IDENTIFYING GRAVITY WAVE EVENTS
from atmospheric 3D-temperature data

Tobias Hauffe-Waschbüsch, Kai Hinzmann, Sarthak Kapoor, Yiğit Köse, Corinna Müller, Alice Muthurajavel, Paul Orschau

&

Dr. Jörn Ungermann, Konstantin Ntokas, Prof. Dr. Benjamin Berkels

Short Description
Our aim was to identify key characteristics (position, direction, spatial wavelengths, and magnitude) of gravity wave packets in several decades of highly resolved 3-D temperature data. Therefore, the constructed algorithm must be able to scale regarding the parameters memory and execution time. The knowledge of the exact information on the gravity waves can then help to improve weather forecasting models as gravity waves play an important role as driver of the large-scale atmospheric circulations.

1 Challenges

Gravity waves arise when wind in the atmosphere is perturbated by obstacles like islands or mountains. The waves then propagate in time and space until they break or dissipate in high atmospheric layers. Beside by wind data, those waves can also be measured by small temperature fluctuations as the waves expand adiabatically. In higher altitudes the air pressure is lower than at lower altitudes so that a cooling can be observed when the wave rises and a heating when the wave drops again. After a preprocessing step of the data, where low frequency perturbations are removed, the small scale fluctuations can be analyzed. To describe those local gravity wave packets localized waves, so called wavelets, are used whose definition will be introduced in section 2.

1.1 Continuous wavelet transform

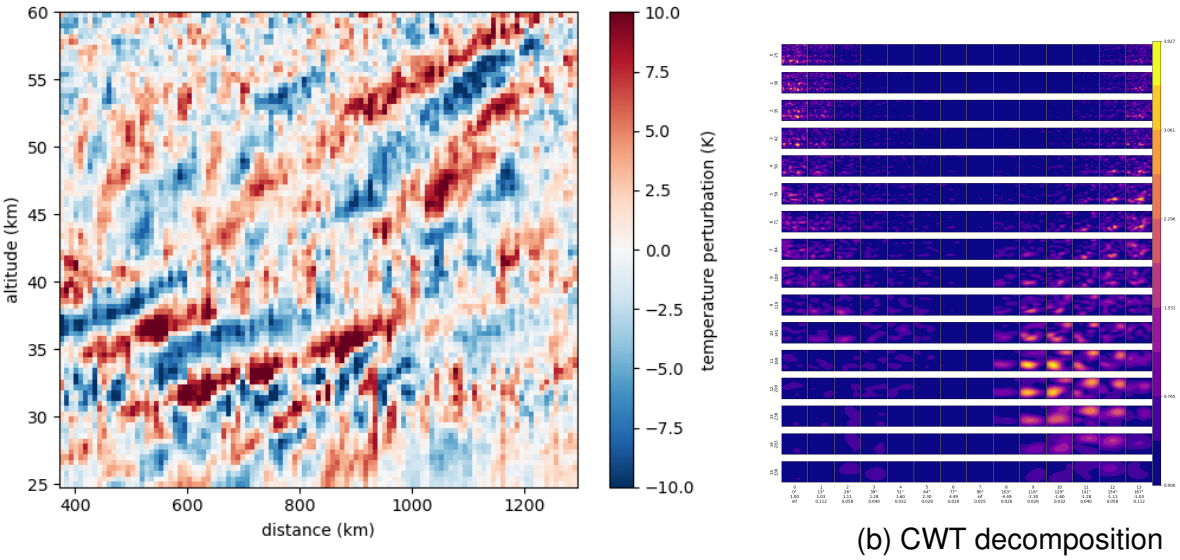
To analyze the wavelets included in a data set, a straightforward approach is to apply the continuous wavelet transform (CWT). This transformation works analogously to a Fourier transform but decomposites the data into a wavelet basis instead of a basis of waves. A detailed description of this procedure can be found in section 3. To implement this method on a computer a discretization has to take place so that the coefficients are only computed for a finite number of possible wavelets. Different wavelets differ in their translation and their scaling. For two dimensional data this results in a four dimensional parameter space as both translation and scaling can be applied in two coordinate directions. So, for two dimensional data like in figure 1a the result of the CWT can be depicted in a 4D plot like the one in figure 1b.

But there are two problems with this algorithm:

1. **Bad scaling**: When applied to 3D data the computational effort and memory needs increase even more as three translation and three scaling parameters are then needed, which form a six dimensional space.

2. **Wasteful**: The method computes much more information than needed as only the dominant and maybe the second most dominant wave are of interest.

The purpose of this project is to find another approach which overcomes those problems.



(a) Preprocessed data

(b) CWT decomposition

Figure 1: CWT decomposition of two dimensional temperature data in the four dimensional parameter space

1.2 New approach

The requirements for the new approach can thus be identified as finding, implementing and testing an algorithm that computes only the parameters for the wavelet of the largest amplitude and has better scaling properties than the old approach. The first idea to reduce the algorithms complexity is to not compute the possible wavelets for each point in space but only for a subset as there will not be much change in the near surroundings of a point. The second idea addresses that only the dominant wave is needed. To find only this wave, an optimization algorithm is used which optimizes the parameters of one wavelet at a specific point to best fit the in the data underlying wave of highest amplitude. As this optimization problem is non-linear and non-convex a good starting point for the procedure is needed which can be found utilizing the Fast Fourier Transform (FFT). This initialization process is described in section 4 before the applied optimization strategy follows in section 5.

1.3 State of the art

Few-wave decomposition[2] is the state of the art technique which minimizes the squared deviations of measurements and wave vectors. An initial wave vector is obtained by the wave with the largest amplitude in Fourier transform. This method involves finding the best fit for each of the components of the wave vector and thus reduces the variance. This method has the added advantage that waves with wavelengths larger than the extent of the dataset can also be identified, unlike FFT. The approach we introduce here follows in its steps, but uses Morlet wavelets instead of waves to fit the measurement data.

