

Towards a New Framework for Simulating Magnetic Resonance Imaging

Paul Simedrea^{a,b}, Luca Antiga^c and David A. Steinman^{a,b,d}

^aBiomedical Engineering Graduate Program, The University Of Western Ontario;

^bImaging Research Laboratories, Robarts Research Institute, London, Canada;

^cBioengineering Department, Mario Negri Institute, Bergamo, Italy;

^dMechanical & Industrial Engineering, University of Toronto, Toronto, Canada

ABSTRACT

Optimization of magnetic resonance imaging (MRI) protocols in the presence of complex anatomical structures and motions is an ongoing challenge, often calling for the use of simulation environments. Still, MRI simulations are traditionally hampered by the practical difficulties of describing anatomically realistic objects and their motions in a compact but sufficiently resolved manner. Inspired by the powerful finite element method, here we present a novel and flexible approach in which a complex object is described via an unstructured mesh of simpler element shapes, over each of which the MRI signal equation is integrated numerically. To render this approach computationally tractable, a scheme for automatically adapting the numerical integration order to the element geometry and frequency content of the MRI signal is presented.

Keywords: MRI simulation, unstructured mesh, finite elements, vascular imaging

1. INTRODUCTION

Magnetic resonance imaging (MRI) has become one of the leading modalities for non-invasive imaging. However, owing to the many independent parameters available, and the inevitable trade-offs between resolution and scan time, it is often challenging to design robust protocols, especially in the presence of complex anatomical configurations and motions. MRI facility operating time is also costly and limited, thus experimental optimization of MRI protocols to improve image quality is not always feasible. Moreover, it is often difficult to assess the accuracy of MRI-derived measurements (e.g., the segmentation of tissue boundaries) from experimental studies alone. As the physics governing MRI are well understood, a cost-effective alternative is to computationally simulate MR image acquisitions.

Traditionally, MRI simulations have relied on structured mesh approaches—where the object is represented by a regular grid of points at which the MR signal is simulated and summed to produce a virtual MR image.^{1,2} However, in practice this approach is limited to due to the large number of points required to resolve the object. The novel approach investigated here stems from previous work involving representing the target object as an unstructured mesh composed of linear tetrahedra,³ and integrating the MRI signal equation analytically. Still, many such tetrahedra are required to resolve objects having complex, curved boundaries and, in practice, analytic integration requires the use of cumbersome symbolic math software engines to avoid singularities.

To get around these limitations we propose a novel alternative, finite-element MRI (FE-MRI), whereby the signal equation is integrated numerically rather than analytically, thus permitting the use of elements of arbitrary geometry or order. In this paper we demonstrate the functionality of the FE-MRI numerical integration engine. We first characterize the numerical error incurred by integrating the MR signal numerically, then present details of the numerical integration software engine using Gaussian quadrature integration techniques. A scheme is presented to predict the necessary order of integration, and results are presented for the case of a simple unstructured grid geometry.

For further information, please contact D.A.S. at steinman@mie.utoronto.ca or 416-978-7781.

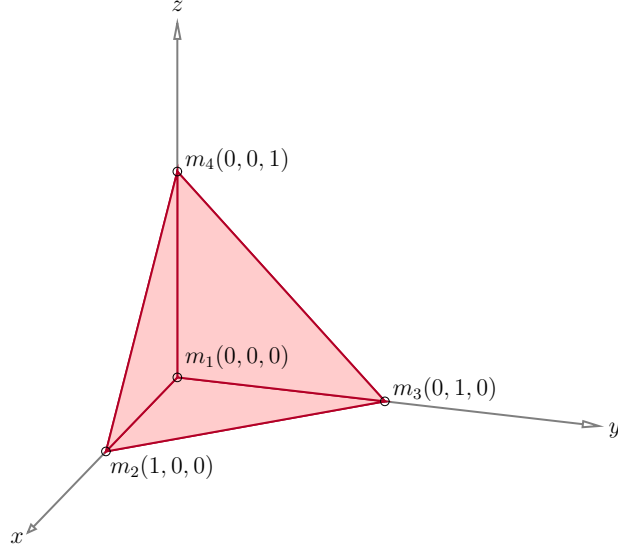


Figure 1. The standard tetrahedral element with magnetization values $m_1 \dots m_4$ at nodes.

