Asociación Argentina de Mecánica Computacional

Mecánica Computacional Vol XXVI, pp.1174-1193
Sergio A. Elaskar, Elvio A. Pilotta, Germán A. Torres (Eds.)
Córdoba, Argentina, Octubre 2007

TESTS OF PERFORMANCE OF THE BOUNDARY ELEMENT DUAL RECIPROCITY METHOD—MULTI-DOMAIN APPROACH FOR 3D PROBLEMS

Bruno Natalini[a] and Viktor Popov[b],

[a]*Facultad deIngeniería, Universidad Nacional del Nordeste, Av. Las Heras 727, 3500 Resistencia, Argentina, bnatalini_2000@yahoo.com.ar, http://ing.unne.edu.ar/tunel/pagina%20tunel.htm*
[b]*Wessex Institute of Technology, Environmental Fluid Mechanics, Ashurst Lodge, Southampton, SO40 7AA, United Kingdom, viktor@wessex.ac.uk, http://www.wessex.ac.uk*

Keywords: Boundary element method; DRM-MD; Optimal implementation; Radial basis functions; 3D problems.

Abstract. In this paper, aspects regarding implementation of the boundary element dual reciprocity method—multi-domain approach (DRM-MD), in respect to 3D problems are reviewed. Results of numerical tests on a 3D advection–diffusion problem with non-uniform velocity field are presented. The sensitivity of the accuracy and stability of the codes to the continuity of the elements, scaling, internal DRM nodes and mesh refinement have been tested. The results show that scaling is essential and that mesh refinement and/or internal DRM nodes improve the accuracy when the non-homogeneous term of the governing equation becomes dominant. The computer code implemented with discontinuous elements offers higher accuracy, especially for advection dominant transport, but is much slower than the computer code with continuous elements. At the present stage, the discontinuous element code has the advantage of flexibility since it can solve non-homogeneous domains.

1 INTRODUCTION

The dual reciprocity method (DRM), which was introduced by Nardini and Brebbia (1983), is acknowledged to be one of the most effective boundary element method (BEM) techniques for transforming domain integrals into boundary integrals.

The accuracy and stability of the DRM is strongly dependent on the function used in the DRM approximation, for which a variety of interpolation functions can be used. Until the 1990s simple ad hoc expansions such as $1+R$, where R is the distance between a collocation point and a field point, were preferred by users. With the application of the theory of radial basis functions (RBFs) in the context of the DRM the procedure gained a solid mathematical foundation. The origin of RBFs can be traced back when they were applied to geophysical data interpolation. In the 1990s, Golberg and Chen (1994, 1996) introduced the theory of RBFs in the DRM field and demonstrated that the DRM converges if RBFs are used as interpolation function. They pointed out that the early success of $1+R$ as approximation function in the DRM is due to the fact that it belongs to RBFs.

A characteristic feature of the DRM is that it uses a set of internal points to improve its accuracy. The accuracy of the method is sensitive to both the number and distribution of the DRM points. These two factors constitute its main drawbacks, as there are no standard criteria to deal with them. There have been some attempts in using adaptive techniques (Schclar, 1993; Rodríguez and Power, 2002) in order to approach this problem.

The DRM has been demonstrated to be a general and reliable procedure. However, as many of the RBFs used are globally supported the matrix of the resulting system of equations is dense and frequently ill conditioned, when applied to large problems. This makes the method computationally expensive and sometimes unstable. There are two ways to avoid these difficulties: by using compactly supported RBFs (CS-RBFs) or by using domain decomposition.

Positive-definite CS-RBFs, which are locally supported, have been explicitly constructed and applied to multivariate safe reconstruction in mid-1990s by Schaback (1995), Wendland (1995) and Wu (1995). Since then, much effort has been focused on building efficient algorithms using CS-RBFs, as a result of which new functions have been proposed (Wendland, 1995; Buhmann, 2001). The implementation of these functions leads to a sparse system of equation that is free of the problems mentioned above, due to the local support. However, the current stateof-the-art CS-RBFs faces two main difficulties: (a) their accuracy and efficiency depend on the scale of the support, with the scale being uncertain, and (b) the convergence rate of CS-RBFs is low.

Domain decomposition is a technique that is commonly used in the BEM when the domain is piecewise homogeneous. After applying the numerical formulation in every subdomain, the final system of equations is obtained by means of a set of matching conditions in the interfaces between subdomains. The resulting system of equations is not dense, and the sparsity of the system increases with the number of subdomains. Popov and Power implemented a scheme using domain subdivision in conjunction with the DRM to avoid domain integration and called it the dual reciprocity method—multi-domain approach (DRM-MD). The initial problem solved using this formulation was the flow of a mixture of gases through a porous media (Popov, Power and Baldasano, 1998; Popov and Power, 1999a, 2000). The DRM-MD has also been applied to linear and nonlinear advection–diffusion problems (Popov and Power, 1999b), driven cavity flow of Navier–Stokes equations (Florez and Power, 2002a) and of non-Newtonian fluids (Florez and Power, 2002b), and the flow of polymers inside mixers with complex geometries (Florez, 2001). DRM-MD does not suffer the two main problems related to standard DRM; the systems of equations produced by DRM-

MD are sparse and well conditioned, and the number and position of DRM nodes is usually not critical, since small sub-domains usually require no or few interior DRM nodes.

Most of the reported results of the DRM-MD in the literature refer to 2D applications. As the method was extended to 3D problems in 2004 (Natalini and Popov, 2004), presently a few facts are known about 3D implementation, though this is an issue that is far from being settled. For instance, Natalini and Popov (2004, 2006) tested 10 different DRM approximation functions, five globally and five compactly supported RBFs, in codes that solve Poisson and advection–diffusion problems. The highest accuracies were obtained by using compactly supported RBFs. However, a suitable size of the support must be known a priory and there is no rigorous guideline to choose it. It is known that a small size of the support guarantees a safe recovery, but a large one reduces the error at expense of stability. Therefore a balance must be reached between these two factors. This topic has been discussed within the framework of multiscattered interpolation theory by Floater and Iske (1996) and Schaback (1995) among others. Floater and Iske proposed a strategy but it is to be used in a hierarchical scheme for smoothly interpolating scattered data, which cannot be applied in the numerical models presented here or the ones of Natalini and Popov (2004, 2006). Consequently Natalini and Popov concluded that the augmented thin plate splines (ATPS) appear to be the best choice at the moment, as ATPS produced one of the most accurate and certainly the most consistent results without introducing any additional parameter.

On the other hand, Peratta and Popov (2006) applied the DRM-MD in a hybrid formulation to large 3D problems of transport in fractured porous media. Based on comprehensive research by Portapila and Power (2001, 2005) on the application of iterative solvers in 2D DRM-MD codes, they used an iterative solver combined with a MC64 preconditioner. The preconditioner showed to be essential since it reduces the CPU time required by the solver by several orders of magnitude.

In this paper, the sensitivity of the method to factors such as continuity of the elements, scaling, mesh refinement and number of internal nodes is tested for an advection-diffusion problem with variable velocity field, in order to have a better insight into the 3D implementation of the DRM-MD.

2 THE DUAL RECIPROCITY METHOD

Let us consider the following Poisson equation, with the transport coefficient equal to one

$$\nabla^2 u(\boldsymbol{x}) = b(\boldsymbol{x}) \tag{1}$$

where $u(\boldsymbol{x})$ is a scalar field (potential function), $b(\boldsymbol{x})$ the non-homogeneous term, and $\boldsymbol{x}$ the position vector in the domain with components e_i.

