LA-UR- 10-003329

Approved for public release;
distribution is unlimited.

Title:	The LANL DSD Application: Overview, Implementation, Status
Author(s):	Toru Aida Tariq Aslam John Bdzil Mark Short John Walter
Intended for:	DSD Implementation Workshop Dept. of MeshSE Univ. of Illinois Urbana, Illinois 27 Jan 2010

Form 836 (7/06)

Abstract

The Detonation Shock Dynamics model is discussed in the context of it's implementation in the LANL DSD Application code. Some results related to DSD's use with the XCP-4 research code, LANL's Pagosa code and LLNL's Ale3d code are presented. On-going work to interface LANL DSD to ALe3d is also dicussed.

The LANL DSD Application Overview, Implementation, Status

Toru Aida, X-1-SMMP

Tariq Aslam, DE-9

John Bdzil, DE-9

Mark Short, DE-9

John W. Walter, X-3-MA

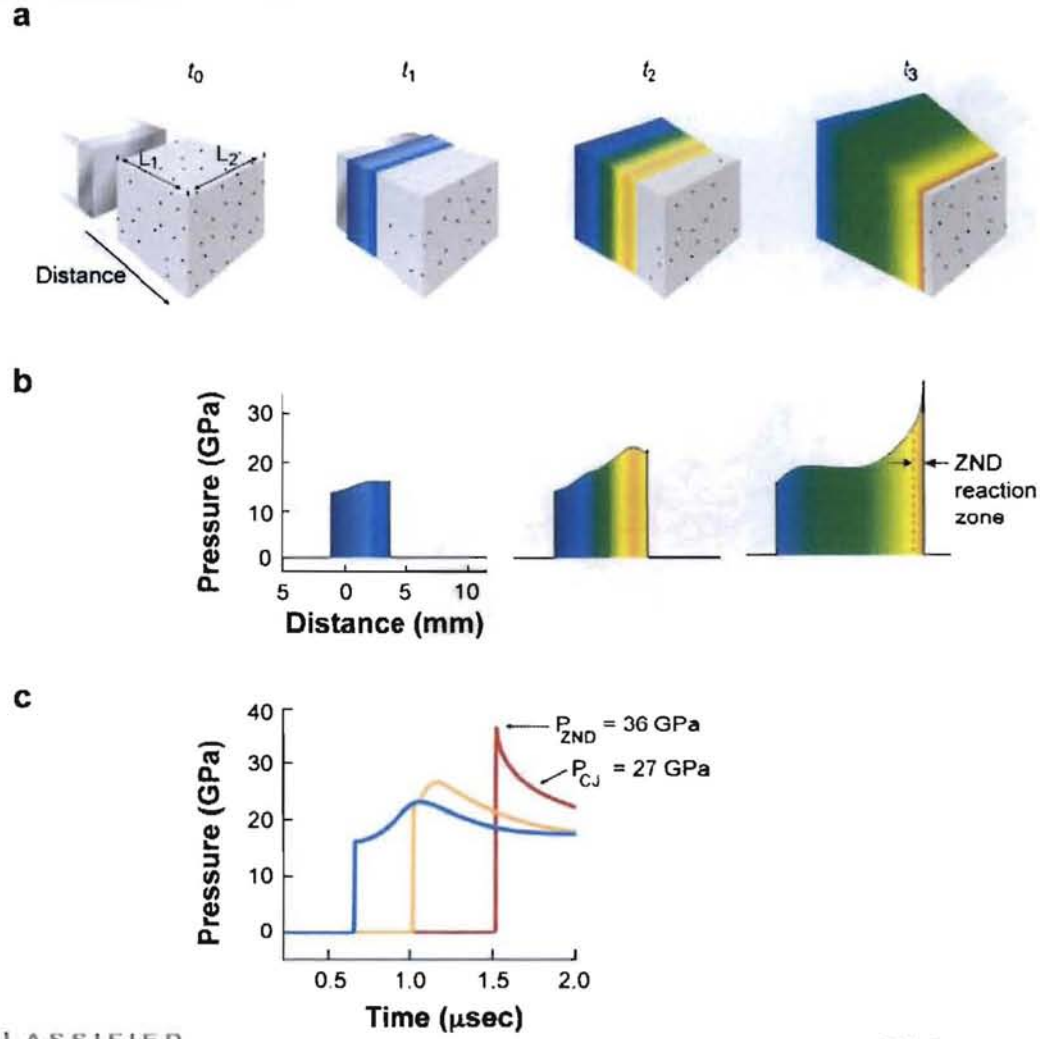
Jan 27, 2010

Outline

- Motivation
- The Detonation Shock Dynamics (DSD) model
- DSD algorithms and code
- Interface to Ale3d
- Parallelization

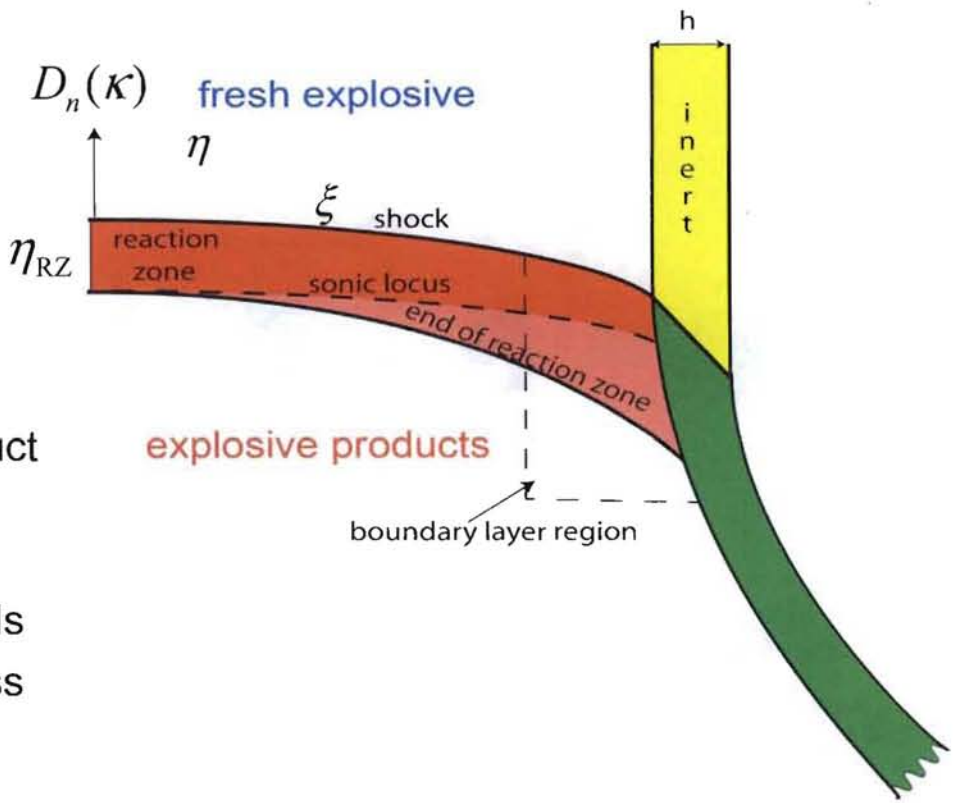
Explosive performance is central to device performance

- **Explosive power delivery depends on:**
  - Speed of detonation propagation
  - Push provided by the pressurized explosive reaction products
- **Propagation speed is set by the balance between energy release and flow in the reaction zone**
- **Push is most strongly a function of the EOS response of the explosive products**



Structure of reaction zone and modeling requirements

- **Reaction zone structure**
  - Thin: $\eta_{RZ} \sim 0.3$mm to 2mm
  - Weakly diverging: $\eta_{RZ}\kappa = O(\varepsilon) \ll 1$
  - Slowly varying in time
  - Sweeping along adjacent inert material
- **Reactive flow modeling**
  - Euler eqs. (gas dynamics) + reactant/product EOS + rate law for reaction progress, λ
  - Exptl. input needed for EOS and rate law
  - DNS using suitable hydrodynamics methods
  - Resolution: need at least 40–50 cells across reaction zone to get D_n to 1% accuracy
  - In 3D for DOE applications, have 10^{15} cell-cycles *just in reaction zone.*

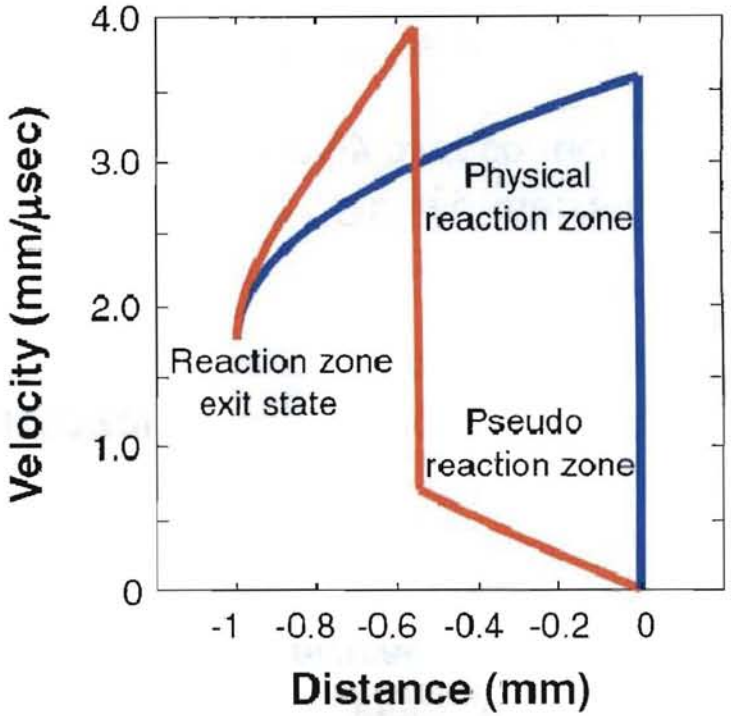


Two main approaches for modeling detonation propagation: reactive flow and programmed burn

- **"Reactive burn" modeling – DNS of the reactive Euler equations**
  - Use an EOS for both unburned explosive and detonation products.
  - Develop a global heat-release rate law for the reaction progress variable, λ.
    - Accurate detonation propagation requires highly-resolved calculations of the relatively narrow detonation reaction zone.
    - The numerical resolution requirements are beyond current capabilities for some 2D and all 3D engineering geometries.
- **The LANL DSD programmed burn approach involves three components**
  - Replace direct numerical simulation (DNS) of reaction zone with a shock curvature-dependent detonation front propagation law: $D_n(\kappa)$.
    - Preferable to constant-velocity programmed burn for IHE.
  - **Requires** only an EOS for the products — but reactant/product EOS is **preferred**.
  - Pseudo reaction zone (PRZ) model to release the explosive energy over correct length / time scale.
    - PRZ approximates react. zone exit state that would obtain from resolved DNS.
    - Requires low numerical resolution compared to DNS.
  - Computationally very efficient – suitable for 3D modeling of explosives.

