PHY 387N: Relativity Theory II Spring 1998 Project 2
Due: Tuesday, March 10, 12:30pm

Note: *This assignment requires the use of several software packages, such as the* `vsxynt` *interface for I/O,* `ser/vs` *and/or* `scivis` *for visualization, and* `RNPL` *for the solution of PDEs using finite-difference techniques. On-line information concerning all of these packages is, or will shortly be, available via the Course Notes page. Note, however, that documentation for* `ser/vs` *is sparse to say the least. Feel free to see the instructor for a demonstration, and note that most Center for Relativity grad students can also serve as resources in this regard.*

Problem 1: Consider the following simple partial differential equation, which we will call the *one dimensional advection equation* or *1d advection equation* (we follow the convention that when discussing time-dependent PDEs, the time dimension is implicit, i.e. *1d* means one *spatial* dimension):

$$(\partial_t - \partial_x)\; u(x,t) = 0\,. \tag{1}$$

We wish to solve this PDE on the domain

$$\Omega = \{(x,t) \;\mid\; 0 \le x \le 1,\;\; 0 \le t \le T\}\;, \tag{2}$$

subject to the initial condition

$$u(x,0) = u_0(x)\,, \tag{3}$$

and periodic spatial "boundary condition"

$$u(0,t) = u(1,t)\,. \tag{4}$$

("boundary condition" is in quotes since we are solving the equation on a domain with topology $\mathbf{R} \times \mathbf{S^1}$ and $\mathbf{S^1}$ has no boundary). Note that for consistency, the initial data, $u_0(x)$, must also be periodic—$u_0(0) = u_0(1)$.

We generate an approximate solution to (1) using finite-difference techniques as follows. We first introduce a discrete domain (also known as a *mesh* or *grid*)

$$\Omega^h = \left\{(x_j, t^n) \;\mid\; 0 \le x_j \le 1,\;\; 0 \le t^n \le T\right\}\;, \tag{5}$$

where

$$\begin{aligned} x_0 &= 0 && (6)\\ x_{j+1} &= x_j + \triangle x && (7)\\ \triangle x &= h && (8)\\ h &= J^{-1}\,, && (9) \end{aligned}$$

and

$$\begin{aligned} t^0 &= 0 && (10)\\ t^{n+1} &= t^n + \triangle t && (11)\\ \triangle t &= \lambda\, h && (12)\\ T &= n_c J \triangle t\,, && (13) \end{aligned}$$

and further, where λ is an adjustable parameter of the difference scheme which we will generally consider in the range

$$0 < \lambda < 1\,, \tag{14}$$

$J = h^{-1}$ is one fewer than the total number of points in the discrete spatial domain $\{x_j\}$, and, finally, n_c is effectively the ratio of the number, N, of discrete times, t^n, to J. For $\lambda \approx 1$, n_c will be approximately the number of times a signal can cross the spatial domain in the interval of integration $0 \le t \le T$. It is crucial to note that, mathematically, in *any* finite-difference solution of a PDE, one is clearly most interested in the limit when mesh spacings tend to 0. Computationally, of course, it will generally be impossible to take this limit; however it will always be possible to compute the variation of a finite-difference solution as a function of resolution. We will refer to such computations as *convergence studies*: the convergence properties of the finite-difference scheme detailed below is a prime focus of this problem. As a final note in this regard, observe that the discrete domain (5) is characterized by a *single* mesh spacing (resolution), h: that is, whenever we perform (or think about performing) a convergence study, we *fix* λ at some specific value. Then, if $h \to h/2$, for example, we will have $\triangle r \to \triangle r/2$ *and* $\triangle t \to \triangle t/2$.

