Comparison between two numerical methods for a magnetostatic problem

J.-F. Gerbeau, C. Le Bris
CERMICS, Ecole Nationale des Ponts et Chaussées. 6-8 av. Blaise Pascal
Cité Descartes - Champs-sur-Marne 77455 Marne-La-Vallée (France)

Abstract

We draw a comparison between two numerical methods to solve a magnetostatic problem set on a bounded convex domain. The problem is of vector Poisson type and is associated with boundary conditions set on the curl of the unknown, here the magnetic field. These boundary conditions therefore introduce a coupling between the components. One of the two algorithms under consideration consists in an adaptation of the influence matrix method introduced by R. Glowinski and O. Pironneau [4] on the biharmonic equation and extended later by L. Quartapelle *et al* [7], [8], [9]. We present a detailed description of the practical implementation of the algorithm. Through various numerical tests, we compare this uncoupled method with a strategy consisting of a direct attack of the coupled problem.

Résumé

Nous comparons deux méthodes numériques pour résoudre un problème de magnétostatique posé sur un domaine convexe borné. Il s'agit d'un problème de Poisson vectoriel dont les conditions aux limites sont posées sur le rotationel de l'inconnu, ici le champ magnétique. Ces conditions aux limites couplent donc les composantes du vecteur inconnu. L'un des deux algorithmes considérés est une adaptation de la méthode des matrices d'influence introduite par R. Glowinski et O. Pironneau [4] pour le problème du bi-laplacien et étendue ensuite par L. Quartapelle *et al* [7], [8], [9]. Nous présentons une description détaillée de l'implémentation pratique de l'algorithme. Nous comparons à travers divers tests numériques cette méthode découplée avec une résolution directe du problème couplé.

1 Introduction

In this article, we consider a magnetostatic problem set on a bounded convex domain Ω of $\mathbb{R}^3$, enclosed in a $\mathcal{C}^{1,1}$ boundary Γ. The magnetic field B we

seek is the solution to a vector Poisson problem with non-classical boundary conditions, namely a system of the form

$$\left\{\begin{array}{rcll} \operatorname{curl}(\operatorname{curl} B) & = & f & \\ \operatorname{div} B & = & 0 & \\ \operatorname{curl} B \times n & = & k \times n & \text{on } \Gamma \\ B.n & = & q & \text{on } \Gamma. \end{array}\right. \tag{1.1}$$

It can be seen on (1.1) that the boundary conditions on $\operatorname{curl} B \times n$, very natural from the physical standpoint, introduce a coupling between the components of B which prevents one from solving the vectorial problem as three scalar Poisson equations, contrarily to the case when there are three Dirichlet boundary conditions. Some difficulties are therefore likely to be encountered from a computational viewpoint. Indeed, whereas in the case of a "classical" vector Poisson problem, three linear systems of size N^2 have to be solved (N denotes the number of degrees of freedom on the mesh), a coupled system is here to be solved, which increases the required memory.

Let us point out that such a situation also occurs in the framework of hydrodynamics when a vector potential associated to the velocity field in incompressible flows is computed. In this latter context, the system is

$$\left\{\begin{array}{rcll} -\triangle u & = & f, & \\ u \times n & = & a \times n & \text{on } \Gamma, \\ \operatorname{div} u & = & b & \text{on } \Gamma. \end{array}\right. \tag{1.2}$$

In order to solve (1.2) L. Quartapelle and coworkers have extended and analysed an interesting uncoupled algorithm (see [7], [8], [9]) based on the influence matrix method introduced by R. Glowinski and O. Pironneau. The idea is basically to transform system (1.2) into the resolution of a standard system with Dirichlet boundary conditions together with the resolution of a problem on the surface Γ. From the computational viewpoint, this strategy replaces the resolution of the original linear system by linear systems of smaller size N^2. The boundary conditions on the vector Poisson equations are different in magnetostatics from the ones used in hydrodynamics, but we show in the present work that this algorithm may however be extended in order to treat the magnetostatic problem.

Our aim is to compare the following two strategies to solve problem (1.1) :

- a "direct method" which treats simultaneously the three components of B,
- an "uncoupled method", following the ideas of Glowinski-Pironneau and Quartapelle and coworkers, which treats successively the three components of B.

Let us briefly derivate the magnetostatic equation we shall consider henceforth. We begin with the stationary Maxwell equations :

$$\operatorname{curl}(B/\mu) = j, \tag{1.3}$$

$$\operatorname{div} B = 0, \tag{1.4}$$

$$\operatorname{curl} E = 0. \tag{1.5}$$

While the magnetostatic problem (1.3)-(1.4) is often solved numerically through the introduction of the so-called scalar and vector potentials (see for instance E. Emson [2] and the bibliography therein), we shall not follow this approach here.

In the sequel, μ is supposed to be constant, thus we may set $\mu = 1$ without loss of generality. On the boundary Γ, we specify some components of the magnetic and electric fields as follows :

$$B.n|_\Gamma = q,$$

$$E \times n|_\Gamma = \frac{1}{\sigma}\, k.$$

where n is the outward-pointing normal to Ω. Using the Ohm's law $j = \sigma E$ (or $j = \sigma(E + u \times B)$ with $u|_\Gamma = 0$ for a magnetohydrodynamics flow), the boundary conditions on E also reads

$$\frac{1}{\sigma} \operatorname{curl} B \times n = \frac{1}{\sigma}\, k \times n.$$

Let us denote j/σ by g in the sequel. Taking the *curl* of (1.3) we obtain the system

$$\begin{aligned}
\operatorname{curl}\left(\frac{1}{\sigma} \operatorname{curl} B\right) &= \operatorname{curl} g \qquad \text{in } \Omega, && (1.6)\\
\operatorname{div} B &= 0 \qquad \text{in } \Omega, && (1.7)\\
\frac{1}{\sigma} \operatorname{curl} B \times n &= \frac{1}{\sigma}\, k \times n \qquad \text{on } \Gamma, && (1.8)\\
B.n &= q \qquad \text{on } \Gamma. && (1.9)
\end{aligned}$$

We suppose that the conductivity σ is constant over the domain (see however Remark 5.1 below) and we set $\sigma = 1$. Therefore, the equations to be solved are :

$$\begin{aligned}
\operatorname{curl}(\operatorname{curl} B) &= f \qquad \text{in } \Omega, && (1.10)\\
\operatorname{div} B &= 0 \qquad \text{in } \Omega, && (1.11)\\
\operatorname{curl} B \times n &= k \times n \qquad \text{on } \Gamma, && (1.12)\\
B.n &= q \qquad \text{on } \Gamma, && (1.13)
\end{aligned}$$

where $f = \operatorname{curl} g$.

