MODELING STRATEGY AND SEMI-AUTOMATED CONNECTIVITY CHECKING FOR LARGE SIZE FINITE ELEMENT MODEL

BY

L. I. NAGY, PRINCIPAL RESEARCH ENGINEER

G. C. CAMPBELL, PRODUCT DESIGN ENGINEER A

FORD MOTOR CO., LIGHT TRUCK DESIGN ANALYSIS

PRESENTED AT THE

MSC/NASTRAN CONFERENCE ON

FINITE ELEMENT METHODS AND TECHNOLOGY

PASADENA, CALIFORNIA

MARCH 19 & 20, 1981

ABSTRACT

This paper outlines modeling strategy for finite element analysis of large size problem. A semi-automated method is presented to assembly component models, tear down over-all model into arbitrary sub-assemblies or Super Elements, and verify the correctness of connectivities. The utilization of the method is demonstrated throughout a simple example problem. The listing of the employed computer codes are also included.

INTRODUCTION

Analyzing large size finite element problems at Ford Motor Company began in 1973, using NASTRAN substructuring analysis capability.(1) The need to automate the cumbersome substructuring techniques became obvious,(2) thus, MSC developed the automated version of Substructuring called Super Element analysis capability(3) in 1976 and made it available for general use in 1977. Since then, Super Element analysis capability has been intensively used at Ford to solve large size finite element problems. Super Element capability was found to be a better and a more satisfactory solution approach by Ford as well

---

Numbers in parentheses designate references at end of paper.

as by other companies.[4] However, it still requires tedious modeling and model debugging procedures. There are many different computerized systems available to automate component modeling and model debugging, but these systems usually do not provide meaningful capabilities to properly assemble component models and/or disassemble the overall model into Super Elements. Errors made in component connectivity are not easily detectable, and uncorrected connectivity error could lead to wrong conclusions.

A semi-automated procedure was developed and utilized to reduce connectivity errors and make the assembly and disassembly of component models easier. In the following, this procedure along with modeling strategy will be discussed in detail.

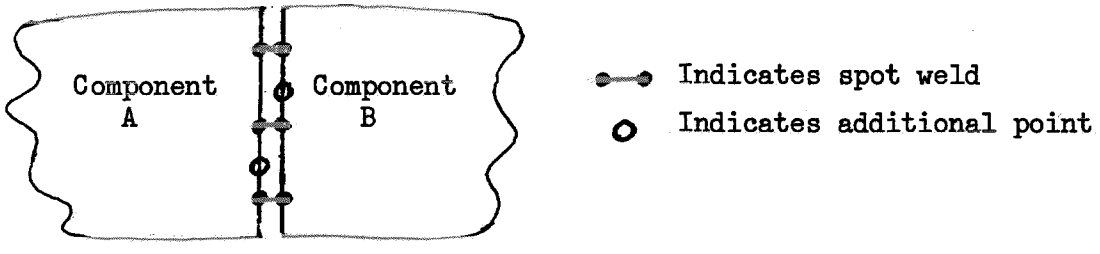
MODELING STRATEGY

1. Define the targeted overall model size. To do that, considering the following may be helpful:
   - Type of analysis required - static analysis usually necessitates finer model than dynamic analysis. Is the model intended to perform only one or both types of analyses?
   - Type of loading and boundary to be applied. Depending on these, one may select a half, full or mirror image model.
   - Available time span to perform the analysis. The best results may become meaningless if not obtained in time to effect design modifications. If possible, provide time for reanalyzing design changes.
   - Available manpower, level of expertise and equipment. Modeling time can be considerably reduced by utilizing more skilled manpower, provided equipment is also available.

MODELING STRATEGY (Cont'd.)

- Allocated budget for the analysis. That could be a serious constraint if the analysis is performed on outside computer.

2. Define model size for each component. This can be accomplished by distributing the targeted overall model size to the components. Do not lose sight of these in the component modeling stage.
3. Allocate a block of sequence numbers for each component. The sequence number range should be greater than the estimated model size. Gaps will not cause any problem but overlapping numbers will.
4. Mark and identify mandatory grid points (MGP) for the entire structure. Load, constraint, and connectivity points are considered to be mandatory points. To model spot weld, rivet or bolt connections, additional grid point must be defined at least for one of the components as shown below:



These additional points must also be marked and identified on all components.

MODELING COMPONENTS

Digitize the MGP's along with all other grid points required for the component mesh, then define element connectivity using available computerized modeling system. Debug each component model for holes, double elements, reverse normals, planarities, etc., in the usual way. Define unique element PID's for each component.

ASSEMBLY OF COMPONENT MODELS

Create an overall bulk data file containing all the unconnected component models. Since each component model has a block of unique sequence numbers, the connectivity points between two adjacent parts will also have different

ASSEMBLY OF COMPONENT MODELS (Cont'd.)

sequence numbers but same coordinates. Sort the entire bulk data for common coordinates, within a user specified limit and list them (DDUPLG*). Select one grid ID for all common coordinates and replace the others with it in both on grid cards and in the element connectivity table (SUBSTG*), and finally eliminate duplicate grid numbers (DDUPLN*). At this point, all component models should be correctly connected. But are they? How can it be checked? This will be discussed next.

CHECKING CONNECTIVITY OF COMPONENTS

Sort (ISOPID*) the overall bulk data file based on element PID's and group them together. Bring these and the associated grid points into local files. The overall model is now subdivided into components, subsystems, or Super Elements as defined by the user. Combine two or more local files and define duplicate grid numbers (DDUPLN). Separate the duplicate grid listing into "Include" and "Exclude" lists (WELDPTS*). Read the overall bulk data file and the "Include" and "Exclude" lists and write a NASTRAN plot set card (SETSPID*). This plot set will plot the mesh according to PID selection and numbers only the connectivity points. This procedure can be used to check components connectivity as well as Super Element connectivity. The codes used in this procedure were developed on the "as needed" base. The codings could be and should be improved to increase efficiency. But the procedure's main advantage is: "IT WORKS".

CLOSURE

The codes described in the foregoing only represent a small segment of the special purpose codes utilized in the modeling phase. As the library of small, special purpose codes expand, it becomes evident that a "better, more efficient way" must be found. A system, which allows direct access to the many functions

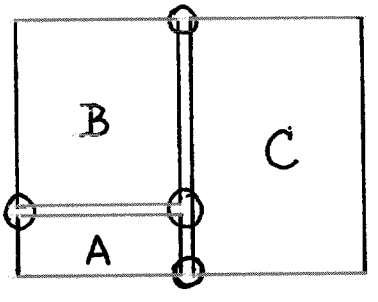
*Indicates in-house written computer code described and listed in the Appendix.

CLOSURE (Cont'd.)

of these small programs and provides easy expansion to include new functions would be ideal.

EXAMPLE PROBLEM

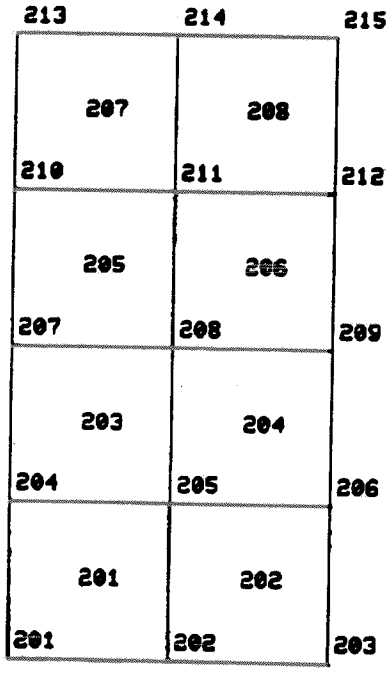
Although, the usefulness of the method presented grows with the problem size, to demonstrate it, a small example problem composed of three parts A, B, and C was selected. These parts are spot welded together as indicated by circles in Figure 1. By marking the connectivity on each drawing and

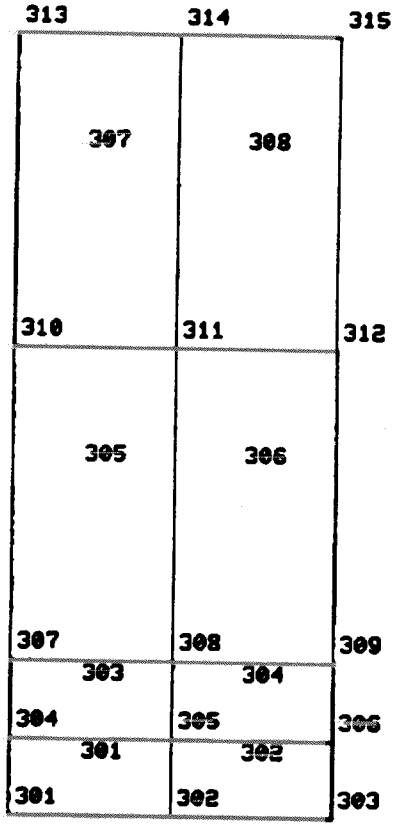


Parts	Allocated Sequence Numbers
A	100 - 190
B	200 - 290
C	300 - 390

Figure 1: Three Components Welded Together

allocating non-overlapping sequence numbers, each component can now be modeled independently of the others. The component models are shown in Figure 2. The model assembly procedure flow chart is shown in Figure 3. The connected over-all model is displayed in Figure 4. The connectivity verification procedure flow chart is shown in Figure 5, and the resulting plots are displayed in Figure 6. The steps in details are given in Appendix"B" according to Figures 3 and 5.





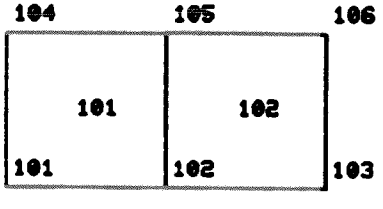
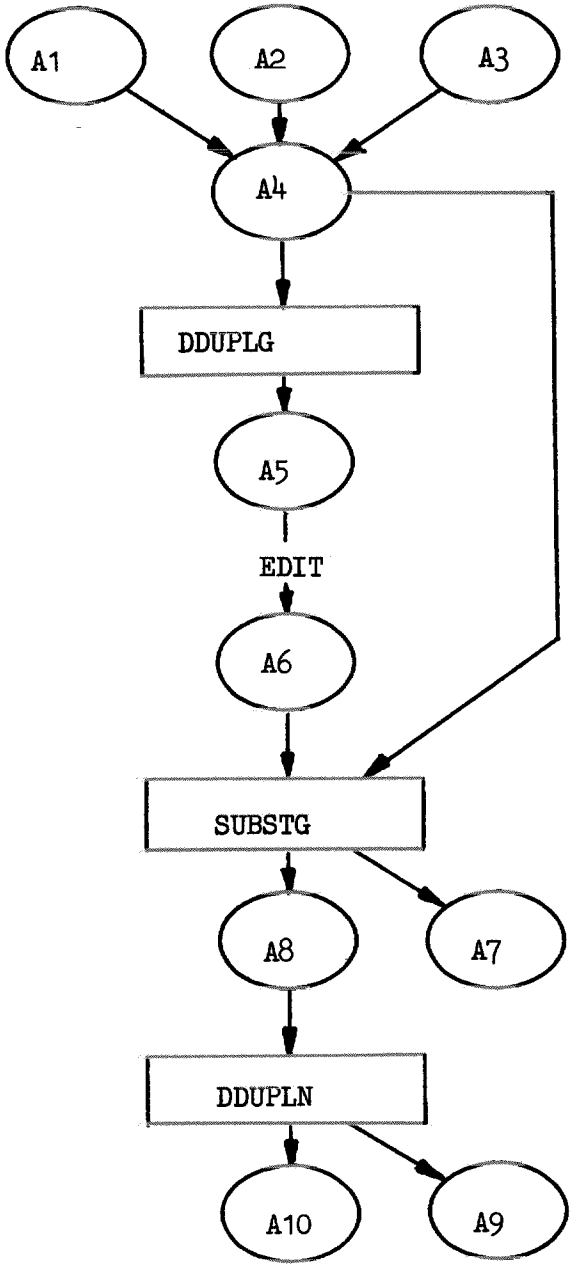


