Block Condensing with qpDUNES

Dimitris Kouzoupis
Rien Quirynen, Janick Frasch and Moritz Diehl

Systems control and optimization laboratory (SYSCOP)

TEMPO summer school
August 5, 2015

Table of Contents

QPs in nonlinear MPC

Multistage QP from direct multiple shooting:

$$\underset{Z}{\text{minimize}} \quad \frac{1}{2} Z^\top H Z + h^\top Z \tag{1a}$$

$$\text{subject to} \quad A_\text{e} Z = b_\text{i} \tag{1b}$$

$$A_\text{i} Z \leq b_\text{i} \tag{1c}$$

with $Z = [x_0^\top \; u_0^\top \; \dots \; x_N^\top]^\top$.

QPs in nonlinear MPC

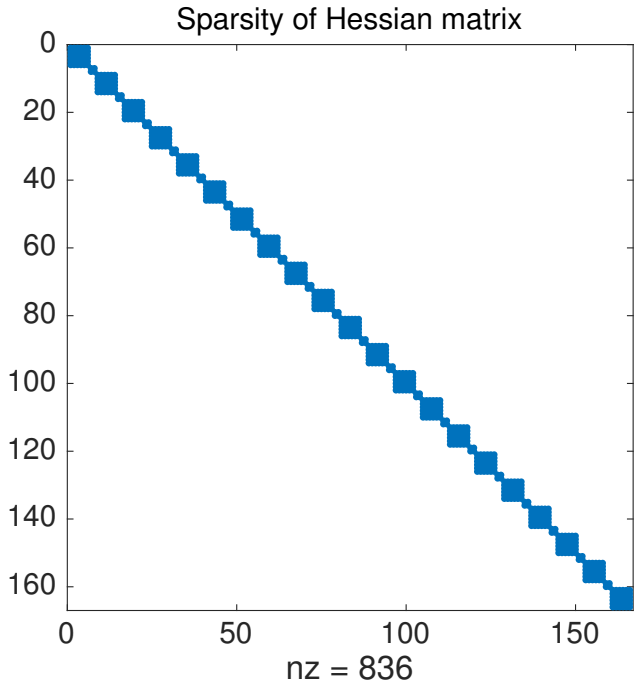
Multistage QP from direct multiple shooting:

$$\underset{Z}{\text{minimize}} \ \frac{1}{2} Z^\top H Z + h^\top Z \tag{1a}$$

$$\text{subject to} \ A_\text{e} Z = b_\text{i} \tag{1b}$$

$$A_\text{i} Z \leq b_\text{i} \tag{1c}$$

with $Z = [x_0^\top \ u_0^\top \ \dots \ x_N^\top]^\top$.



QPs in nonlinear MPC

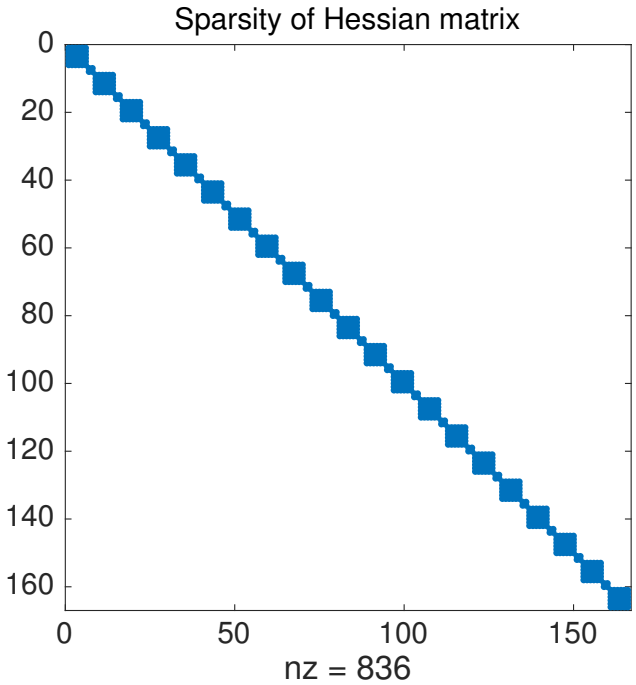
Multistage QP from direct multiple shooting:

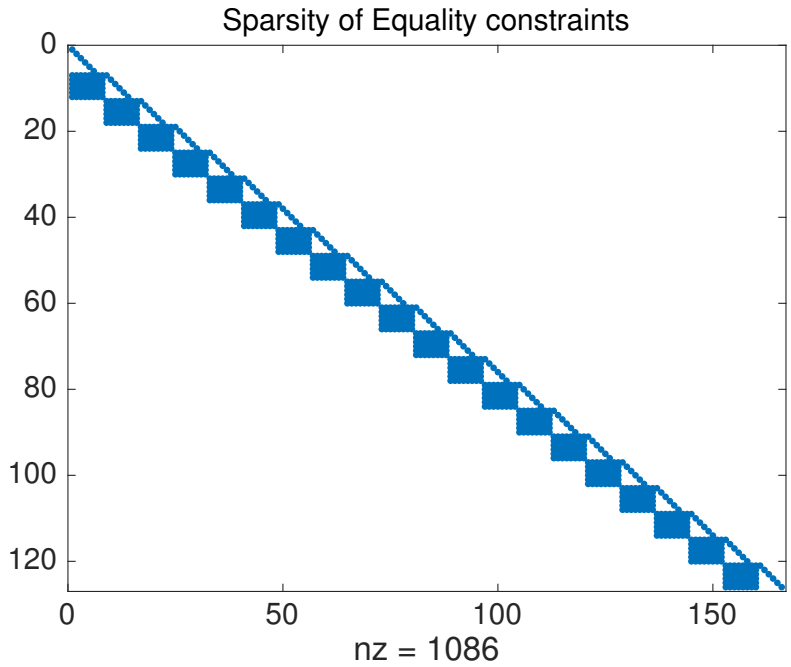
$$\underset{Z}{\text{minimize}} \ \frac{1}{2} Z^\top H Z + h^\top Z \tag{1a}$$

$$\text{subject to} \ A_\text{e} Z = b_\text{i} \tag{1b}$$

$$A_\text{i} Z \leq b_\text{i} \tag{1c}$$

with $Z = [x_0^\top \ u_0^\top \ \dots \ x_N^\top]^\top$.





