Finite-element method for electronic structure

Steven R. White and John W. Wilkins*
Laboratory of Atomic and Solid State Physics, Clark Hall, Cornell University, Ithaca, New York 14853-2501

Michael P. Teter
Process Research Center, Second Floor, Corning Glass Works, Sullivan Park, Corning, New York 14831
(Received 25 July 1988)

We discuss the use of the finite-element method in electronic-structure calculations. Products of orthogonal or nonorthogonal one-dimensional (1D) finite-element shape functions are used to form 3D basis functions on a cubic grid. The strict locality of these functions means that the matrix for any local operator is very sparse, making calculation times proportional to the number of basis functions (N) possible. The completeness of the basis can be increased globally by decreasing the grid spacing and locally by increasing the number of basis functions per site. We discuss algorithms, including the highly efficient multigrid method, for solving the Poisson equation and for the ground state of the single-particle Schrödinger equation in $O(N)$ time. Results are presented for test calculations of H, H_2^+, He, and H_2 using as many as 500 000 basis functions.

I. INTRODUCTION

Many, if not most, electronic-structure calculations are done within a basis set, and one of the first and most important decisions facing a physicist or chemist planning such a calculation is the choice of basis set. A number of different basis sets have been developed and used with great success in the past. For example, Gaussian basis sets (usually multiplied by spherical harmonics) have been used extensively in quantum-chemistry calculations of small- and medium-sized atoms and molecules. Plane-wave basis sets have been used very successfully in density-functional calculations for solids (frequently coupled with pseudopotentials to treat core electrons). We will briefly discuss these two examples below in order to set the stage for a new kind of basis set which we introduce in this paper. These basis functions are based on the finite-element method, a calculational tool widely used in fields ranging from solid and fluid mechanics to structural engineering. We hope to convince the reader that, notwithstanding the successes of existing basis sets, there is room for new types of basis sets and that the finite-element method has great potential for use in large, accurate electronic-structure calculations.

Gaussian basis functions have a number of desirable properties which motivate their extensive use in quantum-chemistry calculations. One major advantage of Gaussian basis functions is that all of the integrals required to compute the matrix elements of the Hamiltonian can be done analytically and yield simple formulas which can be evaluated quickly and to great accuracy on a computer. Since the number of integrals needed is very large, this is a significant advantage. Gaussians also allow for increased resolution where needed—in other words, the completeness of the basis set can be increased in a small region (i.e., near the nuclei) by adding more basis functions there. For atomic problems and for representing core electrons the spherical symmetry of Gaussians is another advantage.

Unfortunately for very large or very accurate calculations some disadvantages of Gaussians become important. The most significant disadvantage is that the number of integrals needed for the electron-electron repulsion terms in the Hamiltonian generally[1] grows as N^4, where N is the number of basis functions $f_i(\mathbf{r})$. These integrals have the form

$$\int d^3\mathbf{r}_1 \int d^3\mathbf{r}_2 f_i(\mathbf{r}_1) f_j(\mathbf{r}_1) \frac{1}{|\mathbf{r}_1-\mathbf{r}_2|} f_k(\mathbf{r}_2) f_l(\mathbf{r}_2) \,. \quad (1)$$

Even though these integrals can be done quite quickly, the number needed grows so rapidly that calculations are currently limited to N not much more than a few hundred. Another drawback is that there is no straightforward way to extrapolate to the infinite-N limit, i.e., to a complete basis.

Planes waves share with Gaussians the property that all the integrals needed are known analytically. Unlike Gaussians, the Coulomb interaction is local in Fourier space—hence solving Poisson's equation, an important step in density-functional calculations, is trivial. The fast Fourier transform allows one to go to real space, where the potential is local,[2] and back to Fourier space with the calculation time proportional to $N \ln N$.

The basis of plane waves has two important disadvantages. The first is that periodic boundary conditions *must* be used. In solids, this is desirable; however, in calculations on clusters or molecules it is not. The second, and more important disadvantage is that the resolution of the basis is exactly the same everywhere. If resolution is used that is fine enough to properly describe core electrons in atoms like silicon, millions of plane waves are needed to describe the simplest silicon unit cell. Pseudopotentials can be used to eliminate core electrons, and the combination of the density functional method in a plane-wave basis with pseudopotentials has had a number of notable successes. The situation is not entirely satisfactory, how-

ever, for two reasons. First, one generally has no way to check the accuracy of the pseudopotential approximation for an atom in its current electronic environment—in other words, temporarily turning off the pseudopotential approximation is not feasible in a plane-wave basis. Second, for accurate calculations some atoms require pseudopotentials which are fairly deep,[3] necessitating a large basis. Both of the difficulties would not arise if one could locally vary the resolution while still retaining the advantages of plane waves. Recent developments in generalized Wannier functions may provide a way of achieving this.[4]

The finite-element method has proven to be very useful in electronic-structure calculations of small atoms and diatomic molecules.[5–9] For example, a recent calculation of H_2 has established a new benchmark Hartree energy to ten-significant-figure accuracy.[5] Unfortunately, the calculations to date use finite-element basis functions on a subset of the system's coordinates (e.g., the cylindrical symmetry of the H_2 Hartree wave function reduces the problem to two dimensions) and cannot be applied readily to more complicated systems. An intriguing question has been whether the finite-element method could be used as successfully in representing, for example, the fully three-dimensional Hartree-Fock orbitals in a several-atom molecule. This work is a first step in that direction. The methods developed here are designed to be useful in a many-atom system with little or no symmetry. Although this means that even calculations on small systems are tedious, the extension to the many-atom case is straightforward.

The finite-element basis functions we discuss here have most of the advantages of both Gaussians and plane waves. Like Gaussians, their resolution can be varied locally (although in the form presented here the extent to which the resolution can be varied is limited). Like plane waves, they have a form of completeness which can be systematically increased. This property allows straightforward extrapolations to the complete basis set limit. The *orthogonal* finite-element basis functions introduced here, also like plane waves, form an orthonormal set. In addition, as we demonstrate in this paper, using the multigrid method one can then iteratively solve Poisson's equation and find the ground state of the single-particle Hamiltonian in calculation times *strictly proportional to* N, usually in less than a dozen iterations.

The two main disadvantages to the finite-element basis functions are that the integrals required to construct the Hamiltonian matrix are not known analytically (and hence must be computed numerically), and storing these matrix elements requires considerable computer memory. These problems are not insurmountable, and as a demonstration of the potential utility of these functions we present here calculations of the electronic structure of H, H_2^+, He, and H_2 (the latter two in the Hartree approximation). It should be emphasized, however, that these methods are still in their infancy, with much room for improvement. Advances in increasing the variability of the resolution between the core and the valence regions, for example, could result in increases in computation speed by orders of magnitude.

In this paper we consider both the basis functions themselves and the algorithms one can use with them in electronic structure calculations. Section II describes the basis functions and explains how they are derived. In Sec. III we use the basis functions to solve a single-particle Hamiltonian, presenting results for the hydrogen atom and the hydrogen molecular ion as test cases. Section IV addresses the additional algorithms needed for doing many-body calculations. One of these algorithms expresses the product of two wave functions written in terms of the basis functions as a single function in terms of the basis functions. This gives us densities needed for the Poisson equation, which we solve to find the electrostatic potential of the charge distribution. We solve the Poisson equation using the multigrid method, an extremely efficient algorithm for solving elliptic differential equations on a grid. We also discuss the more difficult case of adapting multigrid to finding eigenstates of a single-particle Hamiltonian. In Sec. V we use the methods discussed in the previous sections to calculate the Hartree solutions for the hydrogen molecule. Finally, in Sec. VI we conclude with suggestions and discussion of future directions for the finite-element method in electronic-structure calculations.

II. BASIS FUNCTIONS

The finite-element method has been used extensively in engineering[10] and occasionally in electronic-structure calculations.[5–9] The finite-element basis functions, called shape functions, have many real advantages: (a) they possess polynomial completeness; (b) the shape functions are strictly zero outside their respective elements, which results in (c) very sparse Hamiltonian and overlap matrices regardless of the form of the interactions; (d) the mesh can be varied to give high resolution where needed; (e) the ease of representing any arbitrary function is such that potentials and densities can be represented as easily as wave functions; and (f) error in fits do not propagate beyond a local region. Perhaps their most important advantage is (c), which means that potentially all the calculations relevant to electronic structure can be done in times *proportional* to the number of basis functions N, whereas dense matrices require at least N^2 operations. There are many problems, however. Efficient $O(N)$ algorithms have been developed for only some of the calculations relevant to electronic structure. It is not yet known how to define an optimal mesh for a system with many nuclei. For calculations involving moving nuclei, more efficient ways of doing the integrals to calculate matrix elements are needed.

Because in some cases efficient $O(N)$ algorithms are known only for orthogonal basis sets, we have developed orthogonal shape functions. In particular, an efficient method for solving Poisson's equation, the multigrid method, which was developed primarily for finite-difference problems, can be adapted to orthogonal shape functions with little modification (see Sec. IV). Probably the multigrid method can be modified to be just as efficient for the nonorthogonal case, but at present the orthogonal functions have a definite advantage. The penal-

ty one pays for using orthogonal shape functions is that more functions are needed to obtain the same order of completeness than in the nonorthogonal case. In this section we present first nonorthogonal and then orthogonal shape functions; our test calculations for simple atoms and molecules were done with orthogonal shape functions.

A. Nonorthogonal shape functions

We discuss first shape functions in one dimension; the three-dimensional (3D) generalization is straightforward and is discussed at the end of this section. We assume we have a one-dimensional lattice with unit spacing. On each site n there will be centered one or more shape functions, with each shape function extending only out to the neighboring sites. In particular,

$$S_{n,t}(x)=S_t(x-n) , \tag{2}$$

where t indexes the different types of functions on each site, and where $S_t(x)=0$ for $|x|\geq 1$.

