Water Utility Journal 30-31: 33-42, 2022.

OX FreeSurf: Automated free surface calculation in open-channel flow

Athanasios V. Serafeim[1], Anastasios Perdios[1], Dimitrios Gkikas[3], Nikolaos Th. Fourniotis[2,3] and Andreas Langousis[1,3*]

[1] *Department of Civil Engineering, University of Patras, Patras, Greece*
[2] *Department of Civil Engineering, University of the Peloponnese, Patras, Greece*
[3] *School of Science and Technology, Hellenic Open University, Patras, Greece*
[*] *e-mail: andlag@upatras.gr*

Abstract: OX FreeSurf is an open-source software that uses basic principles of open-channel hydraulics to effectively simulate one-dimensional, steady state, non-uniform, free-surface flows in open channels. The developed software is fully automated and it is characterized by simplicity in its use as well as parsimony in input entries, while exhibiting a friendly graphical user interface (GUI). The software becomes available in two forms: a) as an open-source code written in Python programming language, and b) as a single self-extracting installation file for Windows 8.1 or higher. In the latter case no additional actions are required by the user, as the software is installed automatically and can be run directly from the created icons. The functionality of OX FreeSurf was examined using two application examples of open-channel flow. The numerically calculated free-surface profiles were found in quantitative agreement with those obtained by applying the HEC-RAS model as well as one-dimensional analysis of the flow. Given its full automation and simplicity in use, OX FreeSurf is expected to serve as a useful tool/platform for educational/academic purposes, research, and engineering practice in general.

Key words: Open-channel hydraulics; hydraulic jump; hydraulic drop; gradually-varied-flow; prismatic channels; free surface flow; free surface profiles; Python.

1. INTRODUCTION

Computation of the steady state free-surface profile in open channel flow, requires numerical solution of the dynamic equation of gradually-varied-flow (see e.g., Chow, 1959; Henderson, 1966; Mays, 2001; Douglas et al., 2005; Akan, 2006; Cengel and Cimbala, 2014; Ponce, 2014), to determine how the water depth varies along a given channel layout. For this purpose, the hydraulic engineer needs to: a) partition the channel into short reaches with constant bottom slope and given cross-sectional geometry, b) determine the normal and critical depths in each reach, c) use the given set of boundary conditions (i.e., the inflow and outflow depths), along with critical judgment driven by in depth knowledge of hydraulic principles, to determine the control sections of the flow, the corresponding flow profiles (i.e., drawdown and backwater curves), as well as the location of hydraulic jumps along the channel (i.e., a process usually referred to as qualitative flow profile analysis (see e.g., Chow, 1959; Langousis and Fourniotis, 2020), and d) integrate the dynamic equation of gradually-varied-flow between the identified control sections, properly taking into account abrupt depth changes caused by hydraulic jumps (i.e. by solving the specific force continuity equation at locations where the flow transitions from supercritical to subcritical; see Section 2.3). Actually, implementation of steps (a) - (d) constitutes a demanding but fundamental part of most engineering courses in open channel hydraulics taught in civil engineering departments worldwide (see e.g., Chow, 1959; Henderson, 1966; Mays, 2001; Douglas et al., 2005; Akan, 2006; Noutsopoulos et al., 2007; Chadwick et al., 2013; Cengel and Cimbala, 2014; Ponce, 2014; Demetracopoulos, 2018; Langousis and Fourniotis, 2020).

Under this setting, an easy to use online platform for hydraulic calculations (http://uon.sdsu.edu/online_calc.php) has been developed for educational purposes and beyond, by Professor V.M. Ponce (http://ponce.sdsu.edu/). The online platform allows for calculation of normal and critical depths, drawdown and backwater curve profiles, among many other flow related variables and, therefore, it can serve as a useful tool for implementation of steps (b) and (d) above, but it is not suited for hydraulic simulation of steady state one-dimensional free-surface flows, as the latter operation requires implementation, also, of steps (a) and (c). For the latter purpose, there exist several operational software platforms, including HEC-RAS (see USACE, 2016 and https://www.hec.usace.army.mil/software/hec-ras/), MIKE+ powered by DHI (https://www.mikepoweredbydhi.com/), Channel Studio offered by Hydrology Studio (see https://www.hydrologystudio.com/), and OpenFlows FlowMaster powered by Bentley (see https://www.bentley.com/en/products/product-line/hydraulics-and-hydrology-software). Important drawbacks of the foregoing platforms are that: a) with the exception of HEC-RAS, their use is not free of charge, b) they are not open-source, thus, not allowing for interventions/modifications by advanced users, and c) they are suited to address more complex problems than one-dimensional, steady state, free-surface flow integration in open channels, making problem setup by elementary users for educational purposes and routine design applications a rather complex task.

