

Operator Preconditioning in Boundary Element Methods on GPUs

HPCSE 2026, Soláň, May 19

On the occasion of Prof. Zdeněk Dostál's anniversary

D. Lukáš and Z. Machaczek

Dept. Appl. Math., VŠB-TU Ostrava

email: dalibor.lukas@vsb.cz

<http://home1.vsb.cz/~luk76>

 VSB TECHNICAL
UNIVERSITY
OF OSTRAVA

FACULTY OF ELECTRICAL
ENGINEERING AND COMPUTER
SCIENCE

DEPARTMENT
OF APPLIED
MATHEMATICS

Related work

[M. Merta, J. Zapletal, G. Of, O. Steinbach, S. Dohr, R. Watschinger et al. '18-'22]

- BEM4I library,
- Intel Xeon Phi acceleration
- space-time BEM for heat equation
- parallel fast multipole

[A. Prchal '19]

- a 2d GPU-BEM implementation

[J. Homola, '21]

- Acceleration of the Space-Time BEM Using GPUs, MSc. thesis, VSB-TU Ostrava

Operator Preconditioning in Boundary Element Methods on GPUs

Outline

- Evaluation of BEM singular integrals
- Operator preconditioning
- GPU implementation, numerical results
- Conclusion, outlook

Operator Preconditioning in Boundary Element Methods on GPUs

Outline

- Evaluation of BEM singular integrals
- Operator preconditioning
- GPU implementation, numerical results
- Conclusion, outlook

Evaluation of BEM singular integrals

Dirichlet problem for the Laplace equation, direct BIE

$$\left\{ \begin{array}{l} -\Delta u(\mathbf{x}) = 0, \quad \mathbf{x} \in \Omega \\ u(\mathbf{x}) = g(\mathbf{x}), \quad \mathbf{x} \in \Gamma \end{array} \right. \xrightarrow{u=\tilde{V}(t)-W(g)} \left\{ \begin{array}{l} \text{Find } t \in H^{-1/2}(\Gamma) =: Q \\ \langle w, V(t) \rangle = \langle w, (1/2 I + K)(g) \rangle \quad \forall w \in Q \end{array} \right.$$

BEM

(T_1, \dots, T_m) ... triangulation (segmentation in 2d) of Γ ,
 $H^{-1/2}(\Gamma) \supset Q^h = \langle \Psi_1^h, \dots, \Psi_m^h \rangle$... discontinuous elementwise constant functions,
 $H^{1/2}(\Gamma) \supset V^h = \langle \varphi_1^h, \dots, \varphi_n^h \rangle$... hat functions, assume $g = \sum_{j=1}^n g_j \varphi_j^h$,

$$\text{Find } t^h(x) = \sum_{j=1}^n t_j \Psi_j^h(x) : \quad \mathbf{V} \mathbf{t} = \left(\frac{1}{2} \mathbf{M} + \mathbf{K} \right) \mathbf{g},$$