Figure 2: Component Models



FILE DESCRIPTION

A1, A2, A3 = Component Models

A4 = Unconnected Over-All Model

A5 = List of Coincident Grids

A6 = Edited A5

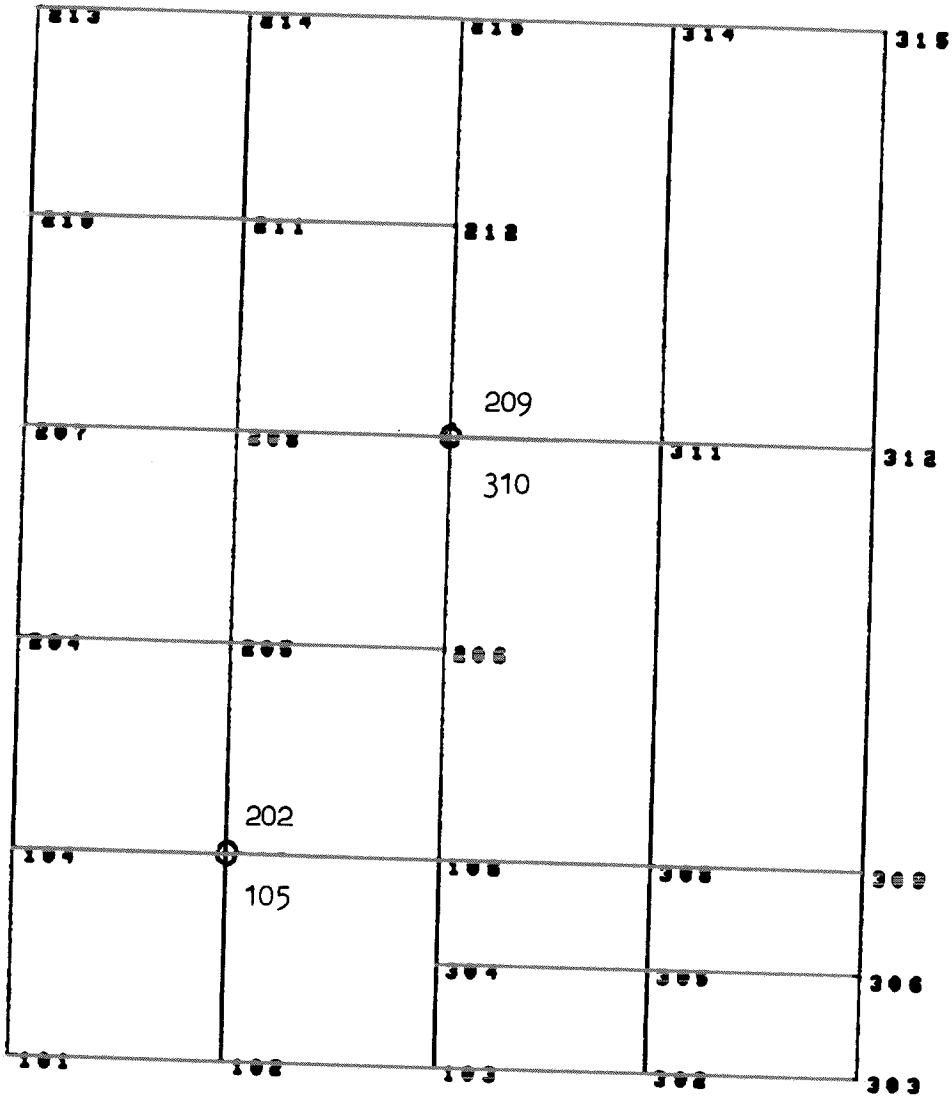
A7 = Cross Reference New to Old Grid List

A8 = New Bulk Data with Duplicate Grid Numbers

A9 = List of Duplicate Grid Numbers

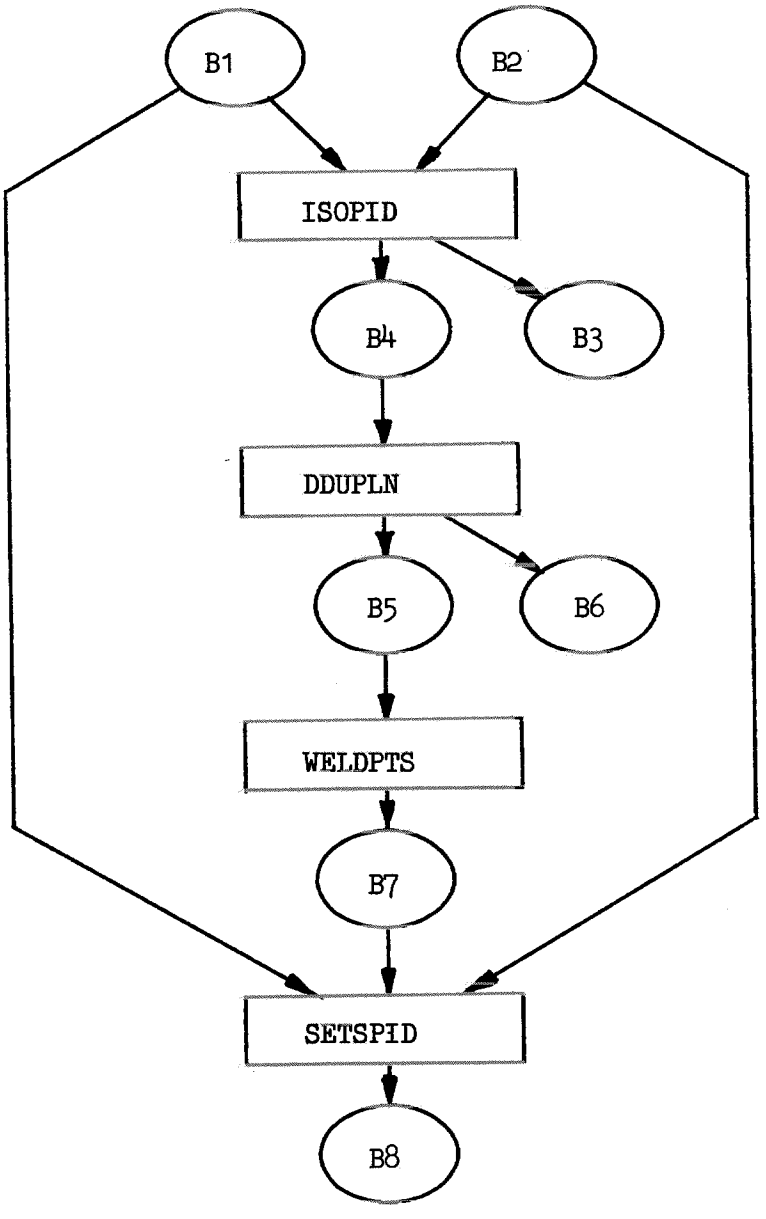
A10 = Connected Over-All Model

Figure 3: Flow Chart of Assembly Procedure



Circle indicates coincident but unconnected grids

Figure 4: Resultant Model



FILE DESCRIPTION

B1 = List of PID's

B2 = Same as A10

B3 = Information

B4 = Component Models

B5 = List of Duplicate Grid Numbers

B6 = Connected Over-All Model

B7 = Reformated Duplicate Grid List (Include, Exclude)

B8 = NASTRAN Pot Set Cards

Figure 5: Flow Chart of Connectivity Verification

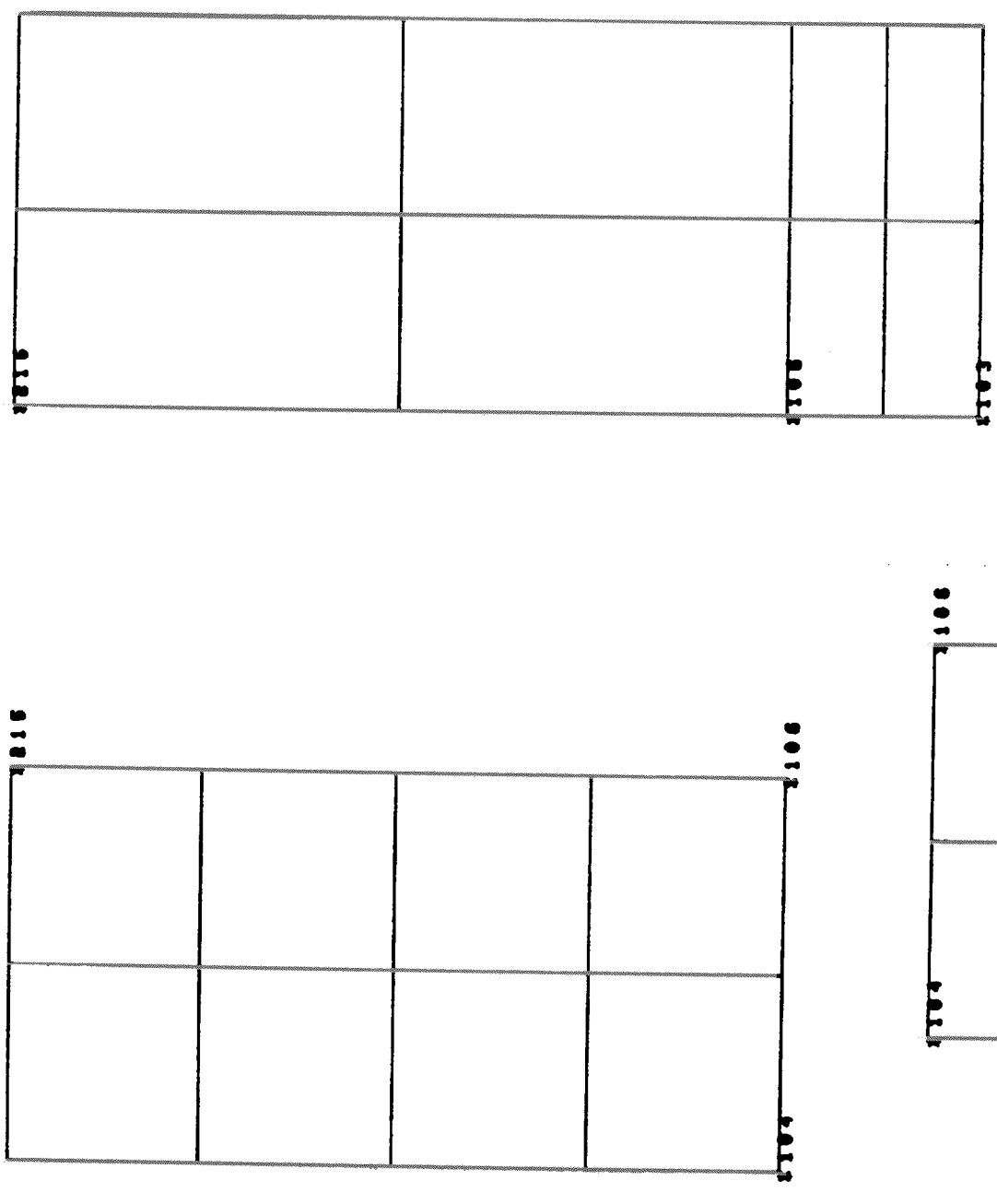


Figure 6: Connectivity Plots

ACKNOWLEDGEMENTS

The authors would like to express their gratitude to Messrs. J. Brancheau, V. Philippopoulos and M. Zebrowski for the help given in developing this procedure and some of the utilized computer codes. Mr. R. Palm's effort to execute and document the example problem is also appreciated.

REFERENCES

1. L. I. Nagy, "Static Analysis via Substructuring of an Experimental Vehicle Front-End Body Structure", Proccedings of International Conference on Vehicle Structural Mechanics, Detroit, 1974.

2. C. W. Hannrich and E. J. Konrath, "Substructure Analysis Techniques and Automation", 3rd Colloquium of NASTRAN User's Experiences, NASA Technical Memorandum, NASA TM X-2893, Langley, 1973.

3. J. A. Joseph, Editor, "MSC/NASTRAN Application Manual, Vol. I.," The MacNeal-Schwendler Corp., Los Angeles.

4. D. W. Lawson, J. L. Webster, F. D. Overbye, "MSC/NASTRAN Superelements in Structures with Reflective Symmetry", MSC/NASTRAN Conference on Finite Element Methods and Technology, Pasadeana, 1980.

APPENDIX A

1. Description of Computer Codes:

DDUPLG - defines and lists grid cards having the same coordinates.

SUBSTG - reads the list of grid cards provided by DDUPLG, selects one grid number for all of the common grid coordinates, and replaces the other grid numbers in the element connectivity card as well as on the grid card. Lists duplicate grid numbers as many times as they occur.

DDUPLN - reads a bulk data file containing multiple grid cards with the same ID and rewrites the bulk data with each grid ID listed only once. Also provides a list of the duplicate grid ID's.

ISOPID - regroups elements according to PID.