Condensed approach

Eliminate equality constraints and solve smaller, dense QP:

$$\underset{U}{\text{minimize}} \ \frac{1}{2} U^\top H_c U + h_c^\top U \tag{2a}$$

$$\text{subject to } A_c U \leq b_c \tag{2b}$$

- Solve for $U = [u_0^\top \ u_1^\top \ \dots \ u_{N-1}^\top]^\top$.
- Condensed QP data are calculated online.
- Solution must be expanded for the next SQP iteration.

Condensed approach

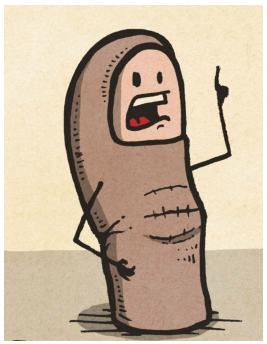
Eliminate equality constraints and solve smaller, dense QP:

$$\underset{U}{\text{minimize}} \ \frac{1}{2} U^\top H_c U + h_c^\top U \tag{2a}$$

$$\text{subject to } A_c U \leq b_c \tag{2b}$$

- Solve for $U = [u_0^\top \ u_1^\top \ \dots \ u_{N-1}^\top]^\top$.
- Condensed QP data are calculated online.
- Solution must be expanded for the next SQP iteration.

Good idea for:



- Short horizons
- Many states
- Few controls

Sparse approach

Alternative: Solve directly the sparse QP

$$\underset{Z}{\text{minimize}} \ \frac{1}{2} Z^\top H Z + h^\top Z \tag{3a}$$

$$\text{subject to} \ A_\text{e} Z = b_\text{i} \tag{3b}$$

$$A_\text{i} Z \leq b_\text{i} \tag{3c}$$

- Keep both states and controls as optimization variables.
- Use a QP solver with linear algebra that exploits sparsity.

Software	**License**	**Method**
FORCES Pro	Commercial	Primal-dual Interior Point, ADMM, FGM
HPMPC	LGPL	Primal-Dual Interior Point
qpDUNES	LGPL	Non-smooth dual Newton method.[1]

[1] Frasch, Sager, and Diehl, "A Parallel Quadratic Programming Method for Dynamic Optimization Problems".

Sparse approach

Alternative: Solve directly the sparse QP

$$\underset{Z}{\text{minimize}} \ \frac{1}{2} Z^\top H Z + h^\top Z \tag{3a}$$

$$\text{subject to} \ A_\text{e} Z = b_\text{i} \tag{3b}$$

$$A_\text{i} Z \leq b_\text{i} \tag{3c}$$

- Keep both states and controls as optimization variables.
- Use a QP solver with linear algebra that exploits sparsity.

Software	**License**	**Method**
FORCES Pro	Commercial	Primal-dual Interior Point, ADMM, FGM
HPMPC	LGPL	Primal-Dual Interior Point
qpDUNES	LGPL	Non-smooth dual Newton method.[1]

[1] Frasch, Sager, and Diehl, "A Parallel Quadratic Programming Method for Dynamic Optimization Problems".

The dual Newton strategy

Block-banded QP:

$$\begin{aligned}
\underset{z}{\text{minimize}} \quad & \sum_{k=0}^{N} \frac{1}{2} z_k^\top H_k z_k + h_k^\top z_k \\
\text{subject to} \quad & E_{k+1} z_{k+1} = C_k z_k + c_k, k = 0, \ldots, N-1, \\
& d_k^l \leq D_k z_k \leq d_k^u, \qquad k = 0, \ldots, N.
\end{aligned}$$

The dual Newton strategy

Block-banded QP:

$$\begin{aligned}
\underset{z}{\text{minimize}} \quad & \sum_{k=0}^{N} \frac{1}{2} z_k^\top H_k z_k + h_k^\top z_k \\
\text{subject to} \quad & E_{k+1} z_{k+1} = C_k z_k + c_k, k = 0, \dots, N-1, \qquad (\lambda_{k+1}) \\
& d_k^l \leq D_k z_k \leq d_k^u, \qquad k = 0, \dots, N.
\end{aligned}$$

Key idea: Dualize equality constraints and apply Newton method to solve dual problem.

$$\begin{aligned}
\underset{\lambda}{\text{maximize}} \quad & \underset{z}{\text{minimize}} \sum_{k=0}^{N} \frac{1}{2} z_k^\top H_k z_k + h_k^\top z_k + \begin{bmatrix} \lambda_k \\ \lambda_{k+1} \end{bmatrix}^\top \begin{bmatrix} -E_k \\ C_k \end{bmatrix} z_k + \lambda_{k+1}^\top c_k \\
\text{subject to:} \quad & d_k^l \leq D_k z_k \leq d_k^u, \; k = 0, \dots, N.
\end{aligned}$$

The dual Newton strategy

Block-banded QP:

$$\begin{aligned} \underset{z}{\text{minimize}} \quad & \sum_{k=0}^{N} \frac{1}{2} z_k^\top H_k z_k + h_k^\top z_k \\ \text{subject to} \quad & E_{k+1} z_{k+1} = C_k z_k + c_k, k = 0, \dots, N-1, \qquad (\lambda_{k+1}) \\ & d_k^l \leq D_k z_k \leq d_k^u, \qquad k = 0, \dots, N. \end{aligned}$$

Key idea: Dualize equality constraints and apply Newton method to solve dual problem.

$$\begin{aligned} \underset{\lambda}{\text{maximize}} \quad & \underset{z}{\text{minimize}} \sum_{k=0}^{N} \frac{1}{2} z_k^\top H_k z_k + h_k^\top z_k + \begin{bmatrix} \lambda_k \\ \lambda_{k+1} \end{bmatrix}^\top \begin{bmatrix} -E_k \\ C_k \end{bmatrix} z_k + \lambda_{k+1}^\top c_k \\ \text{subject to:} \quad & d_k^l \leq D_k z_k \leq d_k^u, \ k = 0, \dots, N. \end{aligned}$$

Separable in primal variables

The dual Newton strategy

Minimization and summation can be interchanged:

$$\underset{\lambda}{\text{maximize}} \sum_{k=0}^{N} f_k^*(\lambda)$$

with $\lambda \in \mathbb{R}^{Nn_x}$ and stage QPs:

$$f_k^*(\lambda) = \min_{z_k} \frac{1}{2} z_k^\top H_k z_k + \left(h_k^\top + \begin{bmatrix} \lambda_k \\ \lambda_{k+1} \end{bmatrix}^\top \begin{bmatrix} -E_k \\ C_k \end{bmatrix} \right) z_k + \lambda_{k+1}^\top c_k$$
$$\text{s.t. } d_k^l \leq D_k z_k \leq d_k^u$$

- Sub-problems can be solved in parallel by any QP solver.
- qpDUNES supports two options:
  - Clipping (for diagonal hessian and box constraints).
  - qpOASES (for the general case).

