kOmega documentation

Giulio Dolcetti[1,a], Borbála Hortobágyi[2], Matthew Perks[3], Simon J. Tait[4]

[1]: Department of Civil, Environmental and Mechanical Engineering, University of Trento, Trento, Italy.
[2]: UMR5600 Environnement Ville Société, Université de Lyon, Lyon, France.
[3]: School of Geography, Politics and Sociology, Newcastle University, Newcastle, UK.
[4]: Department of Civil and Structural Engineering, University of Sheffield, Sheffield, UK.
[a]: corresponding author, giulio.dolcetti@unitn.it

June 2, 2023

Contents

1 Fundamentals of the Method

kOmega is a Matlab script to calculate the average flow velocity and water depth of rivers and open-channel flows from sequences of images of the water surface recorded with a camera. The analysis is based on the method described in Dolcetti et al., 2022. The script computes the frequency-wavenumber spectra of the input set of images, and runs an optimisation routine to compare the measurements with the theoretical dispersion relation of water waves and to identify the set of flow parameters that provide the best fit with the measured data.

The method allows the estimation of the average flow velocity without requiring the presence of artificial tracers. It implements a robust analytical model of the water surface dynamics, therefore the accuracy is not undermined by the presence of gravity waves (including standing/stationary waves). The method is best suited for the analysis of videos or sets of images where surface deformations such as gravity waves are clearly visible, although it can also be applied in the absence of visible waves in the presence of artificial or natural floating tracers with suitable density.

Note: This script is intended for research applications only. The results of the analysis shall be interpreted with caution and always validated against established measurement methods. The Authors decline any responsibility for damage or harm resulting from the inappropriate use of the script or of any part of it.

1.1 Types of surface deformations

The water surface of rivers and open-channel flows shows the presence of multiple types of surface deformations (e.g., Muraro et al., 2021). For the kOmega method, it is important to distinguish between gravity-capillary waves and turbulence-generated or turbulence-forced surface deformations. These surface deformations produce distinct features in the frequency-wavenumber spectra of the water surface of river and open-channel flows (Dolcetti et al., 2016).

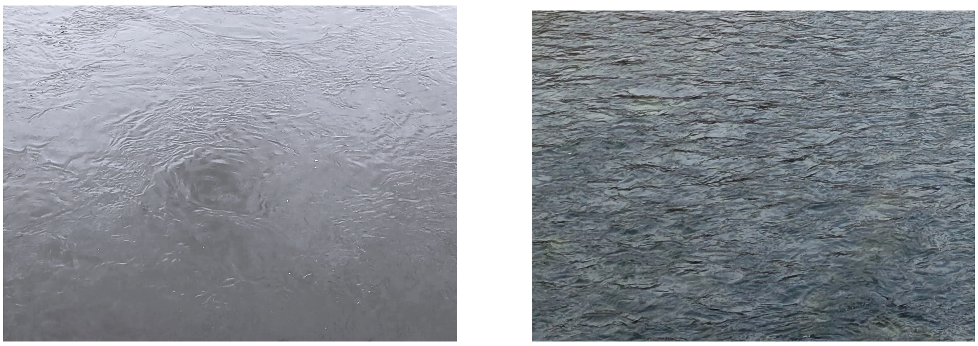

Figure 1: Examples of water surface dominated by different types of deformations. Left: turbulence-forced deformations (boils). Right: gravity-capillary waves.

Gravity-capillary waves are generally defined as sinusoidal long-crested waves comprising multiple crests and troughs, although the combination of multiple gravity-capillary waves can also produce patterns that are difficult to identify visually. Gravity-capillary waves do not move at the same speed of the underlying flow. This can affect the accuracy of standard optical velocimetry approaches if gravity waves or ripples are used as tracers (e.g., Dolcetti et al., 2020). Instead, gravity-capillary waves propagate relative to the flow with the intrinsic celerity c_i, which varies with the wavelength (long waves are faster than short waves) and which also depends on the depth of the flow, d:

$$c_i(d) = \sqrt{gd\frac{(1+B)}{B}\frac{\tanh(kd)}{kd}}, \tag{1}$$

where $B = \rho g k^{-2} \gamma^{-1}$ is the so-called Bond number (which tends to infinity for long gravity-dominated waves), ρ is the water density, g is the acceleration due to gravity, and γ is the surface tension coefficient. Gravity-capillary waves are also advected by the flow, therefore their velocity in the streamwise direction is increased by an amount proportional to the speed of the flow.

Turbulence-generated surface deformations can appear at the surface of turbulent flows with different forms and shapes such as boils, scars, vortex dimples (e.g., Brocchini and Peregrine, 2001; Muraro et al., 2021). Although their dynamics are still largely unknown, they are believed to move downstream approximately at the same speed of the flow. Hence, their speed is independent of their wavenumber.

Note: within kOmega, tracers (either artificial or natural, such as seeding particles, leaves, foam, etc.) are effectively equivalent to turbulence-generated surface deformations, since they also move at the same speed of the flow. Therefore, the method can be used also with artificial or natural tracers.

