ENHANCEMENTS TO SPERRY/NASTRAN

Tadashi Koga
Nippon Univac Kaisha, Ltd.

SUMMARY

This paper reviews the enhancement to NASTRAN program performed by NUK (Nippon Univac Kaisha, Ltd.) added to Level 15.5. Features discussed include intermediate checkpoint-restart in triangular decomposition, I/O improvement, multibanked memory and new plate element. The first three improvements provides the capability to solve significantly large size problem, while the new elements release the analyst from the cumbersome work to constrain the singularities caused by the lack of stiffness of inplane rotation of old plate elements.

INTRODUCTION

Since 1974, NUK has been maintaining and developing a NASTRAN based on COSMIC level 15.5 and distributing it to the UNIVAC customer in Japan and Europe as SPERRY/NASTRAN in co-operation with Sperry Support Services. In Japan, more than 35 customer are using our version of NASTRAN.

Since the leasing policy and export restrictions of source code were placed on post level 15.5, our motivation to offer SPERRY/NASTRAN is to provide state-of-the-art analytical capabilities incorporated in level 16.0 or later and error corrections in a timely manner to the UNIVAC customers outside U. S. A. In addition to maintaining a SPERRY/NASTRAN, we are also publishing a Newsletter each time the new version in released, providing user training seminar 8 times a year and developing and maintaining pre and post processors.

These improved features are listed in Table 1 and some of them are discussed in this paper.

DCOMP2

In static analysis, the computing time associated with triangular decomposition forms 70-80% of total time if the number of grid points exceeds 3000. Since a checkpoint can be taken only after the completion of the module, if the decomposition is interrapted by unexpected error such as max time, dish fault, system hung up, the computation up to this point of error is completly lost. In this case, restart run can save only 10-15% of total run time. The use of new module DCOMP2 provides a method by which any number of intermediate checkpoints can by taken during symmetric matrix decomposition. The module supervises the existing symmetric decomposition routine (SDCOMP) by interrupting the factoring process at intervals specified by the user.

The new module, DCOMP2, must be applied to an analysis through a rigid format DMAP alter. (Fig 1)

DMAP Calling Sequence

```
DCOMP2     KLL / LLL, ULL, SCR / V, N, NPARA / V, N, MPARA $
```

Input Data Block

KLL - Partition of stiffness matrix - 1 set

Note : KLL is assumed to be symmetric and may not be purged.

Output Data Blocks

LLL - Lower triangular factor of KLL - 1 set.

ULL - Upper triangular factor of KLL - 1 set.

SCR - Scratch Data Block for checkpoint/restart, contains the front matrix currently held in open core.

Notes :
1. LLL and ULL may not be purged.
2. ULL is not a standard upper triangular matrix. Its format is compatible only for input to subroutine FBS.

Parameters

NPARA - Input/output - integer - no default
= -1 No operation is taken
= 0 Normal decomposition
= +N Number of intermediate checkpoints required.

MPARA - output - integer - no default, MPARA contains the number of checkpoints completed.

Notes :
1. The number of checkpoints required, NPARA, is decreased by 1 each time an intermediate checkpoint is made.
2. The number of checkpoints made, MPARA is increased by 1 each time an intermediate checkpoint is made.
3. Both parameters are saved in data block XVPS and used at the time of restart to determine current location in the decomposition procedure.

Method

The general procedure for utilizing the intermediate checkpoint feature, is to replace the existing symmetric decomposition instruction, e g. RBMG2 with the new module DCOMP2. This is accomplished by using a set of DMAP alters for rigid format.

The new module contains a parameter, MPARA, which is used to specify the number of intermediate checkpoints to be taken. The user must determine before the first execution, the value of NPARA. If a restart is required of a partially decomposed matrix, the value of NPARA supplied in the DMAP instruction is replaced by the value in the XVPS data block, which can be used to determine at which row in the KLL the last checkpoint occured.

It may be noted that the checkpoint dictionary provided after each intermediate checkpoint, specifies a re-entry into DMAP instruction 90, however, the use of DMAP alter forces re-execution of the inserted module. This action assured that the restart will re-enter the DMAP instruction 89. When the value of NPARA is reduced to 0 a normal exit from the SDCOMP routine will be made and NPARA set to -1. If a restart is made after the decomposition has been completed, NPARA = -1, the DCOMP2 module gives an immediate return, therefore, it is not necessary to remove the DMAP alter statements.

Fig. 1 describes the meaning of the parameter NPARA which determines how the matrix is broken into ΔN rows for decomposition. At the first entry to subroutine SDCOMP, ΔN is decided from NPARA and the matrix size N. When the decomposition is completed for every ΔN rows, SDCOMP copies the values in open core and scratch file (in case spill occures) to SCR, then gives an alternate return to DCOMP2. DCOMP2 calls XCHK to write LLL, ULL and SCR on NPTP and after returning from XCHK, calls SDCOMP again to continue the decomposition and increase the value of MPARA by 1.

In the restart run, SDCOMP recovers open core from SCR and resumes decomposition from MPARA $*\Delta N + 1$ st row of matrix KLL.

IMPROVED I/O

SPERRY/NASTRAN has two improvements in I/O routine, asynchronous I/O and multiple I/O block read.

Asynchronous I/O

In conventional NASTRAN, one buffer area is assigned to each I/O unit and actual I/O request will be initiated when the buffer area is filled in case of write operation. The execution of NASTRAN is suspended until all data in buffer area is transfered to external storage.

In most of computer systems, asynchronous I/O capability is provided which makes it possible to process I/O operation and non-I/O operation simultaneously. To take full advantage of this capability, I/O routines of NASTRAN (GINO) were drastically rewritten.

The new GINO routine divides the I/O buffer into two parts. One of them is used for the data transfer for the higher level subprograms and the other is used to transfer the data from/to the external storage. (Fig 2-1)

This new capability has no effect to computer time, but the elapsed time (wall clock time) may be reduced to 2/3 of the old.

MREAD (Multiple Read Routine)

In real eigenvalue analysis, Invers Power Method is most papulary used due to its efficiency. Iterative procedure of this method is described as follows.

step		description
1	$\lambda_0 \rightarrow \lambda_1$	shift
2	$K-\lambda_1 M \rightarrow D$	form dynamic matrix
3	$D \rightarrow L \cdot U$	triangular decomposition
4	$M \cdot U_n \rightarrow V_{n+1}$	matrix multiplication
5	$(L \cdot U)^{-1} V_{n+1} \rightarrow W_{n+1}$	substitution
6	$W_{n+1}/c \rightarrow U_{n+1}$	normalization
7	convergence check	

where K : stiffness matrix
M : mass matrix
λ : estimated eigenvalue
u, v, w : iteration vector

Generally, for the extraction of one eigenvalue, step 4, 5 will be performed 8～10 times and step 3 will be done once. Regarding to computer time and memory size, step 3 and step 4, 5 have quite different characteristics described below.

step	CPU time	I/O time	memory size
3	$O(NB^2)$	O(NB)	B^2
4, 5	O(NB)	O(NB)	$14 \times N$

where N : size of matrix
B : semi-band width of matrix

As easily shown from this table, CPU time is dominant in step 3 while I/O time is dominant in step 4, 5, and the memory size in step 4, 5 is much less than that of step 3.

To reduce the I/O time in step 4, 5, we can remember the fact that I/O time depends on not only the number of words transfered but also the number of times I/O operations requested.

The new open routine (MOPEN) determines the number of I/O blocks held in open core. The lowest I/O routine (GINOIO) was changed so that the one I/O operation fills all of these I/O blocks and the I/O request of one block to GINOIO is considered to be a change of pointer to current I/O block untill all the I/O blocks held in open core were exhausted.

Since the size of I/O block was not changed, the size of actual I/O request can be determined according to the size of open core available.

This new capability is also applied to FBS (forward backward substitution), TRD (transient response displacement) and TRHT (transient response heat transfer). The numerical examples described in Fig 2.2 thru Fig 2.5 shows us that the I/O time will be reduced to 1/3 to 1/5 of the old.

MULTI BANKED MEMORY

At this time Univac system has no virtual storage capability, and the addressing limit of 262 KWD. Therefore, if the summation of B (semi-band width) and C (number of active columns) of stiffness matrix exceeds nealy 470, front matrix generated during the symmetric real decomposition (performed by SDCOMP) can not be held in the main memory. Some portion of this overflowed area (called 'spill') is processed in the area for active column, but remainder are stored in disk area. The processing of this spilled portion requires frequent I/O operation which results significant decrease of the execution efficiency.

