Accepted Manuscript

Isogeometric Boundary Element Analysis of steady incompressible viscous flow, Part 1: Plane problems

Gernot Beer, Vincenzo Mallardo, Eugenio Ruocco, Christian Dünser

PII: S0045-7825(17)30259-1
DOI: http://dx.doi.org/10.1016/j.cma.2017.08.005
Reference: CMA 11544

To appear in: *Comput. Methods Appl. Mech. Engrg.*

Received date : 8 February 2017
Revised date : 5 August 2017
Accepted date : 7 August 2017

Please cite this article as: G. Beer, V. Mallardo, E. Ruocco, C. Dünser, Isogeometric Boundary Element Analysis of steady incompressible viscous flow, Part 1: Plane problems, *Comput. Methods Appl. Mech. Engrg.* (2017), http://dx.doi.org/10.1016/j.cma.2017.08.005

Isogeometric Boundary Element Analysis of steady incompressible viscous flow, Part 1: plane problems

Gernot Beer[a,*], Vincenzo Mallardo[b], Eugenio Ruocco[c], Christian Dünser[a]

[a]*Institute of Structural Analysis, Graz University of Technology, Lessingstraße 25/II, 8010 Graz, Austria*
[b]*Department of Architecture, University of Ferrara, Via Quartieri 8, 44121 Ferrara, Italy*
[c]*Department of Civil Engineering, Design, Building and Environment, University of Campania "L. Vanvitelli", Via Roma 28, 81031 Aversa, Caserta, Italy*

Abstract

A novel approach is presented to the Boundary Element analysis of steady incompressible flow. NURBS basis functions are used for describing the geometry of the problem and for approximating the unknowns. In addition, the arising volume integrals are treated differently to published work, that is, volumes are described by bounding NURBS curves instead of cells and a mapping is used. The advantage of our approach is that non-trivial boundary shapes can be described with very few parameters and that no generation of cells is required. For the solution of the non-linear equations both classical and modified Newton-Raphson methods are used. A comparison of the two methods is made on the classical example of a forced cavity flow, where accurate solutions are available in the literature. The results obtained agree well with published ones for moderate Reynolds numbers using both methods, but it is found that the latter requires a relaxation scheme and considerably more iterations to converge. Finally, it is shown on a practical example of an airfoil how more complex boundary shapes can be approximated with few parameters and a solution obtained with a small number of unknowns.

Keywords: BEM, isogeometric analysis, flow, incompressible

1. Introduction

Numerous approaches to numerically solve incompressible viscous flow problems can be found in the literature. Most publications use domain methods such as Finite Difference, Finite Elements or Finite Volumes (see for example [1]). A classical example to test published numerical methods is the forced flow in a cavity and very accurate solutions are available for comparison. In [2] for example an extremely fine finite difference mesh is used for the solution. We will use these solutions to compare our results later on.

*Corresponding author. Tel.: +43 316 873 6181, fax: +43 316 873 6185, mail: *gernot.beer@tugraz.at*, web: *www.ifb.tugraz.at*

Here we use the Boundary Element method (BEM). The advantage of this method is that for linear problems, unknowns only exist on the boundary and that the solutions inside the domain satisfy the governing differential equations exactly. For nonlinear problems, such as the one discussed here, volume integrals arise which have to be dealt with and this will be discussed in more detail later on.

1.1. The Boundary Element method for viscous flow

The governing differential equation for steady incompressible viscous flow can be developed from the laws governing the conservation of mass and momentum and assume the following differential forms:

$$\begin{aligned} \frac{\partial u_j}{\partial x_j} &= 0 \\ \mu \frac{\partial^2 u_i}{\partial x_j \partial x_j} - \frac{\partial p}{\partial x_i} - \rho u_j \frac{\partial u_i}{\partial x_j} &= 0 \end{aligned} \tag{1}$$

where x_i is the Eulerian coordinate, u_i is the velocity vector, p is the pressure, ρ the mass density and μ the viscosity.

The requirement for the BEM is the existence of fundamental solutions of the differential equations. These solutions can be found for the nonlinear equations (1) only if we consider the non-linear terms as body forces. We rewrite the equations as:

$$\begin{aligned} \frac{\partial u_j}{\partial x_j} &= 0 \\ \mu \frac{\partial^2 u_i}{\partial x_j \partial x_j} - \frac{\partial p}{\partial x_i} + f_i &= 0 \end{aligned} \tag{2}$$

with

$$f_i = -\rho u_j \frac{\partial u_i}{\partial x_j} \tag{3}$$

Fundamental solutions of equations (2) can now be obtained for an infinite domain by substituting the Dirac-Delta function for the body force.

We define fluid stresses as:

$$\sigma_{ij} = \mu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \tag{4}$$

and the resulting tractions on boundary S:

$$t_i = \sigma_{ij} n_j - p n_i \tag{5}$$

where n_i is the unit vector normal to the boundary. Using the reciprocal theorem, the following integral equation is obtained (for a full derivation refer to [3]):

$$c_{ij}(y)\, \dot{u}_j(y) = \int\limits_S [U_{ij}(y,x)\, t_j(x) - T_{ij}(y,x)\, \dot{u}_j(x)]\, dS(x) + \int\limits_{V_0} U_{ij}(y,\bar{x})\, f_j(\bar{x}) dV_0(\bar{x}) \tag{6}$$

where $c_{ij}(y)$ is an integral free term, depending on the shape of the boundary and $\dot{u}_i$ is the velocity perturbation, i.e. the total velocity can be written as:

$$u_i(x) = \dot{u}_i(x) + u_i^0(x) \tag{7}$$

with u_i^0 is the free stream velocity and $U_{ij}(y,x)$ and $T_{ij}(y,x)$ are fundamental solutions for the velocity and traction at point x due to a source at point y listed in the Appendix.

In Equation (6) f_j appears that involves derivatives of velocities. As has been shown in [4] these derivatives can be computed by using finite differences or by taking derivatives of an approximation of the velocity field. In both cases additional computational work needs to be done and errors are introduced.

Alternatively, the requirement of computing derivatives can be eliminated by applying the divergence theorem to the volume integral in Equation (6) as explained in [5] resulting in:

$$\begin{aligned} c_{ij}(y)\,\dot{u}_j(y) = & \int_S [U_{ij}(y,x)\,t_j(x) - T_{ij}(y,x)\,\dot{u}_j(x)]\,dS \\ & - \int_{S_0} U_{ij}(y,x)\,t_j^0(x)dS_0 + \int_{V_0} U_{ij,k}(y,\bar{x})\,b_{jk}^0(\bar{x})dV_0 \end{aligned} \tag{8}$$

where $U_{ij,k}(y,x)$ is a derived fundamental solution and:

$$\begin{aligned} b_{ik}^0(\bar{x}) &= \rho\,u_k(\bar{x})\,\dot{u}_i(\bar{x}) \\ t_i^0(x) &= b_{ik}^0(x)\,n_k(x) \end{aligned} \tag{9}$$

This is the approach used here.

1.2. Previous work and novelty of our approach

Early work on the solution of viscous flow problems using the BEM appeared for example in [5]. As with most published methods, internal cells were used for the evaluation of the volume integrals. Cells are basically like finite elements with the subtle difference that they are only used for evaluating integrals and not for approximating the resulting fields (fundamental solutions, that satisfy the linear differential equation, are used to approximate these inside the domain). In [5] Equation (8) was used and therefore the computation of derivatives of velocities was avoided.

Remark 1: To overcome the need for a volume discretization, approaches such as the dual reciprocity BEM [6] or the use of radial basis functions [7] have been proposed. However, radial basis functions can not be used for infinite domain problems and a comparison of these methods to the cell based approach found in [8] recommends the latter for accuracy and robustness.

Solving the forced cavity flow problem it was found that a very accurate integration scheme and a fine mesh of cells had to be used to obtain good results. For higher Reynolds numbers it was necessary to use a classic Newton-Raphson method, where the left hand side is updated at every iteration step, to get results that converge to the right solution. Results for Reynolds numbers up to 1000 are presented, but details on how the tangent operator was determined are missing. The application of the BEM to fluid flow problems has also been discussed in textbooks (see for example [9]).

In [10] and [11] the same approach as in [5] is used and the method is extended to three dimensions. The authors explain how the tangent operator for full Newtom-Raphson can be obtained, but results for the forced cavity flow problem are only presented for Reynolds numbers up to 100. A solution of the problem with the BEM can also be found in [4]. Here Equation (6) is used and the derivatives of velocities are computed using either finite differences or by taking the derivatives of basis functions that approximate the solution inside cells. A modified Newton-Raphson method with relaxation is used and good results are obtained for Reynolds numbers up to 1000.