DSD – hydrodynamics coupling: Pseudo-Reaction Zone

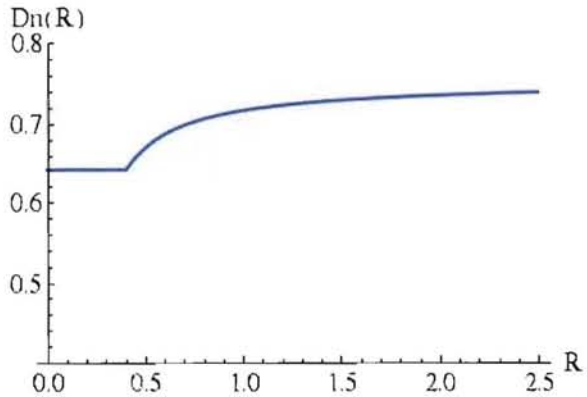
- **DSD is a programmed burn scheme**
  - Hydro-mesh burn front arrival times are computed in pre-processor mode.
  - Complete energy release as burn front sweeps a computational cell is often not correct.
- **PRZ models the real reaction zone exit state**
  - DSD burn times + PRZ reaction rate calibrated to those burn times → accurate exit state.
- **Dey and Shaw One-step Rate for PBX9502**
  - Rate: $\dot{\lambda} = (1-\lambda)/\tau$
  - Burn fraction: $\lambda(t,t_b) = H(t-t_b)(1-e^{-(t-t_b)/\tau})$
- **Wescott Rate for ANFO**
  - Rate: $\dot{\lambda} = k(D_n / D_{CJ})^N \sqrt{1-\lambda}\, H(t-t_b)$



Comparison of DNS and DSD + constant rate PRZ

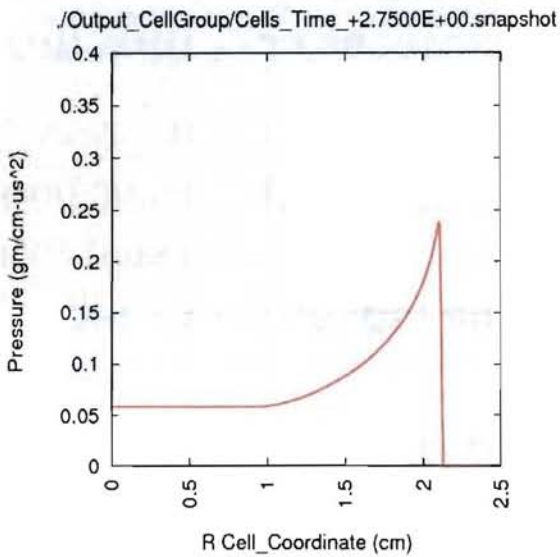
Importance of Pseudo-Reaction Zone for IHE Reactant/Product EOS – PRZ – DSD model for PBX9502

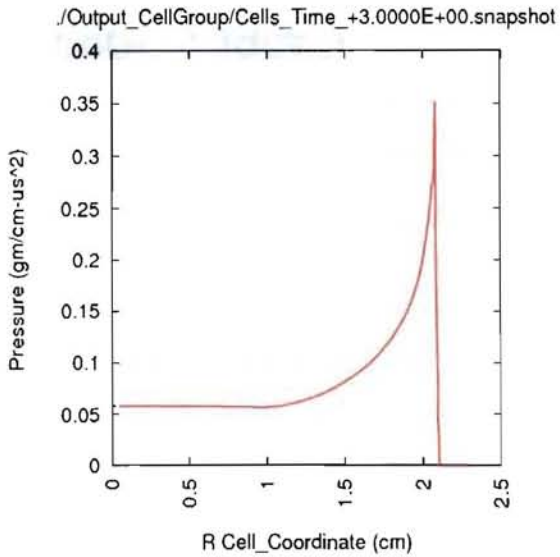
- **Model of Dey and Shaw, LANL rpt. LA-UR-05-7511, 2005.**
  - Linear Us-Up Reactant, Sesame Product, EOS (Shaw).
  - One-step PRZ model, τ = 30 us.
  - Non-linear DSD model.
    Dcj = 0.770 cm/us, kmax = 5.0 cm^{-1}.
    9502 reacts completely in 0.1 – 0.2 us
    or 0.13 – 0.26 cm for steady planar detonation.



- **Consider three cases, all with same EOS.**
  1) Huygens (constant Dn) burn times, standard prog burn.
  2) Non-linear Dn(kappa) burn times, standard prog burn.
  3) Non-linear Dn(kappa) burn times, one-step PRZ burn.
- **All run with X-3 research code, CWNN (conserv. Lagrangian hydro).**
  - 1D spherical mesh, 250 cells, dR = 0.01 cm.

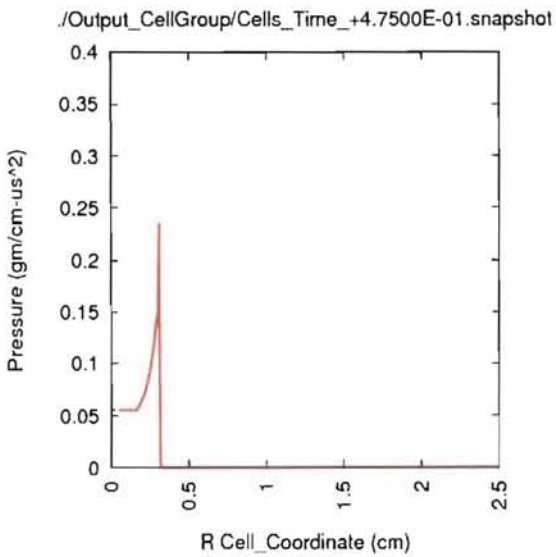
Compare case 1) and case 3)

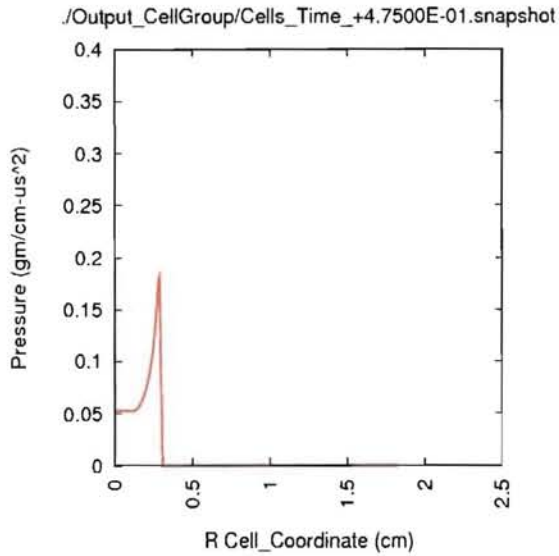




- Both run stably but in case 1) the (programmed) burn front runs ahead of the shock wave that should be coincident with it.
- This causes loss of the von Neumann spike and un-physically lowers the CJ pressure.
- Of course, the arrival times are also wrong: left at 2.75 us, right at 3.0 us.

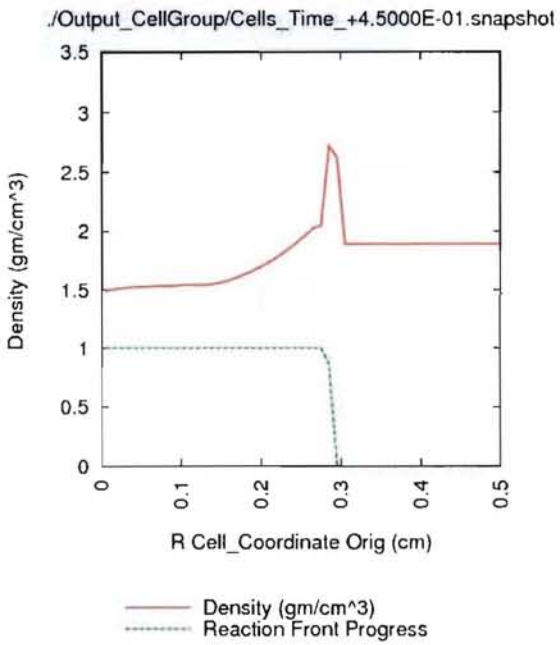
Use DSD to get the correct burn arrival times (case 2). But now simple programmed burn crashes!

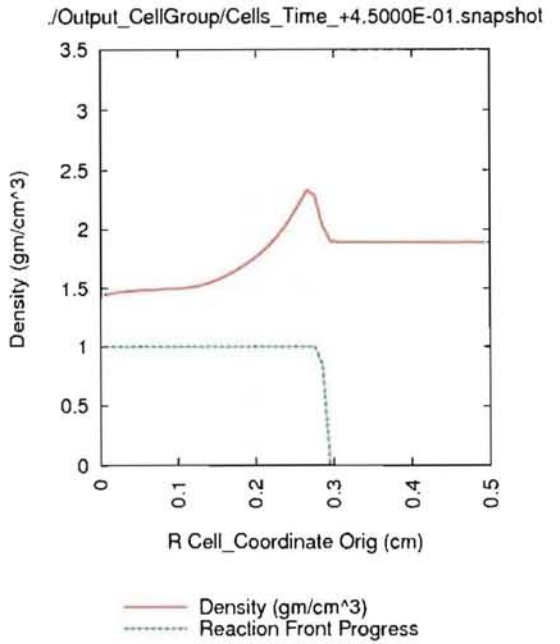




- In case 2 (left) a large von Neumann spike has developed even though the burn is just starting. There is no spike yet in case 3. This suggests the nature of the problem.

