“INTRODUCTION to DMAP”

Special Topic Seminar

October 2000

MSC SOFTWARE
SIMULATING REALITY

Introduction to DMAP

- OBJECTIVE
- What is DMAP?
- Nastran Set Definitions and Structural Matrix Reductions
- Flow of Solution -- Common Subdmaps
- Key Modules
- Where Do I Insert a DMAP
- SSSALTERs
- What is a Qualifier?

PRESENTATION OBJECTIVE

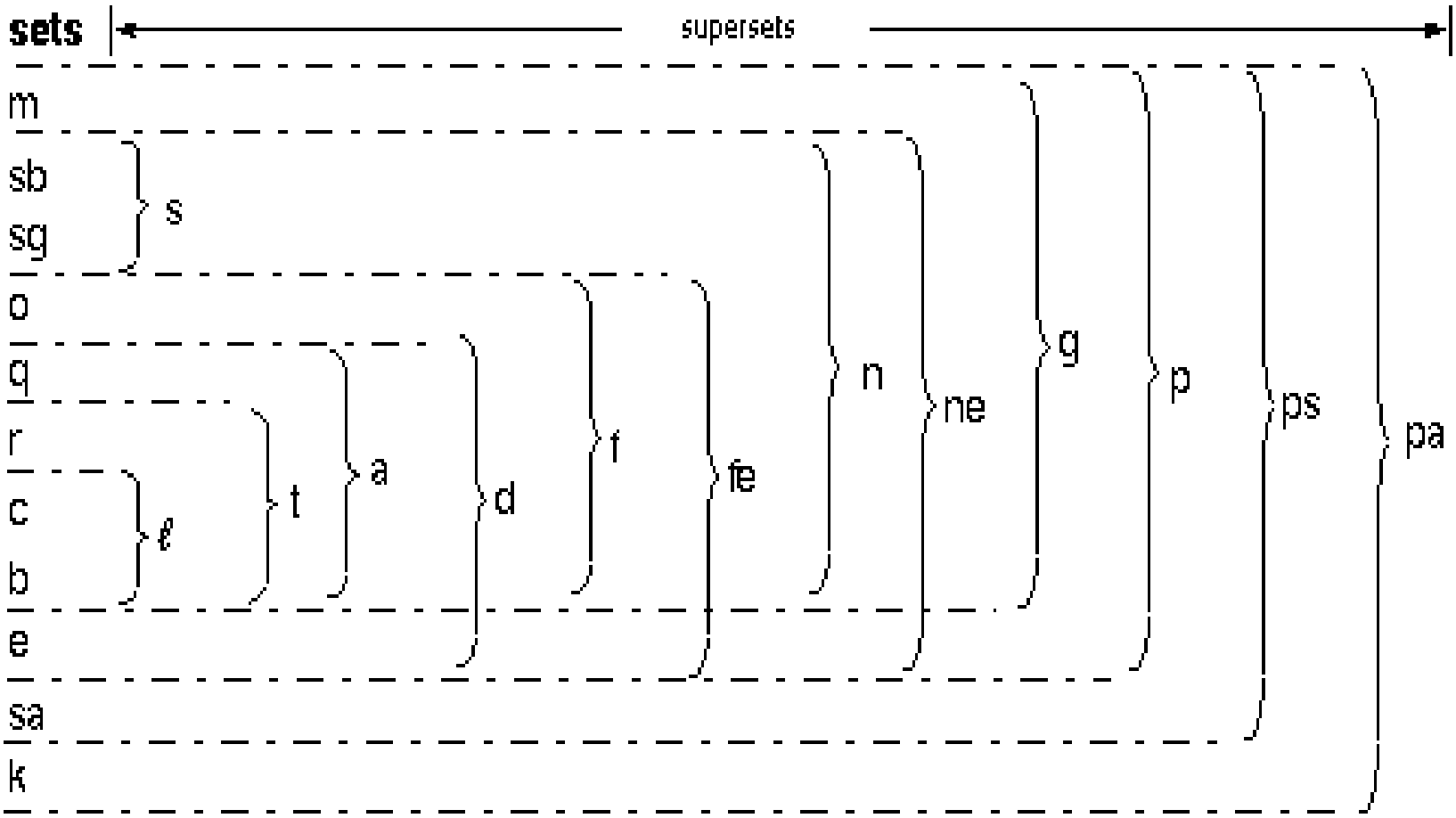
- **Current DMAPpers:**
  - **Reinforce concepts**
  - **Provide a few “tricks”**
  - **Encourage “string based” alters and MALTERs**
- **Potential DMAPpers:**
  - **Introduce NASTRAN set definitions**
  - **Provide insight to the solution process**

What is DMAP?

- DMAP (deemap) Direct Matrix Abstraction Program
  - Developed as “modular” approach to perform linear algebra manipulations on structural matrices
  - MSC.Nastran uses a specific sequence of DMAP statements to solve FEM solutions
- Structured Solution Sequences
  - Subdmaps (similar to subroutines) are collected to perform common tasks
    - Example: SEMG -- Super Element Matrix Generation
      - Used by Sol 101, 103, 105, 106, 111, etc. to assemble the mass, stiffness and other structural matrices for a superelement
    - Helps in writing DMAP ALTERs to modify standard solution

Nastran Set Definitions

Reference: Appendix B, MSC.Nastran Quick Reference Guide



MSC SOFTWARE
SIMULATING REALITY

Structural Set Reductions -- Stiffness matrix

- **Step 1, eliminate dependent dof for MPCs (RBEs)**
  - **Subdmap SEKR0**

Gset = All grids and spoints
Nset = Independent dof
Mset = Dependent dof

$$\left[K_{gg} \right] \Rightarrow \left\{ \begin{array}{c|c} \overline{K}_{nn} & K_{nm} \\ \hline K_{mn} & K_{mm} \end{array} \right\}$$

MCE1 creates Gmn (xformation matrix)
MCE2 creates KNN
KMM is partition

$$[K_{nn}] = [G_{mn}]^T[K_{mm}G_{mn}+K_{mn}] + [K^T_{mn}G_{mn}+\overline{K}_{nn}]$$

Structural Set Reductions -- Stiffness matrix

- **Step 2, partition Single Point Constraints (SPCs)**
  - **Subdmap SEKR**

Nset = Independent dof
Fset = Free dof
Sset = Constrained (SPC) dof

$$[K_{nn}] \Rightarrow \left\{ \begin{array}{c|c} K_{ff} & K_{fs} \\ \hline K_{sf} & K_{ss} \end{array} \right\}$$

This is a simple UPARTN operation

Note: If AUTOSPC is selected, GPSP module checks Knn for Singular stiffness dof, and modifies USET table prior to partition

Structural Set Reductions -- Stiffness matrix

- **Step 3, Static Condensation, Physical dof**
  - **Subdmap SEKR**

Fset = Free dof
Tset = Total set of physical dof (Subset of Aset)
Oset = Omitted dof

$$\left[K_{ff} \right] \Rightarrow \left\{ \begin{array}{c|c} \overline{K}_{tt} & K_{to} \\ \hline K_{ot} & K_{oo} \end{array} \right\}$$

DCMP and FBS solve
$[K_{oo}][G^T_{ot}] = -[K_{ot}]$
for G^T_{ot}

$$[K_{tt}] = [\overline{K}_{tt}] + [K_{ot}][G^T_{ot}]$$

Structural Set Reductions -- Stiffness matrix

- **Step 4, Assemble Aset**
  - **Subdmap SEKMR**

Aset = Assembled (Analysis) set dof
Tset = Total set of physical dof (Subset of Aset)
Qset = Modal dof

$$\left\{\begin{array}{c|c} \mathbf{K_{tt}} & \mathbf{K_{tq}} \\ \hline \mathbf{K_{qt}} & \mathbf{K_{qq}} \end{array}\right\} \Rightarrow \left[\mathbf{K_{aa}}\right]$$

For CMS K_{tq}, K_{qt} are null
M_{tq}, M_{qt} will have coupling terms

Physical dof (Ktt) UMERGEd to Aset Size If Component Modes Synthesis (CMS) is performed, then Qset will exist. CMS performed in subdmap SEMR3

