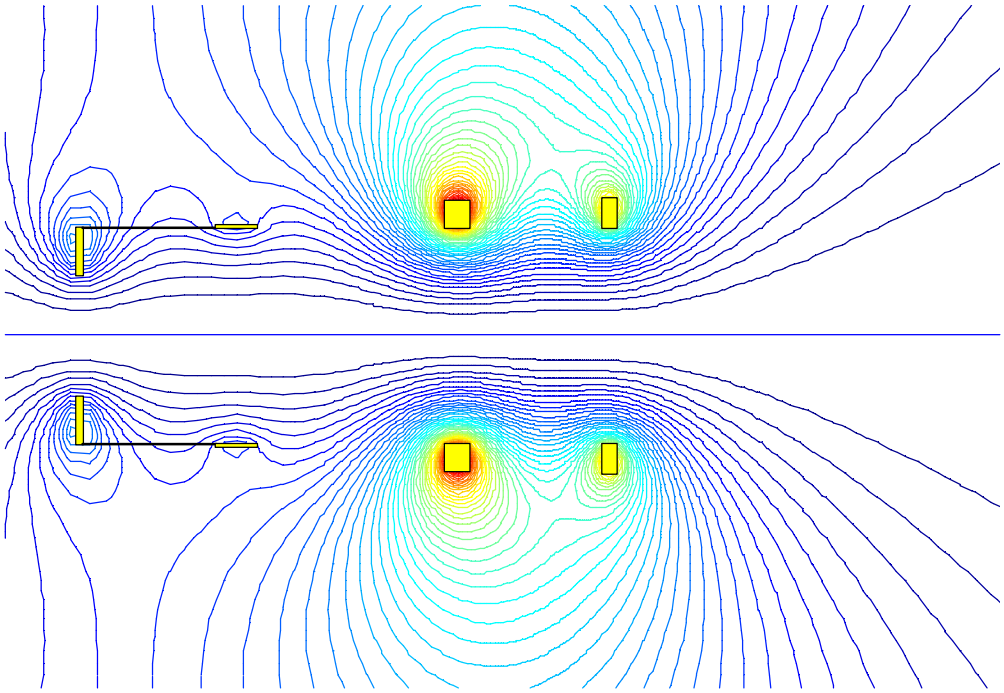
NASA/TP—2000–210194

Accuracy Improvement in Magnetic Field Modeling for an Axisymmetric Electromagnet

Andrew V. Ilin and Franklin R. Chang Díaz
Advanced Space Propulsion Laboratory
Lyndon B. Johnson Space Center
Houston, Texas

Yana L. Gurieva and Valery P. Il'in
Institute of Computational Mathematics and Mathematical Geophysics SD RAS
Novosibirsk, Russia



December 2000

The NASA STI Program Office ... in Profile

Since its founding, NASA has been dedicated to the advancement of aeronautics and space science. The NASA Scientific and Technical Information (STI) Program Office plays a key part in helping NASA maintain this important role.

The NASA STI Program Office is operated by Langley Research Center, the lead center for NASA's scientific and technical information. The NASA STI Program Office provides access to the NASA STI Database, the largest collection of aeronautical and space science STI in the world. The Program Office is also NASA's institutional mechanism for disseminating the results of its research and development activities. These results are published by NASA in the NASA STI Report Series, which includes the following report types:

- TECHNICAL PUBLICATION. Reports of completed research or a major significant phase of research that present the results of NASA programs and include extensive data or theoretical analysis. Includes compilations of significant scientific and technical data and information deemed to be of continuing reference value. NASA counterpart of peer-reviewed formal professional papers, but having less stringent limitations on manuscript length and extent of graphic presentations.

- TECHNICAL MEMORANDUM. Scientific and technical findings that are preliminary or of specialized interest, e.g., quick release reports, working papers, and bibliographies that contain minimal annotation. Does not contain extensive analysis.

- CONTRACTOR REPORT. Scientific and technical findings by NASA-sponsored contractors and grantees.

- CONFERENCE PUBLICATION. Collected papers from scientific and technical conferences, symposia, seminars, or other meetings sponsored or co-sponsored by NASA.

- SPECIAL PUBLICATION. Scientific, technical, or historical information from NASA programs, projects, and missions, often concerned with subjects having substantial public interest.

- TECHNICAL TRANSLATION. English-language translations of foreign scientific and technical material pertinent to NASA's mission.

Specialized services that complement the STI Program Office's diverse offerings include creating custom thesauri, building customized databases, organizing and publishing research results ... even providing videos.

For more information about the NASA STI Program Office, see the following:

- Access the NASA STI Program Home Page at ***http://www.sti.nasa.gov***

- E-mail your question via the Internet to help@sti.nasa.gov

- Fax your question to the NASA STI Help Desk at (301) 621-0134

- Telephone the NASA STI Help Desk at (301) 621-0390

- Write to:
  NASA STI Help Desk
  NASA Center for AeroSpace Information
  7121 Standard Drive
  Hanover, MD 21076-1320

NASA/TP—2000–210194

Accuracy Improvement in Magnetic Field Modeling for an Axisymmetric Electromagnet

Andrew V. Ilin and Franklin R. Chang Díaz
Advanced Space Propulsion Laboratory
Lyndon B. Johnson Space Center
Houston, Texas

Yana L. Gurieva and Valery P. Il'in
Institute of Computational Mathematics and Mathematical Geophysics SD RAS
Novosibirsk, Russia

National Aeronautics and
Space Administration

Lyndon B. Johnson Space Center
Houston, Texas 77058

December 2000

Acknowledgments

This research was partially sponsored by NASA's Lyndon B. Johnson Space Center.

Available from:

NASA Center for AeroSpace Information
7121 Standard Drive
Hanover, MD 21076-1320
301-621-0390

National Technical Information Service
5285 Port Royal Road
Springfield, VA 22161
703-605-6000

This report is also available in electronic form at http://techreports.larc.nasa.gov/cgi-bin/NTRS

Contents

Figures

Tables

Acronyms

ASPL	- Advanced Space Propulsion Laboratory
FDM	- finite difference method
FEM	- finite element method
FVM	- finite volume method
ICRF	- ion cyclotron resonance frequency
ICRH	- ion cyclotron resonance heating
JSC	- Johnson Space Center
MB	- megabytes
MFlops	- million float point operations
RAM	- random access memory
RTD	- radiation and technology demonstrator
VASIMR	- Variable Specific Impulse Magnetoplasma Rocket
1D, 2D	- one- and two-dimensional
5FD2N	- 5-point finite difference second-order on nonuniform mesh technique
9FD4U	- 9-point finite difference fourth-order on uniform mesh technique

Nomenclature

$\boldsymbol{A} = (0, A_q, 0)$	- vector magnetic potential
A_h	- a matrix in algebraic system for the discretized problem
$A^{(l)}$	- a local balance matrix in finite volume discretization
$\boldsymbol{B} = (B_r, 0, B_z)$	- vector magnetic induction
B_z^a, B_z^h	- analytical and numerical values of z-component of magnetic induction
c, c_1, c_2	- test coils
c_p	- coefficients in Tailor expansion
e	- residual function
$f = mr\boldsymbol{r}$	- right hand side in the magnetostatic equation
H	- distance from the coil to the domain boundary
h^r_i, h^z_j	- variable grid step sizes for nonuniform mesh
h_r, h_z	- constant grid step sizes for uniform mesh
I_{ij}	- finite volume integral
i, j	- mesh indices
J	- total current in the coil
K, E	- full elliptic integrals of the first and second orders
k	- order of accuracy
L_r, L_z, L_r^h, L_z^h	- 1D differential operators and their grid approximations
$M(.)$	- elliptic integral function
m	- number of quadrature points in numerical integration
N_r, N_z	- numbers of mesh steps in r- and z-directions
n	- length of the finite Tailor expansion
$n(e)$	- number of iterations needed to solve the algebraic system
P, Q	- the volumes of required computer memory and arithmetic operations
$p0, \ldots p8, f0$	- coefficients of finite difference scheme
$p00, \ldots p88$	- entries of the local balance matrix in finite volume approximation

R_3, Z_0, Z_3 - coordinates of the computational domain boundaries
(r, θ, z) - cylindrical coordinates
r_i, z_j - grid line coordinates
$S_r = \{(r', \theta, z): r'<r\}$ - cross section of the magnetic flux tube
S_{ij} - finite volume surfaces
s - coil independent variable
T_{AM} - number of 2D arrays used by the approximation method
U_{AM} - number of arithmetic operations per one iteration, per one mesh point
$u = r A_\theta$ - magnetic flux
V_{ij} - finite volume
w - finite volume weight parameter
$Z_1 < Z_2$ - left and right edges of the coil
β - auxiliary variable for boundary condition analysis
$\delta, \delta_B, \delta_j, \delta_s, \delta_r$ - calculated and experimental errors
δh - relative increment of the nonuniform mesh
ε_i - iteration criteria
γ_i - finite volume approximation coefficient
μ - magnetic permeability
ρ - current density
$\phi_p(.)$ - auxiliary function for boundary condition analysis
$\omega_n(.)$ - integrable function in boundary condition analysis
Ω - computational domain

Abstract

This paper examines the accuracy and calculation speed for the magnetic field computation in an axisymmetric electromagnet. Different numerical techniques, based on an adaptive nonuniform grid, high order finite difference approximations and semi-analitical calculation of boundary conditions are considered. These techniques are being applied to the modeling of the Variable Specific Impulse Magnetoplasma Rocket. For high-accuracy calculations, a fourth-order scheme offers dramatic advantages over a second-order scheme. For complex physical configurations of interest in plasma propulsion, a second-order scheme with nonuniform mesh gives the best results. Also, the relative advantages of various methods are described when the speed of computation is an important consideration.

1. Introduction

Computing the axisymmetric magnetic field generated by the current in a coil is a numerical problem frequently encountered in the mathematical modeling of plasma flows, charged particle beams [1, 2, 3], and other areas of physics. The problem can be approached in a variety of ways involving numerical integration methods as well as finite difference, finite volume, finite element, or other numerical methods for solving partial differential equations. All methods produce results, which may trade off computational speed for accuracy. The results depend very much on the order of approximation, the use of uniform or nonuniform grids, and other factors. For steady state solutions, the numerical approach may be different than for those solutions involving time dependence.

One important area of interest involves the study of plasma behavior in the Variable Specific Impulse Magnetoplasma Rocket (VASIMR) engine [4]. This new space propulsion concept is being studied at the Advanced Space Propulsion Laboratory (ASPL) of the NASA Johnson Space Center in Houston, Texas. Numerical simulation data are a crucial element of this research. They allow us to accurately predict system performance before developing costly experimental test equipment. In addition, numerical modeling is helpful in the design of plasma heating schemes and understanding plasma behavior in the VASIMR.

A number of factors direct requirements on code accuracy and computational speed. For example, the measurement accuracy of magnetic field with most available laboratory instrumentation is about *0.1*% [5, 6]. Furthermore, magnetic coils can only be positioned with a limited degree of precision. These experimental realities limit the relative error, which can be experimentally verified to about *0.001*. On the other hand, simulating plasma instabilities and other more complex dynamics may require a high-accuracy numerical solution. Such accurate solutions may also be useful in developing plasma control algorithms for an operational device. Still another important issue pertains to solver speed, as the complex iterative mathematical modeling of a magnetoplasma thruster requires multiple fast recalculation of the magnetostatic problem.

Finite difference and finite volume methods are widely used for solving the magnetostatic problem [7-9]. In previous publications, researchers have approached the problem in Cartesian coordinates. We, in turn, use a cylindrical coordinate formulation, assuming axial symmetry, and explore a high-order finite volume approximation. In addition, we analyze and describe the effects of uniform vs. nonuniform grids with respect to accuracy, computational speed and robustness. The results are presented in the context of the VASIMR described briefly in the next section.

2. VASIMR system

The VASIMR system is a high-power-density magnetoplasma rocket, which is capable of real-time exhaust modulation for optimum performance. In the system, shown in Figure 1, hydrogen plasma is created, heated, and expelled through an open-ended magnetic configuration to provide modulated rocket thrust. The general magnetic structure of the VASIMR is that of an asymmetric magnetic mirror, comprising three linked magnetic stages. The present system preserves azimuthal symmetry. The magnetic configuration of the VASIMR—combined with its approach to plasma generation, heating, and acceleration—results in a unique engine whose theoretical performance far exceeds that of present-day rockets [10, 11].

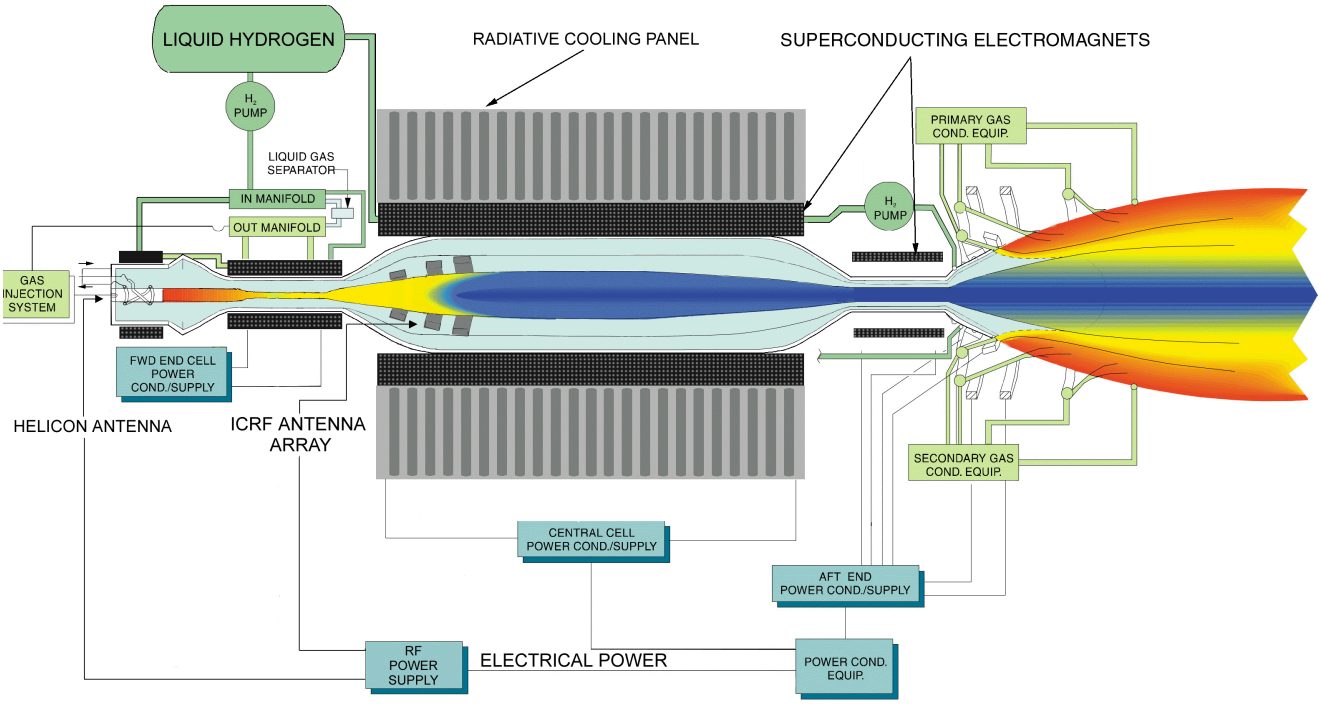


Figure 1. The VASIMR system.

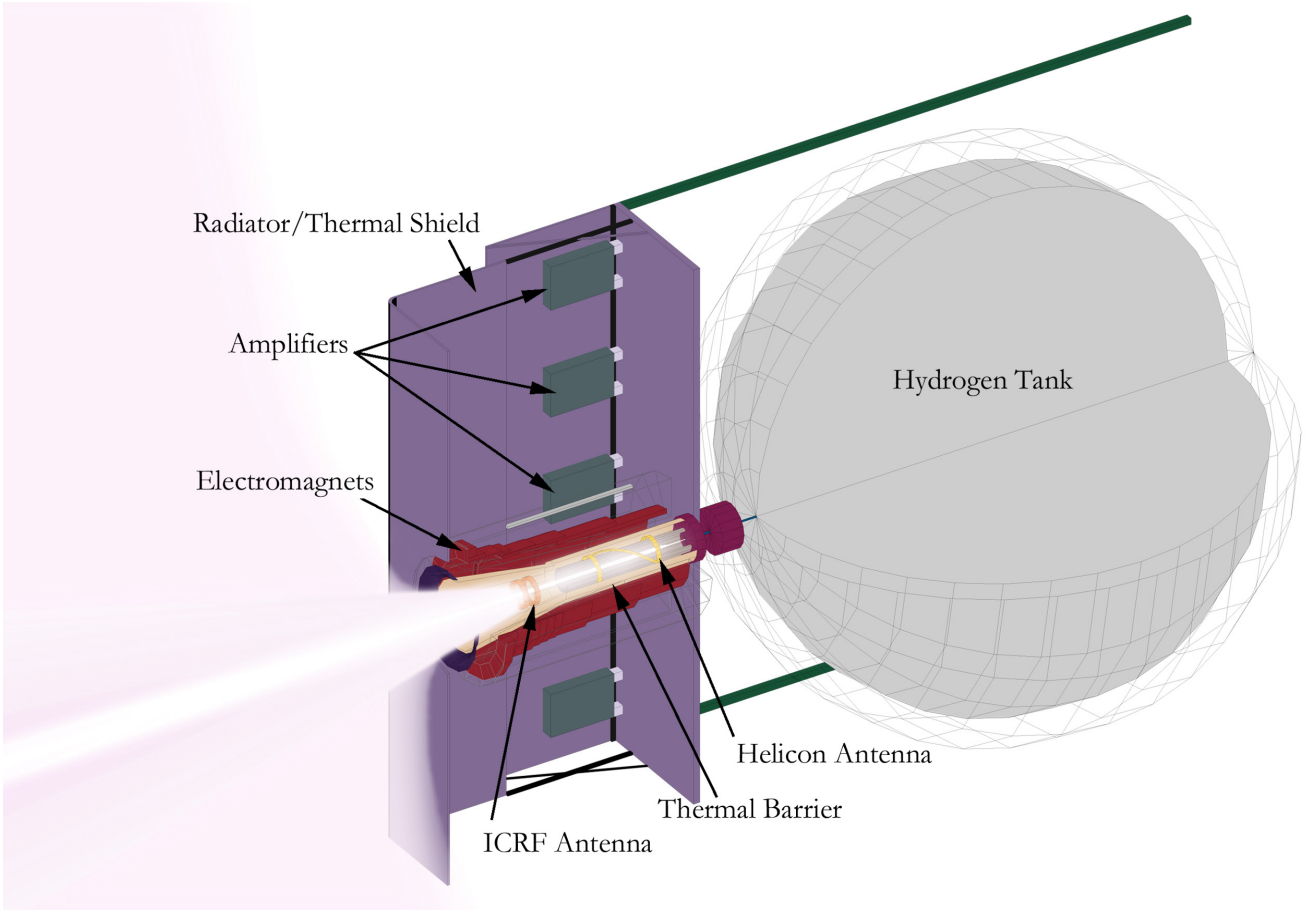


Figure 2. Projected design of the first spaceflight low-power VASIMR thruster.

A simpler variation of the VASIMR system is shown in Figure 2. In this configuration, the exhaust and heating cells are merged to provide a simpler geometry, albeit with some limitations in performance. This simpler VASIMR will be used in the first flight experiment envisioned to demonstrate this technology. Denoted the "radiation and technology demonstrator" (RTD), this experiment is currently the subject of considerable theoretical and experimental study.