To address the needs of students, educators, engineering practitioners and researchers, while not competing with existing commercial tools, we developed OX FreeSurf, an open-source software written in Python programming language that uses basic principles of open channel hydraulics to effectively simulate one-dimensional, steady state, non-uniform, free-surface flows in prismatic open channels. To the best of our knowledge, there is no available open-source software for this purpose that is fully automated, characterized by simplicity in its use and parsimony in input entries, while exhibiting a friendly graphical user interface (GUI), making it suitable for educational/academic purposes and routine design applications. The software becomes available in two forms: a) as an open-source code written in Python programming language, allowing for case-specific modifications/interventions by advanced users, as well as possible extensions to address research questions, and b) as a single self-extracting installation file (i.e., OX_FreeSurf_setup.exe) for Windows 8.1 or higher. In the latter case no additional actions are required by the user, as the software is installed automatically and can be run directly from the created icons.

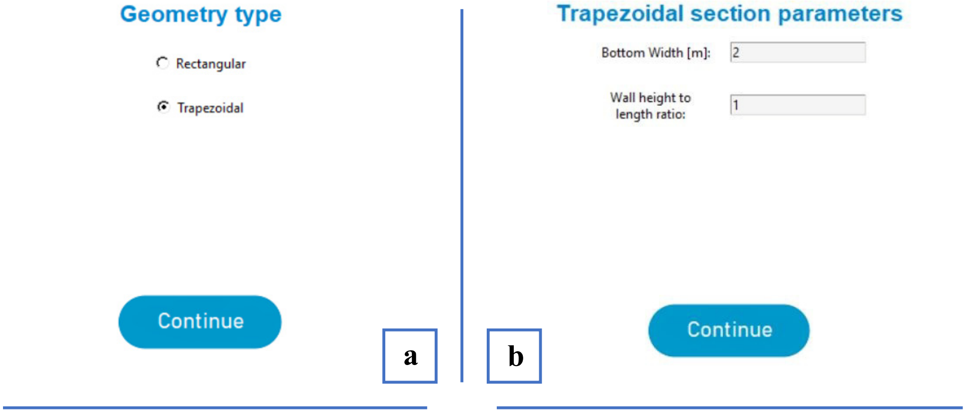
Section 2 outlines the basic architecture of OX FreeSurf, its components and all required details to run the open-source code, followed by two application examples presented in Section 3. Section 4 summarizes the expected impact of OX FreeSurf, and concludes with some remarks and possibilities for future development.

2. SOFTWARE DESCRIPTION

Before executing the open-source code of OX FreeSurf, the user should first download and install an open-source Python release (version 3.7 or higher), which runs smoothly (i.e., not encountering malfunctions/bugs; check maintenance status at python.org/downloads). After successful installation of Python, the user should also install: a) the *tkinter* python package, b) the *pandas* and *numpy* libraries, c) the *matplotlib* library, and d) the modules *sys*, *time* and *os*.

2.1 Input configuration

Problem configuration is accomplished through the OX FreeSurf GUI windows (see Figures 1 and 2). First, the user should select the geometry of the hydraulic section of the open channel (i.e., rectangular or trapezoidal; Figure 1.a) and then set the bottom width, b, and the side slope z (i.e., the wall width to height ratio; see Figures 1.b and 1.c). It is noted that in the case of a rectangular section, the value of z is set by default equal to zero. Next, the user is prompted to set the flowrate value Q [m^3/s], the upstream and downstream boundary conditions (i.e., the inflow depth d_{in} [m] and the outflow depth d_{out} [m]), the Gauckler-Manning coefficient n [m$^{-1/3}$s], as well as the constant step to be used for the numerical integration of the free-surface profile dx [m] (see Figures 2.a and 2.c). As a final step, the user defines the number of nodes of the polyline that best describes the altimetry of the channel's reaches (no limitation exists for the maximum number of nodes), and specifies their coordinates; i.e., *X*: horizontal, and *Y*: vertical (see Figures 2.b and 2.c).



Geometry type

○ Rectangular
● Trapezoidal

Trapezoidal section parameters

Bottom Width [m]: 2
Wall height to length ratio: 1

c Trapezoidal section

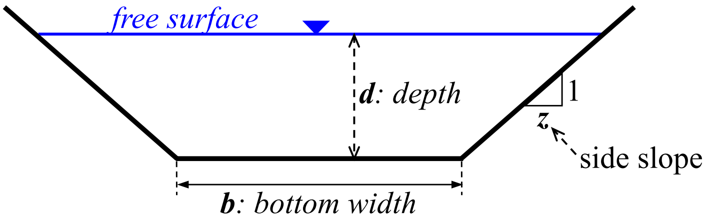
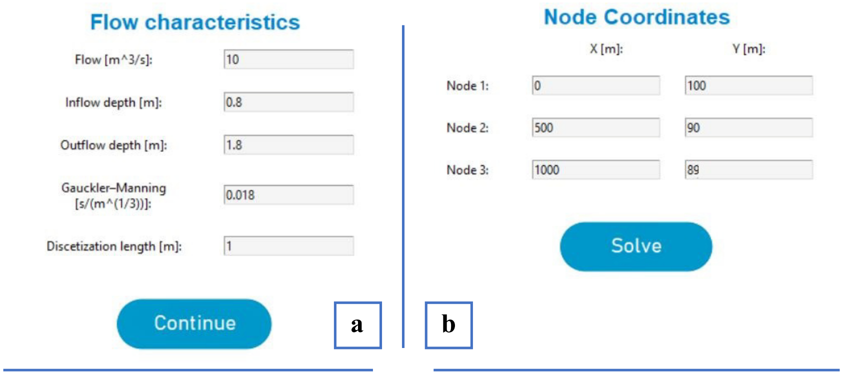


