Available online at www.sciencedirect.com
ScienceDirect
Journal of Computational Physics 227 (2008) 5200–5214
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

Extracting degenerate modes and frequencies from time-domain simulations with filter-diagonalization ☆

Gregory R. Werner [a,*], John R. Cary [a,b]

[a] Center for Integrated Plasma Studies, University of Colorado, Boulder, CO 80309, United States
[b] Tech-X Corporation, Boulder, CO 80303, United States

Received 22 May 2007; received in revised form 20 December 2007; accepted 27 January 2008
Available online 5 February 2008

Abstract

A variant of the filter-diagonalization method, using targeted excitation to filter out unwanted modes, can extract exactly or nearly degenerate eigenmodes and frequencies from time-domain simulations. Excitation provides a particularly simple way to produce filtered states with already-existing time-domain simulations, while requiring minimal storage space. Moreover, using broader excitations that cover the entire range of desired frequencies requires just one-fifth as much computation as using narrow excitations. With this method, almost any time-domain code can be easily turned into an efficient eigenmode solver with little or no change to the code. To distinguish M degenerate modes requires running at least M different simulations, so the computational effort is proportional to the size of the degeneracy, no matter how closely-spaced the modes; however, from those M simulations many other non-degenerate modes can also be extracted with high accuracy, without much extra effort. This method allows relatively simple FDTD algorithms to compete with frequency-domain solvers, offering advantages of simplicity, flexibility and ease of implementation; also, it scales to very large problems and massively parallel computation, and it can be used to extract high-frequency modes without first having to identify lower-frequency modes. The accuracy of this method is demonstrated by finding eigenmodes and frequencies of an electromagnetic resonant cavity.

Keywords: FDM; Filter-diagonalization eigenmode; Degenerate mode; Normal mode analysis; Harmonic inversion; FDTD; Simulation; Time-domain; Electromagnetic

1. Introduction

The time evolution of many systems (namely, linear systems invariant with respect to time translation) can be described as the superposition of various eigenmodes oscillating at their respective eigenfrequencies. Insight into such systems often comes from finding some of the eigenmodes and frequencies, a process equivalent to

☆ This work was supported by the US Department of Energy Grant DE-FG02-04ER41317.
* Corresponding author. Tel.: +1 303 735 2461; fax: +1 303 492 0642.
E-mail address: Greg.Werner@colorado.edu (G.R. Werner).

0021-9991/$ - see front matter © 2008 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2008.01.040

the partial diagonalization of a matrix. In contrast, time-domain methods describe the specific evolution of a system by integrating the equations of motion forward in time from a given initial condition. Usually time evolution and eigensolving are performed by very different algorithms; however, the filter-diagonalization method (FDM) can find eigenmodes easily and efficiently from the time evolution of a system.

We will show how to extract eigenmodes efficiently from nearly any time-domain simulation (whether real or complex) using FDM, using simple excitation to produce filtered states, rather than special frequency filtration operators or post-simulation FFTs. Filtration by excitation performs as well as filter operators, and for very large problems requires much less storage (and computation) than post-simulation FFTs. Moreover, we have developed a faster filtration technique, exciting states containing roughly equal amplitudes of all desired eigenmodes, rather than producing states filtered narrowly around individual frequencies.

Because FDM competes well with dedicated eigenmode solvers, such as the Lanczos block algorithm [1], this paper demonstrates a simple method with which any time-domain code can be turned into an eigenmode solver with very little work; our electromagnetic time-domain code required no modifications to apply this method – we simply had to write some short post-processing code (using standard linear algebra routines, such as those in LAPACK).

Before describing the method in more detail, we list some general advantages common to FDM methods:

- Intensive computation can be done with matrix-vector multiplication; time-domain simulations can be used for the intensive computation (in many cases) with no modification.
- Only one vector must be stored in memory.
- Eigenmodes need not be orthogonal.
- Eigenvalues in any range can be found.
- Degenerate modes can be found.
- Eigenvalue errors can be easily estimated.
- The method knows whether it has found all eigenvalues in the desired range.
- Accuracy increases more than exponentially with computation time, limited by machine precision times the maximum/minimum eigenvalue ratio.

We will discuss the computationally intensive part of FDM in Section 4, where we compare our broad filtering approach with the narrow filtering used by other FDM applications. However, before we get to that most important section, we offer a brief overview of FDM (Section 2), and we review in Section 3 the linear algebra necessary to perform the diagonalization part of filter-diagonalization. Understanding how the diagonalization works will facilitate discussion of the filtering.

We demonstrate the algorithm and determine its accuracy by finding eigenmodes of an electromagnetic resonant cavity: in Section 5 we run five simulations to find 15 eigenmodes and eigenfrequencies of a rectangular cavity with 4-fold degeneracies.

2. FDM overview

The filter-diagonalization method [2–6] extracts eigenmodes and eigenvalues of a linear operator for which the eigenvalues are related to mode frequencies. FDM has been applied in the literature mostly to the Schrödinger equation for vibrational and electronic structure calculations of small molecules; FDM has also been applied to NMR [7–9] (these are two-dimensional applications of FDM) and to a few electromagnetic calculations [10,11] (however, one of these uses a special time-advance that works only with complex fields, the other does not calculate modes or use mode pattern information at all).

Filter-diagonalization partially diagonalizes a large matrix H by transforming H to block-diagonal form with one small block and one large block (that is the filtering part). The large block is ignored, and the small block is diagonalized by standard linear algebra routines appropriate for small matrices (and that is the diagonalization part). The diagonalization of the small matrix can be accomplished in a few seconds, but intensive computation is required to put H in block diagonal form; this computation generally involves many matrix-vector multiplications using H.

H is transformed to block-diagonal form by finding a small, invariant subspace of H; we call this subspace an eigen-subspace, because it has a basis of eigenvectors. Finding an eigen-subspace means finding a set of vectors $\mathbf{s}_1, \ldots, \mathbf{s}_L$ that spans the subspace. From these vectors we can make a basis for the subspace, and, applying H to each basis vector, we can compute the matrix of H restricted to this subspace (the small block).

The $\mathbf{s}_\ell$ are found by filtering eigenvectors based on their eigenvalues, by associating eigenvalues with frequencies. For example, if the vector $\mathbf{s}(t)$ evolves as the sum of eigenmodes oscillating with frequencies that are eigenvalues of H, we can Fourier transform $\mathbf{s}(t)$ to select eigenvectors with eigenvalues near a desired frequency. Eigenvalue-based filtering is convenient because we usually know roughly the range of desired eigenvalues; it can quickly separate modes with far-apart frequencies.

FDM uses Fourier transformation or frequency filtering to separate a small range of eigenvalues from the majority of eigenvalues (which are far away). Simple linear algebra can then be applied to separate the closely-spaced eigenvalues in the desired range. Thus well-separated modes are distinguished by their different frequencies while nearby modes are distinguished by their different mode patterns.

We wish particularly to emphasize that this method can distinguish degenerate eigenmodes, something that time-domain codes are generally considered unable to do. For example, two modes with frequencies separated by a part in 10^5 can be distinguished to high precision after evolution for many fewer than 10^5 oscillation periods; in fact, two modes with exactly the same frequency can be distinguished. To distinguish a 2-fold degeneracy requires twice as much work: two initial conditions must be evolved, rather than one.

FDM identifies frequencies with high precision – well beyond the limits of the uncertainty principle. That is, a frequency may be measured to a part in 10^{10} from a simulation run for only 100 oscillations. The uncertainty principle is not really violated, however [12]; it relates the required simulation duration to the average separation between mode frequencies [13].

3. Diagonalization: extracting eigenvalues and eigenmodes

Although filtering comes before diagonalization in FDM, we first review the diagonalization, because understanding the diagonalization technique will reveal the requirements for the filtering. Unlike other treatments of diagonalization [4,5,7] we avoid inner products and the issue of whether they are Hermitian or complex symmetric.

