Verifying the Geometry in the 2D Boundary Element Method

Introduction

In the application of the boundary element method (BEM) to a particular boundary value problem[1], the *boundary* of the domain must be defined and this is the most important 'geometrical' information that is held in order to describe the problem that is to be solved. By using elements or panels to describe the boundary, there is a resulting issue of approximation for most boundaries. However, this can be accepted in the context of numerical error and the practicalities of using the method. The care in which the boundary is defined - for example maintaining a near-uniform element size or grading elements in order to offset expected errors in certain regions – can also help to minimise the numerical error. However, this document is not directly about the management of the expected numerical error, rather it is about verifying that the boundary that is communicated within the boundary element method is correctly defined.

An error in the definition of the boundary would usually lead to far greater or catastrophic 'error' in the results than the normal numerical error that is reasonably expected in the typical correct use of the BEM. Such errors in setting up the input into the boundary element method could be left to the practitioner. However, the users of the BEM – or of any computational method, are often not familiar, and perhaps should not need to be familiar, with the underlying theoretical framework and hence it would be a useful addendum to the BEM to provide a routine verification of the input geometry. Any such verification is not expected to be entirely water-tight, however this should not mean that developers of the method should not apply verification techniques. In this document a number of strategies will be applied to the geometrical data input in the boundary element method. It is likely that the verification methods will not be exhaustive and that further techniques can be appended in the future. It is also possible that the techniques can be adapted for other methods in which a boundary or surface mesh is included.

The definition and verification of the boundary in the boundary element method is independent of the governing equation or type of problem for which a solution is sought. In this document the discussion is restricted to two-dimensional boundary element methods, and boundaries are approximated by straight line panels[2], although most of the issues and methods discussed will also be applicable or adaptable in three dimensions and to higher order approximating panels.

In order to susbstantiate the verification techniques that are developed in this document, the tests have been emulated in VBA in Excel in a module *VGEOM2.bas*. The spreadsheet environment is is utilised in order to demonstrate the techniques clear. The module is integrated into the VBA boundary element code LIBEM2.bas, which is included in the

[1] Boundary Value Problems and Boundary Conditions
[2] Representation of a line by straight line elements

spreadsheet demonstration of the boundary element method in the Excel spreadsheet LIBEM2.xlsm[3].

Background

In order to verify the boundary, we need to be clear of the meaning of the term 'boundary' in the context of the BEM. Although normally hidden from the users of the method, the method itself must conform to the direction of the integral equation formulation that forms its basis, and this has implications for the classification of the boundary. From the perspective of developing the boundary element method itself, the goal is to work towards its most general implementation. The boundary may therefore be connected, however, the boundary may be a system of individually connected boundaries; for example each disconnected from all other individual boundaries. An important distinction is for the problem to be defined as an interior problem or an exterior problem, as this also has implications for the boundary integral equation formulation.

The following diagrams illustrate the generalised boundary element method that is being alluded to and it stems partially from the work on the 'boundary and shell element method'[4]. The generalised interior domain is illustrated in Figure 1. The domain is shaded in blue and the boundary is shaded in black and the boundary may also be included in the domain. The solution is normally sought at selected points within the domain, or on the boundary. The white areas are void. The boundary consists of an outer closed boundary with two 'holes' or closed boundaries within the outer boundary. There are also two open boundaries, which could represent cracks if the domain is made up of solid material or obstructions (shields) in a fluid domain. The open boundaries have the property that there are normally two solutions associated with each point thereon; there would normally be a different solution either side of the open boundary.

[3] LIBEM2.xlsm
[4] The Boundary and Shell Element Method

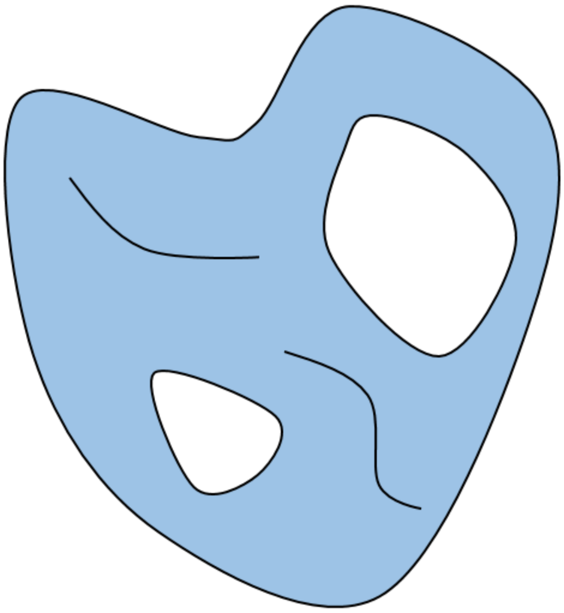


Figure 1. The generalised interior domain

The generalised exterior domain is illustrated in Figure 2, consisting of a domain with two closed boundaries and two open boundaries. The domain is shaded in blue, but in this case it is a (theoretically) infinite domain exterior to a set of closed and open boundaries. The boundary is shaded in black and the boundary may also be included in the domain. The solution is normally sought at selected points within the domain, or on the boundary. The white areas are void; if a solution is sought there then a value of zero is expected in the results from the boundary element method. In the illustration, the boundary consists of an outer closed boundary with two 'holes' or closed boundaries within the outer boundary. There are also two open boundaries, which represent discontinuities; for example cracks if the domain is made up of solid material or obstructions (shields) in a fluid domain.

