EPJ manuscript No.
(will be inserted by the editor)

A fast numerical scheme for computing the response of composites using grid refinement

D. J. Eyre and G. W. Milton

Department of Mathematics, University of Utah Salt Lake City, UT 84112, e-mail: *lastname*@math.utah.edu

January 26, 1999

Abstract. A numerical scheme is presented to compute the response of a composite material to an applied field. The scheme is based on an iteration that uses a Fourier transform based projection method to compute the response of a reference material to the applied field, and a contraction mapping to relate the response of the reference material to the response of the composite material. The scheme also utilizes nested uniform grids to improve performance, and to evaluate the refinement of the grid.

Key words. numerical – composites – grid refinement

PACS. 7 2.80.-r 72.80.Tm

1 Introduction

We consider the numerical solution of the problem

$$\nabla \times E(\mathbf{x}) = 0, \quad \nabla \cdot J(\mathbf{x}) = 0, \quad J(\mathbf{x}) = L(\mathbf{x})E(\mathbf{x}), \quad (1)$$

with a given applied field $E_0 = \langle E(\mathbf{x}) \rangle$, defined in a rectangular domain $\Omega \subset \mathbf{R}^2$, and with periodic boundary conditions. The problem models the local and overall responses of a composite material to an applied external field. The properties of the composite material are defined in the constitutive tensor $L(\mathbf{x})$, and we want to find the fields $E(\mathbf{x})$ and $J(\mathbf{x})$ within the composite material. The domain we work with is chosen as a subsample of the material that is representative of the entire sample.

The numerical methods we present are extensions of a numerical scheme due to Moulinec and Suquet [23]. Our methods improve the performance of the original scheme with an improved Fourier transform based method to solve the problem on a given grid, and a scheme to utilize nested regular grids. The scheme we will present naturally extends to nonlinear problems as well. Methods similar to the method of Moulinec and Suquet can be found in [22, 24].

Numerous other methods have been proposed for solving the problem discussed here. These include finite element methods [17,16,15], integral equation and fast multipole methods [5,6,12,15,9,25,13,14], Fourier approaches [19,27,1], diffusion methods [26,2,18] and Monte Carlo approaches [20].

2 A related problem

To solve the desired equations, we consider a related problem with a constant reference tensor L_0: given a field $P(x)$, and a constant field E_0, find $E(x)$ which solves

$$\nabla \times E(\mathbf{x}) = 0, \quad \nabla \cdot J(\mathbf{x}) = 0, \quad \langle E(x) \rangle = E_0 \quad (2)$$

with

$$J(\mathbf{x}) = L_0 E(\mathbf{x}) + P(\mathbf{x}). \quad (3)$$

If the polarization field were known, the related problem could be solved using the periodic Green's function $\Gamma(\mathbf{x})$ as

$$E(\mathbf{x}) = -\Gamma P + E_0. \quad (4)$$

where ΓP represents the action of the operator Γ on the polarization field. The operator Γ is easily expressed and computed in Fourier space. For the conductivity problem, the Fourier representation of Γ is

$$\hat{\Gamma}(\mathbf{k}) = \frac{\mathbf{k} \otimes \mathbf{k}}{\mathbf{k} \cdot L_0 \mathbf{k}} \quad (5)$$

where $\mathbf{k}$ is the wave vector. The null space of $\Gamma(\mathbf{x})$ consists of all divergence-free vector fields.

3 A solution to the problem

By subtracting the constitutive equations of the model problem and its related problem, we find that the polarization field satisfies

$$P(\mathbf{x}) = (L(\mathbf{x}) - L_0)\, E(\mathbf{x}). \quad (6)$$

Therefore, from the previous expression for $E(\mathbf{x})$, we find that

$$E(\mathbf{x}) - \Gamma(\mathbf{x})\,(L_0 - L(\mathbf{x}))\, E(\mathbf{x}) = E_0 \quad (7)$$

which is a linear integral equation for $E(\mathbf{x})$. The paper of Moulinec and Suquet uses this equation as the basis for a numerical scheme in which they invert the integral operator using a truncation of the series expansion

$$E(\mathbf{x}) = \sum_{j=0}^{\infty} (\Gamma \delta L)^j \, E_0 \tag{8}$$

where $\delta L(\mathbf{x}) = L_0 - L(\mathbf{x})$ and $(\Gamma \delta L)^j$ represents j consecutive applications of the operator to E_0. This expansion, known as a Neumann expansion, is an expression of the inverse of a linear operator of the form $I - A$ as

$$(I - A)^{-1} = I + A + A^2 + A^3 + \cdots$$

and this series converges if $\|A\| < 1$.