The diagonalization part of FDM uses small-scale linear algebra to distinguish nearby eigenvalues within the desired range from one another. The goal is to find eigenmodes $\mathbf{v}_m, m = 1, \ldots, M$, which are eigenmodes of some operator H; that is, $H\mathbf{v}_m = \lambda_m \mathbf{v}_m$. The eigenmodes do not need to be orthogonal. In this section we present the linear algebra necessary to extract the λ_m and $\mathbf{v}_m$, assuming we have found a small number $L \geqslant M$ of vectors $\mathbf{s}_\ell$ spanning the subspace containing the desired eigenmodes.

An elementary theorem of linear algebra states that a matrix is determined by its action on a basis; we determine the matrix H – or rather, the small block of H restricted to the subspace of desired eigenmodes – through the relationship between the $\mathbf{s}_\ell$ and

$$\mathbf{r}_\ell \equiv H\mathbf{s}_\ell. \tag{1}$$

(We could equally well consider application of any sufficiently invertible function of H, such as $\exp(\mathrm{i}\tau H)$, which would yield eigenvalues $\exp[\mathrm{i}\lambda_m\tau]$ with the same eigenvectors [6].)

Because the $\mathbf{s}_\ell$ span the desired eigen-subspace, any $\mathbf{v}_m$ can be written

$$\mathbf{v}_m = \sum_{\ell=1}^{L} a_{m,\ell}\mathbf{s}_\ell. \tag{2}$$

Applying H yields

$$\lambda_m \mathbf{v}_m = \sum_{\ell=1}^{L} a_{m,\ell}\mathbf{r}_\ell. \tag{3}$$

Once the $\mathbf{s}_\ell$ and $\mathbf{r}_\ell$ are found, Eqs. (2) and (3) can be solved for the λ_m, which are eigenvalues of H, and the coefficients $a_{m,\ell}$, which express the eigenvectors of H in terms of the known $\mathbf{s}_\ell$. In practice this set of equations

may be both highly overdetermined and nearly underdetermined, like many parameter-fitting problems; singular value decomposition (SVD) offers a robust solution [14].

To ease the burden of computational linear algebra, we work mainly with P components of the vectors $\mathbf{s}_\ell$ and $\mathbf{r}_\ell$, e.g. $s_{\ell,p}$ and $r_{\ell,p}$ for $p = 1,\ldots,P$. (This is a particularly simple and computationally efficient case of taking the inner product of Eqs. (2) and (3) with a set of vectors Φ_p; i.e. $s_{\ell,p} = \langle \Phi_p | \mathbf{s}_\ell \rangle$ for a chosen set of P vectors Φ_p [6].)

The numerical work begins with the construction of the $P \times L$ matrices S and R

$$S_{p\ell} \equiv s_{\ell,p}, \quad R_{p\ell} \equiv r_{\ell,p}, \tag{4}$$

where the ℓth column of S holds the P components of $\mathbf{s}_\ell$, and similarly each column of R holds components of $\mathbf{r}_\ell$.

Eqs. (2) and (3), written for each of the P components, are

$$v_{m,p} = \sum_{\ell=1}^{L} S_{p\ell} a_{m,\ell}, \quad \lambda_m v_{m,p} = \sum_{\ell=1}^{L} R_{p\ell} a_{m,\ell}. \tag{5}$$

Treating $\mathbf{a}_m$ as a column vector with L components (labeled $a_{m,\ell}$) and treating $\mathbf{v}_m$ as a P-component vector, these equations become

$$\mathbf{v}_m = S\mathbf{a}_m, \quad \lambda_m \mathbf{v}_m = R\mathbf{a}_m, \tag{6}$$

which combine to form a generalized eigenvalue equation

$$R\mathbf{a}_m = \lambda_m S\mathbf{a}_m. \tag{7}$$

S and R are known; if we can solve $R\mathbf{a} = \lambda S\mathbf{a}$ for its eigenvalues λ and eigenvectors $\mathbf{a}$, then we will have found the eigenvalues of H, as well as its eigenmodes $\mathbf{v}_m$ (as linear combinations of the known $\mathbf{s}_\ell$). When solving this generalized eigenvalue equation, intuition may be aided by considering that $R = HS$, although this is not strictly true unless P includes the full set of components.

We solve Eq. (7) in spirit by multiplying both sides by S^{-1} and solving the resulting standard eigenvalue equation. Equivalently (since the null space of $S^\dagger$ is orthogonal to the range of S) we may prefer to solve

$$S^\dagger R\mathbf{a}_m = \lambda_m S^\dagger S\mathbf{a}_m, \tag{8}$$

inverting $S^\dagger S$ instead of S. However, S and $S^\dagger S$ might not be invertible using finite-precision arithmetic, and they might not even be invertible in theory – S might not even be square ($P > L$ facilitates error analysis as we later demonstrate). Truncated SVD techniques can used to accomplish this inversion (see [14] on SVD) as suggested in [4]; however, we prefer to use SVD with Tikhonov regularization [7].

In summary, brief directions for finding eigenvalues and eigenvectors of H:

(1) Find L filtered state vectors $\mathbf{s}_\ell$ that span the subspace of desired eigenmodes of H. L must be at least as large as the number of modes in the subspace, yet small enough that the ensuing eigenvalue problem can be easily solved numerically.
(2) Apply H to each filtered state vector to get $\mathbf{r}_\ell = H\mathbf{s}_\ell$.
(3) Construct the $P \times L$ matrix $S_{p\ell} = s_{\ell,p}$ from P different components of each filtered state $\mathbf{s}_\ell$, where $P \geqslant L$. Similarly construct $R_{p\ell} = r_{\ell,p}$.
(4) Solve the generalized eigenvalue equation $S^\dagger R\mathbf{a} = \lambda S^\dagger S\mathbf{a}$ for eigenvalues λ and eigenvectors $\mathbf{a}$. This can be numerically challenging, so we suggest finding the eigenvalues of $S^\dagger S$ and then solving $(S^\dagger S + \alpha^2_{\text{cutoff}})^{-1} S^\dagger R\mathbf{a} = \lambda\mathbf{a}$, where α^2 is less than the significant eigenvalues and greater than the insignificant eigenvalues of $S^\dagger S$.
(5) Eigenvalues of H are the λ_m. Eigenvectors of H are $\mathbf{v}_m = \sum_{\ell=1}^{L} a_{m,\ell}\mathbf{s}_\ell$.
(6) If $P > L$, estimate the relative error ϵ_m of each mode (see Section 3.1). Modes with errors near one are doubtful; if H has only real eigenvalues, then modes with eigenvalues that have imaginary parts larger than the error are also doubtful. Increasing α_{cutoff} may vanquish doubtful modes.

We note that the eigenvalues λ_m and the coefficients $a_{m,\ell}$ are calculated using only P components of the $\mathbf{s}_\ell$ and $\mathbf{r}_\ell$. The entire vectors ($\mathbf{s}_\ell$ and $\mathbf{r}_\ell$) are needed only to find all components of the eigenmodes.

3.1. Accuracy

Error in calculating eigenvalues and eigenmodes arises from the incomplete isolation of a subspace of M eigenmodes, so that Eqs. (2) and (3) are only approximately valid. The actual vectors $\mathbf{s}_\ell$ and $\mathbf{r}_\ell$ include noise comprising other eigenvectors.

When an eigenvector $\mathbf{v}_m$ is calculated, its error can be told by comparing $H\mathbf{v}_m$ to $\lambda_m\mathbf{v}_m$ [4]. As long as $P > L$ (preferably $P - L \gg 1$), the relative error in each eigenvalue can be easily estimated by

$$\epsilon_m = \frac{\|R\mathbf{a}_m - \lambda_m S\mathbf{a}_m\|}{\|R\mathbf{a}_m\|}. \tag{9}$$

Eigenvalues with large ϵ_m (close to one) are suspicious; they may not be close to eigenvalues of H. Later we will show the accuracy of this estimate in an example (see Fig. 1).

