RESEARCH REPORT NAG-1-544

THE DESIGN AND IMPLEMENTATION OF COST-EFFECTIVE ALGORITHMS FOR DIRECT SOLUTION OF BANDED LINEAR SYSTEMS ON THE VECTOR PROCESSOR SYSTEM 32 SUPERCOMPUTER

By
Augustine S. Samba
Department of Mathematics, Hampton University

January 1985 — December 1985

For submission to the
National Aeronautics and Space Administration

RESEARCH REPORT NAG-1-544

THE DESIGN AND IMPLEMENTATION OF COST-EFFECTIVE ALGORITHMS FOR DIRECT SOLUTION OF BANDED LINEAR SYSTEMS ON THE VECTOR PROCESSOR SYSTEM_32 SUPERCOMPUTER

By

Augustine S. Samba

Department of Mathematics, Hampton University

January 1985 - December 1985

For submission to the

National Aeronautics and Space Administration

Table of Contents

		Page
1.	INTRODUCTION	2
	1.1 An Overview of the Vector Processor System_32 Supercomputer	2
	1.2 Using the Vector Processor System_32 Supercomputer	3
2.	The Vector Cyclic Reduction Method	5
	2.1 The Vector Cyclic Reduction Algorithm	9
3.	Numerical Example	11
	3.1 Speed-Up for the Vector Cyclic Reduction Algorithm	12
4.	The Customized Reduction of Augmented Triangles (CRAT) Method	13
	4.1 Motivations for CRAT	13
	4.2 The CRAT Methodology	15
	4.2.1 Customization of the Banded Algorithm	15
	4.2.2 Problem Decomposition	16
	4.2.3 Reduction of Augmented Triangles	18
	4.4.4 Elimination Step	23
5.	Implementation of the CRAT Algorithm on the VPS_32 Supercomputer	27
	5.1 Data Organization	27
	5.2 Reduction Stage	28
6.	Future Developments	35
7.	Comments	36
8.	Acknowledgements	36
9.	References	36
10.	Appendix	37

The Design and Implementation of Cost-effective algorithms for direct solution of banded linear systems on the Vector Processor System_32 supercomputer, located at NASA Langley

AUGUSTINE S. SAMBA

Department of Mathematics, Hampton University

ABSTRACT

The problem of solving banded linear systems by direct (non-iterative) techniques on the Vector Processor System (VPS)_32 supercomputer is considered. This report describes two very efficient direct methods for solving banded linear systems on the VPS_32.

The Vector Cyclic Reduction (VCR) algorithm is discussed in detail. A Performance of the VCR on a three parameter model problem is also illustrated. The VCR is an adaptation of the conventional point cyclic reduction algorithm.

The second direct method is the "Customized Reduction of Augmented Triangles (CRAT)". CRAT is a new method created and custom designed for the VPS_32 by this principal investigator. CRAT has the dominant characteristics of an efficient VPS_32 algorithm. CRAT is tailored to the pipeline architecture of the VPS_32 and as a consequence the algorithm is implicitly vectorizable.

Contained in this report is a proposal for an additional one year funding in order to undertake further investigations of banded techniques on the Vector Processor System_32 supercomputer.

1.0 INTRODUCTION

1.1 An overview of the Vector Processor System (VPS) 32 supercomputer

The VPS_32 is a high speed vector computer, with the following special characteristics:

(i) It is a Single Instruction stream Multiple Data stream (SIMD) type of computer system. An SIMD system has only one stream of instructions in execution at anytime, but each instruction may affect many different data.

(ii) It is a pipeline computer. The basic idea behind pipeline computers is essentially that of an assembly line: if the same arithmetic operations is going to be repeated many times, throughput can be increased by dividing the operation into a sequence of sub-tasks and maintaining a flow of operand pairs in various stages of completion.

1.2 Using the Vector Processor System 32 supercomputer

Data items of the VPS_32 exist mainly in scalar and vector modes. A matrix may be viewed as a set of column/row or diagonal vectors. The VPS_32 works efficiently in vector mode. The time (T) required to perform an n-element vector operation by the VPS_32 is given by

$$T = S + n/N \qquad (1.2.1)$$

where

T represents the time in minor cycles

S the vector startup time and

N the number of results/cycle emerging from the pipeline.

Equation (1.2.1) demonstrates that by keeping the startup time to a minimum, the overall time needed to perform a given operation on the VPS_32 will depend crutially on the size of n, which also gives a measure of the number of computations. Consequently, algoritms designed for the VPS will generally perform efficiently if the operations involve long vectors as opposed to short vectors. 'Long vectors' are deliverables of efficient data organization (management).

A salient feature about data manipulation on the VPS is that operations on logical items are relatively faster than equivalent operations on other types of compatible items. In practice, efficient algorithms for the VPS_32 supercomputer are uniquely influenced by its pipeline architecture.

Our major interest in the VPS_32 supercomputer is as vehicle for solving Banded linear systems. Two very efficient methods for solving Banded linear systems are described in this report.

Section (2.0) describes the Vector Cyclic Reduction (VCR) algorithm. The VCR is an adaptation of the Conventional Point Cyclic Reduction algorithm [1,2].

The Customized Reduction of Augmented Triangles (CRAT) is a new algorithm created and custom designed for the VPS_32, by this principle investigator. The CRAT algorithm is tailored to the pipeline architecture of the VPS_32 and as a consequence this algorithm is implicitly vectorizable. CRAT is unique; it incorporates the dominant characteristics of a very efficient vector algorithm for the VPS_32 supercomputer. CRAT is discussed in section (4.0).

Section (7.0) outlines a proposal for an additional one year funding in order to undertake further investigation of banded techniques on the VPS_32 supercomputer.

2.0 "Vector Cyclic Reduction" (VCR) METHOD

Recall the banded linear system of equations is defined as

$$A\underline{x} = \underline{y} \qquad 2.1$$

where $a_{i,j} = 0.0$ if $|i - j| > m$; $1 \leq i,j \leq n$. m represents the semi-bandwidth.

The technique here is to successively decouple the banded system into a sequence of smaller systems. The number of steps required to complete the decoupling is $O(\log_2 n)$.

To illustrate this procedure let us identify the entries of the coefficient matrix A, in the following manner,

$$A = \begin{bmatrix} a_1 & b_1 & & & & & \\ c_2 & a_2 & b_2 & & & & \\ & c_3 & a_3 & b_3 & & & \\ & & & & & & \\ & & & & & & \\ & & & & & b_{n-1} & \\ & & & & c_n & a_n & \end{bmatrix}$$

where $c_1 = 0 = b_n$ 2.2

Consider now any three successive equations of the linear system (2.1). The $(i-1)^{th}$, $(i)^{th}$ and $(i+1)^{th}$ equations are respectively.

$$(i-1)\text{th:} \quad c_{i-1}\underline{x}_{i-2} + a_{i-1}\underline{x}_{i-1} + b_{i-1}\underline{x}_{i} = \underline{y}_{i-1}$$

$$(i)\text{th} : \quad c_{i}\underline{x}_{i-1} + a_{i}\underline{x}_{i} + b_{i}\underline{x}_{i+1} = \underline{y}_{i}$$

and

$$(i+1)\text{th:}\quad c_{i+1}\,\underline{x}_i + a_{i+1}\,\underline{x}_{i+1} + b_{i+1}\,\underline{x}_{i+2} = \underline{y}_{i+1} \quad .$$

We can eliminate $\underline{x}_{i-1}$ and $\underline{x}_{i+1}$ from the $(i)^{th}$ equation to yield an equation for $\underline{x}_{i-2}$, $\underline{x}_i$, and $\underline{x}_{i+2}$ as follows.

Normalizing the $(i)^{th}$ equation with respect to the coefficient of $\underline{x}_i$ produces:

$$(i)\text{th}: \quad c_i^{(1)}\,\underline{x}_{i-1} + \underline{x}_i + b_i^{(1)}\,\underline{x}_{i+1} = \underline{y}_i^{(1)} \qquad 2.3$$

where

$$\begin{aligned} c_i^{(1)} &= a_i^{-1}\, c_i \\ b_i^{(1)} &= a_i^{-1}\, b_i \qquad 2.4 \\ \underline{y}_i^{(1)} &= a_i^{-1}\, \underline{y}_i \\ a_i^{(1)} &= a_i \end{aligned}$$

The corresponding $(i-1)^{th}$ and $(i+1)^{th}$ equations are respectively

$$(i-1)\text{th:}\quad c_{i-1}^{(1)}\,\underline{x}_{i-2} + \underline{x}_{i-1} + b_{i-1}^{(1)}\,\underline{x}_{i-1} = \underline{y}_{i-1}^{(1)} \qquad 2.5$$

and

$$(i+1)\text{th}\quad c_{i+1}^{(1)}\,\underline{x}_i + \underline{x}_{i+1} + b_{i+1}^{(1)}\,\underline{x}_{i+2} = \underline{y}_{i+1}^{(1)} \qquad 2.6$$

where the corresponding coefficients and the Right Hand Sides (RHS) respectively satisfy the corresponding relations in (2.4).

