

NASA/CR—97-206226



# Hierarchical Parallelism in Finite Difference Analysis of Heat Conduction

Joseph Padovan, Lala Krishna, and Douglas Gute  
The University of Akron, Akron, Ohio

Prepared under Grant NAG3-644

National Aeronautics and  
Space Administration

Lewis Research Center

---

November 1997

## Acknowledgments

The first author expresses his appreciation for the many helpful discussions with Chris Chamis of NASA Lewis, which motivated this work. Work supported by NASA Lewis under Grant NAG-3-644.

This report contains preliminary findings, subject to revision as analysis proceeds.

Available from

NASA Center for Aerospace Information  
800 Elkridge Landing Road  
Linthicum Heights, MD 21090-2934  
Price Code: A05

National Technical Information Service  
5287 Port Royal Road  
Springfield, VA 22100  
Price Code: A05

## TABLE OF CONTENTS

Title Page

Abstract

Acknowledgment

Table of Contents

List of Figures and Tables - Part I

List of Figures and Tables - Part II

### Part I - Formulations

#### Summary

1. Introduction
2. Previous Work
  - 2.1 Overview of FD Analysis
  - 2.2 New Computer Environments
3. Current Approach
  - 3.1 Substructural Parallelism
  - 3.2 Local Bandwidth Minimization
4. Algorithms Considerations

References

### Part II - Formal Considerations and Benchmarking

#### Summary

1. Introduction
2. Algorithmic Overview
3. Formal Considerations
4. Hierarchical Preconditioning
5. Benchmarking
6. Concluding Remarks

References

## **List of Tables and Figures - Part I**

Table 3.1 - Computational Effectiveness for Load Balance Equalized System:  
Two Level Hierarchy.

Table 3.2 - Computational Effectiveness for Effort Optimized System:  
Two Level Hierarchy.

Figure 3.1 - Partitioned Physical System

Figure 3.2 - Multilevel Hierarchical Scheme

Figure 3.3 - Two Level Architecture

Figure 3.4 - Two Level Partitioned Model

Figure 3.5 - Second Level External/Global Model

Figure 3.6 - Families of Verticals and Horizontals of Second External Global Level

Figure 3.7 - Three Levels Hierarchy

Figure 3.8 - Three Level Hierarchy of Operators

## **List of Figures and Tables - Part II**

Table 5.1 - Iteration of Requirments of Mixed Direct-Iterative Hierarchical Scheme

Table 5.2 - Requirements of Purely Iterative Hierarchical Scheme

Figure 3.1 - The Number of Iterations Increases Linearly as the Problem Size Becomes Larger.

# Hierarchical Parallelism in Finite Difference Analysis of Heat Conduction

Joseph Padovan,<sup>1</sup> Lala Krishna<sup>1</sup> and Douglas Gute<sup>2</sup>  
The University of Akron, Akron, Ohio

## PART I - FORMULATION

### SUMMARY

Based on the concept of hierarchical parallelism, this series of papers develops highly efficient parallel solution strategies for very large scale heat conduction problems. In addition to yielding a many order of magnitude improvement in computational speed, the methodology reduces round off as well as introduces a significant solution stabilization when used in conjunction with iterative procedures. Overall, the method of hierarchical parallelism involves the partitioning of thermal models into several substructured levels wherein an optimal balance in the various associated bandwidths is achieved. The solution to the problem is then developed in parallel via special direct, iterative or mixed (direct/iterative) procedures wherein each partition is monitored for its intrinsic spectral properties so as to enable the choice of the appropriate local solution algorithms. Overall, the paper is organized into two parts. The first develops the parallel modeling methodology and associated multilevel direct, iterative and mixed solution schemes. Part II establishes both the formal and computational properties of the scheme. Here emphasis is given to establishing convergence characteristics, spectral properties as well as the choice of the appropriate solution accelerators.

---

<sup>1</sup> Department of Mechanical Engineering , The University of Akron.

<sup>2</sup> Department of Mathematics, The University of Akron.

## 1. Introduction

In recent years the almost unlimited promise afforded by the first several generations of sequential type computer architectures has essentially been saturated. Motivated by this, extensive ongoing work has been undertaken to develop new types of computer architectures [1]. These principally fall into the following categories [1,2], i.e.:

- i) Vectorized/pipelined systems;
- ii) Parallel systems; and,
- iii) Combined systems.

Many of the new systems will have a hierarchy of operational modes. Namely, depending on the repetition level of a given block of code, three operational modes will be possible. Specifically, below a given repetition level, the code will be performed sequentially. For an intermediate range, the coding block will be performed vectorially. At yet higher levels, the coding is copied to a set of parallel processors where it is performed locally in a multiply vectorized format. Such a many option scheme is essentially dependent on the intrinsic lower and upper bound performance characteristics of the associated scalar vector and parallel processors. Included in the decision making are the associated communications costs.

While such procedures have added to our current capacities, generally, the efficiencies have become saturated as vector lengths and/or the number of processors have grown too large. Prototypically, in parallel systems employing such schemes as the super cube technology [3], running speeds tend asymptotically to be diminishing fractions of the number of processors employed [4]. Similarly, in vectorized systems generally vector length

is limited to certain lower and upper bound dimensions to yield optimal results. This limitation also reduces the full potential of such schemes.

Beyond these restrictions, the general tendency today is to attempt to solve thermal simulation problems in a strictly global manner. As will be seen, such approaches tend to yield extremely large sets of equations with their concomitant bandwidth problems. In an attempt to bypass such difficulties typically some form of bandwidth minimization scheme is employed [5]. While such an approach yields some relief, even optimized global models represent significant difficulties.

In the context of the foregoing, this series of papers will develop an alternative architectural strategy to handle very large scale thermal simulations. Specifically, a multilevel, i.e., hierarchical form of problem partitioning/substructuring will be developed. As will be seen, such a scheme will enable so-called partitional/substructural/local bandwidth minimization at the various hierarchical levels of the scheme. This will enable orders of magnitude improvement of the computational speed of finite difference and element (FD/FE) simulations when separate processors are defined for each partition. Additionally, when used with direct solvers, the associated round off error is significantly reduced thereby lightening machine load.

In addition to establishing the modelling architecture, several associated solution methodologies are developed. These include:

- i) Purely direct schemes;
- ii) Mixed direct and iterative procedures; and,
- iii) Purely iterative methodologies.

For the mixed scheme, various partitions, i.e. substructure, are handled

via either direct or iterative procedures. Such a treatment being contingent on the associated spectral conditioning of the partition. To generalize the development, a variety of schemes are considered and modified, i.e. Jacobi [6], Gauss-Seidel [6], successive overrelaxation (SOR) [6] and conjugate gradient [7]. As will be seen, the hierarchicalism tends to introduce significant improvements in the stability and efficiency of iterative schemes.

Beyond the purely developmental aspects, formal numerical properties will also be given. This will include such items as convergence characteristics, spectral properties, relative efficiencies of the various schemes, as well as the selection of various optimizing parameters for the SOR and conjugate gradient methodologies.

To prove out the scheme, the results of a variety of benchmark examples will be discussed. Overall these include moderate to large scale heat conduction problems ranging in size from models of 10,000 to 250,000 degrees of freedom.

Overall the two-part series of papers is organized as follows:

i) Part I

I.1 Overview of previous work.

- FD analysis of heat conduction
- New computer environments

I.2 Current Approach

- Substructural parallelism
- Local bandwidth minimization

I.3 Algorithmic Considerations



ii) Part II

II.1 Algorithmic overview

II.2 Formal numerical properties

- Convergence characteristics
- Spectral properties
- Relative convergence properties
- Selection of optimal acceleration parameters
- Influence of hierarchicalism

II.3 Benchmarking including heat conduction problems.

II.4 Final summary.

## 2. Previous Work

As noted in the introduction, FD schemes are widely employed in heat transfer analysis. Most of such applications have involved the use of sequential and at best pipe lined/vectorized computers, i.e. CRAY and FPS type systems [1]. To motivate the development of the hierarchical parallel scheme noted earlier in the Introduction, this section will

- i) Briefly overview FD modelling procedures
- ii) Describe new computer environments and their impact on FD analysis; and,
- iii) Overview shortcomings of proposed computer configurations, i.e. pipe lined/vectorized and parallel type systems.

### 2.1 Overview of FD Analysis

Assuming isotropic media for simplicity, the governing Fourier heat conduction relation takes the form [8]

$$\nabla^2 T + Q = 0 \quad (2.1)$$

wherein [8]

$$\nabla^2 ( ) = \frac{\partial^2}{\partial x_1^2} ( ) + \frac{\partial^2}{\partial x_2^2} ( ) + \frac{\partial^2}{\partial x_3^2} ( ) \quad (2.2)$$

such that T is temperature and  $x_1, x_2, x_3$  are the Cartesian coordinates. The boundary conditions associated with (2.1) are either prescribed temperature, flux or of the convection-radiation type, namely [8]

$$\begin{aligned} \text{i) for } \forall \underline{x} \in \partial R_T \\ T = \bar{T} \end{aligned} \quad (2.3)$$

$$\begin{aligned} \text{ii) for } \forall \underline{x} \in \partial R_q \\ -\kappa \frac{\partial T}{\partial x_i} \eta_i = \bar{q}_i \end{aligned} \quad (2.4)$$

iii) for  $\forall \underline{x} \in \partial R_{cr}$

$$-\kappa \frac{\partial T}{\partial x_i} \eta_i = H(T - T_\infty) + \gamma(T^4 - T_\infty^4) \quad (2.5)$$

where  $\bar{T}$ ,  $\bar{q}$ , are the prescribed temperature and heat flux components, and  $Q$ ,  $\kappa$ ,  $H$ ,  $T_\infty$ ,  $\gamma$ ,  $\partial R_T$ ,  $\partial R_q$ ,  $\partial R_{cr}$  are respectively the internal heat generation thermal conductivity, convection coefficient, ambient temperature, radiation coefficient, and the surface areas defining prescribed temperature, flux and convection-radiation.

For demonstration purposes, (2.1) will be simulated via 5-point 2D and 7-point 3D FD expressions. In this context as noted by Varga [6], (2.1) and its associated boundary conditions, i.e. (2.3 - 2.5) can be converted into the following FD formulation, namely

$$[K]\underline{T} = \underline{Q} \quad (2.6)$$

where  $\underline{T}$  defines the  $(N,1)$  column vector of mesh point temperature,  $\underline{Q}$  the  $(N,1)$  column vector defining internal heat generation and boundary condition effects (purely linear) and lastly  $[K]$  is an  $(N,N)$  five diagonal (2-D case) matrix. Since  $[K]$  is generally a positive definite symmetric matrix it fits in the class of matrices termed Stieltjean [9]. As will be seen in Part II, this property will aid us to establish the convergence properties of the parallel solution strategy.

Note depending on problem size, the bandwidth of  $[K]$  will change accordingly. Furthermore, contingent on the boundary conditions and connectivity of the problem, the bandwidth need not be uniform. Generally (2.6) is solved either iteratively or directly via either a skylined [10] or frontal scheme [11]. For extremely large scale problems, typically

some form of out of core blocking <sup>[12]</sup> is required to handle the direct inverse. Generally even with the use of say solid state disks, this requires extensive amounts of CPU (central processor unit) and real clock times. Hence, large scale problems with even modest average bandwidths tend to tax mainframe capacities. This is one of the primary factors motivating the ongoing thrust to develop alternative computer architectures.

## 2.2 New Computer Environments

Currently there appears to be essentially two major forms of new computer architecture, i.e.:

- i) Pipe lined/vectorized; and,
- ii) Purely parallel

For the pipe lined and vectorized machines, while the overall problem is still solved in an essentially segmental sense, similar noninterdependent operations can be performed in vectorized chunks, i.e. matrix multiplication, subtraction, addition, etc. The vector/pipelined architectures involve the use of numerous individual processors all preprogrammed for certain fixed duties. In this context, the size of the chunks operated on are contingent on the number of available individual processors and associated memories.

In the case of so-called parallel systems, the overall machine architecture consists of several to numerous individual processors each with its own distinct I/O and instruction set capacities. Currently the cross-talk between the various separate processors is achieved via various convoluted connection schemes, i.e. the super cube methodologies used in the FPS 100 series <sup>[4]</sup>, the connection machine supported by DARPA, etc.

Each of the various foregoing technologies, while providing significant improvements over the traditional single processor frames nonetheless themselves possess very important shortcomings. Organized according to architectural type, these include:

i) Pipelined/vector architectures

- Extensive memory requirements in main CPU;
- Very awkward to program;
- Difficult to employ in multi-user environment;
- Difficult to arrange programming to efficiently handle problems with multitudinous substructure, i.e. material groups, boundary conditions, complex boundaries;
- Code structure too dependent on machine architecture to be economically viable;
- Many operations required at local level are scalar, hence slowing down overall throughput;
- I/O awkward due to multitudinous degrees of freedom;
- Particularly awkward to provide for mesh refinement, namely wherein any portion of the model may be densified, all storage and array alignments must be reconfigured during refinement process; and,
- Concentration of all computing power in one machine is not cost effective except for large well funded government research installations.

ii) Parallel machines

- Awkward to program;
- Communication between parallel processors usually awkward, i.e. the super cube;

- Controlling languages highly machine dependent; coding lacks portability;
- I/O typically awkward;
- Compute power of given node usually limited; and,
- Parallel architectural methodologies lacking;

Much of the foregoing problems stem from the fact that to date machine architecture has been dictated by hardware-software considerations as well as the needs of the prevailing bulk user, i.e. the service industry. Because of this, the natural generic features of the physics of an engineering problem and its associated analytical numerical formulation must be subverted to satisfy the needs dictated by machine capabilities. The next section will establish a more natural connection scheme.

### 3. Current Approach

For FD simulations such as defined by (2.6), the overall computer load falls into two main categories:

- (1) The generation of mesh point equations and their associated global assembly, and
- (2) The solution of (2.6) with its concomitant large bandwidth.

In the case of large scale simulations, the use of either direct (sky-line, frontal [11]) or iterative (least square [13], Jacobi, Gauss-Seidel, SOR, preconditioned conjugate gradient PCG) schemes tends to tax even the largest main frames whether pipelined, vectorized, or parallel, i.e. the CRAY-XMP, vectorized IBM-3090-400, FPS-200, etc. In this context, it follows that as currently envisioned and configured, neither vectorized/parallelized nor straight parallel systems provide the total answer to improved computer architectures.

From a purely philosophical point of view, the actions and reactions in real physical situations whether steady state or transient occur in an essentially concurrent format. Such behavior is modelled by the equations of continuum mechanics. Because of this, the ideal computer architecture should be able to simulate such behavior in its own architecture, i.e.:

- i) To be able to handle steady problems wherein all the various system components are interacting; and,
- ii) To handle transient wave front problems wherein the zones of concurrent interaction grow as the waves signalling the information flow spread.

Such modelling requirements point to concurrent parallel processors whose individual capacities are enhanced by pipelined/vectorized attributes.

From a purely industrial point of view, as noted earlier, a single concentrated source of computing would not be economically feasible. A corporations prototypical overall organization is divided into separate departments designing/developing/analyzing various components of a given product. Hence; they represent a multiple computer user base. Generally computer requirements fall into several main categories, namely:

- i) To design/develop/analyze individual product components;
- ii) To provide a common data base for the various interconnecting product components; and,
- iii) On occasion to enable running highly refined component models or overall simulations of the entire product.

Such industrial/institutional organizational schemes place competing somewhat contradictory requirements on the computing facilities, i.e.:

- i) The need for localized distributed computing;
- ii) Data base networking; and,
- iii) Significant central processing capability.

Interestingly, these needs are not unlike those required by a particularly large scale FD simulation, in this case, one involving heat conduction.

### 3.1 Substructural Parallelism

Based on the foregoing, a so-called hierarchically parallel methodology and associated direct and iterative solution strategy will be



developed. Specifically, the overall problem will be partitioned into a set of substructures each acting as separate conduction problems. These partitions constitute the first hierarchical level. Note, each of the substructural problems will be provided with the appropriate interlinking boundary conditions to provide the proper global conservation of energy and field variable continuity.

Noting Fig. 3.1, each separate partition will have its own set of internal and boundary/external mesh points. Hence, (2.6) takes the form

$$[K^{\ell}]T^{\ell} = Q^{\ell} ; \ell \in \{1, L\} \quad (3.1)$$

where L is the number of substructure.

$$T^{\ell} = \begin{Bmatrix} T_{-I}^{\ell} \\ T_{-E}^{\ell} \end{Bmatrix} \quad (3.2)$$

$$Q^{\ell} = \begin{Bmatrix} Q_{-I}^{\ell} \\ Q_{-E}^{\ell} \end{Bmatrix} \quad (3.3)$$

and  $[K^{\ell}]$  is partitioned in the form

$$[K^{\ell}] = \begin{bmatrix} [K_{-I}^{\ell}] & [K_{-IE}^{\ell}] \\ [K_{-EI}^{\ell}] & [K_{-E}^{\ell}] \end{bmatrix} \quad (3.4)$$

such that the superscript  $\ell$  and the subscripts I, E, and IE/EI respectively define the partition number, internal, external and connection blocks of  $[K^{\ell}]$ . The mesh points lying on the boundaries of a given substructure may themselves fall into two categories, i.e.:

- i) Those boundary mesh points shared by two distinct sub-domains;  
and,
- ii) Those shared by several, i.e. see Fig. 3.1.