Why does case 2 get in trouble but case 3 doesn't? Compare at earlier time (case 2 dies before 0.5 us).

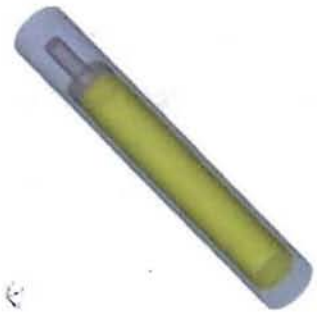


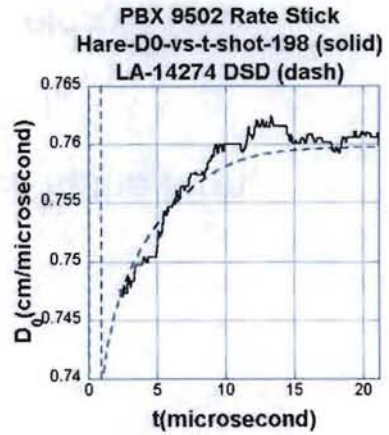


- In case 3 (right) there is no density increase before arrival of the burn front. The PRZ model is releasing the HE energy in a manner consistent with the DSD burn times and the EOS.
- In case 2 the energy is being released too quickly so the lead shock is pushing ahead of the burn front. This pre-compresses the HE and causes the burn to become unstable.

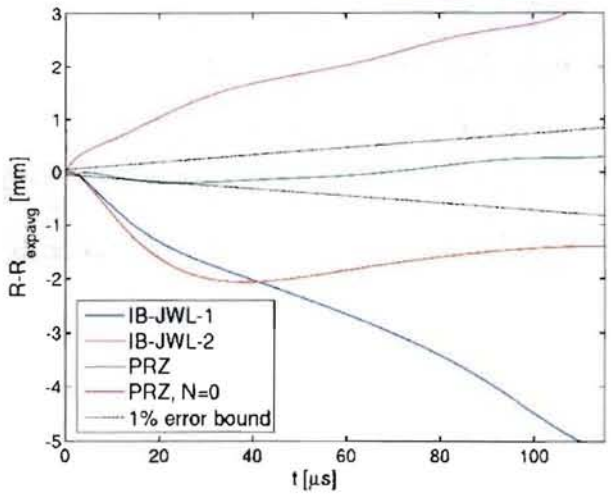
Examples of DSD modeling (PBX 9502): Bdzil, Wescott

- Combining a calibrated DSD $D_n(\kappa)$ propagation law with boundary-condition information, detonation propagation in PBX 9502 can be calculated.
- The variation of detonation speed due to boundary effects, divergence and convergence effects, multiple explosives are accounted for.
- When a pseudo reaction-zone model doses energy to the flow at a rate consistent with $D_n(\kappa)$ (and a consistent products EOS is available), the explosive work is calculated accurately.





Modeling work in ANFO cylinder test: difference (modeling – experiment)



DSD theory

- **A theoretical & computational model for detonation shock propagation**
  - Derived from asymptotic analysis of reactive Euler eqs. in shock coordinates
  - Quasi-steady, multi-dim. perturbation of steady, 1D, ZND reaction zone.
  - Captures the most important effect: dependence of D_n on curvature and thickness
  - In typical geometries: $\eta_{RZ}\kappa = O(\varepsilon) \ll 1$ and $d\,D_n / dt = O(\varepsilon^2)$
  - This yields a standard set of DSD scalings.
  - With $Y = (\rho, u_\eta, p)^T$ the solution is expanded as:

$$Y = Y^{(0)} + \varepsilon Y^{(1)} + \varepsilon^2 Y^{(2)} + \cdots, \quad u_\xi = \varepsilon^{3/2} u_\xi^{(3/2)}$$

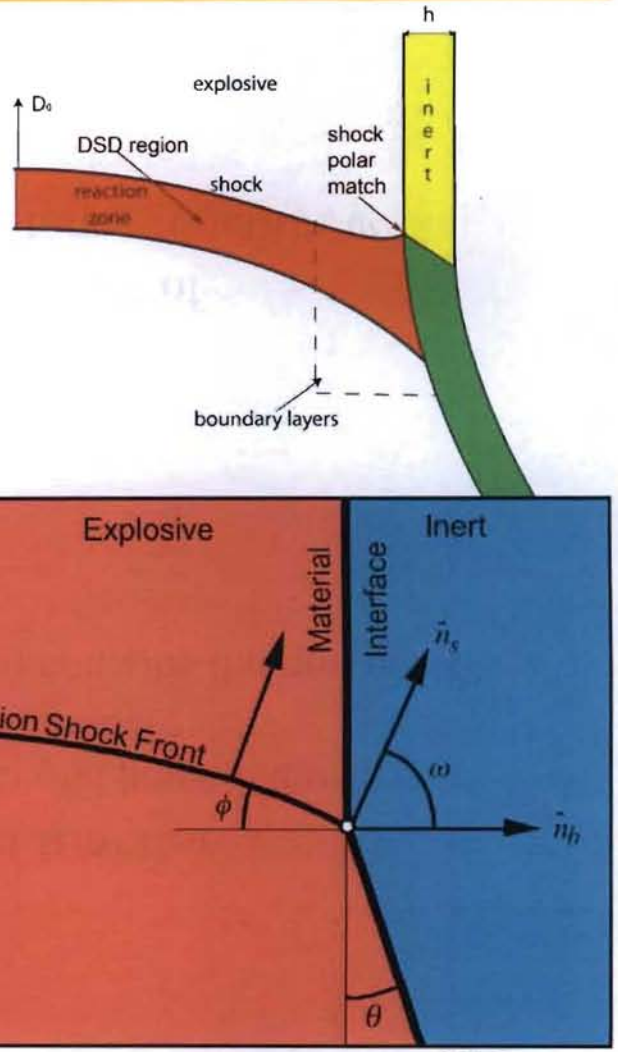
- **Detonation eigenvalue problem and $D_n(\kappa)$**
  - At leading order (ε^1) these scalings produce a "master" equation in the form

$$(U_n^2 - c^2)U_{n,\eta} = c^2[\sigma R(E, p, \lambda) - \kappa u_\eta / (1 - \eta\kappa)], \quad U_n \equiv D_n - u_\eta$$

  - Solving this 1D ODE for a given EOS and rate law $R(p, \rho, \lambda)$ yields $D_n(\kappa)$

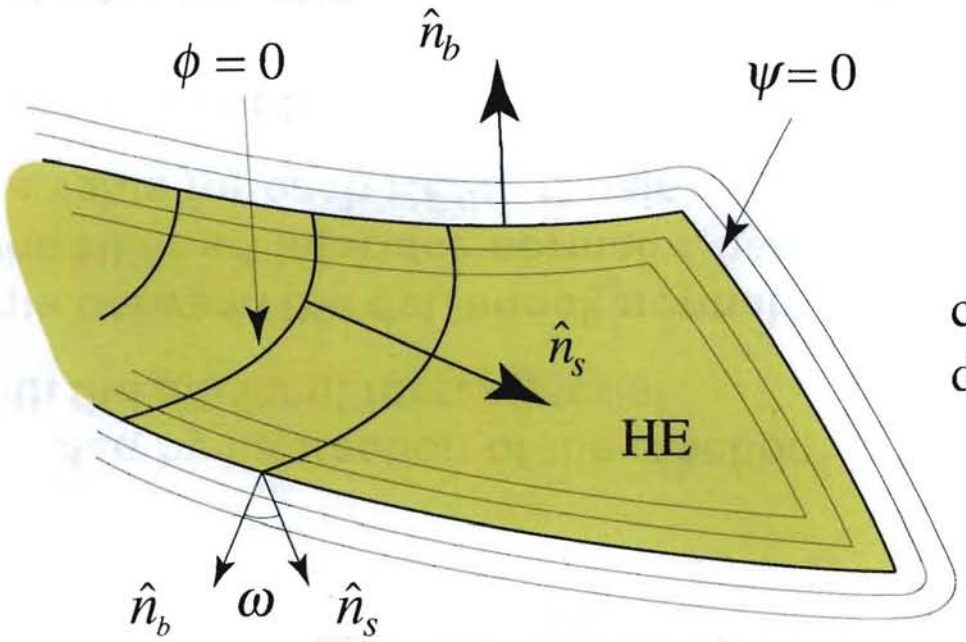
DSD boundary condition (BC) at HE/inert interface
Theory: Aslam, Bdzil, Sharpe, Stewart

- D_n is affected by interaction of the reaction zone with the adjacent inert material
- The angle between the det. shock normal $\hat{n}_s$ and the HE/inert interface normal $\hat{n}_b$ has a unique value for each HE/inert pair
- The value of ω_c is obtained by applying shock polar analysis. Oblique shock Hugoniot data for both HE and inert are plotted in the (p, θ) plane.
  - Confirmatory values have also been obtained using high-resolution DNS of reactive burn
  - Experimental work ("Sandwich" test) is ongoing, L. Hill



DSD numerics: schematic

- **Distance function ψ and detonation front level set function ϕ**
  - ψ is the minimum *signed* distance from an arbitrary DSD grid point to the nearest location on an HE/inert interface or HE/external boundary.
  - Also need matID (material ID) field: which material region contains the grid point?