The second stage in the finite-difference solution of (1) is the *discretization* of the PDE itself. We introduce the grid function, $\hat{u}^n_j$, which is defined on Ω^h, and which is to approximate u^n_j, i.e.

$$\hat{u}^n_j \approx u^n_j \equiv u\left(x_j, t^n\right) , \tag{15}$$

where $u(x,t)$ satisfies (1). Note the somewhat subtle notation—the quantity without a caret satisfies a *continuum* equation, while the quantity with a caret satisfies a *difference* equation. We then approximate (1) using the so-called *leap-frog* scheme:

$$\frac{\hat{u}^{n+1}_j - \hat{u}^{n-1}_j}{2\triangle t} = \frac{\hat{u}^n_{j+1} - \hat{u}^n_{j-1}}{2\triangle x} , \tag{16}$$

which can be solved explicitly for the advanced unknowns, $\hat{u}^{n+1}_j$:

$$\hat{u}^{n+1}_j = \hat{u}^{n-1}_j + \lambda\left(\hat{u}^n_{j+1} - \hat{u}^n_{j-1}\right) . \tag{17}$$

The third and final ingredient in our finite-difference solution is the specification of initial data. The finite-difference scheme (16) is called a *three level* method since it couples unknowns on three "time levels": t^{n-1}, t^n, and t^{n+1}. Initialization thus requires the specification of data at $t^0 = 0$ and $t^1 = \triangle t$. We will postpone a discussion of a particular strategy for initialization until part (e).

We can reexpress (16) using finite-difference *operators* as fference *operators* as follows. Define Δ^h_x and Δ^h_t via

$$\Delta^h_t f(x,t) \equiv \frac{f(x, t+\triangle t) - f(x, t-\triangle t)}{2\triangle t} \equiv \frac{f(x, t+\lambda h) - f(x, t-\lambda h)}{2\lambda h} , \tag{18}$$

$$\Delta^h_x f(x,t) \equiv \frac{f(x+\triangle x, t) - f(x-\triangle x, t)}{2\triangle x} \equiv \frac{f(x+h, t) - f(x-h, t)}{2h} . \tag{19}$$

Then (16) can be written

$$\left(\Delta^h_t - \Delta^h_x\right) \hat{u}^n_j = 0 . \tag{20}$$

The operator notation motivates another view of the grid function, $\hat{u}^n_j$, which will be useful in your subsequent analysis. Specifically, we can view $\hat{u}$ as being defined *everywhere* on the continuum domain Ω (via a sufficiently high order (in h) interpolation, for example), rather than only on the discrete domain Ω^h. We then drop the "grid indices" and simply write

$$\left(\Delta^h_t - \Delta^h_x\right) \hat{u} = 0 , \tag{21}$$

or, changing notation slightly to make the finite-difference solution's h-dependence explicit:

$$\left(\Delta^h_t - \Delta^h_x\right) u^h = 0 . \tag{22}$$

Given the above preliminaries, answer and/or complete the following:

a) Write down the complete solution to (1)-(4). *Briefly* (two or three sentences) describe the nature of the solution.

b) Show that Δ_x^h and Δ_t^h have the following (formal) asymptotic expansions

$$\Delta_x^h = \partial_x + \frac{1}{6} h^2 \partial_{xxx} + O(h^4)\,, \tag{23}$$

$$\Delta_t^h = \partial_t + \frac{1}{6} \lambda^2 h^2 \partial_{ttt} + O(h^4)\,. \tag{24}$$

Use these results to show that the *truncation error*, τ^h, of the difference scheme, defined by:

$$\tau^h \equiv \left(\Delta_t^h - \Delta_x^h\right) u\,, \tag{25}$$

is second order, i.e. that

$$\tau^h = O(h^2)\,. \tag{26}$$

c) Following Richardson (1910), assume that the solution, u^h, of the difference equation (22) admits an asymptotic expansion of the form

$$u^h(x,t) = u(x,t) + h^2\, e_2(x,t) + h^4\, e_4(x,t) + O(h^6)\,, \tag{27}$$

or, more briefly,

$$u^h = u + h^2\, e_2 + h^4\, e_4(x,t) + O(h^6)\,. \tag{28}$$

where e_2, e_4 etc., are *h-independent* functions. Eventually, you should be able to justify the omission of odd-order (in h) terms in the expansion, but you may wish to delay such justification until you have worked through more of the problem. Substitute (28), (23) and (24) in (22) and demand that terms in the resulting equation vanish order by order in h. Show that the $O(1)$ part of the equation yields the original PDE (1), while the $O(h^2)$ part can be written as a PDE for the leading-order error function, e_2, in which u is viewed as a known function. Solve this PDE for the complete solution, $e_2(x,t)$, assuming exact initial conditions for the difference solution:

$$u^h(x,0) = u(x,0) = u_0(x)\,. \tag{29}$$

d) Assuming the validity of the Richardson expansion (28), show that, given two difference solutions, u^h and u^{2h}, computed at resolutions h and $2h$, respectively, one can estimate the leading-order error function, e_2, simply by subtracting the solutions and multiplying by an appropriate factor.

