Computational Improvements for the BEM including Viscothermal Effects

Mikkel Paltorp Schmitt, Vicente Cutanda Henríquez, Niels Aage & Peter Risby Andersen

The Boundary Element Method

The basis of the (Acoustical) Boundary Element Method is the integral equation

$$\alpha\zeta(\mathbf{x})p(\mathbf{x}) - \int_\Gamma \frac{\partial G(\mathbf{x},\mathbf{y})}{\partial \mathbf{n}(\mathbf{y})}p(\mathbf{y})\ \mathrm{d}S_\mathbf{y} + s(\alpha)k\int_\Gamma G(\mathbf{x},\mathbf{y})v_\mathbf{n}(\mathbf{y})\ \mathrm{d}S_\mathbf{y} = 0. \tag{1}$$

In order to discrete the integrals we need to approximate the geometry. This can be done using e.g. elements

$$\mathbf{x}^e(\mathbf{u}) = \mathbf{X}^e\mathbf{N}(\mathbf{u}) \in \Gamma^e, \quad \forall \mathbf{u} \in \mathcal{L}, \quad \cup_{e=1}^N \Gamma^e \approx \Gamma \tag{2}$$

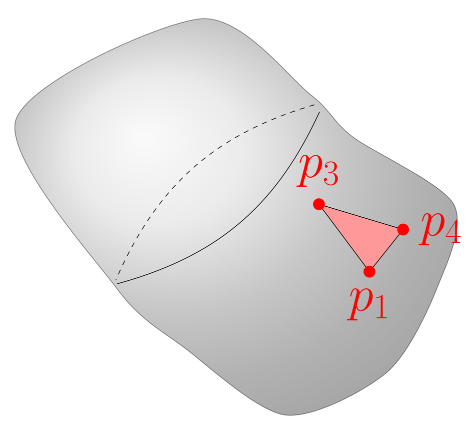


Figure 1. The original domain in shown in gray while a two (linear) elements are shown in red. The red points denote the interpolation nodes of the elements (the columns of $\mathbf{X}^e$).

The pressure (and its normal derivative) on an element can be approximated as

$$p(\mathbf{x}^e(\mathbf{u})) = \mathbf{T}(\mathbf{x}^e(\mathbf{u}))\mathbf{p} = \underbrace{\mathbf{T}(\mathbf{x}(\mathbf{u}))(\mathbf{L}^e)^\top}_{\mathbf{T}^e(\mathbf{u})}\underbrace{\mathbf{L}^e\mathbf{p}}_{\mathbf{p}^e} = \mathbf{T}^e(\mathbf{u})\mathbf{p}^e, \quad \mathbf{u} \in \mathcal{L}, \tag{3}$$

where $\mathbf{L}^e$ is a matrix that extracts the relevant rows of $\mathbf{p}$, i.e. for the element in Figure 1 we have that

$$\mathbf{L}^e = \begin{bmatrix} 1 & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 1 & \dots & 0 \\ 0 & 0 & 1 & 0 & \dots & 0 \end{bmatrix}, \quad \text{so that} \quad \begin{bmatrix} p_1 \\ p_4 \\ p_3 \end{bmatrix} = \mathbf{L}^e\mathbf{p}. \tag{4}$$

The final ingredient is to approximate the integral on the eth element by using a quadrature scheme

$$\int_{\Gamma^e} f(\mathbf{y})\ \mathrm{d}S_\mathbf{y} = \int_\mathcal{L} \text{jacobian}(\mathbf{u})f(\mathbf{u})\ \mathrm{d}\mathbf{u} \approx \sum_{i=1}^{Q} \text{jacobian}(\mathbf{u}_i)w_i f(\mathbf{u}_i). \tag{5}$$

Using all of the above it is found that the discrete form of (1) is

$$(\text{diag}(\boldsymbol{\zeta}) - \mathbf{F})\mathbf{p} + s(\alpha)k\mathbf{G}\mathbf{v_n} = \mathbf{H}\mathbf{p} + s(\alpha)k\mathbf{G}\mathbf{v_n} = \mathbf{0}, \tag{6}$$

References

[1] M. Berggren, A. Bernland, and D. Noreland, "Acoustic boundary layers as boundary conditions," *Journal of Computational Physics*, vol. 371, pp. 633–650, 2018.

[2] M. Brunaeu, P. Herzog, J. Kergomard, and J. Polack, "General formulation of the dispersion equation in bounded visco-thermal fluid, and application to some simple geometries," *Wave Motion*, vol. 11, no. 5, pp. 441–451, 1989.

[3] J. Bezanson, A. Edelman, S. Karpinski, and V. B. Shah, "Julia: A fresh approach to numerical computing," *SIAM review*, vol. 59, no. 1, pp. 65–98, 2017.

