Validation of open source code BEM++ for simulation of acoustic problems

P. Lukashin (ISP RAS, Russia), S. Strijhak, (ISP RAS, Russia), G. Shcheglov (BMSTU, Russia)
Aeroacoustic sources

A rotating 7-bladed propeller on submarine

Compressor stator test rig

Jet’s engines of Launch abort vehicle

Car’s outside rear-view mirror
## Sound-pressure levels

<table>
<thead>
<tr>
<th>(dB(A))</th>
<th>Common sounds</th>
</tr>
</thead>
<tbody>
<tr>
<td>30</td>
<td>whisper</td>
</tr>
<tr>
<td>50</td>
<td>rainfall, quiet office, refrigerator</td>
</tr>
<tr>
<td>60</td>
<td>dishwasher, normal conversation</td>
</tr>
<tr>
<td>70</td>
<td>traffic, vacuum cleaner, restaurant</td>
</tr>
<tr>
<td>80</td>
<td>alarm clock, subway, factory noise</td>
</tr>
<tr>
<td>90</td>
<td>electric razor, lawnmower, heavy truck or road drill at 7 m</td>
</tr>
<tr>
<td>100</td>
<td>garbage truck, chain saw, stereo system set above halfway mark</td>
</tr>
<tr>
<td>110</td>
<td>rock concert, power saw</td>
</tr>
<tr>
<td>120</td>
<td>jet takeoff, nightclub, thunder</td>
</tr>
<tr>
<td>130</td>
<td>jack hammer</td>
</tr>
<tr>
<td>140</td>
<td>shotgun, air raid system</td>
</tr>
<tr>
<td>180</td>
<td>rocket-launching pad</td>
</tr>
</tbody>
</table>
Definition of acoustic’s problem and boundary conditions
### The main approaches for numerical modelling in aeroacoustics

<table>
<thead>
<tr>
<th>Approach</th>
<th>Numerical method</th>
<th>Accuracy</th>
<th>Required resources / Applicability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Solution of «full» Navier – Stokes equations</td>
<td>Direct Numerical Simulation (DNS)</td>
<td>Fine</td>
<td>Very much / Impossible</td>
</tr>
<tr>
<td>Large Eddy Simulation (LES)</td>
<td>Good</td>
<td>Very much / Impossible</td>
<td></td>
</tr>
<tr>
<td>Unsteady Reynolds-Averaged Navier – Stokes (URANS)</td>
<td>Low</td>
<td>Few / Possible but not reliable</td>
<td></td>
</tr>
<tr>
<td>CFD-CAA CFD-ALE CFD-APE</td>
<td>Linearized Euler Equation (LEE) or Acoustic Perturbation Equation (APE)</td>
<td>Good</td>
<td>Needs many resources due to grid resolution for LES and acoustics. Applicable for small-scale cases.</td>
</tr>
<tr>
<td>CFD Surface Integral Analytic Methods (Lighthill, Kirchhoff, FW-H)</td>
<td>Hybrid methods (LES/RANS) and analytical methods for far-field modelling</td>
<td>Sufficient</td>
<td>Can not used in cases with reflection of acoustic waves somewhere in considered region</td>
</tr>
<tr>
<td>CFD-BEM</td>
<td>Connection of the CFD solution (LES/RANS) in the near-field and the solution of Helmholtz equation by BEM (Boundary Element Method) in the far-field</td>
<td>Good</td>
<td>Good for large cases to avoid big grids</td>
</tr>
</tbody>
</table>
The BEM++ Library

- Core library in C++, complete interface via Python
- Support for Laplace, Helmholtz, Maxwell
- Shared-Memory parallelization
- H-matrix assembly
- Open source license
- Mac and Linux supported
Introduction to BEM++

BEM++ project:  www.bempp.org
University College London
Classification of acoustic problems

Acoustic wave equation: \( \nabla^2 u - \frac{1}{c^2} \cdot \frac{\partial^2 u}{\partial t^2} = 0, \quad u = f(x,t) \)

\[ u(x,t) = U(x) \cdot e^{-i\omega t} \quad \Rightarrow \]