Figure 1. a, b) Main windows of OX FreeSurf corresponding to the Geometry_selector class. c) Schematic illustration of a trapezoidal hydraulic section



Flow characteristics

Flow [m^3/s]: 10
Inflow depth [m]: 0.8
Outflow depth [m]: 1.8
Gauckler–Manning [s/(m^(1/3))]: 0.018
Discetization length [m]: 1

Node Coordinates

	X [m]:	Y [m]:
Node 1:	0	100
Node 2:	500	90
Node 3:	1000	89

c Schematic illustration of channel reaches and flow characteristics

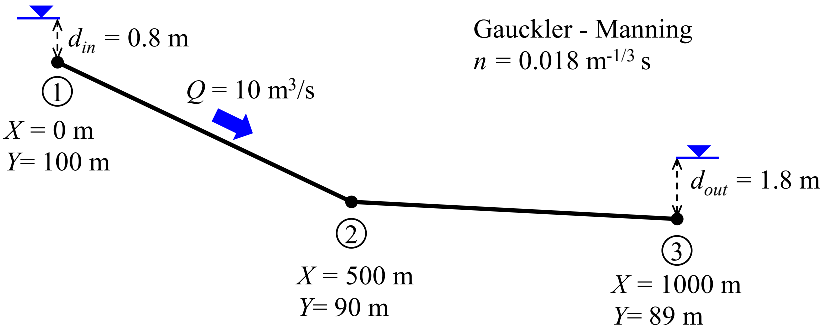


Figure 2. a, b) Main windows of OX FreeSurf corresponding to the Flow_characteristics class. c) Schematic illustration of channel reaches and flow charactreristics, where the parameter entries in panels (a) and (b) are shown.

2.2 Software architecture

OX FreeSurf consists of two main components: a) ox_gui.py, and b) ox_solver.py. ox_gui.py contains the main code of the software, which is structured in 8 Classes, including the design of the GUI main windows, the call to the solver (i.e., ox_solver.py), and error handling during execution. More specifically:

- Class *Splash* generates the initial splash screen of the software.
- *Free_surf_gui* is the main parent class, where the user sets the inputs (i.e., Q, b, z, n, d_{in}, d_{out} and dx) as a shared dictionary (code lines 168-176) between the children classes (i.e., *Geometry_selector*, *Rectangular*, *Trapezoidal*, *Flow_characteristics*, *River_drawing*, and *Reboot_error*).
- *Geometry_selector class* allows the user to choose either the rectangular or the trapezoidal shape for the hydraulic section of the prismatic channel, by leading to the respective classes (see Figures 1.a and 1.b).
- *Flow_characteristics* class passes the user defined flow related variables Q, n, d_{in}, d_{out} and dx (code lines 368-406) to the shared dictionary defined in the parent class *Free_surf_gui*; see Figure 2.a.
- *River_Drawing* class allows the user to define the number of nodes of the polyline that best describes the altimetry of the channel's reaches (code lines 443-448), as well as their coordinates (code lines 471-477). Then, it passes all user inputs to ox_solver.py (code lines 556-559), which returns the free surface elevation along the channel at locations spaced by dx. After successful execution of ox_solver.py, a new window appears that includes a plot of the channel's longitudinal section, together with the corresponding elevations of the free surface (code lines 574-595), where the user can zoom in and out to focus on specific locations of interest (see Figure 3). The user can also choose to export the plot in .jpg (picture) and/or .xlsx (table) formats (code lines 601-604) and, optionally, start a new simulation.
- *Reboot_error* class generates a GUI window, which informs the user in case of an error during execution.