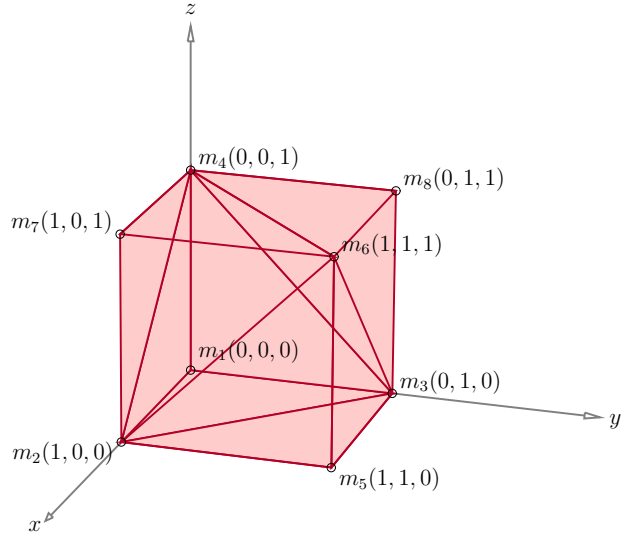


Figure 2. 5-element mesh discretization of the unit cube.

1.1. Signal Equation

Briefly, in MRI magnetic field gradients are manipulated so as to effectively modulate a magnetized object by a sinusoidally varying field. By varying the 3D orientation and frequency of this modulation, an image can be reconstructed by inverse Fourier transform of the MRI signal acquired in this frequency- or k-space. Mathematically, this is embodied in the signal equation:

$$s(t) = \iiint_D m(x, y, z, t) e^{-i2\pi (k_x(t)x + k_y(t)y + k_z(t)z)} dx dy dz, \quad (1)$$

where $D : (x, y, z)$ is the spatial domain, $m(x, y, z)$ defines the instantaneous magnetization of the object, and $k_x(t)$, $k_y(t)$ and $k_z(t)$ define the frequencies and orientations of the modulation. Initially, FE-MRI is strictly designed to efficiently simulate the phase-encode and readout steps given a prescribed magnetization. The magnetization can also be time-varying given that the time scale of the evolving magnetization is greater than the echo time.

The signal $s_e(t)$ for a single linear tetrahedral element (see Figure 1) is then derived to be of the form:

$$s_e(t) = \int_0^1 \int_0^{1-u} \int_0^{1-u-v} m(u, v, w) \exp(-i2\pi [d_0 + d_1 u + d_2 v + d_3 w]) \left| \frac{\partial(x, y, z)}{\partial(u, v, w)} \right| dw dv du \quad (2)$$

where the coefficients $d_0 \dots d_3$ result from the linear transformation from global coordinates (x, y, z) to a reference coordinate system (u, v, w) of the exponent of (1) on the standard tetrahedron. To simplify error analysis we fix $m(x, y, z)$ to 1 at each tetrahedral node. However, in a more realistic situation involving variable magnetization due to multiple tissue types, equation (2) would incorporate an expression for a linear interpolation of $m(x, y, z)$.

1.2. Gaussian Quadrature

In contrast to other numerical integration methods, Gaussian quadrature provides the freedom of choosing the weighting coefficients w_j , as well as the locations of the abscissas x_j , at which function is to be evaluated. It is of the general form given by:

$$\int_a^b W(x) f(x) dx = \sum_{j=1}^N w_j f(x_j) + E \quad (3)$$

The approximate value of the integral is thus found as a summation of the function we wish to evaluate at N discrete abscissas multiplied by the associated N weighting coefficients. In general, if the function is smooth over the integration interval, higher values of N yield lower integration errors E . If $f(x)$ is a polynomial of order $n \leq 2N - 1$, an N -point summation is sufficient to solve the integral of $f(x)$ exactly ($E = 0$).