We shall proceed as follows. We study in Section 2 two variational formulations of (1.1). The formulation presented in 2.2 will be used in the uncoupled strategy whereas the formulation of Section 2.3 will be used in the direct method. The two approaches are then detailed in Sections 3 and 4. We compare the results obtained with the two methods in Section 5 in term of precision, CPU time and memory. Finally, in Section 6 we draw conclusions about our whole work.

2 Variational formulations

We present in this section two variational formulations of (1.10)-(1.13) that will be used in the sequel. We refer for example to F. Kikuchi [5] for other formulations of the magnetostatic problem.

2.1 Functional setting

We recall that Ω is supposed to be a bounded convex domain with a $\mathcal{C}^{1,1}$ boundary. In particular, we exclude for this theoretical study domains with holes. We are aware that this assumption might be too restrictive in some practical cases, nevertheless, such domains are sufficient for the applications we deal with. The first consequence of this assumption of regularity is the continuous embedding (see V. Girault, P.A. Raviart [3], Theorem 3.8 and 3.9)

$$\{B \in L^2(\Omega)^3, \operatorname{curl} B \in L^2(\Omega)^3, \operatorname{div} B \in L^2(\Omega)^3, B.n|_\Gamma = 0\} \hookrightarrow H^1(\Omega)^3. \tag{2.1}$$

In the sequel, we shall need the following functional spaces

$$W = \{B \in (H^1(\Omega))^3, B.n|_\Gamma = 0\},$$

$$W_q = \{B \in (H^1(\Omega))^3, B.n|_\Gamma = q\}.$$

We denote by $(.,.)$ the usual inner product of $L^2(\Omega)^3$, by $< .,. >$ the duality product between $H^{-1}(\Omega)^3$ and $H_0^1(\Omega)^3$, and by $< .,. >_\Gamma$ the duality product between $H^{-1/2}(\Gamma)^3$ and $H^{1/2}(\Gamma)^3$. For B, $C \in H^1(\Omega)^3$ we define :

$$((B,C)) = (\operatorname{curl} B, \operatorname{curl} C) + (\operatorname{div} B, \operatorname{div} C).$$

The second consequence of the assumption on the domain Ω is that there exists a constant $c > 0$ such that for any arbitrary $B \in W$,

$$||B||_{H^1(\Omega)^3} \leq c(||\operatorname{curl} B||_{L^2(\Omega)^3} + ||\operatorname{div} B||_{L^2(\Omega)^3}). \tag{2.2}$$

In other words, $((.,.))$ is a scalar product on W which induces a norm that is equivalent to the $H^1(\Omega)^3$ norm on W (see V. Girault, P.A. Raviart [3], Lemma 3.6).

The data are supposed to have to following regularity :

$$q \in H^{1/2}(\Gamma), k \in H^{-1/2}(\Gamma)^3, g \in H^1(\Omega)^3. \tag{2.3}$$

Moreover, we suppose that :

$$\int_\Gamma q \, d\gamma = 0, \tag{2.4}$$

$$< k \times n, \nabla\Phi >_\Gamma = 0, \quad \forall \Phi \in H^2(\Omega)^3, \tag{2.5}$$

$$(\operatorname{curl} g, \nabla\Phi) = 0, \quad \forall \Phi \in H^2(\Omega)^3. \tag{2.6}$$

Assumption (2.4) is the standard compatibility condition with (1.11). Assumptions (2.5) and (2.6), which are satisfied in physical situations in view of (1.5) and Ohm's law, will be of crucial importance below (see the proof of Proposition 2).

2.2 Classical formulation

We define the following variational problem : find $B \in W_q$ such that

$$((B, C)) = (\operatorname{curl} g, C) + < k \times n, C >_\Gamma, \quad \text{for all } C \in W. \tag{2.7}$$

We then have

Proposition 1
The variational problem (2.7) has a unique solution.

Proof. Remark first that (2.2) implies that $((.,.))$ defines a bilinear symmetric and coercive application on $W \times W$. Since $C \to (\operatorname{curl} g, C) + < k \times n, C >_\Gamma$ belongs obviously to W', the Lax-Milgram Theorem implies there exists a unique $B_0 \in W$ such that

$$((B_0, C)) = (\operatorname{curl} g, C) + < k \times n, C >_\Gamma, \quad \text{for all } C \in W.$$

Next, we define $B_q = \nabla\Phi$, with Φ such that

$$\begin{aligned} -\triangle \Phi &= 0 \qquad \text{in } \Omega \\ \frac{\partial \Phi}{\partial n} &= q \qquad \text{on } \Gamma. \end{aligned}$$

Note that $B_q.n = q$, $\operatorname{div} B_q = 0$ and $\operatorname{curl} B_q = 0$ and that $B_q \in H^1(\Omega)^3$ in view of a classical result of elliptic regularity. Thus $B = B_0 + B_q$ is a solution to (2.7). The uniqueness of B is a straightforward consequence of the coercivity of $((.,.))$ on W. $\diamondsuit$

Proposition 2
Under the assumptions (2.3)-(2.6), system (1.10)-(1.13) is equivalent to the variational problem (2.7).

Proof (sketch). It is straightforward to prove that any solution of (1.10)-(1.13) is a solution of (2.7). Conversely, let B be a solution of (2.7). One checks by standard arguments that B satisfies (1.10), (1.12) and (1.13). The point is to show that B is divergence-free. For this purpose, let us consider an arbitrary $h \in L^2(\Omega)$ with $\int_\Omega h\, dx = 0$ and let us define $\Phi \in H^2(\Omega)$ by

$$\left\{ \begin{array}{rcll} -\triangle \Phi & = & h & \\ \dfrac{\partial \Phi}{\partial n} & = & 0 & \text{on } \Gamma. \end{array} \right.$$

Taking $C = \nabla\Phi$ as a test function in (2.7), we obtain :

$$(\operatorname{curl} B, \operatorname{curl} \nabla\Phi) + (\operatorname{div} B, \operatorname{div} \nabla\Phi) = (\operatorname{curl} g, \nabla\Phi) + < k \times n, \nabla\Phi > .$$

In view of the assumptions (2.5) and (2.6) this equality yields :

$$(\operatorname{div} B, h) = 0.$$

It follows that $\operatorname{div} B$ is constant over Ω, and this constant is zero in view of assumption (2.4). Consequently, (1.11) holds. $\diamondsuit$

