Fast interpolation method for field evaluation in a periodic unit cell
Shaojing Li and Vitaliy Lomakin
Dept. of ECE, UCSD, La Jolla, CA 92093; email: sli@ucsd.edu, vitaliy@ece.ucsd.edu

1. Introduction

Iterative integral equation (IE) solvers, which are widely used to analyze problems in a unit cell of an (infinitely) periodic structure, often are slow due to the need to evaluate the periodic Green’s function (PGF) and its convolution with an extended source. FFT-based methods [1] speed up calculations but are suboptimal for general surfaces problems and require $O(N)$ time consuming PGF evaluations, for $N$ sources and observers. Fast Multipole Methods (FMMs) for statics exist [2]. However, electromagnetic periodic hierarchical-type approaches are more challenging and only one such (FMM) extension for the specific case of doubly periodic free-space problems was very recently presented [3]. Fast hierarchical methods for IE solvers for general periodic problems are yet to be developed with many applications in microwave engineering and optics.

In this paper, we present a fast interpolation method (FIM) to compute fields in a unit cell of general periodic problems with periodicities smaller or on the order of the wavelength. The important properties of FIM are that (i) its computational time scales linearly as $cN$ with a very small constant $c$, (ii) it can handle many periodic problem types for which PGFs are available (i.e. it is kernel independent), (iii) it requires only $O(1)$ PGF evaluations, which can be accomplished with any existing method, and (iv) it can be incorporated into existing IE solvers.

2. Problem formulation

Consider an array of unit cells, each comprising a number $N$ of sources and coinciding observers (Fig. 1). The array can be of any dimensionality and can reside in free space or near more complicated background (e.g. a layered medium). The array’s periodicity $L$ is assumed to be small or moderate (i.e. $L \ll \lambda$ or $L - \lambda$, where $\lambda$ is the wavelength). The sources in neighboring unit cells have a generally complex linear phase shift. The scalar potential in the zeroth unit cell is given by

$$u(r_o) = \sum_{n=1}^{N} G_p(r_o - r_n)q_n$$ (1)

Here, $G_p$ is the PGF describing the periodic array. For example, the PGF for a 1D array in free space is given by $G_p(r) = \sum_{i=-\infty}^{\infty} \exp(-jkR_i - jik_oL)/R_i$, where $k = 2\pi/\lambda$, $k_o$ is the...
phase shift wavenumber, and \( R = [(x - iL)^2 + y^2 + z^2]^{1/2} \). Similar expressions can be given for other PGFs. The task of Eq. (1) appears in various IE methods, e.g. it can be directly used in mixed-potential electric field IE solvers.

The direct evaluation of \( u(r_n) \) in Eq. (1) requires \( O(N^2) \) PGF evaluations. The goal of this work is to reduce the computational cost to \( O(N) \) with \( O(1) \) PGF evaluations.

### 3. Fast Interpolation method for fast field evaluation

#### 3.1 Representations of the PGF and the potential

The scalar potential can be split into its near- and far-field components

\[
u(r_n) = u_{\text{near}}(r_n) + u_{\text{far}}(r_n) ; \quad u_{\text{near}}(r_n) = \sum_{n=1}^{N} G_{\text{near}}^p(r_n - r_o) q_n , \quad u_{\text{far}}(r_n) = \sum_{n=1}^{N} G_{\text{far}}^p(r_n - r_o) q_n
\]

(2)

The near-field component \( u_{\text{near}} \) is given in terms of the near-field Green’s function \( G_{\text{near}}^p \) taking into account the sources in the zeroth and its neighbor unit cells, whereas the far-field component \( u_{\text{far}} \) is given in terms of the far-field Green’s function \( G_{\text{far}}^p \) accounting the rest of the unit cells. For example, for 1D arrays in free space

\[
G_{\text{near}}^p(r) = \sum_{k=-\infty}^{\infty} \exp(-jkR) \bigg/ R, \quad G_{\text{far}}^p(r) = G^p(r) - G_{\text{near}}^p(r)
\]

(3)

The task of evaluating \( u_{\text{near}} \) is identical to that of evaluating potentials in non-periodic structures, with several fast methods available (e.g. FMMs and interpolation based methods with \( O(N) \) cost or FFT methods with \( O(N \log N) \) cost). A new efficient FIM for the evaluation of the far-field potential \( u_{\text{far}} \) is presented next.

#### 3.2 Rapid evaluation of the far-field periodic potential

The far-field PGF \( G_{\text{far}}^p \) and potential \( u_{\text{far}} \) are spatially slowly-varying functions in the zeroth periodic unit cell. The reason is that these potentials are generated by sources residing at a large distance from the observation domain (i.e. the unit cell). This behavior allows using interpolations for their rapid evaluation thus reducing the computational cost to \( O(N) \). The proposed algorithm is accomplished in three stages.