Isogeometric analysis [12] has gained significant popularity in the last decade. The novelty of our approach is that instead of Lagrange polynomials, that are used in the quoted published work, NURBS basis functions are used for describing the geometry and the variation of the unknowns. The use of these functions means that fewer parameters are required to describe complex geometries accurately. Using NURBS to approximate values at the boundary also provides greater flexibility with respect to refinement options. The geometry independent field approximation method, that has already been used successfully in [13] and [14], means that approximation of the unknown is completely uncoupled from the geometry definition. Another novelty is that instead of cells, a mapping method that was first introduced in 2-D in [15] and extended to 3-D in [16], is used, abolishing the requirement of generating a cell mesh. Finally, a comparison between the modified and full Newton-Raphson methods is presented here for the first time.

2. Surface discretisation

As in majority of previous work on the isogeometric BEM [13, 14, 17, 18, 19, 20, 21, 22, 23] we use the collocation method, i.e. we write the integral equations for a finite number (N) of source points at locations $\mathbf{y}_n$. Changing to matrix notation, the integral equations are re-written as:

$$\begin{aligned}\mathbf{c}(\mathbf{y}_n)\,\dot{\mathbf{u}}(\mathbf{y}_n) &= \int_S \mathbf{U}(\mathbf{y}_n,\mathbf{x})\,\mathbf{t}(\mathbf{x})\,\mathrm{d}S - \int_S \mathbf{T}(\mathbf{y}_n,\mathbf{x})\,\dot{\mathbf{u}}(\mathbf{x})\,\mathrm{d}S \\ &\quad - \int_{S_0} \mathbf{U}(\mathbf{y}_n,\bar{\mathbf{x}})\,\mathbf{t}_0(\bar{\mathbf{x}})\,\mathrm{d}S_0 + \int_{V_0} \mathbf{U}'(\mathbf{y}_n,\bar{\mathbf{x}})\,\mathbf{b}_0(\bar{\mathbf{x}})\,\mathrm{d}V_0\end{aligned} \tag{10}$$

with $n=\{1,\ldots,N\}$. In the above $\mathbf{c}(\mathbf{y}_n)$ is a matrix containing integral free terms, $\dot{\mathbf{u}}(\mathbf{x})$ and $\mathbf{t}(\mathbf{x})$ are vectors containing perturbation velocities and tractions at point $\mathbf{x}$ on the boundary. $\mathbf{U}(\mathbf{y}_n,\mathbf{x})$ and $\mathbf{T}(\mathbf{y}_n,\mathbf{x})$ are matrices containing fundamental solutions listed in the Appendix. $\mathbf{b}_0(\bar{\mathbf{x}})$ is a body force vector at a point $\bar{\mathbf{x}}$ inside V_0.

Using the Einstein summation convention we have

$$\begin{aligned}U_{1j,k}(\mathbf{y},\bar{\mathbf{x}})\,b^0_{jk}(\mathbf{x}) &= U'_{111}\,b^0_{11} + U'_{112}\,b^0_{12} + U'_{121}\,b^0_{21} + U'_{122}\,b^0_{22} \\ U_{2j,k}(\mathbf{y},\bar{\mathbf{x}})\,b^0_{jk}(\mathbf{x}) &= U'_{211}\,b^0_{11} + U'_{212}\,b^0_{12} + U'_{221}\,b^0_{21} + U'_{222}\,b^0_{22}\end{aligned} \tag{11}$$

With the fundamental solution listed in the Appendix this can be converted into the matrix multiplication occurring in Equation (10) if:

$$\mathbf{U}' = \frac{1}{4\,\pi\,\mu\,r}\begin{pmatrix} r_1 - 2\,r_1^3, & -r_2 - 2\,r_1^2\,r_2, & r_2 - 2\,r_1^2\,r_2, & r_1 - 2\,r_2^2\,r_1 \\ r_2 - 2\,r_1^2\,r_2, & r_1 - 2\,r_2^2\,r_1, & -r_1 - 2\,r_2^2\,r_1, & r_2 - 2\,r_2^3 \end{pmatrix} \tag{12}$$

and:

$$\mathbf{b}_0(\bar{\mathbf{x}}) = \begin{pmatrix} b^0_{11}(\bar{\mathbf{x}}) \\ b^0_{12}(\bar{\mathbf{x}}) \\ b^0_{21}(\bar{\mathbf{x}}) \\ b^0_{22}(\bar{\mathbf{x}}) \end{pmatrix} = \rho \begin{pmatrix} u_x(\bar{\mathbf{x}})\,\dot{u}_x(\bar{\mathbf{x}}) \\ u_y(\bar{\mathbf{x}})\,\dot{u}_x(\bar{\mathbf{x}}) \\ u_x(\bar{\mathbf{x}})\,\dot{u}_y(\bar{\mathbf{x}}) \\ u_y(\bar{\mathbf{x}})\,\dot{u}_y(\bar{\mathbf{x}}) \end{pmatrix} \tag{13}$$

The initial traction vector is given by:

$$\mathbf{t}_0 = \mathbf{N}\,\mathbf{b}_0 \tag{14}$$

where

$$\mathbf{N} = \begin{pmatrix} n_x & n_y & 0 & 0 \\ 0 & 0 & n_x & n_y \end{pmatrix} \tag{15}$$

The integrals over the domain V_0 and boundary S_0 need only be evaluated if the body force term is not zero. In practice this means that volume integration can be avoided if the body force is negligible, i.e. the integration will usually be restricted to an area near the boundary.

For the discretization of the surface integrals over S we divide the boundary into patches and we adopt a geometry independent field approximation approach for each patch, i.e. we use different basis functions for the description of the geometry and for the field values.

$$\begin{aligned} \mathbf{x}^e &= \sum_{k=1}^{K} R_k(s)\,\mathbf{x}^e_k \\ \dot{\mathbf{u}}^e &= \sum_{k=1}^{K^u} R^u_k(s)\,\dot{\mathbf{u}}^e_k \\ \mathbf{t}^e &= \sum_{k=1}^{K^t} R^t_k(s)\,\mathbf{t}^e_k \end{aligned} \tag{16}$$

In the above equations the superscript e refers to the number of the patch, R_k, R^u_k and R^t_k are NURBS basis functions[1] with respect to the local coordinate s for the geometry, velocities and tractions respectively. $\mathbf{x}^e_k$ specify the location of control points and $\dot{\mathbf{u}}^e_k$

[1] Considering the vastly increasing literature on isogeometric analysis we refrain from presenting the equations for NURBS. The interested reader is referred to [24]

and $\mathbf{t}_k^e$ are the parameters for velocities and tractions. K, K^u, K^t are the number of parameters for each patch. The advantages of NURBS are that they are much better suited than Lagrange polynomials for describing smooth boundaries and that superior refinement strategies such as order elevation, knot insertion and k-refinement, can be used. By manipulating the knot vector one can easily influence the continuity of the basis functions for the approximation of unknown values.

Inserting the approximations into the integral equations and applying the rigid body trick for eliminating the free term and singular integration involving Kernel $\mathbf{T}$ as outlined in detail in [24] the following discretized integral equations can be obtained:

$$\sum_{e=1}^{E}\sum_{k=1}^{K^t} \triangle\mathbf{U}_{nk}^e \mathbf{t}_k^e = \sum_{e=1}^{E}\sum_{k=1}^{K^u} \triangle\mathbf{T}_{nk}^e \dot{\mathbf{u}}_k^e - \mathbf{T}_n \sum_{k=1}^{K^u} R_k^u(s_n)\,\dot{\mathbf{u}}_k^{en} \tag{17}$$

$$+\int_{S_0} \mathbf{U}(\mathbf{y}_n,\bar{\mathbf{x}})\,\mathbf{t}_0(\bar{\mathbf{x}})\,\mathrm{d}S_0 - \int_{V_0} \mathbf{U}(\mathbf{y}_n,\bar{\mathbf{x}})\,\mathbf{b}_0(\bar{\mathbf{x}})\,\mathrm{d}V_0$$

$$\text{for} \quad n = 1,2,3\cdots N$$

with *en* specifying the patch that contains the collocation point. Furthermore

$$\triangle\mathbf{U}_{nk}^e = \int_0^1 \mathbf{U}(\mathbf{y}_n,\mathbf{x}^e(s))\,R_k^t(s)\,J\,ds \tag{18}$$

$$\triangle\mathbf{T}_{nk}^e = \int_0^1 \mathbf{T}(\mathbf{y}_n,\mathbf{x}^e(s))\cdot R_k^u(s)\,J\,ds$$

$$\mathbf{T}_n = \sum_{e=1}^{E}\int_0^1 \mathbf{T}(\mathbf{y}_n,\mathbf{x}^e(s))\,J\,ds \tag{19}$$

In the above J is the Jacobian of the transformation from local s to global (x,y) coordinate systems.