The dual Newton strategy

Minimization and summation can be interchanged:

$$\underset{\lambda}{\text{maximize}} \sum_{k=0}^{N} f_k^*(\lambda) \qquad \begin{cases} \text{Concave} \\ \text{PW Quadratic} \\ \text{Once cont. differentiable} \end{cases}$$

with $\lambda \in \mathbb{R}^{Nn_x}$ and stage QPs:

$$f_k^*(\lambda) = \min_{z_k} \frac{1}{2} z_k^\top H_k z_k + \left(h_k^\top + \begin{bmatrix} \lambda_k \\ \lambda_{k+1} \end{bmatrix}^\top \begin{bmatrix} -E_k \\ C_k \end{bmatrix} \right) z_k + \lambda_{k+1}^\top c_k$$
$$\text{s.t. } d_k^l \leq D_k z_k \leq d_k^u$$

- Sub-problems can be solved in parallel by any QP solver.
- qpDUNES supports two options:
  - Clipping (for diagonal hessian and box constraints).
  - qpOASES (for the general case).
- (Unconstrained) Dual problem is solved using a non-smooth Newton method.

Sparse vs condensed formulation

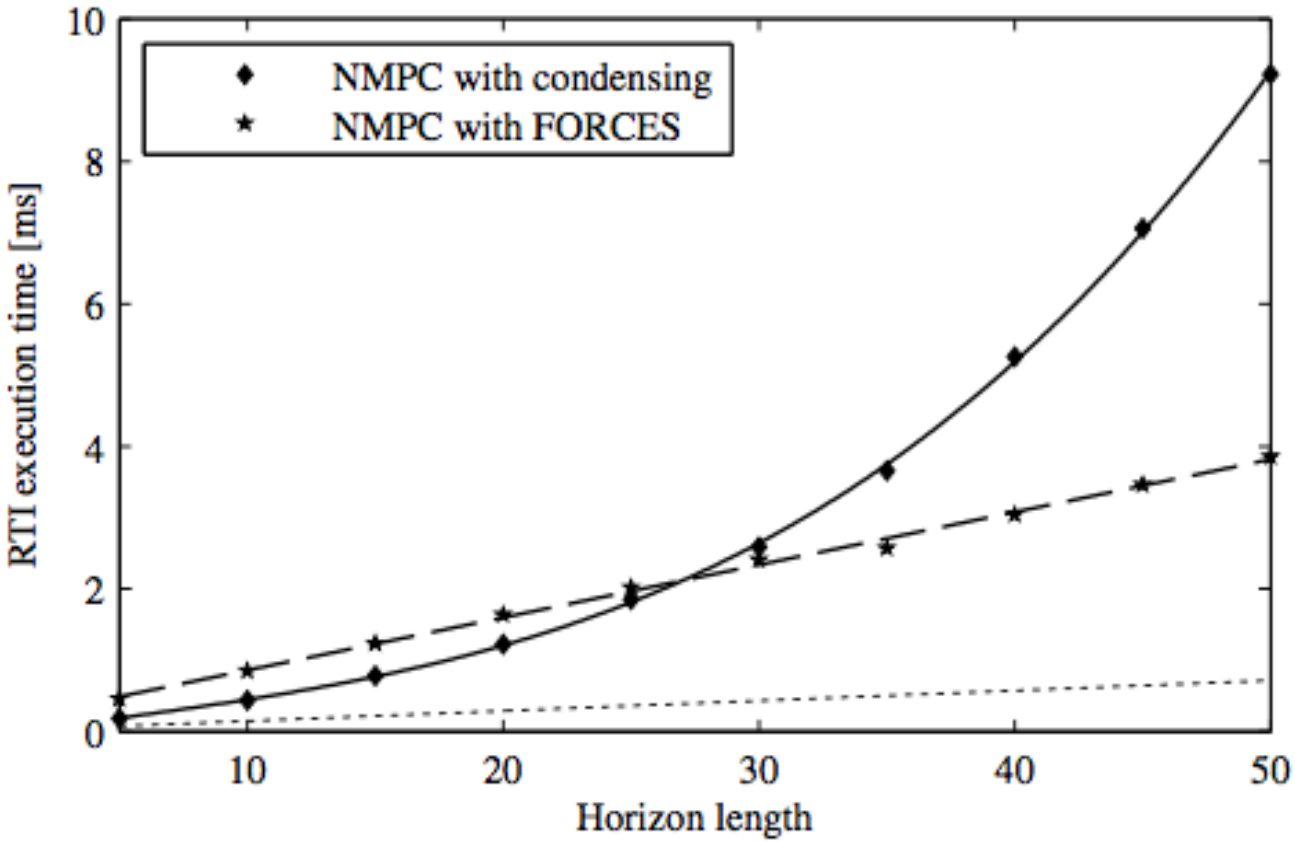


Figure: Comparison taken from[2]

[2] Vukov et al., "Auto-generated Algorithms for Nonlinear Model Predictive Control on Long and on Short Horizons".