[4] V. Cutanda Henríquez and P. R. Andersen, "A three-dimensional acoustic boundary element method formulation with viscous and thermal losses based on shape function derivatives," *Journal of Computational Acoustics*, vol. 26, no. 3, 2018.

[5] T. Askham, Z. Gimbutas, L. Greengard, L. Lu, J. Magland, D. Malhotra, M. O'Neil, M. Rachh, and V. Rohklin, "FMM3D."

Reduced Order Series Expansion Boundary Element Method (ROSEBEM)

The Boundary Layer Impedance (BLI) boundary condition can be used to approximate the viscothermal losses

$$\frac{\partial p}{\partial \mathbf{n}}(\mathbf{x}) = \left[(\gamma - 1)\frac{\mathrm{i}k^2}{k_h} - \frac{\mathrm{i}\Delta^\parallel}{k_v}\right]p(\mathbf{x}). \tag{7}$$

Inserting the above into (1) and doing some manipulation it follows that

$$\begin{aligned}\zeta(\mathbf{x})p(\mathbf{x}) = &\int_{\Gamma_\mathrm{N}} G(\mathbf{x},\mathbf{y})\frac{\partial p(\mathbf{y})}{\partial \mathbf{n}(\mathbf{y})}\ \mathrm{d}S_\mathbf{y} - \int_\Gamma \frac{\partial G(\mathbf{x},\mathbf{y})}{\partial \mathbf{n}(\mathbf{y})}p(\mathbf{y})\ \mathrm{d}S_\mathbf{y} \\ &+ \frac{(\gamma-1)\mathrm{i}k^2}{k_h}\int_{\Gamma_\mathrm{BLI}} G(\mathbf{x},\mathbf{y})p(\mathbf{y})\ \mathrm{d}S_\mathbf{y} + \frac{\mathrm{i}}{k_v}\int_{\Gamma_\mathrm{BLI}} \nabla^\parallel_\mathbf{y} G(\mathbf{x},\mathbf{y}) \cdot \nabla^\parallel_\mathbf{y} p(\mathbf{y})\ \mathrm{d}S_\mathbf{y},\end{aligned} \tag{8}$$

A drawback of the BEM is that the discrete form of (6) depends on the frequency, meaning that for every frequency of interest it needs to be recomputed. A simple Taylor expansion of the Green's function can alleviate this pain as it transforms the discrete form into[6]

$$\left(\sum_{m=0}^{M}\frac{(k-k_0)^m}{m!}\left[\square + \frac{(\gamma-1)\mathrm{i}k^2}{k_h}\square + \frac{\mathrm{i}}{k_v}\square\right]\mathbf{p} + \square\,\partial_n\mathbf{p}\right) = \mathbf{0} \tag{9}$$

This approach, however, increase memory usage due to the storing of M-times the matrices. To solve this a reduced basis can be used, i.e. introduce $\mathbf{U}_\ell$ such that

$$\mathbf{p} \approx \mathbf{U}_\ell\mathbf{p}_\ell, \quad \frac{\partial \mathbf{p}}{\partial \mathbf{n}} \approx \mathbf{U}_\ell\frac{\partial \mathbf{p}_\ell}{\partial \mathbf{n}}, \quad \left(\,| = \square\,|\,\right) \tag{10}$$

As such (9) becomes

$$\left(\sum_{m=0}^{M}\frac{(k-k_0)^m}{m!}\left[\square + \frac{(\gamma-1)\mathrm{i}k^2}{k_h}\square + \frac{\mathrm{i}}{k_v}\square\right]\mathbf{p}_\ell + \square\,\partial_n\mathbf{p}_\ell\right) = \mathbf{0}. \tag{11}$$

In many practical cases storing all the matrices from the above requires about the same memory as the original BEM system from (8).

Results

The results shows that the model accurately describes the pressure while decreasing the computational efforts by 100x using only 1.65x the memory.

