

Dispersion fitted finite difference method with applications to molecular quantum mechanics

Stephen K. Gray
Chemistry Division
Argonne National Laboratory
Argonne, Illinois 60439
gray@anchim.chm.anl.gov

Evelyn M. Goldfield
Department of Chemistry
Wayne State University
Detroit, Michigan 48202
evi@sun.science.wayne.edu

Abstract

An approach to finite difference approximation is presented based on the idea of fitting the dispersion relation up to a limiting accuracy. The resulting approximations to the second derivative can be more accurate than the standard, Lagrangian finite difference approximations by an order of magnitude or more. The locality of the methods makes them well suited to parallel computation, in contrast with pseudospectral methods. The approach is illustrated with application to a simple bound state problem and to a more challenging three dimensional reactive scattering problem.

For the final form of this manuscript, see *J. Chem. Phys.* **115**, 8331 (2001).

I. INTRODUCTION

Grid representations are common in quantum mechanical studies of molecular spectroscopy and dynamics,¹⁻⁵ and many other areas of science and engineering.^{6,7} In molecular quantum mechanics, our main interest, grid representations are well suited to iterative methods, e.g., the Lanczos method⁸ for bound states and wave packet methods for dynamics,⁹ because they lead to favorable computational scalings. Iterative methods involve repeated actions of a Hamiltonian matrix, \mathbf{H} , on a wavefunction or wave packet represented as a vector \mathbf{y} of function values at the grid points. These repeated $\mathbf{H}\mathbf{y}$ products are the computational bottleneck. The numerical effort is largely determined by the choice of representation and how \mathbf{H} is approximated. Often \mathbf{H} is decomposed into kinetic and potential matrices, $\mathbf{H} = \mathbf{T} + \mathbf{V}$, with \mathbf{V} approximated by a diagonal matrix. The main issue is then how to approximate the action of the kinetic energy matrix, $\mathbf{T}\mathbf{y}$. For Cartesian like degrees of freedom \mathbf{T} is proportionate to a second derivative matrix, \mathbf{D}'' . We present here a new, accurate and efficient method for approximating the kinetic energy matrix and its action, within the context of finite difference (FD) ideas.¹⁰⁻¹³

In current molecular quantum mechanics, FD methods are less commonly used than pseudospectral methods.¹⁴ Pseudospectral methods represent a function and its derivatives not only in terms of values at grid points, but in terms of an underlying spectral or basis set representation. The classic pseudospectral method is the Fourier method,^{1,2,14} wherein to evaluate $\mathbf{D}''\mathbf{y}$, \mathbf{y} is transformed into a Fourier (spectral) representation, multiplied by a diagonal matrix corresponding to \mathbf{D}'' in that representation, and then transformed back to the grid representation. The basis need not be a Fourier basis, and one need not explicitly transform to and from the spectral representation but develop a (grid) matrix representation of \mathbf{D}'' consistent with the basis. In this sense, other discrete variable representations (DVRs)^{3,4,15} are pseudospectral methods. Whereas FD methods are local approximations

to derivatives, such as pseudospectral methods are global approximations and the second derivative matrix with respect to one degree of freedom, \mathbf{D}'' , is a full matrix. (When several degrees of freedom are treated, and the kinetic energy operator is of a typical form, the overall kinetic energy matrix is not full but factors into blocks corresponding to the different degrees of freedom.) The inherent locality of derivatives, however, suggests local methods should suffice. FD methods represent \mathbf{D}'' with banded matrices of varying degrees of sparsity, which may result in significant computational gains. They are easily programmed, and portable. Another compelling reason for considering FD methods is their potential, by reducing the amount of communication between processors, to vastly improve the scalability of parallel computing algorithms. The significance for chemical physics research is that the high performance (parallel) computing environment can then be used to tackle large (four or more atoms) molecular quantum mechanics problems.¹⁶⁻¹⁸

Kosloff^{19,20} pointed out that, unlike the Fourier method, FD methods do not in general obey the correct dispersion relation, a matter we discuss below. He also showed how the Fourier method can vastly outperform the standard FD approach, particularly if high accuracy is desired. The success of the Fourier method and other pseudospectral methods, combined with the limitations of FD methods has prevented FD methods from receiving a great deal of attention for quantum chemical applications.

If low order FD methods are used, accuracy is achieved by decreasing the grid spacing. Alternatively, accuracy can be obtained by using higher order FD methods. For example, Thachuk and Schatz¹⁰ explored the use of higher order FD methods for calculating thermal rate coefficients. They showed that under certain circumstances, FD methods are competitive with the Fourier method (applied with fast Fourier transforms). Mazziotti¹¹ compared several high order local methods including the FD method. Applied to accurate calculations of energy levels of a Morse oscillator, he showed that high order FD methods can approach pseudospectral accuracy, while exhibiting exponential convergence with increasing order. Guantes and Farantos^{12,13} presented an interesting, thorough

investigation of FD methods for solving the time-dependent Schrödinger equation in two and three dimensional systems.

A more general type of local method that has been used in a number of applications is the distributed approximating functional (DAF) approach of Kouri and co-workers.²¹⁻²⁴ The Lagrangian DAF is perhaps most closely related to the FD methods.²²⁻²⁴

Our approach to FD methods is different from those above, which are based upon interpolating functions such as Lagrange polynomials. We believe that FD approaches are most useful for applications that do not require extremely high accuracy. There are many such applications in chemical dynamics, e.g., we are generally very satisfied to compute reaction probabilities to three or four decimal places. Our approach embraces this idea of *limiting accuracy*, fitting the FD dispersion relation to the correct one for a range of momenta consistent with a specified accuracy. Our goal is to find a method requiring neither very small grid spacings nor very high order to obtain observables within a desired accuracy. We should note that the idea of fitting a dispersion relation to obtain FD coefficients has been used before in other areas such as acoustic wave propagation. For example, Tam and Webb,²⁵ and Zingg and co-workers²⁶ developed some first derivative approximations using the idea.

Section II presents theory (Secs. IIA-C), discusses computational and efficiency issues (Secs. IID, E), and shows how to treat energy resolved scattering (Sec. IIF). Numerical examples are given in Sec. III, including application to three-dimensional reactive scattering. Sec. IV concludes.

II. THEORY

A. Finite difference (FD) second derivative approximation

Consider an N-point grid in the interval (a,b)

$$x_i = x_0 + i \Delta, \quad i = 1, 2, \dots, N, \quad (1)$$

with

$$\Delta = \frac{b - a}{N + 1} = \frac{L}{N + 1} . \quad (2)$$

The end points a and b are not among the N grid points, with boundary conditions such that the represented function is zero at a and b . This grid is consistent with an underlying sine basis or spectral representation,¹⁵ although other grids consistent with, for example, periodic boundary conditions and complex exponential basis functions could be considered. We are interested in FD approximations to the second derivative of some function $y(x)$ defined on the grid as $y_i = y(x_i)$. Let \mathbf{y} denote the $N \times 1$ column vector of components y_i and \mathbf{y}'' denote the column vector of approximate second derivative components, $y''_i \approx (\partial^2 y / \partial x^2)_{x=x_i}$. A $2n+1$ point FD approximation is

$$\mathbf{y}'' = \mathbf{D}'' \mathbf{y} , \quad (3)$$

where \mathbf{D}'' is a symmetrically banded matrix with non-zero elements of the form

The dispersion relation, Eq. (7), is a relation between the wave number and the eigenvalue spectrum of \mathbf{D}'' . (For finite sized \mathbf{D}'' , Eq. (7) is not precisely the eigenvalue spectrum unless certain “edge” corrections are applied as discussed in Appendix A.)

The Fourier method,^{1,2,14} i.e., the evaluation of the second derivative utilizing Fourier transforms, leads to the correct dispersion relation for the underlying continuous problem,

$$f_{ex}(k) = -k^2. \quad (9)$$

The accuracy of the Fourier method is due to this fact. When implemented with Fast Fourier Transforms (FFTs) it is also efficient, with effort scaling quasilinearly in N as $O[N \log_2 N]$. In contrast, the FD dispersion relation, Eq. (7), deviates from $f_{ex}(k)$ for sufficiently large k , as illustrated in Fig. 1. An FD approximation also has favorable scaling that is explicitly linear in N , $O[(2n+1)N]$ (see Sec. IID).

B. Familiar FD methods as fits to the dispersion relation

FD approximations are typically viewed as $(2n+1)$ point interpolation formulae of $y(x)$.¹⁰⁻¹⁵ We can also view them as arising from fits of the FD dispersion relation, Eq. (7), to the correct relation, Eq. (9). The standard (Lagrangian) FD approximations, for example, result from Taylor series expanding Eq. (7) about $k = 0$,

$$f(k) = \frac{1}{2} d_0 + 2 \sum_{s=1}^n d_s k^{2s} + \frac{2}{4!} \sum_{s=1}^n d_s k^{4s} + \dots, \quad (10)$$

and, of the first $n+1$ non-zero (even) terms in the series, equating all to zero except the k^2 term, which is equated to -1 . For $n = 2$ this results in the three linear equations

$$\begin{aligned}
d_0 + 2d_1 + 2d_2 &= 0 \\
d_1 + 4d_2 &= 1 \\
\frac{d_1}{12} + \frac{4d_2}{3} &= 0
\end{aligned}$$

which yield $d_0 = -30/12$, $d_1 = 16/12$, and $d_2 = -1/12$, the standard 5-point FD result. In terms of the actual second derivative approximation, the error of such a classical $2n+1$ point FD approximation is of order Δ^{2n} . While analytical formulae exist for the Lagrangian FD coefficients of any order,^{10,15} for convenience and later comparison with our results, Table I lists coefficients corresponding to several other Lagrangian FD approximations.

