Class. Quantum Grav. 9 (1992) 721–750. Printed in the UK

A direct comparison of two codes in numerical relativity

M W Choptuik[†], Dalia S Goldwirth[‡] and Tsvi Piran[‡][§]

[†] Canadian Institute for Theoretical Astrophysics, 60 St George St, Toronto, ON, Canada M5S 1A1

[‡] Harvard-Smithsonian Center for Astrophysics, 60 Garden St, Cambridge, MA, 02138, USA

Received 28 January 1991, in final form 2 April 1991

Abstract. We discuss a detailed numerical comparison of the results of two codes which treat the same model problem in numerical relativity. The model consists of a single, massless scalar field minimally coupled to the gravitational field with a further restriction to spherical symmetry. The comparison was complicated by the fact that the codes were based on different formalisms, used different coordinate systems and employed different numerical solution techniques. After briefly reviewing the model and our basic solution methods we describe in detail the additional numerical analysis which enabled us to directly (event-by-event) compare our solutions. We also describe some new algorithms which use Richardson extrapolation to significantly increase the accuracy of one of the codes at a given resolution. Using the basic methodology of convergence testing, and with the aid of high-accuracy (better than 0.001%) numerical results (also generated using extrapolation techniques), we find clear evidence that both codes are convergent even in the regime where the field interactions are significantly non-linear and highly time-dependent. We suggest that techniques such as those described in this paper will be very useful for testing codes which solve more general problems in numerical relativity.

1. Introduction

As numerical relativists, one of our main concerns is the assessment of the reliability of the results which our codes generate. As our field has matured, attention has focused increasingly on this issue—witness the sentiments expressed by Centrella *et al* [3], who suggest that 'newly constructed computer codes should be published with extensive test-bed calculations *or not at all*'. As discussed in [3], such 'test-bed calculations' will generally fall into two classes according to whether the results being evaluated are compared to exact solutions or to solutions which themselves are produced numerically. There has been relatively little work in numerical relativity involving the latter type of test. Among other reasons, there simply have not yet been many cases of different codes which purport to generate approximations of the same solutions of Einstein's equations (spacetimes). This paper is concerned with just such a pair of codes and the results of our evaluation of their relative performance on identical initial data.

The codes we have constructed treat a model problem consisting of a single, massless scalar field, ϕ, which is minimally coupled to the general-relativistic gravitational field. Furthermore, we limit our attention to the case of spherical symmetry;

[§] Permanent address: Racah Institute of Physics, Hebrew University, Jerusalem 91904, Israel.

this makes the task of solving the resulting systems of Einstein/scalar field equations *much* less taxing than it would be in the general non-symmetric case. This is true, not only because the symmetry reduces enormously the complexity of the systems of equations we must deal with, but also because the computational resources (time and memory) which we require to compute a solution to some specified accuracy are many orders of magnitude less than would be needed for the generic case. However, we stress at the outset, and hope that this point will become clear in the following, that the basic methods we have used in making the comparsion, do *not* rely on the existence of a high degree of symmetry in the model. This observation, combined with the fact that we are able to quantitatively assess the accuracy of our numerical results in strong-field (non-linear) computations, suggests that the techniques we use should be useful for more general calculations.

Now, these techniques are far from revolutionary—our basic idea of trying to establish that both codes are approaching some unique continuum limit by examining their respective outputs as a function of discretization scale (resolution) is an obvious one. However, we *do* adopt a somewhat more rigorous approach to our numerical analysis than is the norm in our field. In terms of the general issue of code validation in numerical relativity, we feel the efficacy of such an approach to be self-evident—by definition we *must* be able to trust our error estimates. Moreover, and perhaps more importantly, through *detailed* study of the resolution dependence of our numerical computations, we have frequently been led to an improved algorithm (section 4), or the discovery of a deficiency in a program (section 6).

As described in more detail in the next two sections, the job of comparing the two codes (programs, algorithms) was complicated by the fact that the programs use distinct coordinate systems and, indeed, different basic approaches in their respective treatments of the field equations. One of the codes, which we refer to as CH [9, 10], uses a *characteristic* formulation of the problem (due to Christodoulou [6, 7]), where initial data are specified on some outgoing *null* hypersurface (characteristic surface, surface of constant retarded time). Using the equations of motion, these data can then be propagated to other null slices at later (or earlier) values of retarded time. The other program, which we call CA [4, 5], is based on a *Cauchy* ($3+1$, ADM [1]) formalism. CA's initial data are given on a *spacelike* hypersurface and are then evolved to other spacelike surfaces to the future (or past) using the $3+1$ equations. Therefore, in order to directly compare results from CH and CA, it was necessary to first perform a 'numerical coordinate transformation' on the results from one of the codes and this introduced the potential for additional numerical error having little to do with the intrinsic performances of CH and CA themselves. Thus, after briefly discussing the basic numerical algorithms used in the two codes (section 3), we consider in detail how we designed (section 4) and tested (section 5) the transformation algorithm. The actual results of the comparison are described in section 6 which is followed by some discussion in section 7. We use MTW [13] conventions in the following; in particular, we choose units such that $G = c = 1$. We also presume familiarity with basic notions in the numerical solution of time-dependent partial differential equations, described, for example, in the introductory chapter of [18], or the last chapter of [16].

2. Analytic formulations of the model

In this section, we briefly describe the two different coordinate systems used in this work and the resulting sets of equations which CH and CA must solve to evolve the scalar and gravitational fields. More detailed descriptions of the characteristic [9, 10] and Cauchy (ADM, 3 + 1) [4, 5] formulations may be found elsewhere. Note that we attempt to distinguish between characteristic and Cauchy quantities by using upper and lower case symbols, respectively.

2.1. The characteristic formulation

The characteristic code, CH, uses a coordinate system employed by Christodoulou in his extensive analytic studies of the model system [6]. The spherically symmetric, time-dependent metric is written as

$$\mathrm{d}S^2(R,U) = -G(R,U)\overline{G}(R,U)\,\mathrm{d}U^2 - 2G\,\mathrm{d}U\,\mathrm{d}R + R^2\,\mathrm{d}\Omega^2 \tag{2.1}$$

where $\mathrm{d}\Omega^2$ is, as usual, the metric on the 2-sphere of radius R. The radial coordinate, R, provides a direct measure of proper surface area (and, hence, will sometimes be described as an *areal* coordinate), while surfaces of constant U are outgoing null hypersurfaces which are parametrized by the proper time of a central ($R = 0$) observer. Instead of the scalar field ϕ itself, it proves convenient to write the field equations in terms of the quantity H, defined by

$$H(R,U) = \frac{\partial}{\partial R}(R\phi(R,U)). \tag{2.2}$$

Then ϕ is the mean value of H on 0 to R:

$$\phi(R,U) = \overline{H}(R,U) \equiv \frac{1}{R}\int_0^R H(\tilde{R},U)\,\mathrm{d}\tilde{R} \tag{2.3}$$

where the overbar denotes the mean value operation. With these definitions, Einstein's equations,

$$G_{\mu\nu} = 8\pi(\phi_{,\mu}\phi_{,\nu} - \tfrac{1}{2}g_{\mu\nu}\phi_{,\alpha}\phi^{,\alpha}) \tag{2.4}$$

yield

$$G = \exp\left(4\pi\int_0^R \frac{H-\overline{H}}{\tilde{R}}\,\mathrm{d}\tilde{R}\right) \tag{2.5}$$

and

$$\overline{G} = \frac{1}{R}\int_0^R G\,\mathrm{d}\tilde{R}. \tag{2.6}$$

Applying the method of characteristics [9], the equation of motion for the minimally coupled, massless scalar field,

$$\phi^{;\mu}{}_{;\mu} = 0 \tag{2.7}$$

may be written as a pair of coupled *ordinary* differential equations,

$$\frac{\mathrm{d}H}{\mathrm{d}U} = \frac{1}{2R}(G-\overline{G})(H-\overline{H}) \tag{2.8}$$

$$\frac{\mathrm{d}R}{\mathrm{d}U} = -\tfrac{1}{2}\overline{G}. \tag{2.9}$$

where the second equation defines the trajectories of *ingoing* null geodesics (characteristics). Expressions (2.5)–(2.6) and (2.8)–(2.9) constitute the basic set of equations solved by CH. To generate a particular solution of these equations, initial data, $\overline{H}(R,0) = \phi(R,0)$, are specified along some initial outgoing null geodesic and then the data are propagated to advanced (or retarded) values of the retarded time using discrete versions of the equations of motion. A useful diagnostic quantity is

$$M(R,U) = 2\pi \int_0^R \frac{\overline{G}}{G}(H - \overline{H})^2 \, \mathrm{d}\tilde{R} \tag{2.10}$$

which, in a region of vacuum, is the mass contained within a sphere of radius R at retarded time U.

2.2. *The Cauchy (3 + 1) formulation*

The Cauchy program, CA, is based on the $3+1$ (ADM) [1, 20] formalism, and the particular coordinate system we use might be regarded as a 'natural' extension of the usual Schwarzchild coordinates to the case of *time-dependent* spherically symmetric geometries. The metric is given by

$$\mathrm{d}s^2\,(r,t) = -\alpha^2(r,t)\,\mathrm{d}t^2 + a^2(r,t)\,\mathrm{d}r^2 + r^2\,\mathrm{d}\Omega^2. \tag{2.11}$$

Here, as with the characteristic coordinate system, the radial coordinate, r, measures proper surface area. In spherical symmetry, having chosen such radial coordinates, the time slicing can be fixed by demanding that the 3-metric be diagonal at all times. Equivalently, we may describe our choice of time coordinate in terms of a condition on the trace of the extrinsic curvature tensor. The choice made here is known in $3+1$ parlance as *polar* slicing [2]. In any case, we must (1) choose initial data to be compatible with the slicing choice and (2) choose α to satisfy a certain differential equation at each instant of time in order to preserve the slicing condition.

Again, it is useful to introduce auxiliary variables for the treatment of the scalar field, primarily to avoid the appearance of time derivatives of the *lapse function*, α, which can *not* be evaluated from evolution equations. Specifically, by defining

$$\Phi(r,t) \equiv \frac{\partial\phi}{\partial r} \qquad \Pi(r,t) \equiv \frac{a}{\alpha}\frac{\partial\phi}{\partial t} \tag{2.12}$$

we find that the following system of equations is sufficient to determine the time evolution of the model:

$$\frac{\mathrm{d}a}{\mathrm{d}r} = \frac{1}{2}\left(\frac{a - a^3}{r}\right) + 2\pi r(\Pi^2 + \Phi^2)a \tag{2.13}$$

$$\frac{\mathrm{d}\alpha}{\mathrm{d}r} = \frac{a}{r}\left(a - \frac{\mathrm{d}}{\mathrm{d}r}\left(\frac{r}{a}\right)\right)\alpha \tag{2.14}$$

$$\frac{\partial\Phi}{\partial t} = \frac{\partial}{\partial r}\left(\frac{\alpha}{a}\Pi\right) \tag{2.15}$$

$$\frac{\partial\Pi}{\partial t} = \frac{1}{r^2}\frac{\partial}{\partial r}\left(r^2\frac{\alpha}{a}\Phi\right) = 3\frac{\partial}{\partial(r^3)}\left(r^2\frac{\alpha}{a}\Phi\right). \tag{2.16}$$

We note that (2.13) is the Hamiltonian constraint, (2.14) is the *polar slicing condition* which guarantees that the time coordinate has the property described earlier and (2.15) and (2.16) are equivalent to the wave equation (2.7). From the $3+1$ perspective, the non-trivial Einstein equations missing from this set include: (1) a component of the momentum constraint, which may be regarded as an *algebraic* equation for the single non-vanishing extrinsic curvature component, $K^r{}_r$; and (2) evolution equations for a and $K^r{}_r$. The evolution equations are not used in CA; rather the code implements what is known as a *fully constrained* evolution [14, 15]. Our ability to construct such a scheme is a consequence both of our restriction to spherical symmetry as well as our particular choice of coordinates. Again, we can define a mass function, which, in a region of vacuum, measures the total mass within the sphere of radius r at a given time t:

$$m(r,t) = \int_0^r \frac{\mathrm{d}m}{\mathrm{d}\tilde{r}}\,\mathrm{d}\tilde{r} = 4\pi \int_0^r \tilde{r}^2 \rho\,\mathrm{d}\tilde{r} = 4\pi \int_0^r \tilde{r}^2 \frac{(\Phi^2+\Pi^2)}{2a^2}\,\mathrm{d}\tilde{r} \tag{2.17}$$

or, equivalently,

$$m(r,t) = \tfrac{1}{2} r(1-a^{-2}). \tag{2.18}$$

In the current formulation, the model is solved by specifying initial data, $\Phi(r,0)$, $\Pi(r,0)$ on the initial hypersurface, $t=0$, after which the evolution is determined from (2.13)–(2.16). We also have the following *regularity* conditions at $r=0$:

$$\frac{\partial a}{\partial r}(0,t) = \frac{\partial \alpha}{\partial r}(0,t) = \frac{\partial \Pi}{\partial r}(0,t) = \Phi(0,t) = 0. \tag{2.19}$$

Numerically, since the generic solution admitted by the model involves outgoing waves which eventually reach the outer edge of the computational domain, we must worry about outer boundary conditions on the scalar field variables. Effective numerical conditions based on the known behaviour (outgoing) of the field at sufficiently large radii are easy to formulate in this case and are described in detail in [4]. However, in the context of the computations described later, these conditions are somewhat irrelevant since very little of the scalar field reaches the outer edge of the numerical domain before the calculations are stopped.

