FOURIER METHODS IN COMPUTATIONAL
FLUID AND FIELD DYNAMICS

Robert VICHNEVETSKY
Rutgers University
New Brunswick, New Jersey, 08903, U.S.A.

ABSTRACT

This paper is a review, with examples, of those areas where the theory of Fourier transforms has played a role in the development of computational methods in fluid and field dynamics. These may be separated into three development of computing algorithms using Fourier transforms and [fill the analysis of standard finite difference and finite element algorithms by Fourier methods.

Recent results in the analysis of spurious reflection phenomena at computational boundaries of fluid flow simulation, which involve the theory of wave propagation and the concept of group velocity are given.

1. INTRODUCTION

There are essentially two areas where the theory of Fourier transforms plays a role in the computation of flows and fields.

1. Fourier algorithms which rely on the property of Fourier series to carry out specific calculations.

2. Fourier analysis which rely on the theory as a tool for the analysis of error and stability properties of otherwise standard finite difference or finite element algorithms. Albeit theoretical, this second use of Fourier transforms seems, in a way, to have played a role more important than the first.

The relationship of trigonometric series with the hyperbolic partial differential equations of mathematical physics has been known since the 18th century. The first contributions to the theory of vibrating strings by Brook Taylor (1685-1731), Leopold Euler (1707-1783), Daniel Bernoulli (1700-1782), and Louis Lagrange (1736-1813) among others, do contain indeed many uses of this analytic tool. The name of Fourier became attached later on to trigonometric series, after he applied them to the solution of the heat equation which he had derived (Fourier, 1807).

2. THE FAST FOURIER TRANSFORM

An algorithm that is fundamental to the implementation of Fourier calculations is of course the Fast Fourier Transform (FFT). It uses the symmetry of trigonometric functions to reorganize the equations so as to minimize the computational effort. Whereas computing the discrete Fourier transform of a function represented by N discrete data points requires on the order of N^2 operations, when implemented in the straightforward way, organizing the same calculation as prescribed by the FFT requires only on the order of N log N operations (for N=1024, this represents a saving of more than 100 to 1).

The original description of the fast Fourier transform was given in a paper by Cooley and Tukey published in Mathematics of Computation in 1965, although the technique was not unknown in the field. For example, the basic idea of the method may be found on page 219 of C. Lanczos' Applied Analysis, published in 1956, and earlier results are described in Danielson and Lanczos (1942). Danielson and Lanczos, in turn, refer to Runge (1903, 1905) for the source of their method. Other pre-1965 users of "special techniques" which were in the same vein as the FFT are given in the book The Fast Fourier Transform by Brigham (1974). "Computational tricks" were not held in high esteem in the precomputer days, and attempts to give them visibility under a unified heading were few. The name of "numerical analyst" was coined soon after the advent of electronic computers. The number of people involved in doing scientific calculations grew dramatically, and many techniques that had not been well known came to the surface or were reinvented at that time. The "invention" of the fast Fourier transform in the 1960's is an example.

3. FOURIER METHODS IN COMPUTATIONAL
FLUID DYNAMICS

It is after fast, large scale computers had become available (in the early 1960's), and that the fast Fourier transform algorithm had been popularized (in the mid to late 1960's) that Fourier methods in fluid flow calculations made their appearance. We shall briefly describe their principal fields of application:
Computational fluid dynamics imply the approximation of both first and second order spatial derivatives. The equations which illustrate the relevant concepts are:

(i) Those of the motion of a non viscous fluid (often called Euler's equations; he first described them in 1752-1753), which may be written as:

\[
\frac{DU}{Dt} = \frac{-2U}{t} + U \cdot \nabla U = -\frac{1}{\rho} \nabla P
\]  

(1)

where \(U(x,t)\) is a vector and \(\nabla\) is the gradient operator. Solving these equations numerically requires that the advection term \(U \cdot \nabla U\) be approximated, and Fourier methods have been used mostly with the intent of improving the accuracy of this approximation.

