

ANSYS Duplicate Finite-Element Checker Routine

R. Ortega



ANSYS Duplicate Finite-Element Checker Routine

R. Ortega
Marshall Space Flight Center • MSFC, Alabama

TABLE OF CONTENTS

	Page
I. INTRODUCTION	1
II. THEORY	1
III. ROUTINE DESCRIPTION	2
IV. ATD HPOTP RESULTS	3
V. CONCLUSIONS.....	3
REFERENCES.....	9
APPENDIX – Duplicate Elements Checker ANSYS50a Routine.....	11

LIST OF ILLUSTRATIONS

Figure	Title	Page
1.	Common face of adjoining elements	5
2.	Planar view of points A, B, C, and D	5
3.	Examples of floating elements within a mesh	6
4.	Examples of fully duplicated elements.....	7
5.	Examples of intersecting elements with common faces	8

TECHNICAL MEMORANDUM

ANSYS DUPLICATE FINITE-ELEMENT CHECKER ROUTINE

I. INTRODUCTION

This report documents an ANSYS routine to check for duplicated elements within the volume of a three-dimensional (3-D) finite-element mesh.¹ Recent finite-element submodeling efforts of the space shuttle main engine alternate turbopump development (ATD) high pressure oxidizer turbopump (HPOTP) uncovered duplicate elements in a certain location of the global finite-element model. This initiated an effort to check the entire model of the ATD HPOTP which consists of a total of 54,296 elements distributed over 14 superelements. An automated method was needed to check the large number of elements involved.² A routine was then developed to check 3-D finite-element meshes for the existence of unconnected elements, elements within elements, and elements across other elements.

II. THEORY

Figure 1 shows the common face of adjoining elements (face BEGF) and the tetrahedral volumes formed by joining the corners of this common face to the respective element centroids (points A and D). Point C is formed by the intersection of the line passing through A and D and face BEGF. The intersection of the three planes ABD, AED, and BEF provide the coordinates of point C (XC, YC, ZC). This is done mathematically by defining the following cross products:

$$\overline{RA} = \overline{AB} \times \overline{AD} ,$$

$$\overline{RB} = \overline{AE} \times \overline{AD} ,$$

$$\overline{RC} = \overline{BE} \times \overline{BF} .$$

Vectors RA, RB, and RC represent the normal vectors of the three planes defined above. Using these normal vectors, the equations of the three planes can be written as a system of three linear equations:

$$aaXC + abYC + acZC = PA ,$$

$$baXC + bbYC + bcZC = PB ,$$

$$caXC + cbYC + ccZC = PC .$$

Cramer's rule, found in any algebra book, can be used to solve the above system providing coordinates XC, YC, and ZC of point C.³

Figure 2 shows points A, B, C, and D. The line joining points B and C falls within face BEGF (shown in fig. 1). For adjoining elements not to have any part of their volume crossing each other, it must be demonstrated that points A and D (element centroids) fall on opposite sides of the line joining B and C.

This can be done by comparing the normal vectors from the planes obtained from areas ABC and DBC. The dot product of two unit vectors provides the cosine of the angle between the vectors. If the value is negative, then the angle must be greater than 90° and less than 180°. Therefore, the dot product of the normal vectors from the planes obtained from areas ABC and DBC must be less than zero for the centroids A and D to fall on opposite sides of the line joining B and C. Thus, the following inequality must be satisfied:

$$(\overline{AB} \times \overline{AC}) \cdot (\overline{DB} \times \overline{DC}) < 0 .$$

Otherwise, the elements are deemed to have a volume intersection with each other. There are two key assumptions that must be noted. First, for elements to be adjoining each other, they must have a face in common, which is accomplished by sharing at least three of the four corner nodes. Second, the face of adjoining elements is considered to be flat or with a very minor degree of warping.

III. ROUTINE DESCRIPTION

To implement the check of the above developed inequality, an ANSYS50a routine was written and is enclosed in the appendix. In order to run this routine, a data base file (.db file) of the model in ANSYS50a format must be available. The routine is called checker3d3.dat and runs in batch mode. The data base file of the job that is to be checked should be copied into a file named duplicate.db and placed in a folder together with checker3d3.dat.