Given a point $\boldsymbol{x}$ belonging to a domain Ω, which is enclosed by a contour Γ, the Green integral representation formula for (1) gives the value of u at $\boldsymbol{x}$ in terms of integral equations involving the fundamental solution of the Laplace equation:

$$\lambda(\boldsymbol{x})u(\boldsymbol{x}) + \int_\Gamma q^*(\boldsymbol{x},\boldsymbol{y})u(\boldsymbol{y})d\Gamma_y - \int_\Gamma u^*(\boldsymbol{x},\boldsymbol{y})q(\boldsymbol{y})d\Gamma_y = -\int_\Omega u^*(\boldsymbol{x},\boldsymbol{y})b(\boldsymbol{y})d\Omega_y \tag{2}$$

Here, $u^*(\boldsymbol{x},\boldsymbol{y})$ is the fundamental solution of the Laplace equation given by

$$u^*(\boldsymbol{x},\boldsymbol{y}) = \frac{1}{4\pi}\frac{1}{r} \tag{3}$$

for 3D problems, where r is the distance from the point of application of the concentrated unit

source to any other point under consideration, i.e. $r = |\boldsymbol{x}\text{-}\boldsymbol{y}|$, $q(\boldsymbol{y}) = \partial u(\boldsymbol{y})/\partial n$ and $q^*(\boldsymbol{x},\boldsymbol{y}) = \partial u^*(\boldsymbol{x},\boldsymbol{y})/\partial n$ and n is the unit normal to the boundary of the sub-domain. Notice that in Eq. (2) all the integrals are over the boundary of the domain except for the one corresponding to the term $b(\boldsymbol{y})$, which represents the sum of the non-homogeneous terms. The constant $\lambda(\boldsymbol{x})$ has values between 1 and 0, being equal to 1/2 for smooth boundaries.

To express the domain integral in (2) in terms of equivalent boundary integrals, the DRM approximation is introduced. The basic idea is to expand the $b(\boldsymbol{y})$ term using approximation functions, i.e.:

$$b(\boldsymbol{y}) \cong \widetilde{b} = \sum_{k=1}^{J+I} \alpha_k f(\boldsymbol{y}, \boldsymbol{z}^k) \tag{4}$$

The functions $f(\boldsymbol{y},\boldsymbol{z}^k)$ are approximation functions, which depend only on the geometry of the problem, and the constants α_k are unknown coefficients. The approximation is done at $(J+I)$ nodes, with J boundary nodes around the boundary of the domain and I nodes inside the domain.

Once the DRM approximation of the non-homogeneous term, $b(\boldsymbol{y})$, is implemented, the domain integral can be recast in terms of a series of surface integrals, and one finally arrives at a boundary only integral representation formula

$$\lambda(\boldsymbol{x})u(\boldsymbol{x}) + \int_\Gamma q^*(\boldsymbol{x},\boldsymbol{y})u(\boldsymbol{y})d\Gamma_y - \int_\Gamma u^*(\boldsymbol{x},\boldsymbol{y})q(\boldsymbol{y})d\Gamma_y \cong$$
$$\sum_{k=1}^{J+I}\left\{\alpha_k\left(\lambda(\boldsymbol{x})\hat{u}(\boldsymbol{x},\boldsymbol{z}^k) + \int_\Gamma q^*(\boldsymbol{x},\boldsymbol{y})\hat{u}(\boldsymbol{y},\boldsymbol{z}^k)d\Gamma_y - \int_\Gamma u^*(\boldsymbol{x},\boldsymbol{y})\hat{q}(\boldsymbol{y},\boldsymbol{z}^k)d\Gamma_y\right)\right\} \tag{5}$$

For the numerical solution of the problem, the contour Γ is discretized in Γ_j elements and the density of the integrals in the above equation is defined in terms of nodal values by means of interpolation functions. After application of collocation technique (5) can be written in terms of four matrices, $\mathbf{H}$, $\mathbf{G}$, $\hat{\mathbf{U}}$ and $\hat{\mathbf{Q}}$ which depend only on the geometry of the problem

$$\mathbf{H}\boldsymbol{u} - \mathbf{G}\boldsymbol{q} = (\mathbf{H}\hat{\mathbf{U}} - \mathbf{G}\hat{\mathbf{Q}})\boldsymbol{\alpha} \tag{6}$$

In (6) the vector $\boldsymbol{\alpha}$ is unknown but it can be expressed as $\boldsymbol{\alpha} = \mathbf{F}^{-1}\boldsymbol{b}$, yielding

$$\mathbf{H}\boldsymbol{u} - \mathbf{G}\boldsymbol{q} = (\mathbf{H}\hat{\mathbf{U}} - \mathbf{G}\hat{\mathbf{Q}})\mathbf{F}^{-1}\boldsymbol{b} \tag{7}$$

3 THE DUAL RECIPROCITY METHOD—MULTI-DOMAIN APPROACH

The domain discretization in the BEM is usually used when there are few parts of the domain with different properties. In that case the domain decomposition is often used, in which the original domain is divided into subregions, and on each of them the full integral representation formula are applied. A case of a domain, which is subdivided into four sub-domains, is shown in Figure 1. Though in Figure 1 a 2D domain is considered for the sake of simplicity, the conclusions can be extended to 3D cases as well.

Matching conditions for a potential problem establish that at every node at the interface: (a) the value of the potential is the same for both subdomains. For instance, at node F of Figure 1

$$u_{12}^{F}(\boldsymbol{x}_F) = u_{21}^{F}(\boldsymbol{x}_F) \tag{8}$$

(b) the physical flux is the same for both subdomains

$$\Phi_{12}^{F}(u_{12}^{F}, q_{12}^{F}) = -\Phi_{21}^{F}(u_{21}^{F}, q_{21}^{F}) \tag{9}$$

where the form of the function F depends on the physical problem under consideration.

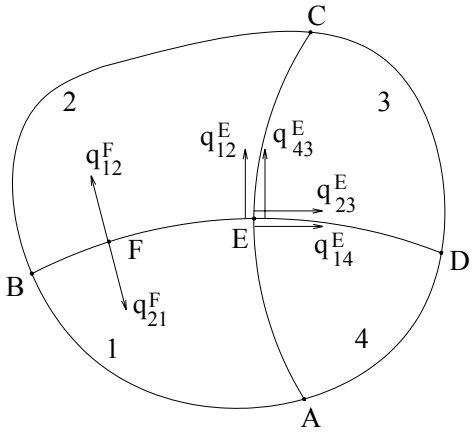

Figure 1: Example of subdivision of a domain into sub-domains.

While the BEM matrices that arise in the single domain formulation are fully populated, the sub-region formulation leads to blocks banded matrix systems with one block for each sub-region and overlaps between blocks when subregions have a common interface. Eq. (10) represents the structure of the system of equations in matrix form that corresponds to domain sub-division shown in Figure 1.

$$\begin{bmatrix} A_1 & A_{12}^{BE} & 0 & A_{14}^{AE} & 0 & 0 & 0 & 0 \\ 0 & A_{21}^{BE} & A_2 & 0 & A_{23}^{CE} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & A_{32}^{CE} & A_3 & A_{34}^{DE} & 0 \\ 0 & 0 & 0 & A_{41}^{AE} & 0 & 0 & A_{43}^{DE} & A_4 \end{bmatrix} \begin{Bmatrix} p_1 \\ p_{12}^{BE} \\ p_2 \\ p_{14}^{AE} \\ p_{23}^{CE} \\ p_3 \\ p_{34}^{DE} \\ p_4 \end{Bmatrix} = \begin{Bmatrix} m_1 \\ m_2 \\ m_3 \\ m_4 \end{Bmatrix} \tag{10}$$