The technique is also valid in 3D, where the integral is now computed in the boundaries defined by the domain Ω_e :

$$\iiint_{\Omega_e} W(x, y, z) f(x, y, z) dx dy dz = \sum_{j=1}^N w_j f(x_j, y_j, z_j) + E \quad (4)$$

The integration domain in this case is the boundaries of a tetrahedral region. Optimal points and weights for tetrahedral and other 3D domains are commonly known for low orders of integration.* For higher orders, since optimal solutions do not exist, tensor-product rules⁴ providing non-optimal points and weights are useful. Non-optimal quadrature has the drawback of being more computationally intensive than optimal because a greater number of points and weights are used.

2. NUMERICAL SIMULATION APPROACH

2.1. 1D Simulation

Before implementing the Gaussian quadrature simulation technique, it is necessary to quantify the expected numerical error. This is because the integration order is expected to be a function of the modulation frequency, with higher frequencies naturally requiring higher integration orders. For FE-MRI to be computationally efficient, we must be able to predict the minimum number of integration points necessary for a particular modulation frequency. To achieve this, we reduce the problem to one dimension, where we can numerically evaluate a function over a linear 1D element. We can then extend the technique to 3D and evaluate the error in the same fashion.

A 1D equivalent to the MR signal equation is derived from the general form of the MR signal equation, which states that the MR signal is the sum of nuclear magnetization m over the element of length L , phase-encoded with the time-dependent k -space coordinates $k(t)$. We note that in the one-dimensional case, expression (1) reduces to an integral of the form:

$$s_1(t) = \int_0^L m(x) e^{-i2\pi(k(t)x)} dx \quad (5)$$

To simplify error analysis, we evaluate the integral over the range $0 \dots L$ —along the length of the 1D finite element. As the signal equation is cyclical, the term $k(t)x$ effectively calculates the number of cycles per element length, since k is represented in cycles/unit length and x is a spatial length variable. In other words, the integration order depends not upon the modulation frequency itself, but on the number of sinusoidal cycles present over the length of the element.

The *Gauss-Legendre* quadrature form is simplest to use for approximating expression (5) because the weight function $W(x) = 1$ and the integration weights and abscissas are easily found. Gaussian quadrature rules and methods for obtaining weights and abscissas are discussed in greater detail by literature.^{5,6}

2.2. Cycles/Element Metric

An equivalent to the 1D metric of cycles/element is required for the technique to be extended to 3D. By examining Figure 3 it is apparent that establishing the number of oscillation cycles within a triangular—and consequently a tetrahedral—region exactly is not possible. Arbitrary triangular geometry does not provide adequate bounds for establishing an exact number of cycles within a region. Thus, an approximate but *consistent* method is required in order to determine the number of cycles per triangular or tetrahedral element. In addition, the cycles/element metric must closely obey the following consistency criteria:

1. Invariance with element spatial orientation.

*Encyclopedia of Cubature Formulas, <http://www.cs.kuleuven.ac.be/~nines/research/ecf/ecf.html>, 2004

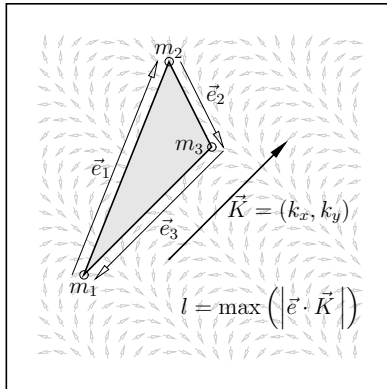


Figure 3. The approximate number of cycles/element is found by first obtaining the dot product of each edge, \vec{e} (length unit), of the triangle with the k -space encoding direction $\vec{K} = (k_x, k_y)$ (cycles/unit length). The maximum dot product absolute value, l (cycles), is considered to represent the number of cycles in a given element. Although shown above for a 2D triangular region, the same idea is true for a 3D tetrahedral region with spatial encoding in the k_x , k_y and k_z directions.