DSD numerics: modified level-set equation

- **Propagating front is described by a modified level-set equation with the front corresponding to** $\phi(x,y,z,t)=0$
  - Standard level-set eqn: $\phi_{,t}+D_n(\kappa)\,|\vec{\nabla}\phi|=0,$ κ … burn front curvature
  - Re-distancing eqn: $\phi_{,\tau}+S(\phi)(|\vec{\nabla}\phi|-1)=0$

- **Solved jointly as**

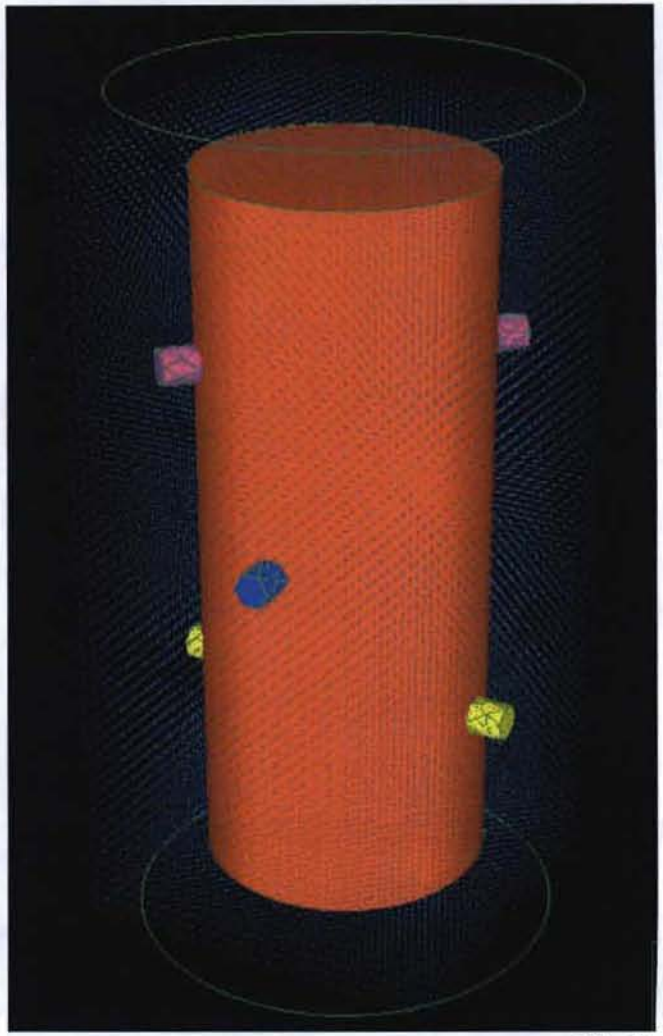
$$\frac{\partial\phi}{\partial t}+D_n(\kappa)\,|\vec{\nabla}\phi|=\frac{S(\phi)}{\varepsilon}\left[1-|\vec{\nabla}\phi|\right],$$ ε … num. parameter

- $D_n(\kappa)$ **form typically used for fitting experimental data**

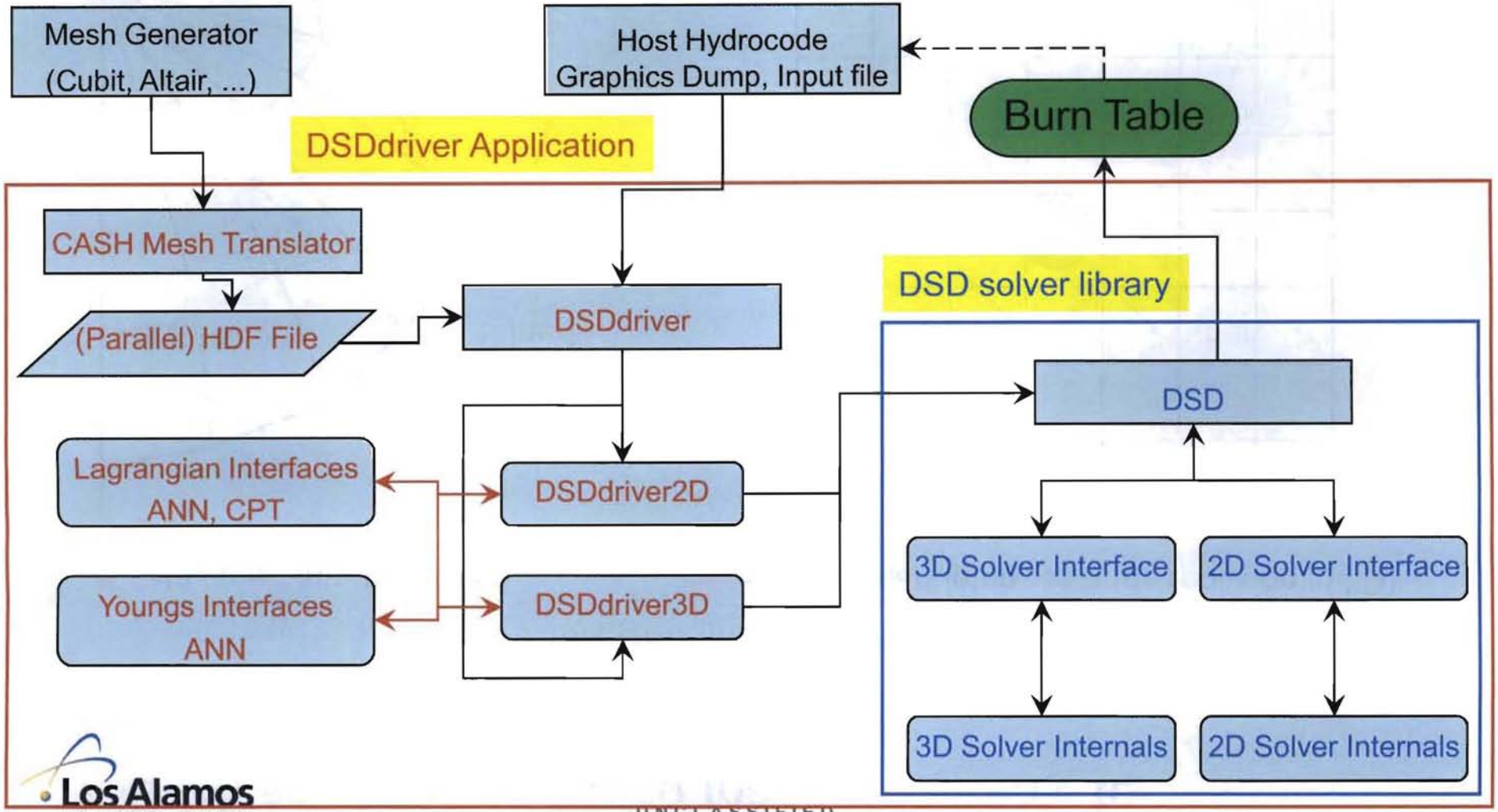
$$\frac{D_n(\kappa)}{D_{CJ}}=1+A\left[\left(C_1-\kappa\right)^{E_1}-C_1^{E_1}\right]-B\kappa\left[\frac{1+C_2\kappa^{E_2}+C_3\kappa^{E_3}}{1+C_4\kappa^{E_4}+C_5\kappa^{E_5}}\right]$$

LANL DSD code overview

- **DSD solver computes the evolution of the detonation front**
  - Front evolution computed on uniform DSD grid containing all HE and nearby adjacent inert materials.
  - Solver apprehends HE/Inert geometry using the distance function ψ and matID fields.
  - DSD grid is independent of hydrodynamics mesh; recommend finer than hydro mesh by 2x – 3x in linear dimension.
- **DSDdriver**
  - Prepares input data for DSD solver.
  - Determines ψ and matID fields using hydro mesh information.
  - Interpolates DSD solution back onto hydro mesh: creates burn table.

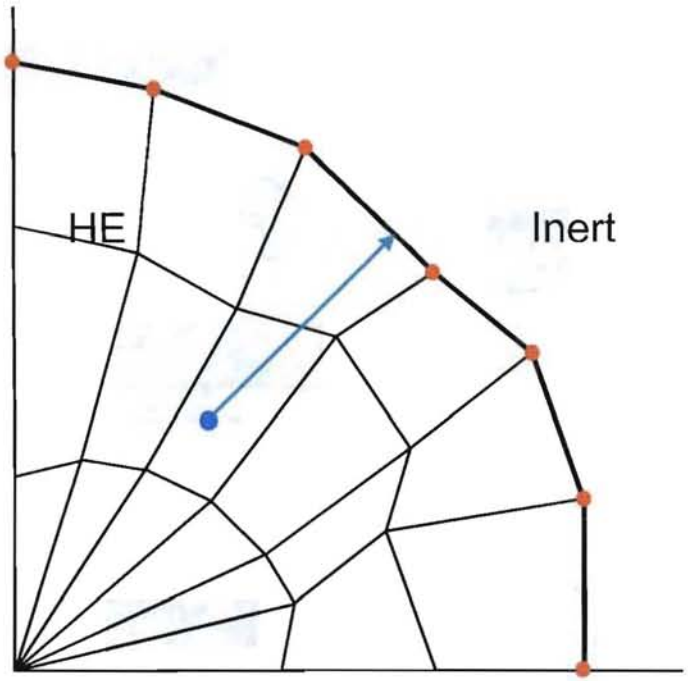


LANL DSD code structure (current)

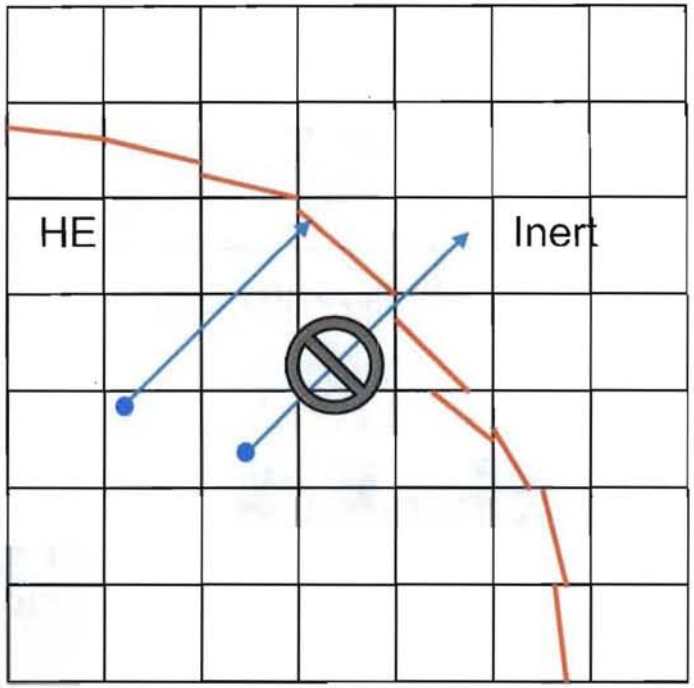


DSDdriver converts hydro mesh geometry to distance function and matID arrays

- Arbitrary DSD grid point
- He / Inert interface node
- HE / Inert interface edge (polygon in 3D)



