Multiscale Calculation of Many Eigenfunctions

Achi Brandt
Department of Computer Science and Applied Mathematics
The Weizmann Institute of Science
Rehovot, Israel

Contents

1 Introduction

1.1 Background

This research is an (important) part of the general drive of developing multiscale computational techniques, where each part contributes ideas and insights to the others. The central new multiscale paradigm currently being developed in various fields is what we call Systematic Upscaling (SU) [13], [12]. Based on methods that emerged in the multigrid research in applied mathematics and on renormalization group methods developed in theoretical physics, and on their joint application in statistical mechanics [15], SU is a methodical approach to derive, from a given system of fine-scale physical equations, equivalent numerical equations at increasingly coarser scales. By climbing the scales by one small factor (typically between 1:2 to 1:10) at a time, and by switching back and forth between finer and coarser scales, SU calculations can be restricted to only small representative regions at each scale, and avoid computational slowdowns typical to single-level calculations over vast domains. The resulting (generally nonlinear) equations derived for each scale can be deterministic or stochastic, in various numerical forms (A Hamiltonian, a conditional probability table, discrete equations, etc.), possibly changing with the scale. Since only representative regions need to be calculated at each scale, the SU approach allows computing indefinitely large systems, provided their equations are repetitive, i.e., the same relatively few equations keep repeating themselves and govern the system everywhere.

Coarsening a system of discrete eigen equations (resulting, e.g., from a differential eigen system) is relatively simple if only few lowest eigenfunctions are to be calculated (see Sec. 1.2 below). Not so if many eigenfunctions are required, since normally they cannot all satisfy the same coarse equations. For one-dimensional (1D) differential systems, a method to derive equivalent coarse-grid equations that accommodate all the eigenfunctions was nonetheless recently developed [2],

[10]. It allows the calculation of N eigenfunctions in just $O(N \log N)$ operators, without assuming physical localization; i.e., without assuming that the physically interesting properties at a point depend only on a local neighborhood containing just N_{local} atoms, where $N_{local} \ll N$. *With* physical localization, the complexity of that algorithm is just $O(N \log N_{local})$. However, the coarse equations of this algorithm are in terms of monodromies (to bypass the problems associated with the high indefiniteness of finite-difference or finite-element discretizations), which is a purely one dimensional concept that cannot be generalized to higher dimensions. Nevertheless, that algorithm has served to show that coarsening many-eigenfunction systems and attaining $O(N \log N)$ efficiency is in principle feasible, leaving open the question of how generally this can be achieved (or at least approached). The main objective of the current work (and the article below) is to answer that question. It also aims at viewing this question in the particular context of systematic upscaling of molecular systems (see more on that in Sec. 6).

1.2 Low eigenfunctions. EIS

As a preliminary step it was needed to study how to coarsen an eigensystem so that at least *low* eigenfunctions, but as many of them as possible, can be calculated on increasingly coarser grids. The multigrid system developed for that purpose, the Exact Interpolation Scheme (EIS), is described in Sec. 8 of [7]. The EIS form of inter-scale transfers has later become a basic ingredient in the algorithms described in [1] and below. It also serves as a basic vehicle in new algebraic multigrid (AMG) solvers, particularly for problems whose operators are plagued with unusually small, and/or unusually many, low eigenvalues.

The number of eigenfunctions at one level that can be accurately approximated by the next-coarser-level EIS equations is limited to the number that can be approximated well enough by one and the same interpolation operator, a number which is much reduced upon each additional coarsening. This naturally leads to the idea of introducing progressively more interpolation operators at increasingly coarser grids, as developed below.

1.3 One-dimensional preliminaries

The next step has been to develop the intended solver in the simpler 1D framework, insisting however on a development that, unlike [2] and [10], can be directly generalized to higher dimensions. The 1D framework has indeed been ideal to accurately study and test the process of separating calculation at progressively narrower bands of eigenvalues. To study this we have developed an algorithm for calculating just two eigenfunctions, with eigenvalues near any given real number. Reported in [1], it uses EIS together with the techniques of [8] for treating a highly indefinite system.

1.4 The MEB algorithms

For organizing many eigenfunctions, we propose the Multiscale EigenBase (MEB) approach, described in Sec. 2 below. Our main example, the linear or nonlinear Kohn-Sham eigen system, is presented in Sec. 3, explaining its suitability for our general studies. The 1D-MEB is described in detail below (Sec. 4). Then its anticipated generalization to higher dimensions is outlined (Sec. 5), and its expected benefits, specifically for electronic structure calculations, are summarized (Sec. 6).

2 The multiscale eigenbasis (MEB) approach

We are generally interested in solving a differential eigenproblem of the form

$$Lu_i(x) = \lambda_i u_i(x) \ , \quad (i = 1, \ldots, N) \tag{1}$$

with some homogeneous boundary conditions, to within some given tolerable error ϵ. Here x is d dimensional and L is a differential operator. We are interested in the N lowest eigenvalues λ_i and the corresponding eigenfunctions u_i.

The MEB algorithms proceed from some finest grid on which the problem is discretized to increasingly coarser grids ("levels"). At each level, with meshsize h say, the algorithm constructs a *"scale-h basis"*, namely, a sequence of functions $u_j^h(x)$, $j = 1, \ldots, n_h$, such that every eigenfunction is a *h-smooth combination* of these basis functions, i.e., each eigenfunction $u(x)$ can be written in the form

$$u(x) = \sum_j (I_H^h W_j^H)(x) \cdot u_j^h(x) \ , \tag{2}$$

where W_j^H is a function defined on a coarser grid of meshsize $H = mh$, $m > 1$ being a small integer (usually $m = 2$) and I_H^h is an interpolation operator, whose order should increase proportionately to $\log \frac{1}{\epsilon}$.

Using the scale-h basis one can upscale the eigen-equations from grid h to grid H, then use the grid-H equations to calculate a scale-H basis. Each scale-H basis function will be described as an h-smooth combination of several scale-h basis functions, and will thus be defined on grid H. The number of scale-H basis functions will roughly be m^d times their number at scale h, so the total amount of discrete values used by the H basis will be the same as that of scale h. The eigen-equations will be transfered from the finest level to increasingly coarser level, then solved at the coarsest level.