In the simplest case, where only one shape function per site is allowed, we have

$$S_0(x)=1-|x| \quad \text{with} \quad |x|\leq 1 . \tag{3}$$

These shape functions are complete to first order, by which we mean that they can represent any straight line exactly. Any functions formed from them will be continuous, but anything except a straight line will have slope discontinuities.

The next set, which has been used several times in electronic-structure calculations,[6–8] is related to Hermite's interpolation formula. The set consists of two shape functions per site, generates functions with continuity up to first derivatives, and is able to fit any third-order polynomial exactly (third-order completeness). The shape functions are formed from piecewise cubic polynomials. One function, $S_0(x)$, corresponds to the function value at $x=0$, and the other, $S_1(x)$ corresponds to the derivative at 0. In particular, $S_0(x)$ for $0\leq x\leq 1$ is defined as the cubic polynomial with the boundary conditions $S_0(0)=1$, $S_0(1)=S_0'(0)=S_0'(1)=0$. Similarly, S_1 is the cubic polynomial satisfying $S_1(0)=S_1(1)=S_1'(1)=0$ and $S_1'(0)=1$. For $0\leq x\leq 1$, we have

$$\begin{aligned} S_0(x)&=1-3x^2+2x^3 , \\ S_1(x)&=x-2x^2+x^3 , \end{aligned} \tag{4}$$

and

$$S_t(x)=(-1)^t S_t(|x|), \quad x<0 . \tag{5}$$

Note that an approximate fit to a function can be generated very rapidly by specifying the coefficient of $S_{n,0}$ to be the value of the function at site n, and similarly the coefficient of $S_{n,1}$ to be the derivative at site n. This Taylor-series fit is exact for cubic polynomials.

Nonorthogonal shape functions of this kind can be generated to any order one wishes. Using m shape functions at each site, representing derivatives up to order $m-1$, we can represent exactly any polynomial up to order $2m-1$. These functions are also continuous through derivatives of order $m-1$.

B. Orthogonal shape functions

The first example of *orthogonal* shape functions which we are going to develop will have three functions per site. If the *nonorthogonal* functions above were used, we would have a basis set complete to fifth order, but we will find that making the functions orthogonal reduces the completeness to second order.

Shape function $S_t(x)$ corresponds to the term of order t in a Taylor series at $x=0$; in particular, we require

$$\frac{1}{j!}\frac{d^j}{dx^j}S_t(x=0)=\delta_{jt}, \quad j,t=0,1,2 . \tag{6}$$

Hence the leading term in $S_2(x)$ is x^2, etc. We also require that derivatives of $S_t(x)$ of order 0–2 vanish at $x=\pm 1$, which guarantees continuity through the second derivative.

The polynomial completeness conditions can be put in equation form by requiring that the Taylor-series fit to the monomials 1, x, x^2, etc. be *exact*. For the interval $0\leq x\leq 1$, only the functions centered at 0 and 1 contribute, and the Taylor-series fit $f_T(x)$ to an arbitrary function $f(x)$ is given by

$$\begin{aligned} f_T(x)=&f(0)S_0(x)+f'(0)S_1(x)+\tfrac{1}{2}f''(0)S_2(x) \\ &+f(1)S_0(x-1)+f'(1)S_1(x-1) \\ &+\tfrac{1}{2}f''(1)S_2(x-1) . \end{aligned} \tag{7}$$

Thus the completeness conditions through quadratic order are [using (6) to evaluate the coefficients in (7)]

$$\begin{aligned} &S_0(x)+S_0(x-1)=1 , \\ &S_1(x)+S_0(x-1)+S_1(x-1)=x , \\ &S_2(x)+S_0(x-1)+2S_1(x-1)+S_2(x-1)=x^2 . \end{aligned} \tag{8}$$

The coefficients of $S_t(x-1)$ in these equations form a Pascal's triangle.

Assuming (5) also holds for the orthogonal functions, we need only specify the functions over the interval from 0 to 1. It is useful to change to the variable u through

$$u=2x-1, \quad -1\leq u\leq 1 . \tag{9}$$

The purpose of this transformation is to be able to expand our shape functions in Legendre polynomials $P_j(u)$, which will be of great help in enforcing orthogonality (and which also reduces roundoff error). We assume the expansion

$$S_t(x)=\sum_{j=0}^{J} C_{tj}P_j(u(x)) , \tag{10}$$

where J as well as the C_{tj}'s remains to be determined. Note that $S_t(x(u)-1)=(-1)^t S_t(x(-u))$.

The orthogonality conditions are different from those usually found because we require orthogonality between functions on *different* sites. An orthogonalization procedure such as Gram-Schmidt cannot produce this form

of shifted orthogonality and retain the locality of the functions. The equations to be satisfied between functions on sites 0 and 1 are

$$\int_0^1 dx\, S_t(x)S_{t'}(x-1)=0 \quad (t,t'=0,1,2)\ . \tag{11}$$

Using (10) this becomes

$$\sum_{j=1}^{J} C_{tj}C_{t'j}\frac{(-1)^{t+j}}{2j+1}=0\ . \tag{12}$$

In writing the completeness conditions (8), we stopped with second order. We now show that the orthogonality conditions make third-order completeness impossible with only three functions per site (with the derivative conditions we have imposed). From (7) the completeness condition for x^3 would be

$$S_0(x-1)+3S_1(x-1)+3S_2(x-1)=x^3\ . \tag{13}$$

Note first that (11) and (13) would imply

$$\int_0^1 dx\, S_t(x)x^3=0\ . \tag{14}$$

Now (8) and (13) can be combined to give

$$-S_0(x)+3S_1(x)-3S_2(x)=(x-1)^3\ . \tag{15}$$

Multiplying by x^3 and integrating, we find a clear contradiction. This inconsistency means we cannot impose (13).

We used a computer algebra package to solve the six unique orthogonality integrals while imposing the completeness and derivative conditions. The nonlinear equations had to be solved iteratively using a linearization procedure. We found that $J\geq 9$ was necessary to satisfy all the conditions; we set $J=9$, which results in the same number of equations as unknowns. Note that so far we have only imposed orthogonality with functions on *neighboring* sites; the functions still need to be orthogonalized with the other functions on the same site. Because $S_1(x)$ is an odd function, it is already orthogonal to $S_0(x)$ and $S_2(x)$, but $S_0(x)$ and $S_2(x)$ are not orthogonal. To orthogonalize them, we subtract $S_1(x)$ times an appropriate constant from $S_2(x)$. Finally, for convenience we normalize each function. The resulting orthonormal functions are plotted in Fig. 1; the coefficients are given in Table I. In the Appendix we give the coefficients for and plot the next higher-order orthonormal basis set, with four shape functions per site and cubic completeness.

To obtain 3D shape functions, we use products of the 1D functions in the form

$$S_t(\mathbf{r})=S_{t_x}(x)S_{t_y}(y)S_{t_z}(z)\ . \tag{16}$$

For the quadratic-complete orthonormal functions we could form 27 different 3D shape functions per site, but not all of these combinations are useful in obtaining higher-order completeness. Since we cannot represent three of the third-order terms (x^3, y^3, and z^3) in this way, it makes little sense to use combinations representing higher-order monomials such as x^2y^2 [which corresponds to $S_2(x)S_2(y)S_0(z)$]. Also, for regions where little resolution is needed (such as the tails of atoms), a lower degree

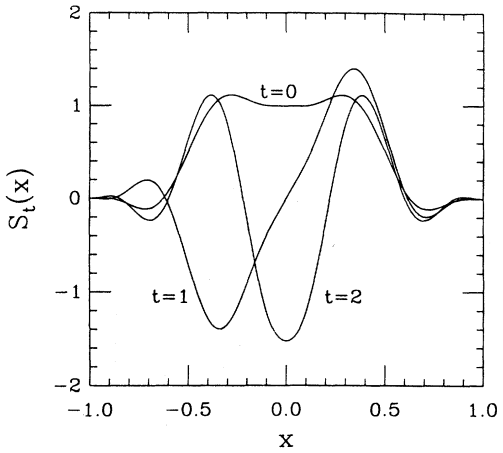

FIG. 1. Second-order orthonormal shape functions $S_t(x)$, $t=0,1,2$. These finite-element basis functions are orthogonal not only among themselves but also with the shifted functions $S_{nt}(x)\equiv S_t(x-n)$, $n=\pm1,\pm2,\ldots$. The functions are piecewise polynomials with $S_t(x)\equiv 0$ for $|x|\geq 1$. The full set $S_{nt}(x)$ can represent any second-order polynomial exactly. Products of these one-dimensional functions are used to form basis functions in three dimensions.