2 Wavelets

First, the definition of wavelets and their parameters are introduced. According to [3] a wavelet in $\mathbb{R}^n$ is defined as a function ψ which fulfills the admissibility condition $\int_{\mathbb{R}^n} \frac{|\hat{\psi}(\vec{\omega})|}{\|\vec{\omega}\|_2^2} \, \mathrm{d}\,\vec{\omega} < \infty$. This condition is met if $\hat{\psi}(\vec{0}) = 0$ and $\hat{\psi}(\vec{\omega})$ is continuously differentiable. The first part is given if ψ has zero mean $\int_{\mathbb{R}^n} \psi(\vec{x}) d\vec{x} = 0$. The continuous differentiability can be reformulated as $\psi \in \mathrm{L}^2(\mathbb{R}^n)$. Additionally, it is often assumed that ψ is normalized, e.g. $\|\psi\| = 1$. Based on this a wavelet basis can be generated by stretching this mother wavelet with a scaling parameter $s \in \mathbb{R}$ and translating it with a transition

vector $\vec{v} \in \mathbb{R}^n$:

$$\left\{ \psi_{\vec{v},s}(\vec{x}) = s^{-\frac{n}{2}} \psi \left(\frac{1}{s} (\vec{x} - \vec{v}) \right) \right\} \tag{1}$$

Together with the normalization factor $s^{-\frac{n}{2}}$ these so called daughter wavelets are normalized again and fulfill the admissibility condition.

2.1 Morlet wavelet

A often used wavelet for the purpose of geopysical analysis is the Morlet wavelet which is a oscillation of a Gaussian curve. The Morlet mother wavelet is defined as

$$\Psi(\vec{x}|k) = \exp\left(ik\vec{x} \cdot \vec{\omega}_0\right) \exp\left(-\frac{|\vec{x}|^2}{2}\right) \tag{2}$$

Here the $\vec{\omega}_0$ defines the direction of the underlying complex wave function which makes up the first part of the equation. The second exponential term represents a Gaussian curve and thus creates an almost compact domain of the wavelet. The wave number k is proportional to the number of osciallations in the wavelet.

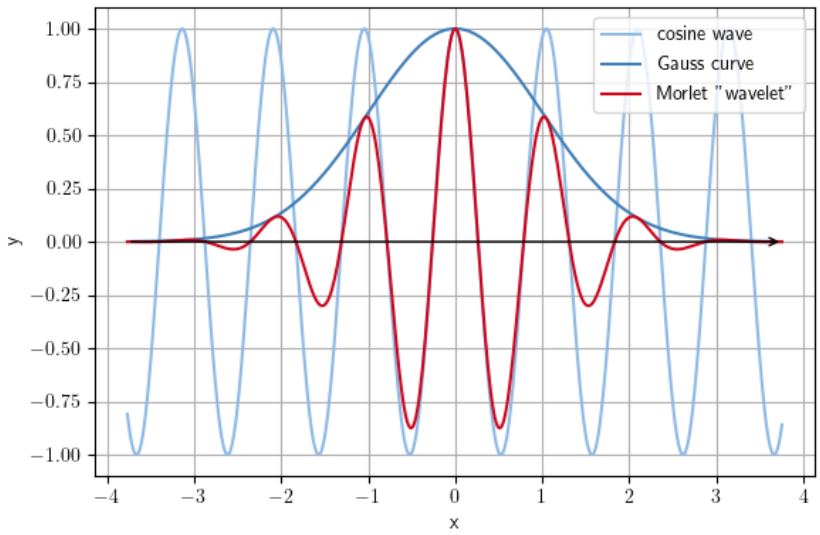


Figure 2: Morlet wavelet in one dimension

It is actually not an admissible wavelet as it does not have zero mean but for high wave numbers k it is quite close so that this can be neglected for our purpose.
For two space dimensions this can be reformulated by baking into account a rotation around the angle $\phi \in [0, 2\pi)$ via $\vec{\omega}_0 = (\cos\theta, \sin\theta)^T$ and scaling the function with a parameter $s \in \mathbb{R}$. This leads to the daughter wavelet formulation:

$$\Psi_{s,\theta}(x, y) = \frac{1}{s} \exp\left(ik \left(\frac{x}{s} \cos(\theta) + \frac{y}{s} y \sin(\theta) \right) \right) \exp\left(-\frac{x^2 + y^2}{2s^2} \right) \tag{3}$$

Here the factor $\frac{1}{s}$ is introduced to ensure a constant norm of ψ for varying s.

2.2 Reformulation of the parameters of the Morlet wavelet

As only the real number space is needed to represent real data, the imaginary part of this function can be neglected. To take into account that a wave in the data does not have to be centered around

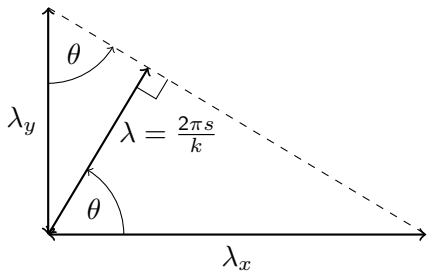


Figure 3: Transformation of the parameters of the 2D Morlet wavelet from scaling s and angle θ to two 1D wavelengths

a point to evaluate in the domain, a phase shift is added to the real cosine function:

$$\Psi_{s,\theta,\phi}(x,y) = \frac{1}{s}\cos\left(k\left(\frac{x}{s}\cos(\theta)+\frac{y}{s}\sin(\theta)\right)+\phi\right)\exp\left(-\frac{x^2+y^2}{2s^2}\right) \tag{4}$$

Another way of interpreting the parameters of the function is to convert them into two one-dimensional wavelengths as it is depicted in figure 3. In general, the wavelength in the direction of θ can be computed as $\lambda = \frac{2\pi s}{k}$ Here, the following trigonometric equations can be read:

$$\cos(\theta) = \frac{\lambda}{\lambda_x} \quad \Leftrightarrow \quad \lambda_x = \frac{\lambda}{\cos(\theta)} = \frac{2\pi s}{k\cos(\theta)} \quad \Leftrightarrow \quad \frac{k\cos(\theta)}{s} = \frac{2\pi}{\lambda_x} \tag{5}$$

$$\sin(\theta) = \frac{\lambda}{\lambda_y} \quad \Leftrightarrow \quad \lambda_y = \frac{\lambda}{\sin(\theta)} = \frac{2\pi s}{k\sin(\theta)} \quad \Leftrightarrow \quad \frac{k\sin(\theta)}{s} = \frac{2\pi}{\lambda_y} \tag{6}$$