Thus, the structure of the solution is that each eigenfunction is represented as a smooth mollification of coarsest-level basis functions, each of which is represented as a smooth mollification of next-finer-level basis functions, and so on. Only very few basis functions need to be represented at the finest level.

It may seem a drawback of the MEB algorithm that, although efficient, it does not yield explicit representation of each eigenfunction at the finest level. But this in fact is a great advantage. The resulting MEB structure is not only much less expensive to construct and store, but also much more efficient to use, for nearly any task, because it allows the use of fast summation techniques, as in [4]. For example, to expand a given function $f(x)$ in terms of the eigenfunctions, one needs to calculate the inner product (f,u) for each eigenfunction $u(x)$. Using (2),

$$\begin{aligned}(f,u) &= \sum_j (I_H^h W_j^H, u_j^h f) \qquad (3)\\ &= \sum_j (W_j^H, (I_H^h)^T u_j^h f)\ .\end{aligned}$$

Hence, at the finest level one only has to calculate $(I_H^h)^T u_j^h f$ for each of the finest-level basis functions, whose number is small. It can similarly be shown that in DFT calculations, the MEB structure yields much faster calculation of the electronic density than its calculation from the eigenfunctions each given explicitly at the finest level.

3 Schrödinger operator and discretization

Our prime example is the Schrödinger operator

$$L = -\Delta + V(x) \qquad (4)$$

where Δ is the Laplace operator and the potential $V(x)$ is either given (the linear problem) or defined in the DFT self-consistent way [3].

The Schrödinger operator appearing in the self-consistent Kohn-Sham equations is not only a very important example; it also offers some simplifications in terms of what we can assume about V, since the resulting effective potential $V(x)$ tends to be smooth and with mild variations. Moreover, the self-consistency requirement by itself is not expected to substantially increase the complexity of the MEB solver (see Sec. 5.5).

Indeed, as will be described below, the MEB algorithm can work separately at different eigenvalue intervals. Hence it can proceed from lower to higher values of λ. Lower λ will give more localized eigenfunctions (e.g., core electrons), which will only marginally be affected by updates to the self-consistent potential introduced by higher eigenfunctions. It is thus expected that just one visit back to lower eigenvalue intervals after calculating higher ones will usually suffice for such updating. When calculating at higher eigenvalue intervals, lower eigenfunctions have already modified the potential V in such a way (e.g., nucleus singularity being masked by the core electrons) that V can indeed be assumed to be smooth and mildly varying. (And see the comment on self-consistent calculations in Sec. 5.5.)

Another simplicity of DFT eigenproblems is that their domain is in principle unbounded. So for simplicity of our first development, and of the description below, we have assumed that $V(x)$ is a given smooth function, mildly varying and periodic. See in Secs. 4.7 and 5.3 for comments on other situations.

For this type of a problem we will see below (as also in [1]) that the MEB structure allows very efficient discretizations, too. Namely, the finest meshsize h_0 does not need to resolve the wavelengths of the (highest) eigenfunctions: it only needs to be fine enough to resolve the landscape of the potential $V(x)$. This can be done because at such a meshsize the scale-h_0 basis functions can be given in an analytical form (see Sec. 4.2).

Moreover, the discretization can be fully adaptive, using different meshsizes and approximation orders as needed at different subdomains. For this purpose, the multigrid structure of local refinements in terms of overset uniform grids (see [5] and [6]) is especially convenient and efficient. The local finer level of that structure will usually correspond to calculations at lower eigenvalue intervals, while calculations at higher intervals will use only coarser grids, carrying pre-calculated finer-grid corrections: either defect corrections as in FAS multigrid, or Galerkin-type approximations based on the finer levels (the Exact Interpolation Scheme (EIS) introduced in [7]), as used in the algorithm below. (See more in Sec. 4.7.)

4 Current one-dimensional MEB algorithm

4.1 General

The publication [1] shows a method for calculating two eigenfunctions with eigenvalues close to some given value λ. Based on the same general approach, the MEB algorithm described below has been designed for calculating *all* the eigenvalues in some given interval $\lambda_{\min} \le \lambda \le \lambda_{\max}$.

As explained above, we aim at solving the eigenproblem (1), with the Schrödinger operator (4), and we first assume for simplicity a uniform finest grid of meshsize h_0 that well resolves the landscape of $V(x)$, with the latter potential being smooth, mildly varying and satisfying the periodicity condition $V(x+1) = V(x)$ for all $x \in \mathbb{R}$. Extensions beyond these assumptions are discussed later.

4.2 Discretization

We divide the interval $(\lambda_{\min}, \lambda_{\max}]$ into the union of disjoint subintervals $J^0_\alpha = (\lambda^0_{\alpha-1}, \lambda^0_\alpha]$, $(\alpha = 1, \ldots, n^0)$, with $\lambda^0_0 = \lambda_{\min}$ and $\lambda^0_{n^0} = \lambda_{\max}$, the length of J^0_α being bounded by some relation like

$$\lambda^0_\alpha - \lambda^0_{\alpha-1} \le h_0^{-1} (\max[h_0^{-2}, \lambda^0_{\alpha-1} - \min_x V(x)])^{1/2} \ . \tag{5}$$

This relation is designed to ensure that at the finest level h_0 just two functions, called $u^0_{\alpha+}$ and $u^0_{\alpha-}$, will suffice as scale-h_0 basis for all the eigenfunctions with eigenvalues in the interval J^0_α. Note that (5) requires only a small number n^0 of intervals; e.g., n^0 does not depend on the number of atoms. The number will remain uniformly bounded and small even for indefinitely large $\lambda_{\max} - \lambda_{\min}$ (cf. Sec. 5.4 below).

