FDM for Curved Geometries in Electrostatics II: the Minimal Algorithm

David Edwards Jr., *Member, IAENG*

Abstract— **FDM (finite difference method) has by its nature difficulties during the relaxation process when finding the values of meshpoints close to the boundary of the geometry. Significant progress at solving this problem had been recently reported but unfortunately this solution has been found to be "brittle" in that occasionally a geometry departing slightly from one in the set of test geometries would be unstable. In addition meshpoints near the boundary required different algorithms dependent upon the local geometry of the meshpoint. The order 8 solution required in fact over 100 different separately determined algorithms. In this paper the solution has been made considerably more robust by the implementation of a new interpolation process concomitant with the development and use of a *single algorithm* for all meshpoints.**

Index Terms— **FDM, curved boundaries, electrostatics, high precision calculations, finite difference method.**

I. Introduction

FDM is a simple computational process for finding the solution to boundary value problems by an iterative method [1]. The FDM solution is a function, having fixed values on the boundary, which satisfies a differential equation at all interior points. The method involves overlaying a set of equally spaced meshpoints on the geometry, creating an algorithm that allows the determination of the value of any mesh point from its neighboring values, and relaxing the mesh using the created algorithm(s).

For boundaries lying on meshpoints, a multi-region process for FDM has been previously described [2-8] demonstrating the high precision capabilities of FDM for this class of geometries.

A serious limitation affecting the precisions obtained by FDM occurs when the boundary does not lie on but passes between meshpoints. A solution to this problem has been recently reported [9], in which the mesh point space was extended to the other side of the boundary and algorithms found to determine the values of these external meshpoints. A solution was found for a reasonably large set of test geometries but suffered from two distinct but likely related limitations. The first was that it was necessary for stability considerations to construct algorithm for a meshpoint depending on the local distribution of its neighbors. For the 8th order algorithm this involved creating 103 distinct algorithms each having been determined by a laborious trial and error process. [9] The result was stable over the set of test geometries for which the algorithms were determined, but when slight departures were made from the test geometries unstable geometries were consistently found albeit somewhat rarely. This situation (and criticism) was explicitly noted in [9]. Thus a solution has continued to be needed that would be considerably more robust than that of [9] and have a smaller number of required algorithms.

Without loss of generality the discussion the boundary value problem will be restricted to cylindrically symmetric electrostatics in which both Laplace's equation and cylindrical symmetry are assumed, the solution of this problem having immediate applicability to field of electron and ion optics

A. Background, the FDM process

In order to both standardize the notation and emphasize certain features of FDM, a quick overview of the FDM process is useful [1]. 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, the outer cylinder being at 10 volts with the ends at 0 volts.

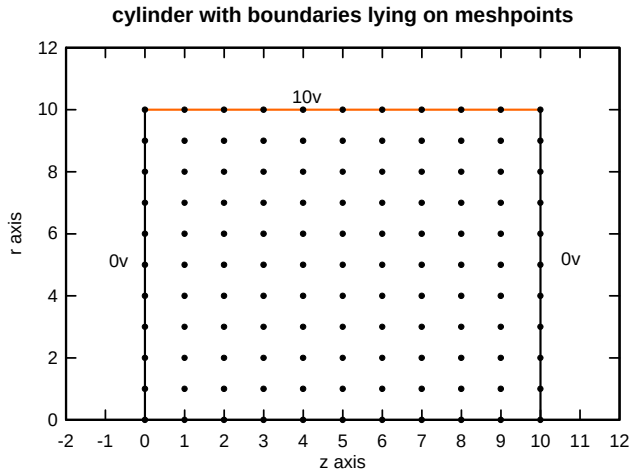

Figure 1. A cylinder is shown with its boundaries lying on meshpoints and the potentials on the various segments are indicated.

In order to relax such a mesh, the points of the mesh are stepped through in a sequential manner. At each meshpoint its value is found by means an algorithm using as input the

Manuscript received December 7 2013
David Edwards Jr.is with IJL Research Center, Newark, Vermont, (email dej122842@gmail.com)

ISBN: 978-988-19252-5-1
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

values of a selection of surrounding mesh points. This process is continued until all values have been determined for the 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 this occurs, the mesh is said to be relaxed and the solution determined.

