Simulating wave propagation with the optical Fourier engine

[*17 minute read*]

Introduction

It's no exaggeration to say that the concept of a wave is one of the most useful ideas in engineering and physics. Waves are so omnipresent in the world around us that just about everyone has an idea of what they are. Some might think of the waves on the ocean surface, while others might think of their use in the vast network of wireless transmitters and receivers that surrounds most of us in daily life. The ability to work out how waves travel, convey information from one point to another, or dissipate away, is essential to their practical use. However, in most cases, working this out isn't a trivial task.

A particularly powerful and common tool for studying the evolution of waves is the Fourier transform. In short, it involves decomposing a function (real or complex) into simple waves which can be manipulated individually, a much easier task. The Fourier transform is used throughout the technical world

At Optalysys, we are developing a Fourier optical computing system that combines silicon photonics with free-space optics, allowing us to perform Fourier transforms and convolutions more efficiently, by orders of magnitude, than CPUs and GPUs can. In this article, we report experimental results obtained with our first silicon photonics demonstrator showing that wave propagation can be simulated accurately using an optical device. To the best of our knowledge, this is the first experimental demonstration of a silicon photonic system in simulating wave propagation.

Different kinds of waves

We've already mentioned a couple of generic examples of waves, but these hardly start to describe the wide range of phenomena they are involved in.

Formally, a 'wave' can be defined as a propagating variation of one or more quantities, often with respect to an equilibrium value. This general definition describes very different phenomena, such as:

- Ocean surface waves, mainly generated by wind or geological events and propagating in the upper layer of the ocean.
- Sound waves, making up everything we can hear (and much more).
- Light, which may be seen as a perturbation of the electro-magnetic field.
- Seismic waves, propagating either on the surface or in the interior of the Earth.
- Gravitational waves: disturbances of space-time itself predicted by Einstein's General Relativity and first observed in 2015 by the LIGO scientific collaboration. (This observation earned Rainer Weiss, Kip Thorne, and Barry Barish the 2017 Nobel Prize in physics.)
- The wave-function associated with a particle in quantum mechanics, and perturbations of quantized fields in quantum field theories.

This list is, of course, far from exhaustive. While each kind of wave has its own specific properties and quirks, they also share many common features and can be described, at least to some extent, by similar or identical mathematical models. Needless to say, simulating these models has important practical applications. Beyond the theoretical interest of understanding phenomena where waves occur, it allows us, for instance, to quickly estimate when and where a tsunami may strike after the detection of sub-oceanic geological activity. To give another scenario with decidedly lower-stakes, wave analysis may guide the design of concert halls to ensure an optimal acoustic quality from any seat.

Wave propagation and free-space optics

The wave propagation problem consists in finding how waves evolve in time. It is one of the most important questions in the study of waves. For instance, going back to the tsunami example given above, if you detect geological activity below the ocean floor then you want to know how seismic and oceanic waves will propagate away from the epicentre to determine if there is a risk the tsunami may strike inhabited land and, if yes, when, where, and with what amplitude. In very few cases, this problem can be solved with as little as pen and paper by writing down some equations and a few lines of algebra. However, these cases are mostly of academic interest. In practice, most situations are far too complicated for that. Numerical simulations on computers are then the only way to reliably estimate how waves propagate.

These numerical simulations can require a lot of compute power, especially when a high precision is needed or when some uncertainty about the initial conditions or medium in which the waves propagate (for instance, in the seismic wave example, there may be some uncertainty about the precise location of the epicentre) require that many simulations are performed to account for all possibilities. As we will explain below, the Fourier transform is often a major bottleneck in these computations. Fortunately, it is possible to make this operation considerably faster and more efficient. The solution is free-space optics.