(ii) Those of viscous fluid, illustrated by the Navier-Stokes equation for an incompressible flow, in primitive form:

\[
\frac{DU}{Dt} = -\frac{1}{\rho} \nabla P + \nabla \cdot \nabla U
\]  

(2)

where \(\nabla^2\) is the Laplacian operator and \(\nu\) is the dynamic viscosity of the fluid. An alternate formulation is the stream function vorticity form:

\[
\frac{D\zeta}{Dt} = \nu \nabla^2 \zeta
\]

\[
\nabla \cdot \zeta = \zeta = \begin{bmatrix} \nu \zeta_{x} \\ \nu \zeta_{y} \end{bmatrix}
\]

(3)

The numerical treatment of these equations requires that the Laplacian \(\nabla^2\) also be approximated. In many cases (in particular if the vorticity form (3) is used, or if (2) is integrated by an implicit time marching method), the solution process requires that a form of Poisson's equation

\[
\nabla^2 \zeta = \zeta(x,y)
\]

be solved at each time step. In this case, Fourier methods are often used to reduce the computing time in the solution of (4).

4. APPLICATIONS IN PLASMA PHYSICS

Questions of the stability of nuclear fusion plasma in the context of the development of controlled nuclear fusion energy devices are often investigated by direct simulation. In one type of approach, the "fluid" consisting of the moving charged particles is represented by a finite number of superparticles: on the order of \(10^4\) superparticles are used in realistic simulation (it is reported that physicists doing these kinds of simulation are sometimes called "particle pushers" by those doing the more conventional continuous model simulations). The particles are in motion in an exterior magnetic field and a self-generated electrical field. The latter is computed by solving Poisson's equation, at each time step (or every few time steps). Since many time steps are needed, it is vital, as with Navier-Stokes' equations, that the solution of Poisson's equation be obtained as economically as possible.

5. FOURIER METHODS FOR FIRST ORDER HYPERBOLIC EQUATIONS

Consider as a model of (4) the linear, scalar equations

\[
\frac{DU}{Dt} + \frac{\partial U}{\partial x} = \zeta(x,t)
\]

over \(x \in [0, L]\) with boundary conditions:

\(U(0,t) = U(L,t)\)

(5)

Periodic boundary conditions of this kind occur for instance in mathematical models of the global atmosphere and Fourier methods have indeed been developed and used to a large extent by geophysical fluid dynamics and meteorologists.

\(U(x,t)\) may be expressed in Fourier series form as:

\[
U(x,t) = \frac{1}{L} \sum_{k=-\infty}^{\infty} \hat{U}_k e^{ik\pi x/L}
\]

(6)

where

\[
\hat{U}_k(t) = \frac{1}{L} \int_{0}^{L} U(x,t) e^{-ik\pi x} dx
\]

(7)

The form of \(U(x,t)\) which solves (5) is then easily obtained. Since

\[
\frac{\partial U}{\partial x} = \frac{\partial}{\partial x} \sum_{k=-\infty}^{\infty} \hat{U}_k e^{ik\pi x/L}
\]

(8)

it follows that each \(\hat{U}_k(t)\) satisfies

\[
\frac{d \hat{U}_k}{dt} + c \frac{k \pi}{L} \hat{U}_k = \hat{\zeta}_k
\]

(9)

where \(\{\hat{\zeta}_k\}\) is the Fourier transform of \(\zeta(x,t)\).

It is Daniel Bernouilli who, with his 1783 paper, may be credited with the invention of the idea of the separation of the dynamics of a string into sinusoidal modes which do not interact with one another. Once more, history had its word to say, and the method of separation of variables is generally called the method of Fourier, who used it more than half a century later.