An experimental evaluation of the VASIMR performance is being conducted in a laboratory device which uses simple liquid nitrogen-cooled copper magnets. These units are integrated into a vacuum structure by placing them in stainless steel enclosures, which can withstand the atmospheric pressure forces. The coils are attached to a structural spacer, which provides diagnostic and view port access. While this magnet assembly would be unsuitable for spaceflight, the resulting magnetic field closely resembles that of the actual RTD flight experiment and can be comfortably studied in the laboratory. Figure 3 shows a composite trimetric view of the present experiment configuration with an actual photograph of the device.

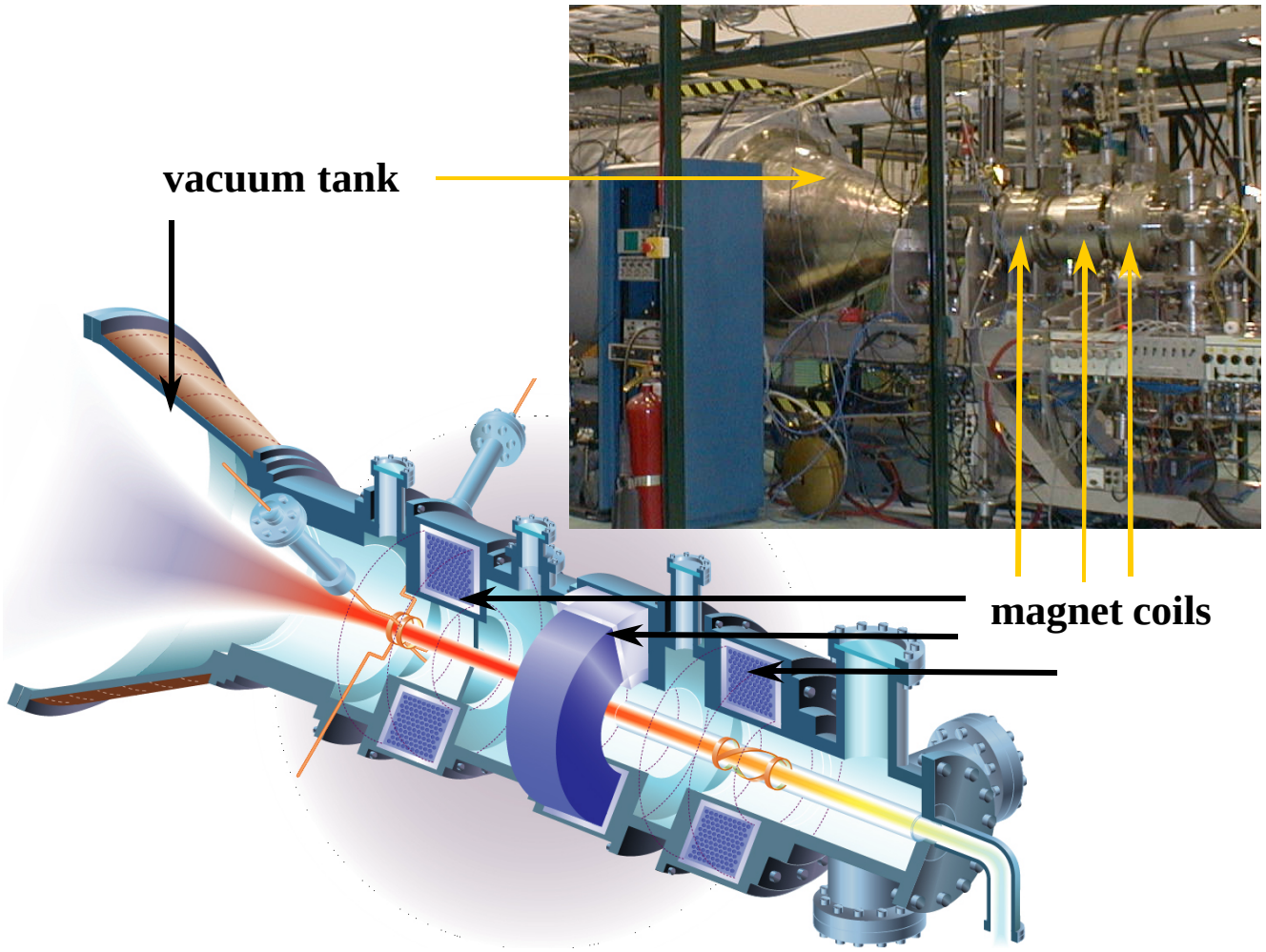


Figure 3. Trimetric view with an actual photo of the present VASIMR configuration at the ASPL.

3. The fundamental magnetostatic problem

The magnetostatic problem is a steady state case of two vector Maxwell equations:

$$\nabla \times \frac{1}{\mu} \boldsymbol{B} = \boldsymbol{j}, \qquad \boldsymbol{B} = \nabla \times \boldsymbol{A}, \tag{1}$$

where $\boldsymbol{B}$ is the magnetic induction vector, μ is the magnetic permeability, $\boldsymbol{j}$ is the current density and $\boldsymbol{A}$ is the magnetic vector potential. This paper is devoted to the fast and accurate calculation of $\boldsymbol{B}$ for a specified azimuthal current density $\boldsymbol{j}$ in the special case of cylindrical symmetry and constant magnetic permeability μ. In that case, the magnetic vector potential $\boldsymbol{A}$, written in the cylindrical coordinate system (r, φ, z), has only an azimuthal nonzero component: $\boldsymbol{A} = (0, A_\varphi(r, z), 0)$ and the problem (1) can be rewritten in the following form:

$$-r\frac{\partial}{\partial r}\frac{1}{r}\frac{\partial u}{\partial r} - \frac{\partial^2 u}{\partial z^2} = f(r,z) = \mu r j, \qquad (r,z) \in W, \tag{2}$$

where $u(r, z) = r A_\varphi(r, z)$ is the magnetic flux, shown in Figure 4, and W is a computational domain.

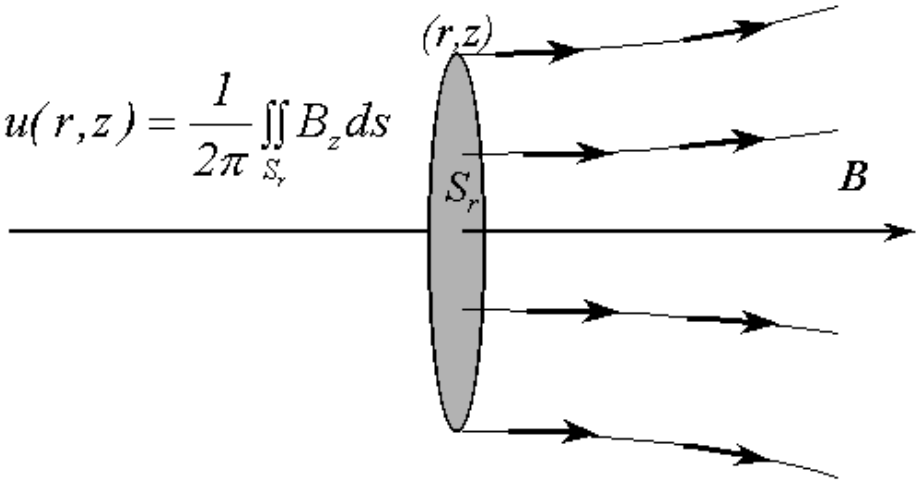


Figure 4. Definition of the magnetic flux u(r, z).

Theoretically, the azimuthal component of the magnetic vector potential for the magnetic field generated by a coil with arbitrary cross section can be computed by numerically integrating the following analytical formula over the coil cross section *c* [12]:

$$A_\theta(r,z) = \mu \iint_c \frac{4\rho(r',z')}{s}\sqrt{\frac{r'}{r}}\left[\left(1-\frac{s^2}{2}\right)K - E\right]dr'\,dz'; \qquad s^2 = \frac{4rr'}{(r+r')^2+(z+z')^2}, \qquad (3)$$

where *K* and *E* are full elliptic integrals of the first and second kind, respectively [13]. If the number of grid points and the number of numerical integration points are large, then the computational time can be excessive. In order to reduce computational time, equation (2) is solved in a gridded domain using finite difference (FDM), finite element (FEM), or finite volume (FVM) methods. In some cases, numerical integration of equation (3) is still useful for calculating the magnetic potential over a much smaller domain, such as for determining boundary conditions.

4. Solution strategy

In order to solve the magnetostatic problem of equation (2) in a computer we need to discretize it from its partial differential form into a system of algebraic equations using an approximation technique. Furthermore, the computational domain, shown in Figure 5, must be discretized also into a mesh of nodal points. Let us take on the domain discretization first.

4.1 Domain discretization

For simplicity, we consider a single coil with rectangular cross section $c = \{R_1 < r < R_2, Z_1 < z < Z_2\}$ in the rectangular computational domain $\Omega = \{0 < r < R_3, Z_0 < z < Z_3\}$. Since the coil is symmetric with respect to $z = 0$, the computational domain can be reduced by half to $\Omega = \{0 < r < R_3, 0 < z < Z_3\}$ with a boundary condition $du/dz = 0$ at $z = 0$, based on symmetry. Another boundary condition, $u = 0$ at $r = 0$, follows from the definition of $u(r, z)$.

Following the geometry of Figure 5, the above domain can be discretized as follows. Let h^r and h^z be mesh steps for the mesh:

$$\begin{cases} r_0 = 0 \\ r_1 = r_0 + h_0^r \\ r_2 = r_1 + h_1^r \\ r_3 = r_2 + h_2^r \\ \vdots \\ r_{N_r} = r_{N_r-1} + h_{N_r-1}^r \end{cases} \qquad \begin{cases} z_0 = 0 \\ z_1 = z_0 + h_0^z \\ z_2 = z_1 + h_1^z \\ z_3 = z_2 + h_2^z \\ \vdots \\ z_{N_z} = z_{N_{zr}-1} + h_{N_{zr}-1}^z \end{cases}$$

Finally, let us define a new array u_{ij} such that $u_{ij} = u(r_i, z_j)$ at each node of the rectangular grid.

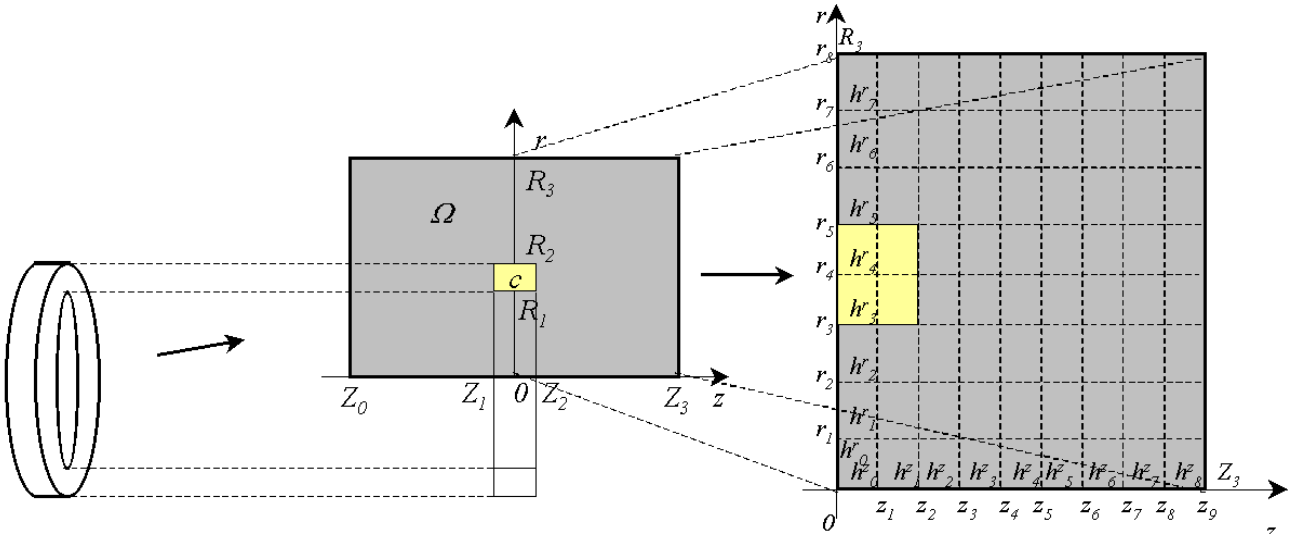


Figure 5. Defining the computational domain Ω for the axisymmetric single coil with rectangular cross section* c *and generating a rectangular grid for the problem. Here $N_z = 9$, $N_r = 8$.

4.2 Equation discretization

After the computational domain is discretized, the second step is to approximate equation (2) into a set of algebraic equations with respect to u_{ij}. We explore two techniques to do this. The first one is a 9-point, finite difference, fourth-order on uniform grid (9FD4U) approximation. The complete details of the derivation are described in Appendix 1. The result is the following system of algebraic equations with respect to u_{ij}:

$$\begin{aligned} &p0_{ij}u_{ij} - p1_{ij}u_{i-1,j} - p2_{ij}u_{i,j-1} - p3_{ij}u_{i+1,j} - p4_{ij}u_{i,j+1} - p5_{ij}u_{i-1,j+1} \\ &- p6_{ij}u_{i-1,j-1} - p7_{ij}u_{i+1,j-1} - p8_{ij}u_{i+1,j+1} = f0_{ij}, \qquad i = 2,...N_r - 1, \;\; j = 2,...N_z - 1, \end{aligned} \tag{4}$$

where the coefficients and right-hand side formulas are given by:

$$p0_i = \sum_k^8 pk_i, \qquad p2_i = \frac{2r_i}{3h_z{}^2(4r_i{}^2 - h_r{}^2)}\left(5 - \frac{h_z{}^2}{h_r{}^2} - 3\frac{h_r{}^2}{2r_i{}^2}\right) = p4_i,$$

$$p3_i = \frac{5h_z{}^2 - h_r{}^2}{6h_r{}^2h_z{}^2r_{i+1/2}} = p1_{i+1}, \qquad p7_i = \frac{1}{12r_{i+1/2}}\left(\frac{1}{h_r{}^2} + \frac{1}{h_z{}^2}\right) = p8_i = p5_{i+1} = p6_{i+1}, \qquad (5)$$

$$f0_{ij} = f_{ij} - \frac{1}{12}\left[\frac{r_i}{r_{i-1/2}}(f_{ij} - f_{i-1,j}) + \frac{r_i}{r_{i+1/2}}(f_{ij} - f_{i+1/2}) + 2f_{ij} - f_{i,j-1} - f_{i,j+1}\right].$$

Figure 6 shows stencils for the 9-point and 5-point schemes described in this paper.

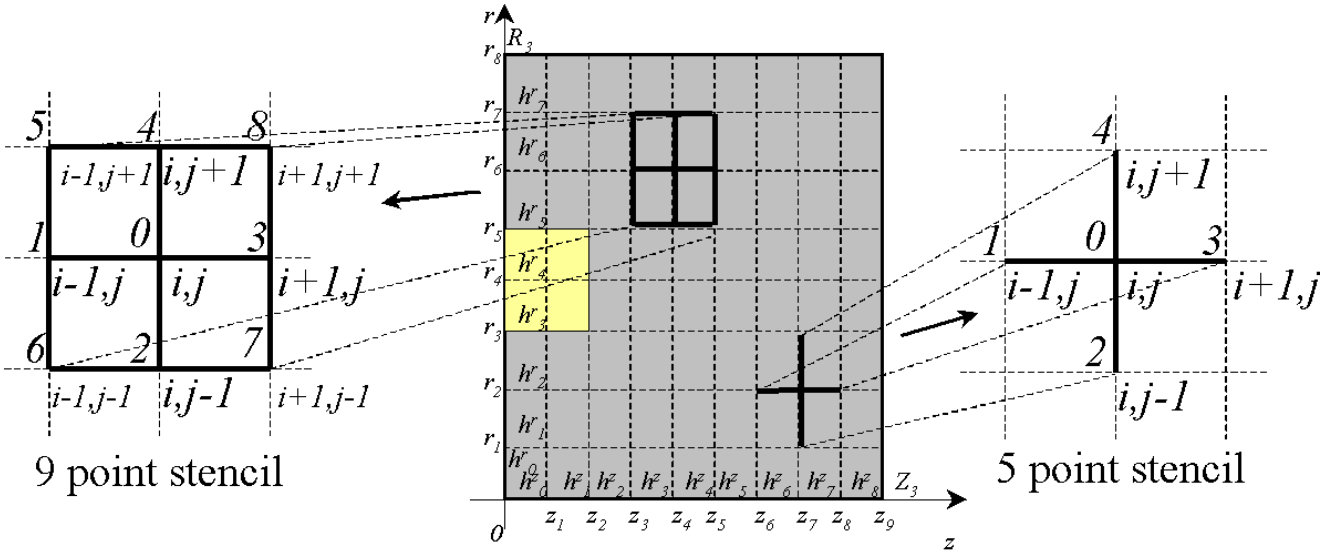


Figure 6. 9-point and 5-point stencils with local numeration of the nodal points.

A similar 9-point finite difference scheme for the Poisson equation in Cartesian coordinates is well known [7-9]. However, such schemes have not been investigated thoroughly for partial differential equations in cylindrical coordinates as in our case.

The efficient use of this method has some limitations. For example, its accuracy is greatly reduced when using a nonuniform mesh. Also, the grid aspect ratio (h_r/h_z) must be close to unity as described in equation (19) of Appendix 1. Further still, use of a uniform mesh requires the computational domain size to be comparable to the characteristic dimension of the magnet. In such cases it may be necessary to calculate non-zero boundary conditions, using analytical formula (3). This may not be convenient, when computing field values over much larger domains or when computing field contributions due to internal plasma currents.

The generalization of the high-order discretization using finite volume approach is described in Appendix 2. Appendix 3 discusses fast and accurate ways to derive boundary conditions by numerical integration of analytical formula (3). The results of the investigation of the 9FD4U method will be presented below after we discuss a second approach.

While the 9FD4U method has grid uniformity limitations, a simpler scheme using a 5-point stencil is available. This method does not suffer from previous mesh uniformity restrictions and yields faster solutions due to the reduced computational load. These benefits, however, come at the expense of accuracy. We have compared the relative advantages and disadvantages that each method offers and present these results in tabular form in Table 1. We now describe the 5-point scheme in detail.

We will use the name "5FD2N" to refer to the 5-point finite difference second-order on nonuniform mesh technique, which can be presented by the following system of algebraic equations with respect to u_{ij}:

$$p0_{ij}u_{ij} - p1_{ij}u_{i-1,j} - p2_{ij}u_{i,j-1} - p3_{ij}u_{i+1,j} - p4_{ij}u_{i,j+1} = f0_{ij}, \quad i = 2,...N_r - 1, \quad j = 2,...N_z - 1, \qquad (6)$$

where the coefficients and right-hand side formulas are given by:

$$\begin{aligned} p0_{ij} &= \sum_{k}^{4} pk_{ij}, \qquad p1_{ij} = \frac{h^z_{j-1} + h^z_j}{r_{i-1/2} h^r_{i-1}} = p3_{i-1,j}, \\ p2_{ij} &= \frac{h^r_{i-1} + h^r_i}{r_i h^z_{j-1}} = p4_{i,j-1}, \qquad f0_{ij} = \frac{(h^r_{i-1} + h^r_i)(h^z_{j-1} + h^z_j)}{2r_i} f_{ij}. \end{aligned} \qquad (7)$$

The scheme has been previously used [2]. One advantage of this method is that the matrix of the system for any mesh steps is less dense than one for the 9-point scheme and the system (6-7) has a solution for any grid. Another advantage is that using a nonuniform mesh allows the use of a large computational domain without sacrificing the minimal mesh size and keeping computational error small. Furthermore, using a nonuniform mesh allows the use of very large domains. This in turn permits the boundary condition $u = 0$ to be used instead of the more computationally expensive numerical integration of formula (3).