WELDPTS - reads the list from DDUPLN and separates it into an "Include" and "Exclude" set.

SETSPID - reads bulk data file and WELDPTS output and writes NASTRAN PLOT SET cards to plot mesh defined by PID and label connectivity grid points.

2. Listings of I/O Files Used in the Example

A1 File

```
100=$---PID 101---
110=CQUAD4        101     101      101      102      105      104
120=CQUAD4        102     101      102      103      106      105
130=GRID          101       0  0.00000  0.00000  0.00000        0
140=GRID          102       0  100.000  0.00000  0.00000        0
150=GRID          103       0  200.000  0.00000  0.00000        0
160=GRID          104       0  0.00000  100.000  0.00000        0
170=GRID          105       0  100.000  100.000  0.00000        0
180=GRID          106       0  200.000  100.000  0.00000        0
```

A2 File

```
100=$---PID 201---
110=CQUAD4        201     201      201      202      205      204
120=CQUAD4        202     201      202      203      206      205
130=CQUAD4        203     201      204      205      208      207
140=CQUAD4        204     201      205      206      209      208
150=CQUAD4        205     201      207      208      211      210
160=CQUAD4        206     201      208      209      212      211
170=CQUAD4        207     201      210      211      214      213
180=CQUAD4        208     201      211      212      215      214
190=GRID          201       0  0.00000  100.000  0.00000        0
200=GRID          202       0  100.000  100.000  0.00000        0
210=GRID          203       0  200.000  100.000  0.00000        0
220=GRID          204       0  0.00000  200.000  0.00000        0
230=GRID          205       0  100.000  200.000  0.00000        0
240=GRID          206       0  200.000  200.000  0.00000        0
250=GRID          207       0  0.00000  300.000  0.00000        0
260=GRID          208       0  100.000  300.000  0.00000        0
270=GRID          209       0  200.000  300.000  0.00000        0
280=GRID          210       0  0.00000  400.000  0.00000        0
290=GRID          211       0  100.000  400.000  0.00000        0
300=GRID          212       0  200.000  400.000  0.00000        0
310=GRID          213       0  0.00000  500.000  0.00000        0
320=GRID          214       0  100.000  500.000  0.00000        0
330=GRID          215       0  200.000  500.000  0.00000        0
```

A3 File

```
100=$---PID 301---
110=CQUAD4        301     301      304      301      302      305
120=CQUAD4        302     301      305      302      303      306
130=CQUAD4        303     301      307      304      305      308
140=CQUAD4        304     301      308      305      306      309
150=CQUAD4        305     301      310      307      308      311
160=CQUAD4        306     301      311      308      309      312
170=CQUAD4        307     301      313      310      311      314
180=CQUAD4        308     301      314      311      312      315
190=GRID          301       0  200.000  0.00000  0.00000        0
200=GRID          302       0  300.000  0.00000  0.00000        0
210=GRID          303       0  400.000  0.00000  0.00000        0
220=GRID          304       0  200.000  50.0000  0.00000        0
230=GRID          305       0  300.000  50.0000  0.00000        0
240=GRID          306       0  400.000  50.0000  0.00000        0
250=GRID          307       0  200.000  100.000  0.00000        0
260=GRID          308       0  300.000  100.000  0.00000        0
270=GRID          309       0  400.000  100.000  0.00000        0
280=GRID          310       0  200.000  300.000  0.00000        0
290=GRID          311       0  300.000  300.000  0.00000        0
300=GRID          312       0  400.000  300.000  0.00000        0
310=GRID          313       0  200.000  500.000  0.00000        0
320=GRID          314       0  300.000  500.000  0.00000        0
330=GRID          315       0  400.000  500.000  0.00000        0
```

A4 File

```
100=$---PID 101---
110=CQUAD4        101     101      101      102      105      104
120=CQUAD4        102     101      102      103      106      105
130=GRID          101       0  0.00000  0.00000  0.00000        0
140=GRID          102       0  100.000  0.00000  0.00000        0
150=GRID          103       0  200.000  0.00000  0.00000        0
160=GRID          104       0  0.00000  100.000  0.00000        0
170=GRID          105       0  100.000  100.000  0.00000        0
180=GRID          106       0  200.000  100.000  0.00000        0
190=$---PID 201---
200=CQUAD4        201     201      201      202      205      204
210=CQUAD4        202     201      202      203      206      205
220=CQUAD4        203     201      204      205      208      207
230=CQUAD4        204     201      205      206      209      208
240=CQUAD4        205     201      207      208      211      210
250=CQUAD4        206     201      208      209      212      211
260=CQUAD4        207     201      210      211      214      213
270=CQUAD4        208     201      211      212      215      214
280=GRID          201       0  0.00000  100.000  0.00000        0
290=GRID          202       0  100.000  100.000  0.00000        0
300=GRID          203       0  200.000  100.000  0.00000        0
310=GRID          204       0  0.00000  200.000  0.00000        0
320=GRID          205       0  100.000  200.000  0.00000        0
330=GRID          206       0  200.000  200.000  0.00000        0
340=GRID          207       0  0.00000  300.000  0.00000        0
350=GRID          208       0  100.000  300.000  0.00000        0
360=GRID          209       0  200.000  300.000  0.00000        0
370=GRID          210       0  0.00000  400.000  0.00000        0
380=GRID          211       0  100.000  400.000  0.00000        0
390=GRID          212       0  200.000  400.000  0.00000        0
400=GRID          213       0  0.00000  500.000  0.00000        0
410=GRID          214       0  100.000  500.000  0.00000        0
420=GRID          215       0  200.000  500.000  0.00000        0
430=$---PID 301---
440=CQUAD4        301     301      304      301      302      305
450=CQUAD4        302     301      305      302      303      306
460=CQUAD4        303     301      307      304      305      308
470=CQUAD4        304     301      308      305      306      309
480=CQUAD4        305     301      310      307      308      311
490=CQUAD4        306     301      311      308      309      312
500=CQUAD4        307     301      313      310      311      314
510=CQUAD4        308     301      314      311      312      315
520=GRID          301       0  200.000  0.00000  0.00000        0
530=GRID          302       0  300.000  0.00000  0.00000        0
540=GRID          303       0  400.000  0.00000  0.00000        0
550=GRID          304       0  200.000  50.0000  0.00000        0
560=GRID          305       0  300.000  50.0000  0.00000        0
570=GRID          306       0  400.000  50.0000  0.00000        0
580=GRID          307       0  200.000  100.000  0.00000        0
590=GRID          308       0  300.000  100.000  0.00000        0
600=GRID          309       0  400.000  100.000  0.00000        0
610=GRID          310       0  200.000  300.000  0.00000        0
620=GRID          311       0  300.000  300.000  0.00000        0
630=GRID          312       0  400.000  300.000  0.00000        0
640=GRID          313       0  200.000  500.000  0.00000        0
650=GRID          314       0  300.000  500.000  0.00000        0
660=GRID          315       0  400.000  500.000  0.00000        0
```

Run DDUPLG, A4, A5

A5 File

```
100=1     36 GRIDS WILL BE PROCESSED
110= GRID COORDS.    301 DUPLICATE TO     103
120= GRID COORDS.    201 DUPLICATE TO     104
130= GRID COORDS.    202 DUPLICATE TO     105
140= GRID COORDS.    203 DUPLICATE TO     106
150= GRID COORDS.    307 DUPLICATE TO     106
160= GRID COORDS.    307 DUPLICATE TO     203
170= GRID COORDS.    310 DUPLICATE TO     209
180= GRID COORDS.    313 DUPLICATE TO     215
```

Manually delete connections that are not to be made i.e. line 130 and 170

A6 File

```
100=1     36 GRIDS WILL BE PROCESSED
110= GRID COORDS.    301 DUPLICATE TO     103
120= GRID COORDS.    201 DUPLICATE TO     104
140= GRID COORDS.    203 DUPLICATE TO     106
150= GRID COORDS.    307 DUPLICATE TO     106
160= GRID COORDS.    307 DUPLICATE TO     203
180= GRID COORDS.    313 DUPLICATE TO     215
```

Run SUBSTG, A6, A4, A7, A8

A7 File

```
100=$$     36 GRIDS WILL BE PROCESSED
110=$$       18  ELEMENTS PROCESSED
120=  NORMAL TERMINATION
130=
140=1                        NEW TO OLD GRID LIST
150=
160=
170=    101    101    102    102    103    103    104    104    105    105    106
106    104    201    202    202    106    203    204    204
180=    205    205    206    206    207    207    208    208    209    209    210
210    211    211    212    212    213    213    214    214
190=    215    215    103    301    302    302    303    303    304    304    305
305    306    306    106    307    308    308    309    309
200=    310    310    311    311    312    312    215    313    314    314    315
315
```

A8 File

```
100=$---PID 101---
110=CQUAD4        101     101      101      102      105      104
120=CQUAD4        102     101      102      103      106      105
130=GRID          101       0  0.00000  0.00000  0.00000        0
140=GRID          102       0  100.000  0.00000  0.00000        0
150=GRID          103       0  200.000  0.00000  0.00000        0
160=GRID          104       0  0.00000  100.000  0.00000        0
170=GRID          105       0  100.000  100.000  0.00000        0
180=GRID          106       0  200.000  100.000  0.00000        0
190=$---PID 201---
200=CQUAD4        201     201      104      202      205      204
210=CQUAD4        202     201      202      106      206      205
220=CQUAD4        203     201      204      205      208      207
230=CQUAD4        204     201      205      206      209      208
240=CQUAD4        205     201      207      208      211      210
250=CQUAD4        206     201      208      209      212      211
260=CQUAD4        207     201      210      211      214      213
270=CQUAD4        208     201      211      212      215      214
280=GRID          104       0  0.00000  100.000  0.00000        0
290=GRID          202       0  100.000  100.000  0.00000        0
300=GRID          106       0  200.000  100.000  0.00000        0
310=GRID          204       0  0.00000  200.000  0.00000        0
320=GRID          205       0  100.000  200.000  0.00000        0
330=GRID          206       0  200.000  200.000  0.00000        0
340=GRID          207       0  0.00000  300.000  0.00000        0
350=GRID          208       0  100.000  300.000  0.00000        0
360=GRID          209       0  200.000  300.000  0.00000        0
370=GRID          210       0  0.00000  400.000  0.00000        0
380=GRID          211       0  100.000  400.000  0.00000        0
390=GRID          212       0  200.000  400.000  0.00000        0
400=GRID          213       0  0.00000  500.000  0.00000        0
410=GRID          214       0  100.000  500.000  0.00000        0
420=GRID          215       0  200.000  500.000  0.00000        0
430=$---PID 301---
440=CQUAD4        301     301      304      103      302      305
450=CQUAD4        302     301      305      302      303      306
460=CQUAD4        303     301      106      304      305      308
470=CQUAD4        304     301      308      305      306      309
480=CQUAD4        305     301      310      106      308      311
490=CQUAD4        306     301      311      308      309      312
500=CQUAD4        307     301      215      310      311      314
510=CQUAD4        308     301      314      311      312      315
520=GRID          103       0  200.000  0.00000  0.00000        0
530=GRID          302       0  300.000  0.00000  0.00000        0
540=GRID          303       0  400.000  0.00000  0.00000        0
550=GRID          304       0  200.000  50.0000  0.00000        0
560=GRID          305       0  300.000  50.0000  0.00000        0
570=GRID          306       0  400.000  50.0000  0.00000        0
580=GRID          106       0  200.000  100.000  0.00000        0
590=GRID          308       0  300.000  100.000  0.00000        0
600=GRID          309       0  400.000  100.000  0.00000        0
610=GRID          310       0  200.000  300.000  0.00000        0
620=GRID          311       0  300.000  300.000  0.00000        0
630=GRID          312       0  400.000  300.000  0.00000        0
640=GRID          215       0  200.000  500.000  0.00000        0
650=GRID          314       0  300.000  500.000  0.00000        0
660=GRID          315       0  400.000  500.000  0.00000        0
```

Run DDUPLN, A8, A9, A10

A9 File

```
100=1     36 GRIDS WILL BE PROCESSED
110=1           5  DUPLICATE NODES HAVE BEEN DELETED
120=
130=
140= PASS NO.     7  DELETED NODE        104
150= PASS NO.     9  DELETED NODE        106
160= PASS NO.    22  DELETED NODE        103
170= PASS NO.    28  DELETED NODE        106
180= PASS NO.    34  DELETED NODE        215
```