e) Write a `f77` (or `C` if you must!) program which implements the finite-difference algorithm defined above. Your program is to be called `advect1d` and is to have usage

```
usage: advect1d <level> <olevel> <nc> <lambda>
```

where

- `<level>` $\equiv$ discretization level. Controls the number of spatial mesh points, J, defined above, via $J = 2^{\texttt{<level>}} + 1$
- `<olevel>` $\equiv$ output level. Output is produced every $2^{\texttt{<level>}-\texttt{<olevel>}}$ time steps (see below).
- `<nc>` $\equiv$ approximate number of crossing times, as above.
- `<lambda>` $\equiv \lambda = \Delta t/\Delta x$, as above.

Note that <level>, <olevel> and <nc> are integer parameters, while <lambda> is real*8.

The command-line parameter <olevel> controls the frequency of program output which is to be performed using the vsxynt interface (documented in more detail via the Course Notes pages). Specifically, every $2^{\texttt{<level>}-\texttt{<olevel>}}$ time steps (including the 0th timestep, $t^0 = 0$), advect1d must produce output via vsxynt as follows

Function	Description	vsxynt name
u	Continuum (exact solution)	u<level>
u^h	Difference solution	uh<level>
$h^2 e_2$	Expected leading-order error	e<level>
e^h	Observed error: $e^h \equiv u^h - u$	eh<level>

For example, if level $= 6$, vsxynt names u6, uh6, e6 and eh6 should be used. Instructions for performing the required character string manipulations in f77 are available online. Note that the "Expected leading-order error" is h^2 times the quantity $e_2(x,t)$ for which you computed a general solution in part (c).

advect1d must use the following initial data

$$u^h(x,0) = u(x,0) = u_0(x) = \exp(-((x-0.5)/0.05)^2) \tag{30}$$

and must initialize both the t^0 and t^1 levels of data using the exact solution corresponding to this initial data. Disregard the fact that the above initial data is not precisely periodic. Also, use <lambda>$= 0.8$ in all of your "production" runs.

Demonstrate that your program has the expected behaviour in the limit of small h (large <level>)—i.e. show that your program is converging "properly" by comparing $h^2 e_2$ and e^h. I will leave it to you to determine what "small h" is in this context, but note that the compute time will be quadratic in <level>—maximum levels of 10 or so should suffice here. Present some typical evidence for convergence using .mpeg or .gif-sequence movies, as well as some summary evidence in the form of regular $X-Y$ plots. Prepare a Web page displaying your results, as well as a LaTeXed report as usual.

f) Use RNPL to solve (1) using the same finite-difference approximation and the same initial data as above. Your RNPL program should also be called advect1d (and naturally, therefore, should live in a distinct directory). Convergence test your program using the *intrinsic* technique suggested by the answer to part (d) and compare the results to those from your "hand-coded" version. How do you account for differences in the calculations, particularly at low resolutions? As before, run your RNPL-based calculations using <lambda>$= 0.8$. Use your Web page(s) for this assignment to display your results, but also include a complete discussion in your LaTeXed report.

Problem 2: *The Wave Equation on the Schwarzschild Background in Eddington-Finkelstein Coordinates*

2.1) *Preamble:*
In this problem, after the derivation and verification of some equations of motion and other results, you will use `RNPL`-generated finite-difference codes to study the spherically-symmetric dynamics of a massless scalar field on a Schwarzschild (black hole) *background*—i.e. the "back reaction" of the scalar field will be ignored, so the spacetime will be fixed, and completely known *a priori.* The chief physics which this model describes, and the physics on which you will focus, is the absorption and/or scattering of spherically-symmetric pulses ("S-waves") of scalar radiation which "infall" onto a black hole, and whose self-gravitation can be ignored. You will solve the problem in the 3+1 form of ingoing Eddington-Finkelstein coordinates (see e.g. MTW, 31.4 & Box 31.2 for a general discussion of Eddington-Finkelstein coordinates and corresponding line-elements), which will allow you to excise the interior of the black hole simply by limiting the spatial domain of integration to the region $r \geq 2M$. Because, the inner boundary, $r = 2M$, of the spatial domain is actually *null*, in principle *no special boundary conditions* are needed there for the scalar field—one simply applies the equations of motion (the covariant wave equation) up to and including $r = 2M$.