Approximations are obtained when \(k\) is restricted in (7) to within a finite range of values, say (-N,N), i.e., \(U(x,t)\) is approximated by a truncated Fourier series. Using this concept leads to several types of feasible approximations, such as:

(i) truncated Fourier series methods in which time integration is done in Fourier space and all calculations
take place in the frequency domain. This is restricted to linear problems.

(iii) collocation methods in which time integration is done by expressing the solution in a finite number of discrete points in physical space. Collocation points, and trigonometric interpolation is used solely with the intent of approximating the spatial partial derivatives in these points.

(iii) collocation methods in which time integration is done in Fourier space, but transformation is not done at each time step to and from physical space, where the non-linear terms are being computed.

Of these, only (ii) and (iii) have general applicability to a variety of problems in fluid dynamics, since they can be applied to non-linear equations.

6. FOURIER METHODS FOR SECOND ORDER EQUATIONS

Special Poisson's equation solvers which use Fourier transforms have been developed in the mid to late 1960's. Their first use was in the simulation of nuclear plasmas, and they have been applied to fluid dynamics later on. Fast Poisson solvers were initially restricted to simple rectangular regions. Reference to subsequent work toward the application of Poisson solvers to less regular domains may be found for instance in Prokuzewski - 1982; see also Vichnevetsky - 1981-a). Consider for instance the solution of (4) in a square with the simple boundary conditions \( \nabla^2 U = 0 \) on \( \partial \Omega \). The simplest method uses a double Fourier series:

\[
U(x,y) = \sum_{p,q} \hat{U}_{p,q} \sin \left( \frac{2\pi px}{L_0} \right) \sin \left( \frac{2\pi qy}{L_0} \right)
\]  

(12)

which is a solution of \( (4) \) when the coefficients \( \{ \hat{U}_{p,q} \} \) are:

\[
\hat{U}_{p,q} = \frac{1}{L_0^2} \int_{-L_0/2}^{L_0/2} \int_{-L_0/2}^{L_0/2} U(x,y) \sin \left( \frac{2\pi px}{L_0} \right) \sin \left( \frac{2\pi qy}{L_0} \right) dx dy
\]

(13)

and \( \{ \hat{f}_{p,q} \} \) is the Fourier transform of \( f(x,y) \).

There is also the popular Hockney method which consists in using Fourier transforms in \( x \) and finite differences in \( y \). The resulting system of tridiagonal equations is then solved by recursive cyclic reduction.

Original ideas and analyses of the efficiency of the Fast Poisson solvers with Fourier transforms may be found in Hockney (1965), Gentleman and Sande (1966), Buneman (1969), Colony and Reynolds (1970), and Le Bail (1972).

7. COMPARISONS

Fourier methods achieve one thing that finite difference methods generally do not: they approximate the partial derivatives with the maximum accuracy which is permitted by a representation of the variables in a finite number of discrete points. Essentially, the derivatives are approximated by the analytic derivative of the trigonometric interpolant between all these points.

Consider the following error analysis:

A numerical solution of (5) consisting of \( 2N \) discrete values in physical space may be expressed in Fourier series forms such as (7) with \( k = -N \ldots N \). The approximation of the advective term \( \partial \nabla \times \partial \nabla \) by standard three point finite differences then results in a Fourier series representation of the advective term by:

\[
\frac{1}{\varepsilon} \sum_{k \neq 0} c_k^a \frac{i \varepsilon}{\varepsilon} \nabla_k e^{i k x n / l} = \sum_{k \neq 0} c_k^a \frac{i \varepsilon}{\varepsilon} \nabla_k e^{i k x n / l}
\]

(12):

where \( c_k^a \) the phase velocity of the \( k \)th Fourier component is given by (see section 9 below):

\[
c_k^a = \frac{\omega_k n h / \varepsilon}{(2 \pi \kappa h / \varepsilon)}
\]

(14)
The difference $C^E - C$ measures the error in the approximation of $C \partial U / \partial x$. By contrast, Fourier methods approximate this term with $(\mathcal{F})$ where the correct velocity $C$ is preserved in all the Fourier components which are included in the computation, and the only source of error is that which results from the elimination or aliasing of the higher wave number components.