The right hand side of Eq. (7) is also a cosine series. As noted by Guantes and Farantos,¹² the standard cosine series

$$\Delta^2 = \frac{\Delta^2}{3} + 4 \sum_{s=1}^{\infty} \frac{(\Delta^2)^s}{s^2} \cos(s\Delta), \quad (11)$$

with $\Delta = k\Delta$, when equated with $f(k)\Delta^2$, leads to the d_s coefficients of the generic DVR given by Colbert and Miller,¹⁵ i.e., $d_0 = -\Delta^2/3$, $d_1 = 2$, etc. It is also known as the sinc-DVR and has been discussed by Boyd.²⁷⁻²⁹ In molecular quantum mechanics, however, the sinc-DVR is generally used as a *full* rather than a banded matrix and gives results with accuracy comparable to the Fourier method.

C. Dispersion fitted finite difference method

The standard FD approximation¹⁰⁻¹³ and a banded sinc-DVR²⁷⁻²⁹ are two extremes regarding the dispersion relation. Standard FD methods describe correctly low magnitude k components, but yield a poor description of high k components of a function unless high orders are used. The banded sinc-DVR, on the other hand, provides an even handed description of all k that, for comparable order gives a much better description of the high k

components . However, the low k components are treated less accurately than with an FD method of comparable order. Here we outline an intermediate approach.

For convenience, we introduce a scaled wave number and dispersion relation,

$$K = k\Delta , \quad F(K) = \Delta^2 f(k) \quad (12)$$

with continuous K range $[0, \Delta]$ and, from Eq. (8), discrete values for finite N of

$$K_j = j\Delta / (N+1), \quad j = 1, \dots, N . \quad (13)$$

The problem of fitting the FD dispersion relation, Eq. (7), to the correct result, Eq. (9), is viewed as the least squares fitting problem of finding d_0, d_1, \dots, d_n that minimize

$$S(K_M) = \sum_{j=1}^M \left(F(K_j) + K_j^2 \right)^2 , \quad (14)$$

where $M \leq N$ and, from Eq. (7),

$$F(K) = d_0 + 2 \sum_{s=1}^n d_s \cos(sK) . \quad (15)$$

Because $F(K)$ is linear in the d_s , the problem is solved with standard general linear least squares methods.³⁰ We call an FD approximation that results from such a fit a dispersion fitted finite difference (DFFD) approximation. If $M = N$, then one fits the entire allowed range of K values, and the resulting coefficients will be similar to (but not exactly the same as) the sinc DVR ones. As one considers smaller M , $S(K_M)$ decreases and the fit, although to a more restricted K range, becomes better for that range. In the limit of small M , i.e.

restricting the fit to small K values, the result of such a fit is similar to the more standard FD approximation.

Because we fit to a discretized form of K , Eq. (13), the specific numerical results depend on N . This is not an issue if the N value employed to generate the fits is larger than any imagined application grid size. The results we subsequently present, based on $N = 1023$, may be used in applications with much smaller grid sizes as we illustrate in Sec. III.

The maximum absolute error, $\epsilon(K_M)$, between $F(K)$ and $-K^2$ in the K interval defined by $[K_1, K_M]$, also decreases with K_M . Fig. 2 illustrates this behavior with a $2n+1 = 9$ DFFD approximation as a solid curve. For comparison, we also show as a dashed curve the maximum absolute error in the same interval that results from the comparable Lagrangian FD method. Clearly, the DFFD errors are one or more magnitudes better than the FD ones.

Each point on the solid curve of Fig. 2 represents a different set of DFFD coefficients, determined by the fitting procedure above. We define a variety of DFFD approximations by the value of $\epsilon(K_M)$, denoting these approximations by $\text{DFFD}_{2n+1}(\epsilon)$. Tables II-V list $\text{DFFD}_{2n+1}(\epsilon)$ coefficients for $\epsilon = 10^{-3}, 10^{-4}, 10^{-5}$ and 10^{-6} , and $2n+1$ ranging from 9 to 23. While this variety of possible methods may seem overwhelming, the next section presents straightforward procedures for determining a suitable method.

D. Using the DFFD method: Computational and efficiency issues

A $\text{DFFD}_{2n+1}(\epsilon)$ method approximates the correct dispersion relation, $-K^2$, for scaled wave numbers, K , up to $K = K_M$ with an accuracy of ϵ or better. We illustrate this in Fig. 3. Provided that the finite DFFD matrix is edge corrected (Appendix A), its eigenvalues are given by the Eq. (7). Fig. 3a shows the absolute error in these eigenvalues relative to $-K^2$ as a function of K/ϵ for $\text{DFFD}_{11}(\epsilon)$ matrices with $\epsilon = 10^{-3}, 10^{-5}$ and 10^{-7} , as well as for the Lagrangian FD_{11} matrix. The FD_{11} errors are very small for low K , but

rise very rapidly and smoothly. The DFFD errors are different, oscillating around an accuracy one order of magnitude better than Δ , and then increasing near $K \approx K_M$. For $K < K_M$, the DFFD accuracy is much more uniform than the FD accuracy. It is also clear that the larger Δ the larger K_M . Fig. 3b shows the error in the eigenvalues for DFFD $_{2n+1}$ ($\Delta = 10^{-5}$) matrices with $2n+1 = 7, 15$ and 23 , as well as for the Lagrangian FD $_{23}$ case. For $K < K_M$, the DFFD errors oscillate around 10^{-6} , and we see that the larger $2n+1$, the larger K_M . Fig. 4 examines in more detail the variation of K_M with $2n+1$. Fig. 4a shows the K_M values associated with the Lagrangian FD method and Fig. 4b shows the DFFD K_M values. The DFFD K_M values in Fig. 4b are generally 50% larger than the corresponding FD values, which we show below leads to greater efficiency.

For a given problem, one estimates the maximum physical wave number, k_{\max} , related to the maximum possible kinetic energies that must be described on the grid, i.e., $\hbar^2 k_{\max}^2 / (2m)$, where m is an appropriate mass. In order for each $k \leq k_{\max}$ to correspond to a $K \leq K_M$ one must choose grid spacings

$$\Delta \leq K_M / k_{\max} . \quad (16)$$

The numerical effort associated with applying a FD or DFFD matrix is proportionate to $(2n+1)N$. Given Eq. (16) and $N \approx (b-a)/\Delta$, this suggests the effort is proportionate to

$$W = \frac{2n+1}{K_M} , \quad (17)$$

and ideally one should choose a method that leads to a relatively small value of W . Fig. 4c illustrates how W varies with $2n+1$ for various ϵ values. An algorithm for choosing a $\text{DFFD}_{2n+1}(\epsilon)$ method is:

1. Decide on a desired accuracy, ϵ
2. From Fig. 4c, determine the optimal $(2n+1)$ to use for the chosen ϵ .
3. From Fig. 4b, determine K_M . Then $\Delta = K_M/k_{\max}$ is the optimal grid spacing and $N = (b - a)/\Delta - 1$ is the number of grid points. (The kinetic energy errors will be $O[\epsilon \hbar^2 / (2\epsilon^2)]$ or less, owing to the actual dispersion errors in Fig. 3 typically being an order of magnitude smaller than ϵ)

A larger ϵ than the most optimal one above, and thus a smaller N , can also be used, but a higher value of $2n+1$ should be employed. The value of $2n+1$ must be such that, for the desired ϵ $K_M = k_{\max} \epsilon$. Fig. 4b can then be used to determine $2n+1$ for given ϵ and K_M . Because the effort in Fig. 4c is a slowly varying function of $2n+1$ near each minimum, there is actually a range of orders that can give nearly optimal performance and so a larger ϵ or higher $2n+1$ may not significantly impact the efficiency. For example, Fig. 4c shows that for $\epsilon = 10^{-4}$, the theoretical efforts for $2n+1$ ranging from 5 to 13 are similar. However, Fig. 4b shows that K_M and thus Δ can be twice as large for $2n+1 = 13$ compared to $2n+1 = 5$ or 7.

We now turn to how the numerical effort compares with other methods such as the Fourier method.^{1,2} It is natural, as we do in Sec. III, to compare results of DFFD or FD calculations with Fourier method results using the very same grids for all methods. However, such comparisons can be misleading because a more efficient, fewer point Fourier method calculation *may* suffice. Similar cautions apply when comparing to other

pseudospectral methods. In the Fourier method,^{1,2} the correct dispersion relation holds for all values of K between 0 and π and so one may use

$$\Delta_{ps} = \Delta/k_{\max} \quad \text{and} \quad N_{ps} \approx (K_M/\Delta) N \quad (18)$$

in a Fourier method or comparable pseudospectral calculation. Comparing with Eq. (16), we see that $\Delta_{ps} = (\Delta/K_M) \Delta > \Delta$ and $N_{ps} < N$, since $K_M < \pi$ for a DFFD or FD method.

We assume the Fourier method is implemented with sine FFTs and use "FFT" to denote it. Let N be the number of DFFD or FD grid points and N_{ps} be the number of Fourier or comparable pseudospectral grid points. The evaluation of the action of the kinetic energy matrix on a vector scales as:

$$\begin{aligned} S_{DFFD} &= \Delta N(2n+1) \\ S_{FFT} &= \frac{3}{2} N_{ps} \log_2 \frac{N_{ps}}{2} \\ S_{DVR} &= N_{ps}^2 \end{aligned} \quad (19)$$

where the factors in S_{DFFD} and S_{FFT} have been chosen to be consistent with efficient implementations of the relevant algorithms. The factor Δ in the DFFD scaling represents an improvement over simple $(2n+1)N$ scaling made possible by a fast banded-matrix algorithm (Appendix B) that utilizes the pipelining³¹ capabilities of most modern computers. It is somewhat machine dependent and turns out to be $\Delta \approx 0.65$ on an IBM RS/6000 Power 3 workstation and $\Delta \approx 0.5$ on a Compaq XP1000 ("Dec-Alpha") workstation. We use $\Delta = 0.65$ in our estimates, with the understanding that it may be greater or smaller on other

platforms. See Appendix C for a discussion of the FFT scaling. S_{DVR} above corresponds to a DVR with a full second derivative matrix.