The ANSYS50a routine first resumes the model from the data base file and counts the number of elements (numb). An array is then set with a size of numb by six. The number six represents the maximum number of sides that a brick element is allowed to have. A do loop is started to check every element on every side. Because of this, the routine takes a very large amount of central processing unit (CPU) time to complete its task. The routine was originally developed as an ANSYS50a macro with the intent of using ANSYS select logic. The assumption was that using an already developed select logic would reduce the time necessary for developing a routine to conduct the element checks. The amount of CPU runtime savings to be obtained from developing a checker FORTRAN routine from scratch (including select logic) is not known, however, it is suspected that the savings could be quite large.

Three element irregularities are checked by the ANSYS50a routine. The first is when there is an element without any other elements attached to it at a common face. This means that the element could be floating within the mesh attached to other element nodes. Figure 3 shows examples of this condition. Duplicate elements are shown in gray in the figure. Unattached floating elements are reported in a file called single.elem. The second element irregularity is when two elements are duplicated, having exactly the same nodes. Fully duplicated elements' numbers, shown in figure 4, are given as pairs in nontriv.elem. Fully duplicated elements are those that contain exactly the same nodes. Finally, the third condition checked is when partial or full volumes of element fall within or cross other elements and have a common face. The third condition is shown in figure 5. The elements

numbers are reported in double.elem in pairs. Elements shown in double.elem should be visually checked by pairs, because element warping could result in the reporting of legitimate elements as intersecting each other. This is due to the fact that a warped face could wrap around a centroid location, resulting in a positive value for the above inequality.

IV. ATD HPOTP RESULTS

The results of the check for duplicate elements on the ATD HPOTP global model are shown in table 1. The table shows the superelement id, number of elements, the number of duplicate elements reported by routine checker3d3.dat, and the final number of duplicate elements after the final visual check of the reported elements. The difference between the reported and final number of duplicate elements represents those elements whose degree of warping was such that the check inequality was not satisfied, but are acceptable to the ANSYS code. The last column of table 1 reports the CPU time required to run the check on a Silicon Graphics ONYX workstation with two 100-mHz processors.

V. CONCLUSIONS

An ANSYS5.0 routine to check for duplicated elements within the volume of a 3-D finite-element mesh was developed. This routine was used to identify ATD HPOTP finite-element model duplicate elements. It is of importance to note that the number of duplicate elements is small compared to the overall model and ended up having little impact on the final results of the ATD HPOTP model. However, these duplicate elements could have occurred in a stress concentration region providing erroneous stress results. Also, the mere existence of such elements can complicate submodeling efforts in regions where they are contained, as it did occur on a submodel effort for the ATD HPOTP weld No. 2.

The ANSYS finite-element code assumes that duplicate elements are modeled on purpose. There are instances where duplicate elements are desirable in a mesh. However, unwanted duplicate elements might appear when a complicated 3-D geometry mesh is modified by hand. Therefore, it is recommended that models be checked when there is knowledge or suspicion of extensive use of manual node and element definitions within a model. Finally, considering the large amount of CPU time needed to run this check, it is suggested that the duplicate element check run be done on a copy of a data base file on a separate folder in the background while work on the model continues as if no duplicate elements exist.

Table 1. Duplicate elements check of ATD HPOTP.

Superelement ID Number	Number of Elements	Number of Duplicate Elements Reported	Final Number of Duplicate Elements	CPU Time (h)
SE 01	3,456	0	0	31.05
SE 02	7,132	32	28	79.29
SE 03	4,007	5	0	44.26
SE 04	3,168	0	0	30.86
SE 05	3,484	0	0	33.16
SE 06	2,924	0	0	19.47
SE 07	3,254	0	0	20.28
SE 08	7,227	24	18	75.39
SE 09	5,972	0	0	54.08
SE 10	2,267	2	0	24.21
SE 11	4,420	17	15	26.27
SE 12	5,414	0	0	84.04
SE 13	687	0	0	7.57
SE 14	884	0	0	3.25