To compute the expansion to m terms, it is useful to notice that if

$$E^m(\mathbf{x}) = \sum_{j=0}^{m} (\Gamma \delta L)^j \, E_0 \tag{9}$$

then

$$\begin{aligned} E^m(\mathbf{x}) &= E_0 + (\Gamma \delta L)\, E^{m-1}(\mathbf{x}) && (10)\\ &= E^{m-1}(\mathbf{x}) - (\Gamma L)\, E^{m-1}(\mathbf{x}). && (11) \end{aligned}$$

The convergence of this series has been analyzed in a number of papers, notably in [8,3,4]. If $L(\mathbf{x})$ is self adjoint and has eigenvalues so that $\alpha I \le L(\mathbf{x}) \le \beta I$, then this scheme converges optimally when $L_0 = (\alpha + \beta)/2$ and the rate of convergence is given by $\|E^m - E^{m-1}\| = \gamma^m \|E^0\|$ where

$$\gamma = \frac{\beta/\alpha - 1}{\beta/\alpha + 1}.$$

Notice that γ is less than 1, and so the scheme will converge for any choice of the initial guess, but the rate of convergence is slow if the contrast ratio of the material, β/α, is large.

Our interest in this scheme stems from its capability of directly solving nonlinear problems. While this paper primarily describes the scheme applied to linear problems, we briefly consider the problem

$$\nabla \times E(\mathbf{x}) = 0, \quad \nabla \cdot J(\mathbf{x}) = 0 \tag{12}$$

with the constitutive relation

$$J = F\,(E, \mathbf{x}) \tag{13}$$

where F is a monotone nonlinear term in its first argument. Using the related problem with the constant reference tensor, we find that the solution has the representation

$$E(\mathbf{x}) - \Gamma \mathcal{F}(E, \mathbf{x}) = E_0 \tag{14}$$

where $\mathcal{F}(E, \mathbf{x}) = F(E, \mathbf{x}) - L_0 E$. This is a nonlinear integral equation for $E(\mathbf{x})$ that can also be solved using a Neumann expansion.

4 The basic numerical scheme

The following numerical scheme is due to Moulinec and Suquet [23]. In two spatial dimensions, the representative sample of the material is divided into $N_1 \times N_2$ grid points that are equally spaced in each grid direction. Let $\mathbf{x}_d$ be the discrete points on the grid, then the tensor $L(\mathbf{x}_d)$ must be determined at each grid point. This may be accomplished by scanning an image of the sample, identifying the materials in scanned image, and using experimentally obtained values for the tensor from the materials in the sample.

Having determined $L(\mathbf{x}_d)$, the scheme exploits the observation that the action of L is local in real space and the action of Γ is local in Fourier space.

Algorithm 1: Set $E^0(\mathbf{x}_d) = E_0$, $J^0(\mathbf{x}_d) = L(\mathbf{x}_d) E^0(\mathbf{x}_d)$, $m = 1$ and iterate

- Using the FFT, compute $\hat{J}^{m-1}(\mathbf{k})$
- Stop if $\|\mathbf{k} \cdot \hat{J}^{m-1}(\mathbf{k})\| \le \varepsilon$
- Otherwise, set $\hat{E}^m(0) = E_0$, and

$$\forall\, \mathbf{k} \ne 0, \text{ set } \hat{E}^m(\mathbf{k}) = \hat{E}^{m-1}(\mathbf{k}) - \hat{\Gamma} \hat{J}^{m-1}(\mathbf{k})$$

- Using the inverse FFT, compute $E^m(\mathbf{x}_d)$
- Compute $J^m(\mathbf{x}_d) = L(\mathbf{x}_d)\, E^m(\mathbf{x}_d)$
- Set $m = m + 1$ and repeat

Notice that the term $\hat{E}^m(0)$ is treated differently. This is required to ensure that the computed field $E(\mathbf{x}_d)$ has the correct average value because the operator Γ projects constant fields to zero.