On the other hand, the model size which is indicated by the number of grid point and number of elements has been increasing significantly due to the requirement of the engineers to obtain more precise simulation result. With the aid of powerful, easy to use preprocessors, the size of finite element model can easily exceeds the above limit.

To override this defect, SPERRY/NASTRAN provides automatic memory expansion capability to use banked memory as internal file up to 4MWD. The spilled portion of the front matrix is not stored in disk space but in this banked memory (called ADDITIONAL CORE). Since the data transfer to (or from) this banked memory is no longer an I/O operation but a simple store (or load) operation, the I/O time was significantly reduced. (Fig 3.1, 3.2)

For the implementation of this capability, following subroutines are developed.

OPENX - According to the number of words requested, reserve the banked memory via MCORE$ and external disk space if necessary.

READX/WRITEX - Transfer the data from/to the banked memory.

CLOSEX - Release the banked memory via LCORE$.

To examine the efficiency of this new capability, comparisons between the new and old NASTRAN are done. Because the comparison with large size problem which causes spill from 262 KWD requires much CPU time, middle size problems are examined restricting program size less than 200 KWD to cause spill processing.

From the results shown in Fig 3.3 and Fig 3.4, we can conclude that the significant reduction in I/O time is achieved while the CPU time reduction is a little. Since the number of words of spilled area is proportional to the square of B+C, the effect of this capability becomes more apparent as the problem size increases.

NEW PLATE ELEMENTS

NASTRAN assumes 6 degrees of freedom (d.o.f.) per grid point. However, all the original elements have 5 or less d.o.f.s at their grid point. Though the lack of stiffness of each grid point is checked and informed by NASTRAN automatically, the generation of appropriate constraint data for these singularities is a difficult work especially in the following case.

inclined flat plate model
curved shell which has a big radius of curvature

Some modules are developed to constrain these singularities by assuming that all the d.o.f. which remains singular should be constrained by SPC processing. Some other modules only made the card images to constrain such d.o.f.s by SPC or MPC processing. Considering the difficulties of such function due to the existence of MPC, we thought it better to incorporate the plate element which has 6 d.o.f.s at their grid points.

Two plate elements named TRIA3 and QUAD4 are incorporated. These elements are formulated from 'Pian's hybrid element'. Assuming the displacement function on each side of the element, the stress function within the element, the equilibrium equation on grid point is formulated based on the principle of complemental virtual work. Membrane element and bending element are produced independently and these are combined without coupling each other. Following are the outline of triangular element.

<u>Membrane Element</u>

The degrees of freedom on a grid point are u, v and θz. U and v are the components of displacements paralle to the axis of the local coordinate system. θz is the average value of the rotation of side, not the ratational angle of grid point itself. The displacement functions are defined on the side and shown as follows.

displacement tangential to side (Us) = linear function
displacement normal to side (Un) = cubic function
rotational angle ($\theta = \partial Un/\partial r$) = quadratic function

where
r is parameter on side
n is normal direction to side
s is tangential direction to side

The stress function is defined within element and as follows.

$$\{\sigma\} = \begin{Bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{Bmatrix} = \left[\begin{array}{ccc|ccc|ccc} \zeta_1 & & & \zeta_2 & & & \zeta_3 & & \\ & \zeta_1 & & & \zeta_2 & & & \zeta_3 & \\ & & \zeta_1 & & & \zeta_2 & & & \zeta_3 \end{array}\right] \begin{Bmatrix} \{\sigma\}_1 \\ \{\sigma\}_2 \\ \{\sigma\}_3 \end{Bmatrix}$$

$$\{\sigma\}i = \begin{Bmatrix} \sigma_x \\ \sigma_y \\ \sigma_{xy} \end{Bmatrix} \quad \text{at grid point i}$$

where ($\zeta_1, \zeta_2, \zeta_3$) are the area coordinates.

In order to satisfy the equilibrium equation within element, 2 components of 9 $\{\sigma\}_1, \{\sigma\}_2, \{\sigma\}_3$ are dependent.

Bending Element

The degrees of freedom on the grid point are W, θx and θy. W is the deflection normal to the x, y plane.

The displacement functions are defined on the side and shown as follows.

deflection W = cubic function
side direction slope $\partial W/\partial s$ = quadratic function
normal slope to side $\partial W/\partial n$ = linear function

The stress function is defined within element as follows.

$$\{\sigma\} = \begin{Bmatrix} m_x \\ m_y \\ m_{xy} \end{Bmatrix} = \left[\begin{array}{ccc|ccc|ccc} \zeta_1 & & & \zeta_2 & & & \zeta_3 & & \\ & \zeta_1 & & & \zeta_2 & & & \zeta_3 & \\ & & \zeta_1 & & & \zeta_2 & & & \zeta_3 \end{array}\right] \begin{Bmatrix} \{\sigma\}_1 \\ \{\sigma\}_2 \\ \{\sigma\}_3 \end{Bmatrix}$$

$$\{\sigma\}i = \begin{Bmatrix} m_x \\ m_y \\ m_{xy} \end{Bmatrix} \quad \text{at grid point i}$$

where $m_x = \int \sigma_x \cdot z \, d_z$
the integration takes place over direction to plate thickness.

9 components of stress are independent.

Composite Shell Element

Shell element is composed of membrane element and bending element without coupling each other.

Quadrilateral Element

Quadrilateral element (QUAD4) is composed from the 4 overlapping TRIA3 elements. (Fig 4.1)

Numerical Evaluation

The model and loading condition are shown in Fig 4.2 and the results are shown in Fig 4.3 thru Fig 4.5. From these results, following conclusions are obtained.

(1) Inplane bending

Because the elements TRIA2 and QUAD2 absorbed the inplane bending as the form of shear, the deformed shape of such elements was not so good. In this viewpoint, TRIA3 and QUAD4 are recognized to be improved, and moreover the stress calculated falls in the safety side.

(2) Out of plane bending

Regarding the displacement, no visible difference between TRIA3 and TRIA2 is recognized. But the distribution of reaction forces reverses.

(3) Inplane tension

The TRIA2 and QUAD2 elements shows good results, on the other hand, the solution of new elements at the end side causes disorder.

(4) Summary

The accuracy of new elements TRIA3 and QUAD4 is better than TRIA2 and QUAD2 in many cases. Moreover, introduction of θz as additional d.o.f. makes lose the necessities of constraint by SPC or MPC card, troubles on modeling will be reduced.

CONCLUSIONS

Several improvements and enhancements performed by NUK (Nippon Univac Kaisha, Ltd.) are described in the paper. These modifications increase the usefulness and efficiency of the NASTRAN program.

Due to the leasing policy and export restrictions placed on post 15.5, and the requirement of Japanese customer such as, timely error corrections, response to technical questions which sometimes need the understanding of source code and Japanese manual, we will continue to develop our NASTRAN. But if NASA can consider alternatives to the current policies so that the Japanese company can respond above requirements, the number of users of COSMIC/NASTRAN will increase steadily.

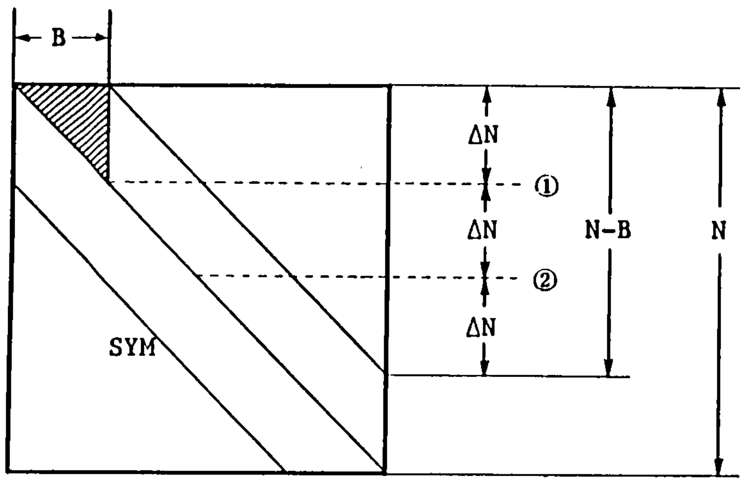
The list of current development priorities for our NASTRAN are :

- FEER method
- HEXA element - 8 to 20 nodes (variable)
- PLOAD3, PLOAD4 - pressure load on solid surface
- BAR element including warping
- Improved solid element - incompatible mode
- Response spectrum
- Output of strain and strain energy