After assembly the following system of equations

$$[\mathbf{U}]\{\mathbf{t}\} = ([\mathbf{T}] - [\mathbf{T}]_0)\{\mathbf{u}\} + \{\mathbf{F}\}_0 \tag{20}$$

is obtained, where $[\mathbf{U}]$, $[\mathbf{T}]$ are matrices assembled from patch contributions (18) and $\{\mathbf{t}\}$, $\{\mathbf{u}\}$ are vectors that collect all traction and velocity components on points $\mathbf{y}_n$. $[\mathbf{T}]_0$ is a matrix relating to rigid body modes and $\{\mathbf{F}\}_0$ relates to the integrals involving body forces. Either $\mathbf{t}$ or $\mathbf{u}$ must be known on the boundary, so for a mixed boundary problem we have

$$[\mathbf{L}]\{\mathbf{a}\} = \{\mathbf{F}\} + \{\mathbf{F}\}_0 \tag{21}$$

where $[\mathbf{L}]$ contains a mixture of $[\mathbf{U}]$,$[\mathbf{T}]$ coefficients and $\{\mathbf{a}\}$ contains a mixture of unknown tractions and velocities. $\{\mathbf{F}\}$ is a vector computed with known boundary values.

3. Integration

For flow problems accurate integration of the Kernel-basis function products is even more critical than for solid mechanics problems and therefore it is discussed in detail here. Depending on the ratio R (= minimum distance from integration region to collocation point $\mathbf{y}_n$) over L (=size of the integration region) the following cases occur:

1. $R/L > lim$ (the collocation point is far from the integration region). We refer to this as regular integration. $0 < R/L < lim$ (the collocation point is near the integration region). We refer to this as nearly singular integration.
2. $R/L = 0$ (the collocation point is inside the integration region and the Kernel is weakly singular). This is referred to singular integration.

where lim is a limiting value (see later).

3.1. Regular and nearly singular integration

If the collocation point is far away relative to the size of the integration region standard Gauss Quadrature can be applied with the number of Gauss points determined depending on the variation of the basis function and of the Jacobian. Note that NURBS basis functions are not polynomials but rational functions and therefore the number of Gauss points has to be higher than for Lagrange polynomials.

If the collocation point is near the integration region then the number of Gauss points has to be increased as a function of R/L. Instead of increasing the number of Gauss points to a large number it is more efficient to subdivide the integration region into subregions. We define $MaxG$ as the maximum number of Gauss points that we want to use and define an array $Rlim$ that determines the limiting value of R/L depending on the number of Gauss points G, i.e. the array $Rlim(1-6)$ contains limiting values for 3 to 8 Gauss points[2]. This means that R/L is checked against $Rlim$ and the required number of Gauss points is determined. If the maximum number specified by $MaxG$ is exceeded then the integration region is subdivided. The limiting value lim above is set to $Rlim(1)$.

For Gauss integration we require a local coordinate system $\xi = [-1,+1]$. For subregion m the relationship between coordinate s and ξ is given by

$$s(\xi) = \frac{d_m}{2}(\xi+1) + s_m \tag{22}$$

where d_m is the size of the subregion and s_m is the local coordinate of its left edge. The Jacobian of this transformation is $J_\xi = \frac{d_m}{2}$. The numerical integration of (18) is for example:

$$\triangle \mathbf{U}^e_{ni} = \sum_{m}^{M} \sum_{g=1}^{G(m)} \mathbf{U}\left(\mathbf{y}_n, \mathbf{x}^e(s(\xi_g))\right) R^t_i(s(\xi_g))\, J(s(\xi_g))\, J_\xi\, W_g \tag{23}$$

where M is the number of subregions, $G(m)$ is the number of Gauss points for subregion m, ξ_g is the Gauss point coordinate and W_g the corresponding weight.

[2] For further details see [24]. Limits are set for integration errors 10^{-2} to 10^{-4}

To make the integration even more efficient we use an interval halving method for determining the size of the subregions. This means that the integration region is first divided into two subregions of equal size. Each subregion is then checked if $R/L < Rlim$ for the maximum number of Gauss points specified. If this is the case then the subregion is halved again. This continues until $R/L \geqslant Rlim$.

3.2. Singular integration

Here we deal with a logarithmic singularity and we have to use a different Gauss formula which integrates exactly any function $f(\gamma)\,\ln\frac{1}{r}$ where $f(\gamma) = a_0 + a_1\,\gamma + \cdots + a_{2L-1}\,\gamma^{2L-1}$ and L is the number of Gauss points:

$$\int_0^1 f(\gamma)\,\ln\frac{1}{r}\,d\gamma \approx \sum_{l=1}^{L} f(\gamma_l)\,W_l \tag{24}$$

In the above γ_l are Gauss-Laguerre abscissae and W_l weights. For the integration the local coordinate $\gamma = [0,1]$ is used which originates from the singularity point. For the case where the singularity point is on the left edge of the integration region the transformation between s and γ is given by:

$$s(\gamma) = d_m\,\gamma + s_m \tag{25}$$

For the case where the singularity point is on the right edge of the integration region we have

$$s(\gamma) = d_m\,(1-\gamma) + s_m \tag{26}$$

The Jacobian of this transformation is $J_\gamma = d_m$. For the integration we have to isolate the singular part of $\mathbf{U}$ (see Appendix):

$$\triangle\mathbf{U}^e_{ni} = \int_0^1 c\left[\mathbf{R}\,(\mathbf{y}_n, \mathbf{x}^e(s)) + \mathbf{I}\ln\frac{1}{r} - \mathbf{I}\ln\frac{1}{\gamma}\right] R^t_i(s)\,J\,ds \tag{27}$$

$$+\int_0^1 c\mathbf{I}\ln\frac{1}{\gamma} R^t_i(s)\,J\,ds \tag{28}$$

where the first integral is now regular and can be evaluated with standard Gauss, whereas the second integral is singular and can be evaluated using Gauss-Laguerre. The singular integration for a subregion m is given by

$$\triangle\mathbf{U}^e_{ni} = \sum_{g=1}^{G}\left[\mathbf{U}\,(\mathbf{y}_n, \mathbf{x}^e(s(\xi_g))) - \mathbf{I}\,\ln\frac{1}{\gamma_g}\right] R^t_i(s(\xi_g))\,J(s(\xi_g))J_\xi\,W_g \tag{29}$$
$$+\sum_{l=1}^{L}\mathbf{I}R^t_i(s(\gamma_l))\,J(s(\gamma_l))J_\gamma W_l$$

4. Basic approach for dealing with volume terms

The basic approach is to solve the problem in an iterative way. First the linear problem is solved. Then the solution is modified to account for the presence of body forces.

The procedure can be summarized as follows:

1. Solve the linear problem and determine the velocity perturbation $\dot{\mathbf{v}}$ inside V_0.
2. Determine the increment in body force $\mathbf{b}_0$ from Equation (9).
3. Compute new right hand side $\mathbf{F}_0$ by evaluating the arising volume integral.
4. Solve for the new right hand side and compute a new increment of boundary velocity perturbation $\dot{\mathbf{u}}$.
5. Repeat 2. to 5. until $\mathbf{b}_0$ is sufficiently small.

5. Geometry definition of V_0

The first task is the description of the geometry of the subdomain V_0. For this we propose to use a mapping method introduced recently for trimmed surfaces in [14] and [24]. This means that the domain is defined by two NURBS curves and a linear interpolation between them.

We establish a local coordinate system $\mathbf{s} = (s,t)^\mathsf{T} = [0,1]^2$ as shown in Figure 1 on the right and perform the integration in this system and then map it to the global (x,y)-system. The global coordinates of a point $\mathbf{x}$ with the local coordinates $\mathbf{s}$ are given by

$$\mathbf{x}(s,t) = (1-t)\,\mathbf{x}^I(s) + t\,\mathbf{x}^{II}(s) \tag{30}$$

where

$$\mathbf{x}^I(s) = \sum_{k=1}^{K^I} R_k^I(s)\,\mathbf{x}_k^I \quad \text{and} \quad \mathbf{x}^{II}(s) = \sum_{k=1}^{K^{II}} R_k^{II}(s)\,\mathbf{x}_k^{II}. \tag{31}$$

The superscript I relates to the bottom (red) curve and II to the top (green) curve and $\mathbf{x}_k^I$, $\mathbf{x}_k^{II}$ are control point coordinates. K^I and K^{II} are the number of control points, $R_k^I(s)$ and $R_k^{II}(s)$ are NURBS basis functions [3]. The derivatives are given by

$$\begin{aligned} \frac{\partial \mathbf{x}(s,t)}{\partial s} &= \quad (1-t)\,\frac{\partial \mathbf{x}^I(s)}{\partial s} + t\,\frac{\partial \mathbf{x}^{II}(s)}{\partial s} \\ \frac{\partial \mathbf{x}(s,t)}{\partial t} &= \quad -\mathbf{x}^I(s) + \mathbf{x}^{II}(s) \end{aligned} \tag{32}$$

where

[3] Note that the advantage of this description is that this allows the definitions of the bounding curves to be different, i.e. the top curve can have different number of control points and order than the bottom curve.

