Your corporate
logo here

Some Issues in the Development of Overset Grids CFD using One-Sided Communication

YikLoon Lee

Naval Air System Command

Patuxent River, MD

CUG 2009
Compute the Future

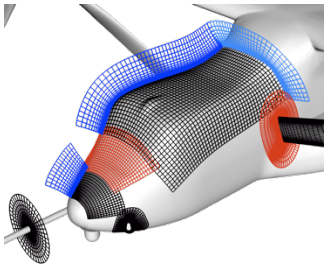
NAVAIR

Outline

- Introduction
  - Overset Grids, Data Communication
- Communication in Overset Grid Connectivity
  - Donor Cell Search
  - Overlap Status, etc.
- Communication in Overset Grid Boundary Interpolation
  - Two-sided Communication
  - One-sided Communication
- Conclusion

Overset Grids in Numerical PDE

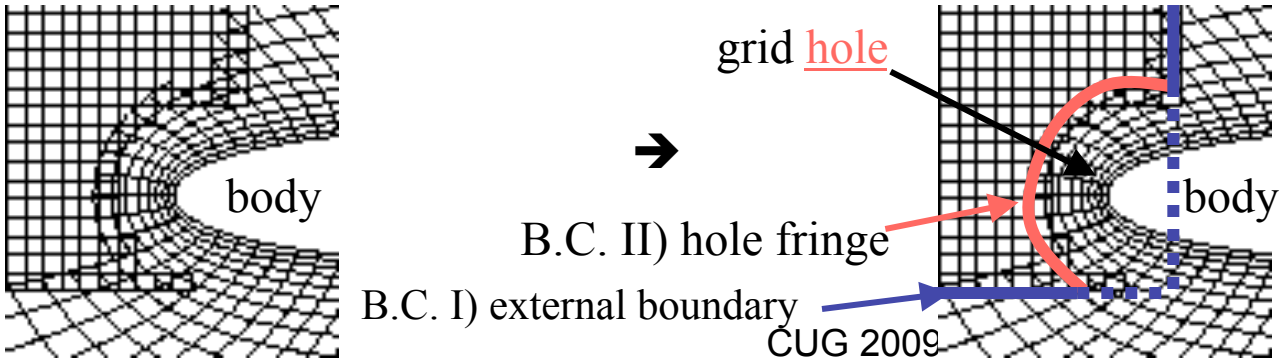
- Multiple arbitrarily overlapping grids
  - structured or unstructured
- Advantage in flexibility
  - Essential if **relative motion between components** is present
  - Especially beneficial for large motion of small objects
- Moving grid examples
  - Flapping wing, rotorcraft, aircraft store separation, rocket stage separation, ship-aircraft interface, pilot ejection, turbomachinery, ….

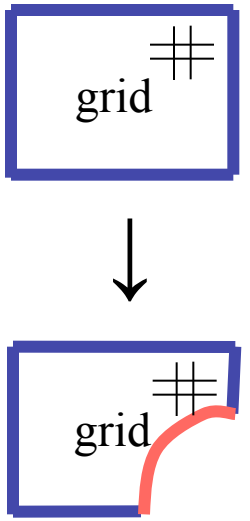


V-22 overset grids
Courtesy NASA Ames

Overset Grid Boundaries

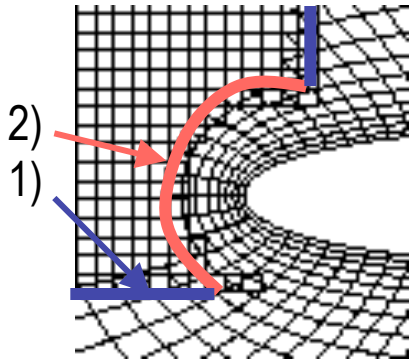
- Price for flexibility of overset grids
  - Grids often extend inside solid bodies (left fig.)
- Points inside bodies (holes) are invalid
  - Must be identified
- This results in **two types of B.C.** (boundary condition) (right fig.) :
  - I) Usual external boundary (blue line) – trivial
  - II) Fringe points of any cut-out hole (red line)





