The Journal of Open Source Software

NGSBEM: potential operators for FEM-BEM couplings in NETGEN/NGSolve

Rafael Dorigo [1*], **Joachim Schöberl** [1*], and **Lucy Weggler** [2*¶]

1 TU Wien, Austria **2** HTW Berlin, Germany ¶ Corresponding author * These authors contributed equally.

DOI: 10.xxxxxx/draft

Software

- Review
- Repository
- Archive

Editor:

Submitted: 03 December 2025
Published: unpublished

Summary

NGSBEM is the dedicated boundary element method (BEM) module within NETGEN/NGSolve. It provides the full set of tools needed to formulate and solve boundary integral equations, including potential operators and their Galerkin discretizations. Seamlessly integrated with the NGSolve core, NGSBEM enables efficient numerical solutions of linear interior and exterior boundary value problems based on the BEM.

Key features of NGSBEM, in conjunction with NGSolve, include:

- Use of finite element spaces already available in NGSolve, including high-order elements on curved surfaces.
- A user-friendly Python interface.
- Declarative programming for problem formulation and solution.
- Standard potential operators for electrostatics, acoustics, and electromagnetics, allowing both scalar and vector-valued densities to be discretized.
- Support for multi-domain problems through FEM–BEM coupling.
- Acceleration via a Multi-Level Fast Multipole Method (MLFMM) for approximating the arising Galerkin matrices.
- Kernel-driven implementations of potential operators, enabling users to add custom operators.
- Extensive documentation and demonstrations supporting users as they start working with.

Statement of Need

Computational engineering frequently involves solving boundary value problems (BVPs), including linear and nonlinear partial differential equations, as well as interior, exterior, and transmission problems. While the finite element method (FEM) excels at handling interior problems with nonlinearities, the BEM is the method of choice for solving linear second-order problems in unbounded domains.

A unified framework that naturally combines both approaches is rare. NETGEN/NGSolve addresses this gap by providing a complete environment for Galerkin BEM and its coupling with FEM.

Usability, Accuracy, and Efficiency

The BEM is attractive due to its dimensional reduction and high accuracy. However, it also introduces challenges, such as singular integral operators and dense matrices. NGSBEM is designed to resolve these difficulties internally, providing robust and efficient implementations while sparing users from the underlying complexity.

Usability

NGSolve follows the philosophy of allowing users to "type formulas as they appear in the book." NGSBEM extends this philosophy to the BEM: users express boundary integral operators declaratively in Python, in a syntax that mirrors their mathematical formulation. This clarity and directness in representing operators is a central design goal.

Consider, for example, the Dirichlet problem

$$\begin{cases} \Delta\phi = 0, & \text{in } \Omega, \\ \gamma_0\phi = m, & \text{on } \Gamma. \end{cases}$$

The representation formula yields:

$$\phi(\mathbf{x}) = \mathrm{LaplaceSL}(j)(\mathbf{x}) - \mathrm{LaplaceDL}(m)(\mathbf{x})\,. \tag{1}$$

To compute the unknown Neumann data j, one solves the Galerkin discretization of the single layer operator. With trial functions u_j and test functions v_i, the matrix entries are

$$V_{ij} = \int_\Gamma \left[\mathrm{LaplaceSL}(u_j)\right] v_i\, ds.$$

In NGSBEM, the matrix is assembled as:

```
fesL2 = SurfaceL2(mesh, order=3, dual_mapping=True)
u, v = fesL2.TnT()
V = LaplaceSL(u * ds) * v * ds
```

Given V, the linear system

$$Vj = b$$

is solved using a standard iterative solver.