The two basis functions for each J^0_α can in fact be given analytically; e.g.,

$$u^0_{\alpha+}(x) = \exp(ik^0_\alpha(x)x) \ , \ u^0_{\alpha-}(x) = \exp(-ik^0_\alpha(x)x) \ , \tag{6}$$

where k^0_α is the local wave number associated with $\hat{\lambda}_\alpha = (\lambda^0_{\alpha-1} + \lambda^0_\alpha)/2$, the center point of J^0_α; i.e.,

$$k^0_\alpha(x) = (\max[0, \hat{\lambda}^0_\alpha - V(x)])^{1/2} \ . \tag{7}$$

(Even better seems to be the Liouville-Green form $u^0_{\alpha\pm} = \exp\left(\pm i \int^x k(\xi)d\xi\right)$, but this does not have the needed localness property, and is particularly less convenient for higher-dimensional analogy). Indeed, it can be shown that each eigenfunction in J^0_α (i.e., whose eigenvalue belongs to J^0_α) can be written as

$$u(x) = w_+(x)u^0_{\alpha+}(x) + w_-(x))u^0_{\alpha-}(x) \ , \tag{8}$$

where the "mollification functions" $w_+(x)$ and $w_-(x)$ are smooth at scale h_0, i.e., they can be interpolated from function w^1_+ and w^1_-, respectively, defined on a grid G^1 with meshsize $h_1 = 2h_0$.

The equations on grid G^1, which is the finest level with discrete equations, is a coupled pair of eigen equations for the pair of functions $w^1_+(x)$ and $w^1_-(x)$. These equations can be derived in two alternative ways. One way is to substitute (8) into (1), (4), and discretize on G^1 the obtained differential equation multiplied by $u^0_{\alpha+}(x)$, with a similar discretization also for multiplication by $u^0_{\alpha-}(x)$. Another way, more in line with the coarsening stages of the algorithm, is to introduce two interpolation operators, $I^0_{\alpha+}$ and $I^0_{\alpha-}$, defined by

$$(I^0_{\alpha\pm}W^1)(x) = u^0_{\alpha\pm}(x) \cdot (I^0W^1)(x) \ , \text{ for any } W^1 \in \overline{G}^1 \ , \tag{9}$$

where I^0 is a polynomial interpolation from G^1 to the continuum, of order proportional to $\log\frac{1}{\epsilon}$, and $\overline{G}^\ell$ will generally stand for the space of complex functions defined on a grid G^ℓ. We denote by $I^{0\dagger}$, $I^{0\dagger}_{\alpha+}$ and $I^{0\dagger}_{\alpha-}$ the adjoints of I^0, $I^0_{\alpha+}$ and $I^0_{\alpha-}$, respectively. That is, if $p^1_j(x)$ re the coefficients of I^0, defined by

$$(I^0W^1)(x) = \sum_{x^1_j \in G^1} W^1(x^1_j)p^1_j(x) \ , \text{ for any } W^1 \in \overline{G}^1 \ , \tag{10}$$

then

$$(I^{0\dagger}w)(x^1_j) = \int w(x)p^1_j(x)dx \ , \text{ for any continuous function } w(x) \ , \tag{11}$$

and

$$(I^{0\dagger}_{\alpha\pm}w)(x^1_j) = \int \overline{u^0_{\alpha\pm}(x)}w(x)p^1_j(x)dx \ , \tag{12}$$

superbar standing for complex conjugate. With this notation, the G^1 eigen equations in J^0_α can be written as

$$A^\ell_\alpha \begin{pmatrix} W^\ell_+ \\ \\ W^\ell_- \end{pmatrix} = \lambda B^\ell_\alpha \begin{pmatrix} W^\ell_+ \\ \\ W^\ell_- \end{pmatrix} \ , \ W^\ell_\pm \in \overline{G}^\ell \ , \tag{13}$$

where $\ell = 1$ (similar equations will later be constructed for coarser levels ℓ) and

$$A^1_\alpha = \begin{pmatrix} I^{0\dagger}_{\alpha+}LI^{0\dagger}_{\alpha+} & I^{0\dagger}_{\alpha+}LI^{0\dagger}_{\alpha-} \\ \\ I^{0\dagger}_{\alpha-}LI^{0\dagger}_{\alpha+} & I^{0\dagger}_{\alpha-}LI^{0\dagger}_{\alpha-} \end{pmatrix} \tag{14a}$$

$$B^1_\alpha = \begin{pmatrix} I^{0\dagger}_{\alpha+}I^{0\dagger}_{\alpha+} & I^{0\dagger}_{\alpha+}I^{0\dagger}_{\alpha-} \\ \\ I^{0\dagger}_{\alpha-}I^{0\dagger}_{\alpha+} & I^{0\dagger}_{\alpha-}I^{0\dagger}_{\alpha-} \end{pmatrix} \ . \tag{14b}$$

4.3 Coarsening the basis by relaxation

Having obtained the equations for J^0_α eigenfunctions on G^1, we now use them to construct the h_1 basis in that interval, by dividing J^0_α into two subintervals $J^1_{2\alpha-1}$ and $J^1_{2\alpha}$, where for any β and any level ℓ we will generally denote $J^\ell_\beta = (\lambda^\ell_{\beta-1}, \lambda^\ell_\beta]$ and $\lambda^\ell_{2\alpha} = \lambda^{\ell-1}_\alpha$, $\lambda^\ell_{2\alpha-1} = \hat{\lambda}^{\ell-1}_\alpha = (\lambda^{\ell-1}_{\alpha-1} + \lambda^{\ell-1}_\alpha)/2$. At each level ℓ and for each interval J^ℓ_β, we construct, on the grid G^ℓ_α with meshsize $h_\ell = 2h_{\ell-1}$, two h_ℓ-basis functions, $u^\ell_{\beta+}$ and $u^\ell_{\beta-}$. Each of these is a 2-vector function (W^ℓ_+, W^ℓ_-) obtained by several Kacmarc relaxation sweeps on Eq. (13) with $\lambda = \hat{\lambda}^\ell_\beta$, starting with the initial approximation $(1_\ell, 0)$ to obtain $u^\ell_{\beta+}$, and with $(0, 1_\ell)$ to obtain $u^\ell_{\beta-}$, where 1_ℓ is the $\overline{G}^\ell$ function whose all entries are 1.

