Building a Numerical Model of the Influence of a Gravitational Wave on an Electromagnetic Four-Potential

The Physics behind
`PhotonGravityWavePerpendicularCase(2+1)-D.py`,
`PhotonGravityWaveInefficient(2+1)-D.py`,
`PhotonGravityWave(2+1)-D.py`.

Varadarajan Srinivasan

Note: All of the following is done in full 3+1 dimensions, but the implementation in Python is in (2+1)-D because that makes the resulting animated visualizations less cumbersome to display, greatly increases the resolution-to-runtime ratio, and suffices to show the desired effects. Nevertheless, the program is structured in such a way that extending it to (3+1)-D would be quick and straightforward.

Let us define a left-handed[1] coordinate system in the presence of an electromagnetic field. Consider a gravitational wave propagating along the z-axis. Let us further define the x- and y-axes to be along the polarization axes of the gravitational wave. This has the effect of stretching and contracting space along those 2 axes. This affects Maxwell's Equations such that the wave equation for a component u of the electromagnetic four-potential in the influence of a gravitational wave can be written (separating the unaffected time component) as

$$\begin{pmatrix}\partial_x & \partial_y & \partial_z\end{pmatrix} \mathbf{M} \begin{pmatrix}\partial_x \\ \partial_y \\ \partial_z\end{pmatrix} u = \frac{1}{c^2}\frac{\partial^2 u}{\partial t^2}, \tag{1}$$

where $\mathbf{M}$ is the space component of the metric tensor and is represented by

$$\mathbf{M} = \begin{pmatrix} 1+f & 0 & 0 \\ 0 & 1-f & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad \text{where } f = \epsilon\cos(kz - kct). \tag{2}$$

Note that when the amplitude ϵ is 0, Eq. 1 reduces to the standard wave equation with no gravitational wave (i.e. flat spacetime),

$$\nabla^2 u = \frac{1}{c^2}\frac{\partial^2 u}{\partial t^2}. \tag{3}$$

Let us generalize this for any inclination of the gravitational wave. We need not generalize for azimuthal angle because we can always define a planar axis along the azimuthal direction of the wave. First, let us define a "stationary" left-handed coordinate system $\langle x\prime, y\prime, z\prime\rangle$ such that the polarization axes of the gravitational wave are parallel to the $x\prime$ and $y\prime$ axes and it propagates with velocity $c\hat{z}\prime$. Second, we must also define a left-handed coordinate system $\langle X, Y, Z\rangle$ fixed to our frame of interest whose XY plane contains those electromagnetic waves. These frames are stationary with respect to each other. $\langle X, Y, Z\rangle$ is the fixed frame our computational algorithms will use; the rows and columns of the numerical method form the discretized representation of the plane. The row and column indices correspond to Y- and X-values respectively with (0,0) as the top-left corner. Note that our initial coordinate system $\langle x, y, z\rangle$ was equivalent to both $\langle X, Y, Z\rangle$ and $\langle x\prime, y\prime, z\prime\rangle$ as that was the case where the gravitational wave is perpendicular to the plane. In our new frame, our tensor $\mathbf{M}$ takes $f = \epsilon\cos(kz\prime - kct)$ with t unchanged.

We know that we can always find a rotation matrix $\mathbf{R}$ such that

$$\begin{pmatrix} X \\ Y \\ Z\end{pmatrix} = \mathbf{R}\begin{pmatrix} x\prime \\ y\prime \\ z\prime\end{pmatrix}. \tag{4}$$

[1] This orientation is more convenient for our computational system because its natural 2D origin is at the top-left corner.

Since the rotation matrix must be orthogonal, $\mathbf{R}^{-1} = \mathbf{R}^T$. Therefore, transposing both sides of Eq. 4 shows that $\begin{pmatrix} X & Y & Z \end{pmatrix} = \begin{pmatrix} x\prime & y\prime & z\prime \end{pmatrix} \mathbf{R}^{-1}$. We can now rewrite the left hand side of Eq. 1 as

$$LHS = \begin{pmatrix} \partial_{x\prime} & \partial_{y\prime} & \partial_{z\prime} \end{pmatrix} \mathbf{R}^{-1}\mathbf{R}\ \mathbf{M}\ \mathbf{R}^{-1}\mathbf{R} \begin{pmatrix} \partial_{x\prime} \\ \partial_{y\prime} \\ \partial_{z\prime} \end{pmatrix} u$$
$$= \begin{pmatrix} \partial_X & \partial_Y & \partial_Z \end{pmatrix} \mathbf{R}\ \mathbf{M}\ \mathbf{R}^{-1} \begin{pmatrix} \partial_X \\ \partial_Y \\ \partial_Z \end{pmatrix} u. \tag{5}$$