3. Basic numerical algorithms

3.1. Notation—summary

In the remaining sections of this paper we use a notation which was designed to help us concisely and accurately describe the various numerical calculations performed in our study. The notation is defined in detail in figure 1(c) and the next subsection. At this point, however, the reader may simply wish to study figures 1(a) and 1(b) and the accompanying captions, then proceed to section 3.3, referring back to figure 1(c) and/or section 3.2 when the meaning of an expression is not clear.

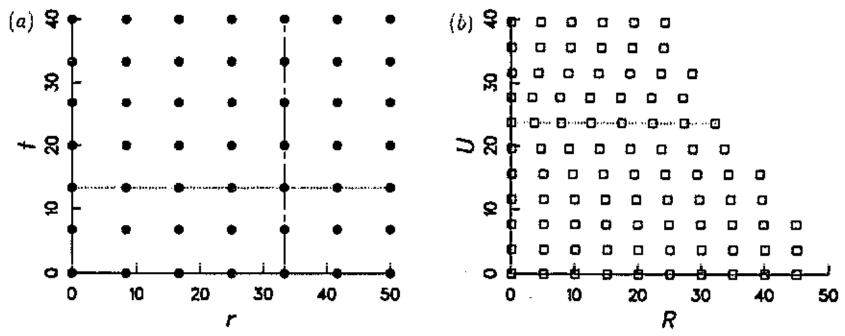


Figure 1. Grids, grid functions and definition of notation used in the text. (*a*) and (*b*) depict the grid structures used in typical CA and CH calculations respectively. The *2-grid* used by CA (filled circles) is *uniform* in both space and time, and may be described in the notation defined in (*c*) as $[\bar{\mathbf{r}}, \bar{\mathbf{t}}]$. Here, $\bar{\mathbf{r}}$ and $\bar{\mathbf{t}}$ denote regularly spaced sequences of radial and temporal coordinates which we refer to as *uniform 1-grids*. Then, for example, the approximation of ϕ computed by CA on the complete grid is the *2-grid function* $\phi^{\mathrm{CA}}[\bar{\mathbf{r}}, \bar{\mathbf{t}}]$. Values of this approximation along the dotted line constitute a *1-grid function* $\phi^{\mathrm{CA}}[\bar{\mathbf{r}}, \bar{t}^2]$; similarly, values along the chain-broken line comprise the 1-grid function $\phi^{\mathrm{CA}}[\bar{r}_4, \bar{\mathbf{t}}]$. The grid used by CH (open squares) is, in general, non-uniform both in space and retarded time. However, the grid may again be decomposed into 1-grids ($U = U^n = \mathtt{constant}$) which we denote by $\mathbf{R}(U^n)$. Collectively, these 1-grids form the *1-grid set* $\mathbf{R}(\mathbf{U})$, and the approximation to ϕ which CH computes is the 2-grid function, $\phi^{\mathrm{CH}}[\mathbf{R}(\mathbf{U}), \mathbf{U}]$. The values of this grid function along the dotted line make up the 1-grid function $\phi^{\mathrm{CH}}[\mathbf{R}(U^6), U^6]$. (*c*) provides a more detailed defintion of the notation as well as definitions of various operators used in the text. Refer to section 3.2 for additional information.

3.2. Notation—details

The primary output quantities from both CH and CA are (we hope) approximations to various continuum functions such as $\phi(r,t)$ or $H(R,U)$. For any specific computation, these approximations are defined at a finite set of events, namely the *grid points* or *mesh points* used in the computation. The basic complexity we must deal with arises from the fact that many distinct grids are involved in our comparison, which, for specific initial data, involves separate CH and CA computations at several different resolutions. Furthermore, additional grids are introduced in the procedure, described in section 4, which transforms CA output into numbers which can be directly compared with those produced by CH.

Due to the simplicity of our model problem, and the nature of our numerical techniques, the 'data structures' we need to represent all of these *grid functions* (functions defined on some collection of grid points) are quite simple. One such structure is the *1-grid*, which, as described in figure 1(*c*), is an ordered vector of coordinates. Generally, any 'syntactic construct' which includes a *single* normal-sized **boldface** symbol is a 1-grid. Thus $\mathbf{r}, \mathbf{U}_l, \bar{\mathbf{r}}_i$ and $\mathbf{R}(U^0)$ are all 1-grids but $\mathbf{R}_{\mathbf{r}}(\mathbf{U}_l)$ is not. When we need to reference *one* specific element of a 1-grid, we convert the boldface symbol to *italic* type and use the standard finite-difference subscript or superscript notation (r_k or $U_l{}^{n+1}$) for specific spatial or temporal coodinate values, respectively. Note that we have implicitly defined a 'last index' operator, #. Because we adopt the convention of labelling 1-grid elements from 0, the number of grid points in a 1-grid $\mathbf{r}$ is $\#\mathbf{r} + 1$. We have also introduced a 'mesh-spacing' operator, Δ, whose operation is only defined on *uniform* 1-grids. The operator MakeU1G[$\cdots$] generates uniform 1-grids as defined in the figure.

(*c*) **1-grid** (ordered 1-vector): $\mathbf{r}, \mathbf{t}, \mathbf{U}, \mathbf{U}_l, \mathbf{R}(U^n), \ldots$

Spatial 1-grid: $\mathbf{r} \equiv [r_0, r_1, \ldots, r_{\#\mathbf{r}}]$ where $r_0 < r_1 < \cdots < r_{\#\mathbf{r}}$

Temporal 1-grid: $\mathbf{t} \equiv [t^0, t^1, \ldots, t^{\#\mathbf{t}}]$ where $t^0 < t^1 < \cdots < t^{\#\mathbf{t}}$

Uniform 1-grid: $\bar{\mathbf{r}}, \overline{\mathbf{R}}_i, \ldots$

$$\bar{\mathbf{r}} \equiv [\bar{r}_0, \bar{r}_1, \ldots, \bar{r}_{\#\bar{\mathbf{r}}}] \text{ where } \bar{r}_1 - \bar{r}_0 = \bar{r}_2 - \bar{r}_1 = \ldots = \bar{r}_{\#\bar{\mathbf{r}}} - \bar{r}_{\#\bar{\mathbf{r}}-1} \equiv \Delta\bar{\mathbf{r}}$$

Uniform 1-grid creator: $\mathrm{MakeU1G}[\cdots]$

$$\bar{\mathbf{r}} = \mathrm{MakeU1G}[\bar{r}_0, \Delta\bar{\mathbf{r}}, \#\bar{\mathbf{r}}]$$

1-grid set: $\mathbf{R}(\mathbf{U}), \mathbf{R}(\mathbf{U}_l), \ldots$

$$\mathbf{R}(\mathbf{U}) \equiv [\mathbf{R}_0(U^0), \mathbf{R}_1(U^1), \ldots, \mathbf{R}_{\#\mathbf{U}}(U^{\#\mathbf{U}})] \quad \text{for 1-grids} \quad \mathbf{R}_i(U^i)$$

2-grid function: $\phi^{\mathrm{CA}}[\bar{\mathbf{r}}, \bar{\mathbf{t}}], H^{\mathrm{CH}}[\mathbf{R}(\mathbf{U}), \mathbf{U}], \phi^{\mathrm{CA}}_{ll'}[\mathbf{R}_r(\mathbf{U}_l), \mathbf{U}_l], \ldots$

$$\begin{aligned}
\phi[\mathbf{r}, \mathbf{t}] &\equiv [\phi[\mathbf{r}, t^0], \cdots, \phi[\mathbf{r}, t^{\#\mathbf{t}}]] \\
&= [[\phi[r_0, t^0], \cdots, \phi[r_{\#\mathbf{r}}, t^0]], \cdots, [\phi[r_0, t^{\#\mathbf{t}}], \cdots, \phi[r_{\#\mathbf{r}}, t^{\#\mathbf{t}}]]] \\
\phi[\mathbf{R}(\mathbf{U}), \mathbf{U}] &\equiv [\phi[\mathbf{R}(U^0), U^0], \cdots, \phi[\mathbf{R}(U^{\#\mathbf{U}}), U^{\#\mathbf{U}}]] \\
&= [[\phi[R_0(U^0), U^0], \cdots, \phi[R_{\#\mathbf{R}(U^0)}(U^0), U^0]], \cdots, \\
&\quad [\phi[R_0(U^{\#\mathbf{U}}), U^{\#\mathbf{U}}], \cdots, \phi[R_{\#\mathbf{R}(U^{\#\mathbf{U}})}(U^{\#\mathbf{U}}), U^{\#\mathbf{U}}]]]
\end{aligned}$$

Interpolation operator: pth-order spatial interpolation.

$$\phi[(\mathbf{U}), \mathbf{U}] := \mathtt{Interpolate}[\phi[\mathbf{R}(\mathbf{U}), \mathbf{U}], (\mathbf{U}), p]$$

Vector (1-grid function) norms: l_∞, l_1 and l_2 norms.

$$\|\phi[\mathbf{R}]\| \equiv \max_{0 \leqslant j \leqslant \#\mathbf{R}} |\phi[R_j]| \qquad \|\phi[\mathbf{R}]\|_p \equiv \left(\#\mathbf{R}^{-1} \sum_{j=0}^{\#\mathbf{R}} |\phi[R_j]|^p \right)^{1/p} \quad \text{for } p = 1, 2$$

Figure 1. (Continued)

A *1-grid set*, such as $\mathbf{R}(\mathbf{U}_l)$, is a set of 1-grids, one for each element of the associated 1-grid ($\mathbf{U}_l$ in this case). *2-grid functions* (or simply *grid functions*) have the structure 'vector of vectors' and are recognizable from the fact that they are *indexed* using the construct $[\ldots, \ldots]$. As described in the figure, there are two basic types of 2-grid functions. The first defines a rectangular array of values, while the second defines a generalized array which has the same 'shape' as the 1-grid set which appears in the first indexing slot. This more general second type allows us to deal with functions defined on grids where the spatial structure is time-varying (either in the specific spatial coordinates used or the number of such coordinates). When one of the indexing slots of a grid function is occupied by a scalar, such as r_k or $U_l{}^{n+1}$, the resulting object is to be identified with the appropriate *section* ('row' or 'column') of the grid function. We will also frequently work with 1-grid functions, which are simply vectors of values defined on some 1-grid—sections of 2-grid functions are examples of 1-grid functions. We note that we index by coordinate value, as we would in the continuum, rather than by specific integer values, as would normally be done in a computer program. Grid functions may be manipulated arithmetically as single entities, provided that their 'shapes' match. For example, $f[\mathbf{x},\mathbf{y}] + g[\mathbf{x},\mathbf{y}]$ is a grid function whose elements are the sums of the corresponding elements of f and g. Similarly, an '=', ':=' or '≡' sign applies on an element-wise basis in expressions where 1-grids and/or 1-grid sets appear as 'free indices' on both sides of the sign (see equations (4.7), (4.11) and (6.2) for some examples).

We also make extensive use of a (spatial) interpolation operator, Interpolate[$\cdots$] which takes a grid function, a 1-grid set and an interpolation order, p, and returns another grid function which is computed using pth-order polynomial interpolation of the original grid function to the coordinates of the 1-grid set. Finally, we define discrete l_1, l_2 and l_∞ norms for grid functions in a standard fashion.

3.3. *The characteristic code,* CH

The numerical algorithm employed by CH is described in detail in [9]. Typically, freely specifiable initial data, $\overline{H}(R,0) \equiv \phi(R,0)$, given as some closed-form function of R, are evaluated on a uniform radial grid, $\overline{\mathbf{R}}^0 \equiv \mathrm{MakeU1G}[\overline{R}_{\mathrm{min}}, \Delta\overline{\mathbf{R}}^0, \#\overline{\mathbf{R}}^0]$, along the initial outgoing null ray, $U = 0$. (The operator MakeU1G[$\cdots$] makes a *uniform* 1-grid—see figure 1(*c*).) Equations (2.8) and (2.9) then yield a coupled set of $2(\#\overline{\mathbf{R}}^0 + 1)$ ordinary differential equations (ODEs) which can be solved numerically using standard techniques and/or software packages. Matters are complicated by the fact that some of the 'coefficients' in these ODEs, namely G, $\overline{G}$ and $\overline{H}$, are defined implicitly through the integrals (2.5) and (2.6). As described in [9], these integrals are approximated using a Simpson-like scheme for unevenly spaced abscissa. For any particular intitial data, CH can advance the solution to essentially arbitrary values of U and it will be useful to view the output from CH as being defined at a set of retarded times constituting a 1-grid $\mathbf{U}$—this 1-grid can be regarded as an input parameter to CH. Then the basic outputs from CH are the grid functions $H^{\mathrm{CH}}[\mathbf{R}(\mathbf{U}),\mathbf{U}]$ and $\phi^{\mathrm{CH}}[\mathbf{R}(\mathbf{U}),\mathbf{U}]$. Note that the output grid structure is non-uniform; apart from the initial grid $\mathbf{R}(U^0) = \overline{\mathbf{R}}^0$, the various radial grids, $\mathbf{R}(U^1), \ldots, \mathbf{R}(U^{\#\mathbf{U}})$, will generally have non-uniform mesh spacings and varying numbers of mesh points (see figure 1(*b*)). Provided that a good ODE integrator with a sufficiently stringent error tolerance is used, the dominant source of truncation error in a CH calculation is expected to arise from the numerical treatment of (2.5) and (2.6). Naively, the scheme is expected to be $\mathrm{O}(h^4)$, provided that $R_j(U^n) - R_{j-1}(U^n) = \mathrm{O}(h)$ for all valid j and n. Thus, the expectation is that the scheme should exhibit, at best, fourth-order convergence to the continuum solution.

