WHY PARALLEL GRID GENERATION ?

- Very Large Problems (Interactive)
- Transient Problems with Moving Bodies/ Boundaries
  - Internal Combustion Engines
  - Turbines
  - Store Separation
  - Shock/Structure Interaction
  - Penetration
  - Multifluid Simulations

$\Rightarrow$

Grid Regeneration May Take as Much CPU-Time as Field Solver

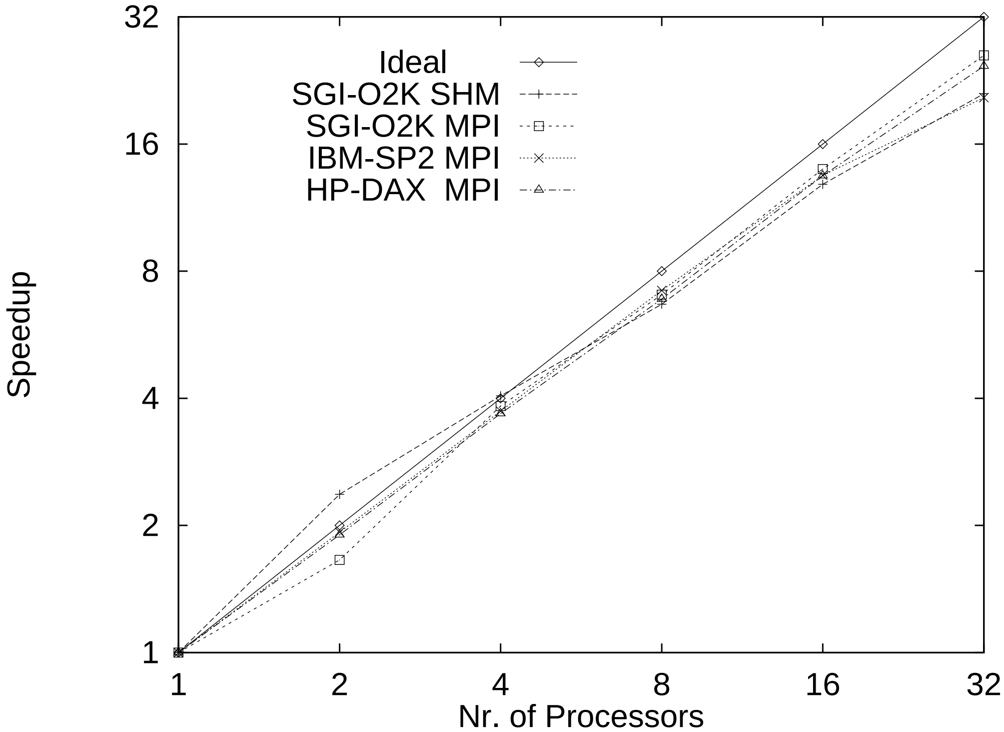
PROBLEM SIZE (1)

Size	Dim	Code	Year	Problem	Machine
$> 10^2$	2-D	FEFLO20	1983	Airfoil	ICL
$> 10^3$	3-D	FEFLO30	1985	Forebody	CYBER-205
$> 10^4$	2-D	FEFLO27	1986	Train	CRAY-XMP
$> 10^5$	3-D	FEFLO72	1989	Train	CRAY-2
$> 10^6$	3-D	FEFLO74	1991	T-62	CRAY-2
$> 10^7$	3-D	FEFLO96	1994	WTC	CRAY-M90
$> 10^8$	3-D	FEFLO98	1998	Village	Origin 2000

PROBLEM SIZE (2)

Case	Year	Mtets	Machine	Proc
WTC	1993	20.0	M90	1
Dharan	1996	6.0	T90	4
Village	1997	60.0	M90	4
LTS2	1998	10.0	O2K	8
Tysons	1998	16.0	O2K	16
Office	1998	52.0	O2K	24
City	1998	110.0	O2K	32
Office	1999	180.0	O2K	32

PARALLEL FLOW SOLVER PERFORMANCE



Euler, 500Ktet, RK3, Roe+MUSCL

ALGORITHMS TO GENERATE UNSTRUCTURED GRIDS

Approaches:

- Fill Empty Space: **Advancing Front**
- Modify Grid: **Delaunay**

Point Distribution in Space:

- From Measure of Grid Quality
- From Background Grid
- From Background Sources

Complex Geometries/Physics:

- Sources/B.G. **Dominate CPU**

UNSTRUCTURED GRID GENERATORS

- Scalar
- Large Variation in OPS
- Parallel by ‘Distance’

$\Rightarrow$ Porting to Parallel Machine:

P.1 Break Up Problem Into Pieces

P.2 Hand Each Processor a Piece

P.3 Provide Interprocessor Transfer of Info

P.4 Assemble

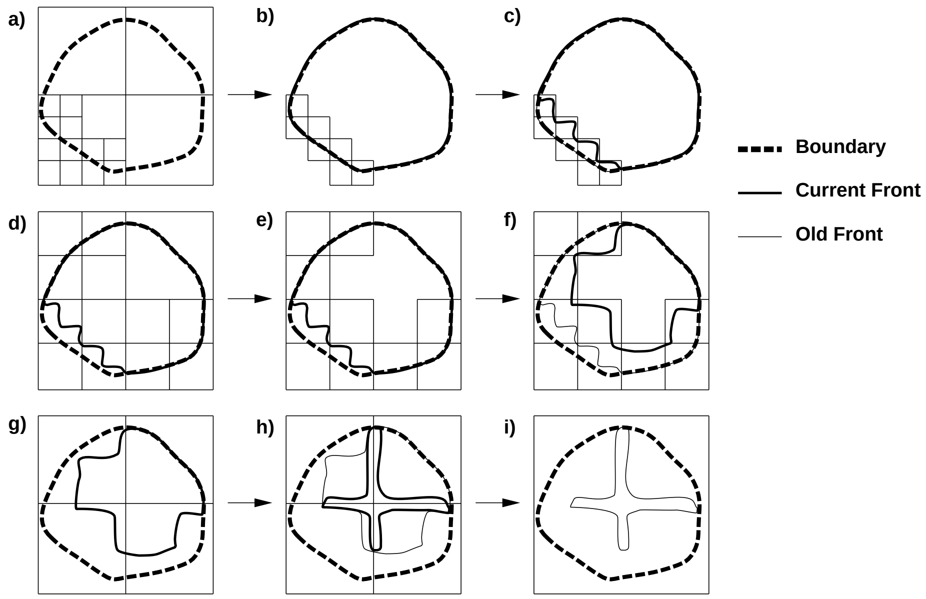
DESIGN GOALS

- Minimal Change of Scalar Advancing Front Technique

- Minimal Overhead for Mono-Processor

- Scalability on SGI O2000

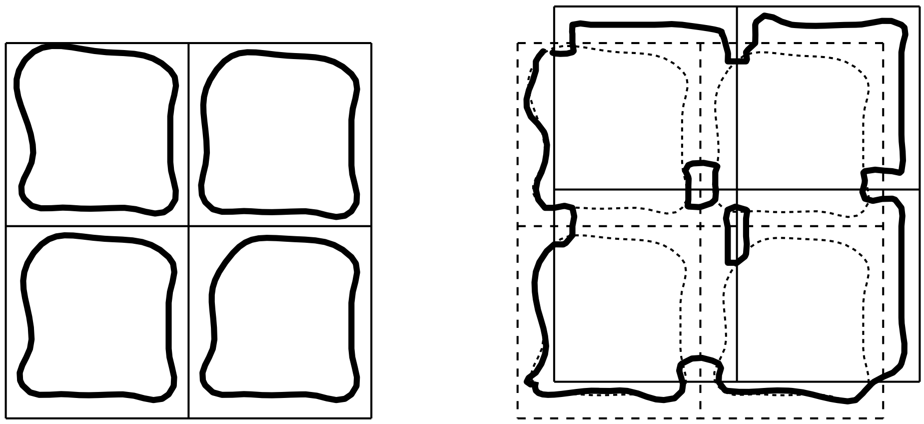
PARALLEL ADVANCING FRONT (1)



PARALLEL ADVANCING FRONT (2)

- WHILE: Active Faces Left:
  - Build Octree of Active Faces
  - Retain Octants With Faces Generating Small Elements
  - (Shift) + Mesh Octants in Parallel
  - Reassemble Remaining Faces
- END WHILE

PARALLEL ADVANCING FRONT (3)

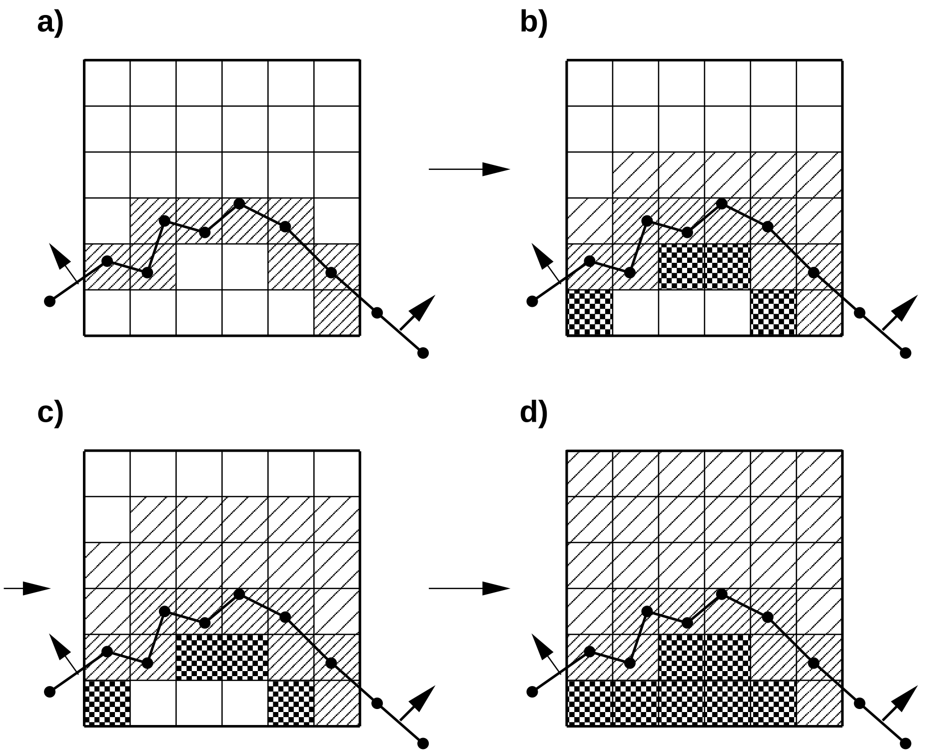


Shift and Regrid Technique

AREAS OF WORK

- Large Variation of Element Size
- Estimation of Work
- Balance of Work
- Reduction of Inter-Box Faces

ESTIMATION OF WORK (1)



ESTIMATION OF WORK (2)

Given:

- Box to be Gridded
- Active Faces

Then: Use Bin-Estimation of Volume

- Subdivide Box Into Voxels
- Obtain Voxels Cut
- Obtain In/Out Neighbour Voxels
- Sweep to Mark In/Out
- Count Voxels In

Remarks:

- Done in Parallel

BALANCING OF WORK

Given:

- Work to be Done In Each Box
- Nr. of Processors

Then: Use Greedy Algorithm

- Obtain Average Work
- Assemble Until Average Exceeded

AGGLOMERATION OF BOXES

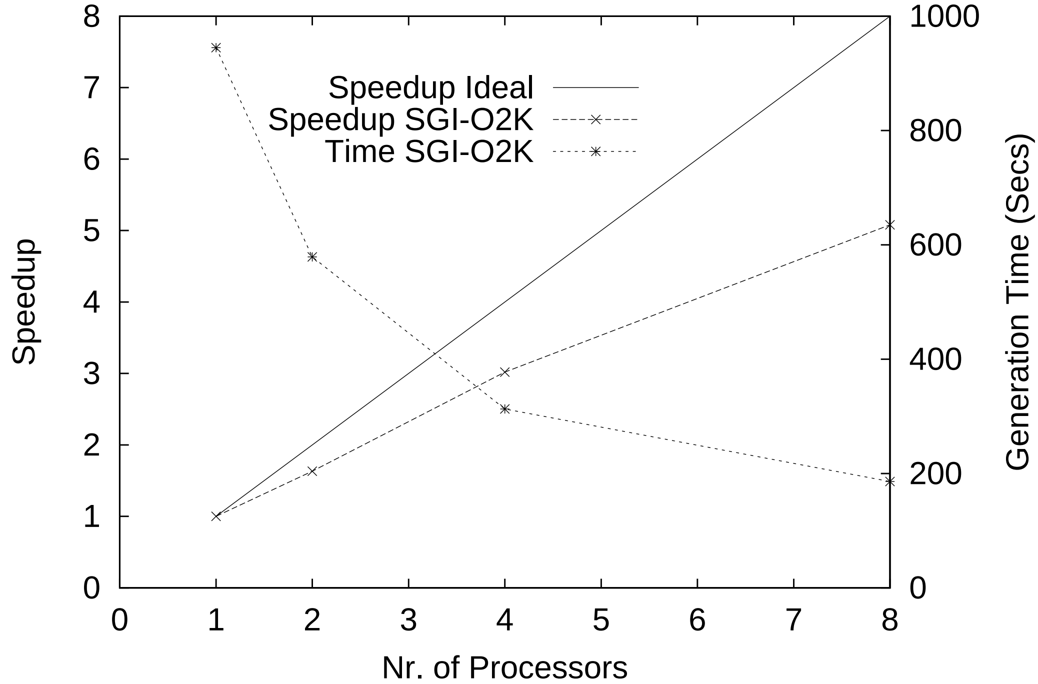
Aim:

- Many Boxes per Processor
- Avoid Excessive Buildup of Faces
- Minimize Interpolation of GGEN-Info

How: Recursive Agglomeration

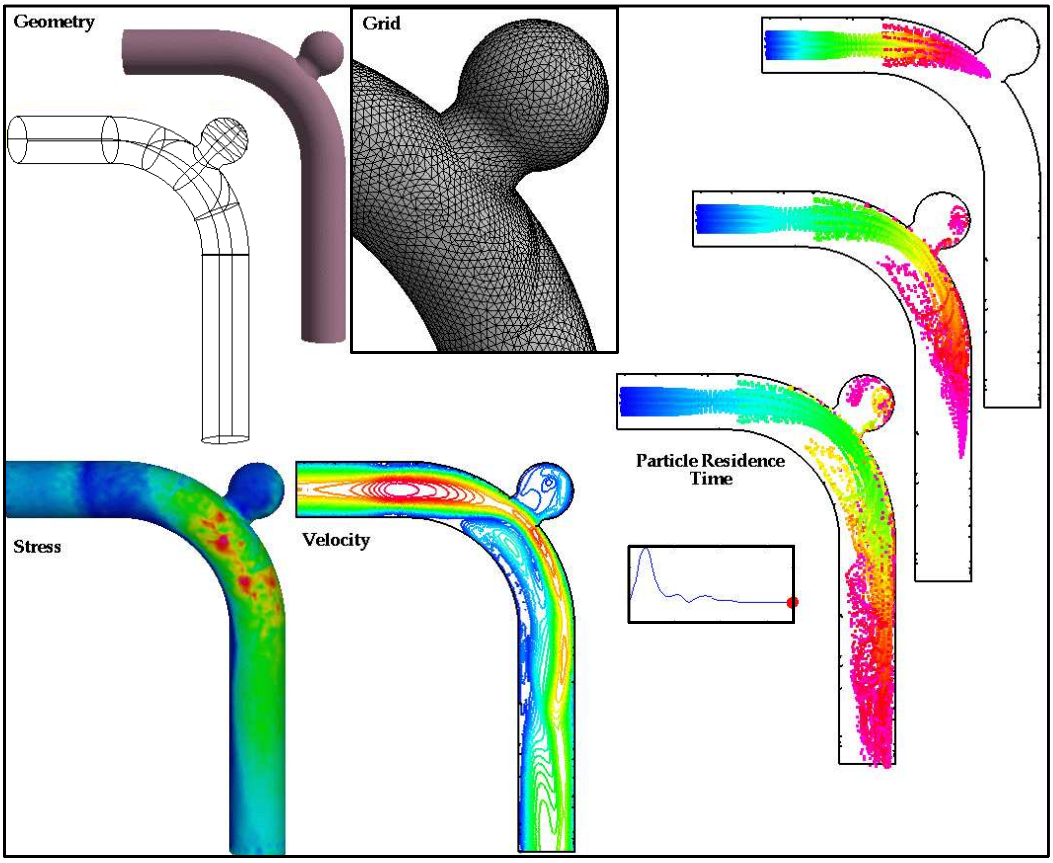
- IF: 2 Dimensions of Box Same
    - IF: 3rd Dimension Coincident
      $\Rightarrow$ Boxes Can be Merged
    - ENDIF
- ENDIF

UNIT CUBE



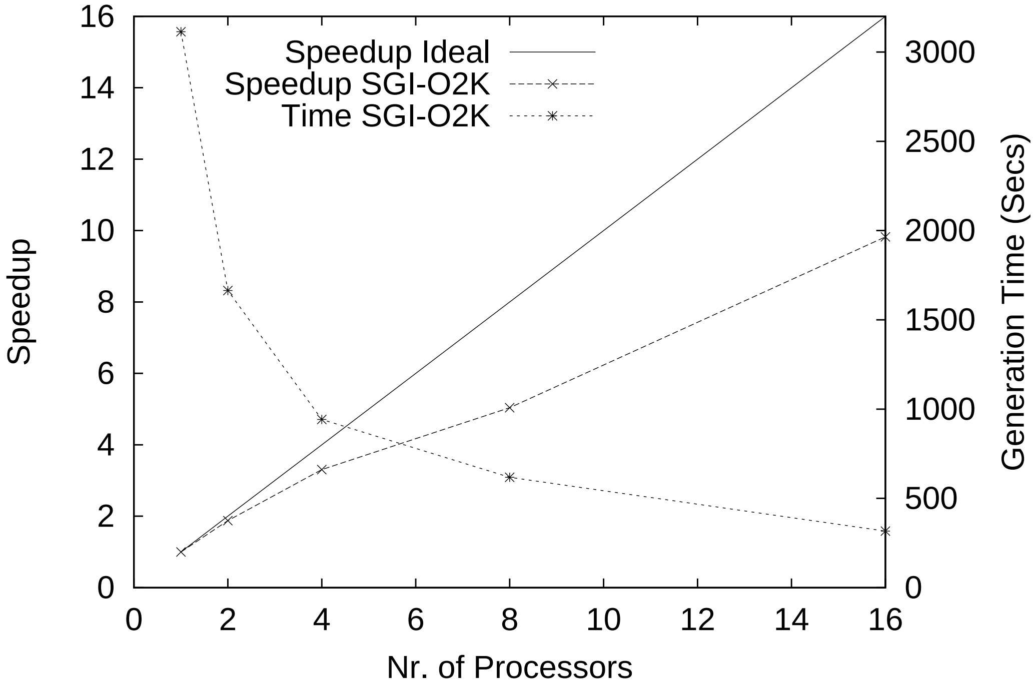
Cube: Speedups Obtained (1 Mtet)