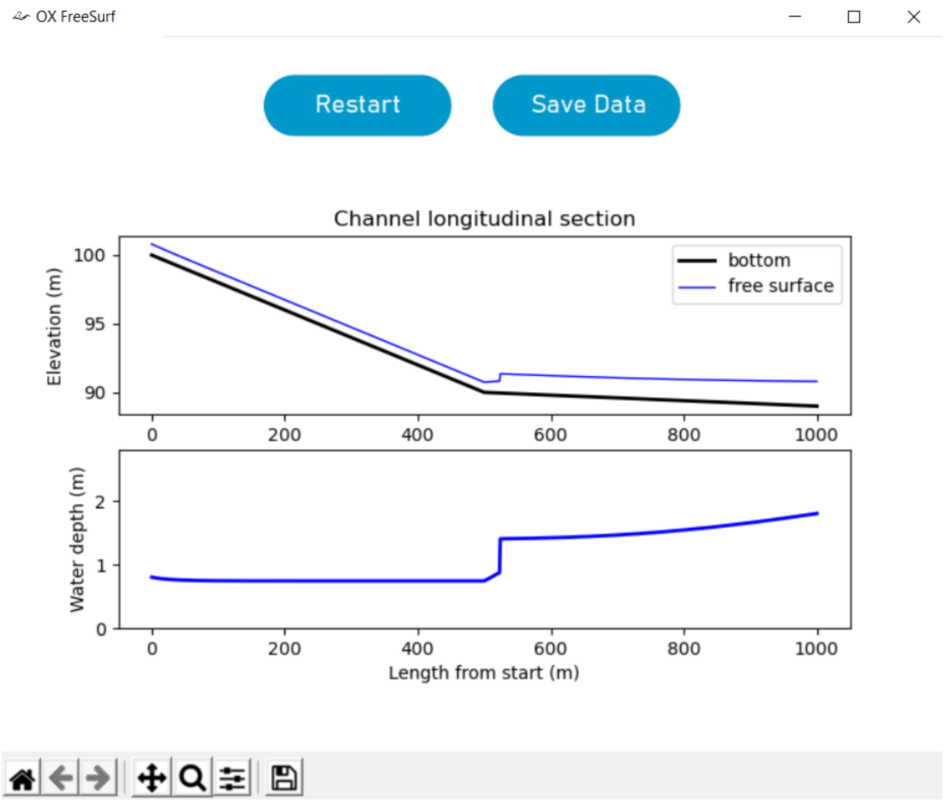


Figure 3. Output screen of OX FreeSurf for the entries shown in Figures 1 and 2.

2.3 OX main engine: Free-surface profile calculator

ox_solver.py includes the source code for the calculation of the free-surface profile along the channel, based on the user defined altimetry of reaches and their specific hydraulic characteristics (geometry of hydraulic cross section, flowrate, inflow and outflow depths etc.; see Figures 1 and 2). Initially, the code checks the user inputs for possible inconsistencies (e.g., negative flowrates, zero or negative geometric distances etc.; code lines 140-148), and then estimates the critical depth $d_{critical}$ (code lines 152, 423-462), the minimum specific energy E_{min} (code lines 153, 465-484), and the minimum specific force M_{min} (code lines 154, 487-504) of the flow (e.g., Chow, 1959; Mays, 2001; Akan, 2006; Chadwick et al., 2013; Cengel and Cimbala, 2014, Ponse, 2014). E_{min} and M_{min} act as constraints, being the minimum possible values of specific energy and force required for a flow with the user defined characteristics to be possible.

Integration of the free-surface profile proceeds using the standard step method (see e.g., Chow, 1959; Mays, 2001; Akan, 2006; Chadwick et al., 2013; Ponse, 2014]), and requires, as a first step, discretization of the channel's longitudinal section into equally sized segments of length *dx* (i.e., the user defined discretization step; see Figure 2, and code lines 167-169). The value of *dx* directly affects the accuracy of the free surface integration, and should be on the order of 1 m or less. After effective discretization of the channel, each segment is characterized based on its slope according to the direction of the flow, which defines the corresponding free-surface profile of the gradually-varied-flow (i.e., A: Adverse, C: Critical, H: Horizontal, M: Mild or S: Steep, see code lines 175-201 and Chow, 1959; Henderson, 1966; Mays, 2001; Douglas et al., 2005; Akan, 2006; Cengel and Cimbala, 2014; Chadwick et al., 2013, Ponse, 2014 for a review). Based on the foregoing classification, the control sections along the channel at which the water depth is known (see e.g., Chow, 1959; Akan, 2006; Langousis and Fourniotis, 2020) are identified, and their location is stored in list *ctrl* (code lines 206-230). The latter list is used to create list *locctr*, which contains the location of all control sections, separating the channel into shorter reaches within which the numerical integration should proceed either downstream (in case of supercritical flow), or upstream (in case of subcritical flow); see below.

For the portions of the channel where numerical integration proceeds downstream of the control section, flow depth values are calculated using function slowly_varying_inter (code lines 254, 660-738), and the obtained results are stored in list *dnsrt* (code lines 255-286). More in detail, the algorithm first checks whether there is only one control section along the channel, excluding its end point. If there is only one control section (code lines 246-255), the free surface profile is integrated starting from the control section to the channel's end. In the case of more than one control sections (code lines 256-294), integration proceeds till the location corresponding to the next entry in *locctr* list, and the resulting depth is compared to the critical depth of the flow $d_{critical}$. In case the resulting depth is less than $d_{critical}$, the corresponding entry in the *locctr* list ceases to be a control section and it is registered in list *flag_elim_ctrl* list. The same procedure is carried out till the last control section along the channel (excluding its end), and the final *flag_elim_ctrl* list is used to update lists *ctrl* and *dnstr* before the algorithm proceeds with upstream integration in the corresponding portions of the channel (code lines 298-327).

Upstream integration is controlled by the updated *locctr* list (code lines 331-337), and the resulting flow depths are stored in list *upstr*. The algorithm first checks whether there is only one control section along the channel, excluding its start point. If there is only one control section (code lines 341-350), the free-surface profile is integrated starting from the control section to the channel's starting point. In the case of more than one control sections (code lines 351-364), integration proceeds till the location corresponding to the preceding entry in *locctr* list.