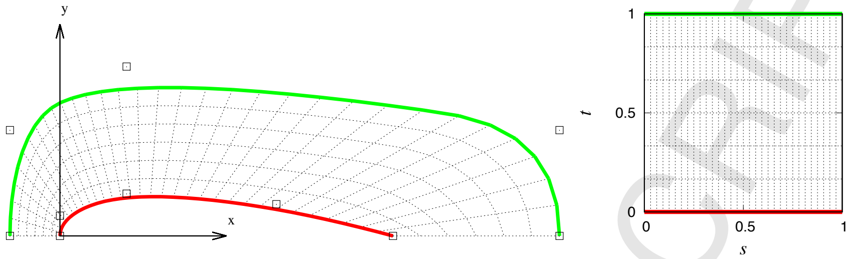

Figure 1: Example of a possible definition of volume V_0 surrounding half of an airfoil in (left) global x,y and (right) local s,t coordinate system. The red curve defines the boundary of the airfoil as well as the bottom boundary of the domain V_0. The green curve defines its top boundary. The associated control points are depicted by hollow squares. Note that only 5 control points are required to accurately define the shape of the airfoil and 5 more control points the surrounding domain.

$$\begin{aligned}\frac{\partial \mathbf{x}^I(s)}{\partial s} &= \sum_{k=1}^{K^I} \frac{\partial R_k^I(s)}{\partial s}\mathbf{x}_k^I \\ \frac{\partial \mathbf{x}^{II}(s)}{\partial s} &= \sum_{k=1}^{K^{II}} \frac{\partial R_k^{II}(s)}{\partial s}\mathbf{x}_k^{II}.\end{aligned} \tag{33}$$

The Jacobian matrix of this mapping is

$$\mathbf{J} = \begin{pmatrix} \frac{\partial x}{\partial s} & \frac{\partial y}{\partial s} \\ \\ \frac{\partial x}{\partial t} & \frac{\partial y}{\partial t} \end{pmatrix} \tag{34}$$

and the Jacobian is $J(s,t) = |\mathbf{J}|$.

An example is shown in Figure 1 where half of the shape of an airfoil is described by NURBS basis functions of order 2 and only 5 control points. Note that the bottom definition of the nonlinear domain V_0 is coincidental with the boundary of the problem.

Remark 2: It is obvious that the following theory is not restricted in any way to the description of V_0 outlined above. Any method that allows the mapping of its geometry to a unit square can be applied.

6. Computation of $\{\mathbf{F}\}_0$

Here we discuss the computation of the right hand side during iteration. This involves the solution of the integrals in Equation (10) over S_0 and V_0.

6.1. Computation of the surface integral over S_0

For the computation of the surface integral the same procedures as explained in section 3 are used.

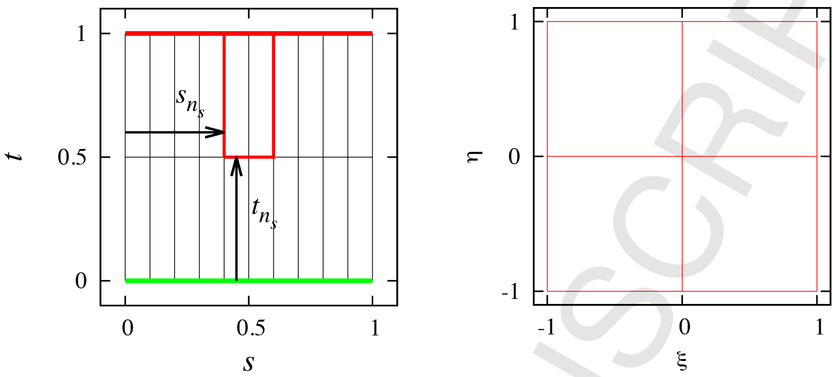

Figure 2: Subdivision of integration region for the volume integration (shown here in the **s** coordinate system) and mapping of the subregion surrounded by the red lines to the local coordinate system $\boldsymbol{\xi}$, in which the integration is carried out.

6.2. *Computation of the volume integral over* V_0

The volume integration also involves the subdivision into subregions. The transformation from **s** coordinates to $\boldsymbol{\xi} = (\xi, \eta)^{\mathsf{T}} = [-1,1]^2$ is given for integration region n_s by (see Figure 2)

$$\begin{aligned} s &= \tfrac{\Delta s_{n_s}}{2}(1+\xi) + s_{n_s} \\ t &= \tfrac{\Delta t_{n_s}}{2}(1+\eta) + t_{n_s} \end{aligned} \tag{35}$$

where $\Delta s_{n_s} \times \Delta t_{n_s}$ denotes the size of the integration region and s_{n_s}, t_{n_s} are the starting coordinates. The Jacobian of this transformation is $J_{\xi}^{n_s} = \frac{\Delta s_{n_s} \Delta t_{n_s}}{4}$.

The sub vector of $\{\mathbf{F}\}_0^{V_0}$ related to collocation point n can be written as:

$$\mathbf{F}_{0n}^{V_0} = \sum_{n_s=1}^{N^s} \int_{-1}^{1} \int_{-1}^{1} \mathbf{U}'\left(\mathbf{y}_n, \bar{\mathbf{x}}(\xi,\eta)\right) \dot{\mathbf{b}}_0\left(\bar{\mathbf{x}}(\xi,\eta)\right) J(\mathbf{s})\, J_{\xi}^{n_s}\, \mathrm{d}\xi\, \mathrm{d}\eta \tag{36}$$

Applying Gauss integration we have:

$$\mathbf{F}_{0n}^{V_0} \approx \sum_{n_s=1}^{N^s} \sum_{m=1}^{M} \sum_{n=1}^{N} \mathbf{U}'\left(\mathbf{y}_n, \bar{\mathbf{x}}(\xi_m,\eta_n)\right) \dot{\mathbf{b}}_0\left(\bar{\mathbf{x}}(\xi_m,\eta_n)\right) J(\mathbf{s})\, J_{\xi}^{n_s}\, W_m W_n \tag{37}$$

where N^s is the number of integration regions and M and N are the number of integration points in ξ and η directions respectively. To determine the number of Gauss points necessary for an accurate integration, we consider that the Kernel $\mathbf{U}'$ is $O(1/r)$ so the number of integration points has to be increased if $\mathbf{y}_n$ is close to V_0.

If the integration region includes the collocation point $\mathbf{y}_n$, then the integrand tends to infinity as the point is approached and a procedure used to deal with weakly sin-

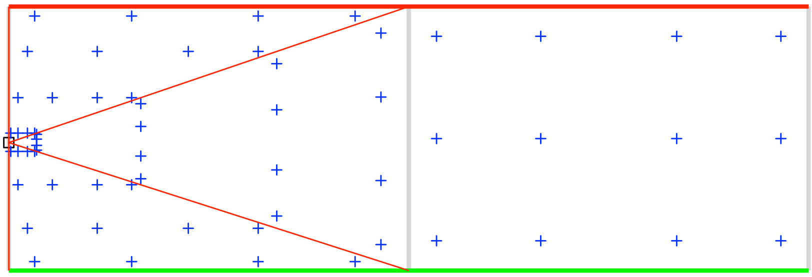

Figure 3: Subdivision into integration regions when the collocation point (marked by a square) is inside V_0. The grey line indicates a subdivision into 2 integration regions and red thin lines the subdivision into triangular subregions. The locations of Gauss points are marked with crosses

gular integrals in three dimensional BEM, can be invoked, which involves triangular subregions.

In this approach we perform the integration in a local coordinate system, where the Jacobian tends to zero as the singularity point is approached. For this we divide the integration region into two, three or four triangular sub-regions depending on whether the collocation point is at a corner, edge or inside. The procedure is well documented in [24] and leads to the following expression:

$$\begin{aligned} \mathbf{F}_{0n}^{V_0} &= \sum_{n_s=1}^{N^s} \sum_{k=1}^{K^\triangle} \int_{-1}^{+1}\int_{-1}^{+1} \mathbf{U}'(\mathbf{y}_n,\mathbf{x})\, \mathbf{b}_0(\bar{\mathbf{x}}(\xi,\eta))\, J(\mathbf{s})\, J_{\triangle_k} J_\xi^{n_s}\, \mathrm{d}\xi\, \mathrm{d}\eta \\ &\approx \sum_{n_s=1}^{N^s} \sum_{k=1}^{K^\triangle} \sum_{m=1}^{M} \sum_{\ell=1}^{L} \mathbf{U}'(\mathbf{y}_n,\mathbf{x})\, \mathbf{b}_0(\bar{\mathbf{x}}(\xi_m,\eta_\ell))\, J(\mathbf{s})\, J_{\triangle_k} J_\xi^{n_s}\, W_m W_l \end{aligned} \tag{38}$$

where $K^\triangle$ is the number of triangles and $J_{\triangle_k}$ is the Jacobian of the transformation from the square to the triangular subregion. An example of this subdivision is shown in Figure 3.