B. The algorithm development process

The algorithm development process has been described previously [9] and only a brief summary of essential features is presented here in order to familiarize the reader with certain concepts of the process.

About any mesh point in 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 origin of the 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 +…+ c44*z*r^8+c45*r^8 +…+ *O* (j)

Where *O* (j) (read order j) means terms of order $r^k z^l$ are neglected for k+l> j. Thus for any meshpoint given the values of {cj} the value of the potential at any neighboring point r, z of the central point can be found. It will be the task of the algorithm to evaluate all of the required coefficients cj. The algorithm at a meshpoint will be a function which has as input the values of a selection of its neighbors, its distance "a" from the axis, and an index j. The output will be the value of cj.

To develop the algorithm a particular set of neighboring meshpoints is selected and using both the power series expansion of (1) and the differential equation governing the boundary value problem and the algorithm is be found by a process described in [9]. For an 8^{th} order algorithm the number of neighboring meshpoints required is 17.

The solution for any ck depends on both the particular set of 17 meshpoints {bj} and "a" the distance of the meshpoint from the axis (coming 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" depending upon the order of the algorithm. The coeff_bj's are determined using the algorithm development process described in [9].

C. Point type definitions.

As discussed in [9] high order algorithms for points near the boundary will require mesh points on the other side of the boundary. To accommodate this an overlaid mesh on the geometry is extended over the entire space.

Different types of meshpoints within the space are defined as follows*: Externalpts* will be those exterior to the geometry or within ~1/2 unit of the boundary. Those within ½ unit of the boundary will be further defined as *near_* points. The remainder are denoted as *internalpts*.

Internalpts are further subdivided into 4 subgroups, groups, ingeom_near, middle, far, and veryfar points. Ingeom_near points are internal pts with near pts as neighbors, ingeom_middle having ingeom_near points as neighbors, the other subgroups being similarly defined. It is noted that each subgroup is further from the boundary than its preceding one. The remaining internalpts points are denoted as ingeometry points.

For all points internal to the geometry which comprise the majority of meshpoints only the value of the central point is required during the relaxation process, its value being given by c0

For any external point its value will be determined by essentially integrating the field from the boundary to the point using the field centered on ***a*** particular internal point, thus allowing for the separation(at variable distances) of the central point from both the boundary and the external point. The details of this process are described below.

II. FINDING THE VALUES AT EXTERNAL POINTS BY INTERPOLATING FROM THE BOUNDARY

Figure 2 shows an external point **s**, the point **b** on the boundary (not necessarily a meshpoint) closest to **s**, and a point **o** about which the field is calculated**.** Point **o** will be considered the central point.

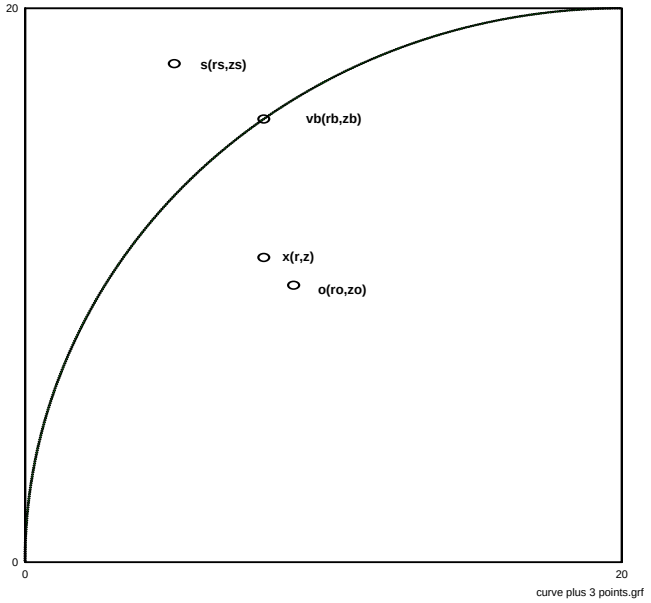

Figure 2. This figure shows the points used in deriving the algorithm for any external point (s). The coordinates for x, b, and s are relative wrt o. interpolations are made from the boundary to s using the field surrounding o.