TABLE I. Coefficients for orthonormal shape functions having second-order completeness. These finite-element basis functions are expressed in terms of Legendre polynomials through (10) and are plotted in Fig. 1.

t	j	C_{tj}
0	0	0.5
0	1	−0.786 889 902 441 989 970
0	2	0.0
0	3	0.491 615 313 206 678 780
0	4	0.0
0	5	−0.299 950 601 616 016 280
0	6	0.0
0	7	0.114 628 031 815 739 560
0	8	0.0
0	9	−0.019 402 840 964 412 152
1	0	0.445 927 185 090 425 120
1	1	−0.538 200 231 906 334 480
1	2	−0.504 166 671 807 620 870
1	3	0.922 261 373 873 345 610
1	4	0.052 033 360 721 463 265
1	5	−0.562 701 865 684 613 440
1	6	0.011 283 001 415 129 673
1	7	0.215 040 099 986 343 020
1	8	−0.005 076 875 419 397 167
1	9	−0.036 399 376 268 740 846
2	0	0.0
2	1	0.358 836 047 975 377 270
2	2	−1.094 977 487 117 334 100
2	3	0.986 392 104 229 709 470
2	4	0.297 516 730 737 903 800
2	5	−0.828 869 858 721 492 330
2	6	0.064 514 028 065 764 367
2	7	0.288 053 362 008 504 730
2	8	−0.029 028 595 428 002 422
2	9	−0.042 436 331 750 430 691

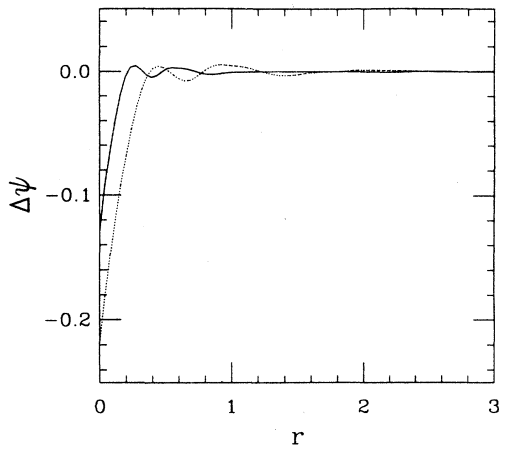

FIG. 2. Errors in calculated hydrogen atom wave functions. Approximations to the H wave function were obtained by diagonalizing the Hamiltonian matrix for two different finite element basis sets. The curves show the error from the exact solution as a function of the distance from the nucleus r along the line $y=z=0$. (Both functions are scaled so that the exact solution is e^{-r}). For both curves $R_1=4$ and $R_2=2$ (see Table II); for the dashed curve $D=7$ and $a=1$, while for the solid curve $D=12$ and $a=0.6$. For both cases the error is much larger near the nucleus.

of completeness may be sufficient. If only linear completeness is desired, four combinations suffice, representing the four 3D monomials up to linear order. Quadratic completeness requires 10 combinations, and up to 17 combinations can be used to get as much of third-order completeness as possible. In the calculations presented in this paper, we put different numbers of functions (4, 10, or 17) on different sites depending on their distance from the center of the atom or molecule. This allowed us a certain amount of variability in the resolution of the basis, but as our test calculations on hydrogen show, more variability would have been desirable in order to make the errors in the wave function near the nucleus and in the tails comparable (see Fig. 2).

III. ONE-ELECTRON PROBLEMS

The solution of the one-electron Schrödinger equation in a basis involves two separate calculations: first, construct the Hamiltonian matrix, and second, diagonalize that matrix to find the ground state (and possibly some excited states). The finite-element basis requires that more attention be paid to the first calculation than is often the case, since the integrals involved are not known analytically. The fact that the resulting matrix is both very large and very sparse means that standard matrix diagonalization techniques are not appropriate. In this section we show how to do both parts of the calculation in an efficient manner. We also give results for two test cases, the hydrogen atom and the hydrogen molecular ion, for which analytic solutions are available.

The basic principle used in solving the Schrödinger equation is the variational method: the ground-state energy E of the Hamiltonian H is given by

$$E=\min_{\psi}\left\{\frac{\langle\psi|H|\psi\rangle}{\langle\psi|\psi\rangle}\right\}, \tag{17}$$

and the ground state ψ is the minimizing state. In our case we minimize wave functions of the form

$$\psi(\mathbf{r})=\sum_{\mathbf{n},\mathbf{t}}\psi_{\mathbf{nt}}S_{\mathbf{nt}}(\mathbf{r}) \tag{18}$$

by varying the $\psi_{\mathbf{nt}}$'s. This immediate leads to the generalized eigenvalue equation

$$\sum_j H_{ij}\psi_j=E\sum_j O_{ij}\psi_j\ , \tag{19}$$

where i and j index basis functions on all sites, i.e., $i=\{\mathbf{n},\mathbf{t}\}$, where

$$H_{ij}=\int d^3r\, S_i(\mathbf{r})HS_j(\mathbf{r}) \tag{20}$$

and

$$O_{ij}=\int d^3r\, S_i(\mathbf{r})S_j(\mathbf{r})\ . \tag{21}$$

Since we use *orthonormal* basis functions, $O_{ij}=\delta_{ij}$, and the generalized equation reverts to the standard eigenvalue equation.

The fact that each 3D basis function is made of a product of 1D basis functions makes the calculation of the kinetic-energy part of the Hamiltonian matrix very simple and the calculation of the potential-energy part tractable. In the kinetic-energy case we have (setting $\hbar=m=e=1$)

$$\int dx\int dy\int dz\, S_{n_xt_x}(x)S_{n_yt_y}(y)S_{n_zt_z}(z)(-\tfrac{1}{2}\nabla^2)S_{n_x't_x'}(x)S_{n_y't_y'}(y)S_{n_z't_z'}(z)$$
$$=K_{n_xt_x,n_x't_x'}\delta_{n_yn_y'}\delta_{t_yt_y'}\delta_{n_zn_z'}\delta_{t_zt_z'}+\delta_{n_xn_x'}\delta_{t_xt_x'}K_{n_yt_y,n_y't_y'}\delta_{n_zn_z'}\delta_{t_zt_z'}+\delta_{n_xn_x'}\delta_{t_xt_x'}\delta_{n_yn_y'}\delta_{t_yt_y'}K_{n_zt_z,n_z't_z'}\ , \tag{22}$$

where

$$K_{n_xt_x,n_x't_x'}=\int dx\, S_{n_xt_x}(x)\left[-\frac{1}{2}\frac{\partial^2}{\partial x^2}\right]S_{n_x't_x'}(x)\ . \tag{23}$$

Note that because of the translational invariance of our grid, $K_{n_xt_x,n_x't_x'}$ depends on n_x and n_x' only through their difference n_x-n_x', and K is 0 for $|n_x-n_x'|>1$. Because the shape functions are expressed in terms of Legendre polynomials, the integrals K are easily evaluated analytically.

In the simplest case of a single atom centered at the origin, the potential-energy matrix elements have the form

$$V_{\mathrm{nt},\mathrm{n't'}} = \int d^3r\, S_{\mathrm{nt}}(\mathbf{r}) \frac{1}{r} S_{\mathrm{n't'}}(\mathbf{r}) \ . \tag{24}$$

To evaluate these, we use an integral transform of $1/r$:

$$\frac{1}{r} = \frac{2}{\sqrt{\pi}} \int_0^\infty dw \frac{1}{w^2} e^{-r^2/w^2} \ . \tag{25}$$

Now r appears in a form which separates into products of x, y, and z terms. Thus we have

$$V_{\mathrm{nt},\mathrm{n't'}} = \int_0^\infty dw \frac{1}{w^2} \left[\int dx\, S_{n_x t_x}(x) S_{n_x' t_x'}(x) e^{-x^2/w^2} \right] \times [x \to y][x \to z] \ . \tag{26}$$

The integrals in the square brackets can be evaluated analytically, but the resulting expressions are both time consuming to evaluate and susceptible to round off error even in double precision. We have found it more expedient to evaluate the integrals numerically as they are needed, storing[11] the results so that the same integral is never done twice. The outer integral must be done numerically. Despite its superficial appearance, the integrand is well behaved and finite near $\omega=0$. The simplest way to deal with the upper limit is to break the integral into two parts,[12] from 0 to ω_c, and from ω_c to ∞, and change variables in the second interval to $1/\omega$. Because the integration routine for the outer integral tends to use the same values of ω for many different integrals, each square-bracketed integral is typically used thousands of times, so the time spent in evaluating them is generally negligible. This means that using the integral transform for $1/r$ for practical purposes has transformed the integrals in (24) into well-behaved one-dimensional numerical integrals.

The number of integrals of the form (24) which need to be calculated is reduced, typically by a factor of 100, by three symmetry relations: (1) symmetry under the interchange of (n_x,t_x) and (n_x',t_x') with (n_y,t_y) and (n_y',t_y') (plus $x \leftrightarrow z$ and $y \leftrightarrow z$); (2) symmetry under $(n_x,t_x) \leftrightarrow (n_x',t_x')$ (and similarly for y and z); and (3) symmetry under $(n_x,n_x') \leftrightarrow (-n_x,-n_x')$ (and similarly for y and z; depending on the parity of the shape functions t_x and t_x' the exchange may cause a change of sign). We calculate $V_{\mathrm{nt},\mathrm{n't'}}$ only once for each symmetry-related set; we obtain the other elements of the set by using these symmetries.[13]

Because of the very short range of the finite-element functions, the Hamiltonian matrix is very sparse. Each basis function overlaps basis functions on 27 sites; those on its site plus the basis functions on the 26 surrounding "neighbor"[14] sites. For the case where there are 10 basis functions per site, each basis function will overlap 270 other basis functions. In practice the actual number of nonzero overlaps is much smaller. In the case of the kinetic-energy matrix the orthogonality of the functions coupled with their factorization into x, y, and z terms results in about 20 nonzero overlaps for the 10 functions per site case [cf. Eq. (22)]. The orthogonality of the basis also enhances the sparsity of the potential energy matrix: since any constant potential would result in a diagonal potential matrix, only the gradient or higher derivatives of the potential can produce off-diagonal terms. Furthermore, note that the potential-energy matrix element corresponding to the overlap function $o(\mathbf{r}) = S_{\mathrm{nt}}(\mathbf{r}) S_{\mathrm{n't'}}(\mathbf{r})$ is the Coulomb energy of the charge $o(\mathbf{r})$ in the potential $1/r$. Orthogonality implies that the total charge is zero, so if $o(\mathbf{r})$ is *spherically symmetric,* then the matrix element is zero. In practice $o(\mathbf{r})$ is never completely spherically symmetric, but if we neglect elements whose absolute value is less than, say, 10^{-8} (as we do in our calculations), the sparsity can often be enhanced dramatically, as is shown in Fig. 3.