Based on the segregation of variables into internal and those on dual and multiple boundaries, we see that (3.1) can be recast in the form:

- i) Internal Regions;

$$\begin{aligned}
 [{}_{I}K_{I}^{\ell}]T_{-I}^{\ell} &= \\
 {}_{I}Q^{\ell} + [{}_{I}K_{DB}^{\ell}]T_{-DB}^{\ell} + [{}_{I}K_{MB}^{\ell}]T_{-MB}^{\ell} & \quad (3.5)
 \end{aligned}$$

- ii) Dual Boundaries;

$$\begin{aligned}
 [{}_{DB}K_{DB}^{\ell}]T_{-DB}^{\ell} &= \\
 {}_{DB}Q^{\ell} + [{}_{MB}K_{DB}^{\ell}]T_{-I}^{\ell} + [{}_{DB}K_{MB}^{\ell}]T_{-MB}^{\ell} & \quad (3.6)
 \end{aligned}$$

- iii) Multiple Boundaries;

$$\begin{aligned}
 [{}_{MB}K_{MB}^{\ell}]T_{-MB}^{\ell} &= \\
 {}_{MB}Q^{\ell} + [{}_{MB}K_{DB}^{\ell}]T_{-DB}^{\ell} + [{}_{MB}K_{I}^{\ell}]T_{-I}^{\ell} & \quad (3.7)
 \end{aligned}$$

such that here:

$[{}_{I}K_{I}^{\ell}]$  - Conductivity matrix of  $\ell^{\text{th}}$  substructure

$[{}_{I}K_{DB}^{\ell}]$  - Coupling block between  $\ell^{\text{th}}$  interior variables and  $\ell^{\text{th}}$  dual boundary variables

$[{}_{I}K_{MB}^{\ell}]$  - Coupling block between interior and multiple boundary variables

$[{}_{DB}K_{DB}^{\ell}]$  - Conductivity matrix of dual boundary variables

$[{}_{DB}K_{MB}^{\ell}]$  - Coupling block between dual and multiple boundary variables

$[{}_{MB}K_{MB}^{\ell}]$  - Conductivity matrix of multiple boundary points

and  $T_I^{\ell}$ ,  $T_{DB}^{\ell}$  and  $T_{MB}^{\ell}$  respectively are the interior, dual boundary and multiple boundary temperatures. Due to the structure of  $[K^{\ell}]$  and its various partitions, it follows that

$$[{}_{I}K_{DB}^{\ell}] = [{}_{DB}K_{I}^{\ell}]' \quad (3.8)$$

$$[{}_{I}K_{MB}^{\ell}] = [{}_{MB}K_{I}^{\ell}]' \quad (3.9)$$

$$[{}_{DB}K_{MB}^{\ell}] = [{}_{MB}K_{DB}^{\ell}]' \quad (3.10)$$

where ( )' designates matrix transposition.

Note, depending on the FD operator employed, i.e. 5, 7, 9, 13 point or higher order, the various off diagonal blocks  $[{}_{I}K_{DB}^{\ell}]$ , ... will have different degrees of coupling. For instance, in the case of low level operator (5, 7 point), the structure of  $[{}_{I}K_{DB}^{\ell}]$ , ... will be essentially empty hence yielding a so called sparse coupling. In contrast, for higher order FD operators the structure of  $[{}_{I}K_{DB}^{\ell}]$ , ... will be quite complex hence yielding a dense coupling. Unlike the FD, use of FE analysis will prototypically yield a dense coupling between levels.

After direct assembly, (3.5) - (3.7) yields the following global second level relations, namely:

$$\begin{aligned} \Sigma(\text{Eq. (3.5)})_{\ell} \rightarrow \\ [{}_{I}K_{I}^{\ell}]T_I - [{}_{I}K_{DB}^{\ell}]T_{DB} - [{}_{I}K_{MB}^{\ell}]T_{MB} = Q_I \end{aligned} \quad (3.11)$$

$$\sum_{\ell} (\text{Eq. (3.6)})_{\ell} \rightarrow$$

$$[{}_{DB}K_{DB}]_{-DB}^T - [{}_{DB}K_I]_{-I}^T - [{}_{DB}K_{MB}]_{-MB}^T = Q_{-DB} \quad (3.12)$$

$$\sum_{\ell} (\text{Eq. (3.7)})_{\ell} \rightarrow$$

$$[{}_{MB}K_{MB}]_{-MB}^T - [{}_{MB}K_{DB}]_{-DB}^T - [{}_{MB}K_I]_{-I}^T = Q_{-MB} \quad (3.13)$$

where for example  $[{}_I K_I]$  is a block diagonal matrix, that is

$$[{}_I K_I] = \begin{bmatrix} [{}_I K_I^1] & & & & \\ & \ddots & & & \\ & & [{}_I K_I^{\ell}] & & \\ & & & \ddots & \\ [0] & & & & [{}_I K_I^N] \end{bmatrix} \quad (3.14)$$

Furthermore,  $T_{-I}$  takes the following partitioned form

$$T_{-I} = \begin{Bmatrix} T_{-I}^1 \\ T_{-I}^2 \\ \vdots \\ T_{-I}^{\ell} \\ \vdots \\ T_{-I}^N \end{Bmatrix} \quad (3.15)$$

The size of the various partitions is given by

$$\begin{aligned}
[K_I] &= (N_I, N_I) \\
[K_{DB}^K] &= (N_I, N_{DB}) \\
[K_{MB}^K] &= (N_I, N_{MB}) \\
[K_{DB}^{K_{DB}}] &= (N_{DB}, N_{DB}) \\
[K_{DB}^{K_{MB}}] &= (N_{DB}, N_{MB}) \\
[K_{MB}^{K_{MB}}] &= (N_{MB}, N_{MB})
\end{aligned} \tag{3.16}$$

where  $N_I$ ,  $N_{DB}$  and  $N_{MB}$  respectively denote the number of total internal, dual and multiple boundary mesh points. Note if  $N$  is the total number of degrees of freedom, then

$$N = N_I + N_{DB} + N_M \tag{3.17}$$

where

$$N_I = \sum_{\ell}^L N_I^{\ell} \tag{3.18}$$

such that  $N_I^{\ell}$  is the number of internal mesh points of the  $\ell^{\text{th}}$  substructure.

Recast in global matrix form, (3.11) - (3.13) yields the expression

$$[K_D] \underline{T} = \underline{Q} + ([K_U] + [K_L]) \underline{T} \tag{3.19}$$

where

$$[K_D] = \begin{bmatrix} [K_I] & [0] & [0] \\ [0] & [K_{DB}^{K_{DB}}] & [0] \\ [0] & [0] & [K_{MB}^{K_{MB}}] \end{bmatrix} \tag{3.20}$$

$$\begin{aligned}
& [K_U] + [K_L] = \\
& \begin{bmatrix} [0] & [{}_I K_{DB}] & [{}_I K_{MB}] \\ [{}_{DB} K_I] & [0] & [{}_{DB} K_{MB}] \\ [{}_{MB} K_I] & [{}_{MB} K_{DB}] & [0] \end{bmatrix} \quad (3.21)
\end{aligned}$$

such that

$$\begin{aligned}
& [K_U] = \\
& \begin{bmatrix} [0] & [{}_I K_{DB}] & [{}_I K_{MB}] \\ [0] & [0] & [{}_{DB} K_{MB}] \\ [0] & [0] & [0] \end{bmatrix} \quad (3.22)
\end{aligned}$$

$$[K_L] = [K_U]^T \quad (3.23)$$

As can be seen from (3.20),  $[K_D]$  is a block diagonal matrix. The matrices  $[K_U]$  and  $[K_L]$  are respectively of upper and lower triangular form with zero blocked diagonals.

Due to its form, (3.19) provides a two level, i.e. hierarchically parallel organization to the governing equations. It can be employed in several ways, namely:

1. If all the internal variables are eliminated, then the external nodes appearing on the various substructure tend to convert them into super elements, i.e. as per finite element analysis; note such a procedure would be undertaken prior to assembly and performed simultaneously, i.e. in parallel;

since it only retains external variables, the resulting assembled global formulation would be significantly reduced in size and hence more manageable;

2. Equation (3.19) could be used to establish three basic types of solution algorithms, i.e.

- Direct elimination both locally and globally;
  - Enable a mixed procedure wherein the various partitions are handled either iteratively or directly; and,
  - Both levels of the formulation are handled iteratively;
- lastly

3. The hierarchical parallelism can be carried out in general global-local levels which enable bandwidth minimization on a local rather than a full problem basis, Fig. 3.2.

Note, due to the restructuring of (2.6), (3.19) is in P-cyclic form [6] for five and seven point FD simulations. Specifically, noting (3.20) and (3.21), it follows from Varga [6] that (3.19) is weakly 2-cyclic. As will be seen in Part II, this property will enable us to establish a very well defined range of choices for parameters to accelerate the successive overrelaxation (SOR) scheme [6].

In a similar context, for FD simulations the various matrices appearing in (3.19) are Stieltjean [9]. Hence, the powerful results of M-matrix theory [6] will enable us to establish formal considerations which define the convergence properties of a wide variety of iterative schemes, i.e. Jacobi, Gauss Siedel, SOR, as well as preconditioned conjugate gradient schemes. These considerations will also enable comparisons between variations of the foregoing techniques.

### 3.2 Local Bandwidth Minimization

For the foregoing two level partitioning process, the appropriate choice of the number of substructure can yield optimal results. As an example, noting Fig. 3.3, consider the case wherein separate processors and dedicated to the individual local substructure as well as to the assembled global level external formulation. While the optimal balance between the requisite number of processors and simulation size is problem dependent, as will be seen from the following example, the central controlling factor is the balance between first and second level bandwidth minimization. In particular, consider the 2-D rectangular uniformly differenced domain defined in Fig. 3.4. For simplicity, the region will be partitioned into  $\kappa_1 \kappa_2$  first level, i.e. local domains. In this context, if  $n_{g1}$  and  $n_{g2}$  denote the total number of degrees of freedom per edge then it follows from Fig. 3.4 that

$$n_{g1} = \kappa_1(n_1 - 1) + 1 \quad (3.24)$$

$$n_{g2} = \kappa_2(n_2 - 1) + 1 \quad (3.25)$$

where  $n_1$  and  $n_2$  denote the number of degrees of freedom along the edges of the local substructure.

Based on (3.24) and (3.25), the total number of degrees of freedom is given by the expression

$$N_g = n_{g1} n_{g2} \quad (3.26)$$

or in terms of  $n_1$  and  $n_2$  we have that

$$N_g = (\kappa_1(n_1 - 1) + 1)(\kappa_2(n_2 - 1) + 1) \quad (3.27a)$$

In the square case ( $\kappa_1 = \kappa_2, n_1 = n_2$ ) then

$$N_g = (\kappa(n - 1) + 1)^2 \quad (3.27b)$$



For situations wherein  $(n, \kappa) \gg 1$  such that  $n \gg \kappa$ , (3.27) reduces to

$$Ng \sim \kappa_1 \kappa_2 n_1 n_2 \quad (\text{rectangular}) \quad (3.28)$$

$$Ng \sim (\kappa)^2 (n)^2 \quad (\text{square}) \quad (3.29)$$

After global assembly the minimum bandwidth associated with the non-substructured full formulation is given by the expression

$$Bg = \kappa_1 (n_1 - 1) + 2 \quad (3.30)$$

where here  $\kappa_1 (n_1 - 1) + 1 < \kappa_2 (n_2 - 1) + 1$ . For the square region, (3.30) reduces to

$$Bg = \kappa (n - 1) + 2 \quad (3.31)$$

In the case that  $(\kappa, n) \gg 1$ ;  $\kappa \ll n$  then

$$Bg \sim \kappa n \quad (3.32)$$

Based on (3.26) - (3.32), the computational effort associated with the direct calculation of (2.6) takes the form

$$Cg \sim \frac{1}{2} Ng ((Bg)^2 + Bg) \quad (3.33)$$

Considering the square and rectangular regions, we yield that

$$Cg \sim \frac{1}{2} (\kappa_1 (n_1 - 1) + 1) (\kappa_2 (n_2 - 1) + 1) \{ (\kappa_1 (n_1 - 1) + 1)^2 + \kappa_1 (n_1 - 1) + 1 \} \quad (3.34)$$

$$Cg \sim \frac{1}{2} (\kappa (n - 1) + 1)^2 \{ (\kappa (n - 1) + 1)^2 + \kappa (n - 1) + 1 \} \quad (3.35)$$

Again for the case where  $(\kappa, n) \gg 1$  such that  $\kappa \ll n$ , then

$$Cg \sim \frac{1}{2} (\kappa_1 n_1)^3 \kappa_2 n_2 \quad (3.36)$$

$$Cg \sim \frac{1}{2} (\kappa n)^4 \quad (3.37)$$

For the local substructural setup, the bandwidth is given by

$$B_l = n_1 + 1 \quad (3.38)$$

$$B_2 = n+1 \quad (3.39)$$

The total number of degrees of freedom are

$$N_2 = n_1 n_2 \quad (3.40)$$

$$N_2 = (n)^2 \quad (3.41)$$

Note for the  $l^{\text{th}}$  substructure, the nearly optimal bandwidth is

$$\beta_2 \sim n \quad (3.42)$$

If a Gaussian scheme [10] is employed to solve the internally/externally partitioned version, then the connectivity block, i.e.,  $[\kappa_{IE}^2]$  tends to yield an increased bandwidth. In particular, from an asymptotic point of view

$$\beta_2 \sim 3n \quad (3.43)$$

Hence, the asymptotic computational effort associated with the local block is given by

$$C_2 \sim \frac{9}{2} n^4 \quad (3.44)$$

The optimal version yields

$$C_{2\text{optimal}} \sim \frac{1}{2} n^4 \quad (3.45)$$

Thus asymptotically

$$C_2/C_{2\text{optimal}} \sim 9 \quad (3.46)$$

This state of affairs can be improved with the use of nested dissection [14] to condense the local partition.

Considering the second level of the partitioned scheme, the estimation of calculation load is complicated by the sparse nature of the region under consideration. Noting Fig. 3.5, the total number of external degrees of freedom is given by the expressions

$$N_e = (\kappa_1(n_1-1)+1)(\kappa_2+1) + (\kappa_2(n_2-1)+1)(\kappa_1+1) - (\kappa_1+1)(\kappa_2+1), \quad (3.47)$$

$$N_e = 2(\kappa(n-1)+1)(\kappa+1) - (\kappa+1)^2 \quad (3.48)$$

In the case that  $(n_1, n_2, \kappa_1, \kappa_2) \gg 1$ ;  $n_1 > \kappa_1$ ,  $n_2 > \kappa_2$  then

$$N_e \sim \kappa_1 \kappa_2 (n_1 + n_2) \quad (3.49)$$

$$N_e \sim 2(\kappa)^2 n \quad (3.50)$$

To establish the bandwidth of the assembled external level formulation, several factors must be taken into account. Specifically, noting Fig. 3.6, the second level formulation consists of families of horizontal and vertical mesh points. In this context, bandwidth is contingent on:

- i) Vertical positioning of horizontals;
- ii) Location within either verticals or horizontals

Based on this, the bandwidths associated with the various external degrees of freedom are given by the following external family of expressions, namely:

- i) First vertical,

$$B_{evij}^1 = i \quad (3.51)$$

$$j \in [1, \kappa_1], i \in [1, n_1 - 1]$$

- ii) 2nd, 3rd, ...,  $\kappa_2 + 1$  verticals,

$$B_{evij}^k = \kappa_1(n_1 - 1) + (\kappa_1 + 1)(n_2 - 2) + i \quad (3.52)$$

$$i \in [2, n_1], k \in [2, \kappa_2 + 1], j \in [1, \kappa_1]$$

$$B_{evol}^k = \kappa_1(n_1 - 1) + (\kappa_1 + 1)(n_2 - 2) + 1 \quad (3.53)$$

$$k \in [2, \kappa_2 + 1]$$

- iii) First horizontal,

$$B_{ehij}^1 = \kappa_1(n_1 - 1) + 1 + (\kappa_1 + 1)(i + 1) \quad (3.54)$$

$$i \in [1, n_2 - 2], j \in [1, \kappa_2]$$

iv) 2nd, 3rd, ...,  $\kappa_1+1$  horizontals,

$$B_{ehij}^k = \kappa_1(n_1-1) + j + (\kappa_1+1)(i-1) - (j-2(n_1-1)) \quad (3.55)$$

$i \in [1, n_2-2], k \in [1, \kappa_2]$   
 $j \in [2, \kappa_1+1]$

such that the subscripts e, h and v define external, horizontal and vertical.