Substituting now, in (2.3) for $\underline{x}_{i-1}$ and $\underline{x}_{i+1}$ from (2.5) and (2.6) leads to

$$(i+1)\text{th:}\quad c_i^{(2)}\,\underline{x}_{i-2} + \underline{x}_i + b_i^{(2)}\,\underline{x}_{i+2} = \underline{y}_i^{(2)} \qquad 2.7$$

where

$$a_i^{(2)} = 1 - c_i^{(1)}\, b_{i-1}^{(1)} - b_i^{(1)}\, c_{i+1}^{(1)}$$

$$c_i^{(2)} = -(a_i^{(2)})^{-1}\,(c_i^{(1)}\; c_{i-1}^{(1)})$$

$$b_i^{(2)} = -(a_i^{(2)})^{-1}\,(b_i^{(1)}\; b_{i+1}^{(1)})$$

and

$$\underline{y}_i = (a_i^{(2)})^{-1}\,(\underline{y}_i^{(1)} - c_i^{(1)}\,\underline{y}_{i-1}^{(1)} - b_i^{(1)}\,\underline{y}_{i+1})$$

Substituting for $\underline{x}_{i-2}$ and $\underline{x}_{i+2}$ in (2.7) from the corresponding $(i-2)^{th}$ and $(i+2)^{th}$ equations respectively, produces a relation in $\underline{x}_{i-4}$, $\underline{x}_i$ and $\underline{x}_{i+4}$ for the $(i)^{th}$ equation.

The elimination proceeds in this manner.

In general, the k^{th} $(k>2)$ step is related to the $(k-1)^{th}$ step by

$$a_i^{(k)} = 1 - c_i^{(k-1)}\, b_{i-1}^{(k-1)} - b_i^{(k-1)}\, c_{i+1}^{(k-1)})$$

$$c_i^{(k)} = -(a_i^{(k)})^{-1}\,(c_i^{(k-1)}\; c_{i-1}^{(k-1)})$$

$$b_i^{(k)} = -\left(a_i^{(k)}\right)^{-1} \left(b_i^{(k-1)} \quad b_{i+1}^{(k-1)}\right)$$

and

$$\underline{y}_i^{(k)} = \left(a_i^{(k)}\right)^{-1} \left(\underline{y}_i^{(k-1)} - c_i^{(k-1)}\, \underline{y}_{i-1}^{(k-1)} - b_i^{(k-1)}\, \underline{y}_{i+1}^{(k-1)}\right)$$

where

$$i = 2^{k-2}$$

The matrix $W^{(k)}$ at each step, k, comprises of three diagonals. The distance of the outer diagonals from the principal diagonal doubles at each step.

Following the final n^{th} ($n = \log_2 N$) step, the two outer diagonals skip out of the coefficient matrix $A^{(n)}$, leaving only the main diagonal.

2.1 The Vector Cyclic Reduction (VCR) Algorithm

Let a_i, b_i and c_i, (i = 1, 2, ...N) represent the block matrices on the main, sub and super-diagonals respectively of the matrix A. Suppose the $\underline{y}_i$, (i = 1, 2, ... N) represent the components of the right band vector $\underline{y}$.

The VCR algorithm then proceeds in the following manner.

Step 1. INITIALIZATION:

Set $c_i \leftarrow 0$

" $b_i \leftarrow 0$

" $k \leftarrow 1$

Step 2. COMPUTATIONS:

For J = 1 STEP 1 UNTIL ($Log_2 N$) do

BEGIN

Compute	c_i	$\leftarrow$	$a_i^{-1} c_i$	$1 \leq i \leq n$
"	b_i	$\leftarrow$	$a_i^{-1} b_i$	"
"	$\underline{y}_i$	$\leftarrow$	$a_i^{-1} \underline{y}_i$	"
"	a_i	$\leftarrow$	$1.0 - c_i \underline{b}_{i-k} - b_i c_{i+k}$	"
"	$\underline{y}_i$	$\leftarrow$	$\underline{y}_i - c_i \underline{y}_{i-k} - b_i \underline{y}_{i+k}$	"
"	c_i	$\leftarrow$	$- c_i c_{i-k}$	"

Compute $b_i \leftarrow -b_i\, b_{i+k}$ $1 \le i \le n$

" $k \leftarrow k + k$ "

END

Step 3. Now Obtain the Solution

Compute $x_i = a_i^{-1} y_i$ "

3.0 Numerical Example

The following banded linear system is a test problem designed to validate the algorithms described in this report.

We wish to solve

$$A\underline{x} = \underline{y} \qquad (3.1)$$

where A is the three-parameter band matrix:

$$A(m;n;s) = \begin{bmatrix} 1 & S & \cdots & S & 0 & & & 0 \\ S & 1 & \cdots & & \ddots & & & \vdots \\ \vdots & \ddots & \ddots & & & \ddots & & \vdots \\ S & & & & & & & 0 \\ 0 & \cdots & & & & & & S \\ & & & & & & & \vdots \\ & & & & & & & S \\ 0 & & & 0 & S & & & 1 \end{bmatrix}$$

The matrix A has semi-bandwidth m, dimensions nxn.

The parameter S is defined for $1/10 \leq S \leq 1/3$. The RHS $\underline{y}$ is given by

$$y_i = \sum_{j=1}^{n} A(i,j)$$

Therefore (3.1) has the unique solution

$$x_i = 1, \quad (1 \leq i \leq n)$$

3.1 SPEED-UP for the VECTOR CYCLIC REDUCTION ALGORITHM: TIMINGS for the VCR ROUTINES

Table (3.1) gives a summary of the timings for the initial tridiagonal VCR Subroutine: PRELUDE. The PRELUDE subroutine is described in figure A3 of Appendix-A.

Two very efficient subroutines VCRI and VCRII are illustrated in figures AI and AII respectively of Appendix-A. Both VCRI and VCRII subroutines yield faster times than the PRELUDE subroutine. This is because both VCRI and VCRII subroutines employ better and more efficient data mapping schemes than the PRELUDE subroutine.

The major difference between the VCRI and VCRII subroutines is that VCRII incorporates Cyber-200 generic vector functions as a means of improving the vector processing power. The VCRI subroutine, on the other hand, seeks to generate and access sets of items (scalar) in contiguous core locations.

System Size (n x n)	Bandwidth (m)	Computing Timings (second)
64 x 64	2	.000663
1024 x 1024	2	.00278
4096 x 4096	2	.00313

Table (3.1). Computing timings for the initial VCR subroutine:PRELUDE

4.0 The Customized Reduction of Augmented Triangles (CRAT) Method

4.1 Motivations for CRAT

The Vector Cyclic Reduction (VCR) algorithm has four disadvantages viewed purely as a technique for solving banded systems on the Vector Processor System_32 supercomputer:

1) The partitioning of a band matrix in tri-diagonal form yields super and sub-diagonal blocks which are triangular; but the technique takes no advantage of this.
2) While it may be easy to process efficiently several RHS simultaneously, it is not possible to re-enter the routine with further righthand sides without repeating the whole reduction process.
3) The diagonal blocks are assumed non-singular throughout, and no pivoting between blocks is possible, although pivoting within blocks can be achieved.
4) Being an adaptation of the conventional serial (point cyclic reduction) algorithm, vectorization is introduced explicitly by employing,
   (a) meticulous data organization and
   (b) selecting computational tools that exploit the data structure. The processing power of these tools is given very little or no consideration.

   I have therefore designed an altenative scheme which is custom made for the VPS_32. The CRAT's prespective on points 1) to 4) above is as follows:

   Point 1: CRAT takes full advantage of the form of the band matrix

Point 2: Not yet studied

Point 3: CRAT works efficiently with singular diagonal blocks

Point 4: CRAT is designed to exploit the structure of the band system and the pipeline architecture of the VPS_32. Therefore vectorization is implicit.

A complete description of the CRAT method now follows.