1.2 Frequency-wavenumber spectra

kOmega exploits the relationship between the speed of the different types of waves and the flow parameters (velocity and water depth) to estimate the flow conditions. It does so by means of a Fourier-based approach applied to sequences of images of the water surface, which produce the so-called frequency-wavenumber spectrum. In fact, the speed or celerity of a wave is proportional to the ratio between its frequency f and wavenumber k:

$$c = \frac{2\pi f}{k}. \tag{2}$$

Frequency and wavenumber describe the temporal and spatial scales of a sinusoidal fluctuation. The frequency f (Hz) is the inverse of the period P (s), $f = 1/P$, and it indicates the number of cycles per second measured at a fixed location in space. The wavenumber $\mathbf{k}$ (rad/m) is the spatial equivalent of frequency, it is proportional to the inverse of the wavelength λ (m), $|\mathbf{k}| = 2\pi/\lambda$, and it indicates how many full wavelengths are found in a 1 m length. The wavenumber is a vector with modulus $k = |\mathbf{k}|$ directed in the direction of propagation of the wave. The two components of the wavenumber vector along the x and y directions are denoted as k_x and k_y, respectively.

The relationship between frequency and wavenumber (the so-called dispersion relation) can be approximated with the following equations (see Dolcetti et al., 2022), for turbulence-generated deformations:

$$f = \frac{1}{2\pi}\mathbf{k} \cdot \mathbf{U_s}, \tag{3}$$

and for gravity-capillary waves:

$$f = \frac{(1-\beta)}{2\pi}\mathbf{k} \cdot \mathbf{U_s} \pm \frac{1}{2\pi}\sqrt{\left(\beta \mathbf{k} \cdot \mathbf{U_s}\right)^2 + (kc_i)^2}, \tag{4}$$

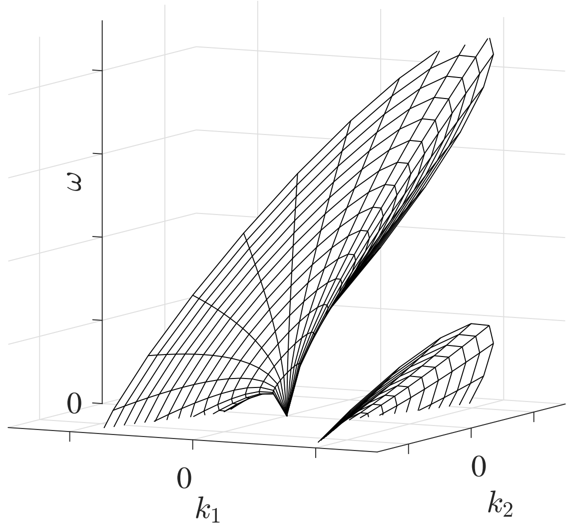

Figure 2: Dispersion relation of gravity-capillary waves in two-spatial dimensions. $\omega = 2\pi f$. Taken from Dolcetti et al., 2022.

where U_s is the surface flow velocity,

$$\beta = (1-\alpha)\frac{\tanh(kd)}{kd}, \tag{5}$$

and $\alpha = \bar{U}/U_s$ is the so-called velocity index, i.e., the ratio between the depth-averaged flow velocity $\bar{U}$ and the surface velocity U_s. Note that the dispersion relation of turbulence-generated surface fluctuations (eq. (3)) depends linearly on the surface flow velocity U_s, while the dispersion relation of gravity-capillary waves (eq. (4)) shows a more complex dependence on both the surface flow velocity U_s and on the water depth d.

The discrete frequency-wavenumber spectrum is employed for decomposing a space-time signal such as a sequence of images into a discrete set of sinusoidal components (waves), each with wavenumber components $k_{x,q}$ and $k_{y,p}$ and with frequeny f_n. This is achieved by means of a Fourier transform in three dimensions (x, y, and time). It is assumed that a set of digital images is represented by a 3D matrix $\mathrm{Z}_{N_y\times N_x\times N_t}$, where N_y and N_x are the number of pixels in the y and x-direction, respectively, and N_t is the number of frames. $\mathrm{Z}_{\eta,\xi,\tau}$ indicates the instantaneous intensity of the pixel corresponding to the spatial co-ordinates $y = \eta L_y/N_y$ and $x = \xi L_x/N_x$ at the time $t = \tau T/N_t$, where L_y and L_x are the physical dimensions covered by the image (in m), and T is the duration of the set of images or video. Then, the discrete frequency-wavenumber spectrum $\mathrm{I}_{N_y\times N_x\times N_t}$ is a 3D matrix obtained by means of three discrete Fourier transforms applied along all three dimensions of the data, i.e.,

$$\mathrm{I}_{p,q,n} = \frac{1}{N_x N_y N_t}\left|\sum_{\eta=0}^{N_y-1}\sum_{\xi=0}^{N_x-1}\sum_{\tau=0}^{N_t-1}\mathrm{Z}_{\eta,\xi,\tau}\exp\left[-\mathrm{i}2\pi\left(q\eta/N_y + p\xi/N_x - n\tau/N_t\right)\right]\right|^2. \tag{6}$$

$\mathrm{I}_{p,q,n} = I(k_{y,p}, k_{x,q}, f_n)$ indicates the contribution in terms of energy that can be attributed to a wave with wavenumber components $k_{y,p} = p2\pi/L_y$ and $k_{x,q} = q2\pi/L_x$ and with frequency $f_n = n/T$. If gravity-capillary waves or turbulence-generated waves are present at the water surface, the frequency-wavenumber spectrum will have a larger amplitude at the frequency-wavenumber combinations that satisfy equations (3) and (4).

