

# SPAI preconditioners for OPM

---

José Eduardo Bueno

August 29, 2022

Universidade Estadual de Campinas, Equinor

## Some theory

---

# Reservoir simulation

- Time and space discretizations of the flow and transport equations on complex domains result on **highly non-linear relations**
- Simulation progression via linearizations → **Newton-like methods**
- Resulting Jacobian matrix **large, sparse and ill-conditioned**

# Linear solver

- Finding adequate points for Newton method requires finding solutions to a series of Jacobian systems

$$Ax = b$$

- Direct solutions require too much memory → **iterative methods** used instead
- OPM Flow uses BiCGStab

# Preconditioning

- Iterative methods can have better convergence if **preconditioned**:

$$MAx = Mb$$

- $MA$  must **approximate the identity** in some sense

# Sparse Approximate Inverse

---

# Method

- Preconditioned matrix approximates the identity in the sense that

$$\min_M \|I - MA\|_F^2 = \sum_{k=1}^n \|(AM - I)e_k\|_2^2$$

- The problem above is equivalent to

$$\min_{m_k} \|Am_k - e_k\|_2, k = 1, \dots, n$$

# Computation

- $\mathcal{J} = \{j | m_k(j) \neq 0\}$
- $\mathcal{I} = \{i | A(i, \mathcal{J}) \neq 0\}$
- $Am_k = e_k$  is equivalent to  $\hat{A}\hat{m}_k = \hat{e}_k$ , where  $\hat{A} = A(\mathcal{I}, \mathcal{J})$ ,  $\hat{m}_k = m_k(\mathcal{J})$  and  $\hat{e}_k = e_k(\mathcal{I})$
- The equivalent system is a **LSQ problem**



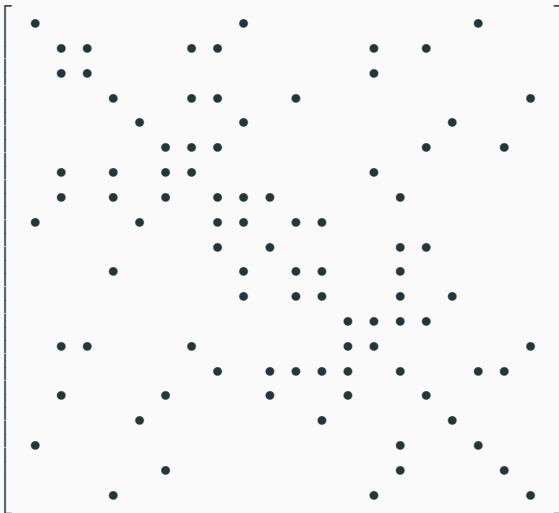
## Computation (2)

- LSQ problem can be solved with QR decomposition
- LSQ solution is spread back to the original sparsity  
 $m_k(\mathcal{J}) = \hat{m}_k$

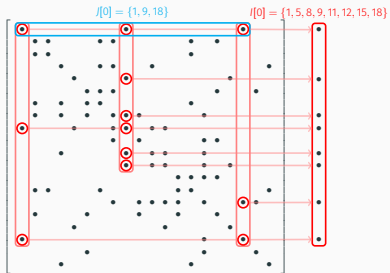
# Enhancement

- If sparsity pattern of  $M$  is chosen to be the **same as of  $A$**  the preconditioner is called SPAI0
- The preconditioner can be enhanced by using the sparsity of  $A^2 \rightarrow$  SPAI1
- The sparsity pattern of the square matrix can be obtained through graph theory:
  - Neighbors with degree  $\leq 2$  in the adjacency graph of the matrix

# Example



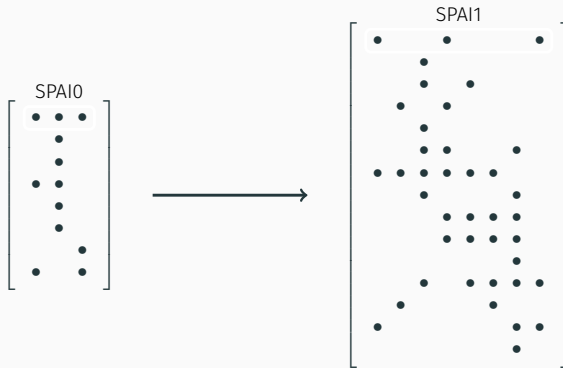
# Example



Resulting matrix for column 0



# Example



# Why SPAI?

---

# OPM preconditioners

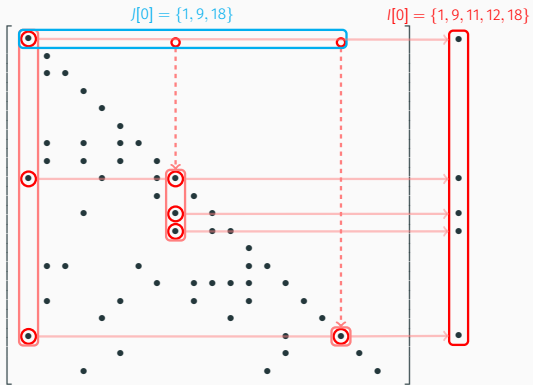
- So far, the path to implementing preconditioners in OPM has been DUNE  $\rightarrow$  OPM
- SPAI is **highly parallelizable**
- The preconditioner can be applied as a simple SpMV