4.2 The CRAT Methodology

The stages in the CRAT method are as follows:

4.2.1 Customization of the Banded Algorithm

The primary goal of customizing the banded algorithm is to efficiently exploit the pipeline feature of the Vector Processor System (VPS)_32 supercomputer. In order to achieve this goal it is necessary to tailor the solution methodology for the Banded System to the architecture of the Vector Processor System_32 Supercomputer. The initial task involves transforming the Banded System to a more suitable (tractable) problem whose solution could be easily and cheaply obtained by utilizing the optimal vector interpretive schemes of the VPS_32 supercomputer.

Recall the nxn banded linear system for the N^{th} order vector $\underline{u}$ is given by

$$W_{i,j}u_j = b_i \quad ; \quad i,j = 1,2, \ldots N \tag{4.1}$$

where $w_{i,j} = 0$ iff $|i-j| > m$; m represents the semi-bandwidth.

We seek to transform (4.1) to an over-determined problem by means of the following technique. Define

i) an $(n+m)^{th}$ order vector $\underline{v}$ in terms of $\underline{u}$ by

$$v_{j+m} = \begin{cases} u_j \text{ , iff } 1 \leq j \leq N \\ 0 \text{ , otherwise} \end{cases} \tag{4.2}$$

and (ii) an Nx(N+2m) matrix A in terms of W by

$$A_{i,j} = \begin{cases} \delta_{i,j}\alpha_j \ , & \text{iff} \quad 1 \le j \le m \ , \quad 1 \le i \le N \\ W_{i,j-m} \ , & \text{iff} \quad m+1 \le j \le N+m \ , \quad 1 \le i \le N \\ \delta_{j-i-m,\, j-N}\ \alpha_{j-N} \ , & \text{iff} \quad N+m < j \le N+2m \ , \quad 1 \le i \le N \end{cases} \tag{4.3}$$

where $\alpha_j \neq 0$ for $1 \le j \le 2m$. $\delta_{i,j}$ is the Kronecker delta, defined as

$$\delta_{i,j} = \begin{cases} 1 \ , & \text{iff} \quad i = j \\ 0 \ , & \text{otherwise} \end{cases}$$

By employing (4.2) and (4.3) the Banded System transforms to the N x (N+2m) over-determined problem,

$$A_{i,j}\ v_j = b_j \tag{4.4}$$

where $1 \le i \le N$ and $1 \le j \le N+2m$.

In order to solve (4.4) efficiently on the VPS_32 supercomputer, it is necessary to reformulate the over-determined problem in the following way.

4.2.2 Problem Decomposition

The coefficient matrix A of (4.4) is partitioned into 2n conformable triangular matrices A_i , B_i, i=1, 2, ... n (= N/2m).

$$A = \begin{bmatrix} A_1 & B_1 & & & & \\ & A_2 & B_2 & & & \\ & & \ddots & \ddots & & \\ & & & \ddots & \ddots & \\ & & & & B_{n-1} & \\ & & & & A_n & B_n \end{bmatrix} \qquad 4.4.1$$

The matrices A_i, $1 \leq i \leq n$ are upper triangular with dimensions 2m x 2m. The B_i, $1 \leq i \leq n$ are 2mx2m lower triangular matrices.

The RHS vector $\underline{b}$ is partitioned into n $(2m)^{th}$ order subvectors $\underline{y}_i$, i=1,2,...,n by the relation:

$$\underline{y}_i^t = [b_j,\ j = 1+2\ (i-1)\ m,\ \ldots,\ 2im], \qquad 1 \leq j \leq n \qquad (4.5)$$

Note the inverse relation

$$\underline{b}^t = [y_i,\ \underline{y}_2,\ \ldots,\ \underline{y}_n]\ . \qquad (4.6)$$

Then if $\underline{v}$ is partitioned into (n+1) $(2m)^{th}$ order subvectors $\underline{x}_i$, i=1,2, ..., n by the relation

$$\underline{x}_i^t = [v_j,\ j = 1+2(i-1)m,\ \ldots,\ 2\ im], \qquad 1 \leq j \leq n\ , \qquad (4.7)$$

with inverse relation

$$\underline{v}^t = [\underline{x}_1,\ \underline{x}_2,\ \ldots,\ \underline{x}_{n+1}] \qquad (4.8)$$

the over-determined problem can be expressed as a vector recurrence problem:

$$A_i \underline{x}_i + B_i \underline{x}_{i+1} = \underline{y}_i, \qquad (1 \leq i \leq n) \qquad (4.9)$$

The task of designing an efficient solution methodology to (4.9), and hence the Banded System, for the VPS_32 supercomputer is the subject of the following section.

4.2.3 Reduction of Augmented Triangles

Observe that the A_i, B_i of (4.9) are upper and lower triangular matrices respectively. Without any loss of generality it will be assumed that n is integer, power 2.

With a view to exploiting the structure of the recurrence problem and the architecture of the Vector Processor System_32 supercomputer, it is instructive to pose equation (3.9) as a set of two subsystems:

Subsystem I:

$$A_i \underline{x}_i + B_i \underline{x}_{i+1} = \underline{y}_i, \qquad 1 \leq i \leq \ell \ (=n/2) \qquad (4.10)$$

and Subsystem II:

$$A_i \underline{x}_i + B_i \underline{x}_{i+1} = \underline{y}_i, \qquad 1+\ell \leq i \leq n \qquad (4.11)$$

The single Instruction stream Multiple Data stream capabilities coupled with the "assembly-line" processing characteristics of the VPS_32 supercomputer provide a clinical and unique means for reducing subsystems I and II simultaneously. This

reduction process is easily accomplished via a culmination of dynamic partitioning technique; and augmentations of the coefficient triangular matrices in (4.10) and (4.11). The reduction process takes only $O(\log_2 \ell)$ steps. The method in detail is as follows.

Consider Subsystem I:

Normalizing the j^{th}, $1 \leq j \leq \ell$, equation with respect to A_j and writing the normalized coefficients with zero superscript gives

$$\underline{x}_j + B_j^{(0)} \underline{x}_{j+1} = y_j^{(0)}, \qquad 1 \leq j \leq \ell \tag{4.10.1}$$

The corresponding expression for the $(j+1)^{th}$, $1 < j < \ell$, equation is

$$\underline{x}_{j+1} + B_{j+1}^{(0)} \underline{x}_{j+2} = \underline{y}_{j+1}^{(0)}, \qquad 1 \leq j \leq \ell-1 \tag{4.10.2}$$

Substituting into (4.10.1) for $\underline{x}_{j+1}$ from (4.10.2) gives

$$\underline{x}_j - B_j^{(0)} B_{j+1}^{(0)} \underline{x}_{j+2} = \underline{y}_j^{(0)} - B_j^{(0)} \underline{y}_{j+1}^{(0)} \tag{4.10.3}$$

for the j^{th}, $(1 \leq j \leq \ell-1)$, equation.

The corresponding $(J+2)^{th}$ equation is given by

$$\underline{x}_{j+2} - B_{j+2}^{(0)} B_{j+3}^{(0)} \underline{x}_{j+4} = \underline{y}_{j+2}^{(0)} - B_{j+2}^{(0)} \underline{y}_{j+3}^{(0)} \tag{4.10.4}$$

By substituting for $\underline{x}_{j+2}$ into (4.10.3) from 4.10.4) we obtain

$$\underline{x}_j + B_j^{(2)} \underline{x}_{j+4} = \underline{y}_j^{(2)} \tag{4.10.5}$$

where

$$B_j^{(0)} = A_j^{-1} B_j$$

$$B_j^{(1)} = -B_j^{(0)} B_{j+1}^{(0)}$$

$$B_j^{(2)} = -B_j^{(0)} B_{j+1}^{(0)} B_{j+2}^{(0)} B_{j+3}^{(0)} \qquad (4.10.6)$$

$$\underline{y}_j^{(0)} = A_j^{-1} \underline{y}_j$$

$$\underline{y}_j^{(1)} = \underline{y}_j^{(0)} - B_j^{(0)} \underline{y}_{j+1}^{(0)}$$

$$\underline{y}_j^{(2)} = \underline{y}_j^{(0)} - B_j^{(0)} \underline{y}_{j+1}^{(0)} + B_j^{(0)} B_{j+1}^{(0)} [\underline{y}_{j+2}^{(0)} - B_{j+2}^{(0)} \underline{y}_{j+3}^{(0)}]$$

Observe that (4.10.5) can be expanded in terms of $\underline{x}_{j+4}$ to get a relation involving $\underline{x}_j$ and $\underline{x}_{j+8}$. In general, the K^{th} stage is related to the $(K-1)^{th}$ stage by the following relations:

$$\underline{x}_j + B_j^{(K)} \underline{x}_{j+2^K} = \underline{y}_j^{(K)} \qquad (4.10.7)$$

where

$$B_j^{(K)} = -B_j^{(K-1)} B_{j+K}^{(K-1)} \qquad (4.10.8)$$

$$\underline{y}_j^{(K)} = \underline{y}_j^{(K-1)} - B_j^{(K-1)} \underline{y}_{j+K}^{(K-1)}, \qquad (1 \le j+K \le \ell) \qquad (4.10.9)$$

The reduction process for Subsystem I advances in this manner. The structure

of Subsystem I at the end of the final reduction step is given by

$$\underline{x}_j + B_j^{(p)} \underline{x}_{\ell+1} = \underline{y}_j^{(p)} \qquad (1 \le j \le \ell) \qquad (4.10.10)$$

where $p = \log_2 \ell = \log_2 n/2 = \log_2 n - 1$.