where

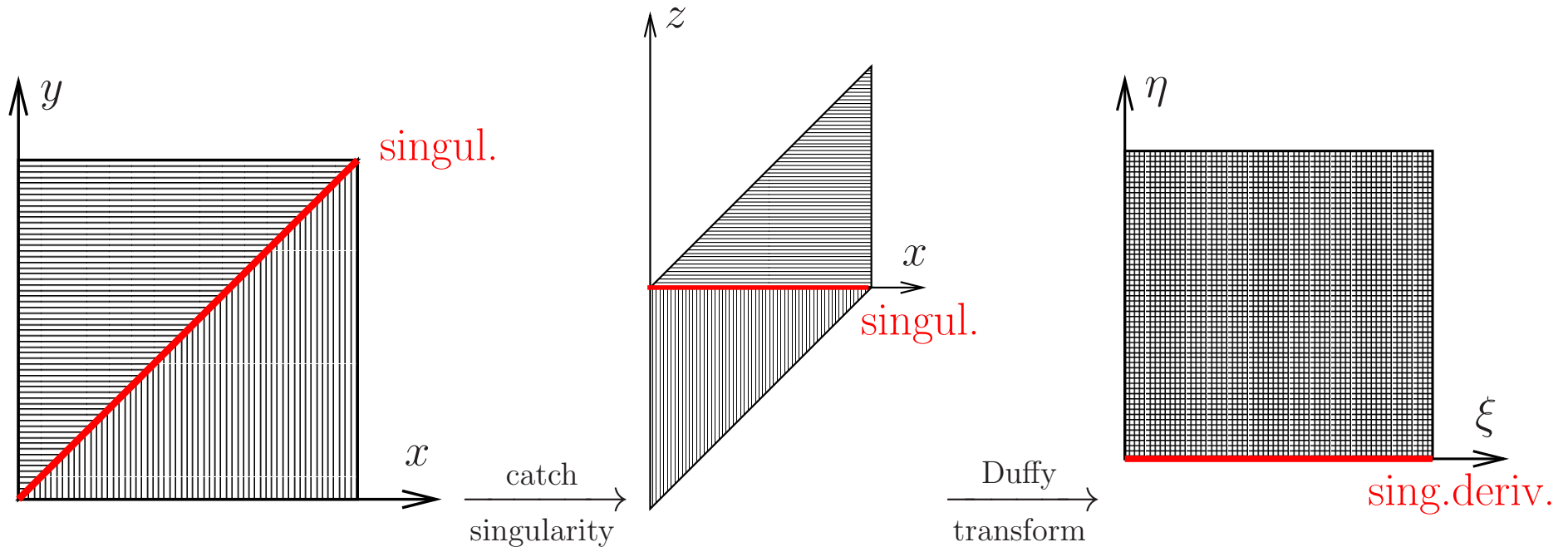
$$(\mathbf{V})_{ij} := \int_{T_i} \int_{T_j} G(\mathbf{x}, \mathbf{y}) ds(\mathbf{y}) ds(\mathbf{x}), \quad (\mathbf{K})_{ij} := \int_{T_i} \int_{\Gamma} \varphi_j^h(\mathbf{y}) \frac{dG}{d\mathbf{n}_y}(\mathbf{x}, \mathbf{y}) ds(\mathbf{y}) ds(\mathbf{x}), \quad \mathbf{M} =$$

with $G(\mathbf{x}, \mathbf{y}) := -\frac{1}{2\pi} \ln \|\mathbf{x} - \mathbf{y}\|$ or $G(\mathbf{x}, \mathbf{y}) := \frac{1}{4\pi \|\mathbf{x} - \mathbf{y}\|}$ in 2d/3d.

Evaluation of BEM singular integrals

Gaussian quadrature of 2d singular kernels, identical segments

$$\int_0^1 \int_0^1 f(x, y) \log |x - y| \stackrel{z:=x-y}{=} \int_0^1 \int_{x-1}^x f(x, x-z) \log |z| \stackrel{\text{Duffy}}{=} \int_0^1 \int_0^1 F(\xi, \eta) \log \eta$$



$$\approx \left\{ \begin{array}{c} \text{Gauss-Legendre} \\ \text{quad. in } \xi \end{array} \right\} \otimes \left\{ \begin{array}{c} \text{„Gauss-Log”} \\ \text{quad. in } \eta \end{array} \right\}$$

Evaluation of BEM singular integrals

CUDA C code example (2d common vertex case)

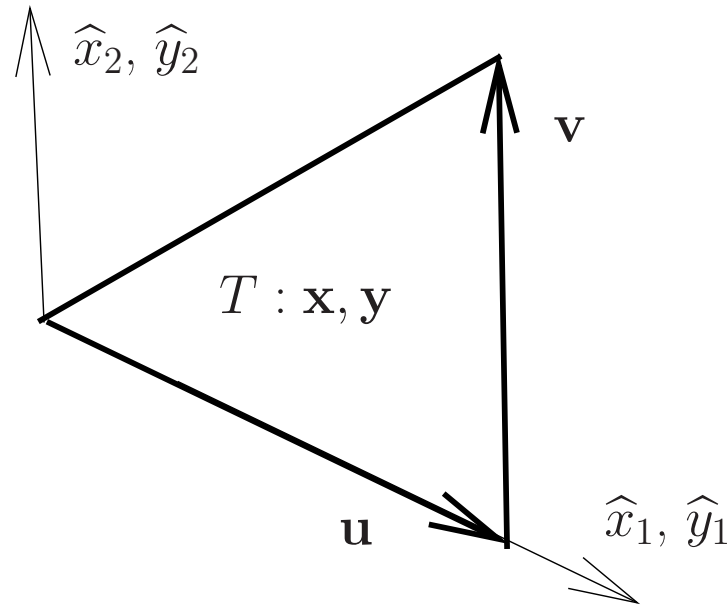
```
__global__ void singleLayerLaplace2d_commonVertex
(float *as, float *bs, int quadn, float *V)
{
    int tid = threadIdx.x + blockIdx.x * blockDim.x, idx = 2*tid, i;
    float a1, a2, b1, b2, J1, J2, xy1, xy2, I, eta, weta;
    a1 = as[idx]; a2 = as[idx+1]; b1 = bs[idx]; b2 = bs[idx+1];
    J1 = sqrt(a1*a1+a2*a2); J2 = sqrt(b1*b1+b2*b2);
    for (i=0, I=0.0f; i<quadn; i++)
    {
        eta = quadPoints[i]; weta = quadWeights[i];
        xy1 = a1-eta*b1; xy2 = a2-eta*b2;
        I += weta * log(xy1*xy1+xy2*xy2);
    }
    V[tid] = -J1*J2/F_PI/2.0f * 0.5f * (-1.0f + I);
}
```