A_j represents the influence coefficients obtained by integration over the external boundary that bounds the sub-domain j and p_i represents the unknown potentials u_j or derivatives q_j at the nodes on this part of the boundary. For example, for the sub-domain 1 the external part of the boundary is given with the curve from A to B. A_{kl}^{i} represents the influence coefficients obtained by integration over the interface of the sub-domains k and l and p_{kl}^{i} represents the unknown potentials and derivatives at the nodes on the interface. When considering nodes on the interface several different situations may occur, of which only the most characteristic two will be explained in this text. The first one will be analysed using the node F on the interface between sub-domains 1 and 2. In this node there are four unknowns, two potentials and two

normal derivatives. Two equations can be written collocating from the F node, one for the sub-domain 1 and the other for the sub-domain 2. Using Eqs. (8) and (9) the contribution of this node towards a closed system of equations is achieved. The situation with node E is more complicated as this node is shared between four sub-domains. In each sub-domain there will be three unknowns, two derivatives and one potential, which would overall make 12 unknowns. However, as the potential is unique in this node, using Eq. (8) three of the unknowns are eliminated reducing the number of unknowns to nine. Further, by using Eq. (9) the number of unknowns will be reduced to five, that is, four normal derivatives, for example the ones shown in Figure 1, and the potential. With four equations that can be written collocating from the node into each of the sub-domains, the contribution of this node towards closed system of equations is not yet achieved and unless the medium is homogeneous and the line/s B–E–D or/and A–E–C are smooth at the node E, this node would need to be converted to a discontinuous node in order that a closed system of equations is achieved. Node E when discontinuous will have four freedom nodes instead, moved for a small distance from the location of node E on the lines A–E–C, in the direction of A and C nodes, and on the line B–E–D, in the direction of B and D nodes. In each of the new freedom nodes a situation equivalent to the situation in node F will appear. When the medium is homogeneous and the intersection lines are smooth in E, it can be shown that $q_{12}=q_{43}$ and $q_{23}=q_{14}$, reducing the number of unknowns to three, making the final system of the equations over-determined. Therefore, when continuous, node E may have three degrees of freedom, contributing towards an over-determined system, or four degrees of freedom contributing towards a closed system of algebraic equations, depending on whether both or just one of the B–E–D and A–E–C lines are smooth in the node E. Node E can be continuous if all of the sub-domains 1–4 are with the same properties, or two by two of the neighbouring domains are of same properties, i.e., 1–2 and 3–4, or, 1–4 and 2–3. In any other combination node E must be discontinuous, which produces eight degrees of freedom, contributing towards a closed system of algebraic equations. Similar analysis could be applied to nodes which are shared between three or more than four sub-domains in 2D or 3D.

Next, let us define degree of overdetermination of acontinuous node, O_v, as

$$O_v = N_{eq} - N_{un} \tag{11}$$

where N_{eq} is the number of equations introduced by the node and N_{un} the number of unknowns at the node.

Provided all the subdomains around the node have the same properties, the value of N_{eq} and N_{un} can be calculated as

$$N_{eq} = N_{sub} + N_{co} \tag{12}$$

$$N_{un} = 1 + N_{in} \tag{13}$$

where N_{sub} is the number of subdomains around the node, N_{co} the number of independent conditions of collinearity (in 2-D problems) or coplanarity (in 3-D problems) that exist in respect to interfaces joining the node and N_{in} the number of interfaces joining the node.

4 MODEL FORMULATION

In this paper, results on problems governed by the advection–diffusion equation

$$D\nabla^2 u - \vec{V}\cdot\vec{\nabla}u - ku = 0 \tag{14}$$

are presented, where D is the coefficient of diffusion, $\vec{V}$ is the vector of flow velocity and k is the reaction constant. Eq. (14) can be written as a non-homogeneous Laplace equation yielding a b term equal to $\frac{1}{D}\left(\vec{V}\cdot\vec{\nabla}u + ku\right)$. After applying the DRM formulation (7), the resulting system of equations becomes

$$\mathbf{H}\boldsymbol{u} - \mathbf{G}\boldsymbol{q} = (\mathbf{H}\hat{\mathbf{U}} - \mathbf{G}\hat{\mathbf{Q}})\mathbf{F}^{-1}\frac{1}{D}\left[\boldsymbol{V}_{e1}\frac{\partial \boldsymbol{u}}{\partial e_1} + \boldsymbol{V}_{e2}\frac{\partial \boldsymbol{u}}{\partial e_2} + \boldsymbol{V}_{e3}\frac{\partial \boldsymbol{u}}{\partial e_3} + k\boldsymbol{u}\right] \tag{15}$$

where $\boldsymbol{V}_{e_1}$, $\boldsymbol{V}_{e_2}$ and $\boldsymbol{V}_{e_3}$ are diagonal matrices containing the flow velocity components in the directions e_1, e_2 and e_3, respectively.

Replacing the partial derivatives by

$$\frac{\partial \boldsymbol{u}}{\partial e_i} = \frac{\partial \mathbf{F}}{\partial e_i}\mathbf{F}^{-1}\boldsymbol{u} \tag{16}$$

and denoting $\mathbf{S}$ the matrix $(\mathbf{H}\hat{\mathbf{U}} - \mathbf{G}\hat{\mathbf{Q}})\mathbf{F}^{-1}$ yields

$$\mathbf{H}\boldsymbol{u} - \mathbf{G}\boldsymbol{q} = \frac{\mathbf{S}}{D}\left[\boldsymbol{V}_{e1}\frac{\partial \mathbf{F}}{\partial e_1}\mathbf{F}^{-1}\boldsymbol{u} + \boldsymbol{V}_{e2}\frac{\partial \mathbf{F}}{\partial e_2}\mathbf{F}^{-1}\boldsymbol{u} + \boldsymbol{V}_{e3}\frac{\partial \mathbf{F}}{\partial e_3}\mathbf{F}^{-1}\boldsymbol{u} + k\boldsymbol{u}\right] \tag{17}$$

Denoting as $\mathbf{T}$ the following matrix: $\frac{\mathbf{S}}{D}\left[\boldsymbol{V}_{e1}\frac{\partial \mathbf{F}}{\partial e_1}\mathbf{F}^{-1} + \boldsymbol{V}_{e2}\frac{\partial \mathbf{F}}{\partial e_2}\mathbf{F}^{-1} + \boldsymbol{V}_{e3}\frac{\partial \mathbf{F}}{\partial e_3}\mathbf{F}^{-1}\right]$ and reordering (17) produces

$$\left(\mathbf{H} - \mathbf{T} - \frac{\mathbf{S}}{D}k\right)\boldsymbol{u} - \mathbf{G}\boldsymbol{q} = 0 \tag{18}$$

5 COMPUTATIONAL IMPLEMENTATION

The implementation of the DRM-MD can be made in a variety of arrangements regarding factors such as domain subdivision, continuity of elements, shape functions, etc. The results presented in this paper were produced by two codes that basically differ in the continuity of the boundary elements. Apart from this, both codes require that the domain be subdivided in tetrahedral subdomains, being every side of every tetrahedron a triangular boundary element which geometry is described by a quadratic shape function; consequently, the geometry of every tetrahedron is described by ten geometrical nodes (Figure 2). This makes the codes very flexible to model complicated geometries. Quadratic shape functions have been also used to approximate the potentials and derivatives. The matrices arising from the model are stored in sparse format. This feature together with the use of iterative solvers makes it possible to run a wide range of problems on a PC.

The main complications arising when using multidomain BEM approaches are due to mesh generation and assembly of equations. The problem of mesh generation is relatively easy to solve since a variety of commercial mesh generators are available. The assembly is more complicated in the subdomain BEM methods since potentials as well as derivatives need to be assembled as described in Section 3. For this purpose an assembling algorithm has been

designed that is general and easily adaptable to problems with different governing equations. The algorithm has been described by Natalini (2005).

Both codes use ATPS as DRM approximation function.

5.1 CODE A: a code using discontinuous boundary elements

CODE A is the same used by Natalini and Popov (2004, 2006) to test different RBFs, which is to the authors' knowledge the first ever reported 3D DRM-MD code. A modified version of this code was produced by Peratta and Popov (2006) to simulate transport phenomena in a waste repository. This code uses discontinuous triangular elements, following the treatment proposed by Do Rêgo Silva (1994), who described in detail the BEM implementation. Over every triangular element, six nodes of freedom are distributed according to Figure 3, where it can be seen that the nodes of freedom occupy positions that are different from the geometrical nodes. At every node of freedom there are two variables, which can be unknown or one of them can be specified as a boundary condition. Inside every tetrahedron, interior DRM nodes can be added. If interior DRM nodes have not been added, this code produces 24 equations for every subdomain.