Squaring and adding the two equations leads to:

$$\cos(\theta)^2+\sin(\theta)^2 = \left(\frac{\lambda}{\lambda_x}\right)^2+\left(\frac{\lambda}{\lambda_y}\right)^2 = 1 \Leftrightarrow \frac{1}{\lambda_x^2}+\frac{1}{\lambda_y^2} = \frac{1}{\lambda^2} = \frac{k^2}{(2\pi)^2 s^2} \Leftrightarrow s = \frac{k}{2\pi}\sqrt{\frac{1}{\frac{1}{\lambda_x^2}+\frac{1}{\lambda_y^2}}} \tag{7}$$

Plugging in equations (5), (6) and (7) into equation (4) leads to the new equation:

$$\psi_{\lambda_x,\lambda_y,\phi}(x,y) = \frac{2\pi}{k}\sqrt{\frac{1}{\lambda_x^2}+\frac{1}{\lambda_y^2}}\cos\left(2\pi\left(\frac{x}{\lambda_x}+\frac{y}{\lambda_y}\right)+\phi\right)\exp\left(-\left(\frac{2\pi}{k}\right)^2\frac{x^2+y^2}{2}\left(\frac{1}{\lambda_x^2}+\frac{1}{\lambda_y^2}\right)\right) \tag{8}$$

Furthermore, defining the spatial frequencies $f_x = \frac{1}{\lambda_x}$ and $f_y = \frac{1}{\lambda_y}$ leads to a formulation of the Morlet wavelet in two dimensions dependent on a spatial frequency in each spatial direction and the phase shift ϕ:

$$\psi_{f_x,f_y,\phi}(x,y) = \frac{2\pi}{k}\sqrt{f_x^2+f_y^2}\cos\left(2\pi\left(xf_x+yf_y\right)+\phi\right)\exp\left(-\left(\frac{2\pi}{k}\right)^2\frac{x^2+y^2}{2}\left(f_x^2+f_y^2\right)\right) \tag{9}$$

3 Theory

In this section we follow [4] and [1]. Let $\psi \in L^2(\mathbb{R})$ with

$$\int_{\mathbb{R}}\psi(t)\,dt = 0 \qquad \text{and} \qquad \|\psi\| = 1.$$

Then by streching and translation of the so-called mother wavelet ψ one can generate a wavelet basis by the set

$$\left\{\psi_{u,s}(t) = \frac{1}{\sqrt{s}}\psi\left(\frac{t-u}{s}\right)\right\}_{u\in\mathbb{R},s\in\mathbb{R}^+}.$$

The continuous wavelet transform, CWT, of a function $f \in \mathrm{L}^2(\mathbb{R})$ is then defined by

$$Wf(u,s) = \langle f, \psi_{u,s} \rangle = \int_{-\infty}^{\infty} f(t) \frac{1}{\sqrt{s}} \psi^* \left(\frac{t-u}{s} \right) dt, \tag{10}$$

where ψ^* denotes the complex conjugate of ψ. To find the wavelet with the largest amplitude, which is given by $\widetilde{Wf}(u,s) = \frac{1}{\sqrt{s}} Wf(u,s)$, that fits the given data we have to identify u', s' with

$$\left| \widetilde{Wf}(u',s') \right| = \max_{u,s} \left| \widetilde{Wf}(u,s) \right|.$$

In order to find these maximizing u' and s' we have to find the wavelet $\psi_{u',s'}$ which maximizes the value of $\left| \widetilde{Wf}(u,s) \right|$ by using the formula (10).
Notice, that one can interpret the integral in (10) as a convolution of f and ψ^*. The calculation of a convolution is expensive, so instead of computing this convolution, we want to use a Fourier transform for the implementation. For this solution, we rewrite (10) as

$$Wf(u,s) = \int_{-\infty}^{\infty} f(t) \frac{1}{\sqrt{s}} \psi^* \left(\frac{t-u}{s} \right) dt = \frac{1}{2\pi} \int_{-\infty}^{\infty} \hat{f}(\omega) \sqrt{s} (\hat{\psi}(s\omega))^* e^{i\omega t} \, d\omega \tag{11}$$

and work with $Wf(u,s)$ in the Fourier representation. In this reprensentation, we do not have to calculate a convolution, there is just a multiplication left.
As the mother wavelet we choose the Morlet wavelet which is given by

$$\psi(t) = \frac{\sqrt{2}}{\pi \sqrt[4]{\pi}} e^{ikt} e^{-\frac{t^2}{2}}, \tag{12}$$

where k is a parameter which is chosen later. After applying a Fourier transform we get

$$\hat{\psi}(\omega) = \frac{2\pi}{\sqrt[4]{\pi}} e^{-\frac{(\omega-k)^2}{2}}.$$

Now, it just remains the task to find the u', s' that maximizes $\left| \widetilde{Wf}(u,s) \right|$ numerical.
Next we are interested in recovering the data, that means we want to recover $f(t)$ if we have a given value of Wf. For that, we want to show that

$$f(x) = \frac{1}{C_\psi} \int_0^{\infty} \int_{-\infty}^{\infty} Wf(u,s) \frac{1}{\sqrt{s}} \psi \left(\frac{t-u}{s} \right) du \frac{ds}{s^2} \tag{13}$$

where

$$C_\psi = \int_0^{\infty} \left| \hat{\psi}(\omega') \right|^2 \frac{d\omega}{\omega'} < \infty.$$