In order to limit the number of internal point evaluations we propose to compute results on grid points inside V_0 and interpolate to the required Gauss point locations [16]. Since the right hand side has to be evaluated for every iteration step it is convenient to precompute matrices that multiply with the values of $\mathbf{b}_0$ at grid points. The value of body force at a location s,t is given by:

$$\mathbf{b}_0(s,t) = \sum_{k=1}^{K} M_k(s,t)\, \mathbf{b}_{0k} \tag{39}$$

where $M_k(s,t)$ are interpolation functions, K is the number of grid points and $\mathbf{b}_{0k}$ are values of body force at grid points. With this the following matrix equation can be written:

$$\{\mathbf{F}_0\} = [\mathbf{B}]\{\mathbf{b}_0\} \tag{40}$$

where $\{\mathbf{b}_0\}$ is a vector of body force values at grid points and $[\mathbf{B}] = \left([\mathbf{B}]^S - [\mathbf{B}]^V\right)$, the sub-matrices of which are given by

$$\mathbf{B}^S_{nk} = \int_{S_0} \mathbf{U}\left(\mathbf{y}_n, \bar{\mathbf{x}}\right) M_k \, \mathbf{N} \, \mathrm{d}S_0 \tag{41}$$

and

$$\mathbf{B}^V_{nk} = \int_{V_0} \mathbf{U}'\left(\mathbf{y}_n, \bar{\mathbf{x}}\right) M_k \, \mathrm{d}V_0 \tag{42}$$

7. Computation of results inside V_0

The solution algorithm requires the evaluation of perturbation velocities inside the inclusion V_0. The velocity vector $\mathbf{v}$[4] at any internal point $\mathbf{y}_i$ can be computed by

$$\begin{aligned} \mathbf{v}\left(\mathbf{y}_i\right) = &\int_S \mathbf{U}\left(\mathbf{y}_i, \mathbf{x}\right) \mathbf{t}\left(\mathbf{x}\right) \mathrm{d}S - \int_S \mathbf{T}\left(\mathbf{y}_i, \mathbf{x}\right) \dot{\mathbf{u}}\left(\mathbf{x}\right) \mathrm{d}S \\ &- \int_{S_0} \mathbf{U}\left(\mathbf{y}_i, \bar{\mathbf{x}}\right) \mathbf{t}_0\left(\bar{\mathbf{x}}\right) \mathrm{d}S_0 + \int_{V_0} \mathbf{U}'\left(\mathbf{y}_i, \bar{\mathbf{x}}\right) \mathbf{b}_0\left(\bar{\mathbf{x}}\right) \mathrm{d}V_0 \end{aligned} \tag{43}$$

The above equation be written in matrix notation as:

$$\{\mathbf{v}\} = [\mathbf{A}]\,\{\mathbf{t}\} - [\mathbf{C}]\,\{\dot{\mathbf{u}}\} + [\mathbf{D}]\,\{\mathbf{b}_0\} \tag{44}$$

where matrices $[\mathbf{A}]$ and $[\mathbf{C}]$ are assembled from element contributions of Kernel basis function products and matrix $[\mathbf{D}] = \left([\mathbf{D}]^V - [\mathbf{D}]^S\right)$ where $[\mathbf{D}]^V$ and $[\mathbf{D}]^S$ are computed as shown in equation (41) and (42) and replacing $\mathbf{y}_n$ with $\mathbf{y}_i$.

8. Iterative procedure

There are two possibilities for the iterative procedure: modified Newton-Raphson or full Newton-Raphson. In the former the left hand side of the system of equations is not changed and only a new right hand side is computed at each iteration, whereas in the latter the left hand side is changed at every iteration. All test examples are pure Dirichlet problems, i.e. $\{\mathbf{a}\} = \{\mathbf{t}\}$ in Equation (21) and therefore the unknown are boundary tractions $\{\mathbf{t}\}$ and the known values are perturbation velocities $\{\dot{\mathbf{u}}\}$ at the boundary. Without loss of generality, the algorithms are detailed for this special case as the extension to mixed boundary conditions is trivial.

[4] The velocities at internal points are referred to as $\mathbf{v}$ to distinguish them from the boundary velocities $\dot{\mathbf{u}}$.

8.1. Modified Newton-Raphson

The iterative procedure for modified Newton-Raphson is essentially the same as used in [4] and is shown in Algorithm 1.

```
INITIALIZATION;
solve for tractions ${\mathbf{t}}^0$ using Equation (45);
compute velocities at internal points ${\dot{\mathbf{v}}}^0$ using Equation (47);
compute body forces ${\mathbf{b}_0}^0$ using Equation (13);
compute vector $\{\mathbf{F}\}_0^0$ using Equation (40);
ITERATION;
for k=1:number of iterations do
    solve for tractions $\{\mathbf{t}\}^k$ using Equation (46);
    set $\{\mathbf{t}\} = \beta\{\mathbf{t}\}^k + (1-\beta)\{\mathbf{t}\}^{k-1}$;
    compute velocities at internal points $\{\dot{\mathbf{v}}\}^k$ using Equation (48);
    set $\{\mathbf{v}\} = \beta\{\mathbf{v}\}^k + (1-\beta)\{\mathbf{v}\}^{k-1}$;
    compute body forces $\{\mathbf{b}_0\}^k$ using Equation (13);
    compute vector $\{\mathbf{F}\}_0^k$ using Equation (40);
    compute residual;
    if residual < Tolerance then
        exit
    end
end
```

Algorithm 1: Modified Newton-Raphson

For the first iteration the unknowns are computed by

$$[\mathbf{U}]\{\mathbf{t}\}^0 = \{\mathbf{F}\} \tag{45}$$

For the subsequent iteration we have

$$[\mathbf{U}]\{\mathbf{t}\}^k = \{\mathbf{F}\} + \{\mathbf{F}\}_0^{k-1} \tag{46}$$

where k is an iteration counter. The velocities at internal points are computed by:

$$\{\mathbf{v}\}^0 = [\mathbf{A}]\{\mathbf{t}\}^0 - [\mathbf{C}]\{\dot{\mathbf{u}}\} \tag{47}$$

for the first iteration and

$$\{\mathbf{v}\}^k = [\mathbf{A}]\{\mathbf{t}\} - [\mathbf{C}]\{\dot{\mathbf{u}}\} + [\mathbf{D}]\{\mathbf{b}_0\} \tag{48}$$

for the subsequent iterations. To ensure convergence for higher Reynolds numbers we apply a relaxation scheme, i.e. the tractions and velocities are computed by a combination of new and previous values:

$$\begin{aligned}\{\mathbf{t}\} &= \beta\,\{\mathbf{t}\}^{k} + (1-\beta)\,\{\mathbf{t}\}^{k-1} \\ \{\mathbf{v}\} &= \beta\,\{\mathbf{v}\}^{k} + (1-\beta)\,\{\mathbf{v}\}^{k-1}\end{aligned} \tag{49}$$

where β is a relaxation coefficient ($0 < \beta < 1$).

8.2. *Full Newton-Raphson*

The iterative procedure for full Newton-Raphson is similar to the one published in [10] and shown in Algorithm 2.

```
INITIALIZATION;
solve for tractions $\{\mathbf{t}\}^0$ using Equation (45);
compute velocities at internal points $\{\dot{\mathbf{v}}\}^0$ using Equation (47);
compute body forces $\{\mathbf{b}_0\}^0$ using Equation (13);
compute vector $\{\mathbf{F}\}_0^0$ using Equation (40);
ITERATION;
for k=1:number of iterations do
    compute $\left[\frac{\partial\{\mathbf{b}_0\}}{\partial\{\mathbf{v}\}}\right]^k$ and update the tangent operator in Equation (51);
    solve the system of equations (51) to compute $\{\Delta\mathbf{t}\}^{k+1}$ and $\{\Delta\mathbf{v}\}^{k+1}$;
    update the tractions by Equation (52a);
    update the velocities by Equation (52b);
    compute body forces $\{\mathbf{b}_0\}^k$ using Equation (13);
    compute vector $\{\mathbf{F}\}_0^k$ using Equation (40);
    compute the residuals $\{\mathbf{R}_t\}^{k+1}$ and $\{\mathbf{R}_v\}^{k+1}$ by Equations (50);
end
if resperc < Tolerance then
    exit
end
```

Algorithm 2: Full Newton-Raphson

For the first iteration the unknowns and the velocities are computed by the same equations of the modified Newton Raphson approach, that is by Equation (45) and Equation (47), respectively. For the subsequent iterations we introduce the following residuals:

$$\{\mathbf{R}_t\}^k = -[\mathbf{U}]\,\{\mathbf{t}\}^k + \{\mathbf{F}\} + \{\mathbf{F}\}_0^{k-1} \tag{50a}$$

$$\{\mathbf{R}_v\}^k = -\{\mathbf{v}\}^k + [\mathbf{A}]\,\{\mathbf{t}\}^k - [\mathbf{C}]\,\{\dot{\mathbf{u}}\} + [\mathbf{D}]\,\{\mathbf{b}_0\}^{k-1} \tag{50b}$$