Evaluation of BEM singular integrals

Gaussian quadrature of 3d singular kernels, identical panels

[Erichsen, Sauter '98], [Sauter, Schwab '04]

$$\int_T \int_T G(\|\mathbf{x}-\mathbf{y}\|) dS(\mathbf{y}) dS(\mathbf{x}) = (J_T)^2 \int_0^1 \int_0^{\hat{x}_1} \int_{-\hat{x}_1}^{1-\hat{x}_1} \int_{-\hat{x}_2}^{\hat{z}_1+\hat{x}_1-\hat{x}_2} G(\|[\mathbf{u}, \mathbf{v}] \cdot \hat{\mathbf{z}}\|) d\hat{\mathbf{z}} d\hat{\mathbf{x}}$$



$$\stackrel{6 \times \text{Duffy}}{=} (J_T)^2 \frac{1}{3} \int_0^1 \{G(\|\eta_3 \mathbf{u} + \mathbf{v}\|) + G(\|\mathbf{u} + \eta_3 \mathbf{v}\|) + G(\|\eta_3 \mathbf{u} + (\eta_3 - 1) \mathbf{v}\|)\} d\eta_3$$

Evaluation of BEM singular integrals

CUDA C code example (3d identical panels case)

```
__global__ void singleLayerLaplace3d_idPanels
(float *us, float *vs, int quadn, float *V)
{
    int tid = threadIdx.x + blockIdx.x * blockDim.x, idx = 3*tid, i;
    float u1, u2, u3, v1, v2, v3, J1, J2, J3, J, xy1, xy2, xy3, I, eta, weta;
    u1 = us[idx]; u2 = us[idx+1]; u3 = us[idx+2]; v1 = vs[idx]; ...
    J1 = u2*v3-u3*v2; J2 = u3*v1-u1*v3; J3 = ...; J = J1*J1+J2*J2+J3*J3;
    for (i=0, I=0.0f; i<quadn; i++)
    {
        eta = quadPoints[i]; weta = quadWeights[i];
        xy1 = eta3*u1+v1; xy2 = eta3*u2+v2; xy3 = eta3*u3+v3;
        I += weta / sqrt(xy1*xy1+xy2*xy2+xy3*xy3);
        ...
    }
    V[tid] = J * I / F_PI / 12.0f;
}
```

Operator Preconditioning in Boundary Element Methods on GPUs

Outline

- Evaluation of BEM singular integrals
- Operator preconditioning
- GPU implementation, numerical results
- Conclusion, outlook

Operator preconditioning

Abstract theory [Steinbach, Wendland '98], [Hiptmair '06]

$$\left. \begin{array}{l} A \in \mathcal{L}(Q, Q^*) \dots Q - \text{elliptic } (c_A), \\ B \in \mathcal{L}(V, V^*) \dots V - \text{elliptic } (c_B), \\ D \in \mathcal{L}(Q, V^*) \dots \text{inf-sup stable } (c_D), \\ Q^h \subset Q, V^h \subset V, \dim Q^h = \dim V^h = n \end{array} \right\} \Rightarrow \kappa(D^{-1} B D^{-T} A) \leq \frac{\|A\| \|B\| \|D\|^2}{c_A c_B c_D^2}.$$