The number of relaxation sweeps should grow proportionately to $\log\frac{1}{3}$. To reduce this number, a short (usually just two-level) *cycle of coarse-level correction* (see Sec.4.5 below) may be useful.

If the off-diagonal operators in A^ℓ_α and B^ℓ_α are sufficiently small compared with those of the diagonal, relaxation can most efficiently be done by relaxing $u^\ell_{\alpha+}$ separately from $u^\ell_{\alpha-}$. The needed diagonal dominance is obtained if the finest basis functions $u^0_{\alpha\pm}$ are locally (at scale h_ℓ) orthogonal enough. This is obtained (already for $\ell = 1$) in the case of the closed form solution (6) when the potential V is smooth enough (on scale h_0). Otherwise this can be obtained by local orthogonalization (see Sec. 4.7).

4.4 Coarsening the equations

Having formed a suitable h_ℓ basis for each eigenvalue J_α^ℓ, equations for that interval at the next coarser grid $G^{\ell+1}$ can be derived. Which is done analogously to the derivation of discretization (Sec. 4.2). Namely, two interpolation operators $I_{\alpha+}^\ell$ and $I_{\alpha-}^\ell$ are introduced, defined by

$$(I_{\alpha\pm}^\ell W)(x) = u_{\alpha\pm}^\ell(x)(I^\ell w)(x) \text{ for any } W \in \overline{G}^{\ell+1},\ x \in G^\ell\ , \tag{15}$$

where I^ℓ is a polynomial interpolation from $G^{\ell+1}$ to G^ℓ of order proportional to $\log\frac{1}{\epsilon}$. Denoting by $I_{\alpha+}^{\ell\dagger}$ and $I_{\alpha-}^{\ell\dagger}$ the complex conjugate transpose of $I_{\alpha+}^\ell$ and $I_{\alpha-}^\ell$, respectively, and by I_α^ℓ and $I_\alpha^{\ell\dagger}$ the 2×2 interpolation operators

$$I_\alpha^\ell = \begin{pmatrix} I_{\alpha+}^\ell & 0 \\ 0 & I_{\alpha-}^\ell \end{pmatrix}\ ,\quad I_\alpha^{\ell\dagger} = \begin{pmatrix} I_{\alpha+}^{\ell\dagger} & 0 \\ 0 & I_{\alpha-}^{\ell\dagger} \end{pmatrix}\ , \tag{16}$$

the coarse-grid eigen equations can again be written as (13), with $\ell+1$ instead of ℓ and with

$$A_\beta^{\ell+1} = I_\alpha^{\ell\dagger} A_\alpha^\ell I_\alpha^\ell\ ,\ B_\beta^{\ell+1} = I_\alpha^{\ell\dagger} B_\alpha^\ell I_\alpha^\ell\ ,\ (\beta = 2\alpha-1, 2\alpha;\ \ell \ge 1)\ . \tag{17}$$

With Eq.(13) now at level $\ell+1$, one can relax and create the $h_{\ell+1}$ basis (as in Sec. 4.3), then construct the equations for level $\ell+2$, etc.

4.5 Coarse level correction cycle (optional)

Actually, before splitting J_α^ℓ into $J_{2\alpha-1}^{\ell+1}$ and $J_{2\alpha}^{\ell+1}$ for creating the $h_{\ell+1}$ basis, one may use grid $G^{\ell+1}$ to accelerate the relaxation on grid G^ℓ. Namely, after a couple of relaxation sweeps on grid G^ℓ, one forms the $G^{\ell+1}$ equations (as in Sec. 4.4), then relaxes them, starting with $W_+^{\ell+1} \equiv W_-^{\ell+1} \equiv 1_{\ell+1}$, then use the resulting $W_\pm^{\ell+1}$ to correct the G^ℓ solution by

$$u_{\alpha+}^\ell \leftarrow I_{\alpha+}^\ell W_+^{\ell+1}\ ,\ u_{\alpha-}^\ell = I_{\alpha-}^\ell W_-^{\ell+1}\ . \tag{18}$$

Adding afterwards several more relaxation sweeps on G^ℓ one can coarsen the equations again, either to create the $h_{\ell+1}$ basis or to have another cycle of correction to the G^ℓ solution.

In the case of more correction cycles, one can also accelerate the $G^{\ell+1}$ relaxation by another coarsening, to grid $G^{\ell+2}$ (without yet splitting J_α^ℓ), and so on. However, only one level, if any, is expected to be useful, since the purpose of these cycles is not to create an exact eigen *solution* at level ℓ, only an h_ℓ *basis*; hence there is no need to converge smooth mollifications, only to somewhat accelerate the convergence of some intermediate ones. (The extent of usefulness of correction cycles really needs to be further explored.)

4.6 Solving at coarsest and accuracy control

At the coarsest level ℓ, the number of gridpoints is very small, hence the obtained equations (13) can inexpensively directly be solved, for one interval J_α^ℓ at a time. Some intervals may turn out to contain no solution with $\lambda_{\alpha-1}^\ell < \lambda \le \lambda_\alpha^\ell$, others will contain several.

The coarsest grid and the solution on it depend of course on the boundary conditions (BC). In the case of periodic BC, the coarsest level ℓ can have just one gridpoint per period, yielding in each J_α^ℓ a simple 2×2 eigen problem. If the domain is unbounded but the potential is periodic $V(x+1) \equiv V(x)$, each eigenfunction has the Bloch form $u(x) = \tilde{u}(x)e^{i\gamma x}$, with $\tilde{u}(x+1) \equiv \tilde{u}(x)$ and $-\pi < \gamma \le \pi$. The finer-level construction of local bases does not depend on γ, since $e^{i\gamma x}$ is a very smooth mollification at all scales up to the scale of the period. Only at the 2×2 coarsest system the value of γ enters as a parameter. Other types of boundary conditions are discussed in Sec. 5.4 below.