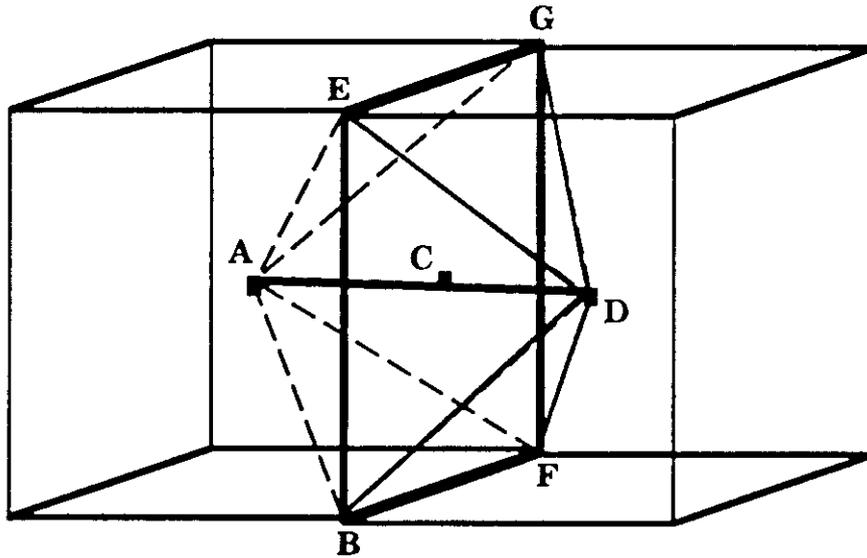


Figure 1. Common face of adjoining elements.

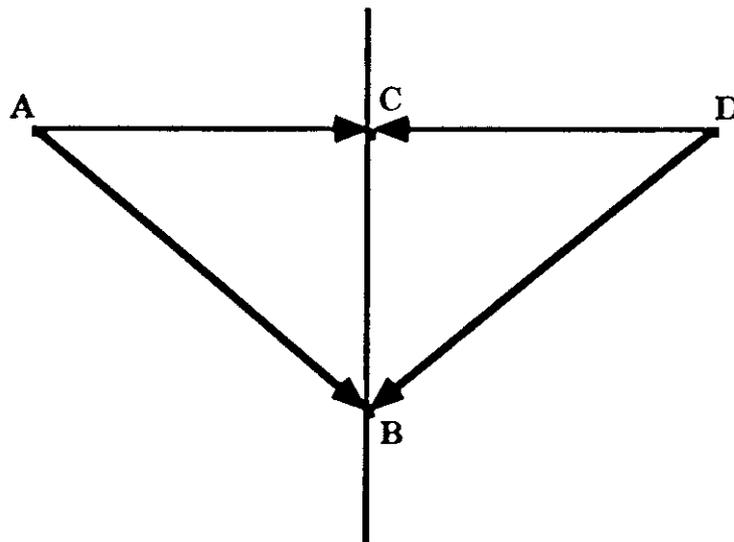


Figure 2. Planar view of points A, B, C, and D.

