及と応用を促進する。
- (3) NII（全米情報基盤）を実現させる重要技術を開発し、NIIアプリケーションの実用性を立証する。

4. HPCC計画の戦略

HPCC計画の戦略は、以下のとおりである。

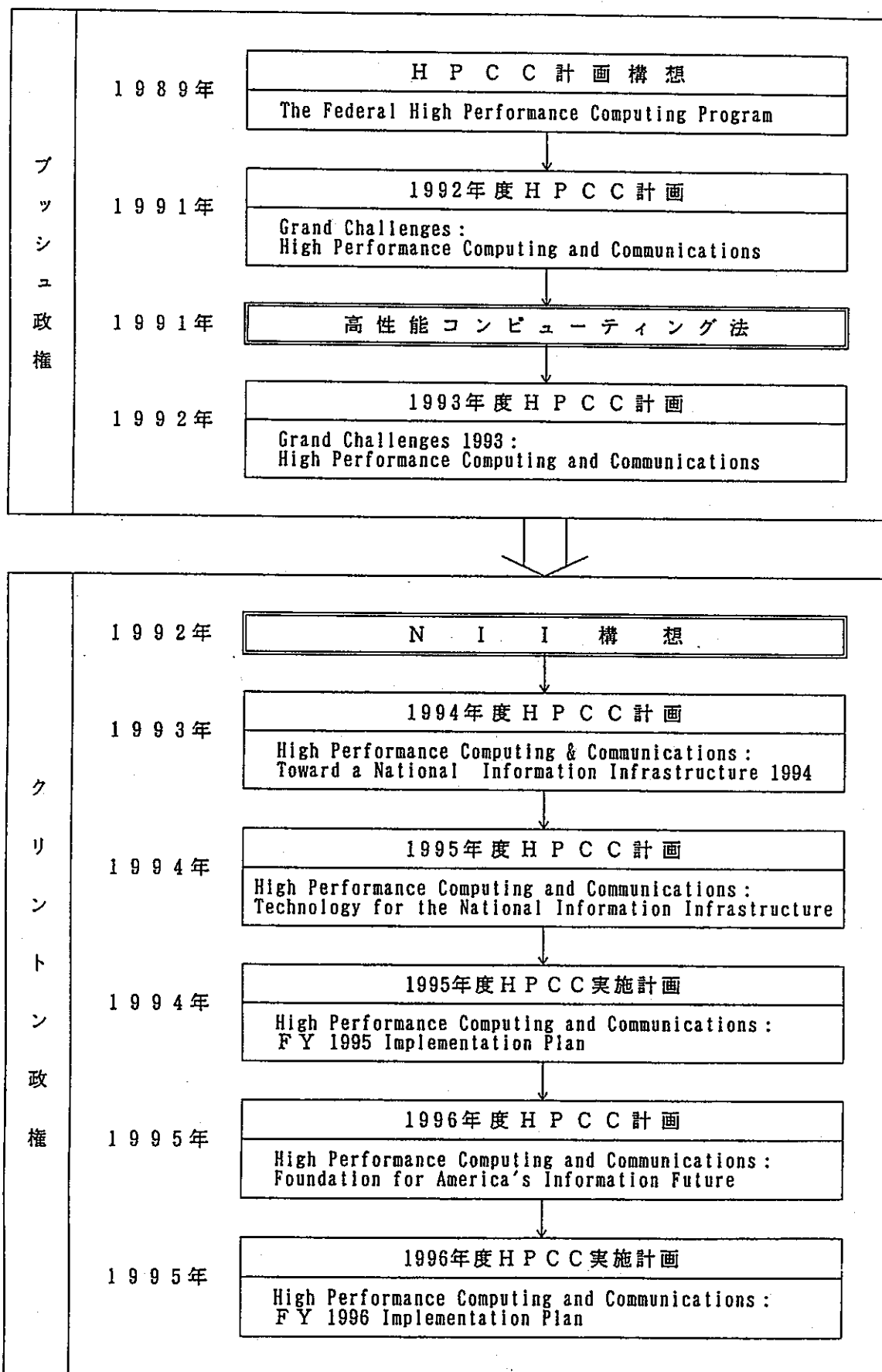
- (1) 産業界との共同により、大規模な科学／工学問題（グランド・チャレンジ問題）に対応可能な、少なくとも1テラ・フロップスの計算能力を有する、スケーラブル並列アーキテクチャを使用した高性能コンピューティング・システムを開発する。
- (2) インターネットの拡張および性能向上を支援することによって、HPCCを構成するすべての要素をサポートする。
- (3) 産業界との共同により、全国的なギガ・ビット・ネットワークを展開するのに必要なネットワーキング技術を開発する。
- (4) グランド・チャレンジおよびナショナル・チャレンジ・アプリケーションの共同研究に役立つ、広域のギガ・ビット・ネットワークの生産性を立証する。
- (5) テラ・フロップス・レベルのコンピューティング性能を活用して、グランド・チャレンジ問題の解析を行い、その妥当性を立証する。
- (6) 国家安全保障、公衆衛生、公衆の安全、環境、教育などに関わる国家の使命遂行能力の向上を可能にする、グランド・チャレンジ、ナショナル・チャレンジおよびその他の先端アプリケーション問題を解決するための、高性能コンピューティング・システム、創作的ソフトウェア／アルゴリズムおよびネットワーク・アクセス技術を提供し、それらを利用することによって技術革新を促進する。
- (7) 米国の研究および産業界におけるHPCC技術の利用と普及を促進するため、高性能コンピューティング研究センター（HPCRC）、ネットワーク、協力体制などを含めた共同研究基盤を構築する。
- (8) 産業界との共同研究により、世界情報基盤（GII）をサポートする情報基盤技術を開発する。
- (9) ナショナル・チャレンジ・アプリケーションを実行するために、産業界との強調によりHPCCへの投資を強化する。
- (10) 高等学校、大学学部および大学院レベルで、全国的に認識され受け入れられる計算科学教育プログラムを確立する。これにより基礎研究のための幅広く認識された学問として、計算科学の質の向上を図る。
- (11) コンピュータ科学、コンピュータ工学、計算科学／工学および情報科学分野における

大学院や博士課程修了者へのフェローシップを強化する。また、大学学部レベルでの計算科学分野の奨学金およびフェローシップ制度を設ける。

5. HPCC計画参加の連邦機関

1995年より新たに厚生省健康・医療政策研究局および軍人復員省が計画に参加することになった。従って現在HPCC計画に参加している連邦機関は以下の12の機関である。

- (1) 厚生省健康・医療政策研究局
- (2) 国防総省口頭研究計画局
- (3) エネルギー省
- (4) 教育省
- (5) 環境保護庁
- (6) 航空宇宙局
- (7) 厚生省国立衛生研究所
- (8) 商務省標準技術研究所
- (9) 商務省海洋大気局
- (10) 国防総省国家安全保障局
- (11) 全米科学財団
- (12) 軍人復員省



図A-1 H P C C 計画の全体的な流れ

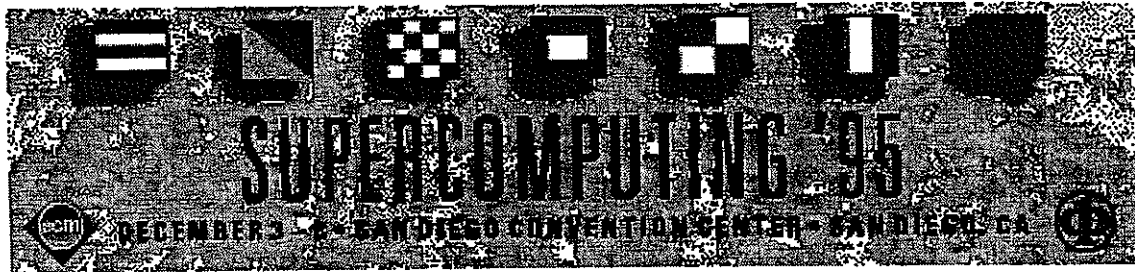
<APPENDIX 2>

スーパーコンピューティング'95会議

○ 論文集の目次

○ 技術論文各セッションの要約

Title Page	SC'95 TOCs: By SESSION	By AUTHOR	INDEX
------------	------------------------	-----------	-------



PROCEEDINGS

SC'95 Table of Contents by Session

Keynote Address

William A. Wulf
And Now for Some *Really* Super Computing

Invited Speakers

Cherri Pancake
The Emperor Has No Clothes: What HPC Users Need to Say and HPC Vendors Need to Hear

Pavel Curtis
Network Places: Concepts, Experiences, and Plans

Sam Milosevich
Thriving on Information Anxiety: A Survival Guide to the Knowledge-Value Revolution

John Schill
Joint Task Force Advanced Technology Demonstration (JTF ATD)

Richard M. Hodur
Numerical Weather Prediction and the America's Cup

Shuichi Iwata
Virtual Engineering: Challenges into Handy Engineering from Advanced and Super Technologies

Jack Donegan
How Many Miles per Gallon Does Your Computer Get?

Seymour Goodman
An Examination of High-Performance Computing Export Control Policy in the 1990s

Papers

ATM in High Performance Networks
 Lead-in Speaker: Jonathon Smith
Multicast Virtual Topologies for Collective Communication in MPCs and ATM Clusters by Y. Huang,
 C. C. Huang, P. K. McKinley
Model and Call Admission Control for Distributed Applications with Correlated Bursty Traffic by

SC'95 Table of Contents by Session
 Jose Roberto Fernandez, Matt W. Mutka

1/16/96 10:59 AM

Applied Algorithms

Lead-in Speaker: TBD

Surface Fitting Using GCV Smoothing Splines on Supercomputers by Alan Williams, Kevin Burrage
Efficient Algorithms for Atmospheric Correction of Remotely Sensed Data by Hassan Fallah-Adl,
 Joseph Jaja, Shunlin Liang, Yoram J. Kaufman, John Townshend

Education Papers

The Living Textbook and the K-12 Classroom of the Future by Kim Mills, Geoffrey Fox, Paul
 Coddington, Barbara Mihalas, Marek Podgorny, Barbara Shelly, Steven Bossert
Pittsburgh Supercomputing Center High School Initiative in Computational Science Report on Findings
School Years: 1991-92, 1992-93, 1993-4 by Casey Porto
Developing Computational Science Curricula: The EarthVision Experience by Ralph K. Coppola, Eva
 Erdosne Toth
The Use of Cellular Automata in the Classroom by H. Albert Lilly

Innovative Educational Issues

Mobile Robots Teach Machine-Level Programming by Patricia J. Teller, Ted Dunning
A Web Interface to Parallel Program Source Code Archetypes by Juan Villacis, Dennis Gannon
HPC Undergraduate Curriculum Development at SDSU using SDSC Resources by Kris Stewart

Data Mining

Computational Methods for Intelligent Information Access by Michael W. Berry, Susan T. Dumais,
 Todd A. Letsche
High-Performance Computing Approaches for Using the WWW to Access a Large-Scale
Environmental Dataset Repository by Bahram Nassersharif, Richard Marciano, Sui-ky Ling, Eugene
 Ho, Curt Edmonds
Distributed Information Management in the National HPCC Software Exchange by Shirley Browne,
 Jack Dongarra, Geoffrey C. Fox, Ken Hawick, Ken Kennedy, Rick Stevens, Robert Olson, Tom Rowan

Applications: Biochemistry

Computational Approach to the Statistical Mechanics of Protein Folding by Ming-Hong Hao, Harold
 A. Scheraga
Surveying Molecular Interactions with DOT by Lynn F. Ten Eyck, Jeffrey Mandell, Victoria A. Roberts,
 Michael E. Pique
I/O Limitations in Parallel Molecular Dynamics by Terry W. Clark, L. Ridgway Scott, Stanislaw
 Wloked, J. Andrew McCammon

Applications: Biology

Microparallelism and High-Performance Protein Matching by Bowen Alpern, Larry Carter, Kang Su
 Gatlin
Parallelizing the Phylogeny Problem by Jeff A. Jones, Katherine A. Yelick
MONSTER - the Ghost in the Connection Machine: Modularity of Neural Systems in Theoretical
Evolutionary Research by Nigel Snode, Terry Bossomaier

Partitioning Algorithms

PMRSB: Parallel Multilevel Recursive Spectral Bisection by Stephen T. Barnard
A Multi-Level Algorithm For Partitioning Graphs by Bruce Hendrickson, Robert Leland
Analysis of Multilevel Graph Partitioning by George Karypis, Vipin Kumar

Performance I

A Structured Approach to Instrumentation System Development and Evaluation by Abdul Waheed,
 Diane T. Rover
Automated Performance Prediction of Message-Passing Parallel Programs by Robert J. Block, Sekhar
 Sarukkai, Pankaj Mehra
Towards Modeling the Performance of a Fast Connected Components Algorithm on Parallel Machines
 by Steven S. Lumetta, Arvind Krishnamurthy, David E. Culler

Applications: Engineering

A Case Study in Parallel Scientific Computing: The Boundary Element Method on a
Distributed-Memory Multicomputer by Ramesh Natarajan, Dilip Krishnaswamy
Parallel Implementations of the Power System Transient Stability Problem on Clusters of Workstations

SC'95 Table of Contents by Session

1/16/96 10:59 AM

by Monika ten Bruggencate, Suresh Chalasani

Parallel Processing of Spaceborne Imaging Radar Data by Craig Miller, David G. Payne, Thanh N. Phung, Herb Siegel, Roy Williams

Parallel Software

A Parallel Software Infrastructure for Structured Adaptive Mesh Methods by Scott R. Kohn, Scott B. Baden

Message Passing Versus Distributed Shared Memory on Networks of Workstations by Honghui Lu, Sandhya Dwarkadas, Alan L. Cox, Willy Zwaenepoel

Storm Watch: A Tool for Visualizing Memory System Protocols by Trishul M. Chilimbi, Thomas Ball, Stephen G. Eick, James R. Larus

Novel Execution Models

Efficient Support of Location Transparency in Concurrent Object-Oriented Programming Languages by WooYoung Kim, Gul Agha

Compiling and Optimizing for Decoupled Architectures by Nigel Topham, Alasdair Rawsthorne, Callum McLean, Muriel Mewissen, Peter Bird

A Hybrid Execution Model for Fine-Grained Languages on Distributed Memory Multicomputers by John Plevyak, Vijay Karamcheti, Xingbin Zhang, Andrew A. Chien

Applications: Physics

Parallel Linear General Relativity and CMB Anisotropies by Paul Bode, Edmund Bertschinger

Balancing Processor Loads and Exploiting Data Locality in N-Body Simulations by Ioana Banicescu, Susan Flynn Hummel

Lattice QCD on the IBM Scalable POWERParallel Systems SP2 by C. Bernard, C. DeTar, S. Gottlieb, U.M. Heller, J. Hetrick, N. Ishizuka, L. Karkkainen, S. R. Lantz, K. Rummukainen, R. Sugar, D. Toussaint, M. Wingate

Gigabit Testbed Experiences

Lead-in Speaker: Darleen Fisher

Distributing a Chemical Process Optimization Application Over a Gigabit Network by Robert L. Clay, Peter A. Steenkiste

Wide-Area Gigabit Networking: Los Alamos HIPPI-SONET Gateway by Wallace B. St. John, David H. DuBois

Compilers I

Symbolic Array Dataflow Analysis for Array Privatization and Program Parallelization by Junjie Gu, Zhiyuan Li, Gyungho Lee

Interprocedural Compilation of Irregular Applications for Distributed Memory Machines by Gagan Agrawal, Joel Saltz

Detecting Coarse - Grain Parallelism Using an Interprocedural Parallelizing Compiler by Mary W. Hall, Saman P. Amarasinghe, Brian R. Murphy, Shih-Wei Liao, Monica S. Lam

Parallel Tools

An Integrated Compilation and Performance Analysis Environment for Data Parallel Programs by Vikram S. Adve, John Mellor-Crummey, Mark Anderson, Ken Kennedy, Jhy-Chun Wang, Daniel A. Reed

Relative Debugging and its Application to the Development of Large Numerical Models by David Abramson, Ian Foster, John Michalakes, Rok Sosic

SCIRun: A Scientific Programming Environment for Computational Steering by Steven G. Parker, Christopher R. Johnson

Performance II

Lead-in Speaker: David Bailey

A Performance Evaluation of the Convex SPP-1000 Scalable Shared Memory Parallel Computer by Thomas Sterling, Daniel Savarese, Peter MacNeice, Kevin Olson, Clark Mobarry, Bruce Fryxell, Phillip Merkey

Predicting Application Behavior in Large Scale Shared-memory Multiprocessors by Karim Harzallah, Kenneth C. Sevcik

Issues in Scheduling, Networking, and Computer Misuse Detection

High-Performance Incremental Scheduling on Massively Parallel Computers - A Global Approach by Min-You Wu, Wei Shu

SC'95 Table of Contents by Session

1/16/96 10:59 AM

High Performance Messaging on Workstations: Illinois Fast Messages (FM) for Myrinet by Scott Pakin, Mario Lauria, Andrew Chien
UNICORN: Misuse Detection for UNICOS(TM) by Gary G. Christoph, Kathleen A. Jackson, Michael C. Neuman, Christine L. B. Siciliano, Dennis D. Simmonds, Cathy A. Stallings, Joseph L. Thompson

Input/Output for High Performance Systems

Server-Directed Collective I/O in Panda by K. E. Seamons, Y. Chen, P. Jones, J. Jozwiak, M. Winslett
Gigabit I/O for Distributed-Memory Machines: Architecture and Applications by Michael Hemy, Peter Steenkiste
Input/Output Characteristics of Scalable Parallel Applications by Phyllis E. Crandall, Ruth A. Aydt, Andrew A. Chien, Daniel A. Reed

Architecture

Chair: Steve Oberlin

The Benefits of Clustering in Shared Address Space Multiprocessors: An Applications-Driven Investigation by Andrew Erlichson, Basem A. Nayfeh, Jaswinder P. Singh, Kunle Olukotun
Lazy Release Consistency for Hardware-Coherent Multiprocessors by Leonidas I. Kontothanassis, Michael L. Scott, Ricardo Bianchini
Architectural Mechanisms for Explicit Communication in Shared Memory Multiprocessors by Umakishore Ramachandran, Gautam Shah, Anand Sivasubramaniam, Aman Singla, Ivan Yanasak

Gordon Bell Prize Finalists

Astrophysical N-body simulations on the GRAPE-4 Special-Purpose Computer by Junichiro Makino, Makoto Taiji
Price and Performance of Simulating Wind Instruments by Panayotis Skordos
Quantum Chromodynamics Simulation on NWT by M. Yoshida, A. Nakamura, M. Fukuda, T. Nakamura, S. Hioki

Applications: Fluid Dynamics

Chair: Thomas Mautner

A Parallel Incompressible Flow Solver Package with a Parallel Multigrid Elliptic Kernel by John Z. Lou, Robert D. Ferraro
Large Eddy Simulation of a Spatially-Developing Boundary Layer by Xiaohua Wu, Kyle D. Squires, Thomas S. Lund
Parallelizing Navier-Stokes Computations on a Variety of Architectural Platforms by D. N. Jayasimha, M. E. Hayder, S. K. Pillay

Compilers II

Chair: Dirk Grunwald

Communication Optimizations for Parallel Computing Using Data Access Information by Martin C. Rinard
Index Array Flattening Through Program Transformation by Raja Das, Paul Havlak, Joel Saltz, Ken Kennedy
An HPF Compiler for the IBM SP2 by Manish Gupta, Sam Midkiff, Edith Schonberg, Ven Seshadri, David Shields, Ko-Yang Wang, Wai-Mee Ching, Ton Ngo

Performance III

Chair: Allen Malony

Lead-in Speaker: Joan Francioni

The Synergetic Effect of Compiler, Architecture, and Manual Optimizations on the Performance of CFD On Multiprocessors by Masayuki Kuba, Constantine D. Polychronopoulos, Kyle Gallivan
Parallel Retrograde Analysis on a Distributed System by Henri Bal, Victor Allis

Matrix Computations

Chair: Lori Freitag

Lead-in Speaker: Michael Heath

Parallel Algorithms for Forward and Back Substitution in Direct Solution of Sparse Linear Systems by Anshul Gupta, Vipin Kumar
Parallel Matrix-Vector Product Using Approximate Hierarchical Methods by Ananth Grama, Vipin Kumar, Ahmed Sameh

Compilers III

Chair: Margaret Simmons

SC'95 Table of Contents by Session

1/16/96 10:59 AM

Automatic Data Layout for High Performance Fortran by Ken Kennedy, Ulrich Kremer
Controlling Application Grain Size on a Network of Workstations by Bruce S. Siegel, Peter A. Steenkiste
A Novel Approach Towards Automatic Data Distribution by Jordi Garcia, Eduard Ayguade, Jesus Labarta

Applications: Ocean and Atmospheric Modeling

Chair: Robert Chervin

Implementation and Performance of a Grand Challenge 3d Quasi-Geostrophic Multi-Grid code on the Cray T3D and IBM SP2 by Clive F. Baillie, James C. McWilliams, Jeffrey B. Weiss, Irad Yavneh
Architecture-Adaptable Finite Element Modelling: A Case Study using an Ocean Circulation Simulation by Santhosh Kumaran, Robert N. Miller, Michael J. Quinn
Performance of a Parallel Global Atmospheric Chemical Tracer Model by James Demmel, Sharon Smith

Panels

University Education

Where is the Supercomputer Software Revolution?

Moderator: Dennis Gannon

Panellists: Larry Smarr, Vince Schuster

Mayors' Panel

Moderator: Jack Donegan

Community Networking I - Applications

Moderator: John Ziebarth

Community Networking II - Technology

Moderator: John Ziebarth

Goldilocks and the Three Bears Confront the Future of Supercomputing

Moderator: Robert Borchers

Panelists: Tom Anderson, Burton Smith, Steven Wallach

Embedded Applications for High Performance Computing

Moderator: Craig Lund

Panelists: Dr. José Muñoz and others TBD

Information Superhighway or Road to Ruin?

Moderator: Becky Bace

Panelists: Gary Christoph, Tsutomu Shimomura, Gene Spafford

Are Tereflops Commercial Flops?

Moderator: Norris Parker Smith

Panelists: Forest Baskett, Irving Wladawski

Workshops

HPF: A User's Perspective by Brian T. Smith

EUROPORT Activities by Francis Wray

Careers for Women in Computer Science and Engineering

Object Oriented Parallel Programming by Dennis Gannon

System Software and Tools for High-Performance Computing Environments by Paul Messina, James C. T. Pool, Thomas Sterling

Research Issues in Scalable I/O by James C. T. Pool

＜技術論文各セッションの要約＞

1. 応用アルゴリズム

以下の3件について発表がなされた。

- (1) Sheared-Memory Emulation for Billion-Atom Molecular Dynamics Simulations
Charles Romine, Martine Marietta Energy Systems, Inc.
- (2) Surface Fitting Using GCV Smoothing Splines on Supercomputers
Alan Williams, Kevin Burrage, University of Queensland, Australia
- (3) Efficient Algorithms for Atmospheric Correction of Remotely Sensed Data
Hassan Fallah-Adl, Joseph JaJa, et al., UMIACS, University of Maryland

(1)は数億個の原子について、数ミリ秒間の分子動力学計算を効率よく行う手法について述べたものである。地中内水理解析／輸送問題、大気拡散問題の解析に用いられている。マシンはインテル社のParagon 1024PEを用い、高いスケーラビリティを達成している。

(2)はオーストラリア各地の離散化された気象データから、GCV(Generalised Cross Validation smoothing algorithm)と呼ばれるアルゴリズムを用い、全土に対しスムージングされたデータを得る方法について述べたものである。本手法を用いて1000～2000程度の観測点データからSmoothing SurfaceをCRAY-C90, Cray-YMP, Intel Paragon, Cray-T3Dなど種々のプラットフォーム上で計算し、そのパフォーマンスを比較した。GCVのソルバーのCG法を使用した場合、Cray-T3Dは比較的良いスケーラビリティを示すのに対し、IBM-SP2やIntel Paragonはあまり良くない。これはCray-T3Dが高いlatency、低いバンド幅を持つためであると考えられる。

(3)はランドサット衛星から送られてくる画像に対し、「大気補正」と呼ばれる画像処理を高速に行う手法について述べたものである。具体的には送られてくる画像の雲などで覆われている部分に補正を加え、地表面の形状を正しく求める手法である。扱うデータ量は膨大となるため、これを領域分割法により並列化を行い高速に処理している。

2. 生化学

以下の3件の発表がなされた。

- (1) Computational Approach to the Statistical Mechanics of Protein Folding
Ming-Hong Hao, Harold A. Scheraga, Cornell University
- (2) Surveying Molecular Interactions with DOT
Lynn F. Ten Eyck, Jeffrey Mandell, San Diego Supercomputer Center
- (3) I/O Limitations in Parallel Molecular Dynamics
Terry W. Clark, L. Ridgway Scott, Stanislaw Wloked, University of Houston

(1)は蛋白質foldingの問題に、並列計算機の使用を前提として新規アプローチで取り組ん

だものである。蛋白質は数万～数十万個の原子からなる高分子であり、そのとり得る立体構造は無限といてよいほど存在する。しかし実際には、その立体構造は"native structure"と呼ばれる特定の構造に落ちつくことが実験的にも知られている。蛋白質foldingの問題とは、ある蛋白質の原子配列が一次的に決まった場合に、その立体構造である"native structure"を予測することである。本発表は、従来のモンテカルロ法による立体構造の予測に代えてESMC(Entropy Sampling Monte Carlo)法を用いて効率化を図ったものである。

(2)は、2つの生体高分子が結合するような問題を計算する場合に課題となってくる、2分子間あるいは他の荷電分子とのelectrostatic potential energyを効率よく計算する手法(DOT)について述べたものである。DOTはmaster/slaveモデルを用いて並列化が行っている。計算はDec 3000/4000システムを9ノード結合したもの、およびIntel Paragon XP/S上で実行され、いづれも従来の手法より高いスケーラビリティを示した。

3. 生物学

このセッションにおける発表は次の3件である。

(1) Microparallelism and High-Performance Protein Matching

Bowen Alpern, IBM Watson Research Center

(2) Parallelizing the Phylogeny Problem

Jeff A. Jones, Katherine A. Yelick, HyperParallel Inc.

(3) MONSTER - the Ghost in the Connection Machine: Modularity of Neural Systems in Theoretical Evolutionary Research

Nigel Snoad, Terry Bossomaier, RSISE, Australian National University

(1)は蛋白質やDNA配列のstring-matchingに用いられるSmith-Waterman法を効率的に並列化する手法およびその解析結果について述べたものである。

(2)はphylogeny problemとして知られている個体進化の系統樹作成に関わる手法の並列化に関する発表である。

(3)は遺伝的アルゴリズム (GA)、ニューラルネットワークという生物学にモデルの基盤を持つ2つのアルゴリズムを組み合わせ並列化したコードMONSTERについての発表である。具体的には、ニューラルネットワークの学習部分にGAを用い、それぞれのPEごとに作られたネットワークを足し併せる手法を用いている。この手法をパターン認識問題に適用しその有効性が確かめられた。

4. エンジニアリング

本セッションは、並列機による実問題に対する取り組みを示したものであり、応用例に対する発表が主となっている。

(1) A Case Study in Parallel Scientific Computing: The Boundary Element Method on a Distributed-Memory Multicomputer

Ramesh Natarajan, IBM Watson Research Center

(2) Parallel Implementations of the Power System Transient Stability Problem on Clusters of Workstations

Monika ten Bruggencate, Suresh Chalasani, University of Wisconsin

(3) Parallel Processing of Spaceborne Imaging Radar Data

Craig Miller, NASA Jet Propulsion Laboratory; David G. Payne, Thanh N. Phung, Intel Corp.

(1)は境界要素法を用い、3次元のラプラス、ヘルムホルツ方程式をIBM-SP2上で効率的に解く手法について述べたものである。境界要素法では問題のサイズが大きくなると、メモリは2乗のオーダーで、ソルバーは3乗のオーダーで必要となってくる。このため本研究では、メモリは係数行列を分割することにより、ソルバーはプログラムの並列化により対応した。プログラムの並列化はIBM Power Visualization System上でHPCベンチマーク用に開発された手法を基に行われた。32PEで約18倍の処理速度向上が図れた。

(2)は電力供給網における負荷の状態を動的に解析するための手法に関する研究である。この問題は最終的には大規模疎行列を解く問題に帰着される。前進後退代入法とW-行列法と呼ばれる2種類の手法を用いて効率的に解かれている。

(3)は、SIR-C/X-SARと呼ばれる宇宙空間に設置されたレーダーで観測された地球環境に関する画像データを並列処理する研究である。基本的には領域分割によりPEごとに画像処理される。また、Intel PFS(Parallel File System)を用い、データを読み込むPE、各PEにデータを分配するPE、処理するPEとの間をデータがオーバーラップしながら渡されていく。I/O、通信、FFT計算、その他の計算にかかる処理時間に偏りがみられず、ほぼリアルタイムに近い処理が施せるようになった。

5. 流体力学

本セッションでは以下の3件の発表がなされた(参考に論文を添付する)。

(1) A Parallel Incompressible Flow Solver Package with a Parallel Multigrid Elliptic Kernel

John Z. Lou, Robert D. Ferraro, California Institute of Technology

(2) Large Eddy Simulation of a Spatially-Developing Boundary Layer

Xiaohua Wu, Kyle D. Squires, University of Vermont

(3) Parallelizing Navier-Stokes Computations on a Variety of Architectural Platforms

D. N. Jayasimha, Ohio State Univ.; M. E. Hayder, S. K. Pillay, NASA

(1)は差分法による非圧縮性流れ解法プログラムの並列化について述べたものである。数値実験により並列ソルバーの有用性が確認されるとともに、PVM等をサポートするシステムに容易に移植できることが示された。

(2)はLESによる空間の乱流発達の発達をシミュレーションするために、乱流入口条件の決定方法を提案したものである。これにより、計算的にコストのかかる層流・乱流遷移領域を計算することなく、空間の発達乱流境界層の基本特性を再現することができた。

(3)は、圧縮性ナビエ・ストークス方程式を用いてジェット問題を解くアプリケーション

ンを、クラスター状に結合したワークステーション、共有メモリ型並列計算機（Cray YMP）、分散メモリ型並列計算機（IBM-SP, Cray-T3D）といった異なるプラットフォームに適用し、通信性能やスケータビリティ等の比較を行ったものである。

A Parallel Incompressible Flow Solver Package with a Parallel Multigrid Elliptic Kernel

John Z. Lou⁺ and Robert D. Ferraro⁺⁺

Abstract

A parallel time-dependent incompressible flow solver and a parallel multigrid elliptic kernel are described. The flow solver is based on a second-order projection method applied to a staggered finite-difference grid. The multigrid algorithms implemented in the elliptic kernel, which is needed by the flow solver, are V-cycle and full V-cycle schemes. A grid-partition strategy is used in the parallel implementations of both the flow solver and the multigrid elliptic kernel on all fine and coarse grids. Numerical experiments and parallel performance tests show the parallel solver package is numerically stable, physically robust and computationally efficient. Both the multigrid elliptic kernel and the flow solver scale very well to a large number of processors on the Intel Paragon and the Cray T3D for computations with moderate granularity. The solver package has been carefully designed and coded so that it can be easily adapted to solving a variety of interesting two and three-dimensional flow problems. The solver package is portable to parallel systems that support MPI, PVM and Intel NX for interprocessor communications.

1. Introduction

The objective of this work is to develop a parallel and scalable incompressible flow solver package which can be used for solving a variety of practical and challenging incompressible flow problems arising from physics and engineering applications. A few examples are convective turbulence modeling in astrophysics, thermally driven flows in cooling systems and combustion process modeling. A Navier-Stokes algorithm for successfully solving these complicated, non-smooth flow problems must be numerically stable, physically robust and computationally efficient. Results from numerical experiments in [2], [3] and [4] indicate that a second-order projection method proposed in [2] is a promising candidate for

⁺lou@acadia.jpl.nasa.gov, Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA 91109

⁺⁺ferraro@zion.jpl.nasa.gov, Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA 91109

simulations of complex incompressible flows.

Our task is to develop an efficient, flexible and portable parallel flow solver package for multiple applications. In terms of efficiency, we want the solver to have high numerical efficiency as well as parallel computing efficiency, which is the reason to use a parallel multigrid elliptic kernel as a convergence accelerator for the parallel flow solver. Flexibility and portability have been emphasized throughout our design and implementation of the solver package. We want to develop the solver package so that it can be used either as a stand-alone flow solver for several types of flow problems or as a flow solver template which can be modified or expanded by the user for a specific application. A reusable or template partial differential equation (PDE) solver, in our view, is a PDE solver package that can be adapted or expanded to solving a variety of problems using different (component) numerical schemes as needed without a major rewriting of the solver code.

A basic assumption in our solver package is the use of finite-difference methods on a rectangular grid or on a composite grids with each of its component a rectangular grid. The use of rectangular grids has several advantages: (1) finite-difference is easy to implement, and for many applications, stable and robust finite-difference methods already exist while the use of a finite-element type scheme may not be desirable for some applications due to physical and numerical considerations; (2) multigrid is easy to apply; (3) parallel implementations are easier than on unstructured grids. Many practical problems are, however, defined in irregular domains. One way to extend our solver package to problems in an irregular domain is to construct a mapping between the irregular domain and a rectangular region. For a variety of non-rectangular domains, such mappings can indeed be constructed (for more detail, see [10]).

A projection method for solving incompressible Navier-Stokes equations was first described in a paper by Chorin [7]. Bell et. al [2] [3] extended the method to second-order accuracy in both time and space, and used a Godunov procedure combined with an upwind scheme in the discretization of the convection term for improved numerical stability. The projection method is a type of operator-splitting method which separates the solutions of velocity and pressure fields with an iterative procedure. In particular, at each time step, the momentum equations are solved first for an intermediate velocity field without the knowledge of a correct pressure field and therefore no incompressibility condition is enforced. The intermediate velocity field is then "corrected" by a projection step in which we solve a pressure equation and then use the computed

pressure to produce a divergence-free velocity field. Our projection step, which is based on a pressure equation derived in [1] and makes use of the highly efficient elliptic multigrid kernel we developed, is mathematically equivalent to but algorithmically different from the projection step described in [2]. In actual flow simulations, this prediction-correction type procedure is usually repeated a few times (1 or 2 iterations seem to be enough from our experiments) until reasonably good velocity and pressure fields have been reached for that time step. In each time step for computing an N -dimensional ($N = 2$ or 3) viscous flow problem, we need to solve $m \times N$ Helmholtz equations for the velocity field and m Poisson equations for the pressure field, where m is the number of iterations performed at each time step. A fast multigrid elliptic solver is thus very useful to improve the computational performance of the flow solver. The multigrid kernel was designed to be a general-purpose elliptic solver. It can solve N -dimensional ($N \leq 3$) problems on vertex-centered, cell-centered and staggered grids, and it can deal with a variety of different boundary conditions as well.

Since the solver package is implemented on rectangular grids, a natural parallel implementation strategy is grid-partitioning: the global computational grid is partitioned and distributed to a logical network of processors; message exchanges are performed for grid points lying on "partition boundary-layers" (whose thickness is usually dictated by the numerical schemes used) to ensure a correct implementation of the sequential numerical algorithms on the global computational grid. In our implementation of the parallel multigrid V-cycle and full V-cycle schemes, we apply this grid-partition to all coarse grids as well. This means on some very coarse grid, only a subset of allocated processors will contain at least one grid point on that grid and they are therefore "active" on that grid, whereas processors which do not contain any grid point will be idle when processing that grid. The appearance of idle processors certainly introduces some complexity for a parallel implementation. For example, the logical processor mesh on which the original computational grid is partitioned can not be used for communications on those coarse grids for which idle processors appear. Depending on the type of finite-difference grid and coarsening scheme, one may also need to consider, on those coarse grids, how to correctly apply boundary conditions in "boundary processors" which contain at least one grid point next to the boundary of the global grid, since boundary processors may change from one grid to another. Grid-partition on all coarse grids is certainly not the only possibility for parallel multigrid. Another approach, e.g., is to duplicate some of the global

coarse grids in every processor allocated, so that processing on those coarse grids can be done without further interprocessor communication, but this coarse-grid-duplication approach involves quite some global communication for grid duplication and it needs some extra storage for global coarse grids. These requirements may severely affect the scalability of the solver when running on a large number of processors. One may also stop further grid coarsening at the coarsest grid for which no idle processor appears yet, and solve the coarse grid problem by some direct or iterative methods. But the cost in solving the coarse grid problem with those methods is not competitive compared to further grid coarsening. Although it seems no approach is perfect for implementing a parallel classical multigrid cycle [6] [9], we do believe the use of grid-partition at all grid levels is an appropriate approach for implementing a general-purpose parallel multigrid solver. The degradation of parallel efficiency due to the idle processors on some coarse grids has been discussed in many papers (e.g. [6] [9] [11]). The performance measurements from our parallel implementations indicate our multigrid solver scales quite well on a 512-node Intel Paragon and on a 256-node Cray T3D for both 2D and 3D problems with moderate sizes of local finest grids. In fact, the percentage of time spent on those coarse grids is insignificant compared to the total computation time. A similar observation was also made in [9]. As shown by a simple asymptotic analysis in [8], the parallel efficiency of multigrid schemes with the grid-partition approach is not qualitatively different from that of a single grid scheme.

The rest of the paper is organized as follows: Section 2 presents numerical algorithms for the multigrid kernel and the second-order projection method for the incompressible flow solver; in Section 3, discussions are made on issues related to the parallel implementations of the solver package; numerical results and parallel performances from the implemented parallel solvers are shown in Section 4.; Section 5 gives some of our observations and conclusions.

2. The Numerical Methods

A. The Multigrid Algorithms

The multigrid schemes implemented are the so-called V-cycle and full V-cycle schemes for solving elliptic PDEs, discussed in some detail in [5] and [9]. The full V-cycle scheme is a generalization of the V-cycle scheme which first restricts the residual vector to the coarsest grid and then performs a few smaller V-cycle schemes

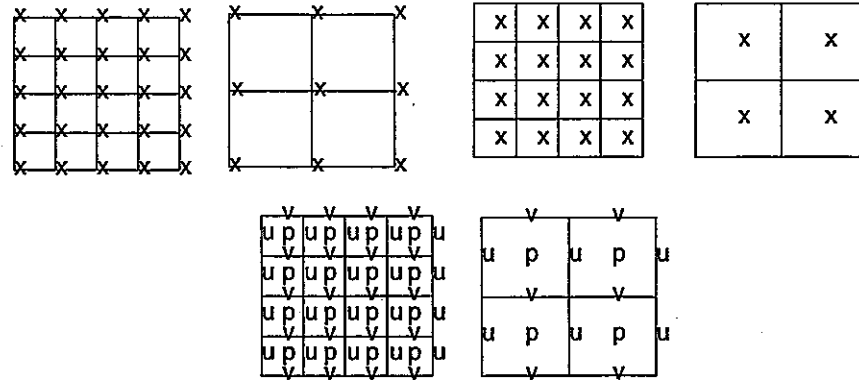


Figure 1: Coarsening of three types of grids: vertex-centered (top-left), cell-centered (top-right) and staggered (bottom).

on all coarse grids, followed by a complete V-cycle scheme on all grids. The full V-cycle scheme often offers a better numerical efficiency than the V-cycle scheme by using a much better initial guess of the solution in the final V-cycle. The parallel efficiency for the full V-cycle scheme, however, is poorer than the V-cycle scheme because it does more processing on coarse grids.

A typical multigrid cycle consists of three main components: relax on a given grid, restrict the resulting residual to a coarse grid, and interpolate a correction back to a fine grid. Our multigrid solver can handle several different types of finite-difference grids commonly used in numerical computations. Figure 1 shows how coarse grids are derived from fine grids for vertex-centered, cell-centered and staggered grids. Although the main steps in a V-cycle are the same for all these grids, restriction and interpolation operators can have different forms on different grids. On a vertex-centered grid we use a full-weighting stencil (9-point averaging on a 2D grid) to make the V-cycle scheme converge well when a pointwise red-black Gauss-Seidal (GS) smoother is used; whereas on a cell-centered grid, a nearest-neighbor stencil (4-point on a 2D grid) can be used with the pointwise red-black GS smoother to achieve a good convergence rate. We also point out that, on a vertex-centered grid, the use of the nearest-neighbor restriction stencil with the point-wise GS smoother does not even result in convergence on our test problems, but the use of a Jacobi smoother with the nearest-neighbor restriction stencil results in convergence but with a slower rate. The operator for transferring from a coarse grid to a fine grid is basically bilinear interpolation for all grids. Since fine and coarse grid points do not overlap on cell-centered and staggered grids, one

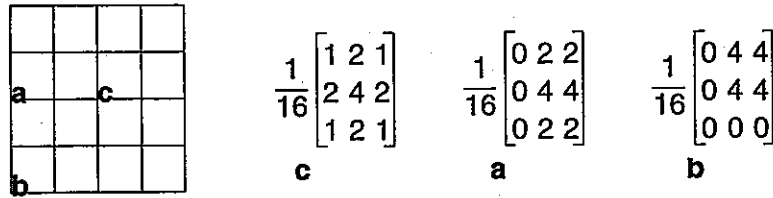


Figure 2: Restriction stencils for interior point c and boundary points a and b.

needs to set the values for grid points at the boundary of coarse grids before a bilinear interpolation operator can be applied. More details on the constructions of restriction and interpolation operators for different types of grids can be found in [14].

Our multigrid solver can solve Dirichlet and Neumann problems for the grids depicted in Figure 1. A periodic boundary condition is also implemented for a special case used in the NS flow solver. The Dirichlet and Neumann boundary conditions are applied only to the original (finest) grid; a homogeneous (zero) boundary condition is used on all coarse grids since residual equations are solved there. In the case of a Neumann boundary condition, where the unknowns are solved on all grid points including those on the grid boundary, restriction stencils are not well-defined for boundary grid points. Take for example the vertex-centered grid in Figure 2, where a 9-point full weighting stencil is used for restriction. This can be done naturally for the interior point c. For boundary points a and b, however, only a subset of the neighboring points are within the grid and therefore weighting stencils on those points still need to be defined in some way. On the other hand, it is reasonable to have the following discrete integral condition satisfied between a pair of coarse and fine grids:

$$\sum_{i,j} U_{i,j} \times A_{i,j} = \sum_{i,j} u_{i,j} \times a_{i,j}, \quad (1)$$

where $U_{i,j}$ and $u_{i,j}$ are solutions on coarse and fine grids, and $A_{i,j}$ and $a_{i,j}$ are areas of grid cells on coarse and fine grids, respectively. Restriction stencils for interior and boundary points that satisfy equation (1) are given in Figure 2.

When solving a Poisson equation with a Neumann boundary condition, the solution is determined up to a constant. We use the following strategy to make sure the application of multigrid cycles converges to a fixed solution: after every relaxation on each grid, we perform a normalization step by adding a constant to the computed solution so that its value at a fixed point (we pick the point located at the center of the grid) is zero. Our numerical tests show

this simple step results in a good convergence rate for Neumann problems.

B. The Second-Order Projection Method

We now give a brief description of the second-order projection method for solving the incompressible Navier-Stokes equations in a dimensionless form

$$\begin{aligned}\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} &= -\nabla p + \text{Re}^{-1} \Delta \mathbf{u} \\ \nabla \cdot \mathbf{u} &= 0\end{aligned}\quad (2)$$

where $\mathbf{u} \in \mathbb{R}^n$ ($n = 2$ or 3) is the velocity field, $p \in \mathbb{R}$ is the pressure field and Re is the Reynolds number. A typical problem is to find $\mathbf{u}$ and p satisfying (2) in a domain Ω for a given initial velocity field $\mathbf{u}_0$ in Ω and a velocity boundary condition $\mathbf{u}_b$ on $\partial\Omega$. The projection method for solving equations (2) is based on the Hodge decomposition which states that any vector field $\mathbf{u}$ can be uniquely decomposed into a sum of $\mathbf{u}_1 + \mathbf{u}_2$ with $\nabla \cdot \mathbf{u}_1 = 0$ and $\mathbf{u}_2 = \nabla \phi$ for some scalar function ϕ . The projection method proceeds as a type of fractional step method by first writing the momentum equation in (2) in an equivalent form

$$\frac{\partial \mathbf{u}}{\partial t} = \mathbf{P} (\text{Re}^{-1} \Delta \mathbf{u} - (\mathbf{u} \cdot \nabla) \mathbf{u}), \quad (3)$$

where $\mathbf{P}$ is an orthogonal projection operator which projects a smooth function onto a divergence-free subspace. Equation (3) can be viewed as the result of applying $\mathbf{P}$ to the momentum equation in (2) which can be rewritten as

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla p = \text{Re}^{-1} \Delta \mathbf{u} - (\mathbf{u} \cdot \nabla) \mathbf{u}. \quad (4)$$

The projection operation removes the pressure gradient in (4) because ∇p is orthogonal to the projection. Thus if we let the right-hand side of (4) be a vector field $\mathbf{V}$, then $\nabla p = (\mathbf{I} - \mathbf{P}) \mathbf{V}$. The second-order projection method in [2] is a modification to the original projection method proposed in [7] to achieve a second-order temporal accuracy and an improved numerical stability for the nonlinear convection. It uses the following temporal discretization on the momentum equation at each half time step $n+1/2$

$$\begin{aligned}\frac{\bar{\mathbf{u}}^{n+1,k} - \mathbf{u}^n}{\Delta t} + [(\mathbf{u} \cdot \nabla) \mathbf{u}]^{n+1/2} &= \\ -\nabla p^{n+1/2,k} + \frac{1}{\text{Re}} \Delta \left(\frac{\bar{\mathbf{u}}^{n+1,k} + \mathbf{u}^n}{2} \right),\end{aligned}\quad (5)$$

where we assume the velocity $\mathbf{u}^n$ is known, and $\mathbf{u}^{n+1,k}$ is an intermediate velocity field that satisfies the same boundary condition as the physical velocity at time step $n+1$. The temporal discretization in (5) is second-order accurate provided that the nonlinear convection term in (5) can be evaluated to the second-order accuracy at the half time step $n+1/2$. The superscripts k in (5) indicate that an iterative process is used for computing the velocity at next time step $\mathbf{u}^{n+1}$, and the pressure at next half time step $p^{n+1/2}$: given a divergence-free field $\mathbf{u}^n$ and the corresponding pressure field $p^{n-1/2}$, we first set $p^{n+1/2,0} = p^{n-1/2}$. For $k \geq 1$, we solve (5) for $\mathbf{u}^{n+1,k}$. Since the correct pressure $p^{n+1/2}$ is not known, the computed $\mathbf{u}^{n+1,k}$ is usually not divergence-free; but $\mathbf{u}^{n+1,k}$ can be used as a guess for $\mathbf{u}^{n+1}$ and it is used to compute $p^{n+1/2,k}$, a new guess for $p^{n+1/2}$, by solving a pressure equation. Once we have a new guess for $p^{n+1/2}$, it is used in (5) to compute $\mathbf{u}^{n+1,k+1}$. This iterative procedure is performed at each time step until $\nabla p^{n+1/2,k} \rightarrow \nabla p^{n+1/2}$ and $\mathbf{u}^{n+1,k} \rightarrow \mathbf{u}^{n+1}$. This iterative process converges because it can be shown that the mapping of errors from state k to state $k+1$ is contractive [2]. In practice, we found 1 to 2 iterations would be enough to get a satisfactory convergence.

The convection term $(\mathbf{u} \cdot \nabla) \mathbf{u}$ is evaluated at the half time step $n+1/2$, using only the velocity $\mathbf{u}^n$ and the pressure $p^{n-1/2}$. On the staggered grid shown in Figure 1, the pressure p is defined at cell centers, horizontal velocity u and vertical velocity v are defined at cell edges. Let us denote cell (i,j) as the cell whose center is located at $(i - 1/2)\Delta x, (j - 1/2)\Delta y$ for $i = 1 \dots I$ and $j = 1 \dots J$. $(\mathbf{u} \cdot \nabla) \mathbf{u}$ is then evaluated at $i, j - 1/2$ for u component and $i - 1/2, j$ for v component. The discretization for u component, for example, has the following form

$$[(\mathbf{u} \cdot \nabla) \mathbf{u}]_u = \frac{u_{i-1/2,j-1/2} + u_{i+1/2,j-1/2}}{2} \left(\frac{u_{i-1/2,j-1/2} - u_{i+1/2,j-1/2}}{\Delta x} \right) + \frac{v_{i,j-1} + v_{i,j}}{2} \left(\frac{u_{i,j} - u_{i,j-1}}{\Delta y} \right)$$

where $u_{i \pm 1/2, j \pm 1/2}$ are velocities at cell centers, $u_{i,j}$ and $v_{i,j}$ are velocities at cell corners and all velocities are assumed to be at time $n+1/2$. Since $\mathbf{u}^n$ is the only velocity available at the start of computations for time step $n+1$ and velocity values are not defined at cell centers and cell corners, we use Taylor expansions of second-order accuracy in both time and space, as was done in [2] and [3], to find

velocities at appropriate locations and at the half time step $n+1/2$ for computing the discrete convection term. To improve numerical stability, a Godunov-type procedure combined with an upwind scheme is used in determining velocity values at cell centers and cell corners. To compute the u velocity component at the cell center of cell (i,j) , for example, we first compute

$$\begin{aligned} u^R &= u_{i-1,j-1/2}^n + \frac{\Delta x}{2} u_{x,i-1,j-1/2}^n + \frac{\Delta t}{2} u_{t,i-1,j-1/2}^n \\ u^L &= u_{i,j-1/2}^n - \frac{\Delta x}{2} u_{x,i,j-1/2}^n + \frac{\Delta t}{2} u_{t,i,j-1/2}^n \end{aligned} \quad (6)$$

where the expansions for u^R and u^L are evaluated on the right side of edge $(i-1, j-1/2)$ and on the left side of edge $(i, j-1/2)$, respectively. The choice of $u_{i-1/2,j-1/2}^{n+1/2}$ is then made by the following upwind scheme:

$$u_{i-1/2,j-1/2}^{n+1/2} = \begin{cases} u^R & \text{if } u^L > 0, u^L + u^R > 0 \\ 0 & \text{if } u^L < 0, u^R > 0 \\ u^L & \text{otherwise} \end{cases} \quad (7)$$

The spatial derivatives in (6) are computed by first using a centered differencing and then applying a slope-limiting step to avoid forming new maxima and minima in the velocity field. Temporal derivatives in (6) are computed by using the momentum equation (4). Derivatives at cell corners are computed in a similar way. More details for the constructions of these derivatives are given in [2] and [3].