The final flow *profile* list is defined by combining the *dnsrt* and *upstr* lists (code lines 368-415). For those parts of the channel where two free-surface profiles are available (i.e., one supercritical and one subcritical), the algorithm: a) identifies the exact locations of hydraulic jumps; i.e., the locations where the calculated supercritical and subcritical flow depths match the corresponding conjugate depths (see e.g., Chow, 1959; Henderson, 1966; Mays, 2001; Douglas et al., 2005; Akan, 2006; Chadwick et al., 2013; Cengel and Cimbala, 2014; Ponce, 2014), and b) eliminates the subcritical (supercritical) flow depths upstream (downstream) of the hydraulic jump, as non-feasible. Finally, the resulting free-surface profile along the channel is returned to ox_gui.py (code line 418), for further processing by the user, in the form of image and/or data table (see Figure 3). For illustration, Figure 4 presents a flow chart of the software.

Please note that, similar to HEC-RAS, and in order to keep the interpretation of the free surface profile as simple as possible (see application examples in Section 3), the current version of the software does not take into account the length of the hydraulic jump, as the latter is generally very small compared to the length of the channel reaches, affecting minimally the obtained results.

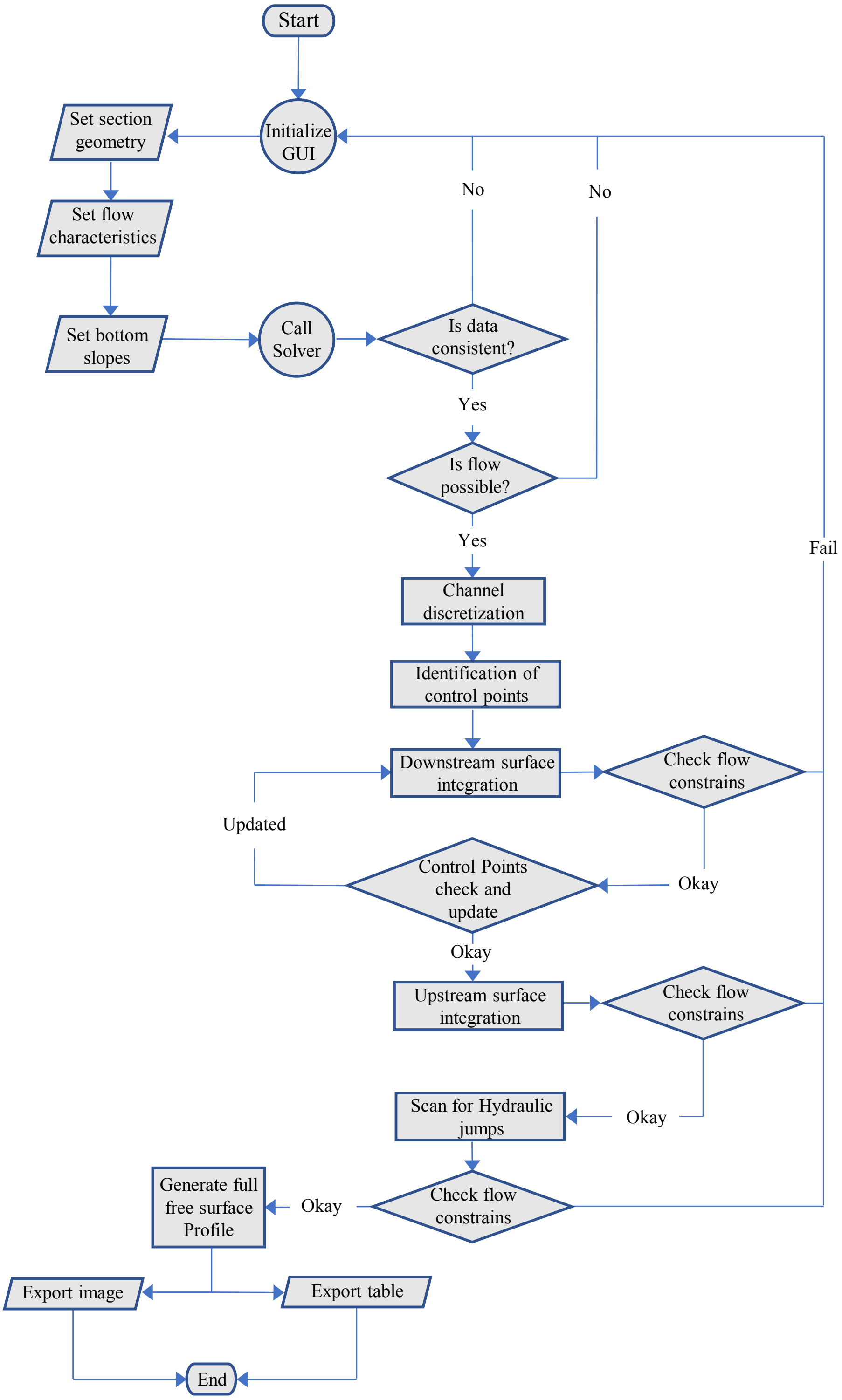


Figure 4. OX FreeSurf flowchart.