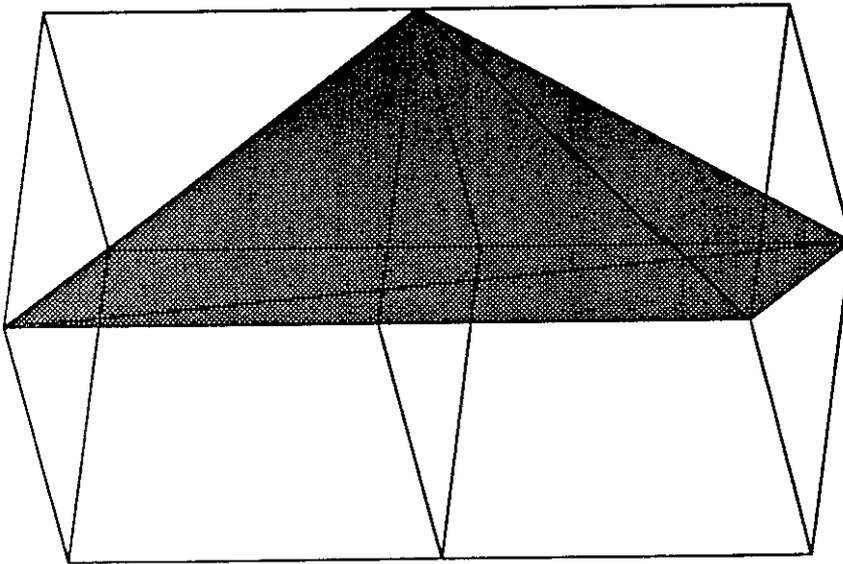
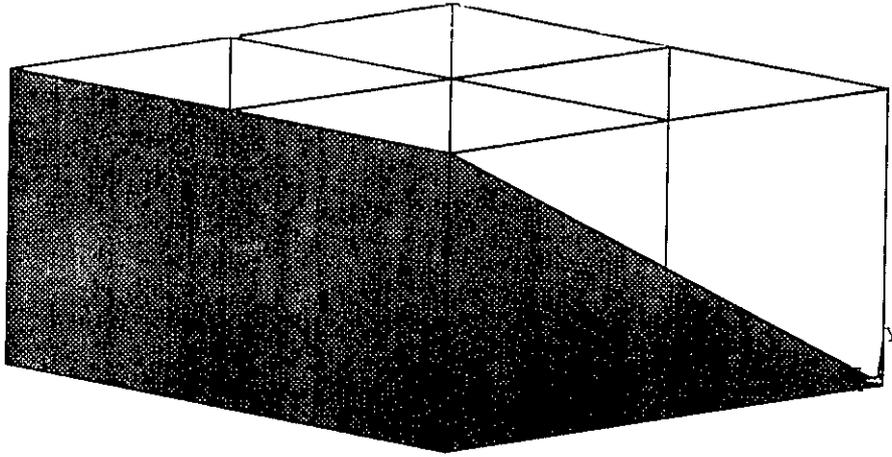


Figure 3. Examples of floating elements within a mesh.

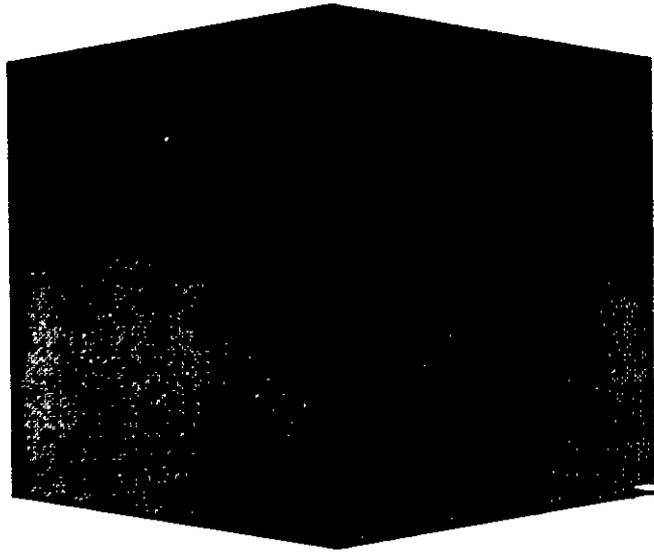


Figure 4. Examples of fully duplicated elements
(Two elements with identical nodes).