Averaging the above noted bandwidths we yield the expressions

i) First vertical;  $\kappa_1(n_1-1)$  entries,

$$\langle B_{evij}^1 \rangle = \frac{n_1}{2} \quad (3.56)$$

ii) 2nd, 3rd, ... verticals,

$$\langle B_{evij}^k \rangle = \kappa_1(n_1-1) + (\kappa_1+1)(n_2-2) + 1 + \frac{n_1}{2}$$

$\kappa_1 \kappa_2 (n_1-1)$  entries (3.57)

$$\langle B_{evol}^k \rangle = \kappa_1(n_1-1) + (\kappa_1+1)(n_2-2) + 1$$

$\kappa_2$  entries (3.58)

iii) First horizontals;  $(n_2-2)\kappa_2$  entries;

$$\langle B_{ehij}^k \rangle = \kappa_1(n_1-1) + 1 + (\kappa_1+1) \frac{n_2-1}{2} - (\kappa_1+1) \quad (3.59)$$

iv) 2nd, 3rd, ... horizontals,  $(n_2-2)\kappa_1\kappa_2$  entries

$$\langle B_{ehij}^k \rangle = \kappa_1(n_1-1) + 1 + \frac{\kappa_1+1}{2} + (\kappa_1+1) \frac{n_2-1}{2} - (\kappa_1+1)$$

$+ 2(n_1-1) - (n_1-1)(1 + \frac{\kappa_1+1}{2})$  (3.60)

For the square region, we yield that

$$\begin{aligned}
\langle B_{evij}^1 \rangle &= n/2 \\
\langle B_{evij}^k \rangle &= \kappa(n-1) + (\kappa+1)(n-2) + 1 + n/2 \\
\langle B_{evol}^k \rangle &= \kappa(n-1) + (\kappa+1)(n-2) + 1 \tag{3.61} \\
\langle B_{ehij}^1 \rangle &= \kappa(n-1) + 1 + (\kappa+1) \frac{n-1}{2} - (\kappa+1) \\
\langle B_{ehij}^k \rangle &= \kappa(n-1) + 1 + \frac{\kappa+1}{2} + (\kappa+1) \frac{n-1}{2} - (\kappa+1) + \\
&\quad 2(n-1) - (n-1)(1 + \frac{\kappa+1}{2})
\end{aligned}$$

In the case that  $(n_1, n_2, \kappa_1, \kappa_2) \gg 1$ ;  $(n_1, n_2) \gg (\kappa_1, \kappa_2)$  then

$$\begin{aligned}
\langle B_{evij}^1 \rangle &\sim n_1/2 \\
\langle B_{evij}^k \rangle &\sim \kappa_1 n_1 + \kappa_1 n_2 \tag{3.62} \\
\langle B_{evol}^k \rangle &\sim \kappa_1 n_1 + \kappa_1 n_2 \\
\langle B_{ehij}^k \rangle &\sim \kappa_1 n_1 + \kappa_1 n_2/2 \\
\langle B_{ehij}^k \rangle &\sim \kappa_1 n_1 + \kappa_1 n_2/2 - n_1 \kappa_1/2
\end{aligned}$$

and

$$\begin{aligned}
\langle B_{evij}^1 \rangle &\sim n/2 \\
\langle B_{evij}^k \rangle &\sim 2\kappa n \\
\langle B_{evol}^k \rangle &\sim 2\kappa n \tag{3.63} \\
\langle B_{ehij}^1 \rangle &\sim \frac{3}{2} \kappa n \\
\langle B_{ehij}^k \rangle &\sim \kappa n
\end{aligned}$$

Based on the foregoing averages of the individual vertical and horizontal external level bandwidths, the overall average takes the form

$$\langle B_e \rangle \sim \frac{1}{N_e} \{ \kappa_1 (n_1 - 1) \langle B_{evij}^1 \rangle + \kappa_1 \kappa_2 (n_1 - 1) \langle B_{evij}^k \rangle + \kappa_2 \langle B_{evol}^k \rangle + (n_2 - 2) \kappa_2 \langle B_{ehij}^1 \rangle + (n_2 - 2) \kappa_2 \kappa_1 \langle B_{ehij}^k \rangle \} \quad (3.64)$$

For the square region, (3.64) reduces to the form

$$\langle B_e \rangle \sim \frac{1}{N_e} \{ \kappa n \langle B_{evij}^1 \rangle + \kappa^2 (n-1) \langle B_{evij}^k \rangle + \kappa \langle B_{evol}^k \rangle + \kappa (n-2) \langle B_{ehij}^1 \rangle + \kappa^2 (n-2) \langle B_{ehij}^k \rangle \} \quad (3.65)$$

As before, for the asymptotic case wherein  $(\kappa_1, \kappa_2, n_1, n_2) \gg 1$ ;  $(n_1, n_2) > (\kappa_1, \kappa_2)$ , we yield the simplified expression

$$\langle B_e \rangle \sim \frac{1}{N_e} \{ \kappa n \frac{n}{2} + \kappa^2 n (2\kappa n) + \kappa (2\kappa n) + \kappa n (\frac{3}{2} \kappa n) + \kappa^2 n (\kappa n) \} \quad (3.66)$$

Since

$$N_e \sim 2\kappa^2 n, \quad (3.67)$$

it follows that

$$\langle B_e \rangle \sim \frac{3}{2} \kappa n \quad (3.68)$$

Employing the foregoing relations, we are now in the position to establish the calculation effort at the 2nd global level. In particular

$$C_e = \frac{1}{2} N_e ((\langle B_e \rangle)^2 + \langle B_e \rangle) \quad (3.69)$$

To simplify the forthcoming discussion of optimization we will confine our development to the asymptotic forms. These can easily be upgraded

for all possible choices of  $n_1, n_2, \kappa_1, \kappa_2$ , i.e. after some extensive algebraic manipulations. In this context,  $C_e$  takes the form

$$C_e \sim \frac{9}{4} (\kappa)^4 (n)^3 \quad (3.70)$$

To choose the optimal values of  $\kappa$  and  $n$ , we must establish the net effort of the hierarchically parallel scheme. In this context, noting the local and external efforts defined by (3.46) and (3.70), we yield that  $C_t$  the total hierarchical effort is defined by the relation.

$$\begin{aligned} C_t &= C_l + C_e \\ &= \frac{9}{2} (n)^4 + \frac{9}{4} (\kappa)^4 (n)^3 \end{aligned} \quad (3.71)$$

At this juncture it is worthwhile determining the ratios between local, external and total hierarchical efforts to the straight full nonsubstructured approach. Recalling (3.37), this yields the following expressions

$$R_{l/g} = C_l/C_g \sim 9/(\kappa)^4 \quad (3.72)$$

$$R_{e/g} = C_e/C_g \sim 4.5/n \quad (3.73)$$

and

$$R_{t/g} = C_t/C_g \sim 4.5/n + 9/(\kappa)^4 \quad (3.74)$$

where  $R_{l/g}$ ,  $R_{e/g}$  and  $R_{t/g}$  define the respective ratios between the first, second and total dual level efforts and that of the full nonsubstructured simulation.

Employing (3.72) - (3.74), we need to establish the optimal choices of  $n$  and  $\kappa$  for a given  $B_g$ . Two approaches can be taken to achieve this, namely:

- i) Equalize problem sizes at both local and external levels; or,
- ii) Minimize  $R_{t/g}$  with respect to  $\kappa$ .

For case (i) namely computer load equalization, we require that

$$C_e = C_l \quad (3.75)$$

This leads to the expression

$$\kappa = \sqrt[5]{2 B_g} \quad (3.76)$$

For case (ii), we require that

$$\frac{d}{d\kappa} (R_{t/g}) \equiv 0 \quad (3.77)$$

Recasting (3.74) in terms of  $\kappa$  and  $B_g$ , we yield the relation

$$R_{t/g} \sim 4.5 \frac{\kappa}{B_g} + \frac{9}{\kappa^4} \quad (3.78)$$

In terms of (3.77), (3.78) yields the expression

$$\kappa = \sqrt[5]{8 B_g} \quad (3.79)$$

Note while (3.76) balances the computing load among processors, (3.79) optimizes, i.e. minimizes the overall computational effort. For instance, noting Table 3.1, as would be expected, as problem size increases, the number of processors needed to balance local and external computing load increases. Concomitant with the increase in processors and problem size, we also see that the relative advantages of hierarchical parallelism also grow significantly. This is a direct outgrowth of the localized bandwidth minimization afforded by the procedure.



For the case of overall load minimization, i.e. (3.17), Table 3.2 illustrates the improvements afforded by the current form of parallelism. As would be expected, (3.17) yields somewhat improved results over the load sharing scheme. This, of course, is at a cost of requiring a modest increase in the number of processors. A comparison of the two schemes illustrates that the increased number of processors is somewhat offset by the decrease in relative loading. In particular, comparing the  $R_{e/g}$  ratios noted in Tables 3.1 and 3.2, it follows that the local processor load is reduced by 300 percent for the optimized scheme. In this context somewhat less powerful individual processors could be employed for the work load optimized approach.

To yield more optimal results, a multilevel hierarchicalism may be employed. Considering a three level system, Figs. 3.7 and 3.8, two levels of partitioning are required. After extensive manipulations, it follows that the asymptotic ratio between the straight classic and the hierarchical methodology takes the form

$$R_{t/g} = C_{2/g} + C_{e/g}^2 + C_{e/g}^3 \quad (3.80)$$

wherein

$$C_{2/g} \sim \frac{9}{(\kappa_2 \kappa_3)^4} \quad (3.81)$$

$$C_{e/g}^2 \sim \left(\frac{49}{4}\right) \frac{\kappa_2}{B_g (\kappa_3)^3} \quad (3.82)$$

$$C_{e/g}^3 \sim \left(\frac{9}{2}\right) \frac{\kappa_3}{B_g} \quad (3.83)$$

such that  $(\kappa_2)^2$ ,  $(\kappa_3)^2$  denote the number of substructure at the 2nd and 3rd levels. Furthermore  $C_{e/g}^2$  and  $C_{e/g}^3$  represent the calculation load defined by the 2nd and 3rd levels.

To establish an optimal choice of substructuring, the appropriate values of  $\kappa_2$  and  $\kappa_3$  must be established. This is achieved by taking the requisite partial derivatives, namely

$$\frac{\partial}{\partial \kappa_2} (R_{t/g}) = 0 \quad (3.84)$$

$$\frac{\partial}{\partial \kappa_3} (R_{t/g}) = 0 \quad (3.85)$$

Employing (3.80) - (3.85) we yield the expressions

$$0 \sim - \frac{36}{(\kappa_3)^4 (\kappa_2)^5} + \frac{49}{4} \frac{1}{Bg (\kappa_3)^3} \quad (3.86)$$

$$0 \sim - \frac{36}{(\kappa_2)^4 (\kappa_3)^5} - \frac{147}{4} \frac{\kappa_2}{Bg (\kappa_3)} + \frac{9}{2Bg} \quad (3.87)$$

Solving (3.86) for  $\kappa_2$  yields the relation

$$(\kappa_2)^5 \sim \frac{144}{49} \frac{Bg}{\kappa_3} \quad (3.88)$$

Since  $\kappa_2$  is non-negative and real, (3.86) can be reduced to the form

$$\kappa_2 \sim \left\{ \frac{144}{49} \frac{Bg}{\kappa_3} \right\}^{\frac{1}{5}} \quad (3.89)$$

In terms of (3.89), we yield the following expression for  $\kappa_3$ , namely

$$\frac{9}{2Bg} \sim \frac{36}{(\kappa_3)^5} \left( \frac{49}{144} \frac{\kappa_3}{Bg} \right)^{\frac{1}{5}} + \frac{147}{4Bg\kappa_3} \left( \frac{144}{49} \frac{Bg}{\kappa_3} \right)^{\frac{1}{5}} \quad (3.90)$$

As can be seen from the preceding development, the three level architecture yields a several fold improvement, namely:

- i) Significantly increased overall speed;
- ii) Loading per individual local processors can be reduced several orders of magnitude;
- iii) Due to reduced loading of local level, the associated processors can be reduced in number and reused in a series format.

In the context of iii), fewer processors are required by the local level. For the optimal case, since  $C_{l/g} \ll C_{e/g}^2$ , rather than having separate processors for each substructure, some may be reused. For such situations, (3.80) takes the following modified form, namely

$$R_{t/g} = f_1 \kappa_2 C_{l/g} + f_2 \kappa_3 C_{e/g}^2 + C_{e/g}^3 \quad (3.91)$$

where here  $f_1$  and  $f_2$  denote factors defining the inverse ratio of the number of processors. In particular

$$\begin{aligned} f_1 &\in [1, 1/\kappa_2] \\ f_2 &\in [1, 1/\kappa_3] \end{aligned} \quad (3.92)$$

In terms of the foregoing, the hierarchical strategy can employ a degree of serialism without any real sacrifice in overall speed.

Note the 3-D cube analogy shows even more significant improvements when the hierarchical substructuring strategy is employed. In this context, it follows that employing the concept of local bandwidth minimization in conjunction with the appropriate mix of local and external variables, significant improvements can be achieved. These can be further enhanced as the number of levels is increased.

#### 4. Algorithmic Considerations

As noted in the previous section, the solution to say the two level architecture can be performed in several ways, namely:

- i) Direct at both levels;
- ii) Direct locally and iterative globally or vice versa; and,
- iii) Mixed direct and iterative at both levels.

Since the purely direct method has essentially been outlined earlier, this section will consider cases ii) and iii). In this context, we shall develop mixed direct/iterative and purely iterative solutions to the two level formulation defined by (3.19).

Note the local level associated with (3.19) involves taking the inverse of the block diagonal matrix  $[K_D]$ . In terms of (3.20) and (3.14), we see that the inverse of  $[K_D]$  can be achieved as a number of independent partitions. Depending on the associated matrix conditioning, i.e. spectral radius, either direct or iterative procedures could be employed. In this context

$$[K_D]^{-1} = \begin{bmatrix} [K_I]^{-1} & [0] & [0] \\ [0] & [K_{DB}^{K_{DB}}]^{-1} & [0] \\ [0] & [0] & [K_{MB}^{K_{MB}}]^{-1} \end{bmatrix} \quad (4.1)$$

where for instance

$$[K_I]^{-1} = \begin{bmatrix} [K_I^1]^{-1} & [0] \\ & \dots \\ [0] & [K_I^2] \\ & \dots \end{bmatrix} \quad (4.2)$$

Here the various partitions making up  $[I^K_I]$ ,  $[DB^K_{DB}]$  and  $[MB^K_{MB}]$  could be handled by entirely separate schemes, i.e.:

- i) Direct; or
- ii) Iterative, for example:
  - Jacobi
  - Gauss-Seidel
  - SOR
  - Steepest descent
  - Conjugate gradient

At the global level, (3.19) can be solved via either a direct or iterative methodology. For the simplest formulation the Jacobi type method, (3.19) yields the algorithm

$$[K_D]T_{-n+1} = Q + ([K_U] + [K_L])T_{-n} \quad (4.2)$$

or after local inversion

$$T_{-n+1} = [K_D]^{-1} \{Q + ([K_U] + [K_L])T_{-n}\} \quad (4.3)$$

For the Gauss-Seidel methodology, (3.19) yields the expression

$$T_{-n+1} = ([K_D] - [K_L])^{-1} \{Q + [K_U] T_{-n}\} \quad (4.4)$$

As will be seen in Part II of this series, since

$$[K] = [K_D] - ([K_L] + [K_U]) \quad (4.5)$$

represents a regular splitting of  $[K]$ , the Stein-Rosenberg theorem [15] can be employed to show that the Gauss-Seidel version is superior to the Jacobi.

In the case of SOR type methodologies, (3.19) yields the following algorithm, namely

$$\begin{aligned}
 [K_D] \underline{T}_{n+1} - \omega [K_L] \underline{T}_{n+1} = \\
 (1-\omega)[K_D] \underline{T}_n + \omega [K_U] \underline{T}_n + \omega \underline{Q}
 \end{aligned}
 \tag{4.6}$$

such that the optimal choice of the over/under relaxation parameter  $\omega$  can yield super Gauss-Seidel convergence rates. Again, as will be seen in Part II, since  $[K]$  is cyclic, i.e. 2-cyclic, it follows that  $\omega \in (1,2)$ . In the case that  $\omega \equiv 1$ , the Gauss-Seidel method is retrieved.

For the five point finite difference operator, prototypical simulations yield  $[K]$  which are Stieltjean type M matrices. As such, the convergence of the Jacobi and Gauss-Seidel schemes are guaranteed. This is also true of the SOR method for optimal choices of  $\omega$ . Note, if the SOR method is employed at the local level, each distinct block of  $[K_D]$  could have its own  $\omega$ . Much of this will be formalized in Part II.

Note each of the various partitions of  $[K_D]$  are symmetric and positive definite. Hence, it follows that locally the conjugate gradient will yield convergent inverses.

To yield a stable and efficient convergence process, the initial guess/starting values need to be relatively accurate. For example, the initial seeding of the iteration process can be achieved via a multi-grid type procedure wherein interpolation/extrapolation is used to obtain the necessary information. For the case of local partitions involving direct solvers, the information flow would occur at partition boundaries. In the case of iterative solvers, information should be generated throughout the interior and on the boundaries of the pertinent partitions.