Helmholtz equation: \( \nabla^2 U + k^2 \cdot U = 0, \quad k = \frac{\omega}{c} \quad – \text{wavenumber} \)

Boundary conditions on \( S \):

\[ \frac{\partial U}{\partial n} - i \beta U = h \]

\[ \beta = 0 \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad |\beta| >> 1 \]

(Neumann BCs) \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad (Dirichlet BCs)

1) Radiation problem: \( h \neq 0 \)
2) Scattering problem: \( h = 0 \)
Formulation of the test problem

Scattering from the unit sphere of plane wave:

\[
\begin{align*}
\nabla^2 U + k^2 \cdot U &= 0 \\
\frac{\partial U}{\partial n} - i \beta U &= 0 \\
U &= U^S + U^I \\
U^I &= U_0 \cdot e^{ik\vec{a}} \\
\lim_{R \to \infty} [R \left( \frac{\partial U^S}{\partial R} - ikU^S \right)] &= 0
\end{align*}
\]

\[U|_S = 0 \quad \text{(Dirichlet problem)}\]

\[\frac{\partial U}{\partial n}|_S = 0 \quad \text{(Neumann problem)}\]

Source data: \(k, U_0, R, \vec{a}\)
BEM problem

Boundary Integral equation (BIE) for

Dirichlet problem:  
\[
\left( \frac{1}{2} I + T - i\eta S \right) U_n^* = \left( \frac{\partial U^I}{\partial n} - i\eta U^I \right) |_S 
\]

Neumann problem:  
\[
\left( H + i\eta \left( \frac{1}{2} I - K \right) \right) U^* = -\left( \frac{\partial U^I}{\partial n} - i\eta U^I \right) |_S 
\]

\( U_n^* \) and \( U^* \) are surface potentials, which need to be calculated.

\[
T[U](x) = \int_S \frac{\partial G(x, y)}{\partial n(x)} \cdot U(y) dS(y) \\
H[U](x) = \frac{\partial}{\partial n(x)} \int_S \frac{\partial G(x, y)}{\partial n(y)} \cdot U(y) dS(y) \\
S[U](x) = \int_S G(x, y) \cdot U(y) dS(y) \\
K[U](x) = \int_S \frac{\partial G(x, y)}{\partial n(y)} \cdot U(y) dS(y) \\
I[U](x) = \int_S U(x) dS(x) 
\]
BEM solution

Integral equation to derive the solution for

Dirichlet problem:

\[ U = U^I - S'[U_n^*] \]

Neumann problem:

\[ U = U^I + K'[U^*] \]

Single-layer potential operator:

\[ S'[U](x) := \int_S G(x, y) \cdot U(y) dS(y) \]

Double-layer potential operator:

\[ K'[U](x) := \int_S \frac{\partial G(x, y)}{\partial n(y)} U(y) dS(y) \]

Green’s function:

\[ G(x, y) = \frac{e^{ik|x-y|}}{4\pi |x-y|} \]
Discretization of the BIE and solving

Continuous piecewise linear functions for Dirichlet data:

\[ \psi_k(x) = \begin{cases} 1, & x \in \tau_k \\ 0, & x \notin \tau_k \end{cases} \]

Piecewise constant functions for Neumann data:

\[ \phi_j(x_i) = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases} \]

Triangulated surface with elements \( \tau_k \) and nodes \( x_i \)

Solving the system by GMRES or CG

\[
\begin{bmatrix}
a_{11} & a_{12} & \cdots & a_{1N} \\
a_{21} & a_{22} & \cdots & a_{2N} \\
\vdots & \vdots & \ddots & \vdots \\
a_{N1} & a_{N2} & \cdots & a_{NN}
\end{bmatrix}
\begin{bmatrix}
\hat{\lambda}_1 \\
\hat{\lambda}_2 \\
\vdots \\
\hat{\lambda}_N
\end{bmatrix}
= 
\begin{bmatrix}
b_1 \\
b_2 \\
\vdots \\
b_N
\end{bmatrix}
\]
Interface basics of BEM++