3.4. *The Cauchy (3 + 1) code,* CA, *and expandability*

A CA computation is performed on the finite domain $[0, r_{\mathrm{max}}] \times [0, t_{\mathrm{max}}]$ in the (r,t) plane, using *two* uniform radial grids $\bar{\mathbf{r}}'$ and $\bar{\mathbf{r}}$ and one uniform temporal grid, $\bar{\mathbf{t}}$:

$$\bar{\mathbf{r}}' := \mathrm{MakeU1G}\left[0, \Delta\bar{\mathbf{r}}, \frac{r_{\mathrm{max}}}{\Delta\bar{\mathbf{r}}}\right]$$

$$\bar{\mathbf{r}} := \mathrm{MakeU1G}[-\tfrac{1}{2}\Delta\bar{\mathbf{r}}, \Delta\bar{\mathbf{r}}, \#\bar{\mathbf{r}}' + 1]$$

$$\bar{\mathbf{t}} := \text{MakeU1G}\left[0, \lambda \cdot \Delta\bar{\mathbf{r}}, \frac{t_{\max}}{\Delta\bar{\mathbf{t}}}\right]. \tag{3.1}$$

The radial mesh spacing, $\Delta\bar{\mathbf{r}} = \Delta\bar{\mathbf{r}}'$, and the 'Courant number' $\lambda \equiv \Delta\bar{\mathbf{t}}/\Delta\bar{\mathbf{r}}$, along with $r_{\max}$ and $t_{\max}$ are parameters of the computation. It is important to note that for fixed initial data, whenever we perform a series of calculations where we vary the mesh spacings, λ is held fixed. Thus, any calculation in that series is effectively characterized by a *single* discretization scale, h, which we can conveniently identify with $\Delta\bar{\mathbf{r}}$. CA is a finite-difference code; as described in [4], equations (2.13)–(2.16) are differenced using second-order, central difference approximations for *all* derivatives. This includes the temporal derivatives in equations (2.15)–(2.16) which are treated with a 'leap-frog' [11] scheme wherein data at time $\bar{t}^{n+1}$ are determined from values at times $\bar{t}^n$ and $\bar{t}^{n-1}$. The centring of spatial derivatives is aided, to a degree, by the use of two separate radial meshes. The scalar field, ϕ^{CA}, for example, is defined on $[\bar{\mathbf{r}}, \bar{\mathbf{t}}]$. Then an $O(h^2)$ approximation to the spatial derivative $\Phi \equiv \partial\phi/\partial r$ is naturally defined on $[\bar{\mathbf{r}}', \bar{\mathbf{t}}]$ simply as the first divided difference of $\phi^{\mathrm{CA}}[\bar{\mathbf{r}}, \bar{\mathbf{t}}]$. Initial data for CA are generated by evaluating some specified function of r on the 1-grid $\bar{\mathbf{r}}$ to produce $\phi^{\mathrm{CA}}[\bar{\mathbf{r}}, 0.0]$. We must also specify the time derivative of $\phi(r, 0)$; in the calculations described later this is done indirectly by demanding that the scalar field configuration be purely *ingoing* at the initial time. Then, for our current purposes, the primary output from CA is a grid function which we denote as $\phi^{\mathrm{CA}}[\bar{\mathbf{r}}, \bar{\mathbf{t}}]$.

CA was constructed to have *precisely* $O(h^2)$ *truncation error*—by which we mean that the CA difference equations, which are derived from the discretization of (2.13)–(2.16), yield $O(h^2)$ residuals when applied to the *exact* solution of the field equations. It has also been established, primarily on empirical grounds, that the actual *solution error* (in ϕ, for example), $\phi^{\mathrm{CA}}[\bar{\mathbf{r}}, \bar{\mathbf{t}}] - \phi[\bar{\mathbf{r}}, \bar{\mathbf{t}}]$, is also $O(h^2)$. In fact, following Richardson [17], we have previously argued [5] that due to the uniformity of the CA mesh structure and the ubiquitous use of second-order, centred difference formulae, we expect $\phi^{\mathrm{CA}}[\bar{\mathbf{r}}, \bar{\mathbf{t}}]$ (as well as other CA grid functions) to admit an asymptotic ($h \to 0$) expansion of the form

$$\phi^{\mathrm{CA}}[\bar{\mathbf{r}}, \bar{\mathbf{t}}] = \phi[\bar{\mathbf{r}}, \bar{\mathbf{t}}] + h^2 e_2[\bar{\mathbf{r}}, \bar{\mathbf{t}}] + h^4 e_4[\bar{\mathbf{r}}, \bar{\mathbf{t}}] + \cdots. \tag{3.2}$$

Here, ϕ is the exact solution and the *error functions* $e_2(r,t), e_4(r,t), \ldots$ are *independent* of h as $h \to 0$ and, roughly speaking, have smoothness comparable to some appropriately high-order derivative of $\phi(r,t)$. As discussed in [5], given the differential equation and the difference equations, it is, in principle, possible to establish the validity of expansions such as (3.2). The basic idea is that, as $h \to 0$, the various error functions such as e_2 and e_4 will *themselves* obey certain auxiliary systems of partial differential equations whose form is apt to be similar to, and at least as complicated as, the original system. Thus, the difficult part of establishing (3.2) will involve demonstrating that, given appropriate initial data, these auxiliary systems of equations for the error functions have unique, bounded solutions. However, in the context of actual numerical calculation, it is not clear how necessary such a rigorous demonstration is, particularly since in this work we have invariably found it straightforward to *empirically* (numerically) establish when an expansion like (3.2) holds.

In brief then, we have found that the expansion provides an accurate representation of $\phi^{\mathrm{CA}}[\bar{\mathbf{r}}, \bar{\mathbf{t}}]$ even at finite values of h, and this provides us with the opportunity to use *Richardson extrapolation* [8, 11, 12] to increase the accuracy of the CA calculations. For example, we could first generate a series of grid functions, $\phi_i^{\mathrm{CA}}[\bar{\mathbf{r}}_i, \bar{\mathbf{t}}_i]$, $i = i_{\min}, \ldots, i_{\max}$, from fixed initial data, but using different basic mesh spacings, h_i. Then, by taking an appropriate linear combination of these grid functions, we could, in the ideal case, eliminate the $i_{\max} - i_{\min}$ leading order error terms from one specific grid function (presumably the one from the highest resolution calculation). Now, the extrapolated values would be defined at some set of (r, t) (Cauchy) coordinates. In fact, we do *not* deal with such results in this paper; rather, as discussed in the next section, we first transform the CA output to (R, U) coordinates and *then* extrapolate. We can understand heuristically how the resulting algorithm works by again following Richardson and arguing that for grid functions which are solutions of second-order centred difference equations, the expansion property should be generally invariant with respect to subsequent numerical operations (such as numerical differentiation, numerical integration and interpolation), provided that these operations are *themselves* based upon second-order, central-differencing.

To be slightly more precise, the algorithms described in the next section can be viewed as series of computations where the output grid function from one stage is the input to the next stage. Let us say that a grid function is *expandable* if it admits an expansion of the form (3.2). In general, we will not be too concerned if the expansion ceases to be valid after the first error term, so that expandable will imply, as a minimum, that e_2 is unique and h-independent. The basic strategy in designing the procedures described later is to construct each stage of the computation so that if the appropriate *continuum* values were supplied as input, the output would be expandable. This is often simply a matter of consistently using central-difference formulae on uniform meshes. By Richardson's argument, when *approximate* values are supplied as input, the output is *still* expandable *provided that the input values are themselves expandable.* The utility of this idea cannot be overemphasized since it allows us to chain together many separate numerical computations which, in the end, produce a final grid function which is still expandable and which is therefore still amenable to extrapolation.

4. Details of the comparison

4.1. *Transforming CA output to CH coordinates*

Because CH and CA use different coordinate systems, it is necessary to perform some numerical 'post-processing' on the results from at least one code in order to directly compare solutions. We trust that is clear that any error estimates which we derive at the end of such a procedure should represent *upper bounds* on the real level of deviation between the two solutions (to the extent that such a measure can be sensibly defined), unless the numerical transformations serendipitously reduce the level of discrepancy. It was most straightforward to set things up so that this post-processing could be viewed as a transformation of the CA output into quantities which were then directly (up to an interpolation) compared to those generated by CH. We discuss this transformation in some detail in this section. We note at this point that we have only compared the functions ϕ (alias $\overline{H}$) and H, rather than attempting to implement a general scheme wherein an arbitrary dynamical quantity (from either code), including metric functions, could be compared. However, since both systems of PDEs involve such an obviously fundamental coupling of the scalar field variables to the gravitational variables, we feel that a demonstration of convergence of the scalar field variables, in a calculation involving significant self-gravitation, implicitly demonstrates convergence of the entire solution.

The simplicity of the model problem, combined with the special and related nature of the coordinate systems we used in constructing CH and CA, made it quite easy to modify CA so that it produced numbers which could be directly compared to the output from CH. This was particularly true when the output we compared was the *scalar* quantity ϕ—then we had to do little more than augment CA so that it could compute trajectories of outgoing null rays and record the calculated values of ϕ along such trajectories. (As will be seen in the next sub-section, the procedure was somewhat more involved when we compared H. In addition, the generation of the CA results was complicated by the use of Richardson extrapolation—this too is discussed in section 4.2.) We note that an obvious alternative approach to the comparison (which could also have been implemented without undue trouble) would have had CH find surfaces of constant CA time. However, as we have just stated, we set up the comparison so that the post-processing of CH results is minimal. Finally, we point out that the modifications to CA, which essentially amount to a transformation of CA results to CH coordinates, provide the means for supplying equivalent initial data to the two codes.

Concentrating on the 3+1 code, then, we recall that we parametrize retarded time using the proper time of a central ($R = r = 0$) observer. We denote this quantity by $U_0(t)$ and observe that CA can easily monitor it by using a discrete version of the following expression, which follows directly from (2.11):

$$U_0(t) \equiv U(0,t) = \int_0^t \alpha(0,t')\,\mathrm{d}t'. \tag{4.1}$$

Here, we have chosen CA's coordinate origin, $(r,t) = (0,0)$, so that $U = 0$ intersects it. The particular $O(h^2)$ discretization used to compute a grid function, $U_0[\bar{t}]$, which approximates $U_0(t)$ is

$$U_0{}^{n+1} := U_0{}^{n} + \tfrac{1}{2}\Delta t \cdot \mathrm{Interpolate}[\tfrac{1}{2}(\alpha[\bar{\mathrm{r}}, \bar{t}^{n+1}] + \alpha[\bar{\mathrm{r}}, \bar{t}^{n}]), 0.0, 4] \tag{4.2}$$

where the interpolation operation returns an $O(h^2)$ approximation of $\alpha(0.0, \bar{t}^{n+1/2})$. Now, for any specific value of U_0, which we will generically refer to as a *launch time*, U_l, CA can approximately compute the trajectory, $R_{U_l}(t)$, of the outgoing null geodesic which is launched from the origin at $U_0 = U_l$. The null geodesic equation which CA approximately solves is just

$$\frac{\mathrm{d}}{\mathrm{d}t} R_{U_l}(t) = \frac{\alpha}{a}(r,t)\bigg|_{r=R_{U_l}(t),\, t=t}. \tag{4.3}$$

The initial condition is

$$R_{U_l}(t_l) = 0 \tag{4.4}$$

where t_l is the value of 3+1 time at which the geodesic is launched and thus satisfies

$$U_0(t_l) = U_l. \tag{4.5}$$

Again, we employ an $O(h^2)$ discretization of (4.3) in CA to compute a grid function, $R_{U_l}[\bar{\mathbf{t}}]$ which we may also view as a 1-grid $\mathbf{R}_{U_l}$. Specifically, we use

$$\frac{R_{U_l}{}^{n+1} - R_{U_l}{}^{n-1}}{2\Delta t} := \text{Interpolate}\left[\frac{\alpha}{a}[\bar{\mathbf{r}}, \bar{t}^n], R_{U_l}{}^n, 4\right] \tag{4.6}$$

so that at every CA time step, $\bar{t}^n$, we must interpolate in the uniform 3 + 1 radial mesh, $\bar{\mathbf{r}}$, to get a value of α/a at the radial coordinate $R_{U_l}{}^n$. Note that (4.6) requires two starting values, since it is a 'leap-frog' integration. In general, the launch time, t_l, will *not* coincide with one of the CA mesh times, $\bar{t}^n$, and some additional care (which, again, involves interpolation in the CA mesh) must be exercised in order that the initial values are set properly so that the whole process retains second-order accuracy. (However, in relation to the extrapolation algorithm described in section 4.2, some of these details are not completely understood, and would benefit from further study. In particular, see figures 6 and 7 and further discussion in the accompanying text.)