Table 1. Major Improvements Added to Level 15.5

Item		Description
1)	Isoparametric Elements	IS2D4, IS2D8, IS3D8, IS3D20
2)	RF14	Thermal transient-structural static
3)	RF13	Normal mode analysis with differential stiffness
4)	ELBOW	curved beam element
5)	PLOAD1	distributed load on BAR element
6)	TRIA3, QUAD4	Plate element with 6 d.o.f.s at each grid point
7)	RBAR, RBE2	Rigid elements
8)	AUTOSPC	Automatic constraint by SPC
9)	READ	Append capability
10)	RF4	Iterative procedure
11)	NOLIN	Iterative procedure
12)	GPSC	SPC, MPC generator
13)	DCOMP2	intermediate checkpoint-restart in SDCOMP
14)	GINO	asynchronous I/O, multiple block read
15)	SDCOMP	banked memory
16)	FBS, INVFBS	non-transmit type UNPACK
17)	TRIA6, QUAD8	higher order shell element
18)	PENTA	15 node wedge element
19)	SPRNG	Spring element

Remark) Items 1) thru 4) are performed by Sperry Support Services.



$$\Delta N = \frac{N - B}{NPARA + 1}$$

(Checkpoint will be taken NPARA times during decomposition)

Following DMAP ALTERs are required for Rigid Format 1.

```
ALTER    3
FILE     LLL=APPEND/ULL=APPEND/SCR=APPEND  $
ALTER    89,89
DCOMP2   KLL/LLL,ULL,SCR/V,N,NPARA=2/V,N,MPARA  $
ENDALTER
```

Fig. 1   Intermediate Checkpoint-restart in SDCOMP

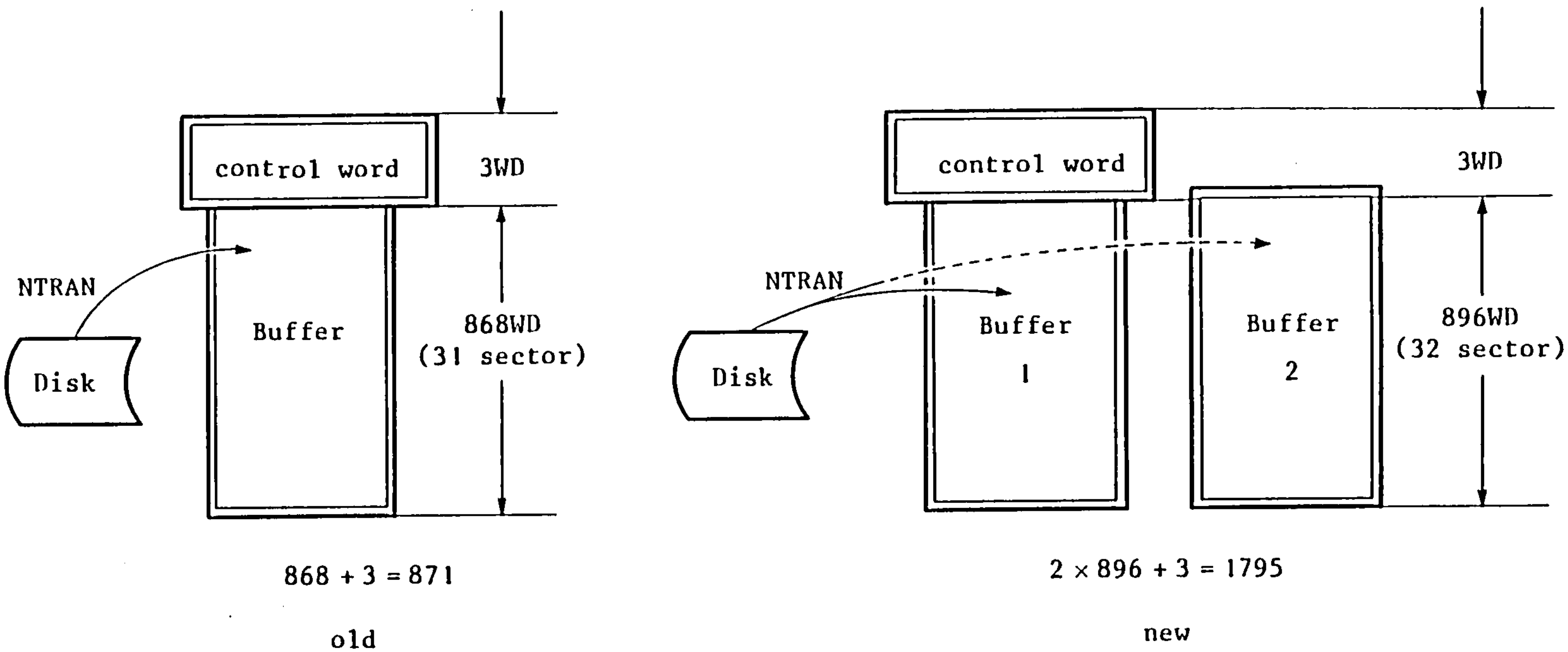


Fig. 2.1 Asynchronous I/O

		READ	No. of Decomp.	No. of Iteration	Memory size (WD)
old	CPU SUP	0.27 1.0	3	30	120K
new	CPU SUP	0.22 0.45	2	37	220K

SUP = CPU + I/O + overhead

No. of GRID points 688

No. of Elements 751

Total d.o.f. 3900

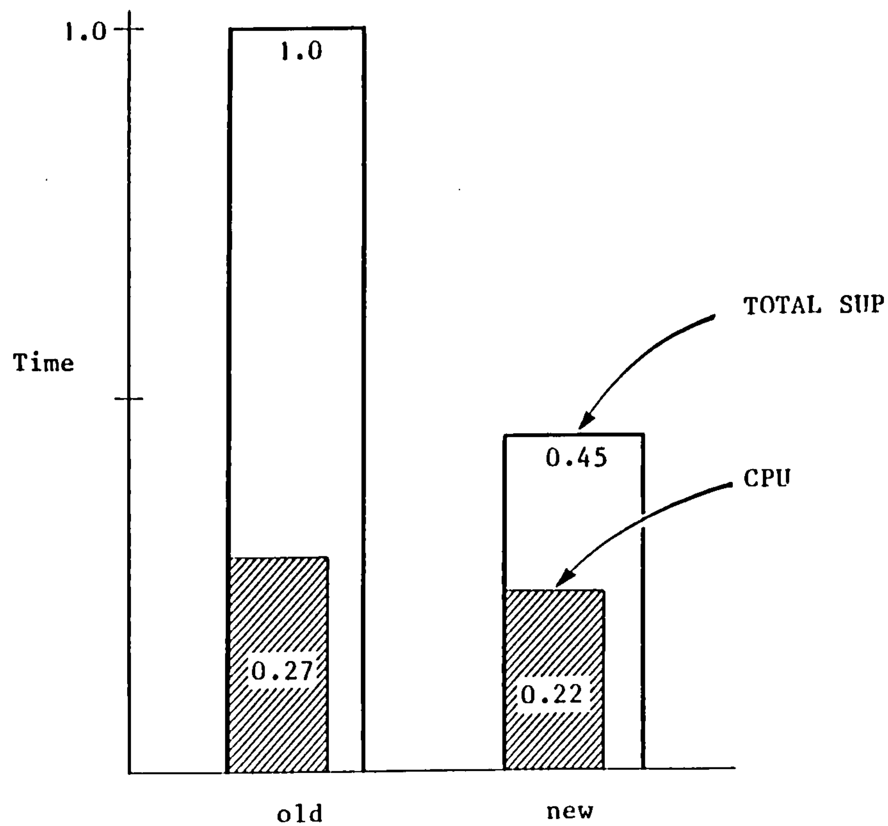


Fig. 2.2 MREAD in READ

		SDCOMP	TRD	Memory size (WD)
old	CPU	0.03	0.26	100K
	SUP	0.05	1.0	
new	CPU	0.03	0.25	257K
	SUP	0.05	0.42	