A broader look at digital signal processing

The study of wave propagation falls in the wider field of digital signal processing (DSP). This is a huge domain with a broad range of applications in:

- Communication systems, where it is used for modulation and demodulation, channel equalization, and echo or background noise cancellation,
- Consumer electronics (notably in audio and video encoding / decoding and speech synthesis / recognition),
- In the music industry (to generate audio effects, reduce background noise, or in synthetic instruments),
- Medical diagnosis (magnetic-resonance imaging, ultrasonic imaging, computer tomography,...),
- Geophysics (mostly in seismology, studies of the Earth upper layers, and oil exploration),
- Astronomy,
- Aviation (through radar and radio navigation),
- Information security (*e.g.* for steganography and digital watermarking),

... in fact, just about any application that requires us to work with signals produced by sensors.

While we will focus on wave propagation in this article, the concepts and tools we demonstrate are more widely applicable to DSP. In particular, the Fourier transform is one of the main ingredients of many DSP applications. The results we show for the optical Fourier engine can thus be generalized to numerous applications beyond wave propagation.

A bit of theory on wave propagation

Some assumptions

In this article, we consider a relatively simple model of wave propagation, which shows the concepts and techniques as directly as possible. Specifically, we make the following assumptions:

- We work in two space dimensions (instead of the usual three). This assumption is mostly made to ease visualisation: it is far easier to see how a wave propagates on a two-dimensional surface than inside a three-dimensional volume.
- We assume the medium in which the waves propagate is homogeneous. The techniques shown below can be extended to heterogeneous media, but it can make them more involved.
- We neglect interactions between waves. The method we follow can also be extended to include interactions, but, again, at the price of a higher complexity. Moreover, many waves we are used to, like sound waves and light, show very little self-interaction. (A prominent counter-example is surface gravity waves in water, whose self-interactions lead to the common "wave breaking" phenomenon that you can see at any beach.)
- Finally, we will make an assumption about the mathematical form of the differential equation describing the wave. In short, it amounts to assuming there is only one 'type' of wave. Seismic waves, for instance, come in different types: compressional P-waves, transverse S-waves, Rayleigh waves, Love waves, and Stoneley waves. The general form of the equation we use could describe the propagation of any one of these types of waves, although describing them all together requires a more complicate model.

A simple mathematical model

Mathematically, these waves can be described by a complex field, *i.e.*, a function associating a complex number to each point of space and time. In the following, we call this function ϕ. Physical quantities, like the pressure for sound waves, can be obtained by taking the real or imaginary parts of this field (or some function of them).

The field ϕ obeys a partial differential equation of the form

$$\mathrm{i}\partial_t \phi = f(-\mathrm{i}\partial_x, -\mathrm{i}\partial_y)\phi,$$

where f is some function, i denotes a complex number such that $\mathrm{i}^2 = -1$, t is the time coordinate, and x and y are two coordinates in the spatial directions. (There are some technical conditions on the function f for this equation to be well defined.) This equation relates the rate of change of the field ϕ describing the wave, on the left of the equal sign, to some function of its space derivatives on the right.

Plane wave solutions

An interesting property of this equation is that it has some (actually, an infinite number of) very simple solutions, of the form

$$\phi(t, x, y) = A \exp[\mathrm{i}\,(k_x x + k_y y - \omega t)],$$

where exp is the exponential function, A is any complex number, and ω, kx, and ky are three numbers such that $\omega = f(kx,ky)$. These solutions are called *plane waves* because their wave-fronts are straight (lines in two dimensions, or planes in three dimensions). The number ω is called the *angular frequency* of the wave, and the vector $\mathbf{k} = (kx,ky)$ its *wave vector*. This relation between the angular frequency and wave vector is called the *dispersion relation*. It can be used to infer some important properties of the waves like the speed at which the information and energy it carries propagate.

These solutions are, however, often not very interesting in real-world scenarios. Indeed, they describe waves extending over the whole space with a uniform amplitude. In contrast, most waves of interest are *localised;* consider the sound emitted from a cello playing one of Bach's six cello suites, or the burst of light emitted by a distant supernova. In these cases, the amplitude of the waves is highest around the point where they are produced, and drops as the wave expands away. In general, there is no solution with a simple expression that can describe these waves. Fortunately, it is possible to simulate them numerically.

Resolution method

In this section, we sketch how wave propagation can be simulated using the Fourier transform, and how free-space optics can make the calculation faster and more energy-efficient. Our aim here is not to give a mathematically rigorous treatment; instead, we try to keep the presentation relatively simple while giving the key elements needed to understand the resolution method.