The sparseness of the Hamiltonian matrix allows the use of large numbers of basis functions without exorbitant computer resources provided efficient ways of dealing with the matrices are known. In some of the larger calculations presented below, the number of basis functions N was more than 500 000. Standard matrix diagonalization routines, with calculation times proportional to N^3, are not feasible for N this large. Fortunately, there are sparse matrix algorithms which can find a few eigenvalues and eigenvectors in a calculation time not much worse than N, one of which we describe later.

The algorithms we use for finding eigenvalues and eigenvectors are based on the *inverse iteration* method.[15] In this iterative method one repeatedly solves the system of equations

$$(H - \lambda)\phi' = \phi \ , \tag{27}$$

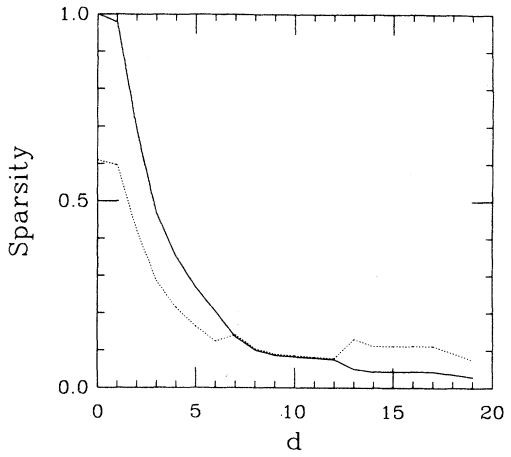

FIG. 3. Sparsity of the potential-energy matrix. For a typical site d spacings from a helium nucleus, the figure shows in two different ways the average number of matrix elements in the corresponding row of the matrix having magnitude greater than 10^{-8}. The number of basis functions per site T was varied as in our electronic-structure calculations, with $T=17$ for $d \le 6$, $T=10$ for $7 \le d \le 12$, and $T=4$ for $d \ge 13$. The solid curve shows the sparsity normalized to unity at $d=0$ (the actual sparcity was 280 at $d=0$). The dashed curve shows the sparsity divided by the maximum possible number of nonzero matrix elements given that each basis function only overlaps with the functions on the 27 sites in its "neighborhood," i.e., $27 \times T$ (symmetry may reduce this number somewhat). For sites fairly far from the nucleus the orthogonality of the basis increases the sparseness dramatically; in a nonorthogonal basis we would expect the dashed curve to remain near unity even for fairly large d.

where λ and ϕ are approximations to the desired eigenvalue and eigenvectors. At each iteration the solution ϕ' of the previous step becomes the new ϕ. If λ is close to an eigenvalue of H, the matrix $(H-\lambda)$ is nearly singular and the component of ϕ in the direction of the eigenvector will be greatly enhanced. For sparse matrices we solve (27) using iterative matrix algorithms such as the conjugate-gradient method[16] or simultaneous overrelaxation[17] (SOR). (These methods both work only for positive-definite matrices, which means that λ must be less than the minimum eigenvalue of H. In finding the ground state, this is not a problem, but the methods must be modified to deal with excited states. In this paper we consider only ground states.) The convergence rate can be increased by making λ closer to the desired eigenvalue; this can be done by correcting λ at each step through

$$\lambda' = \frac{\phi' H \phi'}{|\phi'|^2} = \lambda + \frac{\phi' \phi}{|\phi'|^2} . \tag{28}$$

With a good initial guess for λ this algorithm often converges in only two or three steps. There is a catch, however: most iterative methods for solving (27) converge slowly if λ is very close to the correct eigenvalue. Often the best course is to stop "improving" λ once it is fairly close to the correct eigenvalue.

We can speed up the inverse iteration algorithm often by an order of magnitude by combining the iterations to solve (27) with the iterations to project out the eigenstate. The key idea is that there is no point solving (27) to great accuracy until ϕ is almost the correct eigenstate. We illustrate this by describing the method we used to find the ground state of the hydrogen atom (see below). We start off with a guess for λ and ϕ. Since the conjugate-gradient method is iterative, it needs an initial "guess" for the solution ϕ'; we use the guess

$$\phi'_{\text{initial}} = \frac{|\phi|^2}{\phi(H-\lambda)\phi} \phi . \tag{29}$$

We iterate the conjugate-gradient method until the squared relative error in the solution

$$R^2 = [(H-\lambda)\phi' - \phi]^2 / \phi'^2 \tag{30}$$

is less than 0.1. We then substitute the improved but still relatively inaccurate ϕ' for ϕ, construct ϕ'_{initial} from (29) again, and continue the conjugate-gradient iterations until $R^2 < 0.01$ (this will usually be about the same number of iterations as to get to 0.1, since we are starting closer to the correct solution). We then substitute ϕ' for ϕ again and continue, each time reducing the error criterion by a factor of 10. We stop when the energy obtained from ϕ' differs from that obtained with ϕ by less than 10^{-10}.

Since the inverse iteration technique has reduced the problem of finding eigenvalues and eigenvectors to solving the system of Eq. (27), it is important to have as efficient a method as possible for solving this system. A *very* efficient iterative method for solving (27) when $H-\lambda$ is positive definite is the multigrid method, which we describe in the next section in the context of Poisson's equation. After discussing the simpler problem of solving Poisson's equation we will return to the problem of using multigrid to solve (27). The results presented for H_2^+ were obtained using the multigrid method.

Table II summarizes our results for the hydrogen atom. The first five runs were done with a relatively large lattice spacing to determine how far out the lattice should extend to get an accurate ground state energy. The next three runs show the effects of varying the parameters R_1 and R_2. From the sixth run it is apparent that the region within 2 a.u. of the nucleus is very sensitive to the resolution of the basis; most of the error in the energy comes

TABLE II. Finite-element calculations for the hydrogen atom. The calculations were done a $(2D+1)^3$ lattice with spacing a using the quadratic-complete orthonormal shape functions discussed in Sec. II. There were four (functions per site for sites whose distance from the nucleus r was greater than R_1 (linear completeness), 10 functions per site for $R_2 \le r \le R_1$ (quadratic completeness), and 17 functions per site for $r < R_2$ (nearly cubic completeness). The total number of basis functions is N. The ground state was found using inverse iteration coupled with the conjugate-gradient method. The second to the last column shows the total number of conjugate-gradient steps taken, and the last column shows how closely the approximate solutions satisfied the virial theorem. (The procedure used for finding the ground state in the last calculation was slightly different from the others so the number of iterations is not shown.)

D	a	ΔE (%)	R_1	R_2	N	Iter.	$1-\frac{1}{2}V_{\text{pot}}/E_{\text{kin}}$ (%)
5	1.0	1.623	4.0	2.0	7097	85	0.68
6	1.0	1.537	4.0	2.0	10 561	98	1.12
7	1.0	1.522	4.0	2.0	15 273	99	1.20
8	1.0	1.520	4.0	2.0	21 425	98	1.22
10	1.0	1.519	4.0	2.0	38 817	99	1.23
7	1.0	2.689	4.0	0.0	15 042	96	2.41
7	1.0	1.438	4.0	3.0	15 903	93	1.19
7	1.0	1.369	5.0	2.0	16 821	99	1.53
6	1.2	2.503	4.0	2.0	9947	98	2.14
12	0.6	0.395	4.0	2.0	71 119	142	0.256
18	0.4	0.136	4.0	2.0	231 231	193	0.042
24	0.3	0.0065	4.0	2.0	538 313		−0.007

TABLE III. Finite-element calculations for the hydrogen molecular ion. The basis set was the same as that in Table II, with R_1 and R_2 measured from the center of the molecule. The separation between the nuclei was 2.0 a.u., and the spacings a were chosen so the nuclei always fell on grid sites. The lowest eigenstate of the Hamiltonian matrix was found using inverse iteration coupled with the multigrid method. The exact energy (excluding the nuclear repulsion) is (Ref. 23) −1.102 634 2.

D	a	ΔE (%)	R_1	R_2	N
10	0.5	1.084	2.0	1.5	39 447
12	0.5	1.077	2.0	1.5	64 903
10	0.5	0.141	4.0	2.0	51 497
15	$\frac{1}{3}$	0.462	2.0	1.5	127 437
20	$\frac{1}{3}$	0.452	2.0	1.5	283 957
15	$\frac{1}{3}$	0.514	2.0	0.0	124 714
15	$\frac{1}{3}$	0.158	3.0	0.0	137 590
15	$\frac{1}{3}$	0.108	4.0	0.0	162 082
15	$\frac{1}{3}$	0.101	3.0	2.0	144 065
15	$\frac{1}{3}$	0.051	4.0	2.0	168 557
20	0.25	0.278	2.0	1.5	294 813
20	0.25	0.029	4.0	2.0	392 909
25	0.2	0.184	2.0	1.5	568 155

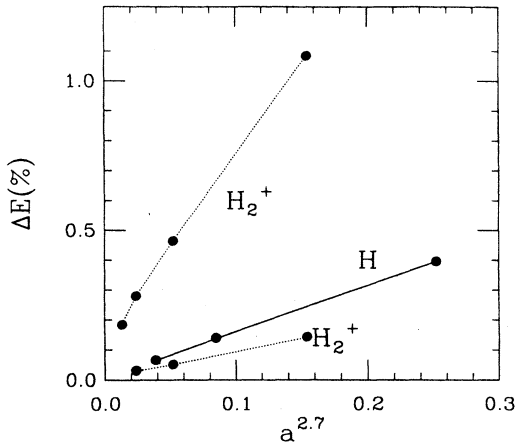

FIG. 4. Extrapolations to the infinite basis-set limit $a \to 0$. The solid curve shows the last three calculations in Table II. The exponent 2.7 was set by requiring that the last *four* calculations in Table II fall on as straight a line as possible; the fourth point ($a=1.2$) is off the scale. Extrapolating to $a=0$ gives $\Delta E \approx 0.01\%$. The same exponent was then used for two sets of calculations for the hydrogen molecular ion. The upper curve is for the runs in Table III with $R_1=2$, $R_2=1.5$, and $Da=5$; the lower curve is for the runs with $R_1=4$, $R_2=2$. The lower curve also extrapolates to $\Delta E \approx 0.01\%$. In both of these cases, the extrapolation has reduced the error by about an order of magnitude from that of the last data point; probably that is all that can be expected from this type of extrapolation. A smaller exponent would be more appropriate for the upper curve, which is not as straight as the lower, since in the calculations for the upper curve the higher-order basis functions were confined to a relatively small region, and significant errors in the energy resulted from the use of only a linearly complete basis in regions of the molecule where the wave function varied rapidly.

from this region. It is also clear that the fitting ability of 17 functions per site is significantly better than 10 functions per site, even though the order of completeness is the same (quadratic order). The last four runs show the results of systematically decreasing the grid spacing. These runs can be used to extrapolate to the complete basis limit $a \to 0$.

