Northwestern

Inter-process Communication & Coupling between Abaqus Solvers

Evanston, IL

10 May 2021

Hao Yin

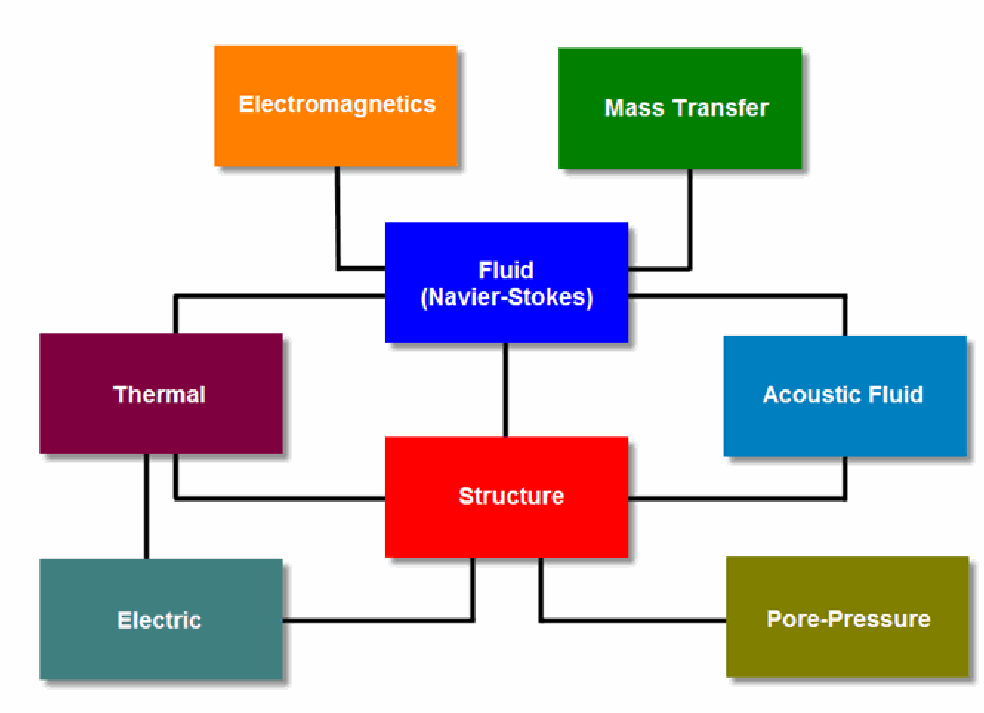
Outline

- Motivations
- Introduction to Inter-process Communication
- Coupling between Abaqus Solvers
- Application – the Multiphysics-LDPM Framework

Motivations

Multiphysics problems

- Multiphysics is defined as the coupled processes or systems involving more than one simultaneously occurring physical fields and the studies of and knowledge about these processes and systems. ------ Wikipedia



Picture source: Adina Multiphysics Homepage

Motivations

Solving Multiphysics problems

- For example, for a thermo-elasticity problem

$$\left.\begin{aligned} \rho\ddot{\mathbf{u}} &= \mathrm{div}\boldsymbol{\sigma}(\mathbf{u},T) + \rho\mathbf{b} \\ \rho c\dot{T} &= -\nabla\cdot\mathbf{q}(\mathbf{u},T) + s \end{aligned}\right\} \quad \text{in } \Omega \times [0,t]$$

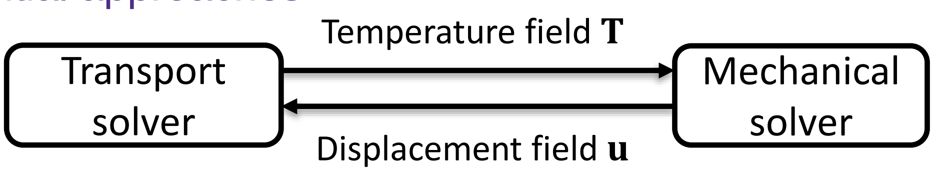
$$\mathbf{u} = \mathbf{u}_g \quad \text{on} \quad \Gamma_{ug} \times [0,t] \qquad T = T_g \quad \text{on} \quad \Gamma_{Tg} \times [0,t]$$

$$\boldsymbol{\sigma}\cdot\mathbf{n} = \mathbf{t}_h \quad \text{on} \quad \Gamma_{uh} \times [0,t] \qquad \mathbf{q}\cdot\mathbf{n} = \mathbf{j}_h \quad \text{on} \quad \Gamma_{Th} \times [0,t]$$

  - Fully-coupled approaches

$$\begin{bmatrix} \mathbf{M}_{11} & \mathbf{M}_{12} \\ \mathbf{M}_{21} & \mathbf{M}_{22} \end{bmatrix} \begin{Bmatrix} \ddot{\mathbf{U}} \\ \ddot{\mathbf{T}} \end{Bmatrix} + \begin{bmatrix} \mathbf{C}_{11} & \mathbf{C}_{12} \\ \mathbf{C}_{21} & \mathbf{C}_{22} \end{bmatrix} \begin{Bmatrix} \dot{\mathbf{U}} \\ \dot{\mathbf{T}} \end{Bmatrix} + \begin{bmatrix} \mathbf{K}_{11} & \mathbf{K}_{12} \\ \mathbf{K}_{21} & \mathbf{K}_{22} \end{bmatrix} \begin{Bmatrix} \mathbf{U} \\ \mathbf{T} \end{Bmatrix} = \begin{Bmatrix} \mathbf{F} \\ \mathbf{Q} \end{Bmatrix}$$