2.3 A mixed formulation to treat $B.n = q$ as a constraint

In order to turn the formulation of Section 2.2 into a numerical method, we must first construct suitable finite elements approximation spaces for W and W_q. If the boundaries of the domain happen not to be parallel to the coordinnates axes, this construction may be tedious from a computational viewpoint. We now suggest a formulation that circumvents this difficulty : we work on the spaces

$$X = H^1(\Omega)^3,$$
$$M = H^{-1/2}(\Gamma).$$

We denote by $||.||_X$ (resp. $||.||_M$) the usual norm on $H^1(\Omega)^3$ (resp. $H^{-1/2}(\Gamma)$) and we define the bilinear form b on $X \times M$ by

$$b(B, \mu) = < B.n, \mu >_\Gamma$$

For ease of notation, we denote by $< .,. >_\Gamma$ the duality product between $H^{-1}(\Omega)$ and $H_0^1(\Omega)$ or between $H^{-1}(\Omega)^3$ and $H_0^1(\Omega)^3$ as above. We introduce the following mixed formulation :

find $(B,\lambda) \in X \times M$ such that, $\forall (C,\mu) \in X \times M$,

$$\left\{ \begin{array}{rcl} ((B,C)) + b(B,\lambda) & = & (f,C) + < k \times n, C >_\Gamma, \\ b(B,\mu) & = & < q, \mu >_\Gamma . \end{array} \right. \tag{2.8}$$

Proposition 3
The mixed variational problem (2.8) has a unique solution.

Proof. First, we prove that there exists a constant $\beta > 0$ such that

$$\inf_{\mu \in M - \{0\}} \sup_{B \in X - \{0\}} \frac{b(B,\mu)}{||B||_X ||\mu||_M} \geq \beta.$$

Indeed, let μ be in $M - \{0\}$. A classical corollary of the Hahn-Banach theorem yields the existence of $\alpha \in H^{1/2}(\Gamma)$ with $||\alpha||_{H^{1/2}(\Gamma)} = 1$ such that :

$$< \frac{\mu}{||\mu||_M}, \alpha >_\Gamma = 1.$$

It is straightforward to build $B_0 \in X$ such that $B_0.n|_\Gamma = \alpha$ and $||B_0||_X \leq c||\alpha||_{H^{1/2}(\Gamma)} = c$, with c independent on B_0 and α. Thus, for all $\mu \in M$,

$$\frac{b(B_0,\mu)}{||B_0||_X} = \frac{< \mu, \alpha >_\Gamma}{||B_0||_X} \geq \frac{1}{c}||\mu||_M,$$

which proves the inf-sup condition. Moreover, the space $\{C \in X, b(C,\mu) = 0, \forall \mu \in M\}$ is equal to space W and, according to (2.2), the bilinear form $((.,.))$ is coercive on W. Therefore, the classical theory on mixed variational problems permits to conclude the proof.$\diamondsuit$

Remark 2.1 *The solution of (2.8) may be seen as the saddle-point of the Lagrange functional* $\mathcal{L}(B,\mu) = \frac{1}{2}((B,B)) - (f,B) - < k \times n, B >_\Gamma + b(B,\mu) - < q, \mu >_\Gamma$.

Proposition 4
Under the assumptions (2.3)-(2.6), system (1.10)-(1.13) is equivalent to the mixed problem (2.8).

Proof.(sketch) Let B be a solution of (1.10)-(1.13). Equation (1.13) yields $b(B,\mu) = < q, \mu >_\Gamma$, $\forall \mu \in M$. Moreover, since $\operatorname{div} B = 0$, we have

$$\operatorname{curl}(\operatorname{curl} B) - \nabla(\operatorname{div} B) = f.$$

Multiplying this equation by $C \in X$ and integrating by part over Ω we obtain

$$\int_\Omega (\operatorname{curl} B.\operatorname{curl} C + \operatorname{div} B \operatorname{div} C)\, dx - \int_\Gamma C.n \operatorname{div} B\, d\gamma = \int_\Omega f.C\, dx + < k \times n, C >_\Gamma$$

Thus (2.8) holds with the Lagrange multiplier $\lambda = -\mathrm{div}\, B|_\Gamma$. Conversely, some analogous arguments prove that the solution of (2.8) satisfies (1.10)-(1.13). In particular, we check that B is divergence-free in the same fashion as in Proposition 4.$\diamondsuit$

3 Direct resolution based on the mixed formulation

3.1 Penalized formulation

The mixed formulation (2.8) can actually be applied to a finite element analysis. However, it requires the computation of the Lagrange multiplier λ. A standard method to avoid this computation is to consider the corresponding penalized formulation. Let $\varepsilon > 0$, we assume that

$$b(B,\mu) - < q, \mu >_\Gamma = \varepsilon < \lambda, \mu >, \quad \forall \mu \in M,$$

and we seek $B \in X$ such that $\forall C \in X$

$$((B,C)) + \frac{1}{\varepsilon} < B.n, C.n >_\Gamma = (f,C) + < k \times n, C >_\Gamma + \frac{1}{\varepsilon} < q, C.n >_\Gamma . \quad (3.1)$$

3.2 Discretisation

Let $h > 0$ be fixed. The domain Ω is approximated by a polyhedron Ω_h with its vertices on Γ. A partition $\mathcal{T}_h$ of Ω_h into elements consisting of tetrahedrons or convex hexahedrons is performed in a standard way. In the sequel, $R_m(K)$ stands for $P_m(K)$ if K is a tetrahedron and for $Q_m(K)$ if K is a hexahedron, where for each integer $m \geq 0$, P_m and Q_m have the usual meaning. For the sake of simplicity, we only consider Lagrangian finite elements.

We denote by Γ_h the boundary of Ω_h, by n its approximated unit outward pointing normal, and by t_1, t_2 an approximated orthogonal set of tangent vectors (see Remark 3.2 below for the treatment of the singular points of Γ). The number of nodes on Ω_h (resp. Γ_h) is denoted by N (resp. M).

$$\begin{aligned} X^h &= \{v_h \in \mathcal{C}^0(\overline{\Omega}); v_h|_T \in R_m(T), \forall T \in \mathcal{T}_h\}, \\ X_0^h &= \{v_h \in \mathcal{C}^0(\overline{\Omega}); v_h|_T \in R_m(T), \forall T \in \mathcal{T}_h\} \cap H_0^1(\Omega), \\ Y^h &= \{v_h|_{\Gamma_h}, v_h \in X^h\}. \end{aligned}$$

Thus, we search $B_h \in X_h^3$ such that for all $C_h \in X_h^3$

$$((B_h,C_h)) + \frac{1}{\varepsilon} < B_h.n, C_h.n >_\Gamma = (f,C_h) + < k \times n, C_h >_\Gamma + \frac{1}{\varepsilon} < q, C_h.n >_\Gamma . \quad (3.2)$$

Remark 3.1 *In the numerical simulations we have performed, the value* $\varepsilon = 1.e-4$ *has given very good results without increasing too much the condition number of the system.*