ANEURISM (1)



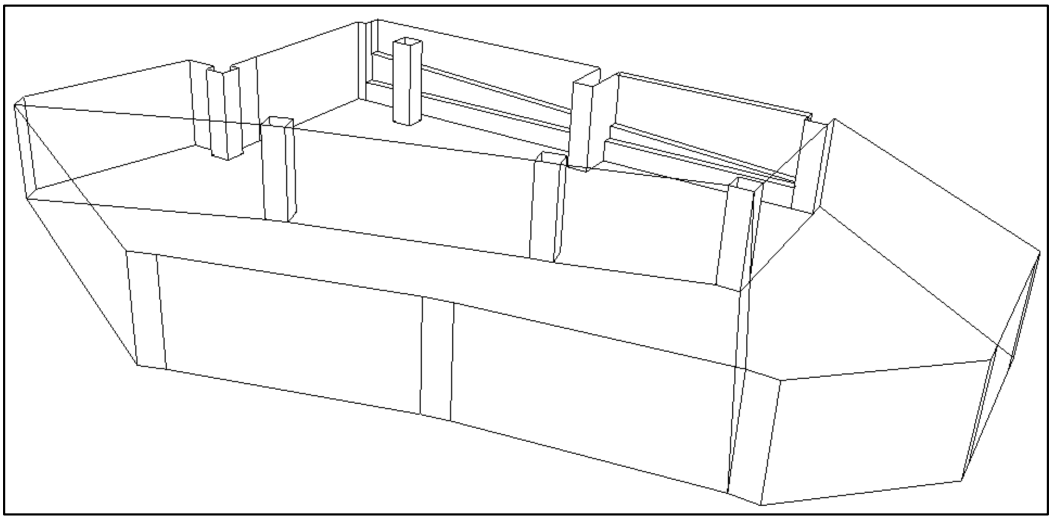
Aneurism: Grid and Solution

ANEURISM (2)



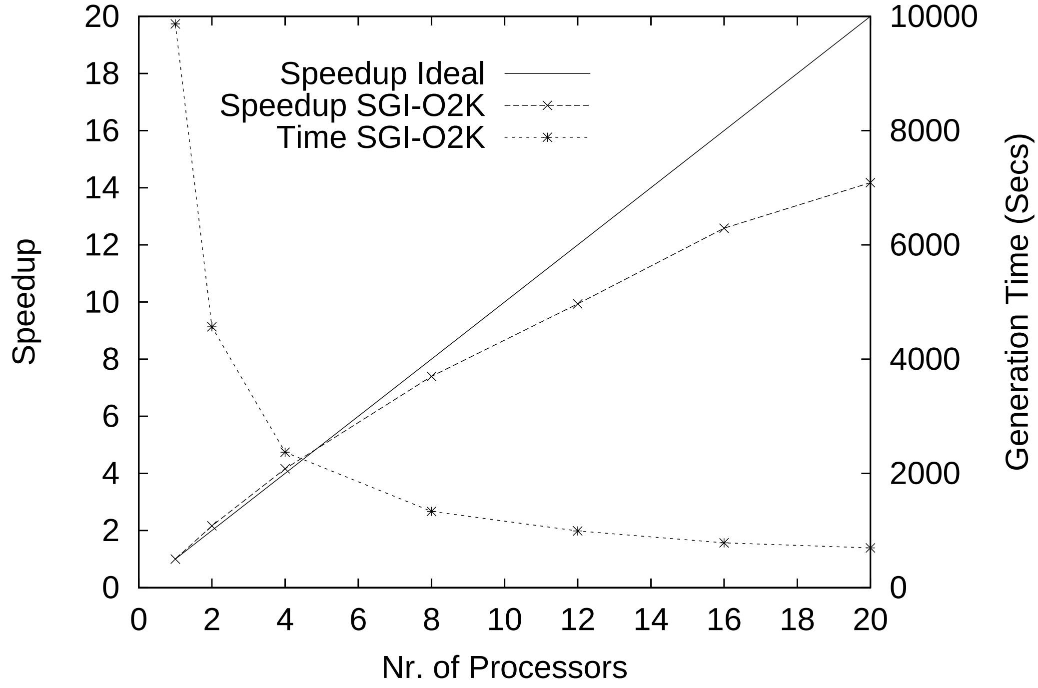
Aneurism: Speedups Obtained (2.7 Mtet)

GARAGE (1)



