A spectral boundary element method

A. Calaon, R. Adey & J. Baynham
Wessex Institute of Technology, Southampton, UK

Abstract

The Boundary Element Method (BEM) is not local and generates a full and non-symmetric matrix, therefore the matrix solution time could easily grow beyond acceptable limits. Furthermore, in many cases a decoupling between geometry description and solution approximation would be advantageous and desirable. This work aims at offering the BEM a new approximation capability, which could reduce the number of degrees of freedom (dofs) needed, and at providing the desired decoupling. The method proposed here takes advantage of the less restrictive requirements in BEM compared with FEM for the field approximation. The base of the new approximation becomes what has been called “Spectral Element” (SE): a super-entity gathering many elements, which are used only for geometry description and numerical integration. Spectral shape functions are generated using smooth positive definite compact supported Radial Basis Functions (RBF). The method was applied to linear elasticity in 3D and numerical tests were performed for different typical cases in order to highlight advantages, disadvantages and limits of the proposed methodology.

1 Approximation field in the BEM

1.1 Some general properties of the “finite element approximation”

Both BEM and FEM, in order to reduce the infinite dimensional space of the solution to a finite one, use the so called finite transform. If \( \mathbf{x} \) represents the vector of space coordinates and \( \mathbf{u} \) the approximation field, a set of “generalised degrees of freedom” in the form of a vector \( \alpha \) must be chosen so that:

\[
\mathbf{u}(\mathbf{x}) = \mathbf{A}(\mathbf{x})\alpha
\]
In equation (1) \( A(x) \) represents the coordinate function matrix, which contains functions fixed in space. The finite elements appear when the continuum under analysis is ideally cut into “small” pieces, the elements of the mesh. In order to ensure that the values of the field in few specific points (nodes) belonging to an element govern the whole behaviour of \( u \) inside it, equation (1) must be imposed simultaneously in each of the \( n \) nodes of an element:

\[
\begin{bmatrix}
  u_1(x) \\
  u_2(x) \\
  \vdots \\
  u_n(x)
\end{bmatrix} = U = C\alpha =
\begin{bmatrix}
  A(x_1) \\
  A(x_2) \\
  \vdots \\
  A(x_n)
\end{bmatrix} \alpha
\]  

(2)

Now if \( C \) is square, i.e. if the number of nodes \( n \) is equal to the number of \( \alpha \) dofs, and is not singular, the following equations hold:

\[ u(x) = A(x)\alpha = A(x)C^{-1}U = \Phi(x)U \quad \text{where} \quad \Phi = A(C)^{-1} \]  

(3)

Generally the \( \Phi \) matrix is called shape matrix. It interpolates the dof of the element, represented by the nodal values, in between the nodes. In the FEM four general properties are needed for the functions in \( \Phi \), called shape functions, each of which belongs to a different node:

1. **Completeness condition**: in order to be able to represent a constant field, (first term in a Taylor series development of any field) the sum of all functions must be one in every point of the element:

   \[ \sum_{i=1}^{n} \phi_i = 1 \quad \forall x \in \text{Element} \]  

   (4)

2. **Interpolation condition**: in order to have “orthogonal” dofs, every function \( \phi \) must be zero at all nodes but its own:

   \[ \phi_i(x_j) = \delta_{ij} \]  

   (5)

3. **Local support condition**: the single \( i \) function vanishes along any element boundary that does not include node \( i \)

4. **Inter-element compatibility condition**: The functions satisfy \( C^0 \) continuity between adjacent elements on any element boundary that includes the node \( i \).

The last two properties are not needed by the BEM for the convergence of the method, and are more related to the mesh rather than the element alone. In fact first numerical approaches in BEM used constant elements, which violate requirements (3) and (4).

From the practical point of view equation (2) isn’t normally used since a set of polynomials with the desired properties is considered to represent directly the \( \Phi \) matrix, avoiding also the problem of dealing with an inverse matrix. Every nodal dof is kept independent from the others (property 2) using orthogonal polynomials.
2 Interpolation error

The interpolation error of a polynomial approximation depends critically on the degree \( p \) of the polynomials used. In fact the following formula holds (see for example [1]):

\[
\| \mathbf{u}(x) - \Pi_h^p \mathbf{u}(x) \|_{H^k(I)} \leq C h^{p+1-k} |\mathbf{u}(x)|_{H^{p+1}(I)},
\]

where \( C \) is a constant, \( I \) is the domain of definition of \( \mathbf{u}(x) \), \( h \) represents the maximum dimension of the elements in the mesh, \( \Pi_h^p \) is the polynomial approximation of degree \( p \) and mesh parameter \( h \), and \( H^k(I) \) is the Sobolev Space of index \( p \).