Eigenmodes are calculated with similar accuracy, except that nearby eigenmodes get mixed together. If eigenvalue λ_1 is close to λ_2, then the relative error in the corresponding eigenmodes will be increased roughly by a factor of $|\lambda_1|/|\lambda_2 - \lambda_1|$ (up to 100% error). Exactly degenerate modes cannot be distinguished at all (not even theoretically, without recourse to some operator other than H). However, the subspace spanned by the two eigenmodes will be accurately computed, even when $\lambda_1 = \lambda_2$.

The accuracy for a given eigenvalue will be limited by how much of the corresponding mode is present in the filtered states. We can estimate the relative contribution of mode $\mathbf{v}_m$ to the $\mathbf{s}_\ell$ by

$$C_m = \sqrt{\frac{\sum_{p=1}^{P}\left|\sum_{\ell=1}^{L} a_{m,\ell}s_{\ell,p}\right|^2}{\sum_{\ell=1}^{L}|a_{m,\ell}|^2}} = \sqrt{\frac{\sum_{p=1}^{P}|v_{m,p}|^2}{\sum_{\ell=1}^{L}|a_{m,\ell}|^2}}, \tag{10}$$

(we will show in Fig. 2 how this relates to the error).

3.2. Source of error

While noise in the $\mathbf{s}_\ell$ may be reduced to nearly machine precision by carefully filtering out unwanted eigenmodes, subsequent application of H disproportionately emphasizes unwanted modes with high eigenvalues. For example, if $\mathbf{s} = \mathbf{v}_1 + \mathbf{v}_{\max}$, then $H\mathbf{s} = \lambda_1\mathbf{v}_1 + \lambda_{\max}\mathbf{v}_{\max}$. Thus modes with eigenvalue magnitude $\lambda_{\max}$ are emphasized in the $\mathbf{r}_\ell$ by a factor $\lambda_{\max}/\lambda_1$ relative to modes with eigenvalue λ_1. If the machine precision is 10^{-16}, and eigenvalues of interest are around 1, while the maximum eigenvalues are near $\lambda_{\max} \sim 10^6$, then one will encounter a relative error of at least 10^{-10} in the $\mathbf{r}_\ell$. With double precision this error is not a major concern, but it may pose problems for single-precision computation.

4. Filtering

In this section, we demonstrate a new method for filtering out unwanted eigenmodes, using excitations with frequency-domain properties that separate the desired from the undesired modes, but do not prefer one frequency over another within the desired range; after all, the diagonalization step is responsible for separating frequencies within the desired range.

We showed in Section 3 how to identify eigenmodes and eigenvalues within an eigen-subspace spanned by a set of vectors $\mathbf{s}_\ell$. Such a set of vectors is found by computing a set of linearly independent vectors from which all the unwanted eigenmodes have been removed.

We will consider systems with oscillatory eigenmodes that evolve as $\exp(-i\omega_m t)$ for various eigenfrequencies ω_m. Such a system can always be described by a state vector $\mathbf{s}$ that evolves in time according to

$$\frac{\mathbf{s}(t+\Delta t) - \mathbf{s}(t)}{\Delta t} = -iH\mathbf{s}(t). \tag{11}$$

An eigenmode of H with eigenvalue λ_m evolves with frequency ω_m, where

$$-i\omega_m\Delta t = \ln(1 - i\lambda_m\Delta t). \tag{12}$$

The operator H need not be explicitly known, as long as an algorithm (a time-domain simulation) can advance any state $\mathbf{s}(t)$ to time $t+\Delta t$; we can then evaluate $H\mathbf{s}(t)$ from $\mathbf{s}(t)$ and $\mathbf{s}(t+\Delta t)$. We emphasize that $\mathbf{s}(t)$ may be real, and may be advanced in time without any complex arithmetic, while still being described as above. This form covers a large range of systems, namely those with linear, time-invariant equations of motion.

The eigenfrequencies may be complex, but if they have large imaginary parts (e.g. if resonances are broad compared to typical mode separation) then the techniques described here may not adequately isolate the desired modes.

4.1. Using broadly-filtered states

To find a set of state vectors $\mathbf{s}_\ell$ that spans the desired subspace, we first drive a simulation to exclude all unwanted modes, so that after time $t=T$ the simulation state $\mathbf{s}(T)$ contains all desired eigenmodes, excluding all others. We then evolve the simulation further to find other linearly independent states containing the same modes; $\mathbf{s}(T)$ and $\mathbf{s}(T+\tau)$ contain the same modes, and they are linearly independent.

A set of vectors spanning the desired subspace is therefore $\mathbf{s}_\ell=\mathbf{s}(T+\ell\tau)$ for $\ell=1,\ldots,L$ and $L\tau\approx\pi/\Delta\omega_{\text{avg}}$ – as long as $L\geqslant M$ – where M is the number of modes in the subspace, and $\Delta\omega_{\text{avg}}$ is the average spacing between the modes. If two modes have frequencies separated by much less than $\Delta\omega_{\text{avg}}$, they will be less accurately calculated; Section 4.3 will explain how to find such degenerate modes.

This strategy can reduce computation time by a factor of five or more (as shown in Section 4.4).

4.2. Excitation to exclude unwanted frequencies

Though often less difficult, driving (or exciting) a system with a certain $f(t)$ is essentially equivalent to filter operators that emphasize frequencies ω with relative strength $|\tilde{f}(\omega)|$, where $\tilde{f}$ is the Fourier transform of $f(t)$ (see [15,16] and Appendix A).

Driving a time-domain simulation is easily done by adding a quantity $\mathbf{g}f(t)\Delta t$ to $\mathbf{s}(t)$ at every time step Δt; thus Eq. (11) becomes

$$\frac{\mathbf{s}(t+\Delta t)-\mathbf{s}(t)}{\Delta t}=-\mathrm{i}H\mathbf{s}(t)+\mathbf{g}f(t). \tag{13}$$

The driver $\mathbf{g}f(t)$ will tend to excite eigenmodes that have field patterns similar to $\mathbf{g}$ and that have frequencies present in $f(t)$. For example, electromagnetic simulations are typically excited by a time-varying current density (at every time step, the current is added to the electric field according to Ampere's Law); a mode with a large axial electric field will be most strongly excited by an axial current oscillating at the mode frequency.

Other FDM variants generate each filtered state around a different frequency, essentially using $f(t)=\Theta(t)\Theta(T-t)\sin(\omega_\ell t)$, where Θ is the Heaviside step function, or a Gaussian-modulated sinusoid $f(t)=\sin(\omega_\ell t)\exp[-\sigma_\omega^2(t-T/2)^2/2]$. Such excitation can be used with time-domain simulations to apply FDM as in, e.g. [4,5,15,16].

However, it is faster to generate the $\mathbf{s}_\ell$ from broadly-filtered states as described in Section 4.1 (Section 4.4 compares the two approaches). The following driver:

$$f_1(t)=-2\frac{\sin[\omega_1(t-T/2)]}{t-T/2}\exp\left[-\frac{\sigma_\omega^2(t-T/2)^2}{2}\right], \tag{14}$$

yields a state $\mathbf{s}(T)$ that excludes frequencies outside the interval $[-\omega_1,\omega_1]$, because it has a Fourier transform of nearly unity amplitude between $-\omega_1$ and ω_1 and nearly zero elsewhere:

$$|\tilde{f}_1(\omega)|=\frac{1}{\sqrt{2\pi}\sigma_\omega}\int_{-\infty}^{\omega}\mathrm{d}\omega'\exp\left[-\frac{(\omega'+\omega_1)^2}{2\sigma_\omega^2}\right]-\exp\left[-\frac{(\omega'-\omega_1)^2}{2\sigma_\omega^2}\right] \tag{15}$$

(here $f_1(t)$ is normalized so that $\tilde{f}_1(0)=1$ in the limit of $\omega_1\gg\sigma_\omega$).

The function $\tilde{f}_1(\omega)$ is to the ideal broadband filter $\Theta(\omega_1^2 - \omega^2)$ what a Gaussian $\tilde{f}(\omega)$ centered around frequency ω_1 is to the ideal narrow-band filter, the Dirac delta-function $\delta(\omega - \omega_1)$. $\tilde{f}_1(\omega)$ has tails slightly steeper than a Gaussian with width σ_ω.