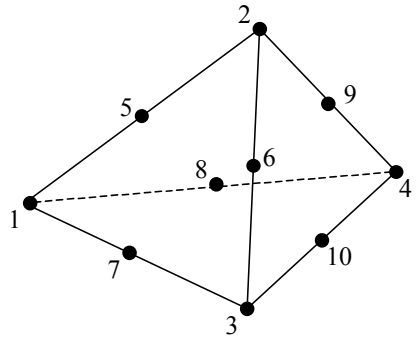

Figure 2: Geometrical nodes of a tetrahedral subdomain.

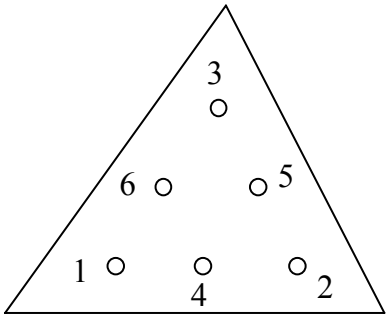

Figure 3: Nodes of freedom on a discontinuous triangular boundary element.

After the assembly is completed, the resulting system of equations is regular. Preliminary tests with different solvers gave the best performance with an iterative solver using a Conjugate Gradient-Normal Residual algorithm (Saad, 1996), which is in agreement with the experience of Peratta and Popov (2006). This solver with no preconditioning has been used in the examples presented in this work.

5.2 CODE B: a code using continuous boundary elements

This code uses continuous triangular elements belonging also to the family of elements proposed by Do Rêgo Silva (1994). The nodes of freedom occupy the same position as the geometrical nodes, hence it produces ten equations for every subdomain, provided interior DRM nodes have not been added. An important factor must be considered when using continuous elements. In Section 3 it has been explained that the contribution of a continuous node towards a closed system of equations cannot be achieved unless the medium is homogeneous and the plane connections between elements are smooth. When the mesh is structured, these conditions can be fulfilled, but in a general case the mesh is unstructured, therefore the condition is not satisfied , unless the node is on the boundary. Consequently, (18) was transformed in such a way that the variables at nodes were potential and **partial**

derivatives instead of potential and normal derivatives. The change of variables has been done in the following way. Let us apply Eq. (18) to a tetrahedral subdomain and express it in index notation.

$$\left(h_{ij} - t_{ij} - \frac{ks_{ij}}{D} \right) u_j - g_{ik} q_k = 0 \qquad \text{with} \qquad \begin{array}{l} 1 \le i \le 10 \\ 1 \le j \le 10 \\ 1 \le k \le 24 \end{array} \tag{19}$$

Note that the subindex k goes from 1 to 24 because there are 24 normal derivatives in a single tetrahedron. At the same time, every normal derivative is the scalar product of the gradient and the unit normal vector:

$$q_k = \left(\frac{\partial u}{\partial e_1} \right)_k n_{e_{1k}} + \left(\frac{\partial u}{\partial e_2} \right)_k n_{e_{2k}} + \left(\frac{\partial u}{\partial e_3} \right)_k n_{e_{3k}} \tag{20}$$

By replacing (20) into (19), one obtains

$$\left(h_{ij} - t_{ij} - \frac{ks_{ij}}{D} \right) u_j - g_{ik} n_{e_{1k}} \left(\frac{\partial u}{\partial e_1} \right)_k - g_{ik} n_{e_{2k}} \left(\frac{\partial u}{\partial e_2} \right)_k - g_{ik} n_{e_{3k}} \left(\frac{\partial u}{\partial e_3} \right)_k = 0 \tag{21}$$

The products between the elements of the **G** matrix and the components of the unit normal vectors can be assembled in three 10×10 matrices that we will denote $\mathbf{G}^{e_1}$, $\mathbf{G}^{e_2}$ and $\mathbf{G}^{e_3}$. The resulting system of equations is

$$\left(h_{ij} - t_{ij} - \frac{ks_{ij}}{D} \right) u_j - g_{ij}^{e_1} \frac{\partial u_j}{\partial e_1} - g_{ij}^{e_2} \frac{\partial u_j}{\partial e_2} - g_{ij}^{e_3} \frac{\partial u_j}{\partial e_3} = 0 \tag{22}$$

A further condition arises in order to apply (22): it must be at least four subdomains around every node in order to have a closed system of equations. This condition is not always fulfilled in apex nodes that are located in the boundary of the domain, where, instead, at least one condition of coplanarity can always be found. In those nodes the problem is defined as in the classical BEM, being the variables the potential and the normal derivative.

The resulting system of equations of the CODE B is over-determined. The over determined system is solved in a least-squares sense by means of the LSQR algorithm. In the examples presented here preconditioning has not been used. LSQR is an iterative method for computing a solution p to equations $Ap = m$, where A is a real matrix with m rows and n columns and m is a real vector. Developed by Paige and Saunders (1982), it is based on the bidiagonalization procedure of Golub and Kahan. It is analytically equivalent to the standard method of conjugate gradients, but possesses more favourable numerical properties.

6 NUMERICAL EXAMPLES

Preliminary tests on Laplace problems showed maximum errors of 10^{-8} and 10^{-11} % for the codes using discontinuous and continuous elements, respectively. These results are not shown here since they do not have a domain integral, which would require the use of the DRM for its solution.

The cases that will be shown correspond to a case of advection diffusion in a prismatic domain of length $L = 1$ in the e_1 direction and width $W = 0.2$ in the e_2 and e_3 directions, see Figure 4. The governing equation is given by (14), whose DRM formulation is given by (18).

The following boundary conditions were applied:

$$u(0, e_2, e_3)=U_0=10 \quad ; \quad u(L, e_2, e_3)=U_1=4 \tag{23}$$

and

$$\left.\frac{\partial u}{\partial n}\right|_{e_2=W/2} = \left.\frac{\partial u}{\partial n}\right|_{e_2=-W/2} = \left.\frac{\partial u}{\partial n}\right|_{e_3=W/2} = \left.\frac{\partial u}{\partial n}\right|_{e_3=-W/2} = 0 \tag{24}$$

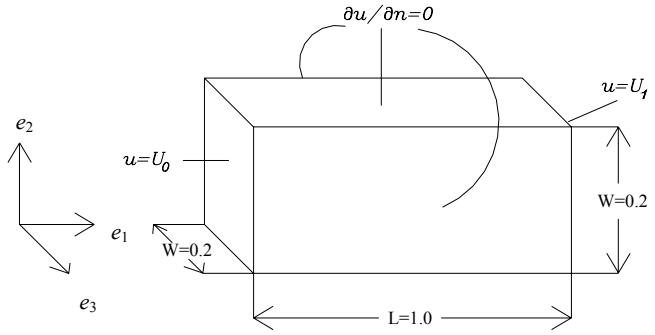

Figure 4: Geometry of the domain and the boundary conditions used in the numerical example.

The flow is in the e_1 direction, and the flow velocity field is a function of the reaction constant, k, and the e_1 coordinate, and is given as

$$V(e_1,k)=\frac{1}{L}\ln\left(\frac{U_1}{U_0}\right)+k\left(e_1-\frac{L}{2}\right) \tag{25}$$

For this geometry, velocity field and boundary conditions, the concentration field is symmetrical to the longitudinal axis, and the results computed on any line parallel to the e_1 axis are the same to the 1D problem with the following known analytical solution:

$$u(e_1,k)=U_0\exp\left\{\left(\frac{ke_1^2}{2}\right)+\left[\frac{1}{L}\ln\left(\frac{U_1}{U_0}\right)-L\frac{k}{2}\right]e_1\right\} \tag{26}$$

Figure 5 shows the distribution of flow velocities and Figure 6 the analytical solution along the domain for different values of k. Two meshes have been used: a rather coarse mesh of 173 subdomains and a finer one of 1456 subdomains (see Figure 7).