From this formula it appears clear why, in principle, the so called \( p \)-refinement is more powerful than the \( h \) one. If it were possible to increase \( p \) letting the element grow over the whole domain the approximation would become spectral, therefore faster than any other polynomial order. The error would then be so bounded:

\[
\exists \gamma > 0 : \| \mathbf{u}(x) - \tilde{\mathbf{u}}_N(x) \|_{H^k(I)} \leq C e^{-\gamma N},
\]

where \( \tilde{\mathbf{u}}(x) \) is the approximation and \( N \) the number of dofs. In the following an approach is presented to provide BEM with this powerful convergence property.

3 Proposed approach

3.1 RBF for generating approximation functions

The RBF approximation method is based on the following interpolant:

\[
\begin{bmatrix}
\mathbf{u}(x)_1 \\
\mathbf{u}(x)_2 \\
\vdots \\
\mathbf{u}(x)_n
\end{bmatrix}
= \begin{bmatrix}
\varphi(\|x_1 - x_1\|) & \varphi(\|x_1 - x_2\|) & \cdots & \varphi(\|x_1 - x_n\|) \\
\varphi(\|x_2 - x_1\|) & \varphi(\|x_2 - x_2\|) & \cdots & \varphi(\|x_2 - x_n\|) \\
\vdots & \vdots & \ddots & \vdots \\
\varphi(\|x_n - x_1\|) & \varphi(\|x_n - x_2\|) & \cdots & \varphi(\|x_n - x_n\|)
\end{bmatrix}
\begin{bmatrix}
\alpha_1 \\
\alpha_2 \\
\vdots \\
\alpha_n
\end{bmatrix}
= \mathbf{C} \alpha,
\]

where \( \varphi \) could be any function, but in 1995 Wendland [2] found a way to retrieve the polynomial \textit{compactly supported positive definite} ones of minimal degree for any continuity and embedding space dimension. The one used in this work was:

\[
\varphi = (1 - r)^8 (32 r^3 + 25 r^2 + 8 r + 1) \quad \text{with} \quad r = \|x_i - x_0\|,
\]

which builds a \( C^6 \) field (the point with lower continuity is the origin) and a numerically invertible \( \mathbf{C} \) matrix.

Equation (1) corresponds to equation (3.1), where the base change from \( \alpha \) to \( \mathbf{u} \) described by equation (3) was introduced. In this case equation (3) becomes

\[
\Phi(x) = \begin{bmatrix}
\varphi(\|x - x_1\|) & \varphi(\|x - x_2\|) & \cdots & \varphi(\|x - x_n\|)
\end{bmatrix} \mathbf{C}^{-1}.
\]

So now the field \( \mathbf{u} \) is based on a set of “semi-classical” shape functions with properties 1 (in the limit of an infinite number of nodes or an infinite RBF support)
and 2: just what is strictly needed by the BEM. The choice of location and number of nodes is now arbitrary and completely independent of the geometrical description, permitting an approximation field based on a super-entity composed of many elements. This “Spectral Element” (SE) has approximation functions extending all over its surface. These functions don’t need mapping, since they are naturally based in the real space and have a degree of continuity depending on the $\varphi$, which can be increased freely. In this approach the normal elements are used only for geometry description and numerical integration. Given that $p$ is no longer restricted to small “finite element type” numbers and the approximation field can have any required smoothness, the approximation error should decrease exponentially with increasing $N$ (here representing the number of nodes on a SE) as described by equation (7).

The scheme described was programmed and used to solve 3D elasticity problems.

3.1.1 Compact support

With Compactly Supported RBFs (CSRBF) it is possible to change the extent of the RBF dividing the distance $\|x - x_i\|$ by the desired support. If the maximum distance between any two SE nodes is bigger than the support the $C$ matrix will be banded, while if not it will remain fully populated. In any case, a larger support corresponds to a higher condition number for $C$. The approximation method proposed, as mentioned before is not able to represent exactly a constant value, but this problem can be reduced by using large supports, since the Wendland $\varphi$ functions offer a plateaux near the origin that can be enlarged. A limit to this strategy is the fact that increasing the support causes the matrix $C$ to become badly conditioned and its inversion starts to give unstable results compromising the whole BEM procedure. A strategy was developed in order to maximise the support as a function of the number of nodes on each SE.