Remark 3.2 *In our numerical tests, we have computed the normals and the tangents at the nodes of the boundary. Denoting by* φ_i *the hat function at the node* i*, the approximated normal is given by :*

$$n_i = \frac{\displaystyle\int_\Omega \nabla\varphi_i \, dx}{\displaystyle\int_\Gamma \varphi_i \, dx}.$$

We then deduce from n_i *the values of* t_1 *and* t_2 *at the node* i*.*

In order to compute $((B_h, C_h))$ we can use the following formula (see [1] or [6])

$$((B,C)) = \int_\Omega \nabla B.\nabla C \, dx + \sum_{k=1}^{3} \int_\Gamma (\nabla B^k \times n).(e_k \times C) \, d\gamma, \tag{3.3}$$

where $(e_k,\ 1 \le k \le 3)$ denotes the canonical basis of $\mathbb{R}^3$ and B^k stands for $B.e_k$. This equality can be easily established in the continuous case. It also holds in the discrete case since the boundary terms only involve tangential derivatives, and thus cancel on the inside faces.

From a computational viewpoint, the formula (3.3) shows that it is useless to allocate memory for a 3×3 system of (sparse) blocks $N^2 \times N^2$: we only need 3 blocks of size $N^2 \times N^2$ for the three laplacians and 6 blocks of size $N \times N_\Gamma$ for the boundary terms. Nevertheless, in some practical problems, this system may still be too large. In such cases, one may use the method presented in Section 4 which allows to solve the problem with a $N \times N$ sparse system.

4 Uncoupled resolution based on the first formulation

As above mentioned, the formulation (2.7) has two drawbacks : first, it needs a finite element basis to approximate the space W, second it leads – like formulation (2.8) – to large systems (even if the formula (3.3) somewhat reduces the size of the matrix). Thus, rather than detailing the direct discretisation of (2.7), we present a method based on the same variational formulation which avoids the coupling induced by the boundary conditions and thus lead to smaller matrices.

4.1 Uncoupled formulation

J. Zhu, L. Quartapelle and A. F. D. Loula have considered in [9] a problem arising in computational fluid dynamics which basically shares the same features as ours. They propose an uncoupled technique that we now adapt to (2.7). We introduce :

$$W_{q,T} = \{B \in H^1(\Omega)^3, B.n|_\Gamma = q, B \times n = 0\},$$

$$\mathcal{H}_N = \{B \in H^1(\Omega)^3, \triangle B = 0 \text{ in } \Omega, B.n = 0 \text{ on } \Gamma\}.$$

Note that $B \in W_q^1$ may be decomposed as :

$$B = B_T + B_{\mathcal{H}}.$$

with $B_T \in W_{q,T}$ and $B_{\mathcal{H}} \in \mathcal{H}_N$ (solve $-\triangle B_{\mathcal{H}} = 0$ on Ω, $B_{\mathcal{H}}.n|_\Gamma = 0$, $B_{\mathcal{H}}|_\Gamma \times n = B|_\Gamma \times n$, and set $B_T = B - B_{\mathcal{H}}$). In the same fashion, $C \in W$ may be decomposed as :

$$C = C_0 + C_{\mathcal{H}}$$

with $C_0 \in H_0^1(\Omega)^3$ and $C_{\mathcal{H}} \in \mathcal{H}_N$.

By linearity, (2.7) reads :

$$\left\{ \begin{array}{rcll} ((B, C_0)) & = & (f, C_0), & \forall C_0 \in (H_0^1(\Omega))^3 \\ ((B, C_{\mathcal{H}})) & = & (f, C_{\mathcal{H}}) - < k, C_{\mathcal{H}} \times n >_\Gamma, & \forall C_{\mathcal{H}} \in \mathcal{H}_N. \end{array} \right.$$

Since $B = B_T + B_{\mathcal{H}}$, we have :

$$\begin{array}{rcl} ((B_{\mathcal{H}}, C_0)) & = & \displaystyle\int_\Omega \operatorname{curl}(\operatorname{curl} B_{\mathcal{H}}).C_0 \, dx + \int_\Omega \operatorname{div} B_{\mathcal{H}} \operatorname{div} C_0 \, dx \\ & = & \displaystyle\int_\Omega -\triangle B_{\mathcal{H}}.C_0 + \nabla(\operatorname{div} B_{\mathcal{H}}).C_0 \, dx - \int_\Omega \nabla(\operatorname{div} B_{\mathcal{H}}).C_0 \, dx \\ & = & 0. \end{array}$$

Therefore, the variational formulation (2.7) is equivalent to the following uncoupled formulation : find $B_T \in W_{q,T}$ and $B_{\mathcal{H}} \in \mathcal{H}_N$ such that

$$((B_T, C_0)) = (f, C_0), \quad \forall C_0 \in H_0^1(\Omega)^3 \tag{4.1}$$

$$((B_{\mathcal{H}}, C_{\mathcal{H}})) = (f, C_{\mathcal{H}}) - ((B_T, C_{\mathcal{H}})) - < k \times n, C_{\mathcal{H}} \times n >_\Gamma, \forall C_{\mathcal{H}} \in \mathcal{H}_N. \tag{4.2}$$

Equation (4.1) is a system consisting of three independent scalar Dirichlet problems. One next remarks that (4.2) may be solved as a problem set on

Γ. Indeed, (4.2) is equivalent to find λ^1, $\lambda^2 \in H^{1/2}(\Gamma)$ such that, for all $\mu^1, \mu^2 \in H^{1/2}(\Gamma)$:

$$\begin{aligned}((C_{\mathcal{H}}(\lambda^1,\lambda^2), C_{\mathcal{H}}(\mu^1,\mu^2))) &= (f, C_{\mathcal{H}}(\mu^1,\mu^2)) - ((B_T, C_{\mathcal{H}}(\mu^1,\mu^2))) \\ &\quad - < k, \mu^1 t_1 + \mu^2 t_1 >_\Gamma \end{aligned} \tag{4.3}$$

with $C_{\mathcal{H}}(\mu^1,\mu^2)$ defined by

$$\left\{ \begin{array}{rcll} -\triangle C_{\mathcal{H}}(\mu^1,\mu^2) &=& 0 & \text{in } \Omega, \\ C_{\mathcal{H}}(\mu^1,\mu^2) \times n &=& \mu^1 t_1 + \mu^2 t_2 & \text{on } \Gamma, \\ C_{\mathcal{H}}(\mu^1,\mu^2).n &=& 0 & \text{on } \Gamma. \end{array} \right.$$