5 A faster converging scheme

Golden and Milton [10] found that the solution could be rewritten using the polarization field, $P(\mathbf{x}) = (L(\mathbf{x}) - L_0) E(\mathbf{x})$, in the form

$$P = -2\, (I - \Upsilon z)^{-1}\, L_0 E_0 \tag{15}$$

where

$$\Upsilon = 2 L_0 \Gamma - I, \quad z = (L_0 + L)(L_0 - L)^{-1}. \tag{16}$$

In this case, the inverse operator $(I - \Upsilon z)^{-1}$ is computed as an expansion of the form

$$(I - \Upsilon z)^{-1} = \sum_{j=0}^{\infty} (\Upsilon z)^j \tag{17}$$

and with an appropriate choice of the reference tensor, this series converges faster than the previous series. In particular, Milton [21] has shown that if $L(\mathbf{x})$ is self adjoint and has eigenvalues so that $\alpha I \le L(\mathbf{x}) \le \beta I$, then this scheme converges optimally when $L_0 = -\sqrt{\alpha\beta}$ and the rate of convergence is given by $\|E^m - E^{m-1}\| = \gamma^m \|E^0\|$ where

$$\gamma = \frac{\sqrt{\beta/\alpha} - 1}{\sqrt{\beta/\alpha} + 1}.$$

Notice again that γ is less than 1, and so the scheme will converge, and that the rate of convergence will be significantly faster than the previous scheme for high contrast materials. Similar expansions are important in the theory of exact relations for composites [11].

To use this scheme in a numerical algorithm, the material is discretized, and the iterative scheme where $P^{m+1} = \Upsilon z P^m - 2L_0 E_0$ is utilized as follows.

Algorithm 2: Set $P^0 = -2L_0 E_0$, $m = 1$ and iterate

- Compute $Q^m(\mathbf{x}_d) = z(\mathbf{x}_d) P^{m-1}(\mathbf{x}_d)$
- Using the FFT, compute $\hat{Q}^m(\mathbf{k})$.
- Set $\hat{P}^m(0) = P^0$, and

$$\forall\, \mathbf{k} \neq 0, \text{ set } \hat{P}^m(\mathbf{k}) = \hat{\Upsilon}\hat{Q}^{m-1}(\mathbf{k})$$

- Using the inverse FFT, compute $P^m(\mathbf{x}_d)$.
- If $\|P^m(\mathbf{x}_d) - P^{m-1}(\mathbf{x}_d)\| \leq \varepsilon$, set

$$E(\mathbf{x}_d) = (L(\mathbf{x}_d) - L_0)^{-1} P^m(\mathbf{x}_d), \text{ and}$$

$$J(\mathbf{x}_d) = L(\mathbf{x}_d)E(\mathbf{x}_d) \text{ and stop.}$$

- Otherwise, set $m = m + 1$ and repeat

This scheme compares successive polarization fields to determine whether the fields have converged. This scheme was chosen for two reasons. First, it only requires the computation of the E and J fields once, and therefore, is lower cost. Second, $\|P^m - P^{m-1}\|$ is a direct measurement of the error that is incurred by taking $E = E^m = (L - L_0)^{-1} P^m$ since

$$\begin{aligned} E^m &= E_0 - \Gamma P^m \\ &= \frac{-L_0^{-1}}{2}\left(P^m + zP^{m-1}E^m(\mathbf{x})\right) \\ &= (L - L_0)^{-1} P^{m-1} + \frac{1}{2}\left(P^m - P^{m-1}\right). \end{aligned}$$

6 Using nested grids to enhance the scheme

In this section, we introduce a two dimensional grid refinement scheme that enhances the overall convergence of the scheme.

In this algorithm, we initially compute the solution on a coarse grid. This process has a low cost. Then we compute a good initial approximation of the solution on a refined grid using a modified spline interpolation. The interpolation operator is denoted by $\mathcal{I}_c^f$ and is defined below. The interpolants we use are accurate and we force the interpolating functions to satisfy the model problem at the new fine grid points. We then use this initial guess to compute the solution on the finer grid. This algorithm is repeated using a nested sequence of refined spatial grids.

Algorithm 3: Define a coarse grid with $N_1 \times N_2$ equally spaced grid points, and let x_c represent the points on the coarse grid. On the coarse grid, set $P^0 = -2L_0 E_0$, and $m = 1$.

- Find $E(\mathbf{x}_c)$ and $J(\mathbf{x}_c)$ using the iteration
  - Compute $P^m(\mathbf{x}_c) = \Upsilon z P^{m-1}(\mathbf{x}_c) + P^0$
  - If $\|P^m(\mathbf{x}_c) - P^{m-1}(\mathbf{x}_c)\| \leq \varepsilon_1$, set

$$E(\mathbf{x}_c) = (L(\mathbf{x}_c) - L_0)^{-1} P^m(\mathbf{x}_c),$$

$$J(\mathbf{x}_c) = L(\mathbf{x}_c)E(\mathbf{x}_c) \text{ and refine.}$$

  - Otherwise, set $m = m + 1$ and repeat
- Define a refined grid and interpolate the solution to the refined grid
  - Set $N_1 = 2N_1$ and $N_2 = 2N_2$, and let x_f represent the points on the fine grid
  - Define interpolated fields

$$[\mathcal{E}(\mathbf{x}_f), \mathcal{J}(\mathbf{x}_f)] = \mathcal{I}_c^f\, [E(\mathbf{x}_c), J(\mathbf{x}_c)]$$