Sparse vs condensed formulation

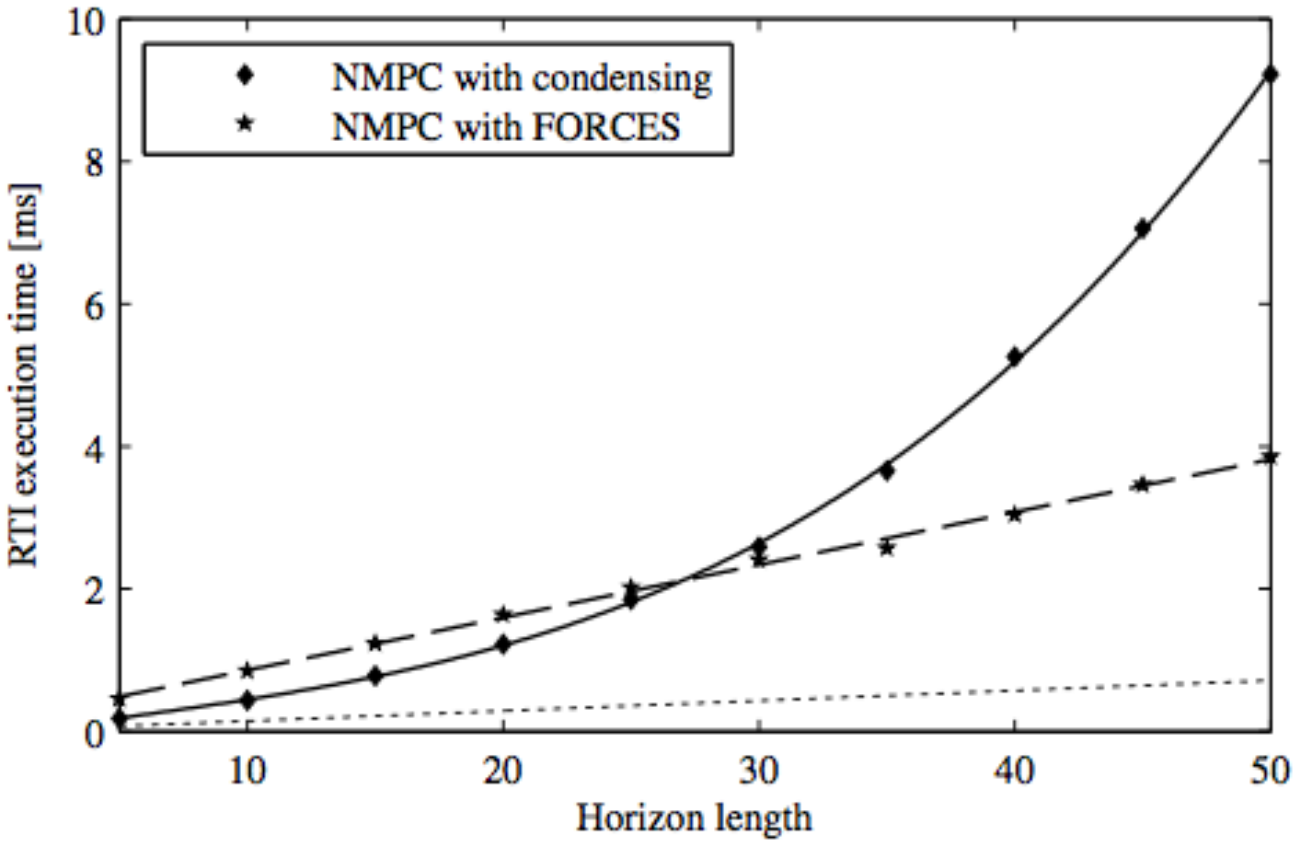


Figure: Comparison taken from[2]

- Can we combine the two approaches?

[2] Vukov et al., "Auto-generated Algorithms for Nonlinear Model Predictive Control on Long and on Short Horizons".

Table of Contents

Block condensing

Main idea:[3]

- Gather M successive stages:

$$\begin{bmatrix} z_k & z_{k+1} & \dots & z_{k+M-1} \end{bmatrix}$$
$$\begin{bmatrix} \overbrace{x_k \quad u_k} & \overbrace{x_{k+1} \quad u_{k+1}} & \dots & \overbrace{x_{k+M-1} \quad u_{k+M-1}} \end{bmatrix}$$

- Eliminate all but the first state variable in each stage:

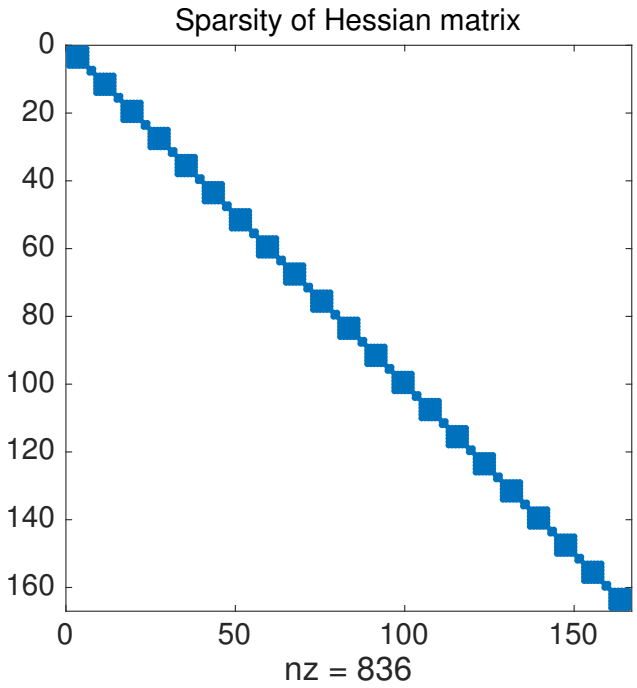
$$\tilde{z}_n = \begin{bmatrix} x_k & u_k & u_{k+1} & \dots & u_{k+M-1} \end{bmatrix}$$