3.2 Integration time with increasing number of nodes on the SEs

An important limitation of the method described is the fact that the integration time grows quadratically with the number of nodes on a SE. In fact, during integration, for each gauss point all the shape functions on a SE must be calculated, and the boundary integration has to be performed for each collocation node. Some possible ways of limiting the growth of the integration time could be using Partition of Unity methods [3] or Fast Multipole Methods [4], but these haven’t been investigated in the present work.

3.3 Superficial stress calculation

Classically in the BEM ([5]) the stress tensor on the boundary surface is computed using traction components and tangential derivatives of displacement. In the case of a RBF displacement approximation, the 2 tangential derivatives can be computed summing directly the derivatives of the $\varphi$ weighted by the solution
coefficient $\alpha$ of equation (3). In the case of the Wendland function tested (9) the continuity of the displacement is $C^6$, therefore generating a $C^5$ tangential derivative field. The stress field calculated will then inherit this smoothness, but only in case of an equally smooth geometric normal field; otherwise any discontinuity in the normal will appear in the stress despite a smoother displacement derivative. In the following only displacement results are presented.

3.4 Nodes near SE boundaries

In order to be able to have SE nodes near every SE boundary a tool was programmed which allows automatic positioning of any number of nodes per element near the SE edges and corners (see fig. 1).

4 Numerical tests

In order to test capabilities and limits of the proposed method, the following two 3D elasticity cases were chosen:

- Cube under tension
- Cantilever beam.

The elastic properties used in the test do not correspond to a real material and are: Young’s modulus $100 \cdot 10^9$, Poisson’s ratio 0.3. In order to reduce end-effects soft springs were applied as restraints. In all tests the SEs have nodes around the boundary and near the corners, as shown in figure 1. This strategy was adopted because it had been shown to give better results in previous tests. The models were solved multiple times in order to assess the convergence rate and to collect data for automatic optimisation of the support as a function of the number of nodes of each
SE. In the first models the RBF support was calculated simply as a linear function of the maximum dimension of the SE:

\[ \text{Support}_i = \gamma d_i, \]  \hspace{1cm} (11)

where \( \gamma \) is a coefficient applied to the whole model and \( d_i \) is the maximum distance centre-periphery on SE \( i \). Afterwards the data collected was used to tune the automatic support calculation, and new calculation series were performed. For the combinations of geometry and boundary conditions used there exist simplified analytical solutions, but the presence of springs, the real 3D geometry and the kind of precision reached made it necessary to compare the results with another numerical solution. This was obtained using the BEM software BEASY (http://www.beasy.com) in series of converging calculations with different boundary conditions, geometries and number of dofs.

4.1 Cube under tension

The geometry of this case is very simple: a unitary cube. On one side is applied a tension of \( 10^8 \) perpendicular to the surface, and on the opposite side the displacement in the direction of the tension is restrained to 0. Perpendicular displacements on the constrained face are restrained through the application of a soft spring. The left part of Figure 2 shows the geometry of this case. Figure 3 shows two series of results obtained using different \( \gamma \) coefficients (see equation (11)). The disposition of SE nodes in the models shown in the graph corresponds to a structural mesh of quadrilaterals on all cube faces, and it doesn’t have nodes on the measuring point, which was chosen as the middle of the cube face with applied traction. The results obtained with the automatic support calculation are plotted with the number of nodes per SE on the abscissa (in this case all SEs in the model have the same number of nodes). The models in the two series shown had a total number of dofs in the range between 90 and 2,592. In both graphs the error enters a phase of exponential convergence, which terminates when the number of nodes per SE grows too high. The same is valid for all other series tested. As expected the number of dofs per SE at which the exponential convergence starts to deteriorate is smaller for bigger
Figure 3: Results for Cube model; error and norm 1 condition number for two different cases ($\gamma$ is 3.5 on the left and 5 on the right).

Figure 4: Results for Cantilever model; error and norm 1 condition number for two different cases ($\gamma$ is 2 on the left and 3.5 on the right).

supports. This trend is common to all the model series investigated and in line with the behaviour expected.