As will be seen later, in addition to optimizing bandwidth minimization, i.e. processor loading, the hierarchical methodology improves the convergence characteristics of iterative schemes. For instance, consider the conjugate gradient procedure. Like the steepest descent method [7], the derivation of the conjugate method is rooted in the quadratic form, namely

$$f = \frac{1}{2} (\underline{T}_I^l)^t [{}_I K_I^l] \underline{T}_I^l - (Q)^t \underline{T}_I^l + c \quad (4.7)$$

where here we consider the inverse of the  $[{}_I K_I]$  diagonal block of  $[K_D]$ . Overall the algorithms various steps are defined by the following sequence of operations namely

$$\tau_k^l = (\underline{g}_k^l)^t \underline{g}_k^l / ((\underline{d}_{k-k}^l)^t [{}_I K_I^l] \underline{d}_{k-k}^l)$$

$$(\underline{T}_I^l)_{k+1} = (\underline{T}_I^l)_k + \tau_k^l \underline{d}_{k-k}^l \quad (4.8)$$

$$\underline{g}_{k+1}^l = \underline{g}_k^l + \tau_k^l [{}_I K_I^l] \underline{d}_{k-k}^l$$

$$\beta_k^l = (\underline{g}_{k+1}^l)^t \underline{g}_{k+1}^l / ((\underline{g}_k^l)^t \underline{g}_k^l) \quad (4.8)$$

$$\underline{d}_{k+1}^l = - \underline{g}_{k+1}^l + \beta_k^l \underline{d}_{k-k}^l$$

such that here

$$\underline{g}_k^l = [{}_I K_I^l] (\underline{T}_I^l)_k \quad (4.9)$$

Note that:

- i)  $\tau_k^l$  defines the optimal step size along the search direction defined by  $\underline{d}_k^l$ ; and,
- ii)  $\beta_k^l$  defines the parameter which yields the optimal search direction.

When the conjugate gradient method is applied globally, there exists but one set of  $\tau_k$  and  $\beta_k$ , i.e. the global set. In this context they reflect the needs of the globally assembled set. Prototypically, the needs of a given partition may differ hence leading to convergence problems. For the current hierarchical methodology, the  $(\tau_k^l, \beta_k^l)$  pair is optimized on a local partition basis. In this context they reflect the intrinsic substructural requirements. As will be seen, this greatly improves the convergence characteristics of the overall problem. This is especially true for problems wherein the conjugate gradient methodology is used locally and say the SOR is employed globally.

To improve the overall convergence process, prototypically after the initial seeding, the local iteration process is allowed to converge to a predetermined accuracy. Subsequently, the global iteration is commenced. This enables the refinement of the seed values. As noted earlier, the Part II of this series will develop the formal properties of the foregoing solution procedures. This will also include benchmarking.



## References

1. Hwang, K., Briggs, F.A., "Computer Architecture and Parallel Processing", McGraw-Hill, New York, 1984.
2. Rehak, D.R., Keiroug, W.T., Hendrickson, C.T., and Cendes, Z.J., "Evaluation of Finite Element System Architecture", Computer and Structures, Vol. 20, pp. 17-29, 1985.
3. Larson, R., "Multi-Microprocessor Architecture for Parallel Computing", Computational Structural Mechanics Workshop, NASA Langley, November 1981.
4. Miles, D., Carlile, B.R., Groshong, J., "Parallel Programming on the FPST Series", International Super Computer Conference, 1988.
5. Cuthill, E.H., "Several Strategies for Reducing the Bandwidth of Matrices", Sparse 2 Conference, pp.157-166.
6. Varga, R., "Matrix Iterative Analysis", Prentice-Hall, New Jersey, 1962.
7. Concus, P., Golub, G.H. and O'Leary, D. P., "A Generalized Conjugate Gradient Method for the Numerical Solution of the Elliptic Partial Differential Equations", Sparse Matrix Computations, Academic Press, New York, 1979.
8. Ozisik, M.N., "Heat Conduction", Wiley, New York, 1980.
9. Bellman, R., "Introduction to Matrix Analysis", McGraw-Hill, New York, 1970.
10. Bathe, K.J., "Finite Element Procedure in Engineering Analysis", Prentice-Hall, New Jersey, 1984.
11. Irons, B.M., "A Frontal Solution Program for Finite Element Analysis", Int., Jr. Numerical Method Engineering, Vol. 2, pp. 5-32, 1970.
12. Lestingi, J. and Prachuktam, S., "A Blocking Technique for Large Scale Structural Analysis", Computers and Structures, Vol. 3, pp. 669-714, 1973.
13. Padovan, J. and Lackney, J., "Constrained Hierarchical Least Square Nonlinear Equations Solvers", Computers and Structures, Vol. 23, pp. 251-263, 1986.
14. George, A. and Liu, J.W., "Computer Solution of Large Sparse Positive Definite Systems", Prentice-Hall, New Jersey, 1981.
15. Stein, P. and Rosenberg, R.L. "On the Solution of Linear Simultaneous Equations by Iteration", J. London, Math. Soc., Vol 23, pp. 111-118, 1948.

Table 3.1 - Computational Effectiveness for Load Balance Equalized System: Two Level Hierarchy.

$B_g$	$\kappa$	$\kappa^2$	$n$	$R_{e/g}$	$R_{t/g}$	$(R_{t/g})^{-1}$
500	3.98	15.8	125.6	.0358	.0717	13.9
5000	6.31	39.8	792.4	.00568	.0113	88.5
10000	7.24	52.5	1379.0	.00326	.00652	153.4
50000	10.0	100	5000	.0009	.0018	555.5
100000	11.48	131	8710	.000516	.00103	967.7

Table 3.2 - Computational Effectiveness for Effort Optimized System:  
Two Level Hierarchy.

$B_g$	$\kappa$	$\kappa^2$	n	$R_{e/g}$	$R_{t/g}$	$(R_{t/g})^{-1}$
500	5.25	27.5	95.2	.0118	.059	16.9
5000	8.32	69.3	600.9	.00187	.00935	106.8
10000	9.56	91.5	1046	.00107	.0053	185.9
50000	13.2	174	3787	.000296	.00147	677.3
100000	15.2	229	6579	.000168	.00085	1178.4