6.1 Tests on the scale of the problem

An important issue in this kind of codes was analysed, which is the scale of the problem. A given problem can be converted into a similar one but with a different scale if a scale factor is properly applied on both the boundary conditions and the parameters of the governing equations. Once a result corresponding to this new problem is obtained, using again the same scale factor the solution of the original problem can be retrieved. From a mathematical point of view the results should be identical no matter whether scaling has been applied or not. In the codes presented here the scale affects the time needed for the iterative solver to converge. The reason is that in the BEM formulation some of the matrices have different dimensions. For instance, let us consider the resulting system of equations of the advection–diffusion

problem described by (18). The term $\left(\mathbf{H} - \mathbf{T} - \frac{\mathbf{S}}{D} k \right)$ has no dimensions, but the matrix **G** has a dimension of length, consequently, the size of the problem affects the condition number of the resulting system of equations. As the DRM formulation is applied locally to every subdomain, a change in the size of the mesh, that is, when refining the mesh, will change the condition number.

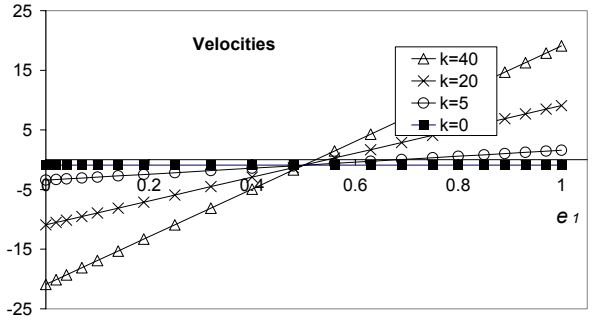

Figure 5: Magnitude of velocities along the domain for different values of k.

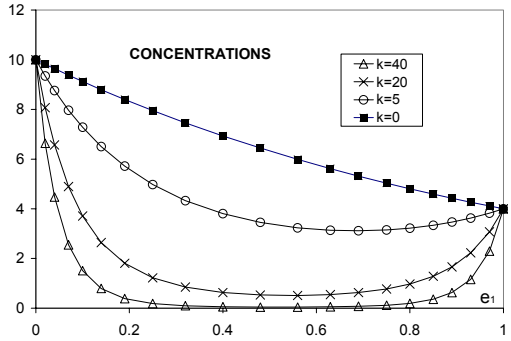

Figure 6: Analytical solution of the test cases.

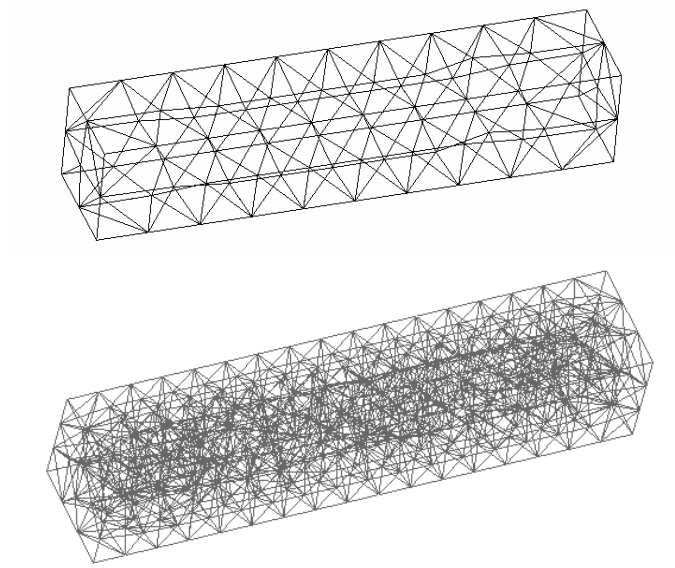

Figure 7: View of the subdivisions of the domain used in the examples.

Figure 8 shows the variation of the condition number with the inverse of the scale factor, when using both CODES A and B and the mesh of 173 subdomains. The inverse of the scale factor indicates how many times the characteristic length of the problem has been enlarged. It can be seen that the condition number has a minimum for an inverse of the scale factor around two for CODE A, and around sixteen for CODE B. Note that CODE A is better conditioned than CODE B. Higher values of the inverse of the scale factor cause changes in the condition number that are more sudden for CODE A and depend on the value of the reaction constant k. Let us recall that in this problem the velocity field is a function of *k* (see Figure 5); the higher the value of *k* the more advective the problem. As both, the advective term and the reactive

term are contained in the term $\left(\mathbf{H} - \mathbf{T} - \frac{\mathbf{S}}{D}k\right)$, this term becomes dominant when k increases; and as it has no dimensions, changes in the scale have less influence. For instance, for $k = 0$ the rate of increment of the condition number with the inverse of the scale factor is higher than for $k = 5$ and $k = 20$.

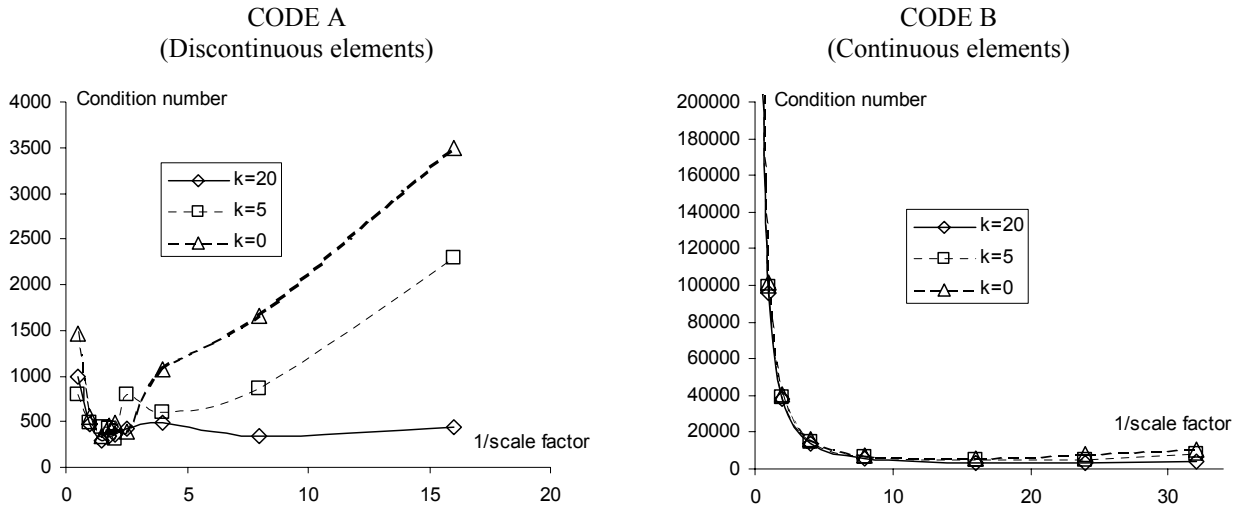

Figure 8: Variation of the condition number with the inverse of the scale factor for both the discontinuous element code and the continuous element code.