Technical assumptions

For the mathematically-oriented readers, let us sketch the assumptions underlying the resolution method we will use. This subsection is a bit more technical than the rest of the article, and can be safely skipped if you are mostly interested in the results — its aim is only to highlight the assumptions under which what we explain below can be made mathematically rigorous.

First, we assume the function ϕ is infinitely differentiable in its last two variables, *i.e.*, that its nth derivative with respect to x and mth derivative with respect to y is well-defined for all integer values of n and m. This assumption is slightly stronger than strictly necessary if the function f is a multinomial; but we adopt it to avoid the complications of distinguishing between the cases where f is a multinomial or not.

Second, we assume that ϕ and all its derivatives in x and y are square-integrable in these two variables over their domain of definition. For the initial conditions, this means that, for all non-negative integer values of n and m, the integral

$$\iint |\partial_x^n \partial_y^m \phi(0, x, y)|^2 \, \mathrm{d}x \, \mathrm{d}y$$

exists and is not infinite. (This assumption could also be slightly relaxed if $f$is a multinomial.)

Solving the wave equation using the Fourier transform

Let us show how the wave equation can be solved in Fourier space. We denote by F the Fourier transform operator in the space directions (*i.e*, the last two variables of ϕ). The wave equation can be rewritten as:

$$\mathrm{i}\partial_t \mathcal{F}(\phi)(t, k_x, k_y) = f(k_x, k_y)\mathcal{F}(\phi)(t, k_x, k_y).$$

This equation is much simpler than the previous one because the arguments of the function f are now numbers rather than derivatives. Moreover, the remaining derivative is of order one, making the resolution formally easy. The general solution may be written as:

$$\mathcal{F}(\phi)(t, k_x, k_y) = \exp(-\mathrm{i}tf(k_x, k_y))\mathcal{F}(\phi)(0, k_x, k_y).$$

This equation gives the Fourier transform of ϕ at time t from its value at some other time (here taken equal to 0). All one has to do to find ϕ is then to inverse the Fourier transform.

In summary, starting from some initial condition at $t = 0$, the value of ϕ at an arbitrary time can be obtained in three steps:

- Compute the Fourier transform of the initial condition.
- Multiply by an array of complex exponentials.
- Compute the inverse Fourier transform of the result.

While the wave propagation problem we are considering is a relatively simple one (as it does not include non-linear effects nor heterogeneous media), the same or a similar method is used in algorithms dealing with more involved ones, the main difference being that time must then generally be discretized and the above procedure repeated for each time step.

The optical advantage

Let us briefly comment on the complexity of the above method. There are three operations to consider: the Fourier transform, multiplication by an exponential, and the inverse Fourier transform.

On a CPU or GPU, the most efficient algorithm to perform the Fourier transform is due to James W. Cooley and John W. Tukey and often called the 'fast Fourier transform' (see also our previous article). If we work on a discretized grid with N pixels, this algorithm has a complexity in $O(N \log N)$. The inverse Fourier transform has the same complexity.

The complex exponential multiplication, on the other hand, has a complexity in $O(N)$. For large grids, it is thus much less demanding than the Fourier transform and its inverse, which are the bottlenecks of the algorithm.

This, at least, is true for electronic hardware. Using free-space optics, the Fourier transform and its inverse can be performed in theoretical $O(1)$runtime. (See our previous articles for more explanation on how we do it.) Free-space optics can thus completely eliminate the Fourier bottleneck in wave propagation simulation as well as other problems using spectral methods (see another of our articles for an example of an application to the Navier-Stokes equations).

Numerical results

Let us now show results from numerical simulations of wave propagation. We focus on the two-dimensional case. We will first show results obtained electronically using a standard CPU and the NumPy library for the Fourier transform. We then show corresponding simulations obtained using the Optalysys Fourier engine.