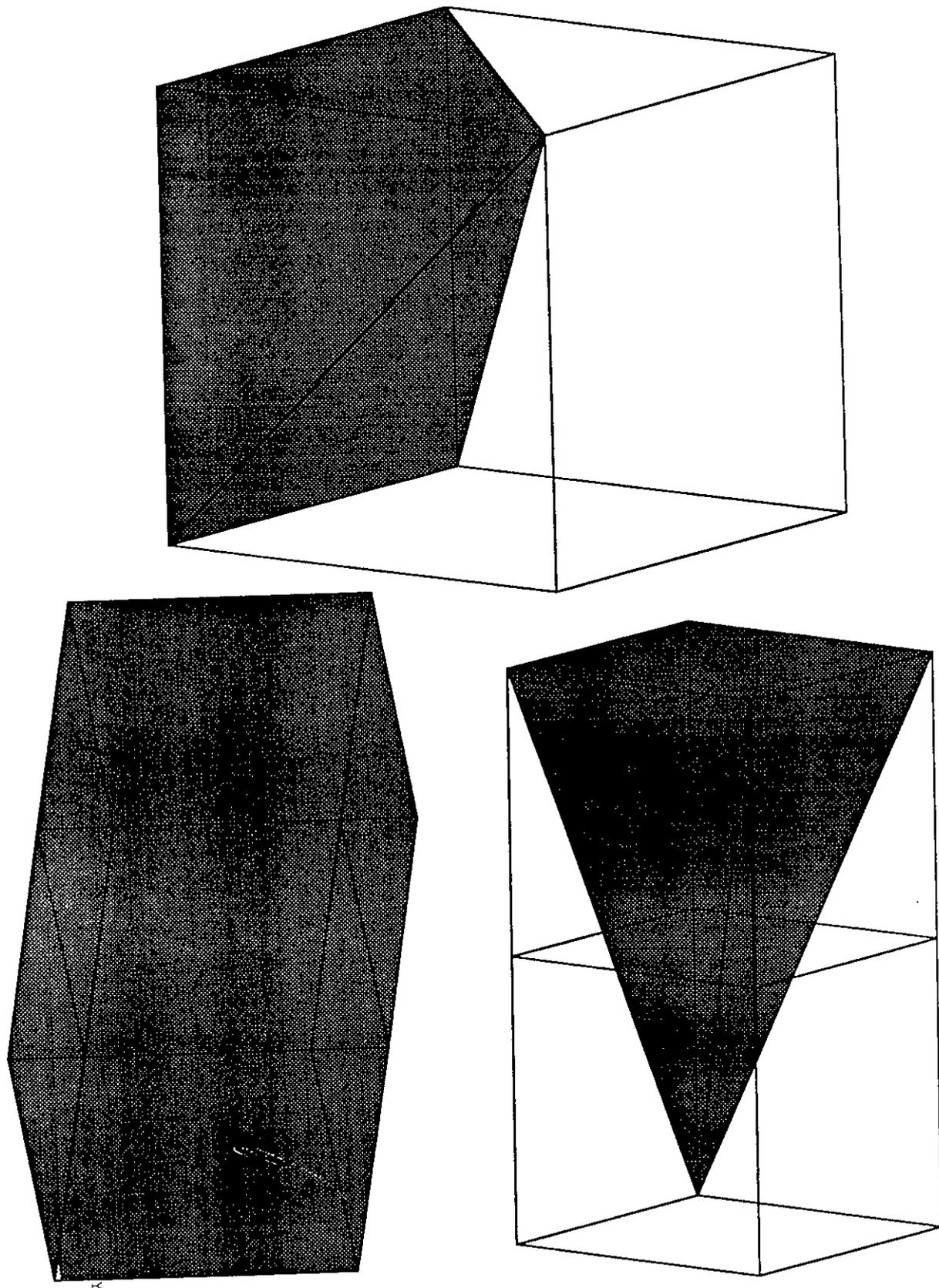


Figure 5. Examples of intersecting elements with common faces.

REFERENCES

1. "ANSYS 5.0 User's Manual," vol. I-IV, Revision 5.0a, Swanson Analysis Systems, Inc., Houston, Pennsylvania, December 23, 1992.
2. "Finite Element Analysis of the Space Shuttle Main Engine (SSME) Alternate Turbopump Development (ATD) High Pressure Oxidizer Turbopump (HPOTP)," vol. I, Part 1, report No. 611-008-93-009, Sverdrup Technology, October 1993.
3. Kreyszig, E.: "Advanced Engineering Mathematics." John Wiley and Sons, Inc., New York, NY, 1972.