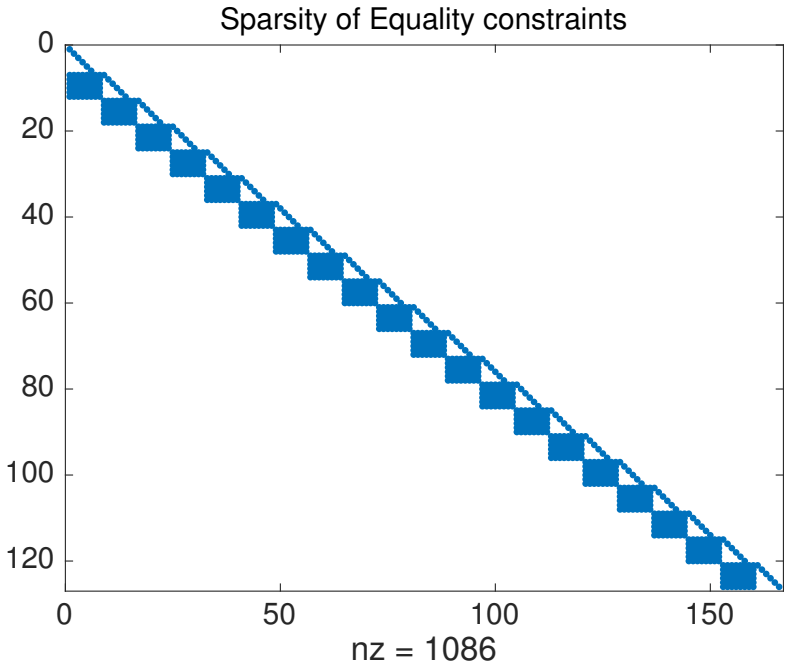
  - Stage index n runs from 0 to N/M.
  - Last stage variable: $\tilde{z}_{N/M} = z_N = x_N$.

- Sparse QPs with different levels of sparsity.

[3] Axehill, "Controlling the level of sparsity in MPC".

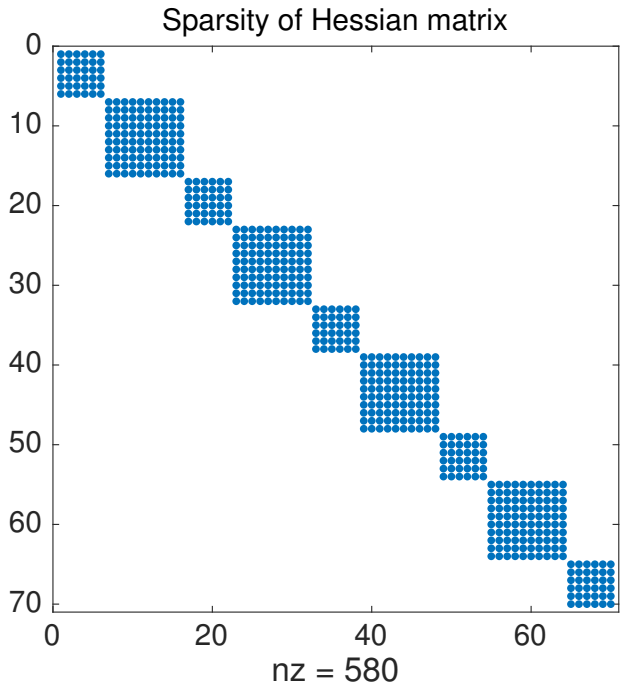
Block condensing

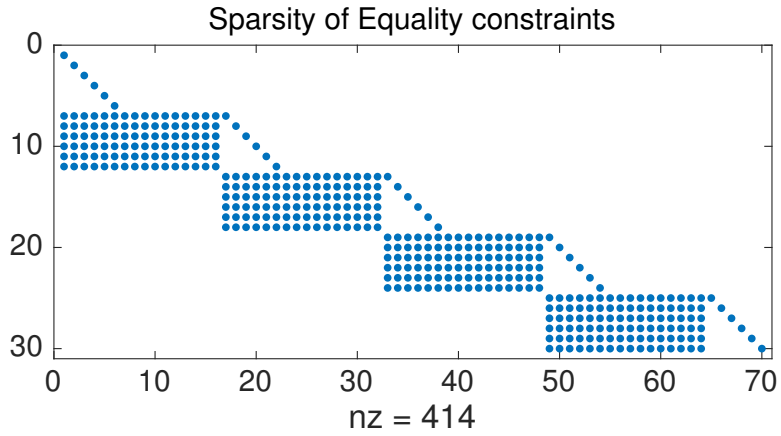




- $n_x = 6,\ n_u = 2,\ N = 20$

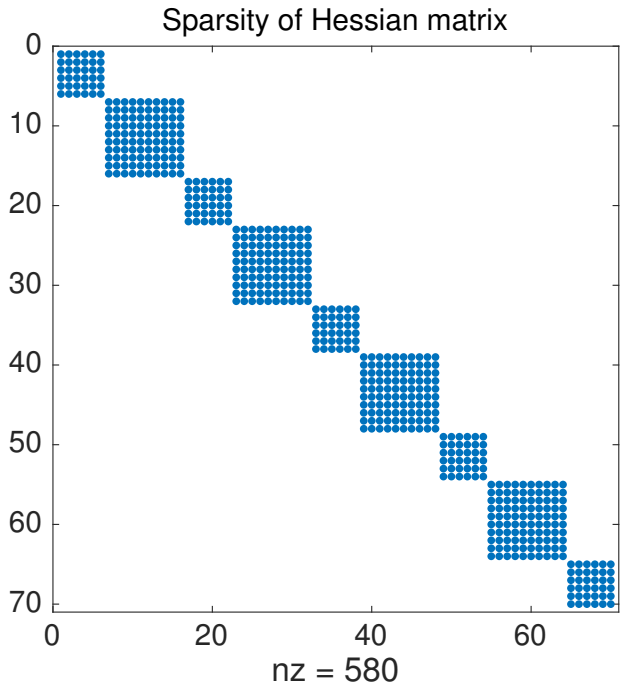
Block condensing

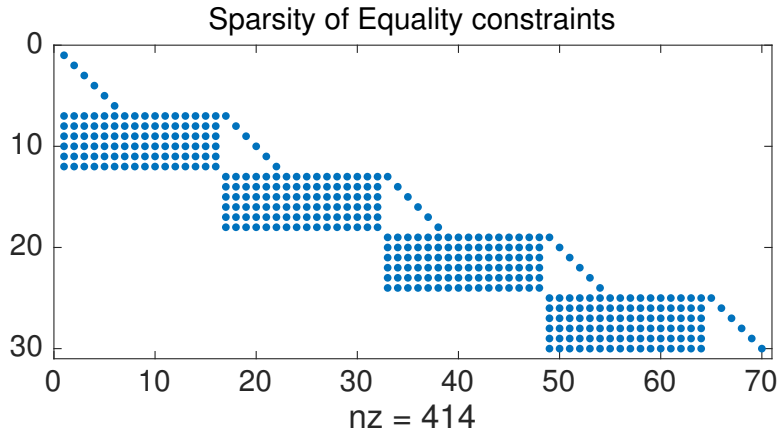




- $n_x = 6,\ n_u = 2,\ N = 20 \rightarrow$ Block size $M = 5$

Block condensing





- $n_x = 6,\ n_u = 2,\ N = 20 \rightarrow$ Block size $M = 5$
- Solve a smaller yet denser QP with the same sparse solver.
  - Complexity still scales linearly with N (for fixed M).
  - Cost of condensing quadratic in M.