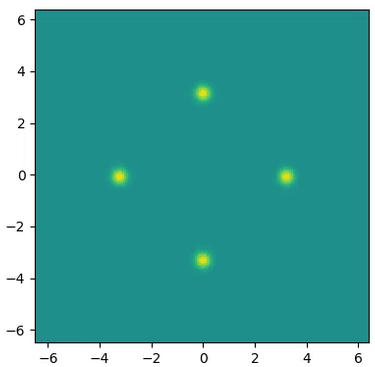
In all the simulations, we work with periodic boundary conditions, *i.e.*, we assume that the solution 'wraps around' the numerical grid, with its top edge identified with the bottom one and the left edge identified with the right one. We made this choice to stay as close as possible to what the Optalysys Fourier engine can do natively and reduce electronic processing to the minimum. Other boundary conditions, such as a reflecting wall, can be straightforwardly implemented using the same optical device by changing the modes which are kept when processing its output.

Electronic results

We first implemented the resolution method described above on an electronic CPU (Intel Core i7–9700K @ 3.60GHz), using the NumPy functions `fft.fft2` to perform the Fourier transform and `fft.ifft2` for its inverse. They should give exact results up to rounding errors (which are too small to be visible in the animations shown below) and thus provide a good baseline against which to compare the optical results.

We first simulated the simplest type of waves, with no dispersion nor dissipation. Such a wave has a simple dispersion relation of the form $\omega^2 = c^2 \mathbf{k}^2$, where ω is the angular frequency, c is the celerity of the wave, and $\mathbf{k}$ is the wave-vector. (In our case, it is a two-dimensional vector.) In practice, it means that waves will propagate away from their emission point with a constant speed, c. Light and sound waves, for instance, follow such a dispersion relation to a very good accuracy. (With, of course, very different values of c: 299,792,458 meters per second for light in vacuum[1] and about 343 meters per second for sound in air at 20°C.) This is why, for instance, red and blue light travel at the same speed[2].

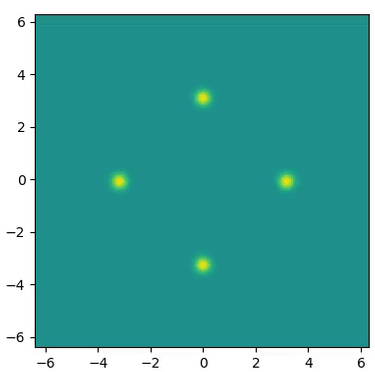
In the animation shown below, we start from four localised perturbations on a grid with size 130 by 130. We set the separation between pixels as 0.05 length units, and simulate the wave evolution for 30 time steps of 0.5 units each, with $c = 1$.



Example of wave propagation in a homogeneous isotropic medium with no dispersion nor dissipation, with periodic boundary conditions. Four localised perturbations generate as many circular waves propagating away with a constant speed and width.

In this case, wave propagation is quite simple: each initial perturbation generates a circular wave-front which expands at a constant speed (equal to c) and width. This is because all Fourier components move with the same speed, and the expansion is only due to the different propagation directions.

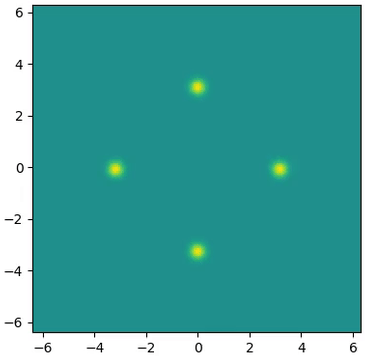
Wave propagation shows a richer behaviour in the presence of dispersion. To illustrate this, the animation below shows results for a dispersion relation of the form $\omega^2 = c^2 \mathbf{k}^2 + \lambda \mathbf{k}^4$, with $c = 0.5$ and $\lambda = 0.05$. Because of the last term, the Fourier components with smaller wavelengths now propagate faster than the long-wavelength ones.



Here the sound speed is equal to 0.5 and the dispersive coefficient is equal to 0.05. The initial localised perturbations still emit one circular wave each, but the later now broaden as they expand and show oscillations. The reason is that their high-frequency components move faster than their low-frequency ones.