Application to BEM

$$Q := H^{-1/2}(\Gamma), V := H^{1/2}(\Gamma),$$

$$Q^h := \langle \Psi_1, \dots, \Psi_n \rangle \text{ element-wise, } V^h := \langle \varphi_1, \dots, \varphi_n \rangle \text{ nodal-wise: } \dim Q^h = \dim V^h!$$

A ... single-layer operator, B ... regularized hyper-singular operator

Laplace: [Steinbach, Wendland '98]

Helmholtz: [Christiansen, Nédélec '00]

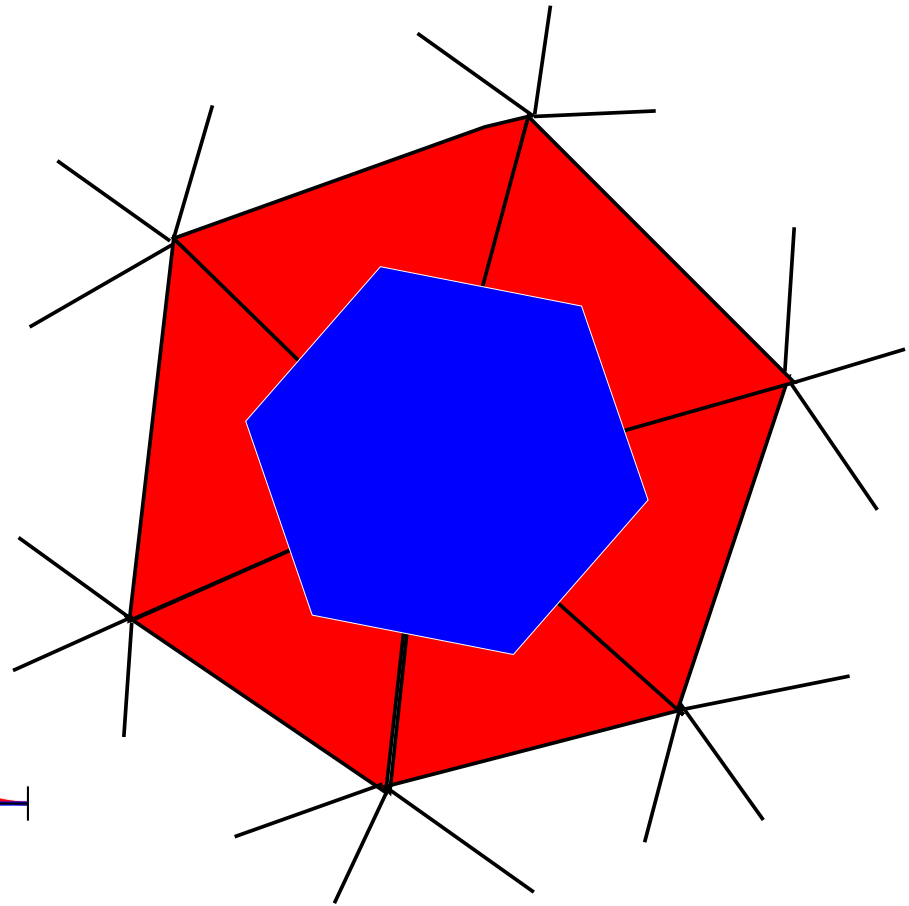
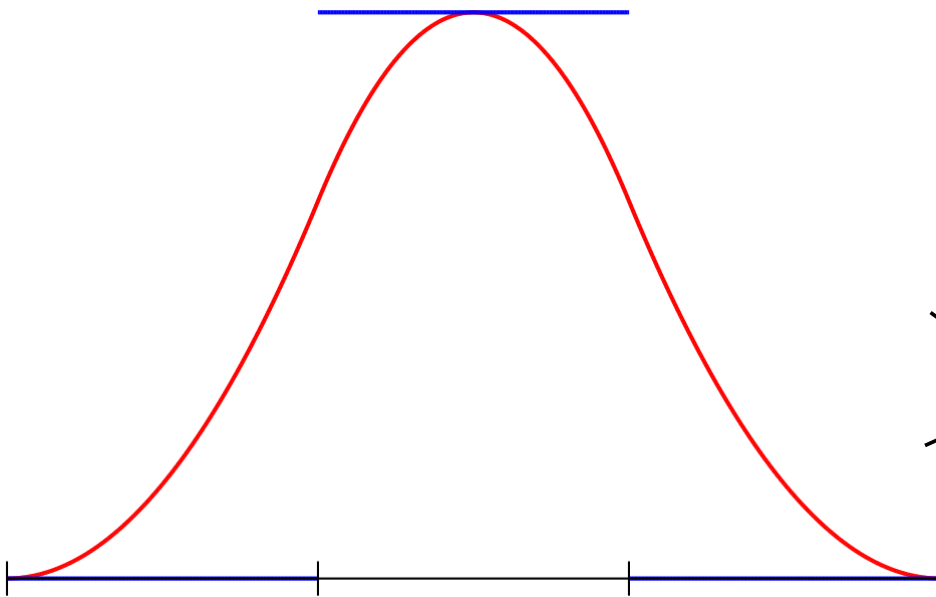
Maxwell: [Buffa, Christiansen '07]