  - Define $E(\mathbf{x}_f) = \Gamma\mathcal{E}(\mathbf{x}_f)$
- If is sufficiently refined, stop.
- Otherwise, initialize the solution for the next iteration
  - On the refined grid, Set $m = 2$, $P_0 = -2L_0 E_0$, and

$$P^1(\mathbf{x}_f) = (L(\mathbf{x}_f) - L_0)E(\mathbf{x}_f)$$

  - Redefine $x_c = x_f$.
- Repeat.

This scheme gives faster convergence because the interpolation operator gives a good estimation of the solution on the refined grid, and consequently, the scheme requires fewer and fewer iterations to compute the solution on successive grids. Furthermore, the scheme is dominated by the work required on the finest and next finest grids, and it requires only a few iterations on these grids. Therefore we see a net savings in the computational effort.

7 Numerical results

The results of numerical tests using Algorithm 3 will now be presented. First, we solve the model problem with a composite consisting of a periodic array of circular inclusions. The representative sample is chosen to be a region of the matrix that contains one inclusion. The tensor is taken to be a piecewise defined scalar with the conductivity in the inclusion equal to one and the conductivity in the matrix equal to σ. In Figure 1, we present the computational effort needed for Algorithms 1 and 2 to compute the solution for 128^2 computational grid, and we present the effort needed for Algorithm 3 to compute the solution on a nested sequence of grids up to 128^2.

In Figure 2, we present the error estimate $\|P^m(\mathbf{x}_c) - P^{m-1}(\mathbf{x}_c)\|$ as a function of the iteration count and the current mesh. Notice that the interpolated solutions are successively better estimate of the solution on the finer grids, and therefore, that the number of iterates required to meet the fixed convergence criteria decreases.

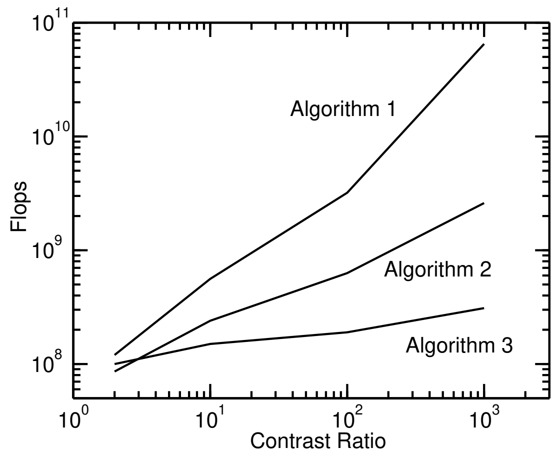

Fig. 1. The computation effort required to compute the electrical field as a function of contrast ratio for the three algorithms.

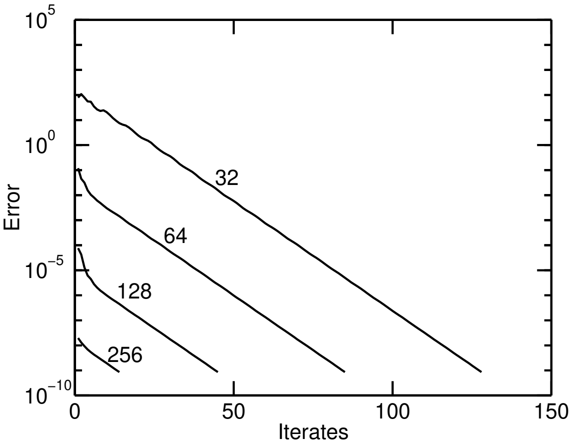

Fig. 2. The iterations required to compute the solution to a fixed tolerance on the nested grids defined in Algorithm 3.

8 The coarse to fine interpolation operator

The coarse to fine spline interpolation operator $\mathcal{I}_c^f$ takes the solution on a coarse grid and computes an approximate solution on the fine grid, i.e.

$$[\mathcal{E}(\mathbf{x}_f), \mathcal{J}(\mathbf{x}_f)] = \mathcal{I}_c^f \left[E(\mathbf{x}_c), J(\mathbf{x}_c)\right].$$

To define the operators, it is useful to have precise definitions of the two grids. The coarse grid points are

$$\mathbf{x}_c(i_1, i_2) = \left(\frac{i_1 L_1}{N_1}, \frac{i_2 L_2}{N_2}\right),$$

where $i_1 = 0, \ldots, N_1 - 1$, and $i_2 = 0, \ldots, N_2 - 1$. The fine grid points are

$$\mathbf{x}_f(j_1, j_2) = \left(\frac{j_1 L_1}{2N_1}, \frac{j_2 L_2}{2N_2}\right),$$

where $j_1 = 0, \ldots, 2N_1 - 1$, and $j_2 = 0, \ldots, 2N_2 - 1$. Notice that $\mathbf{x}_c(i_1, i_2) = \mathbf{x}_f(2i_1, 2i_2)$, and that the fine grid has exactly four times as many points.