### 4.2 Beam in cantilever

This second model, shown on the right of figure 2, is a cantilever beam with a unit square cross section, a length of 7 and an applied transverse end load of $10^8$. As for the cube case, the numerical solution was taken as reference for the error estimation measured in the centre of the beam free end. In the calculation series performed with BEASY the number of dofs needed for an agreement on the $5^{th}$ digit of the displacement result at the beam end was more than 8,500. Figure 4 shows two SE series results obtained with two different values of $\gamma$. In this case the abscissas refer to the number of nodes on the SEs along the beam sides, which are always higher than on the SEs on the ends. The models of the two series shown had a total number of dofs in the range between 936 and 1,974. Again as in the cube case the convergence looks exponential until a certain number of dofs per SE, and this number decreases with increasing support. Some other model series were performed and the trends remained unchanged.
4.3 Optimising the RBF support

For real practical cases the SE will be of arbitrary shape with varying elements and dofs. A general rule is therefore required to determine the support for any SE in order to achieve an optimal convergence. Therefore, based on the results obtained, a formula was developed to predict the largest support compatible with high accuracy and convergence. The idea is to have a linear dependence of the RBF support from the number of nodes on each SE, clearly scaled by a distance representative of the real geometry $d_{average}$. The equation representing the support coefficient for the SE $i$, with a number of nodes equal to $n$, is then:

$$\gamma_i = d_{average} \beta_i \quad \text{where} \quad \beta_i = n_i a + b.$$  \hfill (12)

The definition of $d_{average}$ used was:

$$d_{average} = \frac{d_{min} + d_{max}}{2} \quad \text{where} \quad d_{min} = \sqrt{\frac{d_{max}^2 - \sqrt{d_{max}^4 - A^2}}{4}}.$$  \hfill (13)

$d_{max}$ is the maximum distance between the centre of the SE and any of the points defining its contour, and $A$ is the SE area. The formula for $d_{min}$ in equation (13) comes from easy geometrical considerations and the fact that a representative rectangular shape was chosen. The only parameters which need to be introduced arbitrarily are $a$ and $b$, respectively the slope and constant of a line in a graph $n - \gamma$. The data collected previously provided the information to obtain the two parameters which draw a line of maximum allowable $\gamma$ (while maintaining high convergence) for each $n$: $a = -1.1667$ and $b = 20.167$.

In addition to the strategy outlined, when the number of nodes is bigger than the number giving $\gamma = 2$, here named $n_{max}$, the calculation of the RBF support switches to a criterion trying to keep the band of $F$ equal to the max number of nodes admitted in case of full matrix, and equation (12) must be substituted with the following one:

$$\gamma_i = \tau \sqrt{\frac{n_{max}}{n_i}}.$$  \hfill (14)

In equation (14) $n_i$ is the assigned number of nodes on SE $i$, higher than $n_{max}$, and $\tau$ a coefficient to be optimised. This equation has almost never been used in the present work, since the number of nodes per SE did’t exceed $n_{max}$ for the models considered.

4.4 New results with optimised support

Figure 5 shows the results obtained using the new calculation of the RBF support described. The same precision is reached for lower number of nodes per SE (compare figures 3 and 4). A decrease of precision is shown in the cube case with 100 nodes per SE, probably due to a slightly too large support for that case.
Figure 5: Results for Error and Norm 1 condition number in the two models: cube on the left and beam on the right.

Figure 6: Slab model geometry with boundary conditions on the left and x displacement on the right.

4.5 A more complex case

In order to test the method on a more challenging geometry involving a stress concentration, the case of a slab with a circular hole was considered. Because of symmetry a quarter of a slab was modelled as shown in figure 6, where the boundary condition and the point where the error was estimated are also shown. In this case the result was tested on a corner, where the displacement, due to the discontinuous nature of the field between SEs, was necessarily calculated as an average of the corner result values on the three surfaces as shown in figure 8. Experience showed that near the edges the SE approximation is weaker, therefore the convergence was not as clean as seen in the previous cases, but the stability of the method was confirmed, and a precision of $10^{-3}$ with 828 nodes was reached and practically maintained until the biggest model solved. In comparison classical BEM models obtained such a precision only using at least 5,000 dofs. Figures 6 (on the right) and 7 show the displacement results for x, y and z directions.
Figure 7: Displacement results for the slab model. y displacement on the left and z displacement on the right.

Figure 8: Displacement and norm 1 condition number results for the slab model.

5 Conclusions

A new approximation method was developed and tested, with the aim of reducing the number of dofs and allow for independent modelling of geometry and solution fields. This method uses compact supported RBFs for generating shape functions which have strictly the necessary properties needed in the BEM. The new shape function field was used for modelling the result only, while the mapping of geometry was left to a classical finite element procedure. The method has been applied to 3D elasticity and three cases have been tested. In principle exponential convergence can be achieved, and parameters have been determined to automatically choose the optimum support necessary.

References