2.2) *The Wave Equation for a General, Static, Spherically Symmetric 3+1 Metric:*
Consider the following general 3+1 form of the static, spherically symmetric, vacuum metric (i.e. the Schwarzschild spacetime):

$$ds^2 = \left(-\alpha^2 + a^2\beta^2\right) dt^2 + 2a^2\beta\, dtdr + a^2 dr^2 + r^2 d\Omega^2 \,, \tag{31}$$

where $\alpha \equiv \alpha(r)$, $\beta = \beta(r)$ and $a = a(r)$. α is the lapse function as usual, while β is the radial component of the shift vector, i.e. $\beta^i = (\beta, 0, 0)$, $\beta_i \equiv \gamma_{ij}\beta^j = \left(a^2\beta, 0, 0\right)$.

Problem 2a) Show that the characteristics (null geodesics) of (31) are given by

$$\left(\frac{dr}{dt}\right)_{\pm} = -\beta \pm \frac{\alpha}{a} \,. \tag{32}$$

Problem 2b) Show that, given the metric (31), the massless Klein-Gordon equation (the wave equation):

$$\nabla^a \nabla_a \phi\,(r,t) = 0 \tag{33}$$

can be written as the pair of first-order-in-time (3+1, "Hamiltonian") equations

$$\begin{aligned}
\partial_t \Phi &= \partial_r \left(\beta\Phi + \frac{\alpha}{a}\Pi\right) \,, & (34)\\
\partial_t \Pi &= \frac{1}{r^2}\partial_r \left(r^2 \left(\beta\Pi + \frac{\alpha}{a}\Phi\right)\right) \,, & (35)
\end{aligned}$$

where

$$\begin{aligned}
\Phi\,(r,t) &\equiv \partial_r \phi \,, & (36)\\
\Pi\,(r,t) &\equiv \frac{a}{\alpha}\left(\partial_t \phi - \beta\, \partial_r \phi\right) \,. & (37)
\end{aligned}$$

2.3) *The Schwarzschild Solution in Ingoing Eddington-Finkelstein Coordinates*
Now consider the usual Schwarzschild form of (31), and bear in mind that, as discussed in the preamble, we will be restricting attention to $r \geq 2M$:

$$ds^2 = -\left(1 - \frac{2M}{r}\right) dt^2 + \left(1 - \frac{2M}{r}\right)^{-1} dr^2 + r^2 d\Omega^2 \,. \tag{38}$$

Recall from `PHY 387M` (and Wald 6.4), that defining the "Regge-Wheeler tortoise coordinate", $r_\star$,

$$r_\star \equiv r + 2M \ln\left(\frac{r}{2M} - 1\right) \,, \tag{39}$$

outgoing and ingoing null coordinates, u and v, respectively, of (38) are given by

$$u = t - r_\star\,, \tag{40}$$
$$v = t + r_\star\,. \tag{41}$$

Problem 2c) Show that if we adopt a timelike coordinate, $\tilde{t}$, based on the *ingoing* null coordinate, v, as follows

$$\tilde{t} = v - r = t + 2M\ln\left(\frac{r}{2M} - 1\right)\,, \tag{42}$$

then the Schwarzschild metric (38) takes the *ingoing Eddington-Finkelstein* (IEF) form

$$ds^2 = -\left(1 - \frac{2M}{r}\right)d\tilde{t}^2 + \frac{4M}{r}d\tilde{t}dr + \left(1 + \frac{2M}{r}\right)dr^2 + r^2 d\Omega^2\,. \tag{43}$$

Note that this form differs from the usually quoted IEF metric—for example, from MTW, Box 31.2, equation (2):

$$ds^2 = -\left(1 - \frac{2M}{r}\right)dv^2 + 2dvdr + r^2 d\Omega^2\,, \tag{44}$$

in that, in (43) we adopt a *timelike* coordinate, $\tilde{t}$, rather than the null coordinate, v. Relative to the original Schwarzschild form (38), we can summarize the IEF coordinates as follows:

- We maintain an *areal* radial coordinate, r—i.e. r continues to provide a direct measure of proper surface area.
- We choose our time coordinate, $\tilde{t}$, so that the *ingoing* tangent vector:

  $$\left(\frac{\partial}{\partial\tilde{t}}\right)^a - \left(\frac{\partial}{\partial r}\right)^a\,,$$

  is null.