Let us define another tensor $\mathbf{T} \equiv \mathbf{RMR}^{-1}$, the conjugate of $\mathbf{M}$ by rotation. We can characterize the rotation required to go from the gravitational wave coordinate system to our computational coordinate system as Euler angle rotations about $z\prime, x\prime, z\prime$, but we have enough freedom in our coordinates to be able to always orient our $\langle X, Y, Z\rangle$ such that the last roll rotation about the new $z\prime$ axis does nothing. Thus, for an angle of incidence of θ,[2]

$$\mathbf{R} = \mathbf{R}_{z\prime}\mathbf{R}_{x\prime} = \begin{pmatrix} \cos(\theta) & -\cos(\theta)\sin(\theta) & \sin^2(\theta) \\ \sin(\theta) & \cos^2(\theta) & -\cos(\theta)\sin(\theta) \\ 0 & \sin(\theta) & \cos(\theta) \end{pmatrix} \tag{6}$$

and hence

$$\mathbf{T} = \begin{pmatrix} \frac{1}{8}(4f\cos(2\theta) + f\cos(4\theta) + 3f + 8) & \frac{1}{8}f(6\sin(2\theta) + \sin(4\theta)) & f\sin^2(\theta)\cos(\theta) \\ \frac{1}{8}f(6\sin(2\theta) + \sin(4\theta)) & \frac{1}{8}(-8f\cos(2\theta) - f\cos(4\theta) + f + 8) & -f\sin(\theta)\cos^2(\theta) \\ f\sin^2(\theta)\cos(\theta) & -f\sin(\theta)\cos^2(\theta) & \frac{1}{2}(f\cos(2\theta) - f + 2) \end{pmatrix} \tag{7}$$

We have introduced the parameter θ to convert between our two coordinate systems as our computations require all terms to be in the $\langle X, Y, Z\rangle$ system, but recall that $f = f(t, z\prime) = \epsilon\cos(kz\prime - kct)$. We therefore need to solve Eq. 4 for the primed column vector with $\mathbf{R}$ as above about $x\prime$,

$$\begin{pmatrix} x\prime \\ y\prime \\ z\prime \end{pmatrix} = \mathbf{R}^T \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{pmatrix} X\cos(\theta) + Y\sin(\theta) \\ \frac{1}{2}(-X\sin(2\theta) + Y\cos(2\theta) + 2Z\sin(\theta) + Y) \\ X\sin^2(\theta) + \cos(\theta)(Z - Y\sin(\theta)) \end{pmatrix}$$

Thus,

$$f = f(t, X, Y, Z) = \epsilon\cos\big(k\big(X\sin^2(\theta) + \cos(\theta)\big(Z - Y\sin(\theta)\big) - ct\big)\big). \tag{8}$$

This puts $LHS = \nabla^T\mathbf{T}\nabla u$, the compact form of Eq. 5, entirely in terms of our computational coordinates X, Y, Z. Hence, we have the desired form of the generalized (3+1)-dimensional equation describing $u(t, X, Y, Z)$, a component of the electromagnetic four-potential interacting with a gravitational wave.

$$\boxed{\nabla^T\mathbf{T}\nabla u = \frac{1}{c^2}\frac{\partial^2 u}{\partial t^2} \quad \text{where } f = f(t, X, Y, Z)} \tag{9}$$

with $f(t, X, Y, Z)$ given by Eq. 8.

To numerically compute results, we must discretize Eq. 9. Our process for this will build on that described in WaveEquation.pdf. We define a numerical function[3] $u_{m,n,o,t}$ where $m, n, o,$ and t are the row, column, depth-step, and time-step indices, respectively. This means that, in our graphically conveniently left-handed system, $m, n, o,$ and t are the discretizations of the Y-, X-, Z-, and t-coordinates, respectively. Let

[2]Note that θ is defined clockwise when viewed from above the YZ plane because our coordinate system is left-handed.

[3]If we program in only 2+1 dimensions, we would ignore the third spatial index and Z-derivatives in all of these equations. This would also mean that $f = f(t, X, Y, 0) = \epsilon\cos\big(k\big(X\sin^2(\theta) - Y\cos(\theta)\sin(\theta) - ct\big)\big)$.

us generalize the numerical first derivative from WaveEquation.pdf (using the notation and names defined there) to numerical partial derivatives. We define nD such that, for a function $g(t, X, Y, Z)$,

$$\begin{aligned} \mathrm{nD}_X(g_{m,n,o,t}) &\equiv \frac{g_{m,n+1,o,t} - g_{m,n-1,o,t}}{2\Delta} \approx \frac{\partial g}{\partial X}, \\ \mathrm{nD}_Y(g_{m,n,o,t}) &\equiv \frac{g_{m+1,n,o,t} - g_{m-1,n,o,t}}{2\Delta} \approx \frac{\partial g}{\partial Y}, \\ &\text{and similarly for } Z \text{ (index } o\text{, step size } \Delta) \text{ and } t \text{ (index } t\text{, step size } \Delta t). \end{aligned} \tag{10}$$