As an example, consider $\text{DFFD}_{11}(\Delta=10^{-4})$, which has $K_M/\Delta = 0.54$ and the case $N = 237$. With a different but optimal pseudospectral grid spacing, Eq. (18) implies $N_{\text{ps}} = 128$. We find from, Eq. (19), $S_{\text{DFFD}}/S_{\text{FFT}} = 1.46$, indicating that an optimized FFT calculation could be nearly 50% more efficient than a corresponding DFFD one. It is possible to improve the DFFD efficiency at the cost of accuracy, by increasing Δ or reducing $2n+1$. Similarly, a DFFD calculation becomes even less efficient relative to an optimized FFT one if smaller Δ or higher $2n+1$ values are considered. If the same grid spacing were used for both DFFD and FFT, i.e., if $N = N_{\text{ps}} = 128$, the $S_{\text{DFFD}}/S_{\text{FFT}}$ ratio for $2n+1 = 11$ would be ≈ 0.8 , indicating a more efficient DFFD calculation. The corresponding $S_{\text{DFFD}}/S_{\text{DVR}}$ ratio for our example is 0.10. We conclude that in terms of efficiency, the DFFD method is superior to a full matrix DVR method and can approach or surpass the Fourier method. This is particularly true because, in practice, pseudospectral methods are often implemented using smaller grid spacings than the Δ_{ps} determined from Eq. (18).

Actual timing comparisons between methods depend on their implementation, and the estimates above are based on the assumption of efficient implementations. With the ESSL³² subroutine library on the IBM RS/6000 to evaluate FFTs, and our fast banded matrix algorithm (Appendix B) for the DFFD computation, we achieve timings that are roughly consistent with the theoretical estimates above. (Of course completely different results might occur if any method is implemented in a less efficient manner.)

E. DFFD and parallel computations

A primary motivation for exploring FD methods is related to their usefulness in parallel computing.¹⁶ In order to study large systems with large memory requirements, it is

advantageous to harness the power of many processors either in distributed memory or shared memory parallel systems. In parallel algorithms for iterative bound state or wave packet calculations, a vector \mathbf{y} corresponding to the grid representation of the function is distributed over various processors according to one or more variable,^{17,18} and the calculation is advanced via repeated actions of the Hamiltonian matrix, \mathbf{H} , on \mathbf{y} . Most of the calculation occurs concurrently on the various processors. The calculation of the action of the kinetic energy matrix contribution to $\mathbf{H}\mathbf{y}$, $\mathbf{T}\mathbf{y}$, for the distributed variable(s), requires communication. This communication is generally the major bottleneck that prevents efficient scaling of parallel algorithms.

Because pseudospectral methods are nonlocal in *each specific variable*, the amount of communication required can become considerable. Let N_{proc} be the number of processors. A simple parallel model, based on a global approximation involving each processor to communicate with every other processor, requires that $(N_{\text{proc}}-1)/N_{\text{proc}}$ of the total wave packet be communicated for each calculation of $\mathbf{T}\mathbf{y}$.¹⁷ Because of its local nature, the DFFD promises to drastically reduce the amount of communication. With judicious choice of parameters, each processor needs to communicate only with its nearest neighbors, and only a *fraction* of the wave packet on any given processor needs to be distributed to complete each calculation of $\mathbf{T}\mathbf{y}$. The exact amount of communication will depend on the specific parameters of the problem.

To illustrate the communication requirements for an FD/DFFD method, consider a two coordinate degree of freedom example, with wavefunction or wave packet components $y(\mathbf{R}_i; \mathbf{r}_j)$, $i = 1, \dots, N_R$ and $j = 1, 2, \dots, N_r$. Suppose the wave packet is distributed over two processors according to the variable \mathbf{R} and let $N_R = 256$. Therefore, Processor 0 contains the chunk of the wave packet on $\mathbf{R}_1 \dots \mathbf{R}_{128}$ and all \mathbf{r} , and Processor 1 contains the chunk corresponding to $\mathbf{R}_{129} \dots \mathbf{R}_{256}$ and all \mathbf{r} . Suppose also that the action of the kinetic energy matrix is accomplished via a 7 point FD or DFFD (i.e., $n = 3$). The kinetic energy matrix factors into two terms, \mathbf{T}_R and \mathbf{T}_r . In this example, the evaluation of the action of \mathbf{T}_r

requires no communication, and the computation of $\mathbf{T}_R \mathbf{y}$ requires communication of information only at the edges. For example, the component of $\mathbf{T}_R \mathbf{y}$ at R_{128} , evaluated on processor 0, requires Processor 1 to pass to it $y(R_{129,r})$, $y(R_{130,r})$ and $y(R_{131,r})$ (for all r). Therefore, only minimal information needs to be exchanged between the processors.

In FD or DFFD methods the fraction of the wave packet that must be communicated to each processor to complete an evaluation of $\mathbf{T}_R \mathbf{y}$ depends only on the number of grid points in the distributed variable, e.g. N_R , and the FD/DFFD order, n , and is independent of the number of processors, N_{proc} . Each processor must receive $2n/N$ of the wave packet except Processor 0 and Processor $(N_{\text{proc}} - 1)$ that contain the grid edges; these must receive n/N . Suppose that $N_R = 256$. If $2n+1 = 9$, then $2n/N = 1/32$. When $(2n+1) = 17$, $2n/N = 1/16$. Contrast this with the communication requirements of the simple algorithm given in Ref. [17], where for $N_{\text{proc}} = 8$, $7/8$ of the wavepacket must be communicated to each processor. A recent DVR-based parallel algorithm proposed by Meijer¹⁸ reduces the fraction to $2(N_{\text{proc}} - 1)/N_{\text{proc}}^2$, an enormous improvement for over the method of Ref. [17]. The DFFD methods will usually require less communication than this algorithm, which, for the above example, would require 0.22 of the wave packet be communicated. More efficient implementations of the DFFD method will result from distributing the data according to two or more variables thus taking advantage of larger numbers of processors.

F. Wave packet scattering and FD approximations

In wave packet calculations that aim to obtain energy resolved probabilities, a time sequence is generated and Fourier analyzed.⁹ This analysis includes multiplication by a factor consistent with transforming between wave number (k) and collision energy (E) normalized time-independent scattering states. The wave number to collision energy dispersion relation, for one degree of freedom with associated mass μ , is

$$E(k) = \frac{\hbar^2}{2m} f(k), \quad (20)$$

where $f(k)$, in the case of an FD approximation is given by Eq. (7). Superior energy resolved information can be inferred if Eq. (20) is consistently employed in the analysis. This is a subtle point since one is tempted to employ instead the exact dispersion relation, $E_{\text{ex}}(k) = \hbar^2 k^2 / (2m)$. In the limit of sufficiently small grid spacing, Δx , or high enough FD order, there is no appreciable error in this latter approach. In practice, there are differences between $E_{\text{ex}}(k)$ and $E(k)$, particularly in the higher magnitude k limit, and we find that the actual error in a reaction probability can be reduced considerably by treating the analysis consistently.

More specifically, we outline how the analysis procedure of Ref. [33] is easily modified to treat the FD dispersion relation. Consider a position space normalized initial wave packet $y(x, t=0) = \psi(x)$, where x is the incoming scattering coordinate and $\psi(x)$ is localized in the asymptotic, incoming channel. Time-independent scattering wavefunctions, y_E^+ , normalized such that $\langle y_E^+ | y_{E'}^+ \rangle = 2\pi\hbar \delta(E-E')$, are proportionate to a Fourier transform of the time evolving wave packet under the Hamiltonian H , $y(t) = \exp(-iHt/\hbar) y(t=0)$, according to^{33,34}

$$y_E^+ = \frac{1}{a_\psi(E)} \int dt \exp(iEt/\hbar) y(t), \quad (21)$$

with

$$a_\psi(E) = \left[\frac{\hbar}{d\psi/dk} \right]^{1/2} \int dk \exp(ikR) \psi(R). \quad (22)$$

The energy resolved reaction probability is $P(E) = \langle y_E^+ | F | y_E^+ \rangle$, where F is an appropriate flux operator,^{33,34} and it thus involves a factor of $1/a_{\square}^2(E)$ (Eq. (13) of Ref. [33]). A consistent treatment involves using Eqs. (7) and (20) to evaluate dE/dk in $a_{\square}(E)$ instead of assuming that dE/dk can be replaced by $\hbar^2 k / \square$. (Similar considerations apply to state-to-state reaction probabilities.)

III. ILLUSTRATIONS

A. Harmonic oscillator bound states

As a simple but instructive example, we determine the eigenvalues of a discrete approximation to the unit mass and frequency harmonic oscillator Hamiltonian, in atomic units ($\hbar = 1$),

$$H_{ij} = (1/2) (- D''_{ij} + \square_{ij} x_i^2) , \quad (23)$$

where the grid Eq. (1) and various FD approximations for \mathbf{D}'' are used. Note that all the FD calculations here and in Sec. IIIB are performed with the edge corrected FD matrices of Appendix A. We set $a = -10 a_0$, $b = 10 a_0$ in all our calculations, and focus on the root mean square (RMS) error of the first twenty eigenvalues of Eq. (23) relative to the exact ones. Two sets of calculations are discussed, with $N = 127$ and $N = 63$. The latter set has a grid spacing that is twice as large as the former.

Figs. 5a ($N = 127$) and 5b ($N = 63$) display the RMS error obtained from a variety of calculations with the Lagrangian FD_{2n+1} (dashed curve, open circles) and various

DFFD $_{2n+1}(\Delta)$ (solid curves, see figure for symbols) methods. See Tables I-V for the FD and DFFD coefficients used in the calculations. From Fig 5a, we see that the RMS error of the Lagrangian FD approximation scales exponentially with $2n+1$, as indicated by the linear dashed curve in the linear-log plot of Fig. 5a. The RMS error associated with the various DFFD approximations, for fixed Δ displays near exponential scaling for low values of $2n+1$ but then plateaus out at a value near the Δ value associated with a particular method as one would expect.