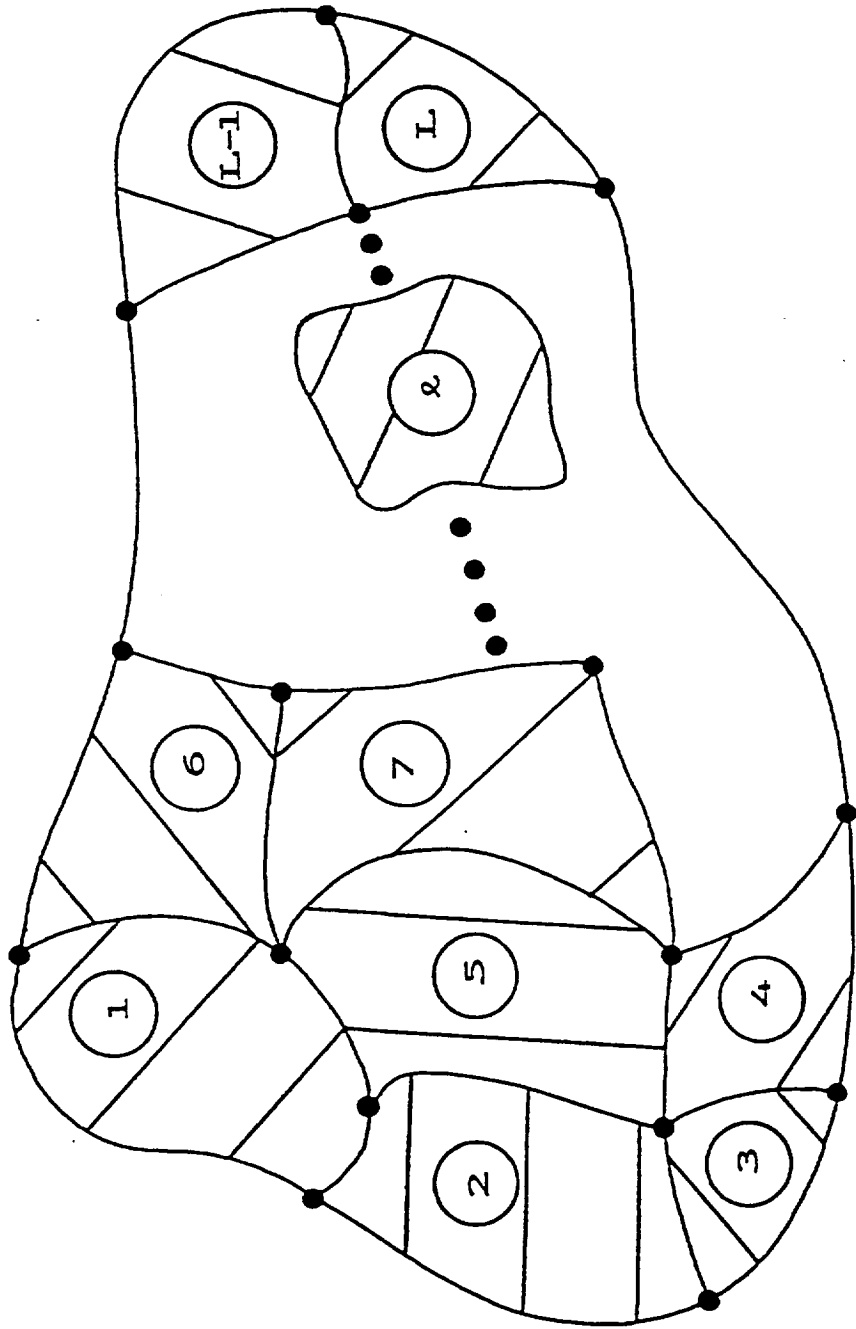


Figure 3.1 - Partitioned Physical System

LEVEL

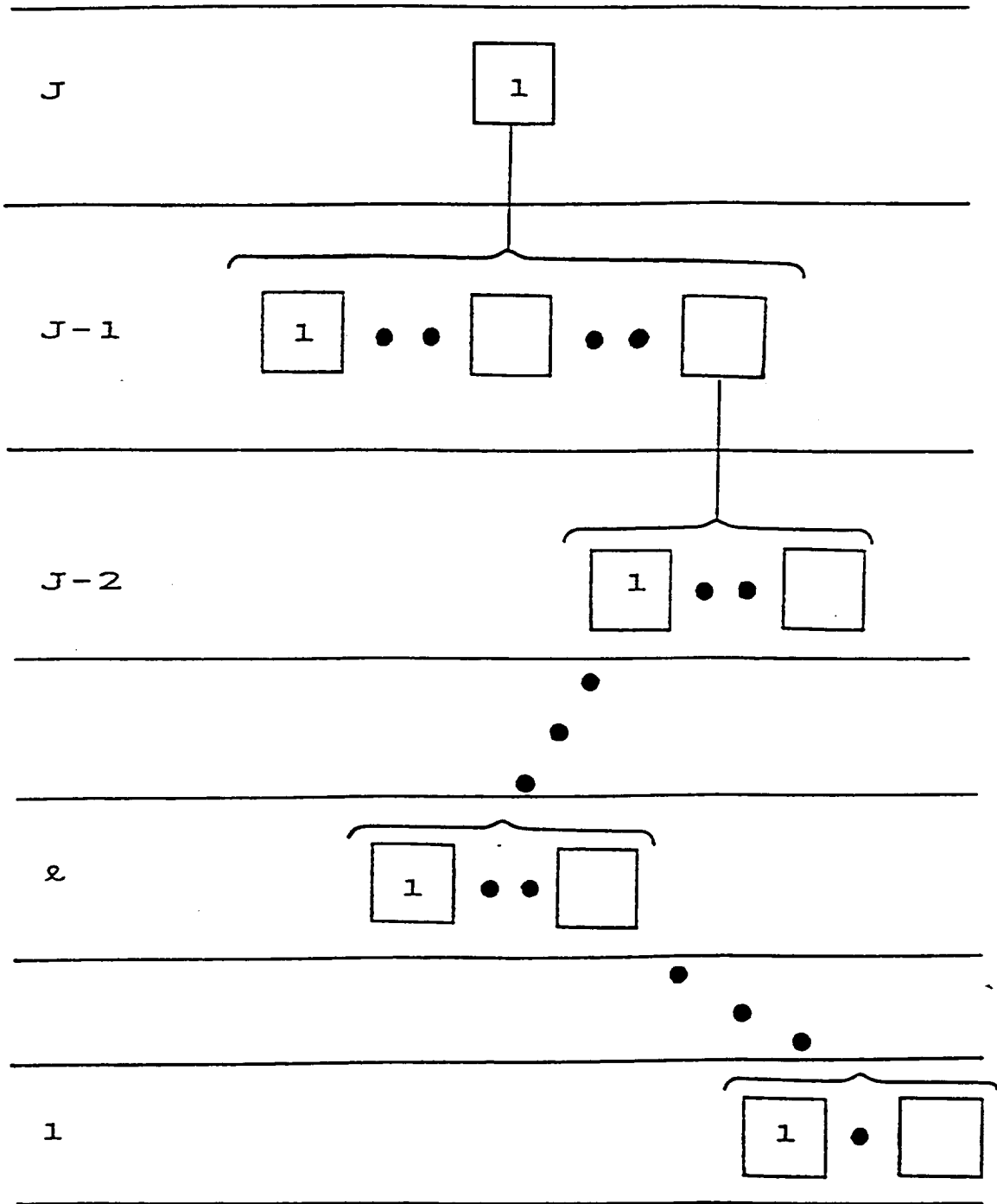


Figure 3.2 - Multilevel Hierarchical Scheme

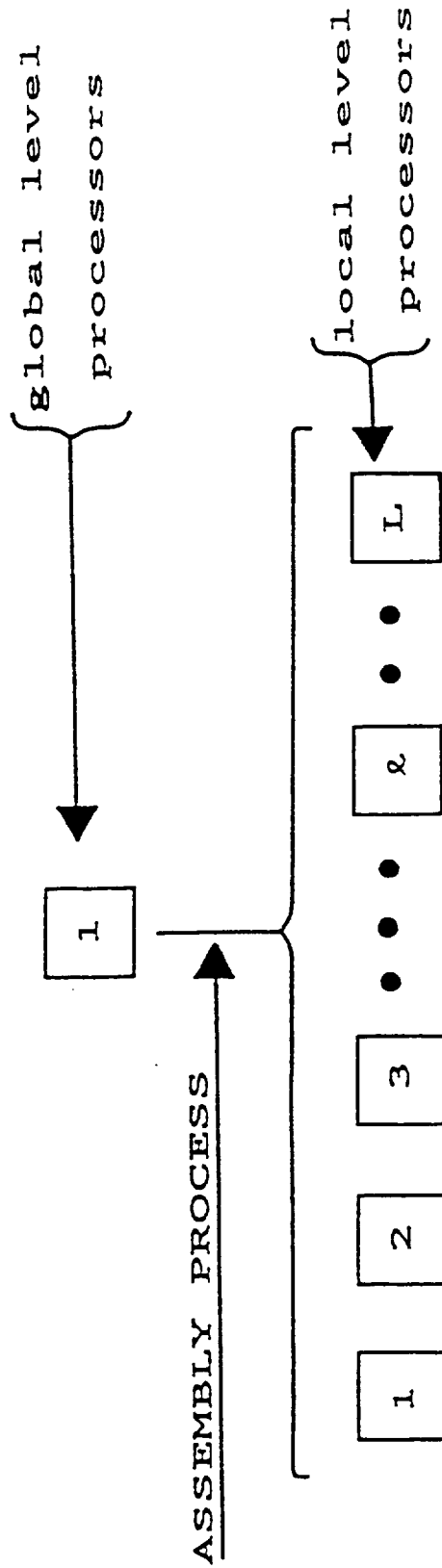


Figure 3.3 - Two Level Architecture

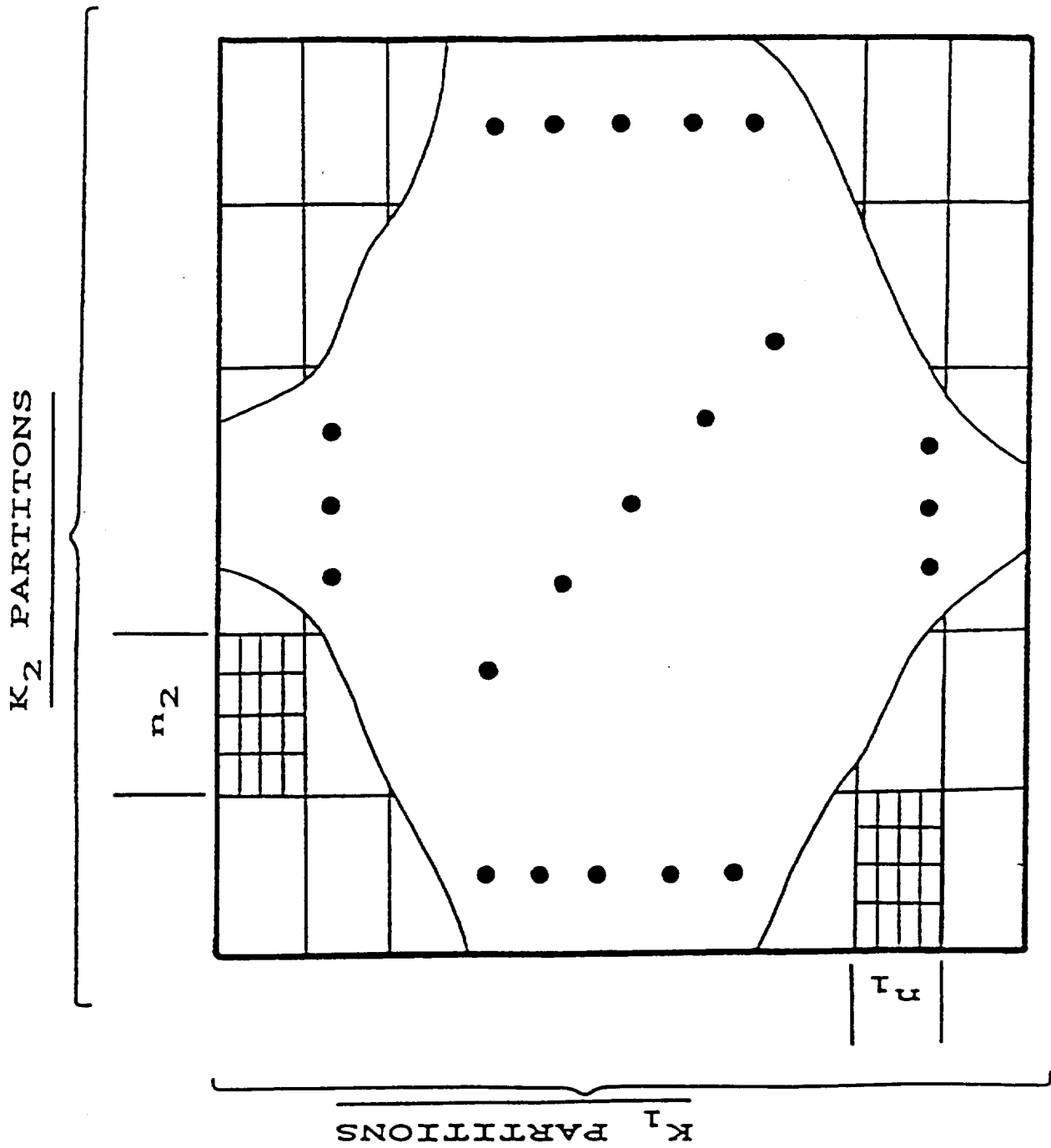


Figure 3.4 - Two Level Partitioned Model

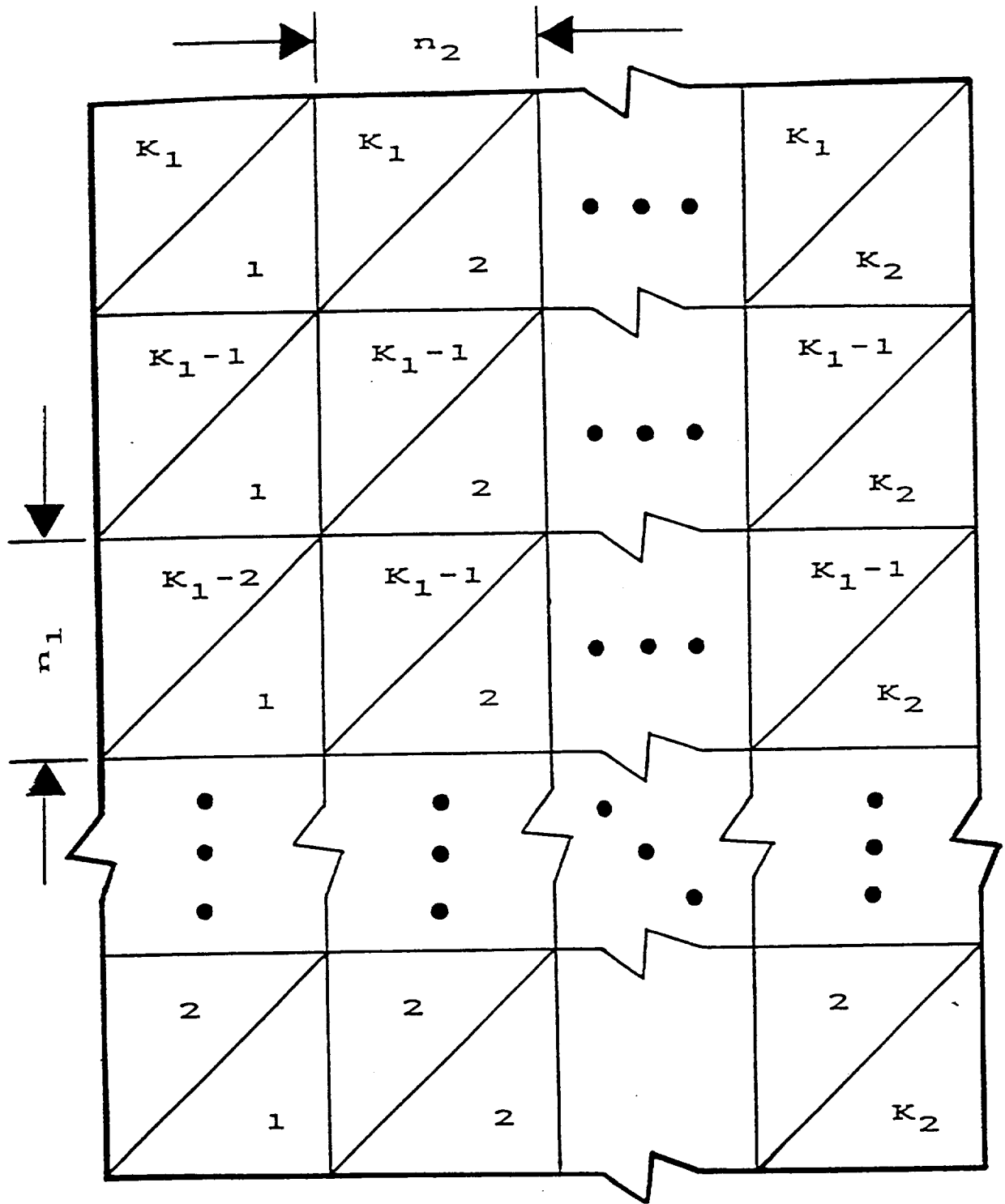


Figure 3.5 - Second Level External/Global Model



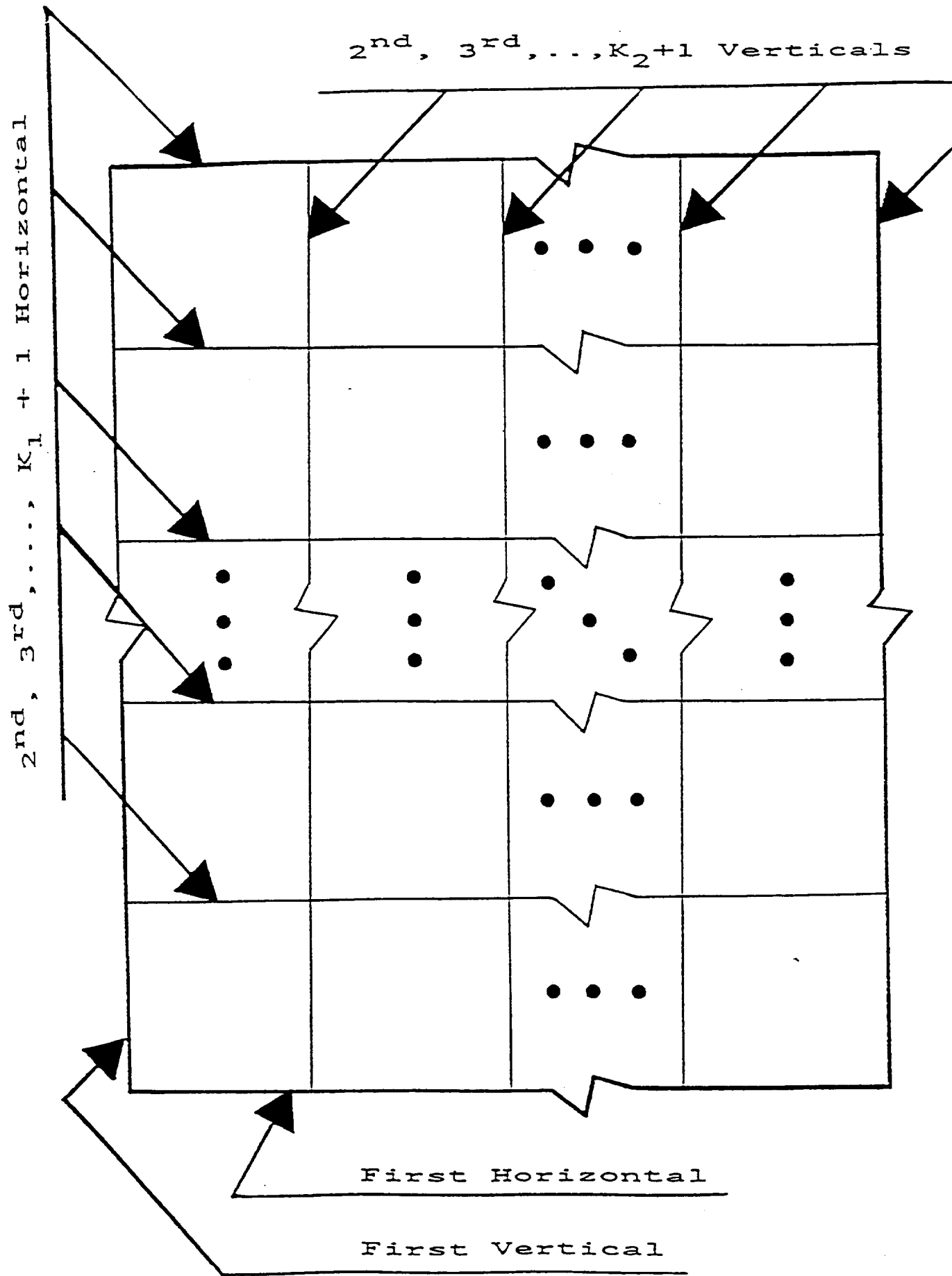


Figure 3.6 - Families of Verticals and Horizontals of Second External Global Level.

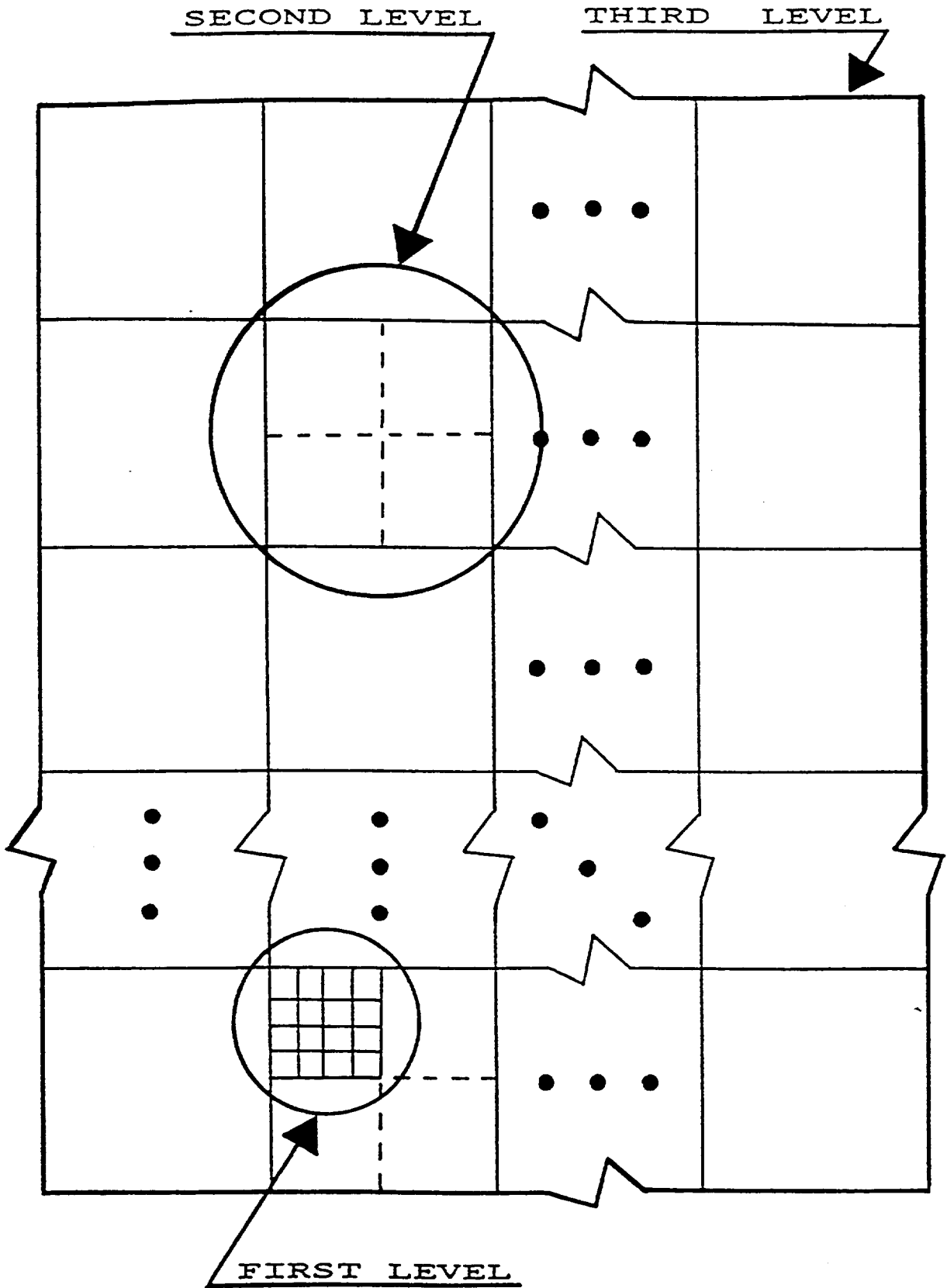


Figure 3.7 - Three Levels Hierarchy

LEVEL

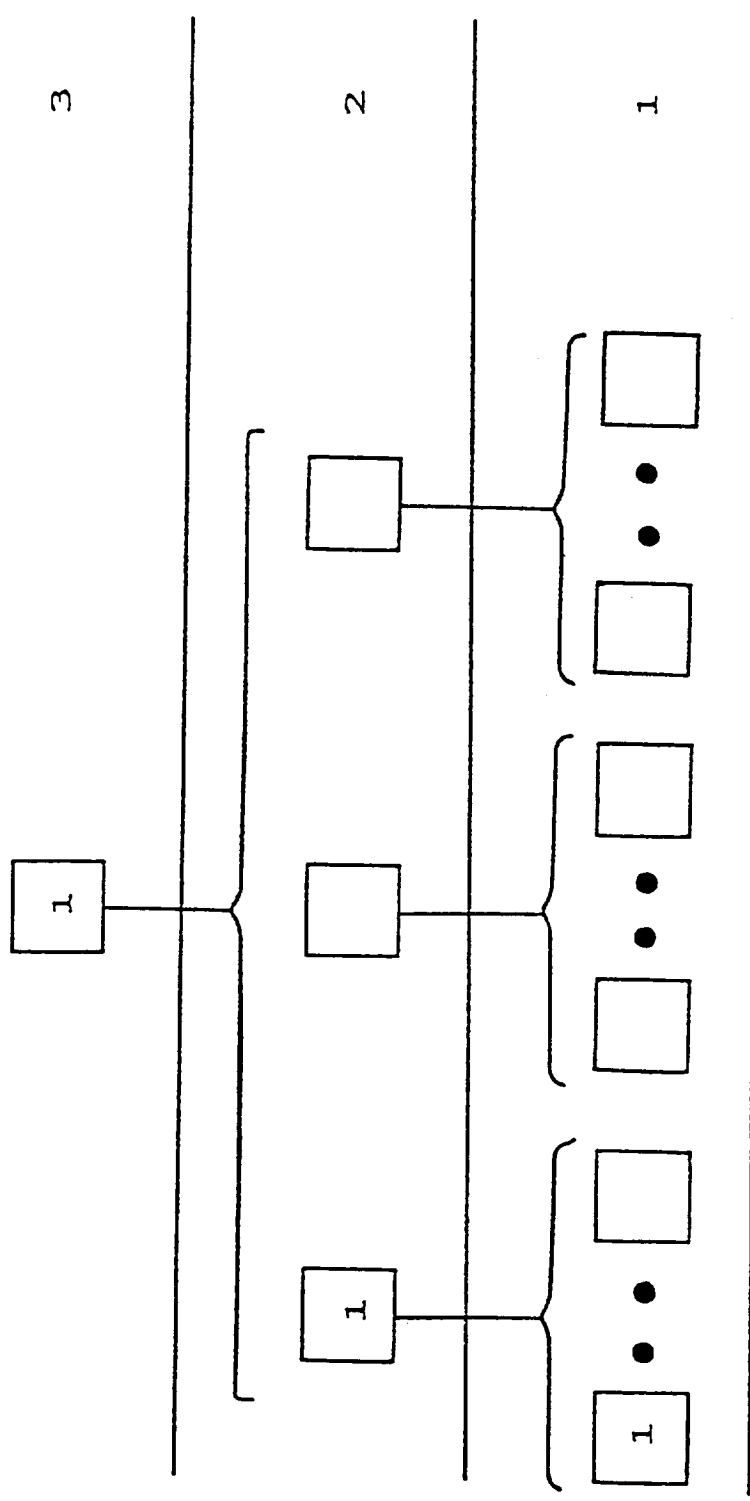


Figure 3.8 - Three Level Hierarchy of Operators

## PART II - FORMAL CONSIDERATIONS AND BENCHMARKING

### SUMMARY

In Part I of this series, a hierarchically parallel finite difference modeling methodology and associated solution scheme was developed. The main purpose was to establish a logical and efficient procedure for use in parallel computer environments. In this part, consideration is given to establish the formal numerical properties of the scheme. Specifically, several theorems and associated proofs are discussed which consider the convergence characteristics of the various solutions algorithms developed in association with the parallel methodology. These are backed up with large scale benchmark experiments performed on a VAX 785, a vectorized IBM 3090-200 and a CRAY-XMP.

