24th WSSP
Dec. 5-6, 2016

Coupled Multi-Physics Simulation Framework on Octree Data Structures

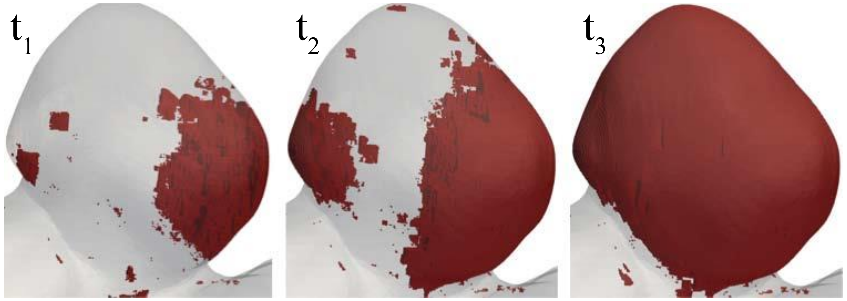
Kannan Masilamani, Verena Krupp, Sabine Roller
Email: kannan.masilamani@uni-siegen.de

Outline

- Introduction to Multiphysics Applications
- Coupling Requirements
- APES Simulation Framework
- APESmate: Integrated Coupling Environment
- Results:
  - Load Balancing
  - Strong scaling
  - Coupled Simulations
- Conclusion and Current Work

Introduction: Application with Multi-physics

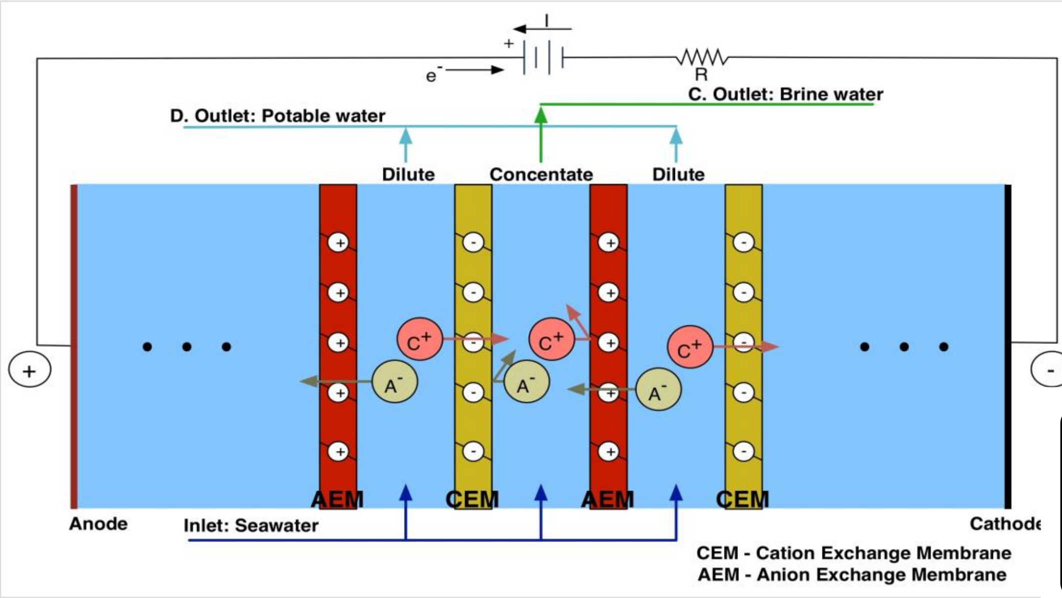
3-field surface coupling
largest domain - acoustic far field
→ large grid cells, higher order

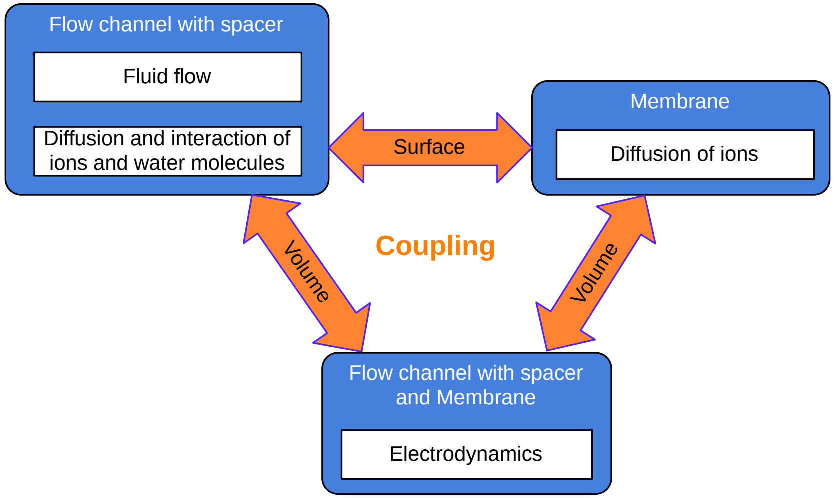


(a) Clotting process inside an aneurysm

2-field volume coupling

Introduction: Application with Multi-physics





Electrodialysis:

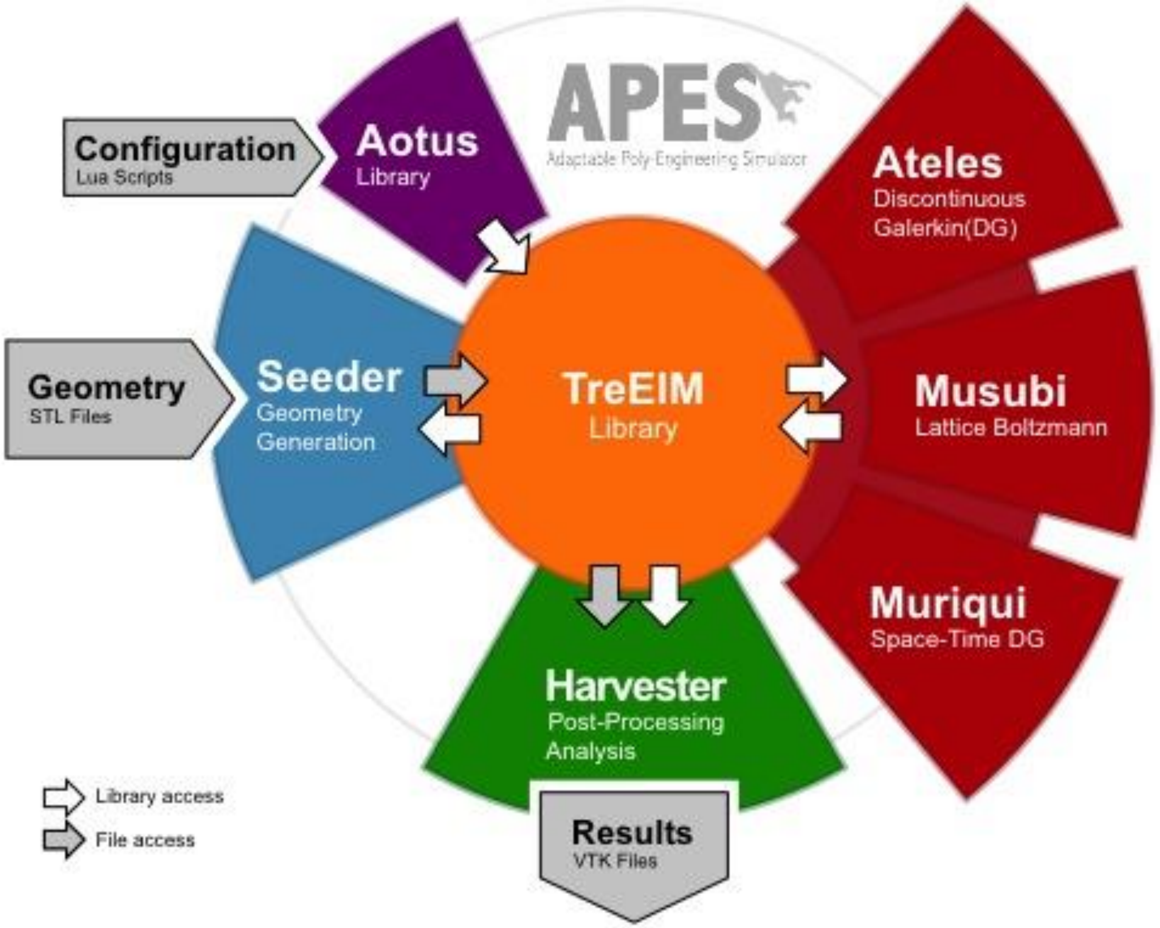
Multi-physics heterogeneous system

3-field surface and volume coupling

Coupling Requirements

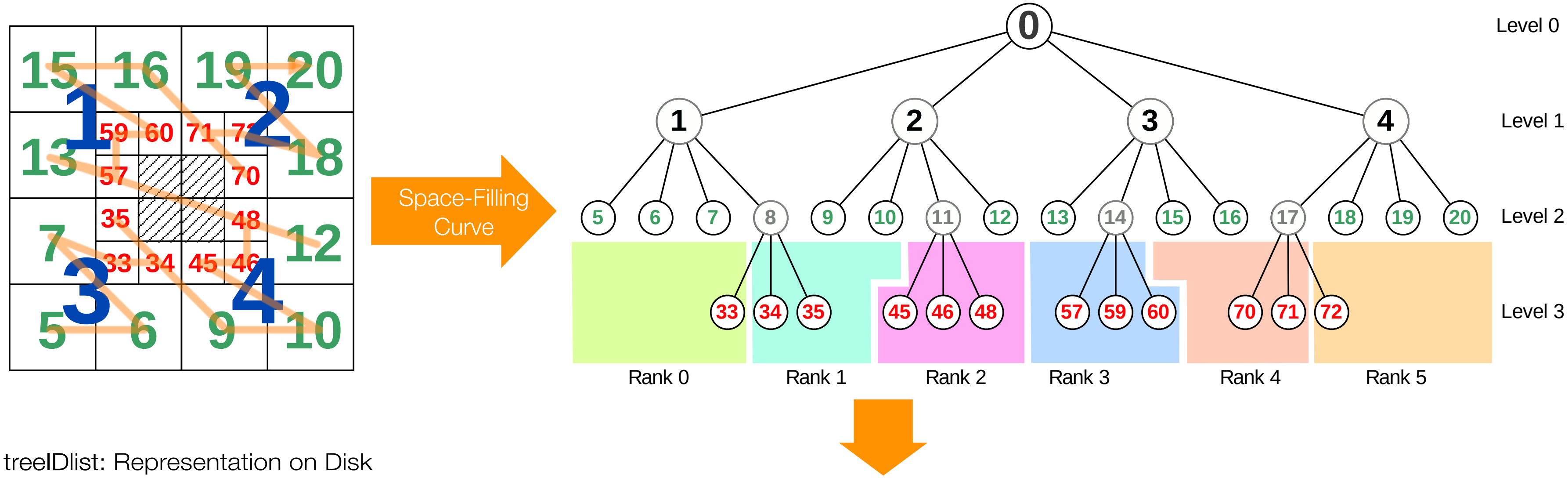
- Exchange of data from simulations in the space time domain between domains/solvers
- Coordinated execution of involved solvers
- Data exchange in the volume or the surface at common points in time or not coinciding points in time
  - Interpolation in space
  - For not coinciding time intervals, interpolate also in time
- Mesh independent
- Exploiting advantages of numerical solver to interpolate spatial values
- Individual numerical solver must be efficient and scalable in supercomputers
- Coupling should be scalable too
  - Load balancing of domains
  - Distribution of domains over nProcs_tot

APES framework



- Simulation framework for large scale parallel computations
- Based on Octree meshes
- End-to-end distributed memory parallel (MPI)
- Allows coupling of solvers
- Musubi:
  - Lattice Boltzmann Method
  - Multi-component LBM
  - Passive scalar
- Ateles:
  - High order discontinuous Galerkin solver
  - Euler equations
  - Linear-euler equations
  - Navier-Stokes equations
  - Maxwell equations

Octree data structure



treeIDlist: Representation on Disk

5	6	7	33	34	35	9	10	45	46	48	12	13	57	59	60	15	16	70	71	72	18	19	20
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24

- 8 Byte Integer: TreeID
- 8 Byte Integer: Property
- Distributed IO straight forward

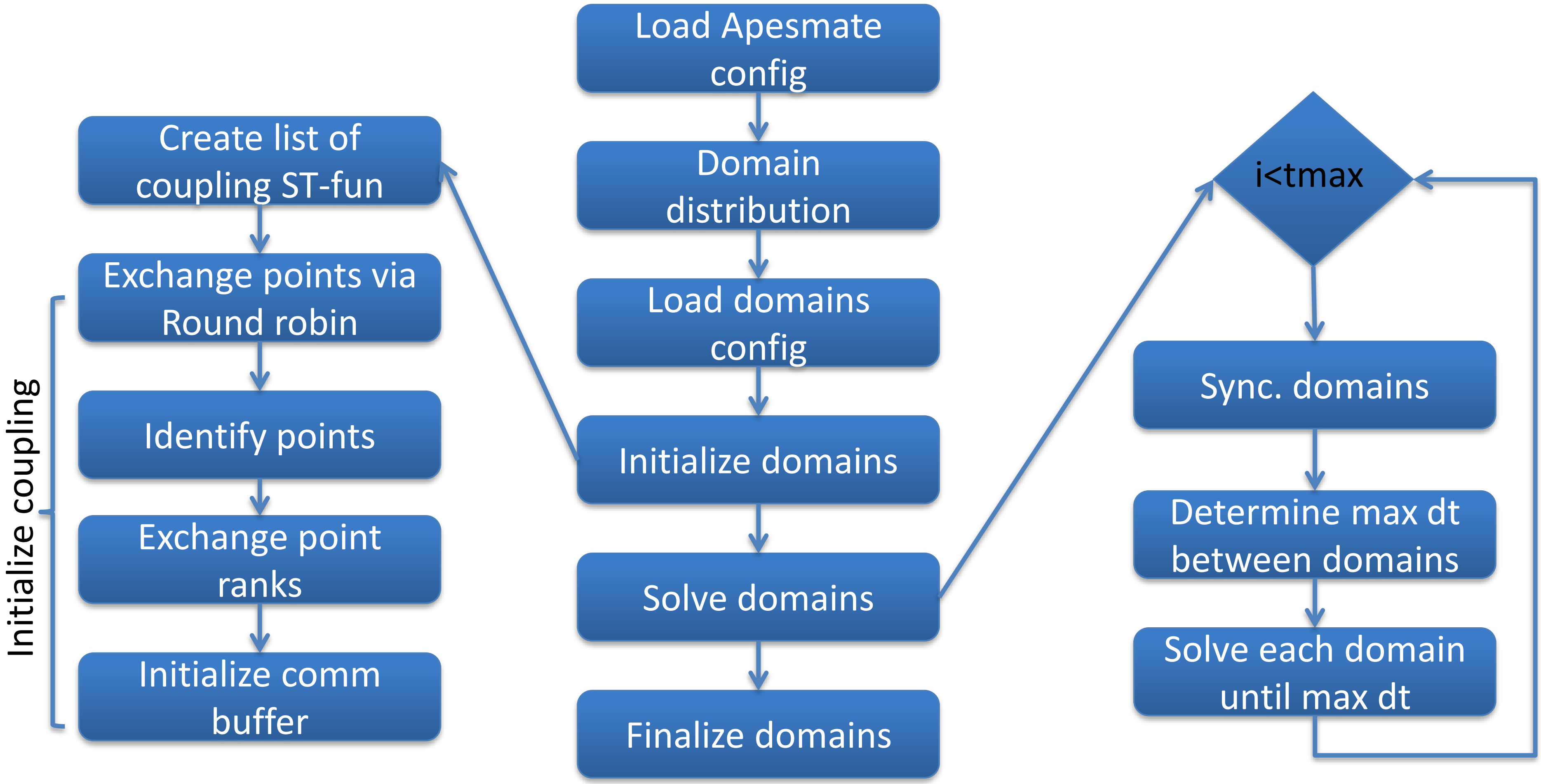
APESmate: Integrated Coupling Environment