ASET in Statics

- **Further reductions occur on the Aset in statics if inertia relief is present (SUPORT entries)**
  - **See V68 Reference Manual Sections 9.4.1 and 9.4.9 for discussion of inertia relief and Rset**

Tset = Total set of physical dof (same as Aset for statics)
Lset = left-over set
Rset = Reference set (for inertia relief)

$$\left[K_{aa}\right] \Leftrightarrow \left[K_{tt}\right] \Rightarrow \left\{\begin{array}{c|c} K_{ll} & K_{lr} \\ \hline K_{rl} & K_{rr} \end{array}\right\}$$

$$\{P_l\} = [K_{ll}]\{U_l\}$$

ASET in Statics

- **Static Solution messages:**

.f04 file message for solution

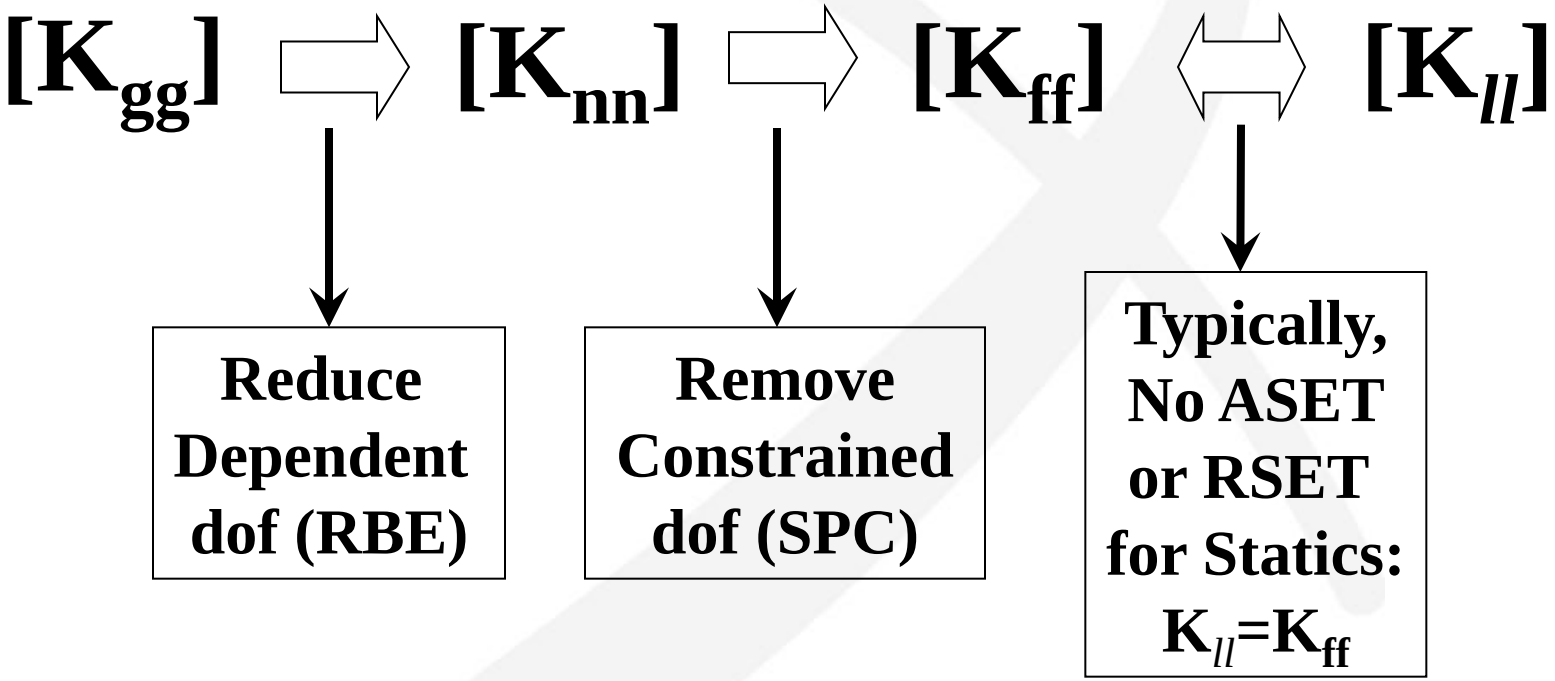
***** USER INFORMATION MESSAGE 4157 (DFMSYN)**
PARAMETERS FOR SPARSE DECOMPOSITION OF DATA BLOCK KLL

.f06 file message for solution

***** SYSTEM INFORMATION MESSAGE 6916 (DFMSYN)**
DECOMP ORDERING METHOD CHOSEN: BEND, ORDERING METHOD USED: BEND
***** USER INFORMATION MESSAGE 5293 (SSG3A)**
FOR DATA BLOCK KLL

LOAD SEQ. NO.	EPSILON	EXTERNAL WORK
1	-6.0279293E-13	1.6411982E+01

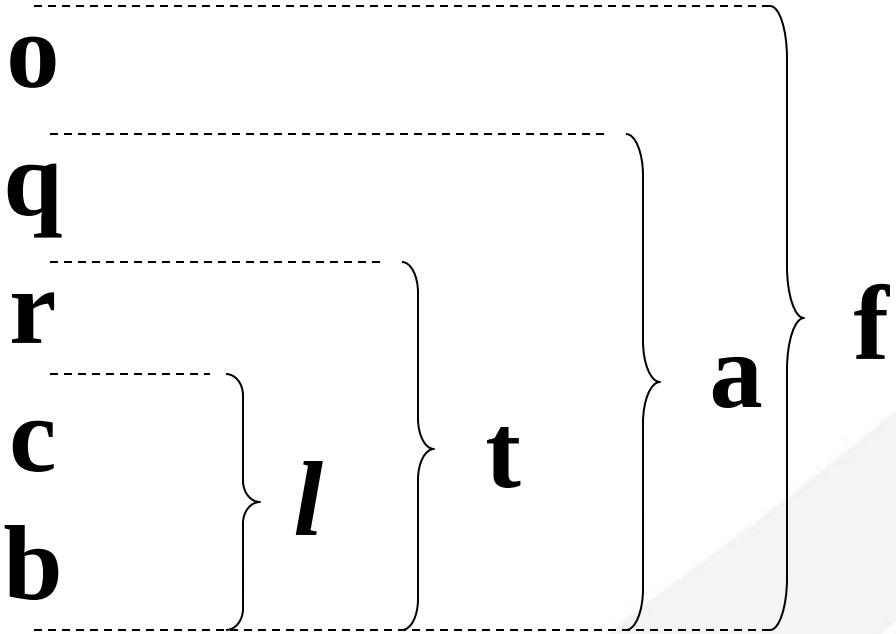
Typical Static Solution Sets



MSC SOFTWARE
SIMULATING REALITY

ASET in Normal Modes with Component Modes Synthesis (CMS)

- **Objective: Calculate modal dof and store in K_{qq}, M_{qq}, M_{qt}, M_{tq} (Shown here w/o GDR)**



Bset = dof fixed during CMS
Cset = dof free during CMS
Rset = reference set (used for "clean" rbmodes)

NEW SET FOR CMS:
Vset = Free to Vibrate Set
= o+c+r

MSC SOFTWARE SIMULATING REALITY

ASET in Normal Modes with Component Modes Synthesis (CMS)