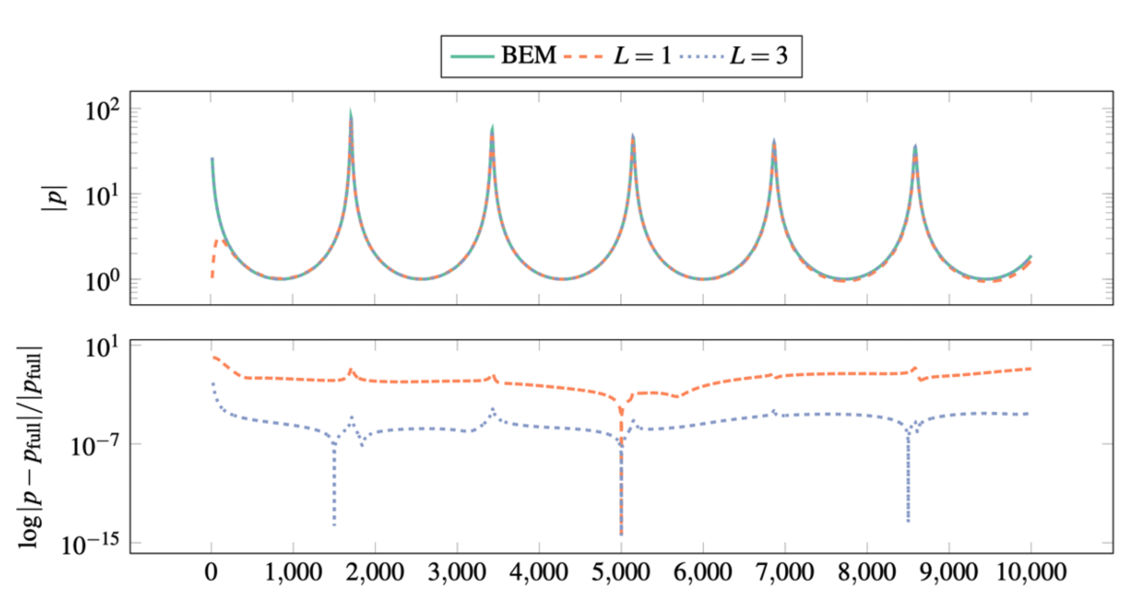
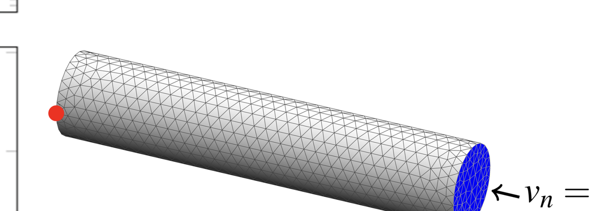


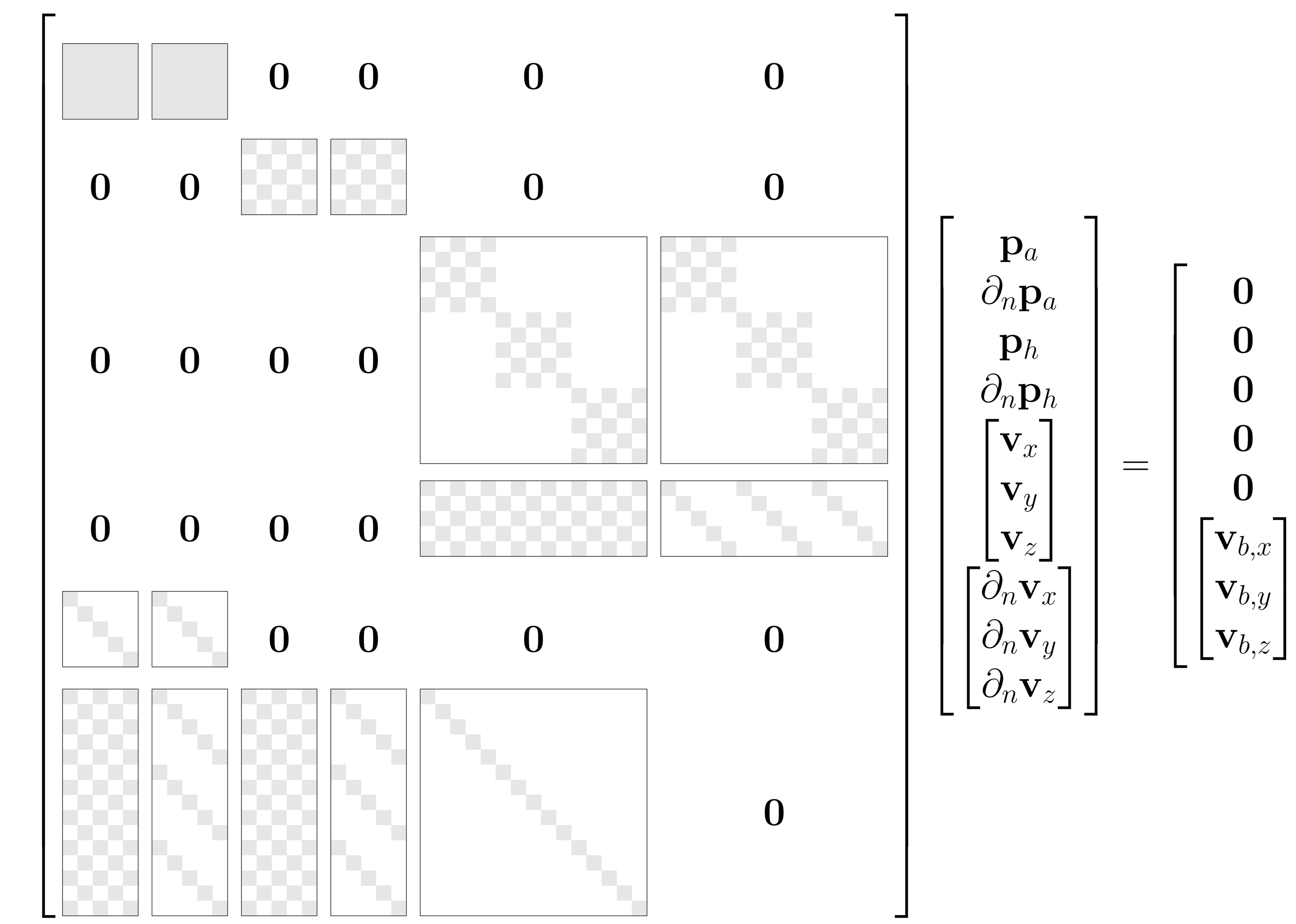
Table 1. Computation efforts: Full BEM vs. ROSEBEM.