Table III give results for H_2^+. The grid was chosen so that both nuclei fell exactly on lattice sites. Hence the potential-energy matrix could be calculated quite simply from the matrix for a single hydrogen nucleus (by adding the matrix to itself with a shift). Surprisingly, more accurate calculations are possible for H_2^+ than for H, since the tails of the wave function fall off faster for H_2^+. This means that fewer basis functions have to be "wasted" in the tails, so more can be concentrated near the nuclei.

Figure 4 shows how to extrapolate to the complete basis-set limit $a \to 0$. Both for H and H_2^+ extrapolation reduces the error in the energy by about an order of magnitude over the most accurate data point.

IV. MANY-PARTICLE ALGORITHMS

Several additional algorithms must be developed pertaining to the electron-electron interaction in order to perform electronic-structure calculations for many-particle systems. The exact nature of the algorithms depends on what electronic-structure method is to be used. In this paper we consider only the Hartree approximation, one of the simplest methods. (For the ground states of the two-electron systems we consider here, the Hartree and Hartree-Fock approximations are identical.)

In the Hartree approximation we write the spatial part of the wave function as a simple product of single-particle orbitals ψ^α, each of which is an eigenstate of an effective single particle Hamiltonian. The effective Hamiltonian for particle α includes, in addition to the kinetic energy and nuclear potential terms, an additional potential $U^\alpha(\mathbf{r})$ arising from the other electrons:

$$U^\alpha(\mathbf{r}) = \int d^3r' \frac{1}{|\mathbf{r}-\mathbf{r}'|} \sum_{\gamma\,(\neq\alpha)} |\psi^\gamma(\mathbf{r}')|^2 \,. \tag{31}$$

In order to use the potential $U^\alpha(\mathbf{r})$ to find ψ^α we must calculate the matrix

$$U^\alpha_{ij} = \int d^3r\, S_i(\mathbf{r}) U^\alpha(\mathbf{r}) S_j(\mathbf{r}) \,, \tag{32}$$

which is analogous to the matrix V of the previous section. Once we have U^α_{ij}, the techniques discussed in the previous section can be used to find ψ^α. Since the orbitals ψ^α are expressed in terms of the S_i's, an approach often taken to evaluate U^α_{ij} involves tabulating all integrals of the form

$$\int d^3r'\, S_i(\mathbf{r}) S_j(\mathbf{r}) \frac{1}{|\mathbf{r}-\mathbf{r}'|} S_k(\mathbf{r}') S_l(\mathbf{r}') \tag{33}$$

and then summing over these stored integrals. The problem with this approach is that the number of these in-

tegrals grows with the number of basis functions N as N^4 for extended basis functions, and as CN^2 for a localized basis. With the basis functions considered here the constant C is on the order of 10^4. We adopt a different approach with a calculation time *proportional* to N. We present in this section two techniques which allow us to do Hartree calculations very efficiently. The first is a procedure for obtaining an expansion of the charge density in terms of our basis set from the expansions of the wave functions. The second is an efficient way of solving the Poisson equation to obtain the electrostatic potential from the charge density. We demonstrate the utility of these techniques by using them to solve the helium atom and the hydrogen molecule within the Hartree approximation.

A. Collapse of the wave function into a density

The first step in our approach is a simplification of the charge density terms of the form $\rho(\mathbf{r})=|\psi|^2$. In the form $|\psi|^2$, the density is expressed as a double sum over the basis functions; we will collapse it into a single sum. More generally, we would like to be able to collapse the product of any two different functions ψ_i^1 and ψ_i^2 into a single function ρ_i. In a complete basis this means determining coefficients ρ_i which satisfy

$$\sum_{i,j}\psi_i^1\psi_j^2 S_i(\mathbf{r})S_j(\mathbf{r})=\sum_i \rho_i S_i(\mathbf{r}) \ . \tag{34}$$

Of course we cannot satisfy this equation exactly in a finite basis, but the completeness properties of the basis mean that the density can be represented with an accuracy which can be increased systematically by decreasing a. One approach to obtaining the ρ_i's is to multiply (34) by $S_k(\mathbf{r})$ and then integrate over $\mathbf{r}$. This leads to

$$\rho_k=\sum_{i,j} O(i,j,k)\psi_i^1\psi_j^2 \ , \tag{35}$$

where the triplet overlap matrix is

$$O(i,j,k)=\int d^3r\, S_i(\mathbf{r})S_j(\mathbf{r})S_k(\mathbf{r}) \ . \tag{36}$$

The ρ_k's obtained this way minimize the error (in the least-squares sense) in (34). Note that $O(i,j,k)$ is zero unless i, j, and k are all mutual neighbors; nevertheless, the number of terms in the double sum over i and j may be over 10^4. This many terms makes direct use of (35) impracticable for large calculations.

Fortunately, the orthogonality of the basis functions allows us to use an approximation to $O(i,j,k)$ which can be evaluated very rapidly. It turns out that $O(i,j,k)$ is dominated by the terms where i, j, and k index basis functions all on the same site. We will construct a truncated version of $O(i,j,k)$ which only connects basis functions on the same site but for smooth functions ψ constructs a good approximation to the ρ produced with the full $O(i,j,k)$.

Note that because of the product form of the basis functions S_{nt} (returning to the expanded notation $i=\{\mathbf{n},\mathbf{t}\}$), we can factor the triplet overlap matrix as

$$O\begin{bmatrix}\mathbf{n} & \mathbf{n}' & \mathbf{n}''\\ \mathbf{t} & \mathbf{t}' & \mathbf{t}''\end{bmatrix}=O_xO_yO_z \ , \tag{37}$$

where

$$O_x=O\begin{bmatrix}n_x & n_x' & n_x''\\ t_x & t_x' & t_x''\end{bmatrix}=\int dx\, S_{n_xt_x}(x)S_{n_x't_x'}(x)S_{n_x''t_x''}(x) \tag{38}$$

with similar formulas for O_y and O_z. Numerical values for O_x are shown in Table IV. If the functions are not all on the same site the elements are at most a few percent of typical on-site elements. In three dimensions, the largest off-site elements will be those where only one of O_x, O_y, and O_z is off site; others will be smaller by at least a factor of about 50. A possible approximation is to simply neglect all offsite elements of $O(i,j,k)$, but this procedure is not exact even in the limit $a\to 0$. A better procedure is to make an approximation reminiscent of local-density-functional theory: in evaluating ρ_{nt} using (35) (for the particular site $\mathbf{n}$), replace $\psi^1_{\mathbf{n}'\mathbf{t}'}$, where $\mathbf{n}'$ is a neighbor of

TABLE IV. One-dimensional triple overlap matrix. Defined in (38), this matrix is useful in forming basis set expansions for the electronic density from the wave functions. The orthogonality of the basis makes the off-site elements ($n_x''=1$) much smaller than the on-site elements ($n_x''=0$).

t_x	t_x'	t_x''	n_x''	$O\begin{bmatrix}0 & 0 & n_x''\\ t_x & t_x' & t_x''\end{bmatrix}$
0	0	0	0	1.0
0	0	1	0	0.0
0	0	2	0	−0.037 871 563 4
0	1	1	0	0.964 088 611 2
0	1	2	0	0.0
0	2	2	0	0.943 683 669 8
1	1	0	0	0.964 088 611 2
1	1	1	0	0.0
1	1	2	0	0.638 049 715 8
1	2	2	0	0.0
2	2	2	0	−0.306 985 143 6
0	0	0	1	0.0
0	0	1	1	0.007 341 346 9
0	0	2	1	−0.018 935 781 7
0	1	0	1	0.007 341 346 9
0	1	1	1	0.0
0	1	2	1	−0.017 893 015 7
0	2	0	1	0.018 935 781 7
0	2	1	1	−0.017 893 015 7
0	2	2	1	0.0
1	1	0	1	0.017 955 694 4
1	1	1	1	−0.011 422 206 8
1	1	2	1	−0.014 262 026 6
1	2	0	1	0.028 613 189 2
1	2	1	1	−0.029 542 012 1
1	2	2	1	0.006 559 968 0
2	2	0	1	0.028 158 165 1
2	2	1	1	−0.032 860 927 0
2	2	2	1	0.016 382 366 5

$\mathbf{n}$, by $\psi^1_{\mathbf{n},\mathbf{t}'}$ (and similarly for ψ^2). By making $\psi_{\mathbf{n}'\mathbf{t}}$ independent of $\mathbf{n}'$, we have made the calculation of $\rho_{\mathbf{nt}}$ a *local* operation, equivalent to using the on-site triple overlap matrix:

$$\bar{O}(\mathbf{t},\mathbf{t}',\mathbf{t}'')=\sum_{\mathbf{n}',\mathbf{n}''} O\begin{bmatrix}\mathbf{n} & \mathbf{n}' & \mathbf{n}'' \\ \mathbf{t} & \mathbf{t}' & \mathbf{t}''\end{bmatrix} \tag{39}$$

and using instead of (35)

$$\rho_{\mathbf{nt}}=\sum_{\mathbf{t}',\mathbf{t}''}\bar{O}(\mathbf{t},\mathbf{t}',\mathbf{t}'')\psi^1_{\mathbf{nt}'}\psi^2_{\mathbf{nt}''} . \tag{40}$$