A10 File

```
100=$---PID 101---
110=CQUAD4        101     101     101     102     105     104
120=CQUAD4        102     101     102     103     106     105
130=GRID          101       0 0.00000 0.00000 0.00000       0
140=GRID          102       0 100.000 0.00000 0.00000       0
150=GRID          103       0 200.000 0.00000 0.00000       0
160=GRID          104       0 0.00000 100.000 0.00000       0
170=GRID          105       0 100.000 100.000 0.00000       0
180=GRID          106       0 200.000 100.000 0.00000       0
190=$---PID 201---
200=CQUAD4        201     201     104     202     205     204
210=CQUAD4        202     201     202     106     206     205
220=CQUAD4        203     201     204     205     208     207
230=CQUAD4        204     201     205     206     209     208
240=CQUAD4        205     201     207     208     211     210
250=CQUAD4        206     201     208     209     212     211
260=CQUAD4        207     201     210     211     214     213
270=CQUAD4        208     201     211     212     215     214
280=GRID          202       0 100.000 100.000 0.00000       0
290=GRID          204       0 0.00000 200.000 0.00000       0
300=GRID          205       0 100.000 200.000 0.00000       0
310=GRID          206       0 200.000 200.000 0.00000       0
320=GRID          207       0 0.00000 300.000 0.00000       0
330=GRID          208       0 100.000 300.000 0.00000       0
340=GRID          209       0 200.000 300.000 0.00000       0
350=GRID          210       0 0.00000 400.000 0.00000       0
360=GRID          211       0 100.000 400.000 0.00000       0
370=GRID          212       0 200.000 400.000 0.00000       0
380=GRID          213       0 0.00000 500.000 0.00000       0
390=GRID          214       0 100.000 500.000 0.00000       0
400=GRID          215       0 200.000 500.000 0.00000       0
410=$---PID 301---
420=CQUAD4        301     301     304     103     302     305
430=CQUAD4        302     301     305     302     303     306
440=CQUAD4        303     301     106     304     305     308
450=CQUAD4        304     301     308     305     306     309
460=CQUAD4        305     301     310     106     308     311
470=CQUAD4        306     301     311     308     309     312
480=CQUAD4        307     301     215     310     311     314
490=CQUAD4        308     301     314     311     312     315
500=GRID          302       0 300.000 0.00000 0.00000       0
510=GRID          303       0 400.000 0.00000 0.00000       0
520=GRID          304       0 200.000 50.0000 0.00000       0
530=GRID          305       0 300.000 50.0000 0.00000       0
540=GRID          306       0 400.000 50.0000 0.00000       0
550=GRID          308       0 300.000 100.000 0.00000       0
560=GRID          309       0 400.000 100.000 0.00000       0
570=GRID          310       0 200.000 300.000 0.00000       0
580=GRID          311       0 300.000 300.000 0.00000       0
590=GRID          312       0 400.000 300.000 0.00000       0
600=GRID          314       0 300.000 500.000 0.00000       0
610=GRID          315       0 400.000 500.000 0.00000       0
```

File listings for connectivity verification

Reference Figure 5

B1 File

```
100=101
110=201
120=301
```

B2 File is the bulk data file, in this example the file A10 was used.

Run ISOPID, B1, B2, B3, B4

B3 File

```
100=1 LIST OF PID'S  TO SORT BY
110=      101
120= NO. OF ELEMENTS PROCESSED =       2
130= NO. OF GRIDS IN CHOSEN SET =       6
140=1 LIST OF PID'S  TO SORT BY
150=      201
160= NO. OF ELEMENTS PROCESSED =       8
170= NO. OF GRIDS IN CHOSEN SET =      15
180=1 LIST OF PID'S  TO SORT BY
190=      301
200= NO. OF ELEMENTS PROCESSED =       8
210= NO. OF GRIDS IN CHOSEN SET =      15
```

B4 File

```
100=CQUAD4      101     101     101     102     105     104
110=CQUAD4      102     101     102     103     106     105
120=GRID        101       0 0.00000 0.00000 0.00000       0
130=GRID        102       0 100.000 0.00000 0.00000       0
140=GRID        103       0 200.000 0.00000 0.00000       0
150=GRID        104       0 0.00000 100.000 0.00000       0
160=GRID        105       0 100.000 100.000 0.00000       0
170=GRID        106       0 200.000 100.000 0.00000       0
180=$EOR
190=$EOF                                                    Delete the end of file mark
200=CQUAD4      201     201     104     202     205     204
210=CQUAD4      202     201     202     106     206     205
220=CQUAD4      203     201     204     205     208     207
230=CQUAD4      204     201     205     206     209     208
240=CQUAD4      205     201     207     208     211     210
250=CQUAD4      206     201     208     209     212     211
260=CQUAD4      207     201     210     211     214     213
270=CQUAD4      208     201     211     212     215     214
280=GRID        104       0 0.00000 100.000 0.00000       0
290=GRID        106       0 200.000 100.000 0.00000       0
300=GRID        202       0 100.000 100.000 0.00000       0
310=GRID        204       0 0.00000 200.000 0.00000       0
320=GRID        205       0 100.000 200.000 0.00000       0
330=GRID        206       0 200.000 200.000 0.00000       0
340=GRID        207       0 0.00000 300.000 0.00000       0
350=GRID        208       0 100.000 300.000 0.00000       0
360=GRID        209       0 200.000 300.000 0.00000       0
370=GRID        210       0 0.00000 400.000 0.00000       0
380=GRID        211       0 100.000 400.000 0.00000       0
390=GRID        212       0 200.000 400.000 0.00000       0
400=GRID        213       0 0.00000 500.000 0.00000       0
410=GRID        214       0 100.000 500.000 0.00000       0
420=GRID        215       0 200.000 500.000 0.00000       0
430=$EOR
440=$EOF                                                    Delete the end of file mark
450=CQUAD4      301     301     304     103     302     305
460=CQUAD4      302     301     305     302     303     306
470=CQUAD4      303     301     106     304     305     308
480=CQUAD4      304     301     308     305     306     309
490=CQUAD4      305     301     310     106     308     311
500=CQUAD4      306     301     311     308     309     312
510=CQUAD4      307     301     215     310     311     314
520=CQUAD4      308     301     314     311     312     315
530=GRID        103       0 200.000 0.00000 0.00000       0
540=GRID        106       0 200.000 100.000 0.00000       0
550=GRID        215       0 200.000 500.000 0.00000       0
560=GRID        302       0 300.000 0.00000 0.00000       0
570=GRID        303       0 400.000 0.00000 0.00000       0
580=GRID        304       0 200.000 50.0000 0.00000       0
590=GRID        305       0 300.000 50.0000 0.00000       0
600=GRID        306       0 400.000 50.0000 0.00000       0
610=GRID        308       0 300.000 100.000 0.00000       0
620=GRID        309       0 400.000 100.000 0.00000       0
630=GRID        310       0 200.000 300.000 0.00000       0
640=GRID        311       0 300.000 300.000 0.00000       0
650=GRID        312       0 400.000 300.000 0.00000       0
660=GRID        314       0 300.000 500.000 0.00000       0
670=GRID        315       0 400.000 500.000 0.00000       0
```

Run DDUPLN, B4, B5, B6

B5 File

```
100=1     36 GRIDS WILL BE PROCESSED
110=1           5  DUPLICATE NODES HAVE BEEN DELETED
120=
130=
140= PASS NO.     7   DELETED NODE        104
150= PASS NO.     8   DELETED NODE        106
160= PASS NO.    22   DELETED NODE        103
170= PASS NO.    23   DELETED NODE        106
180= PASS NO.    24   DELETED NODE        215
```

B6 File

```
100=CQUAD4      101     101       101       102       105       104
110=CQUAD4      102     101       102       103       106       105
120=GRID        101       0   0.00000   0.00000   0.00000         0
130=GRID        102       0   100.000   0.00000   0.00000         0
140=GRID        103       0   200.000   0.00000   0.00000         0
150=GRID        104       0   0.00000   100.000   0.00000         0
160=GRID        105       0   100.000   100.000   0.00000         0
170=GRID        106       0   200.000   100.000   0.00000         0
180=CQUAD4      201     201       104       202       205       204
190=CQUAD4      202     201       202       106       206       205
200=CQUAD4      203     201       204       205       208       207
210=CQUAD4      204     201       205       206       209       208
220=CQUAD4      205     201       207       208       211       210
230=CQUAD4      206     201       208       209       212       211
240=CQUAD4      207     201       210       211       214       213
250=CQUAD4      208     201       211       212       215       214
260=GRID        202       0   100.000   100.000   0.00000         0
270=GRID        204       0   0.00000   200.000   0.00000         0
280=GRID        205       0   100.000   200.000   0.00000         0
290=GRID        206       0   200.000   200.000   0.00000         0
300=GRID        207       0   0.00000   300.000   0.00000         0
310=GRID        208       0   100.000   300.000   0.00000         0
320=GRID        209       0   200.000   300.000   0.00000         0
330=GRID        210       0   0.00000   400.000   0.00000         0
340=GRID        211       0   100.000   400.000   0.00000         0
350=GRID        212       0   200.000   400.000   0.00000         0
360=GRID        213       0   0.00000   500.000   0.00000         0
370=GRID        214       0   100.000   500.000   0.00000         0
380=GRID        215       0   200.000   500.000   0.00000         0
390=CQUAD4      301     301       304       103       302       305
400=CQUAD4      302     301       305       302       303       306
410=CQUAD4      303     301       106       304       305       308
420=CQUAD4      304     301       308       305       306       309
430=CQUAD4      305     301       310       106       308       311
440=CQUAD4      306     301       311       308       309       312
450=CQUAD4      307     301       215       310       311       314
460=CQUAD4      308     301       314       311       312       315
470=GRID        302       0   300.000   0.00000   0.00000         0
480=GRID        303       0   400.000   0.00000   0.00000         0
490=GRID        304       0   200.000   50.0000   0.00000         0
500=GRID        305       0   300.000   50.0000   0.00000         0
510=GRID        306       0   400.000   50.0000   0.00000         0
520=GRID        308       0   300.000   100.000   0.00000         0
530=GRID        309       0   400.000   100.000   0.00000         0
540=GRID        310       0   200.000   300.000   0.00000         0
550=GRID        311       0   300.000   300.000   0.00000         0
560=GRID        312       0   400.000   300.000   0.00000         0
570=GRID        314       0   300.000   500.000   0.00000         0
580=GRID        315       0   400.000   500.000   0.00000         0
```

Run WELDPTS, B5, B7

B7 File

```
100= INCLUDE GRID POINTS,
110=   104,  106,  103,  106,  215,
120= EXCLUDE GRID POINTS 1 THRU 90000
```

Run SETSPID, B1, B2, B8, B7

B8 File

```
100=$  SET CARD(S) FOR PID(S)    101
110= SET    105 = ,
120=   101,  102,
130= INCLUDE GRID POINTS,
140=   104,  106,  103,  106,  215,
150= EXCLUDE GRID POINTS 1 THRU 90000
160=$  SET CARD(S) FOR PID(S)    201
170= SET    205 = ,
180=   201,  202,  203,  204,  205,  206,  207,  208,
190= INCLUDE GRID POINTS,
200=   104,  106,  103,  106,  215,
210= EXCLUDE GRID POINTS 1 THRU 90000
220=$  SET CARD(S) FOR PID(S)    301
230= SET    305 = ,
240=   301,  302,  303,  304,  305,  306,  307,  308,
250= INCLUDE GRID POINTS,
260=   104,  106,  103,  106,  215,
270= EXCLUDE GRID POINTS 1 THRU 90000
```

```
ATTACH,SAMPJCL
 PFN IS
 SAMPJCL:
 PF CYCLE NO. = 001
..E,SAMPJCL,S
..L,A

   100-REP,T50,CM250000.
   110-ACCOUNT,REP,####,NK795200,BLDG1.
   120-REQUEST,PLT2,HQ.
   130-ATTACH,BULK,DECK4.
   140-BEGIN,NASTRAN,,DMAP-ALTER,VER-60.
   150-ROUTE,PLT2,DC-PU,FID-PLOTS,TID-EM.
   160-$EOR
   170-$C RF24D32
   180-$EOR
   190-$DECK SAMPLE
   200-NASTRAN FILES=(PLT2)
   210-ID LIN,REP
   220-SOL 24
   230-TIME .5
   240-$READ ALTER
   250-CEND
   260-TITLE=SAMPLE PROBLEM
   270-SUBTITLE=GEOMETRY CHECKOUT.
   280-OUTPUT(PLOT)
   290-CSCALE=1.8
   300-PLOTTER NAST
   310-PAPER SIZE 26 BY 20
   320-SET 1=ALL
   330-$
   340-$  SET CARD(S) FOR PID(S)   101
   350- SET   105 = ,
   360-   101,  102,
   370-  INCLUDE GRID POINTS,
   380-   104,  106,  103,  106,  215,
   390- EXCLUDE GRID POINTS 1 THRU 99999
   400-$  SET CARD(S) FOR PID(S)   201
   410- SET   205 = ,
   420-201,  202,  203,  204,  205,  206,  207,  208,
   430- INCLUDE GRID POINTS,
   440-   104,  106,  103,  106,  215,
   450- EXCLUDE GRID POINTS 1 THRU 99999
   460-$  SET CARD(S) FOR PID(S)   301
   470- SET   305 = ,
   480-   301,  302,  303,  304,  305,  306,  307,  308,
   490- INCLUDE GRID POINTS,
   500-   104,  106,  103,  106,  215,
   510- EXCLUDE GRID POINTS 1 THRU 99999
   520-$
   530-PTITLE=WELD POINTS
   540-AXES=Z,X,Y
   550-VIEW=0.,0.,0.
   560-FIND SCALE 1, ORIGIN 1, SET 1
   570-PLOT SET 1, ORIGIN 1
   580-PLOT SET 105, ORIGIN 1, SYMBOL 2, LABEL GRID POINTS
   590-PLOT SET 205, ORIGIN 1, SYMBOL 2, LABEL GRID POINTS
   600-PLOT SET 305, ORIGIN 1, SYMBOL 2, LABEL GRID POINTS
   610-BEGIN BULK
   620-$READ BULK
   630-PARAM    AUTOSPC     YES
   640-ENDDATA
```