Hence, $p = \log_2 n - 1$ corresponds to the penultimate step of the CRAT algorithm.

We now focus on Subsystem II:

$$A_i \underline{x}_i + B_i \underline{x}_{i+1} = \underline{y}_i, \qquad 1+\ell \le i \le n.$$

Subsystem II is reduced by employing arithmetic operations identical to those for Subsystem I, but with opposite coefficient triangular matrices.

Normalizing the j^{th}, $[1+\ell \le j \le n]$ equation with respect to B_j and writing the normalized coefficients with zero superscripts gives

$$A_j^{(0)} \underline{x}_j + \underline{x}_{j+1} = \underline{y}_j^{(0)}, \qquad 1+\ell \le j \le n \qquad (4.11.1)$$

The corresponding expression for the $(j-1)^{th}$ equation is

$$A_{j-1}^{(0)} \underline{x}_{j-1} + \underline{x}_j = \underline{y}_{j-1}^{(0)} \qquad (4.11.2)$$

substituting for $\underline{x}_j$ in (4.11.1) from (4.11.2) gives

$$-A_j^{(0)} A_{j-1}^{(0)} \underline{x}_{j-1} + \underline{x}_{j+1} = \underline{y}_j^{(0)} - A_j^{(0)} \underline{y}_{j-1}^{(0)} \qquad (4.11.3)$$

The corresponding $(j-2)^{th}$ equation is

$$-A_{j-2}^{(0)} A_{j-3}^{(0)} \underline{x}_{j-3} + \underline{x}_{j-1} = \underline{y}_{j-2}^{(0)} - A_{j-2}^{(0)} \underline{y}_{j-3}^{(0)} \qquad (4.11.4)$$

Substituting now for $\underline{x}_{j-1}$ from (4.11.4) into (4.11.3) we obtain

$$A_j^{(2)} \underline{x}_{j-3} + \underline{x}_{j+1} = \underline{y}_j^{(2)} \tag{4.11.5}$$

where

$$A_j^{(0)} = B_j^{-1} A_j$$

$$A_j^{(1)} = -A_j^{(0)} A_{j-1}^{(0)}$$

$$A_j^{(2)} = -A_j^{(0)} A_{j-1}^{(0)} A_{j-2}^{(0)} A_{j-3}^{(0)} \tag{4.11.6}$$

and

$$\underline{y}_j^{(0)} = B_j^{-1} \underline{y}_j$$

$$\underline{y}_j^{(1)} = \underline{y}_j^{(0)} - A_j^{(0)} \underline{y}_{j-1}^{(0)}$$

$$\underline{y}_j^{(2)} = \underline{y}_j^{(0)} - A_j^{(0)} \underline{y}_{j-1}^{(0)} + A_j^{(0)} A_{j-1}^{(0)} (\underline{y}_{j-2} - A_{j-2}^{(0)} \underline{y}_{j-3}^{(0)})$$

Equation (4.11.5) is similarly expanded in terms of $\underline{x}_{j-3}$ to get a new relation in $\underline{x}_{j+1}$ and $\underline{x}_{j-7}$.

In general, the K^{th} (K>1) reduction stage for subsystem II is related to the $(K-1)^{th}$ stage by the following relations

$$A_j^{(K)} \underline{x}_{j-2^k+1} + \underline{x}_{j+1} = \underline{y}_j^{(K)} \tag{4.11.7}$$

where

$$A_j^{(K)} = -A_j^{(K-1)} A_{j-K}^{(K-1)} \qquad (4.11.8)$$

and

$$\underline{y}_j^{(k)} = \underline{y}_j^{(K-1)} - A_j^{(K-1)} \underline{y}_{j-K}^{(K-1)}, \qquad 1+\ell \leq j \leq n \qquad (4.11.9)$$

At the end of the final reduction step (p),

$$p = \log_2 \ell = \log_2 n - 1,$$

subsystem II takes the form,

$$A_j^{(p)} \underline{x}_{\ell+1} + \underline{x}_{j+1} = \underline{y}_j^{(p)} \qquad \ell+1 \leq j \leq n \qquad (4.11.10)$$

4.2.4 Final Step: Elimination of $\underline{x}_{\ell+1}$

The $\log_2 n$ step is the final step of the algorithm. Here we simply eliminate $\underline{x}_{\ell+1}$ from (4.10.10) and (4.11.10) as follows.

Combining the equations of the reduced subsystems I and II of (4.10.10) and (4.11.10) respectively leads to

$$\underline{x}_j + C_j^{(p)} \underline{x}_{\ell+1} = \underline{y}_j^{(p)}, \qquad \ell \leq j \leq n \qquad (4.12)$$

where

$$C_j^{(p)} = \begin{cases} B_j^{(p)}, & \text{iff } 1 \le j \le n/2 \\ A_j^{(p)}, & \text{iff } n/2 + 1 \le j \le n \end{cases}$$

and

$$\ell_j = \begin{cases} j, & \text{iff } 1 \le j \le n/2 \\ j+1, & \text{iff } n/2 + 1 \le j \le n \end{cases}.$$

An analysis of equation (4.12) reveals the following.

Case (I).

When J=1, (4.12) becomes

$$\underline{x}_1 + C_1^{(p)} \underline{x}_{\ell+1} = \underline{y}_1^{(p)} \tag{4.13}$$

The components of the $(2m)^{th}$ order vector $\underline{x}_1$ are easily determined from relations (4.7) and (4.2), to be

$$\underline{x}_1 = \begin{bmatrix} 0 \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ 0 \\ u_1 \\ u_2 \\ \cdot \\ \cdot \\ \cdot \\ u_m \end{bmatrix} \qquad \begin{array}{l} \text{Notice the} \\ \text{m leading zero} \end{array} \tag{4.14}$$

Substituting $\underline{x}_1$ from (4.14) into (4.13) gives

$$\sum_{J=1}^{m} \sum_{i=1}^{m} B_1^{(n)}(i,j)\ \underline{x}_{\ell+1}(j) = \underline{y}_1^{(p)}(i) \qquad (4.15)$$

for the leading m - equations.

Case (II).

When $\underline{j=n}$ (4.12) becomes

$$\underline{x}_{n+1} + C_n^{(p)}\ \underline{x}_{\ell+1} = \underline{y}_n^{(p)}(i) \qquad (4.16)$$

The components of the $(2m)^{th}$ order vector $\underline{x}_{n+1}$ are similarly determined from relations (4.7) and (4.2) to be

$$\underline{x}_{n+1} = \begin{bmatrix} u_{N+1-m} \\ u_{N+2-m} \\ \cdot \\ \cdot \\ \cdot \\ u_N \\ 0 \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ 0 \end{bmatrix} \qquad (4.17)$$

Notice the m trailing zeros

Substituting for $\underline{x}_{n+1}$ from (4.17) into (4.15) gives

$$\sum_{J=m+1}^{2m} \sum_{i=m+1}^{2m} A_n^{(p)}(i,j)\ \underline{x}_{\ell+1}(j) = \underline{y}_n^{(p)}(i) \qquad (4.18)$$

for the trailing m equations.

The components of the $(2m)^{th}$ order vector $\underline{x}_{\ell+1}$ are very easily determined from (4.16) and (4.18) through a standard elimination process.

Substituting now for $\underline{x}_{\ell+1}$ in (4.12) yields the remaining (n-1) $(2m)^{th}$ order vector $\underline{x}_j$, ($1 \leq j \leq n$ and $j \neq \ell+1$).

The unknowns u_j of the Banded System are recovered easily from x_j, ($1 \leq j \leq n$) through the inverse relations in (4.8) and (4.2): Evidently $u_j = x_{j+m+1}$, J=1,2, ...N.