The traction and velocity increments are computed by solving the following first-order Taylor expansions:

$$\begin{bmatrix} -[\mathbf{U}] & \left[\frac{\partial\{\mathbf{F}\}_0}{\partial\{\mathbf{v}\}}\right] \\ & \\ [\mathbf{A}] & [\mathbf{D}]\left[\frac{\partial\{\mathbf{b}_0\}}{\partial\{\mathbf{v}\}}\right]-[\mathbf{I}] \end{bmatrix}^k \left\{ \begin{array}{c} \{\Delta\mathbf{t}\} \\ \\ \{\Delta\mathbf{v}\} \end{array} \right\}^{k+1} = -\left\{ \begin{array}{c} \{\mathbf{R}_t\} \\ \\ \{\mathbf{R}_v\} \end{array} \right\}^k \tag{51}$$

The total tractions and velocities are computed by:

$$\{\mathbf{t}\}^{k+1} = \{\mathbf{t}\}^k + \{\Delta\mathbf{t}\}^{k+1} \tag{52a}$$

$$\{\mathbf{v}\}^{k+1} = \{\mathbf{v}\}^k + \{\Delta\mathbf{v}\}^{k+1} \tag{52b}$$

where:

$$\frac{\partial\{\mathbf{F}\}_0}{\partial\{\mathbf{v}\}} = [\mathbf{B}]\left[\frac{\partial\{\mathbf{b}_0\}}{\partial\{\mathbf{v}\}}\right] \tag{53}$$

and

$$\left[\frac{\partial\{\mathbf{b}_0\}}{\partial\{\mathbf{v}\}}\right] = \begin{bmatrix} \frac{\partial\{\mathbf{b}_0\}}{\partial\{\mathbf{v}\}}(\mathbf{y}_1) & \{\mathbf{0}\} & \cdots & \{\mathbf{0}\} \\ & & & \\ \{\mathbf{0}\} & \frac{\partial\{\mathbf{b}_0\}}{\partial\{\mathbf{v}\}}(\mathbf{y}_2) & \cdots & \{\mathbf{0}\} \\ \vdots & \vdots & \vdots & \\ \{\mathbf{0}\} & \cdots & \cdots & \frac{\partial\{\mathbf{b}_0\}}{\partial\{\mathbf{v}\}}(\mathbf{y}_{N_\Omega}) \end{bmatrix} \tag{54}$$

where N_Ω is the number of internal points.

The derivatives of $\mathbf{b}_0$ are given by:

$$\frac{\partial\{\mathbf{b}_0\}}{\partial\{\mathbf{v}\}}(\mathbf{y}_i) = \begin{bmatrix} 2\dot{u}_x(\mathbf{y}_i)+u_x^0(\mathbf{y}_i) & 0 \\ & \\ \dot{u}_y(\mathbf{y}_i)+u_y^0(\mathbf{y}_i) & \dot{u}_x(\mathbf{y}_i) \\ & \\ \dot{u}_y(\mathbf{y}_i) & \dot{u}_x(\mathbf{y}_i)+u_x^0(\mathbf{y}_i) \\ & \\ 0 & 2\dot{u}_y(\mathbf{y}_i)+u_y^0(\mathbf{y}_i) \end{bmatrix} \quad \text{if } \mathbf{y}_i \in \Omega \tag{55}$$

and

$$\frac{\partial\{\mathbf{b}_0\}}{\partial\{\mathbf{v}\}}(\mathbf{y}_i) = [\mathbf{0}]_{4\text{x}2} \quad \text{otherwise} \tag{56}$$

The iteration is halted when $resperc = \frac{\{\Delta\mathbf{t}\}^{k+1}}{\{\mathbf{t}\}^k}$ is less than a pre-set $Tolerance$

9. Numerical results

The implementation of the theory is tested here on two examples. In the first one the results are compared with an available fine grained solution in order to ascertain that good quality of results can be obtained. The second one is used to demonstrate the

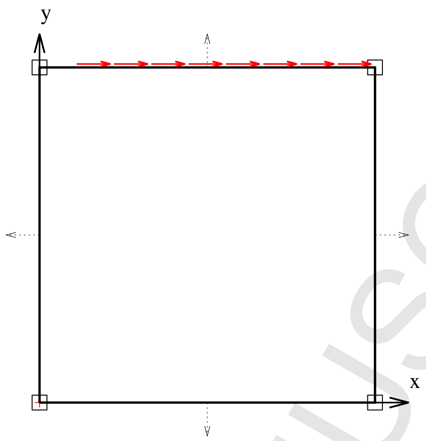

Figure 4: Flow in cavity: Definition of geometry with 4 linear patches and control points. Also shown is the Dirichlet boundary condition on top.

superiority of the proposed approach for describing more complex practical geometries.

9.1. Flow in cavity

A driven cavity problem has become a standard test problem for fluid dynamics codes. An incompressible fluid of uniform viscosity ($\mu = 1$) is confined within a square region of dimension $H = 1 \times 1$. The fluid velocities on the bottom, left and right are fixed at zero, while a uniform velocity $u_x = 1$ is specified at the top, which is tapered off to zero very near the corners. The Reynolds number is defined as $Re = \rho\, U\, H/\mu$. The example is tested for two different Reynolds numbers (= 100, 400) by changing the value of ρ

9.1.1. Definition of geometry and boundary conditions

The boundary of the problem is defined by 4 linear NURBS patches as shown in Figure 4. Using a geometry independent field approximation, the non-zero Dirichlet boundary condition along the top NURBS patch was defined using the following Knot vector for the basis function $R_k^u(s)$

$$\Xi = [0, 0, 0.05, 0.95, 1, 1] \tag{57}$$

with all weights equal to 1. The parameters were specified as:

$$\dot{\mathbf{u}}^e = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \tag{58}$$

This means that the velocity vector at the top is tapered off to zero very near to the corners.

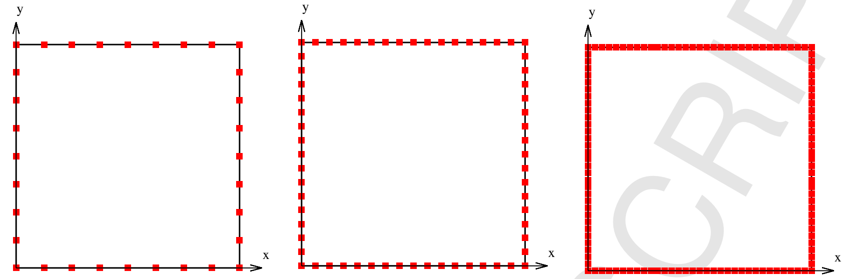

Figure 5: Refinement of solution: Location of collocation points for 3, 7 and 15 knot insertions for each patch

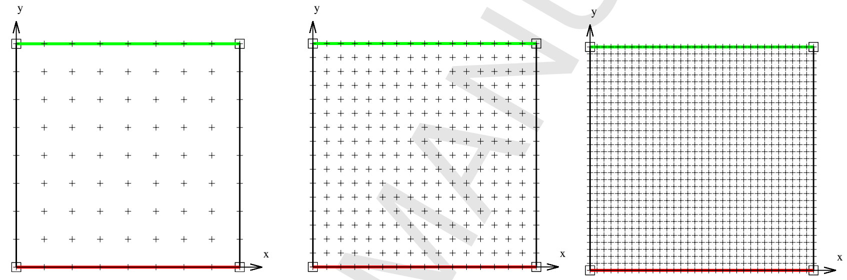

Figure 6: Definition of domain for volume integration with two NURBS curves marked red and green and location of the internal points for the three refinement stages

9.1.2. Approximation of the unknown and refinement

The approximation of the boundary unknown (in this case **t**) was achieved by inserting knots and by order elevating the basis functions for describing the geometry (from linear to quadratic). Three different refinements were investigated and the resulting locations of collocation points computed using Greville abscissa [25] are shown in Figure 5.

9.1.3. Approximation of body forces inside domain

The domain for the volume integration was defined by 2 NURBS curves. The refinement of the boundary values was accompanied by an increased number of internal points as shown in Figure 6. Quadratic interpolation between the points was assumed. The number of degrees of freedom and the number of internal points for the different meshes is shown in Table 1.

9.1.4. Results and comparison

Results were computed for the three meshes and two different Reynolds numbers with a modified and full Newton-Raphson method. Figure 7 shows the velocity vectors for the two Reynolds numbers. A shift in the vortex centre can be clearly seen.