## 1. INTRODUCTION

As noted in Part I, the main thrust of this series has been the development of an hierarchically parallel modelling and solution strategy for finite difference analysis of heat conduction. The basic emphasis of Part I is to formulate the overall strategy and solution methodology. In this part, formal consideration and benchmarking will be undertaken.

Included for consideration will be an evaluation of:

- i) The formal structure of the governing heat conduction equations and associated solution algorithms;
- ii) The convergence properties;
- iii) Spectral properties; and
- iv) A comparison of convergence rates.

The benchmarking of the parallel scheme will consider the various versions of the procedure, i.e., the direct, mixed, and purely iterative formulations.

Note, the formal evaluation of properties will employ the various theoretical aspects of P-cyclic, Stieltjean, nonnegative and M-matrices [1,2,3]. Overall this will enable a very detailed and comprehensive evaluation of the requirements for convergence, the spectral properties of the various versions of the parallel algorithms as well as the relative behavior among the various formulations.

Overall this part is organized into three major sections which:

- i) overview the algorithmic structure and properties of the various coefficient matrices;
- ii) give a detailed consideration of the formal numerical properties and
- iii) thoroughly benchmark the various versions of the scheme on IBM 3090, Vax 895 and CRAY XMP systems.

## 2. ALGORITHMIC OVERVIEW

Recalling Part I of the paper, the global assembled version of the governing FD equations of heat conduction take the form

$$[[K_D] - [K_U] - [K_L]] \underline{T} = \underline{Q} \quad (2.1)$$

where  $[K_D]$ ,  $[K_U]$ , and  $[K_L]$  respectively contain the block diagonal, upper and lower triangular elements of  $[K]$  the global conductivity matrix. Overall the forms are given by the expressions

$$[K_D] =$$

$$\begin{bmatrix} [{}_I K_I] & [0] & [0] \\ [0] & [{}_{DB} K_{DB}] & [0] \\ [0] & [0] & [{}_{MB} K_{MB}] \end{bmatrix} \quad (2.2)$$

$$[K_U] = [K_L]' =$$

$$\begin{bmatrix} [0] & [{}_I K_{DB}] & [{}_I K_{MB}] \\ [0] & [0] & [{}_{DB} K_{MB}] \\ [0] & [0] & [0] \end{bmatrix} \quad (2.3)$$

Equation (2.1) can be solved in either of three ways,

namely:

- i) direct inversion;
- ii) purely iteratively or;
- iii) by mixed direct and iterative procedures.

As seen earlier, this can be achieved at a variety of hierarchical levels. For such a formulation, the various partitions of  $[K_D]$  can be interpreted as follows, namely:

- i)  $[{}_I K_I]$  is a diagonal block matrix whose diagonal elements are nonsingular submatrices corresponding to the internal variables of each substructure.
- ii)  $[{}_{DB} K_{DB}]$  is a diagonal block matrix whose diagonal elements are nonsingular submatrices corresponding to dual shared boundaries between submatrices; and
- iii)  $[{}_{MB} K_{MB}]$  is a nonsingular diagonal matrix corresponding to multiply shared boundaries.

In a similar context,  $[{}_I K_{DB}]$ ,  $[{}_I K_{MB}]$ ,  $[{}_{DB} K_{MB}]$  and their transposes define the assembled connectivity matrices linking the various substructure.

For the direct procedure, (2.1) can be solved either in its global form or at the various substructural levels using static

condensation and forward elimination and backward substitution steps as discussed in Part I.

In the case of matrix iteration, as noted earlier, either a mixed or fully iterative solution methodology can be employed. Overall the solution process involves several levels. These primarily consist of local and global phases of calculation. At the global level, an iterative scheme is employed, i.e., either Jacobi, Gauss-Seidel or SOR methods [1,2]. For the local level, inverses of the various substructural matrices can either be obtained via direct means or by say the robust preconditioned conjugate gradient methodology [4,5].

As an example, consider the Jacobi type of a global formulation. Here (2.1) takes the form

$$[K_D] \underline{T}_{n+1} = ([K_U] + [K_L]) \underline{T}_n + \underline{Q} \quad (2.4)$$

or more directly

$$\underline{T}_{n+1} = [K_D]^{-1} \{([K_U] + [K_L]) \underline{T}_n + \underline{Q}\} \quad (2.5)$$

Overall, (2.5) consists of two levels of calculation, namely:

- i) The local inversion of  $[K_D]$ , and
- ii) The evaluation of  $\underline{T}_{n+1}$  after the appropriate matrix multiplication steps.



Note the inverse of  $[K_D]$  can be established partition by partition. This is possible since there is no coupling between the various blocks making up  $[I K_I]$ ,  $[DB K_{DB}]$  and  $[MB K_{MB}]$ . Hence,

$$[K_D]^{-1} = \begin{bmatrix} [I K_I]^{-1} & [0] & [0] \\ [0] & [DB K_{DB}]^{-1} & [0] \\ [0] & [0] & [MB K_{MB}]^{-1} \end{bmatrix} \quad (2.6)$$

As noted earlier, the inversion of the various blocks can be achieved either directly, or via mixed direct and iterative schemes. Here the conjugate gradient method with preconditioning could be applied to great advantage especially for well conditioned partitions.

For the Gauss-Seidel styled methodology, (2.1) takes the form

$$T_{n+1} = ([K_D] - [K_L])^{-1} ([K_U]T_n + Q) \quad (2.7)$$

Here the inversion process is somewhat more awkward.

To determine the various formal numerical characteristics of such algorithms, the properties of the various coefficient matrices must be defined. To simplify the development, without loss of generality, we shall consider formulations involving the use of 5-point 2-D and 7-point 3-D finite difference representations [1,2]. In this context it follows that  $[K]$  is positive definite Hermitian and  $[K_L]$ ,  $[K_U]$  are nonnegative.

strictly lower and upper triangular matrices. Such properties also apply at the substructural level. The next section will employ these properties to formally illustrate the convergence characteristics of the hierarchical methodology.

### 3. FORMAL CONSIDERATIONS

Overall the formal considerations will have four main thrusts, i.e.:

- i) convergence properties;
- ii) spectral properties;
- iii) comparison of convergence rates among full iterative and mixed approach, and;
- iv) local global attributes.

This will be established in a series of theorems and their associated proofs.

First we will show that the global Jacobi and Gauss-Seidel (GS) iterative method defined by (2.5) and (2.7) respectively converges for any arbitrarily initial vector  $\tilde{T}_0$ , i.e., if  $\tilde{T}$  is the true solution of (2.1) and  $\{\tilde{T}_n\}$  are the sequence derived from (2.5) or (2.6), then  $\lim_{n \rightarrow \infty} \tilde{T}_n = \hat{\tilde{T}}$ .

From (2.5) and (2.7), the Jacobi and GS iteration matrices respectively are given by

$$\begin{aligned} J &= [K_D]^{-1} \{[K_L] + [K_V]\} \\ \varphi &= \{[K_D] - [K_L]\}^{-1} [L_U]. \end{aligned} \quad (3.1)$$



$$J(A) = \begin{bmatrix} 0 & B_{1,2} & & & & \bigcirc \\ B_{2,1} & 0 & B_{2,3} & & & \\ & & \bullet & & & \\ & & & \bullet & & \\ & & & & \bullet & \\ & \bigcirc & & & & \bullet \\ & & & & B_{q,q-1} & \\ & & & & & \bigcirc \\ & & & & & B_{q-1-q} \\ & & & & & 0 \end{bmatrix} \quad (3.3)$$

corresponding to the partitioning of A. Note  $J(A)$  is defined as weakly cyclic of index 2 and A as 2-cyclic.

Let  $J(A) = L + U$ , where L and U are strictly the lower and upper triangular parts of  $J(A)$  in block form.

Definition. If the matrix A of (3.2) is 2-cyclic, then the matrix A is consistently ordered if all the eigenvalues of the matrix

$$J_{\alpha}(A) = \alpha L + \alpha^{-1}U, \quad (3.4)$$

are independent of  $\alpha$ , for  $\alpha \neq 0$ .

Now we give the following convergence theorem for the Jacobi and Gauss-Seidel iterative methods.

Theorem A. The block Jacobi and block Gauss-Seidel iterative methods defined by (2.5) and (2.7) are convergent for the block system of linear equations (2.1) for any initial vector approximations  $T_0$ . Moreover, the Gauss-Seidel Method converges faster than Jacobi method.

Proof. For the matrix  $[K]$  of (2.1) derived from the five point finite difference formula applied to self-adjoint elliptic (Heat conduction) equation, it is known from Theorem 6.4 of Varga [1] that  $[K]$  is positive definite Hermitian and a Steiltjes matrix. We remark that every Steiltjes matrix is an M-matrix.

Define

$$\begin{aligned} M_1 &= [K_D], \quad N_1 = [K_L] + [K_U]; \\ M_2 &= \{[K_D] - [K_L]\}, \quad N_2 = [K_U]. \end{aligned} \quad (3.5)$$

Since  $[K] = [K_D] - [K_L] - [K_U]$  is an M-matrix. It follows from Theorem 3.14 of Varga [1] that  $M_1$  and  $M_2$  are also M-matrices. Thus, it follows that  $[K] = M_1 - N_1 = M_2 - N_2$  are two regular splittings of  $[K]$ . Moreover  $N_1 \geq N_2 \geq 0$ . Hence by theorem 3.15 of Varga [1],

$$0 < \rho (M_2^{-1} N_2) < \rho (M_1^{-1} N_1) < 1 \quad (3.6)$$

From (3.1), it follows that

$$\begin{aligned} J &= M_1^{-1} N_1, \\ \mathcal{J} &= M_2^{-1} N_2. \end{aligned} \quad (3.7)$$

Thus,

$$0 < \rho (\mathcal{J}) < \rho (J) < 1. \quad (3.8)$$

this implies that both the Jacobi and Gauss-Seidel iterative methods converge. Moreover Gauss-Seidel Method converges faster than Jacobi Method since  $\rho(\mathcal{G}) < \rho(J)$

In view of the above theorem, the Gauss-Seidel scheme offers improved convergence rates over the Jacobi version. If the over/under relaxation parameter is properly defined, the SOR prototypically yields improved results over the Jacobi and Gauss-Seidel methods. For the current hierarchical formulation, the SOR takes the following form namely

$$([K_D] - \omega [K_L] \mid T_{\sim n+1} = (1-\omega) [K_D] T_{\sim n} + \omega [K_U] T_{\sim n} + \omega Q \quad (3.9)$$

where  $\omega$  is the relaxation parameter.

If we use the five point finite difference scheme to simulate the heat conduction equation, then  $[I K_{MB}] = 0$  in equation (2.3). Hence, from the earlier definition, it follows that  $[K] = [K_D] - [K_L] - [K_U]$  is a 2-cyclic consistently ordered matrix. The SOR iteration matrix corresponding to (2.1) is then defined by

$$\mathcal{G}_\omega = ([K_D] - \omega [K_L])^{-1} \{(1-\omega) [K_D] + \omega [K_U]\}. \quad (3.10)$$

The convergence and optimal  $\omega$  for the SOR method defined by (3.9) is given in the following theorem which involves the properties of 2-cyclic and consistently ordered matrix theory [1.2.3].

Theorem B. Let the matrix  $[K] = [K_D] - [K_L] - [K_U]$  of (2.1) be a consistently ordered 2-cyclic matrix with nonsingular diagonal submatrices  $[K_D]$ . If all the eigenvalues of the second power of the associated block Jacobi matrix  $J$  are real and non-negative, and  $0 \leq \rho(J) < 1$ , then with

$$\omega_b = \frac{2}{1 + \sqrt{1 - \rho^2(J)}}. \quad (3.11)$$

it follows that

$$\rho(\mathcal{L}_\omega) = (\omega_b - 1)$$

and

$$\rho(\mathcal{L}_\omega) > \rho(\mathcal{L}_{\omega_b})$$

for all  $\omega \neq \omega_b$ . Moreover, the block successive overrelaxation matrix  $\mathcal{L}_\omega$  is convergent, i.e.,  $\rho(\mathcal{L}_\omega) < 1$ , for all  $\omega$  with  $0 < \omega < 2$ .

Proof. Note from Theorem A, we have  $0 \leq \rho(J) < 1$ . Since the block Jacobi matrix  $J$  is symmetric, all its eigenvalues are real. Hence eigenvalues of  $J^2$  will be non-negative. Thus, all the conditions of Theorem B are satisfied.

Since the Gauss-Seidel procedure converges faster than Jacobi's, we shall give an outline of the parallel version of the block Gauss-Seidel iterative method for the five-point finite

difference formula. It is expensive to find the optimal  $\omega$  for the SOR iterative method. However, we also give the analogue version for the SOR method.

There are three levels of computations. Level 1 consists of solving for all unknowns inside the substructures. Level 2 consists of all points on the double boundaries and level 3 consists of all points on the multiple boundaries.

Equations (2.1) can be written in the block form as follows:

$$\begin{bmatrix} [I^K_I] & -[I^K_{DB}] & [0] \\ -[I^K_{DB}] & [DB^K_{DB}] & -[DB^K_{MB}] \\ [0] & -[DB^K_{MB}] & [MB^K_{MB}] \end{bmatrix} \begin{Bmatrix} \tilde{T}^I \\ \tilde{T}^{II} \\ \tilde{T}^{III} \end{Bmatrix} = \begin{Bmatrix} \tilde{Q}^I \\ \tilde{Q}^{II} \\ \tilde{Q}^{III} \end{Bmatrix} \quad (3.12)$$

Here superscript denotes the levels.

Block Gauss-Seidel Version (Algorithm 1)

For  $n = 0, 1, 2, \dots$

Solve:

$$1. \quad [I^K_I] \tilde{T}^I_{(n+1)} = [I^K_{DB}] \tilde{T}^{II}_{(n)} + \tilde{Q}^I. \quad (3.13)$$

$$2. \quad [DB^K_{DB}] \tilde{T}^{II}_{(n+1)} = [I^K_{DB}] \tilde{T}^I_{(n+1)} + [DB^K_{MB}] \tilde{T}^{III}_{(n)} + \tilde{Q}^{II}. \quad (3.14)$$

$$3. \quad [MB^K_{MB}] \tilde{T}^{III}_{(n+1)} = [DB^K_{MB}] \tilde{T}^{II}_{(n+1)} + \tilde{Q}^{III} \quad (3.15)$$



Block SOR Version (Algorithm 2)

$$1. \quad [I^K I] \tilde{T}_{(n+1)}^I = (1-\omega) [I^K I] \tilde{T}_{(n)}^I + \omega [I^K_{DB}] \tilde{T}_{(n)}^{II} + \omega Q_{\sim}^I \quad (3.16)$$

$$2. \quad [DB^K_{DB}] \tilde{T}_{(n+1)}^{II} = \omega [I^K_{DB}] \tilde{T}_{(n+1)}^I + (1-\omega) [DB^K_{DB}] \tilde{T}_{(n)}^{II} \quad (3.17)$$

$$+ \omega [DB^K_{MB}] \tilde{T}_{(n)}^{III} + \omega Q_{\sim}^{II}, \quad (3.18)$$

$$3. \quad [MB^K_{MB}] \tilde{T}_{(n+1)}^{III} = \omega [DB^K_{MB}] \tilde{T}_{(n+1)}^{II} + (1-\omega) [MB^K_{MB}] \tilde{T}_{(n)}^{III} + \omega Q_{\sim}^{III} \quad (3.19)$$

The number of processors required to solve (3.13) are equal to the number of substructures. Number of processors required to solve (3.14) are equal to the number of double boundary lines. Finally, the number of processors required to solve (3.15) are equal to the number of multiple boundaries. The same number of processors are also required for the SOR method.

Note, theorems A and B are based on the assumption that the inverse of  $[I^K I]$ ,  $[DB^K_{DB}]$  and  $[MB^K_{MB}]$  are obtained via direct methods, i.e. Gauss elimination, Cholevsky decomposition etc. However, as will be seen later, such local inverses can be obtained by a convergent iterative scheme, i.e. the conjugate gradient scheme. This gives the overall scheme two phases of iteration, the local and global.

The previous formalism pertained primarily to the global level of the iteration process. As such, it is unaffected by converged local calculations of  $[K_D]^{-1}$ . Specifically the global level formalism remains intact if:

- i) The complete inversion of  $[K_D]$  is performed directly;
- ii) The complete inverse of  $[K_D]$  is obtained via a convergent local iteration process and;
- iii) If the inverse of  $[K_D]$  is obtained via a mixture of direct and convergent iterative scheme.