A second example illustrates the hypersingular operator arising for the Neumann problem. Its Galerkin matrix is given by

$$D_{ij} = \int\limits_\Gamma \left[\mathrm{LaplaceSL}(\mathbf{curl}_\Gamma u_j)\right] \mathbf{curl}_\Gamma v_i(\boldsymbol{x})\,\mathrm{d}s_x\,.$$

implemented simply as

```
D = LaplaceSL(curl(u) * ds) * curl(v) * ds
```

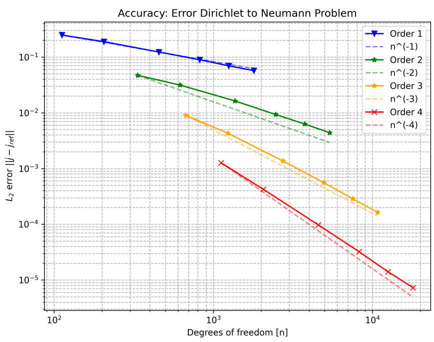
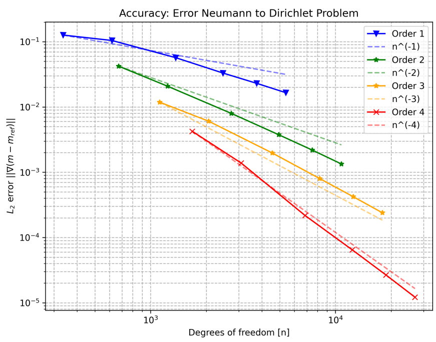
The linear system with Galerkin matrix D is solved using a standard iterative solver.

Single and double layer operators for the Helmholtz and Maxwell equations are provided analogously.

Accuracy

Accurate integration is central to the BEM, particularly because singular kernels must be treated with care. NGSBEM employs the specialized quadrature rules of Sauter and Schwab (Sauter & Schwab, 2011), which support curved elements and enable high-order convergence.

The numerical experiments below compare the computed Cauchy data with the known exact solution for the Laplace equation. The results demonstrate optimal convergence rates.

Laplace Dirichlet Problem	Laplace Neumann Problem
	

Beyond scalar-valued problems, NGSBEM supports vector finite elements and BEM operators for electromagnetics, with numerical evidence of optimal convergence documented in (Weggler, 2025).

Another challenge is the accurate evaluation of potentials, especially in the near field. NGSBEM implements techniques developed by Gumerov and others (Nail A. Gumerov & Duraiswami, 2021; Kaneko et al., 2023). Depending on the geometric configuration, triangles are either subdivided or a line-based approximation is applied, ensuring stable and accurate potential evaluation.

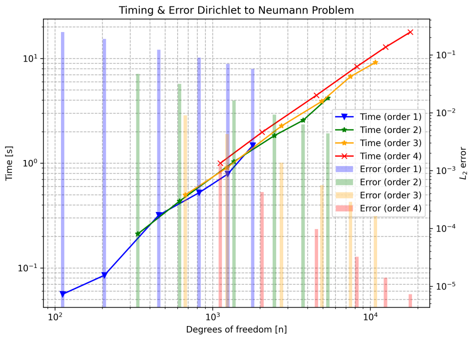
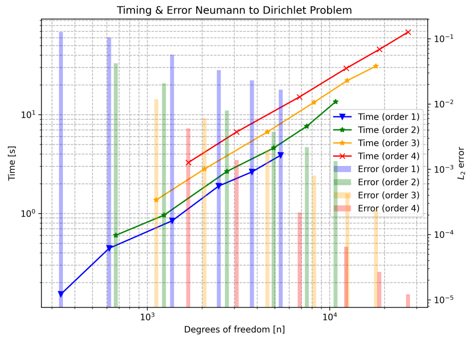
Efficiency

Galerkin matrices are generated on the fly using the multi-level fast multipole method (MLFMM) of Greengard and Rokhlin (Greengard & Rokhlin, 1987), with efficient computation of translation and rotation coefficients following the work of Gumerov and Duraiswami (Dorigo, 2025; Nail A. Gumerov & Duraiswami, 2001; Nail A. Gumerov & Duraiswami, 2015).

Performance is further enhanced by:

- the use of AVX vectorization,
- full parallelization across all major computational stages.

To assess efficiency, we compare accuracy and runtime across different polynomial orders. Higher-order elements provide significantly better accuracy at comparable runtime, despite the increased number of unknowns.

