# 18.336/6.335 Spring 2013
# Fast Methods for Partial Differential and Integral Equations
# Problem Set 3

Handed out: April 8, 2013, Due April 19, 2013

## High-Frequency Magnetic Resonance Imaging Continued

### Problem Statement

Continuing our example from problem set 1, recall that with certain assumptions, the governing equations for MRI scattering can be reduced to the Helmholtz equation in non-conductive media:

$$\Delta u(\mathbf{r}) + k^2(\mathbf{r}) \cdot u(\mathbf{r}) = v(\mathbf{r}) \tag{1}$$

where $u(\mathbf{r})$ is the electric field strength in volts per meter, $v(\mathbf{r})$ is the electric excitation, $k(\mathbf{r})$ is the wave number, and $\mathbf{r} = (x, y, z)$ is the vector defining the location of the observation point. Note that $k(\mathbf{r})$ has the following relationship with the other constants of electromagnetics:

$$k^2(\mathbf{r}) = \omega^2 \mu \epsilon(\mathbf{r})$$

where $\omega = 2\pi f$ is the angular frequency in radians per second, $\mu$ is the magnetic permeability in henries per meter, and $\epsilon(\mathbf{r})$ is the electric permittivity in farads per meter, as a function of the location vector $\mathbf{r}$.

Integral equations are ideal for MRI, because they can readily model the radiation boundary condition without the need to artificially simulate it using a large but finite boundary. Additionally, only the regions of interest need to be modeled, and the remaining regions can be implicitly considered using the appropriate Green's function.

In following section, we will introduce the preconditioned inhomogenous Helmholtz equation, known to be particularly efficient when solved using an integral method. The formulation is realistic, and is commonly used in medical imaging applications. Let $k_0$ be the constant free-space wave number with $k_0^2 = \omega^2 \mu_0 \epsilon_0$. Adding and subtracting this factor from $k^2(\mathbf{r})$ yields:

$$\Delta u(\mathbf{r}) + \left[k^2(\mathbf{r}) - k_0^2 + k_0^2\right] \cdot u(\mathbf{r}) = v(\mathbf{r}). \tag{2}$$

Let $\chi(\mathbf{r})$ be the contrast function:

$$\chi(\mathbf{r}) = (k^2(\mathbf{r}) - k_0^2) = \omega^2 \mu_0 \epsilon_0 [\epsilon_r(\mathbf{r}) - 1],$$

then with some simple rearranging we can separate (2) into two components: one due to the homogenous form of the Helmholtz equation, and the other due to the heterogeneity introduced by the contrast function:

$$\left[\Delta u(\mathbf{r}) + k_0^2 u(\mathbf{r})\right] + \chi(\mathbf{r}) u(\mathbf{r}) = v(\mathbf{r}). \tag{3}$$

Let the Green's function $G(\mathbf{r} - \mathbf{r}')$ be the solution to the homogenous Helmholtz equation

$$\Delta G(\mathbf{r}) + k_0^2 G(\mathbf{r}) = \delta(\mathbf{r} - \mathbf{r}'),$$

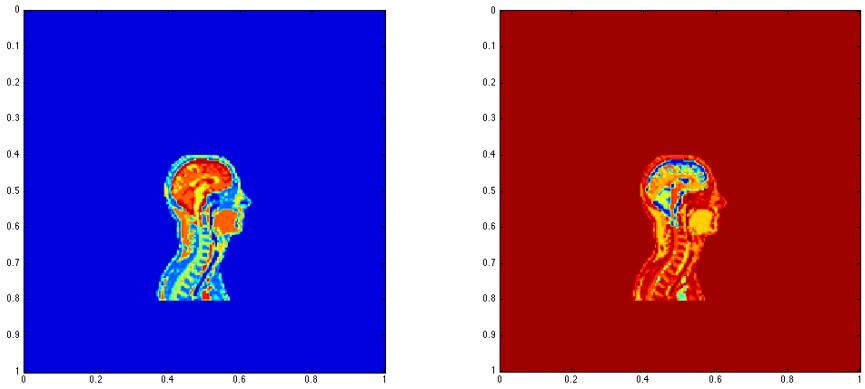

Figure 1: A plot of the $\epsilon_r = \epsilon/\epsilon_0$ provided, generated using the MATLAB code provided. (left) $\mathfrak{Re}\{\epsilon_r\}$; (right) $\mathfrak{Im}\{\epsilon_r\}$.