	Full BEM	ROSEBEM
Computational Time	6.914 min	65 min
Memory Consumption	224.297 MB	370.789 MB



Improvements in the full formulation

Visually the new formulation looks as



The acoustical pressure can be found by solving the following

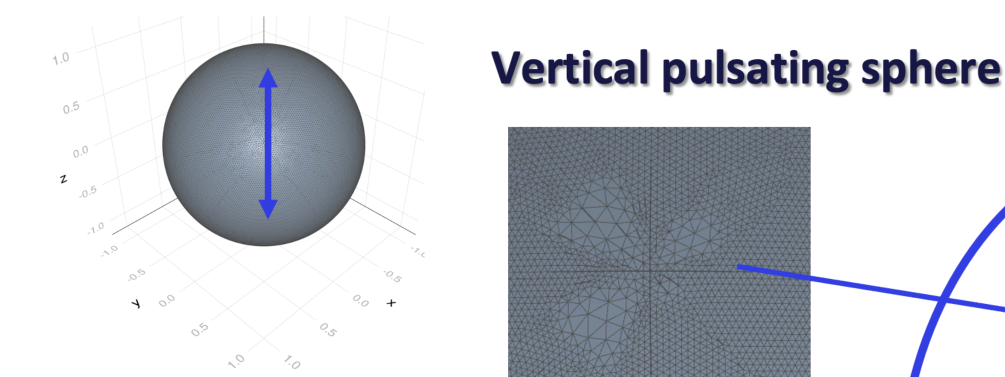
$$\left[\mathbf{G}_a\left(\mu_a(\mathbf{RN})^{-1}\mathbf{RD}_c + \mu_h\mathbf{G}_h^{-1}\mathbf{H}_h\right) - \phi_a\mathbf{H}_a\right]\mathbf{p}_a = \mathbf{G}_a(\mathbf{RN})^{-1}\mathbf{R}\mathbf{v}_s. \tag{12}$$

Note that in (12) every matrix except $\mathbf{G}_a$ and $\mathbf{H}_a$ are *sparse*. However, both matrices can be approximated using an acceleration method such as e.g. the Fast Multipole Method or the H-Matrix approach by approximating the multiplication[4]

$$\left(\underbrace{\int_\Gamma G_a(\mathbf{t}_k,\mathbf{y})\mathbf{T}(\mathbf{y})\ \mathrm{d}S_\mathbf{y}}_{k\text{th row of }\mathbf{G}}\right)\mathbf{z} \approx \underbrace{\left[G_a(\mathbf{t}_k,\mathbf{y}_1)\ G_a(\mathbf{t}_k,\mathbf{y}_2,)\ \dots\ G_a(\mathbf{t}_k,\mathbf{y}_{NQ})\right]}_{\text{multiplication with this can be accelerated}}\mathbf{C}\mathbf{z}, \tag{13}$$

where $\mathbf{C}$ is a sparse matrix.

Results



Storing the systems (solving not included)

No. DOFs	13k	39k	69k	154k	323k
10-Variable	254 GB	2323 GB	7102 GB	35394 GB	156132 GB
1-variable Dense	**6.1 GB**	**60 GB**	183 GB	909 GB	4000 GB
1-variable FMM	**0.054 GB**	**0.16 GB**	**0.29 GB**	**0.64 GB**	**1.16 GB**