5.0 Implementation of the CRAT algorithm on the VPS 32 supercomputer

The performance of the VPS_32 supercomputer, is much more sensitive to the coding technique. Moreover the choice of implementing an efficient algorithm may be different for different supercomputers. The following implementation is proposed for the VPS_32 supercomputer.

5.1 Data Organization

The VPS_32 works efficiently with long vector operands. The following mapping scheme provides the means for processing vector operands of minimum length N, the linear dimension of the Banded System.

Let A_i, B_i, i = 1,2,...n, represent the triangular block matrices described in (4.4.1).

These matrices are held in the arrays C and B as follows:

$$C[i] \longleftarrow \begin{cases} A_i\,, & 1 \le i \le \ell \ (= n/2) \\ B_i\,, & \ell+1 \le i \le n \end{cases}$$

and

$$B[i] \longleftarrow \begin{cases} B_i\,, & 1 \le i \le \ell \ (= n/2) \\ A_i\,, & \ell+1 \le i \le n \quad . \end{cases}$$

C[i] and B[i], $1 \le i \le n$, may be viewed as a set of m Nth order vectors:

$$C[i] \longleftrightarrow C(1 + (i-1)m,\ m;\ N)$$

Given the above, the normalization is easily accomplished via,

$$C[i] \longleftarrow C^{-1}[i]\ B[i]\ , \qquad (1 \le i \le n)$$

5.2 Reduction Stage

Each of the reduction phases for the CRAT algorithm can be efficiently implemented on the VPS_32 by employing dynamic mode partitioning technique.

In a dynamic mode partitioning, the partitions of the subarrays are allowed to vary over the entire parent array, so that composite subarrays behave as single data items during processing.

For example, suppose C is given by:

$$C[i] = \begin{bmatrix} A_1 \\ A_2 \\ A_3 \end{bmatrix}$$

then C may be processed as

$$\begin{bmatrix} A_1^{(1)} \\ A_2^{(1)} \end{bmatrix}$$

where $A_1^{(1)}$ is the composite subarray,

$$A_1^{(1)} = \begin{bmatrix} A_1 \\ A_2 \end{bmatrix} \text{ and } A_2^{(1)} = [A_3].$$

Alternatively, C may be processed as

$$\begin{bmatrix} A_1^{(1)} \\ A_2^{(1)} \end{bmatrix}$$

where $A_1^{(1)} = [A_1]$ and $A_2^{(1)}$ is the composite

subarray,

$$A_2^{(1)} = \begin{bmatrix} A_2 \\ A_3 \end{bmatrix}$$

The dynamic mode partitioning scheme for the CRAT algorithm is illustrated in figures 5.0 to 5.4.

Each of the Reduction phases can be carried out in a time proportional to m; the total time for the CRAT algorithm is then

$$T = k\, m\, [\log_2 n - 1] + V_m.$$

where k is a constant;

V_m is the time required to solve a 2mx2m system of linear equations on the VPS_32 supercomputer.

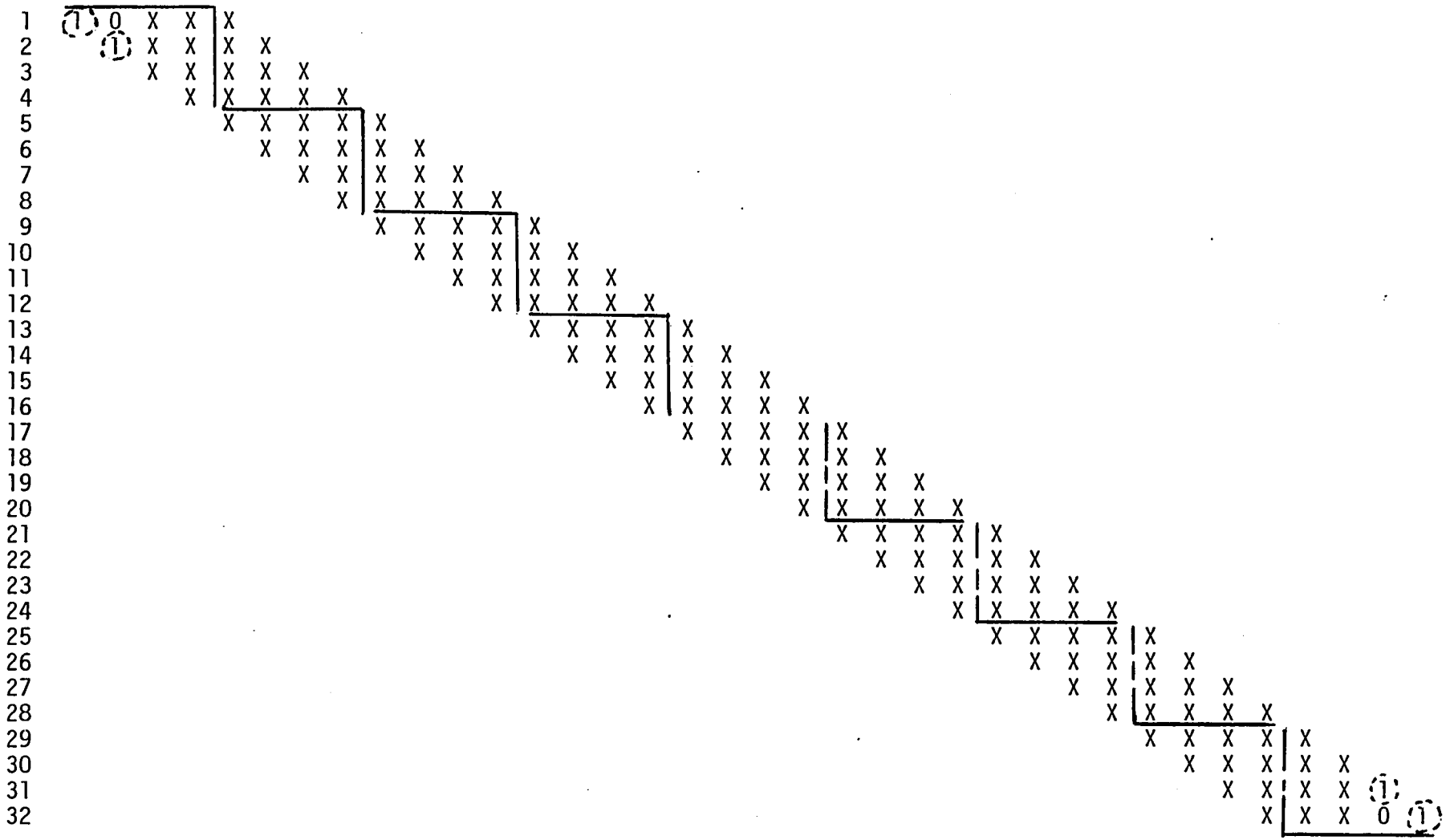

Figure 5.0 Multi-level partitioning for the CRAT algorithm when N=32, m=2. Note the padding of the extreme blocks to "triangular forms".