- **Step 1: Partition Vset**

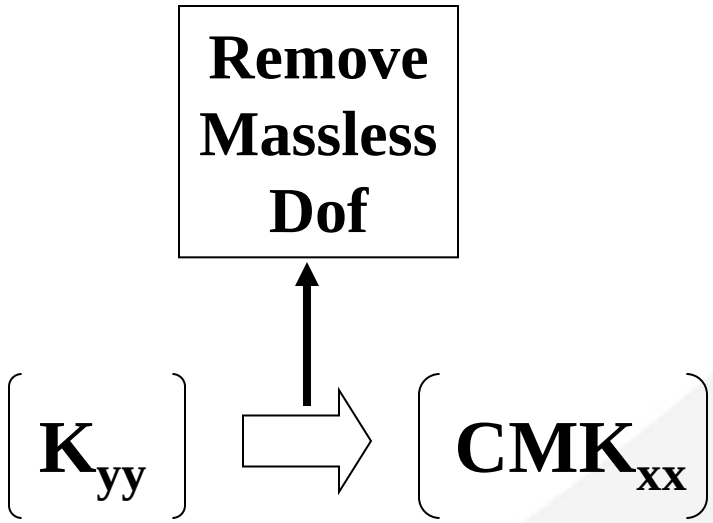
$$\left[K_{ff} \right] \Rightarrow \left\{ \begin{array}{c|c} K_{vv} & K_{vq} \\ \hline K_{qv} & K_{qq} \end{array} \right\}$$

- **Step 2: Perform GDR if requested (never recommended) to form K_{yy}, otherwise $K_{yy} = K_{vv}$**

$$\left[K_{vv} \right] \Leftrightarrow \left[K_{yy} \right]$$

ASET in Normal Modes with Component Modes Synthesis (CMS)

- **Step 3: Remove null rows from mass matrix to form CMK_{xx} and CMM_{xx} , otherwise $K_{vv} = CMK_{xx}$**



Solve Eigensolution CMLAMA, CMPHIXZ

MSC SOFTWARE SIMULATING REALITY

ASET in Normal Modes with Component Modes Synthesis (CMS)

- **Step 4, Reassemble modal dof into Kaa, Maa**
  - **add residual vectors (static shapes)**
  - **diagonalize Kqq, Mqq**
  - **form Mqt, Mtq**

$$\left\{\begin{array}{c|c} K_{tt} & K_{tq} \\ \hline K_{qt} & K_{qq} \end{array}\right\} \Rightarrow \left[K_{aa}\right]$$

For CMS K_{tq}, K_{qt} are null
M_{tq}, M_{qt} will have coupling terms

Bulk Data Entry
Effect on Set Definitions

Name	Bulk Data Entry Name
m	MPC, MPCADD, MPCAX, POINTAX, RBAR, RBE1, RBE2, RBE3, RROD, RSPLINE, RTRPLT, GMBC, GMSPC*
sb	SPC, SPC1, SPCADD, SPCAX, FLSYM, GMSPC*, BNDGRID, (PARAM,AUTOSPC,YES)
sg	GRID, GRIDB, GRDSET (PS field)
o	OMIT, OMIT1, OMITAX, GRID (SEID field), SESET
q	QSET, QSET1
r	SUPORT, SUPORT1, SUPAX
c	CSET, CSET1
b	BSET, BSET1
e	EPOINT
sa	CAEROi
k	CAEROi
a	ASET, ASET1, Superelement exterior degrees of freedom, CSUPEXT

PRINTING USET DEFINITIONS

- **TO PRINT USET DEFINITIONS IN .F06 FILE:**
  - **PARAM,USETPRT,isort**
    - **-1, no print**
    - **0, internal row sort**
    - **1, internal column sort**
    - **2, internal row & column sort**
    - **10, external row sort**
    - **11, external column sort**
    - **12, external row & column sort**
  - **PARAM, USETSEL,**

V70.5 (and prior) USET DEFINITIONS

Set	Bit No.	Dec Eqiv
U1	31	2147483648
U2	30	1073741824
U3	29	536870912
U4	28	268435456
U5	27	134217728
U6	26	67108864
V	25	33554432
FR	24	16777216
T	23	8388608
Q	22	4194304
B	21	2097152
C	20	1048576
PA	19	524288
K	18	262144
SA	17	131072
PS	16	65536

Set	Bit No.	Dec Eqiv
D	15	32768
FE	14	16384
NE	13	8192
P	12	4096
E	11	2048
SB	10	1024
SG	9	512
L	8	256
A	7	128
F	6	64
NE	5	32
G	4	16
R	3	8
O	2	4
S	1	2
M	0	1

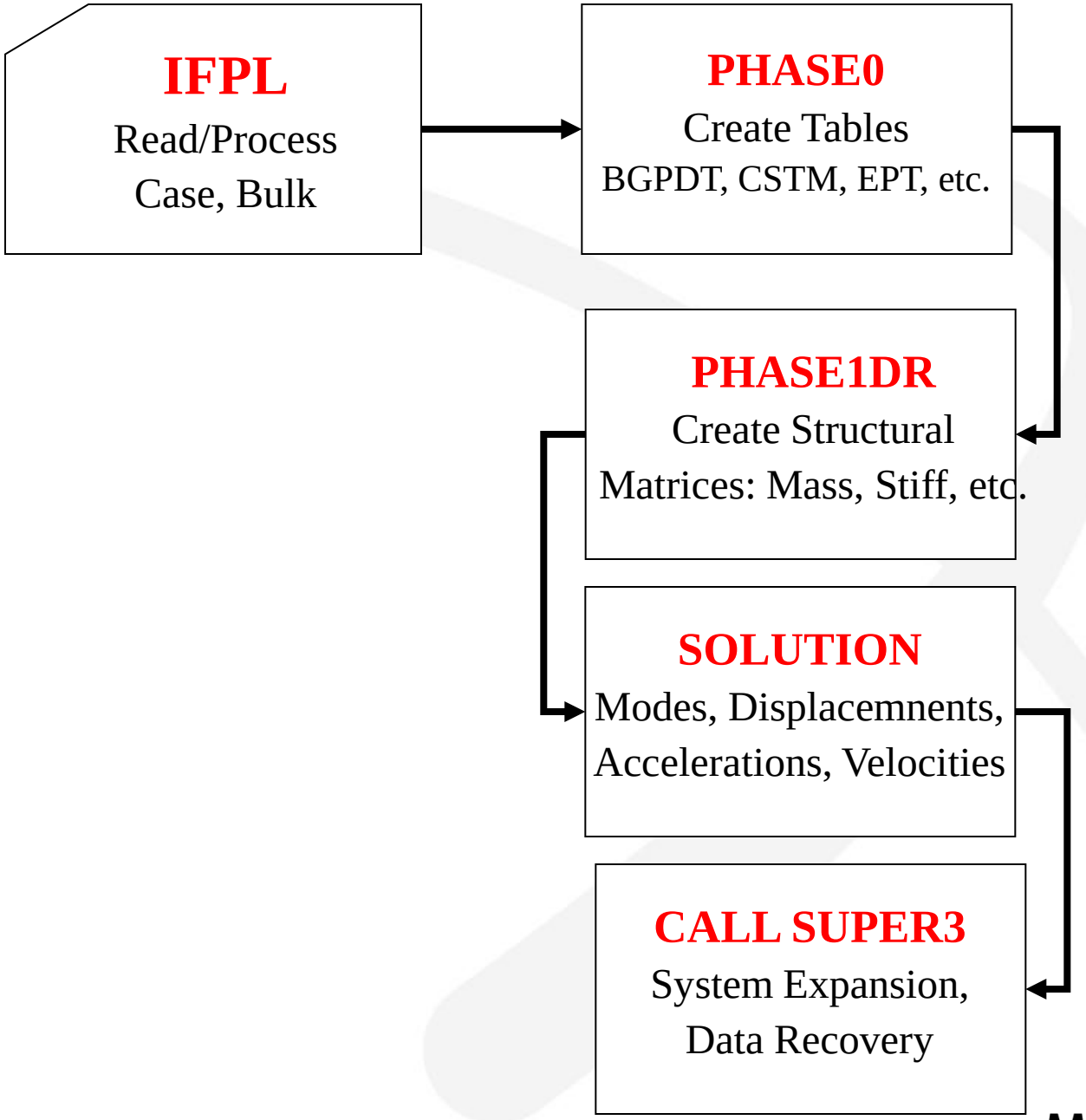
V70.7+ USET DEFINITIONS

Set	Bit	Dec Equiv
U1	31	2147483648
U2	30	1073741824
U3	29	536870912
U4	28	268435456
U5	27	134217728
U6	26	67108864
Q	22	4194304
BE	21	2097152
C	20	1048576
K	18	262144
SA	17	131072
E	11	2048
SB	10	1024
SG	9	512
R	3	8
O	2	4
BF	1	2
M	0	1