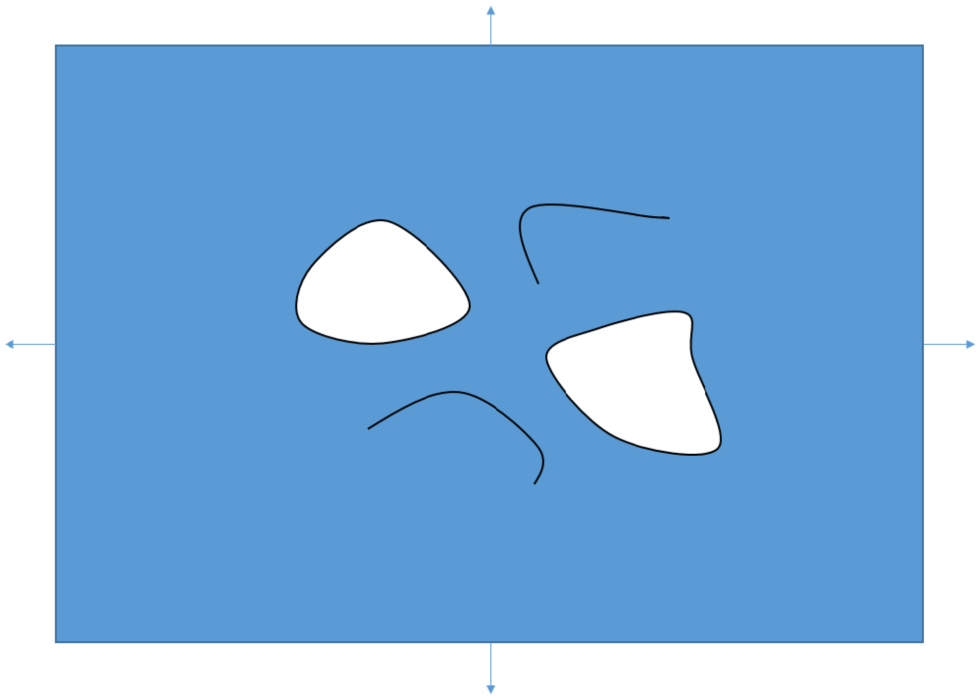


Figure 2. The generalised exterior domain

The boundary integral equations or boundary integrals that are solved within the method generally relate the value of the function at an *observation* point to an integral over a boundary. In the initial part of the boundary element method the observation points are (usually) all on the boundary[5], for example in the case of using collocation[6] as a solution method for the boundary integral equation, the observation points are the collocation points. When observation points lie on the boundary, the integrals are singular and their evaluation requires special treatment.

In the second part of the boundary element method, the observation points are within the domain of the problem; the points at which the solution is sought. An important conPanelsquence of this is that - for point near to the boundary – the integrals are nearly-singular (ie strongly varying) and hence would require more accurate integration methods to maintain the level of accuracy expected in the overall method. With the actual boundary being replaced by an approximate boundary, there is also a blurring of the issue of whether a point in the vicinity of the received approximate boundary is either on the boundary or at one side or the other. Hence, in the geometrical definition of the domain (solution) points, it is important to check if the point is actually within the defined domain or on the defined boundary. For points 'close' to the defined boundary, it would be useful to at least warn the user of the potential loss of accuracy, unless the method is able to adapt to this particular situation.

The most crucial distinction for each boundary is its classification as closed or open, since each has its own formulation. However, an open boundary is an infinitesimally thin discontinuity in the field, which would normally be not entirely physically appropriate. However if a boundary is 'thin' – even in part – then there are dangers of degeneracy in

[5] Outline of the Boundary Element Method
[6] Solution of Fredholm Integral Equations by Collocation

the method if such a boundary is classified as closed. In this latter case, points either side of the boundary will almost coincide, the corresponding integrals over the boundaries will be similar and hence rows in the matrices will effectively be close to being repeated and it follows that an effectively rank deficient or ill-conditioned matrix[7] and correspondingly significant loss of accuracy would ensue.

Definition of the geometry in the boundary element method

Before we can verify the geometrical set up in the boundary element method, we have to consider the method by which the boundary is to be represented. For a two-dimensional problem, the focus is on the simplest boundary approximation – that of using straight line panels – but we also consider alternative methods. As an example, let us consider a mesh similar to the one that has been used a number of times as part of the construction of a test problem[8]. The boundary is that is a unit square, as illustrated in Figure 3. For the purposes of the boundary element method, the boundary of the square is represented by 32 panels of uniform size.

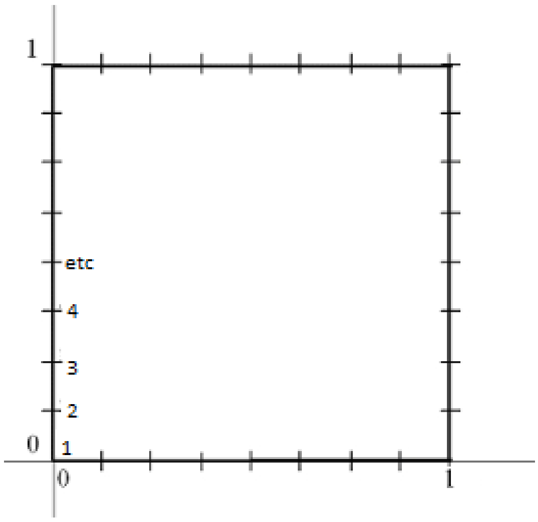


Figure 3. The unit square domain and the 32 panels.

Nodes on the boundary											
Index	x	y	Index	x	y	Index	x	y	Index	x	y
1	0	0	9	0	1	17	1	1	25	1	0
2	0	0.125	10	0.125	1	18	1	0.875	26	0.875	0
3	0	0.25	11	0.25	1	19	1	0.75	27	0.75	0

[7] Condition and Condition Number of a Matrix
[8] A*BEM2 and L*BEM2 codes on www.boundary-element-method.com

4	0	0.375	12	0.375	1	20	1	0.625	28	0.625	0
5	0	0.5	13	0.5	1	21	1	0.5	29	0.5	0
6	0	0.625	14	0.625	1	22	1	0.375	30	0.375	0
7	0	0.75	15	0.75	1	23	1	0.25	31	0.25	0
8	0	0.875	16	0.875	1	24	1	0.125	32	0.125	0

Table 1. The coordinates of the nodes on the boundary of the square.

Panels that constitute the square											
Index	Node 1	Node 2	Index	Node 1	Node 2	Index	Node 1	Node 2	Index	Node 1	Node 2
1	1	2	9	9	10	17	17	18	25	1	0
2	2	3	10	10	11	18	18	19	26	0.875	0
3	3	4	11	11	12	19	19	20	27	0.75	0
4	4	5	12	12	13	20	20	21	28	0.625	0
5	5	6	13	13	14	21	21	22	29	0.5	0
6	6	7	14	14	15	22	22	23	30	0.375	0
7	7	8	15	15	16	23	23	24	31	0.25	0
8	8	9	16	16	17	24	24	25	32	0.125	0

Table 2. The panels that define the boundary of the square.

