Linear multipoint constraints in FEniCSx

FEniCS 2021

Jørgen S. Dokken   Garth N. Wells   Chris Richardson

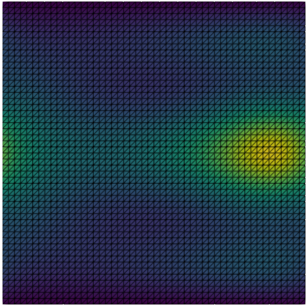
UNIVERSITY OF CAMBRIDGE

March 22, 2021

What is a linear multipoint constraint (MPC)?

A linear combination of degrees of freedom:

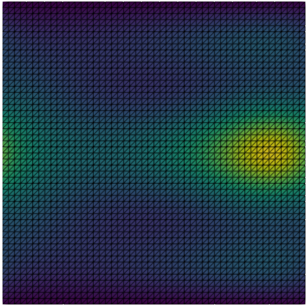
- Periodic conditions: $\mathrm{u}(0, y, z) = \mathrm{u}(L, y, z)$

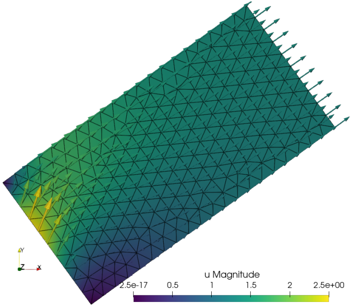


What is a linear multipoint constraint (MPC)?

A linear combination of degrees of freedom:

- Periodic conditions: $\mathrm{u}(0, y, z) = \mathrm{u}(L, y, z)$
- Slip boundary conditions: $\mathrm{u} \cdot \mathrm{n} = 0$

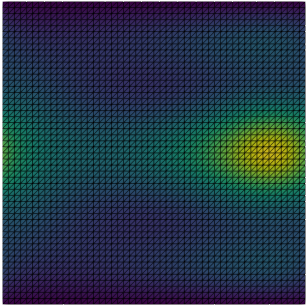


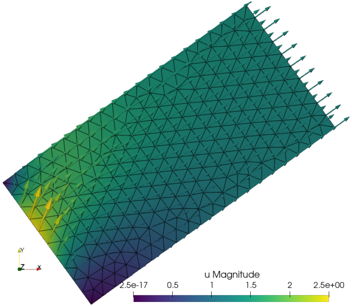


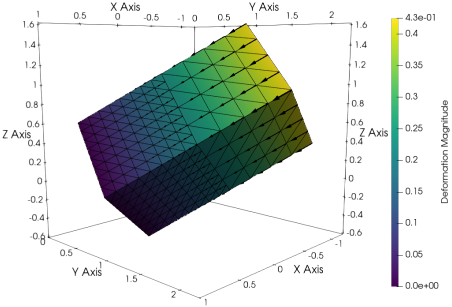
What is a linear multipoint constraint (MPC)?

A linear combination of degrees of freedom:

- Periodic conditions: $\mathrm{u}(0, y, z) = \mathrm{u}(L, y, z)$
- Slip boundary conditions: $\mathrm{u} \cdot \mathrm{n} = 0$
- Frictionless contact: $\mathrm{u}_1 \cdot \mathrm{n}_1 = \mathrm{u}_2 \cdot \mathrm{n}_1, \quad \mathrm{u}_i \in \Omega_i$







To solve a system of linear equations, we eliminate degrees of freedom by using the additional constraints

Find $u = (u_0, \ldots, u_3)^T$ such that

$$Au = b \qquad\qquad u_3 = \zeta u_0$$

To solve a system of linear equations, we eliminate degrees of freedom by using the additional constraints

Find $u = (u_0, \ldots, u_3)^T$ such that

$$Au = b \qquad\qquad u_3 = \zeta u_0$$

Define the prolongation matrix P

$$P\hat{u} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ \zeta & 0 & 0 \end{pmatrix} \begin{pmatrix} u_0 \\ u_1 \\ u_2 \end{pmatrix} = u$$

To solve a system of linear equations, we eliminate degrees of freedom by using the additional constraints

Find $u = (u_0, \ldots, u_3)^T$ such that

$$Au = b \qquad\qquad u_3 = \zeta u_0$$

Define the prolongation matrix P

$$P\hat{u} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ \zeta & 0 & 0 \end{pmatrix} \begin{pmatrix} u_0 \\ u_1 \\ u_2 \end{pmatrix} = u$$

We solve the reduced system

$$(P^T A P)\hat{u} = P^T b$$

where

$$(P^T A P) = \begin{pmatrix} \zeta^2 a_{3,3} + \zeta a_{0,3} + \zeta a_{3,0} + a_{0,0} & \zeta a_{3,1} + a_{0,1} & \zeta a_{3,2} + a_{0,2} \\ \zeta a_{1,3} + a_{1,0} & a_{1,1} & a_{1,2} \\ \zeta a_{2,3} + a_{2,0} & a_{2,1} & a_{2,2} \end{pmatrix}$$