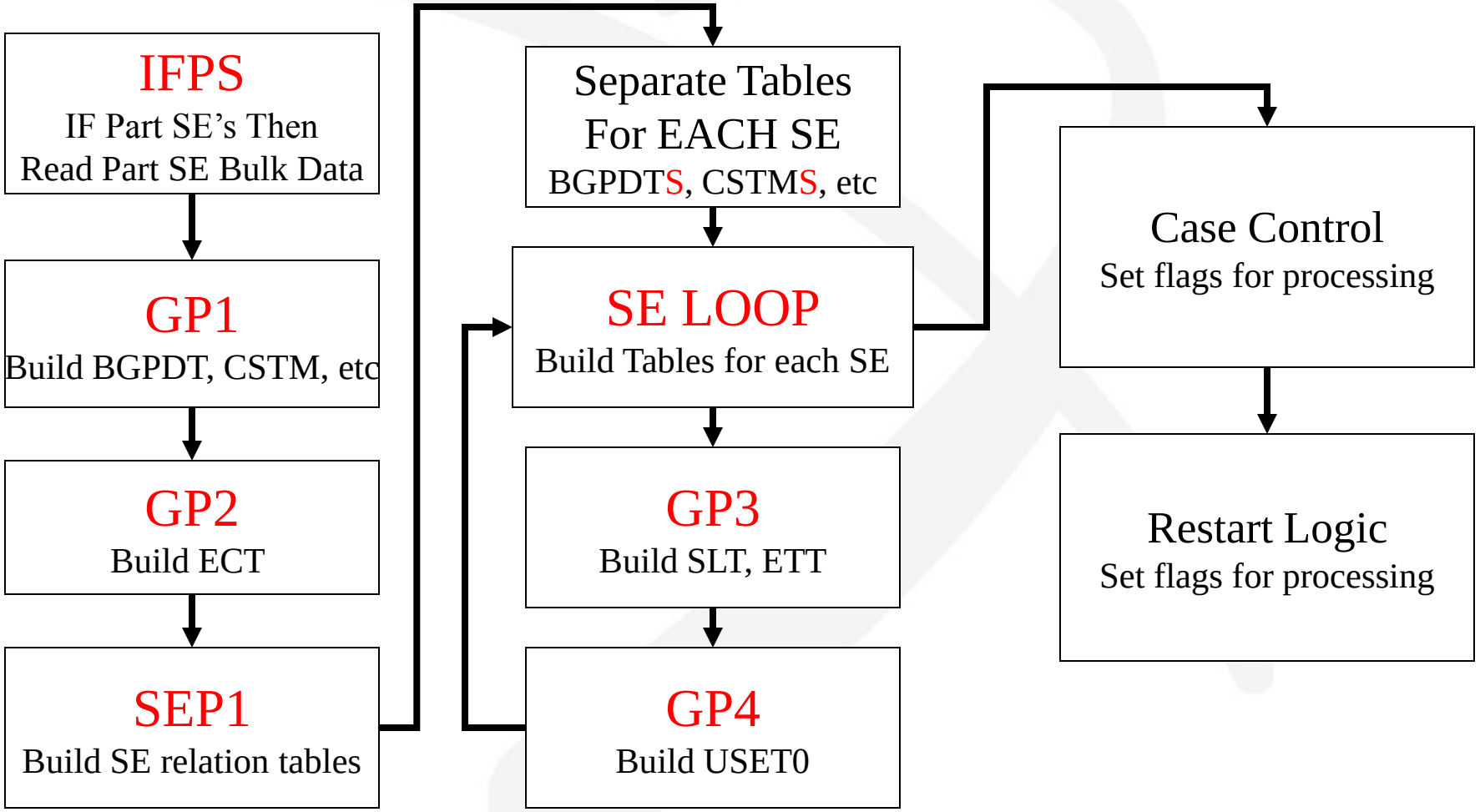
Sets	Dec Eqiv
V=O+R+C	1048588
FR=O+C+BE+BF	3145734
T=R+C+BE+BF	3145738
PA=all sets	7736847
PS=All but K	7474703
D=BE+BF+C+R+Q+E	7342090
FE=BE+BF+C+R+Q+E+O	7342094
NE=BE+BF+C+R+Q+E+O+SB+SG	7343630
P=All but SA and K	7343631
L=BE+BF+C	3145730
A=BE+BF+C+R+Q 7340042	
F=BE+BF+C+R+Q+O	7340046
N=BE+BF+C+R+Q+O+SB+SG	7341582
G=BE+BF+C+R+Q+O+SB+SG+M	7341583

Common Subdmap Flow

- Main DMAP (SESTATIC, SEMODES, SEMTRAN, etc)
  - CALL IFPL
    - Input File Processor Listings
      - Interpret Bulk, Case and Create tables for further processing
  - CALL PHASE0
    - Creates Matrices/Tables for use in creating stiffness, mass, etc.
      - Basic Grid Point Definition Table (BGPDTs), Coordinate System Transformation Matrix (CSTMs), etc.
  - CALL PHASE1DR
    - Builds Structural Matrices for Solution
      - KGG, MGG, BGG | KAA, MAA, BAA, etc.
  - SOLVE for SYSTEM SOLUTION VECTORS
    - SOL 103 -- CALL MODFSRS --
      - solves for LAMA,PHA (eigenvalues, and eigenvectors)
    - SOL 101 -- CALL STATRS --
      - solves for UL = Displacements
  - CALL SUPER3
    - Expand solution, perform data recovery (stress, force, mpcforce, etc.)