Comparisons of the efficiency of Fourier methods with that of standard higher order finite difference approximations have been carried out in some detail in those cases where high accuracy is sought (as in long term atmospheric simulations). These may be found in Oszag (1971), Kreiss and Oliger (1972) and Fornberg (1975).

There are however, cases in the simulation of engineering systems where a lower order of accuracy is sufficient. The possible advantage of Fourier methods Assesses the fact that a very few number of Fourier components may have the potential of giving a realistic approximation, while a very few number of discrete points in physical space may not have that ability. See for instance, discussions about this in Clymer (1960).

Applications of this concept are found in recent work in the simulation and control of what engineers refer to as distributed parameter systems. In fact, these "modal methods" has been known for a long time by nuclear physicists working in the area of the control of nuclear reactors. Not only do they consider sinusoidal or "Fourier" modes, but also other modes which have spatial shapes given by the eigenfunctions of an associated Sturm-Liouville problem — see for instance Kaplan et al. (1965) and Stacey (1967).

The limitation of Fourier methods lies of course in the fact that they prefer regular domains and simple boundary conditions, and that their efficiency fades away when those requisites are not met.

SIMULATION OF ATMOSPHERIC TURBULENCE

A typical application is in the simulation of large scale atmospheric turbulence. An example is illustrated in Figure 3. These results give the stream function and temperature distribution obtained as part of a simulation of the atmosphere in the northern hemisphere by Pinkis and Kowalski (1983), using a Fourier method that was originally given by Oszag (1976). The numerical solution follows (iii) above, namely:

- All the problem's variables are expressed in Fourier space.
- They are transformed (by an FFT) to physical space at each time step for the calculation, point by point, of the non-linear terms.
- Non linear terms are re-transformed to Fourier space by an FFT.

Marching to the next time step is done in Fourier space — a combination of leap-frog and Crank-Nicolson methods was used.

![Figure 3 - Numerical simulation of atmospheric turbulence in the Northern hemisphere obtained with a spectral method (Pinkis and Kowalski, 1983).](image)

9. FOURIER ANALYSIS

Fourier analysis of the accuracy and stability properties of numerical algorithms may be achieved by observing the time evolution of sinusoidal (sometimes called "trivial") solutions. This is the engineers' and physicists' approach. A mathematical refinement consists in observing instead the time evolution of the Fourier transform of numerical solutions. The mathematics are almost identical, but the second viewpoint gives a foundation for more detailed analyses, such as the calculation of energy norms of the error.

In both cases, one assumes that the equations may be locally linearized. The simple scalar equation:

$$\frac{3U}{\partial t} + C \frac{3U}{\partial x} = 0 \quad (\text{35})$$

is used as a model where $C$ is to assume the value of the characteristic velocities of the system of equations being integrated. In the case of compressible gas dynamics, these are the particle velocity, and the sound
velocities upstream and downstream of the flow, respectively.

The underlying assumption in these analyses is that the models are meant to represent the local behaviour of small perturbations of numerical solutions. It may appear as a paradox (but it is not) that such local analyses are modeled by sinusoidal solutions which must be defined on the entire real axis to allow Fourier transform theory to apply with its full analytic power.

Semi-Discretizations

For illustration we consider typical semi-discretizations such as the explicit (finite-difference)

\[
\frac{d U_n}{dt} = - C \left( \frac{U_{n+1} - U_{n-1}}{2 h} \right) \quad (16 a)
\]

and implicit (linear finite element/Galerkin):

\[
\frac{1}{a} \left( \frac{d U_n}{dt} + \frac{d U_{n+1}}{dt} \right) - C \left( \frac{U_{n+1} - U_{n-1}}{2 h} \right) = 0 \quad (16 b)
\]

which may both be expressed in operator notations as:

\[
\frac{d}{dt} \mathbf{U} = \mathbf{A} \cdot \mathbf{U} \quad (17)
\]

where \( \mathbf{A} \) is an appropriately defined discrete (or Toeplitz) operator. Fully discrete algorithms are then obtained by applying to these some numerical time-marching procedure. But a first type of error analysis consists in assuming that time-marching errors may be neglected. The error may then be analyzed by defining the discrete Fourier transform of \( \{ U_n(t) \} \)

\[
\hat{U}(\omega t) = \sum_n U_n e^{-i \omega n t} \quad (18)
\]