In the case of a two-dimensional boundary, such as the square above, it is possible to arrange the nodes and the panels one-by-one along its extent of the boundary, as in the tables above. Also, for closed two-dimensional boundaries, defined in this way, it makes senPanels that the *Node 1* node of the first panel is the same as the *Node 2* node of the final panel: for open boundaries, *Node 1* of the first panel should not be the same as *Node 2* of the final panel. However, for general three-dimensional boundaries, it is not possible to sequence the nodes or panels around the boundary and hence it would make senPanels on the whole not to rely on this. In general, therefore, it would be useful to relax the ordering of nodes and panels.

In this example, there are 32 nodes ($N_V = 32$) and 32 panels ($N_S = 32$). For closed 2D boundaries, like this one, it is reasonable to expect that $N_S = N_V$ and similarly, for open 2D boundaries, it is expected that the number of panels is one less than the number of nodes. However, again, for three-dimensional boundaries, these relations do not apply. It is also possible that some of the nodes that are listed are not used in the definition of the boundary, for example they may be listed in order to be referenced by some other part of the boundary definition. Hence, in general, for validation purposes it seems appropriate not to expect any simple relationship between the number of panels and the number of nodes.

The underlying boundary integral formulations, rely on the values of the normal derivatives of functions on the boundary. Since the unit normal to the boundary could be defined in either direction, I is important to be clear and consistent about the direction of the normal. The convention used in this work is that for a closed domain, such as the unit square in the example above, the normal points outward. In order to be consistent, the panels are defined by their nodes in a clockwise direction around the square. For an interior problem, the unit normal to the surface, therefore points away from the domain and it follows that if there is a boundary (or boundaries) within an outer boundary, as illustrated in figure 1, then for consistency the nodes are defined counter-clockwise around the inner boundary. For open boundaries the normal can be defined to be pointing from one side or the other as long as this is consistent for every panel along the boundary and with the definition of the problem.

Although it is possible that a solution is only sought on the boundary, usually a domain solution is sought. In the second stage of the boundary element method, the solution is sought at selected points in the domain. In the case of the example with a unit square boundary, let us presume that it is an *interior* problem; the points at which the solution is sought must all be within the boundary of the square. The following table illustrates a typical set of coordinates of chosen interior points. Let N_P be the number of domain points. In the case of the data in the table below, $N_P = 5$.

Domain (Interior) Points		
Index	x	y
1	0.25	0.25
2	0.25	0.75
3	0.5	0.5
4	0.75	0.25
5	0.75	0.75

Table 3. The coordinates of the within the square, where the solution is sought.

As discussed earlier, the underlying formulation recommends that closed boundaries and open boundaries are treated separately, that is they are defined using separate tables. However it is possible to have just one table of nodal coordinates, with the panels on either the open or closed boundaries referencing that table. Hence for reasons of simplicity and economy, let it be assumed that there be just one table of nodal co-ordinates, one table listing the panels on the closed boundaries and one table listing the panels on the open boundaries.

Simple Verification Methods

In this section a set of tests are developed in order to verify the geometry that is input into the boundary element method. The tests consider both closed and open boundaries, and it is presumed that the boundary that the boundary is declared as open or closed when it is defined. The verification process is defined as a set of tests and, to some extent, they have to follow a particular order. For example one test may rely on a positive outcome or on a property of the boundary that has been determined. Once a test is failed the verification process is terminated.

The tests are generally based on boundaries defined by just a few nodes and panels, for reasons of simplicity and the economy of processing time. In order to initiate this, two simple boundaries are introduced in Table 4. In the table the indices that are not circled represent the nodes, which in this case are the four corners of the unit square, and the circled indices are the panels.

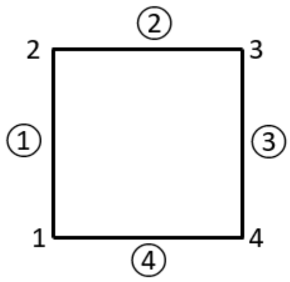
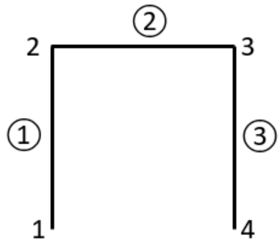
<table>
<tr><th>Illustration of the boundary</th><th>Nodes</th><th>Panels</th></tr>
<tr><td>1. Simple closed boundary
</td>
<td><table>
<tr><td>nodes</td><td>nNodes</td><td>4</td></tr>
<tr><td>index</td><td>x</td><td>y</td></tr>
<tr><td>1</td><td>0</td><td>0</td></tr>
<tr><td>2</td><td>0</td><td>1</td></tr>
<tr><td>3</td><td>1</td><td>1</td></tr>
<tr><td>4</td><td>1</td><td>0</td></tr>
</table></td>
<td><table>
<tr><td>panels</td><td>ns</td><td>4</td></tr>
<tr><td>index</td><td>qa</td><td>qb</td></tr>
<tr><td>1</td><td>1</td><td>2</td></tr>
<tr><td>2</td><td>2</td><td>3</td></tr>
<tr><td>3</td><td>3</td><td>4</td></tr>
<tr><td>4</td><td>4</td><td>1</td></tr>
</table></td></tr>
<tr><td>2. Simple open boundary
</td>
<td><table>
<tr><td>nodes</td><td>nNodes</td><td>4</td></tr>
<tr><td>index</td><td>x</td><td>y</td></tr>
<tr><td>1</td><td>0</td><td>0</td></tr>
<tr><td>2</td><td>0</td><td>1</td></tr>
<tr><td>3</td><td>1</td><td>1</td></tr>
<tr><td>4</td><td>1</td><td>0</td></tr>
</table></td>
<td><table>
<tr><td>panels</td><td>ns</td><td>3</td></tr>
<tr><td>index</td><td>qa</td><td>qb</td></tr>
<tr><td>1</td><td>1</td><td>2</td></tr>
<tr><td>2</td><td>2</td><td>3</td></tr>
<tr><td>3</td><td>3</td><td>4</td></tr>
</table></td></tr>
</table>

Table 4. Two simple boundaries to initiate the verification.