A simple way to check the accuracy of the solution is to compare the eigenvalues of two solutions obtained with two different sets of parameters (different h_0, interpolation orders, number of relaxation sweeps, etc.). One can also compare a specific eigenfunction obtained by the two different solutions, by performing for it all the chain of interpolations from the coarsest level to the finest. Or compare certain important numbers that can inexpensively be calculated, such as the electronic density or the expansion coefficients of some functions (cf. Sec. 2). If the accuracy turns out insufficient, the more accurate of the two solutions (e.g., the one with smaller h_0) is retained and compared to a new solution calculated to greater accuracy (e.g., with still smaller h_0, and/or higher-order interpolations, etc.). And so on until the desired accuracy is obtained. Not much work is wasted, since the final, most accurate solution will always consume most of the work.

If $\hat{\ell}$ is the coarsest level, there is the possibility that two calculated eigenfunctions, with eigenvalues $\lambda_1 \in J_\alpha^{\hat{\ell}}$ and $\lambda_2 \in J_{\alpha+1}^{\hat{\ell}}$ but with $\lambda_2 - \lambda_1$ smaller than the estimated eigenvalue numerical error, represent actually the same physical eigenfunction. If the interpolation orders (and hence the accuracy orders) are higher than the dimension d, this will rarely happen, and can easily be detected by having the basic interval $(\lambda_{\min}, \lambda_{\max}]$ slightly shifted in one of the compared solutions, so that its intervals J_α^ℓ come out also suitably shifted.

4.7 Non-smooth equations. Local orthogonalization. Adaptive grids

The eigenproblem may have regions where the finest level basis cannot be formulated in a closed form such as (6), due to non-smoothness of the potential, boundary conditions, etc. In such regions, for higher eigenvalue intervals, the finest grid G^0 must be fine enough to resolve the eigenfunctions. To form an h_0 basis one would then use relaxation together with local orthogonalization.

The purpose of the local orthogonalization is to create a sufficiently separated h_0 basis, meaning $u^0_{\alpha+}$ and $u^0_{\alpha-}$ such that B^1_α is sufficiently diagonally dominant, i.e.,

$$\|I^{0\dagger}_{\alpha+} I^0_{\alpha-}\|^2_x \ll \|I^{0\dagger}_{\alpha+} I^0_{\alpha+}\|_x \cdot \|I^{0\dagger}_{\alpha-} I^0_{\alpha-}\|_x \ , \ \text{for all } x \in G^1 \ , \tag{19}$$

where $\|\cdot\|_x$ denotes a norm at some local neighborhood of the point x. To achieve that, the relaxation sweeps for $u^0_{\alpha+}$ and $u^0_{\alpha-}$ are intermingled with a process of *smoothly recombining* these functions; e.g., replacing

$$u^0_{\alpha-} \leftarrow u^0_{\alpha-} - I^0_{\alpha+} W \ , \quad (W \in G^1) \ , \tag{20}$$

where the values of W are sequentially adjusted, each value in its turn being chosen so as to minimize $\|I^{0\dagger}_{\alpha+} I_{\alpha-}\|$. This local orthogonalization can be further strengthened by a similar process conducted also on G^2, recombining $u^1_{\alpha+}$ and $u^1_{\alpha-}$.

Since all h_ℓ bases are local, the desired finest-level meshsize can easily change from region to region. One convenient and efficient way to organize such an adaptive discretization is to think in terms of uniform grids covering the entire domain at all levels, starting from the level of the finest meshsize needed anywhere; except that in regions where a certain grid G^ℓ is finer than necessary, the h_ℓ basis functions can be chosen to be identically 1, hence they need not be explicitly stored.

5 Higher Dimensional MEB outline

5.1 General

The MEB algorithm in dimension d is directly analogous to the 1D procedures described above. The most important difference is that each interval J^ℓ_α of eigenvalues may contain many, not just two, h_ℓ basis functions $u^\ell_{\alpha,\nu}$ ($\nu = 1, 2, \ldots, n^\ell_\alpha$); their number n^ℓ_α increases like $n^{\ell+1}_{2\alpha-1} = n^{\ell+1}_{2\alpha} = 2^{d-1} n^\ell_\alpha$. In particular at the discretization level, the h_0 basis functions, analogously to (6), may more generally be

$$u^0_{\alpha,\nu}(x_1, \ldots, x_d) = \exp\left[i \sum_{j=1}^{d} k^0_{\alpha,\nu,j}(x_1, \ldots, x_d) x_j \right] \tag{20a}$$

where the vectors $\mathbf{k}^0_{\alpha,\nu} = (k^0_{\alpha,\nu,1}, \ldots, k^0_{\alpha,\nu,d})$ are selected uniformly in the sphere

$$\{\mathbf{k} : \mathbf{k}^2 = \max[0, \hat{\lambda}^0_\alpha - V(x_1, \ldots, x_d)]\} \ . \tag{20b}$$

At a higher level ℓ, any eigenfunction $u(x)$ with eigenvalue in J^ℓ_α has the form

$$u(x) = I^\ell_\alpha \cdot W^{\ell+1} \ , \quad W^{\ell+1} = \begin{pmatrix} W^{\ell+1}_1 \\ \vdots \\ W^{\ell+1}_{n^\ell_\alpha} \end{pmatrix} \ , \quad W^{\ell+1}_\nu \in \overline{G}^{\ell+1} \ , \tag{21}$$

where $I_\alpha^\ell = (I_{\alpha,1}^\ell, \ldots, I_{\alpha,n_\alpha^\ell}^\ell)$ and each interpolation operator is defined by

$$(I_{\alpha,\nu}^\ell W_\nu^{\ell+1})(x) = u_{\alpha,\nu}^\ell(x)(I^\ell W_\nu^{\ell+1})(x) \ , \tag{22}$$

with I^ℓ being a polynomial interpolation from $G^{\ell+1}$ to G^ℓ of order proportional to $\log \frac{1}{\epsilon}$. (See Sec. 7 for extension to AMG-type interpolations.) The G^ℓ eigen equations, instead of (13), take the more general form

$$A_\alpha^\ell W^\ell = \lambda B_\alpha^\ell W^\ell \ , \quad W^\ell = \begin{pmatrix} W_1^\ell \\ \vdots \\ W_n^\ell \end{pmatrix} \ , \quad W_\nu^\ell \in \overline{G}^\ell \tag{23}$$

where $n = n_\beta^{\ell-1}$, $\alpha = 2\beta - 1$ or $\alpha = 2\beta$, and the matrix operators A_α^ℓ and B_α^ℓ are again defined recursively by (17) above.