A linear combination gives rise to mixed terms between the master nodes

$$Au = b$$

$$u_1 = \alpha u_0 + \beta u_2$$

A linear combination gives rise to mixed terms between the master nodes

$$Au = b$$

$$u_1 = \alpha u_0 + \beta u_2$$

$$(P^T AP)_{0,0} = \alpha^2 a_{1,1} + \alpha a_{0,1} + \alpha a_{1,0} + a_{0,0}$$

$$(P^T AP)_{0,1} = \alpha\beta a_{1,1} + \alpha a_{1,2} + \beta a_{0,1} + a_{0,2}$$

$$(P^T AP)_{0,2} = \alpha a_{1,3} + a_{0,3}$$

$$(P^T AP)_{1,0} = \alpha\beta a_{1,1} + \alpha a_{2,1} + \beta a_{1,0} + a_{2,0}$$

$$(P^T AP)_{1,1} = \beta^2 a_{1,1} + \beta a_{1,2} + \beta a_{2,1} + a_{2,2}$$

$$(P^T AP)_{1,2} = \beta a_{1,3} + a_{2,3}$$

$$(P^T AP)_{2,0} = \alpha a_{3,1} + a_{3,0}$$

$$(P^T AP)_{2,1} = \beta a_{3,1} + a_{3,2}$$

$$(P^T AP)_{2,2} = a_{3,3}$$

We apply both the conditions we have considered so far

$$
\begin{aligned}
Au &= b \\
u_1 &= \alpha u_0 + \beta u_2 \\
u_3 &= \zeta u_0
\end{aligned}
$$

With prolongation matrix P

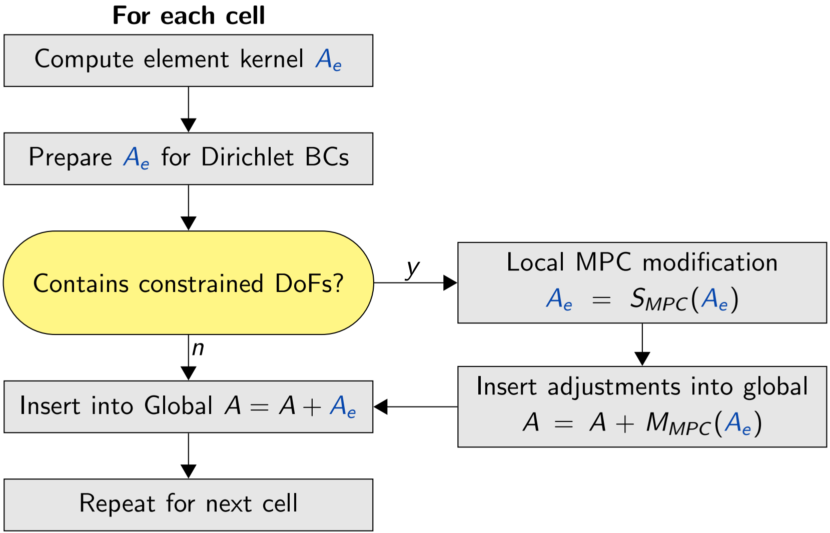
$$
P\hat{u} = \begin{pmatrix} 1 & 0 \\ \alpha & \beta \\ 0 & 1 \\ \zeta & 0 \end{pmatrix} \begin{pmatrix} u_0 \\ u_2 \end{pmatrix} = u
$$

We obtain cross terms between the two constraints

$$Au = b$$
$$u_1 = \alpha u_0 + \beta u_2$$
$$u_3 = \zeta u_0$$

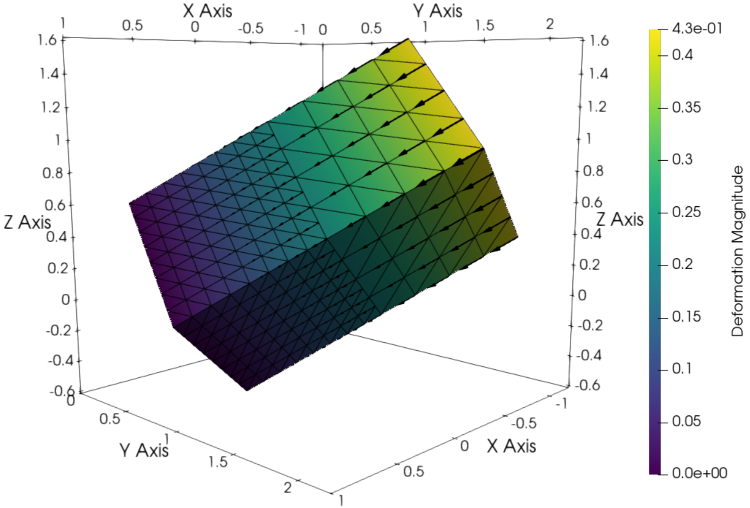
$$\begin{aligned}
(P^T AP)_{0,0} =& \alpha^2 a_{1,1} + \alpha a_{0,1} + \alpha a_{1,0} \\
& + \zeta^2 a_{3,3} + \zeta a_{0,3} + \zeta a_{3,0} + \alpha\zeta a_{1,3} + \alpha\zeta a_{3,1} + a_{0,0} \\
(P^T AP)_{0,1} =& \alpha\beta a_{1,1} + \alpha a_{1,2} + \beta\zeta a_{3,1} + \beta a_{0,1} + \zeta a_{3,2} + a_{0,2} \\
(P^T AP)_{1,0} =& \alpha\beta a_{1,1} + \alpha a_{2,1} + \beta\zeta a_{1,3} + \beta a_{1,0} + \zeta a_{2,3} + a_{2,0} \\
(P^T AP)_{1,1} =& \beta^2 a_{1,1} + \beta a_{1,2} + \beta a_{2,1} + a_{2,2}
\end{aligned}$$