1. Verification of single-value inputs

Early on in the verification it makes senPanels to make some routine checks on the single-valued input parameters, before verifying the data in the tables. Whilst these tests seem trivial, it is useful to include them since a typical error is to omit the value of a parameter. For example for a closed boundary the number of nodes and panels must be at least three ($N_V \geq 3, N_S \geq 3$) and for an open boundary there must be at least two nodes and there must be at least one panel ($N_V \geq 2, N_S \geq 1$). There can be any number of domain points ($N_P \geq 0$).

As mentioned earlier, part of the verification is the determination of whether domain points are on or close to the approximate surface. Given that the coordinates of the nodes and points are real numbers, and may in practice be the result of measurement, a tolerance ε is introduced and it is presumed that if two points approximately within a distance of ε apart then the two points are effectively indistinguishable. Clearly ε must be positive and hence the initial check is that ε>0.

In the first set of tests are on the sheets *1 closed* and *1 open* in which the simple closed boundary and simple open boundary are tested. First they are tested (test P0a) with the original data and with ε=0.000001 in order to achieve a positive outcome. In the tests that follow on the main parameters are given unacceptable values in order to provoke an error message. In *Test 1 closed,* the altered parameters are E1a $N_V = 2$, E1b $N_S = 2$, E1c $N_P = -1$ and E1d ε=0. In *Test 1 open,* the altered parameters are E1a2 $N_V = 1$ and E1b2 $N_S = 0$. Except for test E1a on both sheets the process is terminated once the error has been found.

2. Initial tests on the definition of the boundary panels

In this subsection a set of elementary tests are applied to the definition of the panels. In the first test, E2a, it is checked that all panel indices are in the range 1…N_V.The test problem E2a that is designed to elicit this error is the simple closed boundary except that an extra undefined node '5' is introduced between nodes '4' and '1'. In the definition of the panels on a boundary, any node must be referenced at most once as the starting node and at most once as an end node. Test E2b checks this and the test case is the same as the simple closed boundary with the first panel repeated as panel five.

For closed boundaries every node that is referenced in the definition of the panels should appear once as a starting node and once as an ending node of a panel, as illustrated in Table 2. This is verified by test E2c and the test case E2c the last panel is not included in the simple closed boundary. Similarly, for an open boundary, only the starting node on the first panel on an open boundary and the ending node on the last panel are referenced once, with all other referenced nodes being referenced twice. This is verified by test E2d and the test case is that of the simple closed boundary that is declared as open.

3. Initial tests on nodes, panels and tolerance

In this section a set of simple tests are applied to geometric data. All of the test in this section are applicable to open and closed boundaries. Clearly any boundary, whether it is open or closed, should have a finite size. In the first test (Test E3a), the *size* of the boundary is determined. A trivial error would be for example to omit the data in table 1 and this check would be able to find that error. A measure of the size of the boundary is also a useful property with which to check other information communicated to the boundary element method. Let the *size* of the boundary be denoted by χ and defined as the greatest distance between any two nodes. Clearly the geometric tolerance must be less than the size of the boundary (ε< χ), and in practice it should be minute in comparison to the boundary size (ε≪χ).

The validation code makes the presumption that the condition ε< χ/100 must be met. In test case E3a, the original simple closed boundary is used, so that χ =$\sqrt{2}$, but the geometric tolerance is given a large value ε=1, returning the expected error message.

In test E3b the difference between each pair of nodes is determined and if the difference is effectively zero (that is less than ε) then the nodes are coincident and an error is flagged. The test problem for this is the simple closed boundary above with node 2 repeated.

In test E3c the maximum and minimum lengths of all the panels are calculated. If the largest panel is more than 10 times the size of the smallest panel then a warning message is output. In order to check this test boundary is the simple closed boundary, but with a node with coordinates (0.05,1) is introduced and the second element of unit length is split into two elements with lengths of 0.05 and 0.95. Hence in the test problem the largest panel is of unit length and the smallest panel is of length 0.05 and hence this evokes the warning message.

In test E3d a further check is made on the value of the geometric tolerance ε. The length of the smallest panel must be significant with respect to ε. Hence a further test on ε is carried out to check that the smallest panel is at least ten times the size ε. To evoke this error in test case E3e, the test boundary of the previous paragraph is used, but with ε=0.01. Hence the warning of the previous paragraph is initially evoked, followed by the error message. Note that a warning – rather than an error – message is output. For example in a boundary element mesh the panel sizes may be deliberately decreased in certain areas in order to improve the approximation of the boundary and the boundary functions there.

The General Boundary Definition and Positive Test Cases to support it.

Before we can verify that the boundary is correct, we must be clear about the generality that we are allowing in the boundary definition. For boundaries in two dimensions, it may be reasonably expected that the panels are defined (indexed) conPanelscutively along the boundary. However, this cannot usually be reproduced in three dimensions and hence this expectation is relaxed in the definition of a two dimensional boundary. In the tests so far, the way the boundary is connected has not been addressed. Also the orientation of the panels, which is important, has not been considered. In order to develop more thorough testing, a more general view of the expectations of the boundary is first introduced, and thn further tests are made on to check to see if the boundaries meet these expectations.

1. Separate boundaries

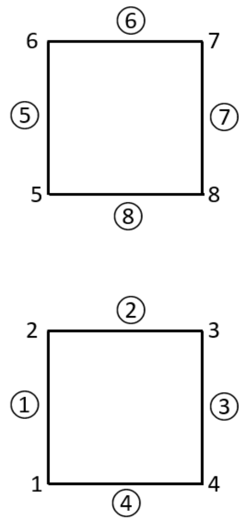
There may be any number of closed and open boundaries and it has been decreed that the closed boundaries are all defined in one list and all open boundaries are defined in a another list. In the following table three new test boundaries are introduced, each made up of two separate boundaries of the same type (ie open or closed). Test boundary 3 represents two closed boundaries as found in an exterior problem. Test boundary 4 is an example of two closed boundaries, in a typical interior problem; the second boundary effectively maps out a hole in the interior domain. Test boundary 5 consists of two open boundaries.

