which for (2.24) is $\lambda \leq 1$ [10] (we discuss this in more detail in the next paragraph). This ties in with the notion of *stability* of FD schemes (which applies to all schemes, implicit and explicit included). Roughly speaking, an unstable FD difference scheme is one that admits solutions that grow faster than an exponential of the form e^{kt}, where k is a constant [10]. Typically, such solutions do not satisfy the continuum PDEs, and hence stability implies convergence, and vice versa. For linear, time-dependent PDEs, a similar statement, known as the *Lax-Richtmyer equivalence theorem*, can be proven rigorously: a consistent finite difference scheme for a partial differential equation for which the initial value problem is well posed is convergent if and only if it is stable (see, for instance [10]). We cannot prove this theorem for the system of PDEs arising from Einstein's field equations, however, we will assume that a similar result *does* hold in our case. This implies that one of the keys to obtaining valid results from a numerical code is to eliminate the instabilities. There are numerous factors that could affect the stability of a simulation, including the particular finite difference stencils used, the set of variables that are solved for, applying the correct boundary and regularity conditions for these variables, etc. Therefore, the approach that we take to construct a stable code, is to begin by using techniques that have been proven to work for similar systems, and then attempt to solve any new problems that may arise.

We now return to the discussion of the CFL condition, which is of relevance to the stability of most hyperbolic systems. For the FD approximation (2.22) to the wave equation, the CFL condition can be derived by performing a Fourier analysis of the update scheme (2.24), and demanding that the single-step growth factor of an eigenmode $\propto e^{i(kx \pm \omega t)}$ be less than or equal to 1 for stability. Intuitively, one can see why the CFL condition should exist for an explicit scheme by the following argument. The characteristic speed of propagation of signals in the 1D wave equation (2.2) is $v = 1$.

Hence, in a numerical evolution, to correctly represent the flow of information to a grid point x_i at time t, grid points within a region of size *greater than* $[x_i - v\Delta t, x_i + v\Delta t] = [x_i - \Delta t, x_i + \Delta t]$ must be referenced at time $t - \Delta t$. The FD difference scheme in (2.24) references the points $[x_i - \Delta x, x_i, x_i + \Delta x]$; and therefore, to satisfy this causality requirement, we need $\Delta x >= \Delta t$, which is exactly the CFL condition.

2.4 Kreiss-Oliger Dissipation

A method that we often use to construct stable FD approximations of hyperbolic equations, is to explicitly add numerical *dissipation*. Dissipation is effectively a low-pass filter applied to the grid function, suppressing high-frequency components in the numerical solution. Here, high-frequency always refers to wavelengths on the order of the mesh spacing h. The reason why dissipation is effective, and usually necessary, is that most FD schemes cannot accurately propagate high-frequency components of the solution; moreover, it is the high-frequency components that tend to exhibit the fastest growth in an unstable scheme. The dissipation technique that we most often use is the Kreiss-Oliger method [11], whereby a term of the form

$$D_{ko}\hat{u}_i \equiv \frac{\epsilon}{16}\left(\hat{u}_{i-2} - 4\hat{u}_{i-1} + 6\hat{u}_i - 4\hat{u}_{i+1} + \hat{u}_{i+2}\right) \tag{2.25}$$

is added to a difference equation in an "appropriate" fashion. ϵ is a positive, adjustable parameter controlling the amount of dissipation added, and must be less that 1 for stability (as can be shown for the appropriately adjusted scheme (2.24) by Fourier analysis, or by a less rigorous argument that we will present shortly, applicable to a general FD scheme).

The Taylor expansion of (2.25) about x_i is

$$D_{ko}\hat{u}_i = \epsilon h^4 u'''' + O(h^5), \tag{2.26}$$

from which it is apparent that adding a term of the form (2.25) to a second-order-accurate FD approximation should not affect the convergence properties of the scheme. Considered as a filter, equation (2.25) is a convolution of the Kreiss-Oliger filter with the grid function $\hat{u}$. Fourier transforming to frequency space $k \equiv \xi/h$: $\hat{u}(x_i) \to \hat{u}(\xi)$, the convolution operator transforms to a multiplication, and (2.25) becomes

$$D_{ko}\hat{u}(\xi) = \epsilon \sin^4(\xi)\hat{u}(\xi). \tag{2.27}$$

ξ takes on values in the range $[-\pi, \pi]$, for the highest possible wave number k that can be represented on a mesh with spacing h is $\frac{\pi}{h}$ (the Nyquist limit). Written as is, equation (2.27) is a high-pass filter; i.e. the low-frequency components of $\hat{u}$ are almost completely suppressed after the convolution with D_{ko}. Hence, we must *subtract* $D_{ko}\hat{u}_i$ from $\hat{u}_i$ (with $\epsilon > 0$) in order to achieve the desired effect of reducing the high-frequency components of the solution. Furthermore, we can see that if $\epsilon > 1$, we will *over-compensate* for frequencies $\xi > \arcsin(\epsilon^{-4})$, i.e. we will just be reintroducing some of the high frequency components but with opposite sign, and possibly even amplifying them if ϵ is large enough.

In general, there are several possible ways to add Kreiss-Oliger dissipation to a given FD scheme. For example, one way to add it to (2.22) is to subtract the high-frequency part of the solution at the most retarded time level $\hat{u}_{n-1,i}$. Then (2.24) becomes

$$\begin{aligned} \hat{u}_i^{n+1} &= 2\hat{u}_i^n - \hat{u}_i^{n-1} + \lambda^2(\hat{u}_{i+1}^n - 2\hat{u}_i^n + \hat{u}_{i-1}^n) \\ &+ \frac{\epsilon}{16}\left(\hat{u}_{i-2}^{n-1} - 4\hat{u}_{i-1}^{n-1} + 6\hat{u}_i^{n-1} - 4\hat{u}_{i+1}^{n-1} + \hat{u}_{i+2}^{n-1}\right) \end{aligned} \tag{2.28}$$

2.5 Iterative Solution of Non-linear PDEs

The update scheme (2.28) for the wave equation is rather trivial, and can easily be solved exactly (i.e. to within machine precision) in an evolution code. However, this is not typical of the FDEs we will obtain from Einstein's equations; moreover, many of the FDEs we must solve are non-linear. The method we use to solve non-linear,