Notice that ψ as defined in (12) is a real wavelet.
Proof: By definition, it holds that

$$\begin{aligned}
RHS &= \frac{1}{C_\psi} \int_0^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(t') \frac{1}{\sqrt{s}} \psi^* \left(\frac{t'-u}{s} \right) dt' \frac{1}{\sqrt{s}} \psi \left(\frac{t-u}{s} \right) du \frac{ds}{s^2} \\
&= \frac{1}{C_\psi} \int_0^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(t') \frac{1}{\sqrt{s}} \left(D_{-\frac{1}{s}} \psi^* \right) (u-t') \, dt' \frac{1}{\sqrt{s}} \psi \left(\frac{t-u}{s} \right) du \frac{ds}{s^2} \\
&= \frac{1}{C_\psi} \int_0^{\infty} \int_{-\infty}^{\infty} \left(f * \left(D_{-\frac{1}{s}} \psi^* \right) \right) (u) \frac{1}{\sqrt{s}} \frac{1}{\sqrt{s}} \psi \left(\frac{t-u}{s} \right) du \frac{ds}{s^2} \\
&= \frac{1}{C_\psi} \int_0^{\infty} \int_{-\infty}^{\infty} \left(f * \left(D_{-\frac{1}{s}} \psi^* \right) \right) (u) \left(D_{\frac{1}{s}} \psi \right) (t-u) \, du \frac{ds}{s^3} \\
&= \frac{1}{C_\psi} \int_0^{\infty} \left(\left(f * \left(D_{-\frac{1}{s}} \psi^* \right) \right) * \left(D_{\frac{1}{s}} \psi \right) \right) (t) \frac{ds}{s^3}.
\end{aligned}$$

Now we can apply the Fourier transform on both sides of (13). We get

$$\begin{aligned}\hat{f}(\omega) &= \frac{1}{C_\psi}\int_0^\infty \left(f * \widehat{\left(D_{-\frac{1}{s}}\psi^*\right)}\right)(\omega)\left(\widehat{D_{\frac{1}{s}}\psi}\right)(\omega)\frac{ds}{s^3}\\ &= \frac{1}{C_\psi}\int_0^\infty \hat{f}(\omega)\left(\widehat{D_{-\frac{1}{s}}\psi^*}\right)(\omega)\left(\widehat{D_{\frac{1}{s}}\psi}\right)(\omega)\frac{ds}{s^3}\\ &= \frac{1}{C_\psi}\int_0^\infty \hat{f}(\omega)\left(|-s|\,\widehat{\psi^*}(-s\omega)\right)\left(|s|\,\hat{\psi}(s\omega)\right)\frac{ds}{s^3}.\end{aligned}$$

By the symmetrie of ψ^* and because of the integration over $s \geq 0$ it follows that $\widehat{\psi^*}(\cdot) = \hat{\psi}^*(-\cdot)$ and

$$\begin{aligned}\hat{f}(\omega) &= \frac{1}{C_\psi}\int_0^\infty \hat{f}(\omega)s\hat{\psi}^*(s\omega)\,s\hat{\psi}(s\omega)\frac{ds}{s^3}\\ &= \hat{f}(\omega)\frac{1}{C_\psi}\int_0^\infty \hat{\psi}^*(s\omega)\,\hat{\psi}(s\omega)\frac{ds}{s}\\ &= \hat{f}(\omega)\frac{1}{C_\psi}\int_0^\infty \left|\hat{\psi}(s\omega)\right|^2\frac{ds}{s}.\end{aligned}$$

Substitution of $\omega' = s\omega$ leads to

$$\hat{f}(\omega) = \hat{f}(\omega)\frac{1}{C_\psi}\underbrace{\int_0^\infty \left|\hat{\psi}(\omega')\right|^2\frac{d\omega}{\omega'}}_{C_\psi} = \hat{f}(\omega),$$

so the identity in (13) holds.
After that, we would like to have a formula like

$$f(t) = \frac{1}{C_\delta}\int_0^\infty Wf(t,s)\frac{1}{\sqrt{s}}\frac{ds}{s} \tag{14}$$

for some C_δ. We can proceed as before. First it holds that

$$\begin{aligned}RHS &= \frac{1}{C_\delta}\int_0^\infty\int_{-\infty}^\infty f(t')\frac{1}{\sqrt{s}}\psi^*\left(\frac{t'-u}{s}\right)dt'\frac{1}{\sqrt{s}}\frac{ds}{s}\\ &= \frac{1}{C_\delta}\int_0^\infty\int_{-\infty}^\infty f(t')\frac{1}{\sqrt{s}}\left(D_{-\frac{1}{s}}\psi^*\right)(u-t')\,dt'\frac{1}{\sqrt{s}}\frac{ds}{s}\\ &= \frac{1}{C_\delta}\int_0^\infty\int_{-\infty}^\infty \left(f * \left(D_{-\frac{1}{s}}\psi^*\right)\right)(u)\frac{ds}{s^2}.\end{aligned}$$

Now we can apply the Fourier transform again to (14). With the same reasons as before we get

$$\begin{aligned}\hat{f}(\omega) &= \frac{1}{C_\delta}\int_0^\infty \hat{f}(\omega)s\hat{\psi}^*(s\omega)\frac{ds}{s^2}\\ &= \hat{f}(\omega)\frac{1}{C_\delta}\int_0^\infty \hat{\psi}^*(s\omega)\frac{ds}{s}.\end{aligned}$$

With substitution of $\omega' = s\omega$ it follows that

$$\hat{f}(\omega) = \hat{f}(\omega)\frac{1}{C_\delta}\int_0^\infty \hat{\psi}^*(\omega')\frac{d\omega}{\omega'}.$$

So the identity in (14) ist true for

$$C_\delta = \int_0^\infty \hat{\psi}^*(\omega')\frac{d\omega}{\omega'} < \infty.$$

The last task is to discretize the formulas (10) and (14) for the implementation. For formula (14) we substitute $s(j) = s_0 2^{\frac{j}{J}}$ for an $s_0 \in \mathbb{R}^+$ and a $J \in \mathbb{N}$. With that we get

$$f(t) = \frac{1}{C_\delta} \int_{s_0}^{2s_0} Wf(t,j) \frac{\ln 2}{J\sqrt{s_0} 2^{\frac{j}{2J}}} \, dj$$

where we suppose that $Wf(t,\cdot)$ is mostly zero outside the interval $[s_0, 2s_0]$. The discrete form is then given by

$$f[n] = \frac{1}{C_\delta} \sum_{j=0}^{J} 1 \cdot \frac{\ln 2}{J\sqrt{s_0} 2^{\frac{j}{2J}}} Wf[n,j].$$