As can be seen in the animation, this term has two effects. First, the wavefront expands much faster (in spite of the smaller value of c). Second, the waves get wider and wider during the propagation as long-wavelength Fourier components 'lag behind' their short-frequency counterparts. The superposition of these wide waves further generates a more complicate interference pattern.

Finally, we added a dissipative term in $-i\,\gamma\,|\mathbf{k}|$ to the angular frequency. For positive values of γ, this term models energy dissipation (which may, physically, be due to friction or other forms of energy exchange with the outside). The following animation shows results obtained with $\gamma = 0.04$.



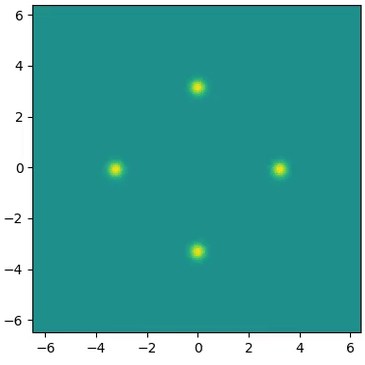
Example of dispersive wave with dissipation. At early times, the behaviour is similar to that of the previous dispersive wave. At later times, however, the wave progressively disappears as its energy is lost by dissipation.

Dissipation makes the amplitude of the wave decrease as time passes, until it becomes barely visible when most of its energy has been lost.

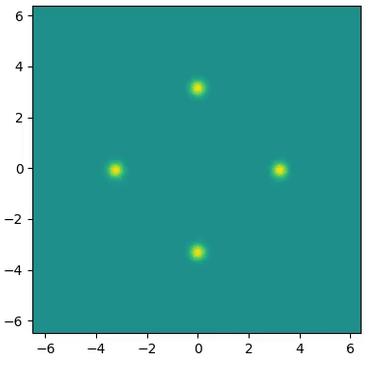
Optical results

We then performed the same simulations using the Optalysys demonstrator Fourier engine for the direct and inverse Fourier transforms. For each of the animations shown below, the initial condition, with a resolution of 130 by 130 pixels, is split twice into sub-images with size 10 by 10 then 5 by 5. The set of sub-images are then processed by the optical Fourier engine to compute the Fourier transform.

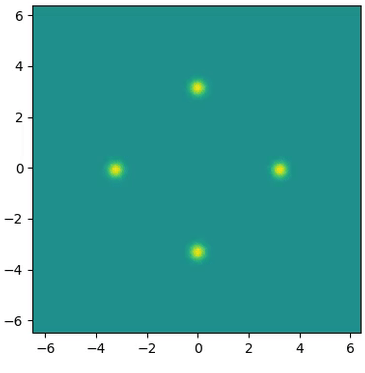
The three animations below show, from first to last, results obtained without dispersion nor dissipation, with dispersion, and with dispersion and dissipation. The parameters are the same as in the previous subsection.



Simulation of wave propagation in a homogeneous isotropic medium with no dispersion nor dissipation using the Optalysys demonstrator Fourier engine.



Simulation of wave propagation in a homogeneous isotropic medium with dispersion using the Optalysys demonstrator Fourier engine.



Simulation of wave propagation in a homogeneous isotropic medium with dispersion and dissipation using the Optalysys demonstrator Fourier engine.

The results look visually close to those shown above. The same differences are obtained between the simulations: an evolution at constant speed and width in the first one, a widening and the presence of an interference pattern in the second one, and a progressive weakening of the wave in the third one. One can check that the results also look similar at a quantitative level. In particular, the mean propagation speed, rate of decay, and relative amplitude of the different Fourier components are indistinguishable from those obtained electronically.

These results show that the optical Fourier engine can be used to perform simulations of wave propagation which are not just *visually* convincing, but also quantitatively accurate. In fact, the only visible differences between the optical and electronic versions of the animations are some echoes visible at early times, especially in the lower right corner. These echoes arise from slight variations in the calibration of the current optical device due to heat exchanges between the modulators used to encode information into the light. As we are moving away from thermal modulators, this issue will not be present in the next version of the optical chip, which will thus provide even more accurate results.