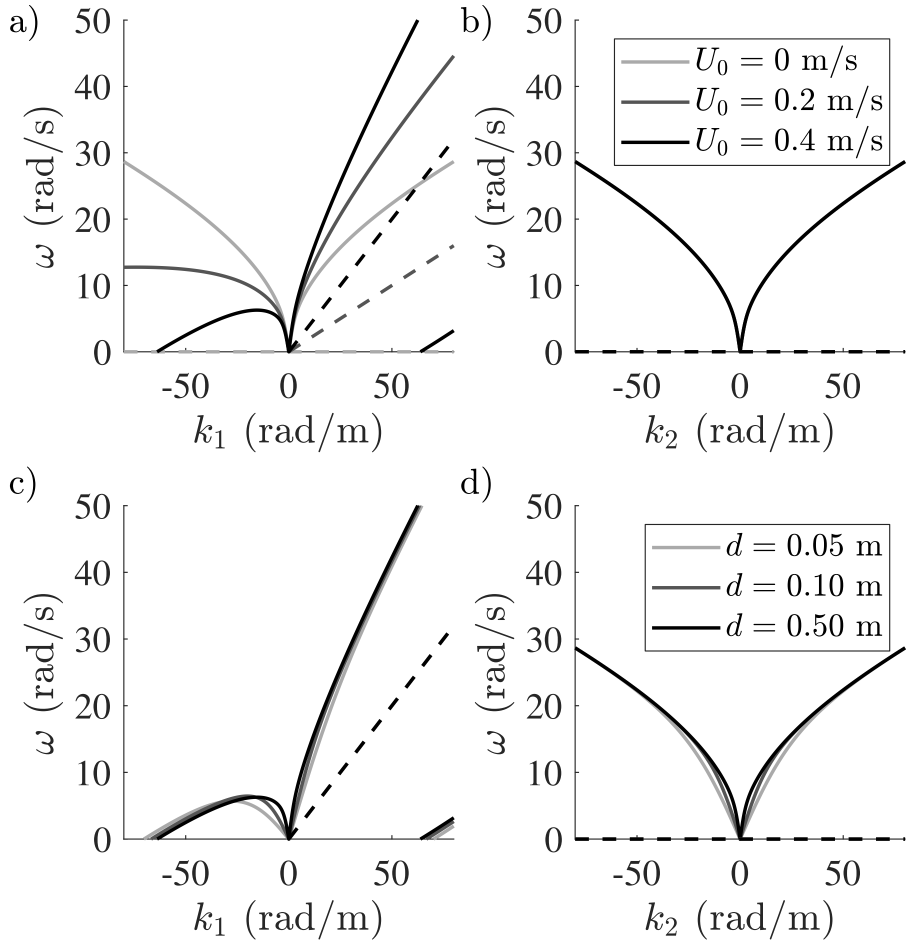

Figure 3: Effect of a variation of the flow velocity (a, b) or of the water depth (c, d) on the frequency-wavenumber spectra of the water surface fluctuations. In (a,b) the depth is kept constant and the flow velocity is varied. In (c,d) the flow velocity is kept constant and the depth is varied. (a,c): cross-section along the streamwise direction. (b,d): cross-section along the lateral direction. Taken from Dolcetti et al., 2022.

1.3 Estimation of the flow velocity and depth

The approach of kOmega consists in fitting equations (3) and (4) to the high-energy peaks of the measured frequency-wavenumber spectrum in order to identify the optimal values of U_x and U_y (the two components of $\mathbf{U_s}$) and eventually of d that better approximate the data. The fitting is performed by means of an optimiser that searches for the maximum of the Normalised Scalar Product:

$$\mathrm{NSP} = \left[\sum_{p,q,n} \mathrm{Z}_{p,q,n}\mathrm{M}_{p,q,n}\right]\left[\sum_{p,q,n} \mathrm{Z}_{p,q,n}\sum_{p,q,n} \mathrm{M}_{p,q,n}\right]^{-1}, \tag{7}$$

where $\mathrm{M}_{N_y \times N_x \times N_t}$ is a synthetic Gaussian-weighed frequency-wavenumber spectrum that follows equations (3) and (4).

Note: Unlike Dolcetti et al., 2022, who used a Self-Adaptive Differential Evolution (SADE) algorithm (Qin and Suganthan, 2005) for the optimisation, kOmega employs the nonlinear constrained multivariable solver fmincon in Matlab, which is part of the **Optimization Toolbox**. **Please ensure that the toolbox is installed in order to run the code.**