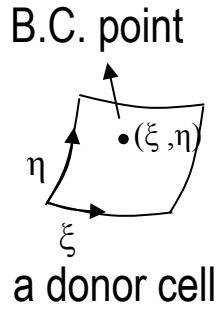
Overset Grids Connectivity

- The determination of -
  - (A) all B.C. points :
    - 1) External boundary – trivial
    - 2) Hole fringe – from **hole-cutting**

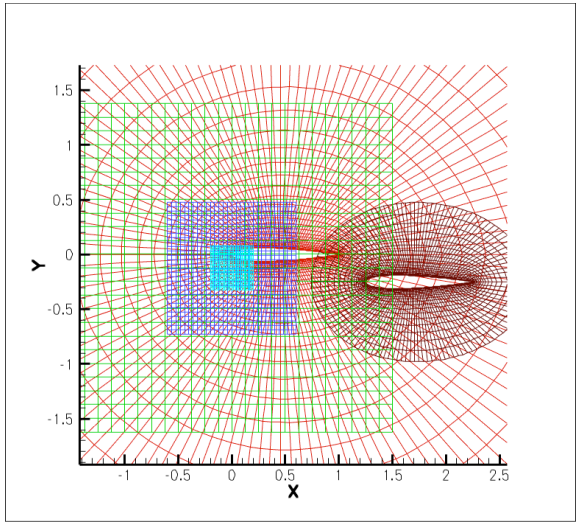


  - (B) donor cell information for each B.C. point :
    - 1) Grid number
    - 2) Cell number
    - 3) Position (ξ ,η) of B.C. point inside donor cell

(Note : (B) is required for interpolation)