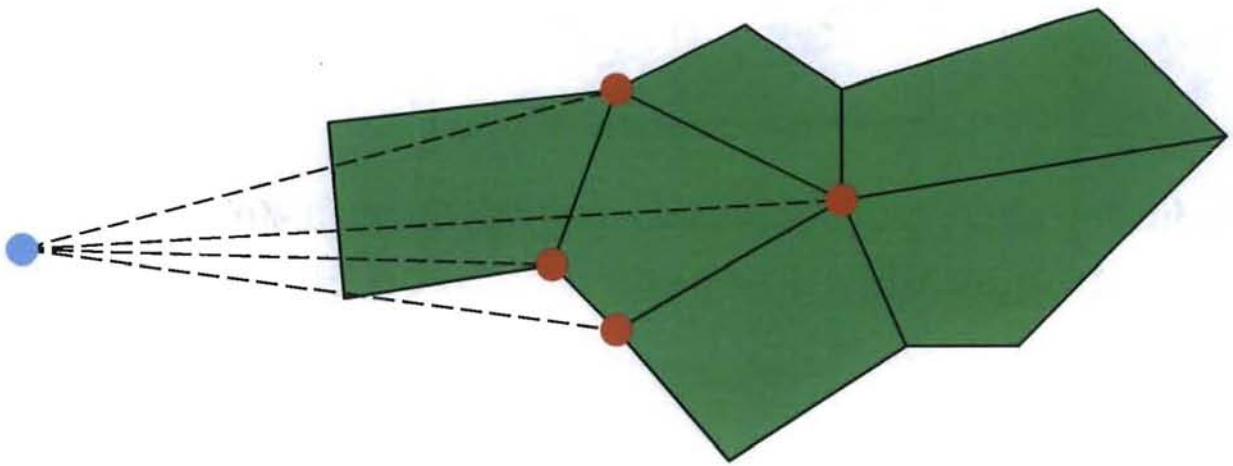
Lagrangian



Eulerian (Youngs')

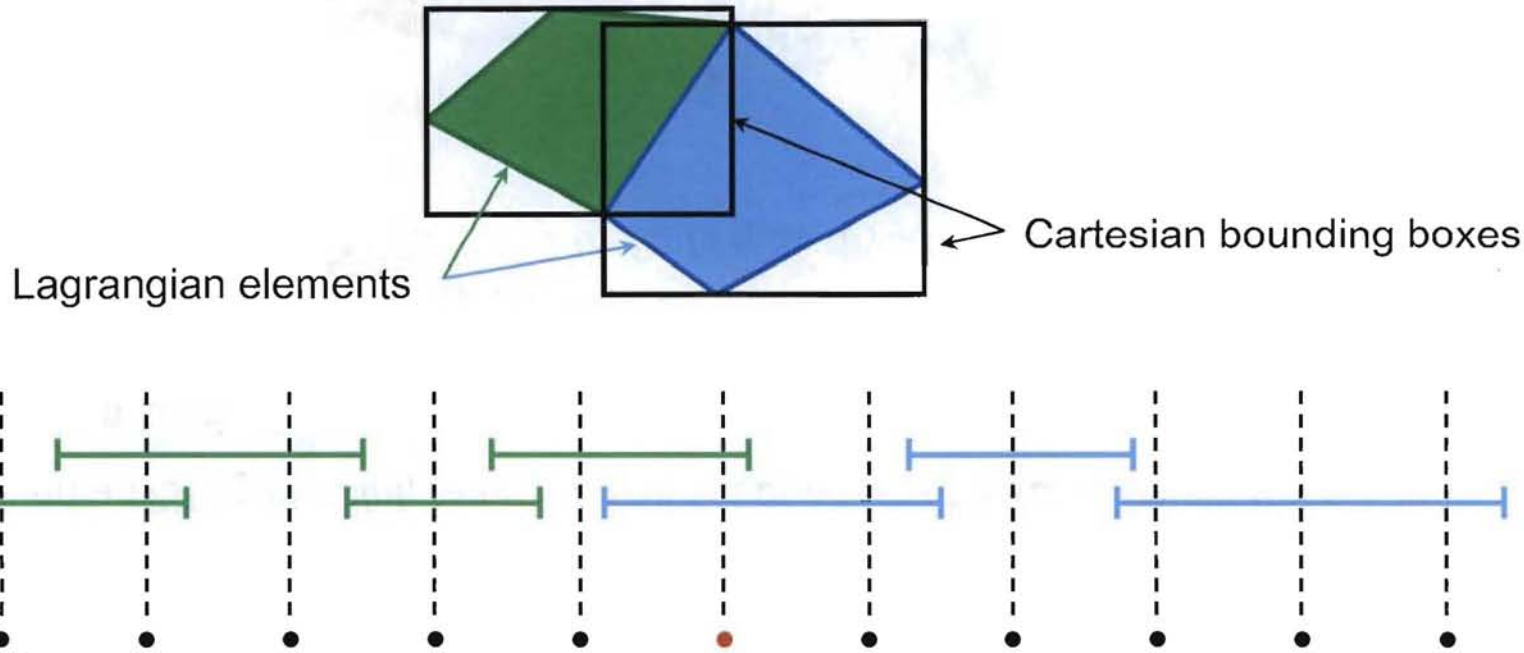
LANL DSDdriver scheme
Used for body-fitted and Youngs'-type interfaces

- **ANN (Approximate Nearest Neighbor) library**
  - ANN — nearest *point* search based on *KD-tree* algorithm.
  - Given a DSD grid *point*, returns small set of nearby hydro mesh interface *nodes.*
- **Algorithm to determine ψ**
  - Identify faces associated with nearby interface nodes.
  - Find minimum unsigned distance.
  - Determine sign of ψ and matID by finding the hydro cell that contains the DSD grid point.

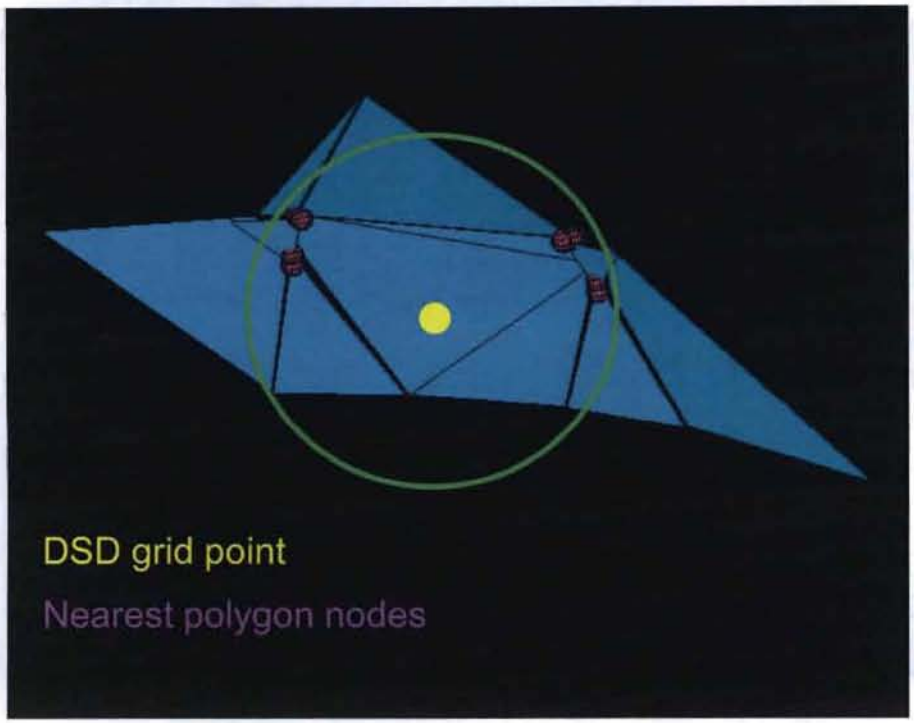


Body-fitted algorithm

- **Uses "Cell bounding boxes" to speed-up sign determination**
  - Application of ray-throwing "point in polyhedron" algorithm directly to hydro regions much too slow.



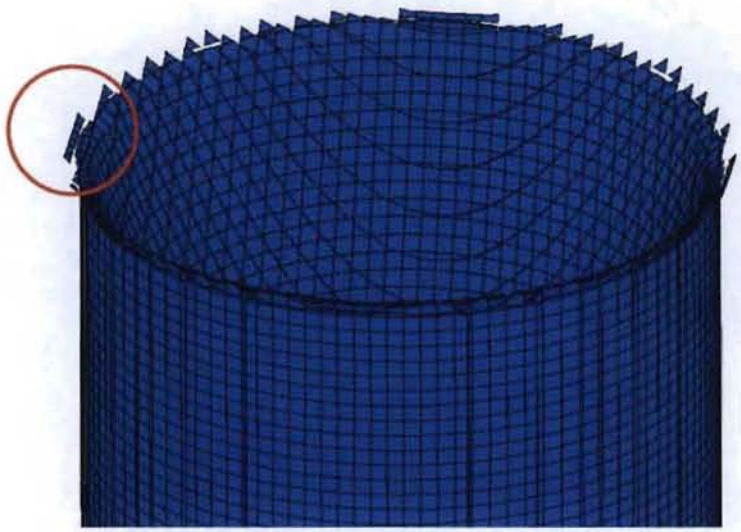
Youngs' interface (Eulerian / ALE) algorithm (Pagosa)