Remark 4.1 *As shown in [9],*

$$((C_{\mathcal{H}}(\lambda^1,\lambda^2), C_{\mathcal{H}}(\mu^1,\mu^2))) = ((C_{\mathcal{H}}(\lambda^1,\lambda^2), w))$$

for any arbitrary vector field $w \in (H^1(\Omega))^3$ which coincides with $C_{\mathcal{H}}(\mu^1,\mu^2)$ on Γ. Indeed :

$$\begin{aligned}((C_{\mathcal{H}}(\lambda^1,\lambda^2), C_{\mathcal{H}}(\mu^1,\mu^2))) &= \int_\Omega curl(curl C_{\mathcal{H}}(\lambda^1,\lambda^2)).C_{\mathcal{H}}(\mu^1,\mu^2)\, dx \\ &\quad + \int_\Gamma curl C_{\mathcal{H}}(\lambda_1,\lambda_2) \times n.C_{\mathcal{H}}(\mu^1,\mu^2)\, d\gamma \\ &\quad - \int_\Omega \nabla(div C_{\mathcal{H}}(\lambda^1,\lambda^2)).C_{\mathcal{H}}(\mu^1,\mu^2)\, dx \\ &= \int_\Gamma curl C_{\mathcal{H}}(\lambda_1,\lambda_2) \times n.w\, d\gamma \\ &= ((C_{\mathcal{H}}(\lambda_1,\lambda_2), w)). \end{aligned}$$

It may be proved following the same lines that :

$$(f, C_{\mathcal{H}}(\mu^1,\mu^2)) - ((B_T, C_{\mathcal{H}}(\mu^1,\mu^2))) - < k, \mu^1 t_1 + \mu^2 t_1 >_\Gamma .$$

$$= (f, w) - ((B_T, w)) - < k, w \times n >_\Gamma$$

These properties will be used in the numerical implementation below.

4.2 Discretisation

We define an approximation of the space $\mathcal{H}_T$ by

$$\mathcal{H}_T^h = \{B_h \in (X^h)^3, \int_{\Omega_h} \nabla B_h.\nabla C_h\, dx = 0, \forall C_h \in (X_0^h)^3 \text{ and } B_h.n|_{\Gamma_h} = 0\}$$

Let $(\varphi_1, ..., \varphi_{N_\Gamma})$ be a basis of Y_h (φ_i is typically the "hat function" at the node i of Γ_h). We construct a basis $(b_1^1, ..., b_{N_\Gamma}^1, b_1^1, ..., b_{N_\Gamma}^2)$ of $\mathcal{H}_T^h$ with b_i^k

such that $b_i^1 = \varphi_i t_1$ and $b_i^2 = \varphi_i t_2$ on Γ_h. In other words, b_i^k satisfies for all $C_h \in (X_0^h)^3$:

$$\left\{ \begin{array}{rcl} (\nabla b_i^k, \nabla C_h) & = & 0 \\ b_i^k & = & \varphi_i t_k \quad \text{on } \Gamma \end{array} \right.$$

An approximation B_T^h of the solution to equation (4.1) may be computed in a classical way through the resolution of three Poisson scalar problems.

In order to solve (4.2), let us now decompose $B_{\mathcal{H}}^h$ on the basis $(b_i^1, b_i^2)_{i=1...N_\Gamma}$:

$$B_{\mathcal{H}}^h = \sum_{i=1}^{N_\Gamma} \lambda_i^1 b_i^1 + \lambda_i^2 b_i^2.$$

The pair $(\lambda_i^1, \lambda_i^2)$ may be seen as the coordinates of $B_{\mathcal{H}}^h$ on the "discrete harmonic basis" $(b_i^1, b_i^2)_{i=1...N_\Gamma}$ as well as the tangential components of $B_{\mathcal{H}}^h$ on Γ_h. That is why (4.2) may be interpreted as a problem set on the boundary.

The discrete approximation of (4.2) reads :

$$((B_{\mathcal{H}}^h, b_i^p)) = (f, b_i^p) - ((B_T^h, b_i^p)) - < k, b_i^p \times n >_\Gamma,$$

for $i = 1, ..., N_\Gamma$ and $p = 1, 2$. More precisely, in order to solve (4.2), we have to find $(\lambda_1^1, ..., \lambda_{N_\Gamma}^1, \lambda_1^2, ..., \lambda_{N_\Gamma}^2)$ such that

$$\left\{ \begin{array}{rcl} \displaystyle\sum_{j=1}^{N_\Gamma} \lambda_j^1((b_j^1, b_i^1)) + \lambda_j^2((b_j^2, b_i^1)) & = & (f, b_i^1) - ((B_T^h, b_i^1)) - < k, b_i^1 \times n >_\Gamma \\ \displaystyle\sum_{j=1}^{N_\Gamma} \lambda_j^1((b_j^1, b_i^2)) + \lambda_j^2((b_j^2, b_i^2)) & = & (f, b_i^2) - ((B_T^h, b_i^2)) - < k, b_i^2 \times n >_\Gamma \end{array} \right. \tag{4.4}$$

for $i = 1, ..., N_\Gamma$.

4.3 Numerical implementation

In this section, we lay some emphasis on the practical implementation of the discrete algorithm we have presented above.

We denote by $\mathcal{A}$ the matrix of the linear system (4.4) and by $\mathcal{M}$ the matrix of the linear system yielded by the discretisation of the original coupled problem (2.7).

The discretisation presented in Section 4.2 has two drawbacks. First, the discrete vector harmonic basis (b_i^1, b_i^2) must be computed, which involves the resolution of $2N_\Gamma$ Poisson problems on Ω_h. Second, the size of $\mathcal{A}$ is actually smaller than the size of $\mathcal{M}$ ($(2N_\Gamma)^2$ instead of $(3N)^2$) but $\mathcal{A}$ is full whereas

$\mathcal{M}$ is sparse, thus it is not clear whether it is much cheaper to store $\mathcal{A}$ rather than $\mathcal{M}$.

In order to overcome both difficulties, we make use of the conjugate gradient algorithm presented by R. Glowinski and O. Pironneau in [4] and that we recall now for the convenience of the reader. As we shall see, this method avoids both the computation of the discrete harmonic basis and the storage of $\mathcal{A}$.