Next, we want to discretize (10) by

$$Wf[n,s] = \sum_{n'=0}^{N-1} 1 \cdot f(n') \frac{1}{\sqrt{s}} \psi^* \left(\frac{n'-n}{s} \right)$$

for a $N \in \mathbb{N}$ and f and ψ^* where f is mostly zero outside the interval $[0, N-1]$ and ψ^* is mostly zero outside $[\frac{-n}{s}, \frac{(N-1)-n}{s}]$. We continue as we did before, but this time we use the discrete convolution to write

$$\begin{aligned} Wf[n,s] &= \frac{1}{\sqrt{s}} \sum_{n'=0}^{N-1} f(n') \left(D_{-\frac{1}{s}} \psi^* \right) (n-n') \\ &= \frac{1}{\sqrt{s}} \left(f * \left(D_{-\frac{1}{s}} \psi^* \right) \right) (n). \end{aligned}$$

With the discrete Fourier transform and by the same reasons as in the proof of (13) we get

$$\begin{aligned} Wf[n,s] &= \frac{1}{N} \sum_{k=0}^{N-1} \frac{1}{\sqrt{s}} \hat{f}(k) s \hat{\psi}^*(sk) e^{i\frac{2\pi k n}{N}} \\ &= \frac{1}{N} \sqrt{s} \sum_{k=0}^{N-1} \hat{f}(k) \hat{\psi}^*(sk) e^{i\frac{2\pi k n}{N}}. \end{aligned}$$

The two discrete formulas can now be impemented.
In the two-dimensional case one can do the same as in the onedimensional case when you use the wavelet

$$\hat{\psi}(\omega_x, \omega_y, \theta) = \sqrt{2\pi} e^{-\frac{(\omega_x - \cos(\theta)k)^2 + (\omega_y - \sin(\theta)k)^2}{2}}$$

where $\theta \in [0, 2\pi)$ is the angle of the wavelet and

$$Wf(u,v,s,\theta) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x,y) \frac{1}{s} \psi^* \left(\frac{x-u}{s}, \frac{y-v}{s}, \theta \right) dx \, dy$$

the related CWT.

4 Initial Guess

4.1 Problem Description

For typical optimizations algorithms there is a necessity of an initial point or a window where to expect the solution. Since the window of possible wave lengths is large and wave functions are highly not convex, just picking a random point to start from is not feasible. The optimization will probably get stuck in a local minimum. It may be easy for the human eye to recognize waves but computers struggle with these kind of problems.

4.2 Basic Idea

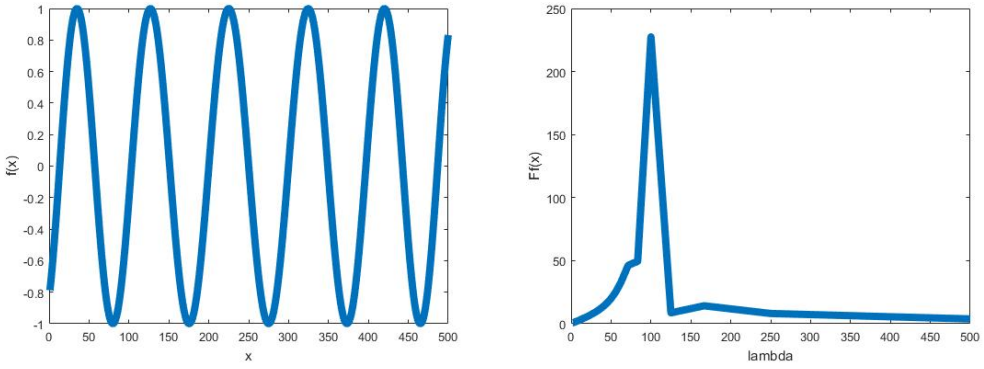
At first we eliminate all linear and constant parts of the data, so the data has a mean of zero and no slope. We expect the data to consist of multiple added waves with different wave lengths and amplitudes. In spectral space waves are resembled by Dirac deltas, which are easy to distinguish.

$$\hat{f} = \mathcal{F}f = \int_{-\infty}^{\infty} f(x)e^{-i\omega x}dx$$

$$\mathcal{F}(cos(fx)) = \frac{\delta(\omega - f) + \delta(\omega + f)}{2}$$

Since we have discretized data we can use the Fast-Fourier-Transformation. The first guess for the wavelength will then be the highest peak in the spectral space.

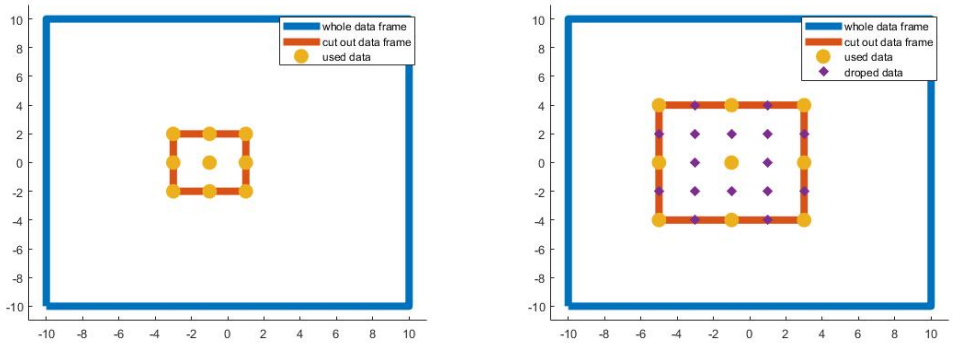
Figure 4: FFT of a wave like function



4.3 Rescale Image

To keep the computation time in control and not scale with the number of grid points, we cut out a little window of data points around the point of interest. For the FFT to work reasonably, the wave length should be smaller then window size. To not loose the ability to recognize large scale waves, we repeat the FFT on different sizes of the cutout window. The number of points in the cutout window stay constant. As we already detected small waves we can afford a coarser grid. In the end we pick the maximum amplitude over all window sizes.

Figure 5: first and second window size for the FFT



4.4 Limits

If the real wave is in between to possible outcomes this method can never find the actual solution. The possible results are determined by:

$$\lambda = \frac{windowsize}{k}, k \in \mathbb{N}_0$$

In later iterations of the window rescaling many data points are dropped. Therefore measurement errors can have an huge impact on the result.