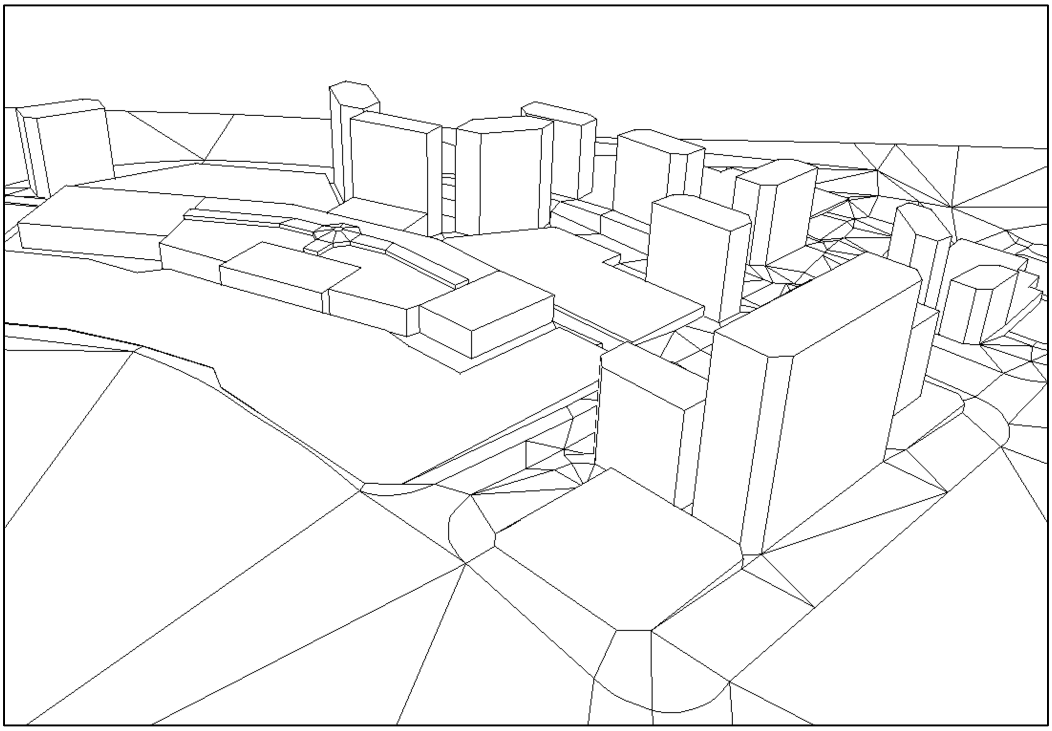
Garage: Wireframe

Garage (2)



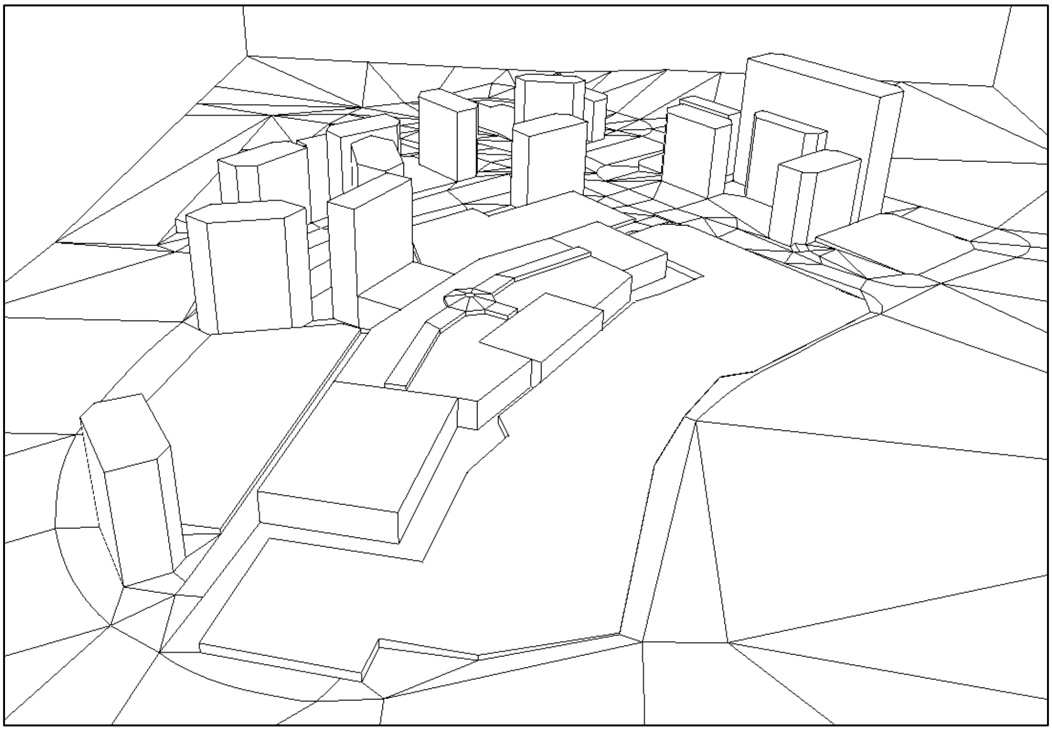
Garage: Speedups Obtained (9.2 Mtet)

TYSONS CORNER (1)



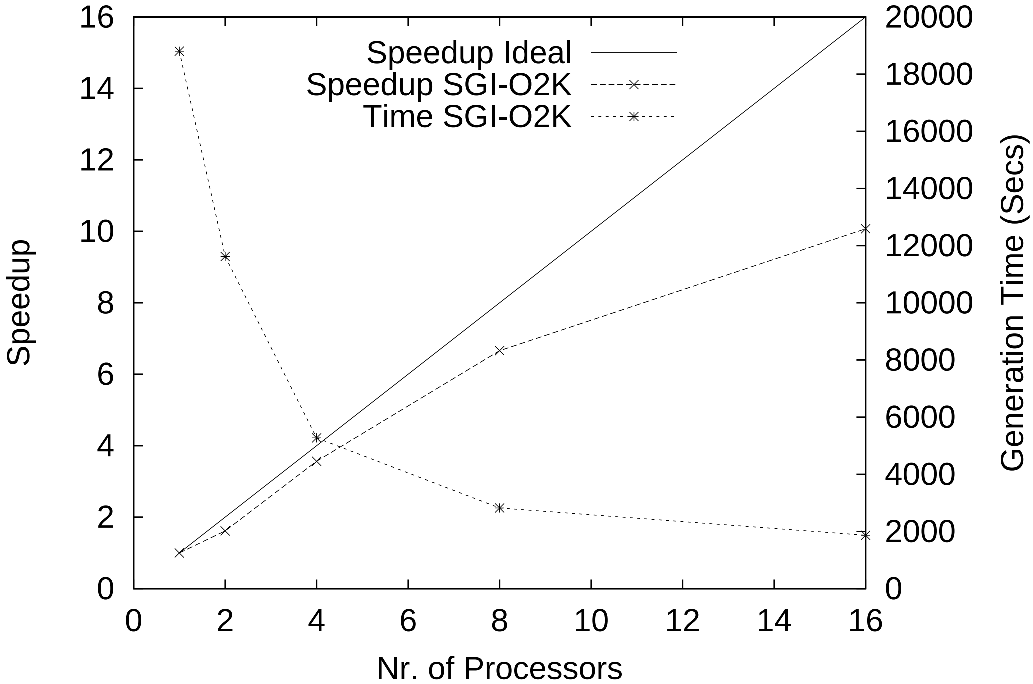
Tysons Corner: Wireframe

TYSONS CORNER (2)