As the trajectory of a given geodesic is calculated, the computed values of the scalar field along the geodesic are accumulated to form a 1-grid function $\phi^{\mathrm{CA}}_{U_l}[\bar{\mathbf{t}}]$, which can also be indexed with the 1-grid, $\mathbf{R}_{U_l}$. Thus,

$$\phi^{\mathrm{CA}}_{U_l}[\bar{\mathbf{t}}] = \phi^{\mathrm{CA}}_{U_l}[\mathbf{R}_{U_l}] := \text{Interpolate}[\phi[\bar{\mathbf{r}}, \bar{\mathbf{t}}], R_{U_l}[\bar{\mathbf{t}}], 4]. \tag{4.7}$$

In general, the set of specified launch times form a 1-grid $\mathbf{U}_l$ and we can view the entire set of values accumulated along all of the null geodesics as the 2-grid function $\phi^{\mathrm{CA}}[\mathbf{R}(\mathbf{U}_l), \mathbf{U}_l]$. Now, this is a grid function which is almost directly comparable to the output from CH. In general, for a given initial radial grid $\overline{\mathbf{R}}^0$ and a set of retarded times, $\mathbf{U}_l$, the output from CH will be $\phi^{\mathrm{CH}}[\mathbf{R}'(\mathbf{U}_l), \mathbf{U}_l]$, for some $\mathbf{R}'(\mathbf{U}_l) \neq \mathbf{R}(\mathbf{U}_l)$. Therefore, at this point we could effect a direct comparison with one more interpolation operation; for example, the computation

$$\phi^{\mathrm{CA}}[\mathbf{R}'[\mathbf{U}_l], \mathbf{U}_l] := \text{Interpolate}[\phi^{\mathrm{CA}}[\mathbf{R}(\mathbf{U}_l), \mathbf{U}_l], \mathbf{R}'(\mathbf{U}_l), 4] \tag{4.8}$$

would suffice. However, as stated previously, we do not view $\phi^{\mathrm{CA}}[\mathbf{R}(\mathbf{U}_l), \mathbf{U}_l]$ as the final output of CA. The generation of the final CA results involves Richardson extrapolation, and is described in more detail in the next sub-section. Recall that extrapolation involves calculations with grid functions generated from fixed initial data, but using different discretization scales, h_i. (A subscript i will often be used in the following to label a quantity computed at resolution h_i.) In order to expedite the extrapolation of its output, it proved useful to have CA perform a final interpolation of the grid function, $\phi^{\mathrm{CA}}[\mathbf{R}(\mathbf{U}_l), \mathbf{U}_l]$, to a *uniform* radial grid, $\overline{\mathbf{R}}$. Then, we conclude this sub-section by stating that at this point we can view CA as an algorithm which takes as input: (1) initial data, (2) a uniform 2-grid $[\bar{\mathbf{r}}_i, \bar{\mathbf{t}}_i]$, with characteristic scale, h_i; and (3) a null geodesic grid $\mathbf{U}_l$, and produces, as output, the grid function:

$$\phi^{\mathrm{CA}}_i[\overline{\mathbf{R}}_i, \mathbf{U}_l] \tag{4.9}$$

where $\Delta\bar{\mathbf{r}}_i$, $\Delta\bar{\mathbf{t}}_i$ and $\Delta\overline{\mathbf{R}}_i$ are all constant multiples of h_i. Specifically, for any given initial data, the ratios $\lambda \equiv \Delta\bar{\mathbf{t}}_i/\Delta\bar{\mathbf{r}}_i$ and $\sigma \equiv \Delta\overline{\mathbf{R}}_i/\Delta\bar{\mathbf{r}}_i$ are always held fixed as we vary $h_i = \Delta\bar{\mathbf{r}}_i$. For all of the calculations subsequently described, we used $\lambda = 0.25$, $\sigma = 1.00$.

4.2. *Richardson-extrapolating the* CA *results*

In the remainder of the paper we will often deal with series of grid functions computed by CH and CA at different resolutions, h_i. Such series are of interest to us for two basic reasons. First, the investigation of the behaviour of the estimated errors in grid functions as a function of h_i—*convergence testing*—is the primary tool we use in our comparison of the two codes, as well as in our testing of the algorithms described in the current section. Second, we need to generate such series for our extrapolation procedures. The discussion in this sub-section is chiefly concerned with this latter function, but we stress that our basic strategy will always be to perform a comparison of final results—even if they are *extrapolated* final results—which are generated using a number of distinct resolutions.

Considering the extrapolation of CA data, then, we adopt a convention of 'increasing resolution' (decreasing mesh-spacing) with increasing resolution-index (subscript), and as a matter of convenience, we invariably use $h_i = 2h_{i+1}$ ('2:1 refinement ratio'). Thus, for example, the grid structures for a sequence of CA computations are specified by

$$\begin{aligned}
\bar{\mathbf{r}}'_i &:= \text{MakeU1G}[0, 2^{-i}\cdot\Delta\bar{\mathbf{r}}_0, 2^i\cdot\#\bar{\mathbf{r}}_0] \\
\bar{\mathbf{r}}_i &:= \text{MakeU1G}[-\tfrac{1}{2}\Delta\bar{\mathbf{r}}'_i, \Delta\bar{\mathbf{r}}'_i, \#\bar{\mathbf{r}}'_i + 1] \\
\bar{\mathbf{t}}_i &:= \text{MakeU1G}[0, \lambda\cdot\Delta\bar{\mathbf{r}}'_i, 2^i\cdot\#\bar{\mathbf{t}}_0].
\end{aligned} \tag{4.10}$$

Here, $\Delta\bar{\mathbf{r}}_0$, $\#\bar{\mathbf{r}}_0$, and $\#\bar{\mathbf{t}}_0$ are parameters defining the extent and scale of the coarsest grid. Although essentially arbitrary, it is sensible to choose these parameters so that the h_0 calculation is computationally inexpensive, yet useful. We have just discussed the computation of $\phi_i^{\text{CA}}[\overline{\mathbf{R}}_i, \mathbf{U}_l]$ (or simply ϕ_i^{CA} where there can be no confusion)—from these values we compute Richardson-extrapolated quantities which we label with a double subscript, ii', where $i' > i$. Specifically,

$$\begin{aligned}
\phi_{ii'}^{\text{CA}}[\mathbf{R}_r(\mathbf{U}_l), \mathbf{U}_l] &\equiv \texttt{Extrapolate}[\phi_i^{\text{CA}}, \phi_{i+1}^{\text{CA}}, \cdots, \phi_{i'}^{\text{CA}}, \mathbf{R}_r(\mathbf{U}_l), \mathbf{U}_l] \\
&:= \sum_{k=i}^{i'} \mathbf{wx}_{k-i}^{i'-i+1}\cdot\text{Interpolate}[\phi_k^{\text{CA}}, \mathbf{R}_r(\mathbf{U}_l), 2(i'-i+1)].
\end{aligned} \tag{4.11}$$

Here, $\mathbf{R}_r(\mathbf{U}_l)$ is a 1-grid set of reference coordinates at which we eventually perform the comparison with CH results—in general we can $\texttt{Extrapolate}[\cdots]$ to any set of coordinates. The extrapolation involves a weighted sum over single-level CA output, which is itself interpolated to the reference coordinates. (Heuristic reasoning based on equation (3.2) suggests that $2(i'-i+1)$ is an appropriately high degree of interpolation.) In general, the weights, $\mathbf{wx}_j^{i'-i+1}$ depend on: (1) the number of levels of data used—$(i'-i+1)$; (2) the various mesh ratios—$h_i/h_{i+1}, h_{i+1}/h_{i+2}, \ldots$ (always 2 in our calculations); and (3) the nature (possibly empirically determined) of the grid function's Richardson expansion.

Now, our empirically motivated presumption of the expansion (3.2) for $\phi^{\text{CA}}[\bar{\mathbf{r}}, \bar{\mathbf{t}}]$, combined with the assumption that the transformation of the CA data to (R, U) coordinates preserves expandability, suggests that we assume

$$\phi_i^{\text{CA}}[\overline{\mathbf{R}}_i, \mathbf{U}_l] = \phi[\overline{\mathbf{R}}_i, \mathbf{U}_l] + h^2 E_2[\overline{\mathbf{R}}_i, \mathbf{U}_l] + h^4 E_4[\overline{\mathbf{R}}_i, \mathbf{U}_l] + \cdots. \tag{4.12}$$

We are particularly interested in values extrapolated from two levels of data. Such values, which ostensibly should be $O(h^4)$, are computed from (4.11) specialized to the case $i' = i + 1$. Thus, given that $h_i/h_{i+1} = 2$, the weights are $\mathbf{wx}_0^2 = -\frac{1}{3}$, $\mathbf{wx}_1^2 = \frac{4}{3}$. As we shall see, there is good empirical evidence that this procedure *does* lead to two-level extrapolated CA values for ϕ which are $O(h^4)$, and it is precisely these values which we view as the final output of CA. We have not studied in detail the issue of whether the expansion (4.12) is valid past the leading order error term (E_4 may not be independent of h, for example). However, we know (again, from empirical evidence) that extrapolation naively based on (4.12) and involving more than two levels of data generally leads to results significantly more accurate than the two-level ones. So, although we cannot claim to have a clear theoretical understanding of multi-level extrapolation, we nonetheless use the procedure to generate very accurate values which have obvious utility in our studies of the errors in the CH and CA results. Thus, for example, we compute the initial data, $\phi_i^{\mathrm{CH}}[\overline{\mathbf{R}}_i, 0.0]$, which is then supplied to the characteristic program CH, using

$$\phi_i^{\mathrm{CH}}[\overline{\mathbf{R}}_i, 0.0] := \mathtt{Extrapolate}[\phi_{i_{\min}}^{\mathrm{CA}}, \cdots, \phi_{i_{\max}}^{\mathrm{CA}}, \overline{\mathbf{R}}_i, 0.0]. \tag{4.13}$$

Here, $i_{\min}$ and $i_{\max}$ are chosen in an attempt to produce a very low level of error in $\phi_i^{\mathrm{CH}}[\overline{\mathbf{R}}_i, 0.0]$. Leaving the specific numerical details for later, we note that by using up to five levels, we can quite easily generate initial data for CH which has fractional accuracy of 10^{-8} or better. This allows us to supply effectively exact initial data to CH, as well as CA, eliminating an obvious potential bias in the comparison calculations described in section 6. Now, the extrapolated results along $U = 0$ are consistently superior to those generated at later retarded times. However, we estimate that *all* of the multi-level extrapolated results used later have fractional errors which are $\leqslant 10^{-5}$. As will be seen, this is more than adequate to elucidate the convergence properties of both the CH and CA (two-level extrapolated) results at resolutions characteristic of the calculations which have previously been reported [9, 10, 4, 5].

We now discuss the generation of extrapolated approximations of the function H from the CA data. Originally, we used a procedure completely analagous to that just described for ϕ; we produced approximate values $H_i^{\mathrm{CA}}[\overline{\mathbf{R}}_i, \mathbf{U}_l]$ at varying resolutions h_i, and then extrapolated. However, we found that the convergence properties of the values so produced were significantly inferior to the convergence properties of the scalar field itself. We now use a method which takes advantage both of the simple relation between ϕ and H,

$$H = \frac{\partial}{\partial R}(R\phi) \tag{4.14}$$

and the fact that we can use (4.11) to extrapolate CA data to any radial coordinate along a given outgoing null ray. Specifically, we compute

$$H_{ii'}^{\mathrm{CA}}[\mathbf{R}_r(\mathbf{U}_l), \mathbf{U}_l] := \sum_{k=-2}^{2} \mathbf{wfd}_k \cdot (\mathbf{R}_r(\mathbf{U}_l) + k\epsilon) \cdot \phi_{ii'}^{\mathrm{CA}}[\mathbf{R}_r(\mathbf{U}_l) + k\epsilon, \mathbf{U}_l] \tag{4.15}$$

where ϵ is some arbitrary (but small) spacing, the finite difference weights, $\mathbf{wfd}_k$, are $(12\epsilon)^{-1}[-1, 8, 0, -8, 1]$ and the values $\phi_{ii'}^{\mathrm{CA}}[\mathbf{R}_r(\mathbf{U}_l) + k\epsilon, \mathbf{U}_l]$ are computed from (4.11). Here, we are essentially using a standard technique for computing numerical derivatives of functions which can be evaluated at arbitrary values of their argument. The basic idea is that the error in $H^{\mathrm{CA}}_{ii'}[\mathbf{R}_r(\mathbf{U}_l),\mathbf{U}_l]$ will have a component due to the approximate nature of $\phi^{\mathrm{CA}}_{ii'}[\mathbf{R}_r(\mathbf{U}_l),\mathbf{U}_l]$ as well as a contribution arising from the use of an $O(\epsilon^4)$ finite difference operation (instead of a differentiation) in (4.15). By an appropriate choice of ϵ we try to minimize the latter type of error, since the former is clearly an 'intrinsic' error at this stage of the calculation. We note that for any specific calculation on a fixed-precision machine, we expect an *optimal* value of ϵ to exist. This follows from the standard argument that as $\epsilon \to 0$, errors due to the finite-precision nature of floating-point arithmetic become proportionally more severe and eventually exceed the actual finite-differencing error. However, in the computations described later, we simply used a fixed and experimentally determined value, $\epsilon = 0.001$. We also note that we could have experimented with other finite difference approximations; however, the $O(\epsilon^4)$ central formula used in (4.15) seemed satisfactory. The results we obtain suggest that the finite-differencing error was generally small in comparison with the intrinsic error (but see figure 4).