For a frequency excitation between ω_1 and ω_2, one can simply subtract $f_1(t)$ from $f_2(t)$ (defined as f_1 above, replacing ω_1 with ω_2).

Roughly, to reduce amplitudes of modes with frequencies above ω_b to a relative level h, we must choose σ_ω so that $\exp[-(\omega_b - \omega_2)^2/(2\sigma_\omega^2)] < h$ and run the simulation for a time T such that $\exp[-\sigma_\omega^2(T/2)^2/2] < h$ (this ensures that the abrupt start and stop of the driving force at $t = 0$ and $t = T$ do not appreciably excite unwanted modes).

The choice of σ_ω is determined by the mode spacing. From σ_ω we can determine the simulation duration T. For example, we might want to reduce unwanted mode amplitudes by $h \sim 10^{-16}$ in a double precision simulation; this requires

$$T > \frac{17}{\sigma_\omega}. \tag{16}$$

Of course, shorter durations may be used, but results will be less precise due to low-level contamination by unwanted modes.

4.3. Distinguishing degeneracies

The filtered states $\mathbf{s}_\ell$, as found above, cannot span degenerate subspaces comprising modes with the same (or nearly the same) frequencies. A simple example illuminates the problem: suppose a system has only two modes, but they have identical frequencies. A random initial state might be $\mathbf{s}(0) = b_1\mathbf{v}_1 + b_2\mathbf{v}_2$, which evolves as $\mathbf{s}(t) = [b_1\mathbf{v}_1 + b_2\mathbf{v}_2]e^{-i\omega t}$, which is always proportional to $\mathbf{s}(0)$. Because the modes evolve with the same frequency, every filtered state $\mathbf{s}_\ell$ has the same linear combination of $\mathbf{v}_1$ and $\mathbf{v}_2$; to span the space of $\mathbf{v}_1$ and $\mathbf{v}_2$, another independent combination of $\mathbf{v}_1$ and $\mathbf{v}_2$ is needed. However, if one computes two sets of filtered vectors from two simulations with different initial conditions, perhaps $\mathbf{s}^{(1)}(0) = b_1^{(1)}\mathbf{v}_1 + b_2^{(1)}\mathbf{v}_2$ and $\mathbf{s}^{(2)}(0) = b_1^{(2)}\mathbf{v}_1 + b_2^{(2)}\mathbf{v}_2$ then one can form arbitrary linear combinations of $\mathbf{v}_1$ and $\mathbf{v}_2$ with different combinations of $\mathbf{s}_\ell^{(1)}$ and $\mathbf{s}_\ell^{(2)}$.

To find eigenmodes with exactly or nearly the same frequency, one can run several simulations, with each simulation excited by a different driving pattern $\mathbf{g}$ (which is the same as applying a filter operator to different initial conditions $\mathbf{g}$). If one expects at most J-fold degeneracies (or near-degeneracies, compared to the average frequency spacing), one must run at least J different simulations, including the $s_{\ell,p}$ from all simulations in the matrix $S_{p\ell}$ and similarly for $R_{p\ell}$. By running $J + 1$ simulations, one can verify that any degeneracies are at most J-fold.

4.4. Broad vs. narrow filtering

Isolating an eigen-subspace using broadly-filtered states takes less computation time than the usual method of computing narrowly-filtered states, as long as eigenmodes are desired and the problem is large enough that storing and performing an FFT of every component of $\mathbf{s}(t)$ takes more time than running the simulation itself (i.e. whenever a low-storage method is necessary). We consider a simplified analysis, extracting M modes with average spacing $\Delta\omega_{\text{avg}}$.

With broad filtering, the bulk of computation is used to compute the first filtered state. For an accuracy of $h \sim 10^{-12}$, a simulation time $T > 15/\Delta\omega_{\text{avg}}$ is required (demanding that $\sigma_\omega < \Delta\omega_{\text{avg}}$ and $T > 2(7.4)/\sigma_\omega$). These parameters suppress modes more than $7.4\Delta\omega_{\text{avg}}$ away from the desired range by at least h. We can extract as many additional independent states as we need by running the simulation for an additional time $\pi/\Delta\omega_{\text{avg}}$. Therefore the total time is approximately $18/\Delta\omega_{\text{avg}}$.

With narrow filtering, using a square window for best performance (see Appendix C), we must calculate at least M different filtered states, each of which takes a time $T > 2\pi/\Delta\omega_{\text{avg}}$ (see Appendix B). The total time is therefore $2\pi M/\Delta\omega_{\text{avg}}$.

Narrow filtering appears to be faster when $M < 3$; however, because a single square window provides poor isolation, one must always use many filtered states at regularly-spaced frequencies around the desired frequencies to achieve reasonable accuracy (for an accuracy of $h \sim 10^{-12}$ we estimate roughly that more than 15 filtered states would be needed); see Appendix C for more explanation.

Therefore, we expect broad filtering to require less than 1/5 the computation time needed for narrow filtering.

Though slower, square-window filtering is more robust than broad filtering. If the broadly-filtered state does not sufficiently exclude unwanted modes (perhaps because the mode density turns out to be higher than expected), then the accuracy of all modes will suffer. This can be a problem when the mode density increases rapidly with frequency. With square-window-filtering, modes in the center of the range may still be well-isolated even when modes at the edge of the range cannot be accurately distinguished.

Of course, when eigenmodes are not needed, Fourier transformation (using a square window) of the time evolution of a small number of components $s_p(t)$ from a single simulation can yield all the filtered state vector components necessary to find the frequencies. In this case, the total simulation time is merely $2\pi/\Delta\omega_{\text{avg}}$ (assuming the Fourier transformation of P components takes negligible time), which should always be faster than running a separate simulation for each filtered state.

5. Modes in a nearly square cavity

To evaluate the technique described in this paper, we calculated TM electromagnetic resonant eigenmodes of a rectangular 2D cavity (we use TM to mean that the magnetic field is transverse to the unsimulated third dimension). Here the operator of interest is the discretized version of $H = \nabla \times \nabla\times$; we used the standard Yee algorithm [17] to discretize H and evolve the electric and magnetic fields.

With sides of $L_x = 1$ m and $L_y = 1.00001$ m, we expect near-degeneracies. For this cavity, the eigenmodes and eigenvalues of the finite-difference equations are known exactly. Eigenmode TM_{mn} has an electric field

$$E_z(x,y) = \sin\left(\frac{m\pi x}{L_x}\right)\sin\left(\frac{n\pi y}{L_y}\right), \tag{17}$$

(regardless of the cell lengths Δx and Δy). The eigenvalues, however, do depend on the grid cell size

$$k_{mn}^2 = \frac{4}{\Delta x^2}\sin^2\left(\frac{m\pi\Delta x}{2L_x}\right) + \frac{4}{\Delta y^2}\sin^2\left(\frac{n\pi\Delta y}{2L_y}\right). \tag{18}$$

The oscillation frequency of mode TM_{mn} is $\omega_{mn} \approx ck_{mn}$, where c is the speed of light; this relationship is approximate because of the discretization of time. We used a 500×500 grid ($\Delta x = L_x/500, \Delta y = L_y/500$).

We chose to isolate modes between $k_1 = 24\ \text{m}^{-1}$ and $k_2 = 27\ \text{m}^{-1}$, well above the lowest TM mode, which has $k \approx 4.4\ \text{m}^{-1}$. From the average mode density (one TM mode per $4\pi^2/(L_xL_y)$ volume of k-space) we can expect about 13 modes in this range; therefore, the average mode spacing should be approximately $\Delta k_{\text{avg}} = (k_2 - k_1)/13$.