We set $\Lambda = (\lambda_1^1, ..., \lambda_{N_\Gamma}^1, \lambda_1^2, ..., \lambda_{N_\Gamma}^2)$ and we denote by β the right-hand-side of (4.4). Suppose now that we solve (4.4) by the conjugate gradient method. The algorithm reads :

$$\begin{aligned}
\Lambda_0 \in \mathbb{R}^{2N_\Gamma}, \quad & \text{arbitrarily chosen} && (4.1)\\
g_0 &= \mathcal{A}\Lambda_0 - \beta && (4.2)\\
z_0 &= g_0, \qquad n = 0 && (4.3)\\
d_n &= \mathcal{A}z_n && (4.4)\\
\rho_n &= z_n.g_n / z_n.d_n && (4.5)\\
\Lambda_{n+1} &= \Lambda_n - \rho_n z_n && (4.6)\\
g_{n+1} &= g_n - \rho_n d_n && (4.7)\\
\gamma_n &= g_{n+1}.g_{n+1} / g_n.g_n && (4.8)\\
z_{n+1} &= g_{n+1} + \gamma_n z_n && (4.9)\\
n &\rightarrow n+1 \text{ and go to (4.4)} && (4.10)
\end{aligned}$$

In order to compute $\mathcal{A}z$ for any vector $z = (z^1, z^2) = (z_1^1, ..., z_{N_\Gamma}^1, z_1^2, ..., z_{N_\Gamma}^2) \in \mathbb{R}^{2N_\Gamma}$ without explicitly knowing $\mathcal{A}$, we define the function $C \in \mathcal{H}_T^h$ by

$$C(z^1, z^2) = \sum_{i=1}^{N_\Gamma} z_i^1 b_i^1 + z_i^2 b_i^2.$$

Recall that $C(z^1, z^2)$ is the solution of the following discrete Poisson problem : find $C(z^1, z^2) \in (X^h)^3$ such that

$$\left\{ \begin{array}{rcl} (\nabla C(z^1, z^2), \nabla D) &=& 0 \quad \text{for all } D \in (X_0^h)^3, \\ C.n &=& 0, \\ C(z^1, z^2) \times n &=& \sum_{i=1}^{N_\Gamma} z_i^1 \varphi_i t_1 + z_i^2 \varphi_i t_2. \end{array} \right. \qquad (4.11)$$

Let us note that this problem may be straightforwardly decoupled in three

scalar Laplace equations. By definition of $\mathcal{A}$, we have

$$\mathcal{A}z = \left(\begin{array}{l} \displaystyle\sum_{j=1}^{N_\Gamma} z_j^1((b_j^1, b_i^1)) + z_j^2((b_j^2, b_i^1)) \\ \displaystyle\sum_{j=1}^{N_\Gamma} z_j^1((b_j^1, b_i^2)) + z_j^2((b_j^2, b_i^2)) \end{array} \right)_{i=1,\ldots,N_\Gamma}$$

$$= \left(\begin{array}{l} ((C(z^1, z^2), b_i^1)) \\ ((C(z^1, z^2), b_i^2)) \end{array} \right)_{i=1,\ldots,N_\Gamma}$$

Therefore, the computation of $\mathcal{A}z$ only requires the knowledge of the vector field $C(z^1, z^2)$ and not the explicit knowledge of $\mathcal{A}$ itself. However, it also requires so far the knowledge of the basis $(b_i^1, b_i^2)_{i=1\ldots N_\Gamma}$.

Let us now indicate how to avoid the computation of b_i^k. We denote by w_i^1 (resp. w_i^2) the vector field of $(X^h)^3$ which takes the value zero at all the nodes of Ω_h except at the node i of Γ_h where it takes the value t_1 (resp. t_2). The function w_i^k coincides with b_i^k on Γ_h, thus, in view of remark 4.1, $((C(z^1, z^2), b_i^k)) = ((C(z^1, z^2), w_i^k))$. Therefore we have

$$\mathcal{A}z = \left(\begin{array}{l} ((C(z^1, z^2), w_i^1)) + ((C(z^1, z^2), w_i^1)) \\ ((C(z^1, z^2), w_i^2)) + ((C(z^1, z^2), w_i^2)) \end{array} \right)_{i=1,\ldots,N_\Gamma} \tag{4.12}$$

and

$$\beta = \left(\begin{array}{l} (f, w_i^1) - ((B_T, w_i^1)) - < k, w_i^1 \times n >_\Gamma \\ (f, w_i^2) - ((B_T, w_i^2)) - < k, w_i^2 \times n >_\Gamma \end{array} \right)_{i=1,\ldots,N_\Gamma} \tag{4.13}$$

Thus, step (4.2) of the conjugate gradient algorithm is replaced by the sequence

- compute the vector field $C(\Lambda_0^1, \Lambda_0^2)$ related to Λ_0 by solving (4.11).
- compute $\mathcal{A}\Lambda_0$ by (4.12).
- compute β by (4.13).

Likewise, step (4.4) is replaced by

- compute the vector field $C(z_n^1, z_n^2)$ related to z_n by solving (4.11).
- compute $\mathcal{A}z$ by (4.12).

The computation and the storage of matrix $\mathcal{A}$ are therefore not necessary, but the price to pay for this saving in memory usage is an increase of the computational time due to the fact that three Poisson problems have to be solved on Ω_h at each step of the conjugate gradient algorithm.

Remark 4.2 *Note that the three Poisson problems of each step of the conjugate gradient algorithm are independent and may easily be solved simultaneously on a parallel architecture.*

5 Numerical results

In the sequel, the method of Section 3 will be referred to as the "direct method" and the algorithm presented in Section 4 will be referred to as the "uncoupled method". We have implemented these algorithms both in 2D and 3D with Q1 finite elements. The tests in two dimensions are the following :

1) $\Omega = [0,1]^2$, $B = (\sin(\pi x)\cos(\pi y)/\pi, -\cos(\pi x)\sin(\pi y)/\pi)$.

2) $\Omega = [-1,1]^2$, $B = (-x^4 y/12 + y x^2/2, x^3 y^2/6 - x^5/60 - y^2 x/2 + x)$.

3) $\Omega = D(0,1)$, $B = (-y/2, x/2)$.

where $D(0,1)$ denotes a disk with center $(0,0)$ and radius 1.

We present in Table 1 the results obtained on various meshes with the two methods. The relative error is computed in L^2 norm. Solutions are plotted on Figures 1, 2, 3.

		Uncoupled method		Direct method	
Test	Grid	Rel. error	$\|\|\text{div}\, B\|\|_{L^2}$	Rel. error	$\|\|\text{div}\, B\|\|_{L^2}$
1	20×20	.0020569	.0319320	.0020570	.0319320
	40×40	.0005141	.0160153	.0005140	.0160154
	80×80	.0001312	.0080139	.0001285	.0080138
2	20×20	.0073202	.0373165	.0073196	.0373145
	40×40	.0037708	.0196841	.0037711	.0196779
	80×80	.0019940	.0115525	.0019938	.0115325
3	169 nodes	.0086348	.0006356	.0086349	.0006352
	649 nodes	.0021503	.0007940	.0021503	.0007926
	2545 nodes	.0005400	.0006745	.0005400	.0006725

Table 1: Tests in two dimensions.