- Software to couple APES solvers via TreElm
- Single executable
- Communication between domains is via global MPI communicator
- Communication within a domain is via its local MPI sub-communicator
- Configurable domain distribution with different process counts per domain
- Minimal access to solver routines enables to create dummy modules to build APESmate only with required solvers

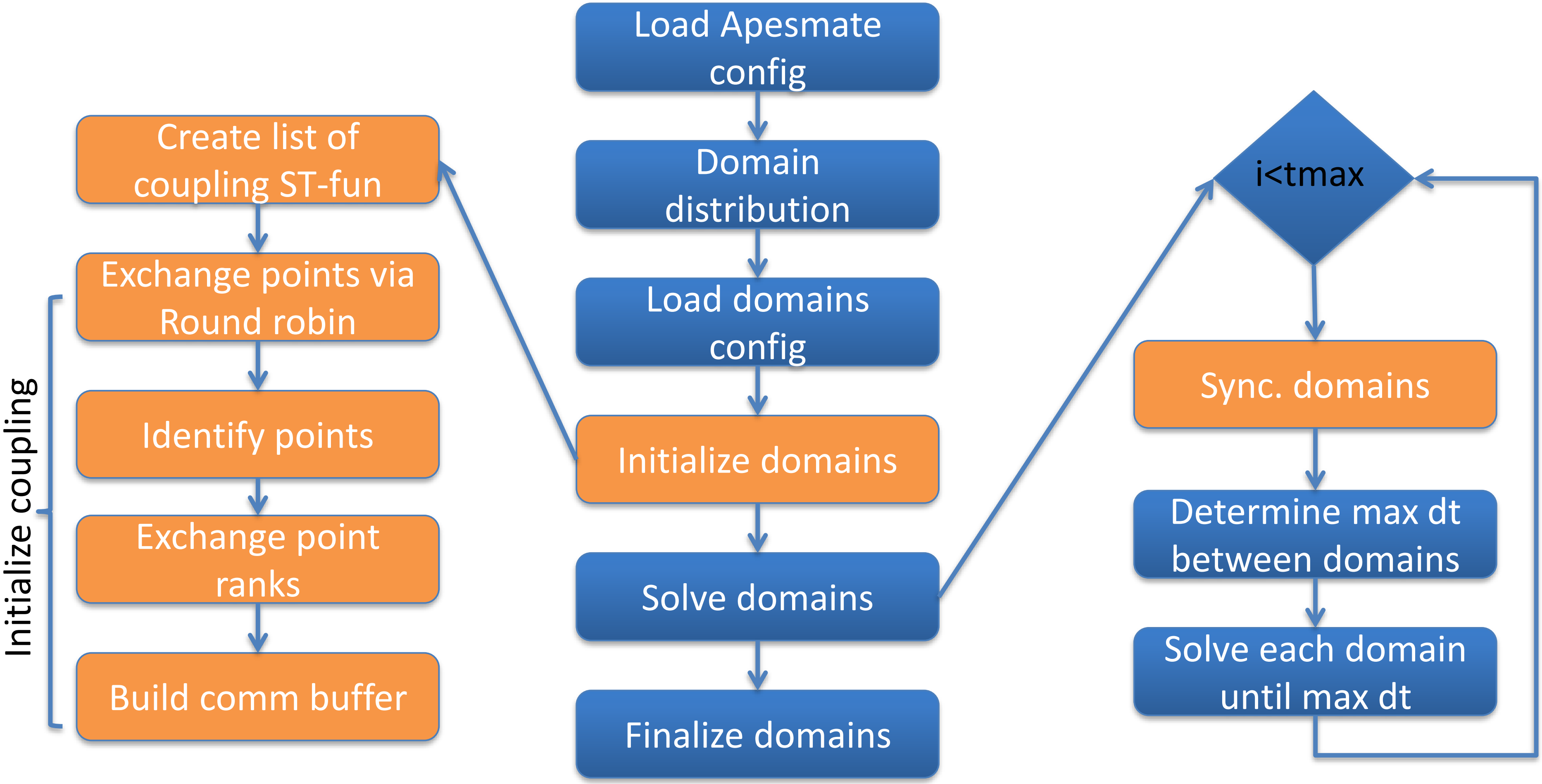
APESmate: Integrated Coupling Environment

- Exchange point values -> Tree independent
- Coupling via space-time function – point data needed to be stored directly in ST-fun for Apesmate to access
  - Surface as boundary condition
  - Volume as source terms
- Easy to configure
- Evaluate point values using derive interface from variable system
- Uses solver data structure to evaluate variable requested by remote domain
- Limitations:
  - Supports only APES solvers
  - Time stepping – domains with different time steps works but without time interpolation