Both numerical approximations are compared below.

Features \ Methods	9FD4U		5FD2N	
1) Approximation accuracy	fourth-order: $k = 4$	☺	second-order: $k = 2$	☹
2) Sparsity of the matrix	9-diagonal	☹	5-diagonal	☺
3) Mesh requirements	Required uniform mesh with h_r/h_z in some range. Composition of uniform meshes introduces increasing error	☹	Nonuniform mesh works as good as uniform, when $\|h_i - h_{i+1}\| = O(h)$	☺
4) Computational domain	Because of 3), domain cannot be very large	☹	Because of 3), mesh can be large	☺
5) Boundary conditions	Because of 4), BC needs to be calculated using numerical integration of analytical formula	☹	For large computational domain, BC can be zero without introducing big error	☺
6) Iterative solver performance (number of iteration)	Because of 3), matrix does not have big condition number, hence the solver needs fewer iterations	☺	Because of 3), the matrix may have a big condition number, hence the solver needs more iterations	☹
7) Number of arithmetic operations per iteration	Because of 2), each iteration requires more arithmetic operations	☹	Because of 2), each iteration requires fewer arithmetic operations	☺
8) Applicability to currents in a complex geometry	Because of 5), it is very difficult to use the method for non-rectangular coils or plasma currents	☹	Because of 5), there is no problem using the method for non-rectangular coils or plasma currents	☺
9) Inclusion of the current source in the domain	If the currents source is inside the computational domain, the error may be large	☹	Inclusion of the current source into the domain does not increase the error	☺

Table 1. Comparison of the 9-point finite difference fourth-order on uniform grid approximation with 5-point finite difference second-order on nonuniform mesh approximation. Symbols ☹ and ☺ are used to show pros and cons for each scheme.

One of the important features of approximation methods, compared in the Table 1, is accuracy. In general, accuracy refers to how close the numerical solution approaches the exact one. We choose to measure the accuracy by bounding the difference between these two solutions to some constant times a power of the step size. This is expressed by

$$max\,|\,u_{ij} - u^a(\,r_i\,,z_j\,)\,| \leq C\,min(\,h\,)^k = O(\,h^k\,)\,, \tag{8}$$

where the analytical solution u^a can be calculated from formula (3). The value of k refers to the order of the numerical error, or order of accuracy. The higher the value of k, the more accurate the solution becomes. In order to find a value of k, one can proceed with the full numerical and analytical evaluation of the left-hand side. However, it has been shown [2] that the k value can be obtained without requiring this lengthy approach, by using a qualitative analysis of the numerical approximations. Using this method, the value of k is found to be $k = 4$ for the 9FD4U scheme. For 4FD2N, if a quasi-uniform grid is used, i.e. $|h_i - h_{i-1}| = O(h^2)$, then $k = 2$. $k = 1$ for both schemes, when an arbitrary uniform grid is used.

4.3 Solving the algebraic system

Both finite difference schemes represent the partial differential problem (2) by a system of linear algebraic equations. The 9FD4U algebraic system is presented in (4) and the 5FD2N in (6). The resulting linear algebraic systems can be represented by a matrix of the form:

$$A_h\,u_h = f_h. \tag{9}$$

Such algebraic systems can be solved either by direct or iterative methods [2, 14].

The resulting linear algebraic systems, presented in this report, are solved by the fast iterative implicit incomplete factorization method [14 - 15]. The stop criteria for the iterative process of calculating u_h^n is the following inequality:

$$\frac{\left| f_h - A_h u_h^{\,n} \right|}{\left| f_h - A_h u_h^{\,0} \right|} \leq \varepsilon_i = 10^{-11}\,. \tag{10}$$

The number of iterations $n(\varepsilon_i)$ depends on the computational domain and grid and normally is between *20* and *40*.

5. Numerical experiments for a single coil

5.1 1D tests

A number of numerical experiments were performed to test the accuracy of the numerical methods. The first and simplest corresponds to the infinite solenoid representing the 1D solution. The solution results are shown in Appendix 4. The error for the 9FD4U scheme is virtually zero. For the 5FD2N, the error is very small and decreases as h^2. The second and more interesting set of tests is shown below and corresponds to a single coil of finite size.

5.2 Coil complexity

The complexity of solving a magnetostatic problem for a finite rectangular cross-section coil is affected by the coil aspect. This is so because the mesh must be compatible with the coil shape. From the mesh-generation point of view, it is difficult to discretize the computational domain for a very thin, very narrow coil or a coil with a very small hole, as illustrated in Figure 7. In general, a nonuniform mesh is preferable when dealing with a difficult coil.

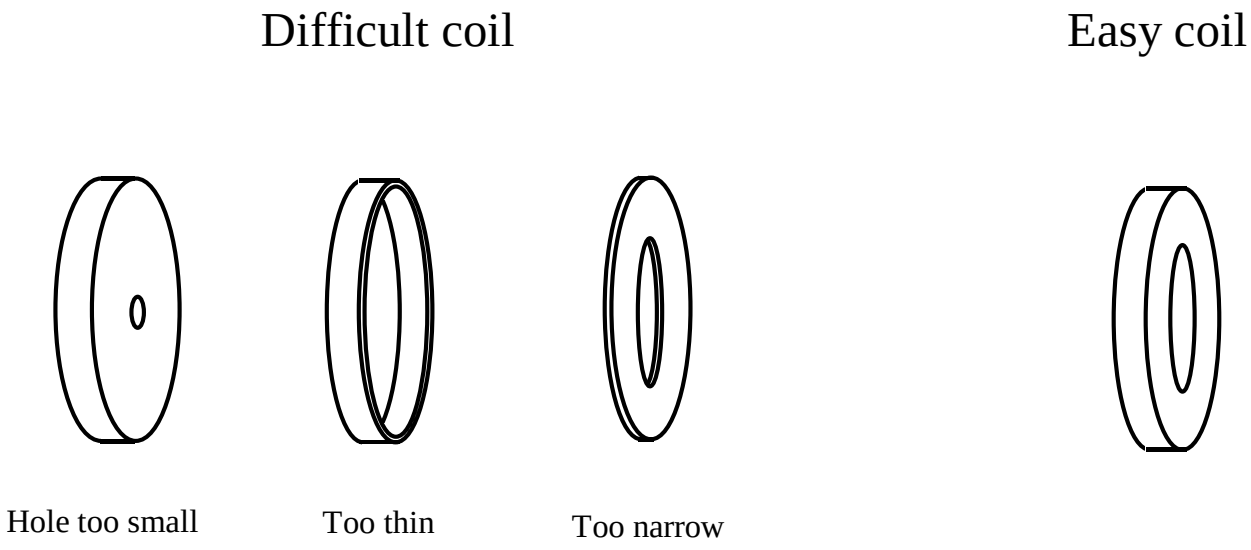


Figure 7. Difficult vs. easy coils from the mesh-generation point of view.

5.3 Deriving an exact solution and calculating the numerical error

The first test was considered for the "easy" coil "c1" with a cross section $c = [0.5, 1] \times [-0.5, 0.5]$ and a current density corresponding $\mu\rho = 1$. The solution for this problem is plotted in Figure 8.

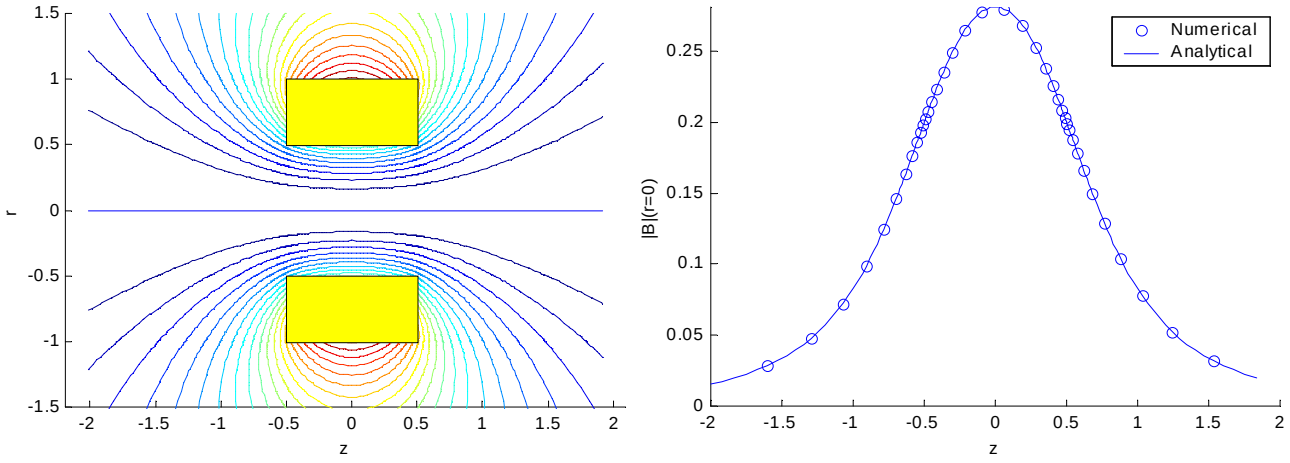


Figure 8. A magnetic field near an "easy" coil. Left: magnetic field lines (contour lines for the magnetic flux u(r, z)). Right: axial magnetic field, calculated numerically and analytically along r = 0.

To study the accuracy of the two methods, the numerical solution was compared with a known analytical solution calculated on the symmetry line $r = 0$ by the formula

$$B_z^a(0,z) = \frac{\mu\rho}{2}\left[(z-Z_1)\ln\frac{R_2+\sqrt{R_2^2+(z-Z_1)^2}}{R_1+\sqrt{R_1^2+(z-Z_1)^2}} - (z-Z_2)\ln\frac{R_2+\sqrt{R_2^2+(z-Z_2)^2}}{R_1+\sqrt{R_1^2+(z-Z_2)^2}}\right], \quad (11)$$

where current density ρ is assumed constant in the coil. The following maximal and point-wise relative errors on the symmetry line were observed:

$$\delta = \max_{z_j}\delta_B(z); \qquad \delta_B(z_j) = \frac{\left|B_z^a(0,z_j) - B_z^h(0,z_j)\right|}{\max_j\left|B_z^a(0,z_j)\right|}, \quad (12)$$

where $B^h(0,z)$ is the numerical solution calculated from the numerical magnetic flux solution u^h at the symmetry line $r = 0$, according to the scheme described in Appendix 5.

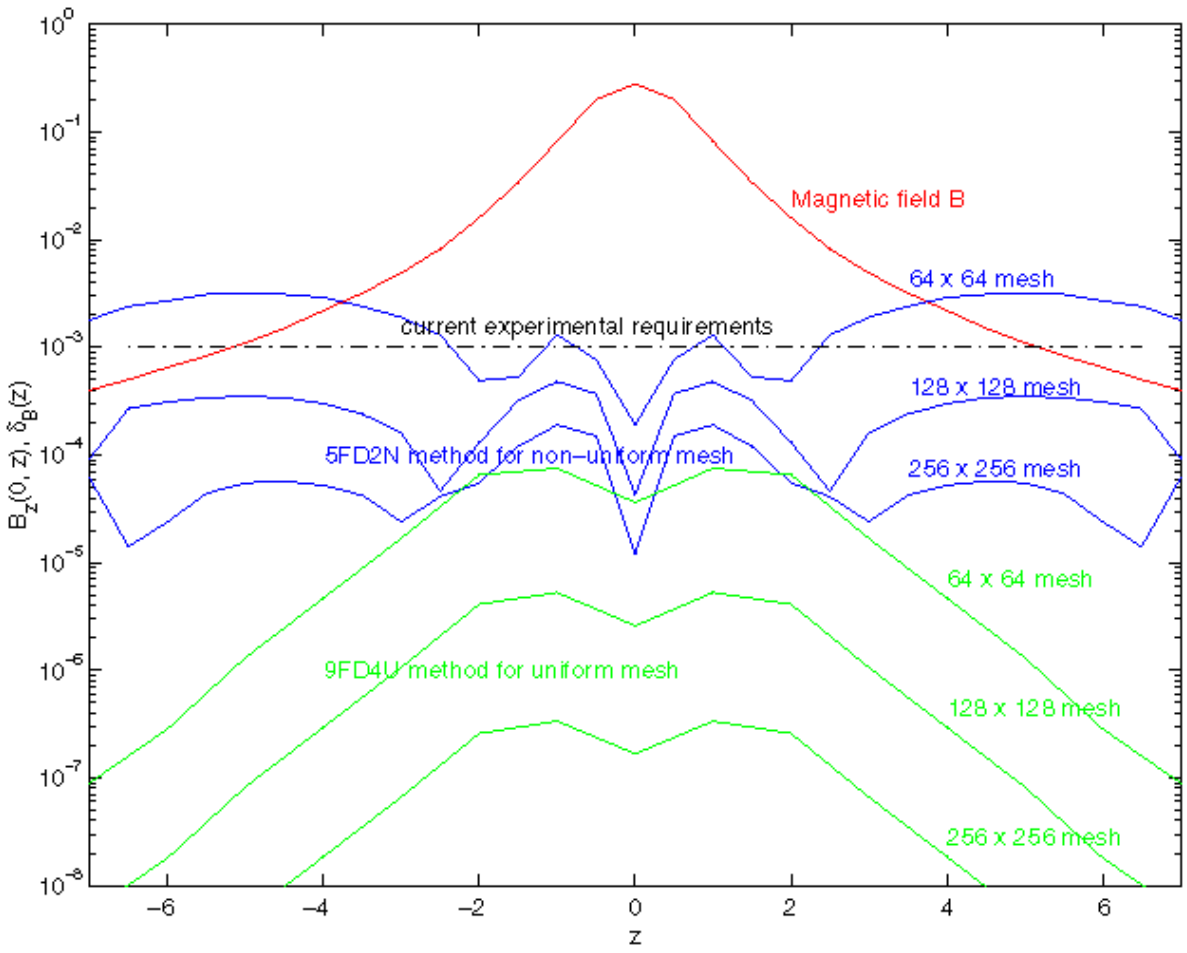


Figure 9. Magnetic field B_z (red line) and relative errors δ_B of the magnetic field B for the single coil "c1", calculated on the symmetry line r = 0 by 5FD2N (blue lines) and 9FD4U (green lines) methods using three different mesh sizes: 64 × 64, 128 × 128 and 256 × 256.

5.4 Finite difference grid

The first comparison of accuracy of the two methods considers the dependence of the relative error on the grid size. This is shown in Figure 9.

Two families of error plots corresponding to 5FD2N and 9FD4U methods are shown in blue and green color, respectively. In each family three mesh structures were used, including 64×64, 128×128 and 256×256 grid points. Several features of Figure 9 merit discussion.

The second-order non uniform-mesh method 5FD2N is generally less accurate but has more constant error along the domain. The fourth-order uniform-mesh method 9FD4U produces more accurate solution but shows diverged error over the computational domain.

In all cases, the error decreases with increasing grid size in accordance with theoretical predictions. For example, the second-order 5FD2N method makes the maximal relative error δ decrease by factors of 4 as *0.003*, *0.0008* and *0.0002*, when the grid step size h is divided by 2. On the other hand, the fourth-order 9FD4U method makes the relative error δ go down by factors of *16* as $8 \cdot 10^{-5}$, $5 \cdot 10^{-6}$, and $3 \cdot 10^{-7}$, when the grid step size h is divided by 2. The strong discontinuities in the error plots are due to several factors, such as the behavior of the analytical solution and its derivatives, the choice of grid and the method used. The use of a fully adaptive grid will tend to smooth out these irregularities. This is beyond the scope of this paper.

One can see that, to meet the current experimental requirements on accuracy of 10^{-3}, the scheme 5FD2N should be used with a 128×128 mesh. For the same requirements, the 9FD4U method can be used with a 64×64 mesh. This shows an advantage of the 9FD4U (uniform) method when the magnetostatic problem is solved for an easier coil.

Mesh configurations for both methods, as applied to an easier coil, are shown in Figure 10. The 2D nonuniform mesh in Figure 10 (b) is a product of two nonuniform 1D meshes. Each of them is fine next to points of singularity: edges of the coil and the symmetry line $r = 0$. It was created independently in the r and z directions, as described in Appendix 4 for the 1D case.

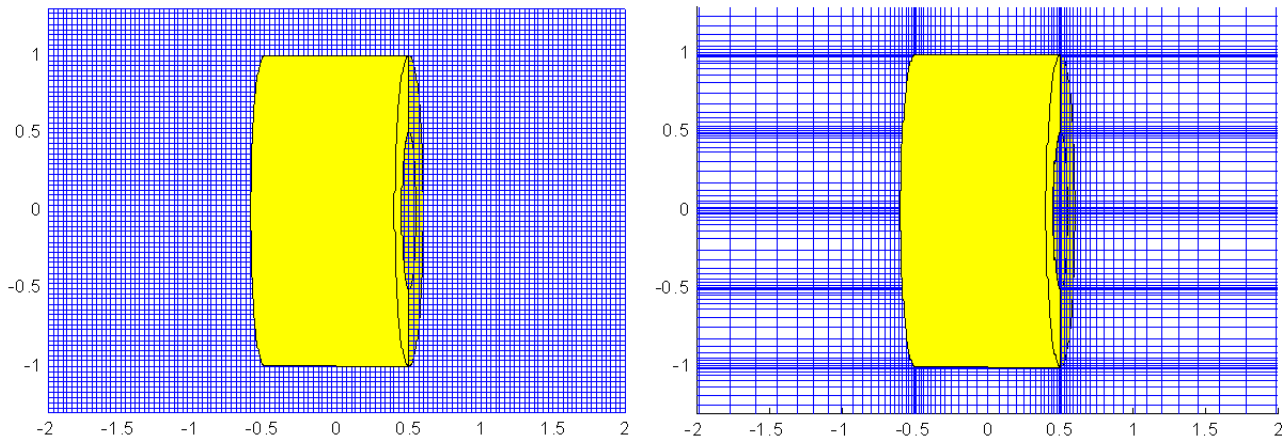


Figure 10. An example for the uniform (a) and nonuniform (b) finite difference meshes for "c1" coil.

5.5 Domain size and boundary conditions

In the above calculation, the size of the computational domain: $\Omega = \{[0, 8] \times [-8, 8]\}$ was chosen to be about ten times the size of the coil. The requirement is particular to the magnetoplasma thruster design. If the magnetic field needs to be calculated only near the coil, the computational domain can be much smaller, simplifying the problem considerably.

Choosing an appropriate domain size is related to choosing appropriate boundary conditions. As mentioned earlier, there are two ways of handling boundary conditions. First, a semi-analytical formula given by equation (3) can be used to compute the field value at the boundary for both methods. Second, the magnetic potential is assumed to be zero at the boundary, which is a good assumption for a large domain.

In the first approach, formula (3) requires the use of a numerical integration technique described in Appendix 3. Convergence of the boundary condition calculation method is presented in Appendix 6. This method works well for both 9FD4U and 5FD2N schemes, when the current source is given as a coil with small cross section.