APPENDIX B

LISTINGS OF COMPUTER CODES

```
      PROGRAM DDUPLG (TAPE5,TAPE6,TAPE7)
C**    READS A BULK DATA DECK AND CHECKS FOR ALL DUPLICATE GRIDS
C        WITHIN  R=10E-5 IN ANY PRINCIPAL DIRECTION
C       AND SAME NODE NUMBER, AND DELETES SUCH GRIDS
C
C      TAPE5  : INPUT DATA
C      TAPE6  : OUTPUT INFORMATION
C      TAPE7  : OUTPUT DECK
C
      DIMENSION NPT(2000),LDUPL(2000),X(2000),Y(2000),Z(2000)
      DIMENSION B(8),C(8),D(8),E(8),G(8),AA(8),BB(24)
      DATA GD/4HGRID/
      IOUT=6
      IDAT=5
      R=.00001
      NG=0
      NDUPL=0
      RR=R*R
    1 READ (IDAT,2) A1,A2,B,C,D,E,G,AA,BB
      IF (EOF(IDAT)) 101,3
    2 FORMAT (2(A4),6(8A1),24A1)
    3 IF (A1.NE.GD) GO TO 1
      NG=NG+1
      BACKSPACE IDAT
      READ (IDAT,4) A1,B,X(NG),Y(NG),Z(NG)
    4 FORMAT (A4,4X,8A1,8X,3F8.0)
      NPT(NG)=INCONV(B,8)
      GO TO 1
  101 REWIND IDAT
      MAXN=NG
      WRITE (IOUT,10) MAXN
   10 FORMAT (1H1,I6,25H GRIDS WILL BE PROCESSED )
      MAXNN=MAXN-1
      DO 20 I=1,MAXNN
      II=I+1
      DO 20 J=II,MAXN
      RG=(Z(J)-Z(I))**2
      RG=RG+(Y(J)-Y(I))**2
      RG=RG+(X(J)-X(I))**2
      IF (RG-RR) 19,19,20
19       WRITE (IOUT,21) NPT(J),NPT(I)
         IF (NPT(J).NE.NPT(I)) GO TO 20
         NDUPL=NDUPL+1
         LDUPL(NDUPL)=NPT(J)
   20 CONTINUE
   21 FORMAT (13H GRID COORDS.,I6,13H DUPLICATE TO,I7)
      IF (NDUPL.LE.0) GO TO 102
C
C     REMOVE DUPLICATE NODES
C
      WRITE (IOUT,32) NDUPL
32    FORMAT(1H1,I10,*  DUPLICATE NODES HAVE BEEN DELETED*,//)
      REWIND IDAT
      IDUP=1
      REWIND 7
      CALL BSORT (LDUPL,NDUPL)
31    READ (IDAT,2) A1,A2,B,C,D,E,G,AA,BB
      IF (EOF(IDAT)) 105,33
33    WRITE (7,2) A1,A2,B,C,D,E,G,AA,BB
```

```
      IF (A1.NE.GD) GO TO 31
      NDUP=INCONV(B,8)
      CALL FIND (NDUP,LDUPL,NDUPL,IFOUND)
      IF (IFOUND.GT.0) GO TO 31
      BACKSPACE 7
      WRITE (IOUT,35) IDUP,NDUP
35    FORMAT (9H PASS NO.,I5,14H  DELETED NODE,I10)
      IDUP=IDUP+1
      GO TO 31
105   CONTINUE
102   STOP
      END
      FUNCTION INCONV(A,LF)
C     FORTRAN FUNCTION TO CONVERT DATA READ AS FREE FORMAT
C      ALPHANUMERIC IN A FIELD OF LENGTH LF TO AN INTEGER
C      A(I),I=1,LF   ARE THE ALPHANUMERICS READ IN A LFA1 FORMAT
C      FUNCTION RETURN IS AN INTEGER VALUE
      DIMENSION A(LF),F(10)
      DATA BLANK,F(1),F(2),F(3),F(4),F(5),F(6),F(7),F(8),F(9),F(10)/
     1       1H , 1H1, 1H2, 1H3, 1H4, 1H5, 1H6, 1H7, 1H8, 1H9, 1H0/
      INT=0
      II=LF
      I=LF
   10 IF(A(I).EQ.BLANK) GO TO 100
      IP=LF-II
      DO 20 J=1,9
   20 IF(A(I).EQ.F(J)) INT=INT+J*(10**IP)
      II=II-1
      I=I-1
      IF(I.EQ.0) GO TO 90
      GO TO 10
  100 I=I-1
      IF(I.EQ.0) GO TO 90
      GO TO 10
   90 INCONV=INT
   99 RETURN
              END
C
      SUBROUTINE FIND(ISECTR,IECTR,N,IF)
      DIMENSION IECTR(N)
      IF=-1
      IF (ISECTR.LT.IECTR(1).OR.ISECTR.GT.IECTR(N)) RETURN
      B=0.
      E=N
   10 M=(B+E+1.)/2.
      IF(ISECTR-IECTR(M)) 15,25,20
   15 IF(ISECTR-IECTR(M-1)) 30,35,40
   30 E=M
      GO TO 10
   35 M=M-1
      GO TO 25
   40 RETURN
   20 IF(ISECTR-IECTR(M+1)) 45,50,55
   45 RETURN
   50 M=M+1
      GO TO 25
   55 B=M
      GO TO 10
   25 IF=M
      RETURN
```

```
      END
C
      SUBROUTINE BSORT (LX,N)
C     TO SORT AN ARRAY IN INCREASING ORDER OF CONTENTS
C      BY THE BUBBLE SORT METHOD
      DIMENSION LX(N)
      NN=N-1
      DO 10 I=1,NN
          II=I+1
          DO 10 J=II,N
              IF (LX(I).LE.LX(J)) GO TO 10
              M=LX(I)
              LX(I)=LX(J)
              LX(J)=M
10    CONTINUE
      RETURN
      END
```

```
      PROGRAM SUBSTG (TAPE2,TAPE5,TAPE6,TAPE7)
C
C     THIS PROGRAM TAKES THE COINCINENT GRID FILE FROM DDUPLG
C     AND DELETES ALL EXCEPT ONE OF THE COINCIDENT GRIDS FROM
C     THE BULK DATA FILE.
C     IT THEN MODIFIES THE ELEMENTS TO REFLECT THE
C     GRID CHANGE.
C
C     CURRENT MODIFIABLE ENTRIES ARE :
C     CQUA,CTRI,CBAR,GRID,CHEX,CPEN,RBAR,RBE2
C
C     ALSO DETECTS MISSING GRIDS
C
C     TAPE2  : INPUT FILE FROM DDUPLG
C     TAPE5  : INPUT FILE
C     TAPE6  : OUTPUT INFORMATION
C     TAPE7  : OUTPUT FILE
C
      DIMENSION B(8),C(8),D(8),E(8),G(8),AA(8),BB(24),CHB(8),CHE(8)
      DIMENSION LN(3000),ELEM(8)
      DIMENSION LD(3000),LDUPL(2,3000)
      DATA ELEM/4HCQUA,4HCTRI,4HCBAR,4HGRID,4HCHEX,4HCPEN,4HRBAR,4HRBE2/
      IOUT=6
      REWIND 2
      IDAT=5
      REWIND 5
      REWIND 6
      REWIND 7
      NELLIB=8
C     ARRAY SIZE
      MAXGD=3000
C     DETERMINE THE STARTING NUMBER FOR RESEQUENCED GRIDS
      READ (IDAT,101) B
101   FORMAT (8A1)
      NSTGD=INCONV(B,8)
      IF (NSTGD.LT.1) NSTGD=0
      IF (NSTGD.EQ.0) REWIND IDAT
C     INPUT AND SORT GRIDS
      CALL INPGRD (IDAT,LN,MAXGD,NCNT)
      IF (NCNT.GT.MAXGD) GO TO 103
      WRITE (IOUT,19) NCNT
      CALL BSORT (LN,NCNT)
      REWIND IDAT
C     READ IN DUPLICATE GRID NO'S
      J = 1
      REWIND 2
      READ(2,73)JK
      IF(JK.EQ.2H G)BACKSPACE 2
   73 FORMAT(A2)
  300 READ(2,301)LDUPL(1,J),LDUPL(2,J)
      IF(EOF(2)) 303,302
  301 FORMAT(13X,I6,13X,I7)
  302 J = J + 1
      GO TO 300
  303 NDUPL = J - 1
C      SORT DUPLICATE GRIDS IN ASSENDING ORDER
C      AND ACCOUNT FOR MULTIPLE DUPLICATES
```

```
      DO 999 I=1,NDUPL
      IF(LDUPL(1,I).GE.LDUPL(2,I)) GO TO 999
      M3=LDUPL(1,I)
      LDUPL(1,I)=LDUPL(2,I)
      LDUPL(2,I)=M3
999   CONTINUE
      CALL BBSORT(LDUPL,NDUPL)
      JJ=NDUPL-1
      DO 1000 JK=1,JJ
      JM=JK+1
      DO 1000 JL=JM,NDUPL
      IF(LDUPL(1,JK).NE.LDUPL(1,JL)) GO TO 1000
      LDUPL(1,JL)=0
      LDUPL(2,JL)=0
1000  CONTINUE
C
C     FIX LD ARRAY
      DO 200 I = 1,NCNT
      LD(I) = LN(I)
      DO 201 II = 1,NDUPL
      IF(LN(I).EQ.LDUPL(1,II)) GO TO 202
  201 CONTINUE
      GO TO 200
  202 LD(I) = LDUPL(2,II)
  200 CONTINUE
C      PROCESS ELEMENTS
      NOMIT=-1
      NEL=0
5     NOMIT=NOMIT+1
    1 READ (IDAT,2) A1,A2,B,C,D,E,G,AA,BB
      IF (EOF(IDAT)) 102,3
    2 FORMAT (A4,A4,6(8A1),24A1)
3     NTYPE=0
      DO 4 I=1,NELLIB
          IF (A1.EQ.ELEM(I)) NTYPE=I
4     CONTINUE
      IF (NTYPE.GT.0) GO TO (40,30,20,10,50,50,60,63),NTYPE
      WRITE (7,2) A1,A2,B,C,D,E,G,AA,BB
      GO TO 1
C               CQUADS
40    NODE=INCONV(AA,8)
      CALL NFIND (NODE,LN,NCNT,ND4)
      NODE=INCONV(G,8)
      CALL NFIND (NODE,LN,NCNT,ND3)
      NODE=INCONV(E,8)
      CALL NFIND (NODE,LN,NCNT,ND2)
      NODE=INCONV(D,8)
      CALL NFIND (NODE,LN,NCNT,ND1)
      IF(ND1.LT.1.OR.ND2.LT.1.OR.ND3.LT.1.OR.ND4.LT.1) GO TO 99
      WRITE (7,41) A1,A2,B,C,LD(ND1),LD(ND2),LD(ND3),LD(ND4),BB
41    FORMAT (A4,A4,2(8A1),4I8,24A1)
      NEL=NEL+1
      GO TO 1
C              CTRIANGLES
30    NODE=INCONV(G,8)
      CALL NFIND (NODE,LN,NCNT,ND3)
      NODE=INCONV(E,8)
      CALL NFIND (NODE,LN,NCNT,ND2)
      NODE=INCONV(D,8)
      CALL NFIND (NODE,LN,NCNT,ND1)
```

```
      IF(ND1.LT.1.OR.ND2.LT.1.OR.ND3.LT.1) GO TO 99
      WRITE (7,31) A1,A2,B,C,LD(ND1),LD(ND2),LD(ND3),AA,BB
31    FORMAT (A4,A4,2(8A1),3I8,8A1,24A1)
      NEL=NEL+1
      GO TO 1
C                 CBARS
20    NODE=INCONV(E,8)
      CALL NFIND (NODE,LN,NCNT,ND2)
      NODE=INCONV(D,8)
      CALL NFIND (NODE,LN,NCNT,ND1)
      IF(ND1.LT.1.OR.ND2.LT.1) GO TO 99
      WRITE (7,21) A1,A2,B,C,LD(ND1),LD(ND2),G,AA,BB
21    FORMAT (A4,A4,2(8A1),2I8,8A1,8A1,24A1)
      NEL=NEL+1
      GO TO 1