Using Eq. 7, we know that the left hand side of our main result, Eq. 9, is given by

$$\begin{aligned} LHS = &\frac{\partial}{\partial X}\left[\frac{1}{8}(4f\cos(2\theta) + f\cos(4\theta) + 3f + 8)\frac{\partial u}{\partial X} + \frac{1}{8}f(6\sin(2\theta) + \sin(4\theta))\frac{\partial u}{\partial Y} + f\sin^2(\theta)\cos(\theta)\frac{\partial u}{\partial Z}\right] \\ &+ \frac{\partial}{\partial Y}\left[\frac{1}{8}f(6\sin(2\theta) + \sin(4\theta))\frac{\partial u}{\partial X} + \frac{1}{8}(-8f\cos(2\theta) - f\cos(4\theta) + f + 8)\frac{\partial u}{\partial Y} - f\sin(\theta)\cos^2(\theta)\frac{\partial u}{\partial Z}\right] \\ &+ \frac{\partial}{\partial Z}\left[f\sin^2(\theta)\cos(\theta)\frac{\partial u}{\partial X} - f\sin(\theta)\cos^2(\theta)\frac{\partial u}{\partial Y} + \frac{1}{2}(f\cos(2\theta) - f + 2).\right. \end{aligned} \tag{11}$$

Now, we discretize Eq. 11 in terms of our previously defined numerical derivatives. For later convenience, let us multiply Eqs. 10 by the spatial step size Δ and multiply the discretized form of Eq. 11 by Δ^2 so as to put LHS in terms of two functions, $\Delta \cdot \mathrm{nD}_X$ and $\Delta \cdot \mathrm{nD}_Y$, independent of parameters. This gives us

$$\begin{aligned} \Delta^2 \cdot LHS_{m,n,o,t} = \Delta \cdot \mathrm{nD}_X\Big[&\frac{1}{8}(4f_{m,n,o,t}\cos(2\theta) + f_{m,n,o,t}\cos(4\theta) + 3f_{m,n,o,t} + 8)\Delta \cdot \mathrm{nD}_X(u_{m,n,o,t}) \\ &+ \frac{1}{8}f_{m,n,o,t}(6\sin(2\theta) + \sin(4\theta))\Delta \cdot \mathrm{nD}_Y(u_{m,n,o,t}) \\ &+ f_{m,n,o,t}\sin^2(\theta)\cos(\theta)\Delta \cdot \mathrm{nD}_Z(u_{m,n,o,t})\Big] \\ +\Delta \cdot \mathrm{nD}_Y\Big[&\frac{1}{8}f_{m,n,o,t}(6\sin(2\theta) + \sin(4\theta))\Delta \cdot \mathrm{nD}_X(u_{m,n,o,t}) \\ &+ \frac{1}{8}(-8f_{m,n,o,t}\cos(2\theta) - f_{m,n,o,t}\cos(4\theta) + f_{m,n,o,t} + 8)\Delta \cdot \mathrm{nD}_Y(u_{m,n,o,t}) \\ &- f_{m,n,o,t}\sin(\theta)\cos^2(\theta)\Delta \cdot \mathrm{nD}_Z(u_{m,n,o,t})\Big] \\ +\Delta \cdot \mathrm{nD}_Z\Big[&f_{m,n,o,t}\sin^2(\theta)\cos(\theta)\Delta \cdot \mathrm{nD}_X(u_{m,n,o,t}) - f_{m,n,o,t}\sin(\theta)\cos^2(\theta)\Delta \cdot \mathrm{nD}_Y(u_{m,n,o,t}) \\ &+ \frac{1}{2}(f_{m,n,o,t}\cos(2\theta) - f + 2)\Delta \cdot \mathrm{nD}_Z(u_{m,n,o,t})\Big]. \end{aligned} \tag{12}$$

For some uses, an explicit expression without nested functions will be necessary. We achieve this by applying Eqs. 10 distributively[4] throughout Eq.12. Computing through and evaluating at time $t-1$ produces (written,

[4]It is easy to check that this useful trick is allowed because $\mathrm{nD}(kg) = k\mathrm{nD}(g)$ for scalar k as nD is, in principle, a linear transformation.