For the large enough domain, the 5FD2N method can work efficiently with zero boundary conditions. This property allows it to work efficiently for problems with complex current source terms, such as plasma currents, when the first approach based on formula (3) cannot be used. The accuracy of the 5FD2N scheme with zero boundary conditions is demonstrated below. The 9FD4U scheme is taken out of comparison there, because a large computational domain with a difficult coil requires too large a uniform mesh.

Since an exact solution may not be small on the boundary points close to the coil, the use of a small domain with zero boundary conditions could be a source of the error. On the other hand, a large domain could be an error source as well, if the grid is too coarse. The optimal domain size depends on the coil geometry and can be found from numerical experiments. The dependence of the numerical error from the domain size R_3 and coil geometry for the fixed number of grid points is presented in Figure 11.

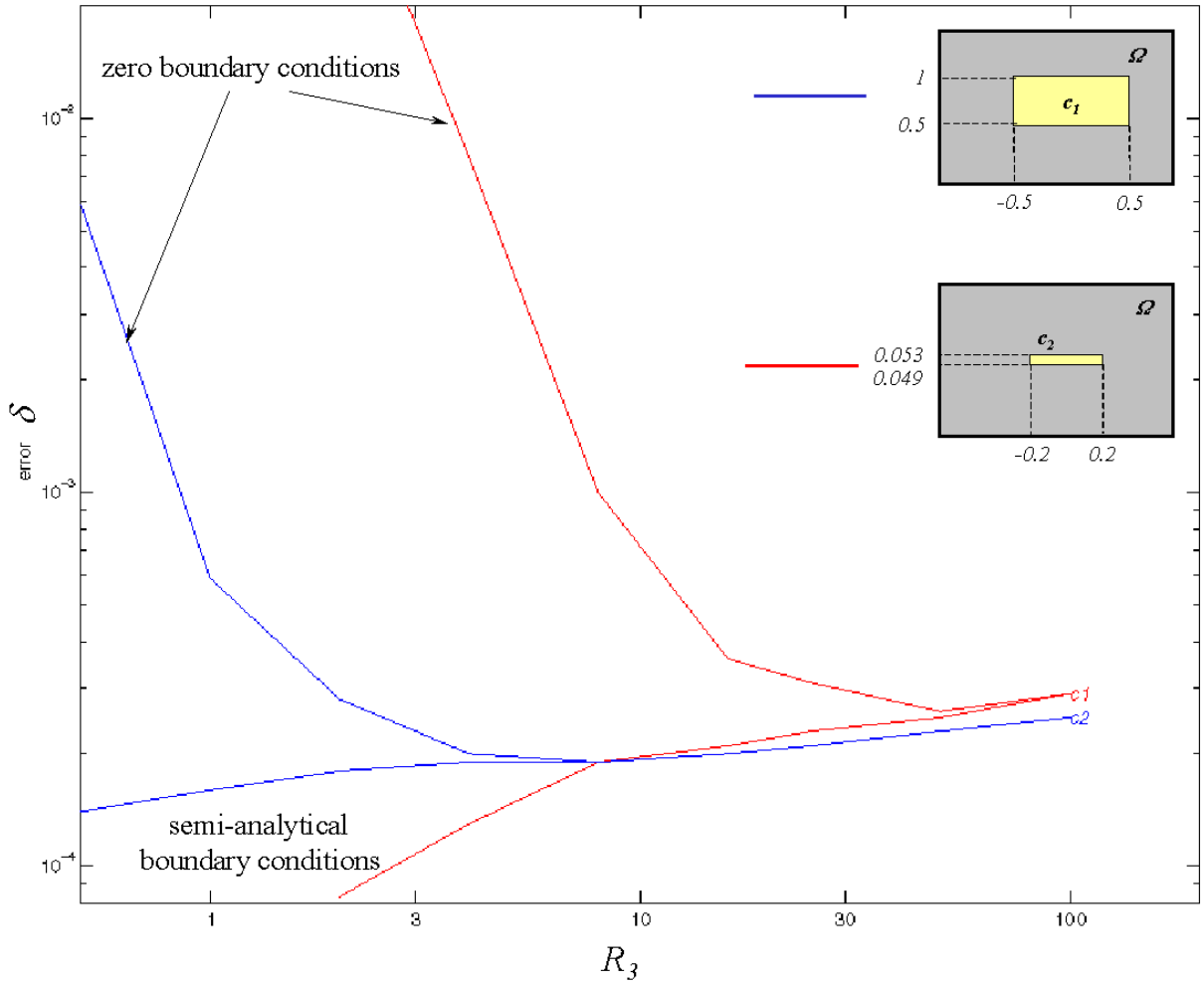


Figure 11. Maximal relative error δ of the 5-point scheme 5FD2N, as a function of the domain size R_3. 2D calculations of the magnetostatic problem for "c1" and "c2" coils were performed, using semi-analytical and zero boundary conditions. For each R_3, the nonuniform mesh 128 by 128 was built to minimize the relative error for 1D problem.

Let's analyze the error for the "easy" coil "c1" and the long-shaped "difficult" coil "c2" with cross section $c = [0.049, 0.053]\times[-0.2, 0.2]$, shown in Figure 11. The use of the semi-analytical boundary conditions provides less error, especially for the small domain size R_3. Use of zero boundary conditions produces a big error for the small domain size. Also, for sufficiently large domain size, the error would grow, because the grid becomes coarse. The zero boundary conditions yield a minimum error corresponding to some "optimal" value for the domain size R_3.

Since the coil "c2" has a smaller and thinner cross section than "c1", the optimal domain size is different. While the total number of mesh points is fixed at *128* by *128*, the plots have minimums at $R_3 = 50$, $\delta = 0.0003$ for coil "c1" and $R_3 = 8$, $\delta = 0.0002$ for the coil "c2". These numerical tests demonstrate the robustness and good performance of the 5FD2N scheme for solving the magnetostatic problem for a difficult coil.

6. Numerical experiments for a multi-coil system

Our ultimate test demonstrated in this section corresponds to the multiple-coil system involved in the magnetoplasma thruster. This system consists of rectangular cross-section coils of different shape. This is why our magnetostatic solver needs to work well for a variety of coil geometries.

In practice, the usual problem consists of computing the magnetic field of the set of various coils of different sizes and placed at varying distances from one another. Fortunately, the principle of superposition applies and the total field of the coil system is the sum of the fields of separate coils. In doing so, the size of the computational domain and the number of mesh-points can be decreased dramatically. As shown in Figure 12, a nonuniform mesh has singular points at the coil edges; hence the number of singular mesh points is bigger for the multi-coil system. By breaking up the magnet into single coils, the number of singular mesh points is greatly reduced.

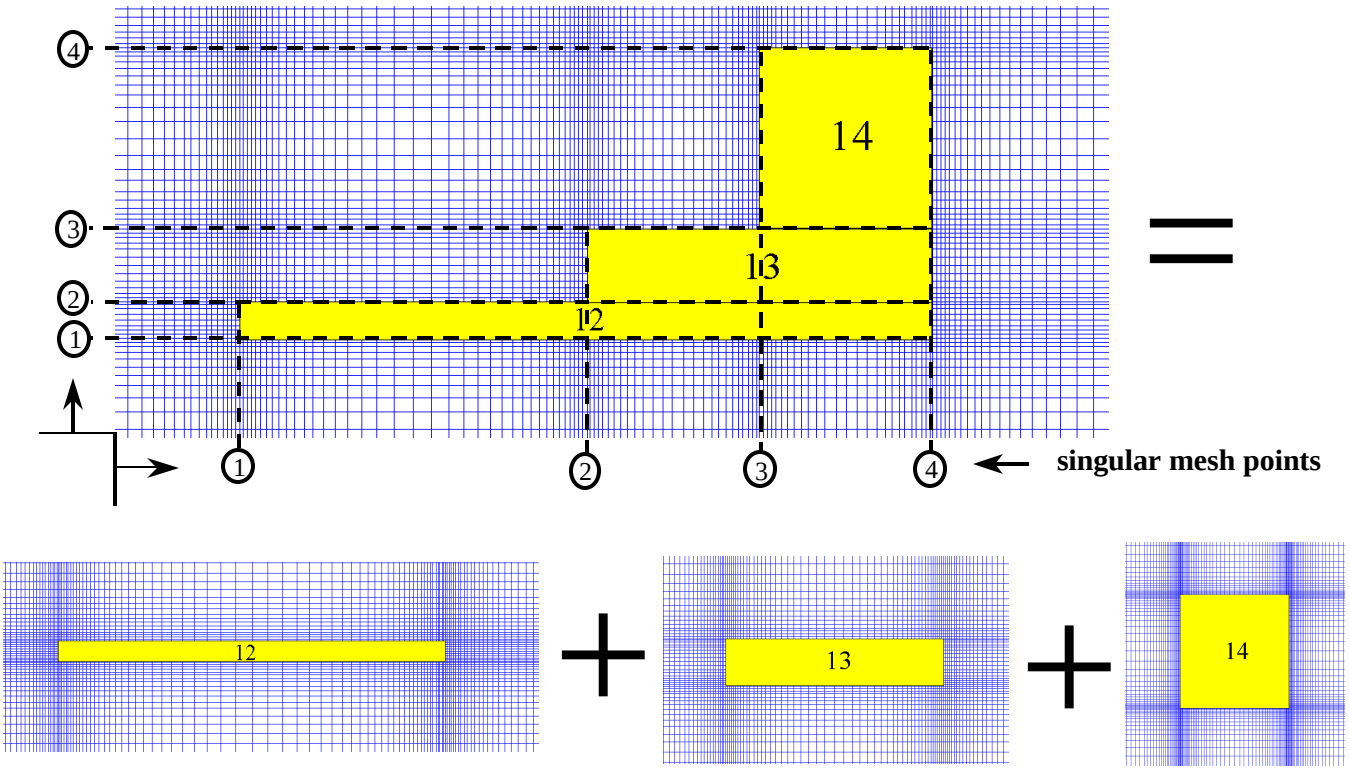


Figure 12. Illustration of the principle of superposition. The magnetic field for the multi-coil system (top) can be calculated as a superposition of the magnetic fields of each individual coil (bottom). Sample nonuniform meshes are shown for the 5FD2N method.

The full magnet field of the low-power VASIMR thruster is shown in Figure 13. Table 2 specifies the various magnet parameters.

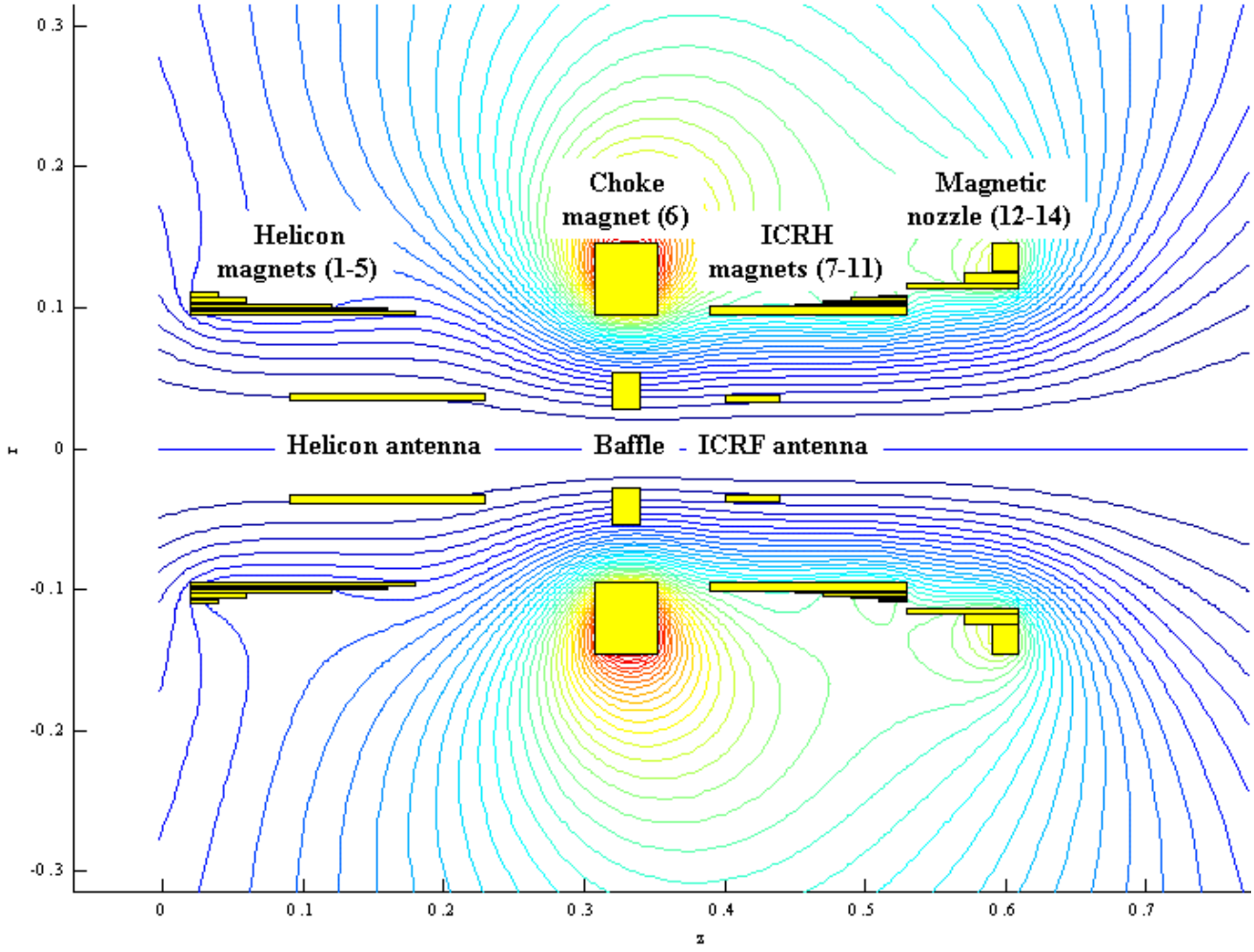


Figure 13. Magnetic field for the 14-coil magnetoplasma rocket configuration.

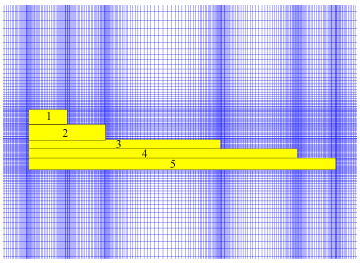
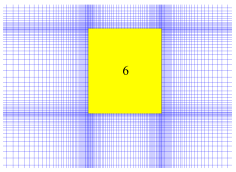
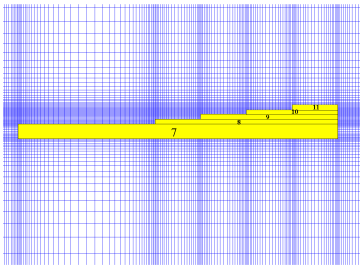
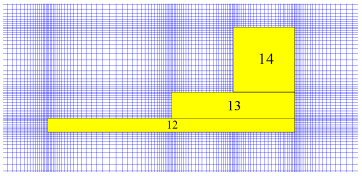
N	R_1	R_2	Z_1	Z_2	J	Section	Sample nonuniform mesh
1	0.1068	0.1108	0.02	0.04	2000.0	Helicon magnets	
2	0.1028	0.1068	0.02	0.06	4000.0		
3	0.1004	0.1028	0.02	0.12	6000.0		
4	0.098	0.1004	0.02	0.16	8000.0		
5	0.095	0.098	0.02	0.18	12000.0		
6	0.095	0.145	0.3075	0.3525	110000.0	Choke magnet	
7	0.095	0.101	0.39	0.53	42000.0	ICRH magnets	
8	0.101	0.103	0.45	0.53	8000.0		
9	0.103	0.105	0.47	0.53	6000.0		
10	0.105	0.107	0.49	0.53	4000.0		
11	0.107	0.109	0.51	0.53	2000.0		
12	0.113	0.117	0.53	0.61	16000.0	Magnetic nozzle	
13	0.117	0.125	0.57	0.61	16000.0		
14	0.125	0.145	0.59	0.61	20000.0		

Table 2. Description of the multi-coil magnetic configuration considered for the RTD VASIMR thruster shown in Figure 13. Here J is the total current in each coil. The current density function ρ in each coil is computed as $\rho = J / (R_2 - R_1)(Z_2 - Z_1)$.

The uniform and nonuniform meshes used are demonstrated in Figure 14. For the 9FD4U method, the computational domain was chosen as the union of two rectangles: $0 < r < 0.075$, $-0.1 < z < 0.7$ and $0 < r < 1$, $0.7 < z < 3$. Such domain includes the area where plasma flow is observed. Since "difficult" coils require too-large uniform meshes, the uniform-mesh domain excludes the coils. A piece-wise uniform grid with 16×72 points is used for the first subdomain and the grid with 240×64 points is used for the second one. The 5FD5N method used nonuniform mesh with 100×300 grid points on the domain $[0\ 5] \times [-5, 5]$.

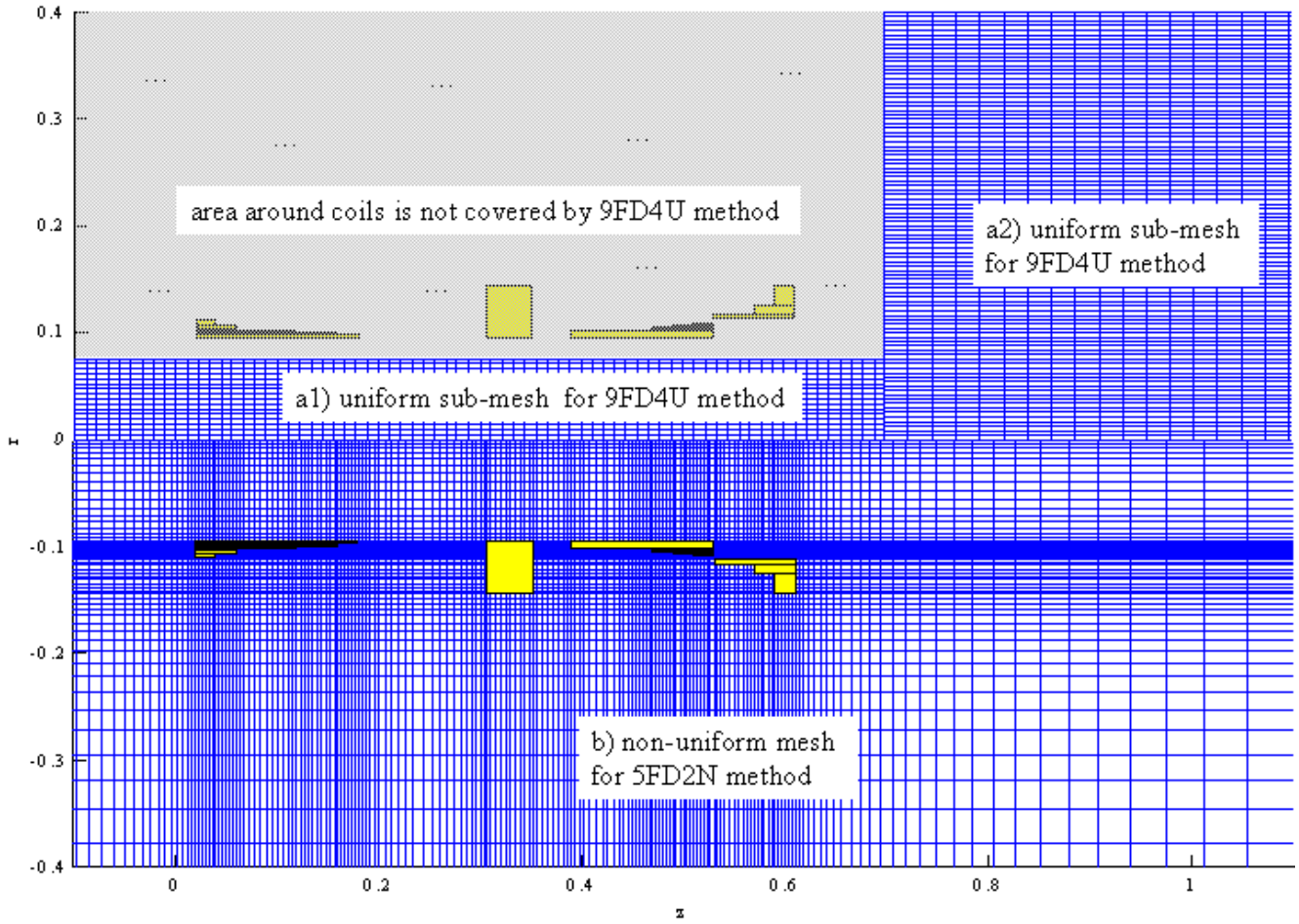


Figure 14. a) Sample uniform meshes used for the 9FD4U method (top half of the picture); area around coils is not covered by the 9FD4U method; b) sample nonuniform mesh used for the 5FD2N method (bottom half of the picture).