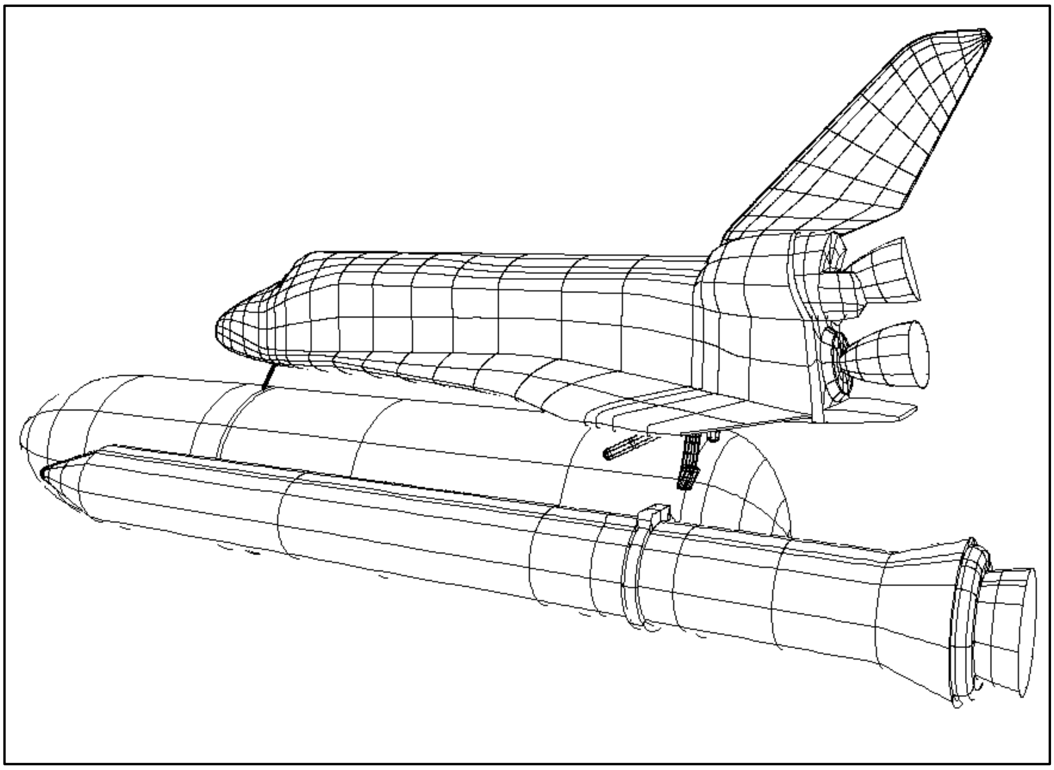
Tysons Corner: Wireframe

TYSONS CORNER (3)



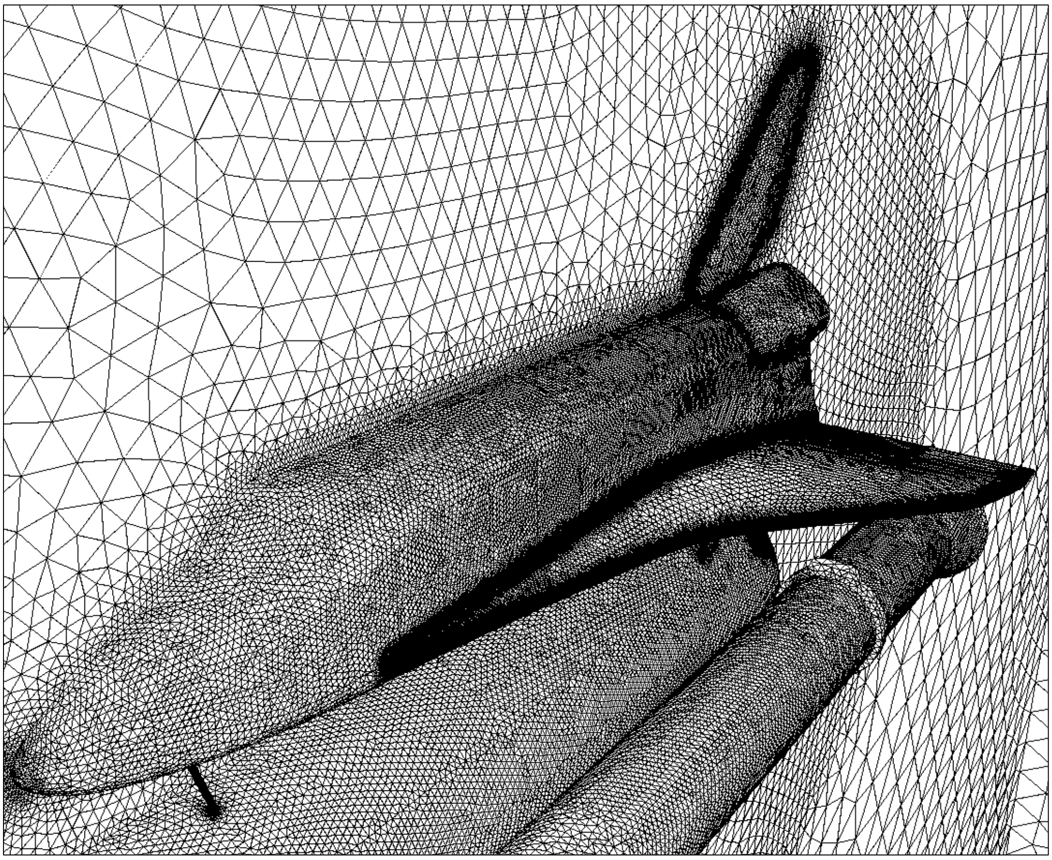
Tysons Corner: Speedups Obtained (16 Mtet)

SPACE SHUTTLE (1)