5.1. Excitation

To isolate the desired subspace of eigenmodes, we used an excitation as described in Section 4.1, with $\omega_1 = c(24.1\ \text{m}^{-1})$ and $\omega_2 = c(26.9\ \text{m}^{-1})$ (where c is the speed of light) and $\sigma_\omega = \Delta\omega_{\text{avg}} = (\omega_2 - \omega_1)/13$. The excitation was run for a time $T = 2(7.4)/\sigma_\omega$. After the excitation turned off, the simulation ran an additional time $\pi/\Delta\omega_{\text{avg}}$, during which field patterns (the $\mathbf{s}_\ell$) were periodically saved to disk.

To excite a simulation, we added a source current J_z to Maxwell's equations (expressed here in convenient units)

$$\begin{aligned}\frac{\partial}{\partial t}\mathbf{B}(x,y,t) &= -\nabla\times\mathbf{E}(x,y,t),\\ \frac{\partial}{\partial t}\mathbf{E}(x,y,t) &= \nabla\times\mathbf{B}(x,y,t) - \hat{\mathbf{z}}J_z(x,y,t).\end{aligned} \tag{19}$$

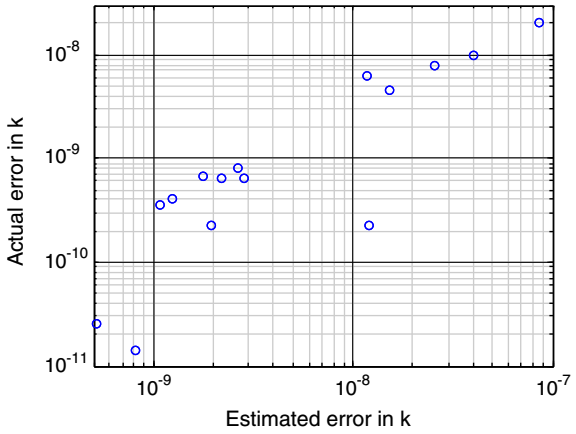

Fig. 1. Actual error in k_{mn} vs. (one-half) the error estimated by Eq. (9).

The source current simply adds an amount $J_z(x,y,t)\Delta t$ to the electric field E_z at each time step. The spatial profile of J_z was chosen by assigning a random number between -1 and 1 to each grid point (x,y); the spatial profile was then modified by a time-dependent prefactor

$$J_z(x,y,t) \propto \begin{cases} 2\left[\frac{\sin[\omega_1(t-T/2)]}{t-T/2} - \frac{\sin[\omega_2(t-T/2)]}{t-T/2}\right] e^{-\sigma_\omega^2(t-T/2)^2/2} & 0 \leqslant t \leqslant T, \\ 0 & \text{otherwise}, \end{cases} \tag{20}$$

selecting $T/2 = 7.4/\sigma_\omega$.

To verify that any degeneracies were at most 4-fold, we had to run 5 different simulations; however, we show results using just the first four of those simulations.

5.2. Choosing H

Eq. (19) is written in the same form as Eq. (13), where the state of the system is uniquely determined by $\mathbf{s} = (\mathbf{E}, \mathbf{B})$. However, we will find eigenmodes of a different but closely-related operator H, which describes the time evolution of $\mathbf{s} = \mathbf{E}$ as a second-order differential equation: (after the driving current turns off)

$$-\frac{\partial^2}{\partial t^2}\mathbf{E}(x,y,t) = \nabla \times \nabla \times \mathbf{E}(x,y,t) = H\mathbf{E}(x,y,t). \tag{21}$$

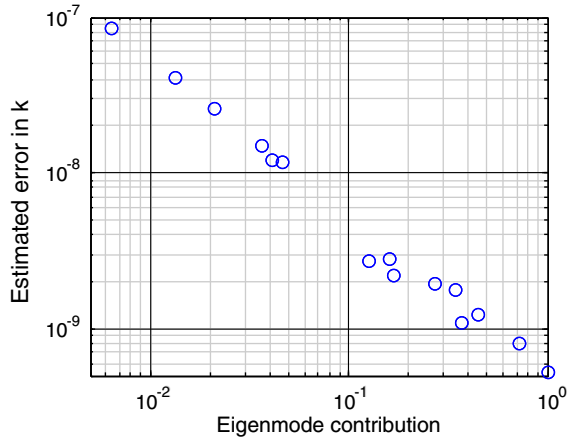

Fig. 2. Estimated error in k (one-half of Eq. (9)) vs. the normalized contribution of the corresponding eigenmode to the $\mathbf{s}_\ell$ from Eq. (10). Eigenmodes with small contributions are drowned out by those that loom larger in the $\mathbf{s}_\ell$.

When Eq. (19) has an eigenmode $(\mathbf{E},\mathbf{B})$ with eigenvalue k, its time-reversed state $(\mathbf{E},-\mathbf{B})$ is an eigenmode with eigenvalue $-k$. The very same $\mathbf{E}$ is an eigenmode of H in Eq. (21) with eigenvalue k^2. By isolating an eigen-subspace of Eq. (19) with $|k| \in [k_1,k_2]$ we have also isolated an eigen-subspace of H with $k^2 \in [k_1^2,k_2^2]$. Once the subspace of an operator is isolated, the techniques in Section 3 can be applied regardless of whether the eigenvalues are frequencies or squared frequencies, or have any connection at all with frequency. The eigenvalue-frequency connection was only important for isolating a mode with a certain eigenvalue by driving the simulation at a certain frequency.

By using this operator H above, we have chosen to work with eigenvalue k^2 instead of k; we can no longer distinguish positive and negative frequencies (so phase information is lost), but we have reduced the size of a state vector from $(\mathbf{E},\mathbf{B})$ to just $\mathbf{E}$.

We could equally well use both $\mathbf{E}$ and $\mathbf{B}$ and find eigenmodes of Eq. (19). This would yield phase information, and would be essential if energy were not conserved (if the frequency had an imaginary component). It would also double the size of each state vector and double the number of state vectors needed, requiring more storage (but not more computation time). However, it would increase the accuracy of the method, because the accuracy is limited by the ratio of maximum to desired eigenvalue (see Section 3.1), which is smaller when the eigenvalue is k than when the eigenvalue is k^2. In any case, with double precision this method is likely to yield greater accuracy than needed.

5.3. Constructing the $\mathbf{s}_\ell$ and $S_{p\ell}$

Expecting around 13 modes, we extracted $I = 5$ different field patterns (at times t_i during the time $\pi/\Delta\omega_{\text{avg}}$ after excitation) from each of $J = 4$ simulations, giving a safety margin to find up to $L = IJ = 20$ modes in case of unexpected modes. In other words, we saved to disk 20 state vectors

$$\mathbf{s}_\ell = E_z^{(j)}(t_i), \tag{22}$$

where $i = 1,\ldots,I$ and $j = 1,\ldots,J$ and $\ell = I(j-1)+i$; the components of $\mathbf{s}_\ell$ are the values of $E_z^{(j)}(t_i)$ at different points in space.

The total simulation time was approximately $4(2\times 7.4+\pi)/\Delta\omega_{\text{avg}} \approx 72/\Delta\omega_{\text{avg}}$; with narrow, square-windowed filtering, we would have required a simulation time (assuming states filtered around 15 regularly-spaced frequencies) of $4(15\times 2\pi)/\Delta\omega \approx 377/\Delta\omega_{\text{avg}}$.

Although we need all components of the $\mathbf{s}_\ell$ to compute the eigenmodes, we performed the linear algebra of Section 3 with just $P = 40$ components; that is, we randomly chose points (x_p,y_p) for $p = 1,\ldots,P$ to compute the $S_{p\ell}$ matrix

$$S_{p\ell} = s_{\ell,p} = E_z^{(j)}(x_p,y_p,t_i). \tag{23}$$

5.4. Constructing $R_{p\ell}$

To find the $\mathbf{r}_\ell$, as in Eq. (1), we did not apply H directly to the $\mathbf{s}_\ell$ (though we have done so with nearly identical results). Instead, we used the known time evolution of the state vectors to reconstruct the effect of H on them, using Eq. (21)

$$\begin{aligned} R_{p\ell} = r_{\ell,p} &= (H\mathbf{s}_\ell)_p = (HE^{(j)}(t_i))_z(x_p,y_p) \\ &= -\frac{E_z^{(j)}(x_p,y_p,t_i+\Delta t) - 2E_z^{(j)}(x_p,y_p,t_i) + E_z^{(j)}(x_p,y_p,t_i-\Delta t)}{\Delta t^2}, \end{aligned} \tag{24}$$

(using a standard leap-frog time-advance to discretize the $\mathrm{d}^2/\mathrm{d}t^2$ operator). This required storing the values of the s_p at $t_i-\Delta t, t_i$ and $t_i+\Delta t$.