Contributions

Block Condensing for Fast Nonlinear MPC with the Dual Newton Strategy

D. Kouzoupis, R. Quirynen, J.V. Frasch and M. Diehl

Proceedings of the 5th IFAC Conference on Nonlinear Model Predictive Control

- Investigate benefits of block condensing in NMPC.
- Efficient, code generated block condensing in ACADO.
- Support for both FORCES Pro and qpDUNES.
- Benchmark example that illustrates speedups up to an order of magnitude.

Benchmark example

- NMPC on a chain of masses (scalable).
  - One point of the chain fixed, the other controlled.
  - Reach the equilibrium position without hitting the wall.
- Code available online.

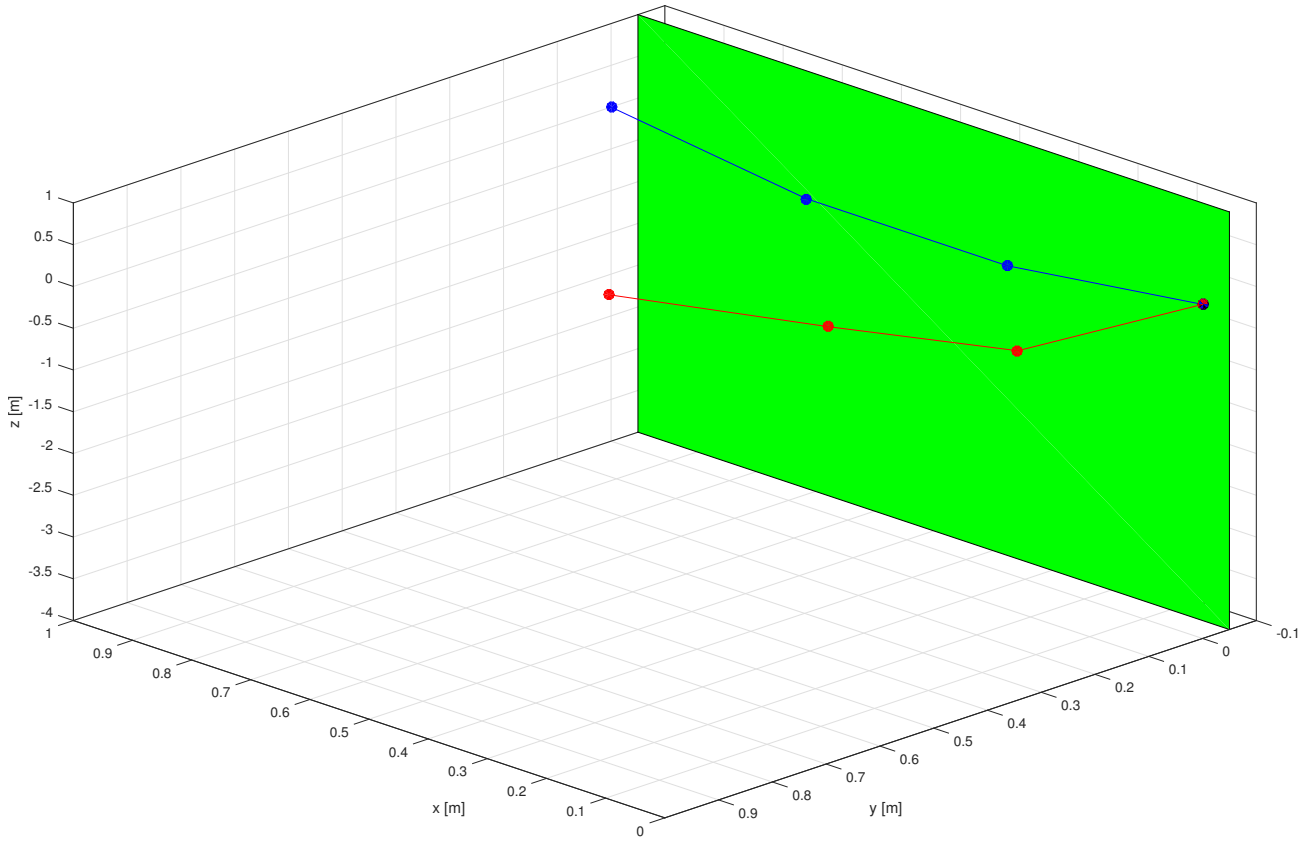
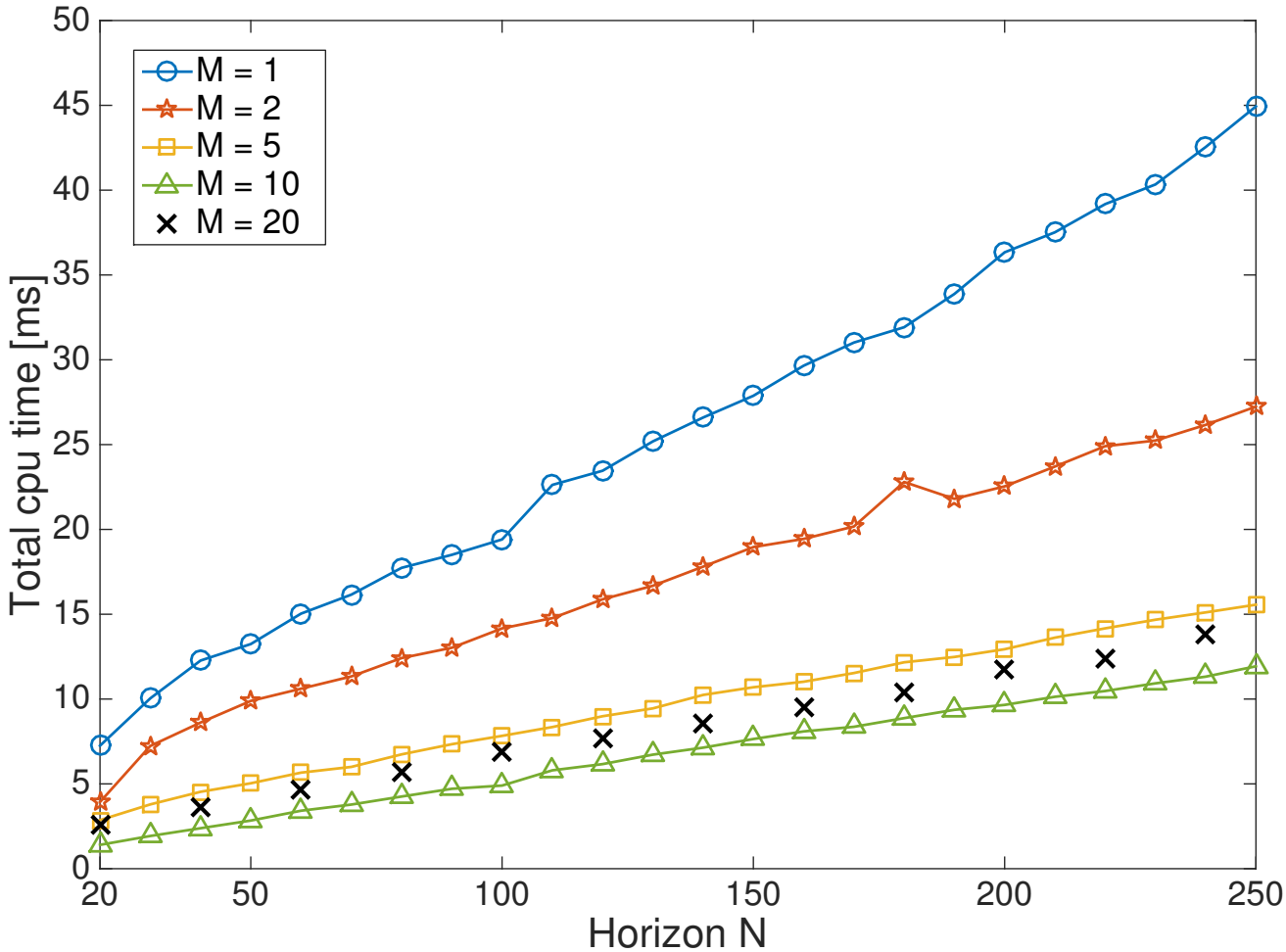


