Efficient Parallel iterative solvers for high order DG methods The curse of the polynomial degree

M. Maischak

Department of Mathematics
Brunel University

Sparse Days 2017

CERFACS, Toulouse

Brunel UNIVERSITY LONDON

Outline

Brunel UNIVERSITY LONDON

Model problem

$$-\Delta u = f \text{ in } \Omega,$$
$$u = g \text{ on } \Gamma.$$

We introduce the mesh $\mathcal{T}_h$ and the space

$$V_h = \{u_h : u_h|_K \in \mathcal{P}_p(K), \forall K \in \mathcal{T}_h\}$$

On the skeleton $\mathcal{E}(\mathcal{T}_h)$ we define

$$\{\{v\}\} = \frac{1}{2}(v^+ + v^-), \qquad \{\{\mathbf{q}\}\} = \frac{1}{2}(\mathbf{q}^+ + \mathbf{q}^-),$$
$$[[v]] = v^+\mathbf{n}_{K^+} + v^-\mathbf{n}_{K^-}, \qquad [[\mathbf{q}]] = \mathbf{q}^+ \cdot \mathbf{n}_{K^+} + \mathbf{q}^- \cdot \mathbf{n}_{K^-}.$$

Brunel UNIVERSITY LONDON

Find $u_h \in V_h$ such that

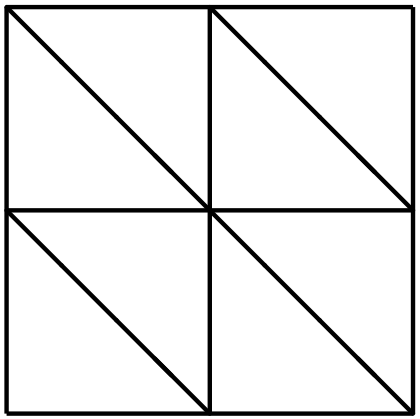
$$A_h(u_h, v) = F_h(v) \qquad \forall v \in V_h$$

$$\begin{aligned} A_h(u,v) = &\sum_{K\in\mathcal{T}_h} \int_K \nabla_h u \cdot \nabla_h v \, dx \\ &- \sum_{\kappa\in\mathcal{E}(\mathcal{T}_h)} \int_\kappa (\{\{\nabla_h v\}\} \cdot [[u]] + \{\{\nabla_h u\}\} \cdot [[v]]) ds \\ &+ \sum_{\kappa\in\mathcal{E}(\mathcal{T}_h)} \int_\kappa c[[u]] \cdot [[v]] ds, \end{aligned}$$

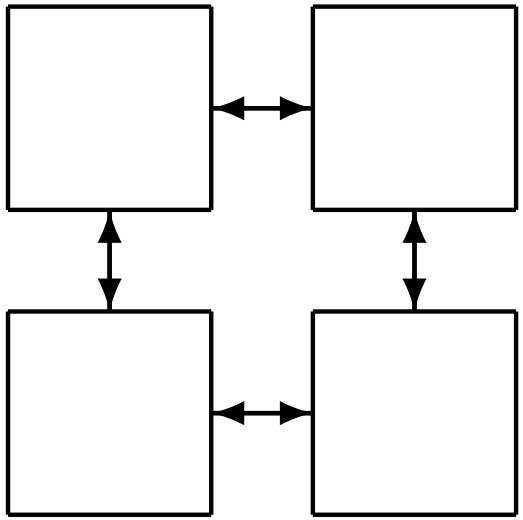
$$F_h(v) = \int_\Omega fv \, dx - \sum_{\kappa\in\mathcal{E}(\mathcal{T}_h)} \int_\kappa g\nabla_h v \cdot \mathbf{n} \, ds + \sum_{\kappa\in\mathcal{E}(\mathcal{T}_h)} \int_\kappa cgv \, ds$$

Brunel UNIVERSITY LONDON

conforming method



DG method



dimension	d
number of elements	n
mesh size	$h \sim n^{-1/d}$
number of coefficients	$dof \sim n \cdot p^d$