Mesh	Degrees of freedom	No. of internal points
mesh1	64	81
mesh2	128	289
mesh3	256	1089
Reference [2]	-	16641

Table 1: Mesh statistics

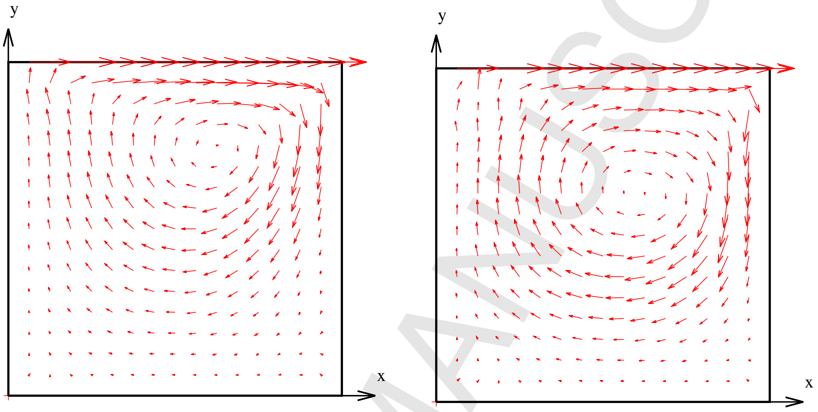

Figure 7: Forced cavity flow: Resulting velocity vectors for Re=100 and Re=400

Figures 8 and 9 show a comparison of the results obtained with the different meshes and iteration methods. The variation of x-velocities along a vertical line though the middle agree well with the extremely accurate published solution, except for mesh 1 and Re=400. It seems that for this Reynolds number mesh 1 is not adequate.

There is very little difference between the results obtained with modified and full Newton-Raphson. However, as shown in Table 2 there is a large difference with respect to the number of iterations required to achieve convergence to a tolerance of 10^{-4} with the modified Newton-Raphson requiring a significant higher number of iterations.

		mesh 1	mesh 2	mesh3
Re=100	modified Newton-Raphson	15	19	18
	full Newton-Raphson	3	3	3
Re=400	modified Newton-Raphson	24	39	100
	full Newton-Raphson	-	5	5

Table 2: Number of iterations required for convergence

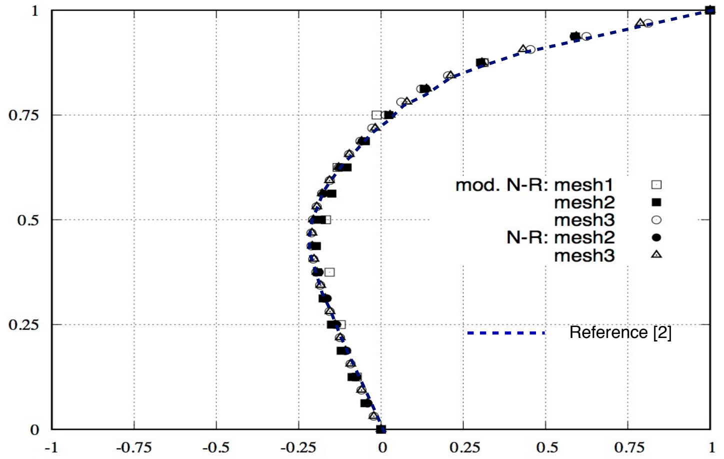

Figure 8: Comparison of velocity in x-direction along a vertical line through centre for Re=100 together with the reference solution

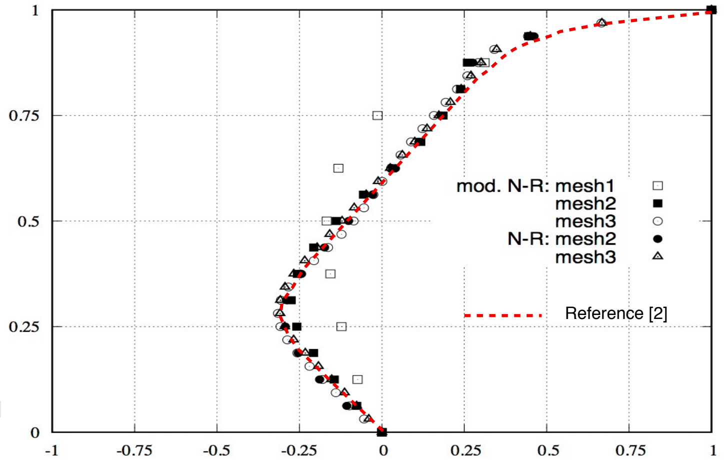

Figure 9: Comparison of velocity in x-direction along a vertical line through centre for Re=400 together with the reference solution

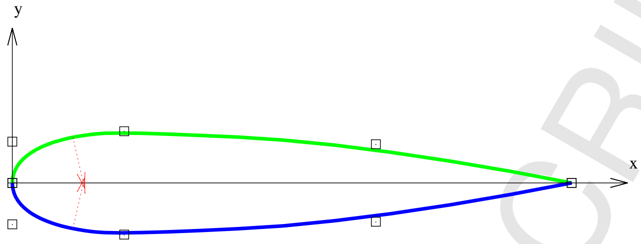

Figure 10: Description of NACA0018 airfoil with two patches and 5 control points and basis functions of order 2 each

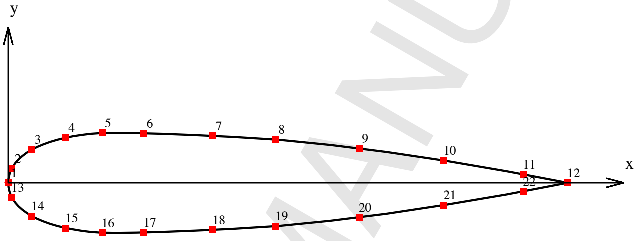

Figure 11: Location of collocation points after refinement by knot insertion

9.2. Airfoil

As a practical example we show the simulation of the flow past an airfoil. We chose the NACA0018 airfoil, where detailed coordinates are available on the internet (see for example www.airfoiltools.com). The airfoil is placed in an infinite domain and subjected to a stream. The free stream velocity is defined by $u_x^0 = 0.984, u_y^0 = 0.173$ which corresponds to a unit velocity vector inclined at 10 degrees.

9.2.1. Description of boundary geometry

NURBS are ideally suited for describing such shapes with very few parameters. Only 5 control points and basis functions of order 2 are able to describe the shape fairly accurately as is shown in Figure 10. More accurate descriptions can be obtained with more control points and higher basis function order. An important fact is that the approximated shape has a C^I continuity throughout each patch, which would not be the case if piecewise continuous Lagrange polynomials are used.

9.2.2. Approximation of the boundary unknown

We use the geometry independent field approximation approach for the variation of the unknown boundary values by inserting 7 knots into the knot vector that is used to describe the boundary, for each patch. The resulting locations of the collocation points are shown in Figure 11.

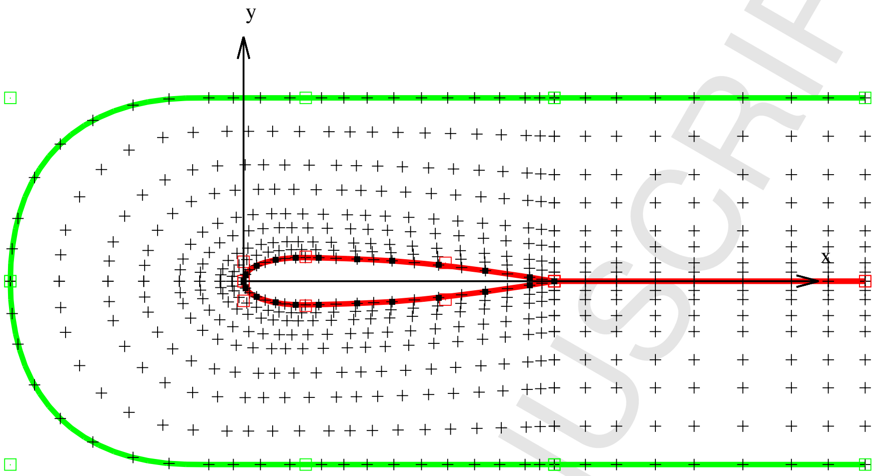

Figure 12: Definition of domain for volume integration

9.2.3. *Approximation of the body forces*

The task left is the description of the domain for the volume integration. Here we use 4 subdomains defined by 2 NURBS curves each. The discretisation and the locations of the internal points are shown in Figure 12. As with the previous example a quadratic interpolation between points is assumed.

9.2.4. *Results*

The problem was analysed by choosing the parameters in such a way that a Reynolds number of 10 was achieved. The resulting velocity vectors are shown in Figure 13.

10. Summary and Conclusions

An implementation of the BEM for incompressible viscous flow was presented. The main novelty of our approach is that instead of Lagrange polynomials, NURBS are used for both the description of the geometry and the approximation of the boundary values. A geometry independent field approximation was used, which means that the approximation of unknown boundary values is completely uncoupled from the description of the geometry. It was shown that the use of NURBS for the description of the geometry results in an accurate, smooth description with few parameters. Another novelty of the paper is that instead of cells, that have been used in the literature for the evaluation of the volume integrals, volumes are defined by bounding NURBS curves and a mapping is used.