In addition to its rapid calculation time, this approximation has several other advantages. First, $\bar{O}$ is symmetric with respect to interchanges of $\mathbf{t}$, $\mathbf{t}'$, and $\mathbf{t}''$, just as O is symmetric with respect to interchanges of i, j, and k. Second, the approximation becomes exact for smooth functions as $a\rightarrow 0$. And third, the total density is preserved exactly; i.e., if ρ_i is given by (40), then

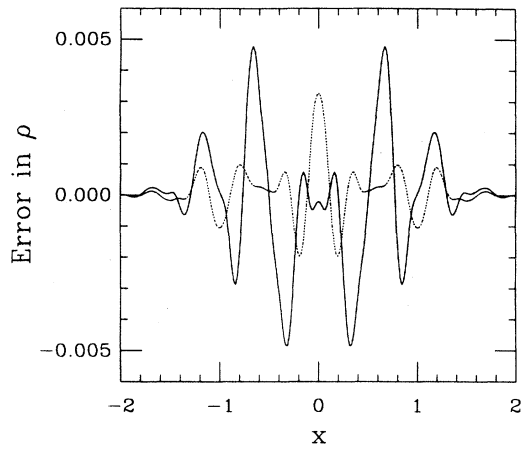

FIG. 5. Errors in reducing the product of two functions into a single function. Given two functions, each expressed as an expansion in terms of a finite-element basis, this figure illustrates errors involved in different methods for writing the *product* of the two functions as an expansion in the basis. The basis used here has 10 functions per site and a lattice spacing $a=0.5$. The two functions $\psi^{\pm}(\mathbf{r})=\sum_i \psi_i^{\pm}S_i(\mathbf{r})$ whose product was taken were approximate fits to the functions $\exp[-(x\pm\frac{1}{2})^2-y^2-z^2]$ (the fits were done with a Taylor-series expansion, but whatever errors are present in the fits are irrelevant since the later comparisons are done with the fits rather than the original Gaussians). The solid curve shows the difference between the exact product $\psi^+(\mathbf{r})\psi^-(\mathbf{r})$ and $\rho(\mathbf{r})$, the best least-squares fit to the exact product, where $\rho(\mathbf{r})=\sum_i \rho_i S_i(\mathbf{r})$ and the ρ_i were evaluated using the full triple overlap matrix O. The same difference function was calculated (but is not shown) for the ρ_i produced by the local approximation $\bar{O}$. The dashed curve shows the additional error arising from the local approximation. Both curves are for the line $y=z=0$. We see that the additional error is fairly small compared to the error arising because of the incompleteness of the basis. The results of additional tests with smaller values of a show that as a is decreased, the relative size of the additional error to the incompleteness error remains approximately the same, but that both errors decrease as a is decreased.

$$\int d^3r \sum_i \rho_i S_i(\mathbf{r})=\int d^3r \sum_{i,j}\psi^1_i\psi^2_j S_i(\mathbf{r})S_j(\mathbf{r}) . \tag{41}$$

To prove this last property, note first that

$$\int d^3r\, S_{\mathbf{nt}}(\mathbf{r})=\delta_{\mathbf{t},0} . \tag{42}$$

This equation and the orthogonality of the S's means that (41) is equivalent to

$$\sum_{\mathbf{n}}\rho_{n0}=\sum_{\mathbf{n},\mathbf{t}}\psi^1_{\mathbf{nt}}\psi^2_{\mathbf{nt}} . \tag{43}$$

From (35) (with $O\rightarrow\bar{O}$), this statement is true provided $\bar{O}(\mathbf{0},\mathbf{t}',\mathbf{t}'')=\delta_{\mathbf{t}'\mathbf{t}''}$, which follows from the symmetry $\bar{O}(\mathbf{0},\mathbf{t}',\mathbf{t}'')=\bar{O}(\mathbf{t}',\mathbf{0},\mathbf{t}'')$ and from

$$\sum_{\mathbf{n}} S_{\mathbf{n},0}(\mathbf{r})=1 . \tag{44}$$

The elements of $\bar{O}(\mathbf{t},\mathbf{t}',\mathbf{t}'')$ can be obtained fairly easily by noting that it can be factorized as

$$\bar{O}(\mathbf{t},\mathbf{t}',\mathbf{t}'')=\bar{O}(t_x,t_x',t_x'')\bar{O}(t_y,t_y',t_y'')\bar{O}(t_z,t_z',t_z'') . \tag{45}$$

In order to test the accuracy of using $\bar{O}$ instead of O, we performed a simple test calculation comparing the two procedures. It is important to note that some of the error is unavoidable because of the incompleteness of the basis. As long as the additional error from using $\bar{O}$ is small compared with the error from the incompleteness of the basis, it is sensible to use $\bar{O}$. As the first step of the test we used a Taylor-series fit to find the expansion coefficients $\psi_i^{\pm}$ of the functions $\exp[(x\pm\frac{1}{2})^2-y^2-z^2]$, using a basis with 10 functions per site. We then used (35) with both O and $\bar{O}$ to determine coefficients of the density ρ_i. We compare $\rho(\mathbf{r})-\psi^+(\mathbf{r})\psi^-(\mathbf{r})$, where $\rho(\mathbf{r})=\sum_i \rho_i S_i(\mathbf{r})$ (and similarly for $\psi^{\pm}$), for the two cases. The results are shown in Fig. 5. We found that, indeed, the extra error from using $\bar{O}$ was small compared to the error due to incompleteness of the basis.

B. Solving the Poisson equation

The second step in obtaining the Hartree potential U^{α} is solving the Poisson equation. This involves two issues: determining appropriate boundary conditions and specifying the method used to solve the linear equations. A number of different approaches are available for each of these problems; we use a multipole expansion to determine the boundary conditions and the multigrid method for solving the system of equations.

In the continuum limit, the solution of the Poisson equation (dropping the superscript α):

$$\nabla^2 U(\mathbf{r})+4\pi\rho(\mathbf{r})=0 \tag{46}$$

is given by

$$U(\mathbf{r})=\int d^3r' \frac{1}{|\mathbf{r}-\mathbf{r}'|}\rho(\mathbf{r}') \tag{47}$$

provided one imposes the boundary condition $U(\mathbf{r})\rightarrow 0$ as $\mathbf{r}\rightarrow\infty$. The problem with using (47) directly is that the calculation time is proportional to N^2; we can solve the linear equations for the finite-element version of (46) much more quickly (with a calculation time proportional to N). To obtain the basis set version of (46), we multiply

both sides of the equation by $S_i(\mathbf{r})$ and integrate over $\mathbf{r}$ (expanding both U and ρ in terms of the basis) to obtain

$$\sum_j [\nabla^2]_{ij} U_j + 4\pi\rho_i = 0 \ . \tag{48}$$

The matrix $[\nabla^2]_{ij}$ is (up to a constant factor) just the kinetic-energy part of the Hamiltonian matrix discussed in the previous section.

Because we use a grid of finite extent, we cannot directly require that $U(\mathbf{r})$ be 0 at infinity. Imposing the correct boundary conditions for a finite region involves using an approximation to the solution (47) at the boundary. We accomplish this with a multipole expansion of $U(\mathbf{r})$ up to the quadrupole moments. The evaluation of the charge, dipole moment, and quadrupole moments of $\rho(\mathbf{r})$ can be done in a single sweep through the grid, using the precalculated moments of the individual basis functions. The multipole moments give an approximation analytic expression for $U(\mathbf{r})$ valid for large $|\mathbf{r}|$. The values U_i for i on the boundary are determined by a Taylor-series fit (see Sec. II) to this expression.[18]

The multigrid method[19] is a procedure developed for solving elliptic differential equations (such as the Poisson equation) very rapidly. Most often the equations are expressed in a finite-difference formulation, but the technique can also be applied to the finite-element method. Multigrid techniques have also been developed to solve a number of additional problems, including nonlinear equations and eigenvalue problems (see below). We will describe just the basic method as we use it here; various enhancements and extensions can be found in the literature.

Relaxation methods, such as Gauss-Seidel,[20] are very efficient at reducing the *high-frequency* error in the solution to (48). They are very slow, however, in reducing the *low-frequency* components of the error, i.e., errors for wavelengths of more than a few lattice spacings. The key idea of the multigrid method is to use relaxation on a sequence of *coarser grids* to reduce all frequency components of the error at the same rate.

We will now describe the basic steps of the multigrid method. We will first describe it in terms of finite differences on just two grids, and then discuss the more general case with finite elements and several grids. First, starting with some guess for U_j, one or more relaxation sweeps are done on (48) (the form of the equation is the same in finite differences as finite elements). The error in the solution after these sweeps will be fairly smooth on the scale of the lattice spacing a. We form the residual

$$R_i = -4\pi\rho_i - \sum_j [\nabla^2]_{ij} u_j \ , \tag{49}$$

where u_j is the current approximation to the correct solution U_j. The error in the solution $\Delta u_j \equiv U_j - u_j$ satisfies

$$\sum_j [\nabla^2]_{ij} \Delta u_j = R_j \ . \tag{50}$$

Since the high-frequency components of Δu are small, it makes sense to solve (50) on a *coarser grid.* Most often one simply doubles a, leaving the boundaries where they are, and obtaining $\frac{1}{8}$ as many sites on the grid. A suitable approximation of the residual R is made on the coarse grid, e.g., a smoothing convolution is done on the fine grid and then the values on the coarse grid are directly copied from the corresponding points on the fine grid. Using some suitable method (e.g., conjugate gradient), one solves for Δu_j on the coarse grid. This is used as an *approximation* to Δu_j for the original fine grid; i.e., we interpolate the coarse-grid solution onto the fine grid and add it into u_j. Since there are many fewer sites on the coarse grid, finding the exact solution there is much faster than on the fine grid. The coarse-grid correction greatly reduces the low-frequency error; the combination of relaxation steps and coarse-grid correction reduces the error for all frequencies. Since only the *error* is passed to the coarser grid, the process can be iterated for increasing accuracy.

