



# A Tutorial on How to Use Implicit LS-DYNA®

Roger Grimes  
LSTC

12<sup>th</sup> International LS-DYNA Users Conference  
June 2012

# What is Different About Implicit?

- The Stiffness Matrix
  - Time is dominated by Numeric Factorization
  - Memory AND Disk Storage is dominated by storing the Factorization.





# Topics

- Memory Management
- I/O
- Debugging Tools
- Details are in the proceedings paper or on LSTC ftp site in `outgoing/grimes/implicit_papers/`



# Memory Management

- This talk focuses on using the Hybrid Parallel version of LS-DYNA.
- Most points can be used for SMP and non-hybrid MPP parallel versions.
- Memory Management is key for using Implicit for large models
  - Understand your computational environment
  - Understand how Implicit is using memory



# Computer Environment

- Distributed memory/massively parallel computing environments are made up of computational nodes.
  - No. of computational cores per node
  - Amount of RAM per node
  - Local or network I/O system for each node

# Why Hybrid?

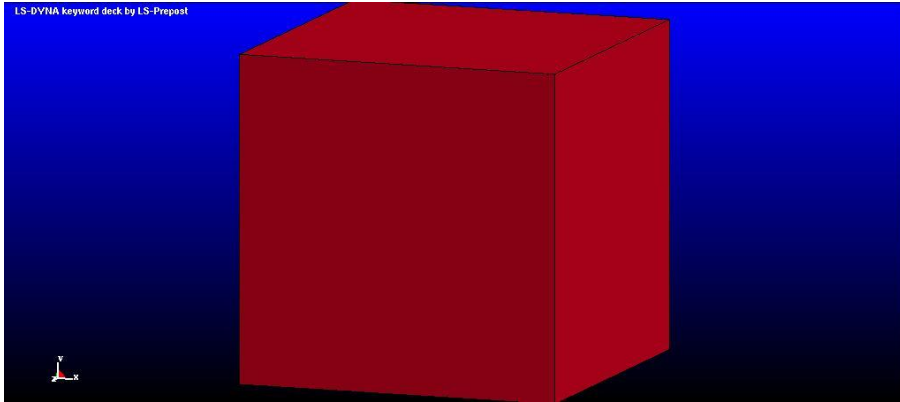
- Using MPI distributes processes across the computational cluster.
- We recommend using Hybrid version and a Round Robin assignment
  - to spread the processes around the cluster
  - keep the number of local processes to 1 or 2.
  - keeps the memory intact



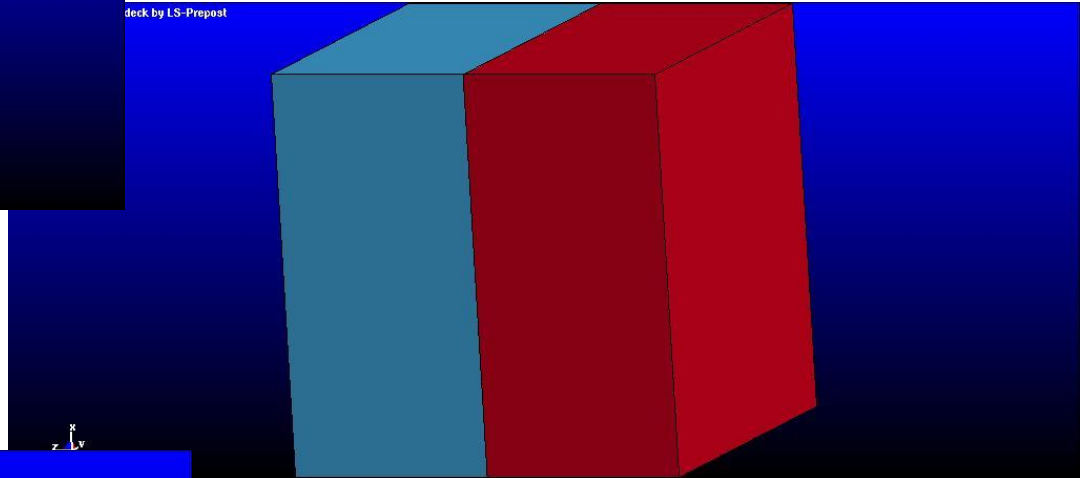
**LSTC**  
Livermore Software  
Technology Corp.