Note since all the various partitions and associated blocks of  $[K_D]$  are Stieltjean, convergence is guaranteed for such iterative methodologies as:

- i) conjugate gradient with and without preconditioning.  
or;
- ii) the more classical Jacobi and Gauss Seidel and SOR schemes.

With the use of five-point 2-D and seven-point 3-D difference representations, the preconditioning follows directly from the structure of the partitions of  $[K_D]$ . For instance, considering the five-point 2-D difference formulation, the various substructural blocks making up the  $[_I K_I]$  partition of  $[K_D]$  are five diagonal. For this case, the preconditioning used in conjunction with the conjugate gradient method would be structured accordingly, i.e. to preserve the five diagonal format [5].

As will be seen during the benchmarking procedure, for the mixed and completely iterative hierarchical methodologies, the overall iteration process can be performed in several different ways, i.e.:

- i) for each cycle of the global iteration, all the distinct blocks are iterated until locally converged;
- ii) for each cycle of the global iteration, various of the local iteration processes are consequential that is, for certain designated blocks, each cycle of local iteration is followed by a global one, and;
- iii) each cycle of all iterated blocks are followed by a global iteration.

Note the proceeding theoretical development guarantees the convergence of case (i) schemes. As will be seen from the benchmarking, during the initial phases of the solution process, case (i) iterations are employed to obtain a better approximation to the solution. Once the desired accuracy is achieved, case (ii) and (iii) procedures can then be used to complete the solution. Such an approach is particularly advantageous to use when conjugate gradient procedure are employed locally. This follows from the fact that the rate of convergence of the conjugate gradient method improves in close neighborhoods of the solution. Recalling Part I, the basic conjugate gradient algorithm [6] takes the following form locally, namely:

$$\tau_k^e = (\underline{g}_k^e)' \underline{g}_k^e / (\underline{d}_k^e)' [\underline{I} K_I^e] \underline{d}_k^e \quad (3.20)$$

$$(\underline{T}_I^e)_{k+1} = (\underline{T}_I^e)_k + \tau_k^e \underline{d}_k^e \quad (3.21)$$

$$\underline{g}_{k+1}^e = \underline{g}_k^e + \tau_k [\underline{I} K_I^e] \underline{d}_k^e \quad (3.22)$$

$$\beta_k^e = (\underline{g}_{k+1}^e)' \underline{g}_{k+1}^e / ((\underline{g}_k^e)' \underline{g}_k^e) \quad (3.23)$$

$$\underline{d}_{k+1}^e = - \underline{g}_{k+1}^e + \beta_{k+1}^e \underline{d}_k^e \quad (3.24)$$

such that here the gradient  $\underline{g}_k^e$  is defined by the expression

$$\underline{g}_k^e = [\underline{I} K_I^e] (\underline{T}_I^e)_k \quad (3.25)$$

Specifically  $\underline{g}_k^e$  represents the gradient of the quadratic form

$$f = (\underline{T}_I^e)' [\underline{I} K_I^e] (\underline{T}_I^e) - (\underline{T}_I^e)' \underline{Q}_I \quad (3.26)$$

Typically, for the method of steepest descent,  $\underline{g}_k^e$  is used to define the search direction. For such an algorithm the subsequent  $\underline{g}_k^e$  are mutually orthogonal, i.e.

$$(\underline{g}_k^e)' \underline{g}_{k-1}^e = 0$$

For the conjugate gradient version of the scheme, the search direction is modified in the manner defined by (3.24). Since  $\beta_k^\ell$  is chosen to satisfy optimality conditions, the subsequent search directions are no longer perpendicular but rather are free to range within the optimal bounds.

Overall  $\tau_k^\ell$  defines the optimal step size along the optimized search direction defined by  $\beta_k^\ell$ . Note, these parameters pertain to the  $\ell^{\text{th}}$  subdomain. Since the  $[{}_I K_I^\ell]$  may all be distinct, the associated search direction and step sizes may all vary. This also applies to various diagonal blocks making up  $[{}_{DB} K_{DB}]$  and  $[{}_{MB} K_{MB}]$ . The overall family of  $\tau$  and  $\beta$  represents one of the distinct advantages of the hierarchical scheme. Namely, rather than one global  $(\tau, \beta)$  pair, the hierarchical scheme provides for locally defined  $(\tau^\ell, \beta^\ell)$  pairs, i.e. for the sets of substructural interior degrees of freedom, as well as for dual and multiply connected boundary variables. Since such pairs more properly reflect the local optimality conditions required for local convergence, the stability and efficiency of the CG is greatly enhanced. Such advantages can be further enhanced through the use of a preconditioned version of the CG, i.e., the PCG. This will be discussed in the next section.

As noted earlier, similar comments apply to the SOR scheme. Namely, distinct  $\omega$  could be generated for each of the various substructure and dual and multiply connected diagonal blocks.

In addition to enabling localized optimization, the hierarchical methodology tends to provide a means to decrease the effects of roundoff generated in all computer hardware. This applies both to the direct as well as iterative schemes. For the direct method, roundoff is decreased due to the significant reductions in bandwidth and problem size associated with substructuring. Another improvement in the roundoff problem follows from the fact that the partitioning process associated with the hierarchical strategy tends to zonalize such effects. This is especially true for the iterative schemes. For instance, recalling the various inner products associated with the CG, i.e. those defined by (3.18) and (3.21), partitioning can introduce significant reductions in the associated roundoff. Due to the substructuring process, such roundoff is somewhat zonally contained.

Such roundoff containment will have very significant impact on computer hardware. Specifically, to contain roundoff, many significant places must be carried for each number stored. This severely impacts memory data transfer, arithmetic operations, etc. By reducing roundoff, the hierarchical scheme can increase usable memory as well as simplify arithmetic and data transfer and storage operations.

#### 4. HIERARCHICAL PRECONDITIONING

To enable a further enhancement of iterative schemes, often times preconditioning is employed to associated matrix equation. This idea has been commonly used [4,5,7,8,9] in conjunction with iterative method such as conjugate gradient and symmetric successive overrelaxation (SSOR) methods at the global level. In the context of the hierarchical methodology developed herein, separate individualized preconditioning can be applied to each distinct local substructures. In this context, the appropriate local characteristics can be taken into account. In what follows, the concept of hierarchical preconditioning will be developed. For demonstration purposes, the preconditioning for the CG iterative method (PCG) will be discussed since it has a number of attractive properties such as

- i) it does not require any estimation of iteration parameter.
- ii) it takes advantage of the distribution of the eigenvalues of the iteration operation.
- iii) it requires fewer restrictions on the matrix for optimal behavior than does such methods as the SOR method.

For the sake of notational convenience, let us assume that the system of linear equations corresponding to a given substructure are defined by

$$A \tilde{x} = \tilde{b} \quad (4.1)$$

where  $A$  is a  $m \times m$  positive definite Hermitian matrix. Let  $A = M - N$  be an incomplete factorization of  $A$  [10] such that  $N$  is the error matrix and

$$M = L U \quad (4.2)$$

where  $L$  and  $U$  are lower and upper triangular matrices and  $M$  is nonsingular. Note  $L$  and  $U$  are selected so as to possess approximately the same sparse structure as the original matrix. This is in contrast to direct factorization which yields densely filled subdiagonal with the upper bounding bandwidth. Such a choice usually yields improved convergence characteristics.

The error matrix

$$N = M - A \quad (4.3)$$

is acceptable if it reduces the spectral condition number which is the ratio of the extreme eigenvalues of  $(LU)^{-1}A$ . Note reductions in the condition number enhance the rate of convergence of the conjugate gradient method. Before giving the error estimates, we give the algorithm for the PCG applied to (4.1) [4.5].



### PCG Algorithm

Let  $\tilde{x}^{(0)}$  be a given vector and arbitrary define the vector  $\tilde{q}^{(-1)}$ .

For  $k = 0, 1, 2, \dots$

(a) solve 
$$L U \tilde{z}^{(k)} = \tilde{b} - A \tilde{x}^{(k)} .$$

(b) compute 
$$b_k = \frac{\tilde{z}^{(k)'} L U \tilde{z}^{(k)}}{\tilde{z}^{(k-1)'} L U \tilde{z}^{(k-1)}} , \quad k \geq 1 .$$

$$b_0 = 0, \quad \tilde{q}^{(k)} = \tilde{z}^{(k)} + b_k \tilde{q}^{(k-1)} ,$$

(c) compute 
$$a_k = \frac{\tilde{z}^{(k)'} L U \tilde{z}^{(k)}}{\tilde{q}^{(k)'} A \tilde{q}^{(k)}} .$$

$$\tilde{x}^{(k+1)} = \tilde{x}^{(k)} + a_k \tilde{q}^{(k)} .$$

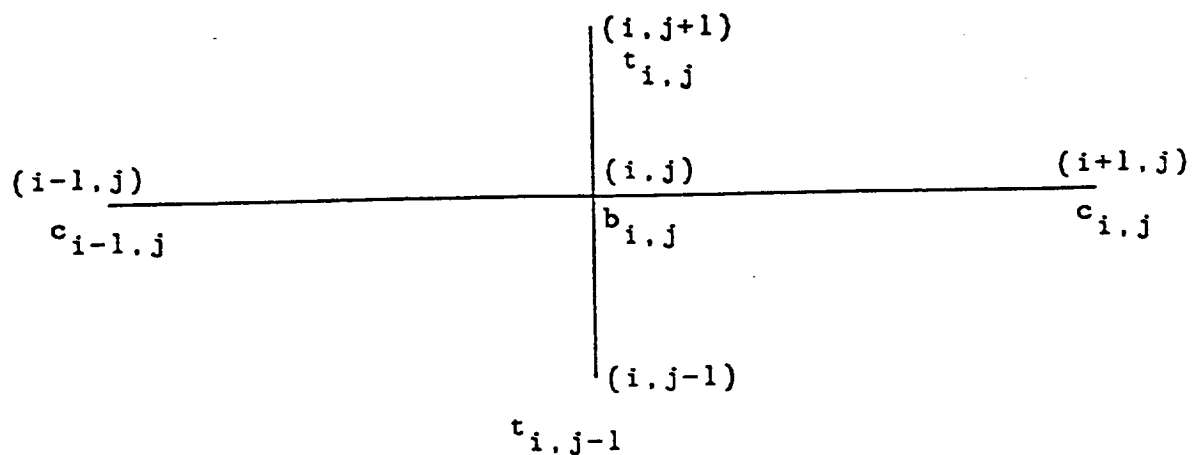
For the model problem, there are several ways to obtain the incomplete factorization [4.7,10] of  $A$  in the form

$$A = LL' - N. \quad (4.4)$$

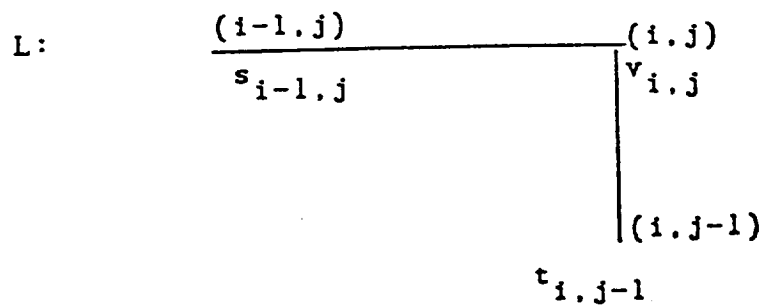
We will use a particular one given by Krishna [4] since it is inexpensive and more stable. The outline is given below.

Suppose we are using a five point difference formula to represent the general heat equation [11.12] then the graph of the nonzero entries of A corresponding to  $(i,j)$  mesh point is given by

A:

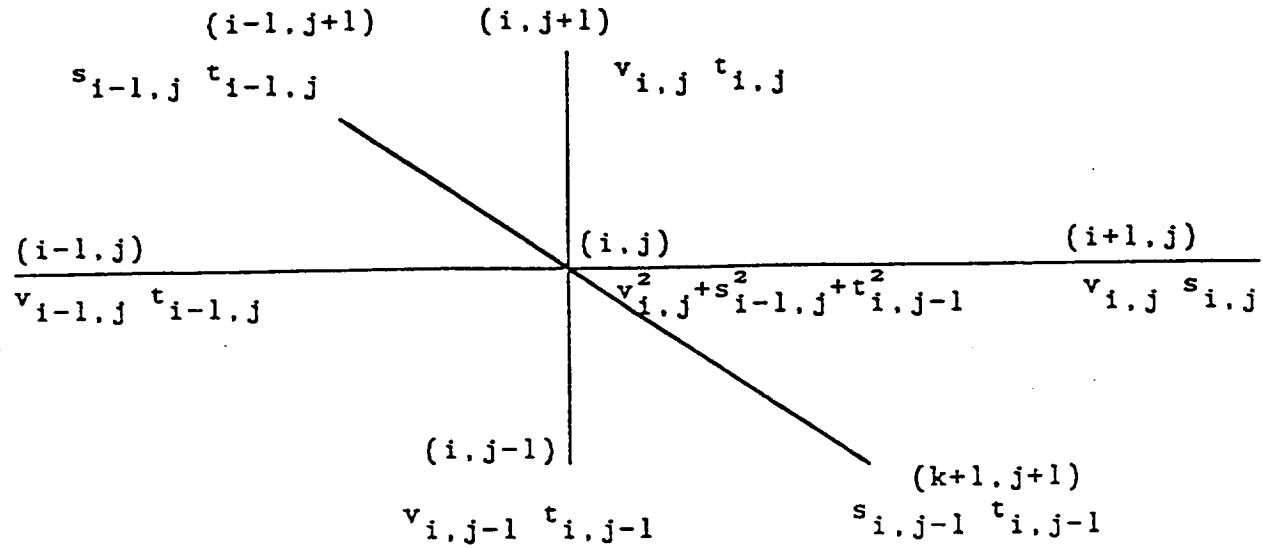


Let us represent the graph of nonzero entries of L corresponding to  $(i,j)$  mesh point by



Then the nonzero entries of  $LL^T$  are given by the graph

$LL^T$ :



We define  $v_{i, j}$ ,  $s_{i, j}$  and  $t_{i, j}$  by

$$\begin{aligned}
 v_{i, j} &= \sqrt{b_{i, j} - s_{i-1, j}^2 - t_{i, j-1}^2} \\
 s_{i, j} &= \frac{c_{i, j}}{v_{i, j}} \\
 t_{i, j} &= \frac{f_{i, j}}{v_{i, j}}
 \end{aligned}
 \tag{4.4}$$

It has been shown in [4] that  $v_{i, j} > 0$ . We remark that in the error matrix  $N$  we have at most two non-zero entries in each row. In step (a) of conjugate gradient algorithm, we can obtain the unknown vector  $\tilde{z}^{(k)}$  very easily by using back and forward substitution.

For the error estimate let the weighted error function after  $(\ell + 1)$  iterations be given by

$$e(\tilde{x}^{(\ell+1)}) = \frac{1}{2}(\tilde{x} - \tilde{x}^{(\ell+1)})' A(\tilde{x} - \tilde{x}^{(\ell+1)}) \quad (4.5)$$

Then it is known [4] that

$$\frac{e(\tilde{x}^{(\ell+1)})}{e(\tilde{x}^{(0)})} \leq 4 \left( \frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1} \right)^{2(\ell+1)} \quad (4.6)$$

where  $\kappa$  is the spectral norm of  $(LL')^{-1}A$ .

## 5. BENCHMARKING

In Part I of this series, a hierarchically parallel modelling methodology was developed. Overall, a given problem is partitioned into a number of separate substructure each with its own distinct internal and interconnecting conductivity matrices. In a parallel computer environment, these can be formulated simultaneously. As has been seen, the solution to the problem then contains at least two phases, i.e. the local and global wherein each substructure can be evaluated in a distinct processor. The solution to the resulting formulation would then be achieved either by:

- i) static condensation and back substitution locally and global assembly and direct solution of the resulting statically reduced global matrix;

- ii) static condensation and back-substitution locally and global assembly and iterative solution of statically reduce matrix globally;
- iii) statically reduce and back-substitute certain chosen local substructure while iteratively solving others in conjunction with global level iterations and
- iv) perform local parallel iterations approximately sequenced with the global phase of iteration.

For problems with poor conditioning, i.e. large spectral radius, method i) would yield the most stable results. In the case that certain substructure are well-conditioned spectrally, then methods ii) and iii) would be of greater advantage. Lastly, if all the various substructural partitions are spectrally well-conditioned, then method iv) would be of greatest advantage. This follows from the fact that the amount of data flow between substructure and the global level is reduced in the flow of calculations associated with the iterative process. Rather than whole matrices, pre and post multiplication reduces most of the data flow to vector form. Secondly, since the iteration process preserves the original matrix sparsity, the storage requirements are significantly reduced. Last, but not least, the direct method suffer from the fact that as problem size grows, roundoff is significantly accelerated. Hence, it requires more significant places thereby further reducing available machine storage.