5.5. Extracting eigenvalues and eigenmodes: results

In comparison to running the simulations, calculating the eigenmodes took a negligible time (not more than a few seconds). After finding the S and R matrices as above, we solved the eigenvalue problem $S^\dagger R\mathbf{a} = \lambda S^\dagger S\mathbf{a}$ as in Section 3, using $\alpha_{\text{cutoff}} = \|S^\dagger S\|\,\varepsilon_{\text{cutoff}}$, where $\|S^\dagger S\|$ is the maximum eigenvalue of $S^\dagger S$ and $\varepsilon_{\text{cutoff}} = 10^{-12}$.

Table 1
Exact eigenvalues and the relative errors in the calculated values, as well as the errors estimated by Eq. (9); here errors for k^2 were converted to errors for k by halving them

Mode (TM_{mn})	k_{mn} (m^{-1})	Relative error	Estimated error
TM37	23.9237706063	7.7×10^{-9}	2.5×10^{-8}
TM73	23.9239355894	2.3×10^{-10}	1.2×10^{-8}
TM56	24.5352071863	1.4×10^{-11}	8.1×10^{-10}
TM65	24.5352514256	4.1×10^{-10}	1.2×10^{-9}
TM18	25.3254543627	2.0×10^{-8}	8.6×10^{-8}
TM81	25.3256998232	9.9×10^{-9}	4.1×10^{-8}
TM47	25.3264357694	6.3×10^{-9}	1.2×10^{-8}
TM74	25.3265643392	4.6×10^{-9}	1.5×10^{-8}
TM28	25.9034160843	2.2×10^{-10}	1.9×10^{-9}
TM82	25.9036446381	6.8×10^{-10}	1.8×10^{-9}
TM66	26.6555857982	2.5×10^{-11}	5.2×10^{-10}
TM38	26.8390177976	6.5×10^{-10}	2.8×10^{-9}
TM83	26.8392199987	3.5×10^{-10}	1.1×10^{-9}
TM57	27.0230051542	8.1×10^{-10}	2.7×10^{-9}
TM75	27.0230927865	6.3×10^{-10}	2.2×10^{-9}

Fifteen of the resulting eigenvalues had estimated errors well below 10%. These eigenvalues k_{mn}^2 were real and corresponded to the actual TM_{mn} modes (no modes between k_1^2 and k_2^2 were missed). Table 1 shows the exact eigenvalues k_{mn} (using Eq (18)) and the relative errors of the calculated values, as well as the error estimated using Eq. (9); Fig. 1 shows that Eq. (9) provides a fairly accurate, if conservative, estimate of the error.

The least error (around 10^{-10} as shown in Table 1) is about as expected; with a maximum/minimum eigenvalue ratio around 10^4 and a machine precision of 10^{-16}, we expected relative errors above 10^{-12} (plus a bit more for errors of finite-precision numerics). However, only the eigenmodes that figure largely in the $\mathbf{s}_\ell$ have such low error; Fig. 2 shows that the error in each eigenvalue is inversely proportional to the contribution (see Eq. (10)) made by the corresponding eigenmode to the $\mathbf{s}_\ell$.

Table 2 shows the relative error in eigenvalues and eigenmodes (in the ℓ^2-norm) for selected modes. The baseline accuracy of the calculated eigenmodes (Fig. 3 shows the calculated TM_{73} mode) is similar, but the accuracy of individual modes is degraded to the extent that eigenvalues are close together – after all, exactly

Table 2
Relative errors in eigenvalues k_{mn}^2 and eigenmodes for selected modes, as well as the squared inner products between calculated and exact modes

Mode	(3,7)	(7,3)	(1,8)	(8,1)	(4,7)	(7,4)	(6,6)
k^2 err.	1.54×10^{-8}	4.51×10^{-10}	4.05×10^{-8}	1.98×10^{-8}	1.25×10^{-8}	9.16×10^{-9}	5.08×10^{-11}
Mode err.	2.77×10^{-4}	2.28×10^{-4}	2.39×10^{-3}	1.43×10^{-3}	6.37×10^{-4}	1.76×10^{-3}	1.90×10^{-8}
Squared inner products $\lvert\langle \mathbf{v}_m \rvert \mathbf{v}_{n,\mathrm{calc}} \rangle\rvert^2$							
TM37	1.00	5.22×10^{-8}	1.09×10^{-13}	2.79×10^{-14}	1.53×10^{-15}	3.67×10^{-15}	3.10×10^{-18}
TM73	7.67×10^{-8}	1.00	1.35×10^{-16}	2.05×10^{-16}	5.65×10^{-19}	2.56×10^{-17}	1.03×10^{-18}
TM18	8.12×10^{-15}	2.13×10^{-15}	1.00	1.05×10^{-6}	3.71×10^{-9}	6.18×10^{-9}	7.42×10^{-18}
TM81	8.60×10^{-15}	1.87×10^{-16}	3.39×10^{-6}	1.00	6.78×10^{-9}	7.82×10^{-9}	2.09×10^{-17}
TM47	2.99×10^{-15}	9.86×10^{-15}	1.89×10^{-6}	8.21×10^{-7}	1.00	3.08×10^{-6}	6.46×10^{-18}
TM74	1.42×10^{-15}	5.51×10^{-16}	4.07×10^{-7}	1.72×10^{-7}	3.95×10^{-7}	1.00	2.55×10^{-18}
TM66	2.55×10^{-14}	7.35×10^{-15}	6.88×10^{-13}	1.46×10^{-13}	9.53×10^{-15}	1.66×10^{-14}	1.00

The main error in TM37 is due to mixing of the nearly degenerate TM73; because it has no near-degeneracy, TM66 has a low error; because TM18, TM81, TM47 and TM74 are all nearly degenerate, they are mixed together at levels much greater than the eigenvalue error.

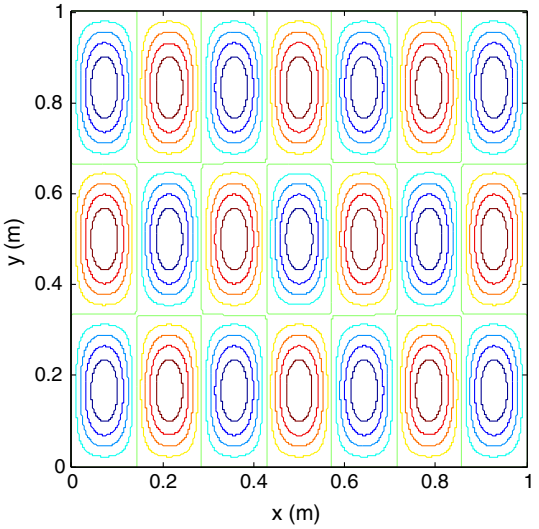

Fig. 3. Contours of E_z for the calculated TM_{73} mode.

degenerate eigenmodes cannot be distinguished. By looking at inner products of calculated and exact modes, one can see that most of the error in the TM_{37} mode is due to spurious amounts of the TM_{73} mode; however, the $TM_{37} \oplus TM_{73}$ subspace is found with an error around 10^{-7}. The TM_{66} mode is very accurate because no other modes have eigenvalues closer than 1%; whereas k_{28} and k_{82} differ by less than 20 ppm.