  - Sequential approaches

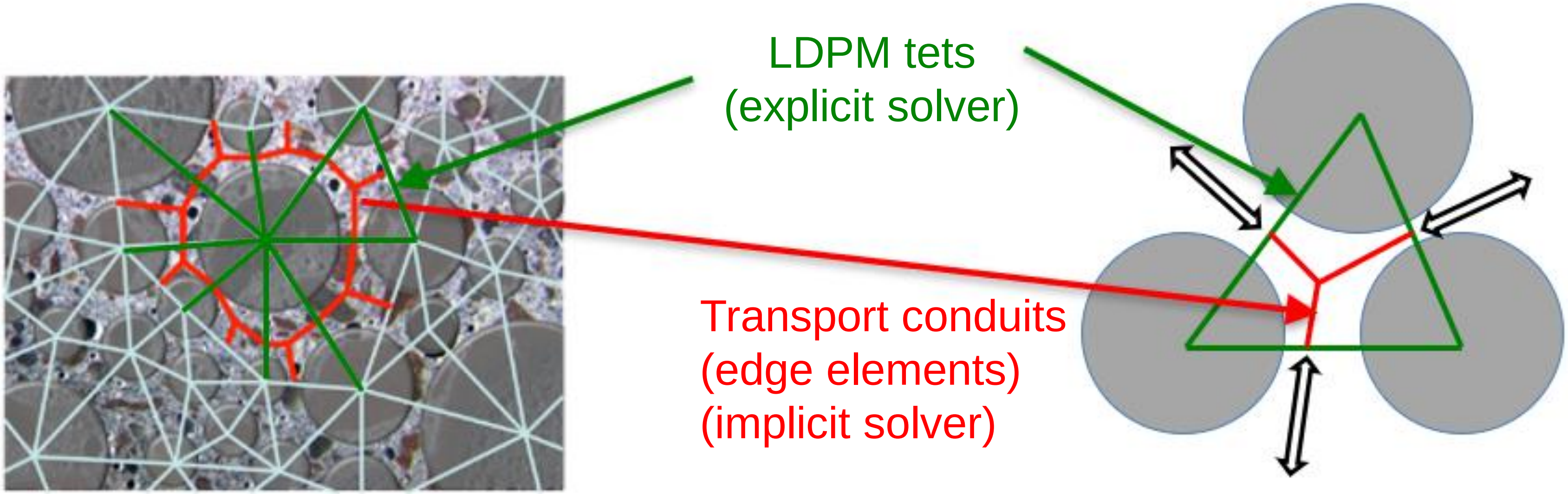


- In Abaqus, we have following built-in procedures to solve Multiphysics problems:
  - Built-in coupled elements (fully-coupled)
  - SIMULIA co-simulation engine (sequential)

However…

Motivations

Dual lattice systems in Multiphysics-LDPM



Different meshes for coupled physical fields and mechanical fields!

- Fully-coupled approaches ✖
- Sequential approaches

spatial mapping?
temporal mapping?

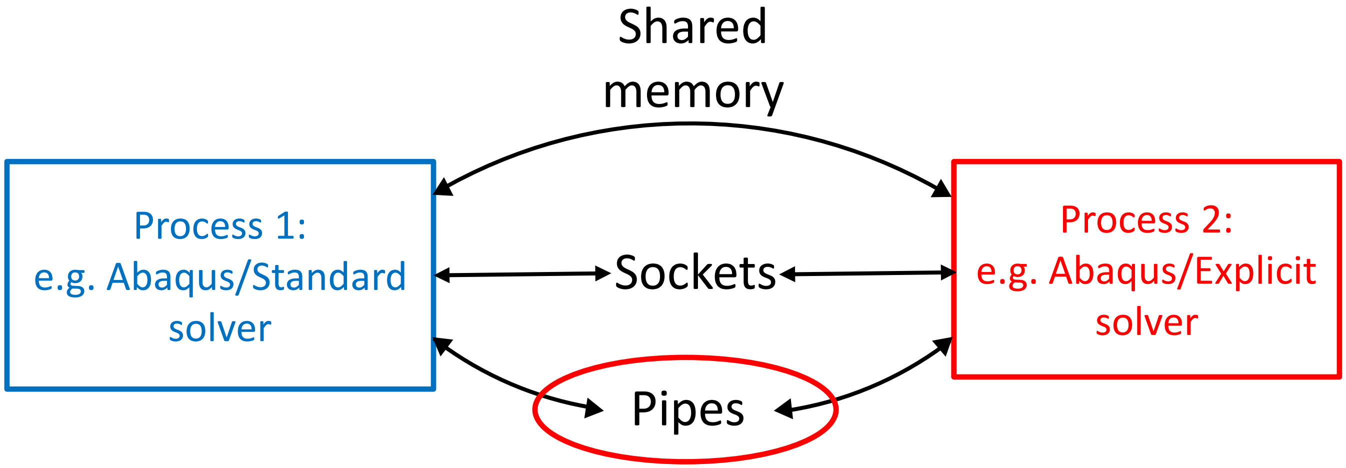
This type of problems is also called “multidomain” or “multimodel” coupling.

Outline

- Motivations
- Introduction to Inter-process Communication
- Coupling between Abaqus Solvers
- Application – the Multiphysics-LDPM Framework

Inter-process Communication

- Inter-process communication (IPC) refers to the coordination of activities among cooperating processes. This communication could involve a process letting another process know that some event has occurred or the transferring of data from one process to another.
- For our applications of IPC in solving Multiphysics problems, the processes are different simulation solvers (e.g. Abaqus/Ansys/in-house codes/other solvers).



Inter-process Communication

Pipes

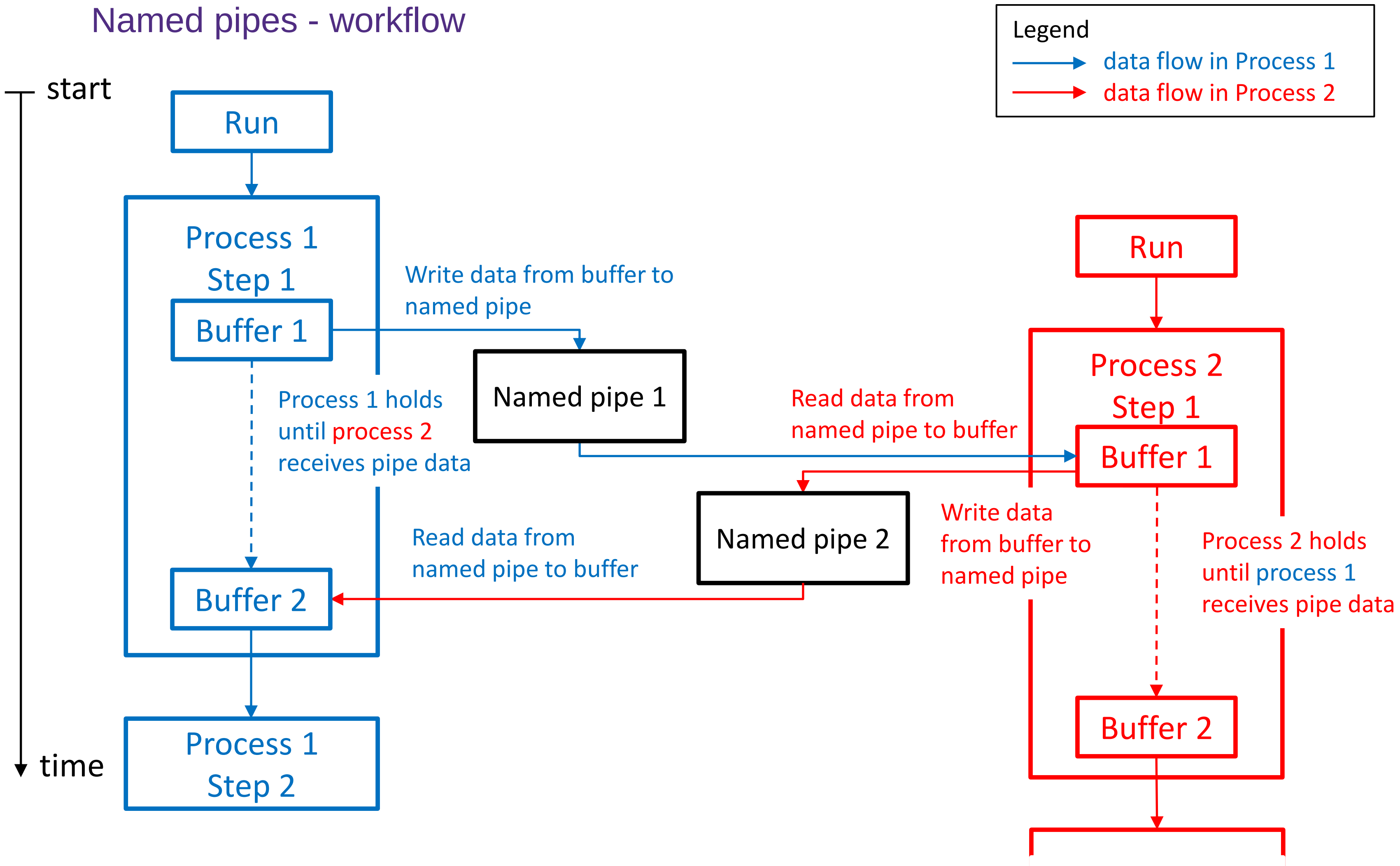
- Using pipe is a simple synchronized way of passing information between two processes. A pipe can be viewed as a special file that can store only a limited amount of data and uses a FIFO access scheme to retrieve data. In a logical view of a pipe, data is written to one end and read from the other.