Matrix vector multiplication costs

sparse	$\mathcal{O}(n)$
dense	$\mathcal{O}(n^2 \cdot p^{2d})$
sparse-p	$\mathcal{O}(n \cdot p^{2d})$

Brunel UNIVERSITY LONDON

Basis functions (classic) ADLP

$$\mathcal{L}_0(x) = \frac{1}{2}(1-x),$$
$$\mathcal{L}_1(x) = \frac{1}{2}(1+x),$$
$$\mathcal{L}_n(x) = \frac{1}{2n-1}(L_n(x) - L_{n-2}(x)) = \int_{-1}^{x} L_{n-1}(y)\,dy, \quad n \geq 2$$

The antiderivatives of Legendre polynomials have the important and simplifying property

$$\mathcal{L}_n(\pm 1) = 0, \quad n \geq 2$$

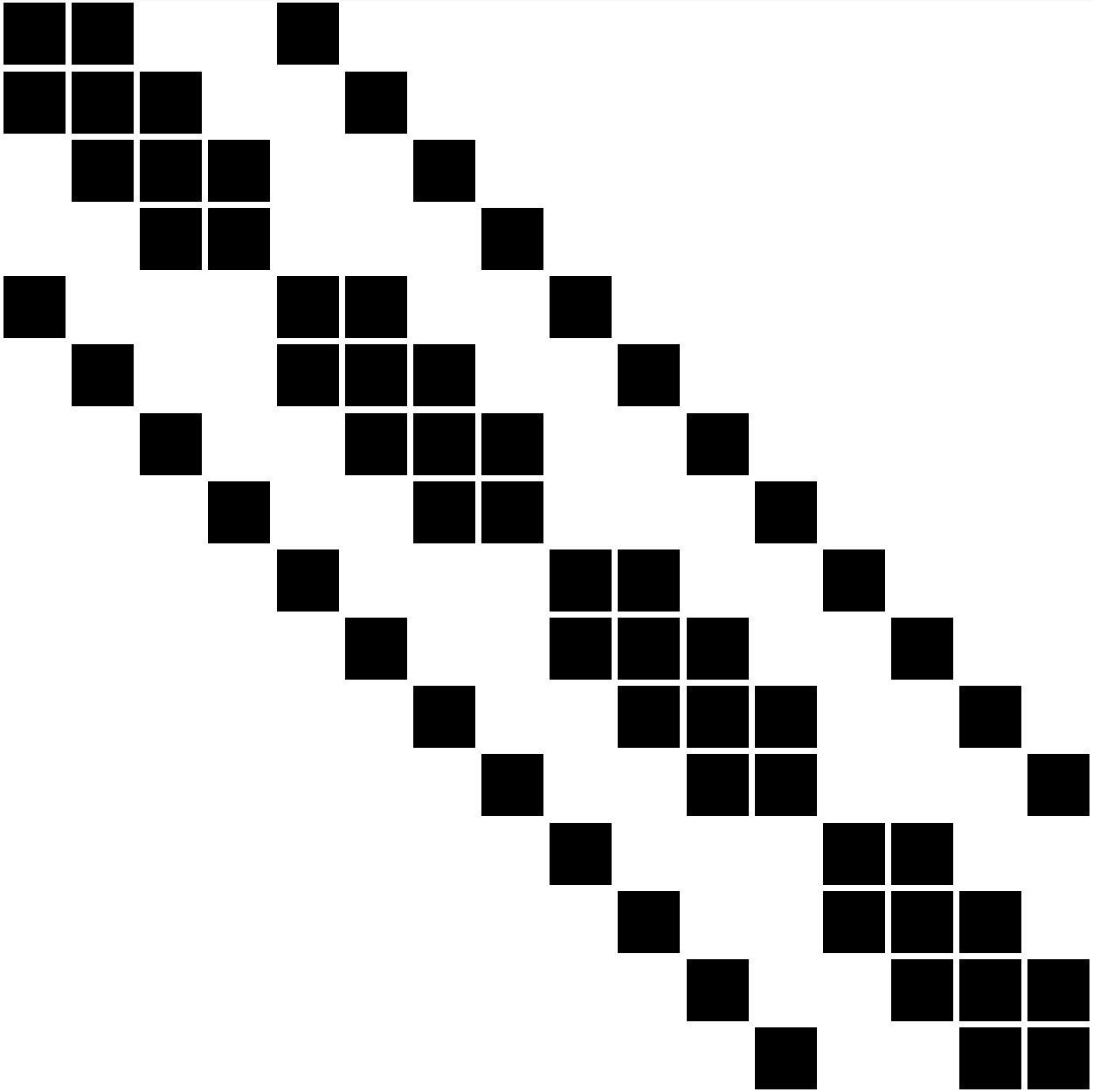
C1-polynomials

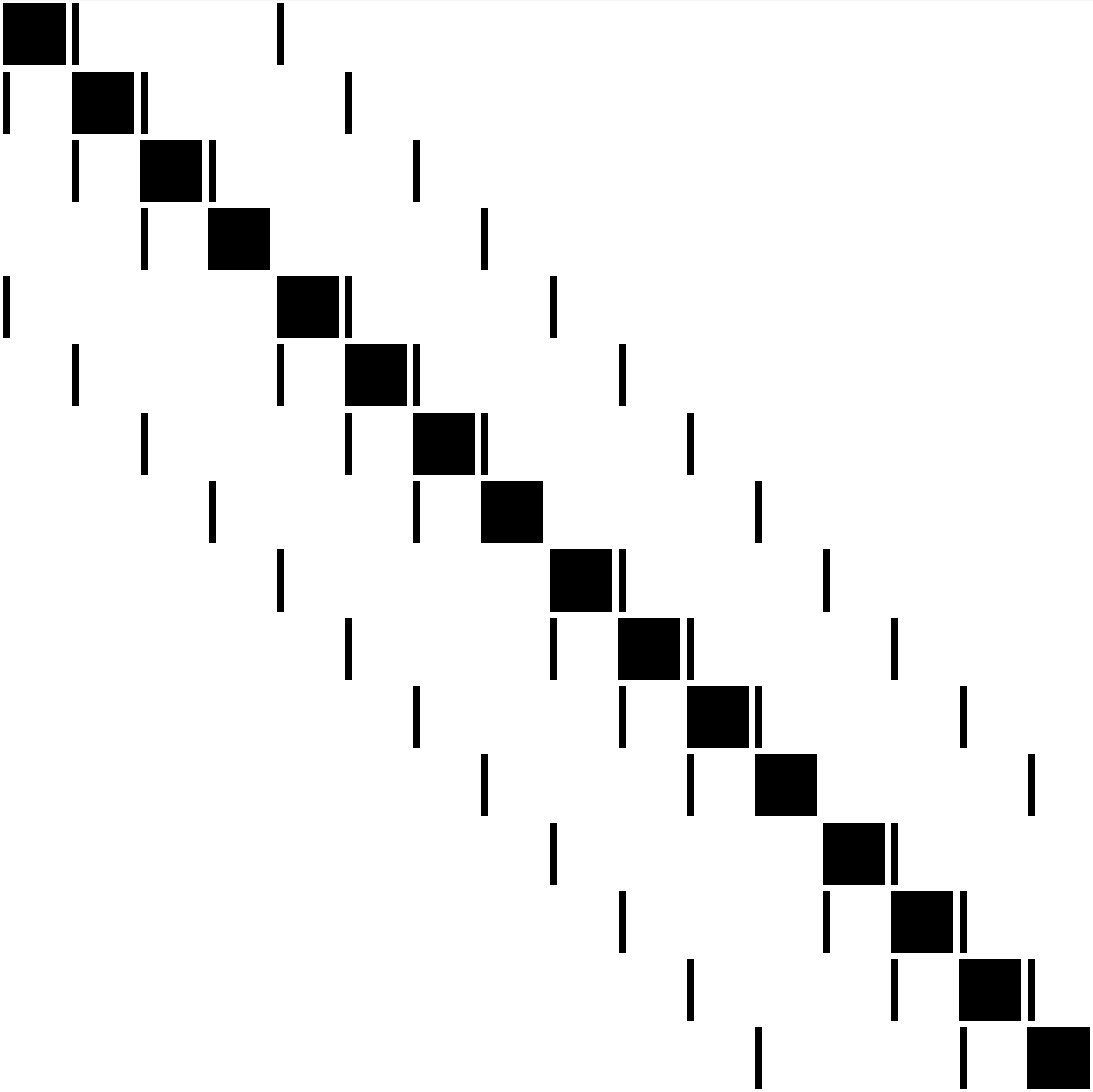
$$C_0(x) = \frac{1}{4}(1-x)^2(2+x), \qquad C_1(x) = \frac{1}{4}(1+x)^2(2-x),$$
$$C_2(x) = \frac{1}{4}(1+x)(1-x)^2, \qquad C_3(x) = \frac{1}{4}(1+x)^2(x-1),$$
$$C_n(x) = (1-x^2)\mathcal{L}_{n-2}(x), \quad n \geq 4$$