It is more significant that the RMS error of any DFFD method, before it plateaus out, is always less than the Lagrangian FD error. For example, the DFFD $_{11}(\Delta = 10^{-7})$ method yields an RMS error that is almost three orders of magnitude smaller than the FD result at $2n+1 = 11$. In fact, Fig. 5a shows that a Lagrangian FD with $2n+1 = 19$ is required to obtain a similar RMS error to the DFFD $_{11}(\Delta = 10^{-7})$ case.

Prior to the plateau region in Fig. 5a, a larger Δ DFFD method may be superior to a smaller Δ DFFD method. For example, at $2n+1 = 9$ we see that the $\Delta = 10^{-6}$ DFFD yields the best RMS error, whereas the $\Delta = 10^{-8}$ DFFD yields the largest error. We can understand this by referring back to Fig. 4b. Here we see for *fixed* $2n+1$, K_M decreases as Δ is decreased, which means that DFFD $_{2n+1}$ fits accurately a smaller and smaller range of wave number values. We also see that with Δ fixed a value consistent with obtaining observables to 5 or 6 significant figures, a low order DFFD may suffice and yield better results than the corresponding Lagrangian FD or, indeed, a DFFD with a much smaller Δ .

The other point to notice in Fig. 5a is that the plateau region is reached at nearly the same values of $2n+1$ that one would expect from the curves presented in 4c. For example, the $\Delta = 10^{-6}$ curve plateaus at $2n+1 = 11$, which is the optimal value as indicated by Fig. 4c. Therefore the grid spacing used in this calculation is optimal. If the grid spacing is doubled, the results are very different, as shown in Fig. 5b. Since the largest good wave number, k_{\max} ,

is given by K_M/Δ , increasing Δ reduces k_{\max} for a given K_M . In Fig. 5b, only $\Delta = 10^{-5}$ has reached its plateau, and not until $2n+1 = 19$. Lower Δ values require much higher orders to attain their accuracy limits. While increasing order rather than decreasing grid spacing is one way to improve an FD calculation^{10,11} beyond a certain point, it leads to reduced overall efficiency. Also note, in Fig 5b that the DFFD curves lies below the FD curve for all $2n+1$.

B. Total reaction probabilities for three-dimensional reactive scattering

We now consider the $D + H_2(v = j = 0) \rightarrow DH + H$ reaction in three dimensions (total angular momentum $J = 0$), with the Liu-Siegbahn-Truhlar-Horowitz (LSTH) potential energy surface.³⁵⁻³⁷ The real wave packet approach³⁸ was used to carry out wave packet propagation in reactant Jacobi coordinates, $R = D - \text{center of } H_2 \text{ distance}$; $r = H_2$ internuclear distance; $\cos\theta = \text{cosine of the angle between vectors associated with } R \text{ and } r$. Grids, as in Eq. (1), were used to represent both R and r ($R_{\min}=0 \text{ a}_0$, $R_{\max}=12.5 \text{ a}_0$; $r_{\min}=0.5 \text{ a}_0$, $r_{\max}=12.5\text{a}_0$). The angular part of the wave packet was represented in a rotational basis set of the first 25 even Legendre functions, but evaluation of the action of the potential involved transformation to an equivalent grid representation.^{38,39} The propagation involves repeated actions of a matrix representation of the Hamiltonian on a vector, with the kinetic energy terms in R and r each being evaluated with the various FD approximations. A flux method³³ was used to evaluate the reaction probability. The initial (Gaussian) wave packet and other details are as in Ref. [38]. The flux calculation involves a first derivative with respect to r at one particular r value. In all cases this was evaluated accurately with FFTs, although we could have used a high order FD approach.

In order to establish the most accurate result with which the various approximations could be compared, we carried out a calculation using the Fourier method in its sine FFT form employing $N_R = N_r = 127$ points in the action of the kinetic energy operator, i.e., $\Delta R = 0.09766 \text{ a}_0$ and $\Delta r = 0.09375 \text{ a}_0$. All the scattering calculations were performed on a

Compaq XP1000 computer and the Digital DXML⁴⁰ library was used for the FFTs. The sine FFTs we employed are such that N_R+1 and N_r+1 should be even, and preferably a product of low prime numbers. Fig. 6 displays the total reaction probability over the 0 - 1.3 eV collision energy our chosen initial wave packet was capable of describing.

One measure of error is the maximum absolute magnitude deviation of the reaction probability from the most accurate result over the energy range. This error is often the error at some energy near the upper collision energy limit in Fig. 6. Fig. 7a displays the error obtained from several calculations employing $N_R = N_r = 95$ points (i.e., $\Delta R = 0.130 a_0$ and $\Delta r = 0.125 a_0$). Each symbol in the figure is the result of a full three-dimensional wave packet calculation and analysis over the 0 - 1.3 eV collision energy range. The DFFD calculations, for each Δ are performed at a low, an optimal (intermediate) and a high value of $2n+1$. The optimal value was chosen by computing $K_M = k_{\max} \Delta$ and then using Fig 4b. With one exception, the DFFD results are superior to the FD results at all $2n+1$. It is also clear that above the optimal value of $2n+1$, there is nothing to be gained by going to higher order. The dashed curve in this plot is the error of applying the Fourier method with $N_R = N_r = 95$. The $\Delta = 10^{-6}$ curve reaches the Fourier limit at its optimal value.

The flux analysis method used to infer the results in Fig. 7a used the formulae from Ref. 31, and did *not* take explicit account of the FD dispersion relation. While the errors in Fig. 7a are reasonable and behave qualitatively as expected, the error magnitudes can be reduced if the simple correction to the analysis procedure, given in Sec. IIF, is used as is shown in Fig. 7b. This simple correction provides major improvements to the less accurate calculations, particularly the Lagrangian FD calculation. It also dramatically improves the $\Delta = 10^{-4}$ results. Once the DFFD results approach the Fourier limit, the FD dispersion relation correction (Sec. IIF) has little effect.

Fig. 7c presents analogous results to Fig. 7b but employing $N_R = N_r = 79$ points, i.e., larger grid spacings of $\Delta R = 0.156 a_0$ and $\Delta r = 0.150 a_0$, where in fact the

corresponding Fourier method result (dashed line) has error $> 10^{-3}$. What is striking in this case is that it is fairly easy here to reach the Fourier limit with low accuracy DFFD and low order calculations. A high order Lagrangian FD method also gives “good” results. This should serve as one cautionary note when comparing FD and spectral methods. If the grid spacing is so large that the FFT results themselves are not well converged, then low accuracy DFFD results may reach the Fourier limit.

Finally, regarding CPU timings, based on extensive calculations with $N_R = N_I = 63, 79, 95$ and 127 we find that the DFFD/FD CPU times for the kinetic energy portion of the calculation vary linearly with $2n+1$ as would be expected. Generally the $2n+1 = 9$ results are always about twice as fast as the corresponding (same number of points) Fourier calculations based on an efficient FFT library,⁴⁰ and the $2n+1 = 19$ results are comparable in time to the Fourier calculations. (These results are somewhat better than suggested by the simple scaling estimates of Eq. (19), which imply $2n+1 \approx 13$ or 15 , not 19 , for the cross over point.)

IV. Concluding Remarks

We presented a new FD method, the DFFD method, for calculating a grid of second derivative approximations corresponding to a grid of function values. The general approach, based on fitting the dispersion relation of the relevant finite difference derivative matrix to the correct result up to a limiting accuracy, is applicable to other derivative orders. (See, for example, earlier work on first derivative approximations in Refs. [25] and [26].) Therefore more general forms of the kinetic energy operator could also be treated with the approach. We also presented a highly efficient algorithm for implementing the method that effectively utilizes the pipelining capabilities of modern computers. For a given problem, and a desired degree of accuracy, it is straightforward to choose the optimal grid spacing, and order.

We also carried out comparisons of our approach with the standard Lagrangian FD and Fourier methods in relation both to a simple eigenvalue problem and a more challenging three-dimensional reactive scattering problem. Depending on the desired accuracy, the DFFD approach, for comparable effort, can be orders of magnitude better than FD and can also approach the Fourier method. We therefore believe that the DFFD approach will be highly competitive with commonly used pseudospectral methods for applications that do not require extremely high accuracy, which comprise the majority of dynamics calculations. In addition to accuracy, simplicity and portability, the locality of the approximation makes it well suited to parallel computation.

ACKNOWLEDGEMENTS

EMG thanks Dr. R. T Pack for pointing her to Ref. 11. We also thank Dr. R. L. Shepard for useful discussions. EMG acknowledges support for this research from NSF grant No. CHE-9970994. SKG acknowledges support from the Office of Basic Energy Sciences, Division of Chemical Sciences, U. S. Department of Energy, under Contract N. W-31-109-ENG-38.

APPENDIX A: THE FD DISPERSION RELATION AND EDGE CORRECTIONS

The basic FD or DFFD second derivative matrix, \mathbf{D}'' , is given by Eq. (4) or (5). With the exception of the first n and last n rows, each row involves a centered $2n+1$ point approximation. If one neglects the upper and lower edge effects, i.e. imagines that there is always a $2n+1$ point centered difference approximation for the second derivative at any grid point, then a simple expression for the eigenvalues of \mathbf{D}'' arises. For example, one can imagine that the grid is sufficiently (infinitely) extended in both the left and right directions.

It is then easy to show by direct substitution that the sine (or particle in a box) basis vectors, $\mathbf{v}^{(j)}$, with grid (x_i) components

$$\mathbf{v}^{(j)} \Big|_i = v^{(j)}(x_i) = \sqrt{\frac{2}{L}} \sin \frac{j x_i}{L} = \sqrt{\frac{2}{(N+1)}} \sin \frac{j i}{N+1}, \quad (\text{A1})$$

for $j = 1, 2, \dots, N$, are eigenstates of each \mathbf{D}_s matrix according to

$$\mathbf{D}_s \mathbf{v}^{(j)} = 2 d_s \cos \frac{s j}{N+1} \mathbf{v}^{(j)}. \quad (\text{A2})$$

(The sine vectors $\mathbf{v}^{(j)}$ are in fact eigenvectors of \mathbf{D}_0 and \mathbf{D}_1 for any grid extent.) Therefore, neglecting edge effects, the sine vectors are also eigenvectors of the full \mathbf{D}'' matrix, Eq. (5), and the eigenvalues of Eq. (5) become simply Eq. (7), the desired dispersion relation, when Eq. (8) is used. (See also Ref. [12] for a related derivation of Eq. (7) with complex exponential basis functions.)