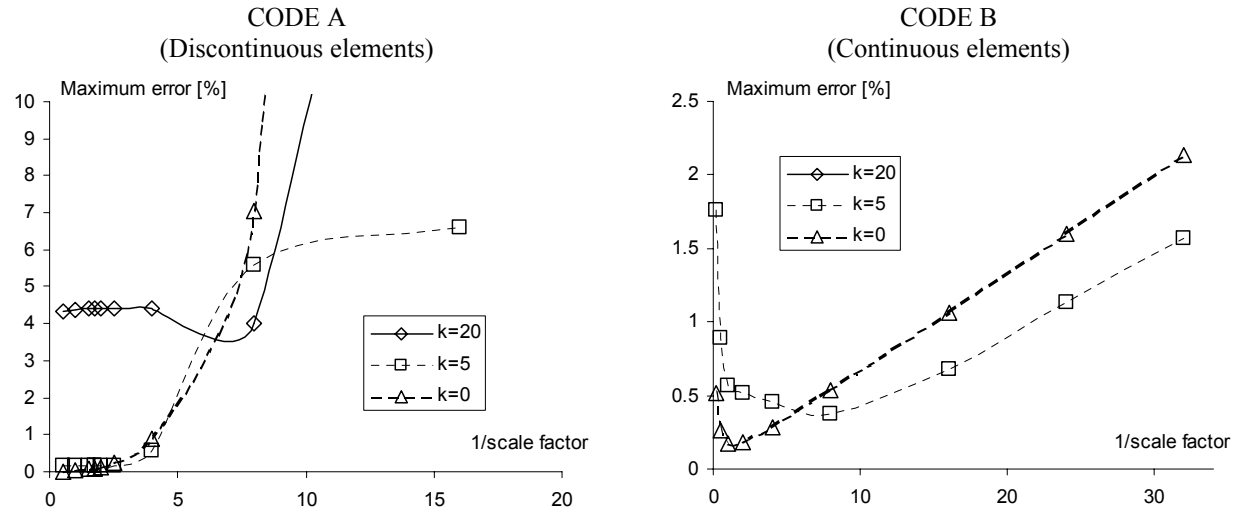

Figure 9: Variation of the maximum error with the inverse of the scale factor for both the discontinuous element code and the continuous element code (the values corresponding to the continuous element code with $k = 20$ are not displayed because they stay uniformly around 14%).

Figure 9 shows the variation of the maximum error with the inverse of the scale factor. For CODE A the best accuracy is in the range where the condition number is minimum. For CODE B the situation is different depending on the value of k. When the term

$\left(\mathbf{H}-\mathbf{T}-\frac{\mathbf{S}}{D}k\right)$ is dominant, the accuracy is not noticeably affected by the scaling, but when it is small or null, the error reaches a minimum for a scale factor that it is not the best from the point of view of the condition number. At the same time these optimal scales vary with the value of k.

It is evident that it is convenient to scale the problem when using these codes. A single change in the unit system in which a given problem is being specified or a refinement of the mesh can cause the code to fail if scaling is not implemented.

Table 1 and Table 2 compare the time that the solver needs to converge to the solution of the problem and the maximum error for both situations: with and without scaling, when using the refined mesh of 1456 subdomains. A better accuracy was expected with regard to the previous cases, in which a mesh of 173 subdomains was used. Note that in order to obtain a suitable scale factor it is the size of the subdomain what must be controlled rather than the size of the whole domain, hence, the scaling criterion was that the average size of the subdomains had to be kept the same as in the 173 subdomains case when the inverse of the scale factor was two, for CODE A, and fifteen, for CODE B.

The results displayed in Table 1 and Table 2 show how the rate of convergence of the solver improves with the scaling. It reduces roughly three times the CPU time for CODE A and ten times for CODE B. The accuracy worsens when k is equal to five and zero, however, for these values of k the codes achieved good accuracy so the error was low. For the CODE B and $k = 40$ the accuracy improves substantially with scaling. The refinement of the mesh clearly reduces the error when $k = 20$ but it does not when k is equal to five and zero. The convergence of the method with the refinement of the mesh shall be analysed in the next section.

CODE A								
	$k = 40$		$k = 20$		$k = 5$		$k = 0$	
	Not scaled	Scaled	Not scaled	Scaled	Not scaled	Scaled	Not scaled	Scaled
Solver time [s]	2366	746	2666	823	2928	895	3005	923
Maximum error [%]	6.3	6.5	0.94	0.92	0.039	0.36	0.0078	0.52

Table 1: CODE A (discontinuous elements). Time needed by the iterative solver to reach convergence and the maximum error with and without scaling, for a refined mesh of 1456 subdomains.

CODE B								
	$k = 40$		$k = 20$		$k = 5$		$k = 0$	
	Not scaled	Scaled	Not scaled	Scaled	Not scaled	Scaled	Not scaled	Scaled
Solver time [s]	1451	103	1503	145	1621	158	1661	169
Maximum error [%]	33.2	23.0	2.0	1.9	0.13	0.78	0.066	1.0

Table 2: CODE B (continuous elements). Time needed by the iterative solver to obtain a solution and maximum error with and without scaling, for a refined mesh of 1456 subdomains.

6.2 Mesh refinement and interior DRM nodes

Both codes were tested using interior DRM nodes. Two situations were considered: (a) when one interior DRM node was added in the mesh in the middle of every subdomain, and (b) when five interior DRM nodes were added, preserving as much as possible equal distance between the nodes in order to avoid ill conditioned system of equations. The distribution of the interior DRM nodes is done automatically by the code, once the number of the interior DRM nodes is defined in the input data.

It was expected that the accuracy would improve as more interior DRM nodes were added. Table 3 and Table 4 show that this was the case for both CODES A and B, using a mesh of 173 subdomains. Table 5 shows that this is still the case for the CODE A when using a refined mesh of 1456 subdomains, but Table 6 shows that not always using interior DRM nodes a better accuracy for the case of CODE B is achieved. For k equal to 5 and 0, when adding one interior DRM node, the accuracy improves in regard with the case with no interior node; but using 5 interior nodes slightly worsen the accuracy in regard with the case with just one interior node. At the same time, by comparing Table 3 and Table 5 and Table 4 and Table 6 it can be seen that both codes do not converge with mesh refinement when $k = 5$ and 0.

In order to understand the results of Table 3–Table 6, let us recall that the BEM formulation (7) uses the fundamental solution of the Laplace operator, hence, if the problem under study is dominated by the Laplace operator, refining the mesh or adding internal DRM nodes would not necesarily improve the accuracy since new equations are being added, which worsen the condition number of the resulting system of equations. In the advection–diffusion problem considered here, the diffusive process is represented by the Laplacian while the advective transport is represented by the non-homogeneous term. Under these conditions, when mesh refinement is applied, in an advection dominated problem the accuracy always improves because at every subdomain the relative importance of the advective transport is reduced, or to put it another way, the local Peclet (Pe) number is reduced, since if a length scale linked to the size of the subdomain is used to calculate the Pe number at every subdomain, every time the mesh is refined, the Pe number decreases. If after refining the mesh the advection is still relevant, the accuracy can be improved by adding internal DRM nodes.

CODE A												
	$k = 40$			$k = 20$			$k = 5$			$k = 0$		
Number of internal DRM nodes	0	1	5	0	1	5	0	1	5	0	1	5
Solver time [s]	97	107	141	95	102	136	102	108	144	104	111	145
Maximum error [%]	29.6	28.3	23.3	4.37	4.14	3.12	0.17	0.16	0.13	0.03	0.03	0.03

Table 3: CODE A (discontinuous elements). Maximum error with various number of interior DRM nodes. Mesh of 173 subdomains.

CODE B												
	$k = 40$			$k = 20$			$k = 5$			$k = 0$		
Number of internal DRM nodes	0	1	5	0	1	5	0	1	5	0	1	5
Solver time [s]	10	17	45	10	19	47	12	20	55	13	24	63
Maximum error [%]	120	99	77	14.4	12.0	9.2	0.63	0.43	0.33	0.99	0.65	0.34

Table 4: CODE B (continuous elements). Time needed by the solver to obtain a solution and maximum error with various number of interior DRM nodes. Mesh of 173 subdomains.

CODE A												
	$k = 40$			$k = 20$			$k = 5$			$k = 0$		
Number of internal DRM nodes	0	1	5	0	1	5	0	1	5	0	1	5
Solver time [s]	746	802	1115	823	885	1220	895	959	1304	923	986	1350
Maximum error [%]	6.5	6.08	4.41	0.92	0.90	0.67	0.36	0.24	0.05	0.52	0.36	0.07

Table 5: CODE A (discontinuous elements). Time needed by the solver to obtain a solution and maximum error with various number of interior DRM nodes, for a refined mesh of 1456 subdomains.