B + C : 193

Total d.o.f. : 2530

No. of time step : 30

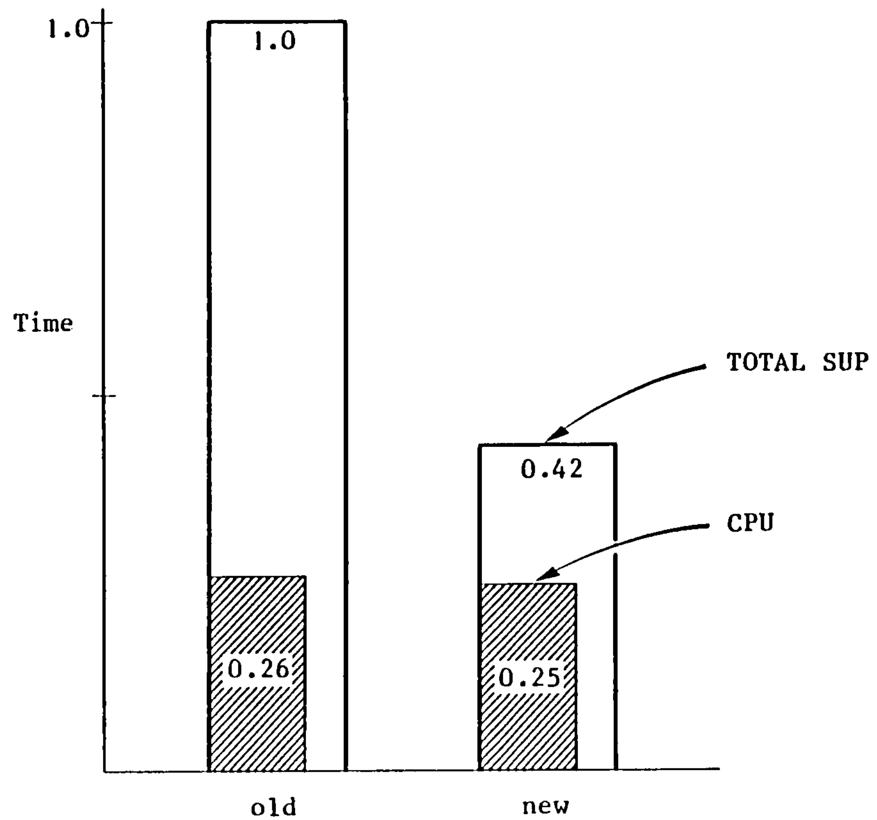


Fig. 2.3 MREAD in TRD

		SDCOMP	TRHT	Memory size (WD)
old	CPU SUP	0.01 0.03	0.24 1.0	70K
new	CPU SUP	0.01 0.03	0.24 0.45	217K

B + C : 22

Total d.o.f. : 1071

No. of time step : 31

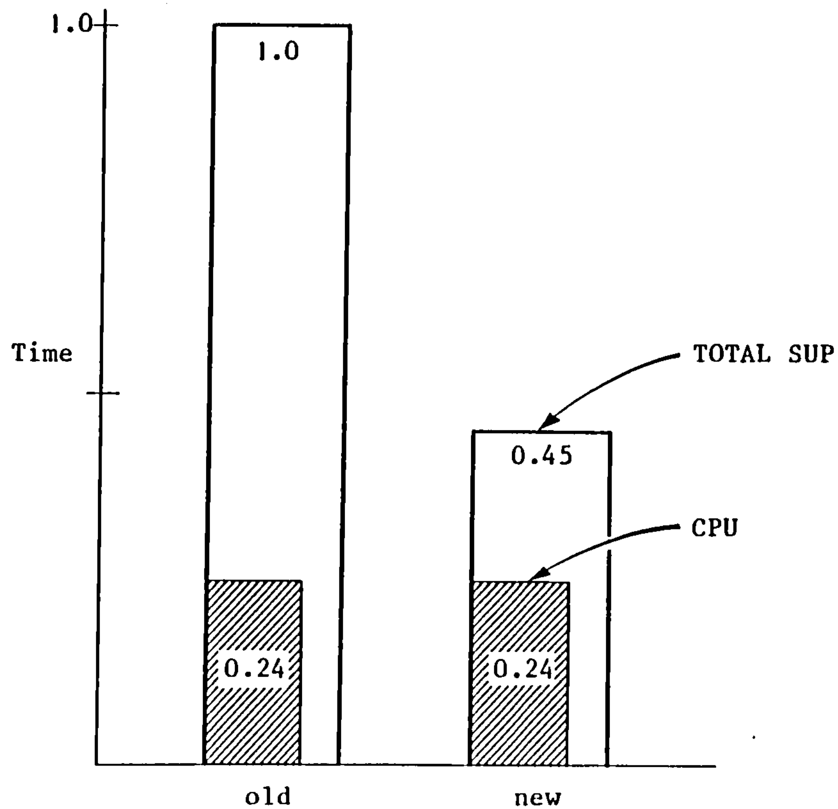


Fig. 2.4 MREAD in TRHT

		FBS	Memory size (WD)
old	CPU SUP	0.05 1.0	80K
new	CPU SUP	0.05 0.18	117K

B + C : 514

Total d.o.f.: 5072

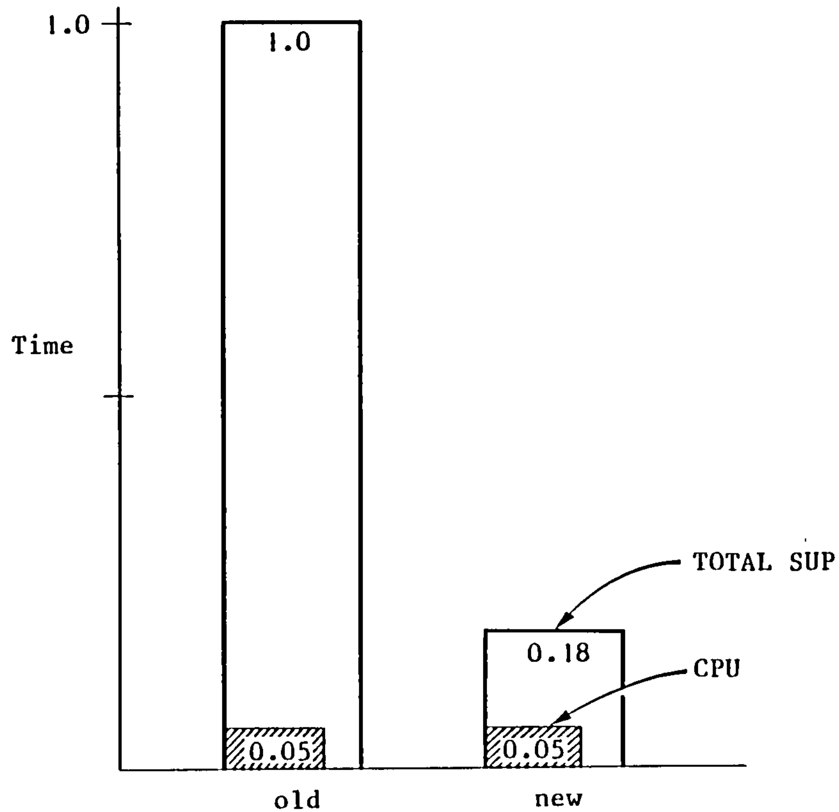


Fig. 2.5 MREAD in FBS

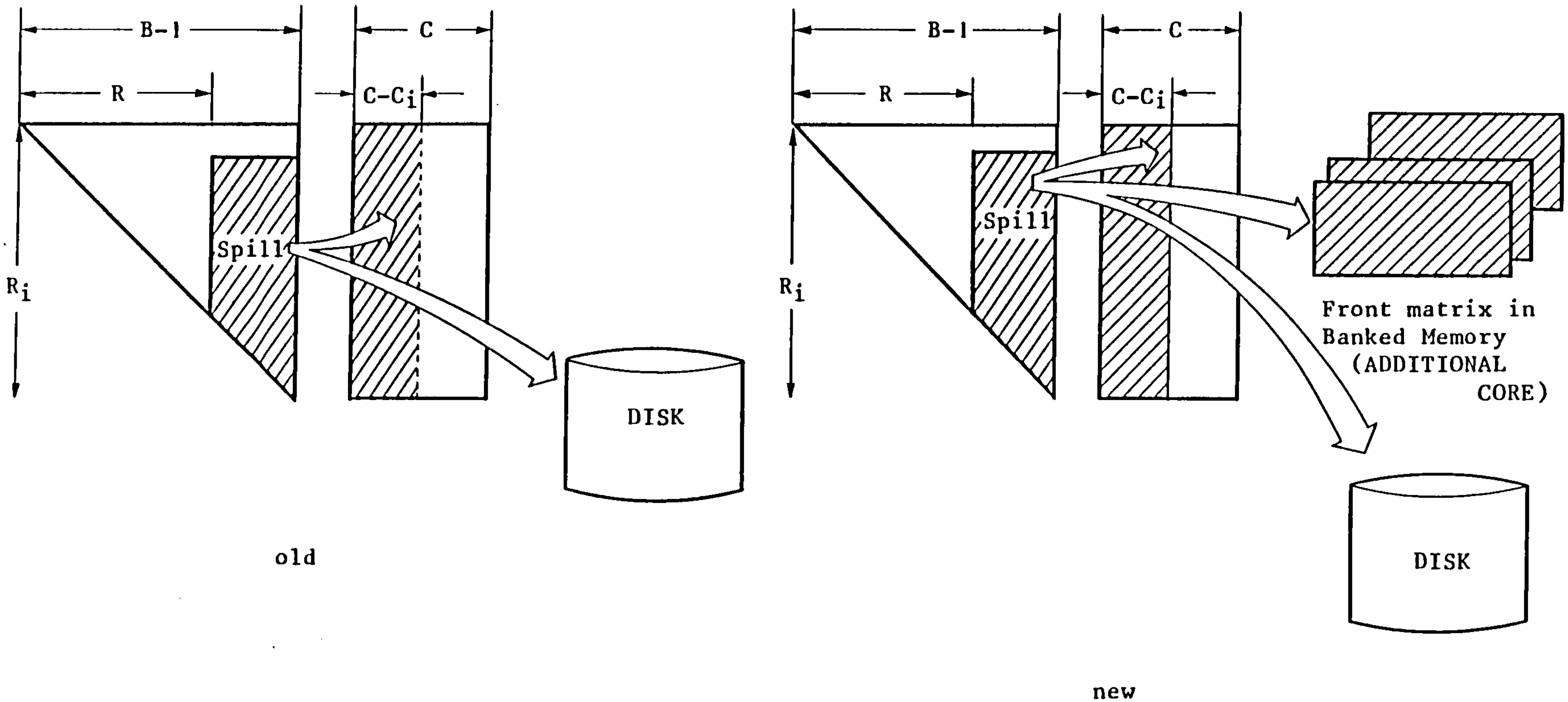
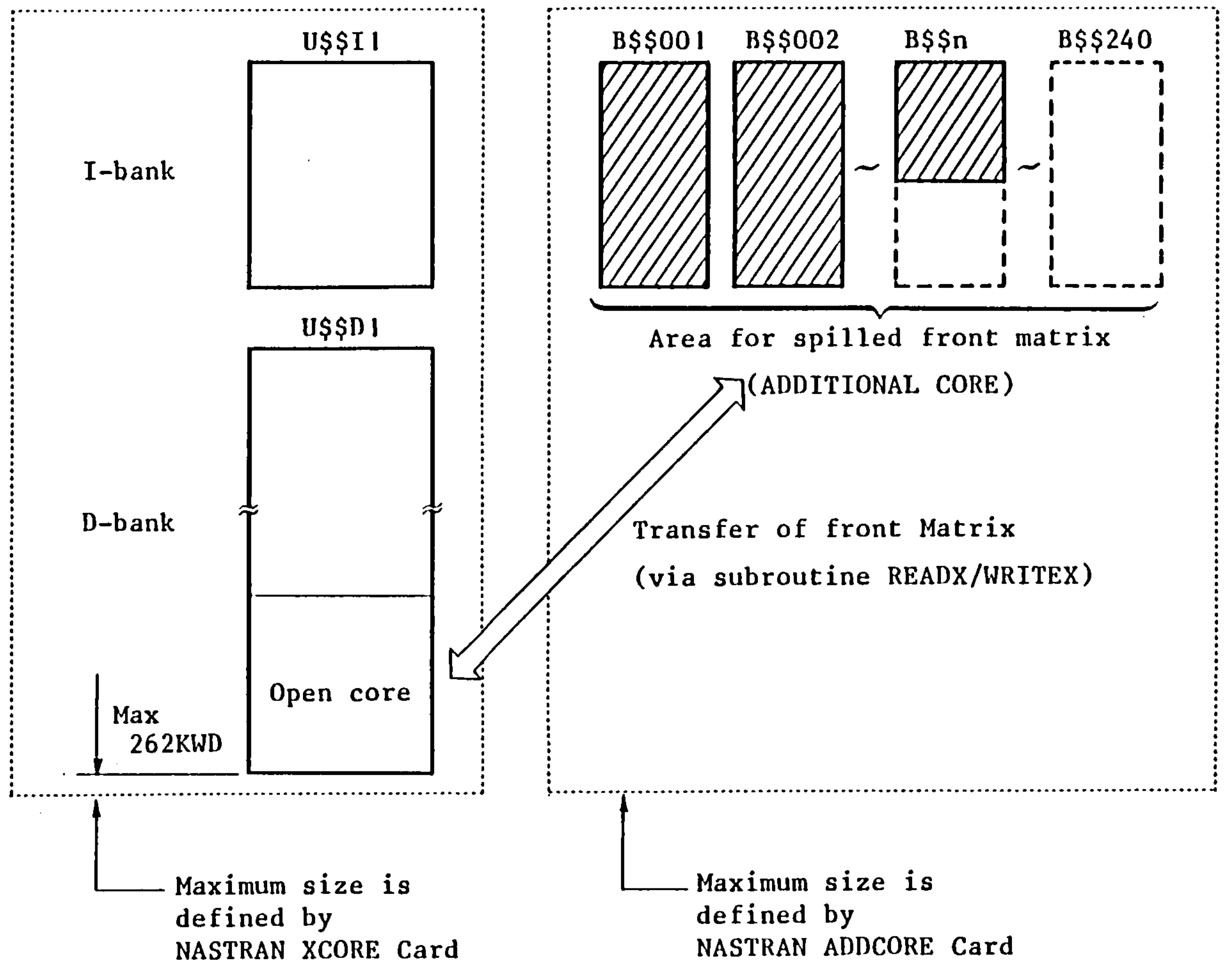


Fig. 3.1 Spill logic in SDCOMP

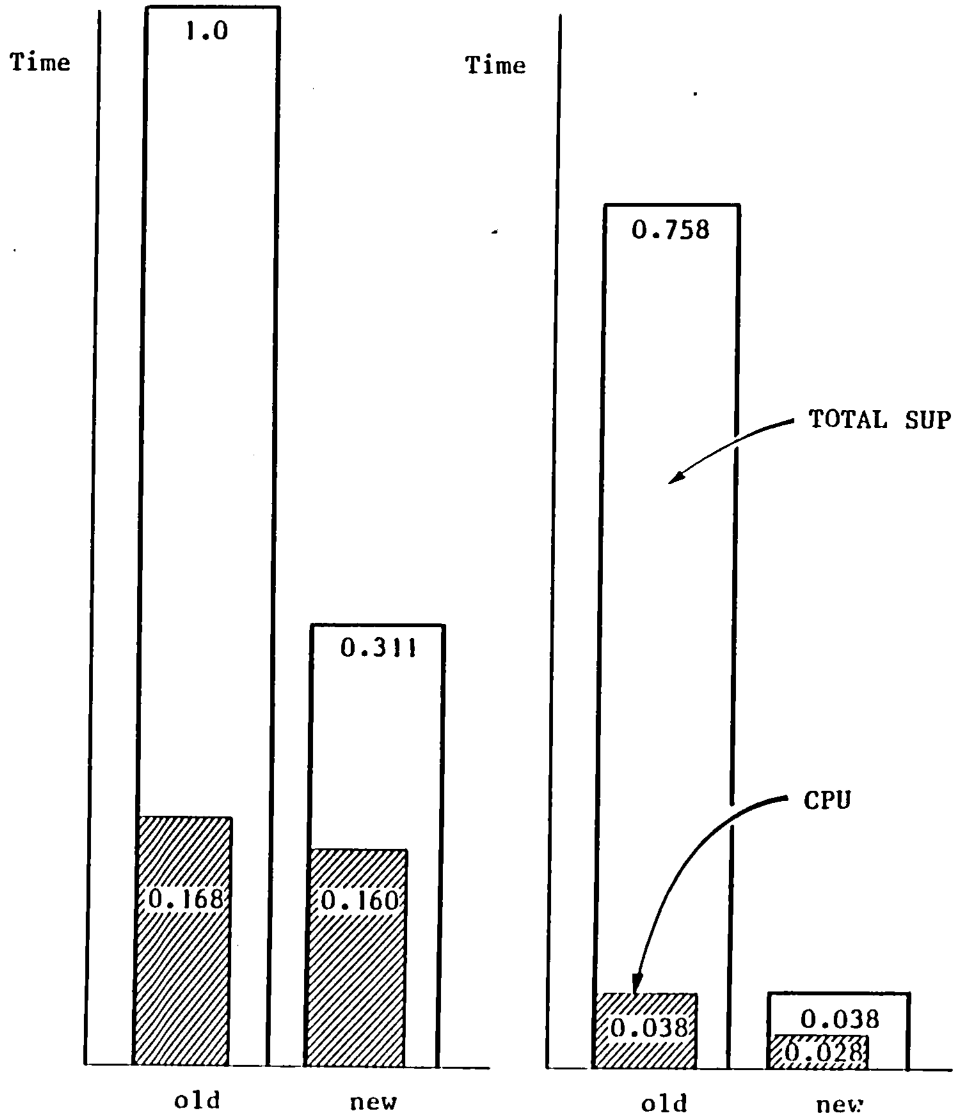


Bank name	Property	Contents
U$$I1	static	NASTRAN I-bank
U$$D1		NASTRAN D-bank
B$$001 ~ B$$240	dynamic	Area for spilled front matrix. Obtained by MCORE$ as much as required. Each bank size is the same as that of U$$I1.