The C1-polynomials have the properties

$$C_0(-1) = 1, C_0(1) = 0, C_0'(-1) = 0, C_0'(1) = 0,$$
$$C_1(-1) = 0, C_1(1) = 1, C_1'(-1) = 0, C_1'(1) = 0,$$
$$C_2(-1) = 0, C_2(1) = 0, C_2'(-1) = 1, C_2'(1) = 0,$$
$$C_3(-1) = 0, C_3(1) = 0, C_3'(-1) = 0, C_3'(1) = 1,$$
$$C_n(\pm 1) = C_n'(\pm 1) = 0, \quad n \geq 4$$

Brunel UNIVERSITY LONDON





Using C1- basis functions save asymptotically up to

80% of memory and operations in 2d
86% of memory and operations in 3d

Data transfer via edges/faces, using C1 functions reduces data transfer costs from $\mathcal{O}(p^d)$ to $\mathcal{O}(p^{d-1})$.
Parallelization using MPI
To take advantage of reduced data transfer, distribute data element wise.

Brunel UNIVERSITY LONDON

Parallel matrix vector multiplication

ADLP functions

$$\tau_{MV} = \frac{5n \cdot p^{2d}}{\#proc} + \#proc \cdot n^{1/d} p^{d}$$

C1 funtions

$$\tau_{MV} = \frac{n \cdot p^{2d} + 4 \cdot n \cdot 4p^{d}}{\#proc} + \#proc \cdot n^{1/d} p^{d-1}$$

Brunel UNIVERSITY LONDON

Iterative solver (CG)

Reduce the number of iterations

- Wirebasket basis functions (e.g. locally discrete harmonic)
- Preconditioner, e.g. 2-level additive Schwarz preconditioner, compatible with distributed element data structure

2-level Hhqp additive Schwarz preconditioner

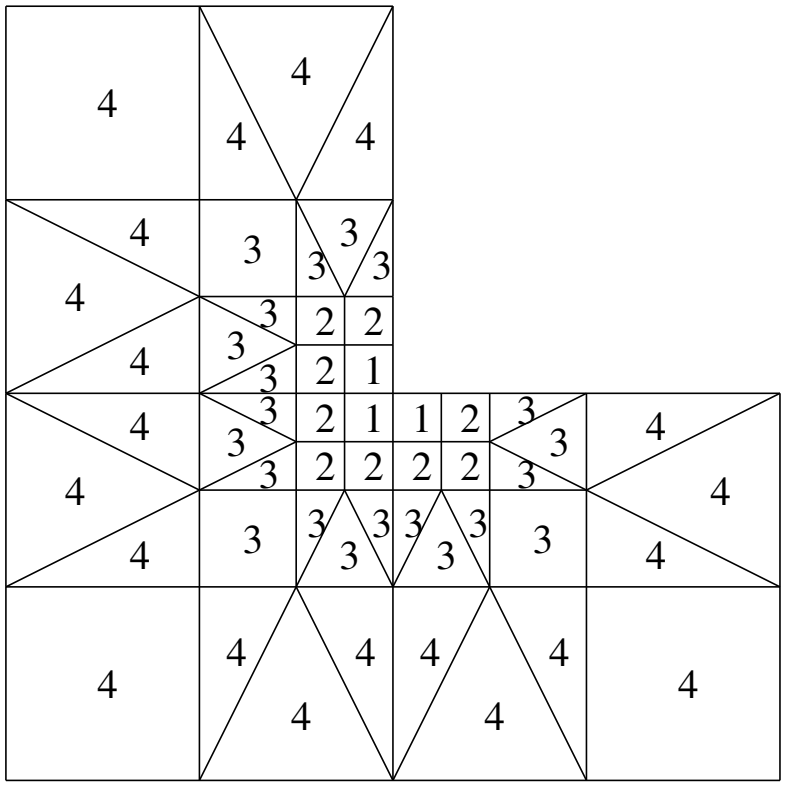
P.F.Antonietti, P. Houston, I. Smears (2015): A note on optimal spectral bounds for nonoverlapping domain decomposition preconditioners for hp-version discontinuous Galerkin methods