Taking the Fourier transform of (17) then results in

\[
\frac{d}{dt} \hat{U}(\omega t) = \mathbf{A} \hat{U}(\omega t) \quad (19)
\]

where

\[
\hat{A}(\omega) = \mathbf{A} e^{i \omega n t} / e^{i \omega n t} \quad (20)
\]

is the spectral function of the operator \( \mathbf{A} \). Comparing this with the exact

\[
\frac{d}{dt} \hat{U}(\omega t) = - i \omega \hat{U}(\omega t) \quad (21)
\]

defines

\[ C(\omega) = - i \text{Im} \hat{A}(\omega) / \omega \quad (22) \]

as the phase velocity at which numerical Fourier components of spatial frequency, \( \omega \), propagate, and

\[ R = \text{Re} \hat{A}(\omega) \quad (23) \]

as the measure of an amplitude error. Semi-discretizations for which \( R = 0 \) are called conservative semi-discretizations. Both (14a) and

\[ C(\omega) = C = \frac{3 \sin(\omega h)}{\omega h (2 + \cos(\omega h))} \quad (24 a) \]

and finite elements:

\[ C(\omega) = C = \frac{1}{\omega h} \quad (24 b) \]

Fully Discrete Algorithms

Many time-marching methods for a system of equations written in the general form (17) may be expressed in operator notation as:

\[
U_n = Z(\Delta t A) U_n \quad (25)
\]

where \( Z \) (sometimes called the amplification operator) is a function of the operator \( \mathbf{A} \), or more precisely of the dimensionless grouping \( \Delta t A \). We have for example, for some of the well known time-marching methods:

Euler's Method:

\[ Z = 1 + \Delta t A \]

Crank-Nicolson Method:

\[ Z = \frac{1 + \Delta t A / 2}{1 - \Delta t A / 2} \]

Leapfrog Method:

\[ Z = \Delta t A \pm i \sqrt{1 + (\Delta t A)^2} \]

To carry out a Fourier analysis of these fully discrete approximations, one replaces, as in the semi-discrete case, \( U_n \) by its Fourier transform \( \hat{U} \).
whence (25) becomes
\[ \mathcal{F}(\omega) = z(\omega) \cdot \mathcal{F}(\omega) \]  
(26)

obtained by replacing the operator $\mathcal{A}$ by its spectral function $\mathcal{B}(\omega)$ in the expression of $z$. This was first done by von Neumann in his classical analysis of numerical stability (somewhat differently: he used sinusoidal trial solutions and inserted them without preliminaries in the full discretizations), and $z(\omega)$ is traditionally called the amplification factor.

But $z(\omega)$ contains information about accuracy as well: By comparison with (24), we have the ratio of Fourier transforms for an exact solution of (15):
\[ \mathcal{F}(\omega) = \frac{-i\omega \xi}{\mathcal{C}(\omega)/\Delta t} \]  
(27)

which allows one to define
\[ \mathcal{C}(\omega) = -i \frac{\Delta t}{\xi} \mathcal{F}(\omega) \]  
(28)
as the numerical phase velocity, and
\[ |\mathcal{C}(\omega)| = 1 \]  
(29)
as an amplitude error: When $|\mathcal{C}(\omega)| < 1$, there is a numerical or spurious damping. And $|\mathcal{C}(\omega)| > 1$ corresponds to numerical instability (von Neumann).