4.5 Runtime

The runtime complexity of the first guess is:

$$O(n \cdot log(n) \cdot r)$$

$n =$ number of data points in the window

$r =$ number of rescaling steps

The FFT for each window is $O(n * log(n))$ and will be executed on every window size. In comparison to the old method the number of points in the data window is limited due to the the down sampling. The number of rescaling steps can also be fixed since the window size grows exponentially and reaches nonphysical sizes after a few iterations.

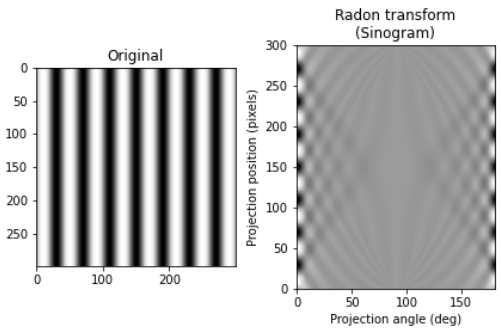
4.6 Outlook

Instead of dropping out data while down sampling one could use an average of neighboring points. This increases the robustness against measurement errors. The cut out window may result in high peaks in the spectral space at high frequencies. The can be solved by multiplying the data window with a gauss function so the data is roughly zero at the boundary.

4.7 Alternative Approach: Radon transform

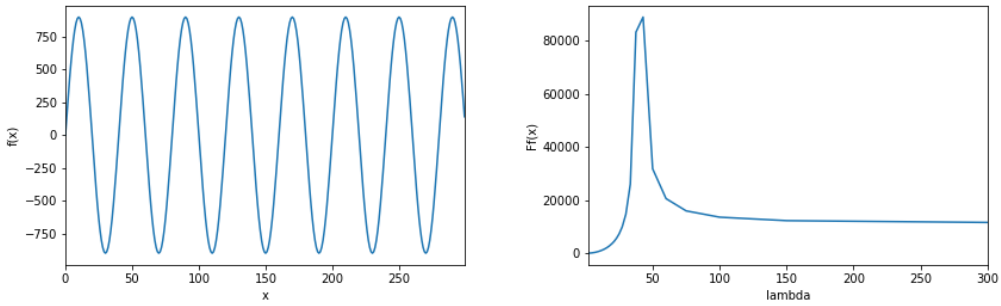
Radon transform is an integral transform which transforms the input function to another function which is defined on a two dimensional space of lines in a plane. Value of each line in the plane is equal to the line integral of the function over that particular line. When a property is "horizontal" with respect to the projection direction (orthogonal to the line) its signal is stronger. Using this fact, dominant waves in the input image can be detected.
If a wave is aligned with the projection direction, signals from its peaks would be stronger, so if we apply Radon transform to the input image, strong signals in the sinogram represent the peaks of the dominant waves.

Figure 6: Radon transformation of a simple input



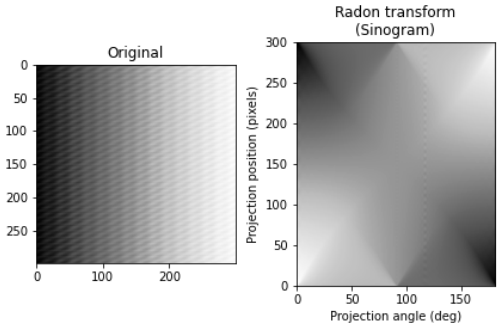
By checking the maximum variance among the data for each different angle value, the angle of the dominant wave can be obtained. By taking a slice of the sinogram at the obtained angle, a one dimensional representation of the dominant wave can be obtained.After getting the one dimensional wave, fast Fourier transform is performed.

Figure 7: FFT of the extracted wave



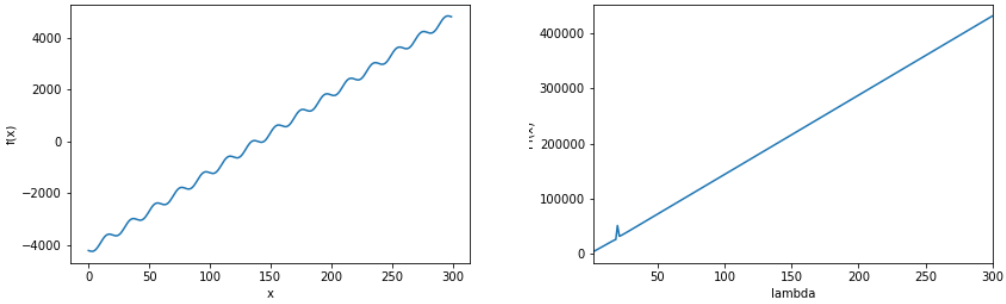
Frequency of the dominant wave is obtained from the FFT. Using the frequency and the angle of the dominant wave x and y components of the wavelength can be computed.
With more complicated data this method begins to fail and require more modifications. In the figure below Radon transform of a more complicated input is given.

Figure 8: Radon transformation of a complicated input



In this figure we have a linear term which makes it hard to detect waves from the sinogram.

Figure 9: FFT of the extracted wave



From these plots it can be seen that linear term disrupts the sinogram and wave information cannot be obtained accurately. Whereas other method can obtain a valid initial guess from the same data so this method was discarded. Another point is that angles in the sinogram are increased with 1 degree increments, while this reduces the computational costs during the maximum variance search, for high frequency data it could lead to under sampling and inaccurate results.

5 Optimization

5.1 Problem Description

With a given initial guess for the parameters of the wavelet that fits best the wave in the given data of the largest amplitude, an optimization is applied to improve this rough approximation. Therefore, two main points have to be considered. First, what is the objective that should be optimized and second, which optimization algorithm is most suitable to solve this kind of problem.