Certain *modified* \mathbf{D}'' matrices, of any *finite* grid extent or dimension N , can also lead to the analytical dispersion relation, Eq. (7). The modified matrices result when one imposes certain boundary conditions to effectively extend the grid and involve altering the upper $n-1$ and lower $n-1$ rows of the \mathbf{D}'' matrix. The idea is to imagine that the grid of available function values, $y_1, y_2, y_3, \dots, y_N$, is extended "off" the grid in both directions:

$$\dots, y_{-2}, y_{-1}, y_0, y_1, y_2, y_3, \dots, y_N, y_{N+1}, y_{N+2}, y_{N+3}, \dots$$

Sine wave boundary conditions, as in the Fourier method applied with a sine Fourier transform,¹⁵ are such $y_0 = 0$, $y_{N+1} = 0$, $y_{-k} = -y_k$, and $y_{N+1+k} = -y_{N+1-k}$ for $k > 0$. This allows one to always employ $2n+1$ points (some obtained by reflection of the available

points) in evaluating the action of the FD second derivative matrix. Of course this is just one particular way of extending the grid of function values, but in practice we have found that it can be a significant improvement over assuming that the function values off the grid are all zero, which is implicitly what is assumed if no modification is made. Consider, for example, evaluation of the first component second derivative of an $n = 3$ DFFD or FD matrix. Without grid extension, one would have, from Eqs. (3) and (4), $y''_1 = [d_0 y_1 + d_1 y_2 + d_2 y_3 + d_3 y_4] / \Delta^2$. The above grid extension idea implies

$$\begin{aligned} y''_1 &= [d_3 y_{-2} + d_2 y_{-1} + d_1 y_0 + d_0 y_1 + d_1 y_2 + d_2 y_3] / \Delta^2 \\ &= [(d_0 - d_2) y_1 + (d_1 - d_3) y_2 + d_2 y_3 + d_3 y_4] / \Delta^2 \end{aligned}$$

Thus, $D''_{11} = (d_0 - d_2) / \Delta^2$ and $D''_{12} = (d_1 - d_3) / \Delta^2$ in this case. Some reflection shows that for an $N \times N$ FD or DFFD matrix, \mathbf{D}'' , the first and last $n-1$ rows are modified according to:

For $i = 1, 2, \dots, n-1$:

For $j = 1, 2, \dots, n-i$:

$$D''_{i,j} = (d_{|i-j|} - d_{i+j}) / \Delta^2 \tag{A3}$$

$$D''_{N-i+1, N-j+1} = D''_{i,j}$$

where the last statement corrects the lower right edge of the FD matrix. For $i = n, n+1, \dots, N-n$, the usual FD matrix, $D''_{i,j} = d_{|i-j|} / \Delta^2$ for $|i-j| \leq n$ and 0 otherwise applies.

By direct substitution, it is possible to show that for any finite N , the edge corrected \mathbf{D}'' matrix defined by Eq. (A3) has eigenvectors given by the sine basis vectors, $\mathbf{v}^{(j)}$, Eq.

(A2), and eigenvalues given by Eq. (7) with the wave number, k , being discretized according to Eq. (8).

It is possible to obtain a similar result if, instead of sine boundary conditions, periodic boundary conditions are employed. In this case the grid would have to be defined slightly differently to include one of the end points, e.g. one could employ $\Delta = (b-a)/N$ in Eq. (2), which implies $y_N = b$. One then assumes "wrap around" boundary conditions, $y_0 = y_N, y_{-1} = y_{N-1}, \dots$, and $y_{N+1} = y_1, y_{N+2} = y_2, \dots$. Considerations similar to those above lead to a different modified \mathbf{D}'' matrix that does not actually have any of the old non-zero matrix elements modified. Instead, some previously zero elements in the upper right and lower left portions of \mathbf{D}'' become non-zero. The eigenvalues of this matrix are also given by Eq. (7). The allowed values of the wave number k can now be both positive and negative according to $k_j = -\pi/\Delta + j 2\pi/(N\Delta)$. (Kouri and co-workers have applied similar wrap around boundary conditions within the context of DAF approximations, allowing the construction of "fast" DAFs,⁴¹ which are useful for large bandwidths.)

APPENDIX B: METHODS FOR COMPUTING THE ACTION OF A FD MATRIX ON A VECTOR

We investigated several approaches for evaluating the action of the FD or DFFD second derivative matrix \mathbf{D}'' , proportionate to the kinetic energy matrix, on a vector \mathbf{y} . A direct evaluation of the form

$$y''_i = \sum_{j=\max(1,i-n)}^{\min(N,i+n)} D''_{ij} y_j \quad (\text{B1})$$

makes use of the banded structure and involves about $N(2n+1)$ multiplications. Approximately the same number of additions are also involved. When programmed directly

as written, these additions are essentially "free" in terms of computational time when run on current computers owing to compilers arranging certain additions and multiplications to be carried out in the same clock cycle ("pipelining").³¹ It is also possible to significantly improve on this computational effort. For example, one could employ Basic Linear Algebra Subroutines (BLAS).⁴² BLAS routines are available on almost every computer system, are highly portable and are tailored by computer manufacturers to run extremely efficiently on their computer platforms. The level 2 BLAS routine DSBMV evaluates the matrix-vector product for a symmetric band matrix and can lead to substantial efficiencies over programming Eq. (B1) in the most obvious manner.

However, we have found an even more efficient algorithm based on the decomposition of \mathbf{D}'' in terms of the $n+1$ \mathbf{D}_s matrices, Eq. (5). If $\mathbf{y}'' = \mathbf{D}'' \mathbf{y}$ is the desired result and if edge corrections are ignored, then the algorithm is, with $\alpha_s = d_s/\alpha^2$:

$$\mathbf{y}'' = \alpha_0 \mathbf{y}$$

For $s = 1, n$:

For $i = 1, s$:

$$y''_i = y''_i + \alpha_s y_{s+i}$$

(B2)

For $i = s+1, N-s$:

$$y''_i = y''_i + \alpha_s (y_{i-s} + y_{i+s})$$

For $i = N-s+1, N$:

$$y''_i = y''_i + \alpha_s y_{i-s}$$

which involves first copying d_0 times the initial vector \mathbf{y} into \mathbf{y}'' and then, for $s = 1, 2, \dots, n$, updating the components of \mathbf{y}'' . The key to efficiency is the update of the central components of \mathbf{y}'' from $i = s + 1$ to $N-s$ for each s . What was two multiplications in Eq.

(B1) has been effectively turned into one multiplication (and an addition). For $N \gg n$, then the number of multiplications is approximately $(2n+1)N/2$, i.e. a factor of two less than the obvious algorithm, Eq. (B1). There are more additions in algorithm (B2). However, we find that the effort associated with these extra additions tends to be significantly masked, owing again to pipelining. This is the source of the \square term in Eq. (19). It is also straightforward to edge correct (Appendix A) the result of Eq. (B2). After applying Eq. (B2), one then modifies the first and last $n-1$ components of \mathbf{y}'' according to:

For $i = 1, n-1$:

For $j = 1, \dots, n-i$:

$$y''_i = y''_i - \square_{i+j} y_j \quad (\text{B3})$$

$$y''_{N+1-i} = y''_{N+1-i} - \square_{i+j} y_{N+1-j}$$

which accomplishes the same thing as the redefinition of the \mathbf{D}'' matrix in Eq. (A3).

APPENDIX C: OPERATION COUNT FOR THE FOURIER METHOD

The Fast Fourier Transform (FFT) involves numerical work that scales as $O(N \log_2 N)$, where N is the number of grid points.³⁰ The actual numerical effort, generally dominated by the number of real multiplications, is more subtle and depends on the specific FFT implementation. (As in Appendix B, we assume that the additions, generally occurring simultaneously with the multiplications, can be pipelined on modern computers.) One also sees a variety of sub-optimal estimates of this count in the literature. An FFT program adopted from a book or taken from a publicly available software library might involve a multiplication count that is two or more times larger than what is achievable with an efficient

and probably computer dependent program. In determining the relative efficiency of the various FD methods with respect to an FFT based Fourier method, we assumed in the text a reasonably optimal implementation of the FFT. This Appendix justifies the multiplication count assumed for such a Fourier method.

With N taken to be a power of two, the number of *complex* multiplications associated with a single complex to complex FFT is approximately⁴³ $(N/2) \log_2 N$. (Most modern FFT libraries allow for N , or sometimes $N+1$ in the case of sine FFTs, to be simply even and, for optimal efficiency, a product of low prime numbers.) Normally a complex multiplication involves four real multiplications. However, symmetries in the details of the FFT allow one to effectively use three and not four real multiplications per complex multiplication,⁴⁰ so that the number of real multiplications per complex FFT is closer to $(3N/2) \log_2 N$. There are more sophisticated formulae for the number of real multiplications⁴⁰ that involve still fewer operations. However, for the N of interest, generally less than 1024, we have found with two modern, optimized libraries (the IBM ESSL³² and the Digital DXML⁴⁰ libraries) that this latter result is quite reasonable. Thus, for the most common form of the Fourier method, which involves two complex FFTs per evaluation of the kinetic energy matrix on a complex vector, the numerical effort is approximately that of $3N \log_2 N$ real multiplications.

We are most interested in an implementation of the Fourier method with a real to real sine FFT. Such a sine FFT is effectively dominated³⁰ by an $N/2$ point complex FFT and so the number of real multiplications per single sine FFT is about $(3N/4) \log_2(N/2)$. Since two sine FFTs are required per evaluation of the kinetic energy matrix acting on a real vector, the number of real multiplications is then about $(3N/2) \log_2(N/2)$.