2. Invariance with triangular/tetrahedral element quality (varied shape and size).
3. Invariance with the direction and position of k -space encoding.

The solution is to reduce the problem to one-dimensional space, where the number of cycles along a linear path is easily determined. The frequency and direction of spin oscillations are determined from the k -space coordinates. In 3D MRI acquisitions these are taken to form the vector $\vec{K} = (k_x, k_y, k_z)$. However, for simplicity, we will use 2D MRI acquisition and 2D geometry (triangles instead of tetrahedra). Shown in Figure 3 is a triangular element with nodal magnetizations $\{m_1, m_2, m_3\}$ and vector edges $\{\vec{e}_1, \vec{e}_2, \vec{e}_3\}$. The edges \vec{e} are given units of length, typically cm or mm. The vector $\vec{K} = (k_x, k_y)$ is in units of (cycles/unit length). Hence a dot product between an edge and \vec{K} results in the number of cycles along that edge in the encoding direction. We thus take the number of cycles within a triangular or tetrahedral region to be the maximum value of the dot product between an element's edges and \vec{K} , given in the expression:

$$l = \max \left(\left| \vec{e} \cdot \vec{K} \right| \right) \quad (6)$$

3. RESULTS AND DISCUSSION

3.1. Numerical Simulation Results

To test the functionality of the FE-MRI simulator, images of the unit cube mesh in Figure 2 were generated and compared against the same image generated using the analytic integration method of Truscott & Buonocore.³ The unit cube was chosen here because geometric discretization errors due to meshing are avoided, and thus only numerical integration errors may be highlighted. The images, shown in Figure 4, were generated using the following imaging parameters, assuming the same arbitrary spatial units as the mesh: FOV=2.5x2.5, image size=32x32 pixels, slice thickness = 0.1. Simulated images are shown at quadrature orders of 10, 14, 18 and 22.

The open source `libMesh`[†] finite-element solver library is used to implement the FE-MRI numerical integrator. Its Gaussian quadrature framework provides optimized abscissas and weights inside tetrahedral regions for up to quadrature order 5 and Gauss/Gauss-Jacobi tensor product rules for orders 6 to 22. Although non-optimal, advantages of the tensor product rules are that they provide much higher-order approximations and they can be extended to any order as necessary.

[†]libMesh: <http://libmesh.sourceforge.net>, 2005

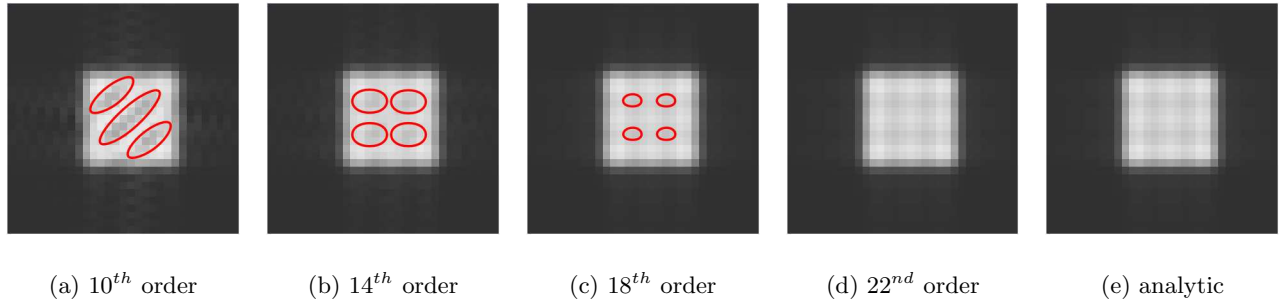


Figure 4. Comparisons of numerical integration over the standard cube shown at quadrature orders 10, 14, 18 and 22, compared with the results of the analytic integration. Most visible numerical errors within the cube are outlined.