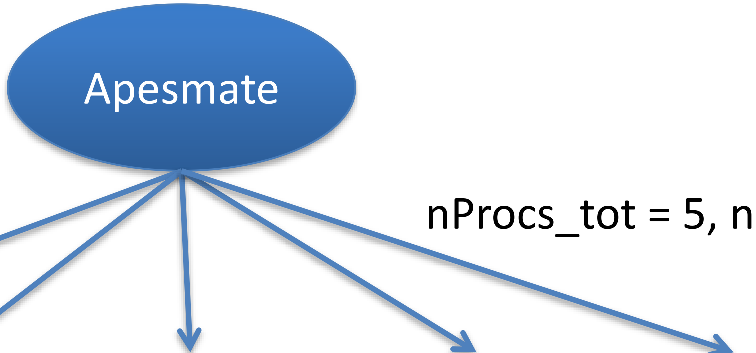
APESmate Algorithm



APESmate Algorithm



Domain Distribution

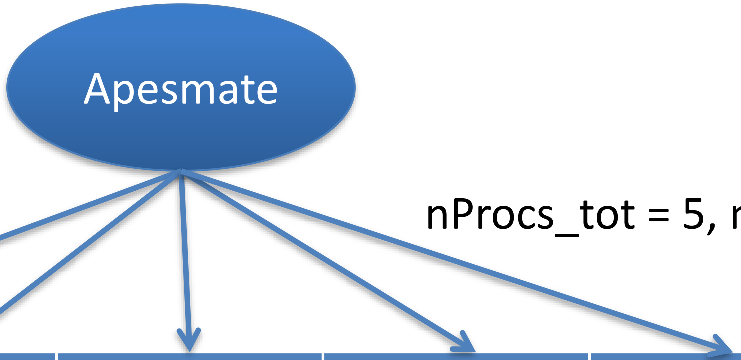


nProcs_tot = 5, nDomains=3

Ranks	P0	P1	P2	P3	P4
Domains	D1	D1			
			D2	D2	
					D3

Nr. Ranks fraction: **D1=2/5, D2=2/5, D3=1/5**

Domain Distribution



nProcs_tot = 5, nDomains=3

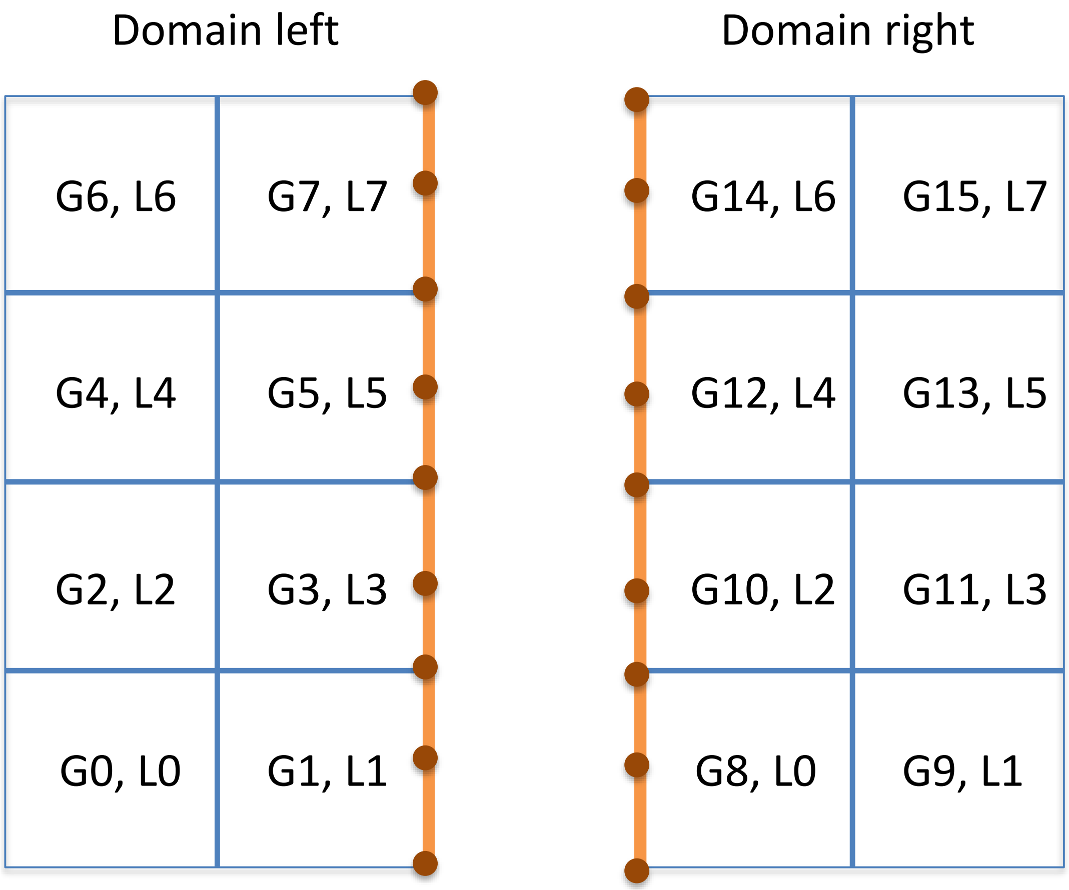
Ranks	**P0**	**P1**	**P2**	**P3**	**P4**
Domains	D1	D1			
		D2	D2	D2	
				D3	D3

Nr. Ranks fraction: **D1=2/5, D2=3/5, D3=2/5**

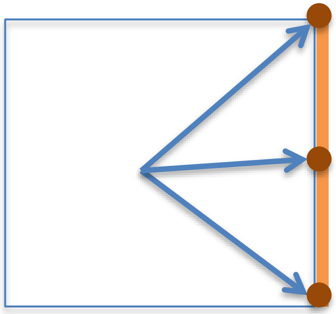
Initialize coupling

- Exchange points and varNames between domains using round robin communication
  - Store requested points and varNames in linked list since we do not know how many request before hand
- Identify points
  - Requires offset_bit for surface coupling
- Check if requested varNames exist in local domain varSys
- Exchange point ranks
- Build point data communication buffer to send point data to correct process
- Exchange point data and varNames by point-point communication
  - Store requested point data and varNames in linked list

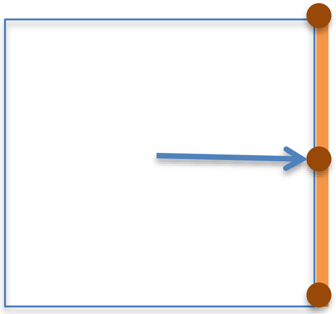
Surface Coupling



Offset bit?



Link based



Face normal

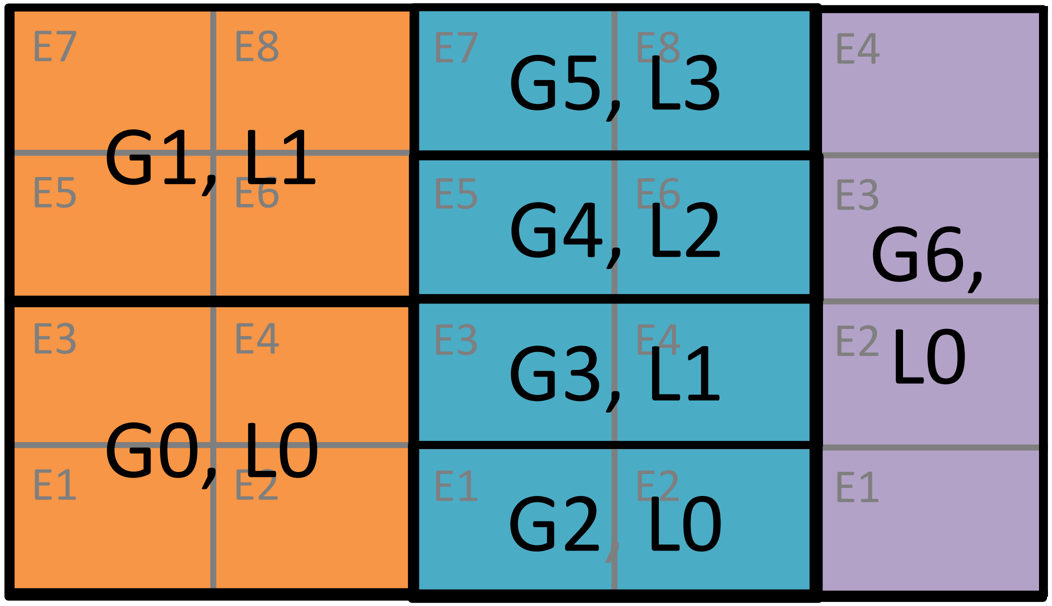
Offset bit

- Required to identify point in local domain
- Points are shifted along offset direction
- Offset direction is converted to character for communication