The orientation of the panels is important, and the verification of this will be considered later in this document. The adopted convention is that for exterior problems the domain is to the left when travelling from the first node to the second node (that together define a panel). For interior problems the domain must always be to the right. This should be clear in the closed boundaries defined so far in this document. An important outcome of this rule is that, for an interior problem, the panel nodes are listed clockwise on the outer boundary, but anticlockwise on the inner boundar(ies). And this latter point is illustrated in test boundary 5 in Table 5.

For the open boundaries, the ‘domain’ is located on both sides of the boundary. In this case it is necessary that the panels are all oriented in the same way. For example if in travelling from the first node to the second node the sections of the domain on the left and right must be the same. In doing this the user is efffectively defining an ‘upper’ and a ‘lower’ side to the boundary, which must be consistent with the definition of the boundary condition and in the interpretation of results. As further verification the three boundaries in Table 5 are implemented in test cases P2a, P2b and P2c, each returning a positive result.

Illustration of the boundary		Nodes	Panels

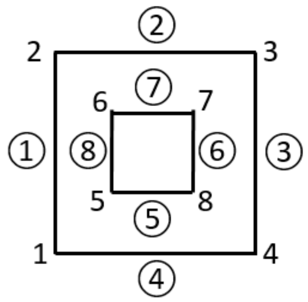
3. Two closed boundaries (exterior)



nodes	nNodes	8
index	x	y
1	0	0
2	0	1
3	1	1
4	1	0
5	3	0
6	3	1
7	4	1
8	4	0

panels	Ns	8
index	Qa	qb
1	1	2
2	2	3
3	3	4
4	4	1
5	5	6
6	6	7
7	7	8
8	8	5

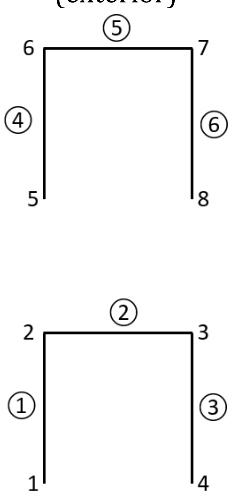
4. Two closed boundaries (interior)



nodes	nNodes	8
index	x	y
1	0	0
2	0	1
3	1	1
4	1	0
5	0.25	0.25
6	0.25	0.75
7	0.75	0.75
8	0.75	0.25

panels	Ns	8
index	Qa	qb
1	1	2
2	2	3
3	3	4
4	4	1
5	5	8
6	8	7
7	7	6
8	6	5

5. Two open boundaries (exterior)



nodes	nNodes	8
index	x	y
1	0	0
2	0	1
3	1	1
4	1	0
5	3	0
6	3	1
7	4	1
8	4	0

panels	ns	6
index	qa	qb
1	1	2
2	2	3
3	3	4
4	5	6
5	6	7
6	7	8

Table 5. Test boundaries to initiate the verification.

2. Organisation of the input boundary

An algorithm is implemented that passes from node to node along each boundary. Each boundary is indexed and each panel is assigned to the boundary index. When the boundary is complete, either by returning to the initial node for a closed boundary or coming to the end node for an open boundary, the boundary index is incremented and a new start node is found. When this algorithm completes, all the panels are assigned to a boundary index.

The nodes that are followed along the boundary are not necessarily followed in the order in the list of nodes or panels. Rather the method links from panel to panel so that the end node of the current panel is the same as the start node of the following panel. This relieves the requirement that panels or nodes need to be listed in any particular order.

Test cases which include three separate boundaries are included. Test P2e is an exterior problem with three separate square boundaries. Test P2e is an interior problem consisting of a square with two square holes within it. With the correct orientaion of the panels, as described earlier, both tests return positive results. These two tests are illustrated in Figures 4 and 5 below, in which now the boundaries are enumerated with respect to the ordering of the first panel on each boundary. These test problems will be revisited later in this document.

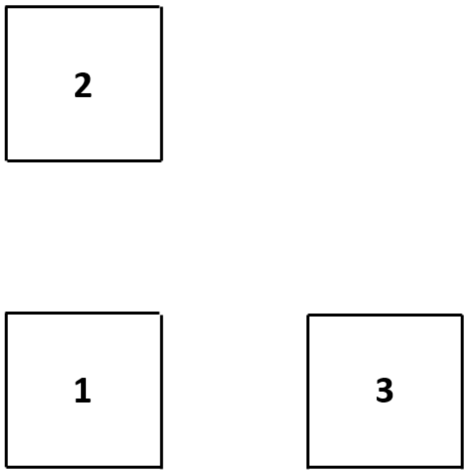


Figure 4. Exterior boundary for test P2d, with 12 panels in total and boundaries enumerated.

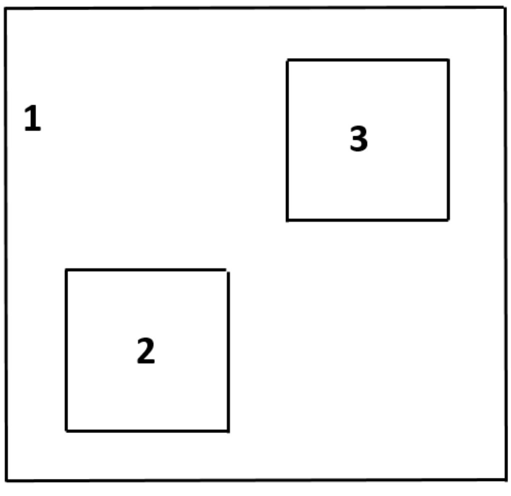


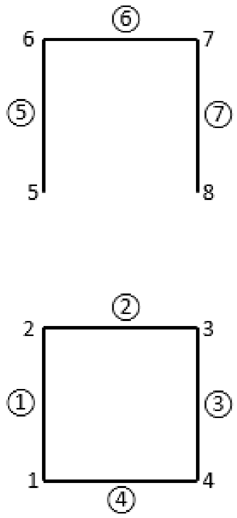
Figure 5. Interior boundary for test P2e, with 12 panels in total and boundaries enumerated.