To define $\mathcal{I}_c^f$, we compute approximate fields from data defined on every rectangular cell on the coarse grid. These cells have corners at points in the set

$$X_c = \{\mathbf{x}_c(i_1, i_2), \mathbf{x}_c(i_1 + 1, i_2), \mathbf{x}_c(i_1 + 1, i_2 + 1), \mathbf{x}_c(i_1, i_2 + 1)\}$$

and these points correspond to the fine grid points in the set

$$X_f = \{\mathbf{x}_f(2i_1, 2i_2), \mathbf{x}_f(2i_1 + 2, 2i_2), \mathbf{x}_f(2i_1 + 2, 2i_2 + 2), \mathbf{x}_f(2i_1, 2i_2 + 2)\}.$$

Within each cell, we define the interpolated fields at four points on the fine grid. These points are the lower left corner,

$$\mathbf{x}_{f,c} = \mathbf{x}_f(2i_1, 2i_2),$$

the center of rectangle cell

$$\mathbf{x}_{f,m} = \mathbf{x}_f(2i_1 + 1, 2i_2 + 1),$$

and the fine grid points at the middle of the bottom edge and the left edge of the cell,

$$\mathbf{x}_{f,b} = \mathbf{x}_f(2i_1 + 1, 2i_2) \text{ and } \mathbf{x}_{f,l} = \mathbf{x}_f(2i_1, 2i_2 + 1).$$

We choose these points for convenience only, and the approximations we compute are the same as those we would compute had we associated $\mathbf{x}_{f,c}$, $\mathbf{x}_{f,b}$ or $\mathbf{x}_{f,l}$ with other neighboring cells.

To compute the approximate fields at the fine grid points, we approximate the fields on the cells with modified bi-cubic Hermite polynomials [7]. Hermite polynomials are described in detail in the next section, but briefly, they interpolate function values and gradients at the corners of the cells, i.e. if $\mathbf{x}_f \in X_f$ and if $\mathbf{x}_c$ is a corresponding point in X_c, then

$$\mathcal{E}(\mathbf{x}_f) = E(\mathbf{x}_c), \quad \mathcal{J}(\mathbf{x}_f) = J(\mathbf{x}_c) \tag{18}$$

and

$$\nabla\mathcal{E}(\mathbf{x}_f) = \nabla E(\mathbf{x}_c) \text{ and } \nabla\mathcal{J}(\mathbf{x}_f) = \nabla J(\mathbf{x}_c). \tag{19}$$

The resulting function is globally continuous and once differentiable.

To define approximations to the matrix valued gradients $\nabla E(\mathbf{x}_c)$ and $\nabla J(\mathbf{x}_c)$, the coarse grid FFTs of E and J are differentiated in Fourier space using multiplication by ik where k is the frequency. The inverse FFT is then applied to the result to compute approximate real space derivatives. The derivatives of the interpolated fields are computed by analytically differentiating the fields. This is possible since the interpolated fields are once globally differentiable.

Additionally, we modify the bi-cubic Hermite polynomials by requiring the constitutive law to hold at the three fine grid points $\mathbf{x}_{f,m}$, $\mathbf{x}_{f,b}$ and $\mathbf{x}_{f,l}$, i.e.

$$\mathcal{J}(\mathbf{x}_{f,\xi}) = L(\mathbf{x}_{f,\xi})\,\mathcal{E}(\mathbf{x}_{f,\xi}), \tag{20}$$

and by requiring that the polynomial approximations analytically satisfy the differential constraints of the fields at these points,

$$\nabla \times \mathcal{E}\,|_{\mathbf{x}_{f,\xi}} = 0 \text{ and } \nabla \cdot \mathcal{J}\,|_{\mathbf{x}_{f,\xi}} = 0. \tag{21}$$

These conditions couple the approximate fields to each other, and force the interpolation function to locally satisfy the physics of the problem. These conditions already hold at $\mathbf{x}_{f,c}$ because this point corresponds to a coarse grid point where we have computed a solution.

Our numerical experiments found that including these additional constraints was crucial for improving the performance of this scheme.