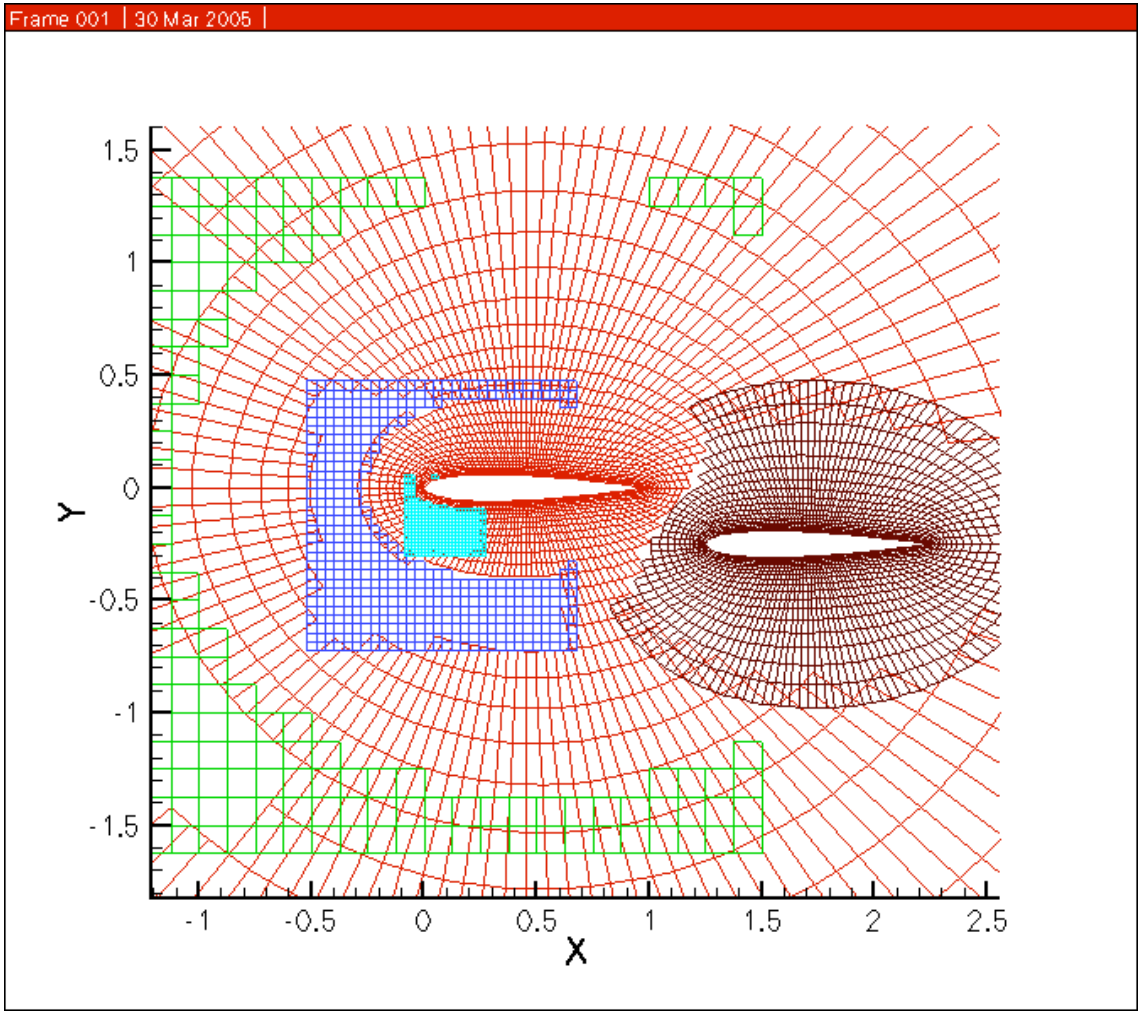
2D Example – Movie



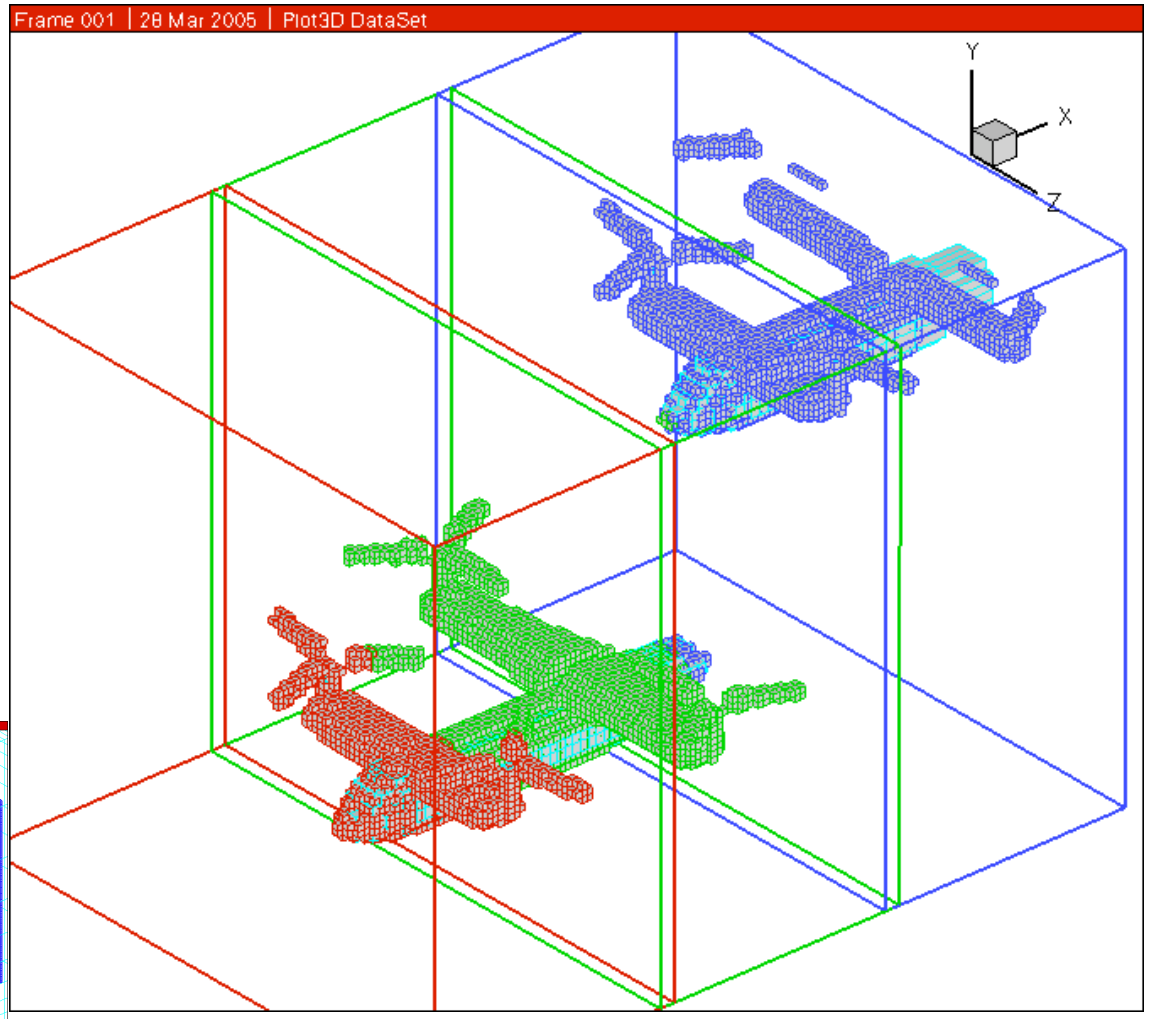
Original grids

5 grids : 2 airfoils + 3 squares

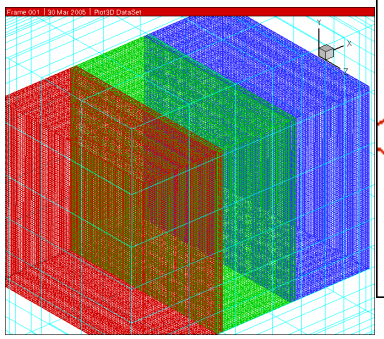
Many holes are being cut out



3D Example – Movie

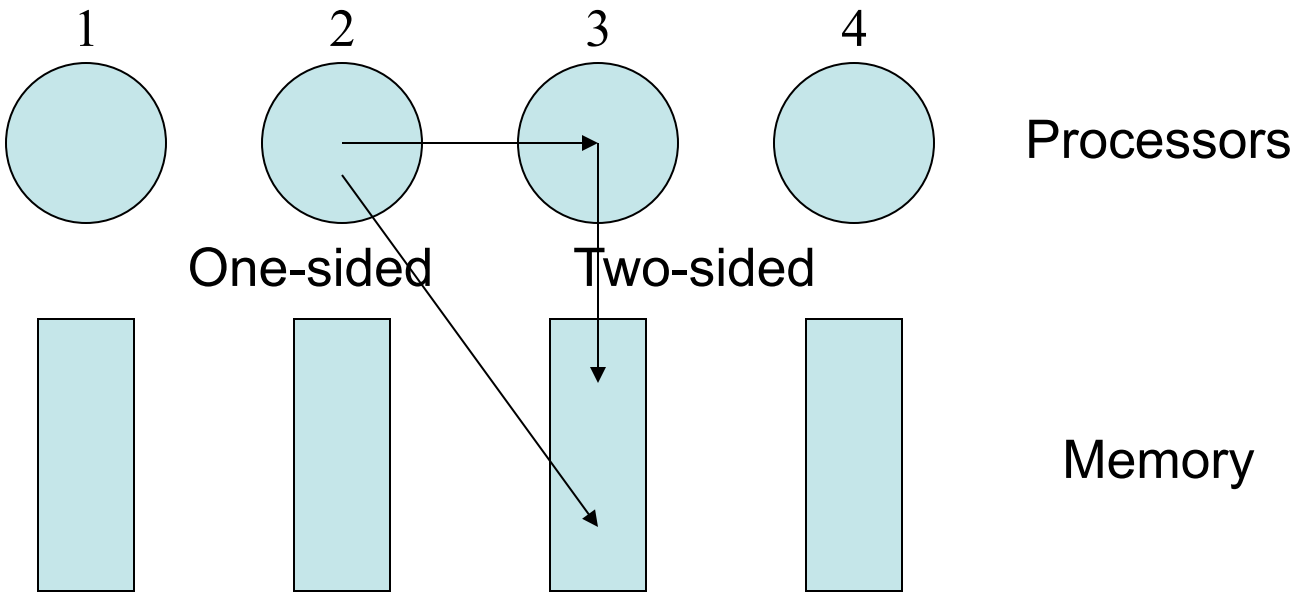


background grids



Communication Models

- Two-sided model : needs remote coordination
- One-sided model : no coordination (PGAS)



NAVAIR

PGAS + Overset Grids

- Most examples studied by PGAS languages are non-overset grids
- PGAS and overset grids communities are relatively small and rarely interact with each other
- PGAS + overset grids
  - virtually non-existent
  - Provides rich environment for research and study

Data Communication in Overset Grids

- Comm. needed at 2 of 3 instances in a time step

  communication needed?

  - 1): overset grid connectivity -> yes
  - 2): solver -> no
  - 3): boundary value interpolation -> yes