In three dimensions, the following cases have been considered :

		Uncoupled Method		Direct method	
Test	Grid	Rel. error	$\|\|\mathrm{div}\, B\|\|_{L^2}$	Rel. error	$\|\|\mathrm{div}\, B\|\|_{L^2}$
4	$5 \times 5 \times 5$	.0331172	.0820945	.0331172	.0820946
	$10 \times 10 \times 10$	.0082390	.0442373	.0082382	.0442374
	$20 \times 20 \times 20$	.0020602	.0225329	.0020570	.0225329
5	$5 \times 5 \times 5$	.2232074	.0664852	.2232074	.0664852
	$10 \times 10 \times 10$	.0514764	.0491199	.0514764	.0491199
	$20 \times 20 \times 20$	.0125658	.0269531	.0125658	.0269530
6	840 nodes	.0722001	.0325454	.0721998	.0325452
	2560 nodes	.0506948	.0150413	.0506958	.0150396
	5436 nodes	.0434860	.0101620	.0434856	.0101527

Table 2: Tests in three dimensions.

4) $\Omega = [0,1]^3$, $B = (\sin \pi x \cos \pi y \cos \pi z/\pi, -\cos \pi x \sin \pi y \cos \pi z/\pi, 0)$.

5) $\Omega = [0,1]^3$, $B = \mathrm{curl}\,(g,g,g)$ with $g = 10^4(xyz(x-1)(y-1)(z-1))^3$.

6) $\Omega =$ Cylinder with height 0.6 and cross section $D(0,1)$, $B = (-y/2, x/2, 0)$.

Table 2 and Figures 4, 5, 6 show the results we obtained in 3D.

We have used the Conjugate Gradient method with Incomplete Cholesky preconditioner to solve the linear systems in both methods. We emphasize that it is necessary to achieve a very good convergence in the resolution of linear systems into the loop of Glowinski-Pironneau conjugate gradient algorithm.

These tables show that the relative errors and the value of $||\mathrm{div}\, B||_{L^2}$ are almost the same for the two methods. The evolution of these values with the step of the grid is good. The only exceptions are the values of $||\mathrm{div}\, B||_{L^2}$ in test 3. Our understanding of this phenomena is rather poor, but we suspect it is due to the non regularity of the mesh on the disk.

In our examples and with our home-made code, the memory required by the uncoupled method is three (resp. *six*) times as small as the memory needed by the direct one in 2D (resp. in 3D). On the contrary, the CPU time required for the uncoupled method is about 1.5 times as large as the CPU time required for the direct method. But, as said above, the uncoupled algorithm can be easily treated on a parallel architecture. For the 3D tests, we have used three computers connected within a PVM network : each machine solves one of the three scalar Poisson problem and compute the third of the expression (4.12). The CPU time is then almost divided by a factor 2. The uncoupled method becomes therefore faster than the direct one.

Remark 5.1 *The only limitation we see today to the use of the uncoupled algorithm is that it is not well-suited for problems involving a non homogeneous conductivity σ. In this case the equations (1.10)-(1.13) have to be replaced by (1.6)-(1.9). Current work is in progress on the subject but we can already suggest three tricks to treat the case when σ is not constant with the uncoupled method.*

The first way is to split $\frac{1}{\sigma}\, curl\, B$ in a gradient and a solenoidal part :

$$\frac{1}{\sigma} curl\, B = curl\, A - \nabla\psi.$$

The unknown ψ is then determined by a scalar Poisson problem, A and B by a vector Poisson problem which can be solved by the uncoupled method.

The second way is to use the vector analysis relation

$$curl\,(\frac{1}{\sigma} curl\, B) = \nabla\frac{1}{\sigma} \times curl\, B + \frac{1}{\sigma} curl\,(curl\, B),$$

and to adopt an iterative strategy : the value B^{n+1} is determined by the resolution of a vector Poisson problem with $\sigma\nabla\frac{1}{\sigma} \times curl\, B^n$ at the right hand side.

In the case when σ is constant over two subdomains Ω_1 and Ω_2 of Ω, a third way consists in solving a vector Poisson problem alternatively on the two subdomains. The boundary conditions on $\partial\Omega_1 \cap \partial\Omega_2$ deal with $curl\, B \times n$ and $B.n$ for the problem set on Ω_1 and with $div\, B$ and $B \times n$ for the problem set on Ω_2.

6 Conclusion

We have proposed two approaches to solve a magnetostatic problem : a direct method, very natural, and an uncoupled algorithm, that draws its inspiration from methods exposed in [7] and [4] in other frameworks. We have studied the variational formulations and the numerical implementation for both approaches. Our numerical results show that the two methods are very similar in term of accuracy. In average the direct method is 1.5 times as fast as the uncoupled one. Conversely the memory required in 3D by the uncoupled method is 6 times as small as the memory needed by the direct one. In a very large problem, the uncoupled method is therefore more attractive. It is indeed all the more attractive since we have shown that the uncoupled algorithm can be straighforwardly used on a parallel architecture of three computers which roughly divides the CPU time by a factor of two.

In addition, we have briefly suggested three tricks to extend the uncoupled algorithm to the case when the electric conductivity is not constant over the domain. Nevertheless, we believe that in this non-homogeneous case, the direct method remains more natural.

In conclusion, our study shows that, in comparison with the direct resolution of the coupled system, the uncoupled method :

- is as accurate as the direct one,

- is far more attractive in term of memory storage,

- does not require a much longer CPU time.

Acknowledgements The authors should like to express their thanks to Prof. O. Pironneau and Prof. M. Bercovier for stimulating discussions.

References

[1] F. Assous, P. Degond, E. Heintze, P.A. Raviart, and Segre J. On a finite-element method for solving the three dimensional Maxwell equations. *Jour. Comp. Phys.*, 109:222–237, 1993.

[2] A. Bossavit, C. Emson, and I.D. Mayergoyz. *Méthodes numériques en électromagnétisme*. Eyrolles, 1991.

[3] V. Girault and P.-A. Raviart. *Finite element methods for Navier-Stokes equations*. Springer-Verlag, 1986.

[4] R. Glowinski and O. Pironneau. Numerical methods for the first biharmonic equation and for the two-dimensional Stokes problem. *SIAM*, 21(2):167–212, April 1979.

[5] F. Kikuchi. Numerical analysis electrostatic and magnetostatic problem. *Sugaku expositions*, 6(1):332–345, June 1993.

[6] O. Pironneau. *Finite element methods for fluids*. Wiley, 1989.

[7] L. Quartapelle and A. Muzzio. Decoupled solution of a vector Poisson equations with boundary condition coupling. In G. de Vahl Davis and C. Fletcher, editors, *Computational Fluid Dynamics*, pages 609–619. Elsevier Science Publishers, (North-Holland), 1988.