<table>
<thead>
<tr>
<th>Implementation in BEM++</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Geometry</strong></td>
</tr>
<tr>
<td>grid = bempp.api.shapes.regular_sphere (n)</td>
</tr>
<tr>
<td><strong>Space function</strong></td>
</tr>
<tr>
<td>space = bempp.api.function_space(grid, &quot;P&quot;, 1)</td>
</tr>
<tr>
<td><strong>BCs</strong></td>
</tr>
<tr>
<td>gr_fun = bempp.api.GridFunction(space, fun=fun)</td>
</tr>
<tr>
<td><strong>Identity operator</strong></td>
</tr>
<tr>
<td>I = bempp.api.operators.boundary.sparse.identity(space, space, space)</td>
</tr>
<tr>
<td><strong>Boundary Integral Operators</strong></td>
</tr>
<tr>
<td>K = bempp.api.operators.boundary.helmholtz.double_layer (space, space, space, k)</td>
</tr>
<tr>
<td>H = bempp.api.operators.boundary.helmholtz.hypersingular (space, space, space, k)</td>
</tr>
<tr>
<td><strong>Potential Operators</strong></td>
</tr>
<tr>
<td>S = bempp.api.operators.potential.helmholtz.single_layer(space, points, k)</td>
</tr>
<tr>
<td>K = bempp.api.operators.potential.helmholtz.double_layer(space, points, k)</td>
</tr>
<tr>
<td><strong>Solving the System</strong></td>
</tr>
<tr>
<td>func.info=bempp.api.linalg.gmres (-H+1j <em>k</em>(0.5*I-K))=gr_fun)</td>
</tr>
<tr>
<td><strong>Deriving the results</strong></td>
</tr>
<tr>
<td>res = np.absolute(u_inc + K.evaluate(func)+1j<em>k</em>S.evaluate(b*func))</td>
</tr>
</tbody>
</table>
Analytical Solution

\( U = U^I + U^S \)

\( U^S = \sum_{l=0}^{n} A_l h_l(kr) P_l(\cos \theta) \)

\( U^I = U_0 \cdot e^{ikx_0} = \sum_{l=0}^{n} (2l + 1)i^l j_l(kr) P_l(\cos \theta) \)

**Dirichlet problem:**

\( U^I + U^S = 0, \text{ at } r = a \)

\( A_l = -U_0 (2l + 1)i^l \frac{j_l(ka)}{h_l(ka)} \)

**Neumann problem:**

\( \frac{\partial U^I}{\partial r} + \frac{\partial U^S}{\partial r} = 0, \text{ at } r = a \)

\( A_l = -U_0 (2l + 1)i^l \frac{lj_{l-1}(ka) - (l + 1)j_{l+1}(ka)}{lh_{l-1}(ka) - (l + 1)h_{l+1}(ka)} \)

- Spherical Bessel function \( j_l \)
- Legendre function \( P_l \)
- Spherical Hankel function \( h_l \)
Compare the results for Dirichlet problem

\[ k = 10, U_0 = 1, R = 1, \vec{a} = \{1; 0; 0\} \]

BEM solution

Analytical solution

OXY plane
Compare the results for Neumann problem

\[ k = 10, U_0 = 1, R = 1, \vec{a} = \{1; 0; 0\} \]

BEM solution

Analytical solution

\[ OXY \text{ plane} \]
Polar plots

\[ k = 10, \quad U_0 = 1, \quad R = 1, \quad \vec{a} = \{1; 0; 0\} \]
Pressure range for a rigid sphere

Point B

Point A

Pressure range at point (-3.0;0;0) for a rigid sphere

Pressure range at point (5*cos45;5*cos45;0) for a rigid sphere
Simulation for prototype of Launch Vehicle

CPU Time

Mesh size: 56500 elements

Pressure plot

\[ f = 100 \, \text{Hz} \]
\[ \dot{a} = \{1; 1; -1\} \]
\[ \frac{\partial U}{\partial n} \big|_{s = 0} \]
Conclusions

• Successful validation of the BEM++ open source library on test case «Sphere»
• The pressure error is less than 5% compared to analytical solution
• Shared memory technology restricts efficiency of solver: sufficiently time of computation for industrials problems with large meshes is possible for low frequencies (~100 Hz)