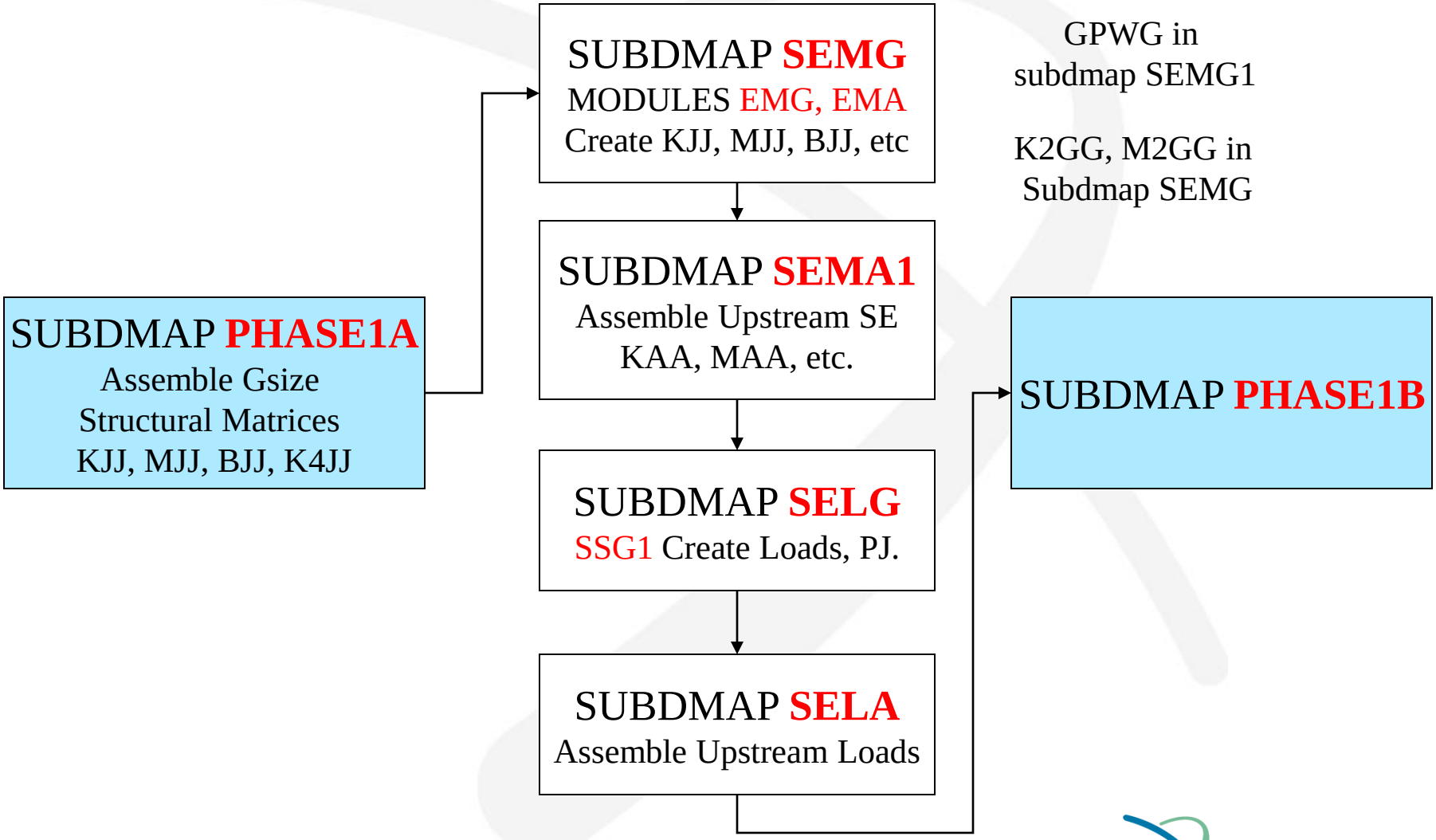


PHASE0 Primary Operations



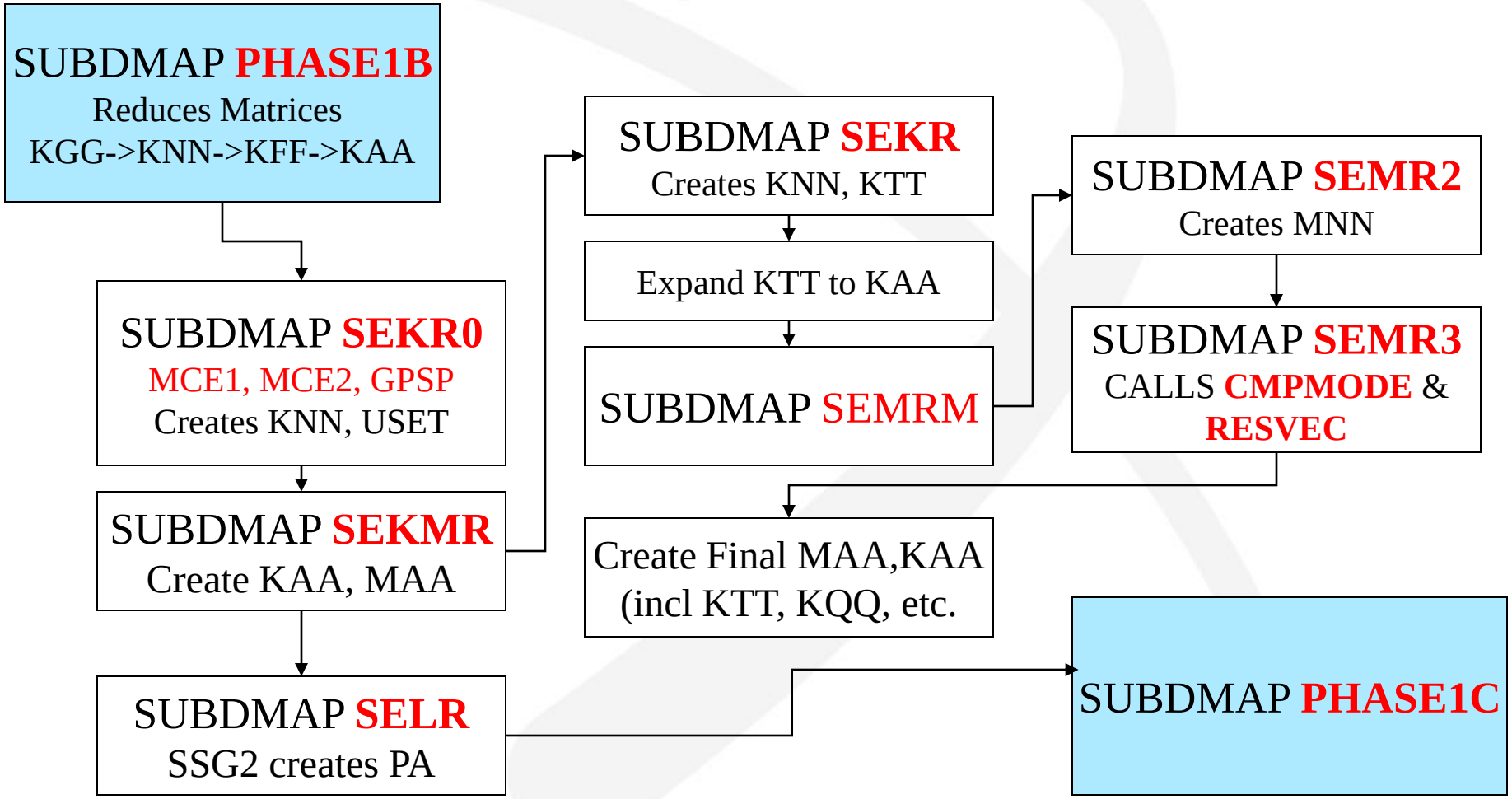
MSC SOFTWARE
SIMULATING REALITY

PHASE1DR Primary Operations



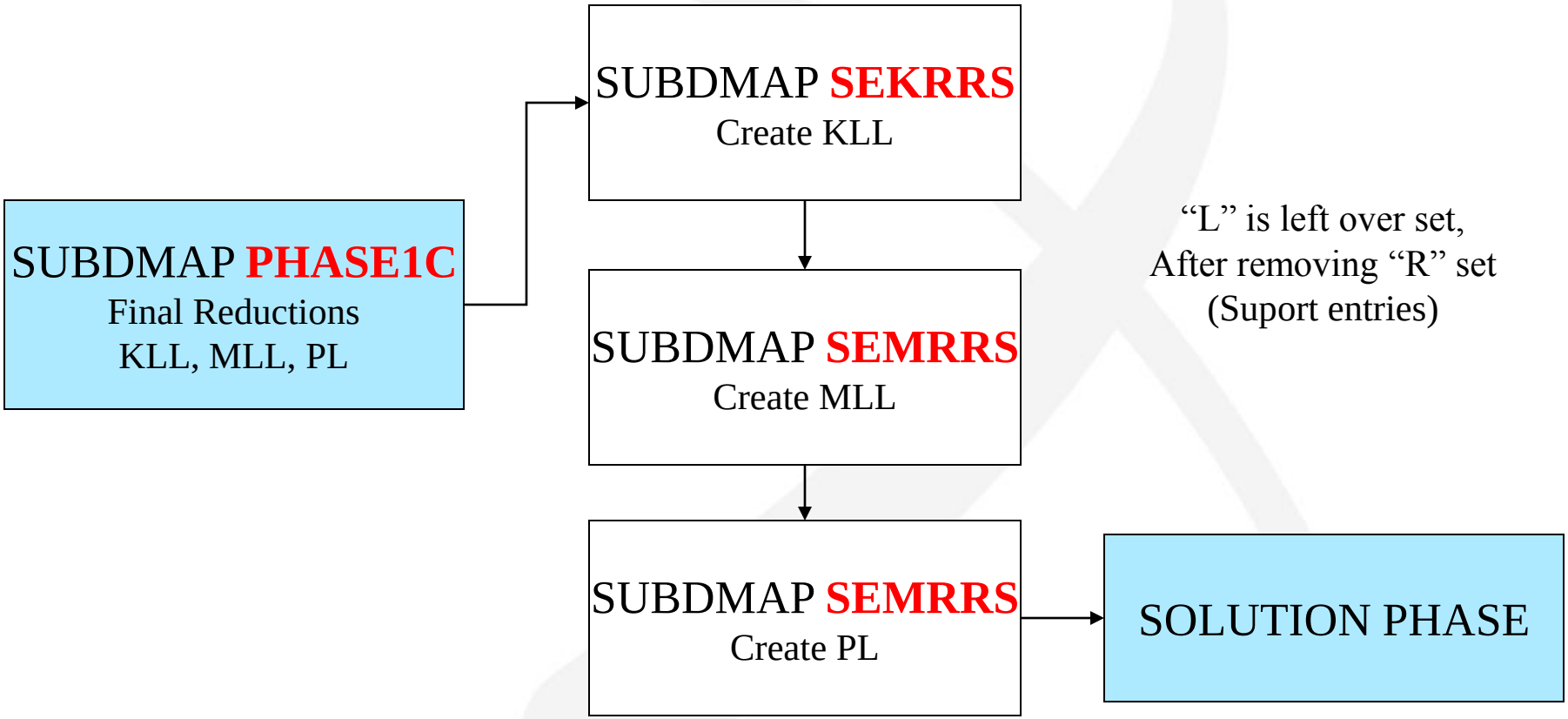
MSC SOFTWARE
SIMULATING REALITY

PHASE1DR Primary Operations



MSC SOFTWARE
SIMULATING REALITY

PHASE1DR Primary Operations



MSC SOFTWARE
SIMULATING REALITY

Solution Phase

SESTATIC (SOL 101)

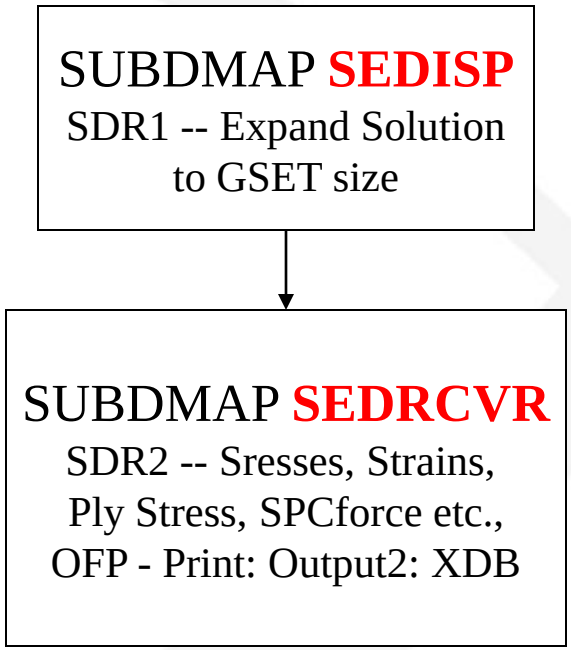
SUBDMAP **STATRS**
Solves for displacements
Create Solution Vector UL

SEMODES (SOL 103)

SUBDMAP **MODEFSRS**
Combines Fluid & Structure
Create Solution Vector PHA

SUBDMAP **MODERS**
READ MODULE for
Eigenvalue solutions

SUPER3 operations



MINIMUM STATIC SOLUTION

```
IFP1   $ case control
XSORT  $ sort bulk data
IFP    $ create geometry
GP1    $ process geometry
GP2    $ element connectivity processing
GP3    $ process static and thermal loads
TA1    $ create element tables
EMG    $ create element stiffness
EMA    $ assemble KGG
GP4    $ create dof table
UPARTN $ partition Kgg to Kff=Kaa=Ktt=Kll
SSG1   $ generate loads
SSG2   $ reduce loads
DCMP   $ decompose stiffness
SSG3   $ for Lset=Aset disp matrix
SDR1   $ expand to Gset disp matrix
SDR2   $ calculate disp, stress, strain, spcf, etc.
OFP    $ print output
END    $ required
```

Only19 lines!!

Barebones Solution

- **Example linear static solution**
  - **Limitations**

```
$ BAREBONES solution for statics
$ stiffness only, no mass
$ MPC/RBE allowed, but no mpcforce data recovery
$ "simple only" -- no checking
$ structural H-elements only (no P-elements)
$ "simple" case control (no output(post), etc.)
$ no k2gg or genel
$ no param,snorm
$ no inertia relief
$ no ASET, BSET, CSET, RSET, QSET, or OMIT
$ no TABLEij processing
$ no MPCForce requests
$ no param,post
$ no composite stresses.
```

Barebones Solution

```
$ ================================
$ read bulk and case control, create tables
$ ================================
IFP1 /CASECC,,,,/S,N,NOGOIFP1/S,N,LASTCC/S,N,BEGSUP $
XSORT,  ,/BULK/S,N,NOGOXSRT $
IFP BULK/