References

1. D. Kosloff and R. Kosloff, *J. Comput. Phys.* **52**, 35 (1983).
2. M. D. Feit and J. A. Fleck, *J. Chem. Phys.* **78**, 301 (1983).
3. J. C. Light, I. P. Hamilton, and J. C. Lill, *J. Chem. Phys.* **82**, 1400 (1985).
4. S. E. Choi and J. C. Light, *J. Chem. Phys.* **92**, 2129 (1990).
5. *Numerical Grid Methods and their Applications to Schrödinger's Equation*, Nato ASI Series C, edited by C. Cerjan (Kluwer Academic, Dordrecht, 1993).
6. C. Canuto, Y. Hussaini, A. Quarteroni, and T. Zhang, *Spectral Methods in Fluid Dynamics*, (Springer, New York, 1988).
7. D. Gottlieb and S. A. Orszag, *Numerical Analysis of Spectral Methods: Theory and Applications*, (SIAM, Philadelphia, 1977).
8. J. Cullum and R. A. Willoughby, *J. Comput. Phys.* **44**, 329 (1981).
9. N. Balakrishnan, C. Sathymurthy and N. Sathymurthy, *Phys. Reports* **280**, 79 (1997) and references therein.
10. M. Thachuk and G. C. Schatz, *J. Chem. Phys.* **97**, 7297 (1992).
11. D. A. Mazziotti, *Chem. Phys. Lett.* **299**, 473 (1999).
12. R. Guantes and S. C. Farantos, *J. Chem. Phys.* **111**, 10827 (1999).
13. R. Guantes and S. C. Farantos, *J. Chem. Phys.* **113**, 10429 (2000).
14. B. Fornberg, *A Practical Guide to Pseudospectral Methods*, Cambridge Monographs on Applied and Computational Mathematics, (Cambridge University Press, Cambridge, England, 1996)
15. D. T. Colbert and W. H. Miller, *J. Chem. Phys.* **96**, 1982 (1992).
16. See articles in *Comput. Phys. Commun.* **128** (2000), a thematic issue on parallel computing in chemical physics.
17. E. M. Goldfield, *Comput. Phys. Commun.* **128**, 178 (2000).
18. A. J. H. M. Meijer, *Comput. Phys. Commun.*, submitted (2001).
19. R. Kosloff, *J. Phys. Chem.* **92**, 2087 (1988).

20. R. Kosloff in *Numerical Grid Methods and their Applications to Schrödinger's Equation*, Nato ASI Series C, edited by C. Cerjan (Kluwer Academic, Dordrecht, 1993), pp 175-194.
21. D. K. Hoffman and D. J. Kouri, *J. Phys. Chem.* **96**, 1179 (1992).
22. G. W. Wei, D. S. Zhang, D. J. Kouri and D. K. Hoffman, *Phys. Rev. Lett.* **79**, 775 (1997).
23. D. K. Hoffman, G. W. Wei, D. S. Zhang and D. J. Kouri, *Phys. Rev. E* **57**, 6152, (1998).
24. G. W. Wei, S. C. Althorpe, D. S. Zhang, D. J. Kouri and D. K. Hoffman, *Phys. Rev. A* **57**, 3309 (1998).
25. C. K. W. Tam and J. C. Webb, *J. Comput. Phys.* **107**, 262 (1993).
26. D. W. Zingg, H. Lomax, and H. Jurgens, AIAA Paper 93-0495, (American Institute of Aeronautics and Astronautics, New York, 1993); D. W. Zingg, *SIAM J. Sci. Comput.* **22**, 475 (2000) and references therein.
27. J. P. Boyd, *Appl. Num. Math.* **7**, 287 (1991).
28. J. P. Boyd, *J. Comp. Phys.* **103**, 243 (1992).
29. J. P. Boyd, *J. Comp. Phys.* **103**, 243 (1992).
30. W. H. Press, S. A. Teukolsky, W. T. Vetterling and B. P. Flannery, *Numerical Recipes in Fortran: the art of scientific computing*, 2nd edition (Cambridge University Press, New York NY, 1992).
31. G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd edition (Johns Hopkins Univ. Press, Baltimore, 1996).
32. Complete documentation for the ESSL library can be found at <http://www.rs6000.ibm.com/tech/essl.html>; see also <http://scv.bu.edu/SCV/IBMSP/ESSL-howto.html>
33. A. J. H.M. Meijer, E. M. Goldfield, S. K. Gray and G. G. Balint-Kurti, *Chem. Phys. Lett.* **293**, 270 (1998).
34. D.H. Zhang and J. Z. H. Zhang, *J. Chem. Phys.* **101**, 3674 (1994).

35. P. Siegbahn, and B. Liu, J. Chem. Phys. **68**, 2457 (1978).
36. D. G. Truhlar and C. J. Horowitz, J. Chem. Phys. **68**, 2466 (1978).
37. D. G. Truhlar and C. J. Horowitz, J. Chem. Phys. **71**, 1514 (1979).
38. S. K. Gray and G. G. Balint-Kurti, J. Chem. Phys. **108**, 950 (1998).
39. C. Leforestier, J. Chem. Phys. **94**, 6388 (1991).
40. DXML Documentation can be found in the Compaq Extended Math Library Guide which can be downloaded from <http://www.compaq.com/fortran/docs>
41. Y. Huang, D. J. Kouri, M. Arnold, T. L. Marchioro II and D. K. Hoffman, J. Chem. Phys. **99**, 1028 (1993).
42. For an overview of the BLAS see <http://www.netlib.org/blas/faq.html>
43. H. J. Nussbaumer, *Fast Fourier Transforms and Convolution Algorithms*, (Springer –Verlag, New York, 1981)

TABLES

Table I. Lagrangian FD second derivative matrix coefficients. Numbers in brackets indicate powers of 10.

$2n+1$	$d_s, s = 0,1,\dots,n$		
9	-2.84722222222 0.253968253968(-01)	1.60000000000 -0.178571428571(-02)	-0.200000000000
11	-2.92722222222 0.396825396825(-01)	1.66666666667 -0.496031746032(-02)	-0.238095238095 0.317460317460(-03)
13	-2.98277777778 0.529100529101(-01) -0.601250601251(-04)	1.71428571429 -0.892857142857(-02)	-0.267857142857 0.103896103896(-02)
15	-3.02359410431 0.648148148148(-01) -0.226625226625(-03)	1.75000000000 -0.132575757576(-01) 0.118928690357(-04)	-0.291666666667 0.212121212121(-02)
17	-3.05484410431 0.754208754209(-01) -0.518000518001(-03)	1.77777777778 -0.176767676768(-01) 0.507429078858(-04)	-0.311111111111 0.348096348096(-02) -0.242812742813(-05)
19	-3.07953546233 0.848484848485(-01) -0.932400932401(-03) 0.507843645099(-06)	1.80000000000 -0.220279720280(-01) 0.128442985586(-03)	-0.327272727273 0.503496503497(-02) -0.115693130399(-04)
21	-3.09953546233 0.932400932401(-01) -0.145687645688(-02) 0.267286128999(-05)	1.81818181818 -0.262237762238(-01) 0.251848991345(-03) -0.108250882245(-06)	-0.340909090909 0.671328671329(-02) -0.321369806664(-04)
23	-3.11606438795 0.100732600733 -0.207390648567(-02) 0.808540540223(-05)	1.83333333333 -0.302197802198(-01) 0.423246221566(-03) -0.623731273886(-06)	-0.352564102564 0.846153846154(-02) -0.682206080813(-04) 0.234309268928(-07)

Table II. DFFD($\square = 10^{-3}$) second derivative matrix coefficients. Numbers in brackets indicate powers of 10.

$2n+1$	K_M/\square	$d_s, s = 0, 1, \dots, n$		
9	0.57	-3.00752631036 0.538284928126(-01)	1.73409774424 -0.715783320850(-02)	-0.277114173738
11	0.65	-3.09317424432 0.857932166572(-01)	1.81136085918 -0.203312767492(-01)	-0.333328232367 0.318623559756(-02)
13	0.70	-3.14695299609 0.112881460866 -0.173392583786(-02)	1.86135425441 -0.353836812411(-01)	-0.373309186173 0.958462639122(-02)
15	0.74	-3.18176011496 0.134006102161 -0.520324386831(-02)	1.89434052267 -0.493134756581(-01) 0.106208422138(-02)	-0.401313224347 0.173732544630(-01)
17	0.78	-3.20606683388 0.150651100423 -0.974588783394(-02)	1.91766953469 -0.615172937772(-01) 0.318373204492(-02)	-0.421909051551 0.253630788949(-01) -0.727196929671(-03)
19	0.80	-3.22299218142 0.163260241868 -0.144905270182(-01) 0.523654248651(-03)	1.93406079669 -0.714498945268(-01) 0.594898976062(-02)	-0.436782406626 0.325692020721(-01) -0.208584225828(-02)
21	0.82	-3.23565073477 0.173261981400 -0.191429182665(-01) 0.147065710509(-02)	1.94639721132 -0.797264050498(-01) 0.904046937689(-02) -0.403718655899(-03)	-0.448194652988 0.389999511305(-01) -0.393083408260(-02)
23	0.84	-3.24511651122 0.181079205310 -0.233799356983(-01) 0.274168249782(-02)	1.95566660040 -0.864348321532(-01) 0.121181195372(-01) -0.108567549576(-02)	-0.456894675479 0.444794495684(-01) -0.600477882385(-02) 0.322646265844(-03)

Table III. DFFD($\square = 10^{-4}$) second derivative matrix coefficients. Numbers in brackets indicate powers of 10.