Simplifying the notation of (2), the potential at any point x in an nhbd of **o** may be written as:

(3) v (r, z) = b0 + sumk ck*fk(r, z) sum k = 1 to kmax,

where b0 is the value at the point **o**, fk(r, z) is r^l*z^m for the kth term in the expansion (1). Note that r and z are relative coordinates with respect to point **o** and that kmax depends upon the order of the algorithm used.

Now the potentials at **b** and **s** may be explicitly written:

(4) vb = b0 + sumk ck*fk (rb_o, zb_o) sum k 1 to kmax

(5) vs = b0 + sumk ck*fk (rs_o, zs_o) sum k 1 to kmax

Where rx_o = rx - ro, zx_o = zx - zo for x being either b or s.

From the algorithm development process of [9], ck may be written as:

(6) ck = sumj abj_ck*bj, the sum being over all meshpoints bj used in creating the algorithm, and abj_ck is a coefficient depending upon both j and k and is determined by the algorithm development process [9].

ISBN: 978-988-19252-5-1
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

After a brief calculation, the following are found:

(7) vs = vb + sumj (coeff_bj *bj)

where:

(8) coeff_bj = sumk abj_ck*gk (rs, zs, rb, zb)

and

(9) gk (rs, zs, rb, zb) = {fk (rs_o, zs_o) - fk (rb_o, zb_o)}

Thus even though we are using all ck's in the evaluation of vs, at the end of the day an equation for vs is formulated in terms of a summation over the neighboring meshpoints, each meshpoint multiplied by coeff_bj. This is of identical form as the expression for the evaluation of all of the ingeometry meshpoints (see (2)) and finds use in the software development. It may be worthwhile to remark that when evaluating the value of **s**, the neighboring meshpoints to be used are those of **o** *and not the neighbors of* **s** itself since the coefficients ck used in (8) for **s** were determined using the neighbors of **o**.

III. The Minimal Algorithm.

It can easily be shown that the average of n algorithms is also an algorithm. By way of contrast it is noted that the sum of algorithms is not an algorithm which may be recognized by its violation of the sum rule for the coefficients of c0 (=1).

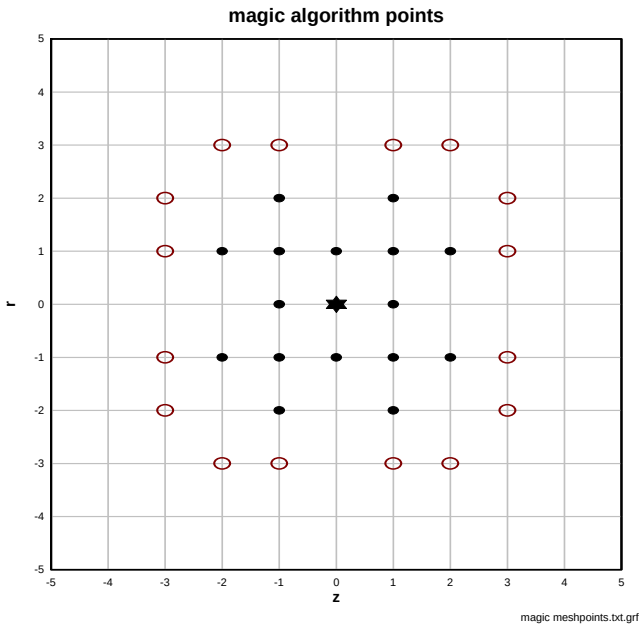

Figure 3. The base set of 16 meshpoints are given by the solid disks excluding the central point. For an order 8 algorithm a total of 17 meshpoints are needed and hence one of the additional empty circles is used to complete the set.