   GEOM1,EPT,MPT,EDT,DIT,DYNAMIC,GEOM2,
   GEOM3,GEOM4,EPTA,UNUSED11,MATPOOL,AXIC,PVT,DMI,
   DMINDX,DTI,DTINDX,DEFUSET,EDOM,DEQATN,DEQIND,
   CONTACT,OINT,UNUSED25/
   S,N,NOGOIFP/S,N,RUNIFP3/S,N,RUNIFP4/
   S,N,RUNIFP5/S,N,RUNIFP6/S,N,RUNIFP7/S,N,RUNIFP8/
   S,N,RUNIFP9/SEID/S,N,RUNMEPT $
```

Barebones Solution

```
$ ==================================
$ basic geometry processing
$ ==================================
GP1 GEOM1,GEOM2,,,,,/
    GPL,EQEXIN,GPDT,CSTM,BGPDT,SIL,/
    S,N,LUSET/S,N,NOCSTM/S,N,NOPOINTS/////-1 $
$ element connectivity processing
GP2 GEOM2,EQEXIN,,,/ECT,/s,n,acoustic $
$ process static and thermal loads
GP3 GEOM3,BGPDT,GEOM2,,,,,/SLT,ETT/
    S,N,NOLOAD/S,N,NOGRAV/S,N,NOTEMP $
$ ==================================
$ combine element data into tables
$ ==================================
TA1 MPT,ECT,EPT,BGPDT,SIL,ETT,CSTM,,,/
    EST,,GEI,GPECT,ESTL,VGFD,/
    LUSET/S,N,NOESTL/S,N,NOSIMP/NOSUP/S,N,NOGENL/-1/// $
```

Barebones Solution

```
$ ==================================
$ generate element stiffness matrix
$ ==================================
EMG EST,CSTM,MPT,,,,ETT,EDT,,,BGPDT,,,,,/
    KELM,KDICT,,,,/ S,N,NOKGG/S,N,NOMGG/
    ////TEMPSID/DEFRMSID/PENFAC //
    //////v,y,K6ROT ///////// S,N,BADMESH $
$ ==================================
$ assemble KGG
$ ==================================
EMA GPECT,KDICT,KELM,BGPDT,SIL,CSTM,,/KGG,// $
```

Barebones Solution

```
$ ==================================
$ create dof table
$ ==================================
GP4 CASECC,GEOM4,EQEXIN,SIL,GPDT,BGPDT,CSTM,,,,GEOM2,/
   RMG,YS0,USET0,/
   LUSET/S,N,NOMSET/S,N,MPCF2/S,N,NOSSET/S,N,NOOSET/
   S,N,NORSET/S,N,NSKIP/S,N,REPEAT/S,N,NOSET/S,N,NOL/
   S,N,NOA/SEID/ALTSHAPE/SEBULK $
```

Barebones Solution

```
$ ==================================
$ reduce out dependent dof
$ ==================================
PURGEX   /GM,,,,/NOMSET $
EQUIVX   KGG/KNN/NOMSET $
IF ( MPCF2>-1 ) THEN $
  UPARTN   USET0,KGG/KMG,,,/'G'/'M'/'N'/1 $
  MCE1     USET0,RMG/GM $
  MCE2     USET0,GM,KGG,,,/KNN,,, $
  UPARTN   USET0,KGG/KMM,,,/'G'/'M'/'N' $
ENDIF $
```