After forming the G^ℓ equations (23), the n_α^ℓ basis functions are first formed by local relaxation, starting with $n_\alpha^\ell = 2^{d-1}n$ different initial approximations. Each of these initial approximations is a vector $W^\ell = (W_1^\ell, \ldots, W_n^\ell)^T$, in which $W_\nu^\ell = 0$ for all $\nu \neq \mu$ while $W_\mu^\ell = F_{\mu,q}^\ell$, $(\mu = 1, \ldots, n;\ q = 1, \ldots, 2^{d-1})$. For each μ, the 2^{d-1} functions $F_{\mu,q}^\ell$ are chosen to be nearly locally orthogonal, i.e., analogously to (19) above, $I^{\ell T}\overline{F}_{\mu,q}F_{\mu,r}I^\ell$ is chosen to be much smaller for $q \neq r$ than for $q = r$. The local near orthogonality between *all* the basis functions $u_{\alpha,\nu}^\ell$, $(\nu = 1, 2, \ldots, n_\alpha^\ell)$, is then further reinforced by the process described next.

5.2 Operator sparsity and local orthogonalization

A central new issue in higher dimensions ($d > 1$) is how to recursively keep the matrices A_α^ℓ and B_α^ℓ as sparse and diagonally dominant as possible, which is of course essential for obtaining full efficiency. This should be done by the process of local orthogonalization (a simple 1D case of which has been described in Sec. 4.7).

For any $1 \leq \nu \leq n_\alpha^\ell$, denote by $S_{\alpha,\nu}^\ell(x)$ the *set of "neighbors"* $u_{\alpha,\mu}^\nu$ of the basis function $u_{\alpha,\nu}^\ell$ in the vicinity of a gridpoint $x \in G^\ell$, defined by

$$S_{\alpha,\nu}^\ell(x) = \{\mu : \|I_{\alpha,\nu}^{\ell\dagger} B_\alpha^\ell I_{\alpha,\mu}^\ell\|_x > \epsilon_\ell\} \ , \tag{24}$$

where the threshold ϵ_ℓ is proportional to (but can be substantially larger than) the desired accuracy ϵ, and we assume the basis function to be roughly *locally normalized*, i.e., $\|I_{\alpha,\nu}^{\ell\dagger} B_\alpha^\ell I_{\alpha,\nu}^\ell\|_x = O(1)$. From (17) it is clear that during relaxation the set $S_{\alpha,\nu}^\ell$ are those inherited from the next-finer level, meaning that at the vicinity of any x they each on the average include 2^{d-1} as many neighbors as in the neighborhood sets of the next finer level.

To trim down the increased number of neighbors, local orthogonalization sweeps are added in between the relaxation sweeps. In those sweeps one sub-

tracts from each $u^{\ell}_{\alpha,\nu}$ a smooth combination of its neighbors:

$$u^{\ell}_{\alpha,\nu} \leftarrow u^{\ell}_{\alpha,\nu} - \sum_{\mu \in S^{\ell}_{\alpha,\nu}} I^{\ell}_{\alpha,\mu} W^{\ell+1}_{\mu} \ , \quad (W^{\ell+1}_{\mu} \in \overline{G}^{\ell+1}) \tag{25}$$

where the values of $W^{\ell+1}_{\mu}$ are adjusted one by one in some order, each in its turn modified so as to lower as far as possible the overlap $I^{\ell\dagger}_{\alpha,\nu} B^{\ell}_{\alpha} I^{\ell}_{\alpha,\mu}$. It is estimated that at each level such local orthogonalization sweeps can reduce the overlap by a factor close to m^{-p}, where m is the coarsening ratio (usually $m = 2$) and p is the order of interpolations. Hence, with p taken high enough, each overlap inherited at some level will disappear in its descendants few coarsening levels later.

5.3 Coarsest level solution: Types of problems

At the coarsest level $\hat{\ell}$, the system can be solved separately for each eigenvalue interval $J^{\hat{\ell}}_{\alpha}$. As in Sec. 4.6 above, one can check the solution accuracy and detect the cases where eigenfunctions appearing at different $J^{\hat{\ell}}_{\alpha}$ actually represent one and the same physical eigenfunction.

The development of the system at the coarsest levels, and the value of $\hat{\ell}$ itself, depend on the type of boundary conditions and may differ for different eigenvalue intervals and at different spatial subdomains. In every case, however, it is expected that at some level $\hat{\ell}$ the relevant part (see below) of grid $G^{\hat{\ell}}$ will be reduced to just one point, so that in the eigen-system (23) each operator A_{ik} and B_{ik} will be just one complex number. Hopefully this system, due to the local orthogonalization at all levels, will come out sparse. Moreover, it is expected that the number of eigenfunctions with eigenvalues in $J^{\hat{\ell}}_{\alpha}$ will be comparable to the size n of the system, and that each eigenfunction will be combined from only one, or very few, of the sets $S^{\hat{\ell}}_{\alpha,\nu}$. Hence it can be expected that the system will be solvable in $O(n)$ computer operations.

For problems in unbounded domains, several situations can be distinguished, as follows.

5.3.1 Localization

For lower eigenvalue intervals, the basis of J^{ℓ}_{α} will become localized in one or several regions, practically vanishing elsewhere. In each such region, a level $\hat{\ell}$ will then be reached at which the region is reduced to only one gridpoint, so the operators A_{ik} and B_{ik} are indeed reduced to mere numbers. Moreover, the system can be solved in each such region separately from other such regions. The level $\hat{\ell}$ at which this happens may of course vary from one region to another.

