A Finite Difference Method for Cylindrically Symmetric Electrostatics having Curvilinear Boundaries

David Edwards, Jr., *Member, IAENG*

Abstract—Difficulties of the finite difference method occur when a boundary does not lie on the mesh points in the overlaid meshpoint array but passes between them. To overcome this limitation and at the same time to effect high precision solutions to the curvilinear problem a solution is described which extends the internal space of the geometry to the other side of the enclosing boundary by analytic continuation. The boundary potential itself is incorporated into the values of mesh points near it by interpolation from the boundary. Using this technique precisions of the order of ~10^{-13} have been obtained for the concentric sphere geometry. Thus a fundamental limitation of the finite difference method has been removed.

***Index Terms*—Finite Difference Method, curvilinear boundaries, analytic continuation, high precision.**

I. Introduction

THE finite difference method (FDM) is a simple computational tool for finding the solution to boundary value problems by an iterative method [1]. The solution is a function, having fixed values on the boundary, satisfying a differential equation at all interior points. The method involves overlaying a set of equally spaced meshpoints over the geometry and then relaxing the mesh

For problems having the boundaries lying on meshpoints, a multi region process has been previously described in a series of papers [2]-[8] which has demonstrated the high precision capabilities of the FDM process when both multi regions and high order algorithms are used.

A serious limitation affecting the precisions obtained by the current FDM method occurs when the boundary does not lie on the meshpoints themselves but passes between meshpoints as will occur for curvilinear boundaries. This report is directed to the solution of this problem.

As the solution is reasonably general the discussion will be restricted to cylindrical symmetric electrostatics in which both Laplace's equation and cylindrical symmetry are assumed. The solution of this problem will have immediate applicability to field of electron and ion optics.

A. Background, the FDM process.

In order to both standardize our notation and emphasize certain features of FDM, a quick overview of the FDM process is useful [2]. Consider figure 1 in which an array of mesh points is overlaid on a geometry represented by three connected line segments all lying on either rows or columns of meshpoints. This geometry represents a closed cylinder in three dimensions. It is noted that potentials of meshpoints falling on the boundary are constant through the relaxation process. Points strictly within the geometry will be designated as ingeometry points

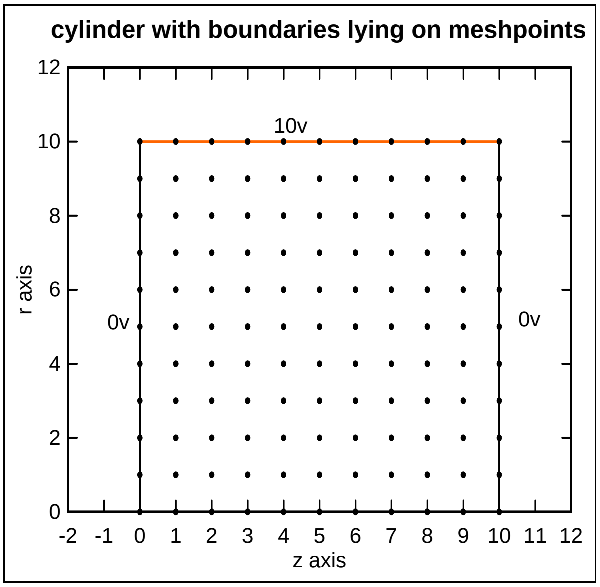

Fig. 1. A cylinder is shown with its boundaries lying on meshpoints and the potentials on the various segments are indicated.

In order to relax such a mesh, the points of the mesh are stepped through in a sequential manner. At each point the value of the potential is evaluated by means an algorithm using the values of the surrounding mesh points as input. This process is continued until all values have been determined during this iteration. Iterations are continued until a suitable end criterion is met for the potentials within the net; i.e. the values at all meshpoints have stopped changing. When the end criterion is met, the mesh is said to be relaxed and the potentials within the mesh determined.