**Stage 1 (evaluating \( G_{\text{far}}^p \) at source and observation grids):** In this stage, \( G_{\text{far}}^p \) is evaluated in highly sparse grids of points in the zeroth periodic unit cell (Fig. 1). Conceptually, separate grids are defined for the source and observation domains. Let the grid \( \{r_n^s\}_{n=1}^{N_s} \) be the sparse source grid and \( \{r_m^o\}_{m=1}^{N_o} \) be the sparse observation grid. The number of the grid points is \( N_s = O(l) \) and the cost of evaluating \( G_{\text{far}}^p(r_n^s - r_m^o) \) scales as \( O(l) \). This evaluation can be performed using any available method. The choice of the grids is very flexible. In this paper, the source and observation grids are chosen to coincide (with \( N_s = N_o \)) and to be uniform in all three dimensions. A benefit of using such grids is that, due to the shift invariant property of \( G_{\text{far}}^p \), the cost of calculating \( G_{\text{far}}^p(r_n^s - r_m^o) \) is only \( cN_o \) (where \( c \) is a constant). In addition, since the evaluation of \( G_{\text{far}}^p \) at the grids only depends on \( L \), \( k_{r0} \), \( k \), it is evaluated only once in the pre-processing stage of an IE solver.

**Stage 2 (evaluating \( u_{\text{far}} \) at the observation grid \( \{r_m^o\}_{m=1}^{N_o} \)):** The potential \( u_{\text{far}}(r_n^o) \) is evaluated at \( N_o = O(l) \) observation grid points. This can be accomplished in two steps. First, \( G_{\text{far}}^p(r_n^o - r_m^o) \) for the source locations \( r_n^o \) and observation grid points \( r_m^o \) is calculated...
by locally interpolating from the source grid points \( \mathbf{r}_i \). In other words,
\[
G_{\text{temp}}^s (\mathbf{r}_i - \mathbf{r}_m) = \sum_{n=1}^{\mathcal{N}_s} \mathbf{w}^s (\mathbf{r}_m, \mathbf{r}_n)^T G_{\text{temp}}^s (\mathbf{r}_n - \mathbf{r}_i),
\]
where \( \mathbf{w}^s (\mathbf{r}_m, \mathbf{r}_n) \) are interpolation coefficients and \( \mathcal{N}_s \) is the number of grid points used for interpolation (\( \mathcal{N}_s = (q + 1)^3 \) with the interpolation order \( q \) for 3D interpolations). Next, \( u_{\text{temp}} (\mathbf{r}_m) \) is found via the summation
\[
u_{\text{temp}} (\mathbf{r}_m) = \sum_{n=1}^{\mathcal{N}_s} G_{\text{temp}}^s (\mathbf{r}_m - \mathbf{r}_n) q_n.
\]
The resulting cost is \( c_1 \mathcal{N}_s \mathcal{N}_o + c_2 \mathcal{N}_s \mathcal{N}_o + c_3 \mathcal{N}_s \mathcal{N}_o \), where \( c_{1,2,3} \) are constants.

Alternatively, combining the two steps and extending the \( n' \) summation to all source grid points with proper zero padding of the interpolation coefficients, \( u_{\text{temp}} (\mathbf{r}_m) \) can be given by
\[
u_{\text{temp}} (\mathbf{r}_m) = \sum_{n=1}^{\mathcal{N}_o} G_{\text{temp}}^s (\mathbf{r}_m - \mathbf{r}_n') Q_n \text{ with } Q_n = \sum_{n=1}^{\mathcal{N}_s} q_n \mathbf{w}^s (\mathbf{r}_m, \mathbf{r}_n').
\]
The resulting computational cost is \( c_1 \mathcal{N}_o \mathcal{N}_o + c_2 \mathcal{N}_s \mathcal{N}_o \). By using identical uniform source and observation grids with FFT to compute the summation for \( u_{\text{temp}} (\mathbf{r}_m) \), the cost is further reduced to \( c_1 \mathcal{N}_o \mathcal{N}_o + c_2 \mathcal{N}_s \mathcal{N}_o \). For both above alternatives, the computational cost of Stage 2 is of \( O(N) \) since \( \mathcal{N}_s, \mathcal{N}_o \), and \( \mathcal{N}_o \) are of \( O(1) \).