Figure 4 shows the effects of numerical integration on image artefacts. Fourier artefacts due to image reconstruction such as ringing inside and on the outside of the object are expected, and are reflected in the analytically simulated image of Figure 4(e). Numerical artefacts are seen in the form of distortions, most obvious at lower orders—see Figures 4(a) and 4(b). Higher orders—see Figures 4(c) and 4(d)—result in more accurate representations of the object. Using a quadrature order of 22 results in a visually indistinguishable image from the “perfect” image, generated via analytic integration. These results demonstrate how the accuracy of the MRI simulation can be controlled merely by increasing the number of integration points, which can be shown to be more computationally efficient than reducing the element size and hence increasing the number of elements. As we have recently demonstrated,⁷ this can be particularly useful for describing complex objects using higher order elements, since fewer elements would obviously be required.

3.2. Error Performance

The performance of absolute numerical error as a function of determined cycles per element is shown in Figure 5 for the 1D simplification. It is shown that as the number of cycles per element increase, a higher order of integration is naturally required to maintain a desired numerical error. The same approach, by using the cycles/element metric described in Section 2.2, was used to characterize the error performance of simulation in linear tetrahedral regions (see Figure 6). For the 3D tetrahedral case a linear k -space trajectory was used starting at the origin and extending diagonally at a step of $\Delta k = 0.1$ to $k_{max} = 30$. This trajectory was chosen to achieve a comparable range of cycles/element to the 1D analysis. The unit tetrahedron in Figure 1 was used as the test element.

The results in Figures 5 and 6 are important in determining the minimum order required for a simulation constrained by imaging parameters. Consequently, this also determines the computational performance of the numerical simulation. As an example, we can consider the imaging of the object of Figure 2, with the same parameters used to acquire the images of Figure 4. We compute the maximum k -space location to be $k_{max} = N/2FOV = 32/2 \cdot 2.5 = 6.4$ cycles/unit length. Since our largest mesh element is of maximum side length $L_{max} = 1$ unit, we find that in the worst case the cycles/element metric would determine 6.4 cycles/element. Thus, using the results of Figure 6, for a desired absolute error of $E = 10^{-3}$, a quadrature order of approximately 20 would be required. Since the results of the 1D and 3D are so similar in trend and use related approaches in determining cycles/element, the 1D error performance data can be used as a more conservative estimate of the required quadrature order.

The practical challenge of numerical integration is using just enough quadrature points to achieve desired accuracy. Based on the data in Figure 6, an optimal algorithm can be implemented. Instead of determining the maximum order required in the worst case (e.g., based on the maximum encoding frequency or k -space location), quadrature order can be determined for each individual element at each k -space location. This approach would allow optimal computational performance while maintaining accuracy. Performance tests using the mesh and imaging parameters outlined in this section indicate we can achieve an approximately five-fold increase in computational speed with the optimal algorithm.

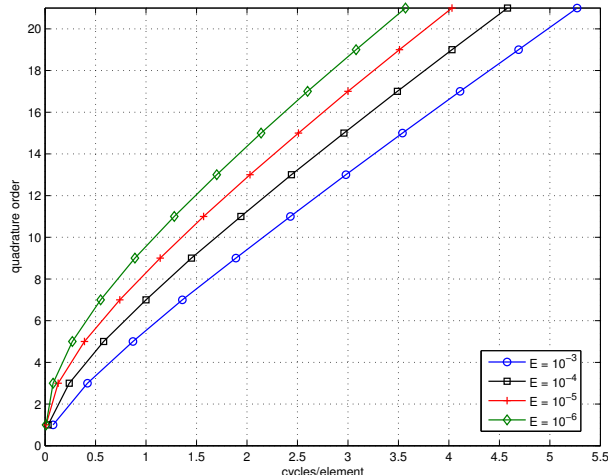


Figure 5. Numerical error performance as a factor of cycles per 1D linear element.