with the Sommerfeld (outgoing) radiation boundary condition. In three dimensions, it has the form

$$G(\mathbf{r}-\mathbf{r}') = \frac{e^{ik_0\|\mathbf{r}-\mathbf{r}'\|}}{4\pi\|\mathbf{r}-\mathbf{r}'\|}.$$

A convolution with $G$ inverts the homogenous component back to $u(\mathbf{r})$:

$$\int \left[\Delta u(\mathbf{r}') + k_0^2 u(\mathbf{r}')\right] G(\mathbf{r}-\mathbf{r}')\,\mathrm{d}\mathbf{r}' = u(\mathbf{r}).$$

A volume integral equation is thus derived by taking the convolution of (3) with $G$:

$$u(\mathbf{r}) + \int \chi(\mathbf{r}')u(\mathbf{r}')G(\mathbf{r}-\mathbf{r}')\,\mathrm{d}\mathbf{r}' = \int v(\mathbf{r}')G(\mathbf{r}-\mathbf{r}')\,\mathrm{d}\mathbf{r}'. \tag{4}$$

## Question 1 - An Integral Equation formulation

In this question, we will discretize (4) using a Galerkin formulation to form the following system:

$$Ax = b \tag{5}$$

where $x \in \mathbb{R}^{N\times 1}$ is the vector of unknowns, $b \in \mathbb{R}^{N\times 1}$ is the vector of excitations, and $A \in \mathbb{R}^{N\times N}$ is the dense coupling coefficient matrix for (4). To model the head, please reuse the MRI data for $\epsilon_r(\mathbf{r}) = \epsilon(\mathbf{r})/\epsilon_0$ in the file `MRI_DATA.mat`. MATLAB code used to generate the reference x and y vectors and the plots shown in Figure 1 are provided below for your reference.

```
load('MRI_DATA');
x = linspace(0,1,257); y = linspace(0,1,257);
subplot(121); imagesc(x,y,real(e_r)); axis image;
subplot(122); imagesc(x,y,imag(e_r)); axis image;
```

Consider the grid created by importing `MRI_DATA.mat`, which contains $N = 256^2$ flat, square and constant-value "pixels", with each side having a length $h = 1/256$ meters. Each pixel can be used as the basis and testing functions in a Galerkin discretization. Let $\psi_j(\mathbf{r})$ denote the $j$-th basis function evaluated at $\mathbf{r}$; both $u(\mathbf{r})$ and $\chi(\mathbf{r})$ can be represented as as a linear combination of the $N$ basis functions:

$$u(\mathbf{r}) = \sum_{j=1}^{N} u_j\psi_j(\mathbf{r}), \quad \chi(\mathbf{r}) = \sum_{j=1}^{N} \chi_j\psi_j(\mathbf{r}).$$

This expression for $u$ substitutes into (4) to yield:

$$u_j\psi_j(\mathbf{r}) + u_j\chi_j \int G(\mathbf{r}-\mathbf{r}')\psi_j(\mathbf{r})\,\mathrm{d}\mathbf{r}' = \int v(\mathbf{r}')G(\mathbf{r}-\mathbf{r}')\,\mathrm{d}\mathbf{r}'.$$

Clearly, the solution function $u(\mathbf{r})$ appears in the matrix equation (5) as the vector of unknowns $x = \begin{bmatrix} u_1 & u_2 & \cdots & u_N \end{bmatrix}^T$.

1. Let $\sigma_i(\mathbf{r})$ denote the $i$-th testing function evaluated at $\mathbf{r}$. Under a Galerkin discretization scheme, the testing functions are the same as the basis functions, i.e., $\sigma_i(\mathbf{r}) \equiv \psi_i(\mathbf{r})$.

   (a) Write down an integral that can be used to evaluate $A_{ij}$, the element in the $i$-th row and $j$-th column of the coefficient matrix $A$. (2 pts)

   (b) What does $A_{ij}$ evaluate to in regions where $\chi_j = 0$? (1 pt)

2. Let $p(\mathbf{r})$ denote the excitation on the right-hand side of (4):

$$p(\mathbf{r}) = \int v(\mathbf{r}')G(\mathbf{r}-\mathbf{r}')\,\mathrm{d}\mathbf{r}'.$$

   Let us set $v(\mathbf{r})$ to an impulse located at $x = 0.6$ m and $y = 0.7$ m, as previously done in Problem Set 1.

   (a) What is the analytical form of $p(\mathbf{r})$? Write down an integral that can be used to evaluate $b_i$, the element in the $i$-th row of the excitation matrix. (2 pts)

   It is worth appreciating the ease with which integral equations can handle impulses and singularities. Notice that the location where $p(\mathbf{r})$ becomes singular is outside of our region of interest (i.e. the head).

