FWI Documentation

Release 1.0

The Computing Research Group

Nov 13, 2019

Contents

It is a repository to document how to solve the wave equation using the finite difference method. It is an exercise for Pysit introduction to Full Waveform Inversion (FWI): https://pysit.readthedocs.io/en/latest/exercises/index.html

Contents:

CHAPTER 1

The Forward Problem

Many types of wave motion can be described mathematically by the equation $u_{tt} = \nabla \cdot (c^2 \nabla u) + f$. We use a more compact notation for the partial derivatives to save space:

$$u_t = \frac{\partial u}{\partial t}, \quad u_{tt} = \frac{\partial^2 u}{\partial t^2}$$

Let's implement a solver for the 1D scalar acoustic wave equation with absorbing boundary conditions. Let $u(x, t)$ be the displacement at time t and at space location x, which is the wavefield. The displacement function u is governed by the following mathematical model.

$$\begin{aligned}
\frac{1}{c(x)^2} u_{tt}(x,t) - u_{xx}(x,t) &= f(x,t), \\
\frac{1}{c(0)} u_t(0,t) - u_x(0,t) &= 0, \\
\frac{1}{c(1)} u_t(1,t) + u_x(1,t) &= 0, \\
u(x,t) = 0 \quad \text{for} \quad t \leq 0,
\end{aligned}$$

where the middle two equations are the absorbing boundary conditions, the last equation gives initial conditions, $x \in [0, 1]$, and $t \in [0, T]$. The model velocity is given by the function $c(x)$.

In our notation, we write that solving this PDE is equivalent to applying a nonlinear operator $\mathcal{F}$ to a model parameter m, where $m(x) = \frac{1}{c(x)^2}$ for the scalar acoustics problem.

We then write that $\mathcal{F}[m] = u$.

Contents:

1.1 Seismic Sources

$f(x, t)$ is the force function or source function in the above equation. We need to define it first.

We define our source functions as $f(x, t) = w(t)\delta(x - x_s)$, where $w(t)$ is the time profile, δ indicates that we will use point sources, and the source location is x_0. In real world applications, the time profile is not known and is estimated as part of the inverse problem. However, it is common to model source signals with the negative second derivative of a Gaussian, also known as the Ricker Wavelet:

$$w(t) = (1 - 2\pi^2\nu_0^2t^2)e^{-\pi^2\nu_0^2t^2}$$

where ν_0 is known as the characteristic or peak frequency (in Hz), because the magitude of w's Fourier transform $|\hat{w}|$ attains its maximum at that frequency. It is also important that this function is causal ($w(t) = 0$ for $t \leq 0$), so we introduce a time shift t_0,

$$w(t) = (1 - 2\pi^2\nu_0^2(t - t_0)^2)e^{-\pi^2\nu_0^2(t-t_0)^2}.$$

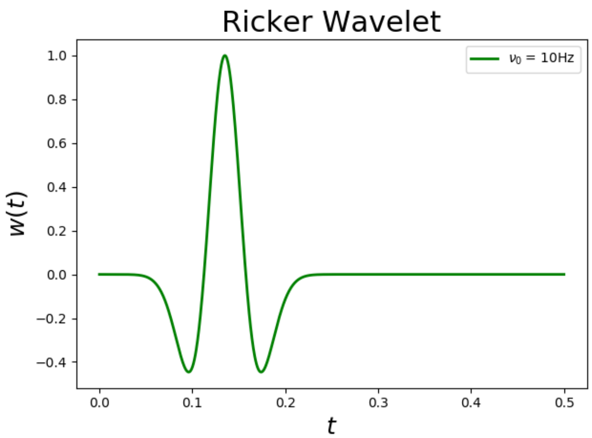
Problem 1.1

Write a Python function `ricker(t, config)` which implements the Ricker Wavelet, taking a time `t` in seconds and your configuration dictionary. This function should assume that your configuration dictionary has a key `nu0` representing the peak frequency, in Hz. Your function should returns the value of the wavelet at time `t`.

You can guarantee causality by setting $t_0 = 6\sigma$ for $\sigma = \frac{1}{\pi\nu_0\sqrt{2}}$, the standard deviation of the underlying Gaussian. You may also want to implement an optional threshold to prevent excessively small numbers.

Plot your function for $t = 0, \ldots, T = 0.5$ at $\nu_0 = 10$Hz and label the plot.

```
# In fwi.py

def ricker(t, config):

    nu0 = config['nu0']
    sigmaInv = math.pi * nu0 * np.sqrt(2)
    cut = 1.e-6
    t0 = 6. / sigmaInv
    tdel = t - t0
    expt = (math.pi * nu0 * tdel) ** 2
    w = np.zeros([t.size])
    w[:] = (1. - 2. * expt) * np.exp( -expt )
    w[np.where(abs(w) < 1e-7)] = 0

    return w

# Configure source wavelet
config['nu0'] = 10 #Hz

# Evaluate wavelet and plot it
ts = np.linspace(0, 0.5, 1000)
ws = ricker(ts, config)

plt.figure()
plt.plot(ts, ws,
         color='green',
         label=r'$\nu_0 =\,{0}$Hz'.format(config['nu0']),
         linewidth=2)
plt.xlabel(r'$t$', fontsize=18)
plt.ylabel(r'$w(t)$', fontsize=18)
plt.title('Ricker Wavelet', fontsize=22)
plt.legend();
```



Ricker Wavelet
ν_0 = 10Hz
$w(t)$
t
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
0.0
0.1
0.2
0.3
0.4
0.5

CHAPTER 2

Indices and tables

- genindex
- modindex
- search