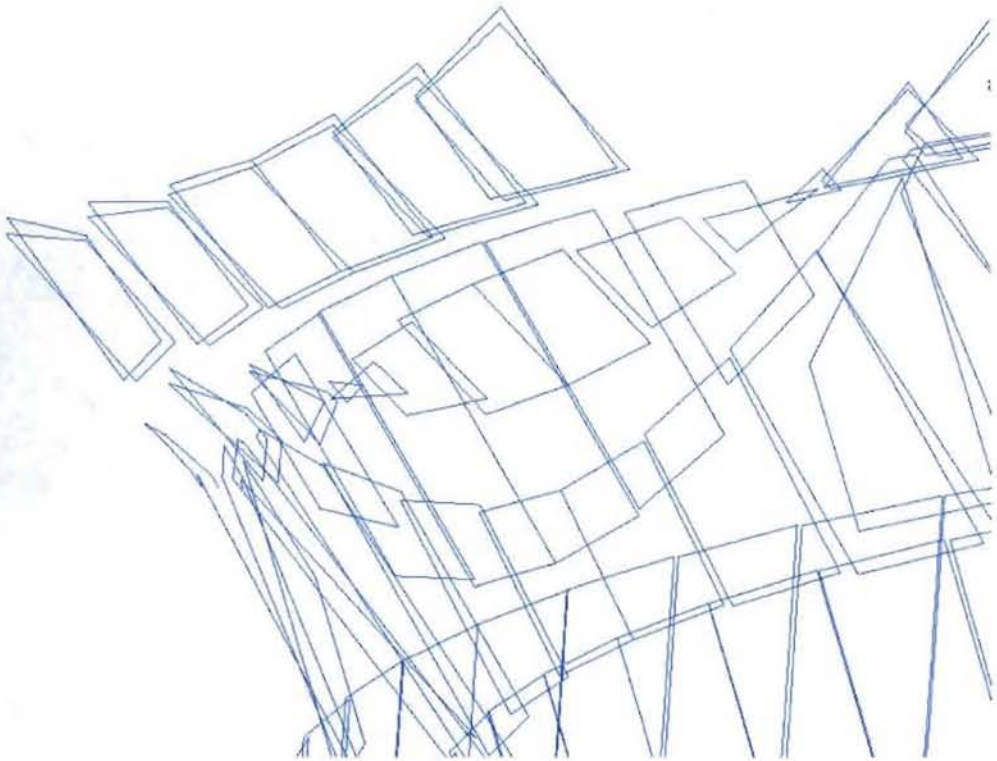


- **Essentially the same algorithm as the body-fitted ANN scheme**
  - Find nearest interface nodes and distances to associated polygons.
  - Generally need larger set of nearest nodes than for body-fitted interfaces.

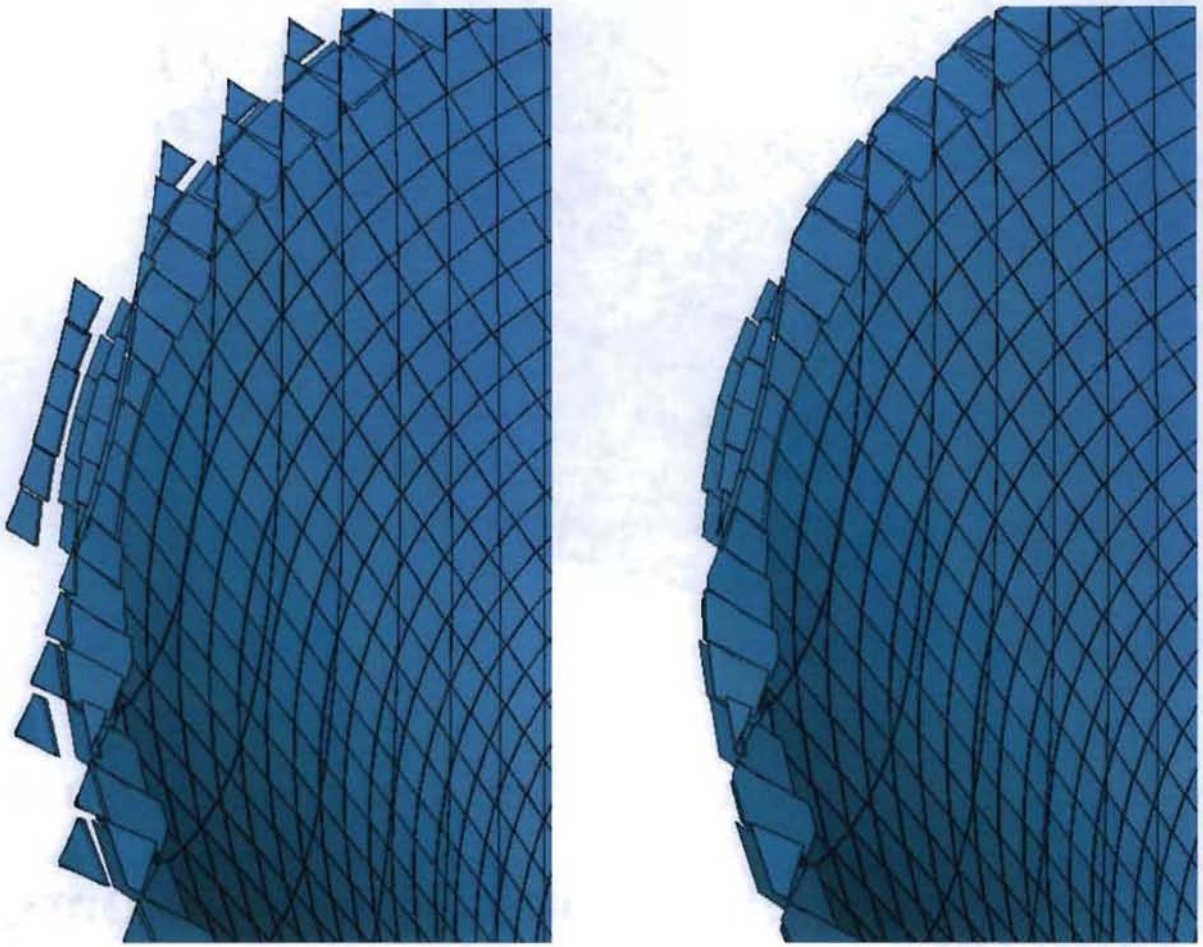
1st-order Youngs' algorithm can produce poorly defined interface (Pagosa)

- **Two-pass operation to create Youngs interface polygons**
  - One against lower priority, one against higher priority materials.
  - Passes are independent.



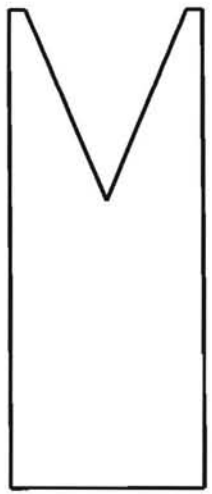


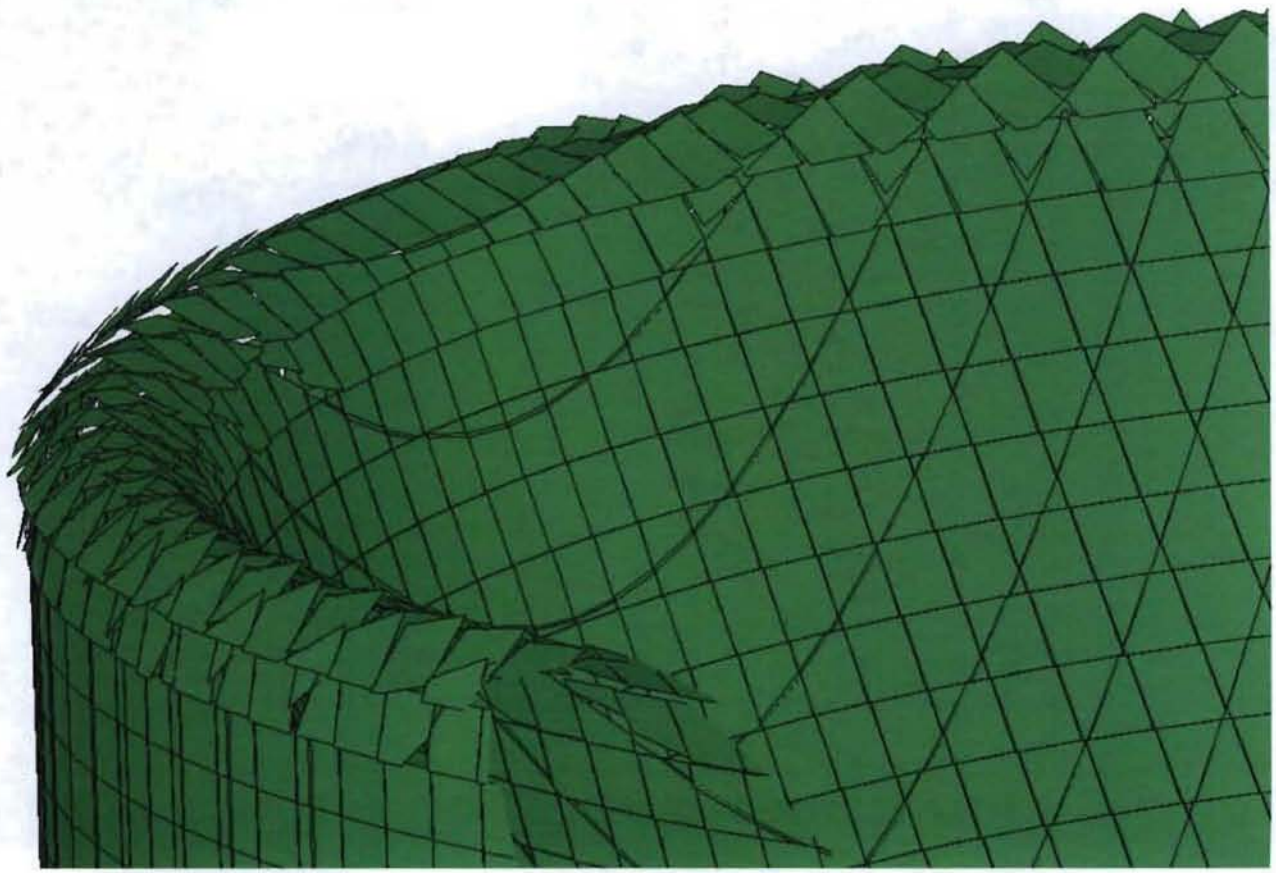
Elimination of intersecting polygons not difficult