- Pipes come in two varieties:
  - **Unnamed**. Unnamed pipes can only be used by related processes (i.e. a process and one of its child processes, or two of its children). Unnamed pipes cease to exist after the processes are done using them.
  - **Named**. Named pipes exist as directory entries, complete with permissions. This means that they are persistent and that unrelated processes can use them.

- Pipes can be used on both Unix and Windows OS platforms, but Windows version need some special treatments.

Inter-process Communication

Named pipes - workflow



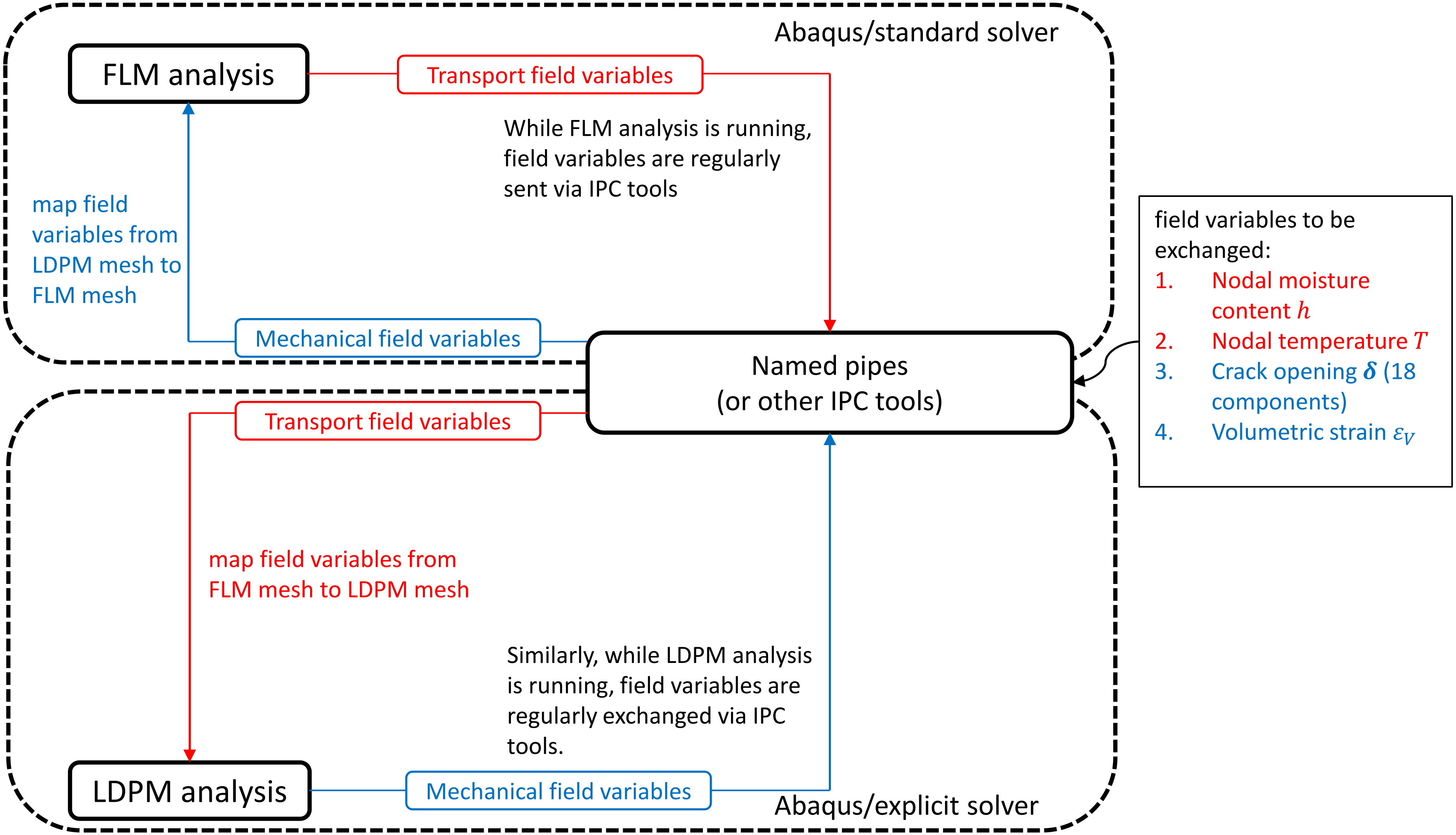
Outline

- Motivations
- Introduction to Inter-process Communication
- Coupling between Abaqus Solvers
- Application – the Multiphysics-LDPM Framework

Coupling between Abaqus Solvers

Legend

- data from FLM to LDPM (red line)
- data from LDPM to FLM (blue line)