Operator preconditioning

BEM spaces — $Q^h := \langle \Psi_i \rangle_i$, $V^h := \langle \varphi_i \rangle_i$

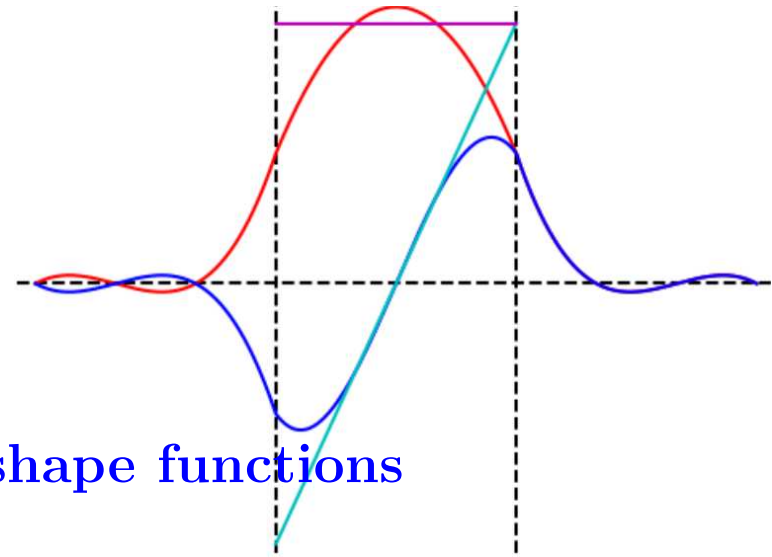
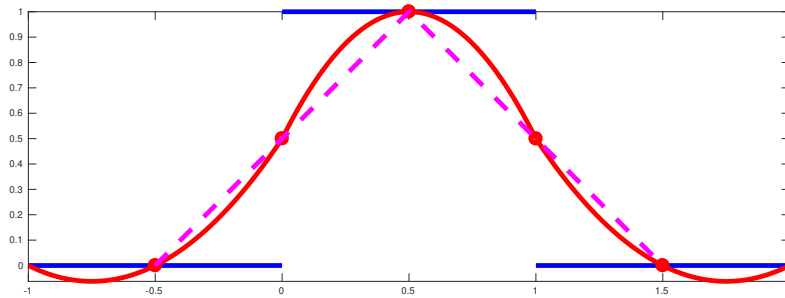
[Steinbach, Wendland '98]:
 C^1 -cont. quadratic B-splines