B. Boundaries lying between meshpoints, low order estimates

If figure 1 is slightly modified (not shown) by having the outer boundary pass between meshpoints rather than lie on meshpoints, a problem for the relaxation process is immediately created, i.e. no mesh point has the boundary potential. If we consider the boundary points as outermost points of the mesh, the values of these points must somehow be estimated in order that the above process is defined. A zeroth order estimate would be to set the value of these outermost points to the potential of the nearest boundary

David Edwards Jr. is a member of the IJL research center, Newark, Vt. 05871. e-mail: dej@kingcon.com (802)274 5845

and would allow the mesh to be relaxed as described above. This estimate has the benefit of simplicity in implementation while providing a process solution converging to the problem solution in the limit of high mesh densities. The downside is of course its inherent lack of precision, since the geometry itself has been modified and no longer corresponds to the model geometry.

A slightly improved estimate [9] of the potential of the outermost points is to place non integral mesh points on the boundary. Using these points algorithms for points near the boundary can be found but include two additional parameters describing the vertical and horizontal distances from the boundary. These additional parameters make the algorithms for near boundary points considerably more complex and although it can provide a better estimate for the potential near a boundary than the zeroth order estimate, this complexity has and likely will preclude its application to any but the lowest order algorithms.

C. The algorithm development process

The algorithm development process has been described previously [2]-[8] and only a brief summary of its essential features is presented here in order to familiar the reader with the origin of and the need for a required set of mesh points as input to any algorithm used in the relaxation process. As a result of this description the class of possible algorithms is systematized and its precision ranked by an algorithm parameter called “order”.

About any mesh point in the mesh overlay of the geometry there is assumed to be a power series expansion of the potential v(r, z) as a function of the relative coordinates r, z with respect to the center of the particular mesh point. (In this notation the potential at the position of the mesh point itself is v (0, 0).)

The power series expansion of v (r, z) is written:

(1) v(r,z)=c0+c1*z+c2*r+c3*z^2+…+c64*z*r^8
+c65*r^8 + … + *O* (j)

where *O* (j) (read order j) means terms of order $r^k z^l$ are neglected when k+l> j. j is called the order of the particular class of algorithms generated by this power series. For example for an order 8 algorithm, there are 45 cj’s in the above expansion. Requiring that v(r, z) satisfy Laplace’s equation in a neighborhood of the point produces *one equation* involving the coefficients cj’s and powers of r and z. Further requiring that this equation be true at any point in the neighborhood of the central mesh point implies that the coefficient of terms $r^k z^l$ must be zero and results in a set of 28 equations which are linear in cj in which only the coefficients cj appear (along with any parameters from Laplace’s equation). Thus an additional 17(45-28) additional equations are required for a solution to the entire set of cjs.

In order to find these additional equations it is noticed that if equation (1) is evaluated at a neighbor bj of the central mesh point its value-v (rj, zj)- may be found using (1), whose value is the value of the meshpoint bj and assumed known. In this way by forming an equation from each of 17 neighboring meshpoints, the 17 additional equations may be found and the complete set of 45 equations determined. As this set is linear all cjs can be determined by using the techniques of linear algebra. (It should be noted that the set of selected neighboring meshpoints must provide a consistent set of linear equations which is established during the solution of the equation set. Further the resulting algorithm must give a stable solution when used in the relaxation process as will be discussed below. This set of selected meshpoints is in fact highly degenerate since there are many such sets that will satisfy the above requirements.)

One such set of selected meshpoints used in the 8th order algorithm is shown in figure 2. Seen is that meshpoints in the second surrounding ring of the central meshpoint are used in the algorithm and must be available, the implication of this observation will be discussed in below. As the actual solution for any cj involves several hundred terms, the visualization of the solution itself is not instructive and is not reported.

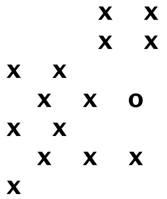

Fig. 2. The set of 17 mesh points around the central mesh point **o** for a general 8th order mesh point algorithm.

As the solution ck depends on both the selection of meshpoints {bj} and “a” the distance of the meshpoint from the axis (a parameter from Laplace’s equation). It may be written:

(2) ck = ck_coeff_b0(a)*b0+ ck_coeff_b1(a)*b1+ ….

where coeff_bj(a) is a truncated power series in a, the highest power of a which depends upon the order of the algorithm.