C                                 SOLIDS
50    NODE=INCONV(D,8)
      CALL NFIND (NODE,LN,NCNT,ND1)
      NODE=INCONV(E,8)
      CALL NFIND (NODE,LN,NCNT,ND2)
      NODE=INCONV(G,8)
      CALL NFIND (NODE,LN,NCNT,ND3)
      NODE=INCONV(AA,8)
      CALL NFIND (NODE,LN,NCNT,ND4)
      NODE=INCONV(BB(1),8)
      CALL NFIND (NODE,LN,NCNT,ND5)
      NODE=INCONV(BB(9),8)
      CALL NFIND (NODE,LN,NCNT,ND6)
      IF(ND1.LT.1.OR.ND2.LT.1.OR.ND3.LT.1.OR.ND4.LT.1.OR.ND5.LT.1.OR.ND6
     1.LT.1) GO TO 97
      WRITE(7,51) A1,A2,B,C,LD(ND1),LD(ND2),LD(ND3),LD(ND4),
     1LD(ND5),LD(ND6),(BB(I),I=17,24)
51    FORMAT (2A4,2(8A1),6I8,8A1)
      NEL=NEL+1
C     DETERMINE IF CONTINUATION FIELD IS PRESENT
52    IF (BB(17).EQ.1H ) GO TO 1
      DO 53 I=1,8
53    CHE(I)=BB(I+16)
      READ (IDAT,56) CHB,B,C,D,E,G,AA,BB
      IF (EOF(IDAT)) 102,531
531   DO 54 I=2,8
      IF (CHB(I).EQ.CHE(I)) GO TO 54
      GO TO 57
54    CONTINUE
      NODE=INCONV(B,8)
      CALL NFIND (NODE,LN,NCNT,ND8)
      NODE=INCONV(C,8)
      CALL NFIND (NODE,LN,NCNT,ND7)
      NODE=INCONV(D,8)
      CALL NFIND (NODE,LN,NCNT,ND1)
      NODE=INCONV(E,8)
      CALL NFIND (NODE,LN,NCNT,ND2)
      NODE=INCONV(G,8)
      CALL NFIND (NODE,LN,NCNT,ND3)
      NODE=INCONV(AA,8)
      CALL NFIND (NODE,LN,NCNT,ND4)
      NODE=INCONV(BB(1),8)
      CALL NFIND (NODE,LN,NCNT,ND5)
      NODE=INCONV(BB(9),8)
      CALL NFIND (NODE,LN,NCNT,ND6)
```

```
      IF(ND1.LT.0.OR.ND2.LT.0.OR.ND3.LT.0.OR.ND4.LT.0.OR.ND5.LT.0.OR.ND6
     1.LT.0.OR.ND7.LT.0.OR.ND8.LT.0) GO TO 96
      WRITE (7,55) CHB,LD(ND8),LD(ND7),LD(ND1),LD(ND2),LD(ND3),
     1LD(ND4),LD(ND5),LD(ND6),(BB(I),I=17,24)
      GO TO 52
55    FORMAT (8A1,8I8,8A1)
56    FORMAT (7(8A1),24A1)
C     READ TOO MUCH,RECTIFY AND CONTINUE
57    BACKSPACE IDAT
      BACKSPACE 7
      READ (7,59) BB,B,C,D,E,G,AA,CHE
      BACKSPACE 7
      WRITE (7,59) BB,B,C,D,E,G,AA
      WRITE (IOUT,58)BB,B,C,D,E,G,AA,CHE
      GO TO 1
58    FORMAT (* THE LINE*,11X,24A1,7(8A1),/,
     1* WAS TRUNCATED TO 72 CHARACTERS*)
59    FORMAT (24A1,7(8A1))
C                                  RBAR
60    NODE=INCONV(C,8)
      CALL NFIND (NODE,LN,NCNT,ND1)
      NODE=INCONV(D,8)
      CALL NFIND (NODE,LN,NCNT,ND2)
      IF (ND1.LT.1.OR.ND2.LT.1) GO TO 99
      WRITE (7,61) A1,A2,B,LD(ND1),LD(ND2),E,G,AA,BB
61    FORMAT (2A4,8A1,2I8,3(8A1),24A1)
      NEL=NEL+1
      GO TO 1
C                                  RBE2
63    NODE=INCONV(C,8)
      CALL NFIND (NODE,LN,NCNT,ND1)
      NODE=INCONV(E,8)
      CALL NFIND (NODE,LN,NCNT,ND2)
      NODE=INCONV(G,8)
      CALL NFIND (NODE,LN,NCNT,ND3)
      NODE=INCONV(AA,8)
      CALL NFIND (NODE,LN,NCNT,ND4)
      NODE=INCONV(BB(1),8)
      CALL NFIND (NODE,LN,NCNT,ND5)
      NODE=INCONV(BB(9),8)
      CALL NFIND (NODE,LN,NCNT,ND6)
      IF(ND1.LT.1.OR.ND2.LT.1.OR.ND3.LT.0.OR.ND4.LT.0.OR.ND5.LT.0.OR.ND6
     1.LT.0) GO TO 97
      WRITE (7,64) A1,A2,B,LD(ND1),D,LD(ND2),LD(ND3),LD(ND4),
     1LD(ND5),LD(ND6),(BB(I),I=17,27)
64    FORMAT (2A4,8A1,I8,8A1,5I8,8A1)
      NEL=NEL+1
C     DETERMINE IF CONTINUATION FIELD IS PRESENT
C     IF YES ,TREAT SAME WAY AS IN SOLIDS
      IF (BB(17).EQ.1H ) GO TO 1
      GO TO 52
C        RENUMBER GRIDS
10      NODE=INCONV(B,8)
        CALL NFIND (NODE,LN,NCNT,NEWN)
        WRITE (7,11) A1,A2,LD(NEWN),C,D,E,G,AA,BB
11      FORMAT (2A4,I8,5(8A1),24A1)
        GO TO 1
99    WRITE (IOUT,98) A1,A2,B,C,D,E,G,AA,BB,LD(ND1),LD(ND2),
     1LD(ND3),LD(ND4)
98    FORMAT (1H ,* UNDEFINED GRID(S) REFERENCED BY*,/,
```

```
     12A4,6(8A1),24A1,/,
     11H ,* $ $ $ $ $ $ $ $ $ $ $ *,6I8)
      GO TO 5
97    WRITE (IOUT,98) A1,A2,B,C,D,E,G,AA,BB,LD(ND1),LD(ND2),LD(ND3),
     1LD(ND4),LD(ND5),LD(ND6)
      GO TO 5
96    WRITE(IOUT,95)CHB,B,C,D,E,G,AA,BB,LD(ND8),LD(ND7),
     1LD(ND1),LD(ND2),LD(ND3),LD(ND4),LD(ND5),LD(ND6)
95    FORMAT (1H ,* UNDEFINED GRID(S) REFERENCED BY*,/,
     17(8A1),24A1,/,1H ,* $ $ $ *,8I8)
      GO TO 5
91    FORMAT (2H$$,I8,*  ELEMENTS PROCESSED*,/,
     1*  NORMAL TERMINATION*,//,1H1,19X,*NEW TO OLD GRID LIST*,//)
102   IF (NOMIT.GT.0) WRITE (IOUT,92) NOMIT
92    FORMAT (*AT LEAST*,I8,*   GRIDS MISSING*)
        IF (NOMIT.GT.0) GO TO 100
        WRITE (IOUT,91) NEL
        DO 105 I=1,NCNT,10
                II=I+9
                IF (II.GT.NCNT) II=NCNT
105   WRITE(IOUT,106)((LD(K),LN(K)),K=I,II)
106     FORMAT (20I6)
100     STOP
103   WRITE (IOUT,104) MAXGD,NCNT
104   FORMAT (*$ARRAY SIZE (*,I8,* ) EXCEEDED ,  CURRENTLY NEED =*,I8)
      STOP
   19 FORMAT (2H$$,I6,25H GRIDS WILL BE PROCESSED )
      END
C
C
      SUBROUTINE INPGRD (IU,IAR,MAXIAR,N)
C     READS TAPE IU, PICKS OUT NODES, STORES IN IAR
      DIMENSION IAR(MAXIAR),B(8),BB(64)
      DATA GRD/4HGRID/
2     FORMAT (2(A4),8A1,64A1)
      REWIND IU
      N=0
1     READ (IU,2) A1,A2,B,BB
      IF (EOF(IU)) 10,3
3     IF (A1.NE.GRD) GO TO 1
      N=N+1
      IF (N.GT.MAXIAR) GO TO 1
      IAR(N)=INCONV(B,8)
      GO TO 1
10    RETURN
      END
C
      FUNCTION INCONV(A,LF)
C    FORTRAN FUNCTION TO CONVERT DATA READ AS FREE FORMAT
C     ALPHANUMERIC IN A FIELD OF LENGTH LF TO AN INTEGER
C     A(I),I=1,LF   ARE THE ALPHANUMERICS READ IN A LFA1 FORMAT
C     FUNCTION RETURN IS AN INTEGER VALUE
      DIMENSION A(LF),F(10)
      DATA BLANK,F(1),F(2),F(3),F(4),F(5),F(6),F(7),F(8),F(9),F(10)/
     1       1H , 1H1, 1H2, 1H3, 1H4, 1H5, 1H6, 1H7, 1H8, 1H9, 1H0/
      INT=0
      II=LF
      I=LF
   10 IF(A(I).EQ.BLANK) GO TO 100
      IP=LF-II
```

```
      DO 20 J=1,9
   20 IF(A(I).EQ.F(J)) INT=INT+J*(10**IP)
      II=II-1
      I=I-1
      IF(I.EQ.0) GO TO 90
      GO TO 10
  100 I=I-1
      IF(I.EQ.0) GO TO 90
      GO TO 10
   90 INCONV=INT
   99 RETURN
            END
C
      SUBROUTINE NFIND(ISECTR,IECTR,N,IF)
      DIMENSION IECTR(N)
      IF=0
      IF (ISECTR.EQ.0) RETURN
      IF=-1
      IF (ISECTR.LT.IECTR(1).OR.ISECTR.GT.IECTR(N)) RETURN
      B=0.
      E=N
   10 M=(B+E+1.)/2.
      IF(ISECTR-IECTR(M)) 15,25,20
   15 IF(ISECTR-IECTR(M-1)) 30,35,40
   30 E=M
      GO TO 10
   35 M=M-1
      GO TO 25
   40 RETURN
   20 IF(ISECTR-IECTR(M+1)) 45,50,55
   45 RETURN
   50 M=M+1
      GO TO 25
   55 B=M
      GO TO 10
   25 IF=M
      RETURN
      END
C
      SUBROUTINE BSORT (LX,N)
C     TO SORT AN ARRAY IN INCREASING ORDER OF CONTENTS
C      BY THE BUBBLE SORT METHOD
      DIMENSION LX(N)
      NN=N-1
      DO 10 I=1,NN
          II=I+1
          DO 10 J=II,N
              IF (LX(I).LE.LX(J)) GO TO 10
              M=LX(I)
              LX(I)=LX(J)
              LX(J)=M
10    CONTINUE
      RETURN
      END
C
      SUBROUTINE BBSORT(LX,N)
C     TO SORT A (2,N) MATRIX BY INCREASING ORDER OF
C     CONTENTS OF THE SECOND ROW USING THE BUBBLE SORT METHOD
      DIMENSION LX(2,N)
      NN=N-1
```

```
      DO 10 I=1,NN
      II=I+1
      DO 10 J=II,N
      IF(LX(2,I).LE.LX(2,J)) GO TO 10
      M1=LX(1,I)
      M2=LX(2,I)
      LX(1,I)=LX(1,J)
      LX(2,I)=LX(2,J)
      LX(1,J)=M1
      LX(2,J)=M2
10    CONTINUE
      RETURN
      END
```

```
      PROGRAM DDUPLN (TAPE5,TAPE6,TAPE7)
C**    READS A BULK DATA DECK AND CHECKS FOR ALL DUPLICATE NODES
C         AND DELETES DUPLICATES
C
C     TAPE5  : INPUT DECK
C     TAPE6  : OUTPUT INFORMATION
C     TAPE7  : OUTPUT DECK
C
      DIMENSION NPT(3000)
      DIMENSION B(8),C(8),D(8),E(8),G(8),AA(8),BB(24)