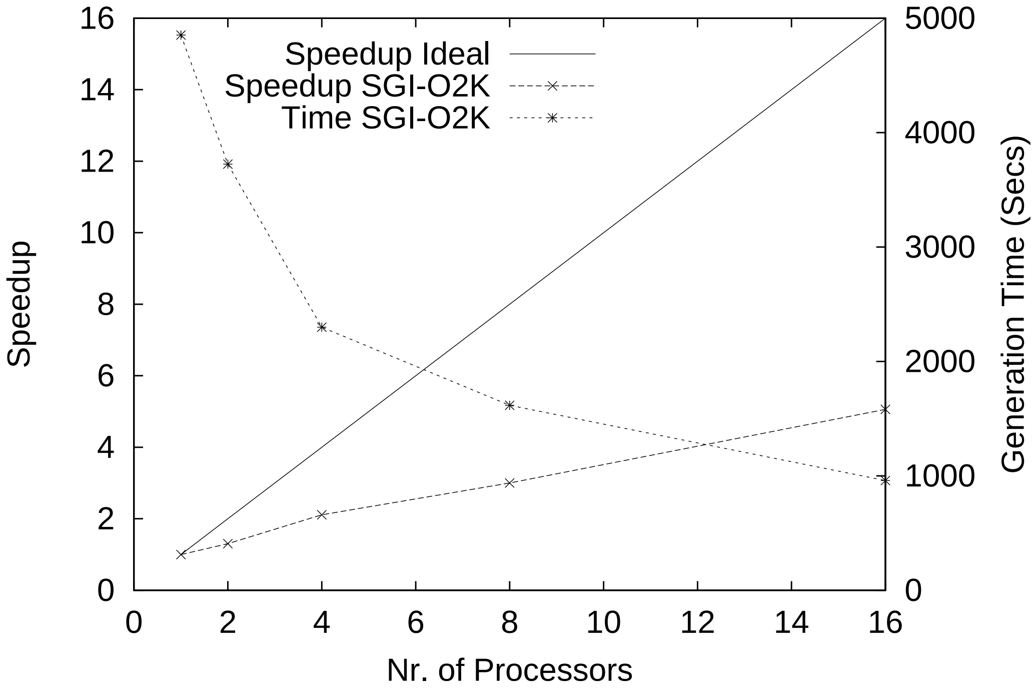
Space Shuttle: Outline of Domain

SPACE SHUTTLE (2)



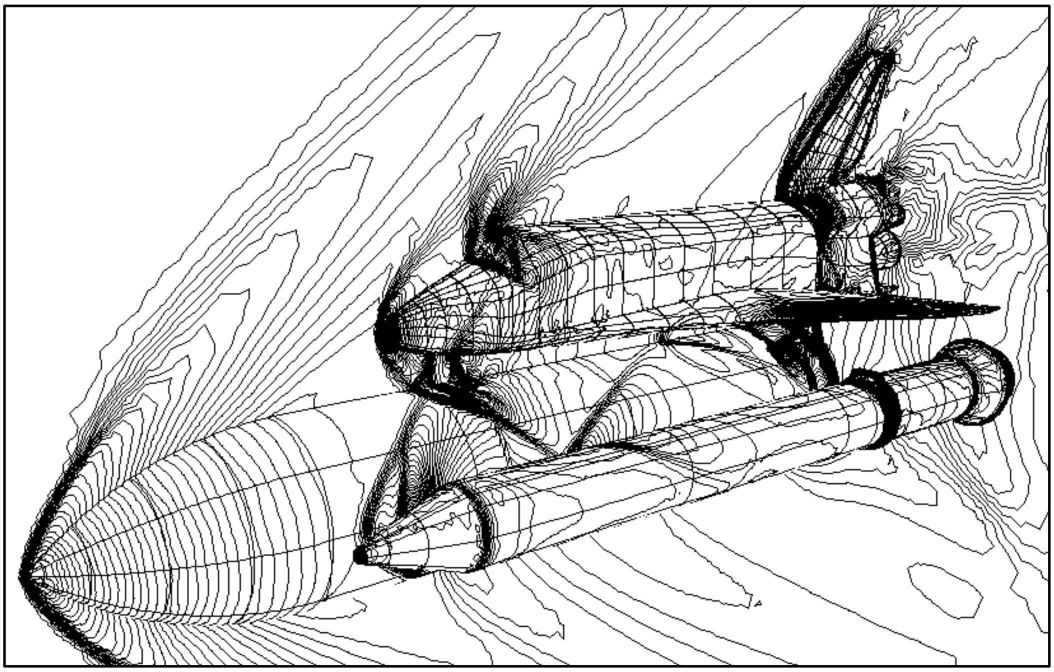
Space Shuttle: Surface Mesh

SPACE SHUTTLE (3)



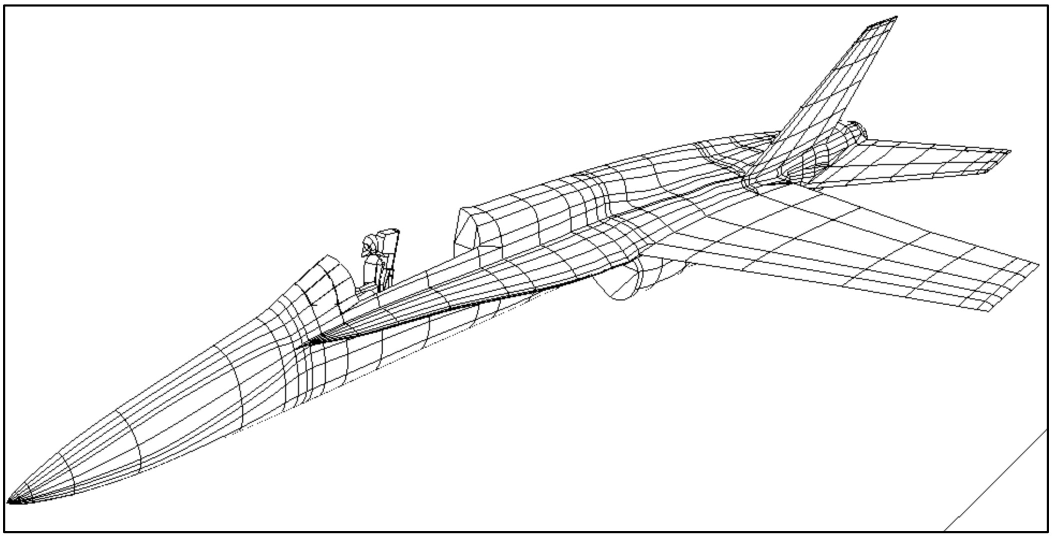
Space Shuttle: Speedups Obtained (4Mtet)