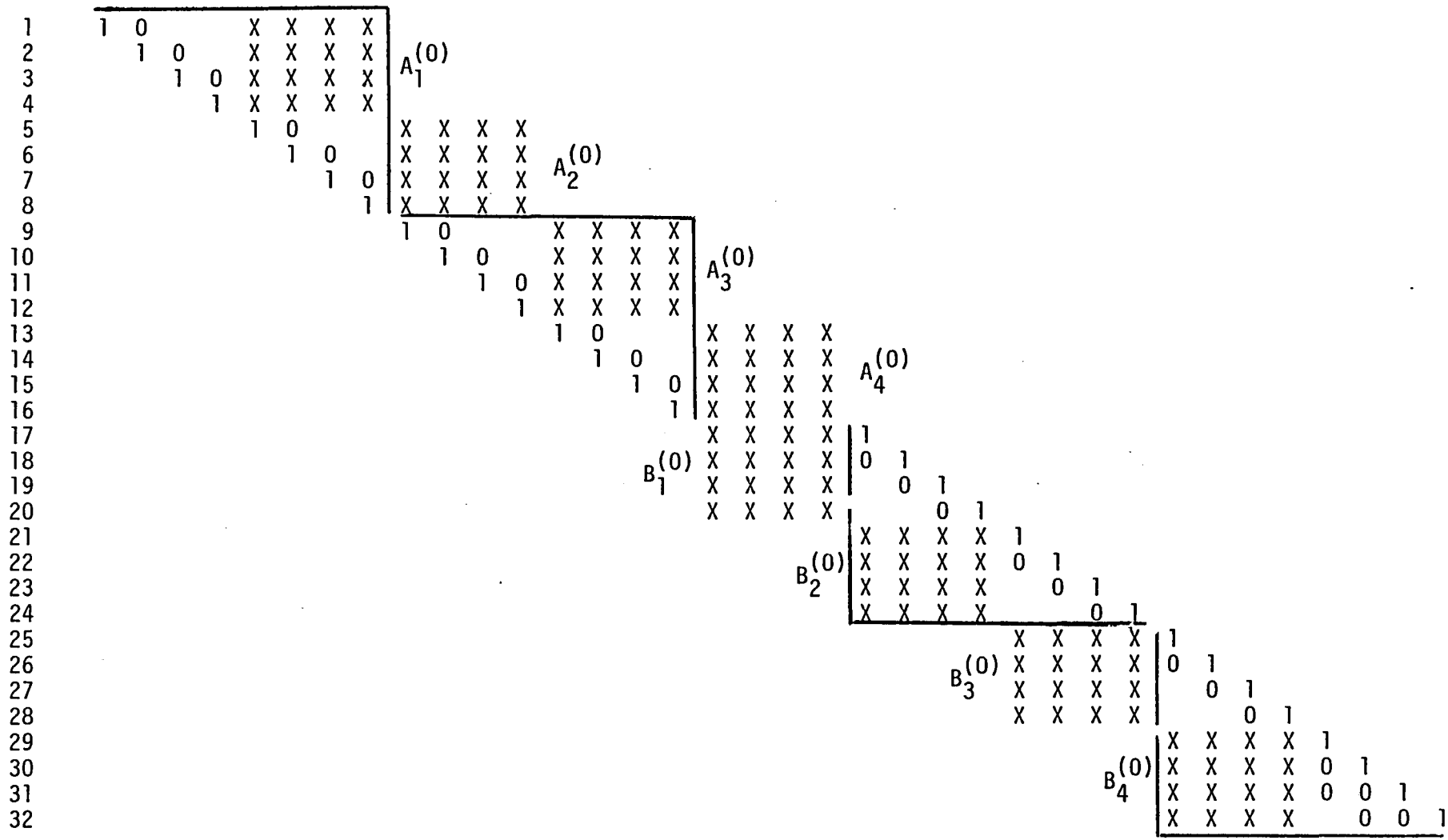

Figure 5.1 First reduction stage.

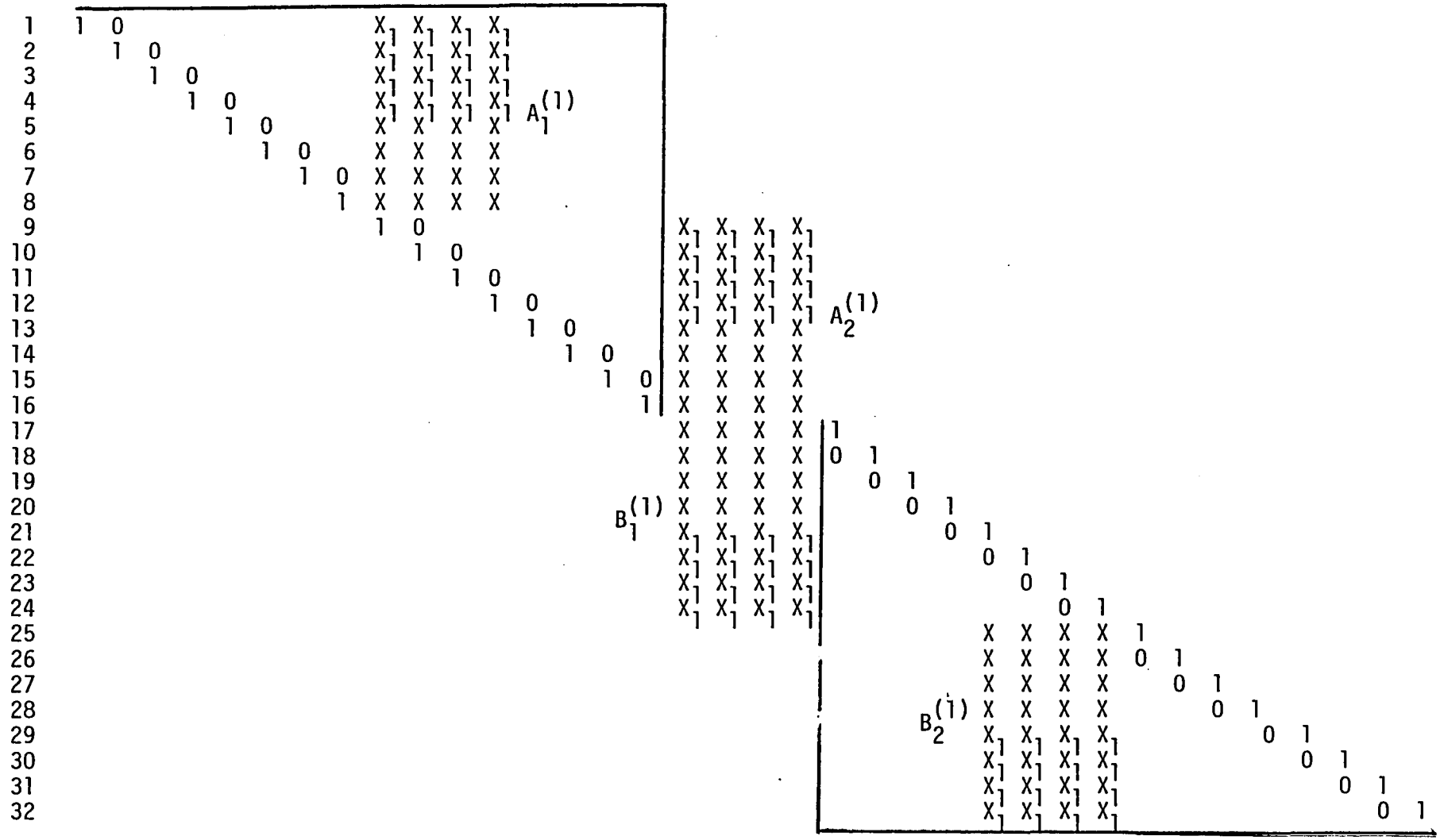

Figure 5.2 Second reduction stage.