After evaluation of the convection term, the intermediate velocity $\mathbf{u}^{n+1,k}$ can be found by solving the following Helmholtz equation for each velocity component:

$$\begin{aligned} -\Delta \bar{\mathbf{u}}^{n+1,k} + \frac{2\text{Re}}{\Delta t} \bar{\mathbf{u}}^{n+1,k} = \\ 2\text{Re} \left(-[(\mathbf{u} \cdot \nabla) \mathbf{u}]^{n+1/2} + \frac{1}{\Delta t} \mathbf{u}^n + \Delta \mathbf{u}^n - \nabla p^{n-1/2} \right) \end{aligned} \quad (8)$$

We notice that the matrix resulted from equation (8) becomes more diagonally dominant as the Reynolds number increases for a fixed grid size and a fixed time step, which is fortunate for computing flows with large Reynolds numbers. For Euler (inviscid) flow problems where $\text{Re} = \infty$, $\mathbf{u}^{n+1,k}$ can be computed explicitly from equation (5). Once $\mathbf{u}^{n+1,k}$ is computed, a projection step is performed to find the pressure field $p^{n+1/2,k+1}$ by solving a Poisson equation:

$$\Delta p = R(u^n, u^{n+1}), \quad (9)$$

where $u^{n+1,k}$ is used in place of u^{n+1} in the right-hand side of equation (5). Mathematically, equation (9) is the result of applying a divergence operator to the momentum equations in (2). Since no boundary condition is defined for the pressure field, some special treatments are needed at the boundary grid points in solving equation (9). The details of deriving the pressure equation (9) on a staggered grid with appropriate treatments at boundary grid points for Dirichlet velocity boundary condition is given in [1]. The treatment of pressure boundary condition for periodic velocity boundary condition using a multigrid scheme is discussed in section 4. In computing a viscous flow, the multigrid elliptic solver is used to solve both equations (8) and (9). After the pressure field $p^{n+1,k+1}$ is found, $u^{n+1,k+1}$ can be computed by using equation (5) with $p^{n+1,k+1}$ in place of $p^{n+1,k}$, and this completes one iteration in the computations for the time step $n+1$. u^{n+1} and p^{n+1} are then obtained at the end of the last iteration. The flow of control for our incompressible Navier-Stokes solver is shown in Figure 3.

3. Parallel Implementations

A. Grid Partition and Logical Processor Mesh

The approach we adopted in parallel implementations of the multigrid elliptic solver and the incompressible flow solver is grid-partition. Our goal is to develop parallel solvers that can partition any N -dimensional ($N \leq 3$) rectangular grids and run on any M -dimensional ($M \leq N$) logical processor meshes. For example, Figure 4 shows the partition of a three-dimensional grid and the assignment of the partitioned subgrids to a three-dimensional torus processor mesh. As shown in Figure 5, logical processor meshes in our code are always constructed as toroidal meshes. Toroidal meshes are useful in the construction of nested coarser processor meshes for the multigrid solver and for dealing with problems with periodic boundary condition.

In the multigrid solver, coarse grids and coarse logical processor meshes are constructed automatically and recursively based on information on a given fine grid. All grid storages are allocated dynamically during the grid coarsening process. In particular, for each multigrid level, a local coarse grid is derived from the local fine grid and storages are allocated for the coarse grid. Processors which will get at least one grid point on that coarse grid will be in an

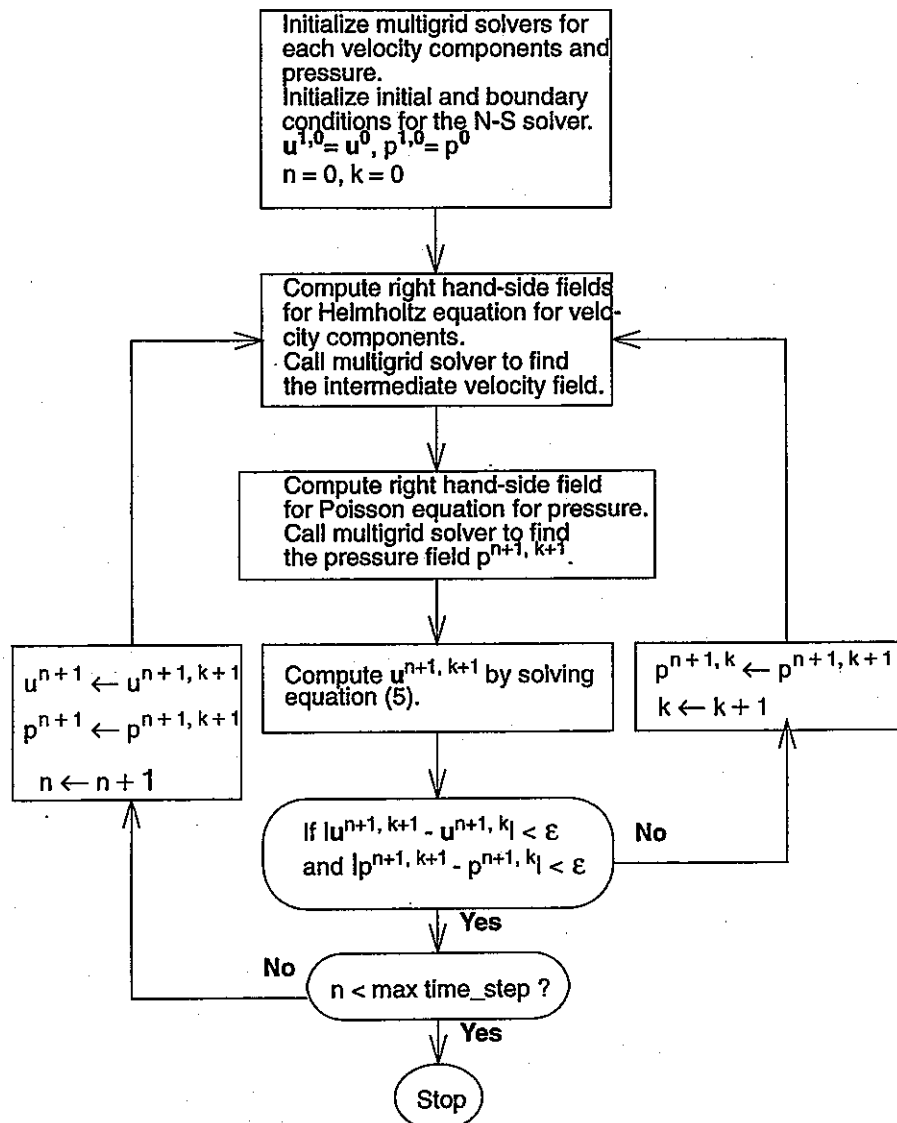


Figure 3: Flow diagram for the Navier-Stokes solver

active state on that grid, otherwise they will be in an idle state on that grid. A flag is then set in each processor for that level depending on the value of the state. A coarse processor mesh for that coarse grid can then be established by communicating the states among processors in the fine processor mesh. This process is repeated recursively until all coarse grids and coarse processor meshes have been constructed. As an illustration, Figure 5 shows a processor mesh and its derived coarse mesh for a problem with a Neumann boundary condition. In our multigrid solver, we put this construction process in an initialization routine which must be called before the first time the multigrid solver itself is called. The cost of running the initialization routine is relatively small when one needs to call the multigrid solver a

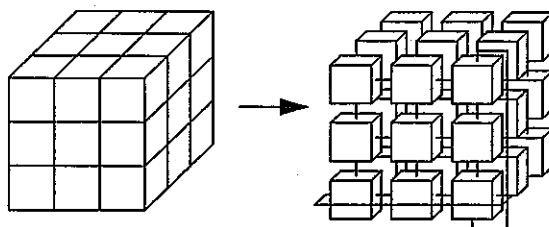


Figure 4: A 3D Grid partition and mapping to a processor mesh. Only two wrap-around connections were shown in the logical processor mesh.

large number of times, as is the case for the Navier-Stokes flow solver. After executing this initialization routine, every processor knows its "role" at each level of the multigrid cycle, and also knows its neighboring processors on that grid level.

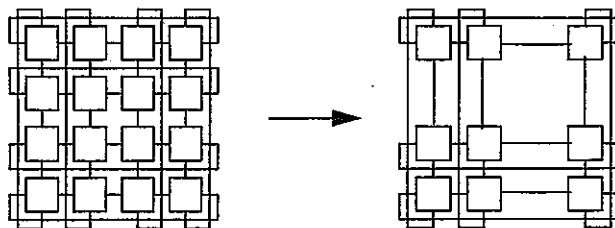


Figure 5: If the left processor mesh contains a 5x5 grid for a Neumann problem on a vertex-centered grid, then the derived coarse processor mesh is the one on the right.

B. Interprocessor Communications

To implement the multigrid scheme and the projection method on a partitioned grid, we need to exchange data which are close to the partition boundaries of each subgrid local to a certain processor. Each processor contains a rectangular subgrid surrounded by some "ghost grid points" which are duplicates of grid points contained in other processors, as shown in Figure 6. The number of ghost points on each side of the subgrid depends on numerical algorithms. For the multigrid elliptic solver using a standard Laplacian stencil, one ghost grid point on each side is needed for the local subgrid at each level, whereas for the second-order projection method, three ghost grid points on each side are needed in computing the nonlinear convection term using Taylor series and upwind schemes. Therefore in the Navier-Stokes flow solver, we allocate storages for three ghost grid points for the fine local grid and one ghost grid point for each coarse grid. For certain operations

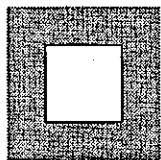


Figure 6: A local subgrid (white area) with surrounding ghost points (shaded area).

in the multigrid scheme (e.g. restriction and interpolation) and for computing the convection term in the projection method, ghost grid points in the diagonal neighbor are also needed, as shown in Figure 7. Since processors P_i and P_j in Figure 7 are not nearest neighbors, direct data exchange between them will introduce a more complicated message-passing pattern. Fortunately, direct data exchange between P_i and P_j is not necessary to get the diagonal ghost grid points. It can be verified that all data exchanges that we need are of nearest neighbor types, as indicated in Figure 8 for 2D problems. As can be seen in Figure 8 that when data lying on partition boundaries are exchanged, the sending blocks always include ghost grid points. After data exchanges in Figure 8 are performed, all ghost grid points shown in Figure 6 will be obtained by appropriate neighboring processors. Each processor, therefore, only needs to know its nearest neighboring processors on each logical processor mesh. In solving problems with periodic boundary conditions, data exchanges are also required among processors lying on the boundary of a processor mesh, and the same message-passing operations as shown in Figure 8 can be used.

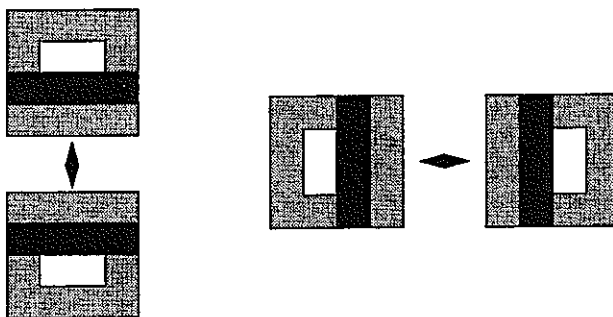


Figure 8: Data exchanges between neighboring processors for 2D problems. The data in black blocks in each processor are sent out, which is stored in the blocks for ghost grid points in the neighboring processors.

The parallel efficiency of a parallel code is largely determined by the ratio of local computations over interprocessor communications. In our solvers, the best parallel efficiency is achieved on the finest grid, where the communication cost could be easily dominated by a large amount of computations, and the parallel effi-

ciency degrades as the grid gets coarser. One way to hide communication overhead and thus improve parallel efficiency on all grids is to overlap communications with computations. In several places within our solvers, we have the following sequence of operations for each processor:

- (1) Exchange data lying on partition boundaries;
- (2) Perform processing on all local grid points.

To overlap communications with computations, we can perform the following sequence of operations for the same result:

- (1) Initiate the data exchange for partition boundaries;
- (2) Perform processing on interior grid points that do not need ghost grid points;
- (3) Wait until data exchange in (1) is complete;
- (4) Perform processing on the remaining grid points.

On the Intel Paragon, we implemented the second set of operations above in the multigrid solver and the flow solver using asynchronous message-passing calls. For one full V-cycle in the elliptic solver, for example, the performance improvement on a 256×256 grid partitioned among 256 processors is about 15%, and the improvement on a 256^3 grid partitioned among 512 processors is about 22%. Faster and asynchronous interprocessor communication can also be achieved on the Cray T3D by using its shared-memory communication model, in which direct memory copy is used at either sending or receiving processors for data exchanges between different processors. Some synchronization between sending and receiving processors, however, is needed before or after a direct memory copy is performed to ensure the correctness of a message-passing. On T3D processor synchronization is provided only for a group of processors with a fixed stride in their processor indexes, this shared-memory communication model can be easily used for exchange of partition boundary data in the flow solver and for multigrid elliptic solver on some fine grids in which data exchanges only occur between nearest-neighbor processors on the original processor mesh.

C. Software Structures

Our solvers were implemented in C because we think it is the language that provides adequate support for implementing advanced numerical software without incurring unreasonably large overhead. Since our goal is to develop reusable and high-performance PDE solvers which can be used either as library routine or as extensible, template-type code for different applications, we empha-

size in our code design both generality and flexibility. First, we want the solvers to be able to run on any M-dimensional rectangular processor meshes for any N-dimensional rectangular grids with ($M \leq N$) (for multigrid processing, N is usually a power of 2). This requirement introduces some complexities in coding the multigrid solver in terms of determining the right global indices for local grids at each grid level. Storages for all grid variables are allocated at run time. For the multigrid solver, storages for local coarse grids are allocated as they are derived recursively from local fine grids. The user is given the option either to supply the storage for variables defined on the original grid or to let the solver to allocate those storages. An array of pointers to an N-dimensional grid (i.e. an N-dimensional data array) is allocated, and each of the pointers points to a grid in the grid hierarchy. N-dimensional data array is constructed recursively from one-dimensional data arrays. This strategy of dynamic memory allocation offers a greater flexibility in data structure manipulations and more efficient use of memory than a static memory allocation, and the user is also alleviated from the burden of calculating storage requirements for multigrid processing.

There are two major communication routines in the solvers: the communication routine for the flow solver exchanges partition boundary data only on the original grid; the communication routine for the multigrid solver can exchange partition boundary data for all fine and coarse grids, using a hierarchy of processor meshes. To make the code portable across different message-passing systems, we defined our own generic message-passing library as an interface with our solvers. To use a new message-passing system, we only need to extend the generic message-passing library to that system without changing any code in our solvers. Currently, our generic message-passing library can accommodate NX, MPI and PVM. A separate data exchange routine has also been implemented for the flow solver, which uses the shared-memory communication library on the Cray T3D.

Simple user interfaces to the parallel solvers have also been constructed. The elliptic multigrid solver can be used as a stand-alone library routine with both C and Fortran interfaces. After initialization of the problem to be solved and some algorithm parameters, a preprocessing routine must be called before the first time the multigrid solver routine is called. The preprocessing routine constructs the set of nested grids and the corresponding set of logical processor meshes. The flow solver can be used as a general-purpose incompressible fluid flow solver on a rectangular, staggered finite-difference grid for problems with Dirichlet or periodic velocity

boundary conditions. To use the multigrid solver as a kernel for evaluating velocity and pressure fields, the preprocessing routine must be called for each velocity component and the pressure, since they are defined on different grid points on a staggered grid. Therefore separate data structures will be constructed in the preprocessing routine for each velocity component and the pressure, which will be used in subsequent calls to the multigrid solver.

4. Numerical Experiments and Parallel Performances

We now report numerical experiments made to examine the numerical properties of the parallel solvers on a few test problems, and parallel performance in terms of speed-up and parallel scaling of the solvers on the Intel Paragon and the Cray T3D systems for problems with different sizes and granularity.

A. The Elliptic Multigrid Solver

The multigrid elliptic solver was first tested on a Helmholtz equation with known exact solutions. Table 1 shows the convergence rate of the multigrid solver from a 3D Helmholtz equation of the form

$$-\Delta u + u = f$$

with a Dirichlet boundary condition. The runs were performed on the Intel Paragon. Errors displayed are the normalized maximum norm of the difference between the computed solution and the exact solution. The table shows the number of cycles needed in each case to reach the order of discretization error (or truncation error). At each grid level, two red-black relaxations were performed. Although the full V-cycle scheme has been shown more efficient than the V-cycle scheme in sequential processing [5], the same statement may not be always true in parallel processing. For a 2D test case with a 2048^2 grid, we found the execution time needed for the computed solution to reach a fixed accuracy is about the same for V-cycle and full V-cycle schemes. As shown for a 3D test case in table 1, the full V-cycle scheme is still a little more efficient. With a grid-partition of both fine and coarse grids, parallel efficiency degrades as the processing moves to coarser grids. Since the full V-cycle scheme does more processing on coarse grids, its parallel efficiency is worse than the V-cycle scheme. For a large computational grid with many levels of coarse grids, the higher numerical efficiency of the full V-cycle scheme may not improve overall computational performance due to its worse parallel efficiency. Although not shown in the paper, the

convergence rates of the multigrid solver were also measured for cell-centered grid and staggered grid. We found for the same model problem the convergence speeds are slightly slower on those grids, which could be due to the use of different restriction operators.

Table 1: Numerical Convergence: 3D Helmholtz Solver

Scheme	Grid	Error	# Nodes	# Cycles	CPU sec.
Full V-cycle	256^3	1.2×10^{-5}	64	4	71.3
V-cycle	256^3	2.6×10^{-5}	64	8	86.7

The parallel performance of an application code is usually judged by two measurements: speed-up and scaling. Speed-up is measured by fixing the problem size (or grid size in our case) and increasing the number of processors used. Scaling is measured by fixing the local problem size in each processor and increasing the number of processors used. Although a nice speed-up can be obtained for many applications with a small number of processors, the reduction in CPU time often diminishes rapidly as the number of processors used exceeds some threshold. This phenomenon is largely inevitable, as stated in Amdahl's law. Because as the number of processors used increases for a fixed problem size, the local communication cost and the cost for global operations will eventually become dominant over the local computation cost after a certain stage. This high ratio of communication to computation makes the influence of a further reduction in the local computation very small on the overall cost of running the application. On the other hand, the scaling performance seems to be a more realistic measure of an application's parallel performance, since a code with a good parallel scaling implies, given enough processors, it can solve a very large problem in about the same time as it requires for solving a small problem, and this is indeed one of the main reasons to use a parallel machine. Figure 9 displays two speed-up plots for a multigrid V-cycle and a full V-cycle for solving 2D and 3D Helmholtz equations with a Dirichlet boundary condition on a vertex-centered grid, measured on the Intel Paragon and the Cray T3D systems. For a comparison, an ideal speed-up curve for one test case is also shown. The code was compiled with the -O2 switch on both machines. The grid size for the 2D problem is 512×512 , and the grid size for the 3D problem is $64 \times 64 \times 64$. The maximum number of processors used for the 2D problem is 256 on both machines. For the 3D problem, all 256 processors were used on the T3D (in which case a rectangular

processor mesh of dimensions 4x8x8 was used) and 512 processors were used on the Paragon.

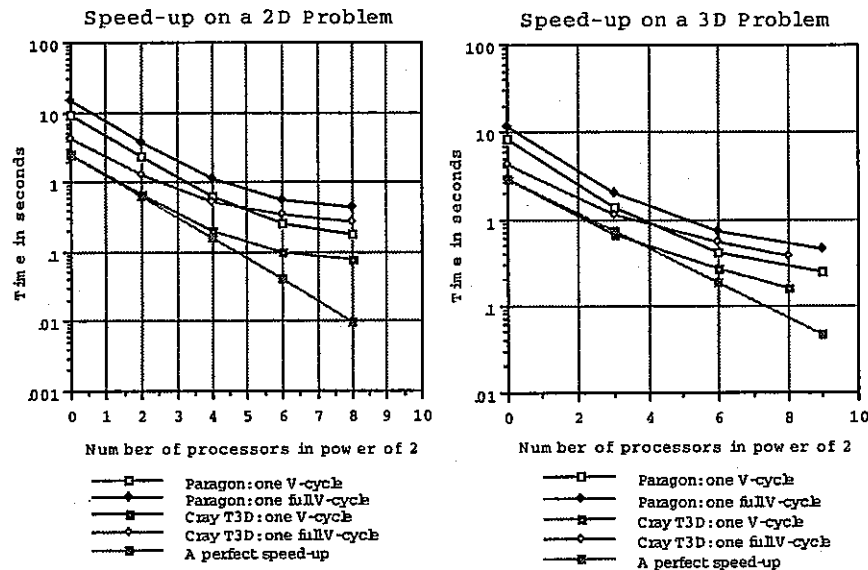


Figure 9: Speed-up performance of the elliptic multigrid solver.

In terms of single processor performance, we found for our multigrid solver that the Cray T3D is about 4 times faster than the Intel Paragon. But since the implementation of PVM on T3D, which we used in our code for message-passing, is relatively slow for inter-processor communication, the performance difference for a parallel application on both machines tends to become smaller as granularity of the problem gets finer. We can see for both 2D and 3D problems that the V-cycle scheme has a slightly better speed-up performance than the full V-cycle scheme, which is expected since the full V-cycle scheme does more processing on coarse grids. For the 2D problem, speed-up started to degrade when more than 16 processors were used, and for the 3D problem, the degradation started when more than 8 processors were used. Despite the degradation in speed-up, we can still see some reduction in CPU time when the largest number of processors was used in each case.

Figure 10 shows the scalings of the parallel multigrid solver on the Intel Paragon for problems with three different granularity, using up to 512 processors. Figure 11 shows the scalings of the same problems on Cray T3D, using up to 256 processors. Shown in the plots are the ratio of CPU times of using n processors versus using one processor. On each of the scaling curves, we fix the local grid size and increase the number of processors, so a flat curve

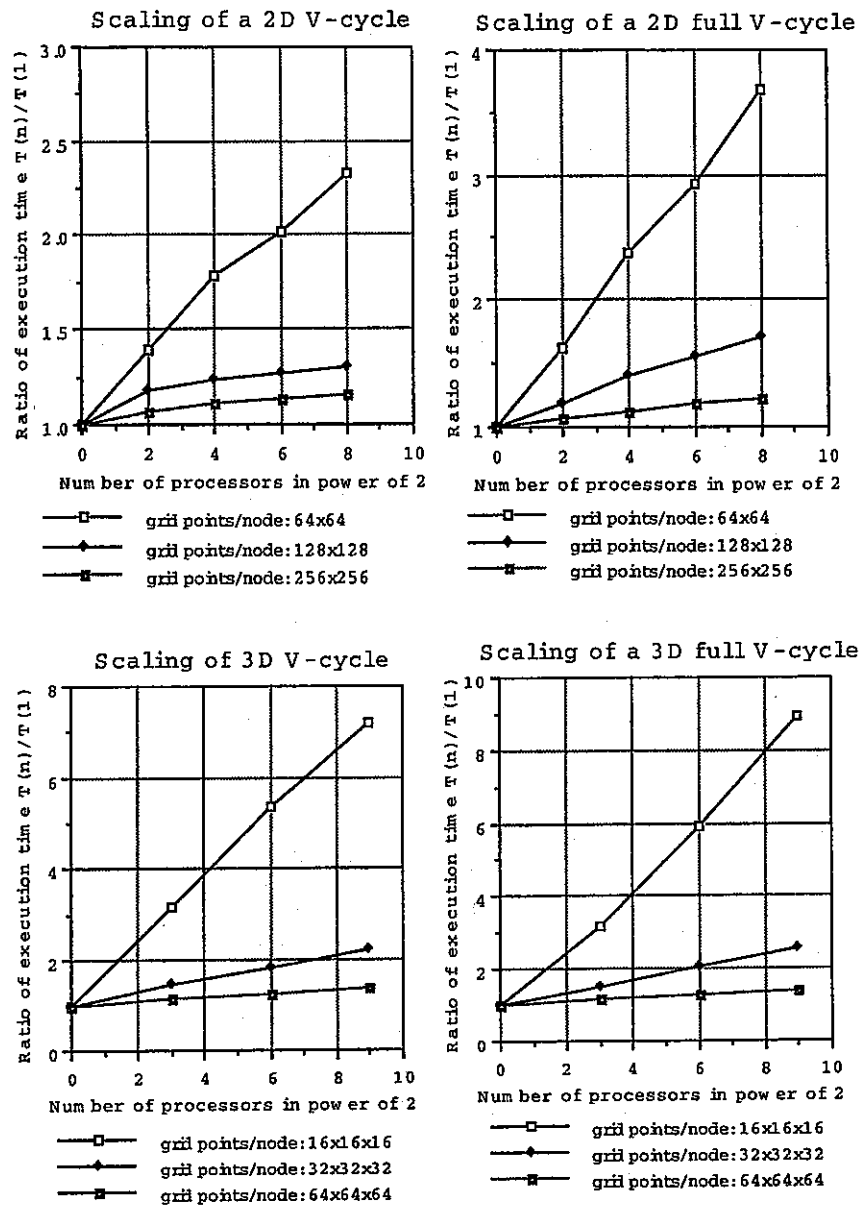


Figure 10: Scaling of the multigrid solver on Intel Paragon.

means a perfect scaling. Since a larger global grid has more coarse grid levels for a complete V-cycle or full V-cycle, cost for processing on coarse grids also rises as the number of processors increases, and therefore it has a negative effect on the scaling performance. Like speed-up performance, scaling performance is also largely determined by the ratio of local computation cost versus communication cost. This ratio can be dependent on both numerical/parallel algorithms and hardware/software performance on each specific

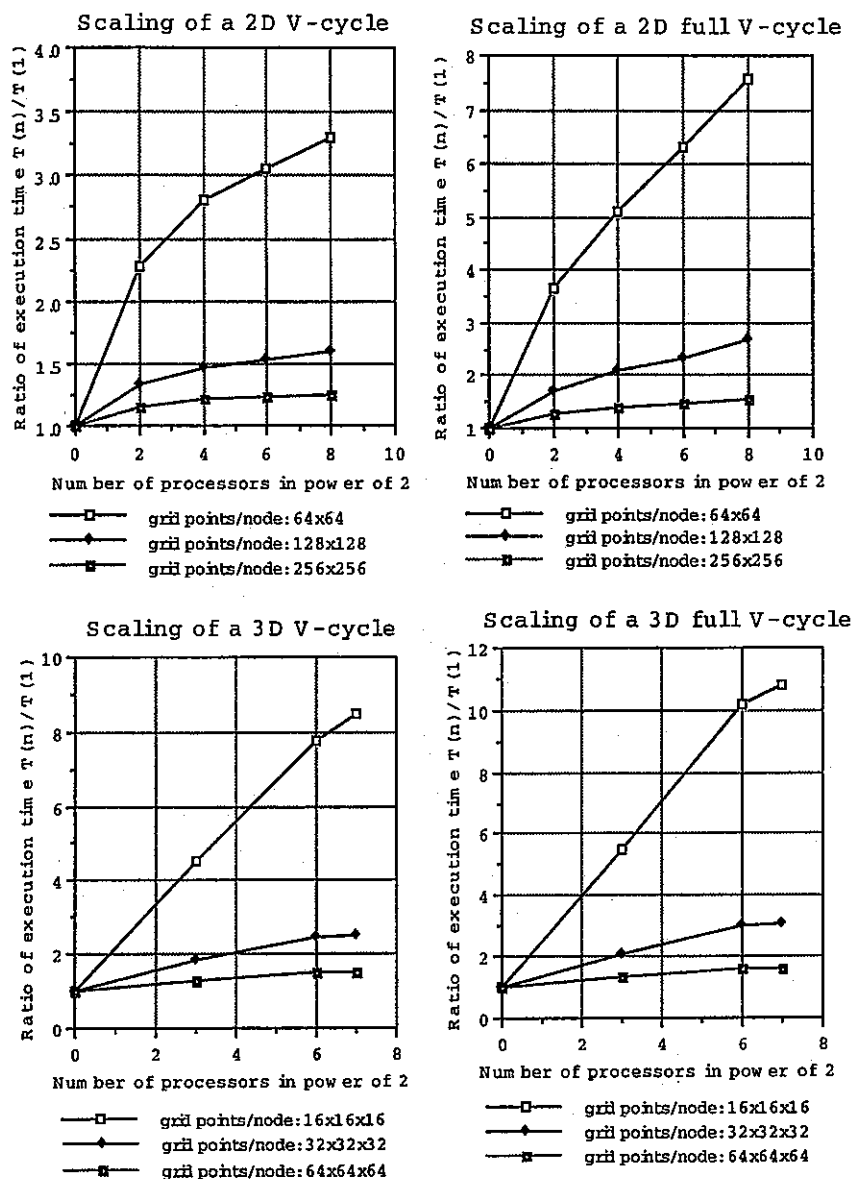


Figure 11: Scaling of the multigrid solver on Cray T3D.

machine. We can see from all the plots in Figure 10 and Figure 11 that scaling performance improves as the size of local grid increases. This improvement is expected for an iterative scheme on a single grid, since the computation cost scales as $O(n)$, where n is the number of grid points in the local grid, whereas the communication cost scales as $O(n^{1/2})$. For a multigrid scheme, it can still be shown that both computation cost and communication cost scale with the same orders as on a single grid [8]. In addition, message-passing latency does not increase as proportionally since the num-

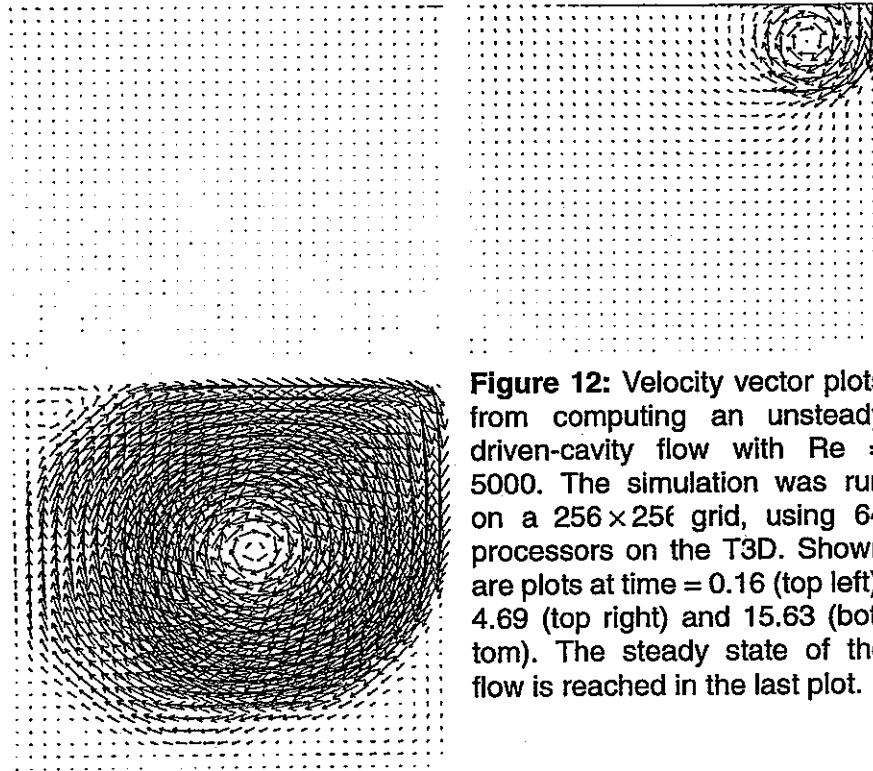


Figure 12: Velocity vector plots from computing an unsteady driven-cavity flow with $Re = 5000$. The simulation was run on a 256×256 grid, using 64 processors on the T3D. Shown are plots at time = 0.16 (top left), 4.69 (top right) and 15.63 (bottom). The steady state of the flow is reached in the last plot.

ber of messages communicated is still roughly the same for a larger local grid (not exactly the same because more coarse levels are involved) though the size of each message is larger. We can also see V-cycle scheme scales somewhat better than full V-cycle scheme, which is expected since the latter does more operations on coarse grids. The scaling plots also show 2D test cases scales better than 3D test cases, which we think is due to the fact that 3D grids have a higher surface to volume ratio than the 2D grids and thus the ratio of computations to communications is smaller for 3D cases. As for a comparison between the Intel Paragon and the Cray T3D, our results show that the scalings on Paragon are slightly better than on T3D. This could be explained by the fact that single processor speed on T3D is much faster than on Paragon, whereas the speed of interprocessor communication on T3D is not proportionally faster when PVM is used for communication.

B. The Incompressible Navier-Stokes Solver

The parallel Navier-Stokes flow solver was first tried out on a test problem in a unit square with a known exact solution. The purpose of the test is to examine the convergence rate of the flow

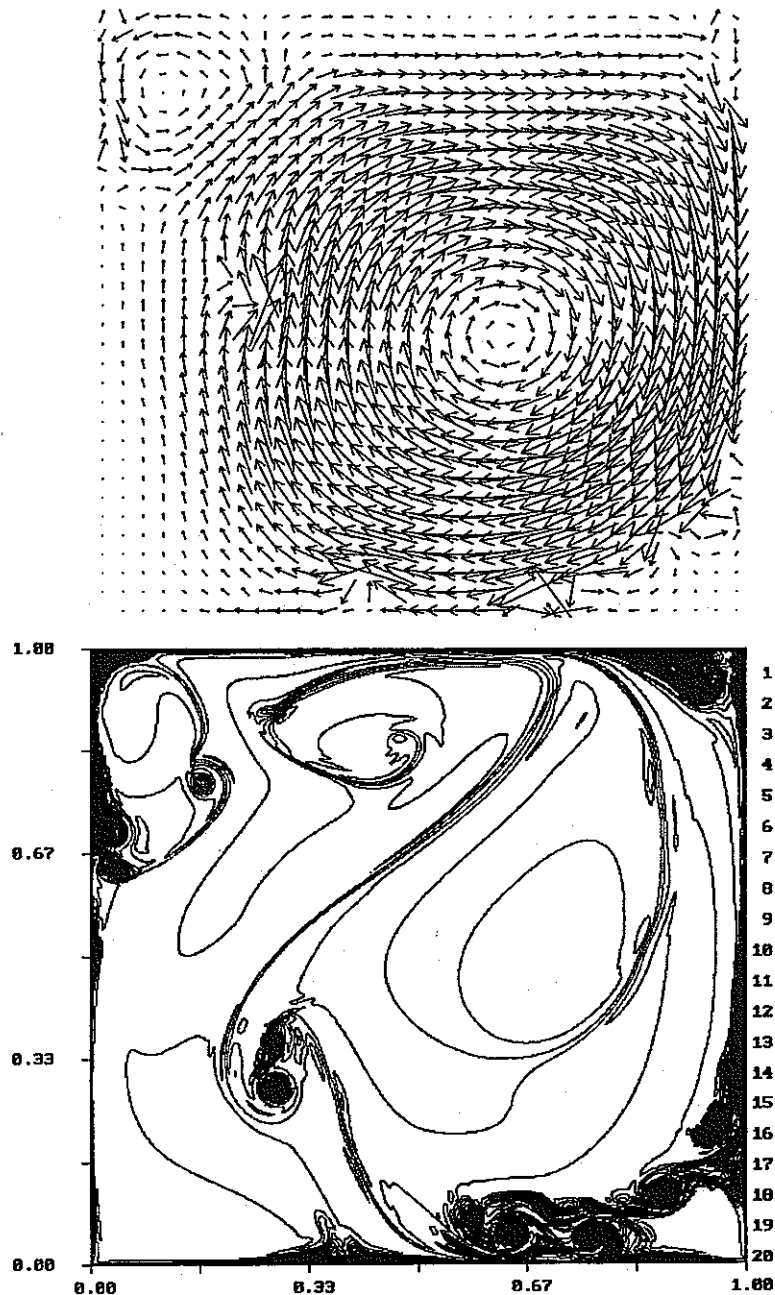


Figure 13: Velocity vector plot (top) and vorticity contour plot (bottom) from a driven-cavity flow with $Re = 10^6$, at time = 5.47. Grid size = 512×512 .

solver on smooth problems. A second-order convergence rate was obtained on the test problem for Reynolds number up to 5000, as expected. For numerical stability of the Godunov scheme used in discretizing the convection term, the time step, Δt , is restricted by the CFL condition

$$\frac{\Delta t}{\Delta x} U_{\max} < 0.5, \quad (10)$$

in the test, where Δx is the size of grid cells and $U_{\max}$ is the maximum value in the current velocity field. A detailed description of such a test is given in [2].

Our next numerical experiment on the flow solver was to simulate an evolving 2D driven-cavity flow. The computational domain is still in a unit box $0 \leq x, y \leq 1$. The no-slip velocity boundary condition is applied to all boundaries except at the top boundary, where the velocity value is given. We first tested the solver on the problem in which the velocity initial condition is specified by $\mathbf{u} = 0$ inside the domain, and the velocity at the top boundary is always one. Figure 12 displays the velocity vector fields which show three stages for time = 0.16, 3.91 and 15.63 in the evolution of the flow, computed on a 256×256 grid with the Reynolds number = 5000. The CFL number (i.e. the right hand-side of (10)) used in the calculation is 0.4, and a total of 10,000 time steps were computed to reach the last state at time = 15.63. We found the vorticity structure at time = 15.63 is similar to those obtained by solving the steady incompressible Navier-Stokes equations (e.g. [12]). In running the parallel solver on the Cray T3D, the global staggered grid was partitioned and distributed to an 8×8 logically rectangular processor mesh. In computing the velocity vector field, velocity components defined on cell edges were averaged to the center of cells. For better visibility, the vector fields shown in Figure 12 are actually 32×32 data arrays which were obtained by averaging the 256×256 velocity vectors from the simulation. Vorticity fields were computed at cell corners by central differencing. The velocity vector plots in Figure 12 show clearly how the cavity flow develops from its initial state to the final steady state which is characterized by a primary vortex in the center of the unit box and two secondary vortices at the two bottom corners and a small vortex at the upper left corner (e.g. [12]). We also noticed, when reaching the final stage in Figure 12, that the change of numerical divergence of the computed velocity field before and after projection is very small. This is because, when the steady state is reached, the intermediate velocity field would be computed using the correct pressure field to result in a correct velocity field. Even though the initial condition is not continuous along the top boundary and the boundary condition is not continuous at the two upper corners of the unit box, the numerical computation of the flow solver turns out to be quite stable.

The flow solver was next tried on a driven-cavity flow problem with some smooth initial and boundary conditions. The top boundary now moves with a slip velocity $u_t(x) = 16x^2(1-x^2)$ and the initial velocity field is specified through a stream function $\psi_0(x, y) = (y^2 - y^3)u_t(x)$. The velocity is then computed by $u = -\psi_y$ and $v = \psi_x$. In this case, we wanted to test the numerical stability of the flow solver on problems with large Reynolds numbers which will result in a very thin boundary layer at the top boundary. Figure 13 displays the result from a calculation with $Re = 10^6$ for a total of 7000 time steps on a 512×512 grid. The computation was performed on the Cray T3D using 64 processors. We noticed the computation using our solver at such a high Reynolds number is still numerically stable, which we can judge by checking the convergence rate of the pressure equation and the numerical divergence of the computed velocity. The computed flow structure at this Reynolds numbers, however, is quite different from that obtained from the computed flow with $Re = 5000$. First, at this high Reynolds numbers, the computed flow does not show any sign of approaching a steady state after computing the large number of time steps; while with $Re = 5000$, for the same initial and boundary conditions, we found a steady state can be reached after computing a much smaller number of time steps. Secondly, we can see some interesting flow patterns which do not exist in the flow with $Re = 5000$. As shown by the vorticity contour in Figure 13, the vorticity structures in this high Reynolds number flow is much more complicated. We can see that a large amount of vortices are generated from the top boundary and then being flushed down along the right wall. Once these vortices reach the neighborhood of the lower right corner, they are pushed toward the interior of the box. We found the vorticity plot in Figure 13 is similar to what reported in [13] where a different algorithm was used on the same problem.

The second problem used to test our flow solver is an inviscid flow for which the Euler equations are solved. The computational domain is again in a unit box, and a periodicity of one is assumed in both horizontal and vertical directions. The initial velocity field is given by

$$u = \begin{cases} \tanh(y - 0.25)/\rho & \text{for } y \geq 0.5 \\ \tanh(0.75 - y)/\rho & \text{for } y < 0.5, \end{cases} \quad (11)$$

$$v = \delta \sin(2\pi x)$$

where $\rho = 0.03$ and $\delta = 0.05$. Thus the initial flow field consists of a jet which is a horizontal shear layer of finite thickness, perturbed by

p	p	p	p
v	v	v	v
p	u	p	u
p	u	p	u
v	v	v	v
p	u	p	u
p	u	p	u
v	v	v	v
p	u	p	u
p	u	p	u

Figure 14: An example of a staggered grid used for computing the doubly periodic shear flow with $N = 4$. Unknowns for velocity and pressure in the grid are shown.

a small amplitude of vertical velocity. Since the viscous term is dropped, the pressure can be updated without computing the intermediate velocity field and the multigrid elliptic solver is only used for solving the pressure equation (9). In addition, only one iteration at each time step is needed for computing $p^{n+1/2}$ and u^{n+1} because the pressure can be computed without the knowledge of u^{n+1} . On the staggered grid we used, the pressure field is defined on a cell-centered grid whose linear dimension, say N , is preferably taken as a power of 2 for the convenience of applying grid coarsening. Thus there are N^2 unknowns for the pressure. Since velocity field is only related to the pressure gradient in the momentum equations, it makes sense to have the velocity defined on an $(N-1) \times (N-1)$ grid, as shown in Figure 14. Therefore there are $(N-1)^2$ unknowns for each velocity component. Since the velocity is periodic, a periodic domain should have a dimension of $(N-1) \times (N-1)$. Since the pressure gradient is a function of velocity, it must have the same dimension of periodicity. Thus the physical boundary condition for the pressure equation (9) in the horizontal direction, for example, can be specified as

$$P_{0,j} = P_{1,j} + P_{N-2,j} - P_{N-1,j} \quad P_{N+1,j} = P_{N,j} + P_{2,j} - P_{1,j} \quad (12)$$

In a multigrid solution of the pressure equation, the boundary condition (12) is clearly for use on the original, finest grid. The use of condition (12) on any coarse grid, however, is incorrect. Our numerical experiments indicate the use of (12) on coarse grids will blow up the computation quickly. Since the unknown vector on a coarse grid is the difference between an exact solution and an approximate solution on the fine grid restricted to that coarse grid, the solution on a coarse grid can be regarded as an approximation to the derivative of the solution on the fine grid. Since a derivative of the pressure field of any order is still periodic with the same period as the velocity field, a reasonable boundary condition for pressure on coarse grids is

$$P_0 = P_N \quad P_{N+1} = P_1 \quad (13)$$

Although condition (13) imposes a period which is one grid cell (of

the finest grid) larger than the velocity period, we found it is easy to apply it to all the coarse grids, and our numerical results show it works well.

Numerical experiments for the inviscid periodic shear flow were performed on Cray T3D, using 64 processors in all cases. Figure 15 shows vorticity contours of two early states of the inviscid periodic shear flow. Figure 16 and 17 show vorticity contours of the flow at time = 1.25 and 2.50, computed on a 128×128 grid, and a 512×512 grid, respectively. The CFL number used in the computations is still 0.4. On the 512×512 grid, a total of 3200 time steps were computed to reach time = 2.50. These vorticity plots show how the shear layers, which form the boundaries of the jet, evolve into a periodic array of vortices, with the shear layer between the rolls stretched and thinned by the large straining field there. A comparison between Figures 17 and 18 clearly shows the resolution of the vorticity structure improves as the computational grid gets finer.

The parallel performance of the incompressible flow solver was also evaluated in terms of speed-up and scalability. In each of the parallel performance measurement, we ran the flow solver on the driven-cavity problem for one time step, excluding any initialization and assignment of initial and boundary conditions. Figure 18 shows the speed-up curves of the flow solver on the Intel Paragon and the Cray T3D for three different problem sizes (ideal speed-up curves are shown again for comparison). The speed-up performance improves as the problem size increases, as expected. For the 512×512 grid, no significant reduction in execution time could be obtained after more than 64 processors were used. By running the flow solver on a single processor, we found T3D is about five times faster than the Paragon for the code compiled with the -O2 switch. But on 256 processors, T3D runs only about 1.5~2.0 times faster than Paragon depending on problem sizes, because, as shown in Figure 18, the speed-up performance of the flow solver on Paragon is better than on T3D. Figure 19 shows scaling performance of the parallel flow solver on T3D and Paragon for three local problem sizes. Again, we see the scaling improves as the size of local grid increases on both machines. In measuring the scalings of the flow solver, we used smaller local problem sizes than we did for the multigrid elliptic solver (see Figure 10 and 11). We expect the flow solver to have better scalings than the multigrid elliptic solver because the flow solver, even though calling the elliptic solver several times at each time step, does substantially additional processings on the finest grid. Indeed, this scaling difference between the two solvers can be verified by looking at the scaling curves for the

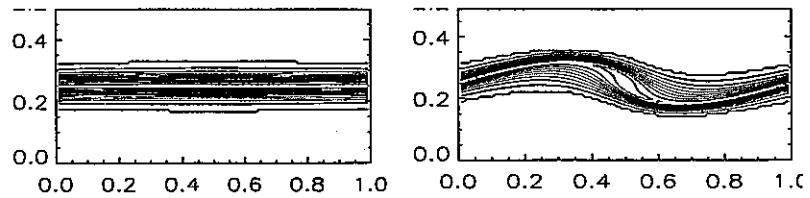


Figure 15: Vorticity contour plots from the periodic shear flow at time = 0.0 (left) and 0.62 (right). Grid size = 128×128 .

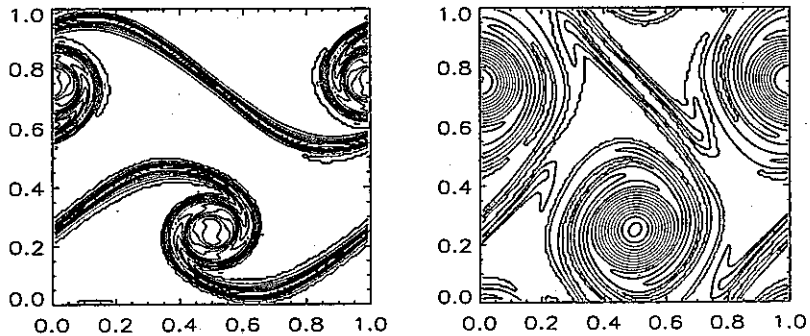


Figure 16: Vorticity contour plots from the periodic shear flow for time = 1.25 (left) and time = 2.50 (right). Grid size = 128×128 .

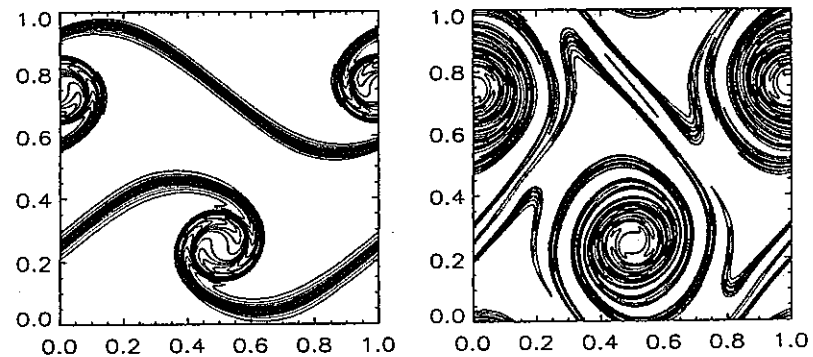


Figure 17: Vorticity contour plots from the periodic shear flow for time = 1.25 (left) and time = 2.50 (right). Grid size = 512×512 .

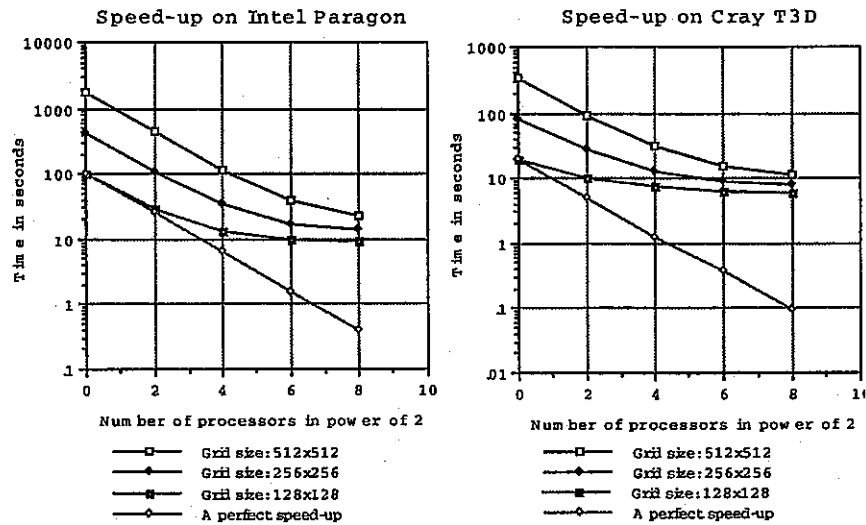


Figure 18: Speed-up performances of the parallel Navier-Stokes solver.

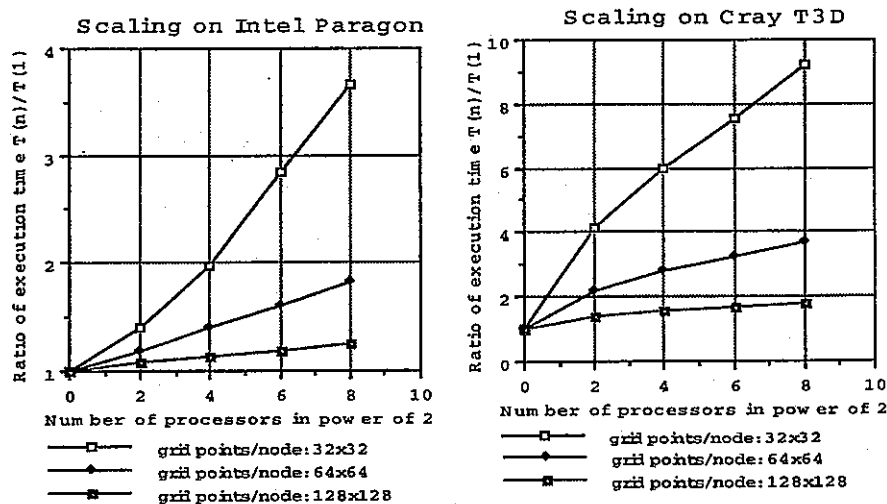


Figure 19: Scaling performances of the parallel Navier-Stokes solve

64 × 64 local grid for the flow solver in Figure 19 and for the multigrid full V-cycle (which is used in the flow solver) in Figure 10 and 11. In view of the scaling performance in Figure 19, we would claim that our parallel flow solver scales quite well on large numbers of processors as long as the local grid size is not smaller than 64 × 64.