The accuracy of both methods is demonstrated in Figure 15. It has been found that there is about the same level of the numerical error for both methods, which is below the current experiment requirements of 10^{-3}. The numerical error for the 5FD2N method with nonuniform mesh is closer to a constant than the error for the 9FD4U method with uniform mesh. The uniform mesh error is higher at areas corresponding to a larger magnetic field. Also, the uniform mesh error has a discontinuity at the interface boundary of sub-meshes.

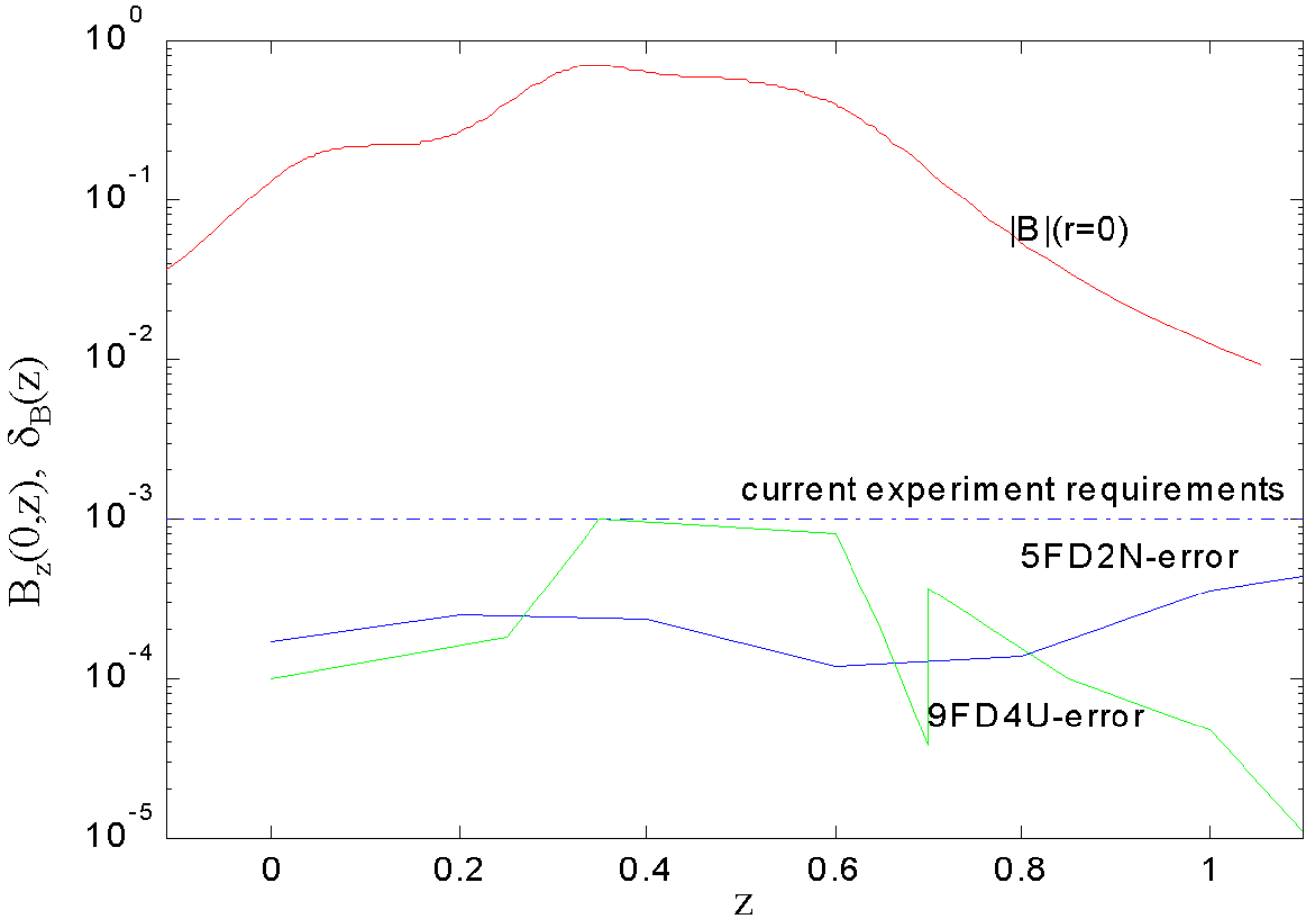


Figure 15. Relative error of the magnetic field for the set of 14 coils calculated on the symmetry line by 5FD2N and 9FD4U methods.

7. Computational complexity

The computational complexity involved in the solution of the problem generally implies requirements on computer memory and speed. For some applications these resources may be limited. Hence the proper choice of computational method is determined in the context of the full engineering problem. To illustrate this point, we examine the tradeoffs in complexity as functions of the computational error for the simple coil shown in Figures 8 and 10.

First, let us discuss computer memory requirements. One of the most efficient available iterative solvers, the iterative implicit incomplete factorization method (IMIF9, [14]), requires the use of a memory volume $P \approx T_{AM}\, N_r\, N_z$, where T_{AM} is the number of 2D arrays used by the approximation method. The size of arrays is equal to the number of mesh points of $N_r\, N_z$. The 9FD4U method uses $T_{AM} = 12$ arrays, and the 5FD2N method uses $T_{AM} = 9$ arrays. Therefore, using double precision, a conventional personal computer with 128 MB of RAM should be able to handle the problem as long as the mesh does not exceed 1000×1000 points.

The requirements on computer speed are a bit more complicated to evaluate, as they depend on the number of iterations in addition to the mesh size. An approximate formula for the number of arithmetic operations Q, required by a numerical method to get a solution with given relative error δ, is given by $Q \approx U_{AT}\, n(\varepsilon_i)\, N_r(\delta)\, N_z(\delta)$. U_{AT} is a number of arithmetic operations used by a method, per one iteration, per one mesh point. 9FD4U method implies $U_{AT} = 45$ and 5FD2N method implies $U_{AT} = 21$. $n(\varepsilon_i)$ is the number of iterations required to get numerical solution with an error less than ε_i. The iteration error ε_i has to be much less than the relative numerical solution error δ. One can use the following simple relation: $\varepsilon_i = \delta^2$. As numerical experiments show, the number of required iterations $n(\varepsilon_i)$ is between *20* and *40* for the iteration error $\varepsilon_i = 10^{-6}$.

The mesh size for the fourth-order 9FD4U method is bounded by a numerical error as $N_r < O(\delta^{-1/4})$, $N_z < O(\delta^{-1/4})$. For the second-order 5FD2N method, the mesh size is defined by the relation $N_r < O(\delta^{-1/2})$, $N_z < O(\delta^{-1/2})$. All this makes the $Q(\delta)$ dependence to be quite complicated. Rather than attempting to evaluate this expression, a set of numerical experiments has been performed for a simple coil (c1). The results are plotted on Figure 16. The number of arithmetic operations is measured in million float-point operations (MFlops).

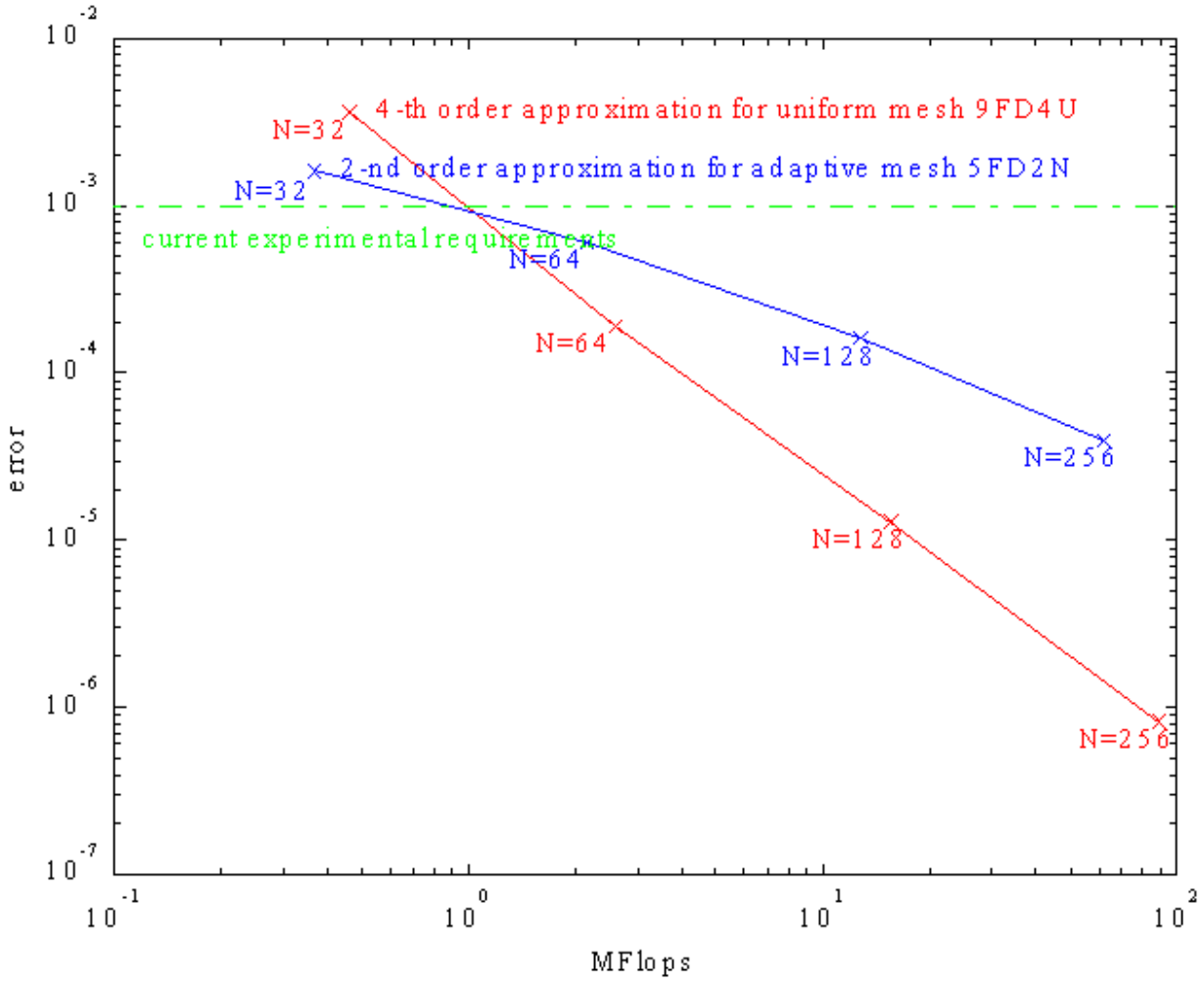


Figure 16. Numerical error δ vs. number of arithmetic operation Q needed for magnetic field calculation in c1-domain.

From this figure, if one is interested in high accuracy, the 9FD4U method is faster, whereas if accuracy is less important, the 5FD2N method is more desirable. As pointed out earlier, these results are dependent on the complexity of the coil structure. For a relative error close to the current experiment requirement of $\delta = 0.001$, the 9FD4U and 5FD2N require about the same number of arithmetic operations for the "easy" c1-coil problem considered here.

However, the second-order scheme using nonuniform mesh is a better tool for calculating a magnetic field for more complex problems. One such complex problem is a multi-coil system including "difficult" coils, when the magnetic field has to be calculated next to the coils. Another complex magnetostatic problem is solving for the magnetic field in the presence of internal plasma currents, where no semi-analytical solution for the calculation of boundary values is available.

The total computational cost of calculating magnetic field must include the cost of computing the boundary conditions, when they are calculated by the numerical integration of formula (3). For simple coil geometries, the numerical calculation of the boundary magnetic field requires many fewer computations and gives much less error than finite-difference solvers. This is why the computational complexity of boundary condition calculation does not affect the overall complexity of the magnetic field calculation.

8. Conclusion

This paper demonstrates advanced computational techniques for an accurate computation of the magnetic field of an axisymmetric electromagnet. These techniques involve the adaptive and uniform grids, second- and fourth-order approximations, and use of a semi-analytical calculation of the boundary conditions. The comparison analysis has been conducted for two approximation methods: the fourth-order 9-point finite volume scheme using uniform mesh (9FD4U) and 5-point finite difference scheme on adaptive nonuniform mesh (5FD2N). The approximation method of choice depends on the complexity of the current sources (coils), specification of the computational domain for finding the magnetic field, computational accuracy requirements, and available computational resources.

For achieving the accuracy imposed by our experimental requirements, both 9FD4U and 5FD2N methods have about the same computational complexity. If higher accuracy is needed, the fourth-order scheme 9FD4U will have dramatic advantage over the second-order method 5FD2N. The 5FD2N scheme is the best tool when the computational domain has to be of the large size or there is no semi-analitical method for boundary values calculation. Such a case exists when the presence of internal plasma currents affects the overall magnetic field. The numerical methods described in this paper have been applied for solving an electrostatic problem, which is a part of the extended mathematical model in the magnetoplasma rocket simulation [2].

Numerical predictions of the field generated by an axisymmetric electromagnet are also being applied to experimental investigations of the VASIMR system in the laboratory. The magnetic field contours shown in Figure 17 are outputs of this predictive tool. Actual photographs of the plasma show plume divergence as predicted by the code.

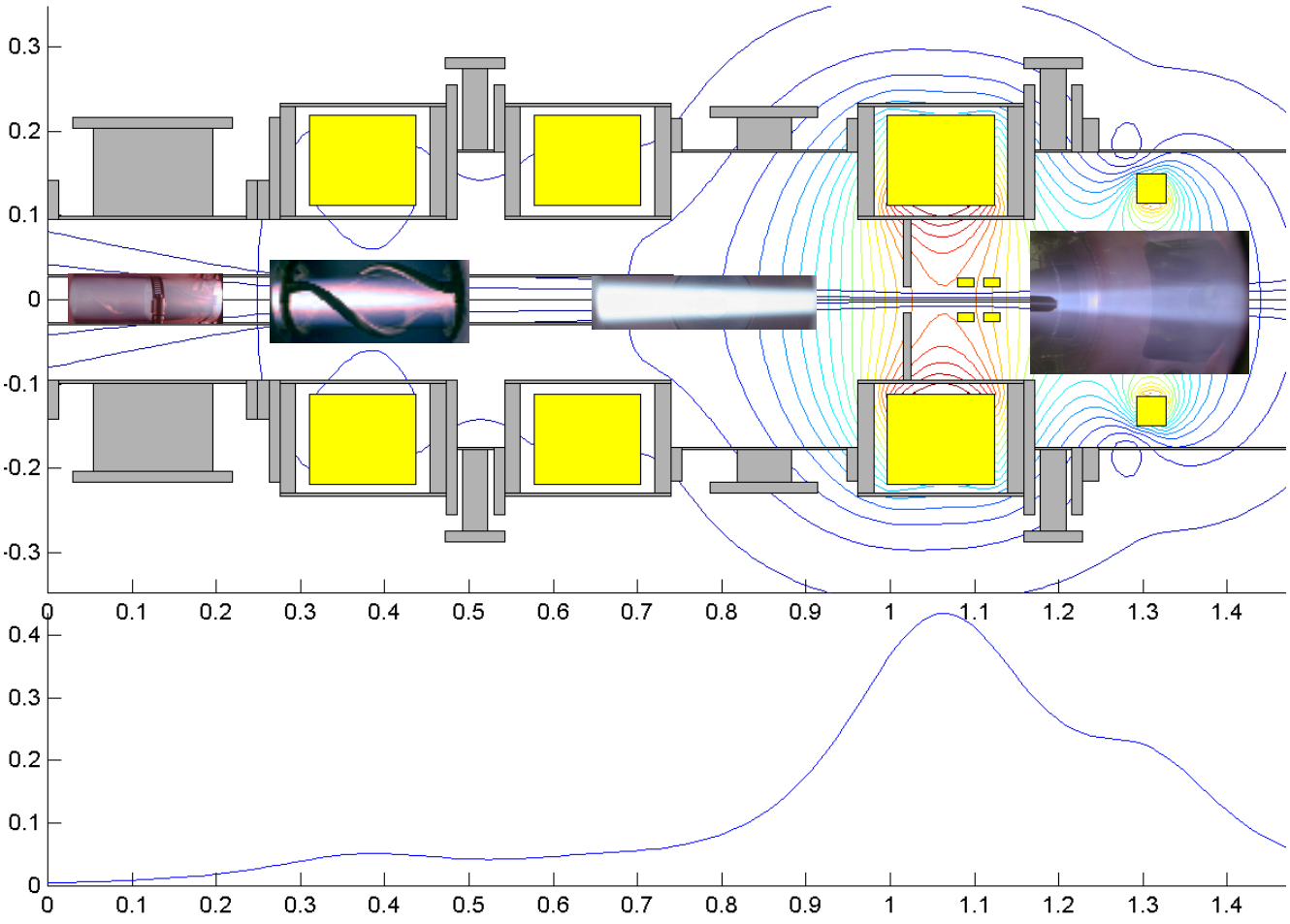


Figure 17. Magnetic field in the present VASIMR configuration calculated by the numerical code together with superimposed photographs of plasma source and plasma exhaust. Magnetic field lines are plotted for the area with a plasma flow, as well as the magnetic field strength contour lines. The bottom part of the figure demonstrates a 1D plot of the axial magnetic field along the symmetry line.

The 5FD2N method is currently used operationally to design an experimental configuration of the VX-10 device at the ASPL. For given electromagnet geometries and current values, the magnetic field is calculated and plotted (Figure 17), as well as contour lines for the magnetic field strength. The present code allows the determination of the electromagnet currents needed to generate a required magnetic field. Another important application of the magnetostatic solver based on the 5FD2N method is a self-consistent simulation of plasma flow in the VASIMR.

References

[1] Ilin A. V., Chang Díaz F. R., Squire, J. P., Breizman, B. N., Carter, M. D. Particle Simulation of Plasma Heating in VASIMR, *Proceedings of 36th AIAA/ASME/SAE/ASEE Joint Propulsion Conference (Huntsville, AL, Jul. 2000), AIAA 2000-3753* (2000).