9. TRANSFER FUNCTIONS

Engineers are familiar with the use of transfer functions for the analysis of input/output properties of linear systems. Transfer functions are defined as a ratio of Fourier transforms. The same concept may be used to evaluate properties of semi-discrete approximations of hyperbolic equations. Consider the Fourier transforms in time:
\[ \mathcal{F}(x, \Omega) = \int U(x, t) e^{-i \Omega t} dt \]  
(30)

If $U(x, t)$ is an exact solution of the advection equation (15), then
\[ \mathcal{F}(x, \Omega) = \mathcal{F}(x, \Omega) e^{-i \Omega x/c} \]  
(31)

and for two adjacent mesh points:
\[ \mathcal{F}(x_{n+1}, \Omega) = e^{-i \Omega h/c} \mathcal{F}(x_n, \Omega) \]  
(32)

where $\mathcal{F}$ is the transfer function between those two points.

Semi-discretizations have the effect of replacing this exact transfer function by approximations. For instance, to the simple upwind approximation
\[ \frac{dU_n}{dt} = -c \frac{U_n - U_{n-1}}{h} \]  
(33)
corresponds the transfer function
\[ \frac{\partial \mathcal{U}}{\partial t} = \frac{1}{1 + i \Omega h/c} \]  
(34)

which we call "call transfer function" of the approximation. By comparison with (32) we find
\[ \mathcal{C}(\Omega) = \frac{-i \Omega h/c}{\sqrt{1 + i \Omega h/c}} \]  
(35)

for the numerical phase velocity and the difference
\[ (1 + i \Omega h/c)^{1/2} = 1 \]

is an amplitude error which may be associated with spurious damping or spurious diffusion.

11. DISPERSION AND GROUP VELOCITY

The concept of group velocity occurs in describing the propagation of wave packets in dispersive media (defined as media where sinusoidal waves of the form $\mathcal{F}(x - \mathcal{C}(\omega) t)$ may exist with a frequency dependent phase velocity $\mathcal{C}(\omega)$). A wave packet is then

\[ \frac{\partial^2 U}{\partial x^2} - c^2 \frac{\partial^2 U}{\partial t^2} = 0 \]  
(15)
defined as a single wave number sinusoidal function modulated by a smooth envelope \( \phi(x, t) \), such as:

\[
U = \phi(x, t) e^{i\omega(x - c^{-}t)}
\]

(37)

\[\text{Figure 6 - Wave packet}\]

It is then found that the envelope of the packet does not propagate in the dispersive medium at the phase velocity \( c \), but instead at the group velocity given by:

\[
\gamma(\omega) = \frac{d}{d\omega}(\omega c^*(\omega))
\]

(38)

Or, in terms of time Fourier transforms,

\[
\gamma(\Omega) = \frac{d}{d\Omega}(\Omega c^*(\Omega))
\]

(39)

Applying this to (24b), (24d), one finds that the group velocity for sinusoidal solutions of wavelength near \( 2\hbar \) is negative. This explains the presence of reflected perturbations when a smooth solution passes across a discontinuity or a boundary of the computational domain; since all numerical phase velocities are positive, a cursory observation of the Fourier series form of the numerical solution does not suggest that any part of it might propagate in the negative direction. But the group velocity analysis shows that reflected solutions may indeed exist in the form of wave packets of short wavelength numerical oscillations which propagate upstream of the flow after having been spurious generated.

The mathematics of propagation in dispersive media and many of the attending developments in Fourier analysis had been well established by the end of the nineteenth century. The concept of group velocity had been used conceptually by Hamilton (1839), was known to Rayleigh (Theory of Sound - 1877) and was fully investigated in the early part of this century with publications of Sommerfeld (1912, 1914) being prominent. An important class of dispersive media is that of crystals, which present periodic structures to the propagation of light. Of interest is to note the fact that numerical discretizations of hyperbolic equations have created new families of similarly periodic structures. It comes therefore as no surprise that applying the same analysis to those discretizations proves to be a very fruitful endeavor.