The above averaging proposition has been used in constructing the minimal algorithm for order 8. As the order 8 algorithm requires 17 neighbors any particular algorithm of the minimal algorithm set may be formed using the solid points of figure 3 together with one of the additional points of this figure. This construction provided a set of 16 distinct algorithms, one for each of the added circles. The set of base points, seen to exhibit notable symmetry, was chosen as the coefficients for c0 for this algorithm was found to be less than other algorithms investigated. The coefficients of the base points themselves were quite insensitive to the additional point chosen and hence the average coefficient for any of the base points was quite similar to the coefficient of any of the individual algorithms. The coefficient of the added meshpoint while being much smaller than any of the coefficients in the base set was not negligible and when an individual algorithm was used by *itself* many geometries were found which were unstable. However when the minimal algorithm was constructed, each of the coefficients of the added meshpoints would be reduced by 16 and hence the sensitivity of the stability to any added meshpoint was considerably reduced. This is likely the reason for its remarkable stability property. It is noted that other likely candidates for a base algorithm (which would then be rotated and reflected to construct an algorithm set) were also tried, and average algorithms constructed. None of these candidates exhibited the degree of stability of our minimal algorithm.

IV. Concentric Sphere Tests

A. The test geometries

The test geometries were chosen to be concentric spheres as these geometries both had a wide variety of point topologies (the type of neighbors surrounding any particular near boundary meshpoint) and allowed for exact solutions so that the precision as well as algorithm stability could be determined. The concentric sphere geometry consisted of two spheres with a common center, the outer having a potential 10 volts, the inner being at 0 volts. The basic geometry is shown in figure 4 from which an enlarged set is constructed by scaling.

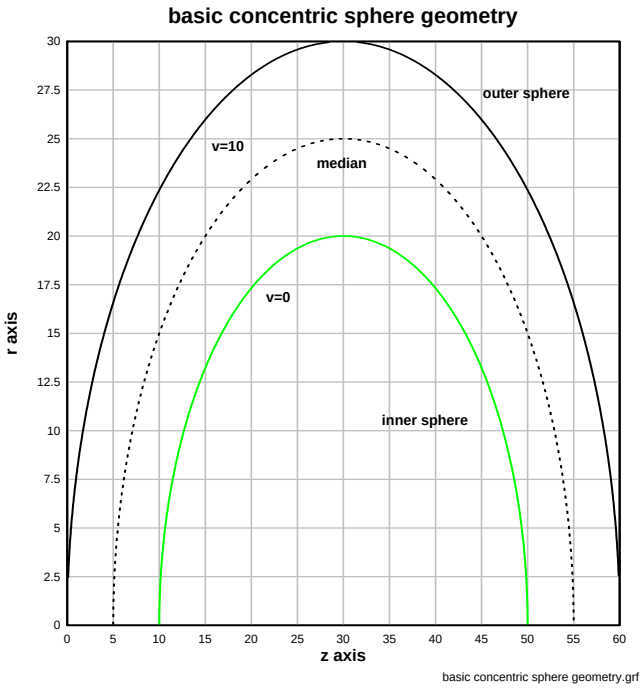

Figure 4 illustrates the basic concentric sphere geometry together with a median sphere over which precisions are evaluated. The outer, inner radius of the basic geometry was 30, 20 resp.

To form a set of 100 test spheres the basic geometry was scaled by 1+j*.025 for j=0 to 99. It is noted that each geometry would represent the same physical geometry while having an ever increasing number of mesh points in the grid overlay.

ISBN: 978-988-19252-5-1
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

For any external point **s**, let **b** be the closest boundary point to **s** and **o** be the central point from which the field is to be calculated for the interpolation of the value at **s** (see section 2 above) from the boundary. **o** is specified as the closest point to **b** *and also being in one of the following 5 sets*: {near_ pts, ingeom_near pts, ingeom_middle pts, ingeom_far pts, ingeom_veryfar pts} defined previously.