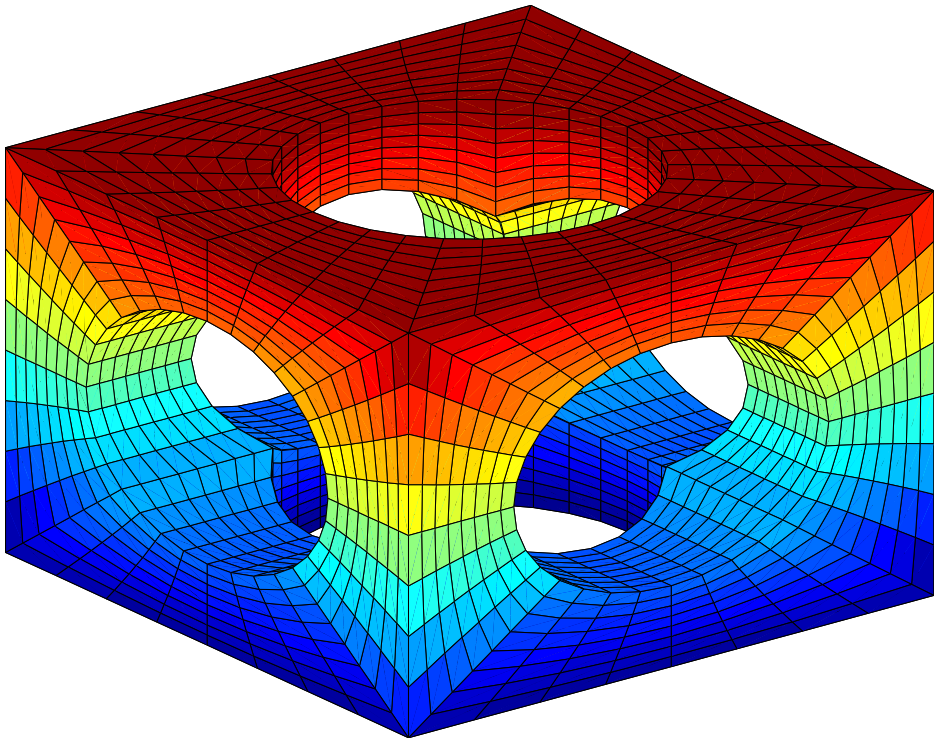
$$\kappa(P_{ad}) \leq C_{\#}^2(N_S + 2), \qquad C_{\#} = C_\sigma \frac{Hp^2}{hq}$$

2-level additive Schwarz preconditioner,
N_S fine grid blocks with pol. degree p,
1 coarse grid block with pol. degree q.