# Implementations and results

---

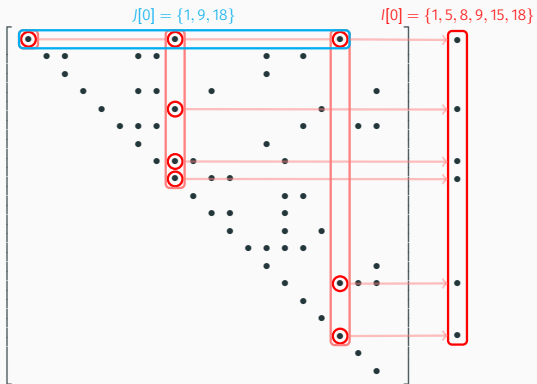


- SPAI scheme is generalizable
- ISAI is SPAI applied to ILU0 decomposition
  - ISAI-L
  - ISAI-U



Resulting matrix for column 0





Resulting matrix for column 0



- Implemented directly into OPM with OpenCL
- Simplifications to solve subsystems
- Merged into `opm-simulators`

# ISAI results (SPE3)

	Linear solve time (s)	Overall Linearizations	Overall Newton Iterations	Overall Linear Iterations
BILU0 (GC)	10.84	569	390	1019
BISAI (GC)	7.54	577	398	5364
BILU0 (LS)	10.95	582	403	3306
BISAI (LS)	7.58	578	399	5271
BILU0 (none)	109.52	582	403	3306
BISAI (none)	12.36	576	397	5226

All results obtained with Intel Integrated Graphics 630 (i7, 7th gen)

# ISAI results (SPE9)

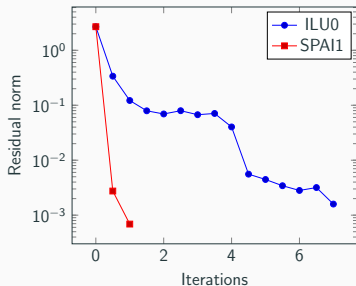
	Linear solve time (s)	Overall Linearizations	Overall Newton Iterations	Overall Linear Iterations
BILU0 (GC)	14.24	318	225	1851
BISAI (GC)	16.97	317	224	2798
BILU0 (LS)	16.65	312	219	1225
BISAI (LS)	16.88	322	229	3159
BILU0 (none)	1131.63	312	219	1225
BISAI (none)	95.42	320	227	3105

All results obtained with Intel Integrated Graphics 630 (i7, 7th gen)

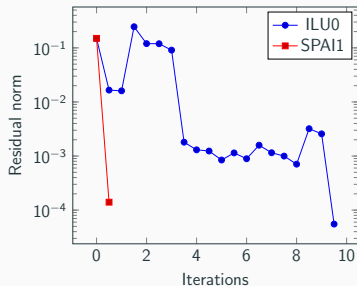
- Initially implemented as a standalone DUNE module
- Solutions for the LSQ problems found with SPQR
- Modifications on DUNE and SPQR to work with blocked matrices (merge request for DUNE under way)
- Currently working within OPM as a CPU preconditioner (pull request for OPM under way)

# SPAI standalone results (SPE1)

BiCGStab - SPE1 (case 1)



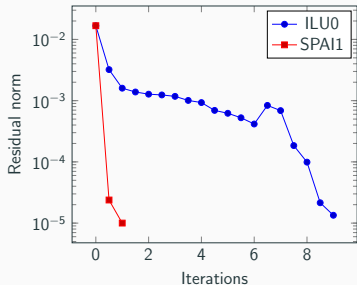
BiCGStab - SPE1 (case 2)



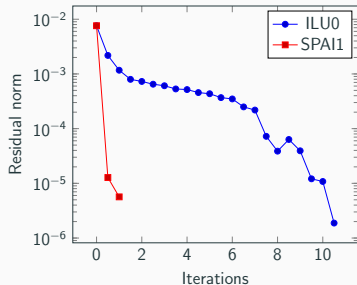


# SPAI standalone results (SPE3)

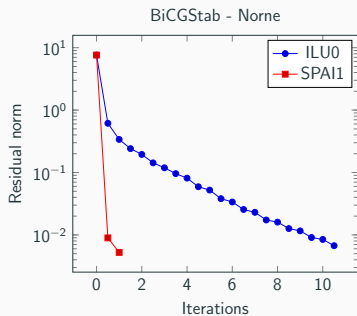
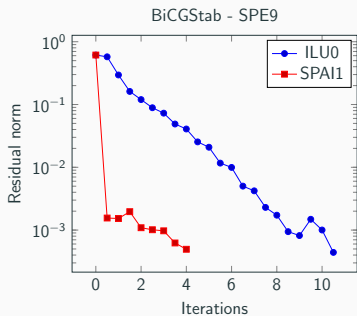
BiCGStab - SPE3 (case 1)



BiCGStab - SPE3 (case 2)



# SPAI standalone results (SPE9 and Norne)



# SPAI OPM results

	Linear solve time (s)	Overall Linearizations	Overall Newton Iterations	Overall Linear Iterations
CPR + GPU (SPE1)	29.91	393	270	823
SPAI (SPE1)	97.56	667	542	5006
CPR + GPU (SPE3)	43.65	543	364	1108
SPAI (SPE3)	103.87	629	450	6195

All results obtained with Intel Integrated Graphics 630 (i7, 7th gen)

# SPAI GPU implementation (OpenCL)

- Still not integrated to OPM

	DUNE (ms)	OpenCL
SPE1	0.095	0.965
SPE3	0.111	0.951
SPE9	4.737	13.578
Norne	28.057	51.099

All results obtained with Intel Integrated Graphics 630 (i7, 7th gen)

## **Conclusions and future work**

---

# Conclusions and future works

- SPAI is in a very imature state
- However, for large models performance seem to catch-up with DUNE
- Assemble phase can also be implemented in GPU
- Fix ISAI!
- Experiment with more powerful GPUs
- Experiment with communication avoiding QR decomposition
- Experiment with inexact solutions for LSQ problems

**THANK YOU!**

---