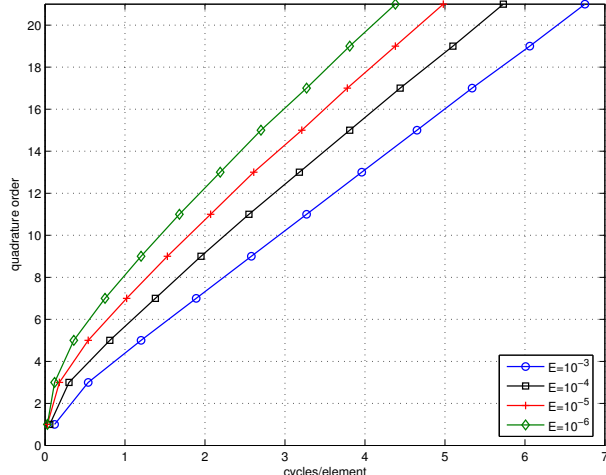


Figure 6. Numerical error performance as a factor of cycles per 3D linear tetrahedral element.

4. CONCLUSIONS

We have demonstrated functionality of the FE-MRI numerical MRI simulator. FE-MRI uses Gaussian quadrature integration within linear tetrahedral regions and is implemented using tensor-product, quadrature rules provided by the libMesh finite-element solver libraries. Error analysis was performed by extending a simplified, 1D, version of the MR signal equation. After determining error trends in 1D, the method was extended to 3D, where similar trends were observed. To achieve this, a metric for describing the number of oscillations within a tetrahedral element was also developed. Results of the numerical simulator were shown and compared with the analytic integration approach in literature. It was found that identical images are produced when using a sufficiently high quadrature order. Finally, an approach for a computationally efficient algorithm was suggested, which uses the error performance data to optimally determine quadrature order on a per-element basis given a location in k -space. Ultimately, we aim to use this approach to develop and apply realistic “computational phantoms” in order to optimize MRI acquisition and post-processing strategies in the presence of complex flow and tissue motions.

REFERENCES

1. D. A. Yoder, Y. Zhao, C. B. Paschal, and J. M. Fitzpatrick, “MRI Simulator With Object-Specific Field Map Calculations,” *Magnetic Resonance Imaging* **22**, pp. 315–328, 2003.
2. H. Benoit-Cattin, G. Collewet, B. Belaroussi, H. Saint-James, and C. Odet, “The SIMRI Project: A Versatile And Interactive MRI Simulator,” *Journal Of Magnetic Resonance* **173**, pp. 97–115, 2005.
3. K. J. Truscott and M. H. Buonocore, “Simulation of Tagged MR Images With Linear Tetrahedral Solid Elements,” *Journal Of Magnetic Resonance Imaging* **14**, pp. 336–340, 2001.
4. G. E. Karniadakis and S. Sherwin, *Spectral/hp Element Methods for Computational Fluid Dynamics*, Oxford Science Publications, 2005.
5. W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C*, Cambridge University Press, New York, Second ed., 1992.
6. P. Davis and P. Rabinowitz, *Methods of Numerical Integration*, Academic Press, New York, 1975.
7. P. Simedrea, L. Antiga, and D. A. Steinman, “FE-MRI: Simulation of MRI Using Arbitrary Finite Elements,” *Fourteenth Scientific Meeting and Exhibition for the International Society of Magnetic Resonance in Medicine (Submitted, November 2005)*, May 2006.
8. M. Abramowitz and I. A. Stegun, *Handbook of Mathematical Functions*, Dover, New York, 1972.