To benchmark the hierarchical procedure described earlier, the following example problem was chosen namely:

i) For all  $0 < x < 1$ ,  $0 < y < 1$ .

$$\nabla^2 T = 2[(5 + x + y) \sin(x + y) - 2 \cos(x + y)] \quad (5.1)$$

$$\nabla^2 ( ) = \frac{\partial^2}{\partial x^2} ( ) + \frac{\partial^2}{\partial y^2} ( ) \quad (5.2)$$

ii) For all  $0 < x < 1$ ,  $y = 0$  or  $1$ ;

$$T = (5 + x + y) \sin(x + y) \quad (5.3)$$

iii) For all  $0 < y < 1$ ,  $x = 0$  or  $1$ ;

$$T = (5 + x + y) \sin(x + y) \quad (5.4)$$

During the evaluation phase, coarse to very refined FD meshes were tested. The most refined model consisted of some 250,000 mesh points representing a like number of total equations. Program development and testing was initiated on an IBM 3090-200 with a vector facility. To enable the largescale evaluations, the program was migrated to a CRAY XMP with an 8 megaword memory, i.e., The University of Pittsburgh machine. While actual parallel processing was not possible (except for four partitions, i.e. CRAY limitations) the overall scheme was run sequentially, i.e. local phase requisite assembly, data flow and global phase.

Three types of solution procedure were considered, namely:

- i) direct global as well as local, i.e.  
condensation-backsubstitution locally and direct  
solution of reduced global matrix;
- ii) locally direct and iterative globally and lastly;
- iii) local iteration and global iteration.

Note all the local iterations were performed employing the preconditioned conjugate gradient methodology [5] noted earlier. Here we recall that the preconditioner preserved the five diagonal form of the various substructural blocks making up the  $[K_I]$  partition.

Note for all test cases involving either global or local-global iterations, an initial guess was employed along the boundaries of the various substructure. The guess was obtained by employing a coarse mesh to generate a preliminary solution. This was then extrapolated/interpolated onto the boundaries of the substructure, i.e. on their associated boundary mesh points.

As will be seen, such an approach greatly improves the numerical efficiency of the iterative scheme. This follows for several reasons, namely:

- i) if direct inversion is used at the local level, the roundoff error associated with the backsubstitution procedure tends to destabilize the global iteration process for very large scale problems. In this context, seeding the solution with a reasonable initial guess tends to limit the roundoff; and

- ii) if iterative techniques are employed at the local level, say the conjugate gradient, the convergence process is significantly speeded up; this is a natural consequence of the fact that such schemes tend to converge more rapidly in small neighborhoods of the solution.

Based on the foregoing, the main thrust of the benchmarking is several fold, i.e.:

- 1) to establish the feasibility of the hierarchical scheme to handle large scale simulations in a parallel setting;
- (2) to compare the direct, mixed and fully iterative versions of the strategy; and
- 3) to compare the parallel and traditional nonparallel solution algorithms.

Note, the main purpose of the comparisons between the various parallel schemes and the traditional nonparallel approach is to ascertain whether any improvements in convergence rate, stability storage requirements and run times are obtained.

For the present purposes, convergence is ascertain by employing the normed ratio test. In terms of the hierarchical partitioning of  $\tilde{T}$  we recall that

$$\tilde{T} = \begin{bmatrix} T_I \\ T_{DB} \\ T_{MB} \end{bmatrix} \quad (5.5)$$



where  $T_{\sim I}$  denotes all the internal points of the substructures and  $T_{\sim DB}$  and  $T_{\sim MB}$  correspond to the points on the double and multiple boundarys respectively.

The ratio test takes the form

$$\frac{\| T_{\sim n+1} - T_{\sim n} \|_2}{\| T_{\sim n+1} \|_2} < \text{TOL.} \quad (5.6)$$

To quantify which of the various substructural partitions are encountering convergence difficulties, local checks can be undertaken. This is achieved through the use of the expression

$$\frac{\| T_{\sim I_{n+1}}^\ell - T_{\sim I_n}^\ell \|_2}{\| T_{\sim I_{n+1}}^\ell \|_2} < \text{TOL} \quad (5.7)$$

such that  $\ell \in [1, L]$  where  $L$  denotes the number of substructural partitions.

Based on the foregoing, Tables 5.1 and 5.2 illustrate various aspects of the convergence requirements of the mixed and purely iterative schemes. For instance, considering the case of local direct calculations, Table 5.1 illustrates the iterative requirements. As can be seen, the requirements remain

essentially stable inspite of the rather dramatic increases in problem size. For the four subdomain problems considered in the parallel mode, the overall running time was essentially 1/3 that of the sequential version of the partitioning. The improvement was a direct outgrowth of the parallel calculation of the local inverses. While in theory, an even better improvement should have been afforded by the parallelism, the recurrent backsubstitution and overall overhead due to data flow increased the time requirements.

Note, the locally iterative version of the partitioned methodology represents a significantly smaller storage burden over the purely direct approach. In particular, for a variable property, problem defined by an  $(n,n)$  square region, decomposition into  $(m,m)$  partitions reduces the storage requirements by a factor of  $\theta(1/m)$  such that the total is proportional to the ratio  $\theta(n^3/m)$ . In the case of uniform properties, the reduction is proportional to  $\theta(1/m^3)$  where in the total storage is  $\theta(n^3/m^3)$ . For the four partition benchmark problem just considered, the storage needs are essentially 1/8 that of the straight global approach. This significant reduction enabled the running of even the largest problem (250,000) in the core of the 8 megaword CRAY. Such storage efficiencies enable the maximized usage of the highspeed core. As is well known, once secondary storage is required, i.e., hard and solid state disks, the resulting out of core solution, is generally very expensive.

Table 5.2 illustrates the convergence requirements of the totally iterative scheme. Seen graphically in fig. 5.1, it follows quite surprisingly that for the given benchmark problem, that proportionately less iterative burden is encountered as problem size was increased. In particular, as can be seen, the problem size to iteration count is a softening curve. Beyond the improved numerical efficiency, the parallel methodology enabled the solution of problems whose size yield either unstable or significantly less efficient iterative processes. In this context, the use of hierarchical parallelism:

- 1) significantly improves the stability of the iterative approach;
- 2) reduced computational time due to the capability of the procedure to perform simultaneous calculations, i.e. the numerical effort for the four partition test case was essentially 1/3 that of the full formulation per iteration; and
- 3) improved iterative efficiency.

Note, the improved iterative efficiency is a direct outgrowth of the partition size reduction introduced by the use of parallelism. Additionally, it should be noted that if the proper ratio between internal substructural and boundary mesh points is obtained, the overhead associated with the data flow between levels can be significantly minimized.

In comparing the mixed and completely iterative schemes, it should be noted that

- 1) for very large scale problems, the mixed method is somewhat sensitive to roundoff error; this is an outgrowth of the direct inversion employed at the various local level substructural partitions; for large scale partitions the number of calculations performed during the forward elimination and backsubstitution phases of calculation tax the place accuracy of even the CRAY system;
- 2) the direct inversion phase of inversion is inherently more storage intensive than the iterative scheme;
- 3) for spectrally ill-conditioned partitions, the direct method can prototypically bypass problems of iterative efficiency and stability;
- 4) due to roundoff errors, the mixed methods are somewhat more sensitive to inaccuracies in the starting guess defines along substructural boundaries. This follows directly from the fact that roundoff initially generated in the forward step and continuously in the backward phase act to disturb the overall convergence process; such behavior is clearly demonstrated by the fact that modest changes in the initial guess accuracy (10%) causes major increases in the iteration count of the mixed method wherein the global level is iterative while the local is direct.

## 6. CONCLUDING REMARKS

Parts I and II of this series of papers has developed a hierarchically parallel modelling methodology and associated solution procedure. Overall, the procedure enables a logical and efficient use of parallel computer environments. The scheme provides a wide variety of solution procedures including direct, mixed, direct-iterative and completely iterative type procedures. Note, due to the local partitioning afforded by the parallelism, the overall stability and efficiency of the iterative phases of computation are greatly enhanced over the classical full single level modelling approach. As has been seen in this part of the series, such behavior has been both formally and empirically verified.

Note, due to the manner of organizing the scheme, it can be directly incorporated in conjunction with a wide variety of general purpose FD codes for example CINDA [12]. Such an undertaking would reduce a code like CINDA to a subroutine residing at a given parallel processor. Here it would generate the appropriate governing FD equations for the given substructure. These would then be locally solved either iteratively or if ill-conditioned directly. The upper solution would be established via a global level which performs the task of overall problem assembly, direct or iterative solution as well as data transfer between iterations and among the various substructural components. In such an undertaking, the main task would be to develop the upper level code.

Note in future activities, the current approach is being generalized for use in finite element type applications. Work is also ongoing to adapt the methodology to use in nonlinear problems.

## 7. REFERENCES

1. Varga, R. S., Matrix Iterative Analysis, Prentice-Hall, New Jersey, 1962.
2. Young, D. M., Iterative Solution of Large Linear Systems, Academic Press, 1971.
3. Berman, A. and Plemmons, R. J., Nonegative Matrices in the Mathematical Sciences, Academic Press, New York, 1979.
4. Concus, P., Golub, G. H., and O'Leary, D. P., "A generalized conjugate gradient method for the numerical solution of the elliptic partial differential equations", Sparse Matrix Computations, J. R. Bunch and D. J. Rose ed., Academic Press, New York, 1979.
5. Krishna, L. B., "An incomplete factorization iterative technique for elliptic equations", submitted for publication the Bulletin of the Australian Mathematical Society.

6. Reid, J. K., "On the method of conjugate gradient for the solution of large sparse systems of linear equations, Large Sparse Sets of Linear Equations, J. K. Reid, ed., Academic Press, New York, 1971.
7. Manteuffel, T. A., "An incomplete factorization technique for positive definite linear systems", Math Comp. 150, 473-497, 1980.
8. Meijerick, J. A. and van der Vorst, H. A., "An iterative solution for linear systems of which the coefficient matrix is a symmetric M-matrix, Math. Comp. 31, 148-162, 1977.
9. Axelsson, O., Brinkkemper, S., Il'in V. P., "On some versions of incomplete block-matrix factorization iterative methods", Lin. Algebra Appl. 58, 3-15, 1984.
10. Varga, R. S., "Factorization and normalized iterative methods", Boundary Problems in Differential Equations, R. E. Longer ed., Univ. of Wisconsin Press, Madison, 1960.
11. Ozisik, M. N., Heat Conduction, Wiley, New York, 1980.
12. Gaski, J. D., and Lewis, D. R., Chrysler Improved Numerical Differencing Analyzer: CINDA3G, Chrysler Corp. Space Div., Rept. TN-AP-67-287, Oct. 67.

Table 5.1 - Iteration of Requirments of Mixed Direct-Iterative Hierarchical Scheme.

<u>Problem Size</u>	<u>Tolerance</u>	<u>No. of Iterations</u>
h = .005 39601 DOF	$10^{-6}$	113
h = .003125 101761 DOF	$10^{-6}$	118
h = .00270 136161 DOF	$10^{-6}$	118

DOF - Degrees of Freedom

Table 5.2 - Requirements of Purely Iterative Hierarchical Scheme

<u>Problem Size</u>	<u>Tolerance</u>	<u>No. of Iterations</u>
h = .01 9801 DOF	$10^{-5}$	31
h = .005 39601 DOF	$10^{-5}$	65
h = .003125 101761 DOF	$10^{-5}$	92
h = .0025 159201 DOF	$10^{-5}$	117
h = .002 249001 DOF	$10^{-5}$	150



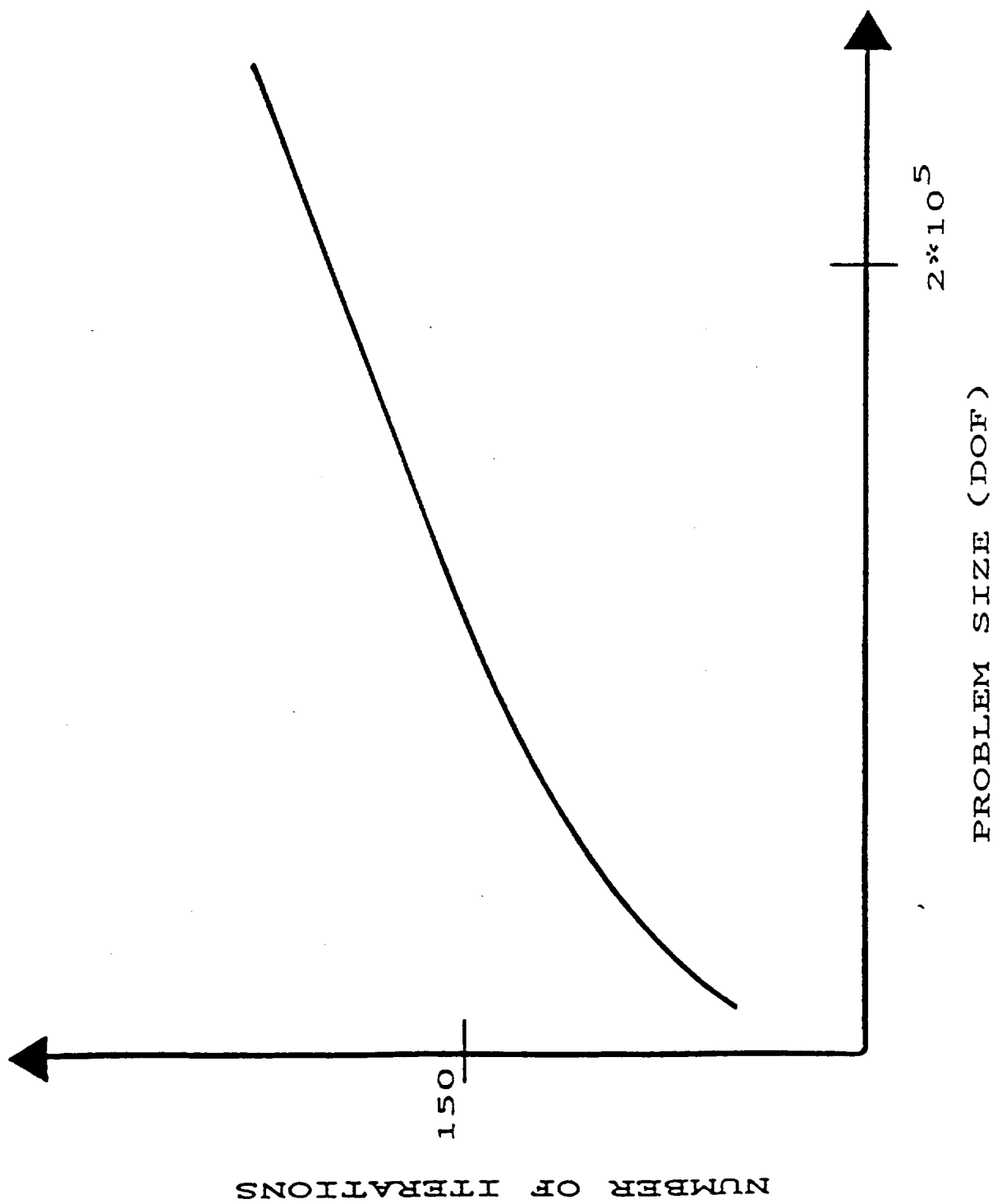


Figure 3.1 - The Number of Iterations Increases Linearly as the Problem Size Becomes Larger.

<b>REPORT DOCUMENTATION PAGE</b>			Form Approved OMB No. 0704-0188	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.				
1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE November 1997	3. REPORT TYPE AND DATES COVERED Final Contractor Report		
4. TITLE AND SUBTITLE  Hierarchical Parallelism in Finite Difference Analysis of Heat Conduction			5. FUNDING NUMBERS  WU-523-22-13-00 NAG3-644	
6. AUTHOR(S)  Joseph Padovan, Lala Krishna, and Douglas Gute				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)  The University of Akron Akron, Ohio			8. PERFORMING ORGANIZATION REPORT NUMBER  E-10963	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)  National Aeronautics and Space Administration Lewis Research Center Cleveland, Ohio 44135-3191			10. SPONSORING/MONITORING AGENCY REPORT NUMBER  NASA CR-97-206226	
11. SUPPLEMENTARY NOTES  Project Manager, Christos C. Chamis, Research and Technology Directorate, NASA Lewis Research Center, organization code, 5000 (216) 433-3252.				
12a. DISTRIBUTION/AVAILABILITY STATEMENT  Unclassified - Unlimited Subject Category: 39  This publication is available from the NASA Center for AeroSpace Information, (301) 621-0390.			12b. DISTRIBUTION CODE  Distribution: Nonstandard	
13. ABSTRACT (Maximum 200 words)  Based on the concept of hierarchical parallelism, this research effort resulted in highly efficient parallel solution strategies for very large scale heat conduction problems. Overall, the method of hierarchical parallelism involves the partitioning of thermal models into several substructured levels wherein an optimal balance into various associated bandwidths is achieved. The details are described in this report. Overall, the report is organized into two parts. Part I describes the parallel modelling methodology and associated multilevel direct, iterative and mixed solution schemes. Part II establishes both the formal and computational properties of the scheme.				
14. SUBJECT TERMS  Solution strategies; Computer machines; Iterative procedures; Convergence speed; Spectral properties; Large-scale problems; Solution algorithms			15. NUMBER OF PAGES 93	
			16. PRICE CODE A05	
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT	