PCI - A PATRAN-NASTRAN MODEL TRANSLATOR

T.J. SHEERER

CHRYSLER TECHNOLOGIES AIRBORNE SYSTEMS, INC.

WACO, TX

INTRODUCTION

The existence of several derivative versions of NASTRAN, which differ significantly in element definitions and result formulation, has caused some difficulties in the interface between NASTRAN and pre- and post- processors such as PATRAN or SUPERTAB. In particular, the PATRAN-COSMIC/NASTRAN interface provided by PDA Engineering has not been updated at the same rate as the equivalent interface with MSC/NASTRAN, and has significantly less capabilities. Many model entities supported by both PATRAN and COSMIC/NASTRAN are not supported by the translator. The well-documented PATRAN neutral file, which is now supported by several other vendors, has provided a means for the user to create his own interface program for model translation, while it has also been possible to pass results form NASTRAN to PATRAN with a user-written program using OUTPUT2 statements and format information from the Patran Users' Manual. In recent years PDA engineering has provided as part of PATRAN a library of subroutines known as gateway utilities, which extract data directly from the PATRAN database file and which can be called from a FORTRAN program. As this eliminates the task of reading the neutral file the work of creating a translator can be produced by the user with little effort. This has been done with the object of producing a PATRAN-COSMIC NASTRAN model translator comparable in scope to the PATRAN-MSC/NASTRAN translator, but also allowing a greater degree of user control than is found therein. The different parts of the program were developed in several locations, as the counterpart to a results translator developed for Texas Instruments by Texas A and M University, which also emphasizes flexibility. Both these programs are public domain programs under the terms of the development agreement between Texas Instruments and Texas A.M.U., and enhancements developed at Chrysler have also been passed to T.A.M.U.

PROGRAM STRUCTURE

PCI supports a range of elements comparable with PATNAS and significantly greater that PATCOS. The structure of the program is such that, in effect, it supports elements not currently extant. In the PATCOS program element types are hard-wired, and if a different NASTRAN card is required a text editor must be used to change the bulk data file. If, for example, the QUAD4 element is required in a model, all elements of this type must be edited from type QUAD2 to QUAD4, since PATCOS does not currently support the QUAD4. Similar difficulties arise with other elements. The PATNAS translator does support a wide range of elements identified by configuration codes, but the equivalence between elements and configuration code is controlled from within PATRAN, and the user can not translate to an element developed in-house, or a newly introduced element, or simply an element not supported by the translator. PCI uses a different approach by having a user text file of twenty NASTRAN element names for each element configuration. This file may be edited by the user to assign whatever correspondence he desires between configuration code and NASTRAN element type. Since most element cards in NASTRAN follow the same pattern of (NAME,PID/MID,NODES) it is not generally necessary to write a new subroutine when adding an element. Frequently the only parameter changed is field one of the NASTRAN data card, and this is controlled from a text file. Because NASTRAN and PATRAN use a different numbering sequence for higher elements it is convenient to use several subroutines to write NASTRAN elements, but frequently only one routine is required for each shape/node combination. All 3-node shells, for example, are written by the same routine. The PATRAN configuration field is used to select an element name from a range of twenty for each configuration of shape/nodes. The user-generated text file ETYPES.DAT contains these names for configurations 1 through 20. Any elements having the default configuration of zero in the PATRAN database are assigned the value 1 in the translator.

Table (1) shows the basic PATRAN element types supported by PCI and the NASTRAN elements obtainable form them. For comparison the elements supported by PATCOS and PATNAS are also shown. note that, in general, failure by PCI to support elements listed for PATNAS is because they are not supported in COSMIC/NASTRAN. In practice, alteration of the text file of element configurations will allow support of these elements if and when they become available

TABLE 1: GEOMETRIC ELEMENT TRANSLATION CAPABILITIES OF PATCOS, PATNAS AND PCI

SHAPE/NODES	PATCOS	PATNAS	PCI
BAR/2	CBAR	CBAR CBEAM CROD	CBAR CROD
TRIA/3	CTRIA2	CTRIA1 CTRIA2 CTRIA3 CTRIARG	CTRIA1 CTRIA2 CTRBSC CTRIARG CTRMEM CTRPLT
QUAD/4	CQUAD2	CQUAD1 CQUAD2 CQUAD4 CQDMEM1 CQDMEM2 CSHEAR CTRAPARG	CQUAD1 CQUAD2 CQUAD4 CQDMEM1 CQDMEM2 CSHEAR CTWIST CQDPLT CQDMEM CTRAPAX
TRIA/6	CTRIM6	CTRIA6 CTRIAX6	CTRIM6 CTRSHL CTRPLT1
QUAD/8	CIS2D8	CQUAD8	CIS2D8
HEX/8	CIHEX1	CHEXA CHEX8	CIHEX1 CHEXA1 CHEXA2
QUAD/20	CIHEX2	CHEXA CHEX8	CIHEX1 CHEXA1 CHEXA2
QUAD/32			CIHEX3

PCI also supports the SCALAR and damping elements CELAS2 and CDAMP2 by generating a scalar element for each degree of freedom specified in a PATRAN SPRING element. The concentrated mass element CONM2 is obtainable from the PATRAN MASS directive. MPCs and rigid elements are supported. Table (2) summarizes the support for these elements. Node translation with embedded SPCs is fully supported.

TABLE 2: SCALAR, DAMPER AND MASS ELEMENT SUPPORT

PATRAN DIRECTIVE	NATRAN CARD WRITTEN BY PCI
MPC	MPC
MPC,RROD	CRIGIDR
MPC,RBAR	CRIGD1
MPC,RBE1	CRBE1
MPC,RBE2	CRBE2
MPC,RBE3	CRBE3
BAR/2/n OR SPRING	CELAS2
BAR/2/n OR MASS	CONM2
BAR/2/n	CDAMP2

LOAD AND CONSTRAINT TRANSLATION:

PCI supports constraint (SPC), nodal force and pressure translation. FORCE and SPC cards are translated in an element-dependent manner as shown in Table (3). Only normal element pressures are supported. PCI will select the appropriate PLOAD card for the element type. In many circumstances involving higher order elements several PLOAD cards must be generated for a single PATRAN pressure load.

TABLE 3: PCI SUPPORT OF LOADS AND SPCS

NASTRAN CARD	USED IN ELEMENTS	SUPPORT
PLOAD4	CQUAD4	YES
PLOAD2	OTHER 1ST ORDER SHELL	YES
PLOAD3	ISOPARAMETRIC SOLID	NOT YET
PLOAD	2ND ORDER SHELL	YES

COORDINATE SYSTEMS:

The various PATRAN coordinate systems are translated to CORD2 cards in NASTRAN, as in the PATNAS translator. Additionally, PATRAN nodes, which are stored in the database in the basic coordinate system, have their locations output to NASTRAN in the local system associated with the nodes. This is of great importance if, for example, constraints are to be applied in a local system.

The PATRAN database includes, associated with each coordinate system, a 3x3 matrix T such that

$$[T] \begin{Bmatrix} x_l \\ y_l \\ z_l \end{Bmatrix} = \begin{Bmatrix} x_b \\ y_b \\ z_b \end{Bmatrix}$$

where the suffixes l, b, denote local and basic coordinate systems respectively. The translator inverts the matrix to produce the matrix T^{-1} where

$$[T]^{-1} \begin{Bmatrix} x_b \\ y_b \\ z_b \end{Bmatrix} = \begin{Bmatrix} x_l \\ y_l \\ z_l \end{Bmatrix}$$

and prints the local values to the NASTRAN data deck.

PROPERTY GENERATION:

It is generally no more difficult to enter property and material cards in NASTRAN that in PATRAN. For this reason property generation has not been implemented in PCI.

CONCLUSIONS:

The amount of programming required to develop a PATRAN-NASTRAN translator was surprisingly small. The approach taken produced a highly flexible translator comparable with the PATNAS translator and superior to the PATCOS translator. The coding required varied from around ten lines for a shell element to around thirty for a bar element, and the time required to add a feature to the program is typically less than an hour. The use of a lookup table for element names makes the translator also applicable to other versions of NASTRAN. The saving in time as a result of using PDA's Gateway utilities was considerable.

During the writing of the program it became apparent that, with a somewhat more complex structure, it would be possible to extend the element data file to contain all data required to define the translation from PATRAN to NASTRAN by mapping of data between formats. Similar data files on property, material and grid formats would produce a completely universal translator from PATRAN to any FEA program, or indeed any CAE system.