5. Conclusions

In this paper we presented multigrid schemes for solving elliptic PDEs and a second-order projection method for solving the Navier-Stokes equations for incompressible fluid flows. Our parallel

implementation strategies based on grid-partition are discussed for implementing these algorithms on distributed-memory, massively parallel computer systems. Our treatment of various boundary conditions in implementing these parallel solvers is also discussed. We designed and implemented these solvers in a highly modular fashion so that they can be used either as stand-alone solvers or as expandable template codes which can be used in different applications. Several message-passing protocols (MPI, PVM and Intel NX) have been coded into the solvers so that they are portable to systems that support one of these interfaces for interprocessor communications.

Numerical experiments and parallel performance measurements were made on the implemented solvers to check their numerical properties and parallel efficiency. Our numerical results show the parallel solvers converge with the order of numerical schemes on a few test problems. Our numerical experiments also show the flow solver is stable and robust on viscous flows with large Reynolds numbers as well as on an inviscid flow. Our parallel efficiency tests on the Intel Paragon and the Cray T3D systems show that good scalability on a large number of processors can be achieved for both the multigrid elliptic solver and the flow solver as long as the granularity of the parallel application is not too small, which we think is typical for applications running on distributed-memory, MIMD machines. For future work, we plan to generalize the parallel solver package to thermally-driven flows and variable density flow problems and extend the flow solver to 3D problems.

Acknowledgments:

The authors wish to thank Dr. Sefan Vandewalle (California Institute of Technology) and Dr. Steve McCormick (University of Colorado) for some helpful discussions on multigrid methods. This work was carried out at the Jet Propulsion Laboratory (JPL), California Institute of Technology (Caltech), under a contract with the National Aeronautics and Space Administration (NASA) and as a part of the NASA High-Performance Computing and Communications for Earth and Space Sciences Project. The computations were performed on the Intel Paragon parallel computers operated by JPL and by the Concurrent Supercomputing Consortium at Caltech, and on the Cray T3D parallel computer operated by JPL.

References:

1. C. Anderson, "Derivation and Solution of the Discrete Pressure-Equations for the Incompressible Navier-Stokes Equations." Lawrence Berkeley Laboratory Report, LBL-26353, 1988, Berkeley, CA (unpublished)
2. J. B. Bell, P. Colella and H. Glaz, "A Second-Order Projection Method for the Incompressible Navier-Stokes Equations," J. Comp. Phys., 85:257-283, 1989
3. J. B. Bell, P. Colella and L. H. Howell, "An Efficient Second-Order Projection Method for Viscous Incompressible Flow." Proceedings, 10th AIAA Computational Fluid Dynamics Conference, Honolulu, HI, pp.360-367, 1991
4. J. B. Bell and D. L. Marcus, "A Second-Order Projection Method for Variable-Density Flows." J. Comp. Phys., Vol 101, No 2, pp. 334-348, 1992
5. W. Briggs, "A Multigrid Tutorial," SIAM, Philadelphia, 1987
6. T. F. Chan and R. S. Tuminaro, "A Survey of Parallel Multigrid Algorithms", in "Parallel Computations and Their Impact on Mechanics", A. Noor, Ed., Vol: AMD 86, 1986
7. A. J. Chorin, "Numerical Solution of the Navier-Stokes Equations," Math. Comp., vol. 22, pp. 745-762, Oct. 1968
8. G. Fox, et. al., "Solving Problems on Concurrent Processors." Vol. I, Prentice Hall, Englewood Cliffs, New Jersey, 1988
9. S. F. McCormick, "Multilevel Adaptive Methods for Partial Differential Equations." Frontiers in Applied Mathematics, SIAM, Philadelphia, 1989
10. W. D. Henshaw, Part I: The Numerical Solution of Hyperbolic Systems of Conservation Laws; Part II: Composite Overlapping Grid Techniques, Ph.D Thesis, Dept. Appl. Math., California Institute of Technology, Pasadena, CA, 1985
11. F. Roux and D. Tromeur-Dervout, "Parallelization of a Multigrid Solver via a Domain Decomposition Method." Manuscript, 1994
12. R. Schreiber and H. B. Keller, "Driven Cavity Flows by Efficient Numerical Techniques." J. Comp. Phys., 49, 310-333, 1983
13. Weinan E and Jian-Guo Liu, "Essentially Compact Schemes for Unsteady Viscous Incompressible Flows." Manuscript, 1994

14. P. Wesseling, "An Introduction to Multigrid Methods", Pure & Applied Mathematics, A Wiley-Interscience Series of Texts, Monographs & Tracts, John Wiley & Sons, 1991

John Z. Lou received his Ph.D in applied mathematics from the University of California at Berkeley in 1991, a M.S. in engineering mechanics and a B.A. in applied mathematics from Shanghai Jiao-Tong University, Shanghai, China in 1985 and 1982. From 1991 to 1993, he worked as a computational scientist in the Naval Command, Control and Ocean Surveillance Center in San Diego. He currently works as a member of technical staff in the High-Performance Computing Systems and Algorithms Group in the Jet Propulsion Laboratory of California Institute of Technology. His current research interests include parallel numerical algorithms, parallel software technologies and implementations for scientific applications.

Robert D. Ferraro received a B.A. in physics from Cornell University, College of Arts and Sciences, in 1978, and an M.A. in 1980 and Ph. D. in 1984 in physics from the University of Rochester. He did post-doctoral work in the Plasma Theory Group of the UCLA physics department before joining the Jet Propulsion Laboratory in 1988 to work on the application of parallel processing to scientific and engineering computations. Currently, Dr. Ferraro is the Technical Group Supervisor for the High Performance Computing Systems and Algorithms Group, and Associate Project Manager for the NASA HPCC Earth and Space Sciences project. His current research interests center around the application of parallel computing technology to scientific applications programming. Dr. Ferraro is a member of the American Physical Society Divisions of Plasma Physics and Computational Physics.

Copyright © 1995 by the Association for Computing Machinery, Inc. (ACM).

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that new copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted.

To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Publications Dept, ACM Inc., via fax at +1 (212) 869-0481, or via email at permissions@acm.org.

Large Eddy Simulation of a Spatially-Developing Boundary Layer

Xiaohua Wu & Kyle D. Squires
Mechanical Engineering Department
Votey Building
University of Vermont
Burlington, VT 05405
wu@emba.uvm.edu

Thomas S. Lund
Center for Turbulence Research
Stanford University
Building 500
Stanford, CA 94305
lund@ctr-next3.stanford.edu

Contact name: Kyle D. Squires
Mechanical Engineering Department
Votey Building
University of Vermont
Burlington, VT 05405
squires@emba.uvm.edu
ph: 802-656-1940

Keywords: Large eddy simulation, turbulent boundary layers, inflow turbulence

Abstract

A method for generation of a three-dimensional, time-dependent turbulent inflow condition for simulation of spatially-developing boundary layers is described. Assuming self-preservation of the boundary layer, a quasi-homogeneous coordinate is defined along which streamwise inhomogeneity is minimized (Spalart 1988). Using this quasi-homogeneous coordinate and decomposition of the velocity into a mean and periodic part, the velocity field at a location near the exit boundary of the computational domain is re-introduced at the inflow boundary at each time step. The method was tested using large eddy simulations of a flat-plate boundary layer for momentum thickness Reynolds numbers ranging from 1470 to 1700. Subgrid scale stresses were modeled using the dynamic eddy viscosity model of Germano et al. (1991). Simulation results demonstrate that the essential features of spatially-developing turbulent boundary layers are reproduced using the present approach without the need for a prolonged and computationally expensive laminar-turbulent transition re-

gion. Boundary layer properties such as skin friction and shape factor as well as mean velocity profiles and turbulence intensities are in good agreement with experimental measurements and results from direct numerical simulation. Application of the method for calculation of spatially-developing complex turbulent boundary layers is also described.

1. Introduction and Background

Calculation of complex turbulent flows by direct numerical simulation (DNS) and large eddy simulation (LES) remains one of the principal challenges of computational fluid dynamics. All complex flows encountered in practice are temporally- or spatially-developing and accounting for their evolution in either space or time is crucial to obtaining an accurate description of mean flow and turbulence characteristics. However, the majority of DNS and LES computations are currently performed for flows in which there are at least two directions of statistical homogeneity. These simulations are motivated primarily by the simplifications associated with their calculation and are typically limited to canonical flow fields in which homogeneity permits application of periodic boundary conditions (e.g., see Kim et al. 1987, Spalart 1988). While simulation of these flows has significantly increased our fundamental understanding of many of the dynamical processes governing turbulence, calculation of more complex spatially-developing flows is required to further advance DNS and LES as a predictive tool.

One of the most important issues in calculation of spatially-developing flows in DNS or LES is the need for prescription of turbulent inflow conditions. Of principal interest in this work is generation of turbulent inflow conditions for simulation of turbulent boundary layers. Prescription of turbulent "inflow" in an experiment is typically accomplished by tripping the boundary layer at a location far enough upstream of the test section such that a well-defined, nominally two-dimensional boundary layer is subsequently obtained. Thus, analogous to experiments, in DNS and LES calculations it is desirable to prescribe a turbulent inflow condition, i.e., a two-dimensional boundary layer, at the inflow boundary of the computational domain in order to calculate spatially-developing flows.

One of the most straightforward methods of specifying turbulent inflow is through prescription of a mean-flow profile together with superposition of random fluctuations. The amplitude of the fluctuations are constrained to satisfy a particular energy spectrum and the phase relationships of the fluctuations are chosen randomly. This approach has been successfully applied by Lee et al. (1992) in computation of grid turbulence. Rai & Moin (1993) employed a similar technique for computation of spatially-developing boundary layers using DNS,

including laminar-turbulent transition and the fully turbulent region. Rai & Moin (1993) were able to capture the essential features of the transition process, e.g., counter-rotating streamwise vortices, in the domain near the inflow boundary. Flow-field statistics in the turbulent region were also in good agreement with experiments. While the work of Lee et al. (1992) and Rai & Moin (1993) demonstrates the viability of specifying random turbulent inflow, the disadvantage of this approach for calculation of wall-bounded flows are the long streamwise lengths required for the flow field to become fully turbulent. Rai & Moin used a multiple-zone technique with a total resolution of approximately $1214 \times 61 \times 361$ grid points in the streamwise, wall-normal, and spanwise directions, respectively. Prescription of a fully-developed turbulent velocity profile at the inflow boundary of the computational domain would allow smaller domain sizes, and consequently fewer grid points, to be used in the simulations.

Another approach for generating turbulent inflow conditions for computation of spatially-developing flows is the separate simulation of a "periodic" flow, i.e., a turbulent flow in which periodic boundary conditions are imposed. Data from the "periodic" calculation is then supplied, time step by time step, at the inflow boundary of the spatially-developing computation. This approach has been successfully applied to calculations of internal flows such as the plane diffuser and backward facing step in which it is possible to perform an independent simulation of fully-developed turbulent channel flow and then use the channel flow data as the inflow condition for the spatially-developing simulation (e.g., see Kaltenbach 1993, Akselvoll et al. 1993). The drawback of this approach for simulations of unbounded flows is that spatially-developing boundary layers are not periodic in the inflow direction and thus the data supplied from a "periodic" calculation is inconsistent with the boundary conditions employed in the spatially-developing calculation. For example, Lund (1993) used velocity fields from simulations of parallel boundary layers as the inflow condition for LES of a curved boundary layer and found that an unphysical adjustment in the velocity field occurred downstream of the inflow boundary in the spatially-developing computation.

Other approaches to specifying turbulent inflow conditions include the work of Tsai & Leslie (1990) in which results from simulations of a temporally-evolving, nominally two-dimensional boundary layer are transformed to those representative of a spatially-developing flow. The spatially-developing boundary layer was approximated by a flow that is homogeneous in the streamwise direction and grows outwards in time. Boundary layer growth was simulated through the growth of a thermal boundary layer in which a step change in temperature diffused outward from one wall in a fully-developed channel flow. The momentum thickness Reynolds number Re_θ varied from 353 to 576

and results were found to be in good agreement with experimental data. The difficulty with using their method as a means for prescription of inflow turbulence is the difficulty in obtaining a sufficient sample of velocity fields since separate computations are required to produce a single inflow profile for a specified momentum thickness Reynolds number.

These studies provide a sampling of the approaches currently available for prescription of inflow turbulence in computations of spatially-developing turbulent boundary layers. Despite the fact that generation of inflow turbulence is an issue central to accurate prediction of complex flows, current methods possess deficiencies which limit their application to calculation of spatially-developing complex flows. Therefore, the principal objective of the present work has been construction of a computationally efficient and simple method that can be used for generation of inflow turbulence and simulation spatially-developing turbulent boundary layers. Relevant to this objective are the DNS computations of flat-plate boundary layers by Spalart (1988). By taking advantage of the fact that both the thickness of the boundary layer and the energy level of the turbulence vary slowly as functions of streamwise distance, Spalart defined a new set of coordinates along which streamwise inhomogeneity is minimized. Through the coordinate transformation as well as addition of growth terms to the Navier-Stokes equations, Spalart (1988) performed computations of turbulent boundary layers for Re_θ ranging from 225 to 1410.

As will be shown, the coordinate system transformation proposed by Spalart (1988) forms the basis for the method of turbulence inflow generation developed in this work. In particular, the definition of a quasi-homogeneous coordinate which minimizes streamwise inhomogeneity provides a convenient coordinate along which a velocity profile near the outflow boundary may be reassigned at the inflow plane. However, unlike the DNS calculations of Spalart (1988) in which the Navier-Stokes equations were also transformed and the flow for a single momentum-thickness Reynolds number was simulated, in this study the Navier-Stokes equations are solved in the original Cartesian coordinate system and no transformation of the governing equations is performed. The coordinate transformation is used only for the purpose of re-introducing turbulence leaving the computational domain and is not directly related to the solution procedure of the governing equations. The primary advantage of this approach is its simplicity and computational efficiency relative to performing independent simulations as in Spalart (1988).

The method of inflow generation developed in this study was tested through LES of a spatially-developing boundary layer. The numerical scheme and subgrid-scale model are described in §2.1 and an overview of the inflow generation scheme is provided in §2.2. LES

results presented in §3 demonstrate that flow-field statistics from the present spatially-developing computations are in good agreement with both experimental measurements and DNS results. A summary of the work may be found in §4.

2. Method Overview

2.1. Governing equations and numerical approach

Simulations of spatially-developing boundary layers were performed using large eddy simulation of the incompressible Navier-Stokes equations. The filtered continuity and momentum equations are

$$\frac{\partial \bar{u}_j}{\partial x_j} = 0, \quad (1)$$

$$\frac{\partial \bar{u}_i}{\partial t} + \frac{\partial \bar{u}_i \bar{u}_j}{\partial x_j} = -\frac{1}{\rho} \frac{\partial \bar{p}}{\partial x_i} + \nu \frac{\partial^2 \bar{u}_i}{\partial x_j \partial x_j} - \frac{\partial \tau_{ij}}{\partial x_j}, \quad (2)$$

where $\bar{\cdot}$ denotes application of the filter. The subgrid-scale (SGS) stress $\tau_{ij} = \bar{u}_i \bar{u}_j - \bar{u}_i \bar{u}_j$, appearing in (1) is parameterized by an eddy viscosity model

$$\tau_{ij} - \frac{\delta_{ij}}{3} \tau_{kk} = -2\nu_T \bar{S}_{ij} = -2C \bar{\Delta}^2 |\bar{S}| \bar{S}_{ij}, \quad (3)$$

in which δ_{ij} is the Kronecker delta and $|\bar{S}| = \sqrt{2\bar{S}_{ij}\bar{S}_{ij}}$ is the magnitude of the large-scale strain rate tensor, $S_{ij} = 1/2 (\partial \bar{u}_i / \partial x_j + \partial \bar{u}_j / \partial x_i)$. The trace of the SGS stress, τ_{kk} , is added to the pressure term. Closure of the SGS stress τ_{ij} is obtained through specification of the model coefficient C appearing in (3).

Following Germano et al. (1991) a second filter, the test filter, (denoted throughout this work by $\hat{\cdot}$) is used to derive an expression for the model coefficient C . Since the resolved turbulent stress $L_{ij} = \bar{u}_i \bar{u}_j - \hat{u}_i \hat{u}_j$, the SGS stress τ_{ij} , and the subtest-scale stress $T_{ij} = \hat{u}_i \hat{u}_j - \hat{u}_i \hat{u}_j$ are related by

$$L_{ij} = T_{ij} - \hat{\tau}_{ij}, \quad (4)$$

an expression for the model coefficient C can be obtained following the least-squares approach of Lilly (1992), i.e.,

$$C = \frac{1}{2} \frac{L_{ij} M_{ij}}{M_{kl} M_{kl}}, \quad (5)$$

where

$$M_{ij} = \hat{\Delta}^2 |\hat{S}| \hat{S}_{ij} - \bar{\Delta}^2 |\bar{S}| \bar{S}_{ij}. \quad (6)$$

The filter widths at the grid and test levels are denoted by $\bar{\Delta}$ and $\hat{\Delta}$, respectively, and the ratio $\hat{\Delta}/\bar{\Delta}$ is the only parameter requiring specification in the dynamic model. The model coefficient given by (6) is,

in principle, a function of the three spatial coordinates and time. In the present computations, however, the numerator and denominator were averaged over the homogeneous spanwise direction in order to eliminate ill-conditioning associated with small values of the denominator in (5). The reader is referred to Meneveau et al. (1994) for further discussion.

The numerical method employed for solution of (1) and (2) is the fractional step method (e.g., see Chorin 1967, Kim & Moin 1985). Spatial derivatives are discretized using second-order central differences and the discrete system is time advanced using second order Adams-Bashforth and Crank-Nicholson. The computational domain has a dimension of $2h \times h \times \pi h/10$ in the streamwise, wall-normal, and spanwise directions, respectively, where h is approximately five boundary layer thicknesses. The momentum thickness Reynolds number at the inflow plane is $Re_\theta = 1470$. The grid size is $81 \times 51 \times 33$ corresponding to resolution in wall units of $\Delta x^+ \approx 80$, $\Delta z^+ \approx 32$ (using boundary layer properties evaluated at the outflow boundary). The first grid point in the wall-normal direction is $y^+ < 1$ and hyperbolic stretching is used to cluster points near the wall.

Test filtering, required for computation of the model coefficient C , was performed in x - z planes using tophat filtering in physical space together with Simpson's rule for numerical integration. The time step was one viscous time scale $\Delta t = \nu/u_\tau^2$ and data were sampled over a period of approximately 100 inertial timescales h/U_∞ .

On the top surface of the computational domain the boundary conditions are

$$\bar{u} = U_\infty, \quad \bar{v} = U_\infty \frac{d\delta^*}{dx}, \quad \frac{\partial \bar{w}}{\partial y} = 0, \quad (7)$$

where δ^* is the boundary layer displacement thickness. At the exit plane a convective boundary condition of the form $\partial \bar{u}_i / \partial t + c \partial \bar{u}_i / \partial x = 0$, is applied (c is the local convection velocity, Han et al. 1983) together with a correction on the streamwise velocity to ensure global mass conservation.

2.2. Generation of inflow turbulence

The governing equations (1) and (2) were solved using large eddy simulation and the dynamic eddy viscosity model as described in §2.1. Since it is desired to obtain the properties of a spatially-developing boundary layer, a turbulent velocity profile must be specified at the inflow boundary of the computational domain. In this section the methodology for specifying the inflow profile is described.

As discussed in §1, periodic boundary conditions are not appropriate for the streamwise coordinate since flow field statistics are not homogeneous in this direction. However, as is well known, scaling

laws exist for the turbulent boundary layer, i.e., the mean velocity obeys the law of the wall in the inner region and the defect law in the outer region. Further, the turbulence intensities also scale with y^+ in the inner region and y/δ in the outer region and both the intensities and mean profile exhibit slow variation in the streamwise direction. Thus, in the inner region the mean velocity and turbulence intensities should exhibit slow variation along a line of constant y^+ and constant y/δ in the outer region. These scalings then suggest the existence of a similarity coordinate, proportional to y^+ near the wall and y/δ near the boundary layer edge, along which the mean-flow and intensities vary slowly in x . Since the normalized mean-flow and turbulence intensities are assumed constant along these lines, the similarity coordinate also minimizes streamwise inhomogeneity. Thus, as shown by Spalart (1988), it is then possible to consider a turbulent signal as being comprised of a "slow" and "fast" part. Along the similarity coordinates the first derivative of a "slow" variable (e.g., the mean flow and turbulence intensities) is assumed constant while periodic boundary conditions are appropriate for the "fast" part of the signal.

This decomposition together with the slow variation of the mean and turbulence intensities and periodicity of the "fast" part of a turbulent signal then provide a means for re-introducing turbulence near the outflow boundary back at the inflow plane. The specific steps in the method for generation of inflow turbulence can be summarized as follows:

- i. At the beginning of each time step $t + \Delta t$, the system (x, η, z) is constructed where η is the similarity coordinate (Figure 1). As shown by Spalart (1988) streamwise inhomogeneity is minimized along the η direction, defined as

$$\eta = \frac{y_2^m (10^{-3} y^+) + y^m (y/\delta)}{y_2^m + y^m} \quad (8)$$

where $y_1^+ = 15$, $y_3/\delta = 0.3$, $y_2 = (y_1 y_3)^{1/2}$, and $m = 5.0/\log_{10}(y_3/y_1)$. The expression for η in (8) is consistent with the multiple-scale idea, i.e., near the wall η is a function of the viscous scale ν/u_τ and in the outer region of the boundary layer η depends upon δ with a smooth blending between the two regions.

- ii. The instantaneous velocity field from the previous time step is mapped from the computational Cartesian system (x, y, z) to the newly constructed system (x, η, z) ,

$$\bar{u}_i(x, y, z, t) \rightarrow \bar{u}_i(x, \eta, z, t), \quad (9)$$

(linear interpolation is used to obtain the velocity field in the (x, η, z) system in this work).

- iii. In the (x, η, z) coordinate system the instantaneous velocity field is decomposed into a mean and periodic part

$$\bar{u}_i(x, \eta, z, t) = \langle \bar{u}_i \rangle(x, \eta) + \langle \bar{u}_{i,rms} \rangle(x, \eta) \bar{u}_{i,P}(x, \eta, z, t) \quad (10)$$

where $\langle \cdot \rangle$ refers to averaged values, $\bar{u}_{i,P}(x, \eta, z, t)$ is the time-dependent "fast" part of the signal that is assumed to be periodic in the transformed coordinate system and is scaled by the root-mean-square (rms) fluctuation, $\bar{u}_{i,rms}$.

- iv. The periodic signal $\bar{u}_{i,P}(x_{out}, \eta, z, t)$ from the last time step is then applied at the inflow boundary at the new time step in order to generate a time-dependent signal. The velocity profile at the inflow boundary can then be expressed as

$$\bar{u}_i(x_{in}, \eta, z, t + \Delta t) = \langle \bar{u}_i \rangle(x_{in}, \eta) + \langle \bar{u}_{i,rms} \rangle(x_{in}, \eta) \bar{u}_{i,P}(x_{out}, \eta, z, t). \quad (11)$$

The mean and rms values are the "slow" variables whose first derivatives with respect to x along a similarity line $\eta = \text{const}$ are assumed constant. This permits evaluation of $\langle \bar{u}_i \rangle(x_{in}, \eta)$ from

$$\frac{\langle \bar{u}_i \rangle(x_{mid}, \eta, t) - \langle \bar{u}_i \rangle(x_{in}, \eta, t + \Delta t)}{\langle \bar{u}_i \rangle(x_{out}, \eta, t) - \langle \bar{u}_i \rangle(x_{in}, \eta, t + \Delta t)} = \frac{x_{mid} - x_{in}}{x_{out} - x_{in}}, \quad (12)$$

where the subscript *mid* refer to a streamwise station between the inlet and outlet (averaging is performed along the spanwise direction). The rms values are treated in the same manner. This procedure is quite similar to the long-scale analysis used in Spalart (1988).

- v. The velocity field at the inflow from (11) is transformed from the non-Cartesian system (x, η, z) back to the computational system (x, y, z) using linear interpolation,

$$\bar{u}_i(x_{in}, \eta, z, t + \Delta t) \rightarrow \bar{u}_i(x_{in}, y, z, t + \Delta t). \quad (13)$$

Given the inflow profile (13) at the new time step $t + \Delta t$ the Navier-Stokes equations are then solved as described in §2.1.

It should again be remarked that although the present method utilizes the multiple-scale approach of Spalart (1988), the approach outlined above differs in that the Navier-Stokes equations are solved in this work on a computational domain which is inhomogeneous in the streamwise direction while in the DNS calculations of Spalart (1988), the Navier-Stokes equations themselves are solved in a transformed system in order to apply periodic boundary conditions. Further, properties of spatially-developing boundary layers using the above method are obtained over a continuous range of momentum thickness Reynolds numbers rather than on a station-by-station basis as in Spalart (1988).

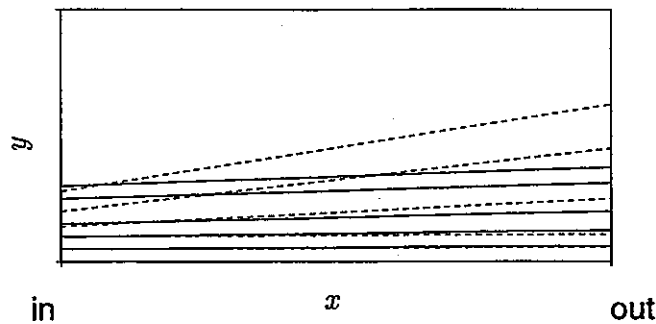


Figure 1: Coordinate system transformation from (x, y, z) to (x, η, z) . Velocity profile near outflow boundary is re-introduced at inflow along η coordinates. — mean-flow streamlines; ---- transformed ($\eta = \text{const}$) coordinates along which spanwise inhomogeneity is minimized.

3. Results

The results to be presented in this section were obtained using the methodology outlined in §2. The primary objective is to demonstrate that a realistic, spatially-developing turbulent boundary layer can be obtained without the need for a prolonged laminar-turbulence transition section. The results presented in this section are not intended to provide a complete survey of the properties of two-dimensional turbulent boundary layers. The reader is referred to Spalart (1988) for a more thorough description of both the statistical and structural properties of canonical flat-plate boundary layers.

Figure 2 shows the computed skin friction together with both experimental data (Purtell et al. 1981, Murlis et al. 1982) and DNS measurements (Spalart 1988, Rai & Moin 1993). The present LES is more similar to the simulations of Rai & Moin in that both simulate a spatially developing flow over a continuous range of Re_θ rather than the DNS of Spalart in which the flow field at a single Re_θ is obtained from a single three-dimensional, time-dependent computation. It may be observed from Figure 2 that in the LES there is essentially no adjustment of the skin friction near the inflow boundary resulting from the recycling used to generate the inflow condition. Any adjustment which occurs is quite different from the extended transition region that is required if random fluctuations are supplied at the inlet. In the latter case, computations (not shown here) demonstrate that the skin friction decreases sharply to the laminar value. The results in Figure 2 also show that overall the LES results are close to the correlation of Coles (1962) and in the best agreement with the experimental data of Murlis et al. (1982). "Noise" in the skin friction data from the LES computations reflect a lack of statistical sample. Shown in Figure 3 is a comparison of the boundary layer shape factor H to existing data

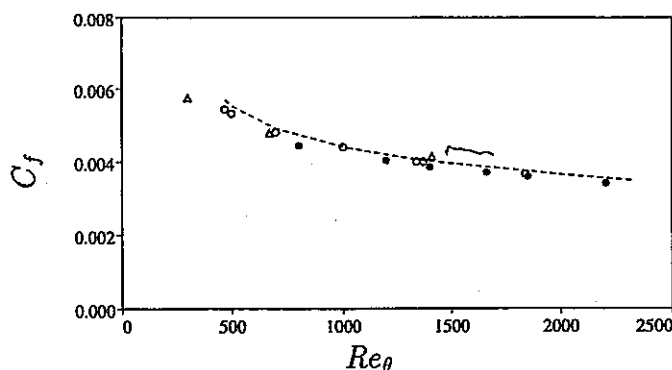


Figure 2: Variation of skin friction coefficient with Reynolds number. — LES; $\circ$ Purtell et al. (1981); $\bullet$ Murlis et al. (1982); Δ Spalart (1988); Rai & Moin (1993); ---- Coles' correlation (1962).

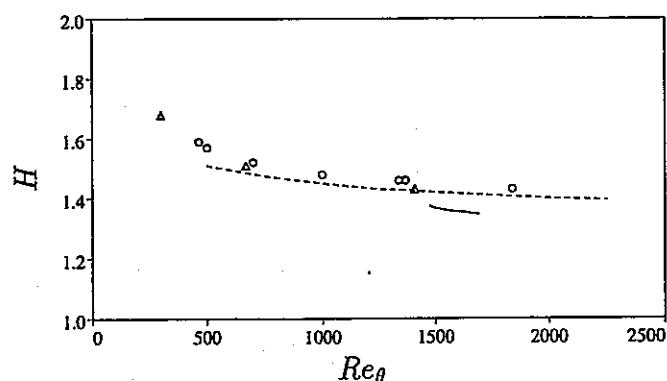


Figure 3: Variation of shape factor with Reynolds number. — LES; $\circ$ Purtell et al. (1981); Δ Spalart (1988), ---- Coles' correlation (1962).

and Coles' (1962) correlation. As can be observed in Figure 3, the boundary layer shape factor in the spatially-developing LES is in good agreement with both experimental measurements and correlations.

Mean-flow profiles in wall coordinates at three streamwise locations from the LES calculations are shown in Figure 4a. The results in Figure 4a are in good agreement with the DNS data of Spalart (1988) as well as universal profiles in the sublayer and logarithmic region. The gradual transition of the mean velocity from a laminar profile to turbulent is absent in the present LES study, indicating that turbulence in the spatially-developing boundary layer is sustained throughout the computational domain. To further demonstrate that the mean-flow from the current LES has the proper universal properties, shown in Figure 4b are profiles of the mean-velocity derivatives in wall units. By definition, a logarithmic layer is a region in which $y^+ d\langle \bar{u} \rangle^+ / dy^+$ is constant and equal to $1/\kappa$, where κ is the Von Karman constant. At high Reynolds numbers this region is extended and bounded on both

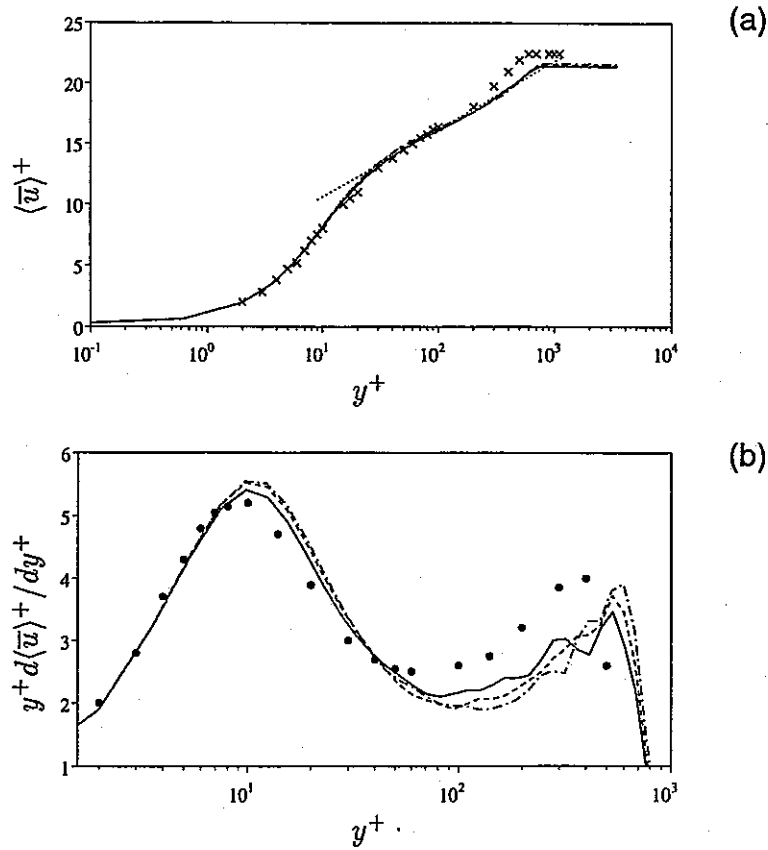


Figure 4: Mean flow distributions. (a) mean velocity, (b) mean velocity derivative. LES: — $Re_\theta = 1520$, ---- $Re_\theta = 1590$, --- $Re_\theta = 1650$; $\times$, $\bullet$ Spalart (1988); ---- $\langle \bar{u} \rangle^+ = 2.44 \ln y^+ + 5.2$ (in a).

sides by regions in which $y^+ d\langle \bar{u} \rangle^+ / dy^+$ takes on values larger than $1/\kappa$. The logarithmic regions shown in Figure 4b are in reasonable agreement with the DNS data of Spalart (1988), the Von Karman constant obtained from Figure 4b is about 2.25, smaller than the classical value 2.44.

Profiles of the resolved turbulence intensities in wall coordinates are presented in Figure 5 from three streamwise locations in the LES calculations. Included for comparison are the DNS data of Spalart (1988) as well as the results of Rai & Moin (1993). It is again noted that the current LES results bear a greater similarity to those of Rai & Moin. For y^+ between approximately 8 and 100, the streamwise intensity from the LES calculations are in very good agreement with the DNS results of Rai & Moin and both are larger than the measurements of Spalart (1988). As also observed by Tsai & Leslie (1990), the wall-normal and spanwise fluctuations show greater attenuation near the wall, possibly due to the use of coarse grids (see also Rai & Moin 1993). In global coordinates the profiles of the streamwise intensities are also in good overall agreement with the experimental

data of Murlis et al. (1982) (Figure 6a).

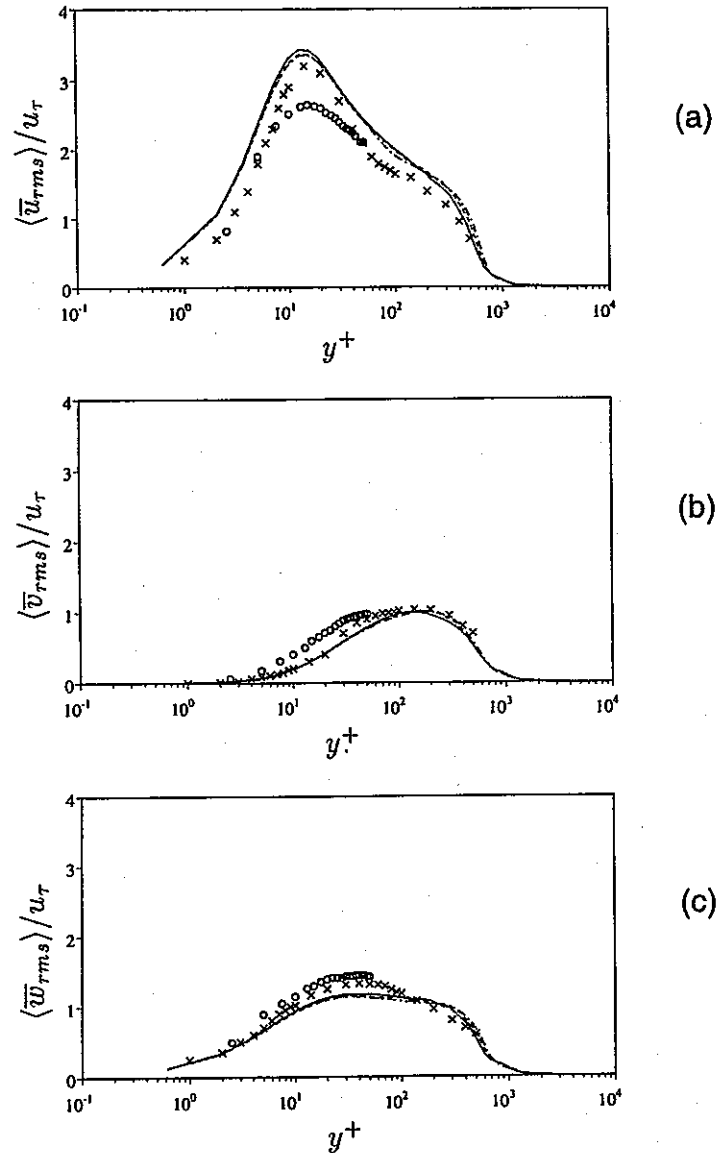


Figure 5: Resolved rms velocity fluctuations. (a) streamwise, (b) wall-normal, (c) spanwise. LES: — $Re_\theta = 1520$, ---- $Re_\theta = 1590$, —·— $Re_\theta = 1650$; o (unfiltered) $Re_\theta = 1410$, Spalart (1988); × (unfiltered) $Re_\theta = 1350$, Rai & Moin (1993).

Comparison of triple correlations $\langle u''^2 v'' \rangle$ and $\langle u'' v''^2 \rangle$ to the experimental data of Murlis et al. (1982) are shown in Figure 7. As may be observed from the Figure, the general shape of the triple products are consistent with the experimental measurements. In the outer region of the boundary layer, $y/\delta > 0.5$, the agreement is good. The fact that the general behavior of higher-order statistics is reasonably well produced in the LES calculations is another indicator that the current method of inflow generation provides a means for calculation

of realistic spatially-developing boundary layers.

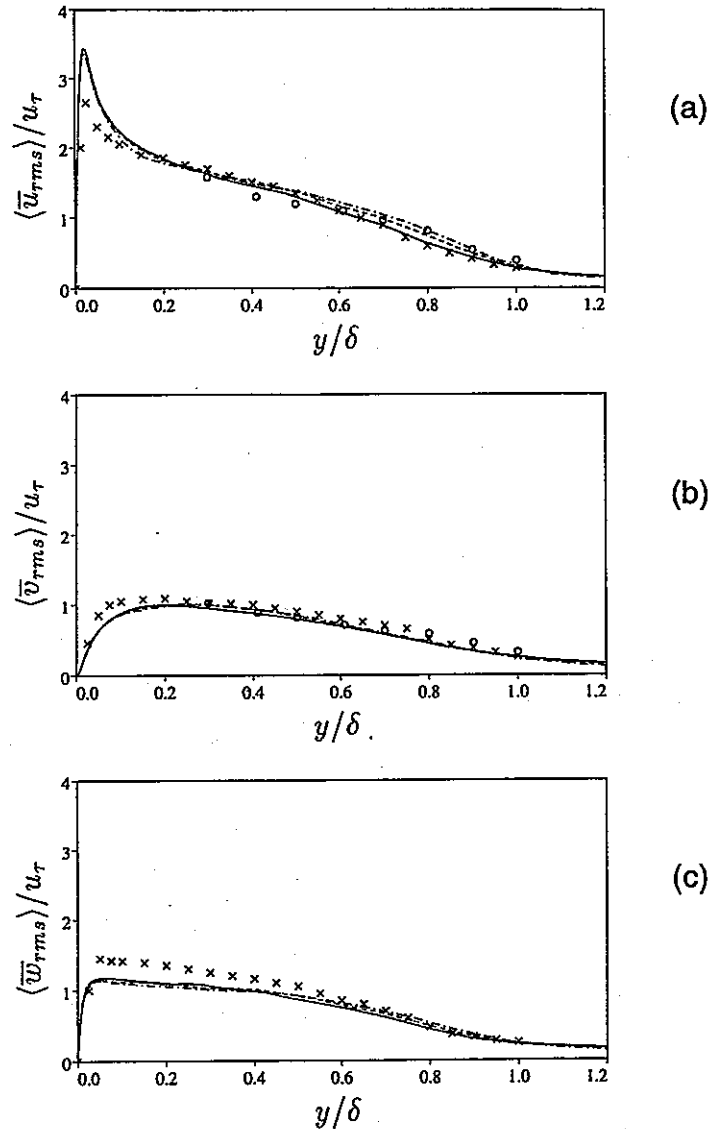


Figure 6: Resolved rms velocity fluctuations. (a) streamwise, (b) wall-normal, (c) spanwise. LES: — $Re_\theta = 1520$, ---- $Re_\theta = 1590$, —·— $Re_\theta = 1650$; $\times$ (unfiltered) $Re_\theta = 1410$, Spalart (1988); $\circ$ $Re_\theta = 1368$, Murlis et al. (1982).

Contours of the instantaneous streamwise velocity field in representative x - y and y - z planes are shown in Figures 8a and 8b, respectively. As expected, these figures show the sharp interface between the turbulent and irrotational regions of the flow; boundary layer growth near the inflow plane in Figure 8a does not appear to be adversely affected by the inflow generation method. Figure 8c offers further evidence that the spatially-developing flow calculated in this work reproduces the essential features observed in experiments and DNS. Shown in

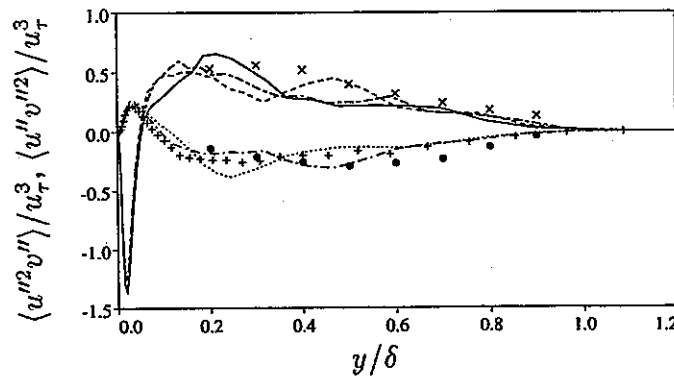


Figure 7: Triple correlations. $\langle u''^2 v'' \rangle / u_\tau^3$: LES: — $Re_\theta = 1520$, ---- $Re_\theta = 1590$, -.- $Re_\theta = 1650$; $\times$ Murlis et al. (1982). $\langle u'' v''^2 \rangle / u_\tau^3$: LES: $Re_\theta = 1520$, — $Re_\theta = 1590$, + $Re_\theta = 1650$; $\bullet$ Murlis et al. (1982).

Figure 8c are contours of the fluctuating streamwise velocity component at $y^+ = 5$. The solid lines represent positive velocities (greater than the mean) while dotted lines represent negative velocities (less than the mean). As expected, the contours near the wall show the streaky structure characteristic of turbulent boundary layers.

4. Summary

A method for generating a physically realistic inflow velocity field for simulation of spatially-developing boundary layers has been presented. The approach developed in this work is based on the multiple-scale analysis of Spalart (1988). By decomposing the velocity profile into a mean and periodic part, turbulent signals near the outflow boundary are recycled along coordinate lines in which streamwise inhomogeneity is minimized. The method is straightforward in implementation and its robustness was demonstrated through large eddy simulation of a flat-plate turbulent boundary layer. LES results demonstrated that boundary layer statistics obtained using this method are in good agreement with experimental data and DNS results. Contours of the instantaneous velocity field also demonstrate that the qualitative nature of turbulent boundary layers, e.g., near-wall streaky structure, is captured. More importantly, the present study demonstrates that turbulence in a spatially-developing boundary layer can be sustained without the need for computation of the development of the flow from an external perturbation.

It should be remarked that the current interest in this method of generating turbulent inflow is to provide a well-defined inflow condition for calculation of complex wall-bounded turbulent flows. It is intended that the approach outlined in this work will be used for pre-computation of a flat-plate turbulent boundary layer. Velocity fields

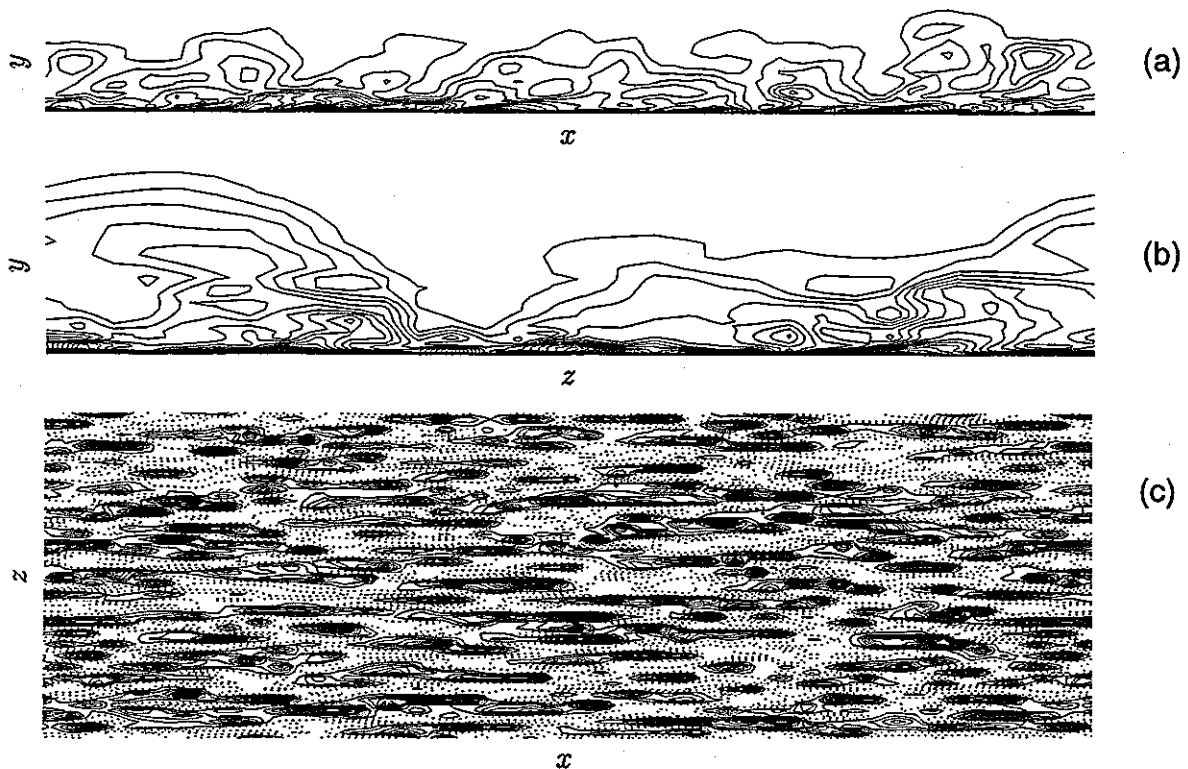


Figure 8: Streamwise velocity, contour levels in (a) and (b): $\bar{u}/U_\infty = 0.05, 0.1, 0.15, \dots$. (a) x - y plane, (b) y - z plane, (c) x - z plane at $y^+ = 5$, — positive values, negative values.

from the pre-computation will then be supplied at the inflow boundary of calculations of complex flows (e.g., curved boundary layers and three-dimensional boundary layers). Preliminary investigations have shown that the overhead required for these precomputations, whether run together with the simulation of the complex flow or prior to simulation of the complex flow and then stored, is not prohibitive.

Finally, the multiple-scale analysis of Spalart (1988) was used in this work to develop a scheme for inflow generation and computation of zero-pressure gradient turbulent boundary layers. It should be remarked that similar analyses could be applied to other flows, e.g., adverse or favorable pressure gradients leading to a different transformed coordinate η , provided that the pressure gradient is mild enough such that boundary layer quantities vary slowly in the mean-flow direction.

Acknowledgements

Financial support for this work was provided from the Office of Naval Research and Air Force Office of Scientific Research. XW and KDS gratefully acknowledge support under ONR grants N-00014-94-1-

0047 and N-00014-94-1-1053. TSL acknowledges support from ONR grant N-00014-91-J-4072 and AFOSR grant F49620-92-J0003. The research was conducted using the resources of the Cornell Theory Center (CTC) and the authors gratefully acknowledge the technical support and assistance of the staff at CTC.

References

1. Akselvoll, K. & Moin, P., "Application of the dynamic localization model to large eddy simulation of turbulent flow over a backward-facing step", *Engineering Applications to Large-eddy Simulation*, (Piomelli, U. & Ragab, S. eds.), ASME, New York, 1993.
2. Chorin, A.J., "A numerical method for solving incompressible viscous flow problems", *J. Comp. Physics*, Vol. 2, pp. 745-762 (1967).
3. Coles, D.E., "The turbulent boundary layer in a compressible fluid", Rep. R403-PR, ARC 24473, Appendix A: A manual of experimental boundary layer practice for low speed flow, 1962.
4. Germano, M., Piomelli, U., Moin, P. & Cabot, W.H., "A Dynamic Subgrid-scale Eddy Viscosity Model", *Phys. Fluids A*, Vol. 3, pp. 1760-1765 (1991).
5. Han, T.Y., Meng, J.C.S. & Innis, G.E., "An open boundary condition for incompressible stratified flows", *J. Comp. Physics*, Vol. 49, pp. 276-297 (1983).
6. Kaltenbach, H., "Large eddy simulation of flow in a plane, asymmetric diffuser", *Annual Research Briefs, Center for Turbulence Research*, pp. 101-109 (1993).
7. Kim, J. & Moin, P., "Application of a fractional-step method to incompressible Navier-Stokes equations", *J. Comp. Physics*, Vol. 59, pp. 308-323 (1985).
8. Kim, J., Moin, P. & Moser, R.M., "Turbulence statistics in fully developed channel flow at low Reynolds number", *J. Fluid Mech.*, Vol. 177, pp. 133-166 (1987).
9. Lee, S., Lele, S.K. & Moin, P., "Simulation of spatially evolving turbulence and the applicability of Taylor's hypothesis in compressible flow", *Phys. Fluids A*, Vol. 4, pp. 1521-1530 (1992).
10. Lilly, D.K., "A proposed modification of the Germano subgrid-scale closure method", *Phys. Fluids A*, Vol. 4(3), pp. 633-635 (1992).