Coupling between Abaqus Solvers

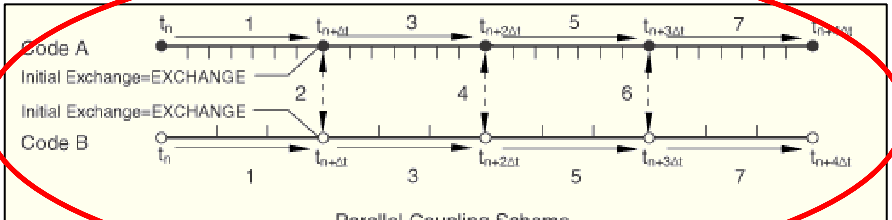
Coupling Scheme

- Coupling Scheme
  - Parallel explicit coupling scheme (Jacobi)

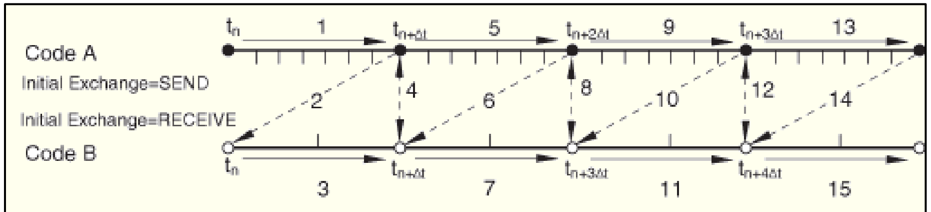
    In a parallel explicit coupling scheme, both simulations are executed concurrently, exchanging fields to update the respective solutions at the next target time. - more efficient use of computing resources; less stable than the sequential scheme

  - Sequential explicit coupling scheme (Gauss-Seidel)

    In a sequential explicit coupling scheme, the simulations are executed in sequential order. One analysis leads while the other analysis lags the co-simulation.



Parallel Coupling Scheme



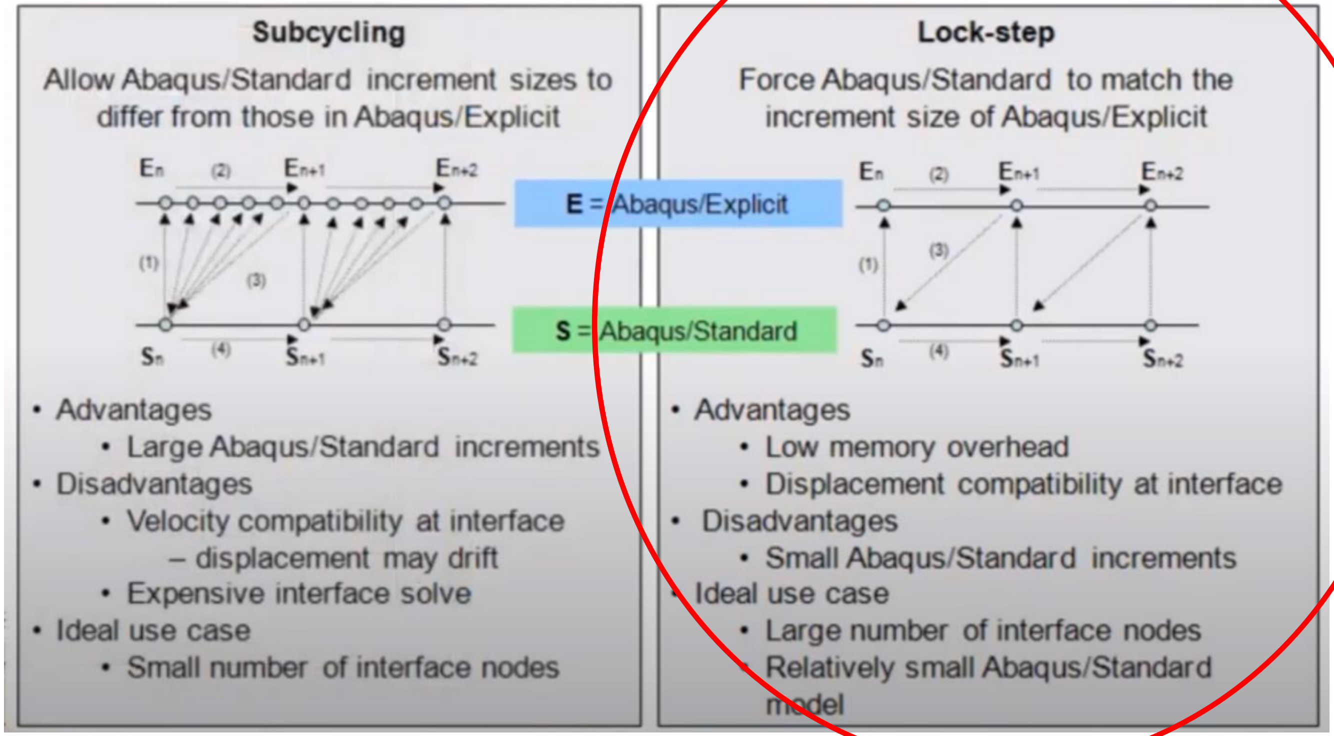
Serial Coupling Scheme

Picture source: [Abaqus Analysis User's Manual - Co-simulation using MpCCI]

Coupling between Abaqus Solvers

Time Incrementation Scheme

- Time Incrementation Scheme



Picture source: [Abaqus Standard & Abaqus Explicit Co-Simulation | SIMULIA How-To Tutorial]

Coupling between Abaqus Solvers

Named pipes – pseudocodes

```
# pseudocode for named pipes in process 1
Subroutine VUEL(...)

# Declare variables
real(dp), dimension(n,m) :: FLMdata
real(dp), dimension(nn,mm) :: LDPMdata
integer,parameter :: V2U = 100
integer,parameter :: U2V = 101 # unit numbers
for named pipes
⋮

# Send data to named pipes
open(unit=V2U,file='pipe_name_for_V2U.pipe',
action='write')
write(unit=V2U,format='any_format') LDPMdata
close(unit=V2U)
# Retrieve data from named pipes
open(unit=U2V,file='pipe_name_for_U2V.pipe',
action='read')
read(unit=U2V,format='any_format') FLMdata
close(unit=U2V)

# Physical model here
LDPMdata = umat_LDPM(...,FLMdata)

End subroutine VUEL
```

```
# pseudocode for named pipes in process 2

⋮

# retrieve data from named pipes
open(unit=V2U,file='pipe_name_for_V2U.pipe',
action='read')
read(unit=V2U,format='any_format') LDPMdata
close(unit=V2U)
# send data to named pipes
open(unit=U2V,file='pipe_name_for_U2V.pipe',
action='write')
read(unit=U2V,format='any_format') FLMdata
close(unit=U2V)

# Physical model here
FLMdata = HTCmodel(..., LDPMdata)

End subroutine VUEL
```

Coupling between Abaqus Solvers

Named pipes on Northwestern Quest

- Batch job is somehow not an ideal choice to submit jobs involving pipes (need to check).
  - If submitting multiple Abaqus jobs using one single batch file, then only the first Abaqus job will be executed.
  - If submitting jobs using multiple separate batch files, then each job will be assigned to a different Quest node. Communication between Quest nodes is possible but difficult – Firewall/network protection applies.