We can conclude our description of the manner in which the comparison is set up by directing our attention to the characteristic code, CH. Here, we have as basic output the grid functions $\phi^{\mathrm{CH}}_i[\mathbf{R}_i(\mathbf{U}_l),\mathbf{U}_l]$ and $H^{\mathrm{CH}}_i[\mathbf{R}_i(\mathbf{U}_l),\mathbf{U}_l]$. To enable direct comparison with the extrapolated CA values, we merely have to interpolate to the reference coordinates, $\mathbf{R}_r(\mathbf{U}_l)$. Thus we compute

$$\begin{aligned}\phi^{\mathrm{CH}}_i[\mathbf{R}_r(\mathbf{U}_l),\mathbf{U}_l] &= \mathrm{Interpolate}[\phi^{\mathrm{CH}}_i[\mathbf{R}_i(\mathbf{U}_l),\mathbf{U}_l],\mathbf{R}_r(\mathbf{U}_l),6]\\ H^{\mathrm{CH}}_i[\mathbf{R}_r(\mathbf{U}_l),\mathbf{U}_l] &= \mathrm{Interpolate}[H^{\mathrm{CH}}_i[\mathbf{R}_i(\mathbf{U}_l),\mathbf{U}_l],\mathbf{R}_r(\mathbf{U}_l),6]\end{aligned} \tag{4.16}$$

where we use sixth-order interpolation in an arguably over-cautious attempt to minimize interpolation error relative to truncation error.

5. Testing the numerical coordinate transformation

In the extreme weak field, flat spacetime limit, where the scalar field decouples from the gravitational field, our model problem (in the 3 + 1 formalism) reduces to the usual linear wave equation for spherically symmetric disturbances:

$$\frac{\partial^2\phi}{\partial t^2} = \frac{1}{r^2}\frac{\partial}{\partial r}\left(r^2\frac{\partial\phi}{\partial r}\right) \tag{5.1}$$

or

$$\frac{\partial^2(r\phi)}{\partial t^2} = \frac{\partial^2(r\phi)}{\partial r^2}. \tag{5.2}$$

A general solution of this equation may be constructed by specifying two arbitrary functions $f(r)$ and $g(r)$ as the ingoing and outgoing components, respectively, of $r\phi$ at the initial time, $t = 0$. Then, initially

$$r\phi(r,0) = f(r) + g(r) \tag{5.3}$$

$$\frac{\partial}{\partial t}(r\phi)(r,0) = f'(r) - g'(r) \tag{5.4}$$

and the complete solution is

$$r\phi(r,t) = f(t+r) + g(r-t) \qquad r \geqslant t \tag{5.5}$$

$$r\phi(r,t) = f(t+r) - f(t-r) \qquad r < t. \tag{5.6}$$

Thus, given the initial data, $f(r)$, $g(r)$, which, along with the ordinary derivatives $f'(r)$, $g'(r)$, we always take as closed form expressions in r, we can compute the values satisfying (5.1) for arbitrary r and t. In particular, we can calculate the scalar field values along an arbitrary outgoing null ray, $U = t - r = \texttt{constant}$. We can also compute the function, H, along such a geodesic using the following relation

$$H(R,U) = H(r, t-r) = \frac{\partial}{\partial R}(R\phi) = \frac{\partial}{\partial r}(r\phi) + \frac{\partial}{\partial t}(r\phi). \tag{5.7}$$

In this section we use such a linear solution to test the algorithms described in the previous section. (The use of linear data to check CH has previously been documented [9].) Although gravitational effects are absent, we can still test the various stages of the process which transforms output from CA to values which can be directly compared to the output from CH. These stages include the initialization and integration of the finite-differenced null-geodesic equations for arbitrary values of retarded time, the accumulation of scalar field values along the null rays, the interpolation of the accumulated values to a uniform mesh, and the interpolation, Richardson extrapolation and finite-differencing operations which produce the final values of ϕ and H at some specified reference coordinates. We note that the convergence testing procedure described later can be used to investigate the implementation of each step of the calculation; this was, in fact, done, and was quite useful both in designing and debugging the algorithm. Here, we will simply examine the *overall* convergence of the method, and assert that, provided the set of reference coordinates is suitably generic, the convergence of the algorithm as a whole implies convergence of the constituent stages.

The convergence tests we make here, and in the next section, are straightforward. For fixed initial data and (R, U) reference coordinates, we directly measure errors in the computed approximations of ϕ and H as a function of the basic mesh spacing, h_i. Given a generic grid function, Q^{C}, and an associated continuum quantity, Q, which can be evaluated on Q^{C}'s mesh, we define $E_1(Q^{\mathrm{C}}, Q)$ and $E_\infty(Q^{\mathrm{C}}, Q)$, which provide *logarithmic measures of per cent relative error* (mean and maximum, respectively) at any given value of retarded time. Specifically, for the values of retarded times $U_l{}^n \in \mathrm{U}_l$, we compute

$$E_1(Q^{\mathrm{C}}, Q)[\mathrm{U}_l] \equiv \log_2\left(100\frac{\|(Q^{\mathrm{C}} - Q)[\mathbf{R}_r(\mathrm{U}_l)\mathrm{U}_l]\|_1}{\|Q[\mathbf{R}_r(\mathrm{U}_l), \mathrm{U}_l]\|_1}\right) \tag{5.8}$$

We use a base-2 logarithm since whenever we perform a convergence test which involves a series of grid functions $Q_i^{\mathrm{C}} = Q_1^{\mathrm{C}}, Q_2^{\mathrm{C}}, \ldots$ we take $h_i = 2h_{i+1}$. Then, for Q_i^{C} which are expandable in the sense defined in section 3.4, with leading error terms of order $(h_i)^p$, we should find for all $U_l{}^n$

$$\lim_{h_i \to 0}(E_{1/\infty}(Q_i^{\mathrm{C}}, Q)[U_l{}^n] - E_{1/\infty}(Q_{i+1}^{\mathrm{C}}, Q)[U_l{}^n]) = p. \tag{5.9}$$

Note that $E_1 = 0$ and $E_1 = -10$ correspond to 1% and about 0.001% relative errors, respectively.

The initial datum for the test is a single, ingoing 'Gaussian' pulse in ϕ, defined by

$$f(r) = f_0 r^3 \exp\left(-\left(\frac{r-r_0}{\delta}\right)^2\right) \tag{5.10}$$

$$g(r) = 0. \tag{5.11}$$

where f_0, r_0 and δ are parameters. We have also used this type of initial data for the intermediate and strong field computations described in the next section; in particular, f_0 is a convenient *control parameter* with which to govern the 'strength' of the gravitational self-interaction and has been used as such in this study. Thus, for all the calculations described later, the other parameters were fixed; specifically, we took $r_0 = 25.0$ and $\delta = 3.0$. Finally, we observe that, strictly speaking, this is *not* a 'linear test', since the full, non-linear version of CA is used; however, we claim that the spacetime which is generated using $f_0 = 1.0 \times 10^{-10}$ is flat to the level of precision with which we are concerned.

In the test, numerical values are compared with exact values on the following reference grid:

$$[\mathbf{R}_r(\mathbf{U}_l), \mathbf{U}_l] = [\overline{\mathbf{R}}_r, \mathbf{U}_l] = [\mathrm{MakeU1G}[0.0, 0.5, 100], \mathbf{U}_l] \tag{5.12}$$

with

$$\mathbf{U}_l = [0.0, 3.6, 7.6, 11.6, 15.6, 19.6, 23.6, 27.6, 31.6, 35.6, 39.6]. \tag{5.13}$$

There is nothing particularly significant about these launch times, other than the fact that they are also used in the actual comparison calculations described in the next section. As will be mentioned later, it *is* significant that the reference grid is radially uniform. Figure 2 displays some of the main features of the test calculation. As described in the caption, the dotted lines show the trajectories of the outgoing null geodesics in the (r, t) plane, while the boundaries of the shaded regions in figures 2(a) demark e^{-1} limits of $|\phi|$ and 1%–99% limits of m, respectively. Various discrete norms of ϕ and H as a function of U are shown in figure 2(b).

The convergence testing procedure, as applied to the 'raw' (non-extrapolated) output of CA is summarized in figure 3. The test used five levels of discretization ranging from $h_1 = \Delta\bar{r}_1 = \frac{1}{2}$ to $h_5 = \Delta\bar{r}_5 = 1/32$. We first note that, in accordance with equation (5.9), the type of pattern seen in these graphs—namely a regularly spaced vertical sequence of symbols with a separation between symbols of p units—is *the* generic signal of $O(h^p)$ convergence. It is quite clear from the plots, then, that both ϕ_i^{CA} and H_i^{CA} have $O(h^2)$ error. Lest the reader be concerned about a lack of convergence at $U_l{}^n = 39.6$, we point out that the norms of ϕ and H are at that time reduced by more than six orders of magnitude relative to $U_l{}^n = 0.0$ (see figure 2). A plot of the convergence of H_i^{CA} shows essentially the same behaviour.

Figure 4 shows the results of the convergence test of the two-level, Richardson-extrapolated quantities, ϕ_{ii}^{CA} and H_{ii}^{CA}. Note the change (from figure 3) in the scale of the y-axes, as well as the fact that the vertical separation between markers is now

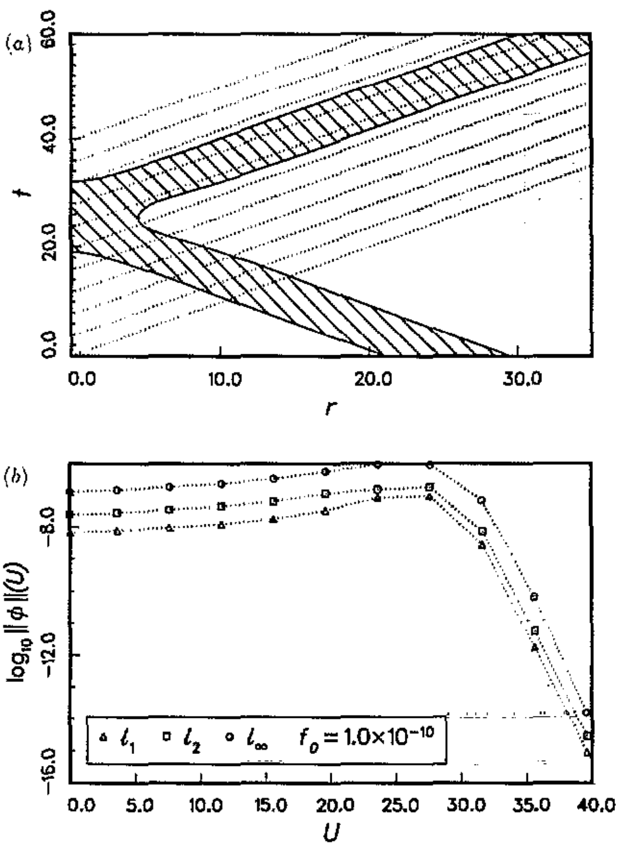


Figure 2. Spacetime plots of the principal features of the linear computation, and various norms of the scalar field variables ϕ and H. In (*a*), the cross-hatched region includes all grid points $(\bar r_j, \bar t^n)$ such that $|\phi[\bar r_j, \bar t^n]| \geqslant e^{-1} \cdot \|\phi[\bar r, \bar t^n]\|2$. The dotted lines are the (r,t) trajectories of outgoing null geodesics labelled by values of retarded time, U, which are elements of the launch vector (1-grid), $U_l = [0.0, 3.6, 7.6, 11.6, 15.6, 19.6, 23.6, 27.6, 31.6, 35.6, 39.6]$. In the current case, the outgoing null geodesics are essentially straight lines with unit slope. (*b*) are plots of various norms (spatial norms at specific values of U, see figure 1(*c*) of ϕ and H. Note that the norms of both functions are very much reduced at late retarded times.

generally four units. Again, this is a clear demonstration that, at least for this linear calculation, the extrapolated values have $O(h^4)$ error. The principal irregularities in these plots appear at and after $U_l{}^n = 23.6$ and are particularly in the l_∞ norm. This behaviour reflects the fact that, for a fixed resolution, the computations are least accurate during, and to a lesser extent, following the period when the pulse is propagating near $r = 0$. As described in the figure caption, difficulties in keeping the various difference equations precisely centred at $r = 0$ are most likely responsible for the dominant error terms in the extrapolated results. Nevertheless, as is usually the case in numerical calculations where Richardson extrapolation works, the gain in accuracy is quite impressive. This is particularly true for the highest resolution calculations, where the errors in the extrapolated results are generally at least 500 times smaller than those of the single-level calculations.