[2] Il'in V. P. Numerical methods in electrophysics (in Russian), Moscow, Nauka (1985).

[3] Glasko V. B., Olhovskaya N. I. On the synthesis problem in NMR-tomography (in Russian), *J. Num. Math and Math. Phys.*, **29**, 1036 -1044 (1989).

[4] Chang Díaz F. R., Squire J. P., Bengtson R. D., Breizman B. N., Baity F. W., Carter M. D. Physics and Engineering of the VASIMR Engine, *Proceedings of 36th AIAA/ASME/SAE/ASEE Joint Propulsion Conference (Huntsville, AL, Jul. 2000), AIAA 2000-3756* (2000).

[5] Current and Voltage Transducers for Industrial Applications. Industrial Transducers Catalogue. LEM Corporate Communications (1996).

[6] Gaussmeters, Probes & Accessories. Industrial Catalogue. F. W. Bell, Inc. (1996).

[7] Forsythe G., Wasow W. Finite-difference methods for partial differential equations. John Wiley & Sons, NY (1959).

[8] Il'in V. P. High accuracy balance approximations for Poisson equation (in Russian.), *Siberian Mathematical Journal*, Novosibirsk, **37**, 151-169 (1996).

[9] Gurieva Y. L., Il'in V. P. Finite volume approaches for 2-D BVPs: algorithms, data structures, software and experiments. *Report No. 9715*, University of Nijmegen, The Netherlands (1997).

[10] Chang Díaz F. R., Braden E., Johnson I., Hsu M. M., Yang T. F. Rapid Mars Transits With Exhaust-Modulated Plasma Propulsion, *NASA Technical Paper 3539*, 10 pp. (1995).

[11] Chang Díaz F. R. The VASIMR Rocket, *Scientific American*, No. 5, **283**, 72–79 (2000).

[12] Lifshitz E. M., Pitaevskii L. P., Landau L. D. Electrodynamics of Continuous Media, Elsevier Science (1985).

[13] Gradshteyn I. S., Ryzhik I. M. Table of Integrals, Series, and Products, Academic Press (1994).

[14] Il'in V. P. Incomplete factorization methods. World. Sci. Pub. Co., Singapore (1992).

[15] Ilin A., Bagheri B., Scott L. R., Briggs J. M., and McCammon J. A. Parallelization of Poisson-Boltzmann and Brownian Dynamics calculation. In *Parallel Computing in Computational Chemistry*, No. 592, Washington D.C., ACS Books, 170-185 (1995).

[16] Szabo B., Babuška I. Finite Element Analysis, John Wiley & Sons, Inc. (1991).

[17] Hildebrand F. B. Introduction to numerical analysis. Dover Publ., Inc., NY (1987).

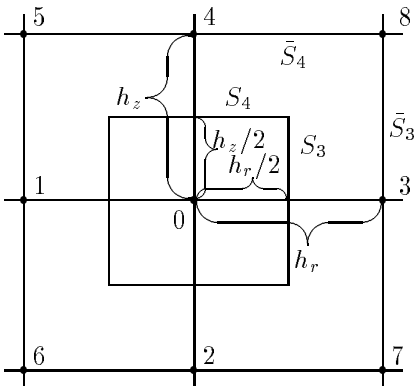


Figure 18: Local numbering used in the compact 9-point finite difference scheme (17) and two boxes around mesh node (i,j), used in finite volume discretization (21).

Appendix 1: Compact finite difference scheme of the fourth order

Our goal is to derive the compact finite difference approximation of equation (2) of the fourth order at the uniform grid with mesh steps $h_i^r = h_r$, $h_j^z = h_z$ (here "compact" means that only the closest to the central point mesh points are included into the stencil for approximation). To get this, we define the scaled one-dimensional (1D) differential operator

$$\tilde{L}_r u = \frac{1}{r} L_r u = -\frac{\partial}{\partial r}\frac{1}{r}\frac{\partial u}{\partial r} \tag{13}$$

and its three-point approximation with the following property:

$$(\tilde{L}_r^h u)_{i,j} \equiv \frac{1}{h_r^2}\left(\frac{u_{i,j}-u_{i-1,j}}{r_{i-1/2}} + \frac{u_{i,j}-u_{i+1,j}}{r_{i+1/2}}\right) = (\tilde{L}_r u - \frac{h_r^2}{12}\tilde{L}_r r \tilde{L}_r u)_{i,j} + O(h^4), \tag{14}$$

where $r_{i\pm1/2} = (r_i + r_{i\pm1})/2$. The validity of this relation can be checked by Taylor expansion.

From (14) and similar definition of another 1D difference operator

$$(L_z^h u)_{i,j} \equiv \frac{1}{h_z^2}(2u_{i,j} - u_{i,j-1} - u_{i,j+1}) = (L_z u - \frac{h_z^2}{12} L_z^2 u)_{i,j} + O(h^4) \tag{15}$$

we have the following nine-point scheme of order $O(h^4)$:

$$\begin{aligned}(L_9^h u)_{i,j} &\equiv \left[r\tilde{L}_r^h u + L_z^h u - \frac{r}{12}(h_r^2 \tilde{L}_r^h L_z^h u + h_z^2 L_z^h \tilde{L}_r^h u)\right]_{i,j} \\ &= f_{i,j} - \frac{1}{12}(r h_r^2 \tilde{L}_r^h f + h_z^2 L_z^h f)_{i,j} + O(h^4).\end{aligned} \tag{16}$$

The validity of this relation is proved by the direct substitution of the expansions (14), (15) in (16) and using the original equation (2) for the terms of the second order. The function f is required to have bounded fourth derivatives.

Let us introduce the local numbers for the nodes of the nine-point mesh stencil in accordance with Figure 18. Then we can write the resulting grid equation of the mesh point (i,j) in the form

$$p0_{i,j} u_0 - \sum_{k=1}^{8} pk_{i,j} u_k = f0_{i,j}, \tag{17}$$

where the coefficients and right-hand side are given for $r_i > 0$ by the following formulas derived by multiplication of 1D finite-difference operators:

$$\begin{aligned}
p0_{i,j} &= \sum_{k=1}^{8} pk_{i,j}, \\
p3_{i,j} &= \frac{1}{6h_r^2 r_{i+1/2}}(5 - \frac{h_r^2}{h_z^2}) = p1_{i+1,j}, \\
p2_{i,j} &= \frac{2r_i}{3h_z^2(4r_i^2 - h_r^2)}(5 - \frac{h_z^2}{h_r^2} - 3\frac{h_r^2}{2r_i^2}) = p4_{i,j-1}, \\
p7_{i,j} &= \frac{1}{12r_{i+1/2}}(\frac{1}{h_r^2} + \frac{1}{h_z^2}) = p8_{i,j-1} = p5_{i+1,j-1} = p6_{i+1,j}, \\
f0_{i,j} &= f_{i,j} - \frac{1}{12}[\frac{r_i}{r_{i-1/2}}(f_{i,j} - f_{i-1,j}) + \frac{r_i}{r_{i+1/2}}(f_{i,j} - f_{i+1,j}) + 2f_{i,j} - f_{i,j-1} - f_{i,j+1}].
\end{aligned} \tag{18}$$

All the coefficients (18) are positive under the conditions

$$2/7 < h_r^2/h_z^2 < 5. \tag{19}$$

If we have the Dirichlet condition on all of the boundary $\partial\Omega$ of the computational domain $u|_{\partial\Omega} = g(r,z)$ (the simplest case is $g = 0$), then an algebraic system of linear equations (17), (18) can be rewritten in vector-matrix form

$$A_h u_h = f_h, \quad u_h = \{u_{i,j}\}, \quad f_h = \{f0_{i,j}\}, \tag{20}$$

where A_h is a symmetric and positive definite matrix for an arbitrary ratio of the mesh steps. A_h is a Stieltjes type matrix, i.e. it is monotone ($A_h^{-1} > 0$, see [14] for example) only under the conditions (19). The convergence rate $O(h^4)$ of the error of this scheme can be proved in the Euclidean or maximum norms by usual techniques, like it was done in [7, 8].

The generalization of the high-order discretization using finite volume approach is described in Appendix 2.

Appendix 2: Finite volume approximations

Using direct finite-difference approximations, it is impossible theoretically to get high-order accuracy if the functions u and f don't have smooth enough derivatives.

So, our second approach consists of constructing the finite volume scheme for the piece-wise smooth function f. The approach is based on the approximation of the integral equation obtained by formal integration of the differential equation (2) and coincides with the fourth-order equation (16) in the case of a uniform grid and a smooth function f.

A similar finite-volume scheme for the Poisson equation in Cartesian coordinates can be found in [8, 9]. Such schemes are not investigated thoroughly for partial differential equations in cylindrical coordinates.

In accordance with [9], we introduce the small and big boxes around the (i,j) mesh-point:

$$\begin{aligned}
V_{i,j} &= \left\{ r_i - h_{i-1}/2 \le r \le r_i + h_i/2, \quad z_j - h_{j-1}/2 \le z \le z_j + h_j/2 \right\}, \\
\overline{V}_{i,j} &= \left\{ r_{i-1} \le r \le r_{i+1}, \quad z_{j-1} \le z \le z_{j+1} \right\}
\end{aligned} \tag{21}$$

and denote their boundaries by $S_{i,j}$ and $\overline{S}_{i,j}$ respectively, as shown in Figure 18.

Let us integrate by parts the left side of equation (2) over the volume

$$\int\limits_{V_{i,j}} \frac{1}{r}(L_r u + L_z u)dV = \int\limits_{z_{j-1/2}}^{z_{j+1/2}} \frac{1}{r}\frac{\partial u}{\partial r}|_{r_{i-1/2}}^{r_{i+1/2}} dz + \int\limits_{r_{i-1/2}}^{r_{i+1/2}} \frac{1}{r}\frac{\partial u}{\partial z}|_{z_{j-1/2}}^{z_{j+1/2}} dr$$

and consider the linear combination with a weight parameter $0 \le w \le 1$ of the integral relations for the small and big boxes

$$I_{i,j} \equiv w \int\limits_{S_{i,j}} I^n dS + (1-w) \int\limits_{\overline{S}_{i,j}} I^n dS = w \int\limits_{V_{i,j}} \frac{f}{r} dr dz + (1-w) \int\limits_{\overline{V}_{i,j}} \frac{f}{r} dr dz. \tag{22}$$

Here I^n is the projection of the vector $I = (I_r, I_z) = (-\frac{1}{r}\frac{\partial u}{\partial r}, -\frac{1}{r}\frac{\partial u}{\partial z})$. We assume also that some jumps of function f are possible only at the grid lines $r = r_i,\ z = z_j$.

To approximate equation (22), we introduce sub-volumes $V_{i,j}^l,\ \overline{V}_{i,j}^l,\ l = 1,2,3,4$ which are the result of subdivision of $V_{i,j}$ and $\overline{V}_{i,j}$ by the grid lines into four parts:

$$V_{i,j} = \bigcup_{l=1}^{4} V_{i,j}^l, \quad \overline{V}_{i,j} = \bigcup_{l=1}^{4} \overline{V}_{i,j}^l.$$

We define corresponding fluxes $I_{i,j}^l$ through the sub-volume surfaces $S_{i,j}^l$ and $\overline{S}_{i,j}^l$:

$$I_{i,j} = \bigcup_{l=1}^{4} I_{i,j}^l \tag{23}$$

and write with the help of (22) the necessary relation for $l = 3$ in more detail (right upper sub-volumes for $r > r_i,\ z > z_j$):

$$-I_{i,j}^{(3)} = w \left(\int\limits_{z_j}^{z_{j+1/2}} \frac{1}{r}\frac{\partial u}{\partial r}|_{i+1/2} dz + \int\limits_{r_i}^{r_{i+1/2}} \frac{\partial u}{\partial z}|_{j+1/2} \frac{dr}{r} \right)$$
$$+(1-w) \left[\int\limits_{z_j}^{z_{j+1}} \left(\frac{1}{r}\frac{\partial u}{\partial r}|_i + \frac{1}{r}\frac{\partial u}{\partial r}|_{i+1} \right) dz + \int\limits_{r_i}^{r_{i+1}} \left(\frac{\partial u}{\partial z}|_j + \frac{\partial u}{\partial z}|_{j+1} \right) \frac{dr}{r} \right]. \tag{24}$$

Additional terms with the integrals over the "internal" boundaries between sub-volumes are included here to improve the approximation of the derivatives. It is important that in the sum (23), additional terms annul because of the continuity property of the solution and its fluxes.

Then by applying the simplest quadrature formulas and the linear interpolation of the terms under the integrals, we can write the flux $I_{i,j}^{(3)}$ as some linear form of four values u_l:

$$I_{i,j}^{(3)} = p00_{i,j} u_0 + p03_{i,j} u_3 + p04_{i,j} u_4 + p08_{i,j} u_8. \tag{25}$$

If the fourth-order vector $\overline{u}^{(3)} = (u_0, u_3, u_4, u_8)$, whose components are the values of the solution in the vertices of the finite volume $V_{i,j}^{(3)}$, and vector $\overline{I}^{(3)} = (I_0^{(3)}, I_3^{(3)}, I_4^{(3)}, I_8^{(3)})$, whose entries are the

"partial" fluxes around corresponding nodes (for example, $I_0^{(3)} = I_{i,j}^{(3)}$ from (24)), are introduced, then one can write in vector form the following relation between them:

$$\overline{I}^{(3)} = A^{(3)}\overline{w}^{(3)}, \quad A^{(3)} = \begin{pmatrix} p00_{i,j} & p03_{i,j} & p04_{i,j} & p08_{i,j} \\ p30_{i,j} & p33_{i,j} & p34_{i,j} & p38_{i,j} \\ p40_{i,j} & p43_{i,j} & p44_{i,j} & p48_{i,j} \\ p80_{i,j} & p83_{i,j} & p84_{i,j} & p88_{i,j} \end{pmatrix}. \tag{26}$$

Here the fourth-order symmetric matrix $A^{(3)}$ ($pkl_{i,j} = plk_{i,j}$) can be called the local balance matrix by analogy with the local stiffness matrix of finite element methods, introduced in [16].

It is easy to show that the elements $pk_{i,j}$ of the "global" balance matrix A_h from system (17), (20) can be written via the entries of local balance matrices by the formulas

$$\begin{aligned} &p0_{i,j} = p00_{i,j} + p33_{i-1,j} + p44_{i,j-1} + p88_{i-1,j-1}, \quad p3_{i,j} = p1_{i,j} = p03_{i,j} + p08_{i,j-1}, \\ &p4_{i,j} = p2_{i,j} = p04_{i,j} + p38_{i-1,j}, \qquad\qquad p8_{i,j} = p7_{i,j} = p5_{i,j} = p6_{i,j} = p08_{i,j}. \end{aligned} \tag{27}$$

In practice, instead of implementing these formulas, it is preferable to assemble the global matrix A_h in an element-by-element technique which can be presented by the following pseudo-code for each sub-volume $V_{i,j}^l$:

$$\begin{aligned} &p0_{i,j} := p0_{i,j} + p00_{i,j}, && p0_{i+1,j} := p0_{i+1,j} + p33_{i,j}, \\ &p0_{i,j+1} := p0_{i,j+1} + p44_{i,j}, && p0_{i+1,j+1} := p0_{i+1,j+1} + p88_{i,j}, \\ &p3_{i,j} := p3_{i,j} + p03_{i,j}, && p3_{i,j+1} := p3_{i,j+1} + p48_{i,j}, \\ &p4_{i,j} := p4_{i,j} + p04_{i,j}, && p4_{i+1,j} := p4_{i+1,j} + p38_{i,j}, \\ &p8_{i,j} := p8_{i,j} + p08_{i,j}, && p7_{i,j+1} := p7_{i,j+1} + p34_{i,j}. \end{aligned} \tag{28}$$

It is possible to use different approximations of the integrals in (24). We apply quadratures which follow from the linear interpolation of the functions under the integral over the surface of $\overline{V}_{i,j}^3$:

$$\begin{aligned} -I_{i,j}^3 = w \Bigg\{ &\frac{3h_j^z}{8h_i^r r_{i+1/2}}[3(u_{i,j} - u_{i+1,j}) + u_{i,j+1} - u_{i+1,j+1}] \\ &+ \frac{1}{16h_j^z}\left[\left(\frac{h_i^r}{r_{i+1/2}}\right)(u_{i+1,j} - u_{i+1,j+1}) + \left(7\gamma_{i+1/2} - \frac{h_i^r}{r_{i+1/2}}\right)(u_{i,j} - u_{i,j+1})\right]\Bigg\} \\ &+ (1-w)\left[\frac{h_j^z}{h_i^r r_{i+1/2}}(u_{i,j} - u_{i+1,j} + u_{i,j+1} - u_{i+1,j+1}) + \frac{\gamma_{i+1/2}}{h_j^z}(u_{i,j} - u_{i,j+1} + u_{i+1,j} - u_{i+1,j+1})\right]. \end{aligned} \tag{29}$$

Here the values $\gamma_{i+1/2}$ are defined from the condition that the entries $pk_{i,j}$ of the global matrix A_h, which are assembled via the elements $pkl_{i,j}$ of the local balance matrices by the formulas (25) – (28), coincide with the coefficients (18) of the compact finite difference scheme of order $O(h^4)$ in the case of uniform grid and $w = 16/15$. This condition has the recurrent form

$$\gamma_{i-1/2} = \frac{h_i^r + h_{i-1}^r}{r_i} - \gamma_{i+1/2}, \quad \gamma_{N_r+1/2} = \frac{h_{N_r+1}^r}{r_{N_r+1/2}}, \quad i = N_r, \ldots, 1 \tag{30}$$

and provides the following formulas for the entries of the local balance matrix:

$$\begin{aligned} p03_{i,j} = p48_{i,j} &= \frac{1}{r_{i+1/2}}\left[\left(1 - \frac{5w}{8}\right)\frac{h_j^z}{h_i^r} - \frac{w}{16}\frac{h_i^r}{h_j^z}\right] - \left(1 - \frac{15w}{16}\right)\frac{\gamma_{i+1/2}}{h_j^z}, \\ p04_{i,j} = p38_{i,j} &= \left(1 - \frac{9w}{16}\right)\frac{\gamma_{i+1/2}}{h_j^z} - \frac{1}{r_{i+1/2}}\left[\frac{w}{16}\frac{h_i^r}{h_j^z} + \left(1 - \frac{7w}{8}\right)\frac{h_j^z}{h_i^r}\right], \\ p08_{i,j} = p34_{i,j} &= \frac{1}{r_{i+1/2}}\left[\left(1 - \frac{7w}{8}\right)\frac{h_j^z}{h_i^r} + \frac{w}{16}\frac{h_i^r}{h_j^z}\right] + \left(1 - \frac{15w}{16}\right)\frac{\gamma_{i+1/2}}{h_j^z}, \\ p00_{i,j} &= p03_{i,j} + p04_{i,j} + p08_{i,j} = p33_{i,j} = p44_{i,j} = p88_{i,j}, \quad pkl_{i,j} = plk_{i,j}. \end{aligned}$$

The approximation of the right-hand side of equation (22) can be made in a similar cell-by-cell approach, so the value $f0_{i,j}$ from (20) is assembled as

$$f0_{i,j} = f^{(3)}_{o,i,j} + f^{(3)}_{3,i-1,j} + f^{(3)}_{4,i,j-1} + f^{(3)}_{8,i-1,j-1}. \tag{31}$$

Here the following "local" terms are defined from the numerical integration on the sub-volume $V^{(3)}_{i,j}$:

$$f^{(3)}_{0,i,j} = \left[(1-\frac{3w}{4})\rho_{i,j} - \frac{8-7w}{16}(\frac{r_i\rho_{i,j} - r_{i+1}\rho_{i+1,j}}{r_{i+1/2}} + \rho_{i,j} - \rho_{i,j+1})\right] h^r_i h^z_j, \tag{32}$$

$$f^{(3)}_{3,i,j} = \left[(1-\frac{3w}{4})\rho_{i+1,j} - \frac{8-7w}{16}(\frac{r_{i+1}\rho_{i+1} - r_i\rho_{i,j}}{r_{i+1/2}} + \rho_{i+1,j} - \rho_{i+1,j+1})\right] h^r_i h^z_j, \tag{33}$$

$$f^{(3)}_{4,i,j} = \left[(1-\frac{3w}{4})\rho_{i,j+1} - \frac{8-7w}{16}(\frac{r_{i+1}\rho_{i,j+1} - r_{i+1}\rho_{i+1,j+1}}{r_{i+1/2}} + \rho_{i,j+1} - \rho_{i,j})\right] h^r_i h^z_j, \tag{34}$$

$$f^{(3)}_{8,i,j} = \left[(1-\frac{3w}{4})\rho_{i+1,j+1} - \frac{8-7w}{16}(\frac{r_{i+1}\rho_{i+1,j+1} - r_i\rho_{i,j+1}}{r_{i+1/2}} + \rho_{i+1,j+1} - \rho_{i+1,j})\right] h^r_i h^z_j, \tag{35}$$

where $\rho_{i,j} = f(r_i, z_j)/r_i$ at the line of jump corresponds to "own" sub-volume. It is easy to check that formulas (31)–(35) provide for $w = \frac{16}{15}$ the term $f0_{i,j}$ from (18) in the case of a uniform grid. One remark: in fact coefficients in (18) in Appendix 1 differ by the multiplier $(1 - \frac{3}{4}w)$ $(h^r_i + h^r_{i-1})$ $(h^z_j + h^z_{j-1})/4$, which is the "weighted" cell volume.