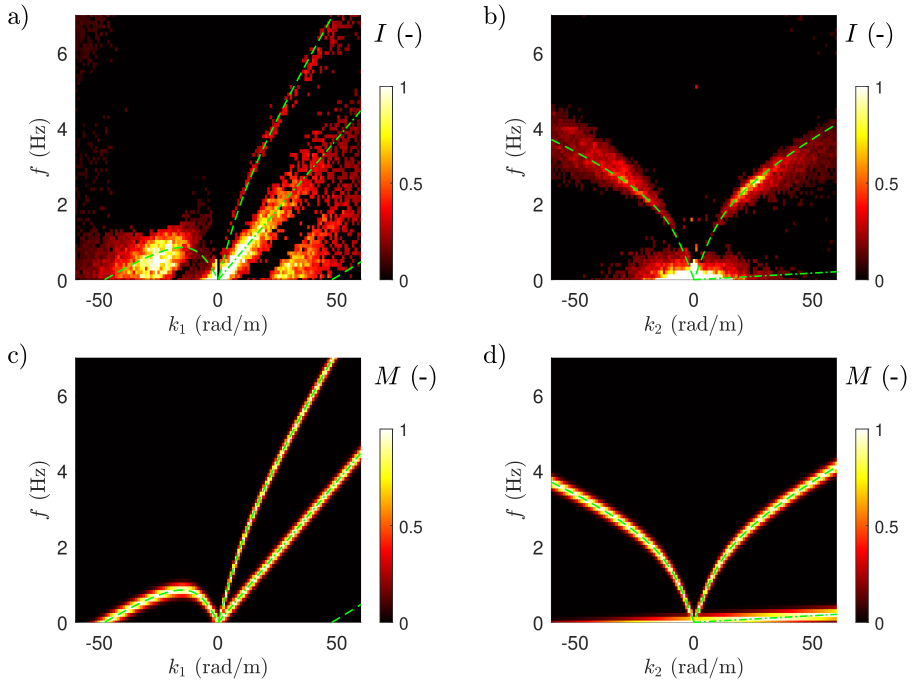

Figure 4: Example of the fitting procedure. (a) and (b) show the streamwise (a) and lateral (b) cross-section through the 3D measured frequency wavenumber spectrum. (c) and (d) show the theoretical Gaussian-weighed spectrum M after the optimisation. The green lines are the theoretical dispersion relations, eq. (3) and eq. (4). Taken from Dolcetti et al., 2022.

1.4 Uncertainties and guidelines

The main sources of uncertainty of the method have been discussed in detail by Dolcetti et al., 2022. As is customary with other optical velocimetry approaches such as LSPIV (Muste et al., 2008), the whole size of an image is usually split into portions (Areas of Interest, AOI) where conditions are assumed to be homogeneous, and for which a single value of velocity and/or depth is estimated. A smaller AOI improves the resolution of the measured distribution of the flow parameters. However, Dolcetti et al., 2022 demonstrated that the uncertainty of the Fourier-based approach implemented in kOmega is strongly dominated by the spectral resolution. This defines the minimum detectable change of the dispersion relation and therefore of the flow parameters, and it is inversely proportional to the size of the AOI and to the duration of the sequence of images.

For accurate results, the area of interest (AOI, the portion of image that is actually used for the analysis) should be large enough to include multiple wavelengths (ideally 6-10).
The velocity estimations rely mostly on short ripples that can be detected accurately even with a small AOI. Therefore, **measurements of the velocity alone can be performed with high accuracy even with a small AOI, thus enabling a high spatial resolution of the velocity distributions.**
On the other hand, the sensitivity of the spectra to depth variations is small and controlled by waves with a wavelength larger than the water depth itself. As a result, **depth estimations require a large AOI, at least 6-10 times larger than the water depth.** Moreover, depth estimations are highly uncertain when long waves are absent. This could be the case of relatively calm, slow and/or deep flows with relatively small bed roughness. In these cases an accurate estimation of the water depth could be impossible. Additionally, the depth can only be estimated in the presence of gravity-capillary waves, while velocity estimations are possible even with turbulence-generated surface deformations alone.

When this is possible without affecting the relevance of the results (e.g., in the case of a relatively flat bathymetry) it is suggested to perform the analysis in two steps. In the first step, it is recommended to use a very large AOI covering almost the entire width of the river/channel to estimate an average water depth and velocity. In the second step, a refined spatial distribution of velocity and eventually depth can be calculated by means of multiple smaller AOI's opportunely distributed in space. During this second step, it is possible to either use the value of depth estimated during the first step, or to select relatively narrow boundaries to at least constrain the depth within reasonable limits. An exemplification of this two-step approach is given in the examples included with the code.

1.5 Camera placement

The relationship between the pixel intensity of an image of the water surface and the local surface deformation depends on multiple factors such as the camera angle and sensitivity to light and the illumination conditions. These factors can affect the detectability of the surface deformations. As discussed by Dolcetti et al., 2022, better visibility of the waves is obtained with oblique-viewing cameras in spite of nadir-looking ones (e.g., from drones), although too low angles may result in increased ortho-rectification errors. Direct sun reflections and sun glint should be avoided as well as shadows from river banks, bridges, trees, etc., whenever possible. The camera should be fixed and stable, or the images stabilised.

Note: kOmega does not include the algorithms to perform the image ortho-rectification or stabilisation. These steps should be performed externally with another software or code prior to the analysis.