*B. Stability for types of central points chosen for point **o***

Taking the central point **o** from the set of near_ pts, all geometries were unstable. When the central point was taken from the set of ingeom_near pts the relaxations were stable but the convergence during the relaxation extraordinarily slow and in for this reason not usable. For the remaining sets, i.e., ingeom_middle, ingeom_far, and ingeom_veryfar the relaxations were stable and the convergence of the relaxation process reasonably rapid.

It is of course not surprising that an unstable situation resulted when the central point was taken from the set of near points as this situation closely mimics the situation of the previous work as in that work the near_ points were evaluated by interpolating from the boundary to the point itself which necessitated that the algorithm be matched to local neighborhood of the particular near_ point. One algorithm fits all manifestly did not work there and does not work here for this choice of central point.

C. Precision tests

For precision tests the base geometry of figure 4 was used and the precision test sets obtained by scaling by 1+.25*j for j =0 to 20. As the concentric sphere geometry has a known potential errors may be determined for any point in the relaxed net. The test plane chosen for error measurements was median sphere and errors for points on or near this surface were determined and then averaged over the test plane. A plot of the average error on the median sphere vs the geometry index is shown in figure 5 together with our previous data [9] using multi algorithms.

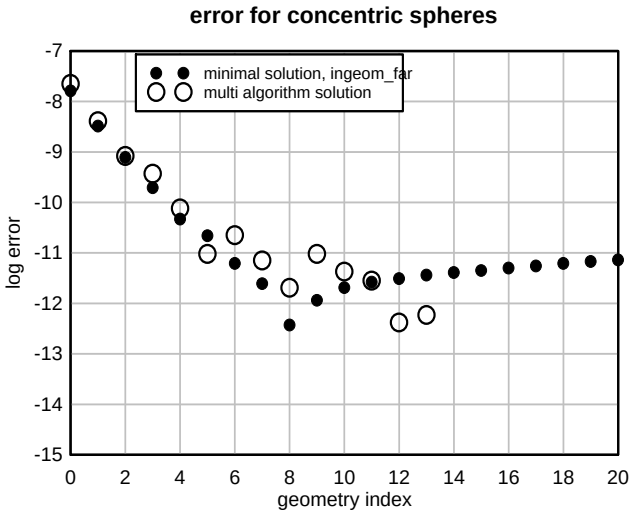

Figure 5. Shown is the average error over the median sphere vs the geometry index for both our previous solution (multi algorithm) and the current solution (minimal algorithm). Seen is the remarkable similarity of the two curves for two distinctly different algorithmic processes.

Shown in this figure is the solution from [9] using many algorithms having the algorithm center point being the point itself and the current solution using the minimal algorithm but centered on an ingeometry point well separated from the external point whose value is being calculated. The central point **o** in the above figure was taken from the set of ingeom_far points. It is clear that the *two solutions have in fact very similar resultant precisions for markedly different point value evaluations*. There does seem to be a noticeably larger scatter in the errors of the solution [9] for adjacent geometries than for the minimal solution. This scatter can also be seen in plots of error vs point position on the median sphere (not shown). It is noted that the error deviations of the current solution were reasonably constant over the test plane.

The minimal solution curve of figure 4 is seen to have a minimum in the error plot near index 8 after which the curve exhibits a slow rise to an error value of 10^{-11}. This type of effect has been seen in previous work and is attributable to the cumulative effect of roundoff errors. It should be noted that it was somewhat suppressed in the multi algorithm curve in that certain operations in the relaxation process were made with a high precision arithmetic unit which was not used in the current version as precisions less then ~10^{-11} were not required.

It is noted that when the central point **o** was taken from the set of ingeom_middle points the set of geometries was also stable as mentioned above but the precision was degraded by several orders of magnitude. At the present it is felt that this is the result of feedback from the setting of the near_ points on the values of the ingeom_middle points themselves. This is further supported by the fact that when the measurement sphere was scanned from the inner to outer sphere the errors were considerably enhanced near the inner or outer measurement spheres.