SPACE SHUTTLE (4)



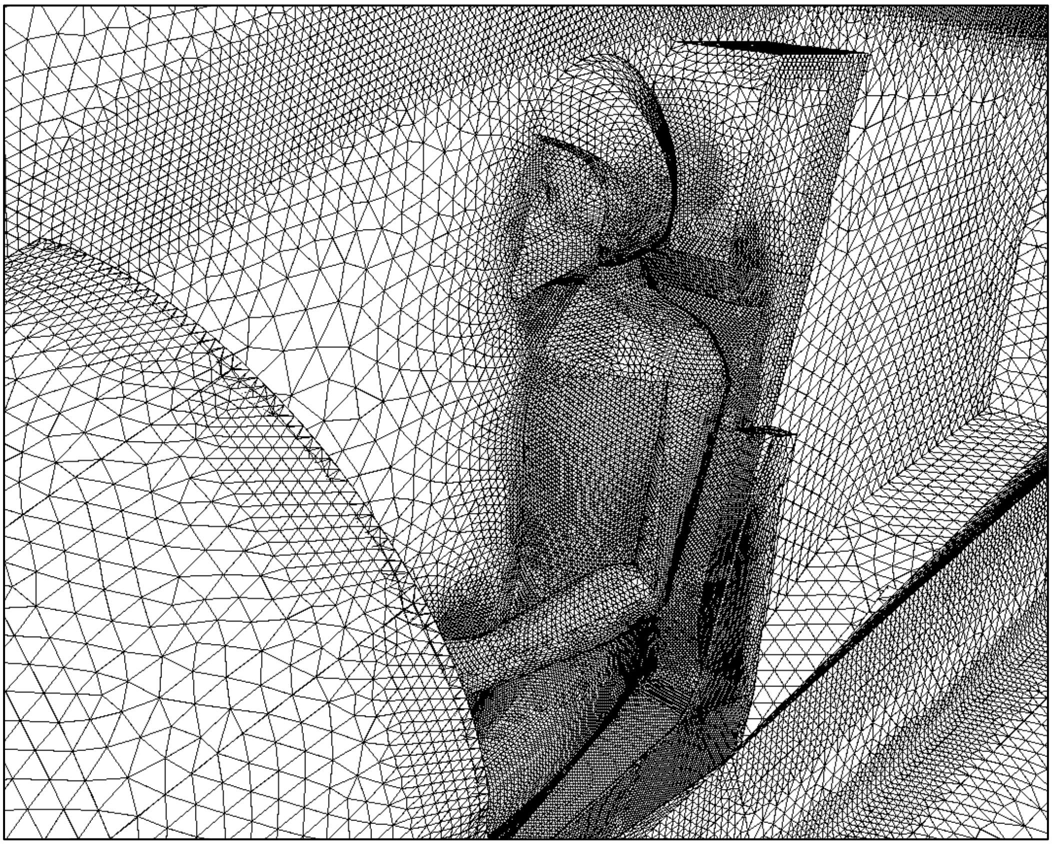
Space Shuttle: Surface Pressures

F18 PILOT EJECTION (1)



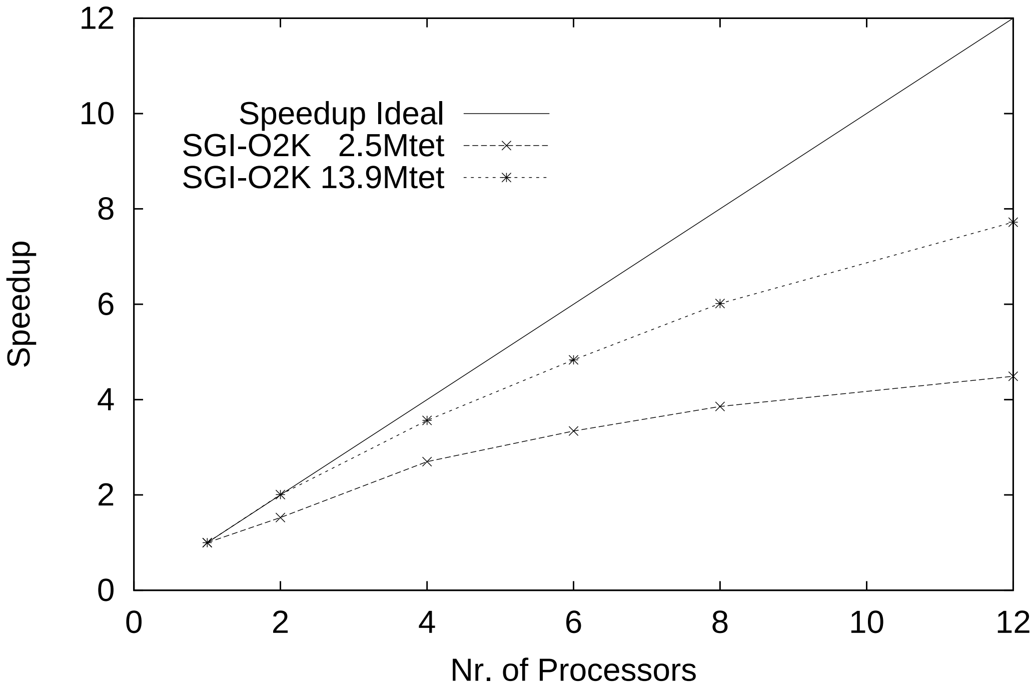
F18 Pilot Ejection: Outline of Domain

F18 PILOT EJECTION (2)



F18 Pilot Ejection: Surface Mesh (Closeup)

F18 PILOT EJECTION (3)



F18 Pilot Ejection: Speedups Obtained

ONGOING DEVELOPMENTS

- Better Estimation of Work
- Load Balancing Towards End
- Large Nr. of Processors