Table of Contents

Benchmark example

Effect of block condensing with qpDUNES on worst-case RTI timings:

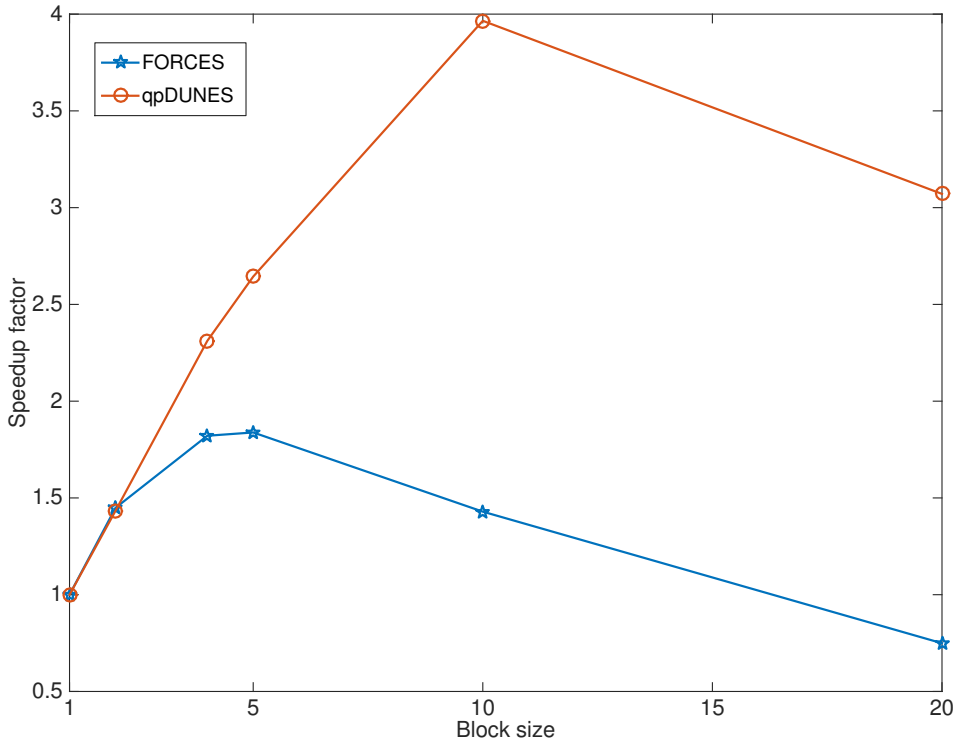
- Different block sizes, wide range of horizons.
- Performance improves up to $M = 10$.



Benchmark example

Comparison of sparse QP solvers.

- Speedup in worst-case total cpu time.
- For block size $M = 1$:
  - FORCES $t_{\max} = 47.14\,\mathrm{ms}$
  - qpDUNES $t_{\max} = 19.41\,\mathrm{ms}$
- Fewer Newton iterations.