CODE B												
	$k = 40$			$k = 20$			$k = 5$			$k = 0$		
Number of internal DRM nodes	0	1	5	0	1	5	0	1	5	0	1	5
Solver time [s]	103	201	637	145	231	707	158	306	972	169	325	1027
Maximum error [%]	23.0	26.3	30.1	1.9	1.61	1.37	0.78	0.51	0.67	1.0	0.65	0.71

Table 6: CODE B (continuous elements). Time needed by the solver to obtain a solution and maximum error with various number of interior DRM nodes, for a refined mesh of 1456 subdomains.

The apex of the subdomains of the coarser mesh is about 1/10 long, and the one of the finer mesh is about 1/17 long. D was set equal to 1 in all the examples. Figure 10 and Figure 11 show the values of the local Pe along the subdomain. Figure 10 shows that the local Pe is below 0.34 and 0.09 for k equal to 5 and 0, respectively, when using the coarser mesh. As the transport is mainly diffusive along the whole domain, the accuracy would not improve by refining the mesh. And this is what has been observed in both CODES A and B. Under these conditions, the results in Table 4 and Table 6 show that internal DRM nodes are not of much help for CODE B. Table 3 and Table 5 show that adding internal DRM nodes can improve the accuracy of CODE A.

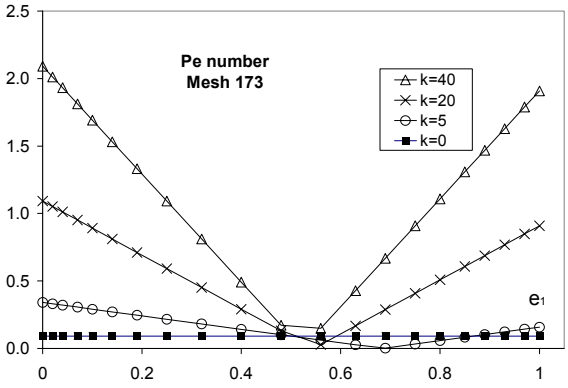

Figure 10: Local Peclet number distribution for the coarser mesh.

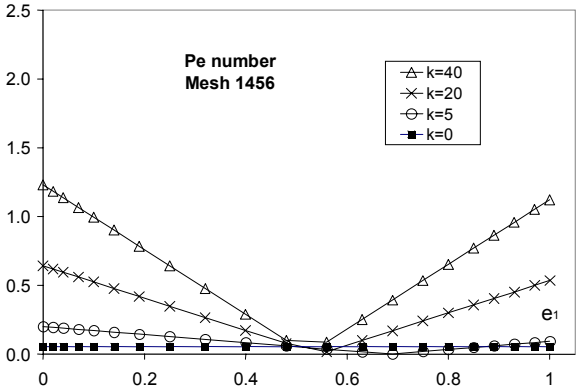

Figure 11: Local Peclet number distribution for the finer mesh.

When k is equal to 20, in vast parts of the domain the problem is advective enough as to show convergence with mesh refinement. When k is above 20, adding DRM nodes improves the accuracy for the same reason, and although at a first sight the results of the CODE B seem not to be in agreement with this conclusion, it shall be seen that they are not an exception.

Figure 12 and Figure 13 show the distribution of the absolute and relative error when k = 40 and the problem is solved with CODE B. The crosses correspond to NO internal DRM nodes, the triangles correspond to 1 internal DRM node and the dots to 5 internal DRM nodes. Figure 14 and Figure 15 show the same for CODE A. Figure 12 shows that adding internal DRM nodes in the case of CODE B reduces the absolute error where it is higher: in the zone where advection is more important, and slightly increases it in the middle of the domain, where diffusion is dominant. In the advective part of the domain, the absolute error is two orders of magnitude higher than in the middle. But, it is in the middle of the domain where the relative error is higher because the concentrations are the lowest, and that is why the maximum relative error increases with internal DRM nodes in Figure 13. Conversely, Figure 15 shows that in the CODE A case the relative error is distributed more evenly, then, adding internal DRM nodes improves both absolute and relative errors.

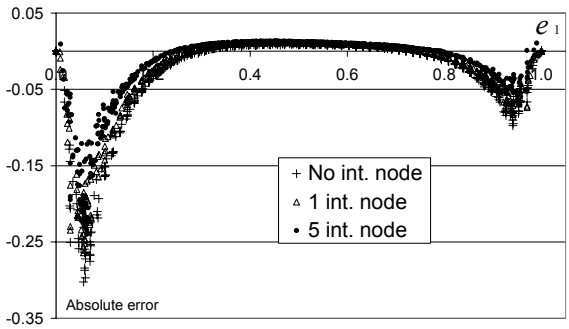

Figure 12: CODE B. Absolute error distribution for k = 40.

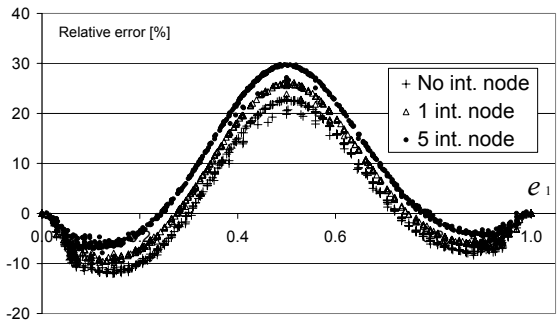

Figure 13: CODE B: Relative error distribution for k = 40.

According to the previous analysis, internal DRM nodes can be used to improve the accuracy but they should be placed only in those subdomains where advection is important. For instance, the test problem was solved with CODES A and B , with k = 40 and using the finer mesh with interior DRM nodes only in those subdomains where Pe > 0.65. Table 7

compares the obtained results with the ones of the codes using DRM nodes in every subdomain. Results of both codes show that similar accuracy is achieved in both situation, that is, when using internal nodes inside every subdomain and when using internal nodes only where Pe > 0.65; but in the later situation a good deal of solver time is gained compared with the former one.

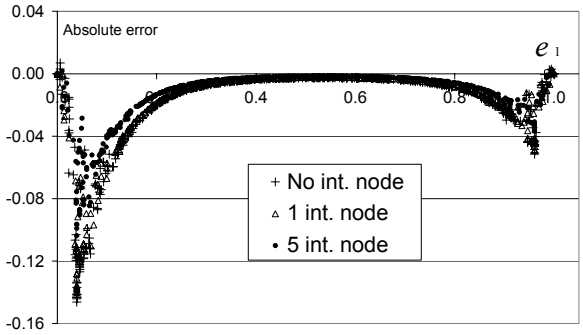

Figure 14: CODE A. Absolute error distribution for k = 40.

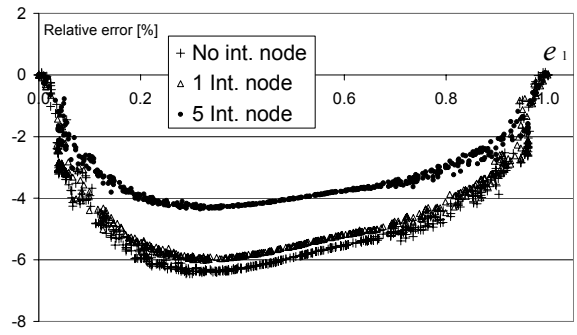

Figure 15: CODE A: Relative error distribution for k = 40.

	CODE A - k = 40				CODE B - k = 40			
	Internal DRM nodes in every subdomain		Internal DRM nodes only where Pe > 0.65		Internal DRM nodes in every subdomain		Internal DRM nodes only where Pe > 0.65	
	1 IDRMN	5 IDRMN	1 IDRMN	5 IDRMN	1 IDRMN	5 IDRMN	1 IDRMN	5 IDRMN
Solver time [s]	802	1115	777	927	201	637	174	381
Maximum error [%]	6.08	4.41	6.11	4.53	26.3	30.1	25.9	29.0

Table 7: Comparison of the performance of the both codes using internal DRM nodes in every subdomain and only where Pe > 0.65.