      DATA GD/4HGRID/
      IOUT=6
      IINP=5
C     MAX.NODES
                MAXN=3000
      NG=0
      NDUPL=0
      REWIND IINP
    1 READ (IINP,2) A1,A2,B,C,D,E,G,AA,BB
      IF (EOF(IINP)) 101,3
    2 FORMAT (2(A4),6(8A1),24A1)
    3 IF (A1.NE.GD) GO TO 1
      NG=NG+1
      IF (NG.GT.MAXN) GO TO 103
      NPT(NG)=INCONV(B,8)
      GO TO 1
  101 REWIND IINP
      MAXN=NG
      WRITE (IOUT,10) MAXN
   10 FORMAT (1H1,I6,25H GRIDS WILL BE PROCESSED )
      MAXNN=MAXN-1
      DO 20 I=1,MAXNN
       IF (NPT(I).LE.0) NDUPL=NDUPL+1
       IF (NPT(I).LE.0) GO TO 20
      II=I+1
      DO 19 J=II,MAXN
19        IF (NPT(I).EQ.NPT(J)) NPT(J)=-NPT(J)
   20 CONTINUE
      IF (NDUPL.LE.0) GO TO 102
C
C     REMOVE DUPLICATE NODES
C
      WRITE (IOUT,32) NDUPL
32    FORMAT(1H1,I10,*  DUPLICATE NODES HAVE BEEN DELETED*,//)
      REWIND IINP
      NG=0
      REWIND 7
31    READ (IINP,2) A1,A2,B,C,D,E,G,AA,BB
      IF (EOF(IINP)) 105,33
33    IF (A1.EQ.GD) GO TO 34
      WRITE (7,2) A1,A2,B,C,D,E,G,AA,BB
      GO TO 31
34    NDUP=INCONV(B,8)
      NG=NG+1
      IF (NPT(NG).NE.NDUP) WRITE (IOUT,35) NG,NDUP
      IF (NPT(NG).NE.NDUP) GO TO 31
      WRITE (7,2) A1,A2,B,C,D,E,G,AA,BB
35    FORMAT (9H PASS NO.,I5,14H  DELETED NODE,I10)
      GO TO 31
```

```
105    CONTINUE
102    STOP
103    WRITE (IOUT,104) MAXN
104    FORMAT (*  ARRAY SIZE EXCEEDED  =*,I8)
       END
       FUNCTION INCONV(A,LF)
C     FORTRAN FUNCTION TO CONVERT DATA READ AS FREE FORMAT
C      ALPHANUMERIC IN A FIELD OF LENGTH LF TO AN INTEGER
C      A(I),I=1,LF   ARE THE ALPHANUMERICS READ IN A LFA1 FORMAT
C      FUNCTION RETURN IS AN INTEGER VALUE
       DIMENSION A(LF),F(10)
       DATA BLANK,F(1),F(2),F(3),F(4),F(5),F(6),F(7),F(8),F(9),F(10)/
      1       1H , 1H1, 1H2, 1H3, 1H4, 1H5, 1H6, 1H7, 1H8, 1H9, 1H0/
       INT=0
       II=LF
       I=LF
   10  IF(A(I).EQ.BLANK) GO TO 100
       IP=LF-II
       DO 20 J=1,9
   20  IF(A(I).EQ.F(J)) INT=INT+J*(10**IP)
       II=II-1
       I=I-1
       IF(I.EQ.0) GO TO 90
       GO TO 10
  100  I=I-1
       IF(I.EQ.0) GO TO 90
       GO TO 10
   90  INCONV=INT
   99  RETURN
               END
```

```
      PROGRAM ISOPID (TAPE4,TAPE5,TAPE6,TAPE7)
C             TO ISOLATE ELEMENTS AND GRIDS BY A PID LIST
C             IN NASTRAN BULK DATA FORM
C
      DIMENSION LN(5000),BPID(80),LPID(10),ELEM(8)
      DIMENSION B(8),C(8),D(8),E(8),G(8),AA(8),BB(24)
      DATA GRID/4HGRID/
      DATA ELEM/4HCQUA,4HCTRI,4HCROD,4HCELA,4HCBEA,4HCBAR,4HCHEX,4HCPEN
     /
C
C     TAPE4  : INPUT PIDS
C     TAPE5  : INPUT BULK DATA DECK
C     TAPE6  : OUTPUT INFORMATIVE STATISTICS
C     TAPE7  : OUTPUT ISOLATED BULK DATA DECK
C
C             PROGRAM LIMITATIONS
C             MAXN   : MAX NODE CONNECTION LIST PER PID
      MAXN=5000
C             MAXPID: MAX NO. OF PIDS PER SORT PASS
      MAXPID=10
C             NO.OF ELEMENTS IN ELEM LIBRARY
      NELLIB=8
C              MAX NODES PER ELEMENT
      MNPEL=20
      REWIND 6
      REWIND 7
C       INITIALIZE COUNTERS
C             READ NEW PID SET
6     READ (4,101) BPID
101   FORMAT (10(8A1))
      IF (EOF(4)) 999,1
1     NSORT=0
      NCNT=0
      NELP=0
C     ESTABLISH LENGTH AND LIST OF PID SET
      DO 11 I1=1,MAXPID
             I1J=(I1-1)*8
             DO 12 J1=1,8
12                  B(J1)=BPID(I1J+J1)
             LPID(I1)=INCONV(B,8)
11           IF (LPID(I1).GT.0) NSORT=I1
      IF (NSORT.LE.0) GO TO 897
      WRITE (6,990) (LPID(I),I=1,NSORT)
C     READ AND ISOLATE ELTS OF CHOSEN PIDS
      REWIND 5
2     READ (5,102) A1,A2,B,C,D,E,G,AA,BB
102   FORMAT (2A4,6(8A1),24A1)
      IF (EOF(5)) 199,3
3     IEL=0
      DO 4 I=1,NELLIB
4            IF (ELEM(I).EQ.A1) IEL=I
      IF (IEL.EQ.0) GO TO 2
      NPID=INCONV(C,8)
C            CHECK IF DESIRED PID
      DO 13 I=1,NSORT
13    IF (NPID.EQ.LPID(I)) GO TO 14
      GO TO 2
C            KEEP THIS ELEMENT
14    CONTINUE
```

```
      GO TO (10,20,30,30,30,30,40,40),IEL
C             QUADS
10    LN(NCNT+1)=INCONV(AA,8)
      NCNT=NCNT+1
C             TRIANGLES
20    LN(NCNT+1)=INCONV(G,8)
      NCNT=NCNT+1
C             TWO NODED ELEMENTS ONLY
30    LN(NCNT+1)=INCONV(E,8)
      LN(NCNT+2)=INCONV(D,8)
      NCNT=NCNT+2
      WRITE (7,102) A1,A2,B,C,D,E,G,AA,BB
      GO TO 99
C           SOLIDS
C     COL 73 MUST BE NON-BLANK
C     SUBSEQUENT RECORD MUST START WITH A +
40    LN(NCNT+1)=INCONV(D,8)
      LN(NCNT+2)=INCONV(E,8)
      LN(NCNT+3)=INCONV(G,8)
      LN(NCNT+4)=INCONV(AA,8)
      LN(NCNT+5)=INCONV(BB(1),8)
      LN(NCNT+6)=INCONV(BB(9),8)
      NCNT=NCNT+6
      WRITE (7,102) A1,A2,B,C,D,E,G,AA,BB
      IF (BB(17).EQ.1H ) GO TO 99
      READ (5,103) A1,A2,B,C,D,E,G,AA,BB
      IF (A1.NE.1H+) GO TO 49
      LN(NCNT+1)=INCONV(B,8)
      LN(NCNT+2)=INCONV(C,8)
      LN(NCNT+3)=INCONV(D,8)
      LN(NCNT+4)=INCONV(E,8)
      LN(NCNT+5)=INCONV(G,8)
      LN(NCNT+6)=INCONV(AA,8)
      LN(NCNT+7)=INCONV(BB(1),8)
      LN(NCNT+8)=INCONV(BB(9),8)
      NCNT=NCNT+8
      WRITE (7,103) A1,A2,B,C,D,E,G,AA,BB
      IF (BB(17).EQ.1H ) GO TO 99
      READ (5,103) A1,A2,B,C,D,E,G,AA,BB
      IF (A1.NE.1H+) GO TO 49
      LN(NCNT+1)=INCONV(B,8)
      LN(NCNT+2)=INCONV(C,8)
      LN(NCNT+3)=INCONV(D,8)
      LN(NCNT+4)=INCONV(E,8)
      LN(NCNT+5)=INCONV(G,8)
      LN(NCNT+6)=INCONV(AA,8)
      NCNT=NCNT+6
      WRITE (7,103) A1,A2,B,C,D,E,G,AA,BB
103   FORMAT (A1,A7,6(8A1),24A1)
      GO TO 99
49    BACKSPACE 5
99    NELP=NELP+1
      IF ((NCNT+MNPEL).GT.MAXN)  GO TO 899
      GO TO 2
199   CONTINUE
      IF (NELP.LE.0) GO TO 895
      IF (NCNT.LT.2) GO TO 896
C             ZEROIZE DUPLICATE NODE NUMBERS
      NCNTM1=NCNT-1
      DO 201 I=1,NCNTM1
```

```
              II=I+1
              DO 201 J=II,NCNT
201           IF (LN(I).EQ.LN(J)) LN(J)=0
C      PACK NODE ARRAY
       J=0
       IZERO=0
       DO 202 I=1,NCNT
              LN(I-IZERO)=LN(I)
202           IF (LN(I).LE.0) IZERO=IZERO+1
       NCNT=NCNT-IZERO
       WRITE (6,991) NELP
       REWIND 5
C      BUBBLE SORT LN ARRAY FOR EXPEDIENCE OF SEARCH
       CALL BSORT (LN,NCNT)
C      IDENTIFY NEEDED GRIDS AND SAVE ON 7
       IGRID=0
301    READ (5,102) A1,A2,B,C,D,E,G,AA,BB
       IF (EOF(5)) 351,302
302    IF (A1.NE.GRID) GO TO 301
       NODE=INCONV(B,8)
       CALL FIND (NODE,LN,NCNT,IFOUND)
       IF (IFOUND.GT.0) WRITE (7,102)A1,A2,B,C,D,E,G,AA,BB
       IF (IFOUND.GT.0) IGRID=IGRID+1
       IF (IGRID.LT.NCNT) GO TO 301
351    WRITE (6,992) IGRID
       IF (IGRID.LT.NCNT) WRITE (6,994) IGRID,NCNT
       ENDFILE 7
       GO TO 6
895    WRITE (INFO,991) NELP
       GO TO 999
896    WRITE (INFO,992) NCNT
       GO TO 999
897    WRITE (6,997)
       GO TO 999
899    WRITE (6,993) MAXN
       GO TO 999
990    FORMAT (1H1,* LIST OF PID'S  TO SORT BY *,/,10I8,//)
991    FORMAT (1H ,*NO. OF ELEMENTS PROCESSED =*,I8)
992    FORMAT (1H ,*NO. OF GRIDS IN CHOSEN SET =*,I8)
993    FORMAT (1H ,*  ARRAY SIZE EXCEEDED  *,I8)
994    FORMAT (1H ,I8,*  GRIDS FOUND OUT*,I8,* NEEDED IN SET*)
997    FORMAT (1H ,*  IMPROPER PID INPUT  *)
999    STOP
       END
C
       FUNCTION INCONV(A,LF)
C     FORTRAN FUNCTION TO CONVERT DATA READ AS FREE FORMAT
C      ALPHANUMERIC IN A FIELD OF LENGTH LF TO AN INTEGER
C      A(I),I=1,LF   ARE THE ALPHANUMERICS READ IN A LFA1 FORMAT
C      FUNCTION RETURN IS AN INTEGER VALUE