Barebones Solution

```
$ ==================================
$ grid point singularity processor
$ ==================================
GPSP    KNN,KMM,USET0,SIL,GPL,YS0,GEOM4,EQEXIN/
        USET,YS/
        S,N,NOSSET/V,Y,AUTOSPC='YES'/
        V,Y,PRGPST='YES'/V,Y,SPCGEN=0/
        V,Y,EPZERO=1.E-8/0/S,N,SING/V,Y,EPPRT=1.E-8/
        S,N,NOSET/S,N,NGERR/RGTYPE $
```

Barebones Solution

```
$ ====================================
$ spc elimination
$ ====================================
IF (NOSSET<0) THEN
  EQUIVX   KNN/KFF/NOSSET $
ELSE
  UPARTN   USET,KNN/KFF,KSF,KFS,KSS/'N'/'F'/'S' $
ENDIF $
$ ====================================
$ do not allow aset, omit, qset, etc: KFF=KAA=KTT=KLL
$ ====================================
EQUIVX KFF/KLL/ALWAYS $ kll=Left over set
```

Barebones Solution

```
$ ==================================
$ process loads
$ ==================================
LCGEN    CASECC,SLT,ETT/CASESX/0/1 $
SSG1     SLT,BGPDT,CSTM,,EST,MPT,ETT,EDT,,CASESX,,,,,,,/
         PG,PTELEM,/
         V,N,LUSET/V,N,NSKIP $
$ apply constraints to static load vectors
IF (nosset>0) THEN
   SSG2     USET,GM,YS,KFS,,,PG/QR,PO,PS,PA,PL $
ENDIF $
```

Barebones Solution

```
$ ==================================
$ solve...
$ ==================================
DCMP    USET,SIL,EQEXIN,KLL,,/LLL,,LRSEQ/
        1/0/BAILOUT/MAXRATIO/'L'/1.E-20///////
        S,N,SING/S,N,NBRCHG/S,N,ERR=0 $
SSG3    LLL,,KLL,PL,,,,,LRSEQ/
        UL,,RULV,,/V,N,OMIT/
        0/V,N,NSKIP/S,N,EPSI $
$ ==================================
$ expand solution vector (UL) to GSET size
$ ==================================
SDR1    USET,PG,UL,,YS,,GM,PS,KFS,KSS,QR/UG,PGG,QG/
                V,N,NSKIP/'STATICS'/NOSPC $
```

Barebones Solution

```
$ ==================================
$ calculate output requests
$ ==================================
SDR2    CASESX,CSTM,MPT,,EQEXIN,,ETT,EDT,BGPDT,PGG,
        QG,UG,EST,,,,,,,,/
        OPG1,OQG1,OUGV1,OES1,OEF1,/
        'STATICS'/S,N,NOSORT2/V,Y,NOCOMPS=1 $
$ print output
OFP OPG1,OQG1,OUGV1,OES1,OEF1// $
END
```

Selected DMAP Modules

- **LOGICAL CONSTRUCTIONS**
  - **IF - THEN - ELSE - ENDIF -- logical branch**
  - **DO WHILE - ENDDO -- counter loop**
  - **EXIT -- terminates program**
- **READ/WRITE operations**
  - **DMIIN -- process DMI bulk data entries**
  - **DTIIN -- process DTI bulk data entries**
  - **MTRXIN -- process DMIG bulk data entries**
  - **INPUTT2/OUTPUT2 -- read/write tables**
  - **INPUTT4/OUTPUT4 -- read/write matrices**

Selected DMAP Modules

- **PRINTING MATRICES**
  - **MATGPR -- print a matrix and associated dof**
  - **MATPRN -- print matrix (just numbers, no dof map)**
- **PRINTING MESSAGES (very useful)**
  - **MESSAGE -- prints a message to the .f06 file.**
- **INTERROGATING MATRICES**
  - **PARAML**
    - **Extract single value (row,col)**
    - **Get Trailer info**

Selected DMAP Modules

- **PARTITION/MERGE OPERATIONS**
  - **MERGE -- merge matrices using partition vectors**
  - **UMERGE -- merge matrices using USET definitions**
  - **PARTN -- partition matrices using partition vectors**
  - **UPARTN -- partition matrices using USET defs.**
- **MATH OPERATIONS**
  - **ADD -- add, subtract, multiply, divide, 2 matrices**
  - **MPYAD -- matrix multiplication**
  - **SMPYAD -- triple product matrix multiplication**

Selected DMAP Modules

- **SPECIAL MANIPULATIONS**
  - **APPEND -- append 2 matrices (useful in DO loops)**
  - **DIAGONAL -- extract diagonals from a matrix, raise matrix to a power**
  - **EQUIVX -- equivalence datablocks**
  - **MATGEN -- generate various matrices**
    - **Identity, Pattern matrix, psuedorandom matrix, partitioning matrix, null matrix, internal-external sort matrix**
  - **MATMOD -- modify matrices. -- many options.**
  - **VECPLOT -- resultants, transforms, rbmodes**

Selected DMAP Modules

- **“FAVORITE” MODULES**
  - **GP4 -- creates “raw” USET table (USET0)**
  - **GPSP -- AUTOSPC, creates “final” USET table**
  - **GPWG -- Grid Point Weight Generator**

Selected DMAP Modules

- **COMPUTATIONAL MODULES**
  - **DCMP -- decompose a matrix**
  - **FBS -- forward backward substitution**
  - **READ -- eigensolution**
  - **SOLVE -- [A][X]=[B], solve for X -- better to use dcmp & fbs**

WHERE DO I PUT AN “ALTER”?

- **What is a DMAP ALTER?**
  - **The ALTER is a way to modify the standard solution sequence.**
  - **Example -- OUTPUT4 MAA,KAA after CMS:**

THE “ALTER”

Old Style: line number

```
compile phase1dr,nolist,noref
alter 164
output4 maa,kaa,,,//-1/13/1//15 $
```

New Style: String Based

```
OR:
compile phase1dr,nolist,noref
alter ‘malter.*kaa.*maa’
output4 maa,kaa,,,//-1/13/1//15 $
```

String Based (w/o compile!)

```
OR:
malter 'malter.*kaa.*maa'
output4 maa,kaa,,,//-1/13/1//15 $
```

THE “ALTER”

- **What is a “STRING BASED” alter?**
  - **Searches SUBDMAP for unique string rather than line number**
  - **ALTER ‘unique string’(occurance,offset)**
  - **Advantage: 95+% success rate converting from one version to the next.**
- **What is an “MALTER”?**
  - **Searches entire solution sequence for a unique string**
  - **Don’t need to know subdmap name or line number**
  - **“built in” malters**

MALTER:TOP OF PHASE 1 SUPERELEMENT LOOP AFTER PARAMETERS AND
MALTER:QUALIFIERS SET

MALTER:AFTER UPSTREAM SUPERELEMENT MATRIX AND LOAD ASSEMBLY
MALTER:(KGG, BGG, MGG, K4GG, PG)

MALTER:AFTER SUPERELEMENT MATRIX AND LOAD REDUCTION TO A-SET,
MALTER:STATIC AND DYNAMIC (KAA, MAA, BAA, K4AA, PA)

MALTER:BOTTOM OF PHASE 1 SUPERELEMENT LOOP

MALTER:AFTER SUPERELEMENT DISPLACEMENT RECOVERY (UG)

MALTER:BOTTOM OF SUPERELEMENT DATA RECOVERY LOOP

MALTER:AFTER PREFACE MODULES

MALTER:AFTER TOTAL SUPERELEMENT STIFFNESS, VISCOUS DAMPING, AND MASS
MALTER:FORMULATED, STRUCTURAL + DIRECT INPUT (KJJ, BJJ, MJJ)

MALTER:AFTER SUPERELEMENT LOAD GENERATION (PJ)

MALTER:AFTER ELEMENT STRESS, STRAIN, ETC. DATA RECOVERY, SORT1
MALTER:(OUGV1, OES1, OEF1, ETC.)