dual meshes [Steinbach '02]: robust, but
computationally demanding

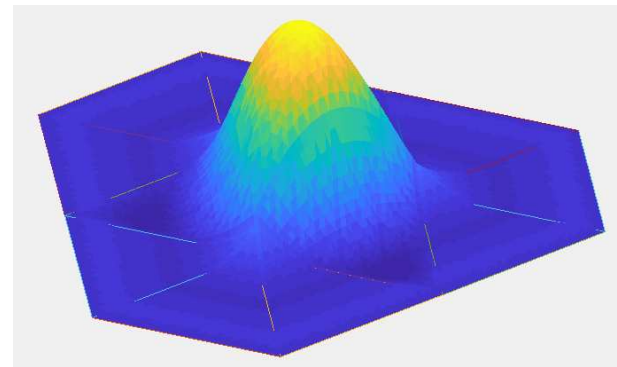
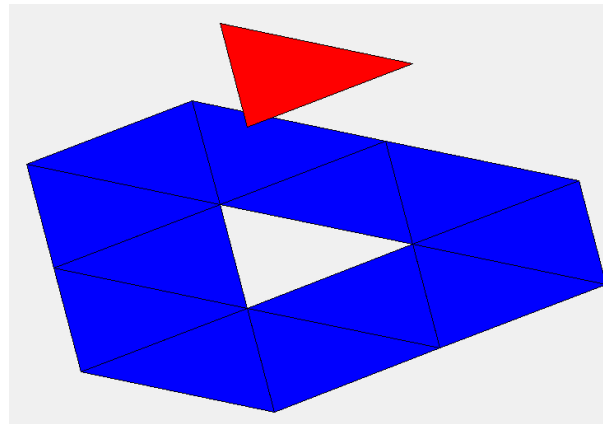
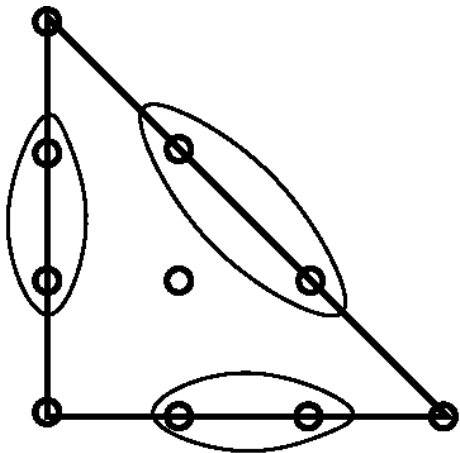


Operator preconditioning

Novel 2d basis $V^h := \langle \varphi_i \rangle_i$ imitating the dual mesh [L., Machaczek '26+]



Novel 3d basis: exploit cubic Lagrange shape functions



Operator Preconditioning in Boundary Element Methods on GPUs

Outline

- Evaluation of BEM singular integrals
- Operator preconditioning
- GPU implementation, numerical results
- Conclusion, outlook

GPU implementation, numerical results

CPU-GPU (CUDA C) implementation

Operator A (1-layer) action (GPU):

- GPU: apply diagonal (identical panels) entries,
- GPU: apply common-edge entries (3d)
- GPU: apply common-vertex entries,
- GPU: apply disjoint-panels entries.

Preconditioner action (CPU-GPU):

- CPU: solve the Gramm matrix D^{-T} ,
- GPU: apply regularized hyper-singular matrix B ,
- CPU: solve the Gramm matrix D^{-1} .

Numerical results, comparison to dual-mesh approach

Dual mesh vers. polynomial preconditioner, $\Omega = (0, 1)^3$, $u(x) = \frac{1}{|x-2|}$

prob. size n	action of \mathbf{V} [s]	action of $\mathbf{G}^{-1}\mathbf{D}_{\text{reg}}\mathbf{G}^{-T}$ [s]	PCG iterations	sol. time [s]	rel. error $\frac{\ t-t_h\ _0}{\ t\ _0}$
768	0.03	0.07	10	3	1.7e-2
3072	0.45	0.8	11	35	7.5e-3
12288	7.1	13	11	554	3.4e-3
49152	120	203	11	4135	1.6e-3

Performed on GeForce RTX 4060 with 3072 cores, PCG tolerance 1e-6.

Numerical results, comparison to dual-mesh approach

Dual mesh vers. polynomial precondition., Ω crankshaft, $u(x) = \frac{1}{|x-2|}$

prob. size n	action of \mathbf{V} [s]	action of $\mathbf{G}^{-1}\mathbf{D}_{\text{reg}}\mathbf{G}^{-T}$ [s]	PCG iterations	sol. time [s]	rel. error $\frac{\ t-t_h\ _0}{\ t\ _0}$
1656	0.14	0.25	13	6	8.8e-4
6624	2.3	3.7	13	88	3.8e-4
24496	36	60	13	1419	1.8e-4
105984	594	944	14	...	1.37e-4

Performed on GeForce RTX 4060 with 3072 cores, PCG tolerance 1e-6.

Operator Preconditioning in Boundary Element Methods on GPUs

Outline

- Evaluation of BEM singular integrals
- Operator preconditioning
- GPU implementation, numerical results
- Conclusion, outlook

Conclusion, outlook

Conclusion

- BEM on GPU,
- an optimal (robust) operator preconditioning,
- higher-order BEM in progress.

Outlook

- an open-source library
- fast parallel BEM on CPU-GPU,
- Helmholtz, Maxwell,

Happy Birthday, Zdeněk!