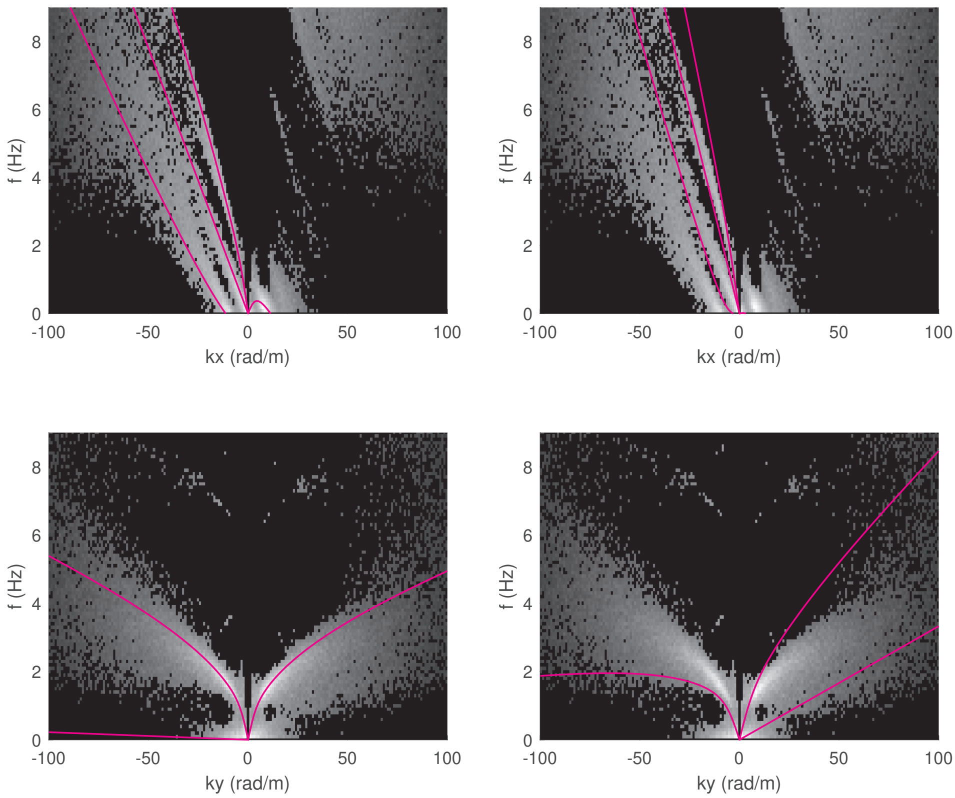

Figure 5: Left: successful optimisation. Right: unsuccessful optimisation caused by a local minimum. Note the discrepancy between the measured spectrum and the theoretical relations in the latter case. Top: streamwise cross-section of the spectrum. Bottom: lateral cross-section of the spectrum.

2 Matlab Script

2.1 Requirements

kOmega uses the nonlinear constrained multivariable solver fmincon in Matlab, which is part of the **Optimization Toolbox**. The toolbox must be installed in order to run the code.

2.2 Algorithm Steps

1. Input data and parameters
2. Images pre-processing
3. Calculation of the frequency-wavenumber spectrum
4. Spectrum pre-processing
5. Optimisation
6. Output

The main input data consists in sets of orthorectified grayscale images provided in the form of a 3-dimensional array, where the 3 dimensions correspond to the spatial y-axis, to the spatial x-axis, and to the time (frame) axis, respectively. By default, kOmega analyses the whole data provided as input. If downsampling in time and/or space is required, or if the analysis should be limited to a smaller Area of Interest, then these should be identified and selected before running the code.