FDM can actually distinguish fairly closely-spaced eigenvalues, though with an accuracy that diminishes with the spacing. For our broad-excitation method, two modes separated by $\Delta\omega$ are best distinguished by comparing simulation states separated by a time $\sim\pi/\Delta\omega$ (or greater) so that the modes evolve to completely different phases. However, the modes still evolve to noticeably different phases after a time $0.01\pi/\Delta\omega$ – but the accuracy will suffer by a factor of about 100.

We showed that with $J = 4$ simulations we extracted all 15 eigenvalues with errors between 10^{-11} and 10^{-8}. If we use just $J = 2$ simulations, we can extract the same 15 modes with errors (in eigenvalue) between 10^{-10} and 10^{-7} (with the worst error being for the 4-fold near-degeneracy).

If we use just $J = 1$ simulation, only 14 modes are found (one of the 4-fold degenerate modes is not found) with errors from 10^{-10} (for the isolated TM_{66} mode) to 10^{-5}.

6. Conclusion

We have described and demonstrated a method to extract degenerate modes and frequencies from time-domain simulations using excitation to create filtered states for use with the filter-diagonalization method. This approach requires low-storage and is very easy to implement; in many cases no changes need be made to the time-domain code to achieve performance on par with dedicated frequency-domain codes (also: this method easily finds non-orthogonal eigenmodes as well as slowly-decaying modes, and can extract modes with eigenvalues in a specified range).

Although excitation can be used to create states filtered around regularly-spaced frequencies, we have demonstrated a new way of filtering – broad filtering – that can produce filtered states roughly five times faster than other FDM schemes.

We also suggested a way of estimating (quickly) the error of calculated eigenvalues, and demonstrated its agreement within an order of magnitude of the real error.

Although we focused on systems with oscillatory modes, the same method could be used even in the presence of some purely growing modes, as in ideal magnetohydrodynamics. Typically, when instability occurs, a few modes grow exponentially, while the rest of the modes oscillate. The fastest-growing modes naturally isolate themselves by outgrowing all other modes; the techniques of Section 3 can then be used to identify them.

Subsequently, these growing modes could be eliminated from the excitation to allow identification of other stable modes.

This method also works with complex frequencies; again, the difficulty lies in mode isolation. Since slowly-growing or -decaying oscillating modes have narrow resonances, they can be easily isolated with the methods described in this paper. Even broader resonances can be isolated as long as the number of other modes overlapping those resonances does not become too large.

Acknowledgments

We would like to acknowledge helpful discussions with Carl A. Bauer and David N. Smithe. To test the method described in this work, we used the simulation framework VORPAL [18], for which we would like to acknowledge support from the Offices of FES, HEP and NP of the Department of Energy, the SciDAC program, AFOSR, JTO, Office of the Secretary of Defense and the SBIR programs of the Department of Energy and Department of Defense. We also acknowledge assistance from the VORPAL team: G.I. Bell, D.L. Bruhwiler, R.S. Busby, J. Carlsson, D.A. Dimitrov, A. Hakim, P. Messmer, P. Mullowney, C. Nieter, K. Paul, S.W. Sides, N.D. Sizemore, D.N. Smithe, P.H. Stoltz, R. Trines, S.A. Veitzer, D.J. Wade-Stein, W.-L. Wang, N. Xiang and W. Ye.

Appendix A. Low-storage FDM: excitation vs. filter operators

For large simulations with small time steps, the Fourier transformation approach to filtering is impractical because it requires storing $\mathbf{s}(t)$ at every time step, and filtering every component of the state vector $\mathbf{s}(t)$, which could have a billion components.

To reduce storage, one operates on an initial state $\mathbf{s}$ with a frequency-domain filter that directly yields a filtered state $\mathbf{s}_\ell$ [15,16,19]. The frequency-domain filter must be applied anew for each $\mathbf{s}_\ell$, but only the final $\mathbf{s}_\ell$ need be stored to disk.

Ref. [15] applies a Gaussian filter operator

$$\mathbf{s}_\ell = \exp[-(H-\omega_\ell)^2/(2\sigma_\omega^2)]\mathbf{s}, \tag{A.1}$$

expanded in polynomials in H (which can be applied to $\mathbf{s}$), using Chebyshev polynomials.

The resulting $\mathbf{s}_\ell$ is in principle the same as if $\mathbf{s}(t)$ had been evolved for a time T and then been filtered via Fourier transformation using a Gaussian window $w(t) = \exp[-\sigma_\omega^2(t-T/2)^2/2]$.

The resulting $\mathbf{s}_\ell$ is also the same as if it had been produced by a Gaussian excitation $f(t) = w(t)\sin(\omega_\ell t)$; both excitation and the filter operator in [15] require the same amount of computation.

The computation required to filter a vector increases with the narrowness and height of the filter and the maximum eigenvalue $\omega_{\max}$ of H. We have empirically found that about $7\omega_{\max}/\sigma_\omega$ applications of H will accurately approximate the Gaussian filter down to about 10^{-12} below its peak (see [15] to verify this for $\sigma_\omega/\omega_{\max} = 0.01/\sqrt{2}$).

A Gaussian excitation requires similar computation time: to simulate the Gaussian tails down to 10^{-12} requires $\exp[-\sigma_\omega^2(T/2)^2/2] \approx 10^{-12}$ or $T > 2(7.4)/\sigma_\omega$. A simple algorithm generally applies H once each time step, which is limited to $\Delta t < 2/\omega_{\max}$ for stability (the Courant–Lewy–Friedrichs condition). Therefore, about $7.4\omega_{\max}/\sigma_\omega$ applications of H are needed.

Appendix B. Simulation time using narrow frequency filtering

When computing narrowly filtered state vectors, that is, filtering the $\mathbf{s}_\ell$ around regularly-spaced frequencies ω_ℓ using an excitation $f_\ell(t) = \Theta(t)\Theta(T-t)\sin\omega_\ell t$ (or equivalently, using square-windowed Fourier transformation), we believe that the simulation should be run for time $T > 2\pi/\Delta\omega_{\text{avg}}$, where $\Delta\omega_{\text{avg}}$ is the average frequency spacing of modes in the desired range.

Other authors have explored the minimum simulation time more carefully; we will try to interpret their conclusions with regard to this paper. Ref. [5] concludes that $T > 4\pi/\Delta\omega_{\text{avg}}$. However, that work does not use the time evolution of different points of a field pattern (i.e. $s_p(t)$ for different points p), but instead considers a sca-

lar-valued time series $s(t)$. Therefore, half the time is used to construct artificial field patterns, e.g. $s_p(t) = s(np + t)$; in the end, all $s_p(t)$ are known for a time $2\pi/\Delta\omega_{\text{avg}}$. Therefore, we believe Ref. [5] agrees with our conclusion that $T > 2\pi/\Delta\omega_{\text{avg}}$.

Ref. [13] also notes the factor of two difference in times. It goes on to claim that $T_{\text{req}} > \pi/\Delta\omega_{\text{avg}}$, but at one point suggests using a window of $\Theta(T^2 - t^2)$ so their simulations might run from $t = -T_{\text{req}}$ to $t = T_{\text{req}}$, which would agree with our conclusion that the total simulation time (for one filtered state) is $T > 2\pi/\Delta\omega_{\text{avg}}$.

From experience we know that there is no sharp dividing line between failure and success of this method. One can extract frequencies using $T = \pi/\Delta\omega_{\text{avg}}$, but using $T = 2\pi/\Delta\omega_{\text{avg}}$ will yield greater accuracy. While we have not thoroughly explored this issue, our limited experiments suggest that the fastest way to reach a given accuracy uses $T \approx 2\pi/\Delta\omega_{\text{avg}}$.

Appendix C. The advantage of the square window

When using narrow filtering with FDM, i.e. each $\mathbf{s}_\ell$ is filtered around different frequency ω_ℓ, a square-window filter works best. By square window, we mean using a sinusoidal excitation $f(t) = w(t)\sin(\omega_\ell t)$ with a square window $w(t) = \Theta(t)\Theta(T - t)$. Because of the abrupt start and stop, this excitation excites modes with frequencies far from ω_ℓ at significant amplitudes; it might therefore be surprising that it works better than other windows (such as a Gaussian $w(t)$) that provide better isolation from distant frequencies.