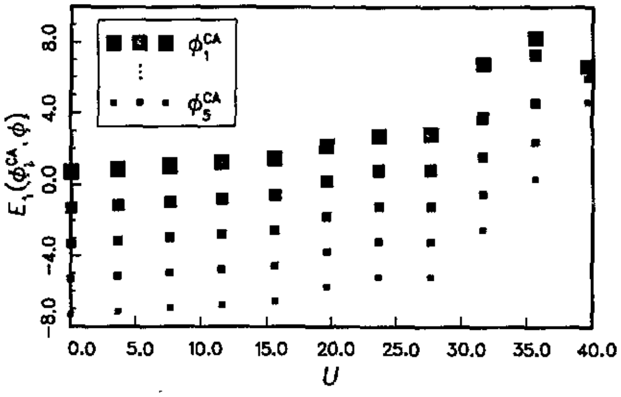


Figure 3. Results of a convergence test of the basic (non-extrapolated) CA grid functions $\phi_i^{CA}[\overline{\mathbf{R}}_i, \mathrm{U}_l]$ and $H_i^{CA}[\overline{\mathbf{R}}_i, \mathrm{U}_l]$. The error measures $E_1(\cdots)$ and $E_\infty(\cdots)$ are defined in the text. At any fixed value of U, the sequence of symbols shows a particular measure of the errors from a series of calulations made using different resolutions h_i, with $h_i = 2h_{i+1}$. Successive symbols should be two units apart (vertically) if the convergence of the quantity is second order ($\mathrm{O}(h^2)$).

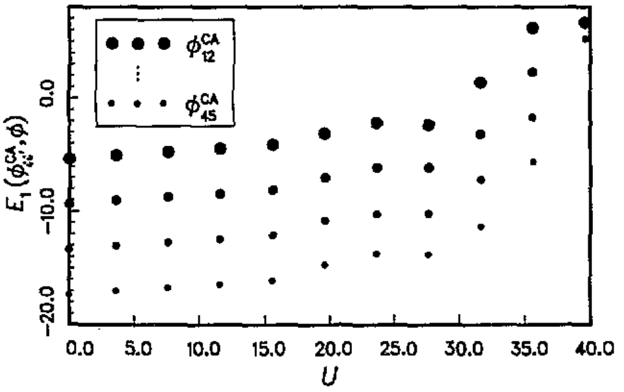


Figure 4. Convergence of the two-level extrapolated CA results $\phi_{ii'}^{CA}[\overline{\mathbf{R}}_r, \mathrm{U}_l]$ and $H_{ii'}^{CA}[\overline{\mathbf{R}}_r, \mathrm{U}_l]$. Note that for all U, the extrapolated values are generated on the uniform radial reference grid $\overline{\mathbf{R}}_r = \overline{\mathbf{R}}_1 = \mathrm{MakeU1G}[0.0, 0.5, 100]$. The convergence of both $\phi_{ii'}^{CA}$ and $H_{ii'}^{CA}$ is generally $\mathrm{O}(h^4)$ (different sized symbols vertically separated by four units) throughout the computation. For the highest resolution calculation ($ii' = 45$), the extrapolated results are about 500 times more accurate than the corresponding single level calculation ($i = 5$). Multi-level extrapolation produces better results still, particularly at early values of U.

6. Results of the comparison of CH and CA

In this section we present the results of our comparison of the CH and CA codes. Here we use initial field configurations which generate spacetimes where there is significant interaction between the scalar and gravitational fields. In contrast to the calculations described in the previous section, we do *not* have exact solutions with which to evaluate the accuracy and convergence of the various numerical results we compute from such initial data. Indeed, to a large extent, it was precisely this fact which motivated us to perform the comparison. However, as will be seen in the following, we have strong evidence that both codes are convergent, at least for the particular class of initial data we have used. Thus, for given initial data, we can, in principle, use either code to generate an arbitrarily accurate *numerical* solution by computing with an appropriately small mesh spacing. Moreover, for the purposes of investigating the convergence of the two codes down to some limiting discretization scale, $h_{\rm min}$, it is clearly sufficient that the error in the numerical reference solution be 'small' compared to the errors in the CH and CA results at that resolution. As described in section 3.4, we can efficiently produce a good reference solution using multi-level Richardson extrapolation of CA data, and this is the approach we have adopted. We wish to stress, however, that the convergence-testing approach we used was equally useful in establishing the relative performance of the two algorithms *before* we obtained the evidence that both codes were converging and that very accurate results could be obtained, even in the non-linear regime, using multiple-level Richardson extrapolation. We remind the reader that we expect $O(h^4)$ convergence *at best* for both the CH and CA results. We also note that we have not addressed the question of the relative computational efficiency of the two codes when operating at comparable resolutions. Our justification for this omission is as follows. Simple considerations indicate that both CH and CA should have run-times proportional to the number of discrete events which are used in the calculation and this expectation is borne out in practice. In terms of a comparison, the specific values of the respective constants of proportionality would be quite relevant if the *convergence rates* of the codes were the same. However, as will be seen, the convergence rates of CH and CA are *not* the same. To be fair in our comparison, we have to demand that the two codes achieve similar accuracies for a given problem, and particularly for high-accuracy computations, the difference in resource usage between the two codes will be dominated by the difference in convergence rates, not by the per-event operation count.

As discussed previously, we study solutions generated from initial data which, from the Cauchy (CA) point of view, describe a single 'Gaussian' pulse (or more properly, a shell, since the solution is spherically symmetric) of scalar field which is *ingoing* at $t = 0$. The resulting profile $\phi(R, 0)$ of the scalar field along the initial data surface, $U = 0$, for the characteristic (CH) code is also essentially 'Gaussian' in such solutions; however, in both of the cases which we consider, $\phi(R, 0)$ has a tail which falls off roughly like R^{-1} at large R. This tail is due to backscatter from the spacetime curvature which is itself induced by the scalar field. At this point we note that it has been well established numerically [4, 9, 10], and analytically [7], that the model we are studying admits black hole solutions. For the particular type of initial data we are considering, we know that if our control parameter, f_0, is chosen sufficiently large, the resulting spacetime will contain a black hole (in the one-parameter family of solutions considered here, a black hole forms at $f_0 = 2.08\ldots \times 10^{-5}$). However, it is also well established that neither CH nor CA is ideally suited for the study of black hole spacetimes since, in both cases, pathologies in the behaviour of the values of some of the discrete dynamical variables and/or their derivatives develop on a dynamical time scale, $t \sim M_{\rm bh}$, where $M_{\rm bh}$ is the black hole mass. The algorithms we have described cannot cope with these difficulties, which chiefly arise from our particular choice of coordinate systems. It is principally for this reason that we have avoided the 'black hole region of parameter space' in our comparison calculations.

We refer to computations we have done using control parameter values (see section 5, equation (5.10)) $f_0 = 1.0 \times 10^{-5}$ and $f_0 = 2.0 \times 10^{-5}$ as 'intermediate-field' and 'strong-field' calculations, respectively. One way of characterizing the 'strength' (non-linearity) of the solutions is to quote the maximal value of $2m(r,t)/r$ which develops in the evolution [9]. This measure is 0.0719 for the intermediate-strength calculation and 0.349 for the strong-field one. Figure 5 shows some of the non-linear (self-gravitating) aspects of the current computations. Although the appearance of the intermediate-field plots is quite similar to the corresponding linear-field graphs, the lag in integrated proper time at the origin relative to coordinate time is noticeable (as mentioned previously, we use the collection of launch times, $\mathbf{U}_l$, given by expression (5.13), in the two comparison computations). In the strong-field computation, additional signs of self-gravity are visible in the plots. These include the broadening of the support of the *outgoing* scalar radiation and the clear 'bending' of the null geodesics passing through the interaction region.

In order to assess the relative accuracies and convergence rates of our algorithms, we again use straightforward convergence tests. The relative levels of error in the codes, and, to a lesser extent, the convergence rate of these errors are, in general, resolution-dependent quantities. Here we compute with a range of mesh spacings characteristic of the discretization scales used, for example, in the computations described in [9]. Although we continue to work with the mesh-spacing sequence $h_i = 1/2, 1/4, 1/8, \ldots$ employed in section 5, it transpires that for both the intermediate- and strong-field calculations, the lowest resolution ($h_1 = 1/2$) CA computation is too coarse to be used effectively in the extrapolation algorithm. Thus, using resolutions down to a limiting value of $h_5 = 1/32$, we consider three separate sets of CA grid functions ($\phi_{23}^{\mathrm{CA}}, \phi_{34}^{\mathrm{CA}}$ and ϕ_{45}^{CA}, for example) and three sets of CH grid functions ($\phi_3^{\mathrm{CH}}, \phi_4^{\mathrm{CH}}, \phi_5^{\mathrm{CH}}$). As discussed earlier, as reference solutions (in lieu of the exact values we have to check linear computations) we use multi-level extrapolated values which are computed from CA computations made on levels 2 through 6 ($h_6 = 1/64$). As previously claimed, we feel that these reference values are generally good to at least a part in 10^5; this is a level of precision which is more than sufficient to reveal the dominant error behaviour in both the CH and (two-level) CA computations. We therefore feel justified in referring to the deviation of a grid function relative to such a multi-level extrapolated quantity simply as the error in that grid function. In addition, the multi-level values are used to provide initial data along $U = 0$ for the CH algorithm. We can argue that these values have errors no worse than a few parts in 10^8 so that any errors in CH values due to inaccuracies in the initial data are negligible.

For given launch times $\mathbf{U}_l$ and reference coordinates $\mathbf{R}_r(\mathbf{U}_l)$, we define then, in analogy to (5.8), the following measure of error of a grid function Q^{C} relative to the reference quantity Q_{26}^{CA}:

$$E_1(Q^{\mathrm{C}}, Q_{26}^{\mathrm{CA}})[U_l] \equiv \log_2\left(100\frac{\|(Q^{\mathrm{C}} - Q_{26}^{\mathrm{CA}})[\mathbf{R}_r(\mathbf{U}_l), \mathbf{U}_l]\|_1}{\|Q_{26}^{\mathrm{CA}}[\mathbf{R}_r(U_l), U_l]\|_1}\right). \quad (6.1)$$

For both computations described here, $\mathbf{R}_r(\mathbf{U}_l)$ were chosen to be the radial coordinates computed and used in the $i = 2$, CH calculation.

The errors, as defined above, for the various grid functions computed in the intermediate field calculations are plotted in figure 6. At early retarded times we see rapid convergence of all monitored grid functions. For the CH code, the convergence of both ϕ^{CH} and H^{CH} appears to be $\mathrm{O}(h^4)$; for the two-level extrapolated CA values, the early time convergence of ϕ^{CA} appears to be $\mathrm{O}(h^4)$ while we measure an $\mathrm{O}(h^3)$

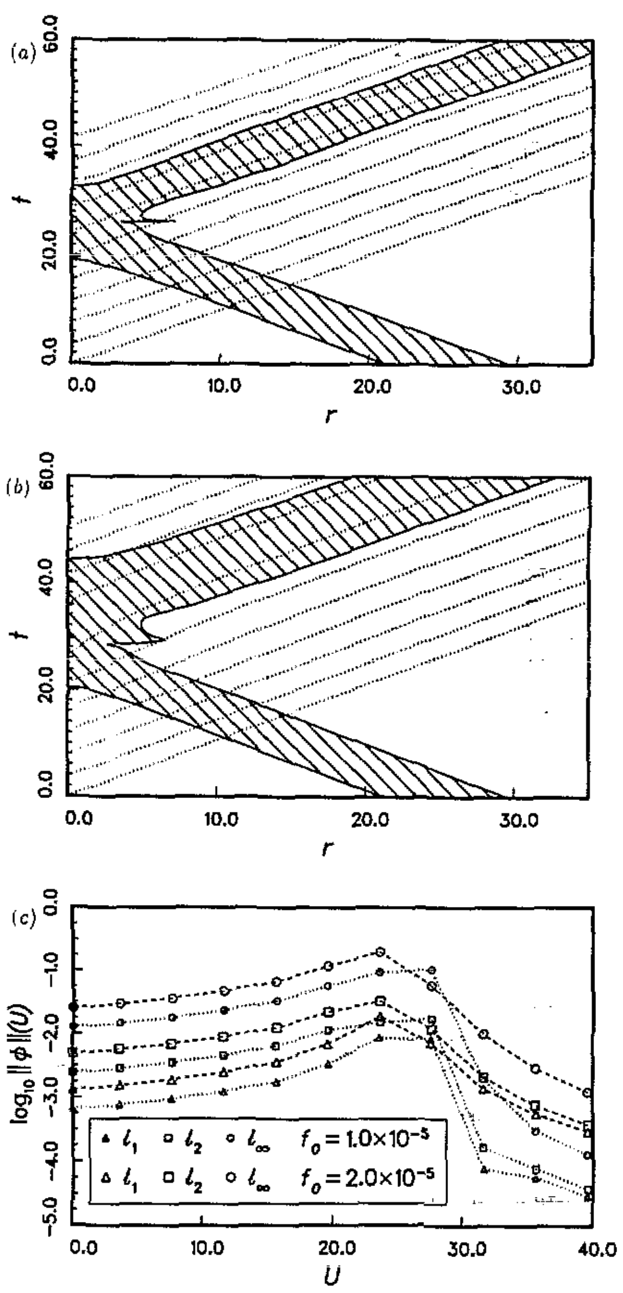


Figure 5. Spacetime plots showing some principal features of the intermediate-field (*a*) ($f_0 = 1.0 \times 10^{-5}$) and strong-field (*b*) ($f_0 = 2.0 \times 10^{-5}$) computations. The hatched areas are defined as in figure 2(*a*), and the dotted lines are again the $U \in \mathrm{U}_l$ null geodesics. Note that the non-linearity in the intermediate-field calculation does not induce much change from the linear case in the qualitative appearance of this plot. However, the lag in central proper time, U_0, relative to coordinate time, t, is noticeable—the last null geodesic is launched at $U = 39.6$. Non-linear effects are considerably more distinct in the corresponding strong-field plots. In particular, note the 'bending' of the outgoing null rays in the 'interaction region' of the computational domain. (*c*) shows various norms of ϕ for both calculations.