1st-order Youngs' construction—blunt edge

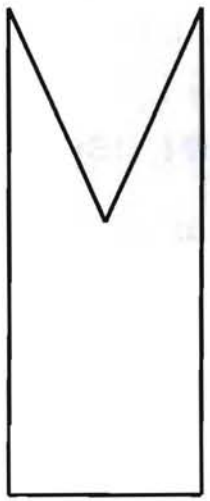
- **DSDdriver handles this without problems.**

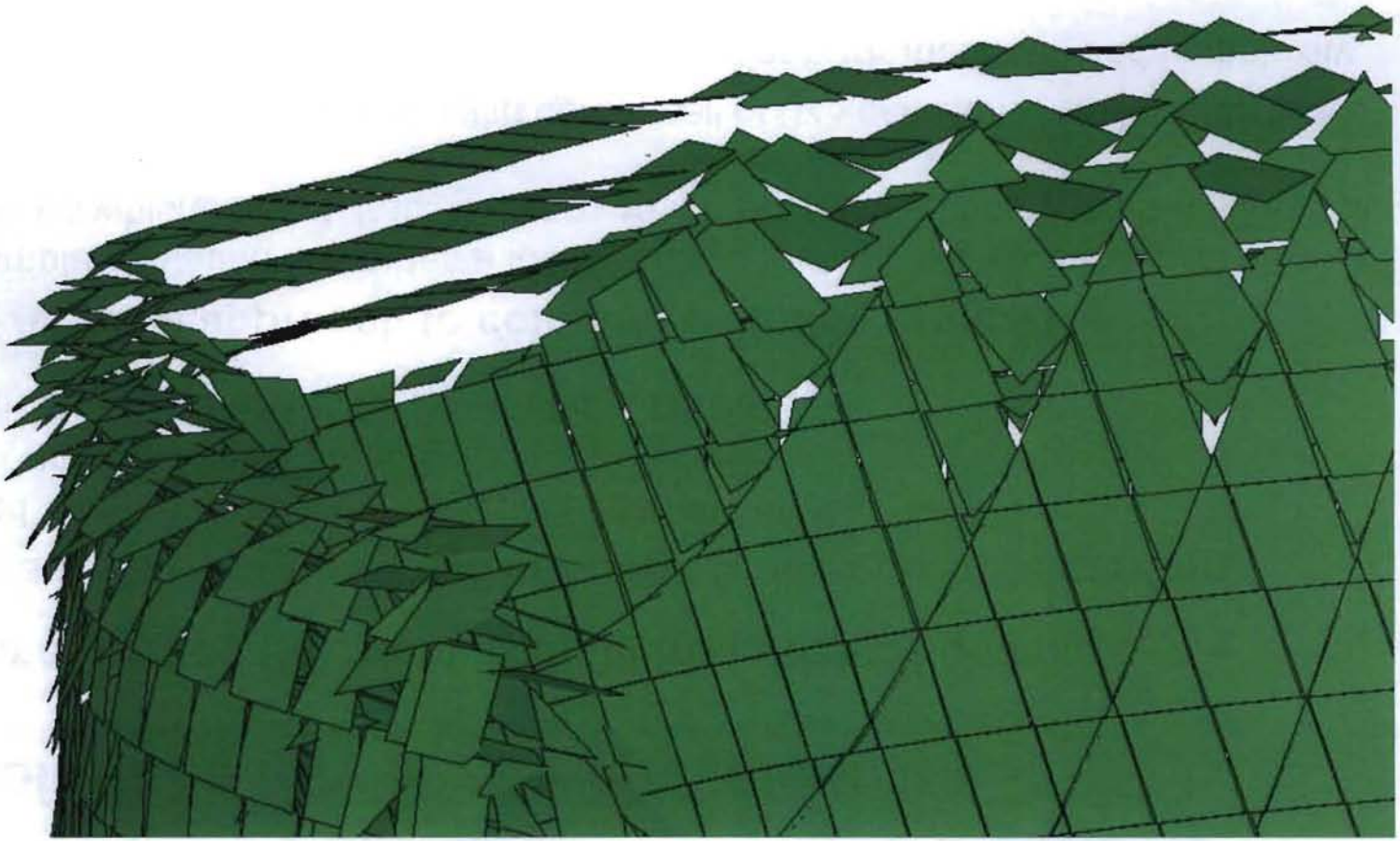




1st-order Youngs' construction—sharp edge

- **Even with intersecting polygon removal DSDdriver can't run this.**





Possible solutions for "shaped-in" meshes

- **Take volume fraction (VF) data from hydro initialization and …**

1) Construct a better Youngs'-type interface & use current algorithm.

- Patterned Interface Reconstruction package (Jay Mosso, Sandia).
  2nd-order accurate, fairly mature.
- Written in C++, will need to be interfaced to DSDdriver.

2) Use Tariq Aslam's approach to compute ψ directly from VF

- 2D trial implementation computes a locally planar "best fit" representation of psi in a cell mesh on which the vol. frac. is given. Point rep. is obtained by averaging to cell rep. to common point.
  Works by varying orientation of psi in center cell of 3x3 cell stencil while VF is fixed.
- Trial impl. is fairly simple but dealing with non-uniform VF mesh will add complexity.

3) Use elements from 1) to help do 2)

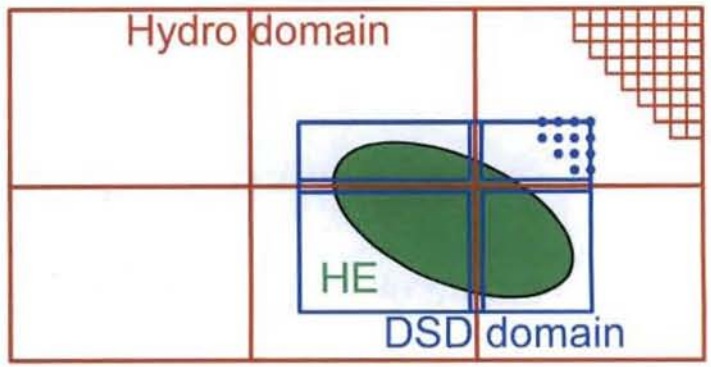
4) Other ???

Parallelization of LANL DSD: driven by run-time and memory requirements

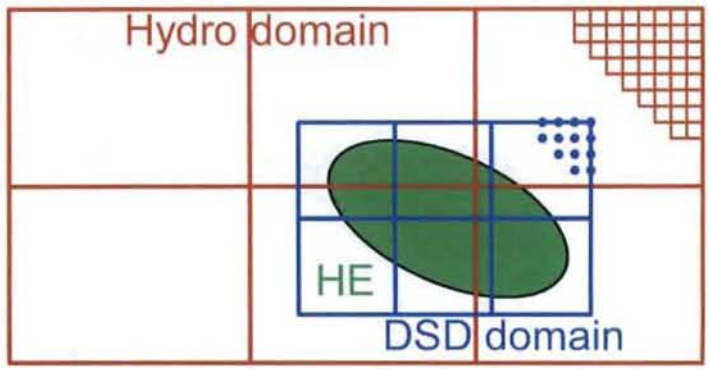
- **Execution time**
  - For a small 3D problem (~2.0M DSD grid points) run time is ~two hours on a 3.8 GHz Linux box.
  - 3D "production" run size is expected to grow from O(10^7) to O(10^9) DSD grid points in the not-distant future.
- **Memory usage**
  - At 32 GB per compute node, the number of DSD grid points that can be handled by is estimated to be at most O(10^7).
  - If run in-line during the hydrodynamics "generator" phase, available memory may be reduced considerably.
- **Advanced Architectures**
  - For high performance, memory footprint must be reduced drastically.

MPI parallelization paradigms

- **Use same partitioning scheme for hydro and DSD sub-domains**
  - DSD uses a subset of the processors requested for hydro.
  - No hydro-DSD mapping issues.
  - Performance hit could be severe if HE occupies a small fraction of full hydro domain. Likely with Eulerian or ALE codes.

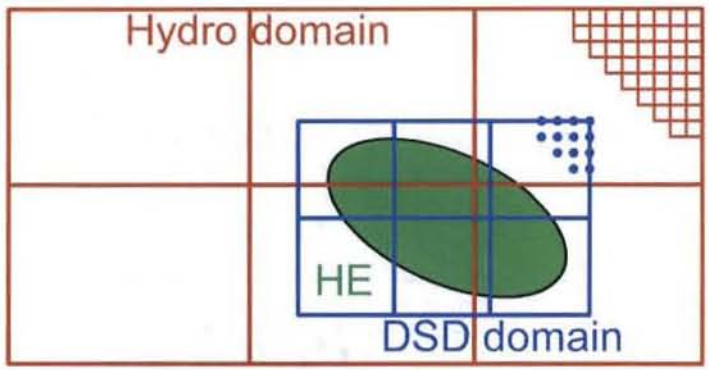


- **Use independent partitioning schemes for hydro and DSD domains**
  - Additional processors required beyond those needed for hydro.
  - Hydro-DSD sub-domain mapping required.
  - Independent DSD sub-domains allow user control of comp. resources vs. run time for burn table generation.