5.2 Objective

For a given data vector $\vec{d}$ on a grid $(x_i, y_i)_i$ the corresponding discretized wavelet can be formulated as $(\psi(x_i, y_i))_i$. We define a measure comparing the data vector and discretized wavelet as the objective to optimize wavelet parameters (wavelengths and phase shift). Our first approach defines this measure as least square difference (15), which is then minimized. By definition, least square minimization has regularization as it considers minimization of individual norms in addition to maximizing the scalar product between the two quantities.

$$\min_{(\lambda_x,\lambda_y,\phi)} \left\| \vec{d} - (\psi(x_i, y_i))_i \right\|_2^2 = \min_{(\lambda_x,\lambda_y,\phi)} \left\| \vec{d} \right\|_2^2 - 2 \left\langle \vec{d}, (\psi(x_i, y_i))_i \right\rangle + \|(\psi(x_i, y_i))_i\|_2^2 \tag{15}$$

Our second approach defines similarity score using scalar product (16) as objective, which is then maximized.

$$\max_{(\lambda_x,\lambda_y,\phi)} \left\langle \vec{d}, (\psi(x_i, y_i))_i \right\rangle \tag{16}$$

Both of the approaches maximizes the amplitude (17) of the fitted wavelet and gives us the estimate of the first dominant wave at $(x_i, y_i)_i$.

$$A = \frac{\left\langle \vec{d}, (\psi(x_i, y_i))_i \right\rangle}{\|(\psi(x_i, y_i))_i\|_2^2} \tag{17}$$

5.3 Optimizaton Algorithm

To optimize the objective, we used derivative-free (Nelder-Mead) and derivative-based methods (Conjugate Gradient, Trust Region Reflective). Nelder-Mead (`N-M`) is a direct search method that maintains a set of $n + 1$ test points arranged as a simplex for n-dimension optimization space, and points are updated at every iteration to minimize the objective. On other hand, Conjugate Gradient (`CG`) method, with non-linear adaptation, uses first derivative of the objective to compute the update direction. For Trust Region Reflective (`trf`), the shape of the trust-regions is determined by direction of the first derivative of objective, and promotes exploration of the parameter space.
`trf` was implemented using SciPy method `optimize.least_squares`, whereas `N-M` and `CG` were implemented using SciPy method `optimize.minimize`. The Jacobian for derivative-based methods were incorporated as callable derivative function of the objective. These were implemented using forward-mode algorithmic differentiation method from JAX library. Additionally, we used just-in-time (JIT) compilation of JAX related implementation for optimized performance.

5.4 Performance

We tested the performance, in terms of runtime, and accuracy, in terms of L2 norm between computed and real parameters, of the optimization algorithm on synthetic data which was generated as described in section 6.1. We found that least squares objective for `N-M` and `CG` gave better accuracy than scalar product objective (by an order of magnitude), however the runtimes were higher (2-3 times). `CG` and `trf` had similar accuracy, however, `trf` was almost twice faster. `N-M` is slightly more accurate than `trf`, however, the runtimes are significantly higher (5-6 times). Therefore, we found optimization with `trf` using the least squares objective as the optimal combination for our use case.

5.5 Outlook

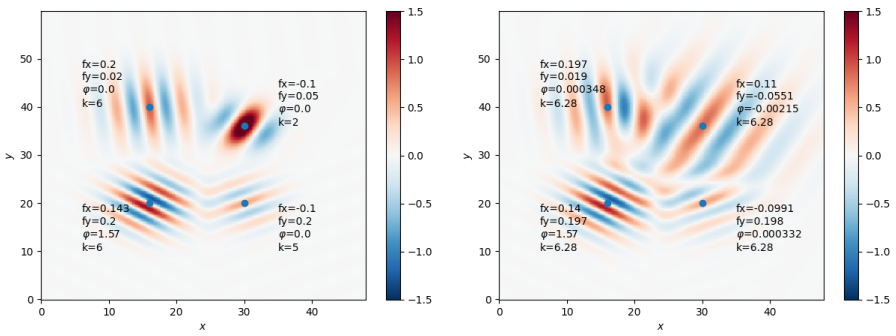
Our method doesn't take into account the uncertainties involved in measurement of temperature data. If measurement-variance data is available, we can replace the objective with weighted-least-squares that gives more weight to data with lower variance. Additionally, our method can be erroneous when the fixed-size of the window cutout around given point is not comparable to the wave features we are trying to fit. This can be perilous especially near to the domain edges, where a fixed-size window might not be available around the given point. To overcome this, our algorithm can be modified to consider a dynamically-sized window, possibly using a coarse-to-fine optimization approach.

6 Results

6.1 Synthetic Data

To test the new two-step approach introduced in section 1.2, we create synthetic data containing wavelets with parameters of our choosing. The example in figure 10 contains four wavelets in different locations, whose parameters can be determined by applying the proposed method in the same, known locations. Using the found parameter values, the four wavelets may then be reconstructed again. The formula used for each of the summed up ground truth wavelets f_i is given in equation 18, the parameter values in table 10c.

$$f_i(x+x_o, y+y_o) = \frac{A}{k\sqrt{2\pi(f_x^2+f_y^2)^{-1/2}}} \cos(2\pi(xf_y+yf_y)+\varphi) \exp\left(-\frac{2\pi^2}{k^2}(x^2+y^2)(f_x^2+f_y^2)\right) \quad (18)$$



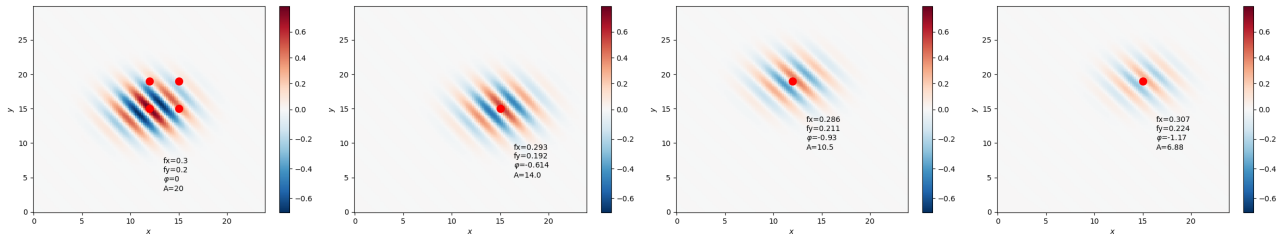
(a) Synthetic wavelets (b) Reconstructed wavelets

	f_1	f_2	f_3	f_4
x_o	16	30	16	30
y_o	40	36	20	20
A	30	30	40	15
k	6	2	6	5
f_x	0.2	-0.1	1/7	-0.1
f_y	0.02	0.05	0.2	0.2
φ	0	0	$\pi/2$	0

(c) Parameter values

Figure 10: Synthetic data used to test the parameter estimation and reconstruction of the method.