APPENDIX

Duplicate Elements Checker ANSYS50a Routine

```

/filename,duplicate
resume
/prep7
arr(1,1)=
yes
esel,none
cm,crset,elem
esel,all
nset,all
cm,dummy,elem
*get,numb,elem,0,count
*get,numbb,elem,0,num,max
*dim,arr,,numbb,6
*do,in,1,numb,1
*get,en,elem,0,num,min
esel,s,elem,,en
nsle
*get,nn,node,0,count
*if,nn,le,4,then
nef=nelem(en,5)
*if,nef,le,0,then
cmsel,s,dummy
esel,u,elem,,en
cm,dummy,elem
*cycle
*endif
*endif
*get,xa,elem,en,cent,x
*get,ya,elem,en,cent,y
*get,za,elem,en,cent,z
imark=0
*do,jn,1,6,1
*if,jn,eq,1,then
b=nelem(en,1)
e=nelem(en,4)
f=nelem(en,2)
*endif
*if,jn,eq,2,then
b=nelem(en,2)
e=nelem(en,6)
f=nelem(en,1)
*endif
*if,jn,eq,3,then
b=nelem(en,3)
e=nelem(en,7)
f=nelem(en,2)
*endif
*if,jn,eq,4,then
b=nelem(en,4)
e=nelem(en,8)

```

```

f=nelem(en,3)
*endif
*if,jn,eq,5,then
b=nelem(en,1)
e=nelem(en,5)
f=nelem(en,4)
*endif
*if,jn,eq,6,then
b=nelem(en,6)
e=nelem(en,7)
f=nelem(en,5)
*endif
nset,s,node,,b
nset,a,node,,e
nset,a,node,,f
cm,stud,node
*get,nnb,node,0,count
*if,nnb,lt,3,then
imark=imark+1
*if,imark,eq,6,then
esel,s,elem,,en
ewrite,single,elem,,1
cmsel,a,crset
cm,crset,elem
*endif
*cycle
*endif
esel,none
cm,elgr,elem
nset,s,node,,b
esln
*get,elb,elem,0,count
cm,elst,elem
*do,jk,1,elb,1
*get,est,elem,0,num,min
esel,s,elem,,est
nsle
cmsel,r,stud
*get,chk,node,0,count
*if,chk,eq,nnb,then
cmsel,a,elgr
esel,u,elem,,en
cm,elgr,elem
*endif
cmsel,s,elst
esel,u,elem,,est
cm,elst,elem
*enddo
cmsel,s,elgr
*get,elc,elem,0,count

```

```

*if,elc,le,0,then
imark=imark+1
*if,imark,eq,6,then
esel,s,elem,,en
ewrite,single,elem,,1
cmsel,a,crset
cm,crset,elem
*endif
*cycle
*endif
*if,elc,eq,1,then
*get,ena,elem,0,num,min
ittt=0
*do,jp,1,6,1
*if,arr(ena,jp),eq,en,then
ittt=1
*endif
*enddo
*if,ittt,eq,1,then
*cycle
*endif
*endif
*do,jl,1,elc,1
*get,ena,elem,0,num,min
esel,s,elem,,ena
esel,a,elem,,en
*get,xd,elem,ena,cent,x
*get,yd,elem,ena,cent,y
*get,zd,elem,ena,cent,z
xb=nx(b)
yb=ny(b)
zb=nz(b)
xe=nx(e)
ye=ny(e)
ze=nz(e)
xf=nx(f)
yf=ny(f)
zf=nz(f)
a1=xb-xa
a2=yb-ya
a3=zb-za
ae1=xe-xa
ae2=ye-ya
ae3=ze-za
ad1=xd-xa
ad2=yd-ya
ad3=zd-za
be1=xe-xb
be2=ye-yb
be3=ze-zb