Once the interpolating polynomial is defined on X_f, the fields at the four points are defined by evaluating the interpolant at those points. We note that at the common points of the coarse and fine grids, the approximate fields agree with the coarse grid fields

$$\mathcal{E}\left(\mathbf{x}_f(2i_1, 2i_2)\right) = E\left(\mathbf{x}_c(i_1, i_2)\right) \tag{22}$$

and

$$\mathcal{J}(\mathbf{x}_f(2i_1, 2i_2)) = J(\mathbf{x}_c(i_1, i_2)). \tag{23}$$

Additionally, the interpolant guarantees that the approximate fields are accurate to at least $O((L_i/N_i)^3)$.

9 Modified bi-cubic Hermite interpolation

In one dimension, cubic Hermite interpolation functions are piecewise defined C^1 functions that interpolate function values and first derivatives defined at a set of discrete nodes. Between any two nodes, these functions are cubic polynomials. The standard basis functions for these polynomials are defined on $[0,1]$, and are plotted in Figure 3. The basis function $H_0(x)$ is defined to satisfy the four constraints, $H_0(0) = 1$, $H_0'(0) = 0$, $H_0(1) = 0$, and $H_0'(1) = 0$, and so $H_0(x) = 2\left(x + \frac{1}{2}\right)(1-x)^2$. The basis function $H_1(x)$ is defined to satisfy the four constraints, $H_1(0) = 0$, $H_1'(0) = 1$, $H_1(1) = 0$, and $H_1'(1) = 0$. and so $H_1(x) = x(1-x)^2$. The function $-H_2(x)$ and $H_3(x)$ are reflections about $x = 1/2$ of $H_1(x)$ and $H_0(x)$ respectively. Then the Hermite approximation of a function $f(x)$ on the interval $[0,1]$ with the data $f(0)$, $f'(0)$, $f(1)$ and $f'(1)$ is

$$\tilde{f}(x) = f(0)H_0(x) + f'(0)H_1(x) + f'(1)H_2(x) + f(1)H_3(x).$$

To compute the approximation on another interval, one maps that interval to $[0,1]$, computes the interpolant and inverts the map. This allows one to connect intervals, thereby creating the piecewise C^1 interpolant.

To define bi-cubic Hermite polynomials on $[0,1]^2$, one computes tensor products of the one dimensional basis functions. We will write

$$\mathcal{H}_{\alpha\beta}(x_1, x_2) = H_\alpha(x_1)H_\beta(x_2)$$

for the tensor product of $H_\alpha(x_1)$ with $H_\beta(x_2)$. Then for each i ranging over the four vertices, there are suitable

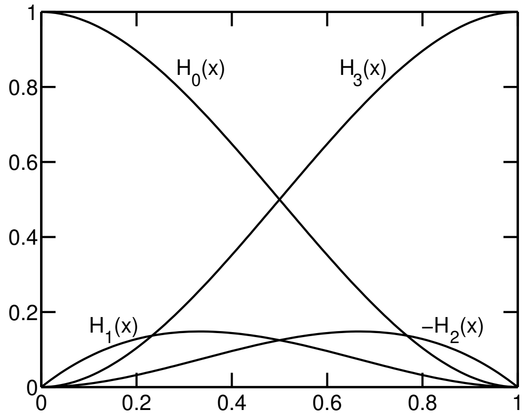

Fig. 3. The one-dimensional Hermite basis functions on the interval $[0,1]$.

values of α and β such that at the specified node $\mathbf{x}_i$ and the other nodes $\mathbf{x}_j$, the function satisfies one of the three sets of conditions

$$\mathcal{H}_{\alpha\beta}(\mathbf{x}_i) = 1,\ \nabla\mathcal{H}_{\alpha\beta}(\mathbf{x}_i) = (0,0),\ \mathcal{H}_{\alpha\beta}(\mathbf{x}_j) = 0 \text{ and}$$
$$\nabla\mathcal{H}_{\alpha\beta}(\mathbf{x}_j) = (0,0),$$

or

$$\mathcal{H}_{\alpha\beta}(\mathbf{x}_i) = 0,\ \nabla\mathcal{H}_{\alpha\beta}(\mathbf{x}_i) = (1,0),\ \mathcal{H}_{\alpha\beta}(\mathbf{x}_j) = 0 \text{ and}$$
$$\nabla\mathcal{H}_{\alpha\beta}(\mathbf{x}_j) = (0,0),$$

or

$$\mathcal{H}_{\alpha\beta}(\mathbf{x}_i) = 0,\ \nabla\mathcal{H}_{\alpha\beta}(\mathbf{x}_i) = (0,1),\ \mathcal{H}_{\alpha\beta}(\mathbf{x}_j) = 0 \text{ and}$$
$$\nabla\mathcal{H}_{\alpha\beta}(\mathbf{x}_j) = (0,0).$$

According to the choice of i and the chosen set of conditions, one generates one of the 12 basis functions. These are the standard bi-cubic Hermite basis functions.