Fig. 3.2 Bank structure

(a) Case A : old : 68KWD

new : 68KWD + ADDITIONAL CORE 24KWD



① SUP in NASTRAN ② SUP in SDCOMP

Case A

No. of Grid points	551
No. of Elements	992
B + C	230
N	2697

SUP in NASTRAN

	old	new
TOTAL SUP	1.0	0.311
CPU	0.168	0.160

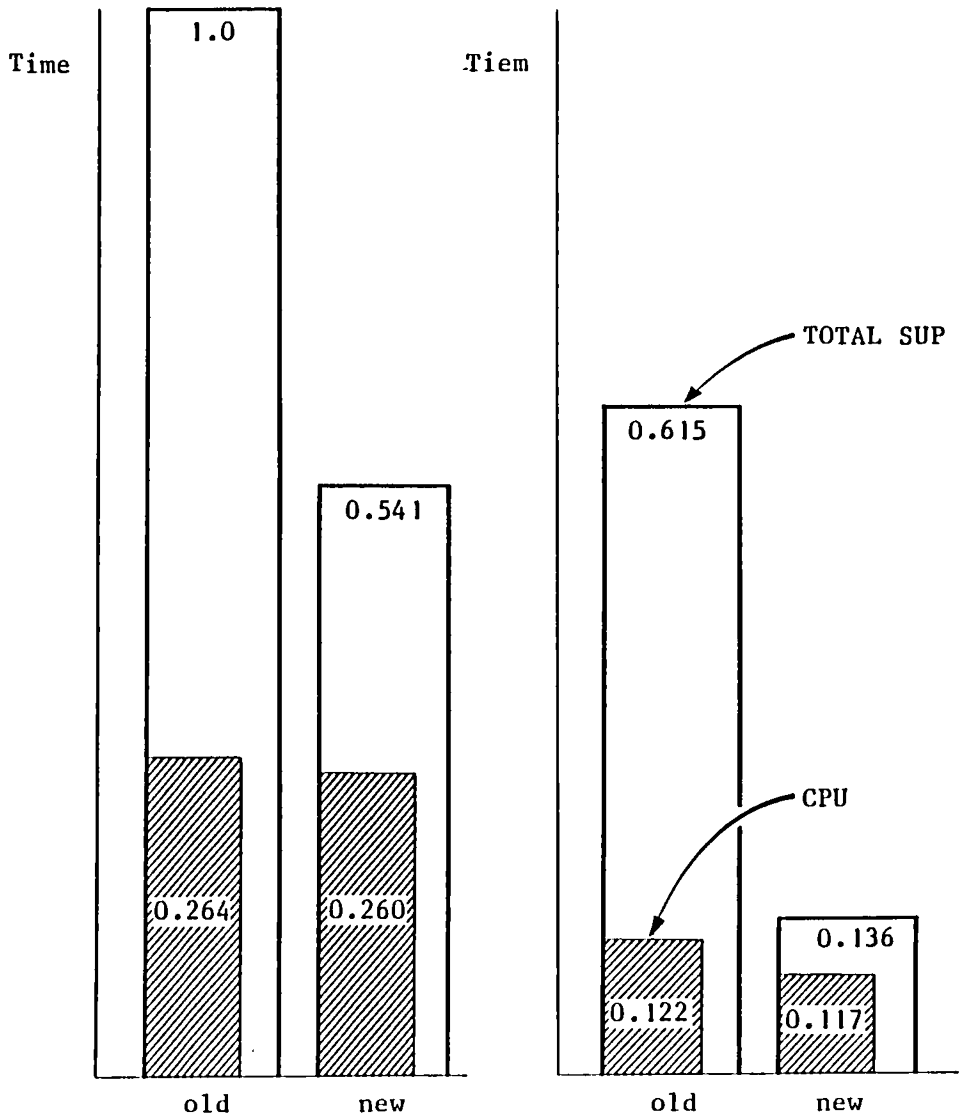
SUP in SDCOMP

	old	new
TOTAL SUP	0.758	0.038
CPU	0.038	0.028

Fig. 3.3 Efficiency test - 1

(b) Case B : old : 200KWD

new : 200KWD + ADDITIONAL CORE 43KWD



1 SUP in NASTRAN

2 SUP in SDCOMP

Case B

No. of GRID points	1061
No. of Elements	900
B + C	450
N	3966

SUP in NASTRAN

	old	new
TOTAL SUP	1.0	0.541
CPU	0.264	0.260

SUP in SDCOMP

	old	new
TOTAL SUP	0.615	0.136
CPU	0.122	0.117

Fig. 3.4 Efficiency test - 2

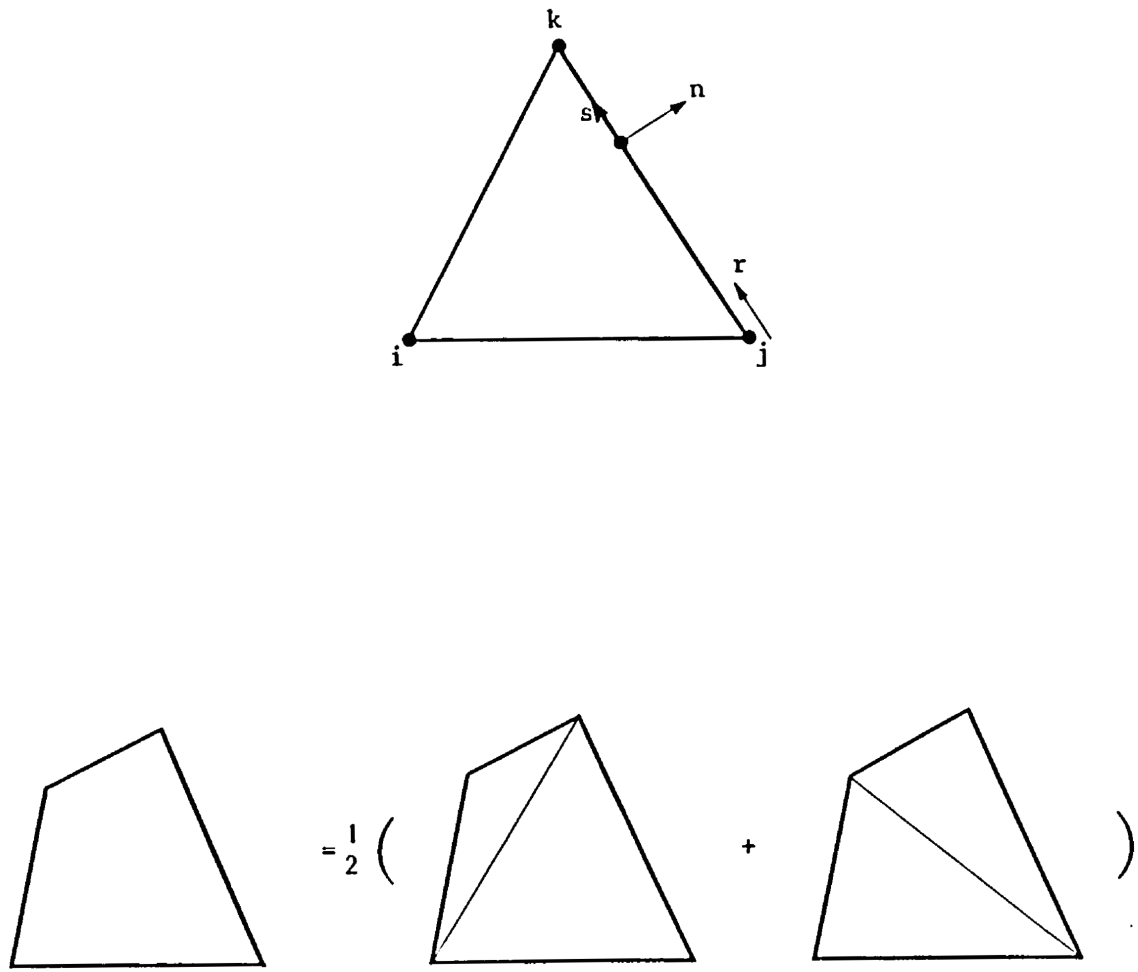
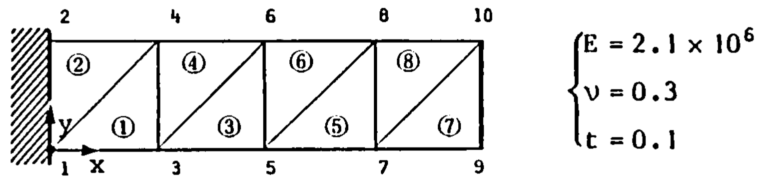