       DIMENSION A(LF),F(10)
       DATA BLANK,F(1),F(2),F(3),F(4),F(5),F(6),F(7),F(8),F(9),F(10)/
      1        1H , 1H1, 1H2, 1H3, 1H4, 1H5, 1H6, 1H7, 1H8, 1H9, 1H0/
       INT=0
       II=LF
       I=LF
   10  IF(A(I).EQ.BLANK) GO TO 100
       IP=LF-II
       DO 20 J=1,9
   20  IF(A(I).EQ.F(J)) INT=INT+J*(10**IP)
```

```
      II=II-1
      I=I-1
      IF(I.EQ.0) GO TO 90
      GO TO 10
  100 I=I-1
      IF(I.EQ.0) GO TO 90
      GO TO 10
   90 INCONV=INT
   99 RETURN
                END
C
      SUBROUTINE FIND(ISECTR,IECTR,N,IF)
      DIMENSION IECTR(N)
      IF=-1
      IF (ISECTR.LT.IECTR(1).OR.ISECTR.GT.IECTR(N)) RETURN
      B=0.
      E=N
   10 M=(B+E+1.)/2.
      IF(ISECTR-IECTR(M)) 15,25,20
   15 IF(ISECTR-IECTR(M-1)) 30,35,40
   30 E=M
      GO TO 10
   35 M=M-1
      GO TO 25
   40 RETURN
   20 IF(ISECTR-IECTR(M+1)) 45,50,55
   45 RETURN
   50 M=M+1
      GO TO 25
   55 B=M
      GO TO 10
   25 IF=M
      RETURN
      END
C
      SUBROUTINE BSORT (LX,N)
C     TO SORT AN ARRAY IN INCREASING ORDER OF CONTENTS
C      BY THE BUBBLE SORT METHOD
      DIMENSION LX(N)
      NN=N-1
      DO 10 I=1,NN
          II=I+1
          DO 10 J=II,N
              IF (LX(I).LE.LX(J)) GO TO 10
              M=LX(I)
              LX(I)=LX(J)
              LX(J)=M
10    CONTINUE
      RETURN
      END
```

```
      PROGRAM WELDPTS (TAPE3,TAPE2,OUTPUT,TAPE6=OUTPUT)
      DIMENSION ISAVE(2000)
C
C
C       TAPE 3 = INPUT FILE
C
C       TAPE 2 = OUTPUT FILE
C
C
C     START READING
C
C       THIS IS FORMATED TO TAKE THE INFORMATION FILE OUTPUT
C            FROM DDUPLN PROGRAM
C
      REWIND 3
      READ(3,19)II
   19 FORMAT(1X,I6)
      IF(EOF(3).NE.0) GO TO 140
      READ(3,20)NSG
   20 FORMAT(1X,I10)
      IF(EOF(3).NE.0) GO TO 140
      READ(3,21)
   21 FORMAT(/)
      II = 0
  130 II = II + 1
      READ(3,122)ISAVE(II)
      IF(EOF(3).EQ.0) GO TO 130
      NSG = II - 1
  122 FORMAT(29X,I9)
      GO TO 150
  140 NSG = 0
  150 CONTINUE
C
C     START WRITING
C
   30 J = NSG
      LCOL=11
      IF (J.GT.LCOL) GO TO 60
      WRITE (2,61)
   61 FORMAT(* INCLUDE GRID POINTS,*)
      LBEG=1
      GO TO 70
   60 CONTINUE
      ROW=J/11.
      LROW=INT(ROW)
      WRITE (2,61)
      LEND=0
      DO 40 K=1,LROW
      LBEG = ((K-1)*LCOL)+1
      LEND=LEND+LCOL
      IF (LEND.NE.J) GO TO 90
C     WRITE (2,23) (ISAVE(L),L=LBEG,LEND)
      WRITE(2,22) (ISAVE(L),L=LBEG,LEND)
      GO TO 50
   90 CONTINUE
   40 WRITE (2,22) (ISAVE(L),L=LBEG,LEND)
      LBEG=LBEG+11
```

```
C   70 WRITE (2,23) (ISAVE (LL),LL=LBEG,J)
    70 WRITE(2,22) (ISAVE(LL),LL=LBEG,J)
    22 FORMAT(1H ,11(I5,*,*))
    23 FORMAT (1H ,11(I6))
    50 CONTINUE
   999 WRITE(6,5)
     5 FORMAT(/,* OUTPUT LIST IS ON TAPE2*)
       WRITE(2,67)
    67 FORMAT(* EXCLUDE GRID POINTS 1 THRU 90000*)
       REWIND 2
       STOP
   897 WRITE(6,62)
    62 FORMAT(* PROBLEMS*)
       STOP
       END
```

```
      PROGRAM SETSPID (TAPE4,TAPE5,TAPE2,TAPE7,OUTPUT,TAPE6=OUTPUT)
C             TO FORM SET CARDS OF ELEMENTS BY PIDS
C
      DIMENSION ISAVE(2000),BPID(80),LPID(10),ELEM(8)
      DIMENSION B(8),C(8),D(8),E(8),G(8),AA(8),BB(24)
      DATA GRID/4HGRID/
      DATA ELEM/4HCQUA,4HCTRI,4HCROD,4HCELA,4HCBEA,4HCBAR,4HCHEX,4HCPEN/
C
C     TAPE4  : INPUT PIDS (10A8 FORMAT) MULTIPLE CARDS YIELD
C              MULTIPLE SETS
C     TAPE5  : INPUT BULK DATA DECK
C     TAPE2  : OUTPUT SETS
C     TAPE7  : INPUT CARD IMAGES TO BE COPIED AFTER EACH
C              SET OF ELEMENTS (OPTIONAL - WRITTEN TO ALLOW INCLUSION
C              OF "WELD POINTS" SUCH AS 'INCLUED GIRD POINTS .... EXCLUD
E
C              GRID POINTS ...')
C
C            PROGRAM LIMITATIONS
C            MAXN   : MAX NODE CONNECTION LIST PER PID
      MAXN=5000
C            MAXPID: MAX NO. OF PIDS PER SORT PASS
      MAXPID=10
C            NO.OF ELEMENTS IN ELEM LIBRARY
      NELLIB=8
C             MAX NODES PER ELEMENT
      MNPEL=20
C       INITIALIZE COUNTERS
C            READ NEW PID SET
      REWIND 2
      REWIND 4
C
C      ESTABLISH IF THERE IS DATA FROM TAPE7 TO BE COPIED
C      TO THE END OF EACH SET
C
      REWIND 7
      READ(7,701)(BPID(K),K=1,20)
      IF(EOF(7)) 600,601
  600 IT7 = 1
      INCMNT = 0
      GO TO 602
  601 IT7 = 2
      INCMNT = 4
  602 CONTINUE
C
6     READ (4,101) BPID
      J = 1
101   FORMAT (10(8A1))
      IF (EOF(4)) 999,1
1     NSORT=0
      NCNT=0
      NELP=0
C     ESTABLISH LENGTH AND LIST OF PID SET
      DO 11 I1=1,MAXPID
            I1J=(I1-1)*8
            DO 12 J1=1,8
12                B(J1)=BPID(I1J+J1)
```

```
            LPID(I1)=INCONV(B,8)
11          IF (LPID(I1).GT.0) NSORT=I1
      IF (NSORT.LE.0) GO TO 897
      WRITE(2,63)(LPID(I),I=1,NSORT)
   63 FORMAT(*$  SET CARD(S) FOR PID(S) *,10I5)
C     READ AND ISOLATE ELTS OF CHOSEN PIDS
      REWIND 5
2     READ (5,102) A1,A2,B,C,D,E,G,AA,BB
102   FORMAT (2A4,6(8A1),24A1)
      IF (EOF(5)) 30,3
3     IEL=0
      DO 4 I=1,NELLIB
4           IF (ELEM(I).EQ.A1) IEL=I
      IF (IEL.EQ.0) GO TO 2
      NPID=INCONV(C,8)
C           CHECK IF DESIRED PID
      DO 13 I=1,NSORT
13    IF (NPID.EQ.LPID(I)) GO TO 14
      GO TO 2
C           KEEP THIS ELEMENT
14    CONTINUE
      ID=INCONV(B,8)
      ISAVE(J)=ID
      J=J+1
      GO TO 2
   30 J=J-1
      LCOL=11
      IF (J.GT.LCOL) GO TO 60
      ISET5 = LPID(1) + INCMNT
      WRITE (2,61) ISET5
      WRITE(6,65)ISET5,J
      LBEG=1
      GO TO 70
   60 CONTINUE
      ROW=J/11.
      LROW=INT(ROW)
      N = INT(((ROW - FLOAT(LROW))*11.)+.5)
      ISET5 = LPID(1) + INCMNT
      WRITE (2,61) ISET5
      WRITE(6,65)ISET5,J
   61 FORMAT(1H ,*SET *,I5,* = ,*)
      LEND=0
   65 FORMAT(/,*  SET *,I5,*   CONTAINS *,I5,* ELEMENTS*)
      DO 40 K=1,LROW
      LBEG=((K-1)*LCOL)+1
      LEND=LEND+LCOL
      IF (LEND.NE.J) GO TO 90
      GO TO (610,611) IT7
  610 WRITE (2,23) (ISAVE(L),L=LBEG,LEND)
      GO TO 50
  611 WRITE (2,22) (ISAVE(L),L=LBEG,LEND)
      GO TO 50
   90 CONTINUE
   40 WRITE (2,22) (ISAVE(L),L=LBEG,LEND)
      IF (N.EQ.0) GO TO 50
      LBEG=LBEG+11
   70 CONTINUE
      GO TO (620,621) IT7
  620 WRITE (2,23) (ISAVE (LL),LL=LBEG,J)
      GO TO 50
```

```
  621 WRITE (2,22) (ISAVE (LL),LL=LBEG,J)
   22 FORMAT(1H ,11(I5,*,*))
   23 FORMAT (1H ,11(I6))
   50 CONTINUE
      REWIND 7
  700 READ(7,701)(BPID(K),K=1,20)
      IF (EOF(7)) 710,711
  711 WRITE(2,701)(BPID(K),K=1,20)
      GO TO 700
  710 CONTINUE
  701 FORMAT(20A4)
      GO TO 6
  999 WRITE(6,5)
    5 FORMAT(/,* OUTPUT LIST IS ON TAPE2*)
      REWIND 2
      STOP
  897 WRITE(6,62)
   62 FORMAT(*  ERROR ON PID*)
      STOP
      END
C
      FUNCTION INCONV(A,LF)
C    FORTRAN FUNCTION TO CONVERT DATA READ AS FREE FORMAT
C     ALPHANUMERIC IN A FIELD OF LENGTH LF TO AN INTEGER
C     A(I),I=1,LF   ARE THE ALPHANUMERICS READ IN A LFA1 FORMAT
C     FUNCTION RETURN IS AN INTEGER VALUE
      DIMENSION A(LF),F(10)
      DATA BLANK,F(1),F(2),F(3),F(4),F(5),F(6),F(7),F(8),F(9),F(10)/
     1       1H , 1H1, 1H2, 1H3, 1H4, 1H5, 1H6, 1H7, 1H8, 1H9, 1H0/
      INT=0
      II=LF
      I=LF
   10 IF(A(I).EQ.BLANK) GO TO 100
      IP=LF-II
      DO 20 J=1,9
   20 IF(A(I).EQ.F(J)) INT=INT+J*(10**IP)
      II=II-1
      I=I-1
      IF(I.EQ.0) GO TO 90
      GO TO 10
  100 I=I-1
      IF(I.EQ.0) GO TO 90
      GO TO 10
   90 INCONV=INT
   99 RETURN
              END
```