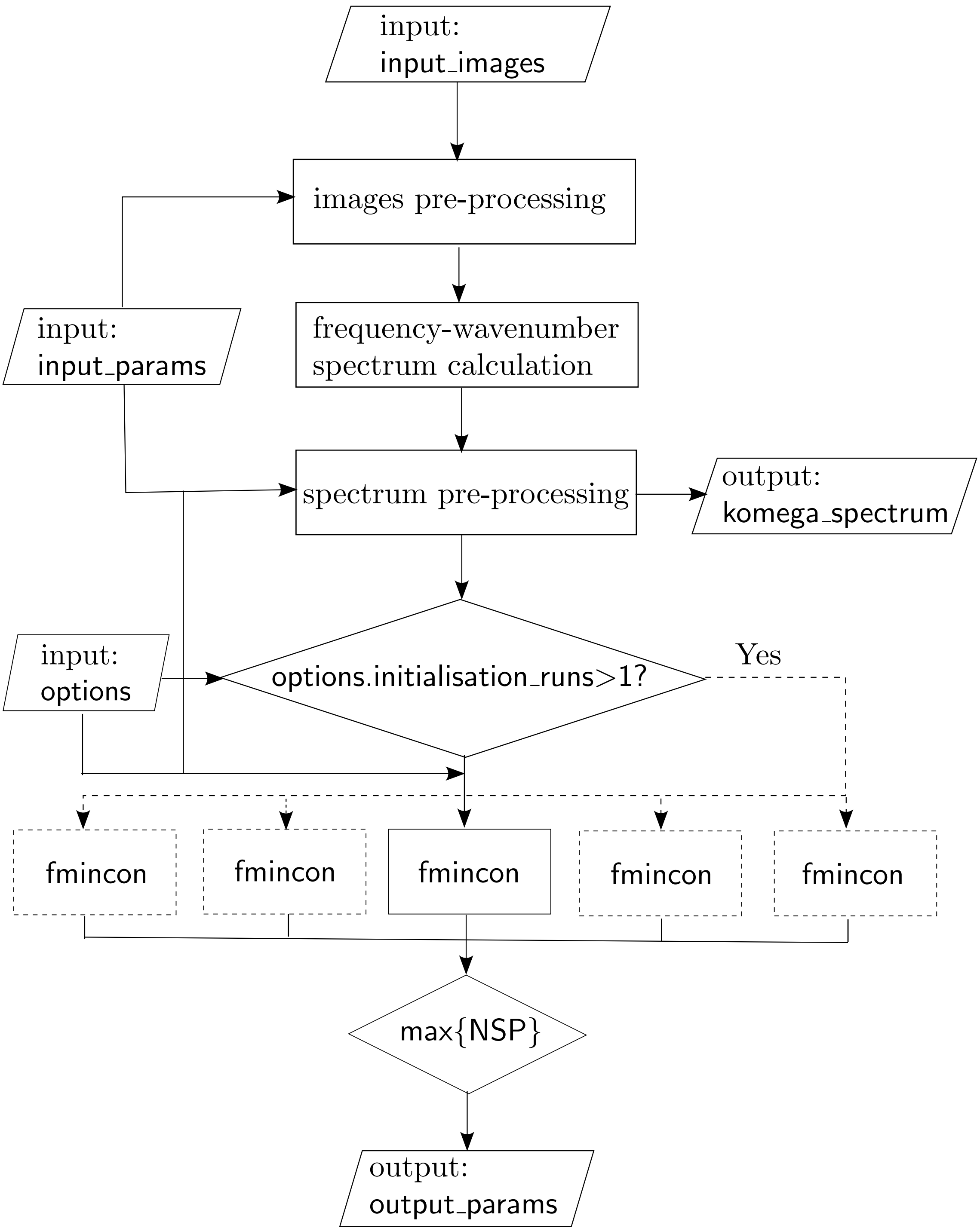

Figure 6: Main steps of kOmega algorithm.

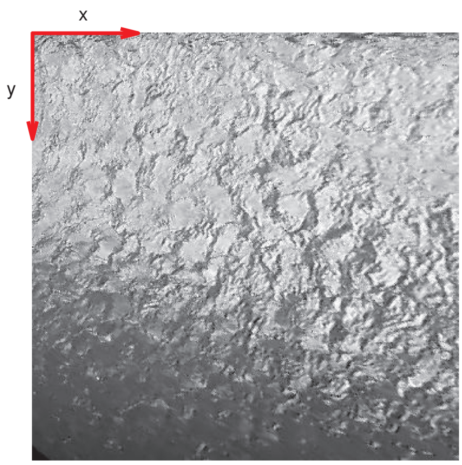

Figure 7: The system of reference used in kOmega.

By default, kOmega runs the optimisation only once, using the average of the lower and upper boundaries for each parameter as starting point (see 2.3). In the case of very noisy data, for small AOI size, or when the starting point is distant from the actual flow conditions, the optimisation algorithm may fail to identify the global maximum of the SNP and give erroneous results. Often these errors are easily identified by comparing the measured spectra with the theoretical relations of eq. (3) and (4) (see Fig. 6). For these cases, kOmega provides an option to run the optimisation multiple times with randomly selected starting points. The options is activated by setting the parameter options.initialisation_runs equal to the desired number of runs. The duration of the computation will increase accordingly. The optimal parameters are then identified as those with the highest SNP value across all runs.

2.3 Input data and parameters

The code requires three inputs: the array of images that need to be analysed (input_images); a structure array containing the input parameters (input_params); and a structure array containing options for the optimisation (options).

2.3.1 input_images

The input_images should be a set of ortho-rectified grayscale images provided in the form of a 3-dimensional array, where the 3 dimensions correspond to the spatial y-axis, to the spatial x-axis, and to the time (frame) axis, respectively. For example, input_images(:,:,10) is the 10-th frame. kOmega accepts input as grayscale images or double.

> Note: In kOmega, the x and y axis correspond to the first and second dimensions of the input images file, respectively. The positive x and y axis point in the direction of increasing indices. Therefore, the x-axis is the horizontal axis pointing towards the right. The y-axis is the vertical axis pointing downwards (see Fig. 7).

2.3.2 input_params

These parameters include metadata for the set of images, expected boundaries of the flow parameters, and pre-processing parameters. Some parameters are required, while others are optional. input_params must be provided as a structure array with the following fields:

input_params.fps	Frame rate of the input images.	s^{-1}
input_params.pxl_size	Size of a pixel in the physical space.	m
input_params.velocity_indx	Velocity index, i.e., ratio between the depth-averaged velocity $\bar{U}$ and the surface velocity U_s. See Hauet et al., 2018 for guidelines on the choice of this parameter[2]. Typical values are between 0.7 and 0.9, usually 0.83 to 0.85.	-
input_params.segment_duration	Segments duration. The whole dataset is segmented into shorter segments and the spectra of each segment are then averaged. Shorter segment durations help the noise converge but also reduce the spectral resolution.	s
input_params.overlap (optional)	Default 0. Overlap between consecutive segments. input_params.overlap must be a numerical value between 0 and 1, where 0 means no overlap and 0.3 means 30% overlap. An overlap can be useful in the case of a small number of frames to improve the spectrum convergence.	-
input_params.boundaries.velx	[lb,ub] where lb and ub are the minimum and maximum allowed values for the target parameter velx. Setting lb = ub effectively fixes the parameter velx = lb = ub.	m s^{-1}
input_params.boundaries.vely	[lb,ub] where lb and ub are the minimum and maximum allowed values for the target parameter vely. Setting lb = ub effectively fixes the parameter vely = lb = ub.	m s^{-1}
input_params.boundaries.depth	[lb,ub] where lb and ub are the minimum and maximum allowed values for the target parameter depth. Setting lb = ub effectively fixes the parameter depth = lb = ub.	m
input_params.depth (optional)	Default [ ]. Fixed water depth value, so that output_params.depth = input_params.depth. If input_params.depth is not declared or if input_params.depth = [ ], then the estimation of the water depth is attempted.	m
input_params.GravityWaves (optional)	Default 'on'. Set input_params.GravityWaves = 'off' to ignore the spectrum of gravity-capillary waves, if needed.	-
input_params.TurbulenceWaves (optional)	Default 'on'. Set input_params.Turbulence Waves = 'off' to ignore the spectrum of turbulence-generated waves, if needed.	-