To incorporate the data at the cell centers, it is necessary to include one more basis function for each component of the fields being calculated. These additional basis functions should not disturb the Hermite polynomials at the corners of the cells, i.e. they should be zero and have zero gradient at the corners. The extended Hermite basis functions we chose are tensor products and sums of the three one dimensional functions $H_4(x) = x^3(1-x)^2$, $H_5(x) = x^2(1-x)^3$ and $H_6(x) = x^2(1-x)^2$ shown in Figure 3. Outside of the interval $[0,1]$, these basis functions are set to zero. Notice that at $x = 0$ and $x = 1$, all three of these basis functions are zero and their first derivatives are zero.

From these functions, we use the following functions

$$\begin{aligned}(\mathcal{H}_{44}(x_1,x_2), \mathcal{H}_{55}(x_1,x_2)) = \\ (H_4(x_1)H_4(x_2) + \epsilon H_{66}(x_1,x_2), \\ H_5(x_1)H_5(x_2) - \epsilon H_{66}(x_1,x_2))\end{aligned}$$

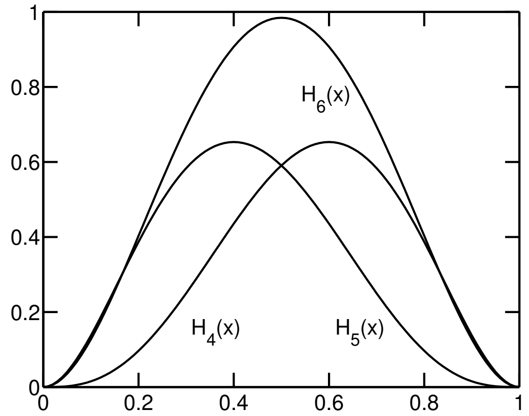

Fig. 4. The one-dimensional modified Hermite basis functions on the interval $[0, 1]$.

as the additional Hermite interpolation basis functions for $E(\mathbf{x})$, and

$$\begin{aligned}(\mathcal{H}_{54}(x_1,x_2),\mathcal{H}_{45}(x_1,x_2)) =\\ (H_5(x_1)H_4(x_2) - \epsilon H_{66}(x_1,x_2),\\ H_4(x_1)H_5(x_2) + \epsilon H_{66}(x_1,x_2))\end{aligned}$$

as the additional Hermite basis functions for $J(\mathbf{x})$. On the cell edges, these basis functions are zero and have zero gradients. We add the slight asymmetry to the basis functions to ensure we can uniquely find the coefficients for these functions, and we choose ϵ to be a small parameter (typically $\epsilon = 0.01$).

To incorporate the data at the cell edge points, it is also necessary to include one more basis function for each component of the fields being calculated, and at each point. On the bottom edge of the cell, we form the additional Hermite interpolation basis functions for $J(\mathbf{x})$ and for $E(\mathbf{x})$ by shifting the Hermite basis functions for the cell center conditions in the x_2 direction, i.e. we use

$$\left(\mathcal{H}_{44}\left(x_1, x_2 + \frac{1}{2}\right), \mathcal{H}_{55}\left(x_1, x_2 + \frac{1}{2}\right)\right)$$

for $E(\mathbf{x})$, and

$$\left(\mathcal{H}_{54}\left(x_1, x_2 + \frac{1}{2}\right), \mathcal{H}_{45}\left(x_1, x_2 + \frac{1}{2}\right)\right)$$

for $J(\mathbf{x})$. On the left edge of the cell, we form basis functions for $E(\mathbf{x})$ and for $J(\mathbf{x})$ by shifting the basis functions for the cell center conditions by $1/2$ in the x_1 direction.

Therefore, to satisfy the 12 constraints at the center and edge points, in total we add three basis functions to the four fields of $J(\mathbf{x})$ and $E(\mathbf{x})$. These added basis functions overlap into neighboring cells, but they are nonzero, and have nonzero first derivatives only at the points where they have been designed to modify the interpolant to satisfy the physics of the problem. Therefore, these basis functions do not overlap any points where the interpolant has already been computed nor do they change the global smoothness properties of the interpolant.