As reported above taking **o** from the set of ingeom_near points was not useable. In spite of this a measurement was made for the basic geometry and the result found (after a lengthy relaxation process) was that the precisions over the median sphere was 3 orders of magnitude less precise than the ingeom_middle situation further supporting the possible interpretation of these errors in the previous paragraph.

In figure 6 the precisions are graphed for central points taken from both ingeom_far and ingeom_veryfar points showing the similarity of the two curves with a slight deterioration (between 4 and 10) in the precision for the ingeom_veryfar points. As the distance from the particular point **s** to the central point is larger for ingeom_veryfar points than from the ingeom_far points the interpolation precision from these points is expected to be less precise than from the points closer to s. This observation is consistent with observed precision decrease. This graph also shows that for geometry indices in the region in which cumulative errors are dominating the two curves merge together which implies that the cumulative errors dominate the single point interpolation errors in this region. This figure gives clear evidence that the precisions of the process are capable of being less than 10^{-11} for either of the two choices of the central point **o**.

ISBN: 978-988-19252-5-1
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

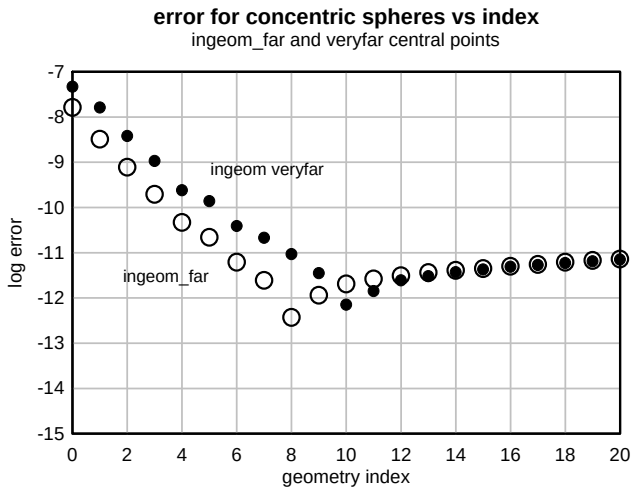

Figure 6. Plotted are the error distributions along the median plane for central points taken from the ingeom_far and ingeom_veryfar points showing both their similarity for large index and the slight precisional deterioration of the ingeom_veryfar points for points not in the region dominated by cumulative errors. (See text.)

V. Conclusion

The attempt of [9] to solve the potential evaluation for points near a curved boundary in FDM was partially successful in that it solved the problem for a particular set of test geometries with the proviso (considered unacceptable) that with departures of a geometry from any of the test set the solution might very well be unstable. The foundation of the present work has been built largely on [9] with modification in two significant ways. The first was to not use the point itself when interpolating to the point but to use a central point some distance from the point being evaluated. The effect of this was to minimize feedback between the external point and the central point on the values calculated and hence constructed a substantially more stable solution. The second modification was to construct a minimal algorithm being the sum of algorithms created using highly symmetric neighboring points as the base neighbor set. This single algorithm was then used in all interpolations *rather than the 103 algorithms required in [9]*.

The result has been a considerably more robust solution with very little if any deterioration in the resultant precision from that reported in [9]. The simplicity of the process makes it accessible to others who wish to take advantage of the enhanced FDM precision that order 8 algorithms provide.

The problem of setting the values of points in an FDM process near curved boundaries has been essentially unsolved since the inception of FDM. *It is now solved.*

References

[1] .D.W.O.Heddle, “Electrostatic Lens Systems”, 2nd ed., 2000, Institute of Physics Publishing, ISBN 0-7503-0697-1, pp 32 to60.W.-K. Chen, *Linear Networks and Systems* (Book style). Belmont, CA: Wadsworth, 1993, pp. 123–135.

[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-

[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] David Edwards, Jr. ,A Finite Difference Method for Cylindrically Symmetric Electrostatics having Curvilinear Boundaries, Lecture Notes in Engineering and Computer Science: Proceedings of The International Multi Conference of Engineers and Computer Scientists 2013, 13-15 March, 2013, Hong Kong, pp 1161-1167

ISBN: 978-988-19252-5-1
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)