11. Lund, T.S., "Large eddy simulation of a boundary layer with concave streamwise curvature", Annual Research Briefs - 1993, Center for Turbulence Research, pp. 91-100 (1993).
12. Moin, P. & Kim, J., "Numerical Investigation of Turbulent Channel Flow", J. Fluid Mech., Vol. 118, pp. 341-377 (1982).
13. Meneveau, C., Lund, T.S. & Cabot, W., "A Lagrangian dynamic subgrid-scale model of turbulence", Proc. of the 1994 Summer Program, Center for Turbulence Research, pp. 271-300 (1994).
14. Murlis, J., Tsai, H.M. & Bradshaw, P., "The structure of turbulent boundary layers at low Reynolds number", J. Fluid Mech., Vol. 122, pp. 13-56 (1982).
15. Purtell, L.P., Klebanoff, P.S. & Buckley, F.T., "Turbulent boundary layers at low Reynolds numbers", Phys. Fluids A, Vol. 24, pp. 802-811 (1981).
16. Rai, M.M. & Moin, P., "Direct numerical simulation of transition and turbulence in a spatially evolving boundary layer", J. Comp. Physics, Vol. 109, pp. 169-192 (1993).
17. Spalart, P.R., "Direct simulation of a turbulent boundary layer up to $Re_\theta = 1410$ ", J. Fluid Mech., Vol. 187, pp. 61-98 (1988).
18. Tsai, H.M. & Leslie, D.C., "Large eddy simulation of a developing turbulent boundary layer at a low Reynolds number", Int. J. Num. Methods in Fluids, Vol. 10, pp. 519-555 (1990).

Author Biographical Sketches

Dr. Xiaohua Wu received his Ph.D. in Mechanical Engineering from the University of Manitoba in November 1993. Since then he has been a postdoctoral associate in the Mechanical Engineering Department at the University of Vermont. His primary research interests include large eddy simulation of complex turbulent flows and hot-wire measurements of perturbed turbulent boundary layers.

Prof. Kyle Squires received his Ph.D. in Mechanical Engineering from Stanford University in September 1990. He then joined the Center for Turbulence Research as a postdoctoral associate, working on large eddy simulation and subgrid-scale modeling of both incompressible and compressible turbulence. Since September 1991 he has been an Assistant Professor in the Mechanical Engineering Department at the University of Vermont. He has also been a Senior Research Fellow at the Institute of Industrial Science at the Tokyo University and a Faculty Fellow of the American Society for Engineering Education. His primary research interests are direct numerical simulation and large eddy simulation of both single- and two-phase turbulent flows.

Dr. Thomas Lund received his Ph.D. in Aeronautics and Astronautics from Stanford University in 1987. From 1987 to 1989 he was an Assistant Professor of Aeronautics and Astronautics at Purdue University. During 1990 he was employed by Rockwell International. From 1990 to the present he has been a research associate within the Center for Turbulence Research at Stanford University. His main areas of interest are numerical simulation of turbulence and turbulence modeling. Presently he is focusing his efforts on large eddy simulation of complex turbulent flows that arise in engineering applications.

Copyright © 1995 by the Association for Computing Machinery, Inc. (ACM).

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that new copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted.

To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Publications Dept, ACM Inc., via fax at +1 (212) 869-0481, or via email at permissions@acm.org.

Parallelizing Navier-Stokes Computations on a Variety of Architectural Platforms

D. N. Jayasimha *

Dept. of Computer and Information Science
The Ohio State University, Columbus, OH 43210
jayasim@cis.ohio-state.edu

M. E. Hayder

ICOMP, Ohio Aerospace Institute
NASA Lewis Research Center
Cleveland, OH 44142
fshyder@icomp.lerc.nasa.gov

S. K. Pillay

Scientific Engg. Computing Solutions Office
NASA Lewis Research Center
Cleveland, OH 44142
spillay@lerc.nasa.gov

*Part of this work was done while this author was a Visiting Senior Research Associate at NASA Lewis Research Center during 1993-94.

Abstract

We study the computational, communication, and scalability characteristics of a Computational Fluid Dynamics application, which solves the time accurate flow field of a jet using the compressible Navier-Stokes equations, on a variety of parallel architectural platforms. The platforms chosen for this study are a cluster of workstations (the LACE experimental testbed at NASA Lewis), a shared memory multiprocessor (the Cray YMP), distributed memory multiprocessors with different topologies— the IBM SP and the Cray T3D. We investigate the impact of various networks, connecting the cluster of workstations, on the performance of the application and the overheads induced by popular message passing libraries used for parallelization. The work also highlights the importance of matching the memory bandwidth to the processor speed for good single processor performance. By studying the performance of an application on a variety of architectures, we are able to point out the strengths and weaknesses of each of the example computing platforms.

1 Introduction

Numerical simulations play an important role in the investigation of physical processes associated with many important problems. The suppression of jet exhaust noise is one such problem which will have a great impact on the success of the High Speed Civil Transport plane. The radiated sound emanating from the jet can be computed by solving the full (time-dependent) compressible Navier-Stokes equations. This computation can, however, be very expensive and time consuming. The difficulty can be partially overcome by limiting the solution domain to the near field where the jet is nonlinear and then using acoustic analogy (see [1]) to relate the far-field noise to the near-field sources. This technique requires obtaining the time-dependent flow field. In this study we concentrate on such flow fields near the nozzle exit. We solve the Navier Stokes equations to compute time accurate flow fields of a supersonic axisymmetric jet. Our code is computationally very intensive and requires many hours of CPU time on the Cray Y-MP. With the advent of massively parallel processors and networks of workstations (NOWs), scientists now have the opportunity to parallelize computationally intensive codes and reduce turnaround time at a fraction of the cost of traditional supercomputers. Recognizing this, a number of researchers [2, 3, 4, 5] have studied CFD (Computational Fluid Dynamics) applications on specific parallel architectures. Our goal in this study is to implement the numerical model derived from the CFD application described above on a variety of parallel architectural platforms.

The platforms chosen for this study, all from the NASA Lewis Research Center, represent a spectrum of parallel architectures that have been proposed to solve computationally intensive problems: a shared memory vector multiprocessor (the Cray YMP), two distributed memory multiprocessors with different topologies— the IBM SP and the Cray T3D, and a cluster of workstations connected via many networks (the Lewis Advanced Cluster Environment (LACE) [6] experimental testbed). One important architecture that has not been considered in our study is cache-coherent, massively parallel processors typified by the DASH architecture [7].

Architectures such as LACE (an example of NOW) are becoming increasingly popular because they show promise as a low cost alternative to expensive supercomputers and massively parallel processors. We have therefore laid more emphasis on this aspect of the study in this paper. An earlier paper by the authors [8] presented the initial results of a study on LACE and the Y-MP. This paper differs from the earlier one in two important aspects: i) It is comprehensive covering a gamut of architectures. ii) It focuses on the relationship of the computation and communication characteristics of the application, and hence its performance, to the architectural aspects of the networks and the processing nodes.

In the next two sections we briefly discuss the governing equations and the numerical model of the application. Section 4 has a discussion of the parallel architectures used in the study and the tools used for parallelizing the application. The parallelization of the application is the subject of Section 5. Section 6 describes the experimental methodology. Section 7 presents a

detailed discussion of the results. The paper concludes with a brief discussion of the lessons learned from this study.

2 Governing Equations

We solve the Navier-Stokes and the Euler equations to compute flow fields of an axisymmetric jet. The Navier-Stokes equations for such flows can be written, in polar coordinates as

$$LQ = S$$

$$\frac{\partial Q}{\partial t} + \frac{\partial F}{\partial x} + \frac{\partial G}{\partial r} = S$$

where

$$Q = r \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ E \end{pmatrix}$$

$$F = r \begin{pmatrix} \rho u \\ \rho u^2 - \tau_{xx} + p \\ \rho uv - \tau_{xr} \\ \rho uH - u\tau_{xx} - v\tau_{xr} - \kappa T_x \end{pmatrix}$$

$$G = r \begin{pmatrix} \rho v \\ \rho uv - \tau_{xr} \\ \rho v^2 - \tau_{rr} + p \\ \rho vH - u\tau_{xr} - v\tau_{rr} - \kappa T_r \end{pmatrix}$$

$$S = \begin{pmatrix} 0 \\ 0 \\ p - \tau_{\theta\theta} \\ 0 \end{pmatrix}$$

F and G are the fluxes in the x and r directions respectively, and S is the source term that arises in the cylindrical polar coordinates, τ_{ij} are the shear stresses and κT_j are the heat fluxes. In the above equations p , ρ , u , v , T , e and H denote the pressure, density, axial and radial velocity components, temperature, total energy and enthalpy. For a perfect gas,

$$E = \frac{p}{(\gamma - 1)} + \frac{1}{2}\rho(u^2 + v^2)$$

$$H = \frac{E + p}{\rho}$$

where γ is the ratio of specific heats. One obtains the Euler equations from the above equations, by setting κ and all τ_{ij} equal to zero.

3 Numerical Model

We use the fourth order MacCormack scheme, due to Gottlieb and Turkel [9], to solve the Navier-Stokes and the Euler equations. This scheme uses predictor and corrector steps to compute time accurate solutions. It uses one sided differences (forward or backward) to compute spatial derivatives at each predictor or corrector step. For the present computations, the operator L in the equation $LQ = S$ or equivalently $Q_t + F_z + G_r = S$ is split into two one-dimensional operators and the scheme is applied to these split operators. We define L_1 as a one dimensional operator with a forward difference in the predictor and a backward difference in the corrector. Its symmetric variant L_2 uses a backward difference in the predictor and a forward difference in the corrector. The predictor step in L_1Q for the one dimension model/split equation $Q_t + F_z = S$ is written as

$$\bar{Q}_i = Q_i^n + \frac{\Delta t}{6\Delta x} \{7(F_{i+1}^n - F_i^n) - (F_{i+2}^n - F_{i+1}^n)\} + \Delta t S_i$$

and the corrector step as

$$Q_i^{n+1} = \frac{1}{2} [\bar{Q}_i + Q_i^n - \frac{\Delta t}{6\Delta x} \{7(\bar{F}_i - \bar{F}_{i-1}) - (\bar{F}_{i-1} - \bar{F}_{i-2})\} + \Delta t S_i]$$

Similarly in L_2 the predictor step is

$$\bar{Q}_i = Q_i^n - \frac{\Delta t}{6\Delta x} \{7(F_i^n - F_{i-1}^n) - (F_{i-1}^n - F_{i-2}^n)\} + \Delta t S_i$$

and the corrector step is

$$Q_i^{n+1} = \frac{1}{2} [\bar{Q}_i + Q_i^n + \frac{\Delta t}{6\Delta x} \{7(\bar{F}_i - \bar{F}_{i+1}) - (\bar{F}_{i+1} - \bar{F}_{i+2})\} + \Delta t S_i]$$

This scheme becomes fourth-order accurate in the spatial derivatives when alternated with symmetric variants [9]. For our computations, the one dimensional sweeps are arranged as

$$Q^{n+1} = L_{1x} L_{1r} Q^n$$

$$Q^{n+2} = L_{2r} L_{2x} Q^{n+1}$$

This scheme is used for the interior points. In order to advance the scheme near boundaries the fluxes are extrapolated outside the domain to artificial points using a cubic extrapolation to compute the solution on the boundary. We use the characteristic boundary condition at the outflow. In our implementation, we solve the following set of equations to get the solution at the new time for all boundary points.

$$p_t - \rho c u_t = 0$$

$$p_t + \rho c u_t = R_2$$

$$p_t - c^2 \rho_t = R_3$$

$$v_t = R_4$$

where R_i is determined by which variables are specified and which are not. Whenever the combination is not specified, R_i is just those spatial derivatives that come from the Navier-Stokes equations. Thus R_i contains viscous contributions even though the basic format is based on inviscid characteristic theory. In implementing these differential equations we convert them to conservation variables ρ , $m = \rho u$, $n = \rho v$, and E . Assuming an ideal gas,

$$p_t = (\gamma - 1)(E_t + \frac{u^2 + v^2}{2} \rho_t - u m_t - v n_t)$$

$$u_t = \frac{m_t}{\rho} - \frac{u \rho_t}{\rho}$$

$$v_t = \frac{n_t}{\rho} - \frac{v \rho_t}{\rho}$$

For subsonic outflow, we calculate R_2, R_3, R_4 from the Navier-Stokes equations. For supersonic flows, all the R_i at the outflow boundary can be calculated from the Navier-Stokes equations or by extrapolation of all the characteristic variables from the interior. This framework of outflow boundary condition implementation is discussed by Hayder and Turkel [10]. Further discussions of our numerical model including other boundary treatments are given in Hayder et al. [11] and Mankbadi et al. [12].

In this study, we consider a jet with the mean inflow profile

$$\bar{U}_r = U_\infty + (U_c - U_\infty)g_r$$

$$\bar{T}_r = T_c + (T_\infty - T_c)g_r + \frac{\gamma - 1}{2} M_c^2 (1 - g_r)g_r$$

$$g_r = \frac{1}{2} [1 + \tanh(\frac{\frac{1}{r} - r}{4\theta})]$$

where θ is the momentum thickness. The subscripts c and ∞ refer to the centerline and free stream values respectively. At inflow, we assume the radial velocity is zero and the static pressure is constant. The size of our computational domain is 50 radii in the axial direction and 5 radii in the radial direction. We excite the inflow profile at location r and time t as

$$U(r, t) = \bar{U}(r) + \epsilon \text{Re}(\hat{U} e^{i n S_t t})$$

$$P(r, t) = \bar{P}(r) + \epsilon \text{Re}(\hat{P} e^{i n S_t t})$$

$$\rho(r, t) = \bar{\rho}(r) + \epsilon \text{Re}(\hat{\rho} e^{i n S_t t})$$

$$V(r, t) = \epsilon \text{Re}(\hat{V} e^{i n S_t t})$$

$\hat{U}$, $\hat{V}$, $\hat{\rho}$ and $\hat{P}$ are the eigenfunctions of the linearized equations with the same mean flow profile, ϵ is the excitation level and S_t is the Strouhal number.

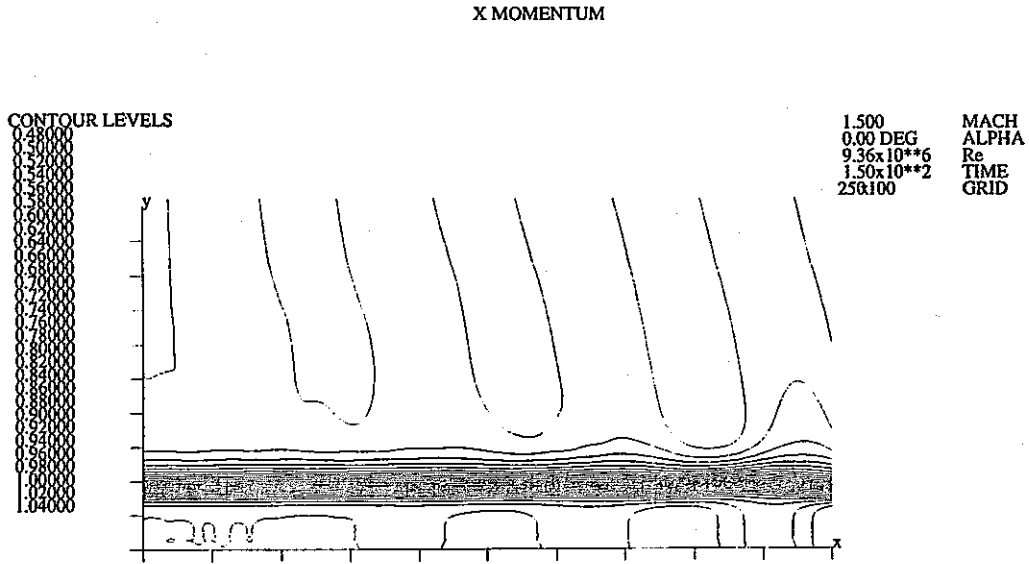


Figure 1: Axial momentum in an excited axisymmetric jet

We consider a case with $\frac{U_\infty}{U_c} = \frac{1}{4}$, $\frac{T_\infty}{T_c} = \frac{1}{2}$, momentum thickness, $\theta = \frac{1}{8}$ and Strouhal number, $S_t = \frac{1}{8}$. The jet center Mach number is 1.5 while the Reynolds number based on the jet diameter is 1.2 million. Our present mean flow and eigen function profiles are same as those in Scott et al. [16].

In Figure 1 we show a contour plot of axial momentum from the solution of the Navier Stokes equations. A grid of size 250x100 was used in this computation. This result was obtained after about 16,000 time steps. For all other results in this paper, we have used the same grid, but ran the experiments for 5000 time steps to keep the computing requirements reasonable.

4 Parallel Computing Platforms

This section contains a brief discussion of the various platforms used in the study together with the parallelization tools used.

4.1 NOW

The LACE testbed is continually upgraded. The present configuration has 32 RS6000 processor nodes (nodes 1-32) and an RS6000/Model 990 (node 0) which is the file server. These nodes or subsets of them are connected through various networks with different speed and connection characteristics. All the nodes are connected through two Ethernet networks (10 Mbits/sec (Mbps)), one of them is for general use and the other is dedicated to "parallel" use.

Nodes 9–24 are interconnected through a FDDI interface with a peak bandwidth of 100 Mbps. It is convenient, for our purposes, to consider the nodes to be partitioned into a lower half (nodes 1–16) and an upper half (nodes 17–32). The lower half has RS6000/Model 590 CPUs (the CPU has a 66.5 MHz clock, 256KB data- and 32KB instruction caches) with the following networks interconnecting the nodes: an ATM network capable of a peak bandwidth of 155 Mbps and IBM's ALLNODE switch, referred to as ALLNODE-F (for fast), capable of a peak throughput of 64 Mbps per link. The upper half has the slower RS6000/Model 560 CPUs (the CPU has a 50 MHz clock, 64KB data- and 8KB instruction caches) and is connected through IBM's ALLNODE prototype switch, referred to as ALLNODE-S (for slow), capable of a peak throughput of 32 Mbps per link. The ALLNODE switch is a variant of Omega interconnection network and is capable of providing multiple contentionless paths between the nodes of the cluster (a maximum of 8 paths can be configured between source and destination processors). The present setup does not permit the use of more than 16 processors using the faster networks. The nodes have varying main memory capacity (64 MB, 128 MB, 256 MB, and 512 MB). We have used the popular PVM (Parallel Virtual Machine) message passing library (version 3.2.2) to implement our parallel programs. We will refer to the LACE cluster with RS6000/Model 560 processors as the LACE/560 and those with the RS6000/Model 590 processors as the LACE/590.

4.2 Shared Memory Architecture

We used the Cray Y-MP/8, which has eight vector processors, for this study. The Cray Y-MP/8 has a peak rating of approximately 2.7 GigaFLOPS. It offers a single address space and the communication between processes executing on different processors is through shared variables. We parallelized the application by using explicit DOALL directives in addition to exploiting the features of the parallelizing compiler on the Cray.

4.3 Distributed Memory Architecture

We parallelized the application on two distributed memory multiprocessors—the IBM SP1 and the Cray T3D. The IBM SP1 has 16 processing nodes (the CPU at each node is a RS6K/370—the CPU has a 50 MHz clock, 32KB data and instruction caches). The original system has been software upgraded to make it function like a SP2. We will refer to this system as the IBM SP in the paper. The nodes of the SP are interconnected through a variant of the Omega network [14]. This network, similar in topology to ALLNODE, permits multiple contentionless paths between nodes. We parallelized the application using MPL (Message Passing Library), IBM's native message passing library and PVMe, a customized version of PVM (version 3.2) developed by IBM for the SP.

The Cray T3D is also a distributed memory multiprocessor with the topology of a three dimensional torus [15]. The machine used in our study has 64

nodes ($8 \times 4 \times 2$) (each node has a CPU with a clock speed of 150 MHz and a direct mapped cache of 8KB) of which only 16 were available in single user mode. Though the T3D supports multiple programming models, we programmed the machine using the message passing paradigm resorting to Cray's customized version of PVM (version 3.2).

5 Parallelization

The factors which affect parallel performance are listed below.

1. Single processor performance: we will explain various optimizations which resulted in 80% improvement in performance.
2. Communication cost: this cost depends on both the number of communication startups and the volume of data communicated. Usually, the startup cost is 2-3 orders of magnitude higher than the per word transfer cost. One method to reduce the effect of startup cost is to **group** data to be communicated into long vectors.
3. Overlapped communication and computation: it is desirable that communication be overlapped with computation as far as possible. Increasing the amount of overlapping, however, usually leads to **finer** granularity of communication which then leads to a higher number of startups.
4. Bursty communication: such communication could overwhelm the network's throughput capacity temporarily leading to increased communication cost and process waiting time. Some amount of burstiness is inevitable since parallel programs are usually written in the SPMD (single program multiple data) style. There is also usually an inverse relationship between bursty data and the number of communication startups.

From the above discussion it is clear that there is a subtle relationship among communication startup cost, overlapping communication with computation, and bursty communication. After some experimentation, we chose to decompose the domain by **blocks** along the axial direction only.

For the solution of Navier-Stokes equations, hereafter referred to as **Navier-Stokes**, each internal subdomain exchanges its two flux values, velocity, and temperature along the boundary with its appropriate (left or right) neighbor. To reduce the number of communication startups, we group communication-first, **all** the velocity and temperature values along a boundary are calculated and then packaged into a single **send**. We use a similar scheme for the flux values that need to be communicated.

The computational and communication requirements of the application are shown in Table 1. It is seen that the solution of Euler equations, hereafter referred to as **Euler**, has roughly 50% of the computation and roughly 75% of the communication requirements of **Navier-Stokes**. Note that the communication requirements are shown on a **per processor** basis. To give some idea of the effects of communication, consider **Navier-Stokes** to be executing on a network of 10 workstations connected via Ethernet. Assume a reasonable throughput of 20 MFLOPS per processor and the maximum throughput of

Table 1: Application Characteristics

Appln	Total Comp. (in FP Ops ($\times 10^6$))	Comm./Processor	
		Start-ups	Volume (MB)
N-S	145,000	80,000	125 (1000Mb)
Euler	77,000	60,000	95 (760Mb)

Table 2: Computation-Communication Ratios

No. of Procs.	FPs/Byte		FPs/Start-up	
	Nav-Stokes	Euler	Nav-Stokes	Euler
1	∞	∞	∞	∞
2	580	405	906K	642K
4	290	203	453K	321K
8	145	101	227K	161K
16	73	51	113K	80K

10 Mbps for Ethernet. The computation time will then be approximately 725 seconds ($145,000/(10 \times 20)$) while a **lower bound** on the communication time, ignoring the effect of startups, is 1000 seconds ($1000 \times 10/10$)! Table 2 shows the ratio of computation to communication for the application in units of floating point operations/byte transferred per processor and floating point operations/startup per processor.

The parallelization on the Cray Y-MP was done differently (it was much easier also) since it is a shared memory architecture: we did some hand optimization to convert some loops to parallel loops, used the DOALL directive, and partitioned the domain along the orthogonal direction of the sweep to keep the vector lengths large and to avoid non-stride access to most of the variables.

6 Experimental Methodology

The performance indicator is the total execution time for **Navier-Stokes** and **Euler**. All experiments were conducted in single user mode. In almost all the experiments, **Navier-Stokes** and **Euler** show similar trends; hence, unless otherwise mentioned, quantitative comments refer to **Navier-Stokes**.

Experiments using a single processor were done on an IBM RS6K (Model 560) workstation of LACE. The performance of the original code for both applications is shown in Figure 2.

We found that most parts of the application were limited by the poor performance of the memory hierarchy involving the cache and the main memory.

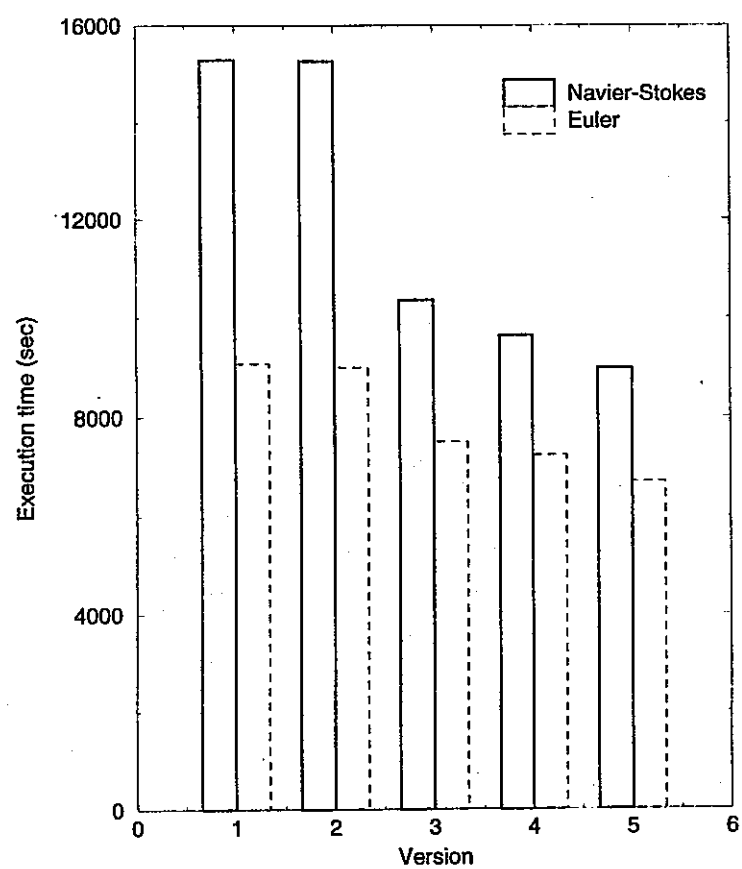


Figure 2: Execution time on a single processor (RS6000/560)

Improved cache performance was the key and this was achieved by accessing arrays in **stride-1** fashion wherever possible (using the loop interchange optimization). The modified program, called Version 3 (the optimizations were performed in a different order than presented in the paper), resulted in this version running faster by approximately 50%, compared to Version 2. We experimented with a number of other modifications, the following of which yielded some improvement: better register usage by collapsing multiple COMMON blocks into a single one (Version 5), strength reduction (replace exponentiations by multiplications wherever feasible- Version 2), replace division by multiplication wherever feasible since the former are relatively expensive (a reduction from 5.5×10^9 divisions to 2.0×10^9 was achieved- Version 4). All these optimizations yielded an overall improvement of roughly 80% (from 9.3 MFLOPS to 16.0 MFLOPS) as illustrated in Figure 2. (The optimizations were incorporated in sequence so that Version 5 contains all the above mentioned optimizations).

We parallelized Version 5 on different computing platforms in accordance with the ideas presented in the last section. On each platform, we measured the execution time as a function of the number of processors (up to 8 with Cray Y-MP, up to 16 with LACE, IBM SP, and Cray T3D). One important goal of the study has been to examine the scalability of NOW. Toward this end, we have studied the performance of LACE with four networks of differing characteristics using "off-the-shelf" PVM as the message passing library.

With the IBM SP, we have studied the impact of parallelizing the application with two message passing libraries- IBM's native MPL and a customized version of PVM called PVMe.

In all experiments, wherever feasible, we have separated the execution time into two additive components: **processor busy time** and **non-overlapped communication time**. The processor busy time is itself composed of the actual computation time and the software overheads associated with sending and receiving messages. An accurate separation of these components is not possible, however, unless we have hardware performance monitoring tools. The non-overlapped communication time could also include the idle time of a processor waiting for a message.

Version 5 of the application does not make any special attempts to overlap communication with computation. Version 6 does overlapping by computing the stress and flux components of the interior part of each subdomain while the processor is waiting for the velocity and temperature vectors from its neighbors. As mentioned earlier, the two "flux columns" nearest each boundary are combined into a single send. We have experimented with sending the flux columns one at a time to avoid bursty communication. This variant is called Version 7.

We found that the execution time improvement with Versions 6 and 7 were either minimal or even worse in many experiments. Hence all our experiments were conducted with Version 5. We do mention, however, the impacts of these versions on different networks of LACE.

The next section presents a detailed discussion of the results of our experiments.

7 Results

The execution times of **Navier-Stokes** and **Euler** have been plotted as a function of the number of processors for each computing platform, using a log-log scale to facilitate meaningful presentation.

7.1 Performance of LACE

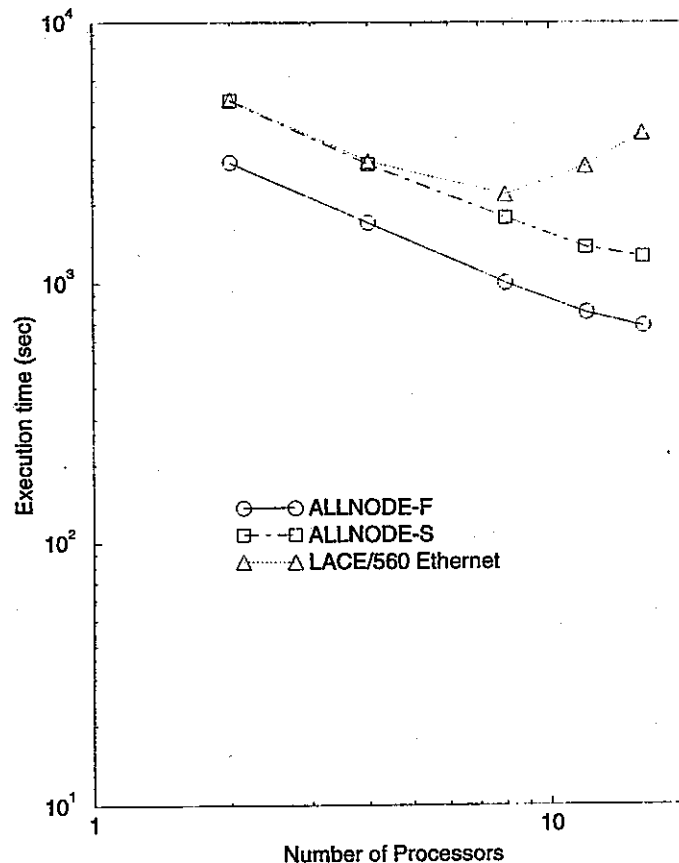


Figure 3: Navier-Stokes execution time on LACE

Figures 3 and 4 show the performance of Navier-Stokes and Euler respectively on different networks of LACE- ALLNODE-F, ALLNODE-S, and the upper-half Ethernet. The performance of the ATM and the FDDI networks are almost identical with ALLNODE-F and ALLNODE-S respectively. Hence the performance of the ATM and FDDI networks are not shown.

The close performance of ALLNODE-F and ATM, and ALLNODE-S and FDDI can be attributed to the following reason: the slower link speed of ALLNODE (64 Mbps/32 Mbps) is balanced by its ability to set up multiple contention-free paths while ATM (155 Mbps) or FDDI (100 Mbps) with their faster links do not permit multiple physical paths in the network.

The execution time falls almost linearly with increasing number of processors with ALLNODE- sublinearity effects begin to show, however, beyond 12

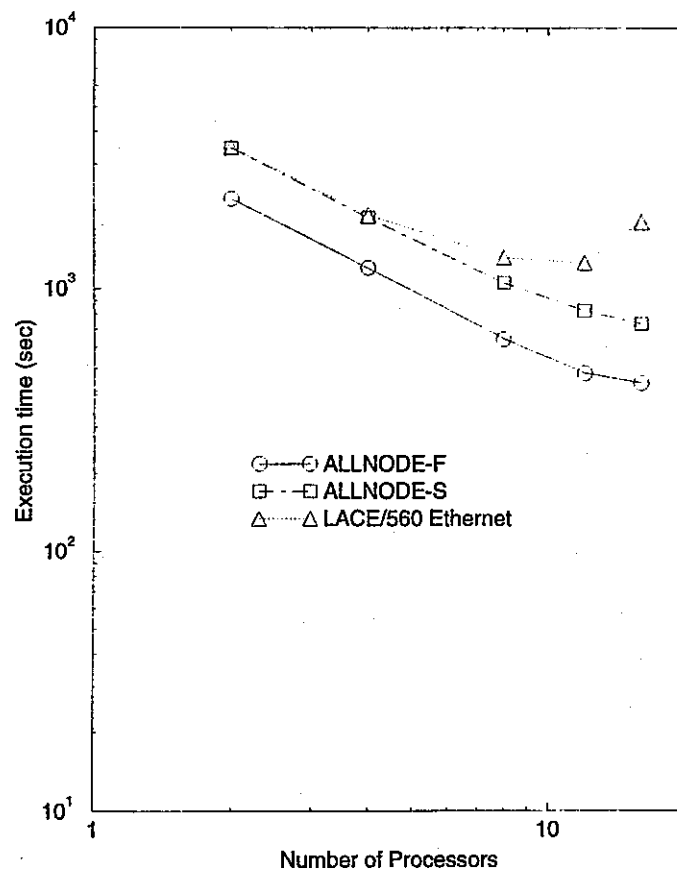


Figure 4: Euler execution time on LACE

processors. The LACE setup does not permit the use of more than 16 processors using the faster networks. ALLNODE-F is about 70%-80% faster than ALLNODE-S. This can be attributed to both an improved network (which is twice as fast) and the superior performance of the 590 model (33% faster clock, data and instruction caches which are 4 times bigger, and memory bus which is 4 times wider than the 560- these contribute to faster instruction execution, better cache hit ratios, and lower cache miss penalty respectively).

Not surprisingly, Ethernet performance reaches its peak at 8 processors for Navier-Stokes and at 10 processors for Euler. Beyond this, the communication requirements of the application overwhelm the network. The inability of Ethernet to handle traffic beyond 8 processors is shown by the following simple argument: Table 2 shows that with 8 processors, Navier-Stokes, on the average, produces a byte for communication after it has completed 145 floating operations on the average. Consider a 1 second interval and each processor operating at 20 MFLOPS. During this interval, each processor produces 0.14 MB or 1.12 Mb for communication, on the average. This translates to approximately 9Mbps from all the 8 processors. Ethernet is capable of supporting 10Mbps peak; it is not surprising, therefore, that Ethernet's performance gets steadily worse beyond 8 processors.

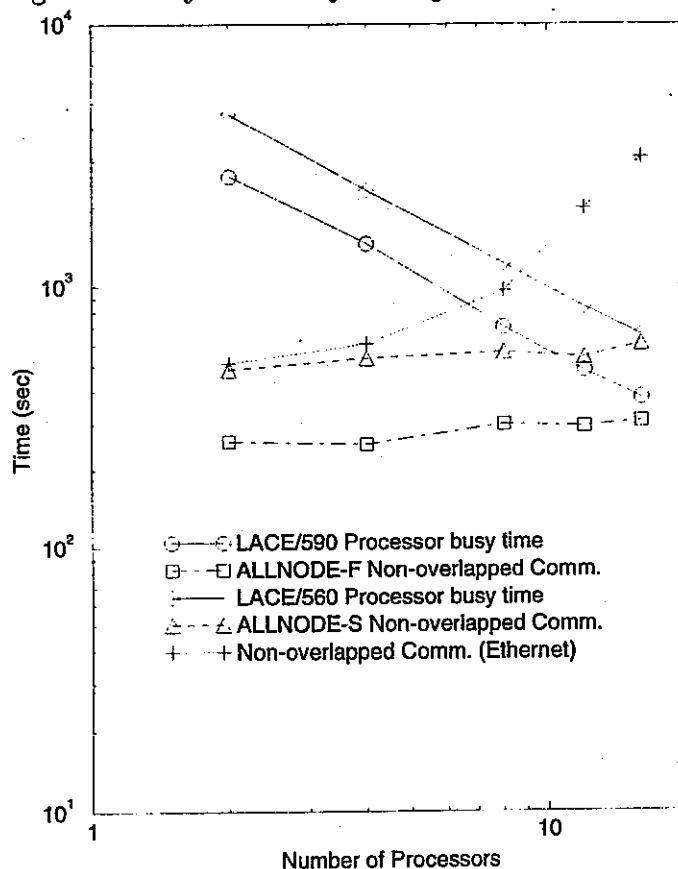


Figure 5: Components of execution time (Navier-Stokes; LACE)

Figures 5 and 6 aid in a more in depth analysis of the performance of LACE. The execution time is separated into two additive components as explained in the previous section. It is seen that the processor busy time falls linearly

with the number of processors. With Ethernet, the non-overlapped communication time increases superlinearly with the number of processors. With

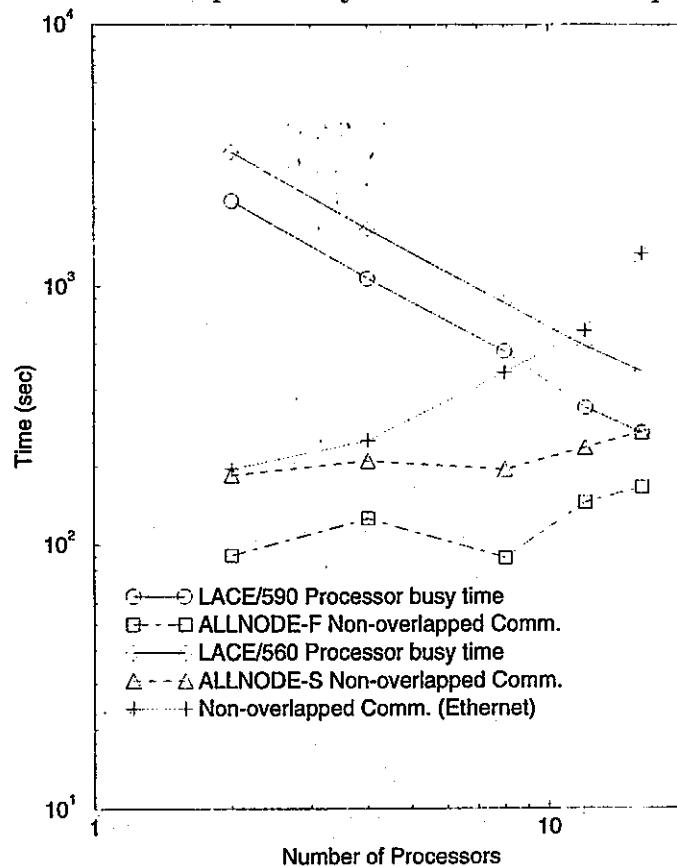


Figure 6: Components of execution time (Euler; LACE)

both ALLNODE switches, this time remains steady up to 10 or 12 processors beyond which it begins to rise. For Navier-Stokes with 16 processors, the communication time is comparable to the computation and PVM setup time while the ratio is about 60% for Euler. The difference in processor busy times and the communication times between the two ALLNODE configurations can be attributed to the superior node and the network respectively. A detailed examination of the data shows that both of these enhancements together contribute to the overall improved performance.

Figures 7 and 8 show the performance of Versions 5, 6, and 7 with Ethernet and ALLNODE-S (the trends are similar with ALLNODE-F). The performance of Version 6 (with overlapped communication and computation as explained in Section 6) is very close to that of Version 5 for both Ethernet and ALLNODE-S. Overlapping does not increase the number of communication startups. With Version 6, since computations for the subdomain have to be broken into separate ones for the interior and the boundary (only the former computations can be overlapped with communication), the loop setup overheads are higher. Further, the cache performance also degrades slightly due to loss of temporal locality. Consequently, these overheads offset any gain due to overlapping.

Version 7 attempts to reduce bursty communication at the cost of increased

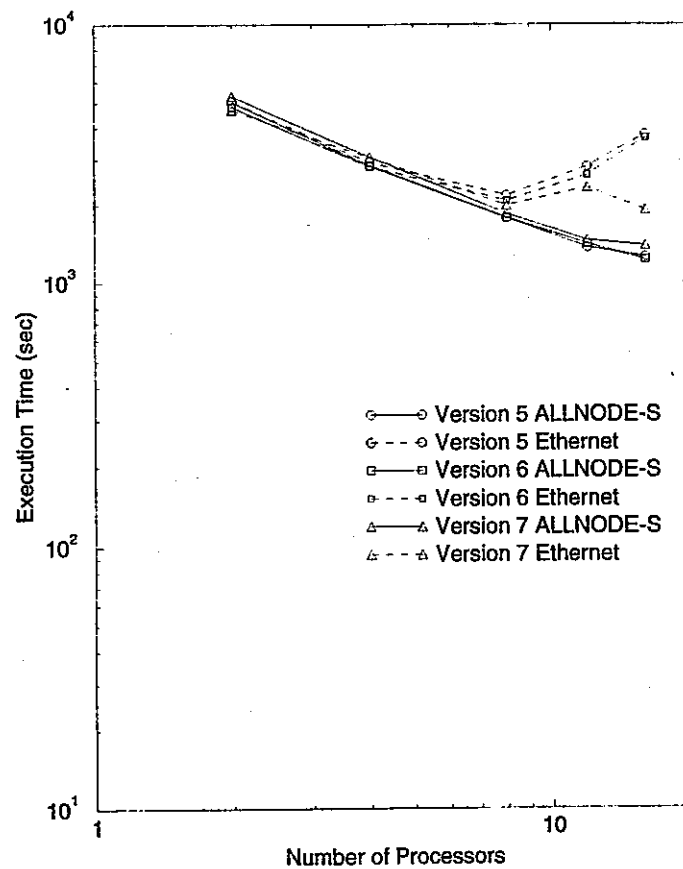


Figure 7: Communication optimization (Navier-Stokes; LACE)

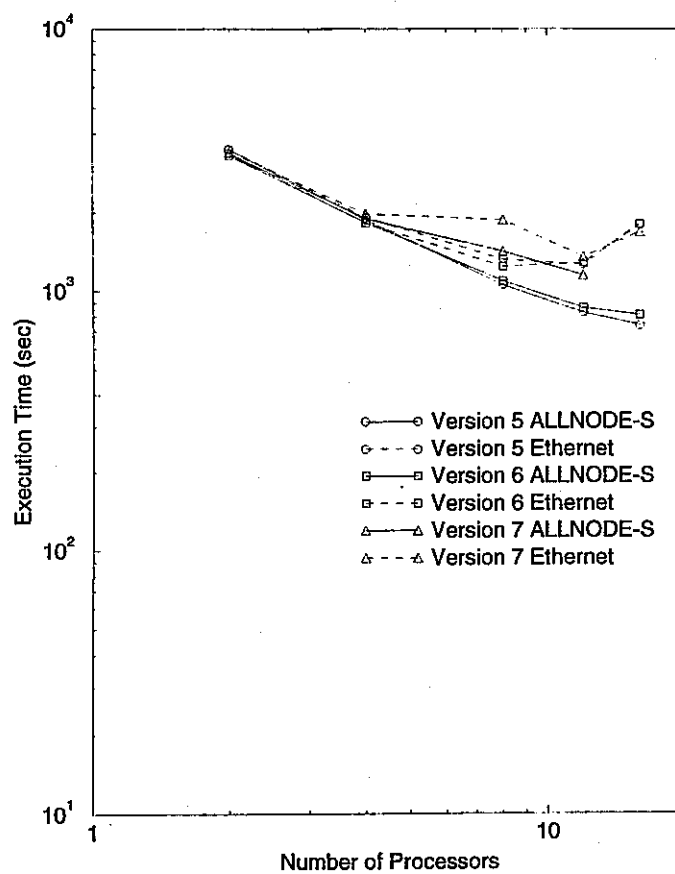


Figure 8: Communication optimization (Euler; LACE)

number of communication startups. Not surprisingly, Ethernet performs better with Version 7 than with Version 5. The performance of ALLNODE-S is appreciably worse than Version 5, however. Since ALLNODE-S can handle the communication requirements of the application, reducing bursty communication only harms the performance since the number of startups increase.

7.2 Comparative Performance

Figures 9 and 10 show the performance of the application on the four computing platforms we have chosen for this study- LACE, Cray Y-MP, Cray T3D and IBM SP. The performance of LACE is reported for ALLNODE-F and ALLNODE-S.

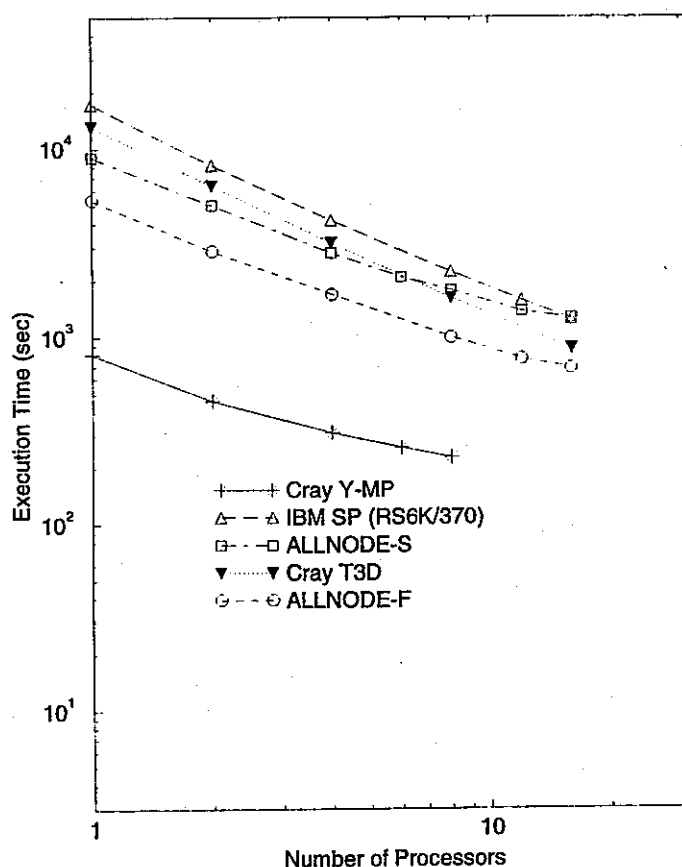


Figure 9: Execution time of Navier-Stokes on computing platforms

Surprisingly, LACE, even with ALLNODE-S, outperforms SP even though the former uses off-the-shelf PVM and the latter uses MPL, IBM's native message passing library. With (our version of) MPL, we were forced to use either blocking `send` or a constrained form of non-blocking `send` (for our communication requirements, both of the `send` primitives give similar results). This could possibly be one contributing factor to the relatively poor performance. The CPU on the SP is intermediate in speed (62.5 MHz clock) between the 560 (50 MHz) and the 590 (66.6MHz). Another contributor to

the poor performance of the SP is attributable to the data cache which is just 32KB (compared to 64KB on LACE/560 and 256KB on LACE/590).

For a comparison of ALLNODE-F and ALLNODE-S, see Section 7.1 (Figures 3, 4, 5, and 6).

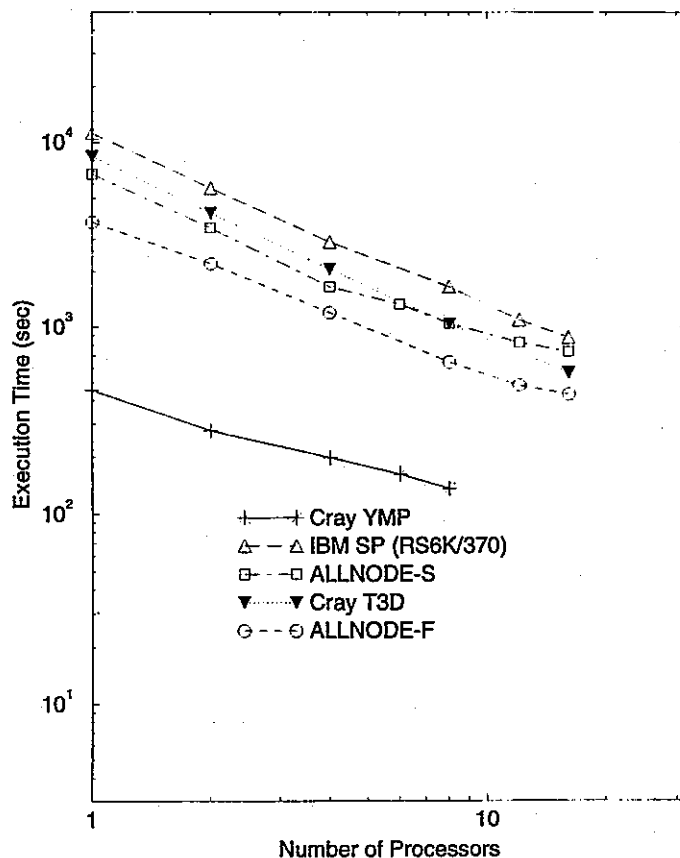


Figure 10: Execution time of Euler on computing platforms

Another surprising result is the relatively poor performance of Cray T3D which is consistently worse than ALLNODE-F and is worse than ALLNODE-S for less than 8 processors. The T3D's CPU has a peak rating which is 2.3X and 3X the rating of the 590 and 560 models respectively. We attribute the T3D's poor performance to the small direct-mapped cache of 8KB size (both the 560 and 590 have 4-way set-associative data caches of sizes 64KB and 256 KB respectively; in addition they have 2-way set associative instruction caches of sizes 8KB and 32KB). Poor single processor performance on the T3D has also been reported elsewhere [17]. Beyond 8 processors, T3D with its superior network speed (150 MB/sec peak transfer rate and a relatively small setup cost) performs better than ALLNODE-S. The T3D is still superior to the IBM SP.

The above results stress the importance of superior cache design to the overall performance.

Both T3D and SP exhibit very good speedup characteristics, with an almost linear drop in the execution time—indicating that the corresponding networks can sustain the communication requirements of the application. With

ALLNODE, a flattening of the speedup is seen beyond 12 processors. It is only reasonable to expect this trend to continue with increasing number of processors.

With architectures which use message passing libraries, the relatively poor performance can be attributed to large setup overheads and the resulting increase in processor waiting times with increasing number of processors. These overheads arise mainly from the multiple times that data to be communicated is copied and from the context switching overheads that arise in transferring a message between the application level and the physical layer of the network for transmission or reception. If NOW architectures are to be feasible as massively parallel processors, it is clear that both the interconnection network and the message passing library be implemented efficiently. Such effort is already under way [18].

Cray Y-MP has by far the best performance. The execution time shown is the connect time in single user mode (this includes the I/O time also which we were not able to separate from the computation time). The performance of LACE/590 with 16 processors is comparable to the single node performance of the Y-MP. The Y-MP (with a maximum of 8 processors) scales quite well for the applications.

7.3 Comparison of Message Passing Libraries

Figures 11 and 12 compare the performance of the PVMe and the MPL message passing libraries on the SP. The graphs show that MPL is consistently faster than PVMe by approximately 75% for Navier-Stokes and approximately 40% for Euler. Observe also that the amount of non-overlapped communication is not only negligibly small but that it decreases with the number of processors though the actual communication increases. This is an interesting phenomenon since it implies that there is increased overlapping of computation and communication with the number of processors. Note however that the computation part also includes the setup overheads of communication. This phenomenon is not seen in case of LACE (see Figures 3 and 4) where the non-overlapped communication increases, further attesting to our previous observation that the MPL (and PVMe) library does not perform as well as PVM does on LACE.