12. REFLECTION PHENOMENA

The appropriate tool for the analysis of spurious reflection phenomena in discretizations of hyperbolic equations is that given by time Fourier transforms. If we denote by \( \{\hat{U}_n(t)\} = \{U_n(t)\} e^{i\Omega t} \) the set of time Fourier transforms of a solution of (16a) or (16b), then an analytic treatment reveals the following (Vichnevetsky 1981 a, b, 1983 a):

\[\text{Figure 7 - Dispersion relation} \]

(frequency \( \Omega \) versus wave number \( \omega \))

To each \( \Omega \) corresponds two wave numbers \( \omega \) and \( \Omega_1 \) which correspond to one another in reflection phenomena. (From reference 25).

\[\text{Figure 8 - Group velocities} \]

To each frequency below the cutoff frequency corresponds a positive group velocity (solution of \( Q_1 \) type) and a negative group velocity (solution of \( Q_2 \) type).
Any solution may be decomposed into the sum of two fundamental types

\[ \{u_n(t)\} = \{p_n(t)\} + \{q_n(t)\} \]  

(42)

whose Fourier transforms satisfy:

\[ \hat{p}_n / \hat{p}_m = \tilde{E}_1(\Omega) \quad \hat{q}_n / \hat{p}_m = \tilde{E}_2(\Omega) \]  

(44)

where \( \tilde{E}_1 \) and \( \tilde{E}_2 \) are cell transfer functions of the semi-discretisation, and are given in Reference (56).

While solutions of the \( [p] \) type have positive phase and group velocities, those of \( [q] \) type have positive phase but negative group velocities; solutions of \( [p] \) type are convergent approximations of genuine solutions, but solutions of \( [q] \) type are entirely spurious. They carry energy upstream of the flow and are precisely the kind of extraneous solutions that are generated when spurious reflection occurs at boundaries of computational domains and at interfaces in numerical mesh refinement.

13. REFLECTION AT BOUNDARIES

When the simple advection equation (15) is approximated over a finite domain, any \( x = 0 \), the approximation at the downstream boundary is different from the approximation in the interior points. This creates spurious reflection. Consider, for instance, the simple upstream approximation in:

\[ \frac{\partial u_n}{\partial x} = -c \left( \frac{u_{n+1} - u_{n}}{h} \right) \]  

(43)

Observe that:

(i) any incident solution arriving at \( x = 0 \) is of \( [p] \) type, since it carries energy downstream.

(ii) the corresponding reflected solution must be of \( [q] \) type, since it carries energy upstream.

(iii) to a component of frequency \( \Omega \) in the incident \( [p] \) correspond a component at the same frequency \( \Omega \) in the reflected \( [q] \). To quantify the reflection, we take the Fourier transform of (45):

\[ \hat{u}_n(\Omega) = \hat{E}_1(\Omega) \hat{u}_{n+1} \]  

(44a)

which together with

\[ \hat{p}_n = \hat{E}_1 \hat{p}_m \quad \hat{q}_n = \hat{E}_2 \hat{q}_m \]  

(44b)

gives the reflection ratio:

\[ \gamma(\Omega) = \frac{\hat{q}_n}{\hat{p}_n} = \frac{-c + i\Omega h/c - \tilde{E}_2}{\tilde{E}_2} = 0 (\tilde{E}_2)^{-1} \]  

(43)

Better treatments of the boundary may be sought by replacing (43) with improved expressions aimed at making \( \gamma(\Omega) \) as small as possible. This has been a subject of considerable interest in recent years.