There are two situations that will be encountered in the following: the first is one in which the potential at a meshpoint itself is desired. In this situation only c0 need be determined since c0 = v (0, 0). The second is one for which the potential is desired at a point in the vicinity of a boundary for which all cj’s must be found. (A benefit to evaluating the complete set of cjs is that the solution set may then be shown to satisfy the equation set.)

II. The solution for external points

When the geometry of figure 1 is relaxed, points one unit from the boundary need to be relaxed using a general mesh point algorithm. Thus using the 8th order algorithm of figure 2 on a point one unit below the upper line segment, the two upper points of figure 2 are required but are not available. If the upper boundary were in fact to pass between the meshpoints, then the upper 6 points of figure 2 would not be available. The solution to this dilemma is to place meshpoints on the other side of the boundary insuring that the points one unit from a boundary will have the necessary potentials available.

Laplace’s boundary value problem is known to have an analytic solution interior to the geometry and that an analytic function may be analytically continued across a boundary by requiring that the function and all of its derivatives be continuous at the boundary [10]. In this way the space of meshpoints is extended to the other side of the boundary. It is noted that in this extended space the boundary itself is assumed to be continuous with continuous derivatives and that Laplace’s equation is also applicable to any point in the continued space. The analytically continued meshpoints are designated as external points and since the differential equation itself is applicable in this region, the algorithms for these points may be found as described above with the exception of points closest to the boundary. These points are interpolated from the boundary itself by using equation (1). In this manner the potential of the boundary is incorporated into net by folding it into the potentials of the points near the boundary.

The set of external points are explicitly constructed from the ingeometry points near the boundary by requiring every such point have a complete set of four surrounding rings available. This will ensuring that an algorithm applied to any such point will have the necessary neighboring potentials available.

In order to be able to create separate algorithms for select types of external points, the external points themselves are classified in the following manner: *Near* points are defined as points of the extended space (external pts + ingeometry pts) within a distance delta of the boundary (typically taken to be ½). *Middle* points are those external points that have near points as neighbors, *far* having middle points as neighbors, etc. This will allow algorithms at an external point to be created based on the point’s near_far classification and hence on the near_far types in its immediate neighborhood.

Figure 3 shows a geometry consisting of 3 line segments illustrating the external points and their classifications.

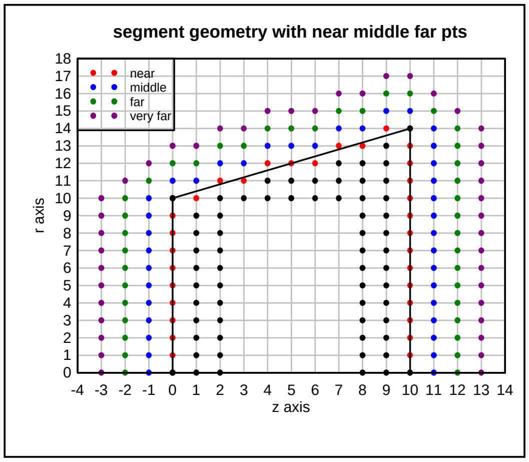

Fig. 3 A geometry illustrating the definitions of the near, middle, far classifications.

In addition each near_far classification is further divided into subtypes depending on the configuration of its surrounding neighbors that have a similar or lower near far classification as the point itself. In this way a point being in a similar neighborhood as other points with the same sub classification will use the same algorithm and will allow algorithms to be generated depending upon the meshpoint’s local neighborhood.

III. The Algorithms for External Points

A. *Algorithms For Middle, Far, …, Pts*

The potential at a middle, far and veryfar pt is found from the expansion coefficient c0, described in section 1.3 and depends only upon the potential at its neighboring meshpoints and the distance “a” of the meshpoint from the axis, It turns out that the stability of an algorithm is considerably less sensitive to the selection of its neighboring mesh points if feedback from points further from the boundary is reduced or eliminated. This can be done by considering points in the selection having a nearfartype less than or equal to that of the meshpoint itself. Thus for example the selection of neighboring of points for a middle point should not include far, or veryfar points. This rule has markedly simplified the search for stable algorithms.

Using (2) expression for c0 may be rewritten:

(3) c0 = Σ_j (c0_coeff_bj(a)* bj)

where bj is the potential of the jth neighbor.