**Stage 3 (evaluating \( u_{\text{temp}} \) at the actual observers via interpolation):** The far-field potential \( u_{\text{temp}} (\mathbf{r}_m) \) at all \( N \) actual observers \( \mathbf{r}_m \) is obtained by interpolating from the potential \( u_{\text{temp}} (\mathbf{r}_n) \) at the observation grid. In other words, \( u_{\text{temp}} (\mathbf{r}_m) = \sum_{n=1}^{\mathcal{N}_o} \mathbf{w}^o (\mathbf{r}_m, \mathbf{r}_n) u_{\text{temp}} (\mathbf{r}_n) \), where \( \mathbf{w}^o (\mathbf{r}_m, \mathbf{r}_n) \) are interpolations coefficients. Because the number of observation grid points is \( O(1) \), the computational cost of Stage 3 scales as \( O(N) \).

Finally, the computational cost of evaluating \( u_{\text{temp}} \) scales as \( O(N) \), which is much smaller than the cost of \( O(N^2) \) of the direct superposition evaluation. This cost is also smaller than the cost of FFT based methods (\( O(N) \) of FIM vs. \( O(N \log N) \) or \( O(N^{3/2}) \) of FFT-based methods), reduced memory consumptions, and drastically reduced cost of the PGF evaluation (\( O(1) \) for FIM vs. \( O(N) \) for FFT-based methods).

4. Numerical results

The FIM described in Sec. 3 was implemented in a Fortran code, compiled with Intel Fortran v11.1 at –O3 optimization. The simulations were run on a single core of Intel i7-

![Fig. 2(a): Computational time for the fast interpolation method (preprocessing and evaluation components) and direct evaluation. Fig. 2(b): RMS error of the FIM for different number of grid samples and different interpolation order.](image-url)
920 2.66GHz CPU. The results are shown for an infinite linear array with the PGF computed as in [4] for \( L = \lambda \) and \( k_{v} = (1.2-0.01)k \). Similar results can be obtained for other periodic problem types.

Figure 2(a) shows the computational time of \( u_{iu} \) using the FIM in Sec. 3 for sources and observers distributed randomly a cube of size \( L \). The times shown include the time of pre-processing (to be executed only once), composed of the calculation of the PGF at the sparse grids (Stage 1 in Sec. 3.2) and calculation of the interpolation coefficients (Stage 2 in Sec. 3.2), as well as the time of the potential evaluation (Stages 2 and 3 in Sec. 3.2) to be executed at every iteration step for an IE solver. For comparison, Fig. 2(a) also shows the time of direct evaluation via Eq. (1). The number of grid points was \( N_u = N_j = 4 \) and cubic (Lagrange) interpolation was used resulting in RMS errors \( \leq 10^{-3} \). The asymptotic computational time and memory of the direct and FIM scale as \( O(N^2) \) and \( O(N) \), respectively. The absolute time of the FIM is also very small for all considered problem sizes. For example, for \( N = 64 \), the speed-up is 14 and for \( N = 2 \) millions the FIM computational time is 13 seconds with a 1.4e6 fold speed-up (with the direct time extrapolated for this large \( N \)). The pre-processing time is on the order of the computation time for larger \( N \) and it saturates at a certain values at smaller \( N \), which is because of the cost of the PGF evaluation at the sparse grids (which is still small).

Figure 2(b) shows the RMS error behavior as a function of the grid densities and interpolation order. It is evident that the error can be controlled by increasing sampling rates and/or the interpolation order. The computational times increase for larger interpolation orders and grid densities, e.g. the time for a linear interpolation is 8 times smaller than that for the cubic one (as in Fig. 2(a)).

5. Summary

We presented, for the first time, a FIM for evaluating electromagnetic fields in a general periodic unit cell in \( O(N) \) operations with only \( O(1) \) PGF evaluations. The specific example shown in Fig. 1 is for the case of a linear periodicity, but the code can work seamlessly for other periodicity types and for problems with more complicated environments (e.g. for arrays above layered media). For such problems, the time of Stages 2 and 3 in Sec. 3.2 (the solid curve in Fig. 2(a)) remains unchanged and only the PGF needs to be re-tabulated at the sparse source and observation grids (Stage 1, which is to be evaluated only once). In addition, the same method can be easily applied to vector problems, for which any differential operator applied to the PGF in Eq. (1) can be handled by replacing the interpolation in Stage 3 of Sec. 3.2 by numerical derivatives.

The FIM performance is very robust for virtually any structure parameters as long as the periodicity is small enough (practically for \( L < 2\lambda \)), with better performance for smaller \( L \). Extensions of the presented FIM to problems with larger \( L \) are under development.

This work is supported by DARPA NACHOS program, and NSF ERC Cian center.