5.3.2 Periodicity

In the case of periodic potential, e.g.,

$$V(x_1, x_2) = V(x_1 + a_1, x_2) = V(x_1, x_2 + a_2) \ , \quad \text{for all } (x_1, x_2) \in \mathbb{R}^2 \tag{26}$$

each eigenfunction has the Bloch form

$$u(x_1, x_2) = \tilde{u}(x_1, x_2) e^{i(\gamma_1 x_1 + \gamma_2 x_2)} \ , \ -\pi < \gamma_1 \ , \ \gamma_2 \leq \pi \tag{27}$$

where $\tilde{u}$ is periodic. The fine level basis functions will be independent of (γ_1, γ_2). The coarsest level will have meshsizes equaling the periods, hence the periodic part $\tilde{u}$ of each eigenfunction will be a mere constant, while each A_{ik} and B_{ik} will be a number that depends on (γ_1, γ_2). The system can be solved for few representative values of (γ_1, γ_2), and the solution interpolated to any other values.

5.3.3 Periodicity with potential defect

Suppose a periodic potential (26) is given, except that it is modified in one of the periodicity cells, called the "defect cell". Due to its purely local character, the system of basis functions at all levels needs then calculations in only two cells: One which represents every regular cell, and one representing the defect cell and few meshsizes around it. One can therefore inexpensively reach very coarse levels with meshsizes much larger than the periods (a_1, a_2), at which the Bloch representation can be assumed outside the defect, so there the system can be solved similarly to the above description.

For problems with many atoms in each periodicity cell, the number of levels with meshsize greater than the period will be significantly less than the number of levels with smaller meshsize, hence the overall computational work will be only a fraction more than twice the work for the purely periodic case.

5.3.4 Periodicity with atomic defect

More usually in electronic structure calculations, the defect (departure from the periodicity) is in the nuclei positions, not in the potential, meaning that even outside the periodicity cell $V(x)$ is not periodic. However, far enough from the defect, the fine-scale structure of $V(x)$ does maintain periodicity (cf. Sec. 5.5). This implies that at each level ℓ of the algorithm, the basis functions calculated in just one periodicity cell (or, at coarser levels, a larger cell of linear size comparable to h_ℓ) can represent all the regions whose distance from the defect is large compared with h_ℓ. It can then be shown that, for large problems, the total solution cost is just a fraction more than in the previous case (Sec. 5.3.3).

5.3.5 Bounded atomic structure

This is just a special case of the former. If the electronic density $\rho = \Delta V$ vanishes outside a bounded domain Ω, the potential V will be very smooth away from Ω. In this case, the basis functions away from Ω can readily be written analytically, with no computation, as pure exponential functions (Fourier components).

5.4 High eigenfunctions and dual-space interpolation

It can be shown that the matrix operators A^ℓ_α and B^ℓ_α in the eigensystem (23) tend to a limit as $\lambda \to \infty$, so one can use one eigenvalue interval J^ℓ_α to cover all values of sufficiently large λ. This would substantially reduce the computational cost wherever $\lambda - V$ is large compared with variations in V.

One can much further reduce the number of λ intervals by realizing more generally that A^ℓ_α and B^ℓ_α are smooth functions of $\hat\lambda^\ell_\alpha$, and can therefore be represented by few values of $\hat\lambda^\ell$ with interpolation to any other value of $\hat\lambda^\ell$. The level-ℓ eigensystem will then have the form

$$A^\ell(\lambda) W^\ell = \lambda B^\ell(\lambda) W^\ell \ . \tag{28}$$

Furthermore, one can expect similar interpolations to be possible also between the component equations of (28), provided suitable parameter(s) are introduced on which these equations depend, thereby substantially reducing the size of that system. We intend to explore what parameters can best be used and how to take full advantage of all such possibilities.

5.5 Preliminary comments on self consistency. Quasi localness

From the point of view of the MEB solver, the important property of the DFT self-consistent potential $V(x)$ is that its *non-smooth* part (e.g., scale-h fluctuations) depends only *locally* on the electronic density $\rho(y) = \sum |u_i(y)|^2$, or on the eigenfunctions $\{u_i(y)\}_i$ themselves (i.e., the dependence decays like $\exp(-|y-x|/h)$. We call this property *quasi-localness.*

Indeed, a quasi-local potential V can be relaxed at each level ℓ simultaneously with the relaxation of the scale-h_ℓ basis functions. Such a simultaneous relaxation leaves undetermined only smooth components of the eigenfunctions and hence only smooth components of V. Such scale-h_ℓ-smooth corrections to V, calculated later at coarser levels, will have secondary feedback effect on the scale-h_ℓ basis functions (except for shifting the eigenvalue itself), hence only one, or very few, transitions from coarse levels back to finer ones will be needed throughout the algorithm to account for such feedbacks.

Quasi locality makes of course a very good physical sense. Indeed, all existing self-consistent functionals have this property, LDA or not (due to a well known corresponding property of Poisson's equation). So the fact that this is the only property that really matters for the MEB solver may bring to mind many new and flexible ways to construct the exchange-correlation potential as a multiscale functional, perhaps even directly in terms of the h_ℓ basis functions.

The dependence of the potential on the location of the nuclei is also quasi local. It is therefore feasible to incorporate nucleus positioning (to minimize the total energy) into the same one-shot MEB solver. A basic ingredient of such a process has been described in Sec. 6.1 of [9], but in the many-atom MEB framework it will be used only at finer levels (to transfer to coarser levels the effect of moving each nucleus on the energy minimization equations). At coarser levels (with meshsize comparable to the inter-nucleus distance) distributive moves of nuclei (moving simultaneously several nuclei at a time so as to leave their gravity center unchanged) to lower total energy should accompany the relaxation of the basis functions. At the next coarser levels similar distributive moves should be made with small blocks of nuclei; then with larger blocks at still coarser levels; etc.