```

```

bf1=xf-xb
bf2=yf-yb
bf3=zf-zb
aa=a2*ad3-a3*ad2
ab=a3*ad1-a1*ad3
ac=a1*ad2-a2*ad1
ba=ae2*ad3-ae3*ad2
bb=ae3*ad1-ae1*ad3
bc=ae1*ad2-ae2*ad1
ca=be2*bf3-be3*bf2
cb=be3*bf1-be1*bf3
cc=be1*bf2-be2*bf1
pa=aa*xa+ab*ya+ac*za
pb=ba*xa+bh*ya+bc*za
pc=ca*xb+cb*yb+cc*zb
md1=aa*bb*cc-aa*cb*bc-ba*ab*cc
md2=ba*ch*ac+ca*ab*bc-ca*bb*ac
md=md1+md2
mx1=pa*bb*cc-pa*ch*bc-ph*ab*cc
mx2=ph*ch*ac+pc*ab*bc-pc*bb*ac
mx=mx1+mx2
my1=aa*pb*cc-aa*pc*bc-ba*pa*cc
my2=ba*pc*ac+ca*pa*bc-ca*pb*ac
my=my1+my2
mz1=aa*bb*pc-aa*cb*pb-ba*ab*pc
mz2=ba*ch*pa+ca*ab*pb-ca*bb*pa
mz=mz1+mz2
*if,md,eq,0,then
cmsel,u,crset
*get,nca,elem,0,count
*if,nca,gt,0,then
esel,s,elem,,en
esel,a,elem,,ena
ewrite,nontriv,elem,,1
cmsel,a,crset
cm,crset,elem
*endif
cmsel,s,elgr
esel,u,elem,,ena
cm,elgr,elem
*cycle
*endif
xc=mx/md
yc=my/md
zc=mz/md
b1=xc-xa
b2=yc-ya
b3=zc-za
c1=xb-xd
c2=yb-yd

```

```
c3=zb-zd
d1=xc-xd
d2=yc-yd
d3=zc-zd
one=(a2*b3-a3*b2)*(c2*d3-c3*d2)
two=(a3*b1-a1*b3)*(c3*d1-c1*d3)
three=(a1*b2-a2*b1)*(c1*d2-c2*d1)
fin=one+two+three
*if,fin,ge,0,then
  cmsel,u,crset
  *get,nca,elem,0,count
  *if,nca,gt,0,then
    esel,s,elem,,en
    esel,a,elem,,ena
    ewrite,double,elem,,1
    cmsel,a,crset
    cm,crset,elem
  *endif
  *else
    arr(en,jn)=ena
  *endif
  cmsel,s,elgr
  esel,u,elem,,ena
  cm,elgr,elem
  *enddo
  *enddo
  cmsel,s,dummy
  esel,u,elem,,en
  cm,dummy,elem
  *enddo
fini
```

APPROVAL

ANSYS DUPLICATE FINITE-ELEMENT CHECKER ROUTINE

By R. Ortega

The information in this report has been reviewed for technical content. Review of any information concerning Department of Defense or nuclear energy activities or programs has been made by the MSFC Security Classification Officer. This report, in its entirety, has been determined to be unclassified.



J.C. BLAIR

Director, Structures and Dynamics Laboratory

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE May 1995	3. REPORT TYPE AND DATES COVERED Technical Memorandum
---	-----------------------------------	---

4. TITLE AND SUBTITLE ANSYS Duplicate Finite-Element Checker Routine	5. FUNDING NUMBERS
--	---------------------------

6. AUTHOR(S) R. Ortega

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) George C. Marshall Space Flight Center Marshall Space Flight Center, Alabama 35812	8. PERFORMING ORGANIZATION REPORT NUMBER
--	---

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) National Aeronautics and Space Administration Washington, DC 20546-0001	10. SPONSORING / MONITORING AGENCY REPORT NUMBER NASA TM - 108492
--	---

11. SUPPLEMENTARY NOTES Prepared by Structures and Dynamics Laboratory, Science and Engineering Directorate.
--

12a. DISTRIBUTION / AVAILABILITY STATEMENT Unclassified-Unlimited	12b. DISTRIBUTION CODE
---	-------------------------------

13. ABSTRACT (Maximum 200 words) An ANSYS finite-element code routine to check for duplicated elements within the volume of a three-dimensional (3-D) finite-element mesh was developed. The routine developed is used for checking floating elements within a mesh, identically duplicated elements, and intersecting elements with a common face. A space shuttle main engine alternate turbopump development high pressure oxidizer turbopump finite-element model check using the developed subroutine is discussed. Finally, recommendations are provided for duplicate element checking of 3-D finite-element models.

14. SUBJECT TERMS finite-element analysis, duplicate element, space shuttle main engine	15. NUMBER OF PAGES 18
	16. PRICE CODE NTIS

17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT Unlimited
--	---	--	--

National Aeronautics and
Space Administration
Code JTT
Washington, DC
20546-0001

*Official Business
Penalty for Private Use, \$300*

Postmaster: If Undeliverable (Section 158 Postal Manual), Do Not Return