Brunel UNIVERSITY LONDON

hp-geometrical refined mesh

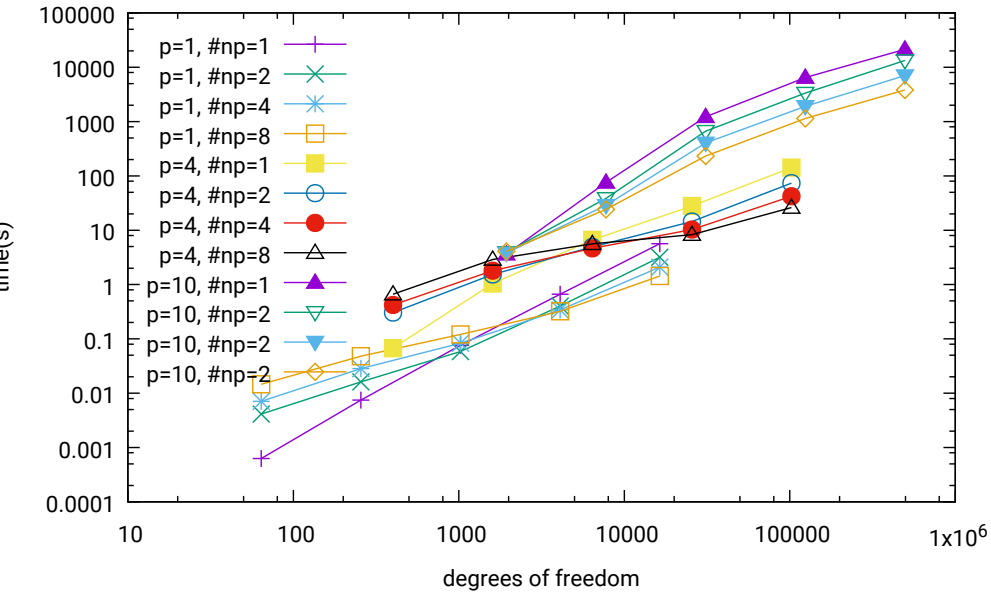


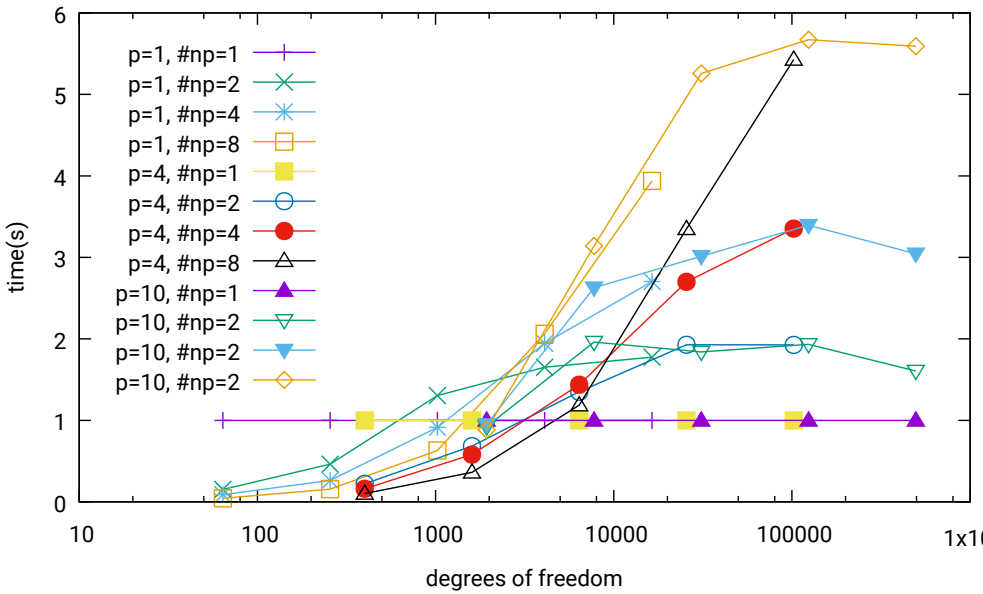


Cluster

12 computing nodes
20 cores, 2.4GHz, 64 GByte per node
1GBit Ethernet

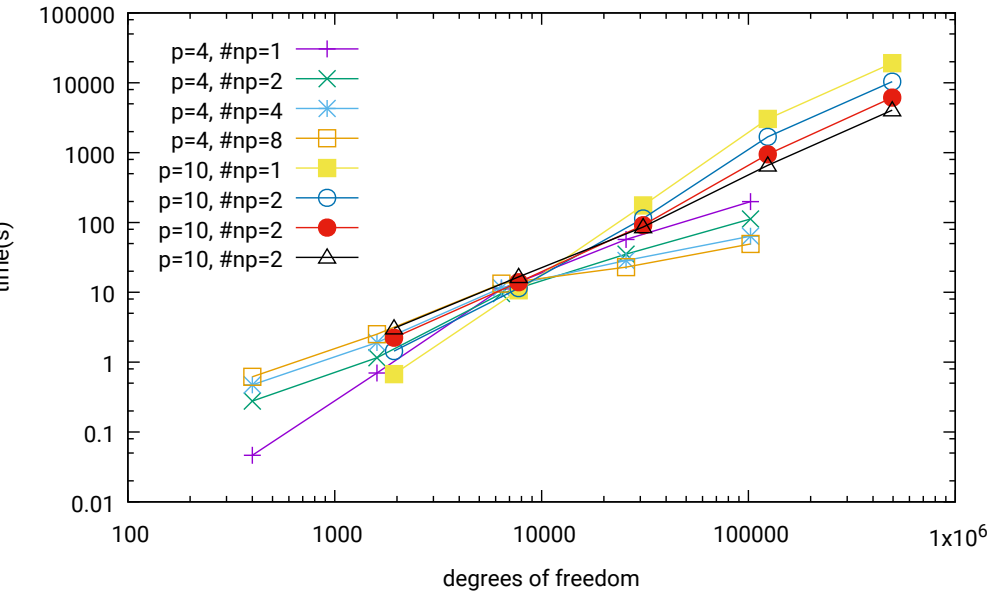
ADLP functions, no preconditioning

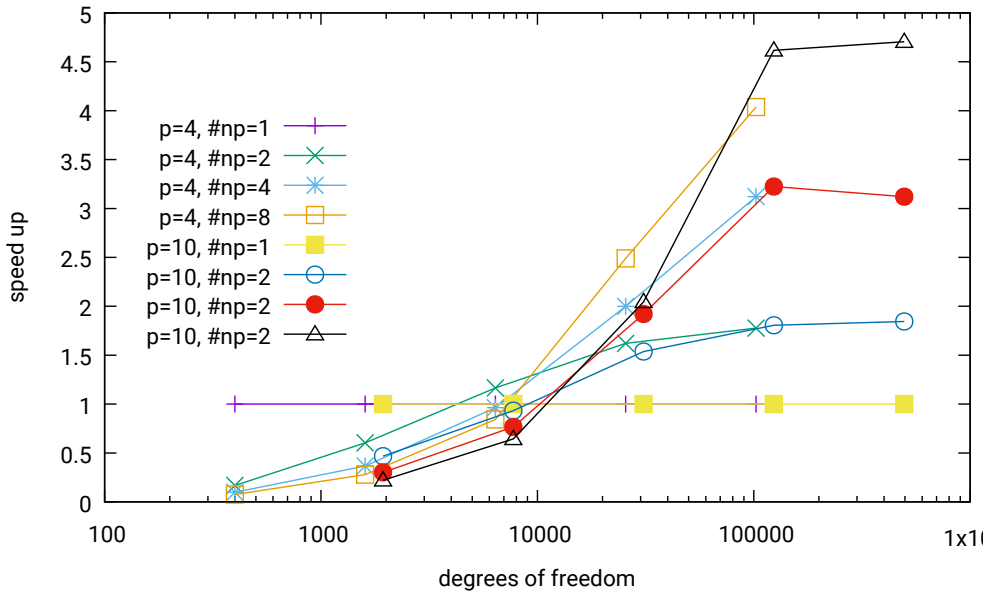




Brunel UNIVERSITY LONDON

C1 functions, no preconditioning





C1 functions, preconditioner, p=10

FDoF	CDoF	#np	Iterations	Solver (s)	speedup
495616	4096	1	480	591.43825	1
495616	4096	2	480	305.07621	1.938657393
495616	4096	4	480	167.29093	3.535387424
495616	4096	8	480	95.92294	6.165764093

Brunel UNIVERSITY LONDON