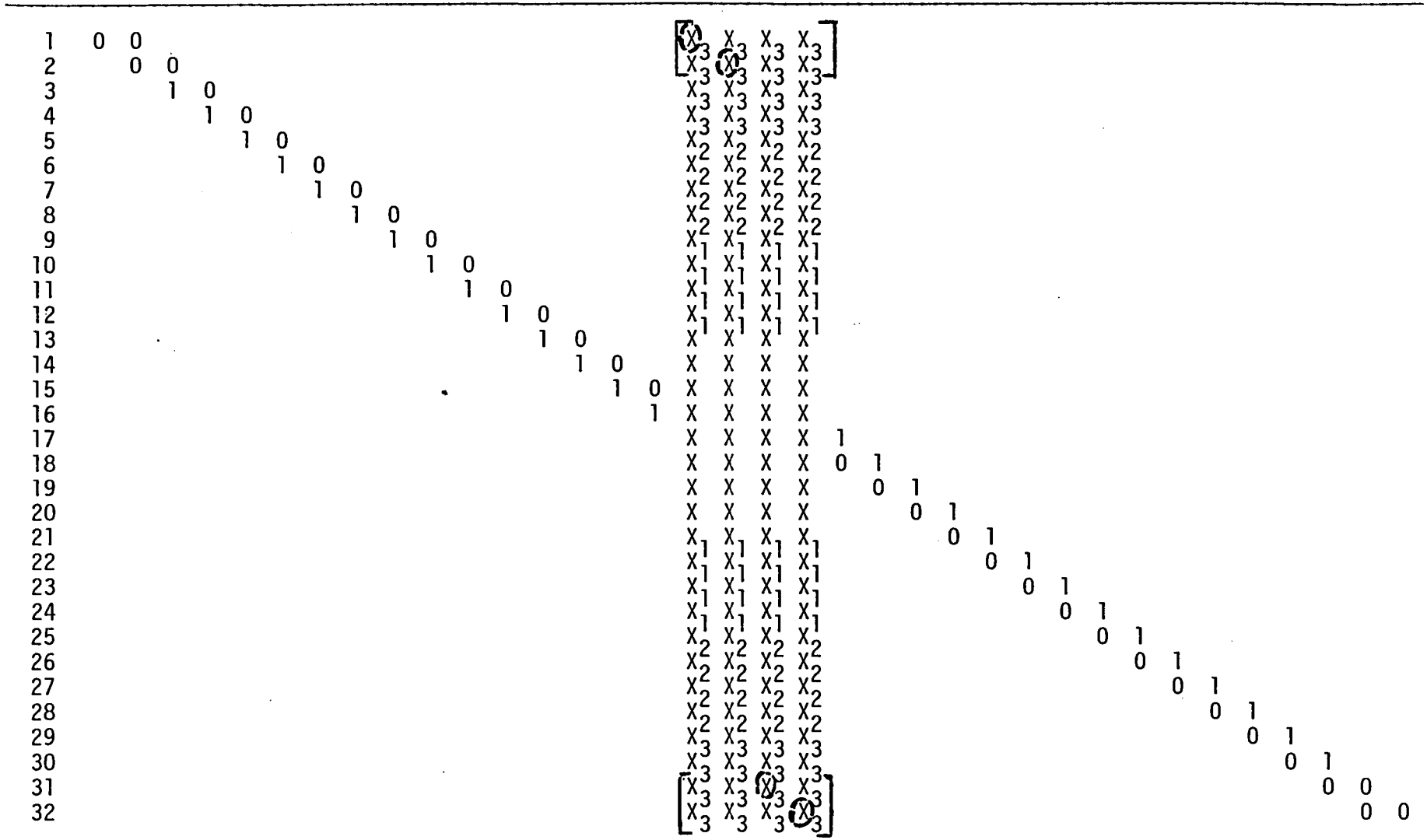

Figure 5.3 Final reduction stage.

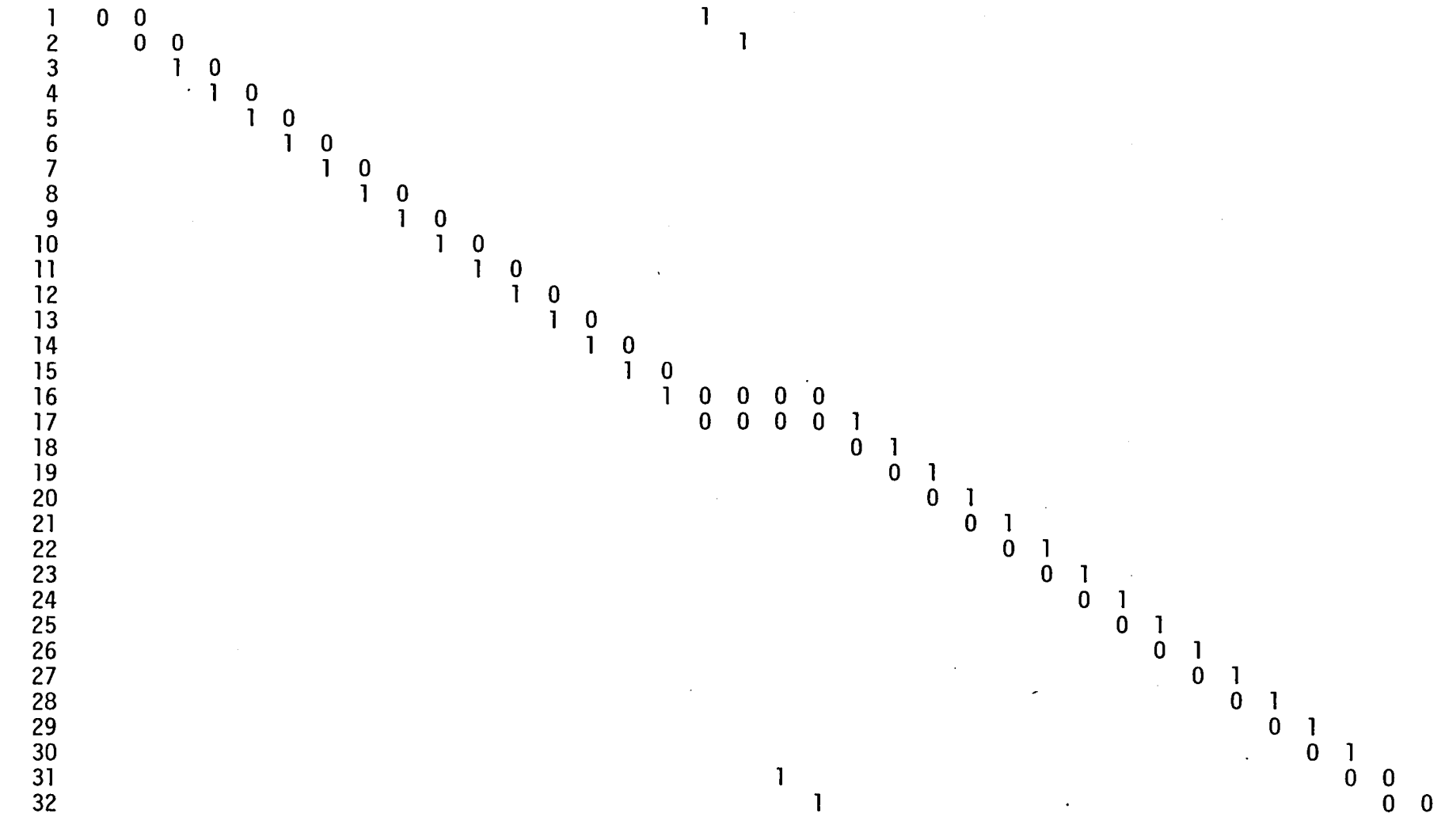

Figure 5.4 Elimination stage.

6.0 Future Developments

We propose a one year program which includes the following tasks:

TASK 1

Development and implementation of the Computer Code for the "Customized Reduction of Augmented Triangles" Algorithm

TASK 2

A study of CRAT algorithm for stability properties and possible variations of the algorithm

TASK 3

Further investigation of the Vector Cyclic Reduction Algorithm and Block LU decomposition Algorithm

TASK 4

Adaptation of the CRAT algortihm for solving banded systems with multiple right hand sides.

7.0 Comments

The CRAT algorithm, in its present form, employs no pivoting strategy within the triangular block matrices. However, column pivoting within the blocks is possible at 'little' cost. The CRAT algorithm successfully avoids the instabilities of the VCR algorithm for singular diagonal block matrices. CRAT is tailored to the VPS_32 pipeline architecture and therefore it should perform efficiently on the VPS_32 supercomputer.

8.0 Acknowledgements

I am grateful to Jules J. Lambiotte (Jr.) of NASA Langley Research Center for the useful discussions on VPS_32 processing. This work was carried out with financial support from the National Aeronautics and Space Administration (NASA) via research grant NAG-1-544 to Hampton University.

9.0 References

1. Hockney, R.W. A fast direct solution of Poisson's equation using Fourier analysis. J. ACM 12,1 (Jan. 1965) 95-113.

2. Heller, D. Some aspects of the Cyclic Reduction Algorithm for Block tridiagonal linear systems. SIAM J. Numer. Anal. 13(1976), 484-496.

Fig. A-I

Subroutine VCR-I (A, D, NDIM, LMAX, N, LG2N)
returns the solution of a tridiagonal linear system of equations.

PARAMETERS ARE:

N --- INTEGER; Must be set to the number of equations in the system; Value unchanged upon exit.

LG2N --- INTEGER; Must be set to $\log_2 N$; Value unchanged upon exit.

LMAX --- INTEGER; Must be set to 3N; Value unchanged upon exit.

D --- VECTOR; Length "LMAX"; Must be set as follows

$$D(i) \leftarrow \begin{cases} d_i & \text{iff} \quad 1 \le i \le N \\ d_{i-N+1} & \text{iff} \quad N+1 \le i \le 2N-1 \\ 1 & \text{iff} \quad i = 2N \\ d_{i-2N} & \text{iff} \quad 2N+1 \le i \le 3N-1 \\ 1 & \text{iff} \quad i = 3N \end{cases}$$

d_i $(1 \le i \le N)$ represents the diagonal entries; Initial values are changed upon exit.

A --- REAL VECTOR; Length 5N; Must be set as follows

$$A(i) \begin{cases} 1 & \text{iff } 1 \leq i \leq N \\ L_{i-n} & \text{iff } N+1 \leq i \leq 2N \\ a_{i-2N+1} & \text{iff } 2N+1 \leq i \leq 3N-1 \\ 0 & \text{iff } i = 3N \\ b_{i-3N} & \text{iff } 3N+1 \leq i \leq 4N-1 \\ 0 & \text{iff } i = 4N \\ L_{i-4N+1} & \text{iff } 4N+1 \leq i \leq 5N \end{cases}$$

L_i, a_i, and b_i represent the right hand vector, sub-diagonal entries and the super-diagonal entries respectively.