# Keeping Memory Intact

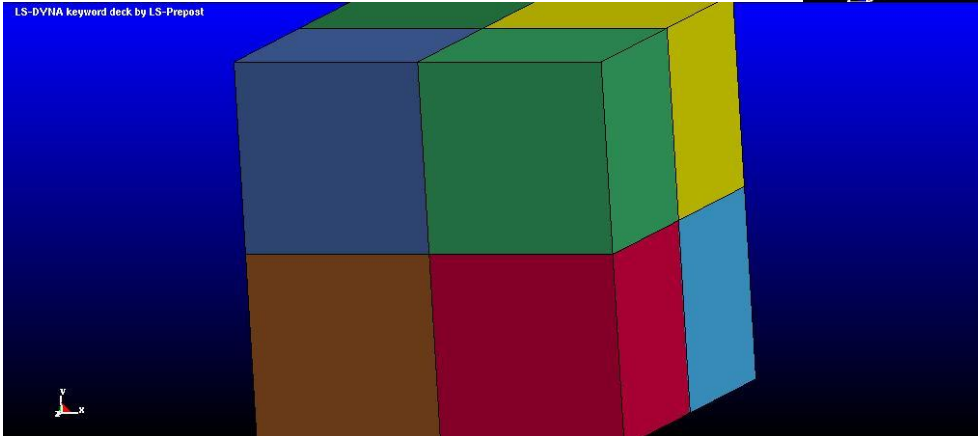
LS-DYNA keyword deck by LS-Prepost



deck by LS-Prepost



LS-DYNA keyword deck by LS-Prepost



# Memory Specification

- For Implicit you can use about 75% of RAM for the memory=xxxxM.

16 Gbytes		48 Gbytes	
$P_\ell$	Memory	$P_\ell$	Memory
1	1500M	1	4500M
2	800M	2	2200M
4	400M	4	1100M
8	200M	8	550M

- Do not use AUTOMEMORY or memory2= for Implicit.



# RAM per Node

- RAM required (estimated)
  - Let N be the global number of nodes in the model
  - Let P be the number of MPP processes being used
  - Past experience shows Implicit needs about  
 $(440/P + 75) * N$
- RAM per node is a limiting factor!!!!
  - Let P\_local be the number of MPP processes being used on a specific computational node
  - Amount of available memory is  
 $(0.75 * RAM / 8) / P\_local$

# Number of Processes

- Need  $P$  and  $P_{\text{local}}$  such that

$$(440/P + 75) * N < (0.75 * \text{MEM} / 8) / P_{\text{local}}$$

- Easiest to start with  $P_{\text{local}} = 1$ 
  - Keeps memory intact
- Then choose  $P$  appropriately



# Where Hybrid Comes in

- Why did I just pay for all of those cores?
  - You want me to use just 1 MPI process per compute node!
  - What about all of those cores?
- Use  $N_{CPU} = \#$  of cores per compute node
  - Uses SMP parallelism on the compute node to recover some if not all of the available parallelism.



**LSTC**  
Livermore Software  
Technology Corp.

# Understanding Implicit use of Memory

- Use LPRINT = 2
- Look for

Start of implicit storage allocation – locend =

End of implicit storage allocation – locend =

expanding memory to xxxx implicit matrix storage

storage currently in use

expanding memory to xxxx linear eqn. solver

# I/O

- Implicit uses disk to store data to reduce requirements for RAM.
- This I/O is all scratch and can be intensive.
- Use local (to the computational node) I/O system. p=pfiler should contain  

```
dir { local /local/directory rmlocal transfer_files transfer_scr }
```
- Do not use network mounted I/O system for these files.

# Model Debugging

- LPRINT = 3
  - Extra debug checking
  - Separable components check indicates small parts not connected to main model
- Eigenvalue computations
  - Finds rigid body modes that causes globally singularities in the stiffness matrix
  - Look for ‘No. of modes to left’

# Model Debugging #2

- If you have a large number of rigid body modes look for
  - Elements not connected to the model.
    - The separable component messages should point these out.
  - Look for shell elements coating solid element parts.
    - These elements probably have zero rotational stiffness because the elements are very very thin.
    - Such elements must be membrane elements.

# D3ITER

- Nonlinear solution for implicit can output each iteration search direction to a d3plot like database
  - d3iter
  - D3itctl is 6<sup>th</sup> field on 2<sup>nd</sup> line of  
\*CONTROL\_IMPLICIT\_SOLUTION





# Implicit Loves Constraints

- Implicit loves constraint equations.
- Penalty based contact can cause problems
- Switch all use of

\*CONTACT\_TIED\_XXXXXX\_OFFSET

or

\*CONTACT\_TIED\_XXXXXX\_BEAM\_OFFSET

to

\*CONTACT\_TIED\_XXXXXX\_CONSTRAINED\_OFFSET



# Details in the Paper

- See the paper in the proceedings for details
- Both paper and presentation will be on LSTC ftp site in `outgoing/grimes/implicit_papers/`
  
- Thanks for listening
  
- Questions?