- Use interactive job instead.
  - Use the command: `srun --account=p12345 --partition=short -N 1 -n 4 --mem=12G --time=01:00:00 --pty bash –l` to run an interactive bash session on a single compute node with four cores, and access to 12GB of RAM for up to an hour, debited to the p12345 account.

  - In bash command line session, run the following commands:
  - `cd /your/abaqus/working/directory on Quest`
  - `module load abaqus/2020`
  - `mkfifo FLM2LDPM.pipe`
  - `mkfifo LDPM2FLM.pipe`
  - `abaqus job=FLMjobname input=FLMjobname.inp USER=UEL_FLM.for ask_delete=OFF`
  - `abaqus job=LDPMjobname input=LDPMjobname.inp USER=VUEL_LDPMM.for double=both ask_delete=OFF`

Coupling between Abaqus Solvers

A few tips about coupling via IPC

- Format
  - when writing and reading with named pipes or sockets, the format should be explicitly declared and consistent in both processes.

In process 1:

```
write(unit=V2U,format='(F20.8)') LDPMdata
```

In process 2:

```
read(unit=V2U,format='(I8)') LDPMdata
```

✖

In process 1:

```
write(unit=V2U,format=*) LDPMdata
```

In process 2:

```
read(unit=V2U,format=*) LDPMdata
```

✖

In process 1:

```
write(unit=V2U,format='(F20.8)') LDPMdata
```

In process 2:

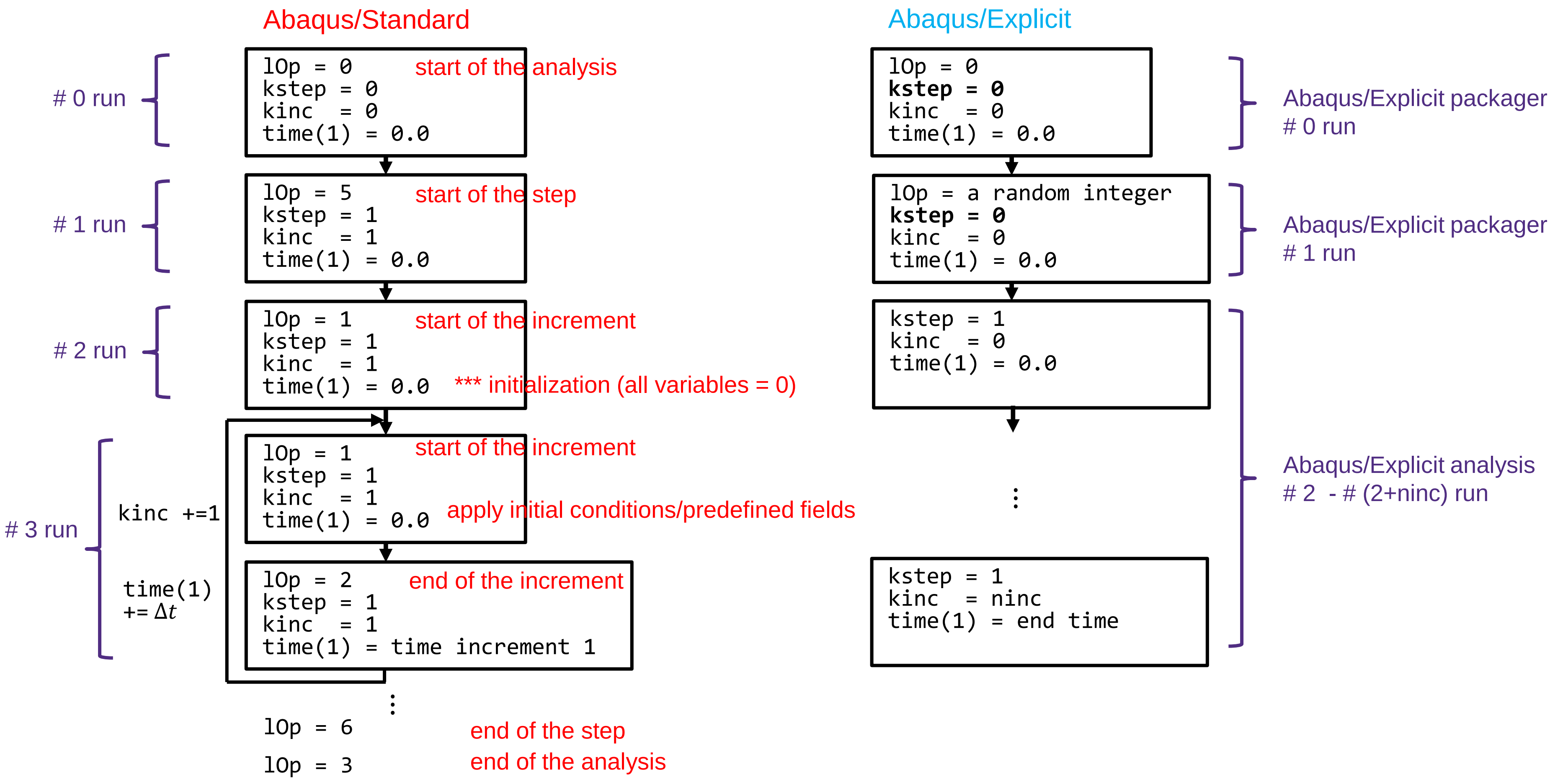
```
read(unit=V2U,format='(F20.8)') LDPMdata
```