Laplace Dirichlet Problem	Laplace Neumann Problem
	

For example, in Laplace Dirichlet Problem with efficiency plot shown on the left:

- order 1 at 1250 DOFs (0.789 s runtime, error 0.07), vs.,
- order 4 at 1120 DOFs (1.000 s runtime, error 0.001)

This corresponds to an error reduction of approximately 98.6% at comparable runtime.

Similar improvements appear across other configurations, demonstrating the practical efficiency of higher-order BEM.

All numerical experiments were run on a MacBook Pro equipped with an Apple M4 Pro (14 cores: 10 performance + 4 efficiency) and 48 GB of RAM. Timings were measured on the host machine under macOS.

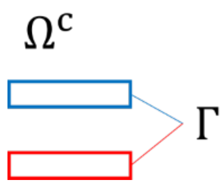
Example

Solving a PDE with the BEM proceeds in two steps:

1. solve an integral equation for the unknown boundary trace(s),
2. evaluate the representation formula to obtain the solution in the domain.

Consider a plate capacitor:

$$\begin{aligned} -\Delta\phi &= 0, && \text{in } \Omega^c\,, \\ \gamma_0\phi &= m\,, && \text{on } \Gamma\,, \\ \lim_{\|x\|\to\infty} \phi(x) &= \mathcal{O}\left(\frac{1}{\|x\|}\right), && \|x\|\to\infty\,. \end{aligned}$$



Using Equation 1, the unknown Neumann data j is computed via a BEM system:

```
j = GridFunction(fesL2)

# generate potential operators V, M and K
with TaskManager():
    SLPotential = LaplaceSL(u*ds(bonus_intorder=3))
    DLPotential = LaplaceDL(uH1*ds(bonus_intorder=3))
    V = SLPotential*v*ds(bonus_intorder=3)
```

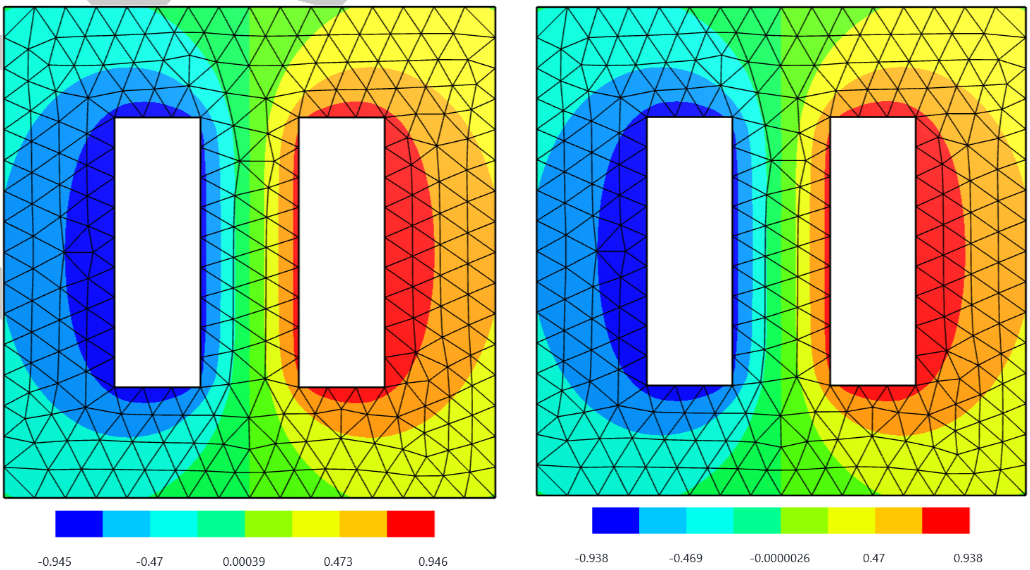
```
    K = DLPotential*v*ds(bonus_intorder=3)
    M = BilinearForm(uH1*v*ds(bonus_intorder=3)).Assemble()

# solve the linear system V j = (-1/2 M + K) m
with TaskManager():
    pre = BilinearForm(u.Trace()*v.Trace()*ds, diagonal=True).Assemble().mat.Inverse()
    rhs = ((-0.5 * M.mat + K.mat) * m).Evaluate()
    CG(
      mat = V.mat, pre=pre, rhs = rhs, sol=j.vec, tol=1e-8, maxsteps=200,
      initialize=False, printrates=False
      )
```