- Transforming offset direction into character:
  - offset_bit = achar((coord(1)+1) + (coord(2)+1)*4 + (coord(3)+1)*16)

- Backward transformation from character to direction
  - coordX = mod(ichar(offset_bit),4) - 1
  - coordY = mod(ichar(offset_bit),16)/4 - 1
  - coordZ = ichar(offset_bit)/16 - 1

Initialize coupling: 3-Domains



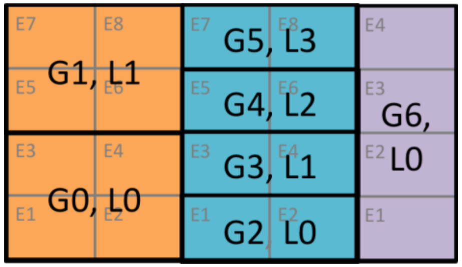
nDomains = 3
nProcs_tot = 6

Dom 1: 2 Proc
4 Elems / proc

Dom 2: 4 Proc
2 Elems / proc

Dom 3: 1 Proc
4 Elems / proc

Initialize coupling: Round Robin Communication



E7 E8 E7 E8 E4

G1, L1

G5, L3

E5 E6 E5 E6 E3

G4, L2

G6,

E3 E4 E3 E4 E2

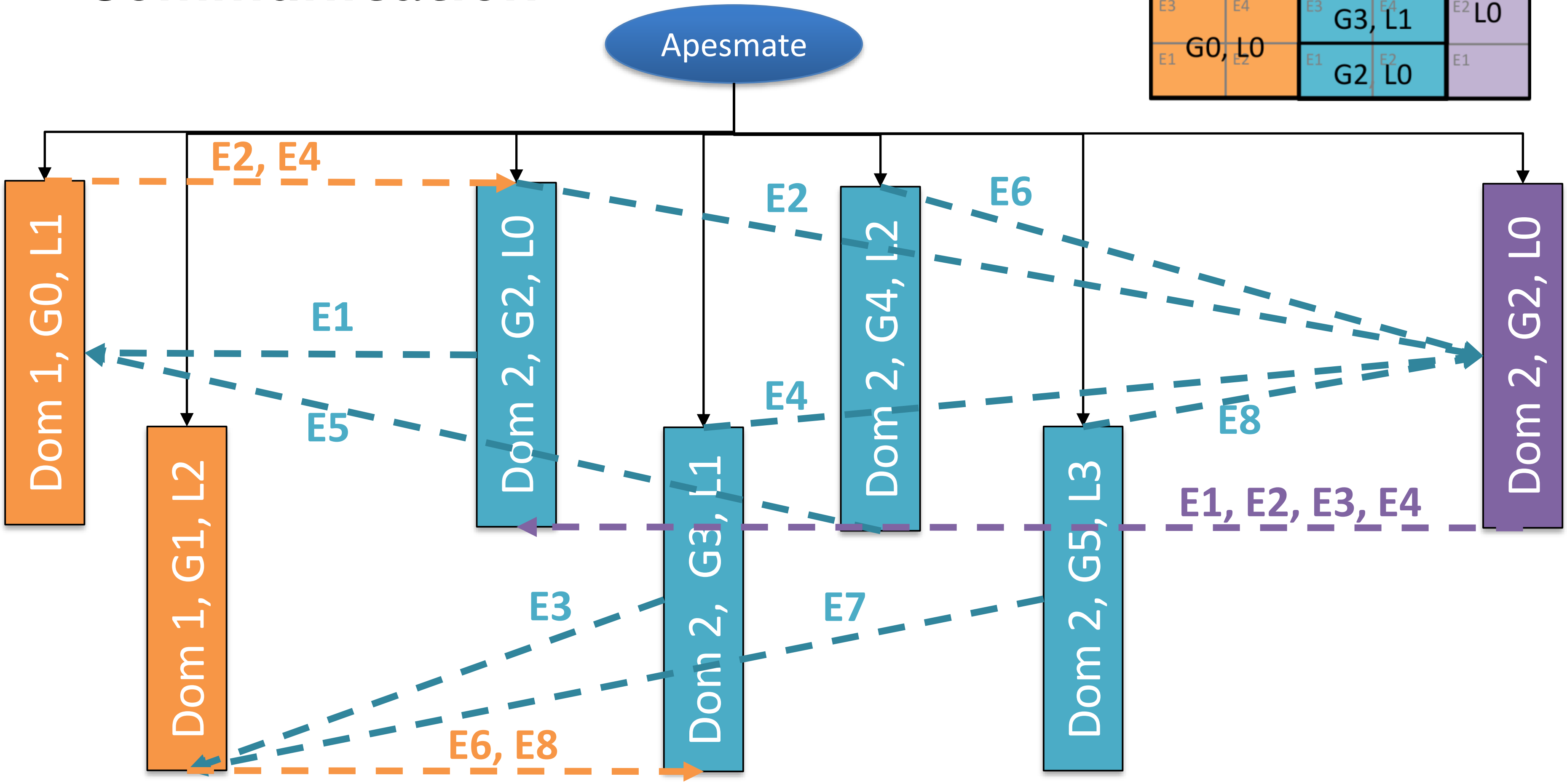
L0

G3, L1

G0, L0

E1 E2 E1 E2 E1

G2, L0



Apesmate