Observe the key property of IEF coordinates—namely that, as is evident from (43), all metric components, $g_{\mu\nu}$ are perfectly well behaved, both on the horizon of the black hole, $r = 2M$, and in the exterior region, $r > 2M$.
Let t now and in the following denote IEF time, so that we have

$$ds^2 = -\left(1 - \frac{2M}{r}\right)dt^2 + \frac{4M}{r}drdt + \left(1 + \frac{2M}{r}\right)dr^2 + r^2 d\Omega^2\,. \tag{45}$$

Problem 2d) Show that, in terms of the general 3+1 form (31), we have

$$\alpha = \left(\frac{r}{r+2M}\right)^{1/2}\,, \tag{46}$$
$$a = \alpha^{-1} = \left(\frac{r}{r+2M}\right)^{-1/2}\,, \tag{47}$$
$$\beta = \frac{2M}{r+2M}\,. \tag{48}$$

2.4) *Asymptotics—Radiation Boundary Conditions*
As $r \to \infty$, the IEF metric (45) approaches the Minkowskii metric in spherical coordinates

$$ds^s = -dt^2 + dr^2 + r^2 d\Omega^2\,. \tag{49}$$

Problem 2e) Show that in this limit, the wave equation (33) can be written

$$\partial_{tt}\left(r\phi\right) = \partial_{rr}\left(r\phi\right)\,. \tag{50}$$

Clearly, the outgoing solution of (50) is

$$(r\phi)(r,t) = g(t-r) \ , \tag{51}$$

where g is an arbitrary function of one variable. Explicitly introducing the characteristic (wave) speed, c_+, associated with the outgoing solution (we have implicitly been assuming $c_+^2 = c_-^2 = c^2 = 1$), (51) becomes

$$(r\phi)(r,t) = g(c_+ t - r) \ , \tag{52}$$

Now, from Problem (2a), we have that, in the curved spacetime case

$$c_+ = -\beta + \frac{\alpha}{a} \, , \tag{53}$$

so that, asymptotically, we should expect

$$(r\phi)(r,t) = g\left(\left(-\beta + \frac{\alpha}{a}\right) t - r\right) \ , \tag{54}$$

which we can also express as

$$\partial_t (r\phi) + \left(-\beta + \frac{\alpha}{a}\right) \partial_r (r\phi) = 0 \, . \tag{55}$$

When solving the wave equation (34)-(35) on a *finite* spatial domain $2M \le r \le R$, we can impose (55) (and other equations derived using it) as a *boundary condition* at $r = R$. Such a condition is called an *outgoing radiation boundary condition*, or, often, a *Sommerfeld condition*.

2.5) *Initial Data*
With the wave equation written in the first order form (34)-(35), we must supply initial conditions

$$\Phi(r,0) = \Phi_0(r) \, , \tag{56}$$
$$\Pi(r,0) = \Pi_0(r) \, , \tag{57}$$

where $\Phi_0(r)$ and $\Pi_0(r)$ are arbitrary functions. Since we are most interested in studying the scattering of pulses of scalar radiation off of, and into, the black hole, we focus attention on data which, at the initial time, is "as ingoing as possible".
Assume that the initial configuration of the scalar field itself, $\phi(r,0) = \phi_0(r)$, describes some "pulse" shape—i.e. that $\phi_0(r)$ (effectively) has compact support—such as a "Gaussian"

$$\phi_0(r;\, A, r_0, \Delta) = A \exp\left(-\left(\frac{r-r_0}{\Delta}\right)^2\right) \, , \tag{58}$$

(where A, r_0 and Δ are adjustable parameters). Further make the approximation that

$$\partial_t (r\phi)(r,0) - \partial_r (r\phi)(r,0) = 0 \, . \tag{59}$$

This approximation is *exact* for an ingoing pulse as the support of the pulse $\to \infty$, and, for finite r, amounts to ignoring curvature–backscatter in attempting to set up *precisely* ingoing initial data.