Afterwards, the potential ϕ can be evaluated anywhere. For example, on a screen cutting through the plates:

```
screen = WorkPlane(
  Axes((0,0,0), X, Z)).RectangleC(4, 4).Face()
  - Box((-1.1,-1.1,0.4), (1.1,1.1,1.1))
  - Box((-1.1,-1.1,-1.1), (1.1,1.1,-0.4)
  )
mesh_screen = Mesh(OCCGeometry(screen).GenerateMesh(maxh=0.25)).Curve(1)
fes_screen = H1(mesh_screen, order=3)
phi_on_screen = GridFunction(fes_screen)
print ("ndofscreen=", fes_screen.ndof)
with TaskManager():
    phi_on_screen.Set(
      -SLPotential(j)+DLPotential(utop)+DLPotential(ubot),
      definedon=mesh_screen.Boundaries(".*"), dual=False
      )
Draw (gf_screen)
```

The BEM solution (left) and a FEM reference solution (right) are shown here:

BEM Solution ϕ	FEM Solution ϕ_{fem}



Because the problem is exterior, the FEM requires either infinite elements or an artificial boundary with prescribed data. The BEM, in contrast, handles the unbounded domain naturally and without mesh refinement at edges or corners. The maximum discrepancy between FEM and BEM is around 2.5%, occurring near corner singularities.

Further examples, including FEM–BEM coupling, mixed formulations, and acoustic or electromagnetic scattering, can be found in the online documentation (Weggler, 2025).

Details on the theoretical background for high order BEM for electrostatics and electromagnetics, can be found in (Weggler, 2011, 2012).

References

Dorigo, R. (2025). *High order boundary element methods accelerated by the fast multipole method in NGSolve* [Diploma Thesis, Technische Universität Wien]. https://doi.org/10.34726/hss.2025.136548

Greengard, L., & Rokhlin, V. (1987). A fast algorithm for particle simulations. *Journal of Computational Physics*, *73*(2), 325–348. https://doi.org/10.1016/0021-9991(87)90140-9

Gumerov, Nail A., & Duraiswami, R. (2001). *Fast, exact, and stable computation of multipole translation and rotation coefficients for the 3-d helmholtz equation*. https://drum.lib.umd.edu/items/f0c54406-2abe-4416-b35f-1d4f669b8c50

Gumerov, Nail A., & Duraiswami, R. (2015). Recursive computation of spherical harmonic rotation coefficients of large degree. In R. Balan, M. J. Begué, J. J. Benedetto, W. Czaja, & K. A. Okoudjou (Eds.), *Excursions in harmonic analysis, volume 3: The february fourier talks at the norbert wiener center* (pp. 105–141). Springer International Publishing. https://doi.org/10.1007/978-3-319-13230-3_5

Gumerov, Nail A., & Duraiswami, R. (2021). *Analytical computation of boundary integrals for the helmholtz equation in three dimensions*. https://arxiv.org/abs/2103.17196

Kaneko, S., Gumerov, N. A., & Duraiswami, R. (2023). *Recursive analytical quadrature of laplace and helmholtz layer potentials in* $\mathbb{R}^3$. https://arxiv.org/abs/2302.02196

Sauter, S., & Schwab, C. (2011). *Boundary element methods* (Vol. 39). Springer. https://doi.org/10.1007/978-3-540-68093-2

Weggler, L. (2011). *High order boundary element methods* [Dissertation]. Saarland University. https://publikationen.sulb.uni-saarland.de/bitstream/20.500.11880/26312/1/thesis_weggler_final_6.1.12.pdf

Weggler, L. (2012). Stabilized boundary element methods for low-frequency electromagnetic scattering. *Mathematical Methods in the Applied Sciences*, *35*(5), 574–597. https://doi.org/10.1002/mma.1597

Weggler, L. (2025). *NGSBEM docu* [Online Documentation]. GitHub Pages. https://weggler.github.io/docu-ngsbem/intro.html