7.4 Load Balancing

Finally, how well is the application load balanced? The amount of computation for the application is evenly distributed but this may not always translate to a load balanced execution. We were able to measure the processor busy times (this time does **not** include the processor waiting time) for Navier-Stokes on each processor of the SP. Figure 13 shows that we were able to achieve almost perfect load balancing.

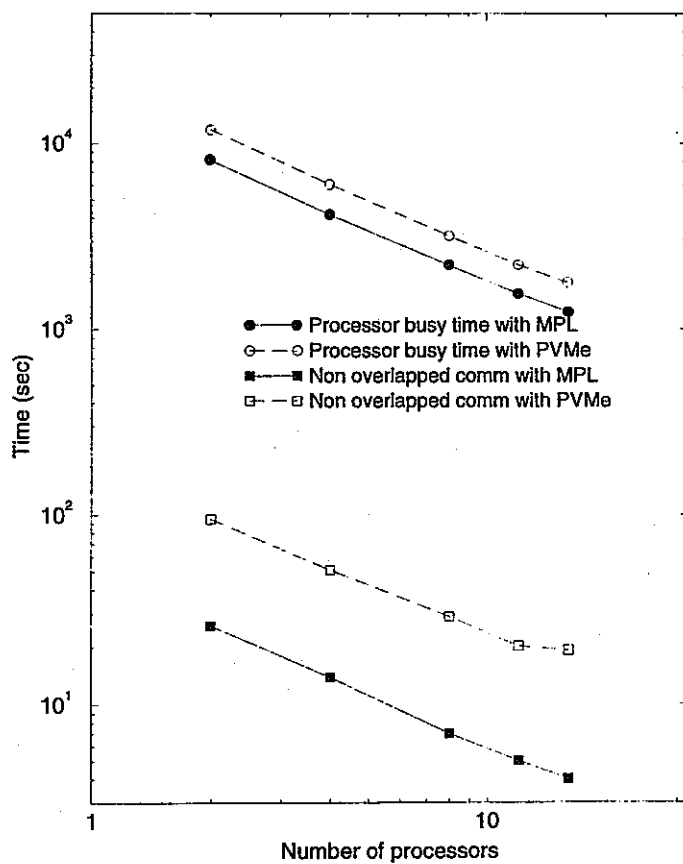


Figure 11: Comparison of MPL and PVMe (Navier-Stokes; IBM SP)

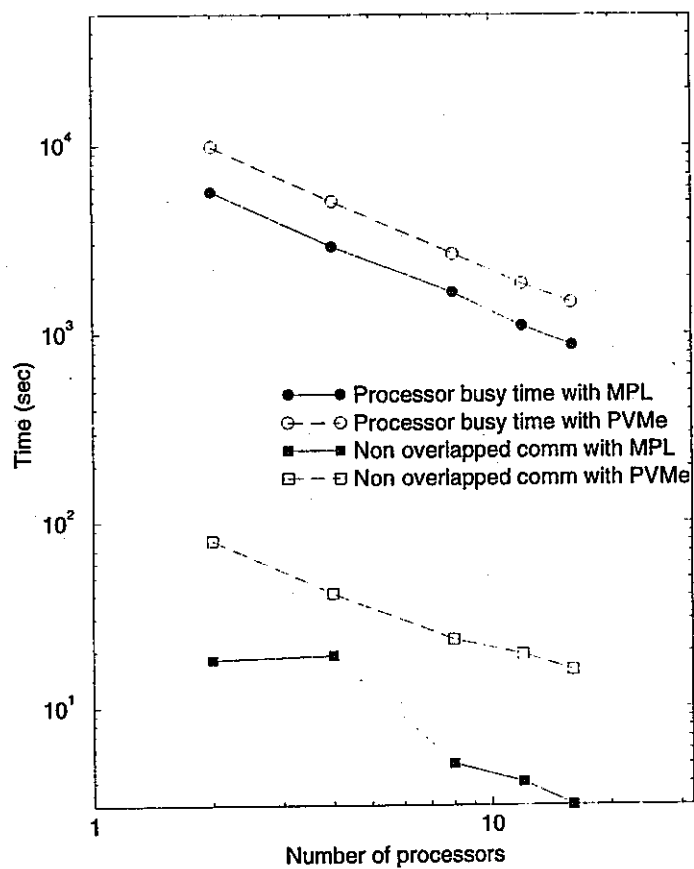


Figure 12: Comparison of MPL and PVMe (Euler; IBM SP)

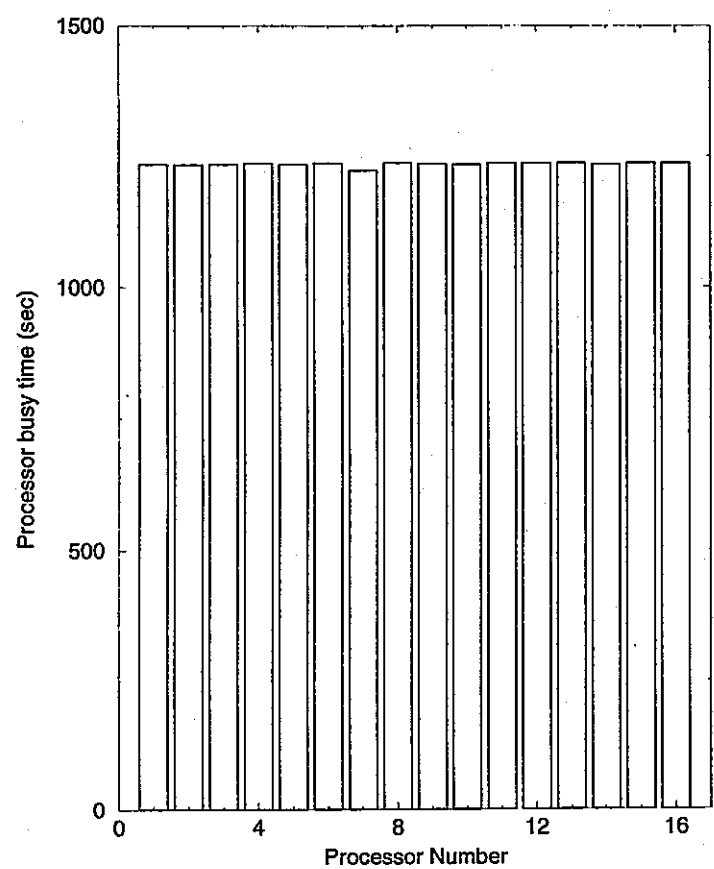


Figure 13: Processor busy times (Navier-Stokes; IBM SP)

8 Conclusion

In this paper we have studied the computational, communication, and scalability characteristics of a typical CFD application on a variety of architectural platforms. The study indicates that NOW have the potential to be cost-effective parallel architectures if the networks are made reasonably fast and message passing libraries are efficiently implemented to circumvent the traditional overheads involved in transferring a message between the application level and the physical layer of the network.

The study also highlights the importance of single processor performance to achieve good performance. With fast, off-the-shelf RISC processors available, the bottleneck seems to be the performance of the cache and the memory hierarchy. A proper cache design is critical to good performance. We believe that the reason for relatively poor performance of the T3D, in spite of a fast processor, is the small, direct-mapped cache.

A traditional vector multiprocessor still outperforms multiprocessors of modest to medium size. Parallelizing an application using message passing libraries is rather tedious and even error-prone but with distributed memory multiprocessors, this effort is worthwhile since good scalability is achievable.

Resource limitations have forced us to limit our study to 16 processors. We hope to extend the study to larger multiprocessors and to other parallelization tools as resources become available. We will then explore other problem decompositions such as blocking along the radial direction, for example, and study their impact on the performance.

Acknowledgments

The authors would like to thank Kim Ciula, Dale Hubler, and Rich Rinehart for their assistance with various aspects of the LACE and IBM SP architectures.

References

- [1] Lighthill, M. J. "On Sound Generated Aerodynamically, Part I, General Theory". *Proc. Roy. Soc. London*, vol. 211, 1952, pp. 564-587.
- [2] Hayder, M. E., Flannery, W. S., Littman, M. G., Nosenchuck, D. M. and Orszag, S. A. "Large Scale Turbulence Simulations on the Navier-Stokes Computer". *Computers and Structures*, vol. 30, no. 1/2, 1988, pp. 357-364.
- [3] Landsberg, A. M., Young, T. R. and Boris, J. P. "An Efficient, Parallel Method for Solving Flows in Complex Three Dimensional Geometries". *32nd AIAA Aerospace Sciences Conference*, AIAA 94-0413, January 1994.

- [4] Morano, E. and Mavriplis, D. "Implementation of a Parallel Unstructured Euler Solver on the CM-5". *32nd AIAA Aerospace Sciences Conference*, AIAA 94-0755, January 1994.
- [5] Venkatakrishnan, V. "Parallel Implicit Unstructured Grid Euler Solvers". *32nd AIAA Aerospace Sciences Conference*, AIAA 94-0759, January 1994.
- [6] Horowitz, J. G. "Lewis Advanced Cluster Environment". Distributed Computing for Aerosciences Applications Workshop, NASA Ames Research Center, October 1993.
- [7] Lenoski, D. E., et al. "The Directory-Based Cache Coherence Protocol for the DASH Multiprocessor". *Int'l Conf. on Computer Architecture*, May 1990, pp. 148-159.
- [8] Hayder, M. E., Jayasimha, D. N. and Pillay, S. K. "Parallel Navier-Stokes Computations on Shared and Distributed Memory Architectures". *33rd AIAA Aerospace Sciences Conference*, AIAA 95-0577, January 1995.
- [9] Gottlieb, D. and Turkel, E. "Dissipative Two-Four Methods for Time Dependent Problems". *Math. Comp.* vol. 30, 1976, pp. 703-723.
- [10] Hayder, M. E. and Turkel, E. "High Order Accurate Solutions of Viscous Problems". *31st AIAA Aerospace Sciences Conference*, AIAA 93-3074, January 1993.
- [11] Hayder, M. E. Turkel, E. and Mankbadi, R. R. "Numerical Simulations of a High Mach Number Jet Flow". *31st AIAA Aerospace Sciences Conference*, AIAA 93-0653, January 1993.
- [12] Mankbadi, R. R., Hayder, M. E. and Povinelli, L. A. "The Structure of Supersonic Jet Flow and Its Radiated Sound". *AIAA Journal*, vol. 32, no. 5, pp 897-906, 1994.
- [13] Geist, A., Beguelin, A., Dongarra, J., Jiang, W., Manchek, R. and Sunderam, V. "PVM 3 User's Guide and Reference Manual", *Technical Report ORNL/TM-12187*, Oak Ridge National Laboratory, Oak Ridge, TN, 1993.
- [14] Stunkel, C. B., Shea, D. G., Grice, D. G., Hochschild, P.H., Tsao, M. "The SP1 High-Performance Switch", *Scalable High Performance Computing Conference*, May 1994, pp. 150-157.
- [15] Oed, W. "The Cray Research Massively Parallel System- Cray T3D", *Technical Report*, Cray Research GmbH, November 1993.
- [16] Scott, J. N., Mankbadi, R. R., Hayder, M. E. and Hariharan S. I. "Outflow Boundary Conditions for the Computational Analysis of Jet Noise". *31st AIAA Aerospace Sciences Conference*, AIAA 93-4366, January 1993.

- [17] Bailey, D. H., Barszcz, E., Dagum, L., Simon, H. D. "NAS Parallel Benchmark Results". *NAS Technical Report NAS-94-001*, October 1994.
- [18] Anderson, T. A., Culler, D. E., Patterson, D. A., NOW team, "A Case for NOW (Networks of Workstations)". *IEEE Micro*, February 1995, pp. 54 - 64.

D. N. JAYASIMHA received the B.E. degree in Electronics Engineering from the University Visvesvaraya College of Engineering, Bangalore, India, the M.E. degree in Computer Science from the Indian Institute of Science, Bangalore, India, and the Ph.D. degree in Computer Science from the Center for Supercomputing Research and Development, University of Illinois, Urbana, IL. He is an Assistant Professor at the Department of Computer and Information Science, The Ohio State University, Columbus, OH. He was a Visiting Senior Research Associate at the NASA Lewis Research Center Cleveland, OH during 1993-94. His research interests are in various aspects of parallel computing and parallel architectures, and in distributed sensor networks. He is a member of the ACM, IEEE, and Sigma Xi.

M. EHTESHAM HAYDER received his BS from Bangladesh University and MS from Tulane University in Mechanical Engineering. He then received his Ph.D. from Princeton University in Mechanical and Aerospace Engineering. His doctoral dissertation was on parallel computations of computational fluid dynamics algorithms on, and evaluation of the architecture of, a research supercomputer, known as the Navier Stokes Computer. After finishing his Ph.D., he worked as a research associate at Princeton before joining the Institute for Computational Mechanics in Propulsion program at NASA Lewis Research Center, Cleveland, OH. He is now a senior research associate at this institute. His current research interests are in the area of computational aero-acoustics and parallel computations.

Dr. SASI KUMAR PILLAY currently manages all scientific and engineering computing that is provided through the Central Computing Organization at the NASA Lewis Research Center. This includes parallel processing environments. In the past, Dr. Pillay has managed work groups responsible for data acquisition and analysis, data networking, mainframe systems, high performance computing and computer graphics.

Dr. Pillay received his Ph.D. in Computer Engineering from Case Western Reserve University and a M. S. in Management of Technology from Sloan School of Management of the Massachusetts Institute of Technology. Dr. Pillay is also the recipient of the NASA Exceptional Service Medal.

Copyright © 1995 by the Association for Computing Machinery, Inc. (ACM).

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that new copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted.

To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Publications Dept, ACM Inc., via fax at +1 (212) 869-0481, or via email at permissions@acm.org.

<APPENDIX 3>

スーパーコンピューティング'95 会議入手資料リスト

<スーパーコンピューティング'95会議入手資料リスト>

(1) 会議発表論文集 (CD-ROM版のみ)

<米国政府関連>

- (2) 米国国家科学技術評議会によるHigh Performance Computing and Communications Programの進捗・計画レポート"FOUNDATION FOR AMERICA'S INFORMATION FUTURE"
- (3) 米国国家科学技術評議会1995年ストラテジープラン

<米国国立研究所関連>

- (4) NASA プロジェクト計画95
- (5) NASA Ames 研究センターパンフレット
- (6) ローレンスリバモア国立研究所計算科学パンフレット
- (7) ローレンスバークレー研究所1994-1995 研究レビュー
- (8) オークリッジ国立研究所計算科学パンフレット"The Path to the 150", "Computing R&D at Oak Ridge National Laboratory"
- (9) サンディア国立研究所計算科学パンフレット"Manufacturing TECHNOLOGIES"
- (10) ロスアラモス国立研究所計算科学パンフレット"Advanced Computing Laboratory"

<軍事関係>

- (11) 米国海軍研究所レビュー(1995)
- (12) 米国陸軍ハイパフォーマンスコンピューティング研究センターパンフレット

<学会関係>

- (13) 米国電子電気学会「コンピュータ」
- (14) ペタ(10¹⁵)フロップス・フロンティア・ワークショップ論文集 (IEEE)
- (15) 計算航空科学ワークショップ95要旨集 (NASA)

<大学関係>

- (16) ボストン大学計算科学パンフレット
- (17) ユタスーパーコンピューティングセンターニュース (ユタ大学)
- (18) ノースカロライナスーパーコンピューティングセンターパンフレット
- (19) マウイハイパフォーマンス計算センターパンフレット
- (20) 東京工業大学計算センター研究概要パンフレット
- (21) 埼玉大学総合情報処理センターパンフレット

<メーカー関係、その他>

- (22) 富士通VPP300シリーズ
- (23) IBM 並列計算用科学ライブラリーパンフレット
- (24) シリコングラフィックスパンフレット
- (25) 日立SR2201超並列計算機パンフレット
- (26) ユーロポート社パンフレット
- (27) ビジュアルニューメリクス社パンフレット
- (28) リアルタイムグラフィクス社

<APPENDIX 4>

マサチューセッツ工科大学メディア・ラボラトリ入手資料

MIT Media Lab Projects

Research at the Media Lab is divided into three general areas

Learning and Common Sense

Society of Mind - Marvin Minsky

"School" of the Future - Seymour Papert

Tools to Think With - Mitchel Resnick

Autonomous Agents - Pattie Maes

Machine Understanding - Kenneth Haase

Gesture and Narrative Language - Justine Cassell

Media That Learn - Henry Lieberman

Visual Intelligence - Matthew Brand

Perceptual Computing

Cognition - Whitman Richards

Looking at People - Alex Pentland

Synthetic Listeners and Performers - Barry Vercoe

Advanced Human Interface - Richard Bolt

Vision Texture - Rosalind Picard

Computers Understanding Action - Aaron Bobick

Speech - Christopher Schmandt

Information and Entertainment

Structured Video - Andrew Lippman

Spatial Imaging (Holography) - Stephen Benton

News in the Future - Walter Bender

Opera of the Future - Tod Machover

Information Appliances - Michael Hawley

Physics and Media - Neil Gershenfeld

Seamless Media Design - Hiroshi Ishii

Object-Oriented Video - V. Michael Bove, Jr.

Interactive Cinema - Glorianna Davenport

Intelligent Graphics - Ron MacNeil

Electronic Paper - Joseph Jacobson

MIT MEDIA LAB RESEARCH CONSORTIUMS

Many of the MIT Media Laboratory research projects described in the Project List are conducted under the auspices of three sponsor-supported, interdisciplinary Media Laboratory consortia. These are:

News in the Future

The News in the Future (NiF) research consortium provides a forum for the MIT Media Laboratory and member companies to explore and exploit technologies that will affect the collection and dissemination of news. The goals include enhancing the efficiency of production, the timeliness of delivery, the convenience of presentation, and the relevance of editorial and advertising content to the consumer. NiF focuses on four areas: description of news by and for computers; observation and modeling of consumer behavior; presentation and interface design; and application. The consortium develops technologies for managing data, building linkages between news providers and consumers, and enabling new approaches to the look and feel of news content. It investigates the application of these technologies by means of experiments at MIT and field experiments set up in cooperation with individual member companies.

Television of Tomorrow

Television of Tomorrow (TVoT) explores the advent of digital television: one of the few realities in today's communications revolution. Tomorrow's television will be more than entertainment-it will be the lingua franca of commerce, education, research, and personal information. The research of this program is the foundation for distributed digital libraries, on-demand audiovisual entertainment, and personalized, content-aware networks. "Television" is used as a metaphor for its breadth and reach: the deeper issue is inventing new computational media that will have the capability to paint a canvas unique for each viewer. This consortium draws on an international group of sponsors who are information providers, channel-makers, and hardware manufacturers. Current research areas include: Media Bank, Structured Video, Structured Audio, Cheops, Holographic Video, Content Coding, Electronic Landscapes, and Story Representations.

Things That Think

This new consortium, officially inaugurated at the Laboratory's 10th birthday celebration, explores ways of moving computation beyond conventional sites, such as PCs or laptops, and adding intelligence to objects that are first and foremost something else. By sensing the movements or feelings of their owners-or by learning their owners' habits-common devices such as toasters, doorknobs, or shoes, will be able, in their own right and through communication with one another, to solve meaningful problems. By becoming truly responsive and unobtrusive, the information technology in the inanimate things around us will enhance the quality of daily living. Things That Think (TTT) brings together an unusual range of interdisciplinary talent and builds upon the close research partnership between the Laboratory and its sponsor community.

Research Groups

The following are some of the major research groups at the Lab

- Autonomous Agents
- Design Interaction Paradigms
- Epistemology and Learning
- Machine Listening
- Physics and Media
- Spatial Imaging
- Visible Language Workshop
- Vision & Modeling

MIT Media Lab Projects 1995

Real-Time Modeling and Characterization

Professor Neil Gershenfeld

Digital systems that are used to describe realities, experiment with possibilities, or realize fantasies require efficient algorithms for the real-time modeling and characterization of complex systems. Because first-principle models of nontrivial systems quickly become computationally intractable, we are studying the use of state-space reconstruction techniques to develop compact models of the solution space of a system from measurements of the nonlinear input-output relationship. One area in which we are exploring this promise is in new musical instruments, both because of its significance for their evolution and because they provide an extremely demanding environment which requires the integration of multiple degrees of freedom of real-time I/O with high-speed computing.

Interface Transducers

Professor Neil Gershenfeld

Intelligent processors continue to be put in dumb boxes, yet the ability of computers to solve useful problems is increasingly limited by the ability of the user to interact with the machine. The interface of many of the most historically successful technologies, such as a Stradivarius violin or a Gutenberg bible, resides in the sophistication of the physical interaction with the device. We are developing the transducers needed for modern technology to capture this sophistication and naturalness. First, we have found that user actions can inexpensively and unobtrusively be measured with millimeter and millisecond resolution by detecting the tiny currents that arise from the interaction between people and varying electric fields. The limits of this technology are being explored, including 3-D imaging, and the use of more sophisticated measurement strategies for biometric characterization of the material in the field. This technology removes the wires from the user, but still requires wires to the sensor. A second project is to develop the use of active remote far-field interrogation of passive near-field structures to detect the ID, location, and state of tags. Finally, we are investigating the mechanisms and applications of smart materials for providing tactile feedback.

Information, Computation, and Physics

Professor Neil Gershenfeld

Information exists as high-level meaning (ideas), and as low-level physical representation (bits). Although these two levels of description are usually clearly separated, many of the most serious constraints and exciting possibilities in information technologies are at this interface. Physical insights can be applied to help understand computers. We are studying the use of techniques from the study of nonlinear time series for optimizing computer performance, and in the thermodynamic limit, and how this leads to fundamental bounds on energy and entropy budgets for practical tasks. Physical insights can also lead to new paradigms for computation. We are exploring the feasibility of classical and quantum computation in bulk materials through resonant spin interactions, and we have developed nonlinear dissipative generalizations of linear feedback shift registers, which overcome some of their important limitations.

MIT Media Lab Projects

THE VISIBLE LANGUAGE WORKSHOP

The thrust of research at the Visible Language Workshop (VLW) is toward the integration of two main directions: first, the study of the design of form and content in the new electronic information medium; second, the potential relationships among artificial intelligence, graphics, and technology. We continue to build a seamless, sophisticated graphical, dynamic, and interactive testbed to use for prototyping tools, editors, and scenarios. Simultaneously we are developing new artificial intelligence methods to help filter, prioritize, and design information on-the-fly, and to capture qualitative and intuitive design knowledge.

Information Landscapes

Ronald MacNeil, David Small and Suguru Ishizaki

In contrast to the discontinuous jumps between flat pages provided by the ubiquitous hyperlinked Web browser, an information landscape provides a continuum of paths from place to place, allowing loose and spatially meaningful relationships between information objects. Navigation of such spaces should not only reveal needed and appropriate information, but should do so in such a way that the journey itself has meaning.

Galaxy of News - This system embodies a scalable approach to visualizing and navigating through large quantities of independently created pieces of information, in this case news stories. It combines the effective aspects of both searching and browsing, and the ability to switch between these modes of operation seamlessly within a single interface. The system automatically organizes disconnected articles into dynamically formed groups, based on the content of the articles, and allows quick access to related information and the ability to understand the relationships among articles quickly.

Financial Viewpoints - (with Lisa Strausfeld): This project was the "virtual" manifestation of an earlier "physical" 3-D model built out of parallel planes of Plexiglas and threaded metal rods. Dynamic, transparent intersecting planes are used to represent complex information. Six mutual funds are compared on the basis of seven features such as risk and rate of return. Depending upon the way in which we approach the information--literally, the direction from which we zoom into the field of data--we can examine differently the features and the context that influence them. For example, we can retrieve straightforward numerical data from a chart to see how much a given fund rose or fell over a particular period of time. We are also able to pull up bar graphs to chart the annual rate of return against risk over the last 10 years, or look up the profiles of the men and women who direct the funds. The transparent quality of the visual structures and the variable focus of data help the viewer maintain the context. This helps prevent the feeling of being lost in a 3-D typographic space.

GeoSpace(with Ishantha Lokuge): The visualization of complex geographic information is a challenging design problem. Our goal was to create an interactive visualization system that can determine the interest of information seekers based on their queries, and can visually guide their attention in the complex data space. We have adopted information-seeking dialogue as a fundamental model of interaction, and have implemented a prototype geographic information system that progressively provides information upon a user's queries. The user-queries trigger presentation plans composed by domain knowledge that reflect a user's interest. The presentation plans are linked by an activation spreading network which creates a highly responsive visual environment. Dynamic use of various visual design techniques, such as transparency, 3-D graphics, and typography, are integrated to enhance the clarity of the visual display.

Millennium Project (with Earl Rennison and Lisa Strausfeld): In 1912 the S.S. Titanic sank on its maiden voyage, Woodrow Wilson won the U.S. presidential election, Sun Yat-sen founded The Kuomintang (the Chinese National Party), C.G. Jung published The Theory of Psychoanalysis, Edwin Bradenburger invented a process for manufacturing cellophane, and Marcel Duchamp painted "Nude Descending a Staircase." How, if at all, do these events relate to one another? Where, when, and what were the confluences of ideas and people that influenced the outcome of these events? How do we acquire the knowledge to understand the complex associations between people and ideas, across time and place, based on the artifacts and events they created?

The Millennium Project set out to address these issues by developing a conceptual and computational approach for enabling understanding of a large, multidimensional set of information. The goal of the Millennium Project is to provide a knowledge seeker with the ability to move through virtual time and space to explore and discover the connections among artifacts of philosophy, painting, music, literature, science, and political events of a pivotal time in world history: the years from 1906 to 1918. This virtual space continually constructs and reconstructs itself based on the knowledge seeker's movements through and within it, much like the process of moving through the conceptual spaces of our minds as we construct meaning. The conceptual framework for this research is based on linguistics and cognitive science. This project shows how our concepts of embodied cognitive models and visual discourse assist us in designing and building a computational environment that enables people to understand large bodies of information.

Dynamic Timelines: Visualizing Historical Information in Three Dimensions (with Robin L. Kullberg): This research reinvents the traditional timeline as a dynamic, 3-D framework for the interactive presentation of historical information. An experimental visualization of the history of photography uses new visual techniques, such as infinite zoom, translucency, and animation, to present a database of over 200 annotated photographs from the collection of the George Eastman House. Dynamic, interactive design solutions address the communicative goals of allowing seamless micro- and macro- readings of information at several levels of detail and from multiple points of view.

Enhancing Air-Traffic Control Information (with Jeffrey Ventrella) - Conventional ATC displays are dangerously confusing. This prototype creates visual analogs for such ATC concepts as "green space" (allowable distance between planes) and flight corridors in a 3-D space environment. Smooth transitions between tower, pilot, and global views maintain context. What-if scenarios dealing with weather conditions and flight paths can be replayed for training scenarios.

Intelligent Graphics: Meta-Design

Ronald MacNeil and Suguru Ishizaki

In order to design information in the age of dynamic, electronic media more effectively, we need a richer understanding of the structure of information (i.e., the architecture of information), and the tools to manipulate that structure. New tools for designers will incorporate these new structural representations to assist during construction, or later during presentation, when the designer may not have the luxury of overseeing information's design.

Dynamic Magical Environment (Ishantha Lokuge and Suguru Ishizaki): This project explores salient principles of close-up magic as a means of creating engaging interactive systems. Although previous research has identified the need to engage the user during the interaction process, there is a dearth of design methods for doing so. The term "engaging" is generally defined as "to embody characteristics that include responsive reaction, unobtrusive interface, guided navigation, suggestive exploration, and unexpected behavior." Principles in magic, such as framing context, focus of attention, continuity, adaptation, element of surprise, and timing and pacing, are studied and used as a guide to designing engaging interactive systems. Two experimental information systems have been implemented in the domain of geographic information: GeoSpace and MediaMagic. Both systems demonstrate the application of the principles and exemplify an approach for designing engaging interaction.

Temporal Typography (Yin Yin Wong and Suguru Ishizaki): Text is no longer limited to a static presentation in electronic communication. Typographic form can change in size, color, and position according to a reader's interaction in real time. This project proposes "temporal typography" as an area of study that incorporates the dynamic visual treatment as an extension of written language. Presently, graphic design lacks ways to conceptualize and describe temporal aspects of typography in a systematic and analytical fashion. We have developed a characterization scheme that provides a set of concepts and terminology that allows for the description of typographic expressions that change dynamically over time. A software tool, along with a scripting language, were implemented based on the scheme. This scheme has been examined through the creation of various design examples that explore dynamic treatment of text, including expressive e-mail, visualization of dialogue, and typographic tone of voice. Our current effort focuses on the design of visual interface that support the writing of temporally expressed text.

VIA (Dr. Louis Weitzman) - This thesis describes research to support the design and presentation of information in computer-based, multimedia documents. The "design and orderly arrangement" of information (i.e., the architecture of information), is a growing problem for future computational environments. Increases in computational power and the increased bandwidth of interconnected networks provide greater access to information. These factors, combined with the realization that not all of this information can now be pre-designed, necessitate new tools and techniques to ensure the effective presentation of computer-based information.

This research investigates the use of relational grammars (an extension to traditional string languages) to enable the design of dynamic documents based on formal visual languages. The description of design as a formal language affords a number of different interaction styles and support tools to the process of design. This research supports design in three distinct ways. First, information given to the system is automatically presented within a predefined language and style. This dynamic design reconfigures the same information, making it sensitive to the output environment in which the documents are delivered. Second, the creation of information can be supported by improved design decisions as the design progresses. These improvements help the user explore the design space with incremental design decisions. Finally, this class of visual languages is constructed by demonstration, using separate visual languages in a meta-design authoring tool.

TYRO - This is an evolving system which uses case-based reasoning for capturing and reusing media communication design knowledge. The task in each domain--such as page layout, video editing, dynamic electronic instruction manual production, or information landscapes--is to:

- express visually in editors and browsers the object attributes and behaviors, along with their underlying representation;
- provide a facility for demonstrating the canonical examples and capturing these in a case library; and
- for any novel case, be able to retrieve the most applicable cases and adapt them, given an evolving set of adaptation reasoning mechanisms.

Typographic Performance (Suguru Ishizaki): This project explores the possibility of developing visual design solutions for computational media that can continuously respond to changes in the information and the goal of its recipient. A design solution, such as a display of online news, is considered a performance composed of a number of active design agents (or performers), each of which is responsible for presenting a particular aspect of information, such as a headline and a news story. The individual design agent is sensitive to changes in its situation (the information itself, the goals of the reader, and the other design agents in the visual scene). The solution as a whole emerges from the dynamic activities of collaborating agents. This model is fundamentally different from the traditional view, which considers a design solution to be a set of affirmative statements. This proposed model suggests a new perspective on the way designers perceive a design solution: an "active entity" consisting of a collection of design agents. A multi-agent design testbed system has been implemented, along with high-quality typographic tools. We are currently developing various design examples in order to help evaluate the model as well as the computational technique.

Darwin Meets Disney (with Jeffrey Ventrella) - This research shows the use of genetic algorithms in graphic design. It emphasizes expression and motion style as demonstrated by animated characters. The animated characters can evolve behaviors like walking, using automatic evolution. They can also evolve expressive style to their motions via interactive techniques from the user.

DataWall: Large-Scale High-Resolution Displays

Ronald MacNeil

A 2000 x 6000-pixel seamless hybrid made by overlapping three 200-inch 2K x 2K Sony monitors, is still the highest resolution "TV" in the world. It supports experiments in group decision-making, testing the limits of just how much information can be displayed at once. A fully dynamic, large-scale DataWall projection display is in the works.

Information Appliances

Professor Michael Hawley

Tools and appliances of all sorts, from wristwatches and notebooks to concert grand pianos and home entertainment systems, are sprouting digital components. To interoperate harmoniously, and to ease the personal interface to a global information system, appliances need to communicate with each other. This project studies the languages and systems required for an open and scalable architecture.

BodyNet

Professor Michael Hawley, Professor Alex Pentland, Professor Rosalind Picard, and Thad Starner

Your interface to the worldwide network should be as convenient as a wristwatch and as comfortable as an old sweater. Like jewelry and clothes, it should be so comfortable and convenient that it is always available, always connected. The BodyNet is a very local, wireless network that will integrate all the information appliances on your person: cameras, watches, calculators, notebooks, datebooks, checkbooks, credit cards. Advanced displays, phones, radios, and other communication systems will be woven together to form an intimate, pervasive interface to the Net.

The Library Channel

Professor Michael Hawley

The future of public libraries is in the living room: the wealth of great libraries will in time be digitized and made universally accessible through home and personal appliances, via a plurality of wired and wireless means. The Library Channel project involves the Library of Congress, National Geographic, WGBH Television, and other partners in the Television of Tomorrow program. Our initial emphasis is on prototype interfaces to media banks.

Personal Embroidery

Professor Michael Hawley

We have built a custom embroidery language and tools to drive computerized sewing peripherals. Think of this as "Media Fab." Think of an embroidery machine as a 254-thread-per-inch multicolor printer, which outputs to garments instead of photocopy paper. And think of thread as potentially being conductive wire, or optical fiber. We are playing not only with custom clothing design, but also with fanciful embroidered circuitry.

MIT Media Laboratories - Vision and Modeling Group

The Vision and Modeling Group of the MIT Media Laboratory was formed in 1987 to study problems in computer vision and scene modeling. Since that time, the group has grown from 2 faculty and 4 graduate students to its current size of 4 faculty, 1 research staff, 2 post docs, 27 students, 3 administrative staff and several visiting faculty and research affiliates. Current research projects range from image compression and synthesis to knowledge-based video interpretation and the perception of human action.

The Vision and Modeling Group
The MIT Media Laboratory
20 Ames Street
Cambridge, MA 02139
Tel: 617-253-0369
Fax: 617-253-8874

**The Vision and Modeling Group
Journal, Book, and Conference Publication List**

Listing (by author) of group publications in academic journals, books, and conference proceedings.

Adelson, Edward, and P. Anandan

"Ordinal Characteristics of Transparency." Proceedings of the AAAI Workshop on Qualitative Vision, Boston, MA, pp. 77-81. July 1990.

Adelson, Edward, and James Bergen

"The Plenoptic Function and the Elements of Early Vision." Chapter 1 in Computational Models of Visual Processing, edited by M. Landy and J.A. Movshon, pp. 3-20. Cambridge, MA: MIT Press, 1991.

Adelson, Edward, and James Bergen

"Spatio-Temporal Energy Models for the Perception of Motion." Journal of the Optical Society of America A, Vol. 2, No. 2, pp. 284- 299. February 1985.

Adelson, Edward, William Freeman, and Eero Simoncelli

"Pyramids and Multiscale Representations." In Pyramids for Early Vision, edited by A. Gorea, pp. 3-16. New York: Cambridge University Press, 1991.

Adelson, Edward, Eero Simoncelli, and Rajesh Hingorani

"Orthogonal Pyramid Transforms for Image Coding." Proceedings of SPIE, Visual Communications and Image Processing II, Cambridge, MA. October 1987.

Adelson, Edward, and Alex Pentland

"The Perception of Shading and Reflectance." In Channels in the Visual Nervous System: Neurophysiology, Psychophysics, and Models, edited by B. Blum, pp. 195-207. London, England: Freund Publishing, 1991.

Adelson, Edward, and Eero Simoncelli

"Subband Image Coding with Three-Tap Pyramids." Proceedings of Picture Coding Symposium, Cambridge, MA. March 1990.

Adelson, Edward, and John Wang

"Single Lens Stereo with a Plenoptic Camera." IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 14, No. 2, pp. 99-106. February 1992.

Adelson, Edward and J.Y.A. Wang

"Single Lens Stereo with a Plenoptic Camera." IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 14, no. 2, pp. 99--106, February 1992.

Adelson, Edward and J.Y.A. Wang

"A Stereoscopic Camera Employing a Single Main Lens." In Proceedings of the IEEE Computer Vision and Pattern Recognition Conference, pp. 619--624, Hawaii, June 1991.

Azarbayejani, Ali; Bradley Horowitz, and Alex Pentland

"Recursive Estimation of Structure and Motion using Relative Orientation Constraints." Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, New York, NY. June 1993.

Bergen, James, and Edward Adelson

"Early Vision and Texture Perception." Nature, Vol. 333, No. 6171, pp. 363-364. May 1988.

Bobick, Aaron

"Representational Frames in Video Annotation." Proceedings of IEEE Signals and Systems Conference, Asilomar. November 1993.

Bobick, Aaron

"Verification Mode Vision: An Example from Egomotion and Optic Flow." IEEE Computer Vision and Pattern Recognition, New York. June 1993.

Bobick, Aaron

"Concurrent Evolving Object Descriptions in Support of Autonomous Navigation." Proceedings SPIE, Boston. 1992.

Bobick, Aaron, and Robert Bolles

"Multiple Concurrent Object Descriptions in Support of Autonomous Navigation." Proceedings of the SPIE. November 1992.

Bobick, Aaron, and Robert Bolles

"The Representation Space Paradigm of Concurrent Evolving Object Descriptions." IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 14, No. 2. February 1992.

Bobick, Aaron, and Robert Bolles

"An Evolutionary Approach to Constructing Object Descriptions." International Symposium on Robotics Research, Tokyo. August, 1989.

Bobick, Aaron, and Robert Bolles

"Representation Space: An Approach to the Integration of Visual Information." Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Diego, CA. June 1989.

Bobick, Aaron, and Robert Bolles

"Exploiting Temporal Coherence in Scene Analysis for Autonomous Navigation." Robotics and Automation, Scottsdale. May, 1989.

Bobick, Aaron, and Yvan Leclerc

"The Direct Computation of Height from Shading." Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Maui, HI. June 1991.

Casey, Michael A., William G. Gardner, and Sumit Basu

"Vision Steered Beam-forming and Transaural Rendering for the Artificial Life Interactive Video Environment (ALIVE)." Proceedings of the 99th Convention of the AES, October 1995 (preprints).

Darrell, Trevor; Azarbajani, Ali, and Pentland, Alex

"Robust Estimation of Multiple Models in the Structure from Motion Domain", Proc. IEEE Workshop on Performance vs. Methodology, CVPR-94, Seattle.

Darrell, Trevor; Essa, Irfan, and Pentland, Alex

"Correlation and Interpolation Networks for Real-time Expression Analysis/Synthesis", G. Tesauro, D. S. Touretzky and T. K. Leen, eds., Advances in Neural Information Processing Systems (NIPS-7), MIT Press, 1995.

Darrell, Trevor; Maes, Pattie; Blumberg, Bruce, and Pentland, Alex

"A Novel Environment for Situated Vision and Behavior", in Landy, M., ed., Exploratory Vision: The Active Eye, Springer-Verlag, 1995. also in Proc. IEEE Workshop on Visual Behaviors, IEEE Computer Society Press, Seattle, 1994.

Darrell, Trevor and Pentland, Alex

"Robust Estimation of Multiple Models using Support Maps", IEEE Transactions on Pattern Analysis and Machine Intelligence, May 1995.

Darrell, Trevor and Pentland, Alex

"Recognition of Space-Time Gestures using a Distributed Representation", in Artificial Neural Networks with Applications in Speech and Vision, Chapman and Hall, London, 1993.

Darrell, Trevor and Pentland, Alex

"Attention-driven Expression and Gesture Analysis in an Interactive Environment", Proc. Intl. Workshop on Automatic Face and Gesture Recognition, Zurich 1995.

Darrell, Trevor and Pentland, Alex

"Classifying Hand Gestures with a View-Based Distributed Representation", in J. Cowan, G. Tesauro, and J. Alspector, eds., Advances in Neural Information Processing Systems (NIPS-6) Morgan Kaufman, 1994.

Darrell, Trevor and Pentland, Alex

"Space-Time Gestures", in Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR '93) IEEE Computer Society Press, New York, 1993.

Darrell, Trevor and Pentland, Alex

"Against Edges: Function Approximation with Multiple Support Maps", in J. Moody, S. Hanson, and R. Lippman, eds., Advances in Neural Information Processing Systems (NIPS) 4, Morgan Kaufman, 1992.

Darrell, Trevor and Pentland, Alex

:"Robust Estimation of a Multi-Layer Motion Representation" Proc. IEEE Workshop on Visual Motion, IEEE Computer Society Press, Princeton, NJ, 1991.

Darrell, Trevor and Pentland, Alex

"On the Representation of Occluded Shapes", Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR '91), IEEE Computer Society Press, Maui, 1991.

Darrell, Trevor and Pentland, Alex

"Segmentation by Minimal Description", Proc. IEEE Third International Conference on Computer Vision (ICCV 3), IEEE Computer Society Press, Tokyo, 1990.

Darrell, Trevor and Simoncelli, Eero

"Nulling Filters and the Separation of Transparent Motions" in Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR '93)), IEEE Computer Society Press, New York, 1993.

Darrell, Trevor and Worn, K.

"Depth from Focus using a Pyramid Architecture", Pattern Recognition Letters, 11(12), pp. 787-796, December 1990.

Essa, Irfan and Alex Pentland

"Facial Expression Recognition using a Dynamic Model and Motion Energy", In Proceedings of International Conference on Computer Vision (ICCV 95), Cambridge, Massachusetts, June 20-23 1995. IEEE Computer Society Press, 1995.

Essa, Irfan and Alex Pentland

"Facial Expression Recognition using Visually Extracted Facial Action Parameters", In Proceeding of International Workshop on Automatic Face and Gesture Recognition 1995, Zurich, Switzerland, June 26-28 1995.

Essa, Irfan; T. Darrell, A. Azarbeyajani, S. Sclaroff and Alex Pentland

"Looking at People: Extracting Human Movements", In Proceedings of International Workshop on Computer Vision and Parallel Processing 1995, Islamabad, Pakistan, January 2-5 1995.

Essa, Irfan; Trevor Darrell and Alex Pentland

"Tracking Facial Motion." IEEE Workshop on Nonrigid and Articulated Motion, Austin, TX., November 1994.

Essa, Irfan and Alex Pentland

"A Vision System for Observing and Extracting Facial Action Parameters." Proceedings of IEEE Computer Vision and Pattern Recognition Conference, Seattle, WA., pages 76-83, June 1994.

Essa, Irfan; Stan Sclaroff, and Alex Pentland

"Physically-based Modeling for Graphics and Vision." Ralph Martin, editor, Directions in Geometric Computing. Information Geometers, U.K., 1993.

Essa, Irfan; Alex Pentland, and Stan Sclaroff

"A Unified Approach for Physical and Geometric Modeling for Graphics and Animation." Computer Graphics Forum, Eurographics '92, Vol. 11, No. 3, pp. C129-C138. September 1992.

Freeman, William, and Edward Adelson

"The Design and Use of Steerable Filters." IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 13, No. 9, pp. 891-906. September 1991.

Freeman, William; Edward Adelson, and David Heeger

"Motion Without Movement." Computer Graphics, Vol. 25, No. 4, pp. 27-30. July 1991.

Friedmann, Martin; Alex Pentland, and Thad Starner

"Device Synchronization Using an Optimal Linear Filter." Proceedings of ACM Symposium on Interactive 3D Graphics, Cambridge, MA, Vol. 26, No. 2. March 1992.

Gorkani, Monika M. and Rosalind W. Picard

"Texture Orientation for Sorting Photos 'at a Glance'." Proceedings of International Conference Pattern Recognition, Jerusalem, Israel, Vol I, pp. 459-464, October 1994.

Heeger, David, and Eero Simoncelli

"Model of Visual Motion Sensing." In "Spatial Vision in Humans and Robots", eds. L. Harris and M. Jenkin, Cambridge University Press, 1992.

Heeger, David, and Eero Simoncelli

"Sequential Motion Analysis." Proceedings of AAAI Robot Navigation Symposium, March, 1989.

Intille, Stephen, and Aaron Bobick

"Disparity-Space Images and Large Occlusion Stereo." Proceedings of the 1994 European Conference on Computer Vision, Stockholm, Sweden. May, 1994.

Intille, Stephen, and Aaron Bobick

"Incorporating Intensity Edges in the Recovery of Occlusion Regions." Proceedings of the 1995 International Conference on Pattern Recognition, Isreal. October 1995.

Izui, Yoshio, and Alex Pentland

"Analysis of Neural Networks with Redundancy." Neural Computation, Vol. 2, No. 2, pp. 226-238. 1990.

Johnson, Michael; Trevor Darrell, and Pattie Maes

"Evolving Visual Routines." In-house Publication, March 1993. 15 pages.

Lippman, Andrew; E. H. Adelson; A. N. Netravali; W. R. Neuman, and W. F. Schreiber

"Single-Channel Backward- Compatible EDTV Systems." SMPTE Journal, pp. 14-19. January 1989.

Liu, Fang and Rosalind Picard

"Periodicity, directionality, and randomness: Wold features for perceptual pattern recognition." Proceedings International Conference Pattern Recognition, Jerusalem, Israel, Vol. II, pp. 184-185, October 1994.

Mann, Steve

"Compositing Multiple Pictures of the Same Scene", Proceedings of the 46th Annual IS&T Conference, Cambridge, Massachusetts, May 9-14, 1993

Mann, Steve and Rosalind Picard

"Virtual Bellows: constructing high-quality images from video" Proceedings of the IEEE first international conference on image processing, Austin, Texas, Vol. I, pp. 363-367, November 1994.

Mann, Steve and Rosalind Picard

"On being 'undigital' with digital cameras: Extending Dynamic Range by Combining Differently Exposed Pictures." Proceedings of IS & T's 48th Annual Conference, Washington, DC, pp. 422-428, May, 1995.

Pentland, Alex

"Modal Descriptions for Recognition and Tracking." Proceedings of IAPR Workshop on Machine Vision Applications, Tokyo, pp. 435-444. December 1992.

Pentland, Alex

"Shape Information From Shading: A Theory of Human Vision." Spatial Vision, Vol. 4, No. 2/3, pp. 165-182. 1989.

Pentland, Alex; Moghaddam Baback, and Thad Starner

"View-Based and Modular Eigenspaces for Face Recognition." IEEE Conference on Computer Vision & Pattern Recognition, 7 pages. 1994.

Pentland, Alex; Trevor Darrell; W. Huang, and Matthew Turk

"A Simple, Real-Time Range Camera." Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, San Diego, CA, pp. 256-261. June 1989.

Pentland, Alex; Glorianna Davenport; Rosalind Picard, and Bill Welsh

"The BT/MIT Project on Advanced Image Tools for Telecommunications: An Overview." Image'Com '93, 2nd International Conference on Image Communications, Bordeaux, France, pp. 397-400, March 1993. OUT OF PRINT.

Pentland, Alex; N. Etcoff; A. Masoiu; O. Oliyide; T. Starner, and M. Turk.

"Experiments with Eigenfaces." Looking At People Workshop, IJCAI '93, Chamberry, France. August 1993.

Pentland, Alex, and Bradley Horowitz

"Recovery of Nonrigid Motion and Structure." IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 13, No. 7, pp. 730-742. July 1991.

Pentland, Alex, and Kenji Mase

"Automatic Lipreading by Optical-Flow Analysis." Systems and Computers in Japan, Vol. 22, No. 6, pp. 67-76. 1991.

Pentland, Alex; Rosalind Picard, and Stan Sclaroff

"Photobook: Tools for Content-Based Manipulation of Image Databases." SPIE PAPER 2185-05 Storage and Retrieval of Image and Video Databases II, San Jose, CA, pp. 34-47, February 6-10, 1994.

Pentland, Alex; Rosalind Picard; Glorianna Davenport, and Stan Sclaroff

"Project Reports: Video and Image Semantics: Advanced Tools for Telecommunications." IEEE Multimedia, pp. 73-75, Summer 1994.

Pentland, Alex, and Stan Sclaroff.

"Closed-Form Solutions for Physically-Based Shape Modeling and Recognition." IEEE Transactions on Pattern Analysis and Machine Vision, Vol. 13, No. 7, pp. 715-730. 1991.

Pentland, Alex, and John Williams

"Good Vibrations: Modal Dynamics for Graphics and Animation." Computer Graphics, Vol. 23, No. 4, pp. 215-222. July 1989.

Perry, Chris and Rosalind Picard

"Synthesizing Flames and their Spreading." Fifth Eurographics Workshop on Animation and Simulation, Oslo, Norway, September 1994.

Picard, Rosalind

"Structured Patterns from Random Fields." Proceedings of Asilomar Conference on Signals, Systems, and Computers, Pacific Grove, CA, pp. 1011-1015. October 1992.

Picard, Rosalind

"Random Field Texture Coding." Proceedings of Society for Information Display 1992 International Symposium, Boston, MA, Vol. XXIII, pp. 685-688. May 1992.

Picard, Rosalind

"Gibbs Random Fields: Temperature and Parameter Analysis ." Proceedings of IEEE Conference on Acoustics, Speech and Signal Processing, San Francisco, CA, Vol. III, pp. 45-48. March 1992.

Picard, Rosalind

"Markov/Gibbs Modeling: Texture and Temperature." Proceedings of SPIE Conference on Intelligent Robots and Computer Vision X: Algorithms and Techniques, Boston, MA, pp. 15-26. November 1991.

Picard, Rosalind, and Ibrahim Elfadel

"On the Structure of Aura and Co-occurrence Matrices for the Gibbs Texture Model." Journal of Mathematical Imaging & Vision, No. 2, pp. 5-25. 1992.

Picard, Rosalind; Ibrahim Elfadel, and Alex Pentland

"Markov/Gibbs Texture Modeling: Aura Matrices and Temperature Effects." Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Maui, HI, pp. 371-377, June 1991.

Picard, Rosalind, and Monika Gorkani

"Finding Perceptually Dominant Orientations." Special Julesz birthday issue of Spatial Vision, Vol. 8, No. 2, pp. 221-253, 1994.

Picard, Rosalind, and Tanweer Kabir

"Finding Similar Patterns in Large Image Databases." Proceedings of IEEE Conference on Acoustics Speech and Signal Processing, Minneapolis, MN, Vol. V, pp. 161-164. April 1993.

Picard, Rosalind; Tanweer Kabir, and Fang Liu

"Real-Time Recognition with the Entire Brodatz Texture Database." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, New York, NY, pp. 638-639. June 1993.

Picard, Rosalind, and F. Liu

"A new Wold ordering for image similarity." Proceedings of IEEE Conference on Acoustics, Speech and Signal Processing, Adelaide, Australia, Vol. V, pp.129--132, April 1994.

Picard, Rosalind, and T. Minka

"Vision Texture for Annotation." ACM/Springer-Verlag, Journal of Multimedia Systems, Vol. 3, pp. 3-14, 1995.