Dom 1, G0, L1

Dom 1, G1, L2

Dom 2, G2, L0

Dom 2, G3, L1

Dom 2, G4, L2

Dom 2, G5, L3

Dom 2, G2, L0

E2, E4

E2

E6

E1

E5

E4

E8

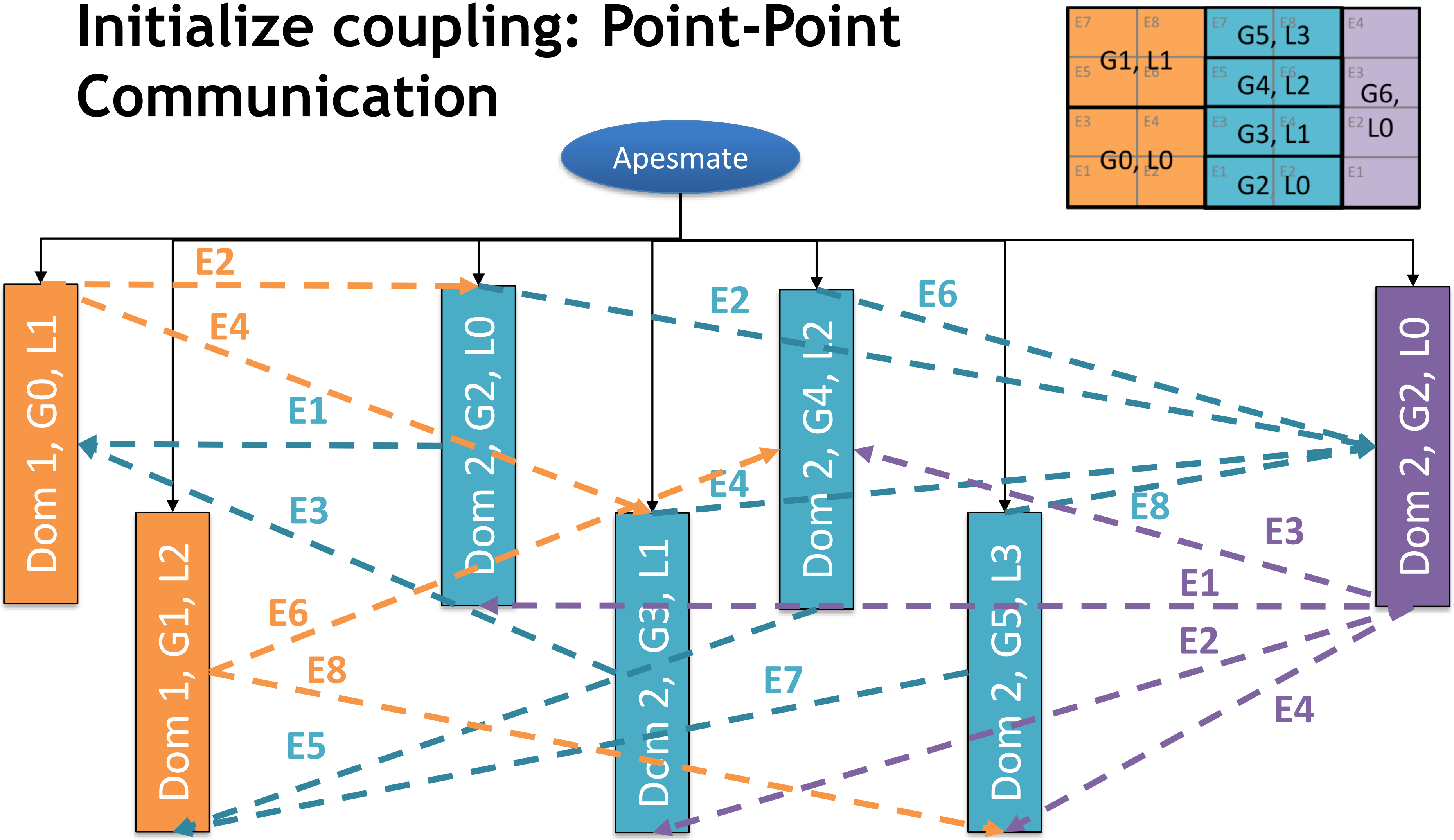
E1, E2, E3, E4

E3

E7

E6, E8

Initialize coupling: Point-Point Communication



Synchronize Domains

- At every synchronize time steps:
  - Evaluate variables on requested points with solver data structure
  - Exchange evaluated values to corresponding domains via global MPI-communicator using point-point communication

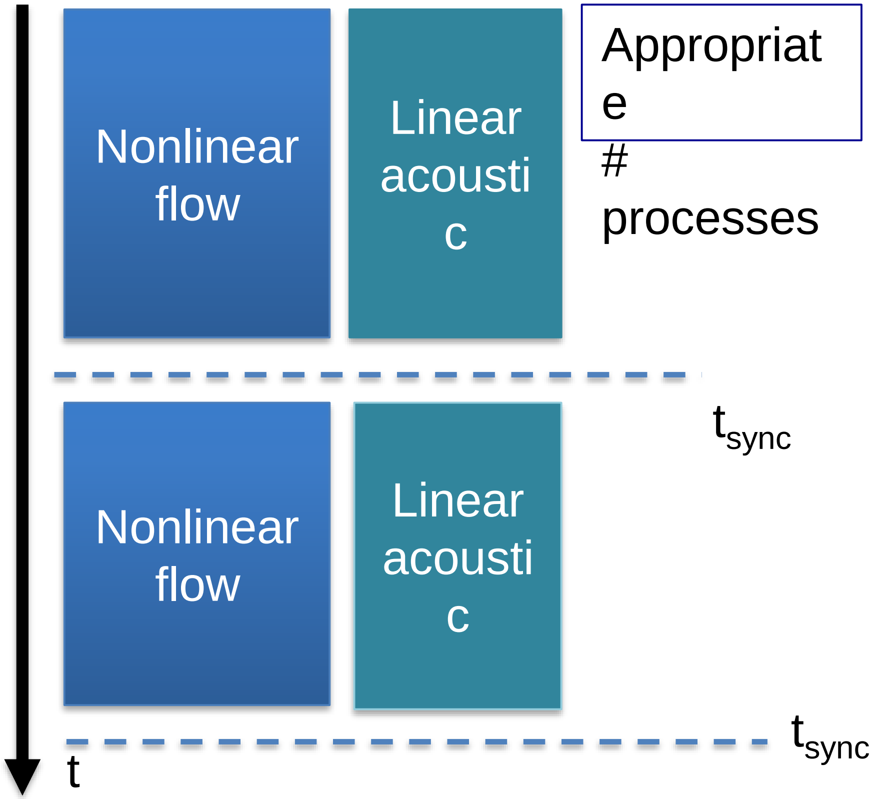
Ideal Load balancing

both participants should spend the same time in computation

→ no idling time

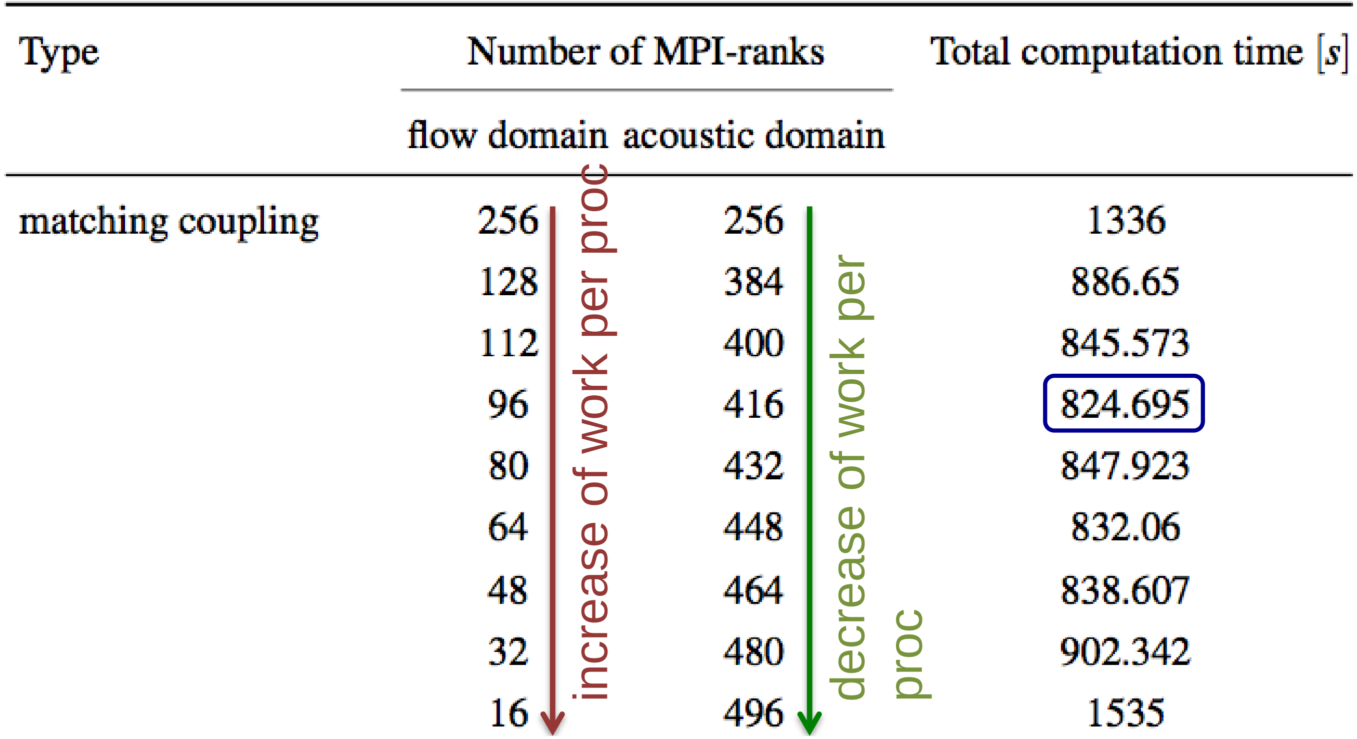
→ only mapping /evaluation + communication