Test cases are implemented in order to show that the order of the panels is irrelevant in the verification of the boundary. In test P3a, the boundary is the same as in test P2a 'Two closed boundaries (exterior) ', except that panels ③ and ⑥ are swapped and no error is reported. Similarly in Test P3b the boundary is the same as in test P2b 'Two closed boundaries (exterior) ', except that panels ③ and ⑧ are swapped. Finally, Test P3c is the same as Test P2c, except that panels ① and ⑤ are swapped.

Geometrical Verification Methods

Illustration of the boundary	Nodes	Panels

6. Closed and open boundaries (exterior)



nodes	nNodes	8
index	x	y
1	0	0
2	0	1
3	1	1
4	1	0
5	3	0
6	3	1
7	4	1
8	4	0

panels	nPanels	8
index	qa	qb
1	1	2
2	2	3
3	3	4
4	4	1
5	5	6
6	6	7
7	7	8

Table 6. Mixed test boundary.

The tests so far have considered the discrete data elements that are input to the the boundary element method. Once these tests have been passed the testing involves considering how the boundaries are defined. The first test on the geometry takes place while the boundary is being organised, as discussed in the previous section. As stated earlier, all boundaries input to the test must be closed or all must be open and one of the parameters of the test routine is the declaration that the boundaries are closed or open. So let us test the routine with a mixture of open and closed boundaries. The test boundary is illustrated in Table 6.

If the boundary in Table 6 is declared 'closed' then the error is picked up by E2c since in such a ase as this some of the nodes are referenced on only one side of the panel. The test case E2c2 consists of the test boundaries in case 6, and this evokes the E2c error. However, if the boundary in Table 6 is declared 'open' then the error is picked up by the method for organising the boundary described above. If the boundaries are all defined as 'open' then a starting node is searched for; if this causes a cycling around he nodes, as would be expected if the boundary was closed, then an error is reported. The test case for this is on sheet G1 and it consists of the boundary in Table 6 and declared ‘open’. This test case evokes the G1 error, which reports that not all the boundaries are open.

In the second test G2, for every node, the angle between the panels on either side is considered. If the angle is too sharp then a warning is output. The test case G2 a triangle with a sharp angle on the third node, generating the warning. The test case for G2 is illustrated in Table 7.

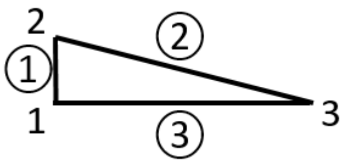
<table>
<tr><th>Illustration of the boundary</th><th>Nodes</th><th>Panels</th></tr>
<tr>
<td>7. Boundary with sharp angle

</td>
<td><table>
<tr><td>nodes</td><td>nNodes</td><td>3</td></tr>
<tr><td>index</td><td>x</td><td>y</td></tr>
<tr><td>1</td><td>0</td><td>0</td></tr>
<tr><td>2</td><td>0</td><td>0.2</td></tr>
<tr><td>3</td><td>1</td><td>0</td></tr>
</table></td>
<td><table>
<tr><td>panels</td><td>ns</td><td>3</td></tr>
<tr><td>index</td><td>qa</td><td>qb</td></tr>
<tr><td>1</td><td>1</td><td>2</td></tr>
<tr><td>2</td><td>2</td><td>3</td></tr>
<tr><td>3</td><td>3</td><td>1</td></tr>
</table></td>
</tr>
</table>

Table 7. Boundary with a sharp angle.

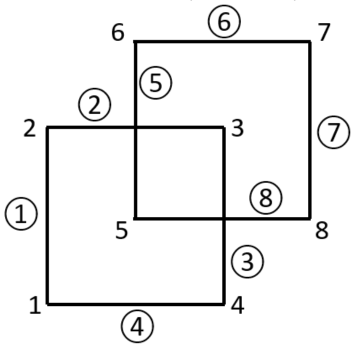
The panels must not overlap and the boundary must not be significantly ‘thinner’ than the panel size. The test G3 compares the distance between the central points of each pair of panels. I the central nodes are very close then an error is reported. It is possible that the central nodes are not as close but the points on the panels still run close together. Hence a second level of testing is implemented if the nodes are quite close in which points are chosen along the panels and tested to see if any pair of individual points are too close.

The test cases for this are illustrated in Table 8. Test case 8 consists of two squares that overlap. In test case 9 the panels ① and ⑤ and panels ② and ④ are too close with respect to their size. Test case 10 is an ‘open’ boundary with intersecting panels. Each of these cases generates the appropriate error message. The test cases G3a, G3b, and G3c are on the sheets with this name and each case generates a G3 error.

The final test based on geometry considers the distance between each domain point that is input to the method and the boundary panels. The reason for this is that the integrands relating the domain point to the boundary panel in the boundary element method are more quickly varying if the domain point is close to the boundary and – with a fixed quadrature rule – this could lead to a loss of accuracy. Hence the distance between each domain point and panel is checked in order to flag a warning message if the domain point is close to the boundary. The test case consists of 1. Simple Closed Boundary with the domain point at (0.05,0.75), and this generates the G4 warning.

Illustration of the boundary	Nodes	Panels

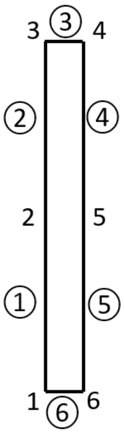
8. Two overlapping closed boundaries (exterior)



nodes	nNodes	8
index	x	y
1	0	0
2	0	1
3	1	1
4	1	0
5	0.5	0.5
6	0.5	1.5
7	1.5	1.5
8	1.5	0.5

panels	nPanels	8
index	qa	qb
1	1	2
2	2	3
3	3	4
4	4	1
5	5	6
6	6	7
7	7	8
8	8	5

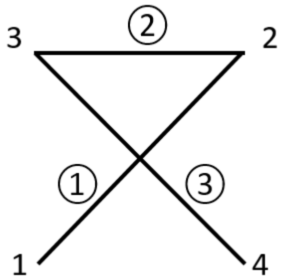
9. Thin closed boundaries



nodes	nNodes	6
index	x	y
1	0	0
2	0	1
3	0	2
4	0.2	2
5	0.2	1
6	0.2	0

panels	nPanels	6
index	qa	qb
1	1	2
2	2	3
3	3	4
4	4	5
5	5	6
6	6	1

10. Intersecting open boundary



nodes	nNodes	4
index	x	y
1	0	0
2	1	1
3	0	1
4	1	0

panels	nPanels	3
index	qa	qb
1	1	2
2	2	3
3	3	4

Table 7. Panels too close; intersecting or 'thin' boundaries.