From these basis functions, we write the components of the fields as

$$\mathcal{K}_i(x_1,x_2) = \sum_{(\alpha,\beta)\in A\cup A'} \mathcal{K}_{i,\alpha\beta}\mathcal{H}_{\alpha\beta}(x_1,x_2) \qquad (24)$$

where $\mathcal{K}_i$ represents a component of either $\mathcal{E}$ or $\mathcal{J}$, A is an index set of the standard bi-cubic Hermite polynomials, and A' is an index set of the modified Hermite basis polynomials.

The coefficients $\mathcal{K}_{i,\alpha\beta}$ where $(\alpha,\beta) \in A$ are the field values and first partial derivatives of the field being interpolated. The coefficients of the twelve additional terms where $(\alpha,\beta) \in A'$ are found by solving three 4×4 linear systems of equations, one at each fine grid point. These equations are found by evaluating the constraints (20) and (21) using the approximate solution (24). Because the basis functions are zero and have a zero derivatives at the other fine grid points, the three sets of linear equations are independent of one another, and their dependence on the standard bi-cubic Hermite polynomials only changes the right hand side of these equations. The asymmetry of the basis functions guarantees that the equations of these systems are linear independent.

Furthermore, because the additional constraints on the cell edges do not depend on data within either of their neighboring cells, and because the interpolant is C^1, the interpolated values at the edge points are independent of which neighboring cell we choose to use in order to define those values.

Finally we note that this interpolant is inexpensive to compute. The most costly requirement is that we must compute derivatives of the fields on the coarse grid. The expense of this is one or two extra FFTs per field, or roughly the same amount of work as one iteration of solution algorithm. Finding and evaluating the interpolating polynomials is an $O(N)$ process, and therefore, is negligible.

The authors would like to thank Pierre Suquet and Hervé Moulinec for their helpful suggestions and support. This work was supported by the General Motors Corporation and the National Science Foundation through grants DMS-9629692, DMS-9501025, DMS-9402763 and DMS-99?????.

References

1. D.J. Bergman and K-J Dunn, Phys. Rev. B, **45**, (1992), 13262
2. R.T. Bonnecaze and J.F. Brady, Proc. R. Soc. Lond. A, **432**, (1991), 445
3. O. Bruno, Asym. Anal., **4**, (1991), 339
4. O. Bruno and F. Reitich, Math. Comp., **63**, (1994), 195
5. H. Cheng, L. Greengard, and V. Rokhlin, preprint, (1998).
6. H. Cheng and L. Greengard, SIAM J. Appl. Math. **58**, (1998), 122
7. C. de Boor, *A Practical Guide to Splines*, (Springer-Verlag, New York, NY
8. A.G. Fokin, Phys. Stat. Sol.(b), **111**, (1982), 281

9. Y. Fu, K.J. Klimkowski, G.J. Rodin, E. Berger, J.C. Browne, J.K. Singer, R.A. van de Geijn and K.S. Vemaganti, preprint, (1998)
10. K. Golden and G.W. Milton, Comm. Pure Appl. Math., **43**, (1990), 647
11. Y. Grabovsky, G.W. Milton, D.S. Sage, submitted Comm. Pure Appl. Math., (1998)
12. L. Greengard and M. Moura, Acta Numerica 1994, Cambridge University Press, pp. 379.
13. J. Helsing, J. Stat. Phys., **90**, (1998), 1461
14. J. Helsing, J. Comp. Phys., **127**, (1996), 142
15. J. Helsing and K. Samuelsson, J. Appl. Phys., **78**, (1995), 2498–2503,
16. T. Y. Hou and X-H Wu, J. Comput Phys, **134**, (1997), 169,
17. T. Y. Hou, X-H Wu and Z. Cai, Math. Comput., to appear.
18. I.C. Kim and S. Torquato, J. Appl. Phys., **68** (1990), 3892
19. H. Ma and P. Sheng, preprint.
20. D.R. McKenzie, R.C. McPhedran and G.H. Derrick, Proc. R. Soc. Lond. A, **362**, (1978), 221
21. G.W. Milton, in preparation.
22. H. Moulinec and P. Suquet, C.R. Acad. Sc. Paris II, **318**, (1994), 1417
23. H. Moulinec and P. Suquet, Comp. Meth. Appl. Mech. Engrg., **157**, (1998), 69
24. W.H. Muller, J. Phys. IV, **6**, (1996), C1-139
25. G.J. Rodin, Int. J. Sol. Struct., **30**, (1993), 1894
26. L.M. Schwartz and J.R. Banavar, Physica A, **157**, (1989), 230
27. R. Tao, Z. Chen and P. Sheng, Phys. Rev. B4l, (1990), 2417