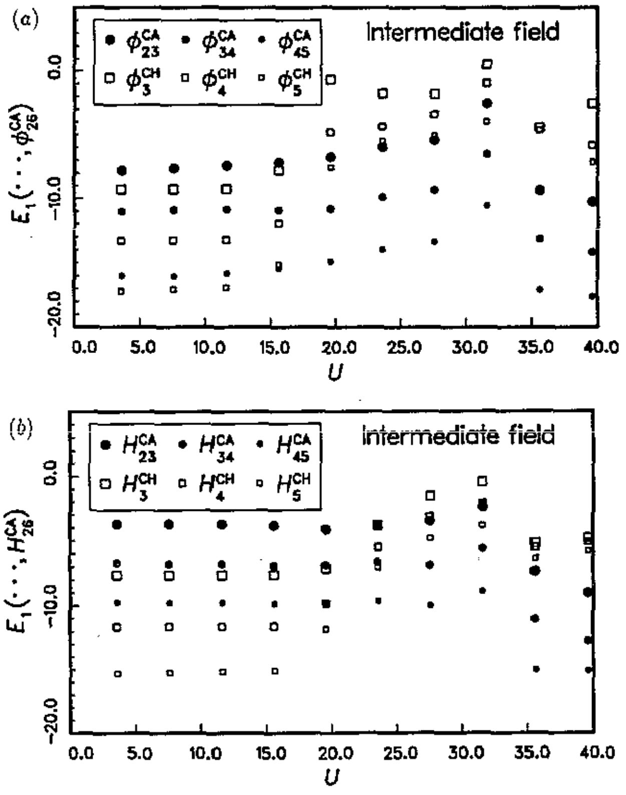
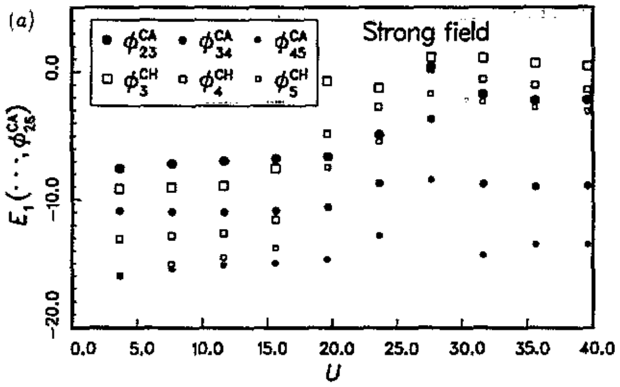


Figure 6. Results of the intermediate-field comparison of the CH and CA codes. The error measure $E_1(\cdots)$, defined in the text, involves a reference solution generated from multi-level extrapolation of CA values. The basic discretization scales for the three separate computations made with each code are $h_3 = \frac{1}{8}, h_4 = \frac{1}{16}$ and $h_5 = \frac{1}{32}$. At early retarded times, we have $O(h^4)$ convergence of ϕ^{CH}_i, H^{CH}_i and $\phi^{CA}_{ii'}$, and $O(h^3)$ convergence of $H^{CA}_{ii'}$. The very low levels of error in H^{CH}_i at the first four retarded times are particularly noteworthy. At later times (once the pulse of scalar field has reached $R = r = 0$), the errors at fixed resolution increase in *all* quantities, but proportionately more for the CH values. As a result, the CA values are significantly more accurate than their CH counterparts at late times. The convergence rate of the CA grid functions remains virtually constant throughout the calculation, while it appears that there is, at best, $O(h^2)$ convergence of ϕ^{CH}_i and H^{CH}_i after the pulse arrives at the origin.

convergence for H^{CA}. The latter behaviour, which is not completely understood at this time, differs from the behaviour observed in the linear calculations of the previous section and can be traced to the fact that the various radial 1-grids comprising the reference coordinates $\mathbf{R}_r(\mathrm{U}_l)$ are in general, non-uniform. At later values of retarded time ($U = 23.6, 27.6, 31.6$), and for any specific mesh-spacing, there is a clear decrease in the accuracy of the values from either code, relative to the corresponding early-time values. This is to be expected since it is along these null geodesics that non-linear effects are most predominant. In addition, at these values of U, there is significant variation in the dynamical variables at and near $R = r = 0$ which is where the truncation errors of both algorithms are likely to be largest. Returning to the question of convergence, it is clear from the plots that the convergence rates of the CA values remain quite steady through the entire calculations, whereas the CH convergences rates deteriorate at late retarded times and, in fact, become somewhat indeterminate with the measure we are using. At worst, we observe about 0.25% fractional error in $H_5^{\rm CH}$ compared with about 0.002% in $H_{45}^{\rm CA}$. Lest the reader miss an obvious point, we stress that these plots (as well as the corresponding strong-field graphs) provide strong evidence that both codes *are* converging to a unique solution.



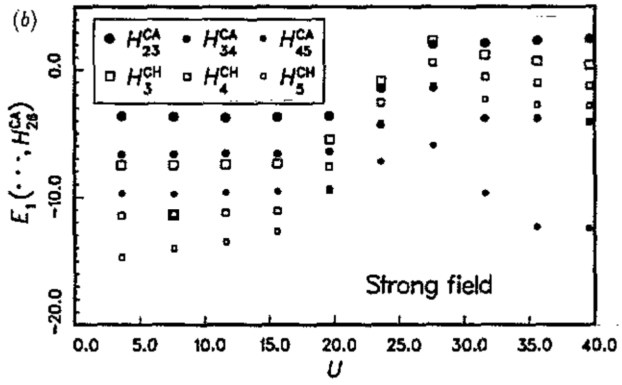


Figure 7. Results of the strong-field comparison of the CH and CA codes. The overall behaviour here, in terms of (1) the relative accuracy of the two algorithms at comparable resolutions; and (2) convergence rates of the various grid functions, is the same as for the intermediate field case. The deterioration in the convergence rate of $\phi_i^{\rm CH}$ and $H_i^{\rm CH}$ begins at earlier values of U in this computation—an improved treatment of boundary conditions in the CH code would likely remedy this situation to a large degree. Note that even in the non-linear regime, the convergence rates of $\phi_{ii'}^{\rm CA}$ and $H_{ii'}^{\rm CA}$ are $O(h^4)$ and $O(h^3)$, respectively.

Figure 7 shows the results from the strong-field computation. The basic features to which we wish to draw attention are essentially the same as for the previous calculation. In particular, the overall observed rates of convergence for the various grid functions before, during and after the non-linear period of the calculation are very similar to those measured in the intermediate-field computation. For the strong-field calculation, we obtain a maximum fractional error of about 1% in $H_5^{\rm CH}$ compared with about 0.01% in $H_{45}^{\rm CA}$ at the same value of U. We note that the strong-field computation is probably quite similar to the $A = 1.8$ (where A is another control parameter) calculation described in [9]. There, relative deviations in $2M(R,U)/R$ of about 3% were measured between $i = 3$ and $i = 4$ computations. This seems entirely consistent with the current observations. We also point out that, in this case, the late-time convergence of the CH values is a little more definitive, and we have even tried to Richardson-extrapolate the CH results based on a presumed second-order convergence (this makes the early time results *worse*, since at early times, we generally have $O(h^4)$ convergence). However, it is not at all clear that we can expect the CH results to be expandable in the sense defined in section 3.4 and the extrapolation does not lead to a significant improvement of the CH data. Finally, we note that after we first examined (graphically) event-by-event deviations of the late-time CH results from corresponding CA values, we were quickly able to isolate what is presumably the dominant source of error in the CH code—namely the use of the 'boundary condition' $\phi[R^0(\mathrm{U}),\mathrm{U}] \equiv \overline{H}[R^0(\mathrm{U}),\mathrm{U}] := H[R^0(\mathrm{U}),\mathrm{U}]$, which has $O(h)$ truncation error if, as appeared to be the case in these computations, $R_0(\mathrm{U}) = O(h)$.

Although figures 6 and 7 provide a good summary of the overall results of the comparison of CH and CA, it is interesting and instructive to examine some of the errors in the various grid functions as computed on an event-by-event basis. Specifically, for a generic grid function, Q^{C}, and a given retarded time, $U_l{}^n$, we compute the following deviation relative to the corresponding reference quantity Q_{26}^{CA}:

$$D_{\%}(U_l^n, Q^{\mathrm{C}}, Q_{26}^{\mathrm{CA}})[\mathbf{R}_r(U_l^n)] \equiv 100 \cdot \frac{\|(Q^{\mathrm{C}} - Q_{26}^{\mathrm{CA}})[\mathbf{R}_r(U_l^n), U_l^n]\|_1}{\|Q_{26}^{\mathrm{CA}}[\mathbf{R}_r(U_l^n), U_l^n\|_1}. \qquad (6.2)$$

Plots of some such values from the strong-field calculations are shown in figure 8. The rapid and regular convergence of both the CH and CA values for $U = 3.6, 7.6$ and 11.6 is again clearly visible (the convergence of H^{CH} is particularly noteworthy) as is the deterioration at later times in the characteristic, and to a lesser extent, Cauchy solutions.

Finally, it is also instructive to display similar plots of event-wise deviations generated from an earlier version of our comparison. The results, some of which are shown in figure 9(*a*), were quite confusing since both groups had strong, justifiable convictions that their respective results could be made arbitrarily accurate at early times. In the case of the CH code, the choice of variables and solution techniques made the algorithm manifestly accurate at such times, while for the CA code, the numerical results were evidently converging as anticipated. The somewhat obvious question which arose was why the CH calculations appeared to be converging (rapidly!) to something which was (1) *not* the exact solution and (2) not even smoothly related to the exact solution. In time, this puzzling phenomenon was traced to the fact that in one of the two steps of the comparison process where we actually transmitted data between the groups, we used a FORMAT statement (we coded exclusively in FORTRAN in this study) which did not provide enough precision in the specification of the radial coordinates. This turned out to be quite crucial, since these radial coordinate values, in the case of the $i = 2$, CH calculation, were exactly the values used as the reference coordinates, $\mathbf{R}_r(\mathrm{U}_l)$, for the comparison. The calculation proceeded under the assumption that the CH values were approximations to continuum quantities at *precisely* those radial coordinates which were produced with the offending FORMAT statement. The specific output format we used produced three digits to the right of the decimal place—we could then expect 'random' errors at roughly the 10^{-3} level in the CH grid functions, which is just what we observe. We feel the difference between the 'before' and 'after'

