Numerical Acoustics – Spring 2025

Exercise 6: Near-singular integrals in BEM

In this demo exercise the result of research related to the BEM is showcased. The ZIP file containing the Matlab files also includes a relevant journal paper on the topic.

Sometimes the numerical integrals performed in the Boundary Element Method become *near-singular*. As explained in the theory, this happens because either a collocation point or a field point is placed very close to the element to be integrated, in relation to the element dimensions. If the near-singularity is strong, the standard Gauss numerical integration may not give good results.

In OpenBEM, a solution based on element subdivision that depends on the relative distance of the point to the element has been implemented. In this exercise, this implementation will be observed in OpenBEM, 3D version.

1. Download the exercise files and open them with Matlab. Run the *run_me* path definition file. Open the script *test_sphere_NUAC*. In this file, the scattering of a plane wave by a sphere is calculated with BEM and compared with an analytical solution. Run the script. The sound pressure is calculated on the surface as a function of the angle with the Z axis, and also on an arc of field points very close to the surface. The results show good agreement in both cases.

2. This BEM code deals with near-singular integrals, using interval division. Disable this feature by turning the flag *nsingON* to zero in line 14 of the script. Run again and observe the result. Which result fails now? Play with the position of the field points (line 9), making them closer to the surface, and the near-singular tolerance (line 10).

3. Let us have a closer look at the subdivision technique. The integrals in the 3D implementation are made over surface elements. The functions *nsing2dQUAD* and *nsing2dTRIA* produce the integration points and weights of the divided element for quadrilateral and triangular elements. If asked to, they can also plot the points. For example: **`nsingIP=nsing2dTRIA(4,[0.1 0.1 0.0001],[1 0 0 1;0 1 0 1;0 0 0 1],1e-6,1);`** will call the function for a triangle situated on *[1 0 0; 0 1 0; 0 0 0]*. Run **`help nsing2dTRIA`** and find out what the input values mean. Interpret the plot. Change the inputs and get other results.

4. Repeat section 3 for quadrilateral elements (*nsing2dQUAD*): **`nsingIP=nsing2dQUAD(4,[0.1 0.1 0.0001],[1 0 0 1;1 1 0 1;0 1 0 1;0 0 0 1],1e-6,1);`**

5. Open either *nsing2dQUAD* or *nsing2dTRIA*. Try to understand how the function works. Observe that it contains a sub-function that is run recursively (it calls itself) until some conditions are reached.

6. You may try other geometries, such as close meshes (as in narrow gaps) and thin structures.

Vicente Cutanda Henriquez - 02-2025