for brevity, in terms of the components of $\mathbf{T}(f(t,X,Y,Z))$, given by Eq. 7)

$$
\begin{aligned}
\Delta^2 \cdot LHS_{m,n,o,t-1} = \frac{1}{4}\Big[& T_{0,0_{m,n,o,t-1}}(u_{m,n+2,o,t-1} - 2u_{m,n,o,t-1} + u_{m,n-2,o,t-1}) \\
& + T_{0,1_{m,n,o,t-1}}(u_{m+1,n+1,o,t-1} - u_{m+1,n-1,o,t-1} - u_{m-1,n+1,o,t-1} + u_{m-1,n-1,o,t-1}) \\
& + T_{0,2_{m,n,o,t-1}}(u_{m,n+1,o+1,t-1} - u_{m,n-1,o+1,t-1} - u_{m,n+1,o-1,t-1} + u_{m,n-1,o-1,t-1}) \\
& + T_{1,0_{m,n,o,t-1}}(u_{m+1,n+1,o,t-1} - u_{m-1,n+1,o,t-1} - u_{m+1,n-1,o,t-1} + u_{m-1,n-1,o,t-1}) \\
& + T_{1,1_{m,n,o,t-1}}(u_{m+2,n,o,t-1} - 2u_{m,n,o,t-1} + u_{m-2,n,o,t-1}) \\
& + T_{1,2_{m,n,o,t-1}}(u_{m+1,n,o+1,t-1} - u_{m-1,n,o+1,t-1} - u_{m+1,n,o-1,t-1} + u_{m-1,n,o-1,t-1}) \\
& + T_{2,0_{m,n,o,t-1}}(u_{m,n+1,o+1,t-1} - u_{m,n+1,o-1,t-1} - u_{m,n-1,o+1,t-1} + u_{m,n-1,o-1,t-1}) \\
& + T_{2,1_{m,n,o,t-1}}(u_{m+1,n,o+1,t-1} - u_{m+1,n,o-1,t-1} - u_{m-1,n,o+1,t-1} + u_{m-1,n,o-1,t-1}) \\
& + T_{2,2_{m,n,o,t-1}}(u_{m,n,o+2,t-1} - 2u_{m,n,o,t-1} + u_{m,n,o-2,t-1})\Big] \\
= \frac{1}{4}\Big[& T_{0,0m,n,o,t-1}(u_{m,n+2,o,t-1} - 2u_{m,n,o,t-1} + u_{m,n-2,o,t-1}) \\
& + T_{1,1m,n,o,t-1}(u_{m+2,n,o,t-1} - 2u_{m,n,o,t-1} + u_{m-2,n,o,t-1}) \\
& + T_{2,2m,n,o,t-1}(u_{m,n,o+2,t-1} - 2u_{m,n,o,t-1} + u_{m,n,o-2,t-1})\Big] \\
+ \frac{1}{2}\Big[& T_{0,1m,n,o,t-1}(u_{m+1,n+1,o,t-1} - u_{m+1,n-1,o,t-1} - u_{m-1,n+1,o,t-1} + u_{m-1,n-1,o,t-1}) \\
& + T_{0,2m,n,o,t-1}(u_{m,n+1,o+1,t-1} - u_{m,n-1,o+1,t-1} - u_{m,n+1,o-1,t-1} + u_{m,n-1,o-1,t-1}) \\
& + T_{1,2m,n,o,t-1}(u_{m+1,n,o+1,t-1} - u_{m-1,n,o+1,t-1} - u_{m+1,n,o-1,t-1} + u_{m-1,n,o-1,t-1})\Big],
\end{aligned} \tag{13}
$$

since each t-slice of $\mathbf{T}$ is symmetric.

The right hand side of Eq. 9 is easily discretized in time. Equating that to LHS gives us the fully discretized form of Eq. 9,

$$\frac{1}{\Delta^2}\ \Delta^2 \cdot LHS_{m,n,o,t-1} = \frac{1}{c^2}\ \mathrm{nD}_t(\mathrm{nD}_t(u_{m,n,o,t-1})). \tag{14}$$

Finally, applying Eqs. 10 to Eq. 14 and collecting parameters on one side, we have our desired iterative equation that solves the wave equation in the presence of a gravitational wave.

$$\boxed{u_{m,n,o,t} = \left(c\ \frac{\Delta t}{\Delta}\right)^2 \Delta^2 \cdot LHS_{m,n,o,t-1} + 2u_{m,n,o,t-1} - u_{m,n,o,t-2}} \tag{15}$$

with $\Delta^2 \cdot LHS_{m,n,o,t-1}$ given by Eq. 13.

The manipulation that allowed us to program the numerical partial derivative functions independently of parameters now gives rise to the accumulated parameter

$$K \equiv \left(c\ \frac{\Delta t}{\Delta}\right)^2. \tag{16}$$

This is the same $K = c^2/c_i^2$ that we discussed in detail in WaveEquation.pdf. Its constraint conditions apply here as well.