The optical advantage: numerical estimates

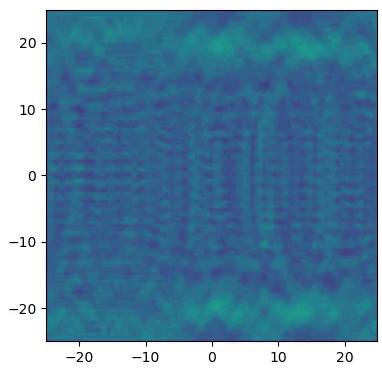
Let us briefly discuss the performance. On the CPU, we found that most of the runtime is, as expected, due to the direct and inverse Fourier transforms, which together took about 500μs for each frame, or 400μs when reducing the size of the image to 128 by 128. (Fast Fourier transform algorithms tend to be more efficient on images whose size is a power of 2.) Multiplication by the complex exponentials requires less than 18μs per frame if the array of complex exponentials is already stored in memory. (These estimates were obtained using the timeit Python module.)

As we explain the our previous articles, a free-space optical Fourier engine can reduce the time needed to perform the Fourier transforms and their inverse by orders of magnitude. The time needed to compute one frame could thus drop to less than 20μs — an improvement over the electronic version by a factor 20!

And this is a somewhat conservative estimate. Indeed, this figure for the speed-up is only limited by the complex multiplications, assuming they are as slow as the electronic version. If said multiplications can also be performed optically (and we have good reason to think they can; they will be presented in a future article along with more details on our next-generation Fourier engine), the speed-up could easily reach two orders of magnitude.

Inverse propagation

To further test the accuracy of the optical Fourier transform, we did an experiment on inverse wave propagation. Starting from an image showing the Optalysys name and logo as initial condition, we first simulated its evolution on the CPU (with dispersion) to generate an unrecognizable wave. After a while, we stopped the simulation and saved the final configuration. We then took the complex conjugate of this configuration (which amounts to reverting the wave velocity) and simulated the subsequent evolution on the Optalysys Fourier engine. The result is shown in the following animation. (The image has size 500 by 500. We arbitrarily set the distance between pixels to 0.05 units.)



Simulation of inverse wave propagation using the Optalysys Fourier engine.

Both the logo and letters are accurately reconstructed, with sharp edges. This constitutes an important test of the accuracy of the Fourier engine: since the forward evolution was performed electronically, there is no room for cancellation of errors in the optical backward simulation. Most systematic types of errors, such as a slightly different effective propagation speed or attenuation of some modes, would have resulted in blurry edges. This test thus constitutes a visual proof of the accuracy of wave propagation simulation using the optical Fourier engine.

Outlook

In this article, we explained how the propagation of waves in a homogeneous medium can be simulated numerically. We showed that the Fourier transform and its inverse are the most demanding aspects on electronic hardware, but that the optical Fourier transform can eliminate this bottleneck, making the algorithm much more efficient. We have implemented it on the Optalysys Fourier engine and found it accurately reproduced results obtained using an electronic CPU. This constitutes an experimental proof that free-space optics can be used to accurately simulate wave propagation.

As we mentioned earlier, this method is applicable much more broadly: a wide range of partial differential equations can be solved using spectral methods for which the Fourier transform is a bottleneck. The Optalysys Fourier engine has the potential to significantly accelerate the resolution of these equations and reduce the energy cost by orders of magnitude. For this reason, we firmly believe that optical computing in general, and free-space optics in particular, will play a prominent role in the near-future evolution of differential equation solving and computational fluid dynamics.

[1] This value is, at the time of writing, used to defined the metre: one metre is defined as the distance travelled by light in vacuum during a fraction 1/299792458 of a second.

[2] Technically, this is true only in vacuum: in any other medium, the speed of light will slightly depend on its wavelength due to interactions with atoms or other particles. This is why, for instance, the colours of a rainbow are separated: water droplets in the air deflect, on average, different wavelengths in different directions because of their different propagation speeds in water. This phenomenon is also responsible for some chromatic aberrations in photography, due to the different propagation speeds of light in the lenses.