+ Ideal load balancing when both domain use appropriate number of processes
+ Static load balancing is based on heuristics (but highly dependent on order)



Performance of Academic Testcase Coupling Euler - Linearized Euler

- monolithic (euler) 1.014s on 512 MPI-ranks
- matching
  flow: dx=1,8000,O(6)
  acoustic: dx=1,208.000, O(6)
  → save 20% compute time
- non-matching
  flow: dx=1,8000, O(6)
  acoustic: dx=5, 1664, O(12)
  → save 50% compute time



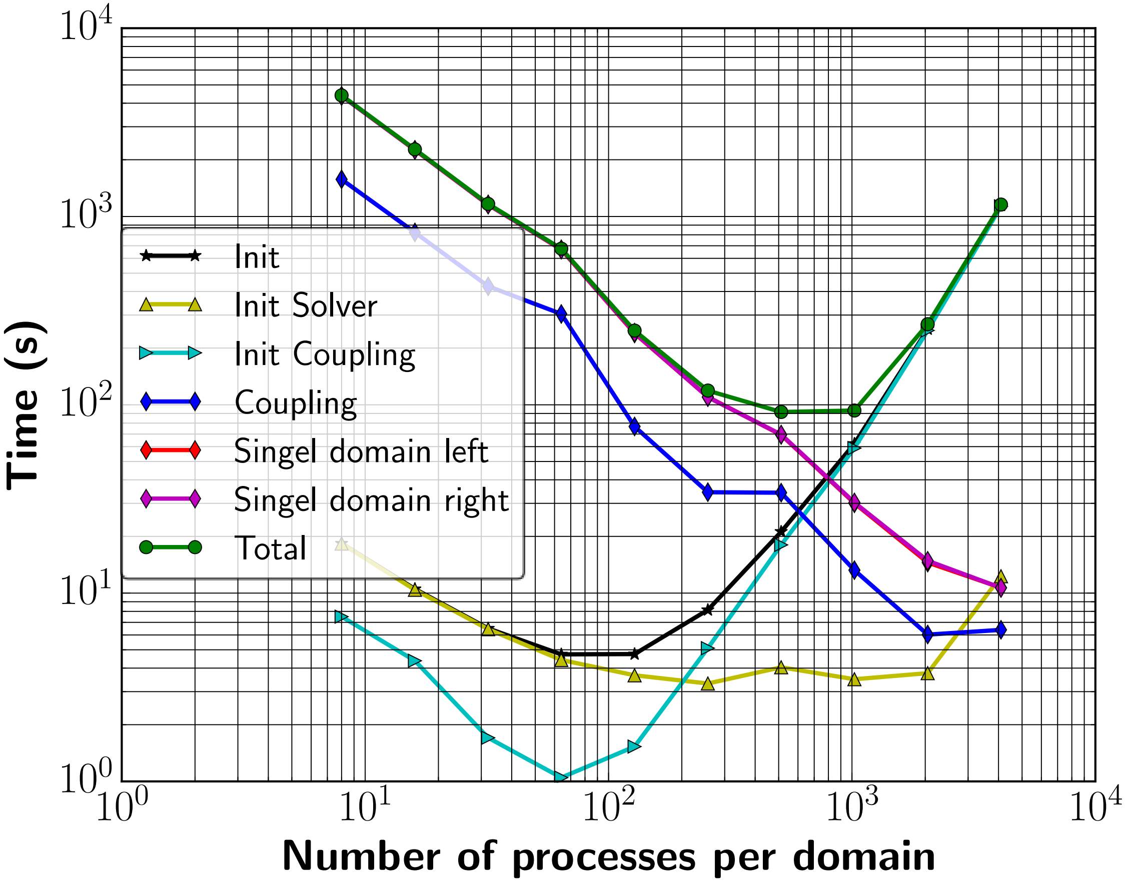
Type	Number of MPI-ranks		Total computation time [s]
	flow domain	acoustic domain	
matching coupling	256	256	1336
	128	384	886.65
	112	400	845.573
	96	416	824.695
	80	432	847.923
	64	448	832.06
	48	464	838.607
	32	480	902.342
	16	496	1535