[2]Note that unlike typical non-contact velocimetry methods where the velocity index is only used for estimating the depth-averaged velocity and thus calculating the flow rate, here the velocity index also affects the wave dynamics (see eq. (4)). Therefore a different value of input_params.velocity_indx will yield different estimates of the flow velocity and depth.

2.3.3 options

These parameters include various options for the optimisation. In addition to a few dedicated parameters listed below, any option for the fmincon optimiser described in the fmincon documentation can be passed within the field options.optimoptions using the dedicated Matlab function optimoptions.

options.output (optional)	Default: '2DSpectra'. Defines the form of the komega_spectrum output structure. By default, this structure contains the 2D cross-sections along the x and y-directions and the theoretical equations (3) and (4). Setting options.output = '3DSpectra' forces the full 3D spectra to be passed to the output.
options.initialisation_runs (optional)	Default: 1. Number of runs for the optimisation. If options.initialisation_runs = 1 the optimisation is run only once and the starting point for the optimiser is the average of the lower and upper boundaries for each parameter. If options.initialisation_runs >1, the optimisation is run multiple times with initial points chosen randomly each time within the provided boundaries. This option can help with noisy data or data with small AOI's when the optimiser can struggle to identify the global optimum.
options.parallel (optional)	Default: 'off'. Parallel computing. If options.parallel = 'on' the multiple optimisation runs are run in parallel on multiple workers (using a parfor loop). This option should only be used if options.initialisation_runs >1.
options.SNR_factor (optional)	Default: 1. Signal-to-Noise-Ratio threshold. The portion of the measured frequency-wavenumber spectrum with amplitude $<$ options.SNR_factor $\times$ the average of the spectrum is identified as a noise-floor and removed from the computation. Setting options.SNR_factor <1 lowers the threshold and could improve the performance with noisy data.
options.GaussWidth (optional)	Default: 1. Width of the Gaussian weighing function. Larger values may improve the performance with noisy data or in the presence of strong velocity gradients.
options.optimoptions (optional)	Use the syntax options.optimoptions = optimoptions(...) to pass any option to the fmincon optimiser. Refer to the official Matlab documentation for the available options.[3]

[3] The examples included with the script show a successful application of the method obtained with the following options: options.optimoptions = optimoptions('fmincon','Algorithm','interior-point','Display','notify','FunValCheck','on','OptimalityTolerance',1e-12,'PlotFcn','optimplotx'); These examples could serve as a starting point for the identification of the optimal parameters, which will ultimately depend on the input data.

2.4 Outputs

By default, kOmega attempts to estimate the three following parameters:

output_params.velx	x-component of the time-averaged surface flow velocity	m s^{-1}
output_params.vely	y-component of the time-averaged surface flow velocity	m s^{-1}
output_params.depth	time-averaged water depth	m

The value of each parameter can be fixed if needed by setting identical lower- and upper-boundary values for that parameter (see Sect. 2.3). An alternative way to fix the water depth is to set the input parameter input_params.depth equal to the desired value (see Sect. 2.3). Both methods are equivalent.

kOmega also outputs the following optimisation diagnostics:

output_params.NSP	Normalised Scalar Product (eq. (7)) calculated for the optimal set of output parameters.
output_params.info	This is a streaming of the output field of the fmincon optimiser, with the diagnostics of the optimisation.

Additionally, kOmega outputs the 2D cross-sections of the measured frequency-wavenumber spectra along the x and y-directions, and the data to plot eq.(3) and (4) according to the estimated velocity and depth:

komega_spectrum.Spectrum_kx	Pre-processed frequency wavenumber spectrum[1]. Cross-section along the x-direction (with $k_y = 0$).	
komega_spectrum.Spectrum_ky	Pre-processed frequency wavenumber spectrum[1]. Cross-section along the y-direction (with $k_x = 0$).	
komega_spectrum.fTurbx	Theoretical frequency of turbulence-generated waves, eq. (3), as calculated based on the estimated flow velocity and water depth. Cross-section along the x-direction, $f(k_x)$.	s^{-1}
komega_spectrum.fTurby	Theoretical frequency of turbulence-generated waves, eq. (3), as calculated based on the estimated flow velocity and water depth. Cross-section along the y-direction, $f(k_y)$.	s^{-1}
komega_spectrum.fGWx_plus	Theoretical frequency of gravity-capillary waves, eq. (4), as calculated based on the estimated flow velocity and water depth. Cross-section along the x-direction, $f(k_x)$.	s^{-1}
komega_spectrum.fGWy_plus	Theoretical frequency of gravity-capillary waves, eq. (4), as calculated based on the estimated flow velocity and water depth. Cross-section along the y-direction, $f(k_y)$.	s^{-1}