Testing based on results from Potential Theory

By considering as many unacceptable boundary configurations as possible, the typical arrangements of a well-formed boundary become clearer. The boundary for an exterior problem consists of one or more separate boundaries, with the panels that describe the boundaries oriented clockwise around the boundary. In moving from the first node to the second node of any panel in an exterior problem, the domain should be on the left. If there are any open boundaries then they are within the exterior domain (not wholly or partially inside any of the closed boundaries).

For an interior problem, the domain is demarcated by an outer closed boundary in which the panels are oriented clockwise around the boundary. The domain may then be demarcated further by any number of inner closed boundaries, which are defined with the panels oriented in the anticlockwise direction. In moving from the first node to the second node of any panel in an exterior problem, the domain should be on the right. If there are any open boundaries then they are within the interior domain (not wholly or partially in the exterior or within any of the closed internal boundaries).

An open boundary has two sides, an 'upper' side and a 'lower' side. In a senPanels it does not matter which side is assigned as 'upper' or 'lower', as long as this is consistent for every panel on the open boundary and with the boundary condition. In moving from the first node to the second node of any panel in an exterior problem, the 'upper' side should be on the left and the 'lower' side on the right.

The verification so far has not considered the points in the domain at which the solution is sought in the boundary element method. Clearly, the domain points must be in the exterior domain in the case of an exterior problem and in the interior domain in the case of an interior problem.

In this section we use some results from potential theory in order to further verify the boundary. These results are also the outcome of solving Laplace's equation and since this is central to the boundary element method then we can use the integrations that are normally used in the boundary element method in order to test the properties. Using this technique on top of the other verification methods and boundary analysis considered earlier in this document completes our comprehensive verification of the geometrical input into the boundary element method.

Useful result from potential theory

The application of Green's second theorem to the interior Laplace equation[9]

$$\nabla^2 \varphi(\mathbf{p}) = 0 \quad (\mathbf{p} \in D)\,, \tag{1}$$

where D is a closed domain, gives the following equations

$$\{M\varphi\}_S(\boldsymbol{p}) + \varphi(\boldsymbol{p}) = \{L\,v\}_S(\boldsymbol{p}) \quad (\boldsymbol{p} \in D)\,,$$

$$\{M\varphi\}_S(\boldsymbol{p}) + \frac{1}{2}\varphi(\boldsymbol{p}) = \{L\,v\}_S(\boldsymbol{p}) \quad (\boldsymbol{p} \in S)\,, \tag{2}$$

$$\{M\varphi\}_S(\boldsymbol{p}) = \{L\,v\}_S(\boldsymbol{p}) \quad (\boldsymbol{p} \in E)\,,$$

where $\varphi(\boldsymbol{p})$ is the potential, $v(q) = \frac{\partial\varphi}{\partial n_q}$ and L and M are two of the Laplace integral operators[10]. Note that the normal to the boundary are taken to be in the outward direction.

[9] Laplace's Equation
[10] The Laplace Integral Operators

Let us consider the special case in which $\varphi(\boldsymbol{p}) = 1$ throughout the domain and on the boundary. It immediately follows that $v(q) = \frac{\partial \varphi}{\partial n_q} = 0$; the derivative of the constant function φ is always zero. In this special case the equations above simplify as follows:

$$\{M1\}_S(\boldsymbol{p}) + 1 = 0 \quad (\boldsymbol{p} \in D),$$

$$\{M1\}_S(\boldsymbol{p}) + \frac{1}{2} = 0 \quad (\boldsymbol{p} \in S), \tag{3}$$

$$\{M1\}_S(\boldsymbol{p}) = 0 \quad (\boldsymbol{p} \in E).$$

where '1' represents the function that equals one for all points on the closed boundary *S. D* is the interior to *S* and *E* is the region exterior to *S*. The implications of this are that the application of the *M* operator to the unit function on any boundary gives the result of -1 for points interior to that boundary, gives the result of -1/2 for points on the boundary (note this only true for a smooth part of the boundary) and gives the result of zero for points exterior to the boundary.

Although there isn't a similar sort of formula to apply to an open boundary, we can apply the complement of this logic. That is if the application of the *M* operator to the unit function must give a result other than these for open boundaries. The integrations will usually be approximated and hence in order to make use of these rules the integrals will need to be computed to a reasonable accuracy. In this work exact integration is applied to the evaluation of integrals when the observation point lies on the panel and 8-point Gaussian quadrature is applied otherwise, through the L2LC module[11].

For example for the *Simple Closed Boundary,* for the collocation point on the panel (0,0.5), the numerical evaluation of the integral over the boundary is -0.499999990577. For a typical interior point (0.25,0.75) the evaluated integral over the boundary is -0.999963286500 and for a typical exterior point (0.5, 1.5) it is -0.000000304642 . From this it can be observed that the results in (3) can be reproduced accurately for a fairly standard quadrature rule and a coarse mesh. The integral for the interior point is somewhat less accurate, and this is attributed to the closeness of the point to the boundary, which has the effect of making the integrands of the operators to be more quickly varying and hence producing a greater numerical error.

For the *Simple Open Boundary,* for the collocation point on the panel (0,0.5), the numerical evaluation of the integral over the boundary is -0.323791804103. For the point (0.25,0.75) the evaluated integral over the boundary is -0.823755095271 and for a typical exterior point (0.5, 1.5) it is 0.102416077707. It is noted that the removal of one panel significantly adjusts the results obtained from the closed boundary.

<u>Testing the simple boundaries</u>

The intial testing is on the boundary itself. In principal then for any point $\boldsymbol{p}$ on the boundary $\{M1\}_S(\boldsymbol{p}) = -0.5$, if the boundary is closed, and $\{M1\}_S(\boldsymbol{p}) \neq -0.5$, if the boundary is open. The earlier tests have been able to find the error of closed boundaries being declared 'open' and open boundaries being declared 'closed' and hence we do not need to use the results here to trap such an error. However, this result can initially help to find errors in the way the panels are oriented.