- Very interesting to study characteristics of one- & two-sided models on 1) & 3)
  - Very different in programming and communication efficiency

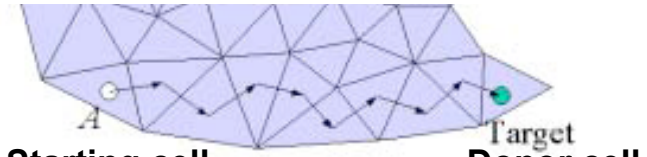
NAVAIR

Outline

- Introduction
  - Overset Grids, Data Communication
- **Communication in Overset Grid Connectivity**
  - Donor Cell Search
  - Overlap Status, Index Range, etc.
- Communication in Overset Grid Boundary Interpolation
  - Two-sided Communication
  - One-sided Communication
- Conclusion

Overset Grid Connectivity
A Few Words

- Search for donor cells of boundary points
  - One of the main operations of connectivity algorithm
- Simple exhaustive search is too slow
- Overlap status of all grid-pairs
  - Smart search for acceleration



NAVAIR

Donor Cell Search

- Remote data needed for search - nodal coordinates
- Each search iteration requires small amount of data
- What data needed next is determined every iteration
- Only one-sided model can accommodate many small messages
- Two-sided model
  - transfer all or most data a priori - not efficient

NAVAIR

Non-Overlap Status of a Grid-Pair

- Several possible statuses, but only consider 'non-overlap' status
- 'Non-overlap' is known when last boundary point of any one of the grid-pair is done
- Once known, a message is immediately sent to the remote CPU of the other grid
  - message is pushed to the remote CPU
  - remote CPU has no knowledge when it may come
- Again, two-sided model is incompatible

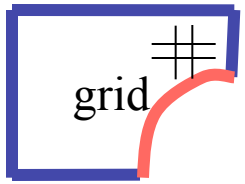
NAVAIR

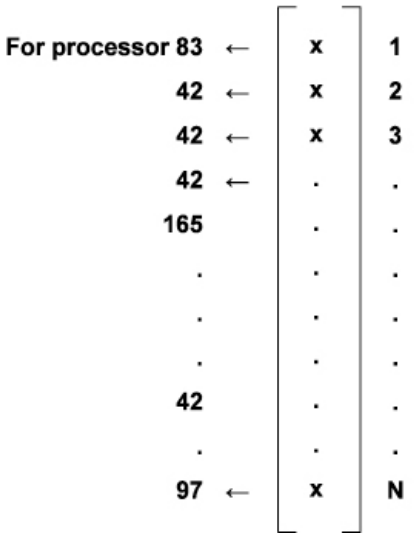
Outline

- Introduction
  - Overset Grids, Data Communication
- Communication in Overset Grid Connectivity
  - Donor Cell Search
  - Overlap Status, etc.
- Communication in Overset Grid Boundary Interpolation
  - **Two-sided Communication**
  - One-sided Communication
- Conclusion