To make the assembly feasible for large systems, we compute the product $P^T A_e P$

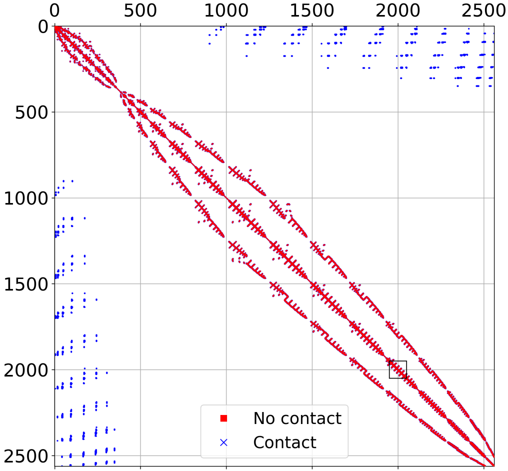


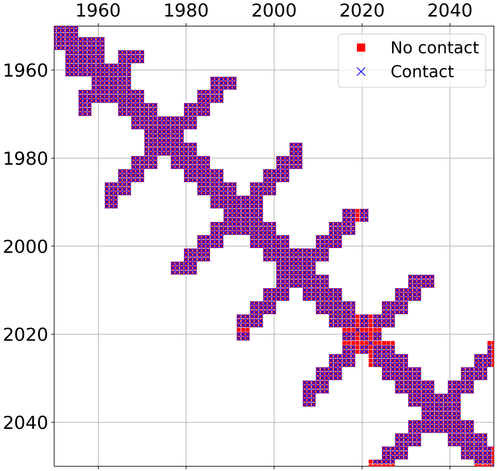
Contact constraint between non-matching meshes

Linear elasticity where a displacement is described on the top (coarse) cube, and the bottom (fine) cube has a slip condition.

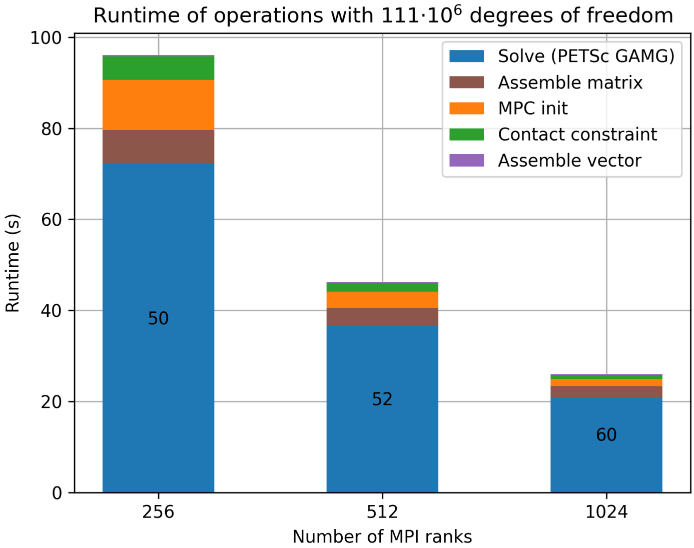


The matrix reduction operation introduces new non-diagonal entries to the sparsity pattern





Strong scaling with 221 million cells



The implementation is written as an add-on to DOLFINx

```python
# Slip constraint on space W using facet-markers
mpc = dolfinx_mpc.MultiPointConstraint(W)
mpc.create_slip_constraint((mt, 1), n, ...)
mpc.finalize()

# Define variational problem using UFL
# ...

# Assemble matrix and vector
A = dolfinx_mpc.assemble_matrix(a, mpc, bcs)
b = dolfinx_mpc.assemble_vector(L, mpc)
A.assemble()

# Solve system using PETSc
# ...

# Backsubstitute from master to puppet dofs
mpc.backsubstitution(uh)
```

Thank you for your attention

https://github.com/jorgensd/dolfinx_mpc