It is noted that for any given meshpoint ck_coeff_bj(a) is a calculated numerical constant dependent on “a” and is constant throughout the relaxation process while bj is the potential of the jth neighbor which changes during the relaxation process. Use of this decomposition considerably facilitates the relaxation process time since the coefficient of each bj, namely, c0_coeff_bj(a), need be calculated only once.

B. *Near Point Algorithms*

As mentioned previously the algorithms for near points are fundamentally different than the other external points in that they involve interpolating the potential from the boundary to the point itself. To do this the difference in the potentials between the meshpoint and the boundary is found and simply added to the boundary potential. The process is given below for a mesh point whose relative distance to the boundary is rb, zb.

v (r, z) from equation (1) can be written:

v (r, z) = c0 + Σ_k ck*fk(r, z) , k = 1 to kmax

where fk(r, z) can be found from the kth term in the following sequence:

f1(r,z) = z, f2(r,z) = r, f3(r,z) = z^2, f4(r,z) = z*r,
f5(r,z) = r^2, ….

and kmax is the index of the last term in the sequence $r^j z^l$, l+j = algorithm order.

After evaluating v(r, z) at a point on the boundary (v (rb, zb) = vb) the potential at the near meshpoint v(0, 0) can be written:

v(0,0) = vb - Σ_k ck*fk(rb, zb) sum k = 1 to kmax, kmax

Inserting (3) for ck it is found after rearranging the summations that:

v(0,0) = vb – Σ_j (Σ_k { ck_coeff_bj(a) *fk(rb, zb)} * value_bj)

Defining:

coeff_bj (a)= Σ_k { ck_coeff_bj(a) *fk(rb, zb) }

which is independent of k, depending only on “a”.

Finally:

(4) v(0,0) = vb – Σ_j coeff_bj (a)*bj.

The distinct advantage of this formulation is that the calculation involves only one coefficient, coeff_bj (a), which as described above may be calculated for the meshpoint once prior to the start of the iteration process and greatly reduces the time required to relax the net. In fact during the relaxation itself there is no time penalty for the interpolation required by the near points as compared with the calculation for c0 for any other point in the net.

IV. Stability tests

Order 2, 4, 6, and 8 algorithms were created for the various classifications of meshpoints described above and subsequently used in the relaxation of various test meshes. Although the algorithms themselves could be readily found, when applied to certain geometries stability problems were frequently encountered. The general feature of an instability was an observation of unbounded growth of the mesh values during successive iterations of process. In view of this an algorithm is considered stable if the end criterion is met for the relaxation process using this algorithm for all geometries in the collection of test meshes.

The collection of meshes created for stability tests consisted of two types. The first was one constructed of line segments while the second constructed using concentric spheres. These test nets provided a variety of different neighboring configurations for each point classification.

A. Linear segmented geometries

Linear segmented geometries were constructed with one segment being above, below, to the right of, and to the left of the ingeometry points. The other segments were either horizontal or vertical. This type of construction was made so that within any particular geometry only one type of meshpoint classification would be tested. The potential on all segments was set at 10 so that the after relaxed potential at any point in the net would also be 10. This selection of the boundary potentials provided a zeroth order check on the algorithm itself and on the availability of its required neighbors.

Figure 3 gives an example of a scaled version of one of geometries for having external points above ingeometry points. For each non horizontal or vertical segment the segment started and ended on meshpoints and its angle was varied in 1 degree increments. A spawned set of geometries with the segment at the same angle was formed by slightly incrementing the position of the starting and ending points. Further sets of geometries were formed by subdividing the angle itself by fractions of a degree. In each of these geometries a given type of meshpoint would have a different surrounding neighborhood and hence be tested under a variation of the configuration of the meshpoints in its neighborhood. This construction generated over 3000 distinct geometries and was the stability test set for line segmented geometries.

Stable algorithm sets were found for order 2, 4, 6, and 8. It should be noted that while for orders 2 and 4 the stable algorithms were easily determined, finding stable algorithms for the higher order algorithms became progressively more difficult. The procedure was one of trial and error and involves selecting a set of neighboring meshpoints, finding the algorithm, and determining its stability. If unstable another set of mesh points was selected and the process continued until a stable algorithm was found. As mentioned above the search was simplified by minimizing the feedback from meshpoints with a higher nearfartype when possible for all except near types for which a reasonably symmetric set of neighboring points could be used. It should be noted that even for the high order algorithms the stable algorithm was in fact highly considerably degenerate meaning an algorithm created using many different neighbor sets would also be stable.

It is clear that the connection point between any two segments shown in figure 3 is a singular point in the potential space. This is true as the potential is not differentiable at this point. The best one can do for points in the vicinity of the singular points is to estimate the potential. This is done here using a low order algorithm, a procedure introduced and successfully applied previously [8].

B. Geometries With Curved Boundaries

To simulate a geometry with general curved boundaries, concentric circular arcs were selected (representing concentric spherical shells in 3 dimensions) an example of which is shown in figure 4.

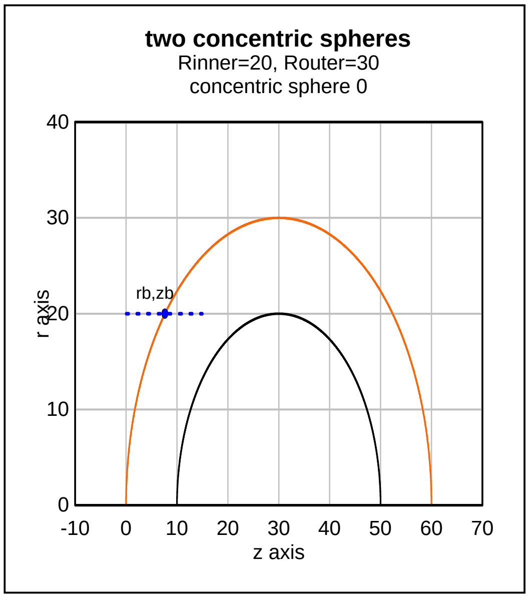

Fig. 4. A geometry consisting of 2 concentric spheres along with a horizontal line at r=20 intersecting the outer boundary at rb, zb.

The collection of test sets for concentric spheres consisted of several hundred geometries of the configuration type shown in figure 4. Tests were done over this set using the stable algorithms found for the segmented geometries and, with few exceptions, were found to be stable for the concentric sphere geometries. For these exceptions, slight modifications to the algorithms were made and the modified algorithms retested for stability.

In this manner stable algorithms were found orders 2 to 8 for both the line segmented and concentric sphere test sets. Being stable is of course a necessary condition for a useful solution, but gives little indication as to the precision of the process itself. Precision tests will be discussed in the next section.

V. Precision tests

Three types of precision tests have been made. The first, a zeroth order test, already mentioned and completed during the stability trials, was to set all boundary potentials to the same value and to verify that all relaxed potentials attained that value. The second was to estimate the potential on the boundary from the potentials in the relaxed mesh at meshpoints near the boundary. The third and the one most indicative of the actual precision was to find the absolute error for the relaxed mesh using the concentric sphere geometry since the errors can be determined for this geometry as theoretical values for the potential at all points are known.

A. Low Order Precision Tests -- Boundary Position Estimation From The Relaxed Mesh

This first low precision test is to see whether an estimate for the boundary position from the relaxed mesh potential data is reasonably close to the actual position of the boundary in the model geometry. As an example a simple segmented geometry similar to that shown in figure 1 is created in which all segments except the left fall on meshpoints. The left vertical boundary being at z = .6 falls in between the overlaid mesh point array and is shown in figure 5.

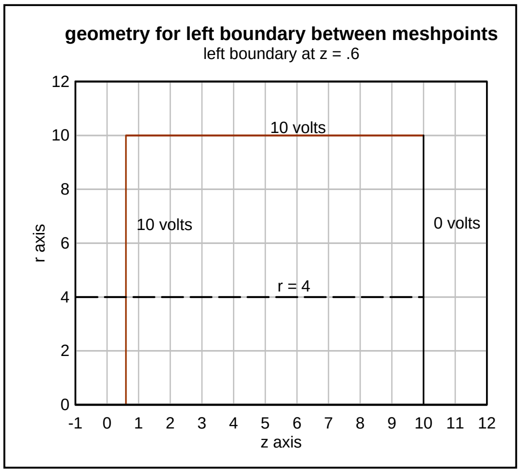

Fig. 5 The geometry used for testing the position of the left boundary from a fit to the potentials along a horizontal line at r=4. The left vertical boundary falls in between meshpoints.

In figure 6 the potential values along the line r=4 are shown near the left boundary along with a fourth order fit to the data points. From the location of the point z(r=4, v=10) on the fitted curve the z coordinate of the left boundary line can be inferred. It was found to be .598 which is within .002 of the position of the actual boundary, .600. The residual error in locating the position of the boundary from the relaxed data is likely due to two causes; first, the coarseness of the mesh overlay and second, the presence of singular points at the upper two corners of the geometry. Within these limitations however, the above test shows that relaxed potential distribution reasonably infers the position of the model boundary.

Also seen in figure 6 is the linear nature of the solution near the boundary which results from the requirement that the analytically continued solution have continuous derivatives at the boundary.

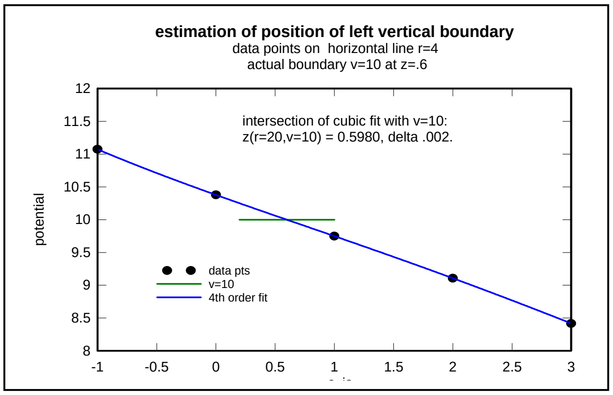

Fig. 6 A plot of the potential at meshpoints near the left vertical boundary along the line r=4.(see Fig. 5). For a discussion see text.

A similar test was made from the relaxed solution of the concentric sphere geometry of figure 4. The potential at meshpoints along the line r = 20 and near the upper sphere (the short line shown in figure 4) is plotted in figure 7 together with a 5th order fit to the data and as described above an estimate of the position of the boundary can be inferred from the location of the point zb(r=20,v=10) on the fitted curve.

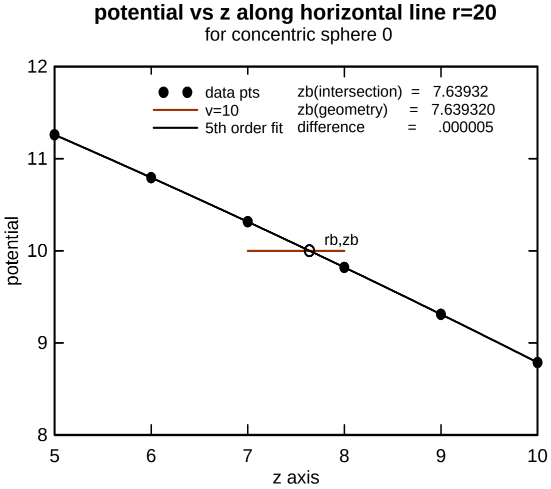

Fig. 7: The solid points are a plot of the after relax potentials at meshpoints along a horizontal line in sphere 0 at r = 20 and near z=7 (see figure 4). A 5th order fit to the data is given by the solid curve.

From this figure it is seen that zb (rb=20, v=10) = 7.63932 whereas the theoretical value of the intersection of the line r=20 with the outer sphere is 7.639320. Thus the position of the actual boundary is again close to that inferred from the after relax potentials indicating that the inferred boundary is in approximately the right place in the relaxed net.

That the estimate in the latter example the boundary position is much closer to that of the geometric model than in the former is likely due to the absence of singular points in concentric sphere geometries.

B. *Higher Order Precision Tests Using Spheres*

In order to do more meaningful precision tests the error in the potential at a selection of points within a soluble geometry must be found. The fact that the segmented geometries both contain singular points and in general do not have soluble solutions makes them not useful for these tests. Suitable however are the concentric sphere geometries as they both have a known theoretical solution and contain no singular points. A collection of concentric sphere geometries were made consisting of the geometry of figure 4 the remaining elements of the collection scaled from this geometry using a scale factor of 2. Thus con_sphere 0 has Rin = 20, Rout = 30, con_sphere 1 (40, 60), etc. Each geometry has been relaxed with the order 8 algorithm and the errors measured for points within .5 of the median plane ((Rin +Rout)/2). From the point error data an average error may easily be found for a particular concentric and is plotted in figure 7 vs the density points at measured that point, the density of the points given by 1/(Dout*Rout), where Dout = 2*Rout.

Seen is the super linear increase of precision with density; namely for the density increase of one order of magnitude the error decreases by over 3 orders of magnitude.

Although figure 7 was taken from data for the measurement sphere in the median plane of the geometry, varying the radius of this plane from the inner to outer shell had little effect on the error as shown in figure 8 evaluated for the geometry with the highest density seen in figure 7. (Similar results pertained to the other test spheres as well, i.e. the error at the median sphere represented to within a factor of 2 or 3 the errors over the entire space, with the error distribution becoming flatter for higher densities.)

As a line drawn from any point to the center of the concentric sphere makes an angle alpha with respect to the z axis the distribution of errors vs angle along points in the test sphere may be made and is shown in figure 9, again for the highest density geometry of figure 7.

Seen is that an average error is found to be ~4.4*10^{-14} and is essentially independent of the point’s position on the test sphere.

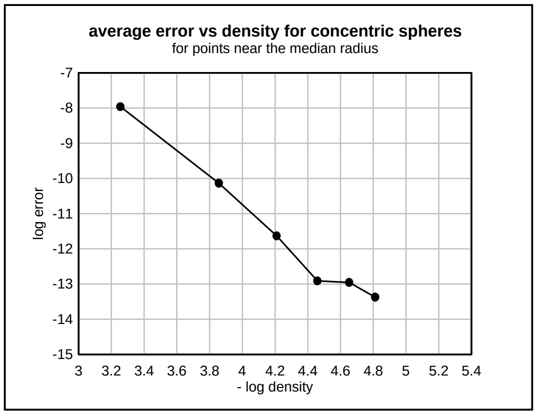

Fig. 7 The average error vs density for points near the median radius of the concentric spheres in the test set.

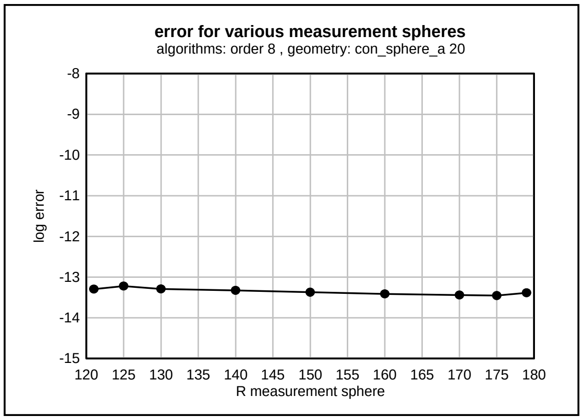

Fig. 8 The results of errors for various measurement spheres going from the inner shell of the geometry to the outer shell.

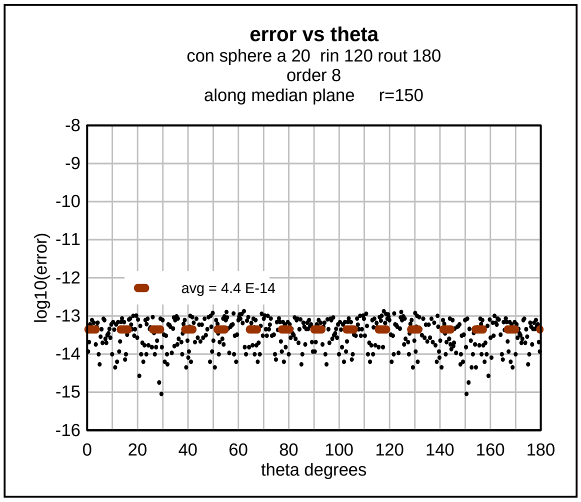

Fig. 9 The error vs theta for our highest density geometry.

VI. Notes of Caution

1) The first comment refers to the possibility of creating an order 10 algorithm set. Although this was done it was observed that finding a stable set of algorithms was considerably more difficult than the order 8 algorithm. The difficulty was due in part to the large number of meshpoints required (21) for the order 10 algorithm but also to the fact that the set of available meshpoints was constrained by software to the 3 surrounding rings of the central meshpoint. Thus any order 10 algorithm required ~1/2 of the available meshpoints contained in these rings. In a number of situations it was not possible to construct a stable algorithm without using meshpoints having a nearfarflag strictly less than that of the central meshpoint resulting in the value of the algorithm at that meshpoint being dependent on neighbors further from the boundary hence providing feed back from these mesh points. In addition relaxing the concentric sphere geometries in many instances was very sluggish and in practice the order 10 algorithm was considered unusable is not considered viable at this time.

2) The second comment refers to the brittleness of the solution. Although the number of test geometries was over several thousand and by design had a large and varied selection of neighbor configurations, the set was finite. Being finite means that there is no assurance of stability when a geometry representing a departure from those of the test set is considered. If such a situation is in fact encountered in practice, a successful algorithm would need to be determined and tested over the now expanded test set. This will both expand the stable test set and enhance the robustness of the algorithm. As it is likely that for most of the actual problems which will be encountered stability will not be an issue, further enlarging the test sets does not seem particularly useful.

VII. Summary and Conclusion

Extending FDM to curvilinear geometries has been enabled by expanding the meshpoints overlaid on the geometry to points on the other side of the boundary. This was done by extending the potential space by analytic continuation to points across the boundary. The process of creating the required algorithms for the expanded set of meshpoints has been detailed and the algorithm itself classified as to the order of the power series used to represent the potential near a meshpoint. Stable algorithms have been created for orders 2, 4, 6, and 8 and tested using a large selection of test geometries.

The precision of the solution and hence the method has been determined for concentric spheres for which there is a known solution. It was found for the higher mesh density overlays that the precision for the order 8 algorithm was ~ $<10^{-13}$ within the region between the spheres and in fact for the maximum density studied the average error for points near the median plane was found to be $4.24x10^{-14}$.

The implication of this study is that the technique described in this report is capable of precisions of the order of 10^{-13} thus solving the FDM curvilinear boundary value problem.

References

[1] D.W.O.Heddle, "Electrostatic Lens Systems", 2nd ed., 2000, Institute of Physics Publishing, ISBN 0-7503-0697-1, pp 32 to60.

[2] David Edwards, Jr, *Accurate Calculations of Electrostatic Potentials for Cylindrically Symmetric Lenses*, Review of Scientific Instruments, 54 (1983) 1229-1235.

[3] David Edwards, Jr, High *Precision Electrostatic Potential Calculations For Cylindrically Symmetric Lenses*, Review of Scientific Instruments, 78 (2007) 1-10.

[4] David Edwards, Jr., *High precision multiregion FDM calculation of electrostatic potential*, Advances in Industrial Engineering and Operations Research, Springer, 2008, ISBN: 978-0-387-74903-7

[5] David Edwards, Jr., *Single Point FDM Algorithm Development for Points One Unit from a Metal Surface*, Proceedings of International Multi Conference of Engineers and Computer Scientists 2008, Hong Kong, 19-21 March, 2008.

[6] David Edwards, Jr., *Accurate potential calculations for the two tube electrostatic lens using a multiregion FDM method*, Proceedings EUROCON 2007, Warsaw, Sept.9-13, 2007.

[7] David Edwards, Jr., *The Use of Shadow Regions in Multi Region FDM: High Precision Cylindrically Symmetric Electrostatics*, D. Taniar et al. (Eds.): ICCSA 2010, Part II, LNCS 6017, pp. 1–13, 2010. © Springer-Verlag Berlin Heidelberg 2010.

[8] D. Edwards, Jr. "*Highly accurate potential calculations for cylindrically symmetric geometries using multi-region FDM: A review*", Nuclear Instruments and Methods in Physics Research A, Elsevier B. V. 2011, pp 283-291.

[9] Anjam Khursheed, "*The Finite Element Method in Charged Particle Optics*", Kluwer Academic Publishers, 1999, ISBN 0-7923-8611-6 pp 46 to 60.

[10] Erwin Kreyszig, Advanced Engineering Mathematics, John Wiley and Sons, New York, 1962.