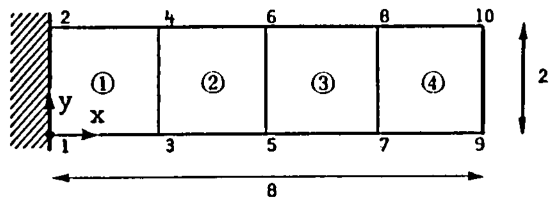
Fig. 4.1 TRIA3 and QUAD4

(3) Test Model

Triangular element



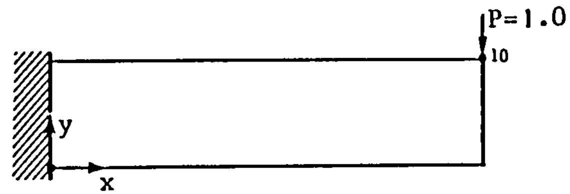
Quadrilateral element



Loading Condition

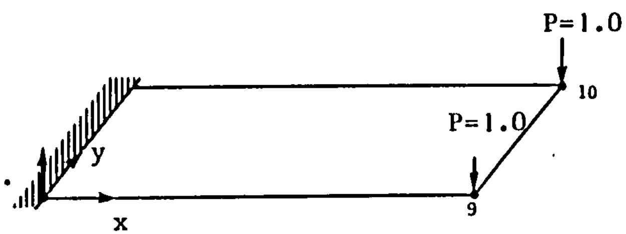
① In plane Bending

-1.0 loading Y-direction on GRID 10.



② Out of plane bending

-1.0 loading Z-direction on GRID 9 and 10.



③ In plane tension

1.0 loaing X-direction on GRID 9 and 10.