Two-sided Model

- Each grid has a list of boundary points (changes in time)
  - Order in the list pre-determined in connectivity algorithm
- Remote CPUs provide interpolated boundary values
- In any remote CPU *n*:
  - Go through the list
  - Gather donor cells :- Identify the subset of points whose donor cells are in *n*
    - In general, this subset is non-contiguous
  - Interpolate boundary values from donor cells in the subset
  - Store values in `Qsend(:,nproc);nproc` = receiver CPU ID





NAVAIR

Two-sided Model (2)

- Synchronize
  - Wait for all interpolation to complete before receiving new and overwriting old boundary values in CPU *n*
- Communicate
  - Send `Qsend(:,nproc);` `nproc` = receiver CPU ID
  - Receive `Qrecv(:,nproc);` `nproc` = sender CPU ID
- Scatter `Qrecv` to boundary points
- Complicated because of the gather/scatter (non-contiguity)
  - Additional book-keeping, receiver point-counting, donor cell-counting, new arrays (variable size), complex code logic

NAVAIR

Two-sided Model (3)

- Two-sided model : remote CPUs have to do the interpolation because only one bulk communication is allowed.
- After boundary points are updated, the next time step can begin immediately
  - Because no more local solution are needed by any CPU
- No more synchronization!
  - Combined load balancing of all 3 - connectivity, solver and interpolation/communication
  - An advantage compared with one-sided model

NAVAIR

Outline

- Introduction
  - Overset Grids, Data Communication
- Communication in Overset Grid Connectivity
  - Donor Cell Search
  - Overlap Status, etc.
- Communication in Overset Grid Boundary Interpolation
  - Two-sided Communication
  - **One-sided Communication**
- Conclusion

NAVAIR

One-sided Model

- Much simpler (than two-sided model) in code logic
- Modification to only one statement (besides co-array declarations and load balancing) -
  - `Q(ngrid)%node(i,j,k)` changed to `Q(ngrid)[np]%node(i,j,k);` where `np=nproc(ngrid)` is CPU ID of donor grid `ngrid`
  - Note : It accesses dataset of the entire donor cell stencil
- But there are two significant downside
  - a) many times more data need to be transferred
  - b) another synchronization required, indirectly because of a)

NAVAIR

One-sided Model (2)

- a) is due to the transfer of entire remote stencil dataset for interpolation - Interpolation is done locally
- If interpolation is done remotely, then remote CPU has to gather its own subset of donor cells - just like back to two-sided model
- Remote CPU can not begin the next time step even after is has completed its own boundary update
  - because the local interpolation uses remote data, which could otherwise be overwritten
- This means another synchronization!
  - Combined load balancing impossible
  - Severe penalty if interpolation is high order

Interim Summary - Communication and Interpolation

- Two-sided model
  - Inefficient programming > bad
  - Much smaller message size > good
  - Only one synchronization > good
- One-sided model
  - Exactly opposite

Interim Summary - Communication and Interpolation (2)

- 1-sided : communicate, then interpolate locally
- 2-sided : interpolate remotely, then communicate

a) Conventional grids	b) Overset Grids
.	.
.	.
Compute	Compute
Communicate	Communicate / interpolate
.	.
Compute	Compute
Communicate	Communicate / interpolate
.	.
.	.

What about combining the two?

- Meaning
  - we want remote interpolation, so we can
    - 1) pass only interpolant (not entire stencil)
    - 2) no 2nd synchronization
  - but remote interpolation -> non-contiguous subset of boundary points (complex code)
- Inherently no way out!
- Only overset grids are prone to this problem
  - Because of inseparability of communication and interpolation

NAVAIR

Conclusion

- Overset grids is powerful for moving body problems
- Naturally avoids difficulties of non-overset grids
- But burden is on A) grid connectivity and B) boundary interpolation, both need data communication
- A) clearly favors 1-sided model
  - Programming efficiency and new algorithmic capability
- B) not so clear
  - 1-sided : saves developer time, wastes CPU time
  - 2-sided : wastes developer time, saves CPU time