We call $w(t)$ a window function (or a weighting function) in analogy with windowed Fourier transforms. When using Fourier transformation to produce filtered states, one evolves a state $\mathbf{s}(t)$ from a random initial state $\mathbf{s}(0)$, and then computes

$$\mathbf{s}_\ell = \sum_{n=0}^{N-1} w(t_n)\mathbf{s}(t_n)\mathrm{e}^{-\mathrm{i}\omega_\ell t_n}. \tag{C.1}$$

This results in essentially the same filtered state as when exciting with $f(t) = w(t)\exp(\mathrm{i}\omega_\ell t)$.

Smoother window functions $w(t)$ are usually introduced for better isolation (see [14] on power spectra), and early FDM schemes used a Gaussian window to achieve the desired resolution [2,4]. However, Refs. [5,13] found the square window to be better, for reasons explained in [6], upon which we will now expand.

Briefly, when using a square window, any individual $\mathbf{s}_\ell$ is poorly isolated from frequencies far from ω_ℓ, meaning that $\mathbf{s}_\ell$ is a superposition of modes including modes with distant frequencies. However, there are linear combinations of nearby $\mathbf{s}_\ell$ that are very well-isolated from distant frequencies. The eigensolving in Section 3 automatically finds modes as linear combinations of the $\mathbf{s}_\ell$ that are well-isolated.

More carefully, we can compare the square window to a window $w(t)$. Filtering a signal $s(t)$ (ignoring temporal discretization for the moment) around ω_ℓ with a square window yields

$$\tilde{s}_{\text{sqr}}(\omega_\ell) \equiv \int_0^T s(t)\mathrm{e}^{-\mathrm{i}\omega_\ell t}\mathrm{d}t, \tag{C.2}$$

(this is the Fourier series coefficient of $s(t)$ for ω_ℓ). Filtering $s(t)$ around ω_ℓ, with a window $w(t)$, yields

$$\tilde{s}_w(\omega_\ell) \equiv \int_0^T s(t)w(t)\mathrm{e}^{-\mathrm{i}\omega_\ell t}\mathrm{d}t \propto \sum_{\ell'=-\infty}^{\infty} \tilde{w}_{\text{sqr}}(\omega_\ell - \omega_{\ell'})\tilde{s}_{\text{sqr}}(\omega_{\ell'}), \tag{C.3}$$

where $\omega_\ell = 2\pi\ell/T$ (since the Fourier series of a product is the convolution of the Fourier series of the factors). A good window $w(t)$ can prevent frequencies far from ω_ℓ from contributing to $\tilde{s}_w(\omega_\ell)$. (By considering the Fourier transform of a Gaussian $w(t)$ truncated to a duration T, it can be shown that frequencies more than $2\pi Q/T$ away from ω_ℓ can be suppressed by a factor $h \sim \exp[-\pi Q/2]$.)

While $\tilde{s}_{\text{sqr}}(\omega_\ell)$ contains significant contributions from signals in $s(t)$ at frequencies far from ω_ℓ, Eq. (C.3) shows that the linear combination of s_{sqr} values

$$\sum_{q=-Q}^{Q} \alpha_q \tilde{s}_{\text{sqr}}(\omega_{\ell+q}), \tag{C.4}$$

is well-isolated from modes far from ω_ℓ if Q is sufficiently large and the coefficients are chosen to be $\alpha_q = \tilde{w}_{\mathrm{sqr}}(\omega_q)$ for a good window $w(t)$.

Using square-window filtering can provide excellent isolation, but only if many states are calculated for many frequencies in a row. Even if only a few modes are desired, one must still calculate many filtered states (using a square window); we estimate from a limited number of tests that more than 15 filtered states must be found to achieve errors below 10^{-12}.

An important advantage when using many square-window filtered states is that the mode-finding algorithm (Section 3) will automatically choose linear combinations that achieve the best isolation – essentially choosing the best window.

Of course, when filtered states are found by Fourier transform, the Fourier transform yields filtered states for all possible ω_ℓ at once. When there is no cost for computing extra filtered states, the square-window filter is clearly the fastest option.

References

[1] T. Carrington Jr., Methods for calculating vibrational energy levels, Can. J. Chem. 82 (6) (2004) 900.
[2] D. Neuhauser, Bound state eigenfunctions from wave packets: time–energy resolution, J. Chem. Phys. 93 (4) (1990) 2611.
[3] D. Neuhauser, Circumventing the Heisenberg principle: a rigorous demonstration of filter-diagonalization on a LiCN model, J. Chem. Phys. 100 (7) (1994) 5076.
[4] M.R. Wall, D. Neuhauser, Extraction, through filter-diagonalization, of general quantum eigenvalues or classical normal mode frequencies from a small number of residues or a short-time segment of a signal. I. Theory and application to a quantum-dynamics model, J. Chem. Phys. 102 (20) (1995) 8011.
[5] V.A. Mandelshtam, H.S. Taylor, Harmonic inversion of time signals and its applications, J. Chem. Phys. 107 (17) (1997) 6756 (erratum);
V.A. Mandelshtam, H.S. Taylor, Harmonic inversion of time signals and its applications, J. Chem. Phys. 109 (10) (1998) 4128.
[6] V.A. Mandelshtam, On harmonic inversion of cross-correlation functions by the filter-diagonalization method, J. Theor. Comput. Chem. 2 (4) (2003) 497.
[7] J. Chen, V.A. Mandelshtam, A.J. Shaka, Regularization of the two-dimensional filter diagonalization method: FDM2K, J. Magn. Reson. 146 (2) (2000) 363.
[8] V.A. Mandelshtam, FDM: the filter diagonalization method for data processing in NMR experiments, Prog. Nucl. Mag. Res. Spectrosc. 38 (2) (2001) 159.
[9] J. Chen, A.A.D. Angelis, V.A. Mandelshtam, A.J. Shaka, Progress on the two-dimensional filter diagonalization method. An efficient doubling scheme for two-dimensional constant-time NMR, J. Magn. Reson. 162 (1) (2003) 74.
[10] D. Neuhauser, R. Baer, A two-grid time-dependent formalism for the Maxwell equation, J. Theor. Comput. Chem. 2 (4) (2003) 537.
[11] S.G. Johnson, J.D. Joannopoulos, Block-iterative frequency-domain methods for Maxwell's equations in a planewave basis, Opt. Exp. 8 (2001) 173.
[12] A. Vijay, R.E. Wyatt, Spectral filters in quantum mechanics: a measurement theory perspective, Phys. Rev. E 62 (3) (2000) 4351.
[13] Gĭrts Barinovs, G. Nyman, On the resolution of the filter diagonalization method, Chem. Phys. 281 (2002) 23.
[14] W.H. Press, S.A. Teukolsky, W.T. Vetterling, B.P. Plannery, Numerical Recipes in C++: The Art of Scientific Computing, second ed., Cambridge University Press, New York, 2002.
[15] R. Chen, H. Guo, A general and efficient filter-diagonalization method without time propagation, J. Chem. Phys. 105 (4) (1996) 1311.
[16] R. Santra, J. Breidbach, J. Zobeley, L.S. Cederbaum, Parallel filter diagonalization: a novel method to resolve quantum states in dense spectral regions, J. Chem. Phys. 112 (21) (2000) 9243.
[17] K.S. Yee, Numerical solution of initial boundary value problems involving Maxwell's equations in isotropic media, IEEE Trans. Antennas Propagat. 14 (3) (1966) 302.
[18] C. Nieter, J.R. Cary, VORPAL: a versatile plasma simulation code, J. Comput. Phys. 196 (2) (2004) 448.
[19] T.P. Grozdanov, V.A. Mandelshtam, H.S. Taylor, Recursion polynomial expansion of Green's function with absorbing boundary conditions: calculation of resonances of HCO by filter diagonalization, J. Chem. Phys. 103 (18) (1995) 7990.