$2n+1$	K_M/\square	$d_s, s = 0,1,\dots,n$		
9	0.46	-2.95463089055 0.417428777709(-01)	1.68853913589 -0.441628133062(-02)	-0.248562076568
11	0.54	-3.04815744881 0.692965489992(-01)	1.77082852536 -0.136466734520(-01)	-0.304011079096 0.162170037812(-02)
13	0.61	-3.10927537741 0.945879391564(-01) -0.740070089962(-03)	1.82646558524 -0.255340850297(-01)	-0.345740310257 0.558957488054(-02)
15	0.66	-3.15085169959 0.116052549350 -0.269839370338(-02)	1.86518705522 -0.379010120945(-01) 0.395506541984(-03)	-0.376916188202 0.113143689741(-01)
17	0.70	-3.18023073755 0.133684283047 -0.574480088363(-02)	1.89298475711 -0.494986655648(-01) 0.147746811877(-02)	-0.400414666665 0.178581512346(-01) -0.238422885519(-03)
19	0.73	-3.20176404869 0.148062930615 -0.949077758166(-02) 0.158804903573(-03)	1.91359155930 -0.598600651418(-01) 0.325180256103(-02)	-0.418450724893 0.245221555108(-01) -0.896881440188(-03)
21	0.76	-3.21751621028 0.159436234545 -0.134249789800(-01) 0.578959401646(-03)	1.92879277734 -0.686107768660(-01) 0.547787183108(-02) -0.111188043051(-03)	-0.432100334454 0.306902900963(-01) -0.197690696198(-02)
23	0.78	-3.22963300053 0.168702094698 -0.173419936914(-01) 0.128712230192(-02)	1.94055852097 -0.760868634860(-01) 0.797189229880(-02) -0.398753371880(-03)	-0.442866285531 0.363158574157(-01) -0.340211389200(-02) 0.827475903337(-04)

Table IV. DFFD($\epsilon = 10^{-5}$) second derivative matrix coefficients. Numbers in brackets indicate powers of 10.

$2n+1$	K_M/ϵ	$d_s, s = 0,1,\dots,n$		
9	0.37	-2.91759389235 0.350634616485(-01)	1.65741376974 -0.319702023083(-02)	-0.230484490985
11	0.46	-3.01370652873 0.589695114516(-01)	1.74046022973 -0.101371712423(-01)	-0.283431293138 0.993085049736(-03)
13	0.53	-3.07881836216 0.820816159831(-01) -0.386029108501(-03)	1.79876288173 -0.196945035156(-01)	-0.325015107320 0.365934989885(-02)
15	0.58	-3.12461689293 0.102775755548 -0.157051246937(-02)	1.84080635663 -0.303765129408(-01) 0.179013024771(-03)	-0.357414797410 0.791002084598(-02)
17	0.63	-3.15781834378 0.120561652244 -0.366079318978(-02)	1.87183556340 -0.410391882516(-01) 0.773693759881(-03)	-0.382681365371 0.132142505869(-01) -0.954475109185(-04)
19	0.66	-3.18244388407 0.135490686594 -0.648410950406(-02) 0.566900898173(-04)	1.89515466702 -0.509874619869(-01) 0.189432759599(-02)	-0.402450768489 0.189736207522(-01) -0.424959280580(-03)
21	0.70	-3.20092520468 0.147772740797 -0.972653138106(-02) 0.252413676250(-03)	1.91282871361 -0.598273241656(-01) 0.348133007385(-02) -0.362222030285(-04)	-0.417891713058 0.246738254974(-01) -0.106531944706(-02)
23	0.72	-3.21518337405 0.157927077948 -0.131409791880(-01) 0.642919547766(-03)	1.92656623086 -0.675664604252(-01) 0.540883900417(-02) -0.160366587811(-03)	-0.430169270519 0.300734143346(-01) -0.201375021728(-02) 0.246678536943(-04)

Table V. DFFD($\epsilon = 10^{-6}$) second derivative matrix coefficients. Numbers in brackets indicate powers of 10.

$2n+1$	K_M/ϵ	$d_s, s = 0,1,\dots,n$		
9	0.30	-2.89304092672 0.312901200130(-01)	1.63712687766 -0.259734807694(-02)	-0.219299316595
11	0.38	-2.98831371176 0.524220917154(-01)	1.71843373456 -0.817942221261(-02)	-0.269220835679 0.701403490614(-03)
13	0.45	-3.05488720754 0.734848372139(-01) -0.236624386935(-03)	1.77730652621 -0.161047793856(-01)	-0.309651152665 0.264469260617(-02)
15	0.51	-3.10281332019 0.929503728074(-01) -0.101357502373(-02)	1.82079637407 -0.253160104212(-01) 0.954570337130(-04)	-0.342008377870 0.590251288462(-02)
17	0.56	-3.13843684694 0.110289646360 -0.249588415711(-02)	1.85374329229 -0.349341866959(-01) 0.448999032235(-03)	-0.368001153554 0.102124928230(-01) -0.448692315917(-04)
19	0.60	-3.16533895294 0.125276752875 -0.464558721629(-02) 0.237641441984(-04)	1.87898399911 -0.442635004806(-01) 0.118398335173(-02)	-0.388810109332 0.151436283853(-01) -0.223373361303(-03)
21	0.64	-3.18590882326 0.137969267339 -0.728249556210(-02) 0.121501716503(-03)	1.89849622445 -0.528710978038(-01) 0.232093751449(-02) -0.137504414901(-04)	-0.405443625771 0.202721728793(-01) -0.614797570847(-03)
23	0.67	-3.20204582907 0.148739722477 -0.102259922299(-01) 0.345796563483(-03)	1.91393371099 -0.606652897814(-01) 0.381242063987(-02) -0.715909901745(-04)	-0.418947212154 0.253500050681(-01) -0.125720700084(-02) 0.862124142694(-05)

Figure Captions

Figure 1. The dispersion relation for the $2n+1 = 9$ point Lagrangian FD approximation (dashed curve) is compared to the exact result (solid curve).

Figure 2. The maximum absolute magnitude error between the scaled dispersion relation, $F(K)$, and the exact result $(-K^2)$, over K values in $[K_1, K_M]$. [$N = 1023$ was used in the discretization of K , Eq. (13).] The DFFD (solid curve) and Lagrangian FD (dashed curve) results for the $2n+1 = 9$ point case are shown.

Figure 3. (a) The absolute magnitude error in the eigenvalues of a 95×95 $DFFD_{11}(\Delta)$ matrix as compared to the exact result $(-K^2)$, as a function of K , for $\Delta = 10^{-3}, 10^{-5}, 10^{-7}$; (b) the error in the eigenvalues of a 95×95 $DFFD_{2n+1}(\Delta=10^{-5})$ matrices with $2n+1 = 7, 15,$ and 23 .

Figure 4. (a) K_M vs $2n+1$ for Lagrangian FD consistent with a maximum absolute error in the scaled dispersion relation given by Δ (b) K_M vs $2n+1$ for $DFFD_{2n+1}(\Delta)$; (c) The work parameter, $W = (2n+1)/K_M$, as a function of $2n+1$ for $DFFD_{2n+1}(\Delta)$.

Figure 5. The RMS error for the first 20 eigenvalues of the unit mass and frequency harmonic oscillator problem. Each symbol on the figure represents the RMS error result of a calculation with the FD (open circles) or various DFFD (see figure) approximations.

Figure 6. The total reaction probability for $D + H_2(v = j = 0) \rightarrow DH + H$ reaction in three dimensions (total angular momentum $J = 0$), based on the LSTH surface. See text for further details.

Figure 7. Maximum absolute error in the reaction probability for the $D + H_2$ reaction as a function of order for FD (open circles), and several DFFD methods. The Fourier limit for each case is given by a horizontal dashed line. (a) $N_R = N_r = 95$ grids in R and r, or $\Delta_R = 0.139 a_0$, $\Delta_r = 0.125 a_0$; (b) Same as (a), but correct FD dispersion relation is used in the analysis (see text); (c) $N_R = N_r = 79$ or $\Delta_R = 0.156 a_0$, $\Delta_r = 0.150 a_0$ and correct FD dispersion relation is used in the analysis.