Popat, Kris, and Rosalind Picard

"Cluster-based Probability Model Applied to Image Restoration and Compression." Proceedings of IEEE Conference on Acoustics, Speech, and Signal Processing, Adelaide, Australia, Vol. V, pp.381--384, April 1994.

Popat, Kris, and Rosalind Picard

"Novel Cluster-Based Probability Model for Texture Synthesis, Classification, and Compression." Proceedings of the SPIE Visual, Communications and Image, Boston, pp. 756-768. Nov. 1993.

Popat, Kris and Rosalind Picard

"Exaggerated Consensus in Lossless Image Compression." Proceedings of IEEE First International Conference on Image Processing, Austin, Texas, Vol. III, pp. 846--850, November 1994.

Popat, Kris, and Kenneth Zeger

"Robust Quantization of Memoryless Sources Using Dispersive FIR Filters." IEEE Transactions on Communications, Vol. 40, No. 11, pp. 1670-1674. November 1992.

Sclaroff, Stan, and Alex Pentland

"A Modal Framework for Correspondence and Description." Proceedings of the Fourth International Conference on Computer Vision, pp. 308-313. May 1993.

Sclaroff, Stan, and Alex Pentland

"Generalized Implicit Functions for Computer Graphics." Computer Graphics, Vol. 25, No. 4, pp. 247- 250. July 1991.

Sherstinsky, Alex, and Rosalind Picard

"M-Lattice: A Novel Non-Linear Dynamical System and Its Application to Halftoning." Proceedings of the IEEE Conference on Acoustics, Speech, and Signal Processing, Adelaid, Australia, Vol. II, pp. 565--568, April 1994.

Sherstinsky, Alex, and Rosalind Picard

"Reaction-Diffusion Systems for Image Processing." Paper presented at Workshop on Synergetic Engineering: Applications of Nonlinear Dynamical Systems to Image Processing, Sydney, Australia. April 1993.

Sherstinsky, Alex and Rosalind Picard

"Orientation-Sensitive Image Processing with M-Lattice: A Novel Non-linear Dynamical System." Proceedings of IEEE First International Conference on Image Processing, Austin, Texas, Vol. III, pp. 152--156, November 1994.

Sherstinsky, Alex and Rosalind Picard

"Restoration and Enhancement of Fingerprint Images Using M-Lattice -- A Novel Non-Linear Dynamical System." Proceedings International Conference Pattern Recognition, Jerusalem, Israel, Vol. II, pp. 195--200, October 1994.

Simoncelli, Eero, and Edward Adelson

"Subband Transforms." Chapter 4 in Subband Image Coding, ed. J. Woods, pp. 143-192. Norwell, MA: Kluwer Academic Publishers, 1990.

Simoncelli, Eero, and Edward Adelson

"Non-Separable Extensions of Quadrature Mirror Filters to Multiple Dimensions." Proceedings of the IEEE, Vol. 78, No. 4, pp. 652-664. April 1990.

Simoncelli, Eero, and Edward Adelson

"Subband Image Coding with Hexagonal Quadrature Mirror Filters." Proceedings of Picture Coding Symposium, Cambridge, MA. March 1990.

Simoncelli, Eero; Edward Adelson; William Freeman, and David Heeger

"Shiftable Multiscale Transforms." IEEE Transactions on Information Theory, Vol. 38, No. 2, pp. 587-607. March 1992.

Simoncelli, Eero; Edward Adelson, and David Heeger

"Probability Distributions of Optical Flow." Proceedings of IEEE Computer Vision and Pattern Recognition Conference, Maui, HI, pp. 310-315. June 1991.

Turk, Matthew, and Alex Pentland

"Eigenfaces for Recognition." Journal of Cognitive Neuroscience, Vol. 3, No. 1, pp. 71-86. 1991.

Wang, John Y.A.; Edward Adelson, and U.Y. Desai

"Reducing Coding Artifacts in a Layered Representation Coding System." In Proceedings of the Picture Coding Symposium, Sacramento, California, September 1994.

Wang, John Y.A. and Edward Adelson

"Representing Moving Images with Layers." In IEEE Transactions on Image Processing Special Issue: Image Sequence Compression, vol. 3, no. 5, pp. 625-638, September 1994.

Wang, John Y.A. and Edward Adelson

"Spatio-Temporal Segmentation of Video Data." In Proceedings of SPIE: Image and Video Processing II, vol. 2182, pp. 120-131, San Jose, CA, February 1994.

Wang, John Y.A. and Edward Adelson

"Apply Mid-level Vision Techniques for Video Data Compression and Manipulation." In Proceedings of SPIE: Digital Video Compression on Personal Computers: Algorithms and Technologies, vol. 2187, pp. 116-127, San Jose, CA, February 1994.

Wang, John Y.A. and Adelson, Edward

"Layered Representation for Motion Analysis." In Proceedings of the IEEE Computer Vision and Pattern Recognition Conference, pp. 361-366, New York, June 1993.

Wang, John Y.A. and Edward Adelson

"Layered Representation for Image Sequence Coding." In Proceedings IEEE ICASSP, vol. 5, pp. 221-224, Minneapolis, MN, April 1993.

**MIT Media Laboratories
Vismod Tech-Reports**

Rosalind W. Picard
TR#358: Toward a Visual Thesaurus

Aaron F. Bobick and Claudio S. Pinhanez
TR#357: Divide and Conquer: Using Approximate World Models to Control View-Based Algorithms

Thad Starner, Steve Mann, Bradley Rhodes, Jennifer Healey, Kenneth B. Russell, Jeffrey Levine, and Alex Pentland
TR#355: Wearable Computing and Augmented Reality

Rosalind W. Picard
TR#354: Digital Libraries: Meeting Place for High-Level and Low-Level Vision

Christopher R. Wren, Ali Azarbayejani, Trevor Darrell, Alex Pentland
TR#353: Pfinder: Real-Time Tracking of the Human Body

Michael A. Casey, William G. Gardner, and Sumit Basu (1995)
TR#352: Vision Steered Beam-forming and Transaural Rendering for the Artificial Live Interactive Video Environment (ALIVE)

A. Pentland (1995)
TR#350: Machine Understanding of Human Action

T. P. Minka and R. W. Picard (1995)
TR#349: Interactive learning using a "society of models"

Steve Mann (1994)
TR#348: Recording 'lightspace' so shadows and highlights vary with varying viewing illumination.

M. Szummer (1995)
TR#346: Temporal Texture Modeling

Yair Weiss and Edward H. Adelson (1995)
TR#344: Motion Estimation and Segmentation Using a Recurrent Mixture of Experts Architecture

A. Azarbayejani and A. Pentland (1995)
TR#341: Camera self-calibration from one point correspondence

R. W. Picard (1995)
TR#339: Light-years from Lena: Video and Image Libraries of the Future

Steve Mann and Rosalind W. Picard (1995)
TR#338: Video orbits of the projective group; A new perspective on image mosaicing

A. Wilson and A. Bobick (1995)
TR#337: Learning Visual Behavior for Gesture Analysis

Chahab Natar and Alex Pentland (1995)
TR#334: Matching and Recognition Using Deformable Intensity Surfaces

S. Niyogi (1995)
TR#332: Detecting Kinetic Occlusion

- D. P. W. Ellis and D. F. Rosenthal (1995)
TR#330: Mid-level representations for Computational Auditory Scene Analysis
- A. Bobick and C. Pinhanez (1995)
TR#329: Using Approximate Models as Source of Contextual Information for Vision Processing
- T. Starner (1995)
TR#328: Human Powered Wearable Computing
- Matthew Brand and Irfan Essa (1995)
TR#327: Causal Analysis for Visual Gesture Understanding
- Baback Moghaddam and Alex Pentland (1995)
TR#326: Probabilistic Visual Learning for Object Detection
- Irfan Essa and Sandy Pentland (1995)
TR#325: Coding, Analysis, Interpretation, and Recognition of Facial Expressions
- C. Pinhanez and A. Bobick (1995)
TR#324: Intelligent Studios: Using Computer Vision to Control TV Cameras
- R. W. Picard (1995)
TR#321: Affective Computing
- F. Liu and R. W. Picard (1995)
TR#320: Periodicity, directionality, and randomness: World features for image modeling and retrieval
- S. A. Niyogi (1995)
TR#319: Kinetic Occlusion
- Thad Starner (1994)
TR#318: Wearable Computing (replaced by #355)
- Baback Moghaddam and Alex Pentland (1995)
TR#317: An Automatic System for Model-Based Coding of Faces
- Thad Starner (1995)
TR#316: Visual Recognition of American Sign Language Using Hidden Markov Models (S.M. Thesis)
- Y. Weiss and E.H. Adelson (1995)
TR#315: Perceptually organized EM: A framework for motion segmentation that combines information about form and motion
- T. Darrell and D. Fleet (1995)
TR#314: Second-order Method for Occlusion Relationships in Motion Layers
- (1995)
TR#313: (substituted by TR # 329)
- Trevor Darrell and Alex. P. Pentland (1995)
TR#312: Attention-driven Expression and Gesture Analysis in an Interactive Environment
- Stan Sclaroff (1995)
TR#311: Modal Matching: A Method for Describing, Comparing, and Manipulating Digital Signals [PhD Thesis]
- L. Campbell and A. Bobick
TR#309: Recognition of Human Body Motion Using Phase Space Constraints

A. Wilson and A. Bobick

TR#308: Using Configuration States for the Representation and Recognition of Gesture

I. Essa and A. Pentland (1995)

TR#307: Facial Expression Recognition using a Dynamic Model and Motion Energy

T. Starner and A. Pentland (1995)

TR#306: Visual Recognition of American Sign Language Using Hidden Markov Models

K. Russell, T. Starner, and A. Pentland (1995)

TR#305: Unencumbered Virtual Environments

S. Sclaroff and A. P. Pentland

TR#304: Modal Matching for Correspondence and Recognition

Irfan A. Essa (1994)

TR#303: Analysis, Interpretation and Synthesis of Facial Expressions [9385 kBytes]

R. W. Picard and T. P. Minka

TR#302: Vision Texture for Annotation

Baback Moghaddam and Alex Pentland (1994)

TR#301: Face recognition using view-based and modular eigenspaces

A. S. Sherstinsky (1994)

TR#300: M-Lattice: A System For Signal Synthesis And Processing Based On Reaction-Diffusion

A. Sherstinsky and R. W. Picard

TR#299: Applications of M-Lattice to Image Processing

A. Sherstinsky and R. W. Picard

TR#298: On Stability and Equilibria of Analog Feedback Networks

S. Intille

TR#296: Tracking Using a Local Closed-World Assumption: Tracking in the Football Domain

R. W. Picard

TR#295: Content Access for Image/Video Coding: "The Fourth Criterion"

S. Intille and A. Bobick

TR#294: Visual Tracking Using Closed-Worlds

F. Liu and R. W. Picard

TR#293: Periodicity, directionality, and randomness: Wold features for perceptual pattern recognition

M. Gorkani and R. W. Picard

TR#292: Texture Orientation for Sorting Photos "at a Glance"

A. Sherstinsky and R. W. Picard

TR#291: Restoration and Enhancement of Fingerprint Images Using M-Lattice -- a Novel Non-Linear Dynamical System

S. A. Niyogi and E. H. Adelson (1994)

TR#290: Analyzing gait with spatiotemporal surfaces

K. Popat and R. W. Picard

TR#289: Exaggerated Consensus in Lossless Image Compression

- A. Sherstinsky and R. W. Picard
TR#288: Orientation-Sensitive Image Processing with M-Lattice: A Novel Non-linear Dynamical System
- C. H. Perry and R. W. Picard
TR#287: Synthesizing Flames and their Spreading
- Ujjaval Yogesh Desai
TR#286: Coding of Segmented Image Sequences
- Steve Mann
TR#285: Lightspace and the Wyckoff principle: algebraic relationships between pictures of the same scene
- T. Darrell, I. Essa, A. Pentland
TR#284: Correlation and Interpolation Networks for Real-time Expression Analysis/Synthesis
- A. Pentland, R. Picard, G. Davenport, K. Haase
TR#283: Project Reports: Video and Image Semantics: Advanced Tools for Telecommunications
- T. Starner, J. Makhoul, R. Schwartz, G Chou (1994)
TR#282: On-line Cursive Handwriting Recognition Using Speech Methods
- T. Darrell, A. Azarbajani, and A. Pentland (1994)
TR#281: Robust Estimation of Multiple Models in the Structure from Motion Domain
- Bill Gardner and Keith Martin (1994)
TR#280: HRTF Measurements of a KEMAR Dummy-Head Microphone
- J. Y. A. Wang and E. H. Adelson (1994)
TR#279: Representing Moving Images with Layers
- S. Mann and R. W. Picard
TR#278: "Video Orbits" for Scene Change Detection
- Christopher H. Perry
TR#277: Synthesizing Interactive Fires
- J. Martin, A. Pentland, R. Kikinis
TR#276: Shape Analysis of Brain Structures Using Physical and Experimental Modes
- Stan Sclaroff and Alex Pentland (1994)
TR#275: On Modal Modeling for Medical Images: Underconstrained Shape Description and Data Compression
- Alex Pentland and Martin Bichsel (1994)
TR#274: Extracting Shape-From-Shading
- Stan Sclaroff and Alex Pentland (1994)
TR#273: Physically-Based Combinations of Views: Representing Rigid and Nonrigid Motion
- Irfan Essa, Trevor Darrell & Alex Pentland (1994)
TR#272: Tracking Facial Motion [2048 kBytes]
- A. Sherstinsky and R. W. Picard (1994)
TR#271: On The Efficiency Of The Orthogonal Least Squares Training Method For Radial Basis Function Networks
- Nassir Navab and Amnon Shashua
TR#270: Algebraic Description of Relative Affine Structure: Connections to Euclidean, Affine and Projective Structure

E. Adelson and A.P. Pentland
TR#268: The Perception of Shading and Reflectance

S. Sclaroff and A. P. Pentland
TR#267: Object Recognition and Categorization Using Modal Matching

Pentland, Darrell, Essa, Azarbayejani, Sclaroff
TR#266: Visually Guided Animation

I. M. Elfadel and R. W. Picard
TR#265: Gibbs Random Fields, Cooccurrences, and Texture Modeling

A. Sherstinsky and R. W. Picard
TR#264: Restoration and Enhancement of Fingerprint Images Using M-Lattice -- a Novel Non-Linear Dynamical System

J. Y. A. Wang, E. H. Adelson, U. Desai
TR#263: Apply Mid-level Vision Techniques for Video Data Compression and Manipulation

J. Y. A. Wang, E. H. Adelson
TR#262: Spatio-Temporal Segmentation of Video Data

Trevor Darrell, Pattie Maes, Bruce Blumberg, Alex P. Pentland (1994)
TR#261: A Novel Environment for Situated Vision and Behavior.

S. Mann and R. W. Picard
TR#260: The Virtual Bellows: A new perspective on the rigid planar patch

S. Mann and R. W. Picard
TR#259: Virtual Bellows: Constructing High Quality Stills from Video

Ali Azarbayejani, Alex Pentland
TR#258: A Generalized Extended Kalman Filter with Applications to Visual Motion Estimation

Maes, Darrell, Blumberg, Pentland
TR#257: The ALIVE system: Full-body interaction with animated autonomous agents

A. Pentland, R. Picard, S. Sclaroff
TR#255: Photobook: Tools for Content-Base Manipulation of Image Databases

R. W. Picard and A. P. Pentland
TR#254: Temperature and Gibbs Image Modeling

Kris Popat and R. W. Picard
TR#253: Cluster-based Probability Model Applied to Image Restoration and Compression

D. P. W. Ellis
TR#252: A simulation of vowel segregation base on across-channel glottal-pulse synchrony

Aaron Bobick
TR#251: Representational Frames in Video Annotation

S. Mann and S. Haykin
TR#250: The Chirplet Transform: A new signal analysis technique based on affine relationships in the time-frequency plane.

J.C.Brown
TR#249: Determination of the meter of musical scores by autocorrelation

J.C.Brown and M.S.Puckette

TR#248: A high resolution fundamental frequency determination based on phase changes of the fourier transform

I. Essa and A. Pentland [1822 KBytes]

TR#247: A Vision System for Observing and Extracting Facial Action Parameters

S. Intille and A. Bobick (1993)

TR#246: Incorporating Intensity Edges in the Recovery of Occlusion Regions

Alex Pentland, Baback Moghaddam and Thad Starner (1993)

TR#245: View-Based and Modular Eigenspaces for Face Recognition

T. Darrell and E. P. Simoncelli (1993)

TR#244: Separation of Transparent Motion into Layers using Velocity-Tuned Mechanisms

A. Azarbayejani and A. P. Pentland

TR#243: Recursive Estimation of Motion, Structure, and Focal Length

A. Azarbayejani, T. Galyean, B. Horowitz, and A. P. Pentland

TR#242: Recursive Estimation for CAD Model Recovery

G. T. Chou

TR#239: Visual Motion Perception: Uncertainty, Estimation and Control

R. W. Picard and F. Liu

TR#237: A new Wold ordering for image similarity

Amnon Shashua

TR#236: On Geometric and Algebraic Aspects of 3D Affine and Projective Structures from Perspective 2D Views

K. Popat and R. W. Picard

TR#234: Novel Cluster-Based Probability Model for Texture Synthesis, Classification, and Compression

Jeff A. Bilmes

TR#232: Techniques to Foster Drum Machine Expressivity

A. Sherstinsky and R. W. Picard

TR#231: M-Lattice: A Novel Non-Linear Dynamical System And Its Application To Halftoning

Kris Popat and Kenneth Zeger

TR#230: Robust Quantization of Memoryless Sources Using Dispersive FIR Filters

R. W. Picard and M. Gorkani (1993)

TR#229: Finding Perceptually Dominant Orientation in Natural Textures

E. H. Adelson and J. Y. A. Wang

TR#228: Representing Moving Images with Layers

Alex P. Pentland

TR#227: Modal Descriptions For Vision And Graphics

A. Pentland, S. Scherock, T. Darrell, and B. Girod

TR#226: Simple Range Cameras Based on Focal Error

Judith C. Brown

TR#225: Musical fundamental frequency tracking using a pattern recognition method

D P W Ellis

TR#224: A Computer Implementation of Psychoacoustic Grouping Rules

Sourabh A. Niyogi and Edward H. Adelson (1993)

TR#223: Analyzing and recognizing walking figures in XYT

M. M. Gorkani

TR#222: Designing an Orientation Finding Algorithm Based on Human Visual Data

J. Y. A. Wang and Edward H. Adelson

TR#221: Layered Representation for Motion Analysis

S. Intille and A. Bobick (1993)

TR#220: Disparity-Space Images and Large Occlusion Stereo

D P W Ellis

TR#219: Hierarchic models of hearing for sound separation and reconstruction

J.C. Brown and M.S. Puckette

TR#218: An Efficient Algorithm for the Calculation of a Constant Q Transform

J.C. Brown

TR#217: Musical Fundamental Frequency Tracking using a Pattern Recognition Method

J.C. Brown

TR#216: Calculation of a Constant Q Spectral Transform

R. W. Picard, T. Kabir, and F. Liu

TR#215: Real-Time Recognition with the Entire Brodatz Texture Database

Pawan Sinha

TR#211: Interactions between motion and contour perception processes in the primate visual system

Pawan Sinha and Edward Adelson

TR#210: Structure from Motion is Only Half the Story

Eero Simoncelli (1993)

TR#209: Distributed Representation and Analysis of Visual Motion

Aaron Bobick

TR#208: Verification Mode Vision: An example from egomotion and optic flow

A. Azarbayejani, B. Horowitz, and A. Pentland

TR#206: Recursive Estimation of Structure and Motion using the Relative Orientation Constraint

R. W. Picard and T. Kabir

TR#205: Finding Similar Patterns in Large Image Databases

I.M. Elfadel and R. W. Picard

TR#204: Gibbs random fields, co-occurrences, and texture modeling

R. W. Picard and T. Kabir

TR#203: The Brodatz Texture Database: Characterization by Shift-Invariant Principal Components

Eero Simoncelli (1992)

TR#202: Distributed Representations of Image Velocity

S. Sclaroff and A. Pentland
TR#201: A Finite-Element Framework for Correspondence and Shape Description

R. W. Picard (1992)
TR#200: Structured Patterns from Random Fields

P. Sinha and E. Adelson (1992)
TR#199: Recovering reflectance and illumination in a world of painted polyhedra

Trevor Darrell and Eero Simoncelli (1992)
TR#198: On the use of Nulling Filters to Separate Transparent Motions

T. J. Darrell and A. P. Pentland (1992)
TR#197: Space Time Gestures

W. T. Freeman (1992)
TR#196: Exploiting the generic view assumption to estimate scene parameters

David Heeger, Eero Simoncelli, EJ Chichilnisky (1992)
TR#195: OBVIUS: Object-Based Vision and Image Understanding System, version 2.2

Pentland, A., Starner, T., Etcoff, N., Masoiu, A., Oliyide, O., and Turk, M. (1993)
TR#194: Experiments with Eigenfaces

David J. Heeger and Eero P. Simoncelli (1992)
TR#191: Model of Visual Motion Sensing

William T. Freeman
TR#190: Steerable Filters and Local Analysis of Image Structure

Martin Friedmann and Alex Pentland (1992)
TR#189: Distributed Physical Simulation

A. Sherstinsky and R. W. Picard (1992)
TR#188: On Training Gaussian Radial Basis Functions for Image Coding

Martin Bichsel and Alex Pentland
TR#186: Human Face Recognition and the Face Image Set's Topology

R. W. Picard
TR#185: Random Field Texture Coding

Irfan A. Essa, Stan Sclaroff and Alex P. Pentland (1992)
TR#184: Physically-based Modeling for Graphics and Vision

Matthew A. Turk
TR#183: Interactive-Time Vision: Face Recognition as a Visual Behavior

Edward H. Adelson (1991)
TR#181: Layered Representations for Image Coding

A. Azarbayejani, T. Starner, B. Horowitz, Alex P. Pentland (1991)
TR#180: Visually Controlled Graphics

Irfan A. Essa, Stan Sclaroff and Alex P. Pentland (1992)
TR#179: A Unified Approach for Physical and Geometric Modeling for Graphics and Animation (See TR 184)

Irfan A. Essa and Alex P. Pentland (1991)
TR#178: Control of Nonrigid Robotic Systems using Visual Data

R. W. Picard

TR#177: Gibbs Random Fields: Temperature and Parameter Analysis

Friedmann, Starner, Pentland (1992)

TR#176: Device Synchronization using an Optimal Linear Filter

R. W. Picard and A. P. Pentland (1991)

TR#175: Markov/Gibbs Image Modeling: Temperature and Texture

Alex Pentland

TR#173: ThingWorld Users Manual

Martin Bichsel and Alex Pentland (1991)

TR#172: A Simple Algorithm for Shape from Shading

Yasufumi Amari (1991)

TR#171: Single-Eye Range Estimation by Using Displaced Apertures With Color Filters

Alex P. Pentland

TR#170: Surface Interpolation Networks

Alex P. Pentland (1991)

TR#169: Surface Interpolation Using Wavelets

R. W. Picard

TR#168: Texture Modeling: Temperature Effects on Markov/Gibbs Random Fields

Stephen F. Scherrock

TR#167: Depth From DeFocus of Structured Light

Sven J. Dickinson, Alex Pentland, Azriel Rosenfeld

TR#166: From Volumes to Views: An Approach to 3-D Object Recognition

Eero P. Simoncelli and Edward H. Adelson (1990)

TR#165: Computing Optical Flow Distributions Using Spatio-Temporal Filters

R. W. Picard, I. M. Elfadel, and A. P. Pentland

TR#164: Markov/Gibbs Texture Modeling: Aura Matrices and Temperature Effects

Trevor Darrell and Alex Pentland (1991)

TR#163: Cooperative Robust Estimation Using Layers of Support

Eero Simoncelli, William Freeman, Edward Adelson, and David Heeger (1991)

TR#161: Shiftable Multi-scale Transforms

R. W. Picard and I. M. Elfadel

TR#160: On the Structure of Aura and Co-occurrence Matrices for the Gibbs Texture Model

I. M. Elfadel and R. W. Picard

TR#159: New Miscibility Measure explains the Behavior of Grayscale Texture Synthesized by Gibbs Random Fields

Martin Friedmann, T. Starner, and Alex Pentland (1991)

TR#157: Synchronization in Virtual Realities

William T. Freeman, Edward H. Adelson, and David J. Heeger

TR#156: Motion without Movement

Edward H. Adelson, Eero P. Simoncelli, and William T. Freeman
TR#155: Pyramids and Multiscale Representations

Matthew Turk and Alex Pentland
TR#154: Eigenfaces for Recognition

Dickinson, S. and Pentland A.
TR#153: A Unified Approach to the Recognition of Expected and Unexpected Geon-Based Objects

Alex Pentland and Bradley Horowitz
TR#152: A Practical Approach to Fractal-Based Image Compression

Stanley Sclaroff and Alex Pentland (1991)
TR#151: Generalized Implicit Functions For Computer Graphics

Edward Adelson and P. Anandan
TR#150: Ordinal characteristics of transparency

Edward Adelson and James R. Bergen
TR#148: The plenoptic function and the elements of early vision

Edward Adelson, John Wang
TR#147: Single lens stereo with a plenoptic camera

Alex Pentland
TR#145: Thingworld Modeling System

Dickinson, Alex Pentland, and Rosenfeld
TR#144: 3-D Shape Recovery Using Distributed Aspect Matching

Alex P. Pentland
TR#143: Finite element and regularization solutions using wavelet basis

Bernd Girod and Stephen Scherock
TR#141: Depth from DeFocus of Structured Light

Edward H. Adelson and Alex P. Pentland (1990)
TR#140: The Perception of Shading and Reflectance

Takanori Senoo, Bernd Girod (1990)
TR#139: Vector Quantization for Entropy Coding of Image Subbands

Takanori Senoo, Bernd Girod, Andrew Lippman (1990)
TR#138: Optimization of Subband Vector Quantization for Moving Images

Eero Simoncelli and Edward Adelson (1989)
TR#137: Subband Transforms

Alex Pentland, Bradley Horowitz (1990)
TR#136: Recovery of Non-Rigid Motion and Structure

Alex Pentland, Stan Sclaroff (1990)
TR#135: Closed-Form Solutions for Physically Based Shape Modeling and Recognition

William Freeman
TR#134: Steerable Filters for Early Vision, Image Analysis, and Wavelet Decomposition

Emerson, Bergen, Adelson
TR#133: Directionally Selective Complex Cells and the Computation of Motion Energy in Cat Visual Cortex

Matthew Turk, Alex Pentland (1990)
TR#132: Representing Faces for Recognition

Alex Pentland, Jeff Kuo (1990)
TR#131: Three-Dimensional Interpretation via Local Processing

Alex P. Pentland, Irfan Essa, Martin Friedmann, Sclaroff, S., Bradley Horowitz, Stanley Sclaroff (1990)
TR#130: The ThingWorld Modeling System: Virtual Sculpting by Modal Forces

Alex P. Pentland
TR#129: Computational Complexity versus Stimulated Environments

Yoshio Izui and Alex Pentland
TR#128: Analysis of Neural Networks with Redundancy

Michael Sokolov (1990)
TR#127: Visual Motion: Algorithms for Analysis and Application

William T. Freeman and Edward H. Adelson (1990)
TR#126: Steerable Filters for Image Analysis

David J. Heeger (1989)
TR#125: Computational Model of Cat Striate Physiology

David J. Heeger and Allan Jepson (1989)
TR#124: Visual Perception of Three-Dimensional Motion

David J. Heeger and Eero Simoncelli (1989)
TR#123: Sequential Motion Analysis

David J. Heeger, Greg Hager
TR#122: Egomotion and the Stabilized World

Alex P. Pentland (1989)
TR#120: Photometric Motion

Eero P. Simoncelli and Edward H. Adelson (1989)
TR#119: Non-separable Extensions of Quadrature Mirror Filters to Multiple Dimensions

W.T. Freeman and E.H. Adelson
TR#118: The Design and use of Steerable Filters

A.P. Pentland and K. Mase
TR#117: Lip Reading: Automatic Visual Recognition of Spoken Words

A.P. Pentland
TR#116: A Neural Mechanism for Computing Shape From Shading

A.P. Pentland and J. Williams
TR#115: Modulations: Modal Dynamics for Graphics and Animation

A.P. Pentland and J. Kuo
TR#114: The Artist at the Interface

A.P. Pentland and J. Williams
TR#113: Perceiving Material Properties

A.P. Pentland, T. Darrell, M. Turk, and W. Huang
TR#112: A Simple, Real-Time Range Camera

A.P. Pentland and J. Williams
TR#111: Virtual Manufacturing

J. Williams and A.P. Pentland
TR#110: Superquadric Object Representation for Dynamics of Multi-body Structures

P. Kube and A. P. Pentland
TR#109: On the Imaging of Fractal Surfaces

A.P. Pentland
TR#108: Parallel Part Segmentation for Object Recognition

A.P. Pentland
TR#106: Linear Shape From Shading

A.P. Pentland
TR#104: Automatic Extraction of Deformable Part Models

A.P. Pentland
TR#103: Shape Information From Shading: A Theory About Human Perception

A.P. Pentland
TR#102: On the Extraction of Shape Information From Shading

E. Simoncelli, E. H. Adelson
TR#101: Multi-Scale Image Transforms

E. Simoncelli
TR#100: Orthogonal Sub-band Image Transforms

E.H. Adelson, E.P. Simoncelli, R. Hingorani
TR#99: Orthogonal pyramid transforms for image coding

**MIT Media Laboratories
Vismod Projects**

This is a listing of the major faculty-directed projects in Vismod

Mid-Level Vision

Professor Edward Adelson

We are developing early and mid-level vision mechanisms that emulate the processing that occurs in primate visual cortex and are designing algorithms that apply them with high computational efficiency. The mechanisms are useful for edge detection, texture analysis, motion analysis, and image enhancement.

X-Y-T Image Analysis

Professor Edward Adelson
Professor Aaron Bobick

We treat a sequence of images as a three-dimensional volume, with the dimensions of x, y, and t (time). Motion analysis involves orientation-selective filtering within this volume. We are developing techniques for dealing with difficult situations such as motion occlusion and motion transparency.

Dynamic Scene Annotation

Professor Aaron Bobick

In a dynamic scene, what is in the image is less important than what is happening in the scene. We are developing dynamic description mechanisms capable of extracting the important aspects of the behavior or motion present in a scene. Two domains we are exploring are charting football plays and extracting choreography from a ballet sequence.

Smart Cameras

Professor Aaron Bobick

We are developing techniques that will allow a camera control system to understand enough about a dynamic scene that it will be able to respond to requests for specific camera shots. Such a system must understand enough about the intent of a director, the semantics and dynamics of a scene, and the computer vision techniques capable of localizing specific objects and action to decide where to aim a camera. The integration of this knowledge requires a common framework for representing the action in the scene.

Looking at People

Professor Alex Pentland
Professor Aaron Bobick
Dr. Irfan Essa

This large, multiyear research project called "Looking at People" is composed of several different subprojects including real-time tracking people's body positions as they point and move about in the work environment, gesture and expression recognition, and continued development of our real-time face recognition system. Currently there are two "test bed" applications of this technology: a real-time virtual reality system called ALIVE (with Professor Pattie Maes) and a "smart" teleconferencing system.

Model-Based Image Coding

Professor Alex Pentland
Dr. Irfan Essa

This research project is developing generic, physically based models that allow ultra-low bandwidth image compression. Using such models we can concisely describe an object's appearance, and predict how its appearance will change as the object and camera move. Using these techniques we have been able to achieve high-quality still-image compression with 50:1 to 100:1 compression ratios, and high-quality video compression at only 8 kilobits/second.

Video and Image Libraries: Representation and Retrieval

Professor Rosalind W. Picard
Professor Alex Pentland

One of the most significant problems with multimedia technology is that you can't find what you want. This is because, unlike text-only systems, you can't ask a computer about the contents of images or video. For instance, you can't ask the computer to "find another video clip like this one, but shot from another angle," or "find a video clip of me on the beach." We are working to solve these problems by making computers able to "see" the contents of images and video.

Affective Computing

Professor Rosalind W. Picard

Recent neurological evidence indicates that emotions are not a luxury; they are essential for "reason" to function normally, even in rational decision-making. Furthermore, emotional expression is a natural and significant part of human interaction. Whether it is used to indicate like/dislike or interest/disinterest, emotion plays a key role in multimedia information retrieval, user preference modeling, and human-computer interaction. Affective computing is a new area of research focusing on computing that relates to, arises from, or deliberately influences emotions. The focus of the present project is on giving computers the ability to recognize affect. Current applications include better learning systems (computer recognizes interest, frustration, or pleasure of pupil), and smarter "things" such as a steering wheel/seatbelt that sense when a driver is angry or incapacitated.

Virtual Bellows for Video

Professor Rosalind W. Picard

Artists exercise viewpoint freedom with cubism and collage, and photographers flex their camera bellows; both desire to express what they see from multiple perspectives. This research harnesses the power of perspective mathematically, allowing one to extract and modify perspective in video. Applications include image mosaicing, high-resolution digital cameras, high-resolution printing, and recognition of video scene changes and camera motion.

Perceptual Similarity Measures

Professor Rosalind W. Picard

People are great at identifying similar patterns -- in pictures, sound, or human behavior. But, how they do it remains a mystery. Based on results from how humans recognize visual patterns, we are building computer models to mimic recognition of perceptual similarity. Particular attention is given to how humans interpret directionality, periodicity, randomness, contrast, translation, rotation, perspective and scale in natural scenes.

Semantic Image Modeling

Professor Rosalind W. Picard

If I state "Atlanta is in Cincinnati" today, it is unlikely you will think I am coherent. If, however, we are talking baseball, then the sentence is very clear. The context makes the interpretation not only easier, but possible. Similarly, with pictures, if you see blue at the top then it's probably sky. The goal of this work is to begin setting up two-way interaction between available contextual information and the models used to represent visual information. The ultimate goal is the one Shannon missed - putting semantic meaning into "information" theory.

WearCam: Video Orbits for Visual Memory

Professor Rosalind W. Picard

We are building a wearable wireless head-mounted video camera (WearCam) for diverse uses such as a hands-free sports camera or a system for augmenting visual memory. However, the digital video processing problems are immense. This research explores combinations of physiological signals from "Affective Computing" with signal processing algorithms from "Video Orbits" for reducing the amount of video that must be processed and assisting the user in deciding what video should be "remembered."

MIT Laboratories
Physics and Media Publications

Field Mice: Extracting Hand Geometry From Electric Field Measurements Joshua R. Smith, to appear in the IBM Systems Journal.

Signal Entropy and the Thermodynamics of Computation Neil Gershenfeld, to appear in the IBM Systems Journal.

Musical Applications of Electric Field Sensing Joe Paradiso and Neil Gershenfeld, preprint.

Entrainment and Communication with Dissipative Pseudorandom Dynamics Neil Gershenfeld and Geoff Grinstein, Physical Review Letters (74), p. 5024 (1995).

Applying Electric Field Sensing to Human-Computer Interfaces Tom Zimmerman, Joshua R. Smith, Joe Paradiso, David Allport, and Neil Gershenfeld. Proceedings of CHI-95.

Electric Field Sensing and the "Flying Fish" David Allport, Thomas G. Zimmerman, Joseph A. Paradiso, Joshua R. Smith and Neil Gershenfeld. To appear in the December 1995 ACM special issue on Multimedia and Multisensory Virtual Worlds.

Lattice Polymer Automata Steen Rasmussen and Joshua R. Smith. Figures missing. Berichte der Bunsengesellschaft für Physikalische Chemie 98, 1185-1193 (1994) No. 9.

Information in Dynamics Neil Gershenfeld, in Proceedings of the Workshop on Physics of Computation, D. Matzke ed., pp. 276-280 (IEEE Press, New York 1993).

The Future of Time Series Neil Gershenfeld and Andreas Weigend, in Time Series Prediction: Forecasting the Future and Understanding the Past, Andreas Weigend and Neil Gershenfeld eds., Santa Fe Institute Studies in the Sciences of Complexity, (Addison-Wesley, Reading MA, 1993).

<APPENDIX 5>

クレイ・リサーチ社での入手資料

December 11, 1995

PNC VISIT

December 12, 1995

GUEST: Hiroyuki Oshima

CRAY COUNTRY ESCORT: Kazuya Terauchi

GUEST SERVICES: Linda Yetzer - 612/683-3556

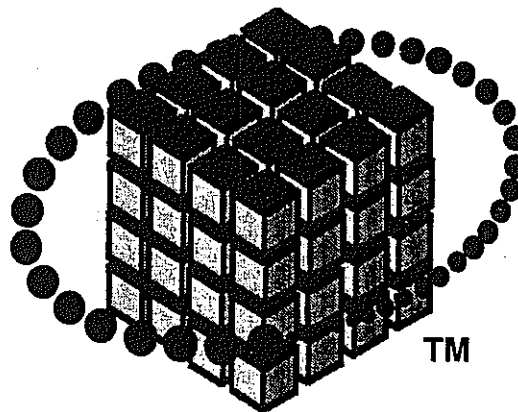
HOTEL ACCOMMODATIONS: Embassy Suites (airport) - 612/854-1000

AGENDA

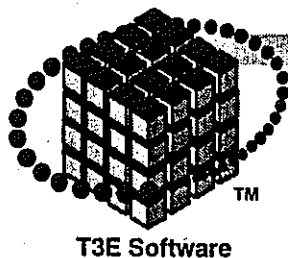
TUESDAY, DECEMBER 12 - CRAY RESEARCH PARK - 655A - ROOM 285A

9:00 am	Company and Products Overview	Vito Bongiorno, Director Marketing Programs
10:00 am	T3E Overview and Futures	William White, MPP Product Manager
10:30 am	Programming Environment on MPP	Kathy Nottingham, Director Software Product Management
11:00 am	Applications	Jef Dawson, Manager CFD Applications Steve Behling, CFD Applications
12:00 am	Data Center Tour	Jim Kramer, Manager CCN Eng.
12:30 pm	Lunch - (catered 234.2D)	Steve Behling
1:30 pm	Departure	

CRAY T3E Software Plans

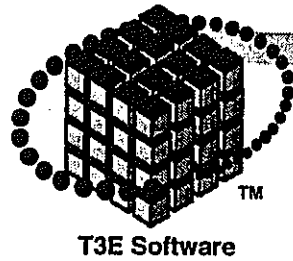


Kathy Nottingham
Director, Software Product Management
Cray Research, Inc.
kln@cray.com

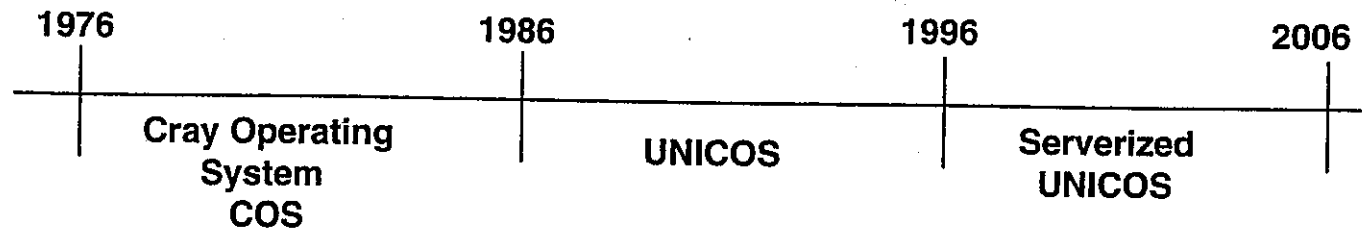


Agenda

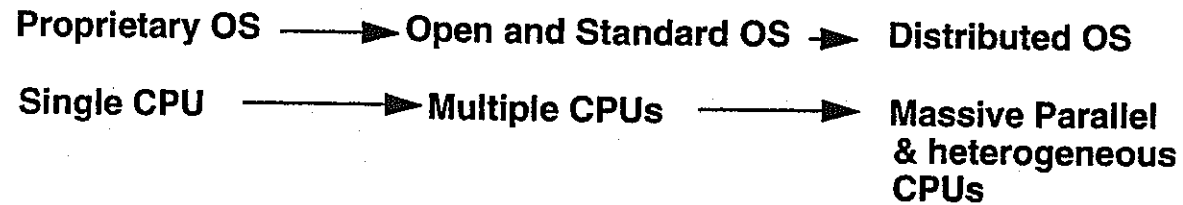
- **Software Strategies**
- **Serverized UNICOS Operating System**
 - Strategies
 - Evolving features/functionality
- **Programming Environments**
 - Compilers/languages
 - Libraries
 - Programming tools



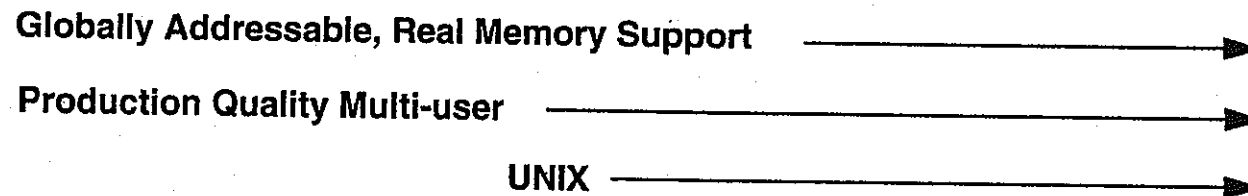
Industry/Technology Trends

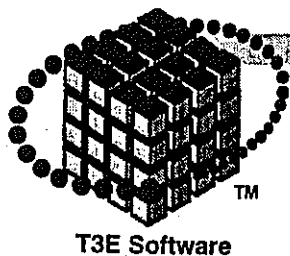


Changing Trends



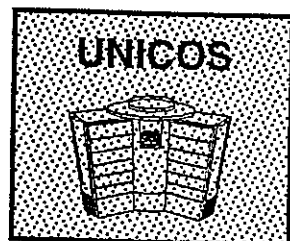
Continuing Trends



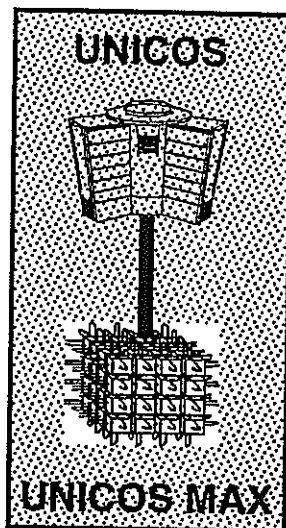


Evolution of UNICOS

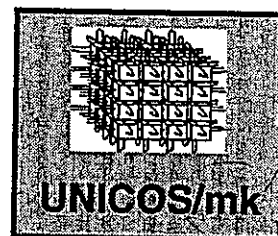
(Common OS for Cray systems)



Multi-threaded
UNICOS on
Vector SMP

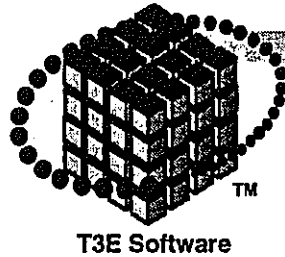


μ K on MPP PEs
with UNICOS host



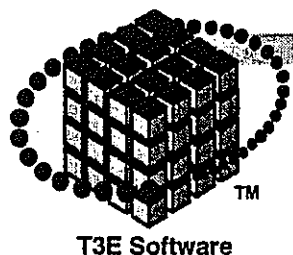
Self hosted μ K
based UNICOS
for MPP

μ K based
UNICOS is
common OS
for future Cray
systems



Serverized UNICOS Strategies

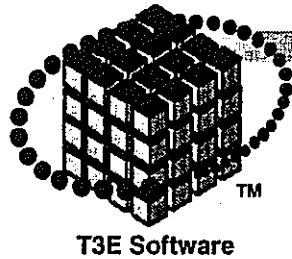
- **Scalability**
 - Efficient memory and processor usage, less replication
- **Single system image**
 - Centralized administration and support
 - Centralized resource management
 - Long term: SSI across multiple, heterogeneous systems
- **Customization**
 - Configure servers to meet workload demands
- **Distributed Operating System**
 - Distribute OS functions for improved scalability of OS services
 - Long term: distribute OS functions across heterogeneous systems for even greater scalability



T3E Software

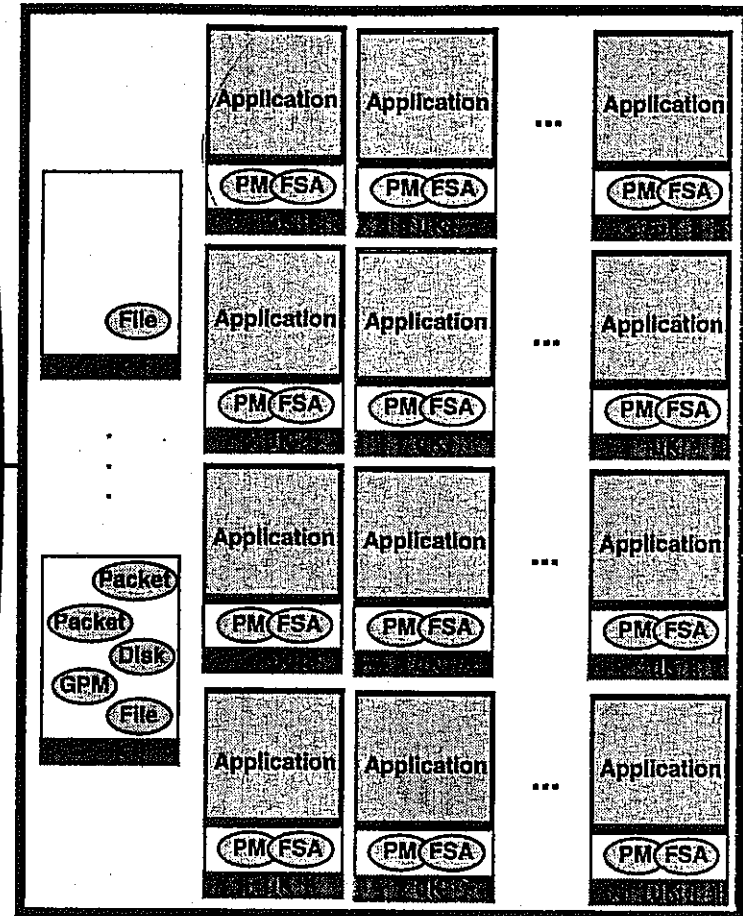
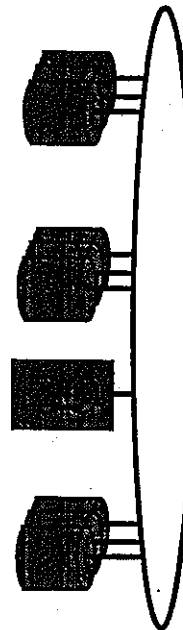
UNICOS/mk Compatibility Requirements

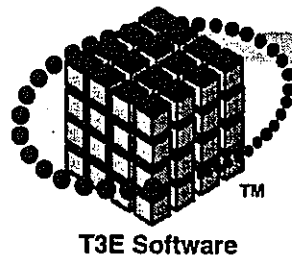
- **Compliance with current and emerging standards**
- **Full functionality, production-quality environment**
 - Support UNICOS extensions: fast I/O, batch, on-line tapes, checkpoint/restart, security, enhanced accounting, multi-user support, etc.
- **Consistent UNICOS interface**
 - Simplify transition from UNICOS on Vector SMP (PVP) systems
 - User-level
 - commands (cc, ls, grep, etc.)
 - utilities (TotalView, make, etc.)
 - Application-level
 - libraries, system calls, signals



UNICOS/mk

- Distributed implementation of UNICOS
- Single OS, single system image
- OS services distributed across processors
- User/application interface equivalent to UNICOS
- Global access to data from any PE





Programming Environments

- **Priorities**

- Single-PE performance (explicit parallel programming)
- Native implementation
- Implicit parallel programming model

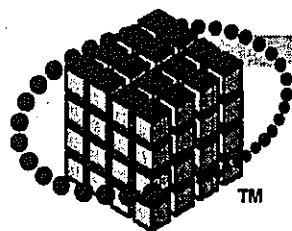
- **Compilers, languages, and programming models**

- **Libraries**

- System libraries
- Scientific libraries

- **Tools**

- Cray TotalView Debugger
- MPP Apprentice



T3E Software

Programming Environment

T3E

High performance
message passing/data passing

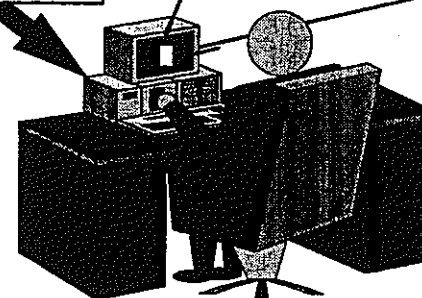
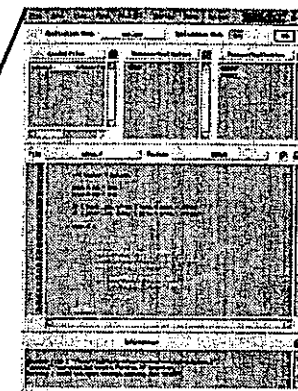
Implicit parallel
programming model

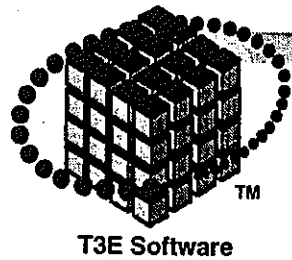
Optimizing compilers

Parallel libraries

Advanced
programming tools

Advanced programming
environment simplifies the creation
of efficient parallel code - allows you
to concentrate on your application,
not the system architecture





Fortran

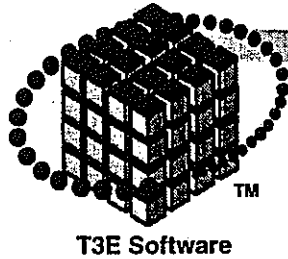
- **CF90 (Fortran 90) Programming Environment**

- Full Fortran 90 compiler (supports Fortran 77 also)
- CF90 2.0 “equivalent” for CF77 for performance/optimization

- **Programming Models**

- Message-passing: PVM and MPI
- Data-passing/one-way messaging: libsma (SHMEM_GET, SHMEM_PUT)
- Implicit global addressing model: HPF and CRAFT

Portability

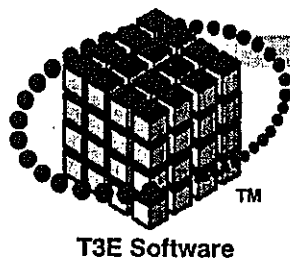


Parallel Programming Model Priorities

- **Portability**
- **Performance**
- **Ease of use**

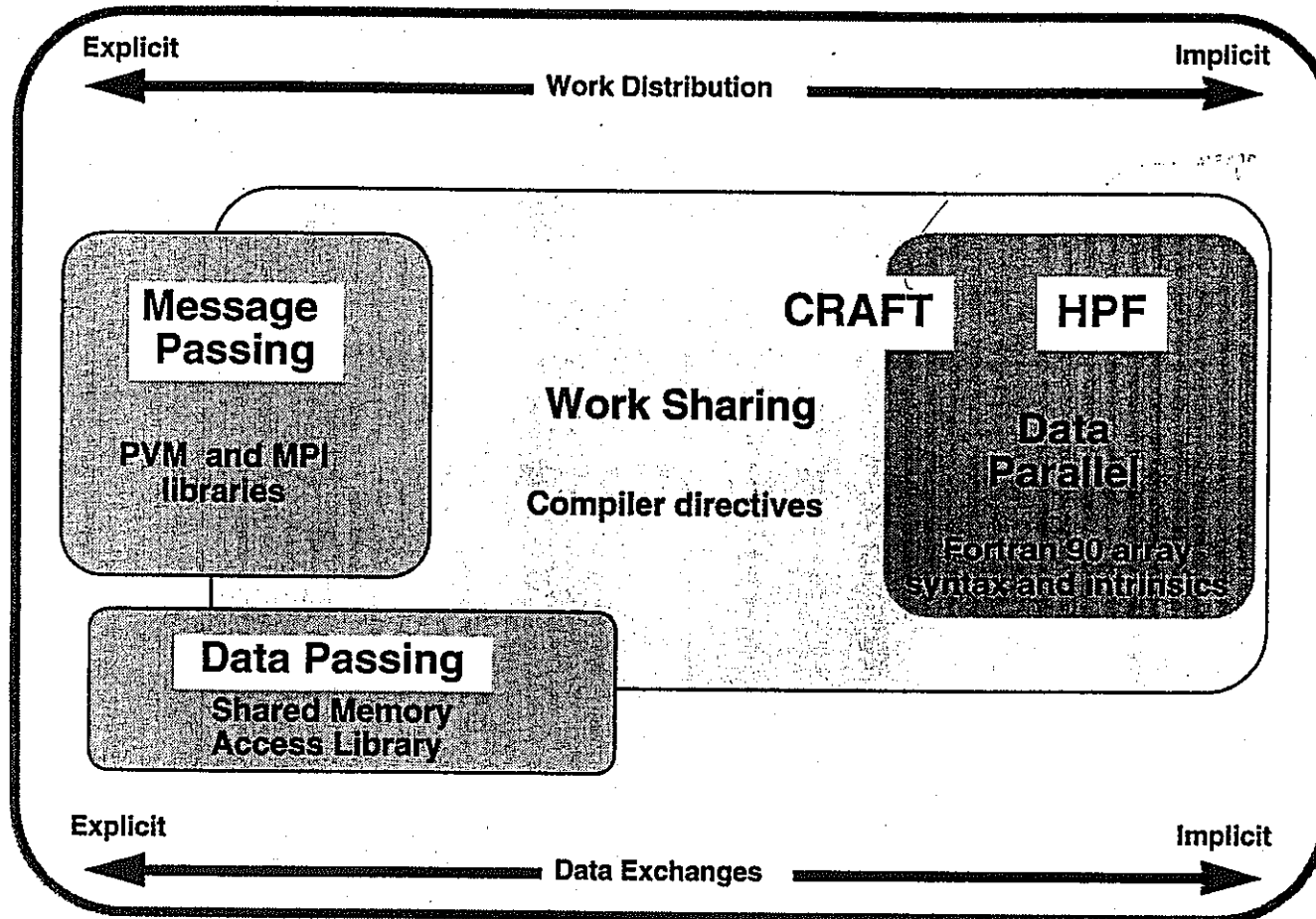
A-147

PNC TN9600 96-004



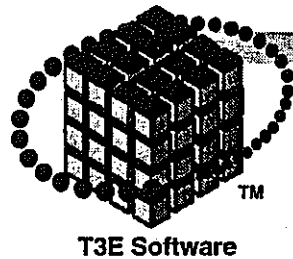
T3E Software

Fortran Parallel Programming Models



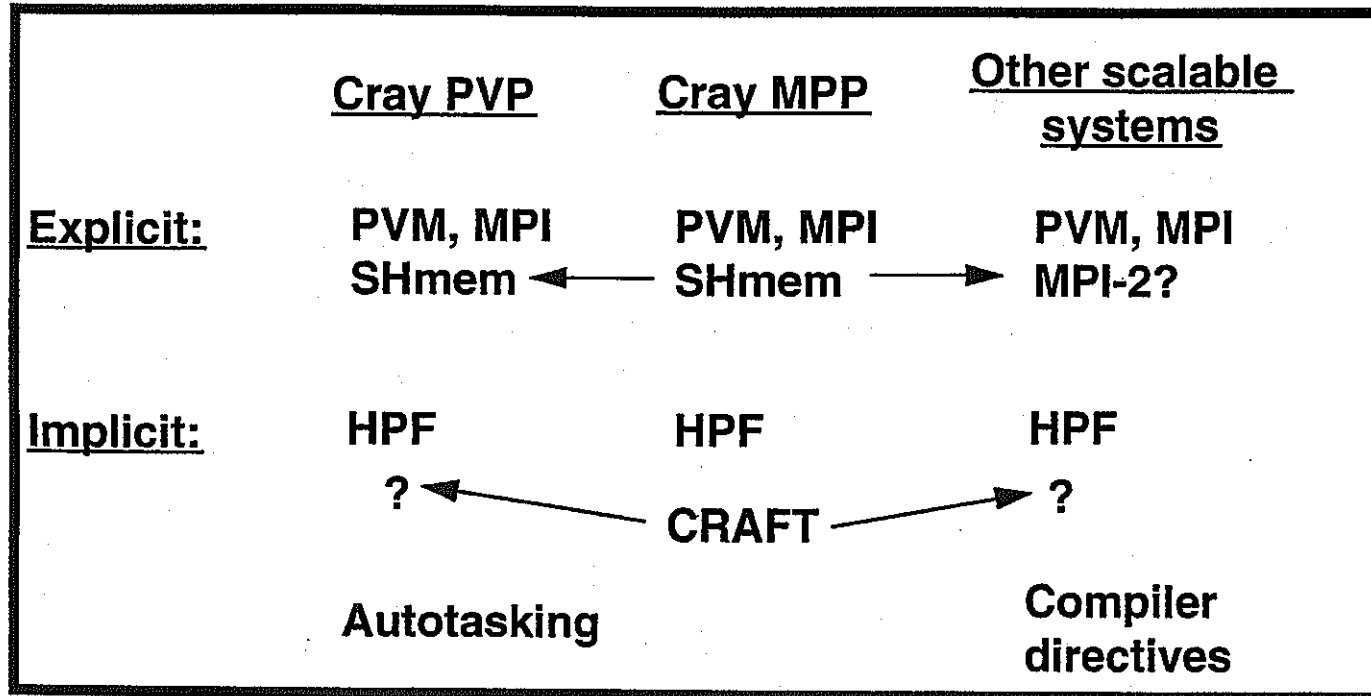
12/12/95
Slide 22

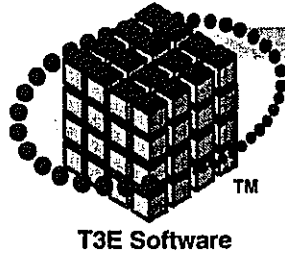
CRAY
RESEARCH, INC.