One can see that the parameter values for the spatial frequencies f_x and f_y, as well as for the phase shift φ, have been captured fairly well in all four cases. The most striking visual difference occurs for the top-right wavelet, which can not be reconstructed exactly, since the wave number k remains fixed at a predefined values of $k = 2\pi$ in our case. Hence, the observed difference is as expected, and the algorithm still finds the "closest approximation" to the ground truth wavelet within the set of optimization variables it can manipulate.
In a second example in figure 11, a single wavelet was generated at (x_o=12, y_o=15), and the algorithm was evaluated at several offsets $(u, v) = \{(3, 0), (0, 4), (3, 4)\}$ from this point. The generated wavelet uses A=20, k=6, f_x=0.3, f_y=0.2 and φ=0, and the reconstructions all use k=6 for their fitting as well.
The example shows that the parameter values for f_x and f_y found by the algorithm are still close to the true values, while the values for φ and A are bound to be different, since choosing an evaluation point at some offset must result in a smaller amplitude and a different phase shift.

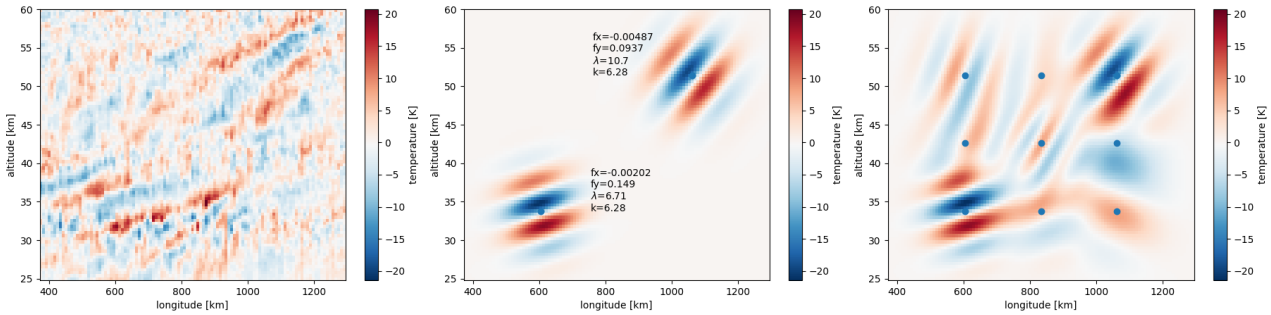


(a) Target wavelet (b) $(3, 0)$ (c) $(0, 4)$ (d) $(3, 4)$

Figure 11: Synthetic wavelet, evaluated at several offsets.

6.2 Real Data

Now the algorithm may be used to examine the given temperature dataset, at some predefined evaluation points. Fig. 12 shows how the algorithm can be applied on just two chosen evaluation points, or on a coarse 3-by-3 grid, to capture the coarse structure of the dataset.



(a) Given dataset (b) Two evaluation points (c) Nine evaluation points

Figure 12: Algorithm evaluated on given dataset.

Examining just figure 12b and the evaluation point in the lower left corner, one can calculate the spatial wavelengths to be $\lambda_x = f_x^{-1} \approx (0.002\,02\,\mathrm{km}^{-1})^{-1} \approx 500\,\mathrm{km}$ and $\lambda_y = f_y^{-1} \approx (0.149\,\mathrm{km}^{-1})^{-1} \approx 6.71\,\mathrm{km}$. Figure 12c shows that the algorithm may in principle be used to coarsely examine a large dataset, while still capturing most of its original structure. Choosing an appropriate coarse grid for this evaluation will depend on the data that should be examined.

7 Application / Outlook

The described method scales with the points of interest we consider to identify our wavelet, unlike CWT which scales with the number of data points and parameters chosen. As this method does not need to store the entire dataset to process, it has a smaller memory footprint. The parameter estimation is continuous and can result in more accurate estimation of the parameters. The parallelization capabilities of this method are also better as each point of interest can be computed independant of each other.

7.1 Application

Scalability, parallelization capabilities and lower memory requirement makes the method a suitable candidate for processing large datasets obtained from the atmospheric temperature measurements.

7.2 Further Testing

After successfully applying the described method to a set of test problems, further tests are necessary before applying it to real data sets:

- Influence of noise: Gaussian noise in the data should be no problem as it has average zero and much higher frequency than the wavelet that is to be extracted.
- Influence of overlaying waves

7.3 Outlook

Additionally, further extensions have to be made to the algorithm to ensure that it is applicable to real world problems:

- Handling of overlays of multiple equally dominant waves: This case can be handled by utilizing the fast Fourier transform (cf 4) to inspect the ratio of the amplitudes of the first and the second most dominant wave. If this ratio is close to one, the second most dominant wave can not be neglected. Instead the optimization procedure can be adjusted to optimize the two most dominant waves or by subtracting an approximation of the second most dominant wave from the data.
- Three-dimensional data: To extend the proposed algorithm to three dimensional data, all applied principles have to be adjusted for three dimensions. First, the general Morlet mother wavelet (2) can also be applied to three dimensions by not only introducing a scaling and the rotation around one axis but by a scaling and the rotation around two different axes. Second, the Fourier transform (cf. 4) can effortlessly be extended to three dimensions as this algorithm is already implemented in python. Third, the optimization procedure (cf. 5) can be the same but another optimization variable and thus one more dimension is introduced to the optimization problem.

References

[1] Cao Chen and Xinzhao Chu. Two-dimensional morlet wavelet transform and its application to wave recognition methodology of automatically extracting two-dimensional wave packets from lidar observations in antarctica. *Journal of Atmospheric and Solar–Terrestrial Physics*, 2016.

[2] C. I. Lehmann, Y.-H. Kim, P. Preusse, H.-Y. Chun, M. Ern, and S.-Y. Kim. Consistency between fourier transform and small-volume few-wave decomposition for spectral and spatial variability of gravity waves above a typhoon. 5(7):1637–1651, 2012.

[3] Stéphane Mallat. *A Wavelet Tour of Signal Processing*. 01 1999.

[4] Jörn Ungermann. *Presentation: WAVELETS AND FRIENDS*. Forschungszentrum Jülich, 2021.