MALTER:AFTER ELEMENT STRESS, STRAIN, ETC. DATA RECOVERY, SORT2
MALTER:(OUGV2, OES2, OEF2, ETC.)

MALTER:AFTER X2PP MATRICES READ (K2PP, M2PP, B2PP)

MALTER:AFTER SUPERELEMENT STIFFNESS, VISCOUS DAMPING, MASS, AND
MALTER:ELEMENT STRUCTURAL DAMPING GENERATION (KJJZ, BJJX, MJJX, K4JJ)

MALTER:AFTER X2JJ MATRICES READ (K2JJ, M2JJ, B2JJ)

Matrix Trailer

```
*8** Module  DMAP Matrix      Cols      Rows  F T     NzWds     Density
     EMA     110  KJJZ         132       132  6 2       36 8.73508D-02
(continued)
BlockT  StrL   NbrStr    BndAvg    BndMax    NulCol
4     2      621       66        81        22 *8**
```

Word	Contents
1	Number of columns in matrix
2	Number of rows in matrix
3	Form of the matrix
4	Type of matrix
5	Largest number of nonzero words among all columns
6	Density of the matrix multiplied by 10000
7	Size in blocks
8	Maximum string length over all strings
9	Number of strings
10	Average bandwidth
11	Maximum bandwidth
12	Number of null columns

Turn on DIAG,8 in Executive Control Section (after SOL, before CEND)

Matrix Forms & Types

Form	Meaning
1	Square
2	Rectangular
3	Diagonal
4	Lower triangular factor
5	Upper triangular factor
6	Symmetric
8	Identity
9	Pseudoidentity
10	Cholesky factor
11	Trapezoidal factor
13	Sparse lower triangular factor
15	Sparse upper triangular factor

Type	Meaning
1	Real, single precision
2	Real, double precision
3	Complex, single precision
4	Complex, double precision

SSSALTERs

- MSC provides several useful alters as part of the delivery of Nastran.
  - These may already solve the problem you are trying to solve
  - A good source of “go-by” DMAP alters
- SSSALTER location:
  - /msc/msc707/nast/misc/sssalter/*
  - INCLUDE ‘SSSALTERDIR:alternname’
    - SSSALTERDIR is a “symbol” that Nastran understands

SSSALTERs

- Over 60 alters available--see readme file
  - All alters contain a small sample and description
- Commonly used SSSALTERs:
  - checka -- multilevel strain energy checks
  - effmassa -- modal effective mass (sol 103)
  - nlgyroa -- nonlinear gyroscopic effects
  - alter1ha, alter2ha, alter3ha -- coupled loads analysis
  - appenda -- combine eigensolutions
  - delmodea -- remove a mode from solution
  - mfreqea -- contribution of each mode to strain energy

SSSALTERs

- **Commonly used SSSALTERs (cont):**
  - **mica -- initial conditions in modal transient**
  - **modevala -- checks for adequate modal content for a specified load**
  - **oloadcda -- oload resultant in user-specified coord sys**
  - **premaca -- cross ortho & MAC between Gset & Aset**
  - **postmaca -- cross-ortho & MAC between model/test**
  - **sebloada -- dynamic SE interface loads**
  - **spc101a -- separate explicit SPC's from AUTOSPC's**
  - **111_by_modea -- Sol 111 data recovery for indiv. Modes**

What is a Qualifier?

- Qualifiers are used in the Structured Solution Sequences (Sol 101-200) for each datablock
- Example: Qualifiers for KAA
  - KAA will exist on the database for every superelement
  - Nastran must keep track of “which” KAA is being used
  - This is done with qualifiers.

What is a Qualifier?

- **Example: find the datablocks and qualifiers on a MSC.Nastran database (.DBALL/.MASTER):**

```
acquire nddl
assign basedb='dbmaker.MASTER'
dbloc logi=basedb
dbdir
endjob
```

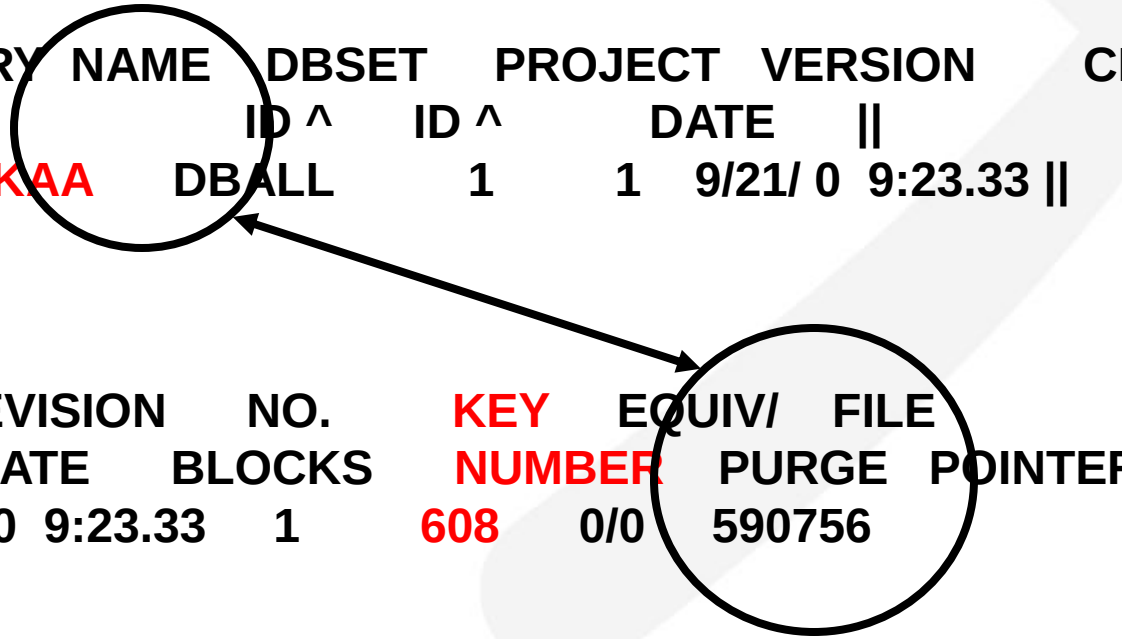
What is a Qualifier?

- **DBDIR listing:**

```
NDDL DATABLOCK LISTING

 ENTRY  NAME    DBSET    PROJECT   VERSION       CREATION    ||
                 ID ^     ID ^         DATE      ||
   68 KAA      DBALL        1         1    9/21/ 0  9:23.33 ||


    REVISION      NO.        KEY     EQUIV/    FILE
      DATE       BLOCKS     NUMBER    PURGE   POINTER
 9/21/ 0  9:23.33    1        608      0/0     590756
```



MSC SOFTWARE
SIMULATING REALITY

What is a Qualifier?

```
KEY NUMBER =   608        QUALIFIER NAME AND VALUE
                    SEID    =          0
                    PEID    =          0
                    MPC     =          0
                    SPC     =          0
                    DYRD    =          0
                    MTEMP   =          0
                    NLOOP   =         -1
                    DESITER =          0
                    PVALID  =          0
                    APRCH   =
                    HIGHQUAL=          0
                    K2GG    =
                    M2GG    =
                    AUXMID  =          0
                    DESINC  =          0
```

What is a Qualifier?

- **Why are qualifiers important**
  - **If DMAP'ing and superelements, then you will need to use the "right" KAA matrix.**
    - **For example, the SPC, MPC, SEID, etc. qualifiers must be correct, or Nastran won't find it, and issue a fatal message**
    - **DBVIEW can be used to get proper matrix**
- **What are some common uses?**
  - **"CSUPER" external superelements**
  - **If reading a file from an external source (i.e. OUTPUT4) and creating a database that will be used in another run, the Qualifiers must be correct.**

Summary

- Nastran Set Definitions and Structural Matrix Reductions
- Flow of Solution -- Common Subdmaps
- “Barebones” dmap solution
- Key Modules
- Where Do I Insert a DMAP (alter, malter)
- SSSALTERs
- What is a Qualifier?