Parallel Programming Model Directions

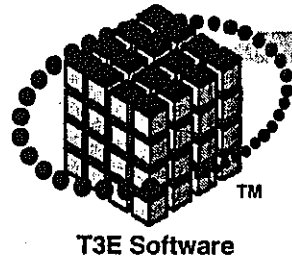
Portability - Performance - Ease of use





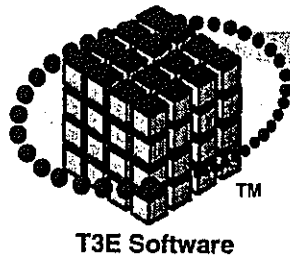
C/C++

- **C++/C Programming Environment**
 - Native C++ compiler
 - ANSI C compiler
- **Programming Models**
 - Message-passing: PVM and MPI
 - Data-passing/one-way messaging: libsmma (shmem_get, shmem_put)
- **Implicit Model?**
 - No clear industry direction in the foreseeable future



Communication and I/O Libraries

- **Communication libraries**
 - PVM-3 message-passing library
 - MPI message-passing library
 - libsm (SHMEM) library
 - get/put transfers, fast global operations, synchronization
- **Standard I/O libraries (libc, libf, libu)**
 - Private Fortran I/O on private data
 - Standard (stdio), asynchronous queued I/O
 - 64-bit Cray to IEEE floating point format conversion

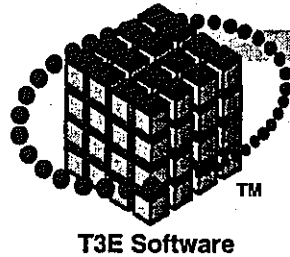


Shared-Memory (*libsma*) Library

- Read or write any word on any PE at any time
- T3E timings and estimates not yet available
 - 1 to 2 microsecond latency on CRAY T3D
 - shmem_put -- up to 130 MB/sec bandwidth on CRAY T3D
- Callable from Fortran and C

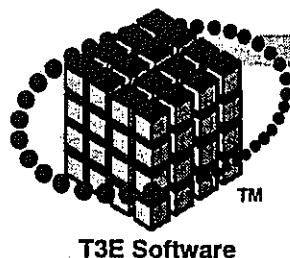
```
shmem_get ( A(10), B(20), I_WORDS, I_PE)
           source  dest.   length  PE#
```

```
shmem_put( A(10), B(20), I_WORDS, I_PE)
           source  dest.   length  PE#
```



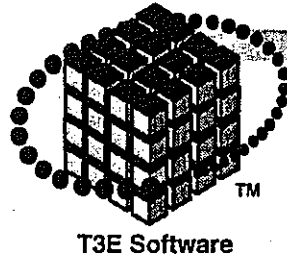
Message-Passing

- **Most portable, scalable programming option**
 - Workstations, clusters, Vector SMP servers, SuperCluster, MPP
- **PVM-3**
 - Compatible and interoperable with ORNL PVM
 - Optimized for PE-PE communications
 - Network connections for communications outside CRAY T3E
- **MPI**
 - Full MPI specification
 - Optimized for PE-PE communications



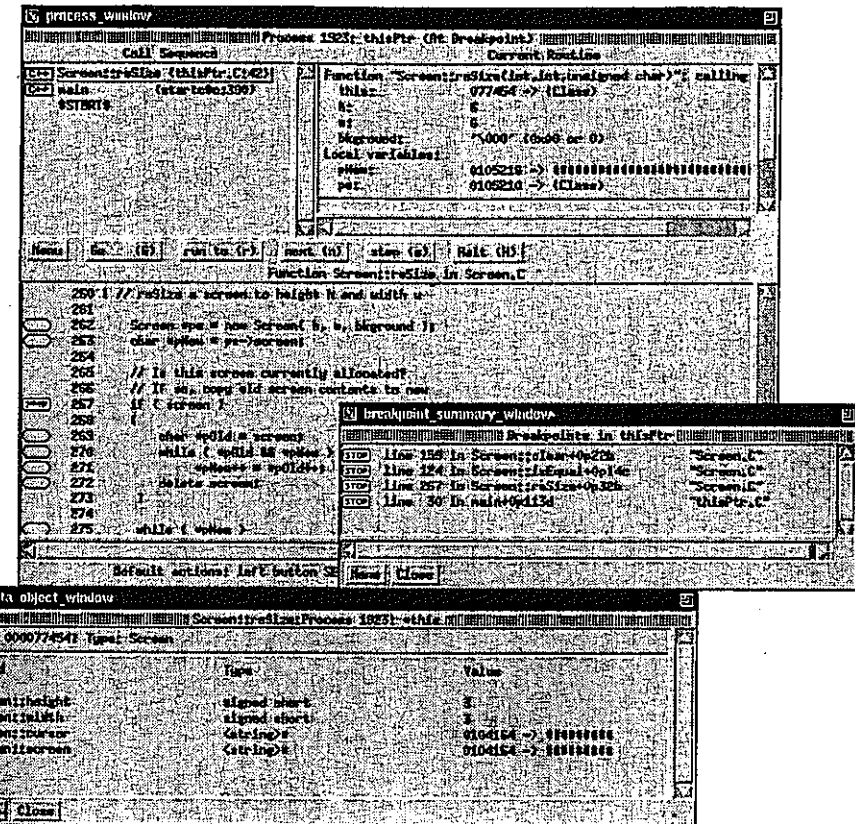
Scientific and Math Libraries

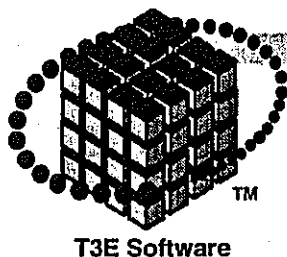
- **Libsci - Cray scientific library**
 - Optimized single-PE and parallel functions
 - BLAS Levels 1, 2, and 3
 - LAPACK routines
 - ScaLAPACK (BLACS, PBLAS, ...)
 - FFTs, solvers, others
- **Mathematical library**
 - 32-bit and 64-bit elementary math functions
- **IMSL and NAG**
 - Third party "industry standard" scientific libraries
 - Written in Fortran, call libsci routines for performance



Cray TotalView Debugger

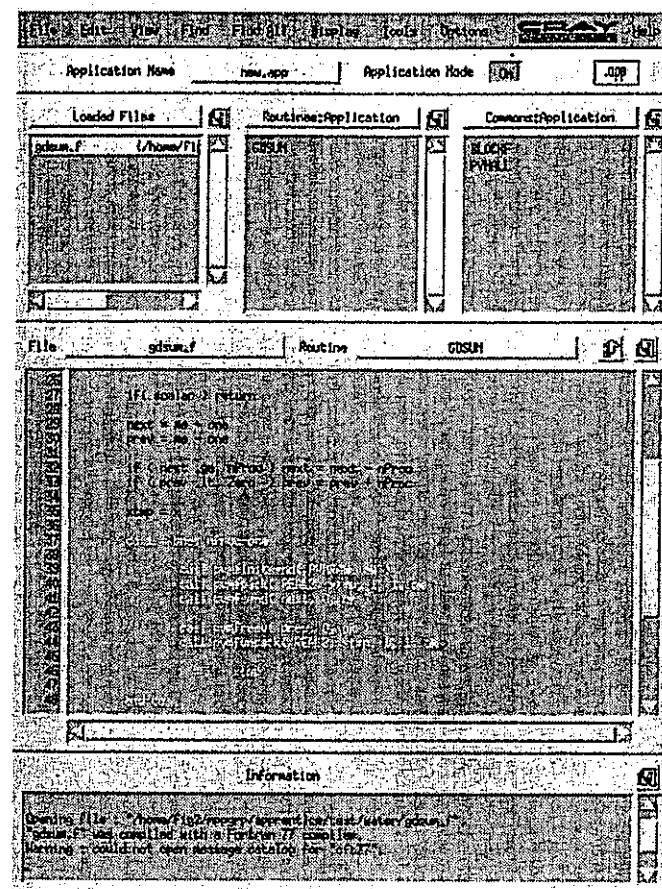
- Based on BBN's TotalView debugger
- Single and multi-PE interactive debugger
- Live, core file, and attach to running process (future)
- Supports Fortran and C/C++
- Supports message-passing and CRAFT-90
- Window-oriented multiprocess debugger





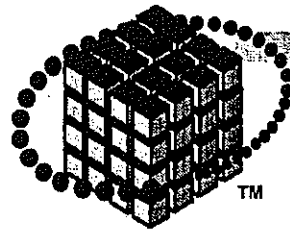
Cray Program Browser

- Interactive environment for browsing and editing application source code
- Language/sensitive code browser
 - Fortran and C
- Source code analysis features
 - Cross Reference
 - Variable tracing



12/12/95
Slide 31

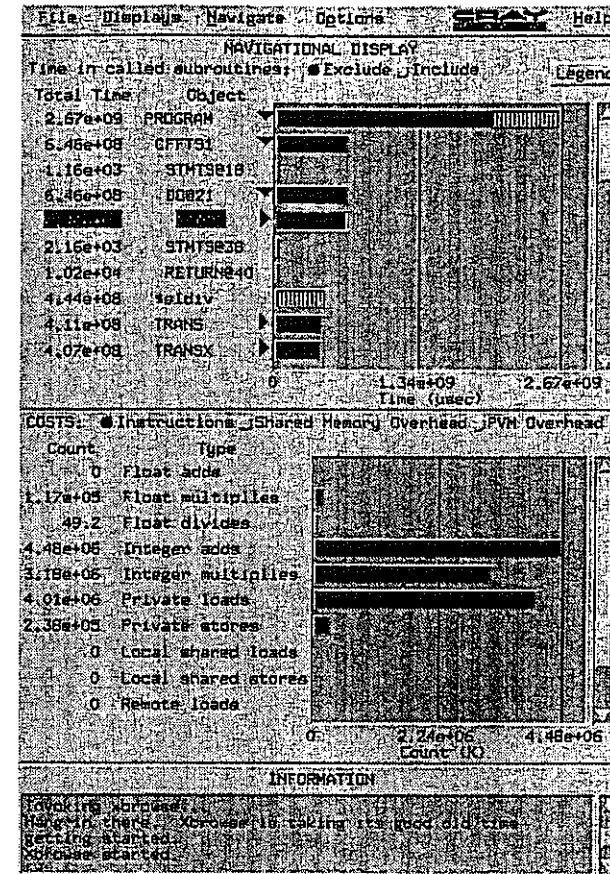
CRAY
RESEARCH, INC.

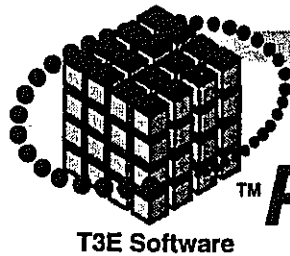


T3E Software

MPP Apprentice

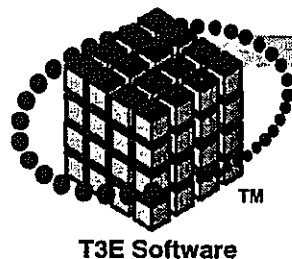
- Source-oriented performance analysis
- Supports Fortran and C/C++
- Supports message-passing and CRAFT-90
- Integrated language-sensitive source code browser
- Scalable, minimized impact on program behavior
- Expert system offers performance observations and suggestions





Delivery Schedule for Programming Environments

- **Early 2Q96 (first T3E volume ships)**
 - CRAY PVP-based distributed programming environments
 - cross-compile
 - Single-PE code generation
 - Message-passing and data-passing applications
- **Late 2Q96**
 - Native programming environments
 - Single-PE performance optimizations
 - Message-passing and data-passing applications
- **4Q96**
 - Implicit parallel programming model support for Fortran 90
 - Library enhancements



CRAY T3E Software

- **UNICOS/mk**
 - Scalable, high performance
 - Production quality
 - Reliable and resilient
- **Programming Environments**
 - High performance
 - Productive tools
 - Advanced parallel programming environment

STAR-CD at Cray Research

Stephen R. Behling

A-160

Third Party Applications

CRAY
RESEARCH, INC.
Applications Division

STAR-CD at Cray Research

- CRI analyst is assigned to port and optimize code
- CRI regularly works with industry partners to demonstrate value of large CFD analyses
- CRI has frequent contacts with Computational Dynamics and ADAPCO (New York)
(London)
- Computational Dynamics has Cray Y-MP EL on site
- CRI is committed to solving our customer's problems as fast as possible

Mesh - CAD $\Rightarrow$ SAMM (Semi-Automatic Mesh Methodology)

STAR-CD History with Cray Research

- **Version 2.1 and earlier**
 - Ran poorly, no optimization had been done
- **Version 2.112+**
 - Significant optimization efforts by CD and CRI.
 - Excellent vector performance for some common routines.
 - No optimization of higher order difference schemes.
 - Single CPU only.
- **Version 2.21**
 - Excellent vector performance for all major sections.
 - Multiple CPU version.
 - Released simultaneously with other platforms.
- **Version 2.3**
 - Preparing for release now.

STAR-CD users on Cray supercomputers

Mercedes Benz

BMW

EDS/Opel

Volkswagen

Audi

Ford (U.S. and Europe)

General Motors

Hyundai

Chrysler

General Electric

Lotus Engr. (UK)

Renault

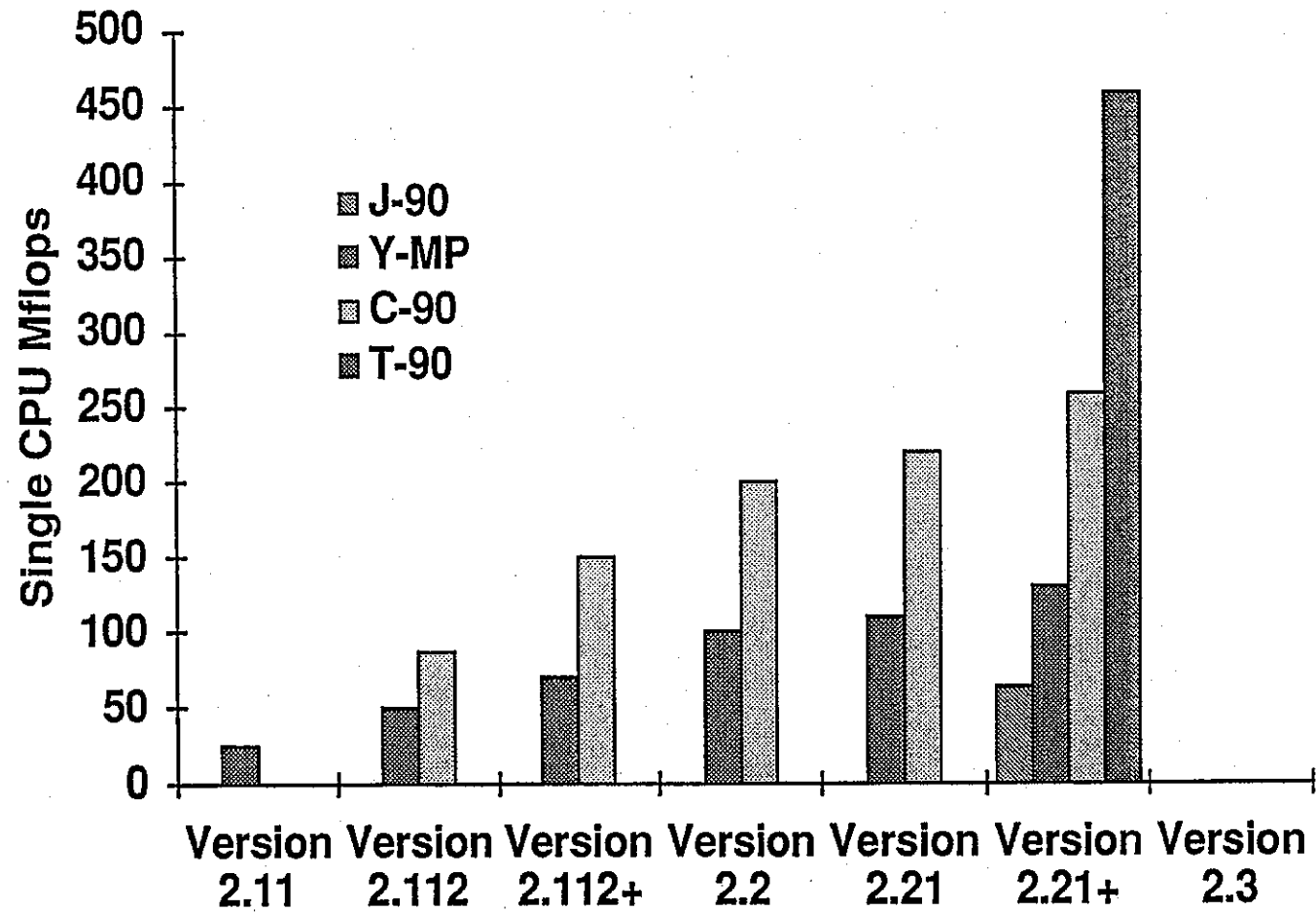
SNCF (France)

Nissan

Nippondenso

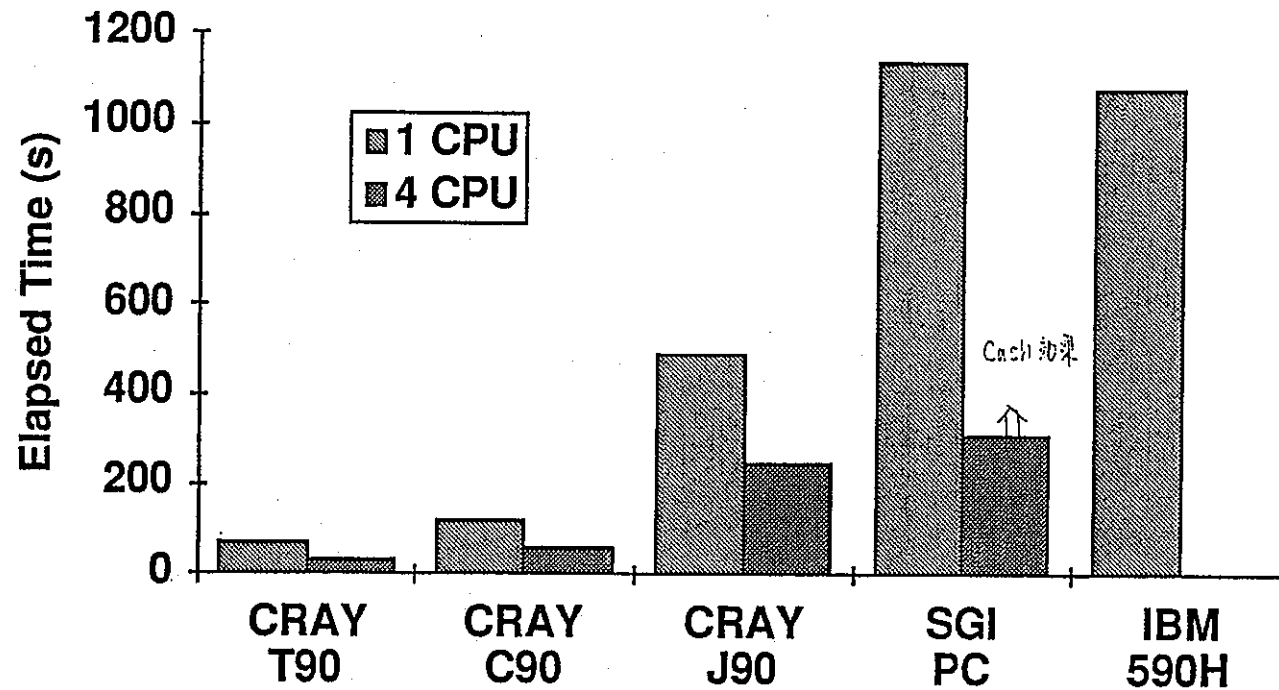
Matsushita

Improvements in STAR-CD Performance



STAR-CD Version 2.210 Performance

Engine benchmark: 157000 cells, 83000 boundaries, 20 iterations

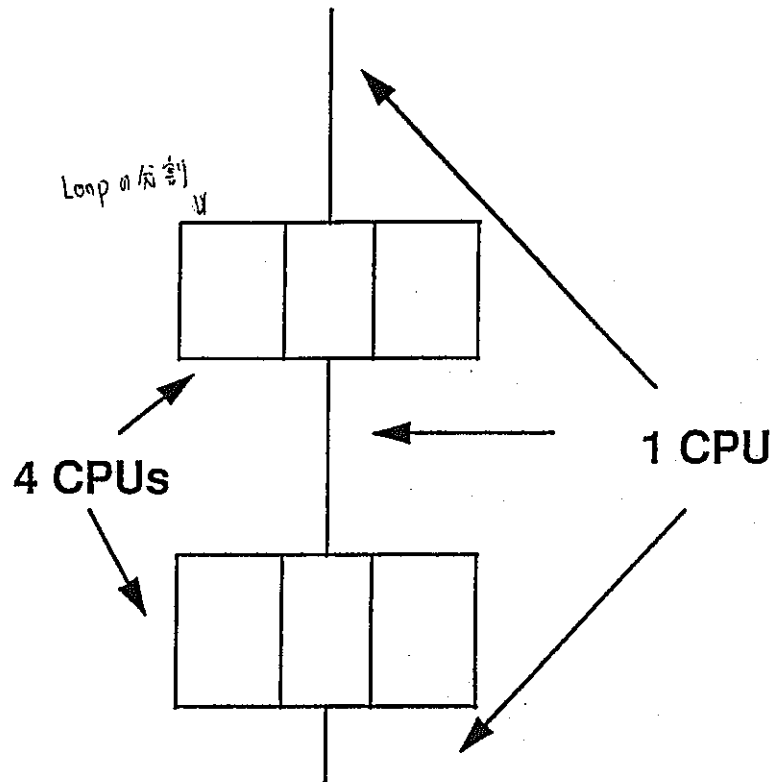


Engineering Applications Group

CRAY
RESEARCH, INC.
Applications Department

STAR-CD Shared Memory Parallelization

Version 2.21 parallelizes at the loop level on shared-memory machines. Serial and parallel sections of code alternate throughout program.



```

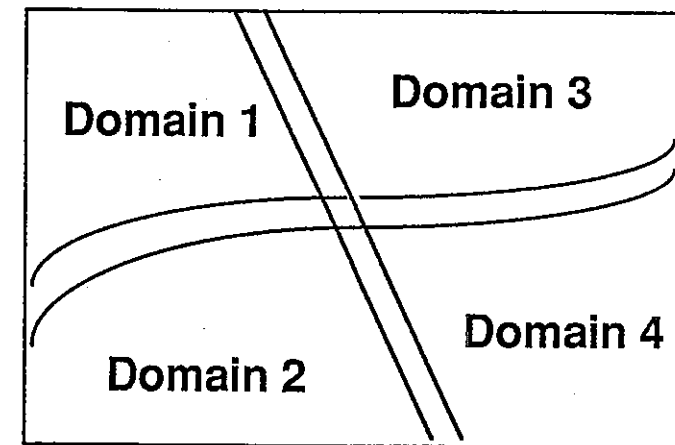
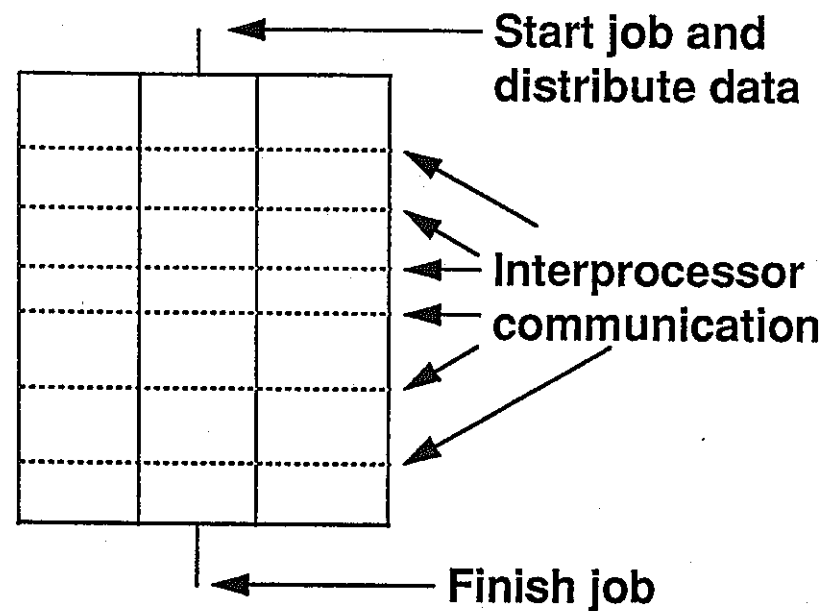
      (setenv NCPUS 4)
      .....
C      serial work
      ITER = ITER + 1
C      parallel work
CMIC$ do autoscope parallel vector
      DO I=1,NCELL
      X(I) = A(I)*S+B(I)
      ENDDO
C      more serial
      Z = X(1) + X(NCELL)
C      more parallel
CMIC$ do autoscope parallel vector
      DO I=1,NCELL
      X(I) = X(I) / Z
      ENDDO
      .....
  
```

E-0717: EUROPORT

STAR-HPC Distributed Memory Parallelization (1996 release)

Problem is divided into subdomains. Each subdomain is a separate task assigned to a separate processor. Communication is needed at common boundaries and for global sums.

FVM



0-processor 依存

0-processor 依存
2009 年 10 月latency
bandwidth

STAR-HPC functionality

- STAR-HPC functionality is a subset of the regular STAR-CD code
- STAR-HPC will initially be based on Version 2.210 and not Version 2.3
- The following models cannot be used in STAR-HPC:
 - Radiation
 - Multi-phase flows (droplets)
 - Mesh connectivity changes using Events
 - Other Events which require concurrent Prostar processing.
- Eventually, all models will be supported in STAR-HPC

STAR-HPC on Cray systems

- **Cray J90**

- CRI has developed shared memory message passing libraries for the parallel-vector machines (Cray J90, C90, T90) based on the Cray Macrotasking techniques.
- CRI will be releasing the Cray Message Passing Toolkit as a product to support PVM, MPI and perhaps other message passing formats.
- Elapsed time on “quiet” machine will be excellent when using STAR-HPC.
- Elapsed time on “busy” machine will be better using the Version 2.3 autotasked code.

- **Cray T3E**

- The Cray T3E will be based on the **DEC EV5** processors with a very high bandwidth network.
- STAR-HPC performance on EV5 is approximately the same as other high end workstations.

HEXARの機能と応用

Automatic, Parallel and Fault Tolerant Mesh Generation from CAD, on Cray Research Supercomputers

望月 義彦

Cray Research, Inc.
Engineering Applications Group
655E Lone Oak Drive, Eagan, MN 55121, USA.
Tel: (1) (612) 683-3645
Fax: (1) (612) 683-3099
Email: mochi@dione.cray.com



CRAY RESEARCH, INC.

説明項目

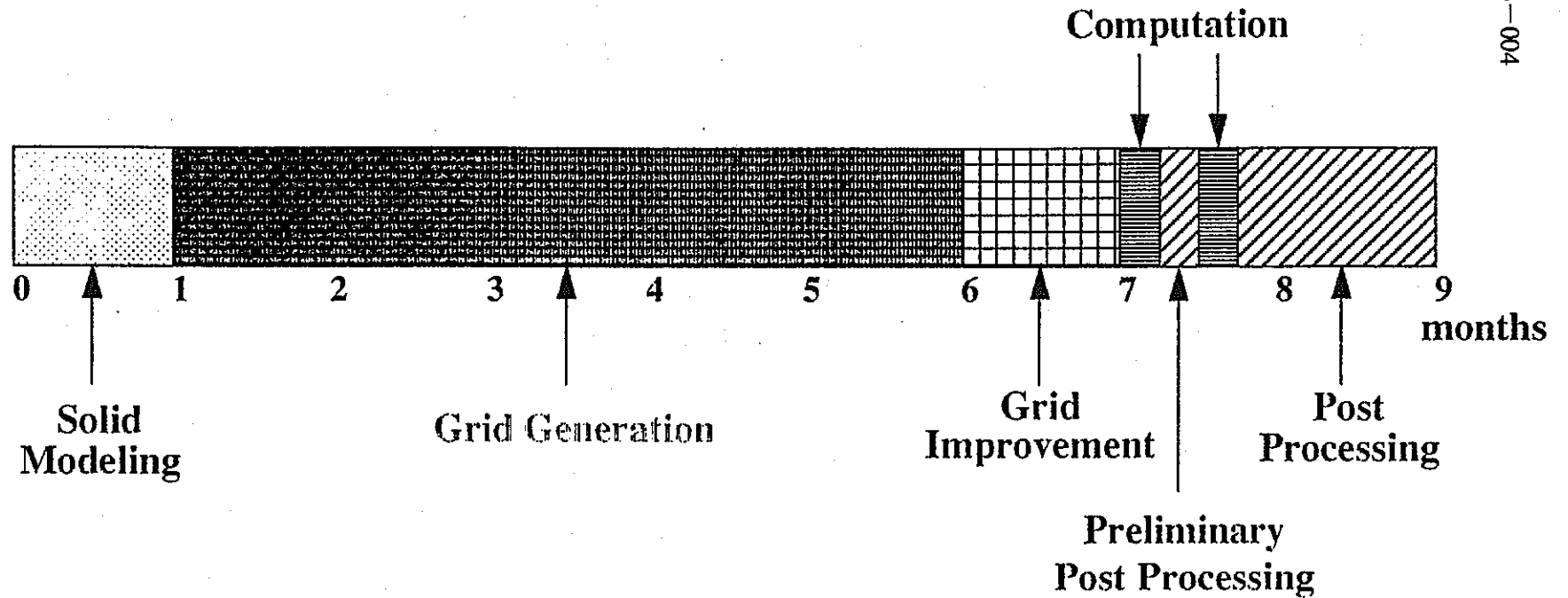
- [1] HEXAR開発の背景
- [2] HEXARの特長
- [3] HEXARの構成
- [4] HEXARの各モジュールについて
- [5] HEXARの応用例
- [6] HEXAR 1.1の機能およびHEXAR開発の今後
- [7] HEXARのデモ・ビデオ
- [8] アンケートについて

**HEXAR : Automatic, Parallel and Fault Tolerant Mesh
Generation from CAD, on Cray Research
Supercomputers**



CRAY RESEARCH, INC.

大規模かつ複雑な流体問題の解析プロセス



Example of a 1,000,000 element Automotive Under-Hood CFD Simulation

**HEXAR : Automatic, Parallel and Fault Tolerant Mesh
Generation from CAD, on Cray Research
Supercomputers**



CRAY RESEARCH, INC.

どうして、メッシュ作成作業が、膨大な時間を必要とするのか？

人間が、ソリッド（解析対象）を目で見て

手作業 で要素を作成しなければ

ならないからである。



**HEXAR : Automatic, Parallel and Fault Tolerant Mesh
Generation from CAD, on Cray Research
Supercomputers**



CRAY RESEARCH, INC.

6面体要素 v.s. 4面体要素 (要素生成手法および解析精度の観点から)

要素生成
手法

6面体要素

これまでは、半自動的な
手法しかなかった。

- BFC法
- PATRAN、I-DEASの手法
- ブロック分割は手動

4面体要素

自動的に要素を分割する手法として

- デラウニ法
 - アドバンシング・フロント法
- が確立されている。

解析精度
(同一自由度)

6面体要素の精度

>

4面体要素の精度

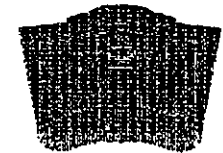


- (a) 要素数 (自由度) を増やす。
(b) 要素の次数を上げる。

>

- 精度を向上させるには、
(a) 要素数 (自由度) を増やす。
(b) 要素の次数を上げる。

**HEXAR : Automatic, Parallel and Fault Tolerant Mesh
Generation from CAD, on Cray Research
Supercomputers**



CRAY RESEARCH, INC.

理想的なメッシュジェネレーターの満たすべき要件

- ・ 高速である。



数秒で、複雑形状ソリッドを1千万要素に分割する。

- ・ 全自動で要素生成を行える。



数行のコマンドを入力するだけで、メッシュを生成できる。

- ・ ロバストである。



CADデータに不具合（ギャップ・オーバーラップなど）が存在しても、メッシュを生成できる。

**HEXAR : Automatic, Parallel and Fault Tolerant Mesh
Generation from CAD, on Cray Research
Supercomputers**



CRAY RESEARCH, INC.

HEXARとは

Crayのスーパーコンピュータを使用し、
CADデータに不具合が存在しても、
自動的に複雑形状ソリッドを6面体要素
に分割するメッシュジェネレーター

**HEXAR : Automatic, Parallel and Fault Tolerant Mesh
Generation from CAD, on Cray Research
Supercomputers**



CRAY RESEARCH, INC.

HEXARの特長

- ・一般的なCADシステムの表面形状データをそのまま利用できる。
- ・CADデータに不具合（ギャップ、オーバーラップなど）が存在しても対処できる。
- ・ソリッドが複数存在する場合もメッシングできる。
- ・ソリッドの表面形状を認識し、自動的に要素密度を決定できる。
- ・ソリッドの内部に空洞が存在しても、3次元ボリュームスキャン機能により対処できる。
- ・表面境界領域のメッシュ層数を自由に指定できる。
- ・高レベルのスミージング機能を有する。
- ・メッシングを高速に行える（CRAYのベクトル・パラレル処理機能）。
- ・ソリッドの主軸が直交座標系に対して傾いていても対処できる。

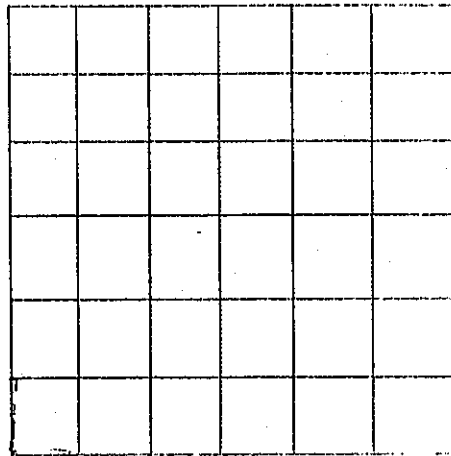
**HEXAR : Automatic, Parallel and Fault Tolerant Mesh
Generation from CAD, on Cray Research
Supercomputers**



CRAY RESEARCH, INC.

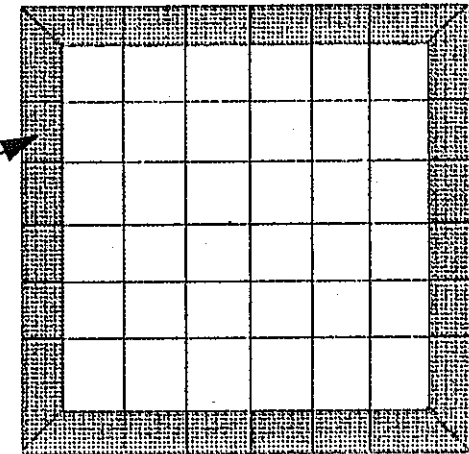
表面境界領域のメッシュ層とは？

hexar



構造メッシュ
(マップファイル)

hexarpost



表面境界領域
のメッシュ層

非構造メッシュ

**HEXAR : Automatic, Parallel and Fault Tolerant Mesh
Generation from CAD, on Cray Research
Supercomputers**

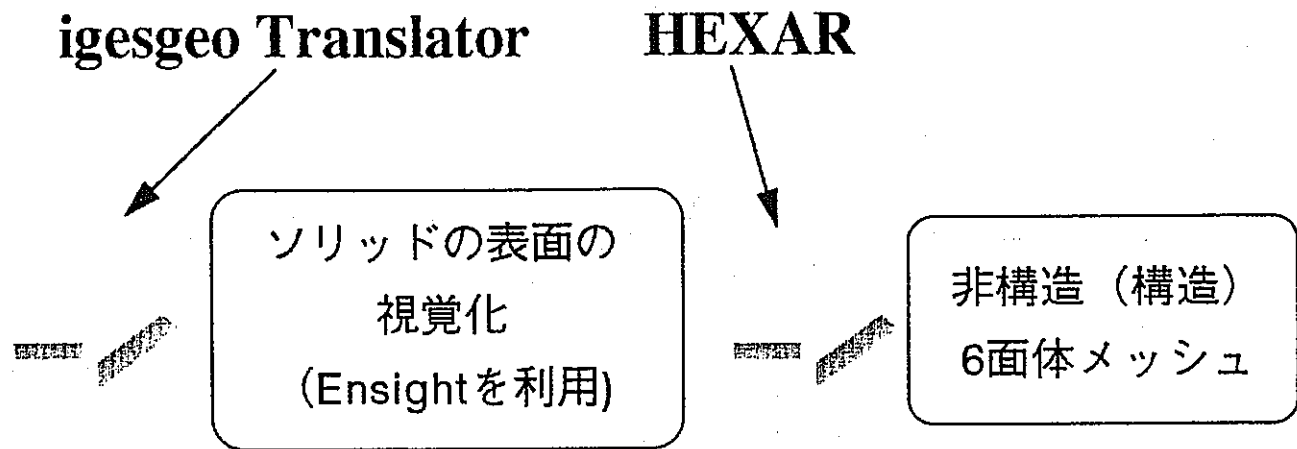


CRAY RESEARCH, INC.

CAD Systems

- CADAM
- CALMA
- CATIA
- Computervision
- Ford_PDGS
- Geodraw
- GM_CGS
- Intergraph
- PROCADAM
- VDA
- PRO-Engineer
- Nissan
- EUCLID-IS
- UniGraphics
- SET
- AutoCad

HEXARを用いたメッシング処理のフロー

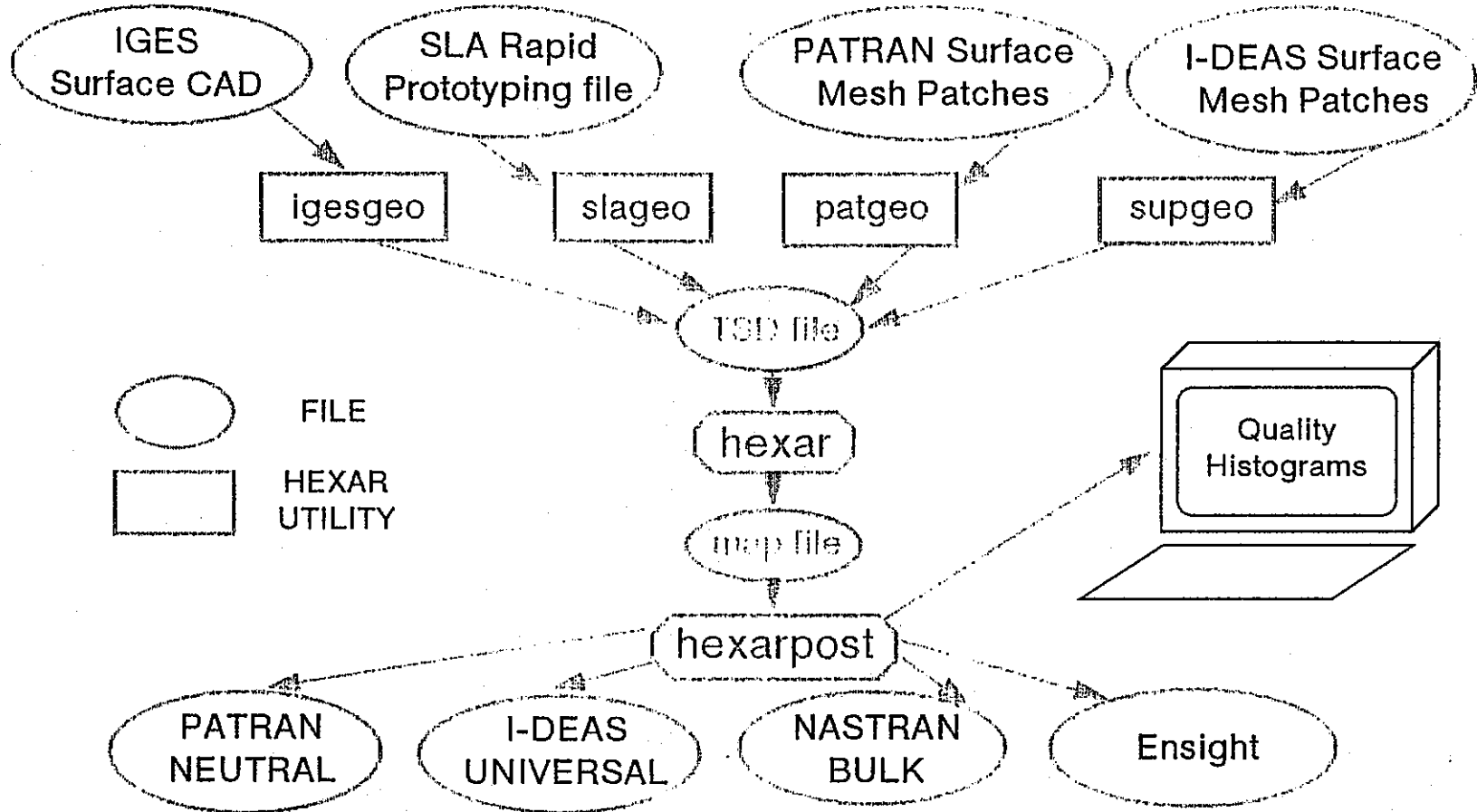


**HEXAR : Automatic, Parallel and Fault Tolerant Mesh
Generation from CAD, on Cray Research
Supercomputers**

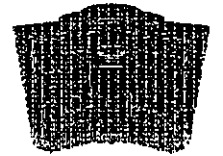


CRAY RESEARCH, INC.

HEXARのシステム構成



**HEXAR : Automatic, Parallel and Fault Tolerant Mesh
Generation from CAD, on Cray Research
Supercomputers**



CRAY RESEARCH, INC.

各種ユーティリティーについて

： 汎用的な表面CADデータを、hexar[HEXARの本体]の入力ファイルTSD(Triangulated Surface Definition)ファイルに変換する。

(a) igesgeo : IGES CAD Surface Definitionファイルを、TSDファイルに変換する。
利用法 → igesgeo -i aero.iges -o aero.t

(b) slageo : 3-D Systems社のSLA(Stereo Lithography Apparatus)フォーマットファイルをTSDファイルに変換する。

(c) patgeo : 3角形要素で構成されたPATRAN neutral フォーマットの表面メッシュファイルをTSDファイルに変換する。

(d) supgeo : 3角形要素で構成されたI-DEAS universal フォーマットの表面メッシュファイルをTSDファイルに変換する。

(b), (c), および(d)は、コマンドを入力した後、入力ファイル名および出力ファイル名を対話的に入力する。

**HEXAR : Automatic, Parallel and Fault Tolerant Mesh
Generation from CAD, on Cray Research
Supercomputers**



CRAY RESEARCH, INC.