Initial Values are changed upon exit: A(i) contains the solution vector x_i (i = 1, 2 .. N) respectively.

```
      SUBROUTINE VCRI(A,D,NDIM,LMAX,N,LG2N)

      DIMENSION A(NDIM), D(LMAX)
      LI=N+1
      LJ=LI+N

      LK=LJ+N
      LL=LK+N
      LM=N+N

C NORMALIZE : RHS, SUB-, SUPER- DIAGS
      A(LI;LMAX)= A(LI;LMAX)/D(1;LMAX)

C REMAP RHS:
      A(LL;N)=A(LI+1;N)
      ISTEP=LG2N-1
      K=1
      KK=2
      NK=N-1
      DO 40 I=1,ISTEP

C COMPUTE PRODUCTS:
        D(1;LMAX)=A(LI;LMAX)*A(LJ;LMAX)

C UPDATE RHS & MAIN DIAGONAL:
        A(1;N)=1.0
        A(1;LM)=A(1;LM)-D(LI;LM)
        A(KK;NK)=A(KK;NK)-D(LI;NK)
       A(LI;NK)=A(LI;NK)-D(K;NK)

C UPDATE SUB-, SUPER- DIAGONALS:
       MK=LM-K
       A(LJ;MK)=-A(LJ;MK)*A(LJ;MK)
       K=K+K
        KK=K+1
        NK=N-K
C NORMALIZE RHS, SUB- SUPER-DIAGS:
       A(LI;N)=A(LI;N)/A(1;N)
       A(LJ;NK)=A(LJ;NK)/A(KK;NK)
       A(LK;NK)=A(LK;NK)/A(1;NK)

  40   A(LL;NK)=A(LI+K;NK)

C SOLVE FOR UNKNOWN VECTOR:
      A(LK+K;K)=A(LI;K)
       A(1;K)=1.0-A(LJ;K)
      A(KK;K)=A(1;K)
      A(1;N)=(A(LI;N)-A(LK;N))/A(1;N)
      RETURN
      END
```

Fig. A-II

Subroutine VCR-II (CV, A, D, NDIM, LMAX, N, LG2N)
returns the solution of a tridiagonal linear system of equations.

PARAMETERS ARE:

N --- INTEGER; Must be set to the number of equations in the system; Value unchanged upon exit.

LG2N --- INTEGER; Must be set to $\log_2 N$; Value unchanged upon exit.

LMAX --- INTEGER; Must be set to 3N; Value unchanged upon exit.

D --- VECTOR; Length "LMAX"; Must be set as follows

$$D(i) \begin{cases} d_i & \text{iff} \quad 1 \leq i \leq N \\ d_{i-N+1} & \text{iff} \quad N+1 \leq i \leq 2N-1 \\ 1 & \text{iff} \quad i = 2N \\ d_{i-2N} & \text{iff} \quad 2N+1 \leq i \leq 3N-1 \\ 1 & \text{iff} \quad i = 3N \end{cases}$$

d_i $(1 \leq i \leq N)$ represents the diagonal entries; Initial values are changed upon exit.

A --- REAL VECTOR; Length 5N; Must be set as follows

$$A(i) \leftarrow \begin{cases} 1 & \text{iff } 1 \le i \le N \\ L_{i-n} & \text{iff } N+1 \le i \le 2N \\ a_{i-2N+1} & \text{iff } 2N+1 \le i \le 3N-1 \\ 0 & \text{iff } i = 3N \\ b_{i-3N} & \text{iff } 3N+1 \le i \le 4N-1 \\ 0 & \text{iff } i = 4N \\ L_{i-4N+1} & \text{iff } 4N+1 \le i \le 5N \end{cases}$$

L_i, a_i, and b_i represent the right hand vector, sub-diagonal entries and the super-diagonal entries respectively.

Initial Values are changed upon exit: A(i) contains the solution vector, x_i (i = 1, 2 .. N) respectively.

CV — BIT VECTOR; Length 'LMAX'; Must be set as follows:

CV(i) = B'1', (1 ≤ i ≤ LMAX)

```
      SUBROUTINE VCRII(CV,A,D,NDIM,LMAX,N,LG2N)

      DIMENSION A(NDIM), D(LMAX)
      BIT CV(LMAX)
      LI=N+1
      LJ=LI+N

      LK=LJ+N
      LL=LK+N
      LM=N+N

C NORMALIZE : RHS, SUB-, SUPER- DIAGS
C REMAP RHS:
      A(LL;N)=A(LI+1;N)
      ISTEP=LG2N-1
      K=1
      KK=2
      NK=N-1
      DO 40 I=1,ISTEP

C COMPUTE PRODUCTS:
        D(1;LMAX)=A(LI;LMAX)*A(LJ;LMAX)

C UPDATE RHS & MAIN DIAGONAL:
        A(1;N)=1.0
        A(1;LM)=A(1;LM)-D(LI;LM)
        CV(1;MK)=Q8VMKO(NK,N;CV(1;MK))
        D(1;MK)=Q8VMERGE(D(LI;NK),D(1;N),CV(1;MK);D(1;MK))
        A(KK;MK)=A(KK;MK)-D(1;MK)

C UPDATE SUB-, SUPER- DIAGONALS:
       MK=LM-K
       A(LJ;MK)=-A(LJ;MK)*A(LJ;MK)
       K=K+K
        KK=K+1
        NK=N-K
C NORMALIZE RHS, SUB- SUPER-DIAGS:
        CV(1;MK)=Q8VMKO(N,NK;CV(1;MK))
        D(1;MK)=Q8VMERGE(A(1;N),A(KK;NK),CV(1;MK);D(1;MK))
        A(LI;MK)=A(LI;MK)/D(1;MK)
       A(LK;NK)=A(LK;NK)/A(1;NK)

  40  A(LL;NK)=A(LI+K;NK)

C SOLVE FOR UNKNOWN VECTOR:
      A(1;K)=1.0-A(LJ;K)
      A(KK;K)=A(1;K)
      D(1;N)=Q8VMERGE(A(LI;K),A(LK+K;K),CV(KK;N);D(1;N))
      A(1;N)=(A(LI;N)-D(1;N))/A(1;N)
      A(1;N)=A(LI;N)/A(1;N)
      RETURN
      END
```

Fig. A-3

Subroutine PRELUDE (P,R, X, D, NMAX, LG2N, N) returns the solution of a tridiagonal linear system of equations.

PARAMETERS ARE:

N --- INTEGER; must be set to the number of equations in the system; Value unchanged upon exit.

LG2N --- INTEGER; Must be set to $\log_2 N$; Value unchanged upon exit.

NMAX --- INTEGER; Must be set to 5N; Value unchanged upon exit.

D --- REAL VECTOR; Length N; Must be set to zero. Upon exit D(i) contains the solution $x_i (1 \leq i \leq N)$ respectively.

X --- REAL VECTOR; Length 5N; Must be set as follows

$$X(i) \leftarrow \begin{cases} a_i\ , & 1 \leq i \leq N \\ d_i\ , & N+1 \leq i \leq 2N \\ b_i\ , & 2N+1 \leq i \leq 3N \\ y_i\ , & 3N+1 \leq i \leq 4N \\ 0\ , & \text{otherwise} \end{cases}$$

where y_i, a_i, b_i d_i represent the right hand vector, sub-, super-, and main diagonals of

		the band matrix; Values changed upon exit.
P	---	REAL VECTOR; Length N; Working array.
R	---	REAL VECTOR; Length N; Working array.

FIG. A3

```
      SUBROUTINE PRELUDE (P,R,X,D,NMAX,LG2N,N)
      DIMENSION DIAG(N),X(NMAX),P(N),R(N)

      I=N/2+1
      J=N*2+1
      L=N*3+I

       X(L+N/2;N)=0.0
       R(1;N)=X(J+N;N)
       X(3*N+1;N/2)=0.0
       DIAG(1;N)=X(N+1;N)
       X(N+1;N)=0.0
       P(1;N)=X(1;N)
        X(1;N/2)=0.0
        K=1

      DO 10 ISTEP=1,LG2N
         X(J;N)=P(J;N)/DIAG(1;N)
         X(L;N)=R(1;N)/DIAG(1;N)
         DIAG(1;N)=1.-X(I;N)*X(J-K;N)-X(J;N)*X(I+K;N)
         R(1;N) =X(L;N)-X(I;N)*X(L-K;N)-X(J;N)*X(L+K;N)
         P(1;N)=-X(I;N)*X(I-K;N)
         X(J;N)=-X(J;N)*X(J+K;N)

10    K=K+K
      DIAG(1;N)=R(1;N)/DIAG(1;N)
      RETURN
      END
```

End of Document