Multigrid with several grids is essentially a recursive application of the two-grid algorithm. Instead of solving the coarse-grid equation exactly, relaxation steps are done, the residual computed, and a coarse grid correction is obtained from an even coarser grid. The recursion stops when the grid is small enough to be solved very rapidly with, say, conjugate gradient. There is no need to iterate each coarse-grid equation to an exact solution before passing it to the finer grid, since the solution is only used as an approximate correction to the next finer grid. All one really wants is to get the low-frequency errors to be reduced as fast as the high-frequency errors, so generally only one or two relaxation and correction passes are needed on each grid before the correction is passed along. One multigrid "pass" consists of a few relaxations on the finest grid plus sweeps through all the coarser grids to obtain a correction to the finest grid. Generally the finest grid residual will be reduced by roughly constant factor (which can be as high as 10) with each pass. Since there are so many fewer points on the coarse grids, the computational time spent on them is usually negligible compared to that on the finest grid. Normally very accurate solutions (with errors on the order of 10^{-10}) can be found with only about 20 times as much work as a single fine-grid relaxation, regardless of the size of the problem.

One possible way to adapt multigrid to our finite-element basis set would involve using basis functions defined on coarser grids, but in our calculations we used a different method. To determine a "coarse-grid" correction to the error in the finite-element version of (48), a finite-difference grid is used with the same spacing as the finite-element grid. (Since the basis functions have 4 to 17 degrees of freedom per site, this is in some sense a "coarser grid.") The advantage of the method is that the standard finite-difference multigrid methods discussed above can be used for the coarser grids. Then the only new things to be specified are how to transfer R_i to the finite difference mesh, and how to interpolate Δu_j back to the basis functions. We should emphasize that since the finite-difference grids only provide *approximate* corrections to the finite-element solution, there is no single correct set of procedures for the transfer to and from the finite-difference grids. Poor choices of the procedures only affect the speed of convergence, not the final answer.

For the transfer to the finite-difference mesh, we have found that using the finite-element coefficient $R_{n,0}$ as the value of R_n on the finite-difference grid works well. For the transfer back, we calculate the derivatives Δu_n up to quadratic order for each site of the grid using standard finite-difference formulas, and then use a Taylor-series fit to get the finite-element coefficients.

C. Multigrid for eigenstates

Several procedures are available for adapting the multigrid to calculate eigenstates.[21] In the method we use inverse iteration is coupled with multigrid, with multigrid taking the same role as the conjugate gradient method in the algorithm discussed in Sec. III. We use multigrid to solve the system

$$\sum_j [H-\lambda]_{ij}\phi'_j=\phi_i \ . \tag{51}$$

We use the same finite-difference coarse grids as the Poisson equation. There are two principal differences between applying multigrid to (51) and to (48). The first is the presence of the potential-energy terms V_{ij} in the matrix. These terms present no difficulty to the method, but we must specify how to write the potential-energy terms on the finite-difference grids. We chose one of the simplest procedures: for the finest finite-difference grid we set

$$V^{\rm FD}_{n,n'}=\delta_{n,n'}V_{n0;n'0} \ , \tag{52}$$

where $V_{n0;n'0}$ is the diagonal element of the finite-element potential energy matrix. This is an especially convenient choice since the $V_{n0;n'0}$ are already tabulated. We determine the matrix elements of V for the coarser grid by working our way down from the finest grid, taking for the coefficient of $V^{\rm FD}_{n,n'}$ on a coarse grid a weighted average of $V^{\rm FD}_{n,n'}$ on points in the neighborhood of n on the next finer grid.

The second difference between (48) and (51) is that, depending on the choice of λ, $[H-\lambda]_{ij}$ may not be positive definite (whereas the corresponding matrix for the Poisson equation is always positive definite). Gauss-Seidel relaxation converges only for positive-definite matrices,[22] which in this case means λ must be less than the minimum eigenvalue of H. Thus the method can only be used to find the ground state (unless the ground state is found first and the ϕ_i is orthogonalized to it at each step; we only calculate ground states in this paper).

As with inverse iteration coupled with the conjugate gradient method, we can combine the two types of iterations—multigrid and inverse iteration. The result is an algorithm which is much faster than the conjugate-gradient algorithm. As a comparison of the two methods, we applied them both to the same problem: the third calculation in Table III. The outer loop in both calculations was the same, inverse iteration with $\lambda=2$, and both required 13 steps (for convergence of the energy to within 10^{-8}). The inner loops were different, with the conjugate-gradient method taking as many steps as necessary to achieve a given accuracy (which was increased for each inverse iteration step; see Sec. III). A total of 230 conjugate-gradient passes were required. In the multigrid calculation, only one pass was necessary for each inverse iteration step. Each multigrid pass had three fine-grid relaxations, for a total of 39 relaxations (the computer time for the coarse-grid relaxations was very minor). The dominant part of a conjugate-gradient pass is a multiplication by the Hamiltonian matrix, which takes roughly the same amount of time as a fine-grid relaxation. Hence the multigrid calculation (neglecting set-up times) was about $\frac{230}{39}=5.9$ times as efficient as the conjugate-gradient calculation. For larger calculations we expect the difference to be even greater.

TABLE V. Finite-element Hartree calculations for the helium atom. The basis set was the same as that in Table II. In each of the runs $R_1=4$ and $R_2=2$. The exact Hartree energy for helium is (Ref. 8) $-2.861\,680\,0$.

D	a	E	ΔE (%)
10	0.6	−2.8101	1.8
11	$\frac{6}{11}$	−2.8209	1.4
15	$\frac{6}{11}$	−2.8209	1.4
15	0.4	−2.8436	0.63
19	$\frac{6}{19}$	−2.8522	0.33

V. HARTREE CALCULATIONS

The procedure for carrying out Hartree calculations is a combination of the algorithms discussed above. First, we use a guess for the Hartree orbital to determine an electronic density. The density is used to find an electrostatic potential via the Poisson equation, the potential is put into an effective Hamiltonian, and the ground state of the effective Hamiltonian becomes the new orbital. This process is iterated until self-consistency is reached. In the Hartree calculations presented here, we have used the conjugate gradient method coupled with inverse iteration for diagonalizing the effective Hamiltonian, and the multigrid method for solving the Poisson equation.

One new technique is needed to obtain a complete Hartree procedure. After the Poisson equation has been solved to obtain the Hartree potential U_i^α, we must still find the matrix elements of U^α with the basis functions as given in (32). It is clear that the exact matrix elements can be obtained using the triple overlap matrix O, but as in Sec. IV we instead use the local approximation $\bar{O}$. An

TABLE VI. Finite-element Hartree calculations for the hydrogen molecule. The basis set was the same as that in Table III, with an internuclear separation of 1.4 a.u. The exact Hartree energy (excluding the internuclear repulsion) is (Ref. 6) $-1.847\,915\,3$. For the first two runs, $R_1=5$ and $R_2=3$, and for the last run $R_1=4$ and $R_2=2$.

D	a	E	ΔE (%)
11	0.7	−1.841 50	0.35
15	0.7	−1.841 50	0.35
19	0.35	−1.847 35	0.03

important property of $\bar{O}$ is its symmetry with respect to interchanges of t, t', and t''; this guarantees the matrix obtained from U^{α} using $\bar{O}$ is symmetric. Since the Hartree potential is usually quite smooth compared to the wave functions or density, the local approximation is quite good.

TABLE VII. Coefficients for orthonormal shape functions having third-order completeness.

t	j	C_{tj}
0	0	0.5
0	1	−0.793 000 822 653 327 836
0	2	0.0
0	3	0.484 225 548 092 482 332
0	4	0.0
0	5	−0.265 078 315 259 477 876
0	6	0.0
0	7	0.085 858 192 877 879 419
0	8	0.0
0	9	−0.012 004 603 057 556 039
1	0	0.448 584 202 954 864 000
1	1	−0.557 718 642 566 173 813
1	2	−0.492 273 006 292 793 268
1	3	0.921 709 410 002 345 821
1	4	0.011 713 918 004 094 849
1	5	−0.504 568 952 474 117 419
1	6	0.055 871 919 994 894 147
1	7	0.163 428 601 842 841 202
1	8	−0.028 893 199 978 292 442
1	9	−0.022 850 416 804 895 791
1	10	0.004 996 165 317 232 713
2	0	0.0
2	1	0.344 635 683 647 103 622
2	2	−1.091 146 864 218 575 225
2	3	1.066 133 335 081 282 872
2	4	0.073 026 142 500 267 162
2	5	−0.715 675 450 255 091 707
2	6	0.348 313 074 232 240 353
2	7	0.034 376 292 862 846 634
2	8	−0.180 124 100 080 427 303
2	9	0.149 087 511 437 541 317
2	10	0.031 146 767 484 934 356
2	11	−0.070 460 383 193 024 763
2	12	0.0
2	13	0.010 687 990 500 902 682
3	0	0.013 101 010 284 009 823
3	1	−0.039 303 030 852 029 470
3	2	0.116 246 329 145 453 556
3	3	0.225 402 676 931 866 057
3	4	−0.900 899 255 704 480 010
3	5	0.432 614 442 173 855 762
3	6	1.289 424 862 991 679 046
3	7	−1.439 352 143 094 399 434
3	8	−0.676 924 038 192 844 153
3	9	1.213 984 450 479 868 221
3	10	0.203 436 459 672 219 361
3	11	−0.463 681 250 089 613 053
3	12	−0.055 399 111 961 588 983
3	13	0.070 334 854 450 451 917
3	14	0.011 013 743 765 551 359

In our calculations we were able to reduce the calculation time substantially by combining all of the iterations involved in the calculation: those in solving Poisson's equation, those in the inverse iteration procedure for finding the ground state, and those for attaining overall self-consistency. The method for combining them is very similar to that described in Sec. III. In the first step a guess for the orbital, which does not have to be very accurate, is used [using (39)] to obtain a density to use in the Poisson equation. The boundary conditions are set using a multipole expansion, and then enough multigrid passes are done to determine the Hartree potential to a specified, initially relatively poor accuracy. The matrix elements of the Hartree potential are found and added to the Hamiltonian. Then we use the initial guess for the orbital as the starting guess for an inverse-iteration conjugate-gradient pass to a limited accuracy. We use the same accuracy as we used for the Poisson equation with multigrid. The improved orbital is used to start the process all over again, this time with higher accuracy. The whole process is iterated to convergence.