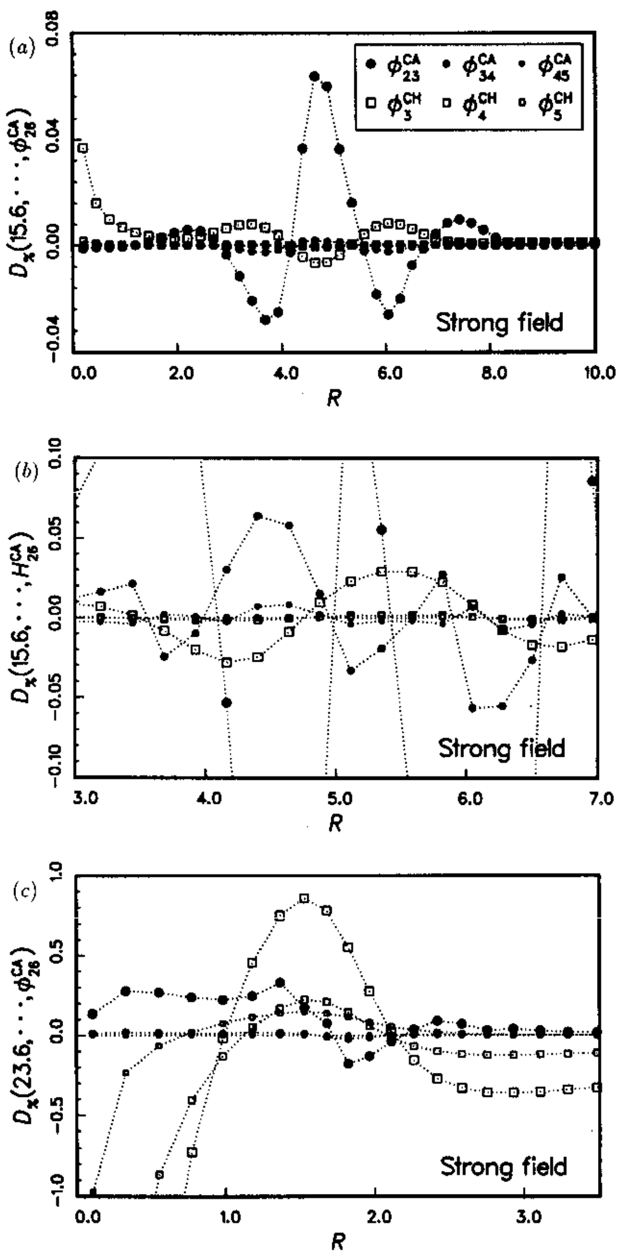
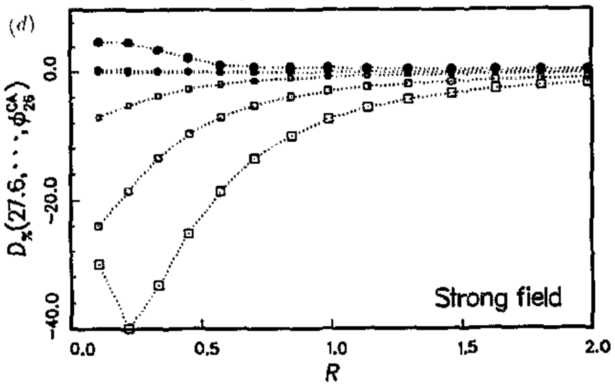


Figure 8. Some of the event-by-event deviations computed from the strong field CH and CA calculations. The error measure $D_{\%}(\cdots)$ is defined in the text. Note that only a portion of the radial domain is plotted in each case and that the finest mesh spacing used in these calculations was $h_5 = \frac{1}{32}$. The results shown in (*a*) and (*b*) are typical of the early-time ($U_l{}^n \leqslant 19.6$) convergence patterns observed in both the intermediate and strong field calculations. Although the increase in the level of error in the CH results at later times (particularly near $R = 0$ is quite evident in (*c*)–(*e*), it seems clear that the two codes produce results which converge to the same solution.

pictures (the data transferral was finally done with greater than 15 digits precision) is



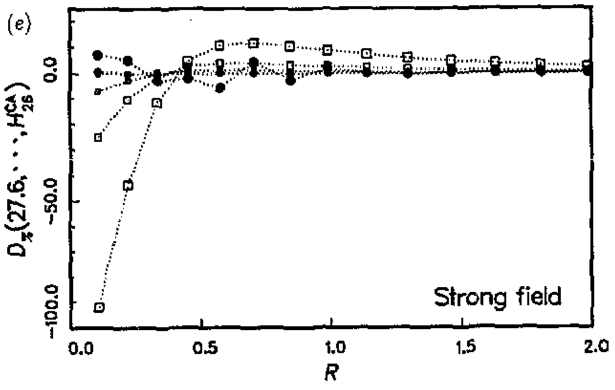


Figure 8. (Continued)

quite remarkable and provides yet another illustration of the level of error we have been able to detect as well as the utility of convergence tests.

7. Summary and discussion

We first summarize the computations we have described earlier and then proceed to some discussion of the implications of our study for the general problem of code testing in numerical relativity. We have set up and performed 'numerical test bed' [3] calculations which compare the output from two different numerical relativity codes. These codes (CH and CA), which treat the problem of spherically symmetric evolution of a minimally coupled, self-gravitating, massless scalar field, are based on different formalisms, use different coordinate systems, and employ different numerical solution methods. In order to 'directly' compare results from the two programs, we had to agree on a set of events at which to perform the comparison, and, since one of the functions (H) we compared was not a scalar, we also had to choose a reference coordinate system. Here, we always chose to compare solutions in the *characteristic* coordinate system and at a subset of the events used in some particular CH computation. This meant that we had to transform CA results into characteristic coordinates and, knowing that the CA output in (r, t) coordinates, could be Richardson-extrapolated,

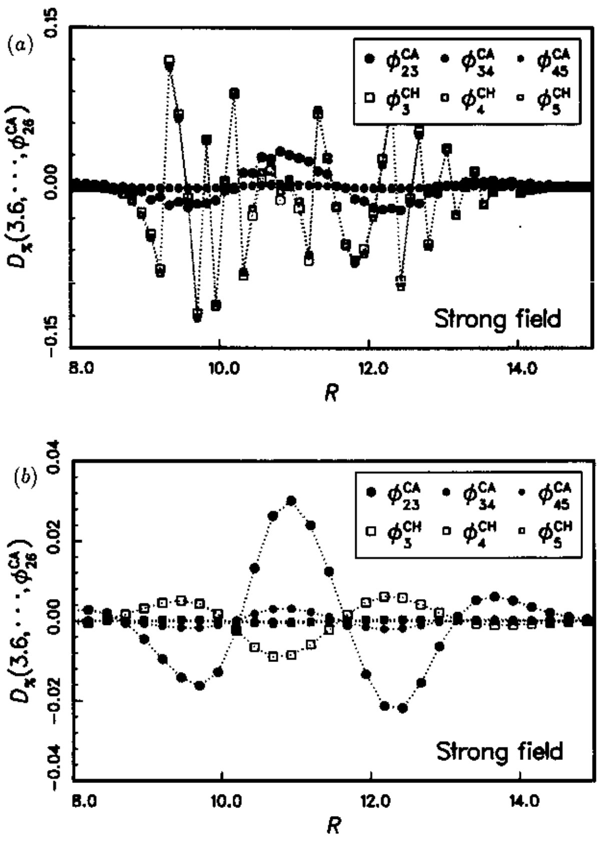


Figure 9. Event-wise deviations calculated before and after a problem with data transferral was corrected. As described in section 2.1, the characteristic code, CH, *generates* its grid structure, $\mathbf{R}(\mathbf{U})$, in the course of a calculation, and produces its output on some sub-grid, $\mathbf{R}(\mathbf{U}_l)$. In (*a*), the fractional parts of the coordinates in this sub-grid were rounded to three digits prior to the comparison calculations. In (*b*), no such rounding occurred. Refer to the text for more details.

we designed the transformation algorithm so that the transformed CA values could also be extrapolated in (R, U) coordinates. By contrast, the 'post-processing' of CH results was kept to a minimum and involved, at most, a spatial interpolation of the basic CH output.

The results of our comparison of the two codes consisted of measurements of the errors in the CH and (two-level extrapolated) CA results at several different resolutions, h_i. We selected initial data which generated spacetimes with significant self-gravitational effects (but no black holes) and, in the absence of exact solutions with which to determine the errors in the various computations, we used high-accuracy (effectively 'fully converged') numerical results which were computed via multi-level extrapolation of CA output. For both sets of initial data with which we worked, the results of the comparison indicated (1) that the two codes were converging to the *same* continuum solution (worst case deviations $\leqslant 1\%$ between the highest resolution CH and CA calculations); and (2) that, although the levels of error in the CA and CH results at a given resolution were quite comparable at early retarded times (with the exception of the particularly rapid convergence of H^{CH}), the CA values were significantly more accurate than the CH data once the pulse of scalar field had reached $R = 0$. Furthermore, the convergence testing allowed us to make *detailed* examination of the errors in the computations (even in the non-linear regime) and provided indications of how the CH algorithm might be improved.

In terms of the general issue of testing codes in numerical relativity, we again refer to the discussion of Centrella *et al* concerning the development of test-bed calculations. In discussing the treatment of problems which do not have 'analytic' solutions, these authors conclude that '(the) only method of insuring ... reliability is to run the same simulations on two or more independent codes'. Of course, this is precisely what we have done and in terms of being convinced of 'correctness', the psychological importance of having agreement of two codes cannot be denied. However, we wish to stress that the methods we have used in constructing our test-bed calculation are applicable to *a single code*. Here, we are saying nothing more than that, as numerical analysts, we should *expect* our numerical solutions to converge to the continuum solution as $h \to 0$. By performing convergence tests (which do not require extant numerical results) we should be able to establish whether we *have* convergence and, if we do, we should then be able to estimate the level of error in the solution. Provided the problems we study are sufficiently well posed, this should apply *even in highly dynamic, non-linear regimes*, and we feel that our work has demonstrated this point clearly. Now one can raise the question: 'Could not an incorrect difference equation or a correct discretization of a mistranscribed differential equation lead to an apparently convergent numerical procedure?' This indeeed seems possible and is a principal reason that the notion of comparison with an *independently generated solution* is so attractive. However, we point out that there are other routes to 'independent' checks which do not involve another approximate solution of the differential equations, just as one does not have to know how to integrate to verify that one function is the antiderivative of another. Another possible concern is that for more complex systems of equations than the ones we have studied here, it may be possible to extract interesting physics from a computation which for some reason (such as the nature of the discrete equations which are used or the maximum level of resolution which is available) does not produce a particularly clear signal of convergence. All we can say here is that in such instances it seems probable that the resolution dependence of the calculations will still allow us to estimate a *lower bound* for the error in the computations.

Finally, Shapiro [19] has recently pointed out that scalar field solutions, such as the ones considered here, can also be generated by general relativistic hydrodynamical codes and suggests that 'such solutions (of the scalar field equations) can be checked by spherical hydrodynamical codes'. We fully agree that '(it) would be useful to perform such checks', and look forward to reports of a hydro code capable of efficiently assessing the level of error in our most accurate scalar field results.

Acknowledgments

This work was supported by NSERC, BSF grant 86-00346 and by NSF grants PHY 82-17853 (supplemented by funds from NASA) and PHY 86-0324. The authors gratefully acknowledge the hospitality of the Institute for Theoretical Physics, Santa Barbara, CA where some of this research was carried out. Some of the computations described were performed using the resources of the Cornell National Superconducting Facility.

References

[1] Arnowitt R, Deser S and Misner C W 1962 *Gravitation—An Introduction to Current Research* ed L Witten (New York: Wiley)
[2] Bardeen J M and Piran T 1983 *Phys. Rep.* **96** 205–50
[3] Centrella J M *et al* 1986 *Dynamical Spacetimes and Numerical Relativity* ed J M Centrella (Cambridge: Cambridge University Press)
[4] Choptuik M W 1986 *PhD Thesis* University of British Columbia (unpublished)
[5] Choptuik M W 1988 *Phys. Rev.* D **44** 3124–35
[6] Christoudolou D 1986 *Commun. Math. Phys.* **105** 337; 1986 *Commun. Math. Phys.* **106** 587; 1987 *Commun. Math. Phys.* **109** 591
[7] Christoudolou D 1987 *Commun. Math. Phys.* **109** 613–47
[8] Dahlquist G and Björck A 1974 *Numerical Methods* (Englewood Cliffs, NJ: Prentice-Hall)
[9] Goldwirth D S and Piran T 1987 *Phys. Rev.* D **36** 3575–81
[10] Goldwirth D S, Ori A and Piran T 1989 *Frontiers in Numerical Relativity* ed C R Evans, L S Finn and D S Hobill (Cambridge: Cambridge University Press)
[11] Kreiss H-O and Oliger J 1973 *Methods for the Approximate Solution of Time Dependent Problems* (Garp Publications Series No 10) (Lanham, MD: Unipub)
[12] Marchuk G I 1982 *Methods of Numerical Mathematics* (New York: Springer)
[13] Misner C W, Thorne K S and Wheeler J A 1973 *Gravitation* (San Francisco: Freeman)
[14] Piran T 1980 *J. Comput. Phys.* **35** 254–83
[15] Piran T 1983 *Gravitational Radiation* ed N Deruelle and T Piran (Amsterdam: North-Holland) pp 203–56
[16] Press W H *et al* 1986 *Numerical Recipes* (Cambridge: Cambridge University Press)
[17] Richardson L F 1910 *Phil. Trans. R. Soc.* **210** 307–57
[18] Richtmyer R D and Morton K W 1957 *Difference Methods for Initial-Value Problems* 2nd edn (New York: Wiley)
[19] Shapiro S L 1989 *Phys. Rev.* D **39** 2839–47
[20] York J W 1979 *Sources of Gravitational Radiation* ed L L Smarr (Cambridge: Cambridge University Press)