3. APPLICATION EXAMPLES

In this section, we demonstrate the functionality of OX FreeSurf using two application examples. More precisely, we compare the free-surface profiles calculated by OX FreeSurf with those obtained using HEC-RAS: a widely used software for free-surface flow calculations in open channels; see e.g., Mays (2001), Lee et al. (2006), Rao and Hromadka (2016), Gkikas (2019) and Introduction. Both examples regard calculation of the free-surface profile in a rectangular open channel with bottom width b = 2 m, flowrate Q = 10 m^3/s, Gauckler-Manning coefficient n = 0.018 m$^{-1/3}$s, and boundary conditions: d_{in} = 1 m (inflow depth), and d_{out} = 1.4 m (outflow depth). The discretization step used, in both cases, for the numerical integration of the free-surface profile is dx = 1 m.

In Example 1, the channel is composed by two distinct reaches: reach AB has supercritical slope J_{AB} = 3% and is followed by reach BC with subcritical slope J_{BC} = 2‰ (see black line in Figure 5.a, and Table 1 for the coordinates of points A, B and C of the channel's longitudinal section).

Table 1. Coordinates of the points/nodes that define the longitudinal sections of the two channels in Examples 1 and 2 (see main text for details).

	Coordinates (*X*, *Y*) (m)				
	Node A	Node B	Node C	Node D	Node E
Example 1	(0, 32)	(1000, 2)	(2000, 0)	-	-
Example 2	(0, 100)	(300, 91)	(600, 90.4)	(900, 81.4)	(2000, 79.2)

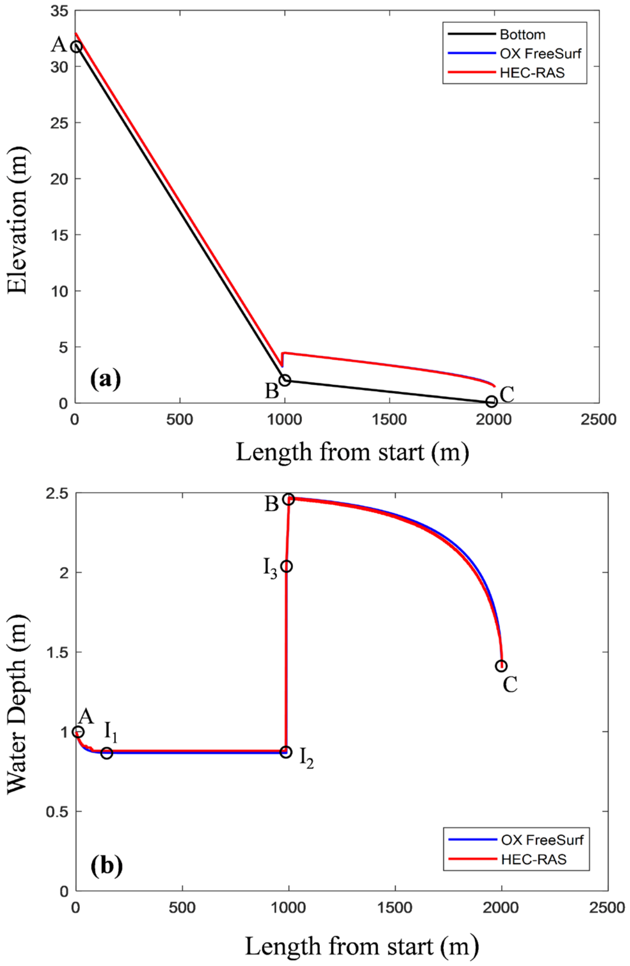


Figure 5. a) Water surface elevation, and b) water depth, along the channel in Example 1, as calculated by OX FreeSurf (blue line) and HEC-RAS (red line).

Example 2 concerns a more complex open channel, which exhibits three changes of bottom slope, and is composed by four distinct reaches: reach AB with supercritical slope J_{AB} = 3%, reach BC with subcritical slope J_{BC} = 2‰, reach CD with supercritical slope J_{CD} = 3%, and reach DE with subcritical slope J_{DE} = 2‰ (see black line in Figure 6.a, and Table 1 for the coordinates of points A, B, C, D and E of the channel's longitudinal section).