These formulas provide the symmetry of the local balance matrix $A^{(3)}$ and global matrix A_h. The values $\gamma_{i\pm1/2}$ are positive for the quasi-uniform grid. For the general nonuniform grid, the approximation of this scheme is of order $O(h)$ only and the matrix A_h is monotone under strong enough conditions for the ratio of the mesh steps (19).

Appendix 3: Computation of boundary conditions

It is known [12] that the magnetic field of a single current ring for the unit magnetic permeability is given analytically by the following formulas for the axisymmetric azimuthal component of the vector potential $\boldsymbol{A}$ and r- and z-components of the induction vector $\boldsymbol{B}$:

$$A_\theta(r,z) = \frac{4J}{s}\sqrt{\frac{r'}{r}}\left[\left(1-\frac{s^2}{2}\right)K - E\right], \quad s^2 = \frac{4rr'}{(r+r')^2 + (z-z')^2}, \tag{36}$$

$$B_r(r,z) = \frac{2J}{r\sqrt{(r'+r)^2 + (z-z')^2}}\left[\frac{(r')^2 + r^2 + (z-z')^2}{(r'-r)^2 + (z-z')^2}E - K\right], \tag{37}$$

$$B_z(r,z) = \frac{2J}{\sqrt{(r'+r)^2 + (z-z')^2}}\left[\frac{(r')^2 - r^2 - (z-z')^2}{(r'-r)^2 + (z-z')^2}E + K\right]. \tag{38}$$

Here r' is the ring radius, J is the ring current, z' is the coordinate of its center in cylindrical coordinates r, z and K, E are full elliptic integrals of the first and second kind respectively (see [13]), for which the following series are valid:

$$\frac{1}{\pi}K = \frac{1}{2}\left[1 + \left(\frac{1}{2}\right)^2 s^2 + \left(\frac{1\cdot3}{2\cdot4}\right)^2 s^4 + \left(\frac{1\cdot3\cdot5}{2\cdot4\cdot6}\right)^2 s^6 + \ldots\right],$$

$$\frac{1}{\pi}E = \frac{1}{\pi}K - s^2\left[\frac{1}{1}\left(\frac{1}{2}\right)^2 + \frac{2}{3}\left(\frac{1\cdot3}{2\cdot4}\right)^2 s^2 + \frac{3}{5}\left(\frac{1\cdot3\cdot5}{2\cdot4\cdot6}\right)^2 s^4 + \frac{4}{7}\left(\frac{1\cdot3\cdot5\cdot7}{2\cdot4\cdot6\cdot8}\right)^2 s^6 + \ldots\right].$$

By the definitions of s, E and K, it is possible to show that $A(0,z) = B_r(0,z) = 0$.

The computation of the magnetic field generated by a coil with arbitrary cross section can be made by numerical integration of analytical formulas (37), (38) over the coil cross section

$$c = \{0 < R_1 < r' < R_2,\ Z_1 < z' < Z_2\}. \tag{39}$$

The following formula, which can be obtained from (36), is used to compute the boundary values of the solution:

$$u(r,z) = \rho \int_{R_1}^{R_2}\int_{Z_1}^{Z_2} \sqrt{(r+r')^2 + (z-z')^2}\, M(s^2)\, dr'dz', \quad M(s^2) = \frac{1}{\pi}(1 - \frac{s^2}{2})K - \frac{2}{\pi}E, \tag{40}$$

where ρ is a uniform current density and the coil cross section is the region $R_1 < r' < R_2$, $Z_1 < z' < Z_2$. For the underintegral term $M(s^2)$, the following series can be derived via the expansion for full elliptic integrals K, E:

$$M(s^2) = s^4 \left[\left(\frac{1}{2}\right)^2 \frac{1}{8} + \left(\frac{1\cdot 3}{2\cdot 4}\right)^2 \frac{2}{12} s^2 + \left(\frac{1\cdot 3\cdot 5}{2\cdot 4\cdot 6}\right)^2 \frac{3}{16} s^4 + \ldots \right] = \sum_{p=0}^{\infty} c_p s^{4+2p}. \tag{41}$$

Instead of $M(s^2)$ we use a calculation of its approximation for $s < 1$:

$$M_n(s^2) = \sum_{p=0}^{n} c_p s^{4+2p}. \tag{42}$$

To understand the convergence rate of $M_n(s^2)$ when $n \to \infty$, the values of

$$s^2 = \left(1 + \frac{(r-r')^2 + (z-z')^2}{4rr'}\right)^{-1}$$

can be estimated for (r,z) at the boundary of the computational domain ($r = R_3$ or $z = Z_2$) and (r',z') from the coil cross section (39) by the following inequality:

$$s^2 \le \max\left\{ \left(1 + \frac{(R_3 - R_2)^2}{4R_3R_2}\right)^{-1},\ \left(1 + \frac{(Z_3 - Z_1)^2}{4R_3R_2}\right)^{-1} \right\}.$$

For example, if $R_3 \ge 3R_2$ and $Z_3 - Z_1 \ge 2R_2$, then $s^2 \le 0.75$. So, if the domain boundary is away from the coil, then the value of s^2 decreases and the error of approximation of $M(s^2)$ by $M_n(s^2)$ for the fixed n is improved. More exactly, the cutoff error ϵ_n can be estimated by the following relation:

$$\epsilon_n = M(s^2) - M_n(s^2) = \sum_{p=n+1}^{\infty} c_p s^{4+2p} \le c_{n+1} s^{4+2(n+1)} \sum_{l=0}^{\infty} s^{2l} \le c_{n+1} s^{4+2(n+1)}/(1-s^2).$$

Our computational strategy is a mixed one and consists of numerical integration with respect to r' by the simplest central rectangle quadrature rule

$$u(r,z) = \rho \int_{R_1}^{R_2} \omega_n(r')dr' + \tilde{\epsilon_n} = \rho h_c \sum_{l=1}^{m} \omega_n(r'_l) + \tilde{\epsilon_n} + \epsilon_m^{(1)}, \quad r'_l = R_1 + (l - \frac{1}{2})h_c, \quad h_c = \frac{R_2 - R_1}{m}, \tag{43}$$

or by the Simpson's quadrature

$$u(r,z) = \rho\frac{h_c}{6}(\omega_n(r_1'') + 4\omega_n(r_2'') + 2\omega_n(r_3'') + \ldots + \omega_n(r_{2m+1}'')) + \tilde{\epsilon_n} + \epsilon_m^{(2)}, \quad r_l'' = R_1 + \frac{(l-1)}{2}h_c, \tag{44}$$

or by the Gauss quadrature

$$u(r,z) = \rho h_c\,(c_1\omega_n(r_1) + \ldots + c_m\omega_n(r_m)) + \tilde{\epsilon_n} + \epsilon_m^{(3)}, \tag{45}$$

where

$$\omega_n(r') = \int\limits_{Z_1}^{Z_2} \sqrt{(r+r')^2 + (z-z')^2}\,M_n(s^2)dz',$$

$$\tilde{\epsilon_n} = \rho \int\limits_{R_1}^{R_2}\int\limits_{Z_1}^{Z_2} \sqrt{(r+r')^2 + (z-z')^2}\,\epsilon_n dr'dz' \le \rho\sqrt{(R_3+R_2)^2 + (Z_3-Z_1)^2}(R_2-R_1)(Z_2-Z_1)\epsilon_n$$

with the given integer m. Here $\epsilon_m^{(i)},\ i = 1,\ 2,\ 3$ are the errors of different quadratures. The total error of such approximation is

$$\epsilon = \tilde{\epsilon_n} + \epsilon_m^{(i)} = O(h_c^{\gamma} + \overline{s}^{2n+6}), \quad i = 1,2, \quad \overline{s} = \max_{\substack{(r',z')\,\in\, c,\\ (r,z)\,\in\,\partial\Omega}} \{s(r,r',z,z')\},$$

where $\gamma = 2$ for quadrature (43) and $\gamma = 4$ for (44) and can be made as small as necessary with sufficiently large integers m and n. The optimal Gauss quadrature points r_k and the coefficients c_k in (45) are defined in [17], for example. The error $\epsilon_m^{(3)}$ of the Gauss quadrature is estimated by the inequality

$$\epsilon_m^{(3)} \le \frac{(R_2-R_1)^{2m+1}(m!)^4}{[(2m)!]^3(2m+1)} \max_{R_1<r'<R_2}\left\{\left|\frac{d^{2m}\omega_n(r')}{(dr')^{2m}}\right|\right\}.$$

As for the error ϵ_n, for $s^2 \le 0.75$, for example, we have $\epsilon_{23} \le 4.2\cdot 10^{-9},\ \epsilon_{33} \le 9.5\cdot 10^{-12}$.

To compute $\omega_n(r')$ we use the auxiliary recurrent relations for the following integrals:

$$\phi_p(x,t) = \int \frac{dx}{(\sqrt{x^2+t})^p}, \quad \phi_3(x,t) = \int \frac{dx}{(\sqrt{x^2+t})^3} = \frac{x}{t\sqrt{x^2+t}},$$

$$\phi_{2p+3}(x,t) = \frac{1}{t(2p+1)}\left(\frac{x}{(\sqrt{x^2+t})^{2p+1}} + 2p\phi_{2p+1}\right), \quad p = 1,2,\ldots \tag{46}$$

If we denote now

$$\overline{\phi}_p(r,r',z,Z_1,Z_2) = \phi_p(z-Z_2,(r+r')^2) - \phi_p(z-Z_1,(r+r')^2),$$

then it is possible to write the resulting formula to compute $\omega_n(r_l')$:

$$\omega_n(r_l') = \sum_{p=0}^{n} c_p(4r_l'r)^{p+2}\overline{\phi}_{2p+3}(r,r_l',z,Z_1,Z_2), \tag{47}$$

where coefficients c_p are defined in (41).

The stable computation of $\omega_n(r_l')$ is held by using the recurrent procedure. Namely, we define the new functions

$$\tilde{\phi}_{2p+3} = (4rr')^{p+3/2}\overline{\phi}_{2p+3}, \quad p = 0,1,\ldots,$$

which are computed from the relations

$$\tilde{\phi}_{2p+3} = \frac{1}{(2p+1)\beta}\left[\left(\frac{z-Z_2}{\left(\sqrt{\beta+\frac{(z-Z_2)^2}{4rr'}}\right)^{2p+1}} - \frac{z-Z_1}{\left(\sqrt{\beta+\frac{(z-Z_1)^2}{4rr'}}\right)^{2p+1}}\right)\sqrt{4rr'} + 2p\tilde{\phi}_{2p+1}\right], \beta = \frac{(r+r')^2}{4rr'}$$

and calculate $\omega_n(r'_l)$ by the formula

$$\omega_n(r'_l) = \omega_{n-1}(r'_l) + \tilde{\phi}_{2n+3}\frac{n}{8+4(n-1)}\prod_{m=1}^{n}\left(\frac{2m-1}{2m}\right)^2.$$

The presented formulas provide high accuracy of computing the boundary conditions for increasing integers $m,\ n$ if the value of s^2 is not very close to unity.

Appendix 4: 1D test

This section describes convergence of finite difference methods of the magnetostatic problem in a case when the solution depends only on one (radial) variable. Then we can present an algorithm of generation of 1D nonuniform meshes.

Consider a coil in the shape of the infinite solenoid with internal and external radii R_1 and R_2 correspondingly. Assume that the coil has an azimuthal electric current with unit density, and the coil is placed inside an infinite conducting cylinder with radius R_3. The magnetic field for such system is one dimensional, $\boldsymbol{B} = (0, 0, B_z(r))$ and has the following analytical expression:

$$B_z(r) = \begin{cases} R_2 - R_1 - 2d, & 0 \le r \le R_1, \\ R_2 - 2d - r, & R_1 \le r \le R_2, \\ -2d, \quad d = (R_2^3 - R_1^3)/(6R_3^2), & R_2 \le r \le R_3. \end{cases} \tag{48}$$

That solution is displayed in Figure 19.

Corresponding magnetic flux $u(r)$ has the following analytical formula

$$u(r) = \begin{cases} \left(\frac{R_2 - R_1}{2} - d\right) r^2, & 0 \le r \le R_1, \\ \left(\frac{R_2}{2} - d\right) r^2 - \frac{R_1^3}{6} - \frac{r^3}{3}, & R_1 \le r \le R_2, \\ d(R_3^2 - r^2), & R_2 \le r \le R_3, \end{cases} \tag{49}$$

which can be found as a solution to the following 1D problem

$$-\frac{\partial}{\partial r}\frac{1}{r}\frac{\partial u}{\partial r} = \rho, u(0) = u(R_3) = 0 \tag{50}$$

with a piece-wise uniform current density $\rho = 1$ for $R_1 \le r \le R_2$ and $\rho = 0$ otherwise. The coil radii are $R_1 = 0.4$, $R_2 = 0.6$, and domain size $R_3 = 1.0$ for the field shown in Figure 19.

When the 9FD4U method is used, the computations give zero error (in the sense of accuracy of the iterative method) both for the uniform and nonuniform grids. It was expected, that the error should be of order $O(h^2)$ because of the discontinuity of the right-hand side and nonuniform grid. In fact, the error is even better than $O(h^4)$.

The application of the 5-point scheme 5FD2N (6) for the boundary value problem (50) with the same 1D solution, but for different R_3, provides the following results presented in Figure 20. Here the

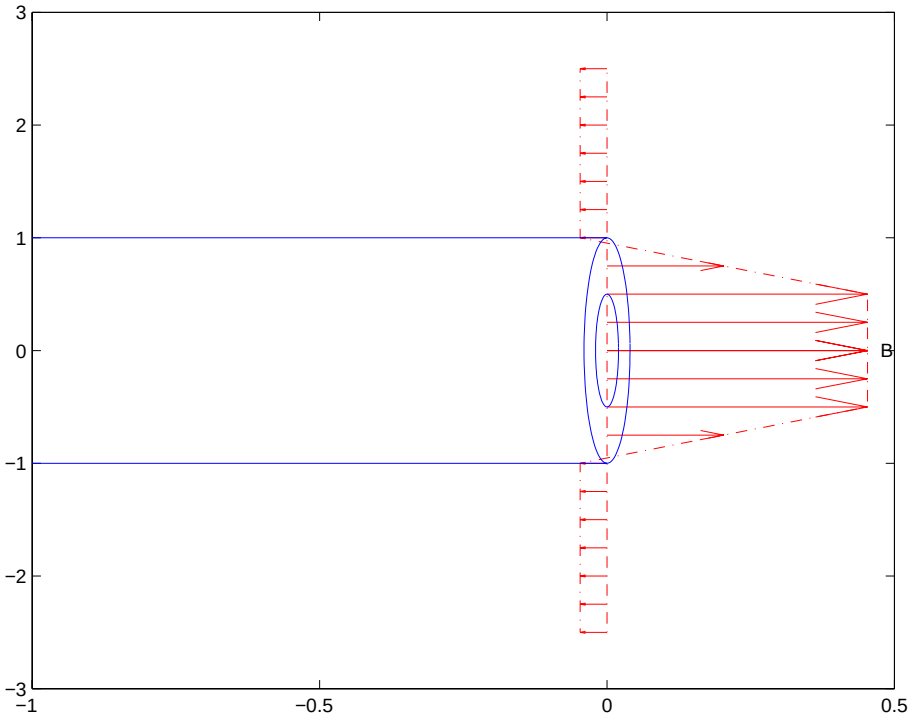


Figure 19: One-dimensional magnetic field for the infinite solenoid with internal radius $R_1 = 0.5$ and external radius $R_2 = 1.0$. Outside conducting wall is assumed to be placed at $R_3 = 2.5$.

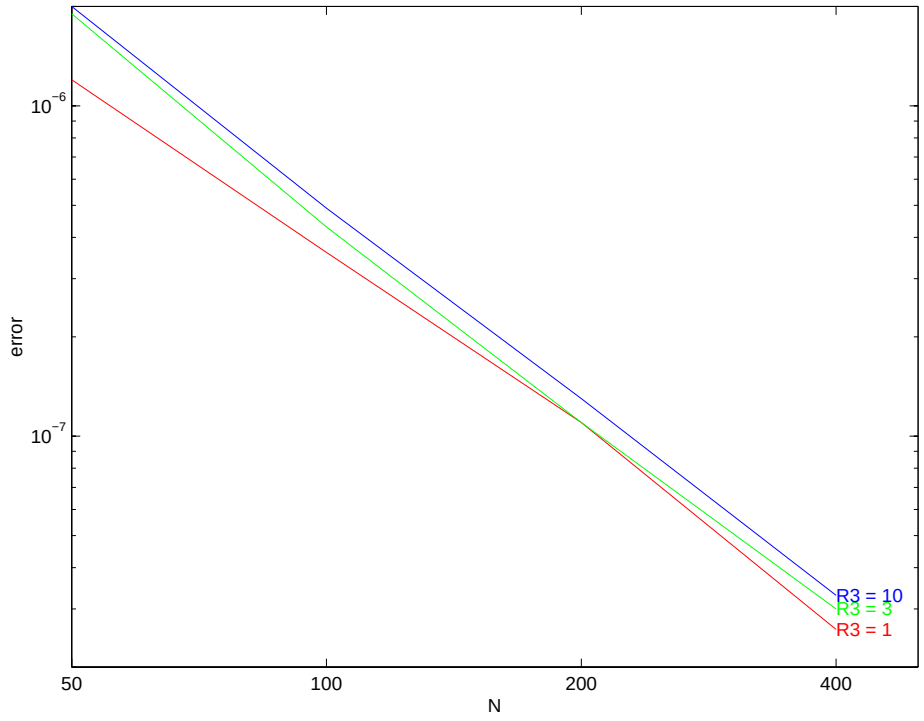


Figure 20: Relative error of the 1D calculation using 5-point scheme (6) on nonuniform mesh. $R_1 = 0.49, R_2 = 0.51, R_3 = 1,\ 3,\ 10$. For each N_r and R_3, the nonuniform mesh was built to minimize the relative error.

relative error is defined as

$$\delta = \frac{\max\limits_{0<r_i<R_3}\{|u(r_i) - u_h(r_i)|\}}{\max\limits_{0<r_i<R_3}\{|u(r_i)|\}}. \tag{51}$$

The plots show that the error is going down as fast as N_r^2, where N_r is the number of grid points. Also, error is less for smaller domain size R_3 due to smaller mesh size.