Less obvious is the question of spurious reflection at an upstream or inlet boundary. The usual numerical treatment consists in letting

\[ u_n(t) = U(0, t) \]  

(46)

i.e., letting the numerical value be equal to the imposed value. But this creates reflection when a spurious solution (of \( [p] \) type) arrives at the boundary. An improved treatment should attempt to absorb those spurious solutions instead of reflecting them. Appropriate non-reflecting inlet boundaries for finite difference discretizations are given in Vichnevetsky, Scibba and Pak (1981).

14. REFLECTIONS IN MESH REFINEMENT

Mesh refinement is commonly used when a greater accuracy is sought in some subdomain of the problem. A typical example is that of grids used in numerical weather prediction. Whereas coarse grids may be used over the oceans, a fine grid is desirable over land where the predictions count. A simple model of mesh refinement is illustrated in Figure 6. The mathematics for the description of reflection at the interface are similar to (44), leading to the reflection ratio for the spurious solution reflected toward \( x < 0 \):

\[ \gamma(\Omega) = \frac{1 - (\Omega h/c)^2}{1 - (\Omega / c)^2} = \frac{1}{1 - (\Omega / c)^2} \]  

(44a)

\[ \gamma(\Omega) = \frac{-c + i\Omega h/c - \tilde{E}_2}{\tilde{E}_2} = 0 (\tilde{E}_2)^{-1} \]  

(43)

Better treatments of the boundary may be sought by replacing (43) with improved expressions aimed at making \( \gamma(\Omega) \) as small as possible. This has been a subject of considerable interest in recent years.

\[ \text{Figure 9} \quad \text{Reflection of a smooth function at a mesh crossing interface. The reflected solution is a wave packet of wave length near } 2b. \]
15. NUMERICAL WAVE PROPAGATION IN 2 DIMENSIONS - ANISOTROPY

The wave equation in two space dimensions
\[ \frac{\partial^2 U}{\partial t^2} = c^2 \left( \frac{\partial^2 U}{\partial x^2} + \frac{\partial^2 U}{\partial y^2} \right) \]  
(47)

has propagation characteristics which are isotropic: a plane sinusoidal wave
\[ U = e^{i(\omega t + k_x x + k_y y)} \]  
(48)

propagates in the \( k_x \) direction at the phase velocity \( c \) which is independent of \( \omega \). By contrast, numerical simulations on a discrete grid introduce, in addition to dispersion (\( c^* \) depending on \( \omega \)) also anisotropy (\( c^* \) depending on \( \theta \)). Consider, for instance, the simple semi-discretization of (47) on a square grid

\[ \frac{d^2 U_{n,m}}{dt^2} = c^2 \left( \frac{U_{n+1,m} - 2U_{n,m} + U_{n-1,m}}{h^2} \right) \]  
(49)

If we insert the expression of a plane sinusoidal trial solution such as (48), we find, after integration
\[ U_{n,m,t} = \phi e^{i\omega(t - c h \sin \theta t)} \]  
(50)

where the numerical phase velocity is:
\[ c^*(t, \omega) = \frac{\omega}{\omega t} \left( 2 - \cos(\omega h \cos \theta) - \cos(\omega h \sin \theta) \right) \]  
(51)

(see also Birkhoff and Douglas (1975)).

Anisotropy properties of general explicit and implicit semi-discretizations of the wave equation on a square grid are given in Vichnevetsky and Bowles (1982). Similar results for finite element discretizations (bilinear elements on rectangles and linear elements on triangles) are given by Mulinos and Belytschko (1982). Anisotropy of phase velocity implies of course anisotropy of group velocity as well. These properties have been analysed by Trefethen (1982 ...) for semi- and full discretizations — figure below.

Figure 12 — Propagation of a wave packet in a discretization of the wave equation on a square grid.
\( \bigcirc \) is the ideal position of the packet and \( \bigboxdot \) is the position predicted by the group velocity analysis.

(Reprinted with permission)
16. ACKNOWLEDGMENTS

The assistance of Elissa C. Partser in the preparation of this paper is gratefully acknowledged.

17. BIBLIOGRAPHY AND REFERENCES