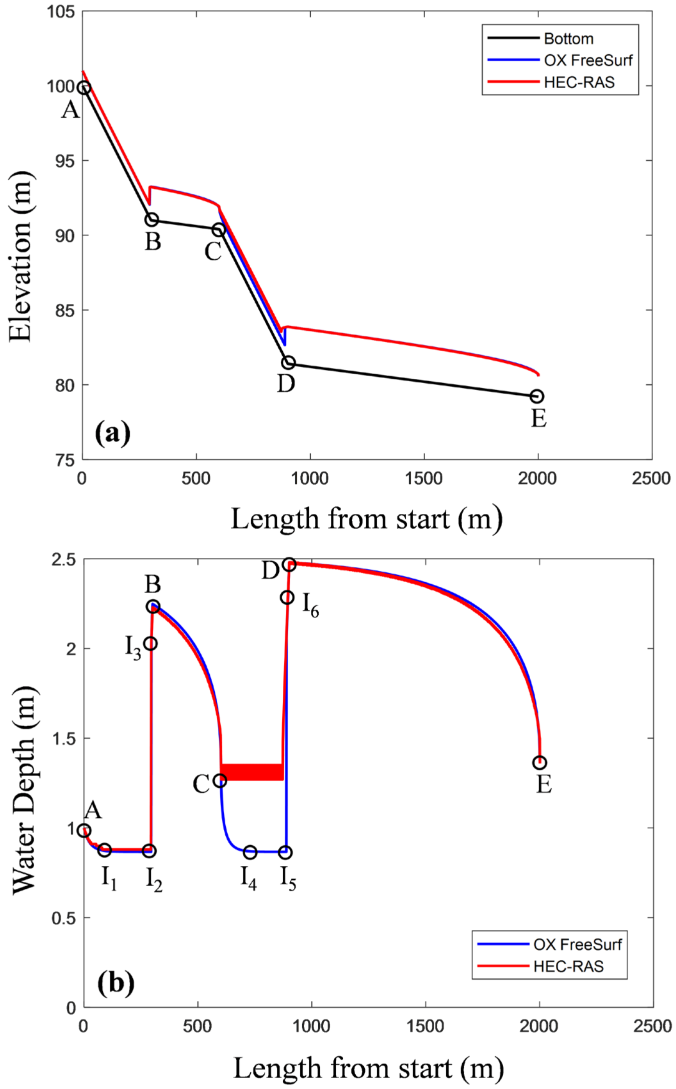


Figure 6. a) Water surface elevation, and b) water depth, along the channel in Example 2, as calculated by OX FreeSurf (blue line) and HEC-RAS (red line).

For the case of Example 1, Figures 5.a and 5.b show the water surface elevation and the water depth, respectively, along the channel as calculated by OX FreeSurf (blue line) and HEC-RAS (red line). The flow starts at section A with depth d_{in} = 1 m and proceeds to section I_1 where the depth of the flow becomes equal to the normal depth: d_n = 0.867 m (as calculated by OX FreeSurf and cross checked using the online platform: http://uon.sdsu.edu/online_calc.php for one-dimensional hydraulic calculations). The free-surface profile between sections A and I_1 follows a S_2 drawdown curve, as the normal depth lies below the inflow depth. Between sections I_1 and I_2 the flow remains uniform with normal depth d_n = 0.867 m, and at section I_2 a hydraulic jump occurs as the flow abruptly transitions from a supercritical state (at section I_2 with flow depth $d_{I_2} = d_n$ = 0.867 m) to a subcritical state; i.e., at section I_3 with flow depth d_{I_3} = 2.05 m. The latter depth value has been obtained from OX FreeSurf and, as expected, it equals the conjugate subcritical depth of d_{I_2} = 0.867 m (cross checked using the online platform http://uon.sdsu.edu/online_calc.php). Between sections I_3 and B the flow is subcritical with free-surface profile that follows a S_1 backwater curve with ending depth d_B = 2.47 m at section B. The flow in reach BC remains subcritical with the free-surface profile following a M_2 backwater curve till the outflow depth d_{out} = 1.4 m at Section C. In general, apart from a slight overestimation of the normal depth in reach I_1I_2 by HEC-RAS (i.e., d_n = 0.88 m) relative to OX FreeSurf and that resulting from one-dimensional analysis (i.e., d_n = 0.867 m), both OX FreeSurf and HEC-RAS provide accurate water depth estimates.

Similar to Figures 5.a and 5.b for Example 1, Figures 6.a and 6.b show the water surface elevation and the water depth, respectively, for the case of Example 2. Within reaches AB and BC the flow profile matches that calculated in Example 1, with the only difference that the ending depth of the M_2 profile at Section C, d_c, is equal to the critical depth $d_{critical}$ = 1.366 m (as calculated by OX FreeSurf and cross checked using the online platform: http://uon.sdsu.edu/online_calc.php). This is due to the slope change of the channel at section C from subcritical (i.e., within reach BC, J_{BC} = 2‰) to supercritical (i.e., within reach CD, J_{CD} = 3%), resulting in a hydraulic drop. As correctly computed by OX FreeSurf (blue line), between sections C and I_4 the free surface profile follows the S_2 drawdown curve until the normal depth is reached at section I_4 ($d_{I_4} = d_n$ = 0.867 m), and then the flow remains uniform till section I_5, where a hydraulic jump occurs, resulting in subcritical depth d_{I_6} = 2.05m at section I_6 (see also Example 1). However, the same does not hold for the free-surface profile calculated by HEC-RAS, as the water depth cannot capture the hydraulic drop at section C, with the computed water depths between sections C and I_5 fluctuating around the critical depth $d_{critical}$ = 1.366 m (the fluctuations appear as a thick red line in Figure 6.b); see USACE (2016) on possible numerical instabilities in HEC-RAS calculated free surface profiles. Between sections I_6 and D the flow is subcritical with free-surface profile that follows a S_1 backwater curve with ending depth d_D = 2.48m at section D. The flow in reach DE remains subcritical with the free-surface profile following a M_2 backwater curve till the outflow depth d_{out} = 1.4 m at Section E.