Tables V and VI show results for Hartree calculations of helium and the hydrogen molecule. More accurate results are possible for H_2 because the nuclear cusp in the Hartree orbital varies more sharply for He than for H_2. Since the spherical symmetry in He is not used in our calculations, the computational difficulty is roughly the same except for the difference in the nuclear cusps.

VI. SUMMARY AND CONCLUSIONS

In this paper we have demonstrated that the finite-element method is practical for electronic-structure calculations of small molecules. Specifically, in Sec. II we discussed nonorthogonal finite-element basis functions (shape functions), which have been used extensively in engineering and occasionally in electronic structure, and introduced orthogonal shape functions, which to our

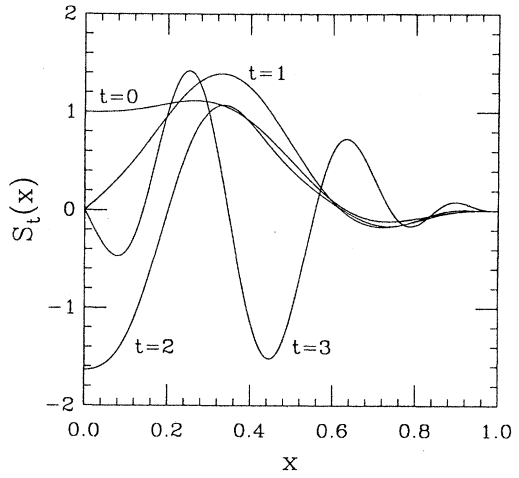

FIG. 6. Third-order orthonormal shape functions $S_t(x)$, $t=0,1,2,3$. These functions have the same orthonormality properties as the second-order functions in Fig. 1. This basis set can represent any third-order polynomial exactly.

knowledge have not been used before. The strict locality of these functions give them many advantages, including flexibility (e.g., more functions can be placed near the nuclei) and calculation times proportional to the number of basis functions.

In Sec. III we discussed the use of the finite element for single-particle problems. Included were techniques for quickly doing the integrals needed to set up the Hamiltonian matrix and efficient ways of solving the sparse eigenvalue problem which gives the ground state. The eigenvalue methods were all based on inverse iteration, sometimes coupled with the conjugate-gradient method, and, as discussed in Sec. IV, the multigrid method. These techniques were used to find the ground states of H and H_2^+.

Section IV dealt with some of the algorithms needed in order to do many-particle systems. One of these is an efficient way of representing the product of two functions (each expressed as an expansion in the basis functions) in terms of a single similar expansion in the basis. This algorithm is useful in obtaining an expansion of the electronic density, to be used in solving the Poisson equation, from the single-particle orbitals. We discussed the multigrid method for solving the Poisson equations, a very efficient and flexible method able to solve a wide variety of matrix problems arising from partial differential equations. We coupled the multigrid method with a multipole expansion to produce an algorithm for obtaining the electrostatic potential from a charge density. In Sec. V we used these methods to solve the Hartree equations for He and H_2.

The most important difference between this work and previous applications of the finite-element method to electronic structure is the three dimensional, cartesian coordinate basis used here. Previous applications have been to atoms or diatomics where symmetry could be used to reduce the dimensionality of the problem. This approach has produced very accurate results, but cannot be extended to larger molecules. While our test calculations have been done only on atoms and diatomics, the extension of the method to larger molecules will be straightforward.

We believe there is only one major hurdle remaining before the finite-element method can become competitive with other commonly used bases: adapting the basis to better deal with the region near the nuclei. We have shown that the computational advantages resulting from the locality of the shape functions make calculations with very large (10^5–10^6) bases feasible. Unfortunately, the difficulty of fitting the nuclear cusps meant that so many functions were needed per atom that calculations could be done only on small systems. An advance in adapting the basis to the nuclei which reduced the number of basis functions needed per atom from 10^5 to 10^3 would allow a basis the same size as we used here to fit a 100 atom system. Such an advance, which we believe is quite conceivable, might allow the previous successes of the finite-element method in atoms and diatomics to be extended to much larger systems.

ACKNOWLEDGMENTS

This work was supported in part by the U.S. Department of Energy (Division of Materials Research, Office of Basic Energy Sciences). This research was conducted with use of the Cornell National Supercomputer Facility (Ithaca, NY), funded in part by the National Science Foundation, by New York State, and by IBM.

APPENDIX

In this appendix we describe cubic-complete orthonormal shape functions, the next higher order than the ones used in the calculation in the main text. Table VII shows the coefficients for and Fig. 6 shows plots of the shape functions. The procedure by which this set, which has four shape functions per site, was generated was very similar to the procedure for the quadratic-complete set, but more effort was required to obtain solutions to the nonlinear equations. It is not clear to what order this can be continued, since in solving nonlinear sets of equations there is no guarantee of real solutions.

---

*Present address: Physics Dept.–OSU, 174 West 18th Avenue, Columbus, Ohio 43210-1106.

[1]For large molecules where not all of the basis functions overlap it is possible to construct neighbor lists to make the number of integrals needed $\propto N^2$.

[2]Plane-wave matrix elements cannot be evaluated with the fast Fourier transform when nonlocal pseudopotentials are used. See J. Ihm, A. Zunger, and M. L. Cohen, J. Phys. C **12**, 4409 (1979), for the matrix elements with nonlocal pseudopotentials.

[3]Oxygen is an example of an atom requiring a deep pseudopotential. See G. B. Bachelet, D. R. Hamann, and M. Schlüter, Phys. Rev. B **26**, 4199 (1982).

[4]D. J. Sullivan, J. J. Rehr, J. W. Wilkins, and K. G. Wilson (unpubished); K. G. Wilson (unpubilshed).

[5]D. Heinemann, D. Kolb, and B. Fricke, Chem. Phys. Lett. **137**, 180 (1987); W. Schulze and D. Kolb, *ibid.* **122**, 271 (1985).

[6]F. S. Leven and J. Shertzer, Phys. Rev. A **32**, 3285 (1985), and references therein; W. K. Ford and F. S. Levin, *ibid.* **29**, 43 (1984).

[7]J. L. Gázquez and H. J. Silverstone, J. Chem. Phys. **67**, 1887 (1977).

[8]D. P. Carroll, H. J. Silverstone, and R. M. Metzger, J. Chem. Phys. **71**, 4142 (1979).

[9]M. Friedman, Y. Rosenfeld, A. Rabinovitch, and R. Theiberger, J. Comput. Phys. **26**, 169 (1978).

[10]O. C. Zienkiewicz, *The Finite Element Method* (McGraw-Hill, London, 1977).

[11]Storage methods are discussed in D. E. Knuth, *The Art of Computer Programming,* Vol. 3 of *Sorting and Searching* (Addison-Wesley, Reading, Mass., 1973), p. 506.

[12]The final result must be independent of ω_c, but the *efficiency* of the numerical integration will depend on a sensible choice of this parameter.

[13]In particular, we obtain the other symmetry related matrix elements by applying all possible combinations of the symmetry interchanges to the given element. To avoid double counting, only those interchanges which generate *different* elements are applied.

[14]In this paper we will consistently use the term neighbor in this sense.

[15]W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical Recipes: The Art of Scientific Computing* (Cambridge University Press, Cambridge, 1986), p. 377.

[16]J. Stoer and R. Bulirsch, *Introduction to Numerical Analysis* (Springer-Verlag, New York, 1980), p. 572.

[17]W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, Ref. 15, p. 655.

[18]We tested the adequacy of carrying out the expansion to only the quadrupole moments in our calculations on H_2 by performing two test calculations with expansions carried out only to the monopole moment (charge) and then to the quadrupole moment. The resulting total energies differed by less than 10^{-6}.

[19]W. Hackbusch and U. Trottenberg, *Multigrid Methods,* Vol. 960 of *Lecture Notes in Mathematics* (Springer-Verlag, Berlin, 1982).

[20]W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, Ref. 15, p. 653.

[21]W. Hackbusch, SIAM (Soc. Ind. Appl. Math.) J. Numer. Anal. **16**, 201 (1979); R. E. Bank, *ibid.* **19**, 886 (1982); A. Brandt, S. McCormick, and J. Ruge, SIAM (Soc. Ind. Appl. Math.) J. Sci. Stat. Comput. **4**, 244 (1983); F. F. Grinstein, H. Rabitz, and A. Askar, J. Comput. Phys. **51**, 423 (1983).

[22]J. Stoer and R. Bulirsch, Ref. 16, p. 547.