For the solution of the non-linear system of equations modified and full Newton-Raphson methods are used. Although these two methods have been applied in the cited publications using conventional BEM approaches, this is the first time the two approaches have been compared on the test example of a forced cavity flow. It was shown

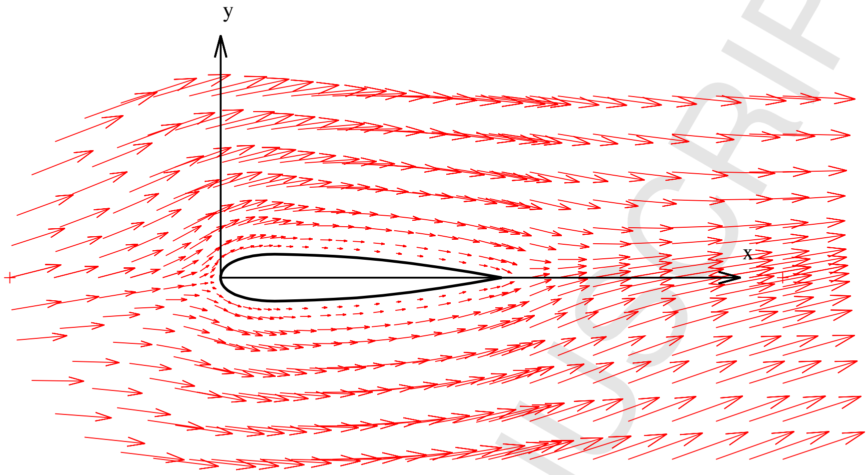

Figure 13: Resulting velocity vectors for Re=10

that for higher Reynolds numbers and for the modified Newton-Raphson method a high number of iterations is required for convergence. The results of the simulation with the new technology agree very well with very accurate results available in the literature. On a practical example of an airfoil the advantages of our approach (accurate definition of geometry with few parameters, solution with few unknowns...) were demonstrated. Only plane problems are discussed here, the extension to 3-D problems is planned.

Whilst the paper is concerned only with steady, viscous incompressible flow an extension of the BEM to deal with unsteady, compressible flow is possible (for details see [10]). It is hoped that this paper provides some impetus for applying isogeometric methods and for much needed research in this area of application.

Appendix A. Fundamental solutions

The fundamental solutions for equations (2) for the velocity at point $\mathbf{x}$ due to point sources at $\mathbf{y}$ is

$$\mathbf{U}(\mathbf{y},\mathbf{x}) = c(\mathbf{R}+\mathbf{I}\ln\frac{1}{r}) \tag{A.1}$$

with $c = \frac{1}{4\pi\mu}$, $\mathbf{I}$ is the identity matrix and

$$\mathbf{R} = \begin{pmatrix} r_1^2 & r_1 r_2 \\ r_2 r_1 & r_2^2 \end{pmatrix} \tag{A.2}$$

For the tractions acting on a boundary S:

$$\mathbf{T}(\mathbf{y},\mathbf{x}) = -\frac{\cos\theta}{\pi r}\mathbf{R} \tag{A.3}$$

In the above r is the distance between **x** and **y**, n_i is a unit vector normal to S and

$$\begin{aligned} r_i &= \frac{1}{r}(x_i - y_i) \\ \cos\theta &= r_i n_i \end{aligned} \tag{A.4}$$

The derived fundamental solution is

$$U'_{ijk} = \frac{1}{4\pi\mu r}\left(\delta_{jk} r_i + \delta_{ik} r_j - \delta_{ij} r_k - 2 r_i r_j r_k\right) \tag{A.5}$$

where δ_{ij} is the Kronecker Delta.

References

[1] M. D. Gunzburger, Finite Element analysis for viscous incompressible flows, Academic Press, 1989.

[2] U. Ghia, K. N. Ghia, C. T. Shin, High-Re Solutions for incompressible flow using the Navier-Stokes equations and Multigrid Method, Journal of computational physics 48 (1982) 387–411.

[3] G. Dargush, P. Banerjee, Boundary Element Methods in Nonlinear Fluid Dynamics, Vol. 6 of Developments in boundary element methods, Elsevier, 1990, Ch. Advanced boundary element methods for steady incompressible thermoviscous flow.

[4] M. Aydin, R. T. Fenner, Boundary element analysis of driven cavity flow for low and moderate Reynolds numbers, International Journal for Numerical Methods in Fluids 37 (2000) 45–64.

[5] G. Dargush, P. K. Banerjee, A Boundary Element method for steady incompressible thermoviscous flow, International Journal for Numerical Methods in Engineering 31 (1991) 1605–1626.

[6] D. P. Henry, P. K. Banerjee, A new BEM formulation for two- and three-dimensional elastoplasticity using particular integrals, International Journal for Numerical Methods in Engineering 26 (9) (1988) 2079–2096.

[7] X.-W. Gao, A boundary element method without internal cells for two-dimensional and three-dimensional elastoplastic problems, Journal of Applied Mechanics 69 (2) (2002) 154–160.

[8] M. Ingber, A. Mammoli, M. Brown, A comparison of domain integral evaluation techniques for boundary element methods, International Journal for Numerical Methods in Engineering 52 (2001) 417–432.

[9] L. Wrobel, The Boundary Element Method, Volume 1, Applications in Thermo-Fluids and Acoustics, Wiley, 2002.

[10] X.-W. Gao, A boundary-domain integral equation method in viscous fluid flow, International Journal for Numerical Methods in Fluids 45 (2004) 463–484.

[11] X.-W. Gao, A promising boundary element formulation for three-dimensional viscous flow, International Journal for Numerical Methods in Fluids 47 (2004) 19–43.

[12] T. Hughes, J. Cottrell, Y. Bazilevs, Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement, Computer Methods in Applied Mechanics and Engineering 194 (39–41) (2005) 4135–4195.

[13] B. Marussig, J. Zechner, G. Beer, T.-P. Fries, Fast isogeometric boundary element method based on independent field approximation, Computer Methods in Applied Mechanics and Engineering 284 (2015) 458 – 488, isogeometric Analysis Special Issue.

[14] G. Beer, B.Marussig, J. Zechner, A simple approach to the numerical simulation with trimmed CAD surfaces, Computer Methods in Applied Mechanics and Engineering 285 (2015) 776–790.

[15] G. Beer, B. Marussig, J. Zechner, C. Duenser, T.-P. Fries, Isogeometric boundary element analysis with elasto-plastic inclusions. part 1: plane problems, Computer Methods in Applied Mechanics and Engineering 308 (2016) 552–570.

[16] G. Beer, V. Mallardo, E. Ruocco, B. Marussig, J. Zechner, C. Duenser, T. P. Fries, Isogeometric boundary element analysis with elasto-plastic inclusions. part 2: 3-d problems, Computer Methods in Applied Mechanics and Engineering 315 (2017) 418–433.

[17] R. Simpson, S. Bordas, J. Trevelyan, T. Rabczuk, A two-dimensional isogeometric boundary element method for elastostatic analysis, Computer Methods in Applied Mechanics and Engineering 209–212 (0) (2012) 87–100.

[18] M. Scott, R. Simpson, J. Evans, S. Lipton, S. Bordas, T. Hughes, T. Sederberg, Isogeometric boundary element analysis using unstructured T-splines, Computer Methods in Applied Mechanics and Engineering 254 (0) (2013) 197 – 221.

[19] B. Marussig, G. Beer, C. Duenser, Isogeometric boundary element method for the simulation in tunneling, Applied Mechanics and Materials 553 (2014) 495–500.

[20] G. Beer, B. Marussig, J. Zechner, C. Duenser, T.-P. Fries, Boundary element analysis with trimmed NURBS and a generalized IGA approach, in: E. Oñate, J. Oliver, A. Huerta (Eds.), 11th World Congress on Computational Mechanics (WCCM XI), 2014, pp. 2445–2456.

[21] G. Beer, Mapped infinite patches for the NURBS based boundary element analysis in geomechanics, Computers and Geotechnics 66 (2015) 66–74.

[22] V. Mallardo, E. Ruocco, An improved isogeometric Boundary Element Method approach in two dimensional elastostatics, CMES - Computer Modeling in Engineering and Sciences 102 (2014) 373–391.

[23] V. Mallardo, E. Ruocco, A NURBS boundary-only approach in elasticity, European Journal of Computational Mechanics 25 (2016) 71–90.

[24] G. Beer, Advanced numerical simulation methods - From CAD Data directly to simulation results, CRC Press/Balkema, 2015.

[25] R. W. Johnson, Higher order B-spline collocation at the Greville abscissae, Applied Numerical Mathematics 52 (1) (2005) 63 – 75.