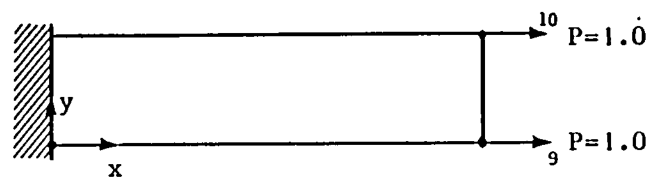


Fig. 4.2 Test Model for TRIA3 and QUAD4 element

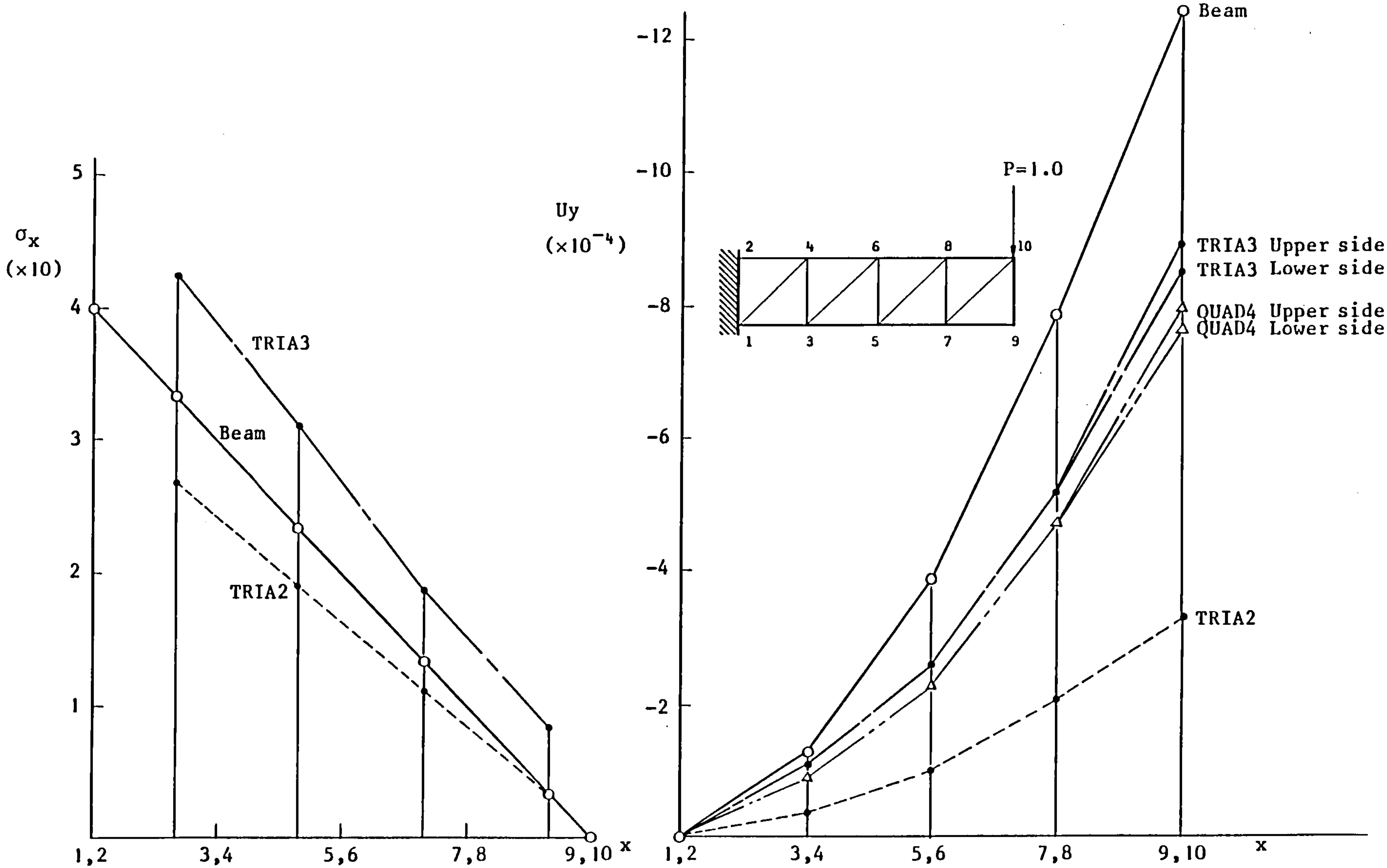


Fig. 4.3 Test Results of New Element - Inplane Bending

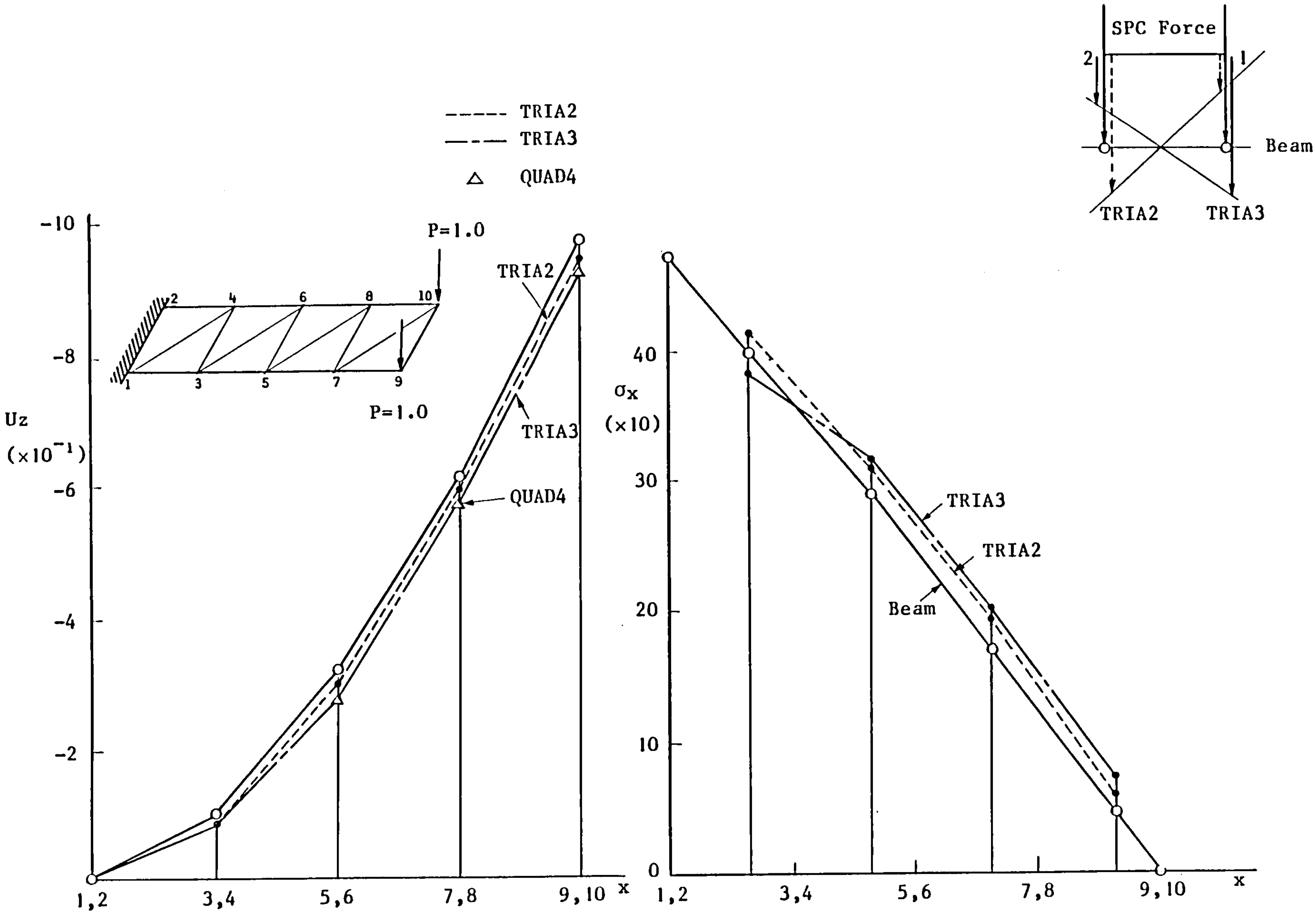


Fig. 4.4 Test Results of New Element - Out of Plane Bending

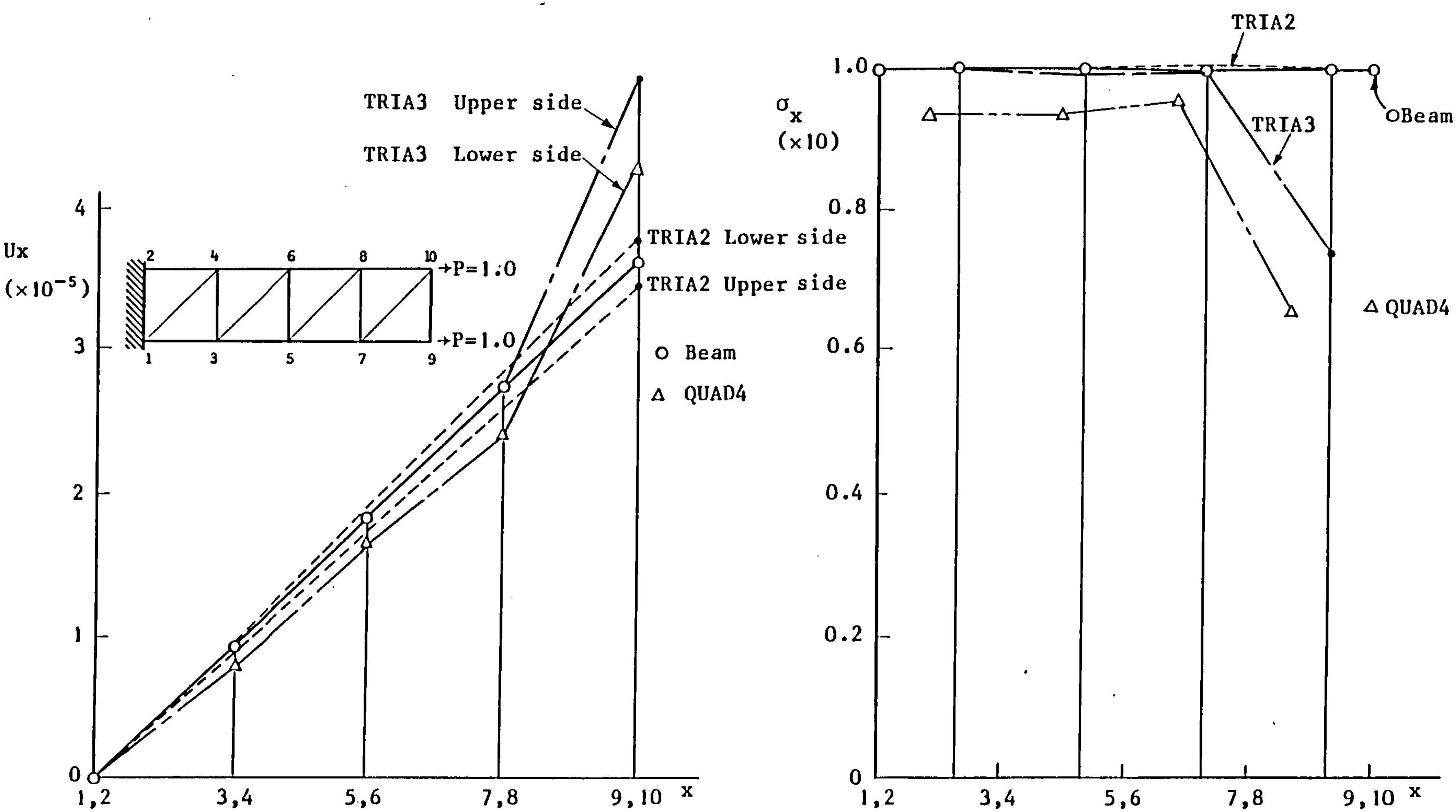


Fig. 4.5 Test Results of New Element - Inplane Tension

REFERENCES

1. Archie J. Jordan, Jr., William G. Ward : Modifications and Additions to NASTRAN at Marshall Space Flight Center.

   NASTRAN User's Experiences
   NASA TM X-2637, 1972

2. Caleb W. McCormick : Review of NASTRAN Development Relative to Efficiency of Execution.

   NASTRAN User's Experiences
   NASA TM X-2893, 1972

3. Ronald P. Schmitz : NASTRAN Maintenance and Enhancement Experiences.

   NASTRAN User's Experiences
   NASA TM X-3278, 1975

4. Edwin N. Hess : Dynamic Storage Expansion in NASTRAN.

   Seventh NASTRAN User's Colloquium
   NASA CP-2062, 1978

5. Shinichiro Harano : Improvements in Sparse Matrix Operations of NASTRAN.

   Ninth NASTRAN User's Colloquium
   NASA CP-2151, 1980

6. P. R. Pamidi : Recent Improvements and Enhancements to NASTRAN.

   Tenth NASTRAN User's Colloquium
   NASA CP-2249, 1982

7. "Sperry NASTRAN News letter" Vol.3 No.2 Sperry Support Services, Huntsville, Alabama, July 15, 1976

8. "Sperry NASTRAN Application Brief" 3.0 Sperry Support Services, Huntsville, Alabama, September 1, 1976

9. The NASTRAN User's Manual NASA SP-222 (01)

10. The NASTRAN Programmer's Manual NASA SP-223 (01)

11. Caleb W. McCormick : Sparse Matrix Operations in NASTRAN Proc. of 1973 Tokyo seminar on FEA, 1973 pp.611-631

12. T. H. H. Pian : Derivation of Element Stiffness Matrices by Assumed Stress Distribution. AIAA Journal Vol.2, 1964 pp.1333-1336

13. Yutaka Yoshida et al : A Flat Finite Element for Thin Shell Analysis Derived by Assumed Stress Approach. Proceedings of The Japan Society of Civil Engineers. No. 211, 1973