6 Anticipated performance and benefits. Upscaling

It is not yet really clear whether the high-dimensional ($d > 1$) MEB algorithm presented above can *generally* reach the ideal $O(N \log N)$ efficiency (or $O(N \log N_{local})$ – see Sec. 1.1), in terms of both CPU time and storage. This will depend on the full efficiency of the local orthonormalization steps and the sparsity they can produce (see Sec. 5.2). On the other hand, even better efficiency may be obtainable when additional devices, like the dual space interpolations (Sec. 5.4), are fully utilized. These factors remain to be studied.

However, more important even than the exact computational complexity of the algorithm in the worst scenario are other advantages it can offer, due to its true local nature. This localness makes it possible to detect and separate out localized eigenfunctions (see Sec. 5.3.1). It yields very inexpensive solvers at unbounded domains, without detailed resolution of superlarge periodicity cells, clusters or such (Secs. 5.3.3, 5.3.4, 5.3.5). More generally, the localness of the algorithm makes it possible to practically solve just once for all repeating regions, i.e., all small or large regions, within the same problems *or in other problems*, which exhibit the same atomistic structure. Only the coarsest level of calculation in each such region should be repeated, to account for interactions with surrounding regions. The coarse-level basis functions calculated in such a region by one researcher can be used by others for other problems.

This of course will fit very well into the general Systematic Upscaling paradigm (see Sec. 1.1). Moreover, upscaling can lead beyond the electronic calculations

and link directly to molecular dynamics (MD) at coarser levels, by including one-shot nucleus positioning in the coarsening process (cf. Sec. 5.5). It means that electronic calculations may only need to be done in rather small regions. In a work on upscaling a polymer [14], for example, it has been shown that MD simulations of rather short parts of the polymer are enough to derive a coarse-level MD Hamiltonian that can accurately (and much faster) reproduce the behavior (all the statistics of interest) of the fine-level system. Similarly, the basic force field (the *fine*-level MD Hamiltonian) can be derived from electronic calculations done only with even shorter parts of the polymer: either isolated parts or (later, to enhance accuracy) temporary "windows" in selected locations of the MD simulation (see [12]).

The local nature of the algorithm also allows of course very efficient utilization of many parallel processors, using domain decomposition.

Other potential benefits mentioned above are the practical elimination of the need to iterate for self consistency, and the enhanced flexibility in introducing multiscale exchange-correlation functionals (Sec. 5.5). Also, as partly described in Sec. 2, the MEB structure is particularly suited for various fast calculations in terms of the eigenfunctions, calculations that can only much more slowly be performed when the eigenfunctions are each explicitly represented at the fine level (requiring also much larger, $O(N^2)$ storage).

Acknowledgements

This research was supported by The Israel Science Foundation Grant No. 295/01, by the US Air Force Office of Scientific Research Contracts F33615-97-D5405 and F33615-03-D5408, and by US Air Force EOARD Contract F61775-00-WE067.

References

[1] A. Brandt and I. Livshits, AMG wave ray algorithm for solving eigenvalue problem for Helmholtz operators, *Numer. Linear Algebra Appl.*, to appear.

[2] Oren E. Livne, Multiscale Eigenbasis Algorithms, Ph.D. Thesis, The Weizmann Institute of Science, 2000.

[3] R.G. Parr and W. Yang, *Density Functional Theory of Atoms and Molecules*, Oxford University Press, New York, 1889.

[4] A. Brandt, Multilevel computations of integral transforms and particle interactions with oscillatory kernels, *Comp. Phys. Comm.* **65** (1991) 24–38.

[5] A. Brandt, Multi-level adaptive solutions to boundary value problems, *Math. Comp.* **31** (1977) 333-390.

[6] A. Brandt, Guide to multigrid development, in *Multigrid Methods* (W. Hackbusch and U. Trottenberg, eds.), Springer-Verlag, 1982, pp. 220–312.

[7] A. Brandt and D. Ron, Multigrid solvers and Multilevel Optimization Strategies, in *Multilevel Optimization and VLSICAD* (J. Cong and J.R. Shinnerl, eds.), Kluwer Academic Publishers, Boston, 2002, pp. 1–69.

[8] A. Brandt and I. Livshits, Wave-ray multigrid method for standing wave equations, *Electronic Trans. Num. An.* **6** (1997), 162–181.

[9] A. Brandt, The Gauss Center Research in Multiscale Scientific Computation, *Elec. Trans. Num. An.* **6** (1997) 1–34.

[10] O.E. Livne and A. Brandt, Multiscale Eigenbasis Calculations: N Eigenfunctions in $O(N \log N)$. In Barth, T.J., Chan, T.F. and Haimes, R. (eds.): *Multiscale and Multiresolution Methods: Theory and Applications*, Lecture Notes in Computational Science and Engineering, Springer-Verlag, Heidelberg, **20** (2001) 347–358.

[11] A. Brandt, J. Bernholc and K. Binder (Eds.), *Multiscale Computational Methods in Chemistry and Physics*. NATO Science Series: Computer and System Sciences, Vol. 177, IOS Press, Amsterdam (2001).

[12] A. Brandt, Multiscale scientific computation: review 2001. In Barth, T.J., Chan, T.F. and Haimes, R. (eds.): *Multiscale and Multiresolution Methods: Theory and Applications*, Springer Verlag, Heidelberg, 2001, pp. 1–96.

[13] A. Brandt, Multiscale Computation: from fast solvers to systematic upscaling. In: *Proceedings of the Second M.I.T. Conference on Computational Fluid and Solid Mechanics*, (K.J. Bathe, ed.), Elsevier (2003), 1871–1873.

[14] D. Bai and A. Brandt, Multiscale computation of polymer models. In [11], pp. 250–266.

[15] A. Brandt and D. Ron, Renormalization multigrid (RMG): Statistically optimal renormalization group flow and coarse-to-fine Monte Carlo acceleration, Gauss Center Report WI/GC–11, June 1999. *J. Stat. Phys.* **102** (2001) 231–257.