increase of work per proc

decrease of work per proc

Strong Scaling: SuperMUC

Euler-Euler coupling
Order: O(20)
nElems: 4096 / domain

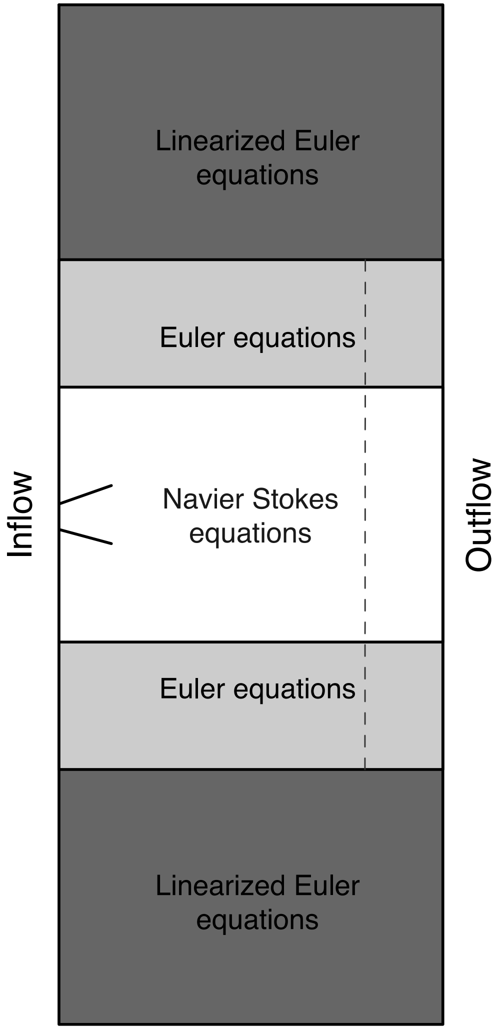


Subsonic Jet in 2D

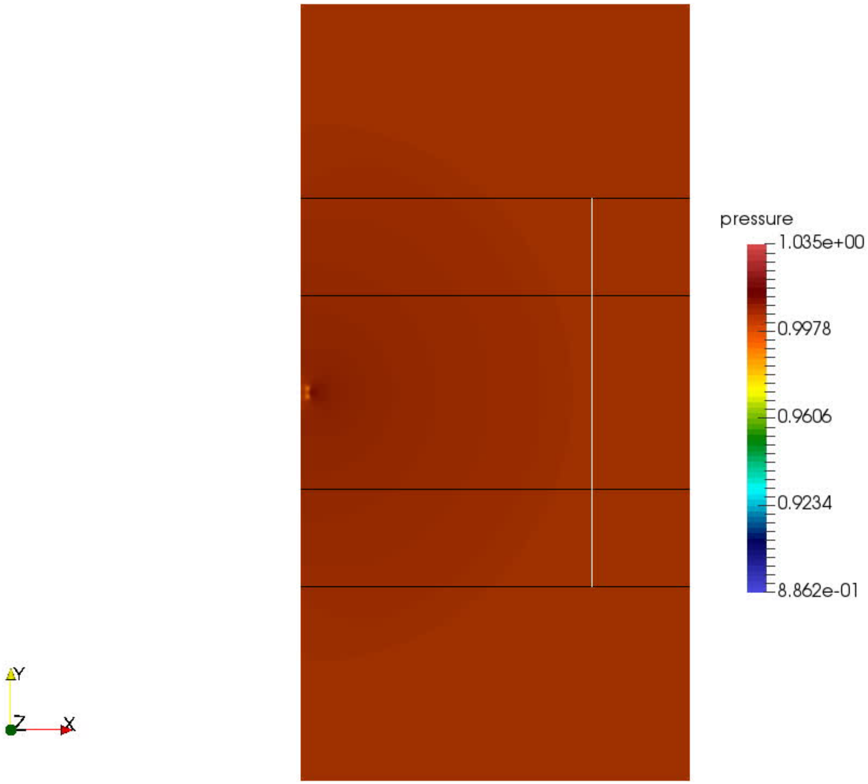
- Inflow condition: expanded jet

- Re = 23 (mu = 1e-6, d=1e-4, v= 0.23)
- Ma = 0.2
- Ramping of initial shock from over first 10s

- Sponge zone in Navier and Euler domain

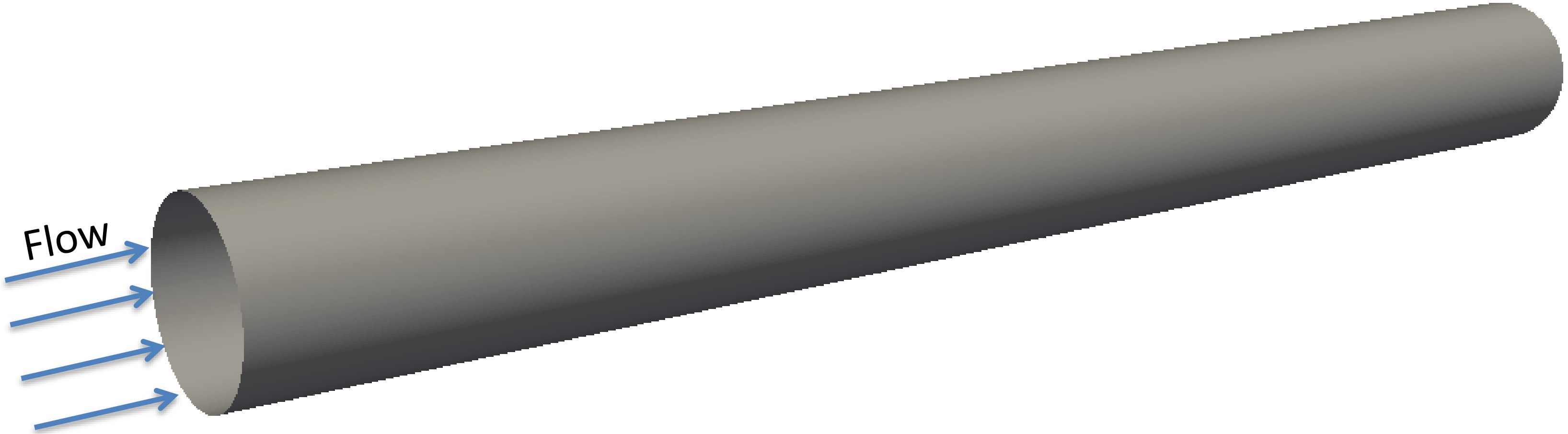


Coupled Simulation for 2D Jet

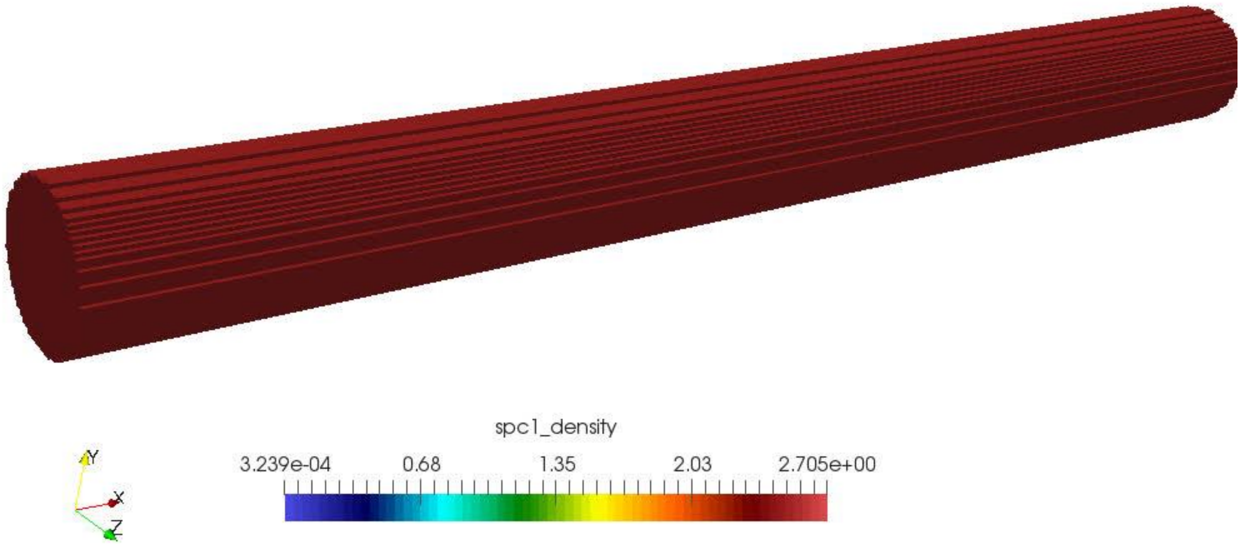


Pipe flow in 3D

- Incompressible flow (D3Q19) + Passive scalar species transport (D3Q7)
- 1-way coupling - Species driven by velocity from flow field



Coupled Simulation for 3D Species Transport



Conclusion and Current Work

- APESmate implementation is working
- Tested with simple test cases for both surface and volume coupling
- Currently work:
  - Production test cases like electrodialysis process, 3-field acoustic and 2D jet are under testing phase
  - Scaling analysis
- Future work: Run production test cases

Thanks for your attention ☺

Questions?