4. CONCLUSIONS

OX FreeSurf is an open-source software that uses basic principles of open channel hydraulics to effectively simulate one-dimensional, steady state, non-uniform, free-surface flows in open channels. It is fully automated, characterized by simplicity in its use and parsimony in input entries, while exhibiting a friendly graphical user interface (GUI). To the best of our knowledge, there is no available open-source software for this purpose, which may allow students, educators, engineering practitioners and researchers to conduct routine hydraulic calculations in the comfort of a user friendly interface and a widely used programming language. The results obtained by applying the software to two application examples were validated using one-dimensional analysis of the flow, and compared to HEC-RAS simulations for the free-surface profile.

Additional opportunities for future development include modifications and extensions of the developed software to: a) account for the effects of flow obstacles, gates, inline weirs etc., and b) generalize its applicability also to river reaches. In this case, it becomes important to establish a linkage between OX FreeSurf and existing CAD and GIS software platforms, so that the river geometry can be readily imported with a minimum of effort.

5. SOFTWARE AVAILABILITY AND REQUIREMENTS

OX FreeSurf software is openly available at GitHub: https://github.com/alangousis/OXFreeSurf in two forms: a) as an open-source code written in Python programming language, and b) as a single self-extracting installation file (i.e., OX_FreeSurf_setup.exe) for Windows 8.1 or higher. In the latter case no additional actions are required by the user, as the software is installed automatically and can be run directly from the created icons.

To execute the open-source code of OX FreeSurf, the user should first download and install an open-source Python release (version 3.7 or higher), which runs smoothly (i.e., not encountering malfunctions/bugs; check maintenance status at python.org/downloads). After successful installation of Python, the user should also install: a) the *tkinter* python package, b) the *pandas* and *numpy* libraries, c) the *matplotlib* library, and d) the modules *sys*, *time* and *os*. Table 2 below summarizes some additional information regarding the distributed code and self-extracting installation file.

Table 2. OX FreeSurf software metadata.

Code metadata description	
Current code version	v1.0.1
Permanent link to code/repository used of this code version	https://github.com/alangousis/OXFreeSurf
Legal Code License	*Apache 2.0*
Software code languages, tools, and services used	Python 3.7, Inno Setup Compiler 6.2.1
Compilation requirements, operating environments & dependencies	Windows 8.1 and above
Support email for questions	athanseraf@hotmail.com, andlag@upatras.gr

ACKNOWLEDGMENTS

This work has been supported by the Region of Western Greece as part of the research project "Investigation of the hydraulic behavior and suggestions for the regulation of the hydrodynamic system of Trichonida - Lysimachia lakes, in Western Greece". The Authors are grateful to two anonymous Reviewers, the Associate Editor and the Editor in Chief, for their insightful comments and suggestions, which enhanced the quality of the presented work.

REFERENCES

Akan, A.O., 2006. Open Channel Hydraulics. 1st edition, Elsevier, 377 p.

Cengel, Y.A., Cimbala, J.M., 2014. Fluid Mechanics Fundamentals and Applications. 3rd edition, McGraw-Hill, New York, 1031p.

Chadwick, A., Morfett J., Borthwick, M., 2013. Hydraulics in Civil and Environmental Engineering. 5th edition, CRC Press, Taylor & Francis Group

Chow, V.T., 1959. Open-channel Hydraulics. McGraw-Hill Book Company, New York, 680 p.

Demetracopoulos, A.C., 2018. Elements of Hydraulics for Closed and Open Conduits. Gotsis Publications, Greece, 197 p. (in Greek)

Douglas, J.F., Gasiorek, J.M., Swaffield J.A., Jack, L.B., 2005. Fluid Mechanics. 5th edition, Prentice Hall, 993 p.

Gkikas, D., 2019. Development of an open-source software for hydraulic simulation of one-dimensional free surface flow in a rectangular channel. MSc Thesis, Hellenic Open University, 65 p. (in Greek)

Henderson, F.M., 1966. Open Channel Flow. Macmillan Publishing Co., Inc., New York, 522 p.

Langousis, A.S., Fourniotis, N.Th., 2020. Elements of Design of Water Supply and Sewerage Works. 722 p., Gotsis Publications, Greece (in Greek)

Lee, K.T, Ho, Y.H., Chyan, Y.J., 2006. Bridge blockage and overbank simulations using HEC-RAS in the Keelung river during the 2001 Nari typhoon. J. Hydraul. Eng., 132(3): 319–323, doi: 10.1061/(ASCE)0733-9429(2006)132:3(319)

Mays, L.W., 2001. Water Resources Engineering. 1st edition, John Wiley & Sons Inc., New York, 768 p.

Noutsopoulos, G., Christodoulou, G., Papathanasiadis, T., 2007. Open Channel Hydraulics. Fountas Publications, Greece, 325 p. (in Greek)

Ponce, V.M., 2014. Fundamentals of Open Channel Hydraulics. Available online: http://ponce.sdsu.edu/openchannel/index.html

Rao, P., Hromadka, T.V., 2016. Numerical modeling of rapidly varying flows using HEC-RAS and WSPG models. SpringerPlus, 5, 662, doi: 10.1186/s40064-016-2199-0

USACE, 2016. HEC-RAS River Analysis System User's Manual, Version 5.0. US Army Corps of Engineers Hydrologic Engineering Center, 960 p.