[11] L2LC.bas

Test case T4a is the *Simple Closed Boundary*, but with the panels defined in the anticlockwise direction. Test case T4b uses the Two closed boundaries (exterior) (3), except that the orientation of the panels on one of the squares is reversed. In test case T4c the test case (4) Two closed boundaries (interior) is used except that the orientation of the panels on the inner square is reversed. All of these tests generate a T4 error.

The final test on the simple boundaries considers the possibility of a boundary being declared open, but being found to be closed. This error would normally be captured with E2d. However, there is a scenario in which the closure of the boundary is ambiguous. This error is denoted T5 and the test case for it consists of the *Simple Closed Boundary,* except an extra node (node 5) is introduced that is very close to node 1; the point (0.00001,0). There are still four panels, except the final panel links between nodes 4 and 5 instead of nodes 4 and 1.

Multiple Boundaries

The testing decribed above for one boundary can initially be readily extended to multiple boundaries. Hence the testing finds the total boundary integral for one observation point on each separate boundary. The observation point on the boundary can be any point, but it is chosen to be the point ($\boldsymbol{p}^*$) in the centre of the first panel that is listed in *panels.* By finding the integral $\{ M1\}_S$ ($\boldsymbol{p}^*$) for each point $\boldsymbol{p}^*$, the result should always equal -0.5 for closed boundaries and not equal =0.5 for open ones..

In this document it has been shown how methods of determining the organisation of the boundaries have been carried out the alongside the testing. The method described in the previous section enables us to determine the separate boundaries that are input all together in the tables that are input into the boundary element method. The method described in this section, based on the methods of potential theory, can now be applied to the individual boundaries in order to ensure that all are correctly defined. However, precise results for the integrals are available only for closed boundaries and so we can only proceed with this testing with closd boundaries.

For closed boundaries a useful pattern emerges that may be exploited in order to further verify multiple boundaries. For example for the multiple boundaries in figure 4, dividing the boundary into three boundaries and finding the integral over each boundary with respect to a point on each boundary gives the following table. Hence, the general result, which also follows from the earlier theory, is that for closed exterior problems we expect the integral over the boundary that the point is on to be -0.5 and the integral over any boundary that the point is not on to be zero.

	boundary 1	boundary 2	boundary 3
point on boundary 1	-0.5	0	0
point on boundary 2	0	-0.5	0
point on boundary 3	0	0	-0.5

Table 8. Integral over each separate boundary with respect to an observation point on each boundary for the boundaries in figure 4.

For closed interior problems, a slightly different pattern emerges. For example for the problem in figure 5, the integrals are shown in the following table. Firstly, when the point lies on the individual boundary the integral is -0.5 in the case of the outer boundary (boundary 1) and +0.5 in the case of the inner boundaries. The reason for this is that the outer boundary is oriented clockwise and he inner boundaries are oriented anticlockwise and hence the sign of the integral is reversed.

	boundary 1	boundary 2	boundary 3
point on boundary 1	-0.5	0	0
point on boundary 2	-1.0	0.5	0
point on boundary 3	-1.0	0	0.5

Table 9. Integral over each separate boundary with respect to an observation point on each boundary for the boundaries in figure 5.

The other difference with the exterior boundaries is that the integral over the external boundary with respect to a point on the integral boundary is -1.

<u>Testing domain points</u>

In the boundary element method the solution is normally sought at points in the domain.The results from potential theory (equation 3) informs us that if a point is exterior to a closed boundary S then $\{M1\}_S\,(\boldsymbol{p}) = -1$ and if it is exterior then $\{M1\}_S\,(\boldsymbol{p}) = 0$. In test case P6a the Simple Closed Boundary as an interior problem with the domain point (0.25, 0.25) and no error is reported. However, in test T6a, the same boundary is used and the problem is again declared as being interior, but with the domain point of (0.5, 1.5), an error message is generated, since the point is exterior to the boundary.

Tests P7a and T7a carry out the opposite test scenario. In this the *Simple Closed Boundary* is again used and the problem is declared as an exterior problem. In test case P7a the domain point of (0.5, 1.5) does not generate a 'T7' error but in test case T7a the domain point (0.25, 0.75) does.

The methods that result from the application of potential theory, can equally be used in more complex boundary arrangements. In tests P6b and T6b test boundary *4. Two closed boundaries (interior)* is used. In test P6b the point (0.5,0.125) lies within the interior domain and no error is reported. However, in test T6b, the point (0.5,0.5) does not lie in the domain and an error is reported.

Finally a more complex interior boundary is tested. In P7b and T7b test boundary *3. Two closed boundaries (exterior)* is used and the problem is declared exterior. For P7b there are three test points (0.5, -0.5), (0.5, 1.5) and (0.5, 3.5), all of which are exterior to the two boundaries, with the second domain point lying between the two squares, and no error is reported. However, in T7b, the point (0.5,0.5) lies within the lower square and an error is reported.

Summary of Errors and Warning messages

Error Code	**Message**
E1a	for closed boundaries, we must have nNodes>=3 (nNodes=number of nodes)
E1a2	we must have nNodes>=2 (nNodes=number of nodes)
E1b	for closed boundaries, we must have nPanels>=3 (nPanels=number of panels)
E1b2	we must have nPanels>=1 (nPanels=number of panels)
E1c	must have nPoints>=0 (nPoints=number of domain points)
E1d	tolGeom must be positive
E2a	panelVertices : some vertex indices are out of range
E2b	some indices are repeated as the start or end nodal indices of panels
E2c	for a closed boundary all nodes that are referenced must be referenced on either side of two panels
E2d	each open boundary must have a start node and an end node
E3a	tolGeom is too large
E3b	boundary nodes listed in nodes are not all distinct
E3c	panels are of disproportionate size
E3d	panel size must be significant in comparison with tolGeom: tolGeom is too large?
G1	not all boundaries are open
G2	*boundary has sharp angles*
G3	panels are too close or intersect

G4	*some domain points are close to the boundary*
T4	boundary is said to be closed but it could be open or the panels may be incorrectly oriented
T5	boundary is said to be open but it seems to be closed
T6	problem is 'interior' but some domain points are exterior
T7	problem is 'exterior' but some domain points are interior