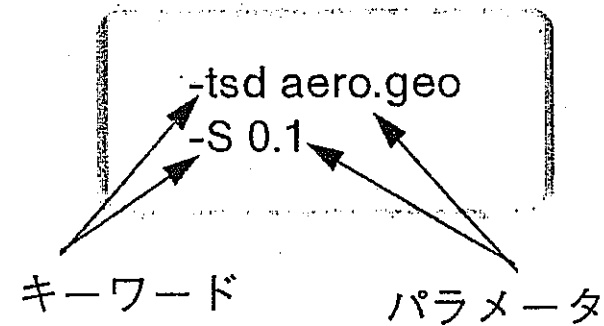
hexar[HEXARシステムの本体]について

: TSDファイルを読み込んで、6面体要素のメッシュを生成するとともに hexarpostの入力データであるmapファイルを出力する。

利用法 :

表面のメッシュデータのためのバイナリーファイル

- [1] grid.dataを作成する。
grid.dataの例、



- [2] hexarコマンドを入力する。

**HEXAR : Automatic, Parallel and Fault Tolerant Mesh
Generation from CAD, on Cray Research
Supercomputers**



CRAY RESEARCH, INC.

hexarpostについて

: mapファイルを読み込んで、以下の処理を対話的に行い、各種の汎用フォーマットに対応したメッシュファイルを出力する。

- (a) 表面境界領域にメッシュ層を追加する。
- (b) メッシュを緩和する（スムージングをかける）。
- (c) 各種のヒストグラムを表示する。—————▶

要素の

- ・体積
- ・アスペクト比
- ・辺のなす角
- ・ジャコビアン[27点]
- ・面のゆがみ度

: hexarpostのコマンドを入力した後、[mapxxxxx i]を入力する。

**HEXAR : Automatic, Parallel and Fault Tolerant Mesh
Generation from CAD, on Cray Research
Supercomputers**



CRAY RESEARCH, INC.

HEXARの応用例

(a) HEXARの有効性を示す適用される分野

(1) 自動車

- Under-hoodの熱解析
- 車内の空気流の解析
- エンジンブロックの熱流動解析
- 車体の外部流の解析

(2) 航空機

- 機体の外部流の解析、客室内の空気流の解析

(3) 製造

- 鋳造シミュレーション、プラントの空調関連の解析

**HEXAR : Automatic, Parallel and Fault Tolerant Mesh
Generation from CAD, on Cray Research
Supercomputers**

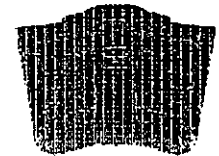


CRAY RESEARCH, INC.

(b) これからの応用分野

- (1) 医学 → 超音波・MRIイメージを利用した流体（血液）、力学、電磁場解析
- (2) 環境 → 建築用のCADデータを利用した大規模建造物の風および火事の解析
- (3) 宇宙工学 → ロケットの固体燃料の燃焼解析
- (4) 電子工学 → 大規模かつ実際的な熱解析
- (5) 石油産業 → 石油タンクに関する解析
- (6) 物理 → プラズマ解析、結晶の成長に関する解析
- (7) 製造 → 鍛造シミュレーション、タンク内における液体の混合解析
- (8) 土木 → 河川・港湾の流体解析、土壌力学

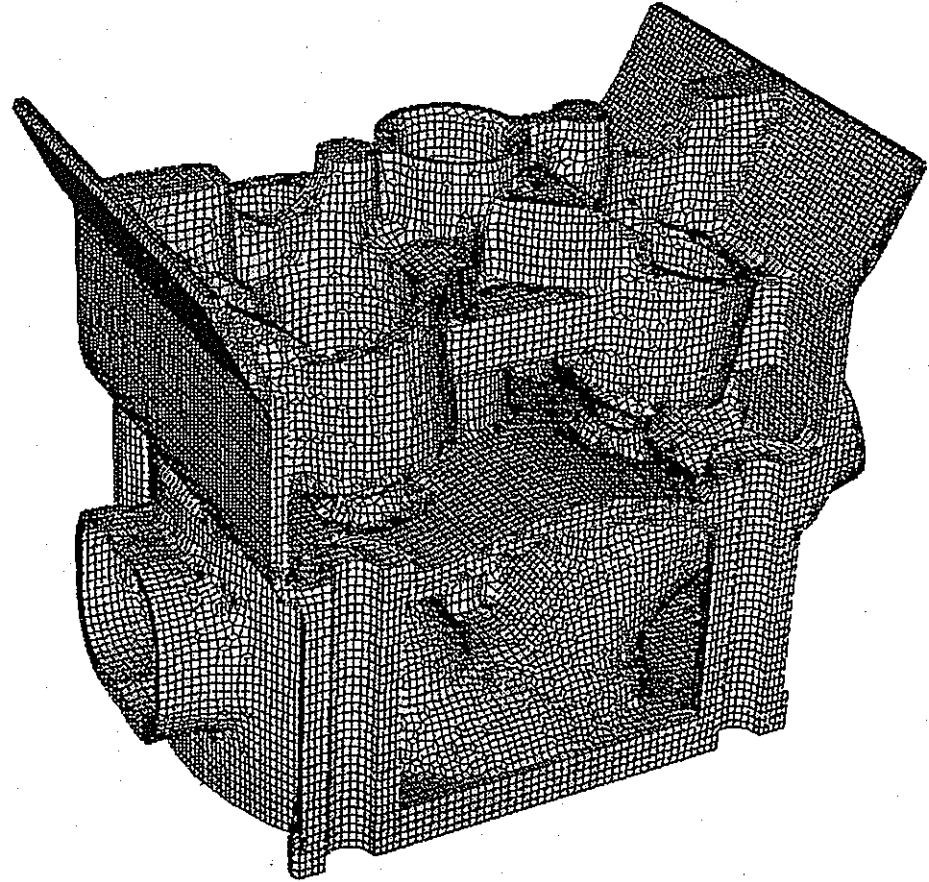
**HEXAR : Automatic, Parallel and Fault Tolerant Mesh
Generation from CAD, on Cray Research
Supercomputers**



CRAY RESEARCH, INC.

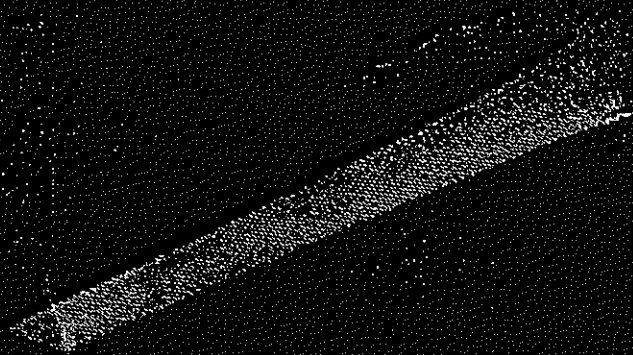
HEXARの応用事例

- ・ エンジンブロックヘッド
 - SLAファセット数20000
 - 節点数340000
 - メモリ32Mw
 - C90CPU時間20分



BLOOD FLOW THROUGH A BRANCHING ARTERY

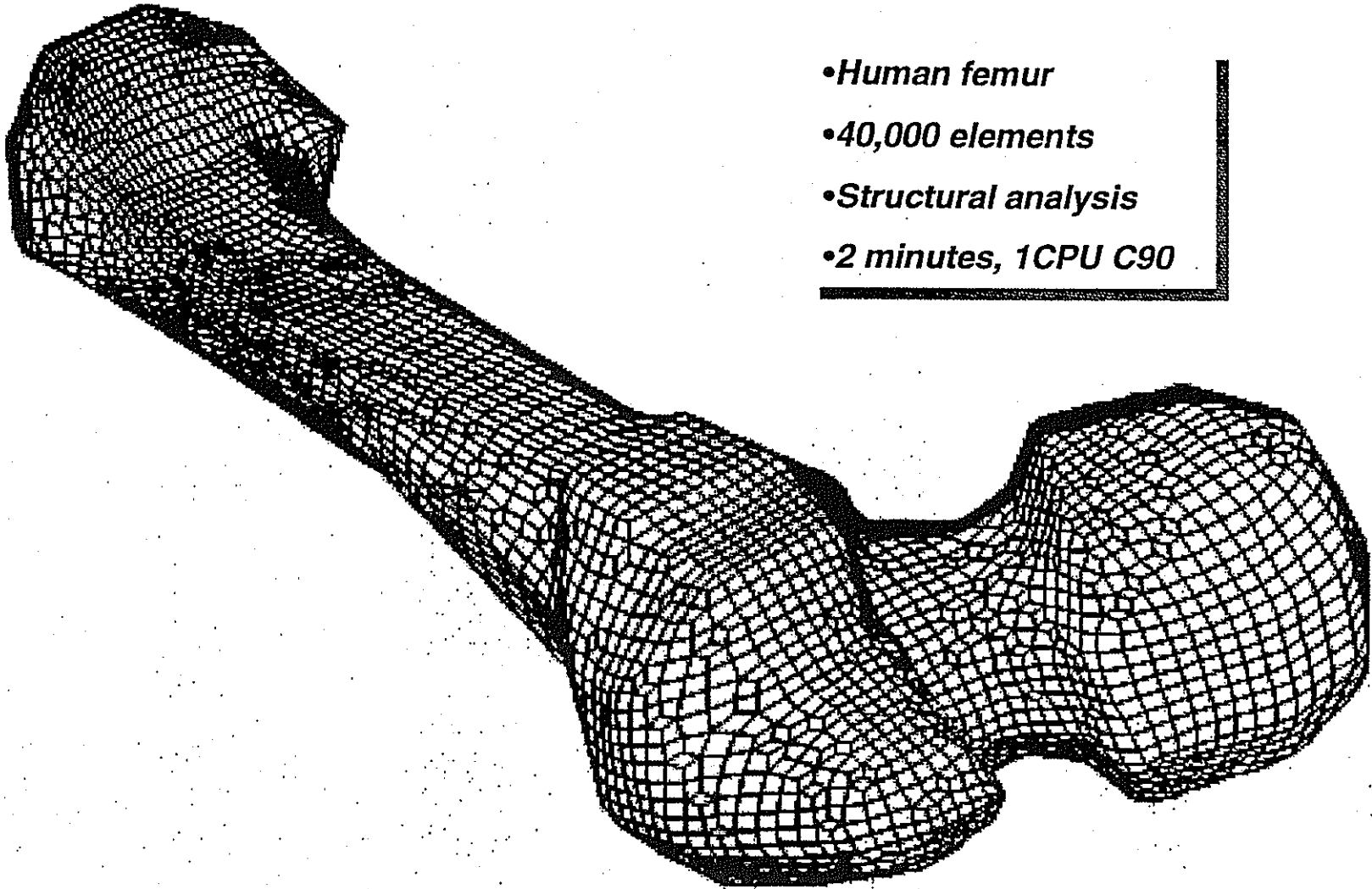
- *Artery*
- *200,00 elements*
- *Blood flow*
- *1 minute, 1CPU C90*

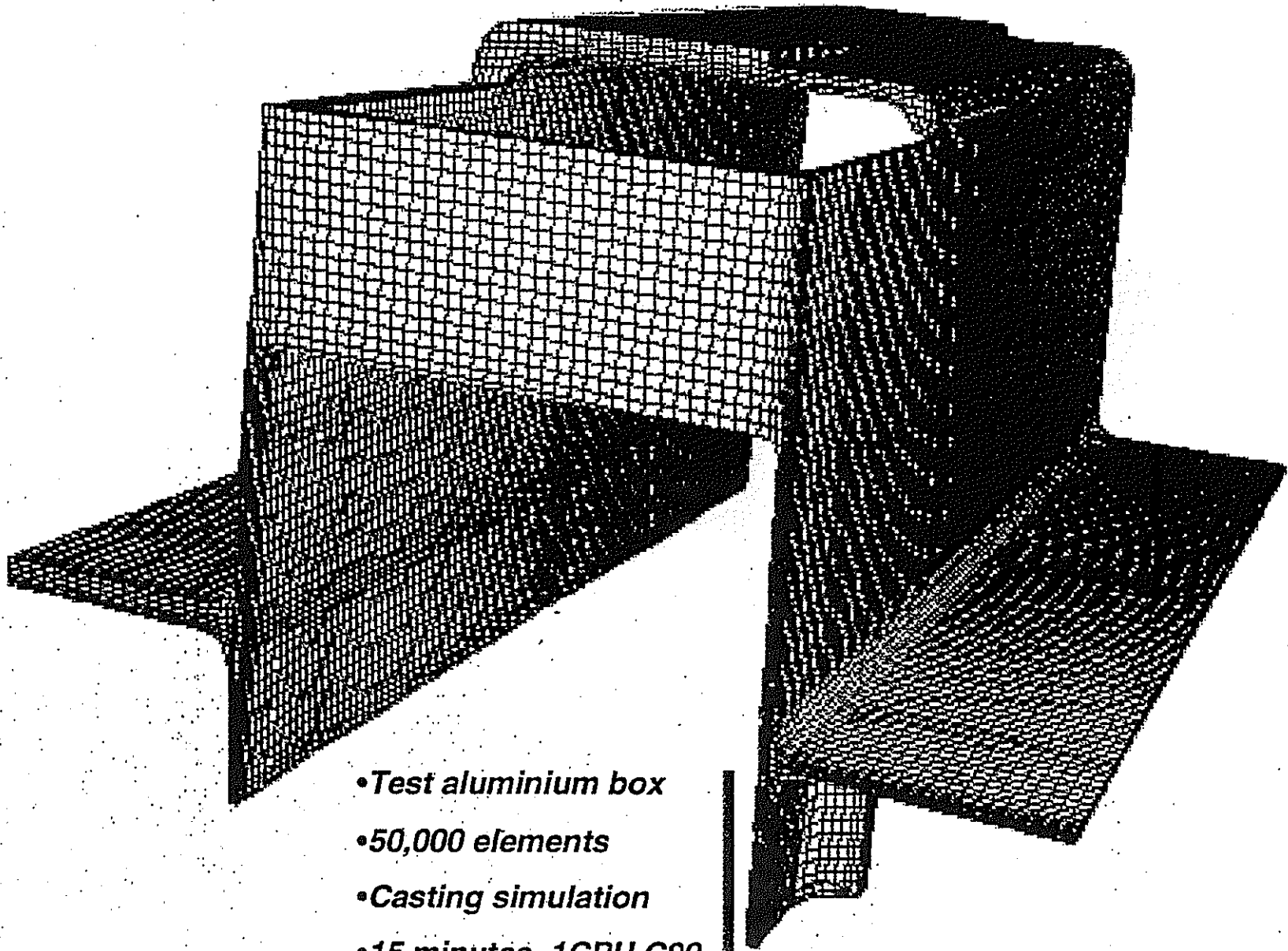


ANALYSIS WITH STAR
MESH WITH HEXAR 1.1

3M CORPORATION

- *Human femur*
- *40,000 elements*
- *Structural analysis*
- *2 minutes, 1CPU C90*





- *Test aluminium box*
- *50,000 elements*
- *Casting simulation*
- *15 minutes, 1CPU C90*

<APPENDIX 6>

ミシシッピ大学エンジニアリング研究センター
入手資料

Numerical Simulation Of Ultra-Relativistic, Heavy-Ion Collisions

Hazem Tleimat^{1,2}, R. B. Piercey¹ and B. K. Soni²

(1) Department of physics and astronomy.

(2) NSF Engineering Research Center
Mississippi State University
Mississippi State, MS 39762

Abstract

The collision of two ultra-relativistic, heavy-ions is simulated by solving the time dependent relativistic hydrodynamics equations on an Eulerian grid. A high resolution second order upwind scheme based on the Harten Lax Van Leer (HLL) method is used in the calculations. We have extended this method for simulating ultra-relativistic, heavy-ion collisions. This strategy is shown to be faster and more accurate than other published methods for the simulation of ultra-relativistic heavy-ion collisions.

Introduction

Under normal conditions like those in an isolated atomic nucleus, nuclear matter is considered to be essentially incompressible. This 'normal' nuclear matter has a density of $\rho = \rho_0 = 0.15 \text{ fm}^{-3}$ and a unique volume energy per nucleon. In order to study the properties of nuclear matter at higher densities, extreme compressional forces are required.

The compression of nuclear matter is thought to occur in two instances in nature, in the hearts of neutron stars, and during the supernova collapse. However, only limited observation of these two instances can be made. The recent development of very high energy accelerators is providing new opportunities to study the properties of nuclear matter at densities, $\rho \gg \rho_0$ and/or temperatures $T > B_0 = 16 \text{ MeV}$.

In the ultra-relativistic, heavy-ion collisions, that can currently be produced in large

accelerators, energies of up to 100 GeV per nucleon can be obtained. This energy corresponds to a Lorentz gamma factor of $\gamma = 10$. In future accelerators, energies of up to tera electron volts are anticipated. Due to the high temperature, pressure and energy involved in such collisions a new, compressed form of nuclear matter may be produced. Current theories, based essentially on the standard model of elementary particles, predict a new state of nuclear matter called the quark-gluon plasma (QGP). It is not yet clear what conditions may lead to this new state, but compressions of up to five times normal density may be required. The possibility of producing the quark-gluon plasma, in relativistic heavy-ion collisions is currently the subject of intense theoretical and experimental investigations.

The conditions for the formation and evolution of the QGP have been investigated using several techniques. These can be roughly divided into microscopic and collective models. One of the most successful collective models is the relativistic nuclear hydrodynamics model. In this model, the nucleus is considered to consist of a compressible fluid and the methods of fluid dynamics are used to describe the dynamical properties. The collective motion of the fluid is used to describe the system, instead of the interaction among the individual particles of the system [3].

The validity of the model depends on the assumption that the mean free path is sufficiently small and the relaxation time is sufficiently small that local equilibrium can be established. This allows the laws of classical hydrodynamics to be applied to the problem.

While there is some question as to the validity of the assumption of local equilibrium, the hydrodynamics model has been very successful in describing some of the reaction phenomena [14]. For example, the hydrodynamics model is particularly well suited to describe phenomena such as shock formation and phase transitions.

At high energies relativistic effects become important and must be included in the formulation of the model. The relativistic nuclear hydrodynamics model has been very important in the development of the theory of high-energy, heavy-ion collisions. One of the most important successes was the discovery of collective flow through a comparison between experiments. Transverse momentum, which was predicted by the nuclear hydrodynamics model, was observed experimentally [17].

The collision of two heavy-ions at high energy is represented schematically in Figure 1, which illustrates four distinct stages. The collision is assumed to take place in the center of momentum reference frame [11, 14]. In stage one, assuming straight line geometry, the nuclei approach each other with non zero impact parameter. Although the nuclei are spherically symmetric in their rest frames, they become significantly distorted in the lab frame. This distortion is due to the well known Lorentz contraction. In stage two, a region of overlap, known as the participant region, develops. The other parts of the nuclei, the spectator regions, which did not participate in the collision keep moving without any knowledge the collision. The separation of the participant and spectator regions occurs due to the fact that the relative speed of the nuclei is faster than the speed of sound in the nuclear matter (about $0.2c$). In stage three the participant region rapidly comes into thermal equilibrium producing high temperature and pressure. If the temperature and pressure are high enough, a QGP may develop. Stage four represents the cooling process, in which the participant region disintegrates emitting particles and photons. In the first stages of the collision depicted the figure, the nucleus may be described as a fluid and the collision considered as hydrodynamic in nature.

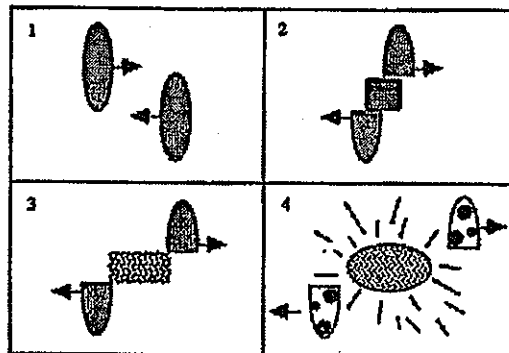


Figure 1. A schematic representation of a nuclear-nuclear collision at high energy.

Several different methods and techniques have been developed to solve the relativistic hydrodynamics equations. Schneider et al. [13] have described three explicit methods based on ultra-relativistic nuclear hydrodynamics. In their paper, they used (HLLE) a high resolution scheme [8,9], which was supplemented by ideas from Einfeldt [6]. Schneider et al. adapted the HLLE scheme for relativistic conditions. They also used the ideas of Book and Boris [1] of flux corrected algorithms and the extensions, SHASTA [2]. Dean et al. [3,4] have developed a simulation based on an implicit, basis-spline, collocation approach. They solved the one dimensional problem and compared it with the analytical solution. They claimed that their method can be applied to non-uniform grids. A comparison between the result of the two groups was done [5] using the extension of the spline method and the SHASTA method in three dimensional calculations. Trivedi et al. [16] have calculated the ultra relativistic heavy ion collision using the first order relativistic HLLE. In this work we extend the HLLE scheme to the second order.

Relativistic Nuclear Hydrodynamics Equations and The Equation of State

The relativistic nuclear hydrodynamics equations can be derived from the classic hydrodynamics equations, after taking into account the relativistic factor and the nuclear equation of state. The equation that describes the nuclear hydrodynamics problem in one space dimension is

$$\frac{\partial U}{\partial t} + \frac{\partial F}{\partial x} = 0 \quad (1)$$

where U and F are the following vectors :

$$U = \begin{bmatrix} R \\ M_u \\ E \end{bmatrix} \quad F = \begin{bmatrix} Ru \\ M_u u + p \\ (E + p)u \end{bmatrix} \quad (2)$$

In three space dimension the equation takes the form:

$$\frac{\partial U}{\partial t} + \frac{\partial F}{\partial x} + \frac{\partial G}{\partial y} + \frac{\partial H}{\partial z} = 0 \quad (3)$$

Where U, F, G and H are the following vectors :

$$U = \begin{bmatrix} R \\ M_u \\ M_v \\ M_w \\ E \end{bmatrix} \quad F = \begin{bmatrix} Ru \\ M_u u + p \\ M_v u \\ M_w u \\ (E + p)u \end{bmatrix} \quad (4a)$$

$$G = \begin{bmatrix} Rv \\ M_u v \\ M_v v + p \\ M_w v \\ (E + p)v \end{bmatrix} \quad H = \begin{bmatrix} Rw \\ M_u w \\ M_v w \\ M_w w + p \\ (E + p)w \end{bmatrix} \quad (4b)$$

R is the mass density, and M_u , M_v and M_w are the momentum density on x, y and z respectively. M the total momentum density, can be written as

$$M = \sqrt{M_u^2 + M_v^2 + M_w^2}$$

and E is the energy density.
u, v and w are the value of the velocity on x, y and z respectively. The total velocity is:

$$V = \sqrt{u^2 + v^2 + w^2}.$$

These quantities are related to the laboratory frame quantities via the relativistic transformation equations:

$$\begin{aligned} R &= \gamma n \\ E &= \gamma^2 (e + p) - p \\ M &= \gamma^2 (e + p)V \\ \gamma^2 &= \frac{1}{1 - V^2} \quad V \leq 1 \end{aligned} \quad (5)$$

and the equation of state (the relativistic case):

$$p = (\Gamma - 1)(e - n) \quad (6)$$

where Γ is the gas constant.

The relativistic nuclear equation of state is different from the ideal gas equation of state, since the kinetic energy and the internal energy both have to be considered. For the case of ultra-relativistic nuclear hydrodynamics, when $e \ll n$, and $\gamma \approx 10$, n can be neglected. Thus, the equation of state becomes

$$p = (\Gamma - 1)e \quad (7)$$

R, M, E and V are related via the equation

$$\begin{aligned} g(v) &= [\Gamma(E - MV) - M(1 - V^2)]^2 \\ &- (1 - V^2)V^2(\Gamma - 1)^2 R^2 = 0 \end{aligned} \quad (8)$$

To find the velocity, the above equation has to be solved. Finding procedure like Newton-Raphson method can to be used. In the case of the ultra-relativistic equation of state the formula becomes simpler, and the velocity can be obtained by solving the quadratic equation:

$$e = \frac{1}{2(\Gamma - 1)} [(\Gamma - 2)E + \sqrt{\Gamma^2 E^2 - 4(\Gamma - 1)M^2}]$$

$$p = (\Gamma - 1)e$$

$$V = \frac{M}{(E+p)}$$

$$u = \frac{M_u}{(E+p)} \quad v = \frac{M_v}{(E+p)} \quad w = \frac{M_w}{(E+p)}$$

The speed of sound can be calculated using the equation

$$C_s = \sqrt{\frac{\Gamma(\Gamma-1)(e-n)}{n+\Gamma(e-n)}}. \quad (9)$$

In this equation if $n \ll e$, which is the case of ultra-relativistic flow when the kinetic energy is much larger than the internal energy, n can be neglected in the equation and the relativistic speed of sound takes the form:

$$C_s^2 = \Gamma - 1 \quad (10)$$

Mathematical Approach

Recently new nontypical approaches in solving non relativistic computational fluid dynamics problems, called "high resolution schemes", were developed. These methods are based on upwind schemes which incorporate into the numerical algorithm the direction in which the physics of the problem propagates. A very simple upwind scheme was discussed by [8] and supplemented by [6]. In this report, the relativistic extension, relativistic HLLE, is used to simulate the collision of two heavy-ions in the ultra-relativistic case. The accuracy of first and second order approximations is discussed. The HLLE method is a so called Godunov-type upwind method in which the discretized distributions of the conserved variables; mass, momentum and energy density, are assumed to be constant within each grid zone. This defines an initial value problem consistent with the Riemann problems. The information contained in the solution of these Riemann problems can be used to determine the fluxes between grid zones.

An explicit numerical scheme for equation (1), which retains the integral conservation properties, is given by

$$U_i^{n+1} = U_i^n - \lambda(G_{i+1/2}^n - G_{i-1/2}^n) \quad (11)$$

with $\lambda = \Delta t / \Delta x$.

The subscript n denote the number of the time step, the indices i define cell centers, and the indices $i \pm 1/2$ correspond to cell faces. Harten [Harten 83] showed how to construct a simple approximation of the Riemann problem. The approximate Riemann solution has the form

$$U(x, t; U_l, U_r) = \begin{cases} U_l & \text{for } x < b_l t, \\ U_r & \text{for } b_l t \leq x \leq b_r t, \\ U_r & \text{for } x > b_r t. \end{cases} \quad (12)$$

This construction assumes that prior bounds for the smallest and largest signal velocities b_l and b_r , as well as the unperturbed states on the left and the right U_l and U_r , are known. The intermediate state, U_r , is determined by requiring consistency of the above approximate equations over a grid zone.

$$U_r = \frac{b_l U_l - b_r U_l - F(U_l) + F(U_l)}{b_r - b_l}. \quad (13)$$

The numerical flux for a first order approximation associated with the previous two equations is

$$G(U_l, U_r) = \frac{b_r^* F(U_l) - b_l^* F(U_r) + b_l^* b_r^* (U_r - U_l)}{b_r^* - b_l^*} \quad (14)$$

where

$$\bar{C}_s = 0.5(C_{sl} + C_{sr})$$

$$\bar{v} = 0.5(v_l + v_r)$$

$$b_l^* = \min(0, \frac{\bar{v} - \bar{C}_s}{1 - \bar{v} \bar{C}_s}, \frac{v_l - C_{sl}}{1 - v_l C_{sl}}) \quad (15)$$

$$b_r^* = \max(0, \frac{\bar{v} + \bar{C}_s}{1 + \bar{v} \bar{C}_s}, \frac{v_r + C_{sr}}{1 + v_r C_{sr}}) \quad (16)$$

for strong shocks, Roe's [13] procedure is used to calculate the average signal velocities as

$$\bar{v} = \frac{\sqrt{R_r} v_r + \sqrt{R_l} v_l}{\sqrt{R_r} + \sqrt{R_l}} \quad (17)$$

$$\begin{aligned} \bar{C}_s^2 &= \frac{\sqrt{R_r} C_{sr}^2 + \sqrt{R_l} C_{sl}^2}{\sqrt{R_r} + \sqrt{R_l}} \\ &+ \frac{\Gamma-1}{2} \frac{\sqrt{R_r R_l}}{(\sqrt{R_r} + \sqrt{R_l})^2} (v_r - v_l)^2. \end{aligned} \quad (18)$$

The extension of the HLLE scheme to second order is done using the MUSCL approach [10] in which second-order accuracy is achieved by introducing more upwind points in the scheme, using the slope between the points i and $i+1$:

$$U_{i\pm}^n = U_i^n \pm \frac{\Delta x}{2} S_i^n \quad (19)$$

where S_i^n is the slope between the points i and $i+1$ and:

$$U_{i\pm}^{n+1/2} = U_{i\pm}^n - 0.5 \frac{\Delta t}{\Delta x} (F_{i\pm}^n - F_{i\pm-1}^n) \quad (20)$$

$$\begin{aligned} G_{i\pm 1/2}^{n+1/2} &= \frac{b_{i\pm 1/2}^+ F_{i\pm}^{n+1/2} - b_{i\pm 1/2}^- F_{i\pm-1}^{n+1/2}}{b_{i\pm 1/2}^+ - b_{i\pm 1/2}^-} \\ &+ \frac{b_{i\pm 1/2}^+ b_{i\pm 1/2}^- (U_{i\pm-1}^{n+1/2} - U_{i\pm}^{n+1/2})}{b_{i\pm 1/2}^+ - b_{i\pm 1/2}^-} \end{aligned} \quad (21)$$

The extension of the second order HLLE to the three dimensions is straight forward. The slope required for equation (19) is calculated in each space dimension (x, y, and z) for each parameter (R , M_x , M_y , M_z and E). Then the first intermediate parameters are calculated as in equation (19). The intrinsic frame parameters (pressure, velocity in each dimension and mass density). Then the second intermediate parameters are calculated as in equation (20). Then the pressure, velocity in each dimension and mass density are calculated at the half time step. $G_{i\pm 1/2, k}^{n+1/2}$, $G_{i-1/2, k}^{n+1/2}$, $G_{y, i\pm 1/2, k}^{n+1/2}$, $G_{y, i-1/2, k}^{n+1/2}$, $G_{z, i\pm 1/2, k}^{n+1/2}$

and $G_{z, i\pm 1/2, k}^{n+1/2}$. The new time step is calculated using the equation:

$$\begin{aligned} U_{jk}^{n+1} &= U_{jk}^{n+1/2} - \lambda_x (G_{x, i+1/2, jk}^{n+1/2} - G_{x, i-1/2, jk}^{n+1/2}) \\ &- \lambda_y (G_{y, i+1/2, k}^{n+1/2} - G_{y, i-1/2, k}^{n+1/2}) \\ &- \lambda_z (G_{z, i\pm 1/2, k}^{n+1/2} - G_{z, i\pm 1/2, k}^{n+1/2}) \end{aligned} \quad (22)$$

Results and Conclusion

The initial condition considered here is similar to the initial condition considered in [5]. We have used the collision of 238 uranium nucleus with 28 silicon nucleus with velocity equal to 0.995c and -0.995c respectively, which correspond to Lorentz gamma factor equal to 10 and a center-of-velocity energy of 200 GeV/A. Rest frame initial conditions to be that of a Woods-Saxon density distribution

$$\varepsilon(r) = \frac{\varepsilon_0}{1 + e^{[(r-r_0)/a_0]}} \quad (23)$$

where $\varepsilon=0.156$ GeV/fm³ is the rest frame nuclear density, r_0 is the nuclear density calculated using the equation

$$r_0 = 1.128 A^{1/3} - 0.89 A^{-1/3} \quad (24)$$

where A is the mass number of the nucleus.

The calculations described here were obtained using second order HLLE method described on an Eulerian grid.

The time evolution of the collision is shown in Figures 2 and 3. The mass density is plotted in these figures to illustrate the development of high density regions. In Figure 1, the results for a head-on collision ($b=0$) is shown. In Figure 2, the results for a non central¹ collision with impact parameter $b=5$ (fm) is plotted. Both figures show cuts in the xy plane of three dimensional grid. The first picture represents the initial conditions. The next seven pictures

¹ - Non central collision : is a collision in which the impact parameter does not equal to zero.

represent the time evolution of the collision with a time interval of 0.5 (fm/c). The density profile is calibrated between (0 and 4) for all pictures (the normal nuclear density in the laboratory frame is about 1.3 fm^{-3}). The nuclei in this simulation exhibit a semi-transparent behavior since one notes a continuation of material along the original direction of each projectile.

The maximum energy density versus time is plotted in Figure 4 for different impact parameters. The highest peak refers to a head on collision with impact parameter ($b=0$). The peak height drops as the impact parameter rises. At impact parameter ($b > r_1+r_2$), it is expected that there will be no peak in the plot, and the maximum value of energy density is the initial conditions.

Figure 5a. shows a comparison among our results using second order relativistic HLLE and the results of two other methods; Spline [3] and SHASTA [13] methods (both groups have compared their results in [5]). In this figure, the HLLE results seem to be shifted to earlier times, but this could be due to a difference in the initial conditions. A small offset in the graph (as can be seen in Figure 5b) shows very good agreement with both methods, particularly with the results of the spline method. The SHASTA method overpredicts the value of the energy density for times greater than three (fm/c).

The time evolution of the momentum on the collision axis and transverse momentum are plotted in Figure 6; the momentum is calculated using the equations:

$$p_x = \sum_{\text{all cells}} |p_x| \quad p_{\text{trans}} = \sum_{\text{all cells}} \sqrt{(p_y^2 + p_z^2)}$$

where p is momentum and the collision axis is the x axis.

The time evolution of the total absolute transverse momentum is plotted in Figure 7 for different impact parameters. It can be seen that large transverse flows occur when the impact parameter is low. However, the probability of the collision of two nuclei with small impact parameter is correspondingly small. This probability rises linearly with impact parameter.

Figure 8 shows the transverse momentum multiplied by the probability of a particular impact parameter plotted versus impact parameter for the near asymptotic solution at time ($T=5 \text{ fm/c}$).

The time evolution of the energy density along the beam axis in the three-dimensional calculations is plotted in Figure 9. The first frame (labeled $T=0$) is the initial conditions. The second frame ($T=0.5$) shows the start of the collision. We can see in this frame that the back part of the large projectile (distance between 7-8 fm on the graph) has not yet suffered any change, due to the fact that the collision is supersonic and the velocity of the collision is faster than the speed of sound in the nuclear matter (speed of sound $C_s \approx 0.2 c$ the speed of light). The frame labeled $T=1$ shows an increase in the energy density to a factor of three due to the collision, and the rest of the frames show the expansion process. We can see from the last three frames ($T=3, 3.5$ and 4) that part of the projectile manages to escape the collision, due to the fact that the nuclei at such high energy (200 GeV A) appear to be semi-transparent.

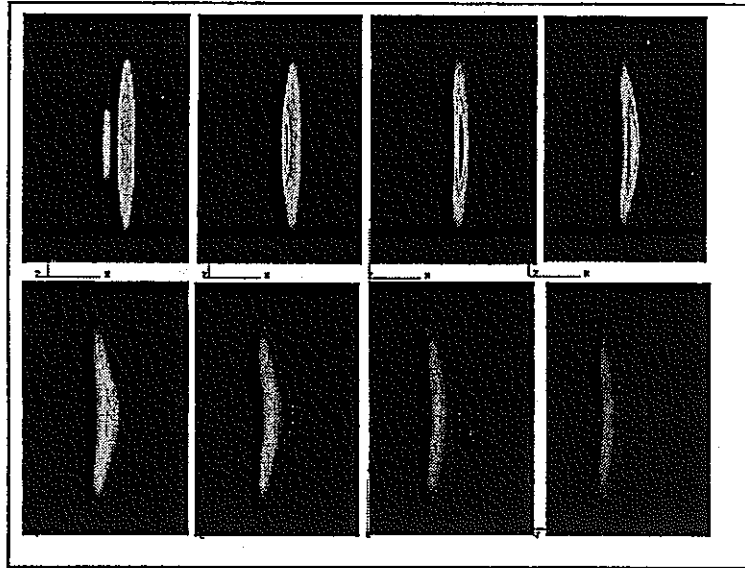


Figure 2. The figure shows the different stages of a head on collision of a $^{238}\text{Uranium}$ nucleus with a $\text{Silicon } 28$. The times t for these stages are 0, 0.5, 1, 1.5, 2, 2.5, 3, 3.5 (fm/c) receptively. The variable plotted is the mass density. The intensity of shading is normalized between black (density =0) and white (density =4). The calculation was done using second order HLLE on an Eulerian grid in three dimension. The plots are cuts on the xy plane, where x is the collision axis.

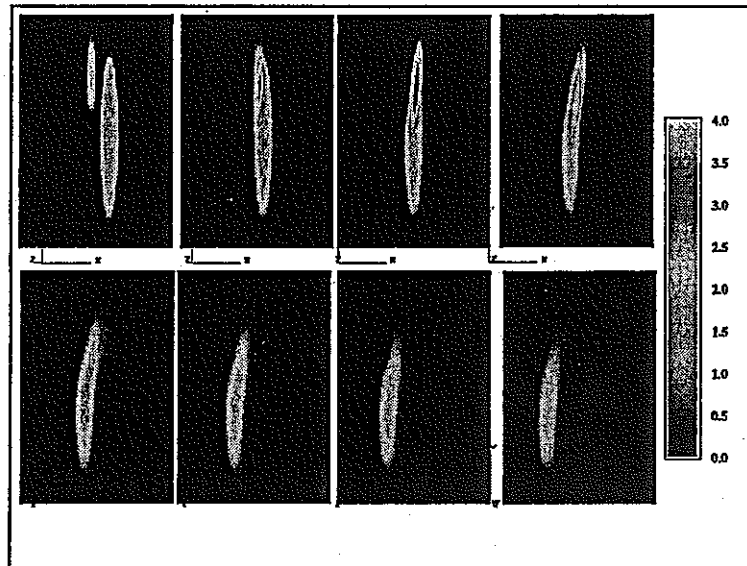


Figure 3. Shows the different stages of a collision of a $^{238}\text{Uranium}$ nucleus with a $\text{Silicon } 28$ nucleus with impact parameter $b=5(\text{fm})$. The times t for these stages are 0, 0.5, 1, 1.5, 2, 2.5, 3, 3.5 (fm/c) receptively. The variable plotted is the mass density. The intensity of shading is normalized between black (density =0) and white (density =4). The calculation on this figure was done using second order HLLE on Eulerian grid in three dimension. The plots are cuts on xy plane, where x is the collision axis. Normal nuclear density considered to be equal to 1.3.

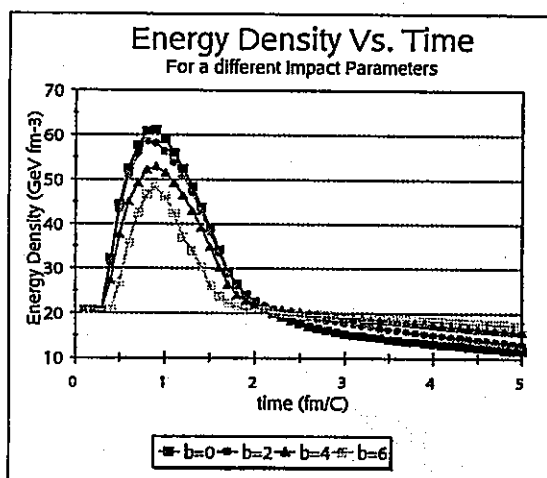


Figure 4. The maximum energy density versus time for the collision of a 238 Uranium nucleus with a 28 silicon nucleus for different impact parameters is shown. In the plot $b=0$ refers to head-on collision.

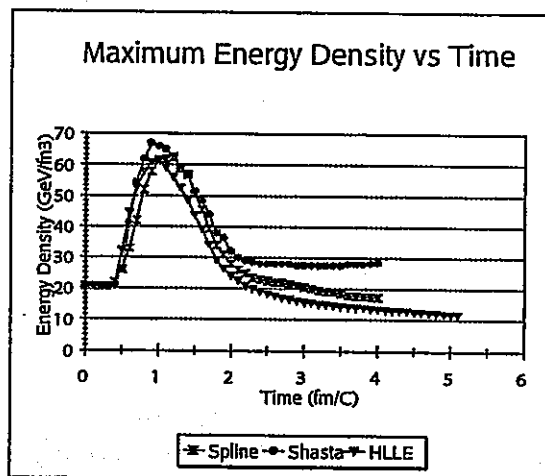


Figure 5b. A plot of maximum energy density versus time for the collision of a 238 Uranium nucleus with a 38 Silicon nucleus for a head-on collision is shown. In this figure, the HLLE data is offset horizontally by two points to aid in comparing the curves.

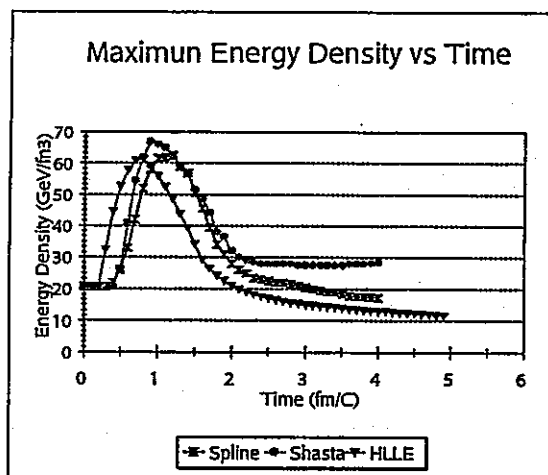


Figure 5a. A plot of maximum energy density versus time for the collision of a 238 Uranium nucleus with a 28 silicon nucleus for head-on collision is shown. The spline and SHASTA calculations [5] are compared to HLLE.

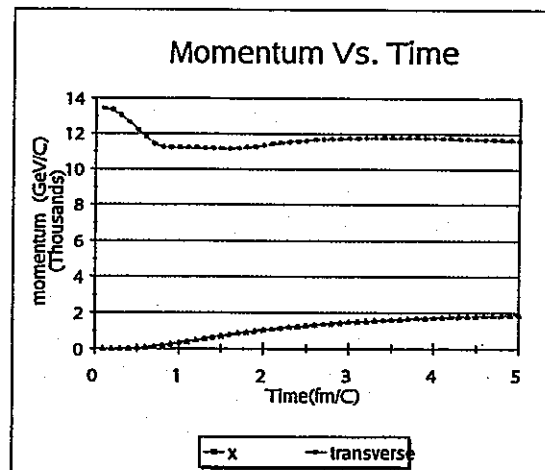


Figure 6. The total of the absolute momentum versus time for the collision of 238 Uranium with 28 silicon for head-on collision is shown.

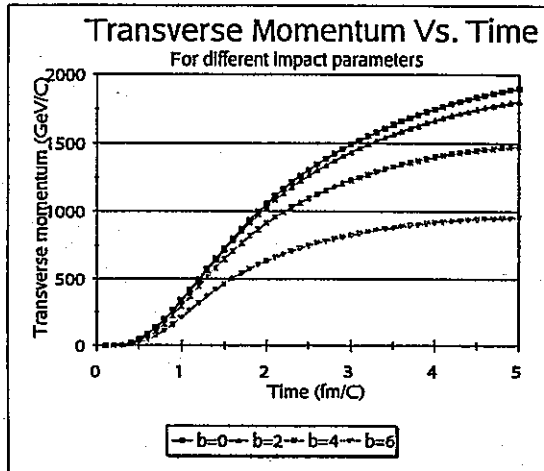


Figure 7. The total of the absolute transverse momentum versus time for the collision of a 238 Uranium nucleus with a 28 silicon nucleus for a different impact parameters is shown, where $b=0$ refers to head on collision.

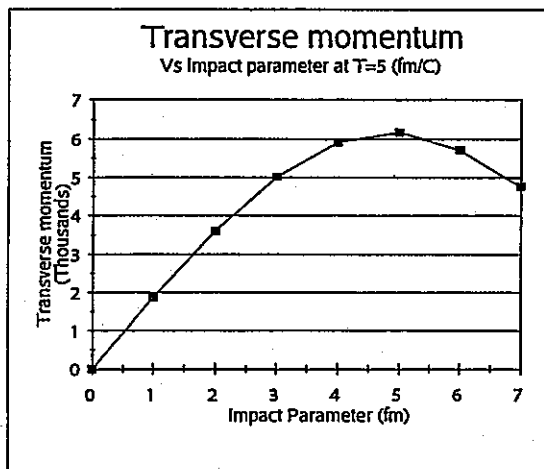


Figure 8. The total of absolute transverse momentum normalized to the geometrical cross section versus impact parameter ($P_{trans} * B$ Vs B) for the collision of a 238 Uranium nucleus with a 28 silicon nucleus is shown.

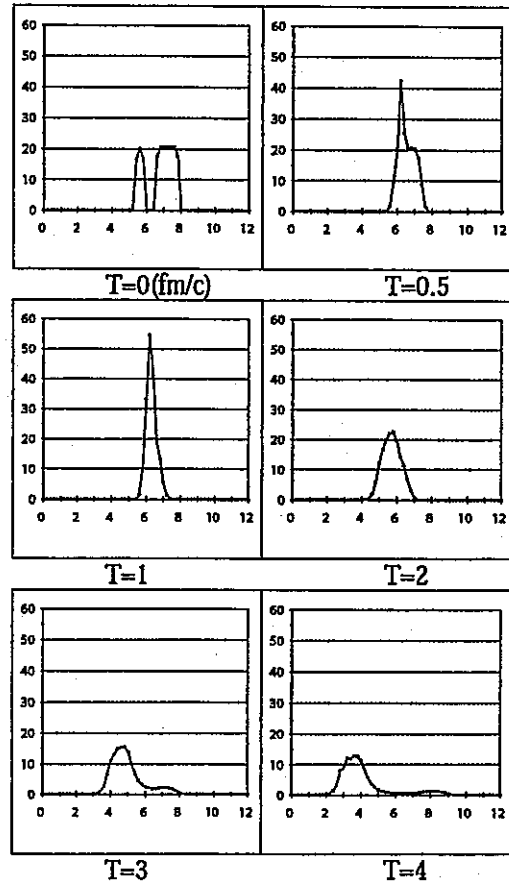


Figure 9. The time evolution of energy density along the beam axis for different time intervals is shown. The horizontal axis represent the distance along the collision axis in fermi (fm). The vertical axis represent the energy density GeV/fm^3 . These graphs were calculated for a three-dimensional head on collision ($b=0$) of a 238 uranium nucleus with 28 silicon nucleus using the second order HLLE.

It is interesting to compare the CPU time of our calculations to that used in the other mentioned calculations [5]. Dean et. al have made a comparison between the SHASTA and the basis spline method. SHASTA calculations was done using a grid of $128 \times 128 \times 64$ points $n_x=n_y=128$, $n_z=64$ (the collision axis) and with $dx=dy=dz=0.2$ (fm). The calculations were done on an IBM 3090-600J VF computer and took 120 of minutes CPU time. The basis spline method used a $32 \times 32 \times 32$ grid with $dx=dy=1$ (fm) and $dz=0.2$ (fm), where z was the collision axis. This calculation was done on an

IPSC/860, 32 node parallel computer and took 3.5 hours of CPU time. Our calculations using the second order HLLE were completed on a $60 \times 20 \times 20$ grid with $dx=0.2$, and $dy=dz=1$ (fm) where x was the collision axis. The calculation was done on a Sun Sparcstation 2 computer and the CPU time was 60 minutes. A meaningful comparison between computations on different computers is very difficult, however, it should be noted that both the IBM 3090-600J VF computer and the IPSC/860, 32 node, parallel computer are much more powerful than the Sparcstation 2 computer. From these preliminary observations we may conclude that our 3-D HLLE method produces simulations which agree well with the results of others but require much less CPU time.

BIBLIOGRAPHY

1. A. Boris and David Book, Flux-Corrected Transport, *J. Compu. Phys.* 11,38(1973).
2. A. Boris, A. M. Landsberg, E. S. Oran and J. H. Gardner, LCPFCT-A Flux Corrected Transport Algorithms for Solving Generalized Continuity Equations, Microfiche NRL/MR/ 6410--93-7192, Naval Research Laboratory (1993).
3. D. J. Dean, C. Bottcher, M. R. Strayer, Spline Techniques for Solving Relativistic Conservation Equations, *Int. J. Modern Phys. C*, 4, 723(1993).
4. D. J. Dean, C. Bottcher, M. R. Strayer, J. C. Wells, Parallel Implementation of 3+1-Dimensional Relativistic Hydrodynamics. *Int. J. Modern Phys. C* 4, 1023(1993).
5. D. J. Dean, C. Bottcher, M. R. Strayer, J. C. Wells, A. von Keitz, Pürsün, D. H. Rischke, and J. A. Maruhn. Comparison of flux-corrected and spline algorithms for solving (3+1)-dimensional relativistic hydrodynamics. *Phys. Rev. E* 49, 1726 (1994).
6. B. Einfeldt, C. D., *SIAM J. Numer. Anal.* 25,294(1988).
7. B. Einfeldt, C. D. Munz, P. L. Roe and B. Sjögreen, On Godunov-Type Methods Near Low Densities, *J. Comp. Phys.* 92, 273(1991).
8. A. Harten, P. D. Lax and B. van Leer, On Upstream Differencing And Godunov-Type Schemes for Hyperbolic Conservation Laws, *SIAM Rev.* 25, 35(1983).
9. A. Harten, *J Comp. Phys.* 49,357(1983).
10. C. Hirsch, *Numerical Computalons of External and Internal Flows*, John wiley & sons (1988).
11. R. B. Piercey, B. K. Soni, H. Tleimat and Anita Trivedi, Grid-based numerical algorithms for nuclear hydrodynamics, Mississippi State University Annual Conference on differential equation and computational simulations, 1993.
12. W. Schmidt, U. Katscher, B. Waldhauser, J. A. Maruhn, H. Stöcker and W. Greiner, Viscosity and the equation of state in high energy heavy-ion reactions. *Phys. Rev. C* 47, 2782(1993).
13. V. Schneider, U. Katscher, D. H. Rischke, B. Waldhauser and J. A. Maruhn, New algorithms for ultra-relativistic numerical hydrodynamics. *J. Comp. Phys* 105, 92 (1993).
14. Schukraft, Ultra-relativistic heavy-ion collisions, *Nuclear Physics News* 2, 14 (1992).
15. A. C. Trivedi, R. B. Piercey, and B. K. Soni, Numerical Simulation of High Energy Heavy Ion Collision, AIAA-95-0154.
16. A. C. Trivedi, Computational Field Simulation of Relativistic Hydrodynamics, Ph.D. Dissertation, Department of physics and astronomy, Mississippi State University, 1994.
17. B. Waldhauser, U. Katscher, W. Schmidt, J. A. Maruhn H. Stöcker and W. Greiner, High energy nuclear fluid dynamics. *Physica Scripta.* 132,195-201(1990).