Problem 2f) From (56), (57) and (59), derive initial conditions, $\Phi_0(r)$ and $\Pi_0(r)$, in terms of $\phi_0(r)$ and $d\phi_0(r)/dr$.

2.6) *Solution of the Equations of Motion*
Problem 2g) Write an RNPL program to solve the wave equation

$$\begin{aligned}
\partial_t \Phi &= \partial_r \left(\beta\Phi + \frac{\alpha}{a}\Pi\right) \, , \\
\partial_t \Pi &= \frac{1}{r^2}\partial_r \left(r^2 \left(\beta\Pi + \frac{\alpha}{a}\Phi\right)\right) \, , \\
\Phi(r,t) &\equiv \partial_r \phi \, , \\
\Pi(r,t) &\equiv \frac{a}{\alpha}\left(\partial_t \phi - \beta\, \partial_r \phi\right) \, ,
\end{aligned}$$

on the Schwarzschild background, and in IEF coordinates

$$\begin{aligned}\alpha &= \left(\frac{r}{r+2M}\right)^{1/2}, \\ a &= \alpha^{-1} = \left(\frac{r}{r+2M}\right)^{-1/2}, \\ \beta &= \frac{2M}{r+2M}.\end{aligned}$$

Take as the solution domain

$$2M \le r \le R \qquad 0 \le t \le T\,, \tag{60}$$

and use ingoing initial data and outgoing radiation boundary conditions as discussed above.

Once you have your program thoroughly tested (including convergence testing, which is virtually automatic with `RNPL`!), perform a parameter space survey using the Gaussian initial data (58), where Δ (the width of the pulse) is varied, while A and r_0 are held fixed. Derive an expression for a conserved mass in the model, i.e. find a function $m(r,t)$ such that, in cases where no scalar field falls down the black hole, we have

$$\lim_{r\to\infty} m\,(r,t) = \text{constant}\,. \tag{61}$$

Of course, in cases where some of the scalar radiation *does* fall into the hole, the mass on $r \ge 2M$ will not be conserved, and you will be able to compute the amount of mass which falls in. The discussion of your results should *include* a plot of fractional mass absorption as a function of pulse width (relative to the size of the black hole), as well as `.mpeg` or `.gif`–sequence movies of representative results. Document your findings as thoroughly as time permits, both in your project Web pages(s), and in your LaTeXed report.

2.7) *Hints and Further Instructions*

You may wish to start with an `RNPL` program for the solution of the flat-space, Cartesian wave equation

$$\partial_{tt}\phi\,(x,t) = \partial_{xx}\phi\,(x,t)\ , \tag{62}$$

cast in first order form

$$\partial_t\Phi = \partial_x\Pi\,, \tag{63}$$

$$\partial_t\Pi = \partial_x\Phi\,, \tag{64}$$

where

$$\Phi\,(x,t) \equiv \partial_x\phi\,, \tag{65}$$

$$\Pi\,(x,t) \equiv \partial_t\phi\,, \tag{66}$$

solved on the domain

$$0 \le x \le 1 \qquad 0 \le t \le T \tag{67}$$

with radiation conditions imposed at *both* $x = 0$ and $x = 1$. Use leap-frog differencing (as in Problem 1) for (63) and (64), and "Kreiss–Oliger" dissipation as discussed in "`ad`: *An Implementation of the Berger-Oliger Mesh Refinement Algorithm for the Wave Equation in Spherical Symmetry*", available on-line. Also refer to that reference for suggestions on finite-differencing radiation boundary conditions. Such a program should be relatively easy to get going, and will give you further experience with `RNPL`, the visualization tools, etc.

For the "real" problem, I suggest you follow the differencing approach just outlined closely. Remember that the goal here isn't to come up with the "best" finite-difference approximation—rather the aim is to, as quickly and confidently as possible, simulate and analyze the physics of scalar wave interactions with a black hole in spherical symmetry.