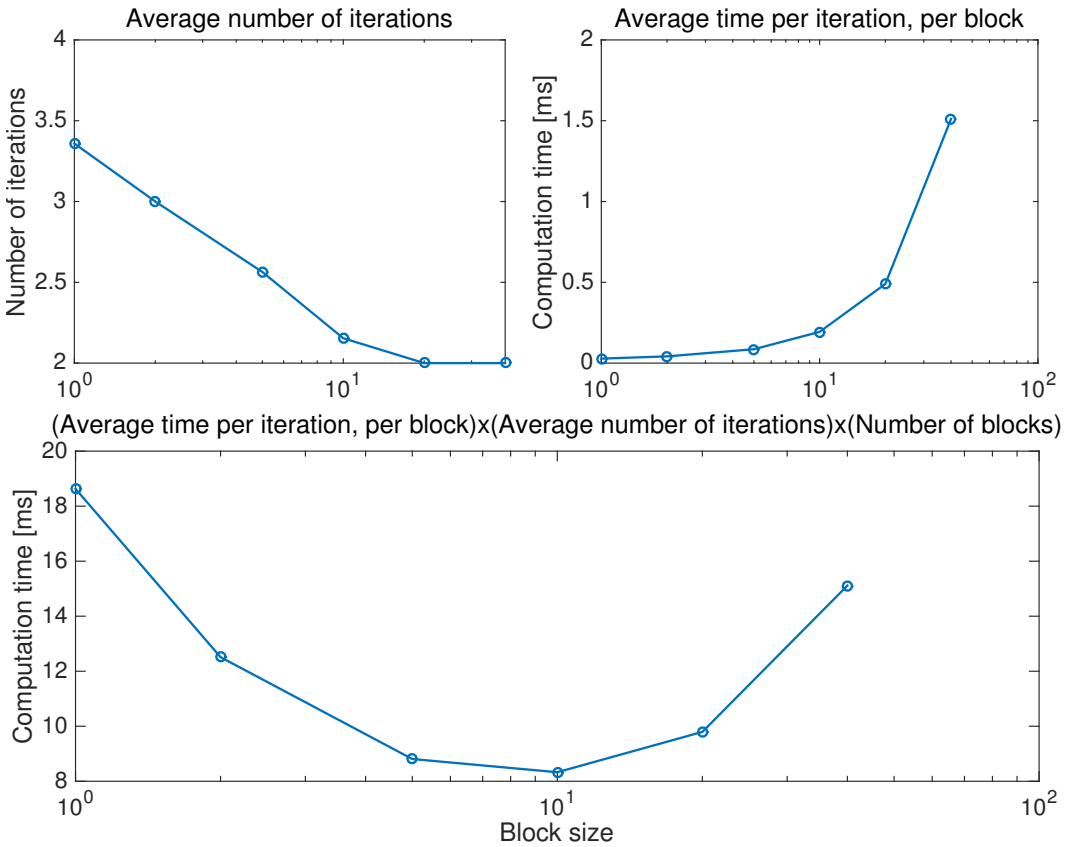
FORCES VS qpDUNES (N=100)

Benchmark example

Increasing the block size:

- More expensive block condensing.
- Condensing of fewer blocks.
- Stage problems:
  - Need longer preparation.
  - Are more difficult to solve.
  - Contain more information.
- Fewer Newton iterations.

Computational load is moved from the Newton steps to the QP sub-problems.

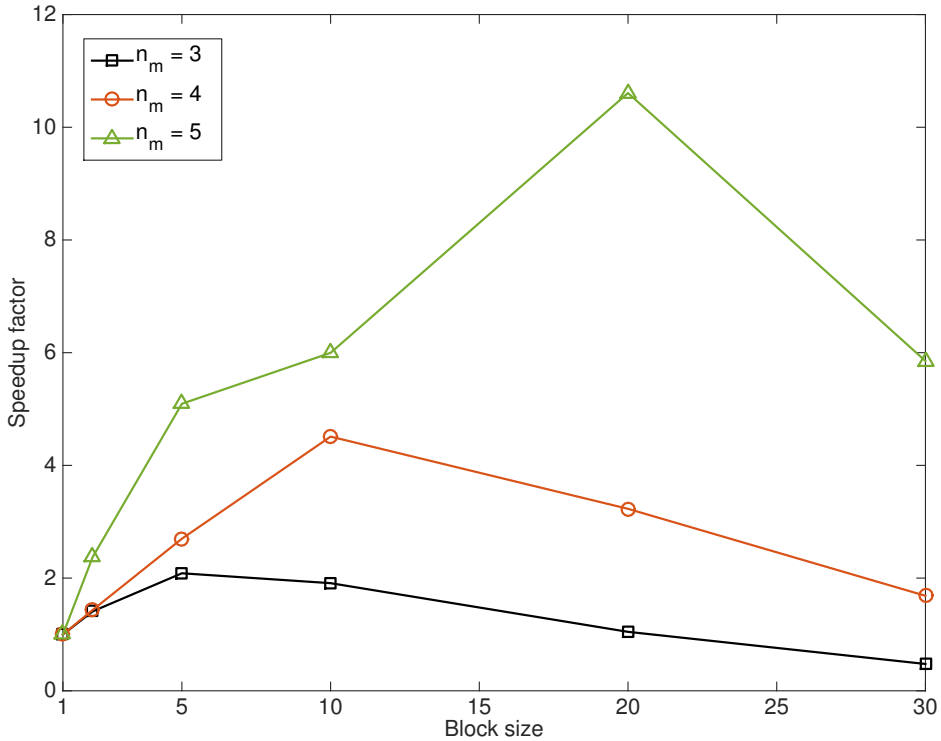


Choice of optimal block size (N=200)

Benchmark example

Comparison for different number of masses.

- Each mass introduces 6 states.
- Benefit of block condensing increases.
- Optimal block size changes.
- Effect on feedback times stronger as condensing is part of the preparation step.



Different number of masses (N=60)

Try it yourself

- Condensing:

```
mpc.set( 'QP_SOLVER', 'QP_QPOASES');
mpc.set( 'SPARSE_QP_SOLUTION', 'FULL_CONDENSING_N2');
```

- Sparse:

```
mpc.set( 'QP_SOLVER', 'QP_QPDUNES'); % or QP_FORCES
mpc.set( 'SPARSE_QP_SOLUTION', 'SPARSE_SOLVER');
```

- Block condensing:

```
mpc.set( 'QP_SOLVER', 'QP_QPDUNES'); % or QP_FORCES
mpc.set( 'SPARSE_QP_SOLUTION', 'BLOCK_CONDENSING_N2');
mpc.set( 'CONDENSING_BLOCK_SIZE', M);
```

qpDUNES-dev on github:

`https://github.com/qpDUNES/qpDUNES-dev`

Benchmark of the paper:

`https://github.com/dkouzoup/BC-RTI`