[1]Note that this is the non-dimensional normalised spectrum used for the optimisation, which has been pre-processed in order to facilitate the fitting, and which may differ significantly from the raw frequency-wavenumber spectrum.

komega_spectrum.fGWx_minus	Like komega_spectrum.fGWx_plus, but considering the solution with the '-' sign.	s^{-1}
komega_spectrum.fGWy_minus	Like komega_spectrum.fGWy_plus, but considering the solution with the '-' sign.	s^{-1}

If the full 3D arrays of the frequency-wavenumber spectra are needed, these can be obtained in alternative to the 2D cross-section by setting options.output = 'Spectra_projections' in the input options structure. In that case, the komega_spectrum structure will be as follows:

The following outputs require setting options.output = 'Spectra_projections'		
komega_spectrum.kx	k_x array.	rad m^{-1}
komega_spectrum.ky	k_y array.	rad m^{-1}
komega_spectrum.f	f array.	s^{-1}
komega_spectrum.Spectrum	Pre-processed 3D frequency wavenumber spectrum[1].	
komega_spectrum.fTurb	Theoretical frequency of turbulence-generated waves, eq. (3), as calculated based on the estimated flow velocity and water depth. 2D array, $f(k_x, k_y)$	s^{-1}
komega_spectrum.fGW_plus	Theoretical frequency of gravity-capillary waves, eq. (4), as calculated based on the estimated flow velocity and water depth. 2D array, $f(k_x, k_y)$	s^{-1}
komega_spectrum.fGW_minus	Like komega_spectrum.fGW_plus, but considering the solution with the '-' sign.	s^{-1}

3 Reusing and Sharing

This work is licensed under a Creative Commons CC BY 4.0 license. This license allows reusers to distribute, remix, adapt, and build upon the material in any medium or format, so long as attribution is given to the creator. The license allows for commercial use. If you remix, adapt, or build upon the material, you must license the modified material under identical terms.

When sharing products obtained with kOmega or derived from kOmega and/or from this document please reference the following article:

Dolcetti, G., Hortobágyi, B., Perks, M., Tait, S. J. & Dervilis, N. (2022). Using non-contact measurement of water surface dynamics to estimate water discharge. *Water Resources Research, 58(9)*, e2022WR032829. https://doi.org/10.1029/2022WR032829.

[1]Note that this is the non-dimensional normalised spectrum used for the optimisation, which has been pre-processed in order to facilitate the fitting, and which may differ significantly from the raw frequency-wavenumber spectrum.

4 Support

For support, questions, and to flag eventual bugs, please contact Dr Giulio Dolcetti: giulio.dolcetti@unitn.it

5 Acknowledgements

This work was funded by the UK Engineering and Physical Sciences Research Council Grant No. EP/R022275/1 and by the UK Natural Environment Research Council Grant No. NE/K008781/1 “Susceptibility of catchments to INTense RAinfall and flooding (SINATRA).”

References

Brocchini, M., & Peregrine, D. H. (2001). The dynamics of strong turbulence at free surfaces. part 1. description. *Journal of Fluid Mechanics*, *449*, 225–254. https://doi.org/10.1017/S0022112001006012

Dolcetti, G., Horoshenkov, K. V., Krynkin, A., & Tait, S. J. (2016). Frequency-wavenumber spectrum of the free surface of shallow turbulent flows over a rough boundary. *Physics of Fluids*, *28*(10), 105105. https://doi.org/10.1063/1.4964926

Dolcetti, G., Hortobágyi, B., Perks, M., & Tait, S. J. (2020). The effect of surface gravity waves on the measurement of river surface velocity. *River Flow 2020: Proceedings of the 10th Conference on Fluvial Hydraulics*. https://doi.org/10.1201/b22619

Dolcetti, G., Hortobágyi, B., Perks, M., Tait, S. J., & Dervilis, N. (2022). Using noncontact measurement of water surface dynamics to estimate river discharge. *Water Resources Research*, *58*(9), e2022WR032829. https://doi.org/10.1029/2022WR032829

Hauet, A., Morlot, T., & Daubagnan, L. (2018). Velocity profile and depth-averaged to surface velocity in natural streams: A review over alarge sample of rivers. *E3s web of conferences*, *40*, 06015. https://doi.org/10.1051/e3sconf/20184006015

Muraro, F., Dolcetti, G., Nichols, A., Tait, S. J., & Horoshenkov, K. V. (2021). Free-surface behaviour of shallow turbulent flows. *Journal of Hydraulic Research*, *59*(1), 1–20. https://doi.org/10.1080/00221686.2020.1870007

Muste, M., Fujita, I., & Hauet, A. (2008). Large-scale particle image velocimetry for measurements in riverine environments. *Water resources research*, *44*(4). https://doi.org/10.1029/2008WR006950

Qin, A. K., & Suganthan, P. N. (2005). Self-adaptive differential evolution algorithm for numerical optimization. *2005 IEEE congress on evolutionary computation*, *2*, 1785–1791. https://doi.org/10.1109/CEC.2005.1554904