[8] J. Zhu, A.F.D. Loula, and L. Quartapelle. Finite element analysis of a nonstandard vector poisson problem. Preprint.

[9] J. Zhu, L. Quartapelle, and A.F.D. Loula. Uncoupled variational formulation of a vector Poisson problem. *C. R. Acad. Sci. Paris, série 1*, (323):971–976, 1996.

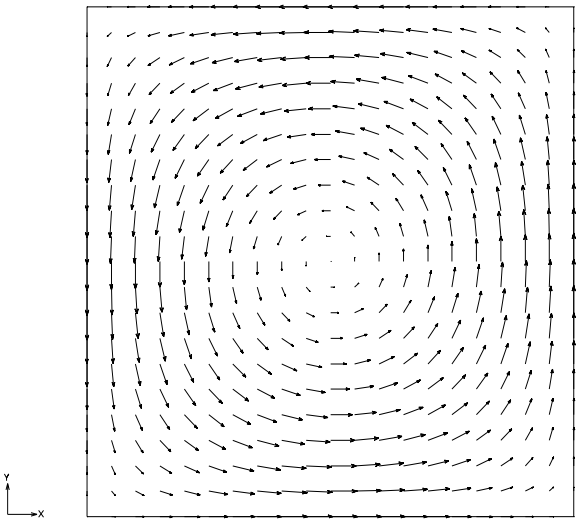

Figure 1: B field computed in test 1.

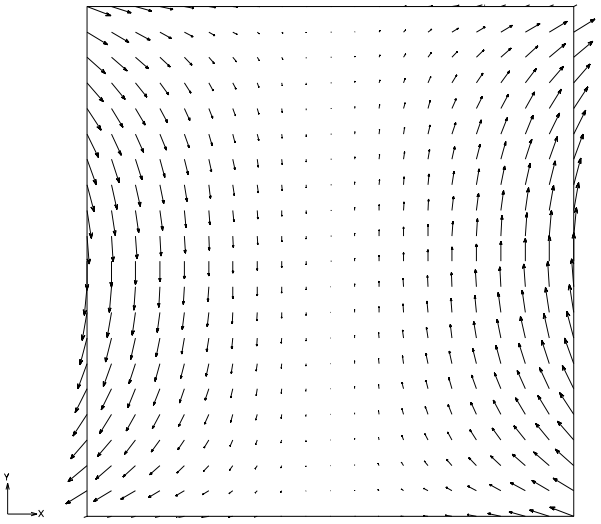

Figure 2: B field computed in test 2.

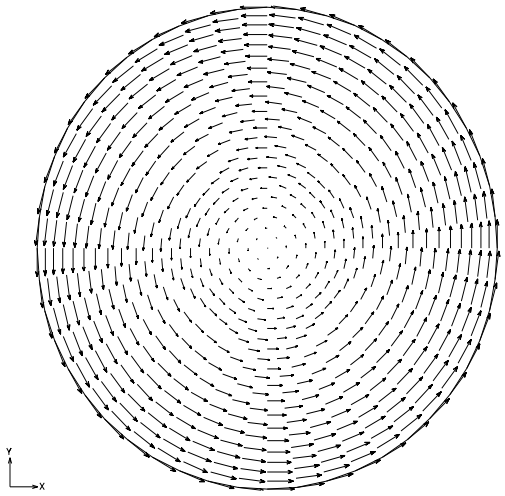

Figure 3: B field computed in test 3.

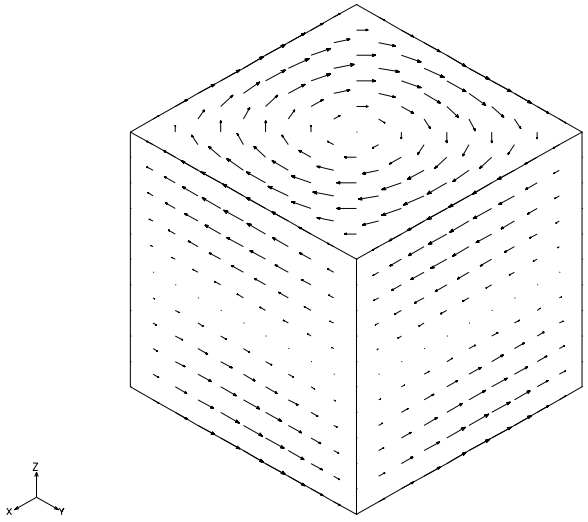

Figure 4: B field computed in test 4.

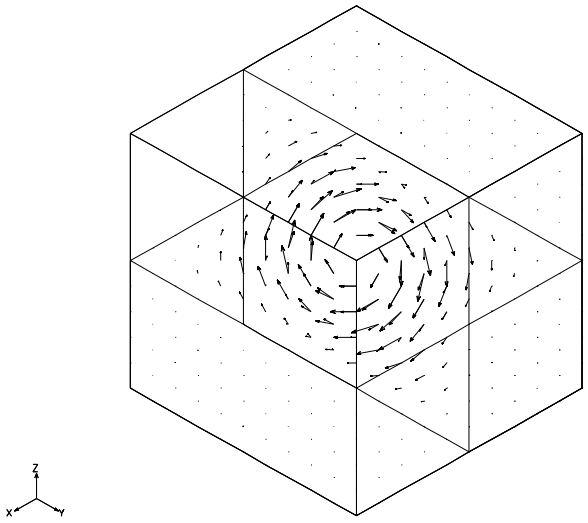

Figure 5: B field computed in test 5.

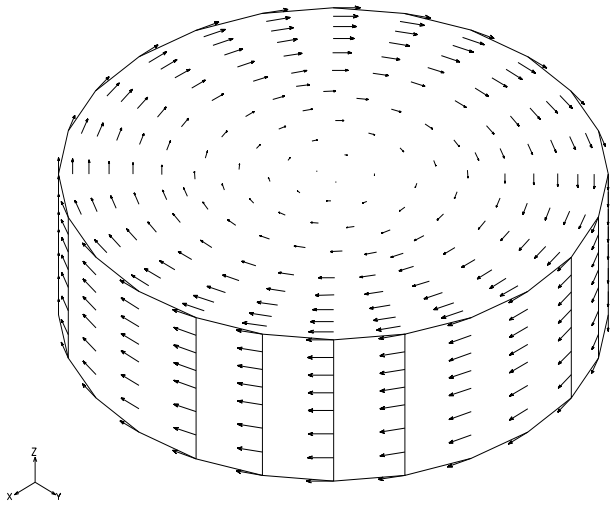

Figure 6: B field computed in test 6.