3. We have provided two functions, `DEMCEM_ST`[1] and `nwspgr`[2]. The first function calculates the following integral:

$$\text{DEMCEM\_ST}(k;h) = \int_0^h \int_0^h \int_0^h \int_0^h G(k;\sqrt{(x-x')^2+(y-y')^2})\,\mathrm{d}x'\,\mathrm{d}y'\,\mathrm{d}x\,\mathrm{d}y$$

   The second function generates multidimensional quadrature points via the sparse grid method. Some demo code is shown below:

```
k0 = 2*pi*f/299792458; % wavenumber k0 = omega / c
st = DEMCEM_ST(k0,h); % h is the size of the discretization

% Quadrature for some 4-dimensional function "func"
[quad_nodes,quad_weights] = nwspgr('KPU', 4, 3); % 4-dimensional quad of order 3.
quad_ans = quad_weights'*func(quad_nodes(:,1),quad_nodes(:,2),...
                              quad_nodes(:,3),quad_nodes(:,4));
```

   (a) Describe which matrix elements of $A$ can be computed via `DEMCEM_ST` (1 pts)

   (b) Using the coefficients generated by `DEMCEM_ST` and `nwspgr`, produce code that will compute any element $A_{ij}$ in the coefficient matrix $A$ for given $i$ and $j$. Measure the time it takes to fill a single column of the coefficient matrix $A$. Notice that $A$ has a translation-invariant structure; estimate the time it would take the fill the entire matrix $A$ in case this structure goes unnoticed. (3 pts)

   (c) Describe how the formation of the $A$ matrix can be speeded up by using a low-rank approximation (no need to code it up) (1 pt)

[1] Athanasios G. Polimeridis, Direct Evaluation Method in Computational ElectroMagnetics (DEMCEM), http://web.mit.edu/thanos_p/www/DEMCEM.html.

[2] The function `nwspgr` implements sparse grid quadrature, and was authored by Florian Heiss, Viktor Winschel at http://sparse-grids.de/. The sparse grid quadrature scheme makes it possible to perform numerical integration simultaneously over many dimensions, using a much lower number of quadrature nodes than required by integrating separately over each dimension.

## Question 2 - A Fast Integral Equation Solver

Without leveraging the translation-invariant structure, forming the coefficient matrix along requires the evaluation of $N^2$ 4-dimensional integrals using quadrature. Then, to invert the matrix directly with Gaussian quadrature is of arithmetic complexity $\mathcal{O}(N^3)$.

1. Fortunately, the coupling matrix $A$ has a block-Toeplitz structure that can be exploited to rapidly solve (4) for a given $v(\mathbf{r})$ with $\mathcal{O}(N \log N)$ complexity.

   (a) Describe the steps required to implement the (non-cyclic) convolution implicit in $A$ using the FFT. Include the details of each operation in your description. (2 pts)

   (b) Describe how the multiplier values in the cyclic convolution operation are computed, using the $A_{ij}$ function implemented in Question 1.3b. (2 pts)

   (c) Implement a fast matrix-vector product function that will quickly evaluate the product $Ax$ for a given $x$. Verify that your function is indeed a convolution, by applying it to a few columns of the identity matrix, and demonstrating that the result is translation-invariant to within machine precision. (2 pts)

2. Embed the forward matrix-vector product function from above into GMRES to allow us to quickly solve the integral equation without explicitly forming the dense coupling matrix.

   (a) Solve the scattering problem using GMRES for $f = 21.3$ MHz and $f = 298.3$ MHz, corresponding to the older 0.5 T permanent magnet MRI and the modern 7 T superconductor magnet MRI respectively. Produce iteration-residual plots at both frequencies, and the scattered image using the code below. (6 pts)

```
sol = gmres(A,b,[],[],maxit); % solve matrix equation.
sol(real(e_r) <2)=0; % crop solution to the head.
new_x = linspace(0.35,0.65,100); new_y = linspace(0.4,0.8,100);
[gx,gy]=meshgrid(new_x,new_y); sol = interp2(x,y,sol,gx,gy,'cubic');
figure; imagesc(new_x,new_y,abs(sol)); axis image;
```

   (b) Why do the scattering images from this Problem Set differ to those obtained using finite differences? (1 pt)

   (c) Why does this formulation require a relatively small number of GMRES iterations? (1 pt)