√

Coupling between Abaqus Solvers

A few tips about coupling via IPC

- stage, step and increment flags in Abaqus

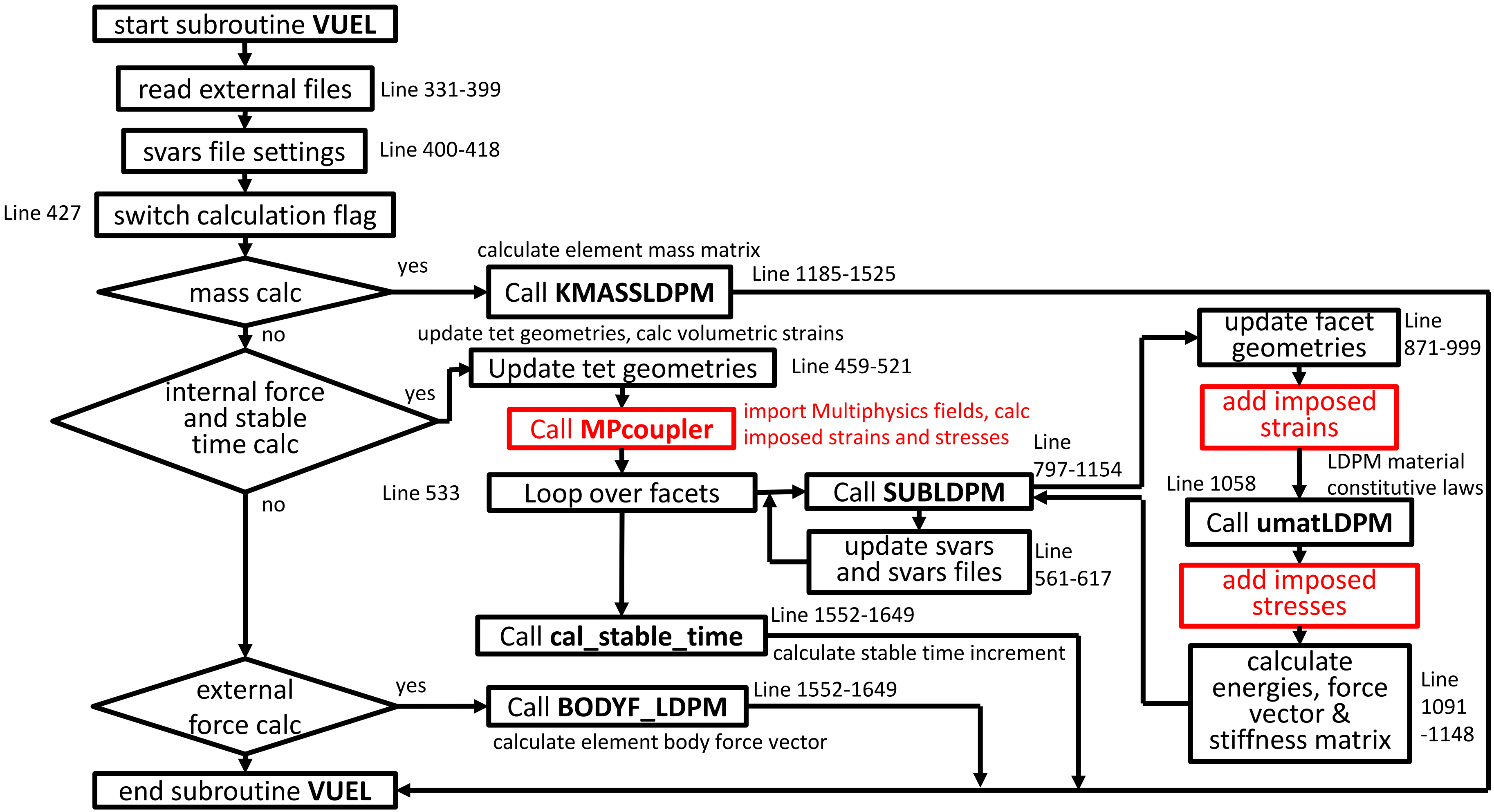


Outline

- Motivations
- Introduction to Inter-process Communication
- Coupling between Abaqus Solvers
- Application – the Multiphysics-LDPM Framework

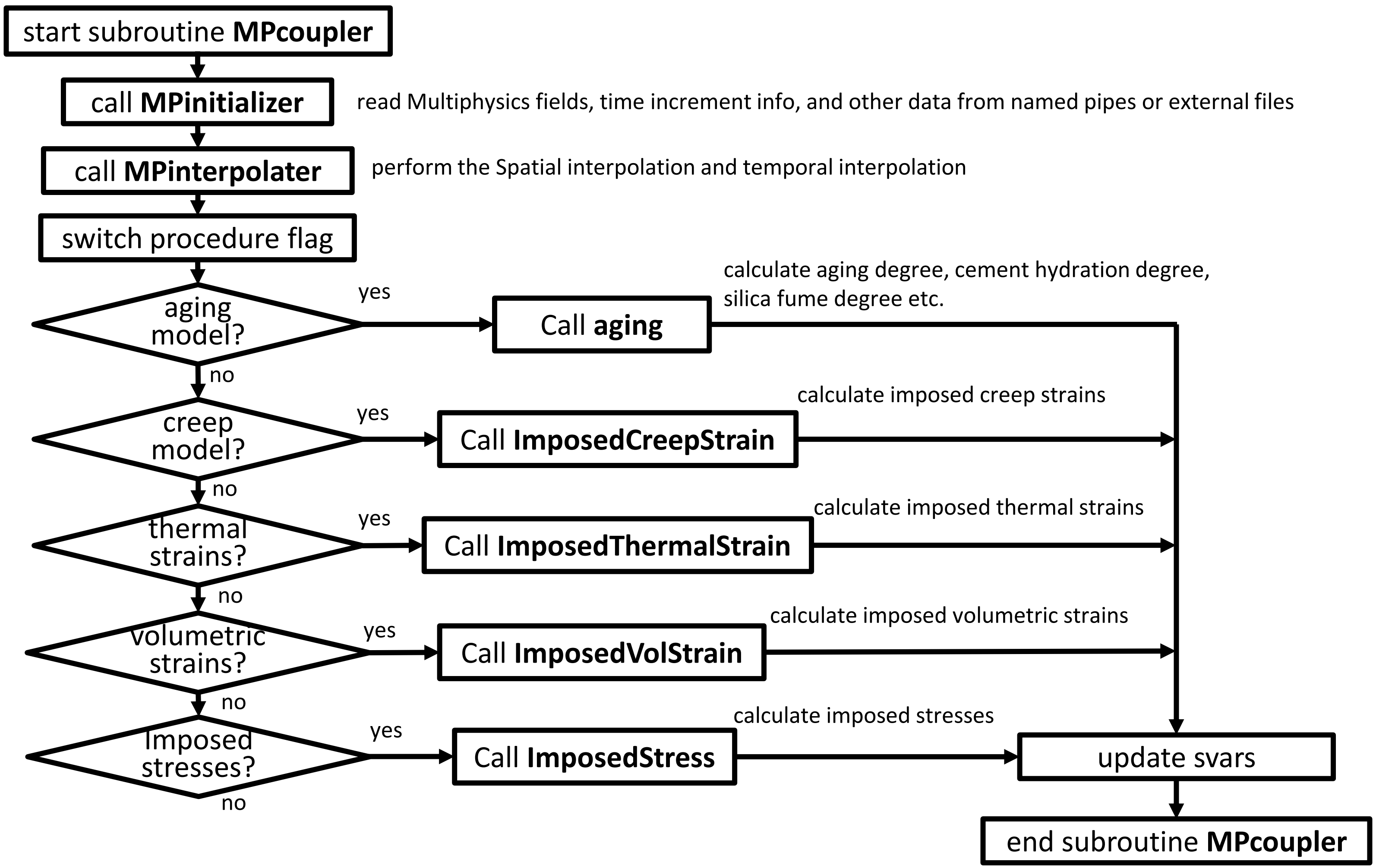
Application – the Multiphysics-LDPM Framework