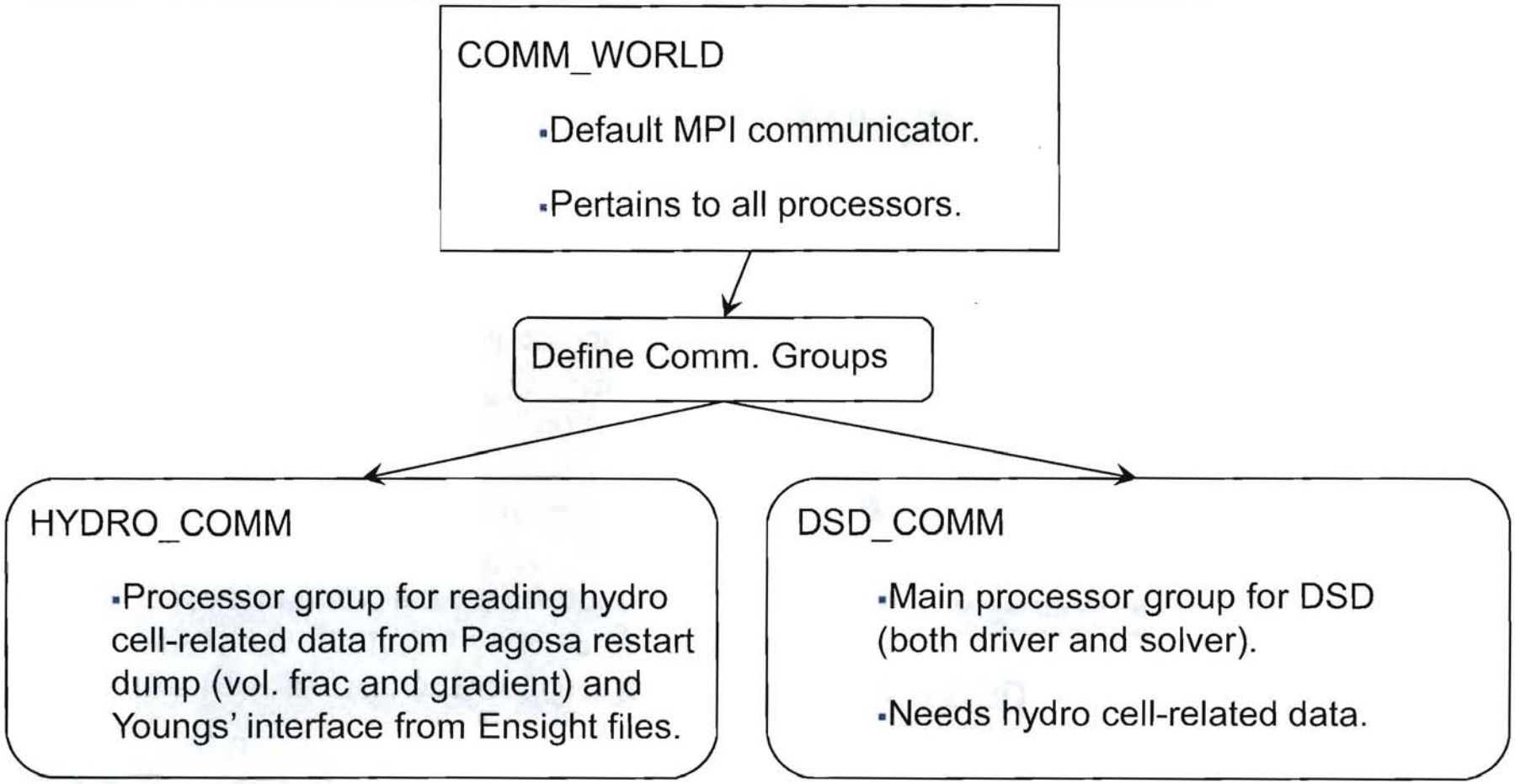


Design being implemented

- **Use independent partitioning of hydro and DSD domains**
  - Initial target hydro code, Pagosa, has separate generator and physics executables. Requesting different numbers of processors for setup (including DSD) and the actual hydro calculations can be handled in batch queue scripts.
  - After parallel DSD sub-domains are setup by DSDdriver, the DSD solver is invoked on each separately.
  - MPI allows for parallel communication at the boundaries of the DSD sub-domains during generation of the DSD solution.
  - Similar approach will be used for other host hydrocodes: Ale3d, Rage, etc.



Parallel DSD flowchart (MPI)



Parallel DSD flowchart (MPI)—cont.

HYDRO COMM GROUP

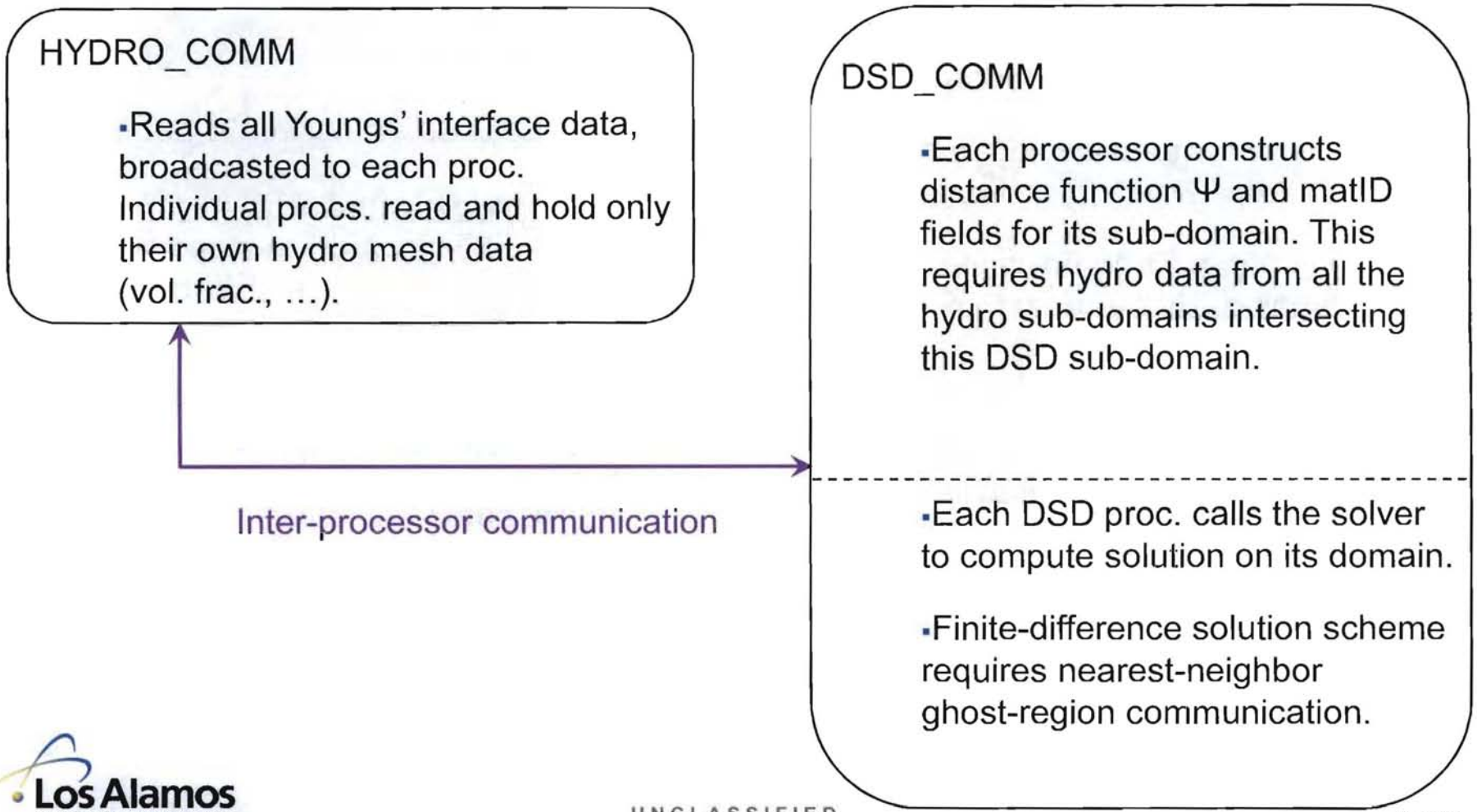
- Creates hydro sub-domains.
- Gathers sub-domain size data (begin / end coordinates) on the root processor.
- Root sends hydro sub-domain size data to DSD COMM GROUP using MPI_SEND.

Inter-processor communication

DSD COMM GROUP

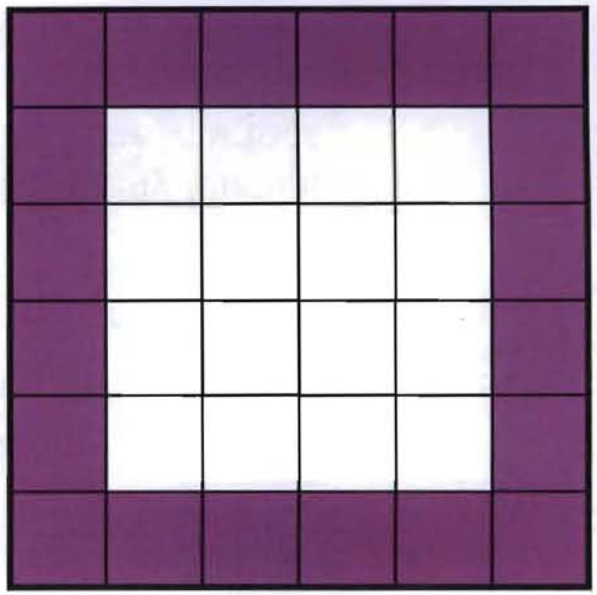
- All processors in this comm. read all the Youngs' interface polygons.
- Creates DSD sub-domains.
- Root receives hydro sub-domain data using MPI_RECV.
- Creates DSD-to-hydro sub-domain map.
- Broadcasts DSD-to-hydro map to all DSD processors.

Parallel DSD flowchart (MPI)—cont.



PARALLEL BURN FRONT SOLUTION GHOST REGIONS

- **Ghost regions flank sub-domains on interior boundaries.**
- **Facilitate exchange of data between adjacent processors.**
- **Keeps data current across the mesh at each time step.**
- **Difference scheme is compact: 3x3x3 stencil (19 points) yields 2nd-order accurate κ.**
  - At interior DSD sub-domain boundaries ghost region thickness need be just one dx.



ARDEC - ALE3D work

- **Ultimate Objectives**
  - An in-line, parallel interface (2D and 3D) to allow LANL DSD burn table to be created and run for general Ale3d meshes (body-fitted and/or shaped-in).
  - A small set of well-characterized DSD verification problems.
  - DSD parameter sets and pseudo-reaction zone (PRZ) model(s) for IHEs of interest as time and $ permit.
- **Status**
  - Code interface work begun, "first-cut" file-based interface nearly functional. Najjar-Neely code to collect mesh data will form basis of in-line interface.
  - Mathematica package to generate "exact" solution to 2D rate stick problem almost complete. Similar approach will be used for other verification problems.
  - A PRZ model for PBX9502 has been provided to LLNL. Model(s) for Army materials await experimental results.