6.3 Comparison of performance of CODES A and B

Table 3–Table 6 show that as long as the same mesh is used, CODE A is more accurate than CODE B, specially when the problem is dominated by advection, but the time required by CODE A to reach the solution is one order of magnitude higher than the time needed by CODE B. Let us recall that CODE A produces 24 equations for every subdomain while CODE B produces 10, provided no internal DRM nodes are added. In order to compare the performance of both codes having a similar number of degrees of freedom, a mesh of 413 subdomains was produced and the example was solved with code B, with no internal DRM nodes (Table 8). In this case, CODE B produces a system of equation of 4130 degrees of freedom, which is very close to the 4152 degrees of freedom of the system produced by CODE A when using the 173 subdomain mesh (Table 3). If the comparison is done in this way, the difference between both codes regarding both accuracy and solver time is reduced, being the reduction more important as the problem is more advective. However, the trend is the same as it was observed previously, the CODE A shows to be more accurate and slower than CODE B.

Note that the time appearing in Table 1–Table 8 corresponds only to the time needed by

the solver to converge to a solution; it does not compute the time required by other tasks of the codes such as the assembling of the system of equation. Let us recall that CODE A produces a regular system of equations while CODE B produces an overdetermined system, hence they use different solvers (Section 5). There is no simple argument to explain the magnitude of the difference observed between the solvers time. For example, the efficiency of the solvers is influenced by factors such as the condition number and the structure of the matrix of coefficients of the system. The two codes produce different matrices, which affects the efficiency of the solvers. A proper answer requires further research, similar to the one already completed for 2D DRM-MD codes (see Portapila and Power, 2001, 2005), which is currently being carried out.

CODE B				
	$k = 40$	$k = 20$	$k = 5$	$k = 0$
Solver time [s]	28	29	35	37
Maximum error [%]	46.2	5.52	0.73	1.01

Table 8: Solver time and maximum error for CODE B using a mesh of 413 subdomains.

7 CONCLUSIONS

In this paper, the sensitivity of the 3D implementation of the boundary element dual reciprocity method—multidomain approach (DRM-MD) to the continuity of the elements, scaling, mesh refinement and number of internal nodes, has been tested in an advection–diffusion problem with non-uniform velocity field.

It has been shown that implementation of scaling is essential for these kind of problems. The examples presented here are sensitive to length scaling, but it must be noted that when changing the governing equation other scales can become relevant. Tests were made on an advection–diffussion problem with non-uniform velocity field using both discontinuous and continuous elements (CODES A and B, respectively). The size of the subdomains, which depends on the size of the problem and the degree of refinement of the mesh, affects the condition number of the resulting system of equations and consequently the time needed by the iterative solvers to converge. By using scaling of the equations, the solver time was reduced by three and by ten times for the CODES A and B, respectively.

For the examples tested, interior DRM nodes in every subdomain improved the accuracy of both codes provided they are used in subdomains where the advective transport, that is, the non-homogeneous term, plays a noticeable role, for instance, where the local Pe is above 0.5. Adding interior DRM nodes where the local Pe is lower than 0.5 would only increase the CPU time required by the iterative solvers without improving or even worsening the accuracy. The same can be said about mesh refinement. Every time the mesh is refined the DRM formulation becomes more diffusive at the level of every subdomain, and as the diffusive term is computed by the Laplacian, it does not make sense to refine the mesh further in areas where the local Pe is below 0.5.

Regarding the use of continuous and discontinuous elements, it is difficult to say which strategy is better. The discontinuous element code offers higher accuracy, especially for highly advective transport, but it is much slower than the continuous element code. At the present stage, the discontinuous element code has the advantage that it can solve non-homogeneous domains. The most convenient strategy is to have a choice to use both types of

elements in a single mesh, which would preserve the versatility of dealing with non-homogeneous domains, while offering possibility for CPU and memory usage reduction.

REFERENCES

Buhmann, M.D., A new class of radial basis function with compact support. *Math Comput*, 70/233:307–18, 2001.

Do Rêgo Silva, J.J., *Acoustic and elastic wave scattering using boundary elements*. Southampton and Boston: Computational Mechanics Publications, 1994.

Floater, M.S., and Iske, A., Multistep scattered data interpolation using compactly supported radial basis functions. *J Comput Appl Math*, 73:65–78, 1996.

Florez, W.F., *Nonlinear flow using dual reciprocity*. Southampton: WIT Press, 2001.

Florez, W.F., and Power, H., DRM multidomain mass conservative interpolation approach for the BEM solution of the two-dimensional Navier–Stokes equations. *Comput Math Appl*, 43/3-5:457–72, 2002a.

Florez, W.F, and Power, H., Multi-domain mass conservative dual reciprocity method for the solution of the non-Newtonian Stokes equations. *Appl Math Model*, 26/3:397–419, 2002b.

Golberg, M.A., and Chen, C.S., The theory of radial basis functions applied to the BEM for inhomogeneous partial differential equations. *Bound Elem Commun*, 5:57–61, 1994.

Golberg, M.A., and Chen, C.S., A bibliography on radial basis functions approximation. *Bound Elem Commun*, 7:155–63, 1996.

Nardini, D., and Brebbia, C.A., A new approach to free vibration analysis using boundary elements. *Appl Math Model*, 7:157–62, 1983.

Natalini, B., The boundary element dual reciprocity method—multidomain approach for solving 3D potential problems. Ph.D. thesis, University of Wales, 2005.

Natalini, B., and Popov, V., Radial basis function selection in 3D DRMMD for potential problems. In: Brebbia CA, editor. *Boundary elements XXVI*. Southampton: WIT Press, 101–10, 2004.

Natalini, B., and Popov, V., Tests of radial basis functions in the 3D DRMMD. *Commun Numer Methods Eng*, 22:13–22, 2006.

Paige, C.C., and Saunders, M.A., Algorithm 583: LSQR: sparse linear equations and least squares problems. *ACM Trans Math Software*, 8/2:195–209, 1982.

Peratta, A., and Popov, V., A new scheme for numerical modelling of flow and transport processes in 3D fractured porous media. *Adv Water Resour*, 29:42–61, 2006.

Popov, V., and Power, H., DRM-MD approach for the numerical solution of gas flow in porous media with application to landfill. *Eng Anal Bound Elem*, 23:175–88, 1999a.

Popov, V., and Power H., The DRM-MD integral equation method: an efficient approach for the numerical solution of domain dominant problems. *Int J Numer Meth Eng*, 44:327–53, 1999b.

Popov, V., and Power, H., Numerical analysis of the efficiency of landfill venting trenches. *J Environ Eng ASCE*, 126/1:32–8, 2000.

Popov, V., Power, H., and Baldasano, J.M., BEM solution of design of trenches in a multi-layered landfill. *J Environ Eng ASCE*, 124/1:59–66, 1998.

Portapila, M.I., and Power, H., Efficiency and accuracy of an iterative solution for the DRM-MD. *Bound Elem Commun*, 12/1:26–37, 2001.

Portapila, M.I., and Power H., Iterative schemes for the solution of system of equations arising from the DRM in multi-domain approach, and a comparative analysis of the performance of two different radial basis functions used in the interpolation. *Eng Anal Bound Elem*, 29/2:107–25, 2005.

Rodríguez, J.J., and Power, H., An adaptative dual reciprocity scheme for the numerical solution of the Poisson equation. *Eng Anal Bound Elem*, 26:283–300, 2002.

Saad, Y., *Iterative methods for sparse linear systems*. New York: PWS Publishing, 1996.

Schaback, R., Creating surfaces from scattered data using radial basis functions. In: Lyche, T., Dæhlen, M., Schumaker, L., editors. *Mathematical methods for curves and surfaces*. Nashville: Vanderbilt University Press, 477–96, 1995.

Schclar, N.A., Application of the boundary element method to the structural analysis of three dimensional anisotropic material. Ph.D. thesis, University of Portsmouth, 1993.

Wendland, H., Piecewise polynomial, positive definite and compactly supported radial function of minimal degree. *Adv Comput Math*, 4:389–96, 1995.

Wu, Z., Multivariate compactly supported positive definite radial functions. *Adv Comput Math*, 4:283–92, 1995.