In Abaqus user subroutine code VUEL_LDPM.for (Mar 26, 2021 version)



Application – the Multiphysics-LDPM Framework

In subroutine MPcoupler



Application – the Multiphysics-LDPM Framework

In subroutine MPcoupler

Input: Multiphysics fields, Multiphysics geometries (e.g. spatial interpolation dictionary), Multiphysics time increment info, analysis procedure flags, and other info from basic LDPM

Intermediate output: Interpolated Multiphysics fields `MPfields`

field θ_1 rate $\dot{\theta}_1$ field θ_2 rate $\dot{\theta}_2$ ... field θ_n rate $\dot{\theta}_n$

`MPfields =` facet 1, facet 2, ⋮, facet 12 $\begin{bmatrix} \\ \\ \\ \end{bmatrix}$

output: Multiphysics state variables `stvMP`

stvMP 1 stvMP 2 ... stvMP n

`stvMP =` facet 1, facet 2, ⋮, facet 12 $\begin{bmatrix} \\ \\ \\ \end{bmatrix}$

then in subroutine VUEL , update:

```
svars(21:20+nstvMP) = stvMP(1:nstvMP)
```

Application – the Multiphysics-LDPM Framework

Poroelasticity problem, radial expansion in a thick-walled cylinder due to pore pressure (Grassl et al. JMPS 2015).

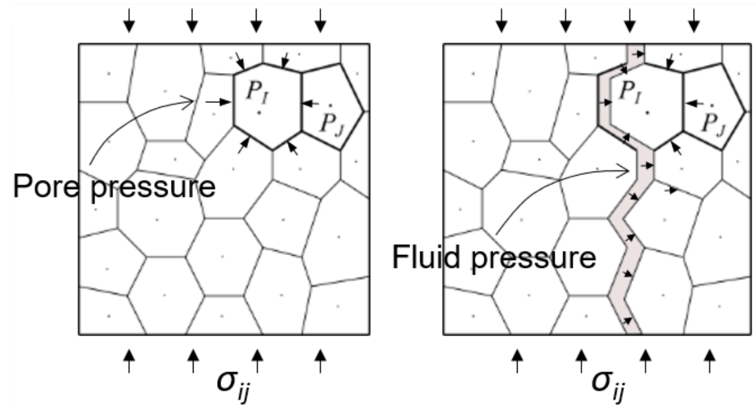
Effective facet normal stress

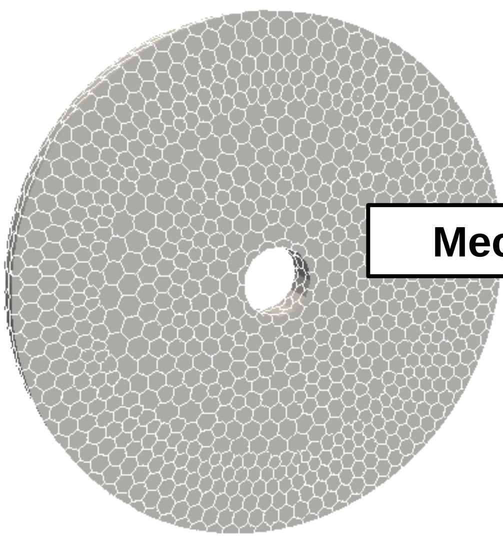
$$\bar{\sigma}_N = \sigma_N + \sigma_N^p = f(\varepsilon_N, \varepsilon_T, \dots)$$

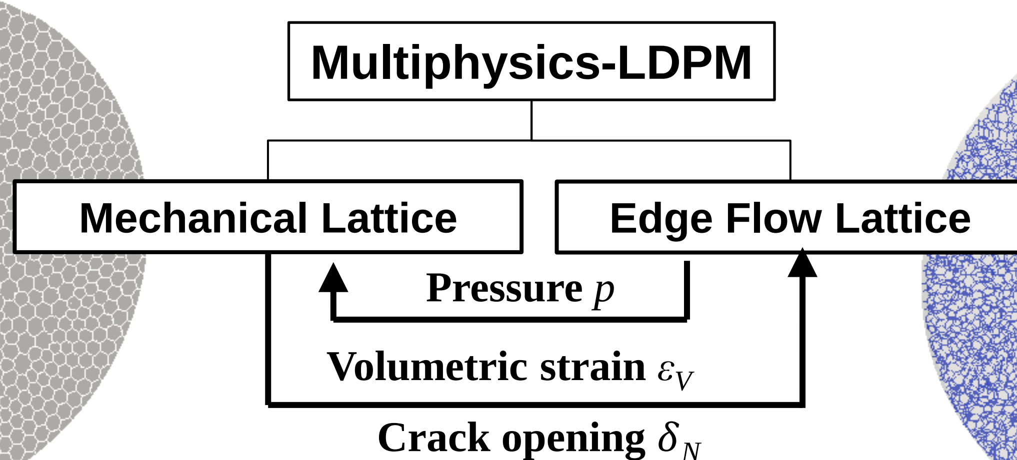
$$\sigma_N^p = -bp$$

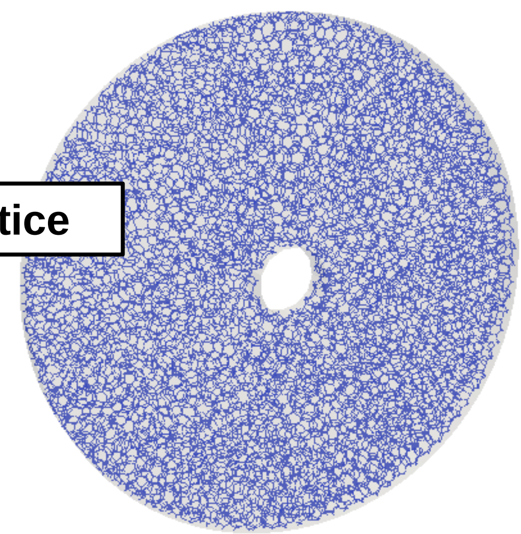
Imposed stress

Biot's coefficient









Application – the Multiphysics-LDPM Framework

Poroelasticity problem, radial expansion in a thick-walled cylinder due to pore pressure.