Let's describe the algorithm of generating an adaptive nonuniform mesh for the following 1D domain in radial direction. The computational domain $\Omega = [0, R_3]$ was divided into five subdomains: $\Omega_1 = [0, R_1/2], \Omega_2 = [R_1/2, R_1], \Omega_3 = [R_1, (R_1 + R_2)/2], \Omega_4 = [(R_1 + R_2)/2, R_2], \Omega_5 = [R_2, R_3]$. The grid was created to be fine next to singular points: $r = 0, r = R_1$ and $r = R_2$. In Ω_1, Ω_3 and Ω_5 the grid step size satisfies the following relation: $h_{i+1} = h_i(1 + \delta h)$, in Ω_2 and Ω_4: $h_{i+1} = h_i/(1 + \delta h)$, where $\delta h > 0$ is a relative increment of the grid-step. This means that in each subdomain Ω_k, the length $L(\Omega_k)$ can be expressed through a number of mesh steps N_k^r as following:

$$L(\Omega_k) = h_{min} + h_{min}(1 + \delta h) + h_{min}(1 + \delta h)^2 + ...h_{min}(1 + \delta h)^{N_k^r - 1} = h_{min}\left((1 + \delta h)^{N_k^r} - 1\right)/\delta h.$$

The 1D adaptive mesh generation has the following three steps.

1) For given N_r number of points and for given relative mesh increment δh, create nonuniform mesh in the domain Ω. This requires us to find 6 unknowns: the number of grid steps in subdomains $N_1^r, N_2^r, N_3^r, N_4^r, N_5^r$ and the minimal mesh size h_{min}. The unknowns must satisfy the following 6 equations:

$$h_{min}((1 + \delta h)^{N_k^r} - 1)/\delta h = L(\Omega_k), k = 1, ...5; \sum_{k=1}^{5} N_k^r = N_r; \tag{52}$$

The following iteration process works efficiently to resolve the above system:
a) Start with initial guess for N_5^r;
b) Calculate $h_{min} = L(\Omega_5)\delta h/((1 + \delta h)^{N_5^r} - 1)$;
c) Calculate $N_k^r = \ln(1 + L(\Omega_k)\delta h/h_{min})/\ln(1 + \delta h)$ for $k = 1, ...4$;
d) Check: if $\sum_{k=1}^{5} N_k^r < N_r$, then increase N_5^r by one and return to b). If $\sum_{k=1}^{5} N_k^r > N_r$, then reduce N_5^r by one and return to b) until convergence.

2) Calculate error δ (51) of the numerical solution for the generated mesh, using analytical solution $u(r)$ (49).

3) Optimize mesh by minimizing the error δ with respect to the relative mesh increment δh. It was found that the function $\delta(\delta h)$ is a convex smooth function with one minimum (see Figure 21), which can be found using the following iteration process:
a) Start with minimal δh^1, maximal δh^4 and intermediate δh^3: $\delta h^1 < \delta h^3 < \delta h^4$ and calculate corresponding errors δ_1, δ_3 and δ_4, using step 2);
b) Calculate another intermediate $\delta h^2 = \sqrt{\delta h^1 \delta h^3}$ with corresponding error δ_2;
c) If $\delta_2 > \delta_3$, then assign $\delta h^1 = \delta h^4, \delta h^4 = \delta h^2$ and return to step b), else assign $\delta h^4 = \delta h^1, \delta h^1 = \delta h^3, \delta h^3 = \delta h^2$ and return to step b).

The described nonuniform mesh in radial direction can be efficiently used for solving 2D problems, because the 2D solution at the vertical symmetry plane $z = 0$ behaves similarly to the 1D solution (48).

Now let us describe the method of generating an adaptive nonuniform mesh in the axial domain. To do this we use a similar approach as used for the radial grid. Consider behavior of the magnetic field of a single coil along the middle radius line $r = (R_1 + R_2)/2$. For values of z inside the coil: $z < Z_2$, radial component B_r goes up from zero to its maximal value at the coil edge. Outside the coil, B_r goes down, approaching zero at large z. It is hard to find a 1D magnetostatic problem in

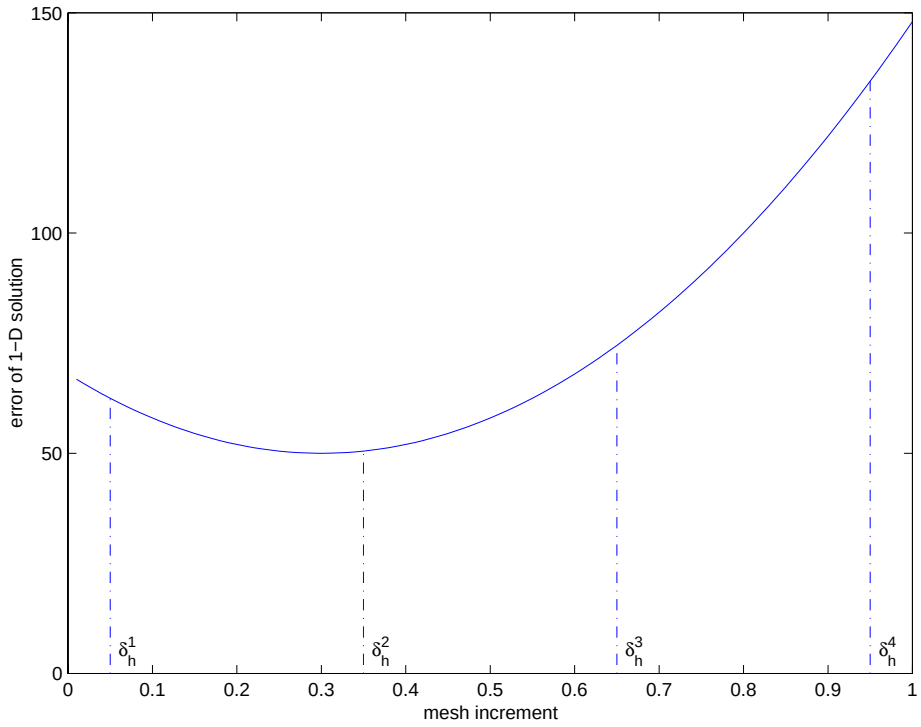


Figure 21: Minimization of the numerical error with respect to the relative mesh step increment.

Cartesian coordinates with a solution having that behavior, but a solution of the 1D problem for an infinite cylindrical wire with uniform current behaves in a very similar way:

$$B(z) = \begin{cases} \dfrac{z}{2}, & 0 \le z \le Z_2, \\ \dfrac{Z_2^2}{2z}, & Z_2 \le z \le Z_3, \end{cases} \tag{53}$$

That solution is displayed in Figure 22.

Corresponding magnetic potential $A(z)$ has the following analytical formula

$$A(x) = \begin{cases} \dfrac{Z_2^2 - z^2}{4} + \dfrac{Z_2^2}{2} ln\left(\dfrac{Z_3}{Z_2}\right), & 0 \le z \le Z_2, \\ \dfrac{Z_2^2}{2} ln\left(\dfrac{Z_3}{z}\right), & Z_2 \le z \le Z_3, \end{cases} \tag{54}$$

which can be found as a solution to the following 1D Poisson problem

$$-\frac{1}{z}\frac{\partial}{\partial z}\left(z\frac{\partial A}{\partial z}\right) = \rho, \frac{\partial A}{\partial z}(0) = 0, A(Z_3) = 0 \tag{55}$$

with a piece-wise uniform charge density function $\rho = 1$ for $0 \le z \le Z_2$ and $\rho = 0$ otherwise.

In formulas (53, 54, 55) we used variable z for the radial coordinate because the formulas are used to generate an adaptive z-mesh as described below.

The computational domain $\Omega = [0, Z_3]$ was divided into two subdomains: $\Omega_1 = [0, Z_2], \Omega_2 = [Z_2, Z_3]$. The grid was created to be fine next to the only singular point: $z = Z_2$. In Ω_1 the grid step size satisfies to the following relation: $h_{i+1} = h_i/(1 + \delta h)$, in Ω_2: $h_{i+1} = h_i(1 + \delta h)$, where $\delta h > 0$ is

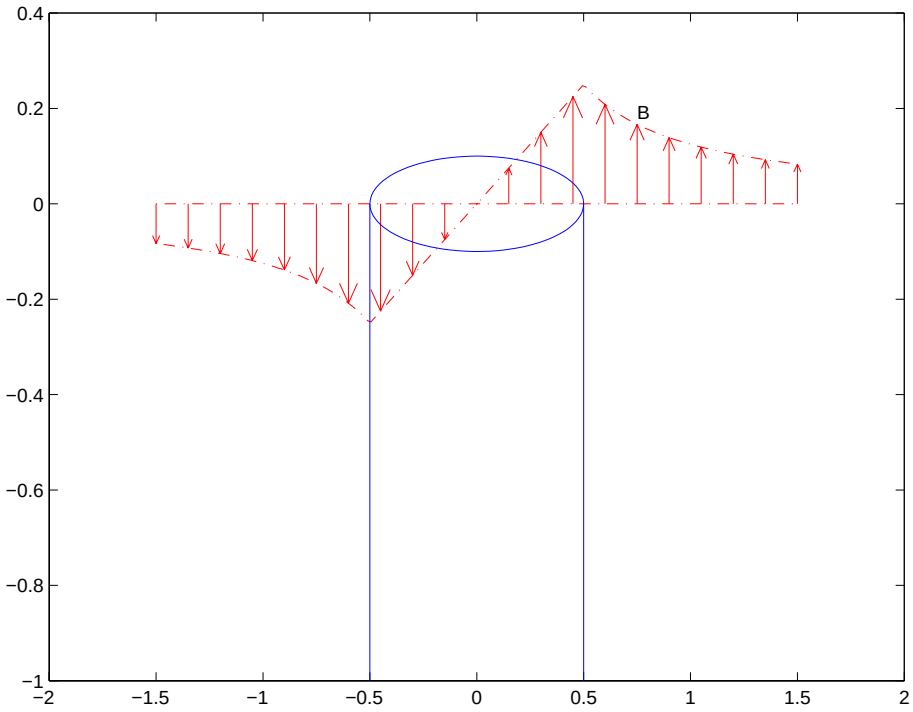


Figure 22: One-dimensional magnetic field for the infinite wire with radius $Z_2 = 0.5$.

a relative increment of the grid-step. It means that in each subdomain Ω_k, the length $L(\Omega_k)$ can be expressed through a number of mesh steps N_k^r as following:

$$L(\Omega_k) = h_{min} + h_{min}(1+\delta h) + h_{min}(1+\delta h)^2 + ...h_{min}(1+\delta h)^{N_k^r - 1} = h_{min}\left((1+\delta h)^{N_k^r} - 1\right)/\delta h.$$

The 1D adaptive mesh generation has the following three steps.

1) For given N_r number of points and for given relative mesh increment δh, create a nonuniform mesh in the domain Ω. This requires us to find 3 unknowns: the number of grid steps in subdomains N_1^r, N_2^r and the minimal mesh size h_{min}. The unknowns must satisfy the following 3 equations:

$$h_{min}((1+\delta h)^{N_k^r} - 1)/\delta h = L(\Omega_k), k = 1,2; N_1^r + N_2^r = N_r; \tag{56}$$

The following iteration process works efficiently to resolve the above system:
a) Start with initial guess for N_2^r;
b) Calculate $h_{min} = L(\Omega_2)\delta h/((1+\delta h)^{N_2^r} - 1)$;
c) Calculate $N_1^r = \ln(1 + L(\Omega_1)\delta h/h_{min})/\ln(1+\delta h)$;
d) Check: if $N_1^r + N_2^r < N_r$, then increase N_2^r by one and return to b). If $N_1^r + N_2^r > N_r$, then reduce N_2^r by one and return to b) until convergence.

2) Calculate error δ (51) of the numerical solution for the generated mesh, using the analytical solution $u(r)$ (54).

3) Optimize the mesh by minimizing the error δ with respect to the relative mesh increment δh. The minimization is done using the same method as for generating the adaptive r-mesh, described above.

Appendix 5: Calculation of the magnetic field on the symmetry line

Accurate calculation of the magnetic field B_z on the symmetry line requires use of both the fine grid in the radial direction and the high-order finite difference approximation. Taking into consideration the zero conditions for magnetic flux: $u(0,z) = 0$ and the magnetic potential: $A(0,z) = 0$, the following analytical formula can be used for computing the axial magnetic field:

$$B_z^a(0,z) = \lim_{r\to 0}\left(\frac{1}{r}\frac{\partial u}{\partial r}\right) = \frac{\partial^2 u}{\partial r^2}(0,z).$$

The formula can be approximated be the following second-order finite difference expression:

$$B_z^h(0,z_j) = \frac{2u_{1,j}}{(h_1^r)^2} + O(h^2), \tag{57}$$

or the fourth-order finite difference approximation:

$$B_z^h(0,z_j) = \frac{2}{h_2^r(2h_1^r + h_2^r)}\left[\left(\frac{h_1^r + h_2^r}{h_1^r}\right)^2 u_{1,j} - \left(\frac{h_1^r}{(h_1^r + h_2^r)}\right)^2 u_{2,j}\right] + O(h^4). \tag{58}$$

These approximations are derived from the Taylor expansion of $u(r,z)$ near the axis under the condition $u_{1,j} = 0$, as well as the smoothness and symmetry property of the solution. Let us note that for a test function $u = r^2 + ar^4$, the exact value is $B_z^a = 2$ and its approximations B_z^h of the second and fourth orders equal $2(1 + ah_r^2)$ and 2 respectively.

Appendix 6: Convergence in the boundary condition calculation

The numerical error of the magnetic field depends on several factors: the distance $H = min\{Z_3 - Z_2, R_3 - R_2\}$ of the domain boundary from the coil, the number m of quadrature points in the numerical integration (43), the number n of the roundoff in the series (42) and the characteristic mesh-step $(R_3 - R_2)/m$. We consider the influence of these parameters on the example of the rectangular coil $c1$.

We use three variants of the domain boundary at a different distance from the coil. These boundaries are defined by the following computational domains ($\Omega = \{0 \le r \le R_3,\ \ 0 \le z \le Z_3\}$):

$$\Omega_1 : R_3 = Z_3 = 2,\ \Omega_2 : R_3 = Z_3 = 4,\ \Omega_3 : R_3 = Z_3 = 8.$$

Here we take into account that, due to the symmetry property of the field for the single coil, the Neumann boundary condition $\partial u/\partial z = 0$ is used at $z = 0$ and the computational domain includes the half of the coil for $z > 0$. We need to distinguish these domains because the value of s^2 decreases when the distance H increases. So in the series (42) we can take a smaller number of terms n and decrease the computational costs. Moreover, the underintegral function in (43) is smoother in this case. So it is possible to use a smaller number m in the quadrature formulas.

In Table 3 we present the relative errors

$$\hat{\delta} = \max_j\left\{\hat{\delta}_j = \left|\frac{u^a(R_3,z_j) - u^{(m)}(R_3,z_j)}{u^a(R_3,z_j)}\right|\right\}$$

of the boundary values for different integers m, distances R_3, and sufficiently big $n = \overline{n}$. Errors of numerical integration are denoted by δ_r and δ_s for the rectangular quadrature formula (43) and the Simpson's quadrature formula (44) respectively. Actually, we define

$$u^a(R_3,z_j) = u^{(\overline{m})}(R_3,z_j),$$

where $\overline{m}$ is the minimal value of m which satisfies the condition

$$\max_j \left| \frac{u^{(m)}(R_3, z_j) - u^{(m/2)}(R_3, z_j)}{u^{(m)}(R_3, z_j)} \right| \le \epsilon = 10^{-13}.$$

It means that, in some sense, $u^{(\overline{m})}(R_3, z_j)$ is an "almost exact" numerical solution at the boundary. Here $\overline{m} = 200$, $\overline{n} = 32$ were taken.

m	1	2	4	8	16
$\Omega_1: \delta_s$	$1.1{\cdot}10^{-4}$	$8.2{\cdot}10^{-6}$	$5.7{\cdot}10^{-7}$	$3.7{\cdot}10^{-8}$	$2.3{\cdot}10^{-9}$
$\Omega_1: \delta_r$	0.045	0.011	0.0028	0.00069	0.00017
$\Omega_2: \delta_s$	$2.7{\cdot}10^{-5}$	$1.7{\cdot}10^{-6}$	$1.1{\cdot}10^{-7}$	$6.7{\cdot}10^{-9}$	$4.2{\cdot}10^{-10}$
$\Omega_2: \delta_r$	0.038	0.0095	0.0024	0.00060	0.00015
$\Omega_3: \delta_s$	$5.6{\cdot}10^{-6}$	$3.5{\cdot}10^{-7}$	$2.2{\cdot}10^{-8}$	$1.4{\cdot}10^{-9}$	$8.6{\cdot}10^{-11}$
$\Omega_3: \delta_r$	0.036	0.0091	0.0023	0.00057	0.00014

Table 3. Errors of numerical integration for coil "c1"

The table shows that the error is of order $O(m^{-2})$ for the central rectangle and $O(m^{-4})$ for the Simpson's quadratures and the last ones give an essentially smaller error. For a big distance H the quadratures are more precise. Let us note that the calculations using Gauss quadratures provide considerably higher efficiency. Its error for $m = 7$ satisfies the inequality $\delta_g < 10^{-14}$ for all three types of the computational domain.

Table 4 includes the errors

$$\check{\delta} = \max_j \left\{ \check{\delta}_j = \left| \frac{u^a(R_3, z_j) - u^{(n)}(R_3, z_j)}{u^a(R_3, z_j)} \right| \right\}$$

of the boundary values for a different number of series terms in (42). The quadrature used is Simpson's one. Here $u^a(R_3, z_j)$ is the "ϵ-asymptotically" exact numerical solution with $\overline{m} = 200$, $\overline{n} = 64$.

n	2	4	8	16	32
Ω_1	0.44	0.22	0.072	0.011	$5.9{\cdot}10^{-4}$
Ω_2	0.19	0.045	0.0034	$3.5{\cdot}10^{-5}$	$8.4{\cdot}10^{-9}$
Ω_3	0.067	0.0056	$5.8{\cdot}10^{-5}$	$1.2{\cdot}10^{-8}$	$2.0{\cdot}10^{-15}$

Table 4. Errors of computing the elliptic integrals for coil "c1"

Note that $n = 32$ is not enough for Ω_1 if high accuracy is required.