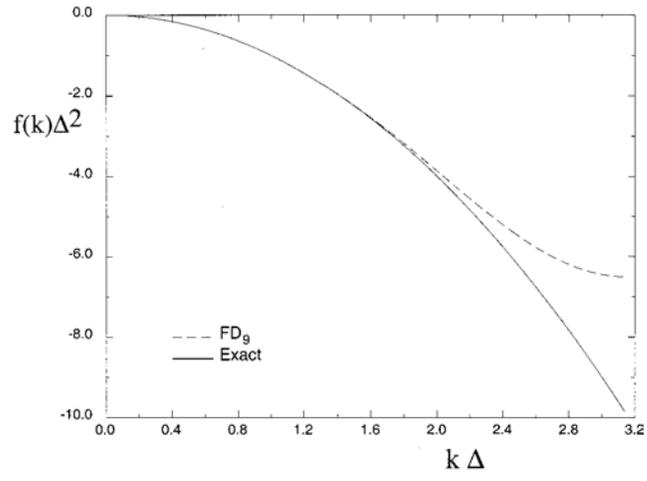


Fig. 1. Gray and Goldfield

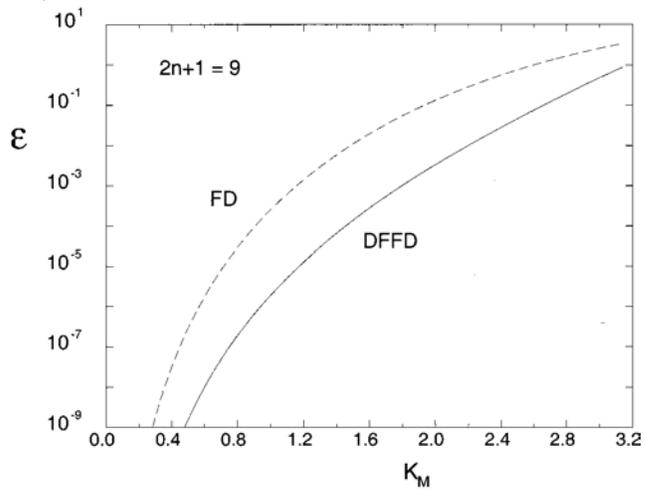


Fig. 2. Gray and Goldfield

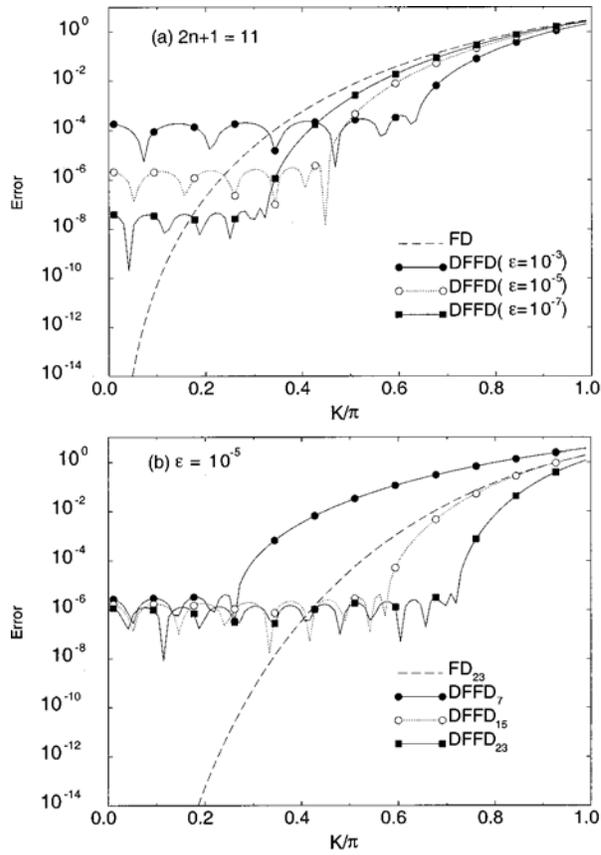


Fig. 3. Gray and Goldfield

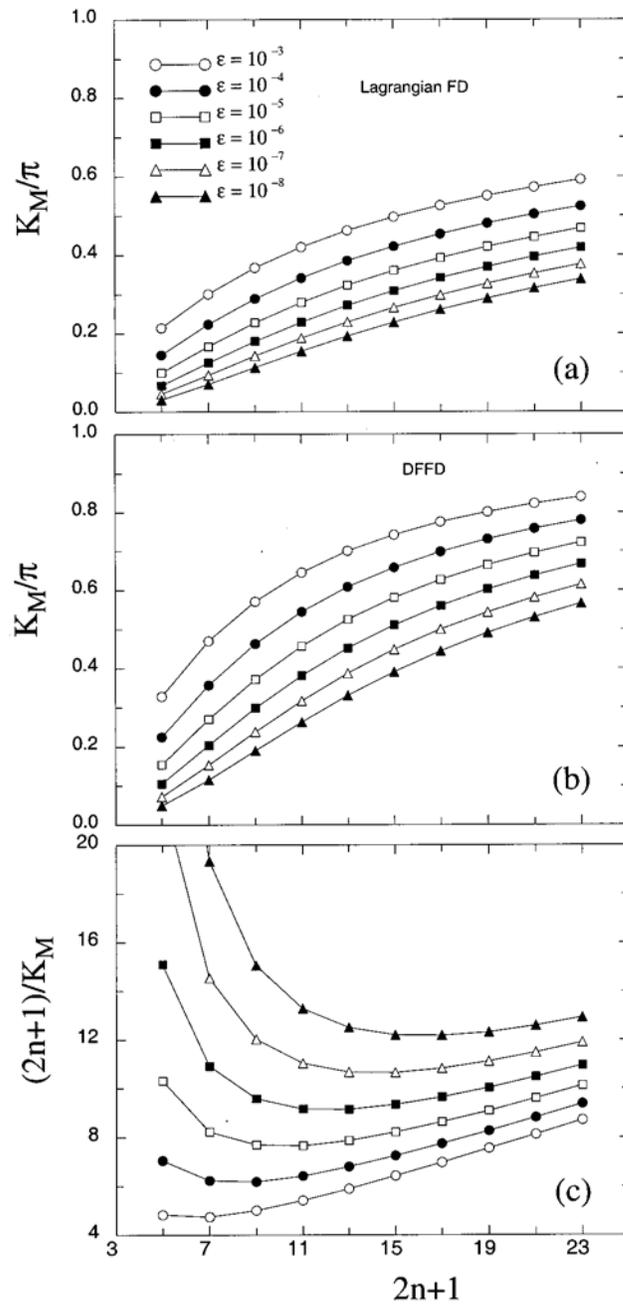


Fig. 4. Gray and Goldfield

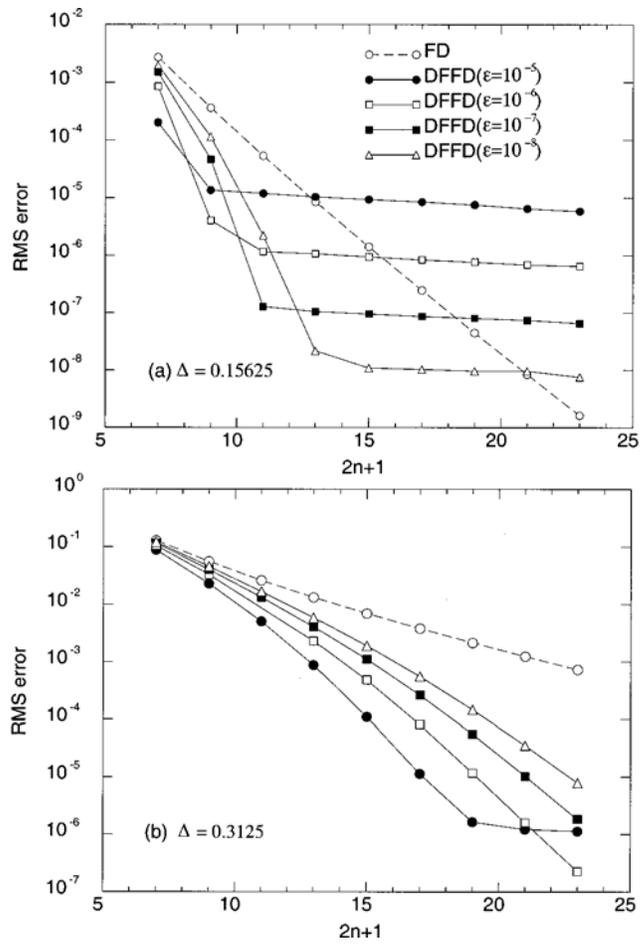


Fig. 5. Gray and Goldfield

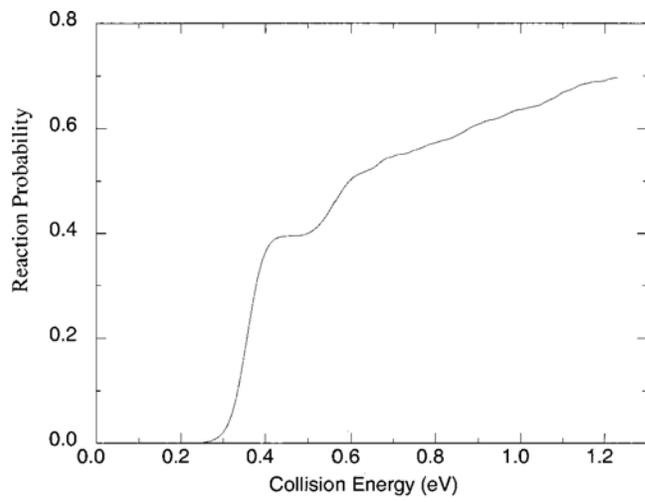


Fig. 6. Gray and Goldfield

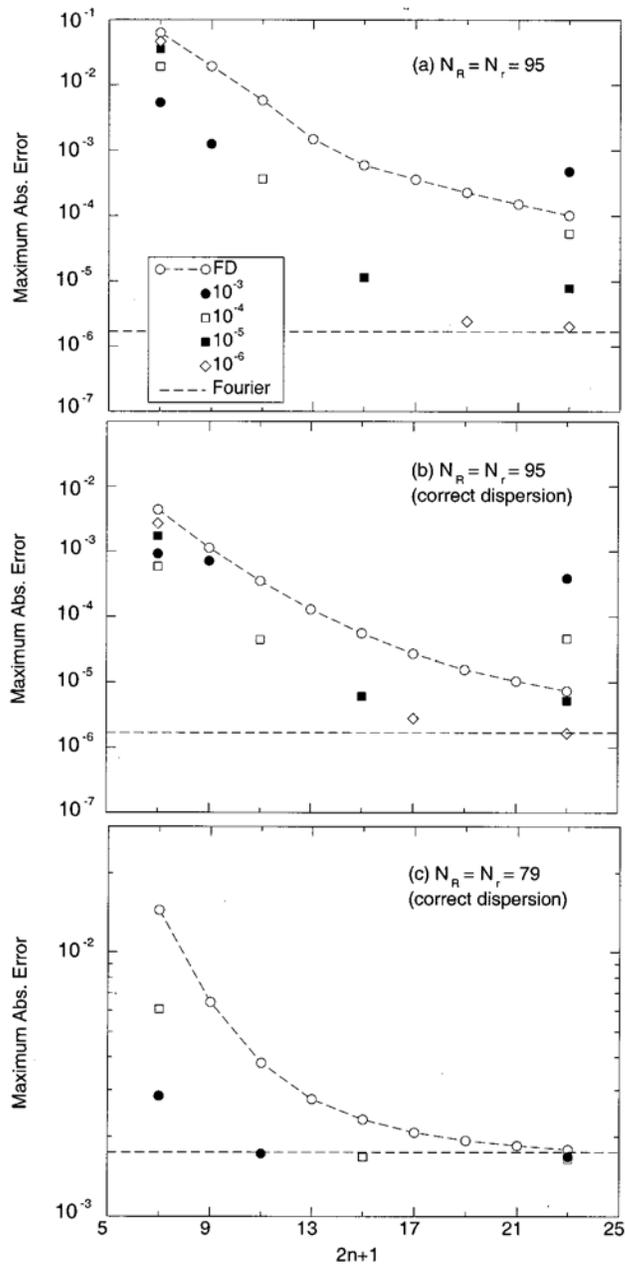


Fig. 7. Gray and Goldfield