Analytical solution:

$$\bar{u} = -b\bar{P}_{\mathrm{fi}}\frac{1-\nu^2}{2}\left[\frac{\bar{r}_o^2}{\bar{r}_o^2-1}\left(\frac{1+\nu}{1-\nu}\frac{1}{\bar{r}}+\bar{r}\right)+\bar{r}\frac{\frac{1}{1+\nu}-\ln\bar{r}}{\ln\bar{r}_o}\right]-(1-b)\bar{P}_{\mathrm{fi}}\frac{\bar{r}_o^2}{\bar{r}_o^2-1}\left(\frac{1+\nu}{\bar{r}}+\frac{\bar{r}(1-\nu)}{\bar{r}_o^2}\right)$$

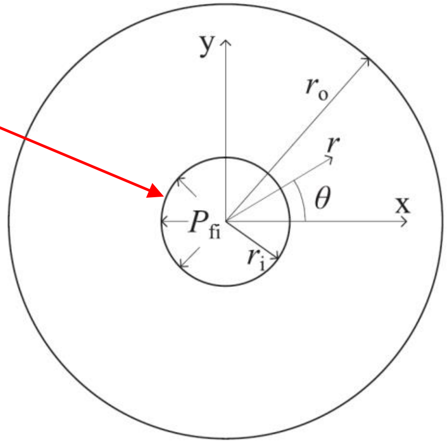
where:

$$\bar{u} = \frac{u}{r_i} \qquad \bar{r} = \frac{r}{r_i} \qquad \bar{r}_o = \frac{r_o}{r_i}$$

$$\bar{P}_f = \frac{P_f}{E_c} \qquad \bar{P}_{fi} = \frac{P_{fi}}{E_c}$$

$$E_c = \frac{2+3\alpha}{4+\alpha}E_0 \qquad \nu = \frac{1-\alpha}{4+\alpha}$$

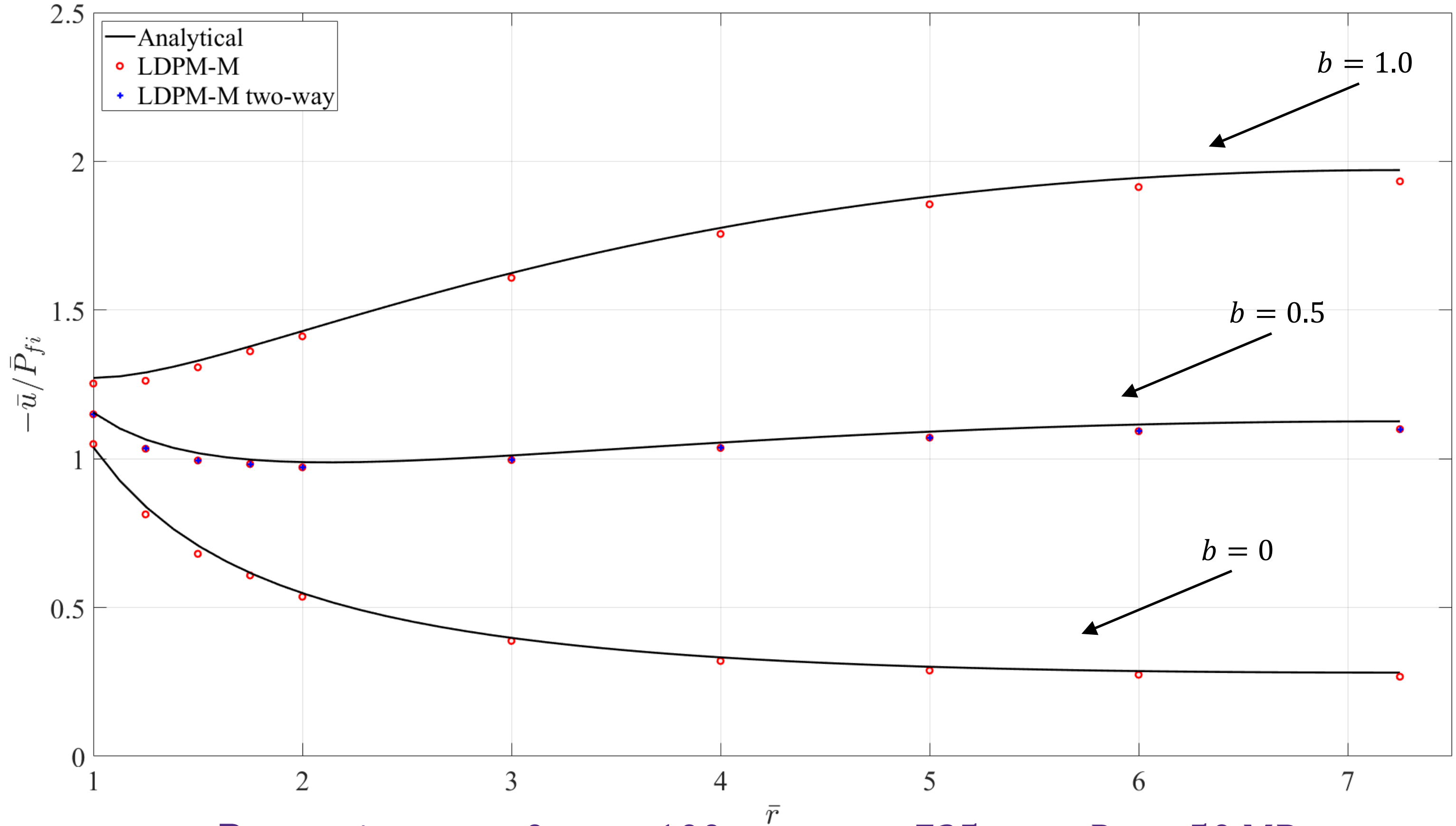
Note: the total stress on the boundary in mechanical analysis is equal to the fluid pressure on the boundary in diffusion analysis.



E_c - Macroscopic Young's modulus, ν – Macroscopic Poisson's ratio
E_0 - Mesoscopic Young's modulus, α – Shear-normal coupling coefficient

Application – the Multiphysics-LDPM Framework

Poroelasticity problem, radial expansion in a thick-walled cylinder due to pore pressure.



Parameters: $\nu = 0,\ r_i = 100\ \text{mm},\ r_o = 725\ \text{mm},\ P_{fi} = 50\ \text{MPa}$

To be continued…