VELAS

VELAS

Visualization and analysis of ELAstic aniSotropy

User’s Guide

August 16, 2023

Description

The VELAS User's Guide describes how to run and use various features of the visualization and analysis of elastic anisotropy program VELAS. This guide demonstrates the capabilities of the program, how to use them, and the necessary input files and formats.

VELAS License

Contact Information

Zheng Ran
Email: ranzheng@outlook.com
Please don't hesitate to contact us if you have any questions about using VELAS or suggestions for improving VELAS.

If you use VELAS in your research, please cite:

Z. Ran, C.M. Zou, Z.J. Wei, et al., VELAS: An open-source toolbox for visualization and analysis of elastic anisotropy. Computer Physics Communications, 283 (2023) 108540.
DOI:https://doi.org/10.1016/j.cpc.2022.108540.

Update information

You can download the latest version of VELAS through the following link.
GitHub: https://github.com/ranzhengcode/VELAS

Contents

1. Introduction

VELAS is a user-friendly open-source toolbox for the visualization and analysis of elastic anisotropy written in **GNU Octave** that can be used for any crystal symmetry.

1.1 Meaning of VELAS

VELAS is derived from the combination of the letters **V**, **ELA** and **S** in "**V**isualization and analysis of **ELA**stic ani**S**otropy" and has no connection or relationship to any known trademarks, places or people that might be called "VELAS".

VELAS

Visualization and analysis of ELAstic aniSotropy

1.2 Logo of VELAS

VELAS

1.3 Why VELAS?

As far as we know, there are already several excellent tools for visualising and analysing elastic anisotropy, such as ElAM, ELATE and ELATools, so why do we need to redevelop a similar tool? First of all, VELAS is not just a tool for visualising and analysing elastic anisotropy, we want to integrate more practical features in future developments, such as the calculation of second, third and higher order elastic constants, the visualisation and analysis of stresses and strains, the analysis of various physical properties based on derivatives of third and higher order elastic constants, a deep learning framework for material property prediction and evaluation. Secondly, VELAS offers a more user-friendly graphical user interface than the known ElAM, ELATE and ELATools, allowing the user to perform calculations and plots directly only through the GUI without having to concern themselves with complex keywords. Furthermore, the programming language used is GNU Octave, whose syntax is highly compatible with the commercial software MATLAB, allowing users with access to MATLAB to use VELAS directly without any modifications.

1.4 Highlights

- **Easy** to install and use, **no** compilation required and **no** dependence on any third-party libraries.
- A fully interactive graphical interactive interface (**GUI**).
- Support for a wide range of visualisation schemes such as **map projection** and **unit sphere projection**.
- Supports the determination of the mechanical stability of crystals at **atmospheric** and **high pressures** using the **Born mechanical stability criterion**.
- Support for analysis of properties such as **hardness**, **fracture toughness**, **average Cauchy pressure**, **areal Poisson's ratio**, etc.
- Provides a **native interface** for calling data from the new (default) and legacy APIs of the Materials Project's database.

2. VELAS Program

VELAS is an easy to use, flexible and user-friendly open source toolbox with an interactive graphical interface (GUI), written in GNU octave for visualising and analysing elastic anisotropy. VELAS supports running via both pure script and GUI. The GUI for VELAS is shown in **Fig. 1**.

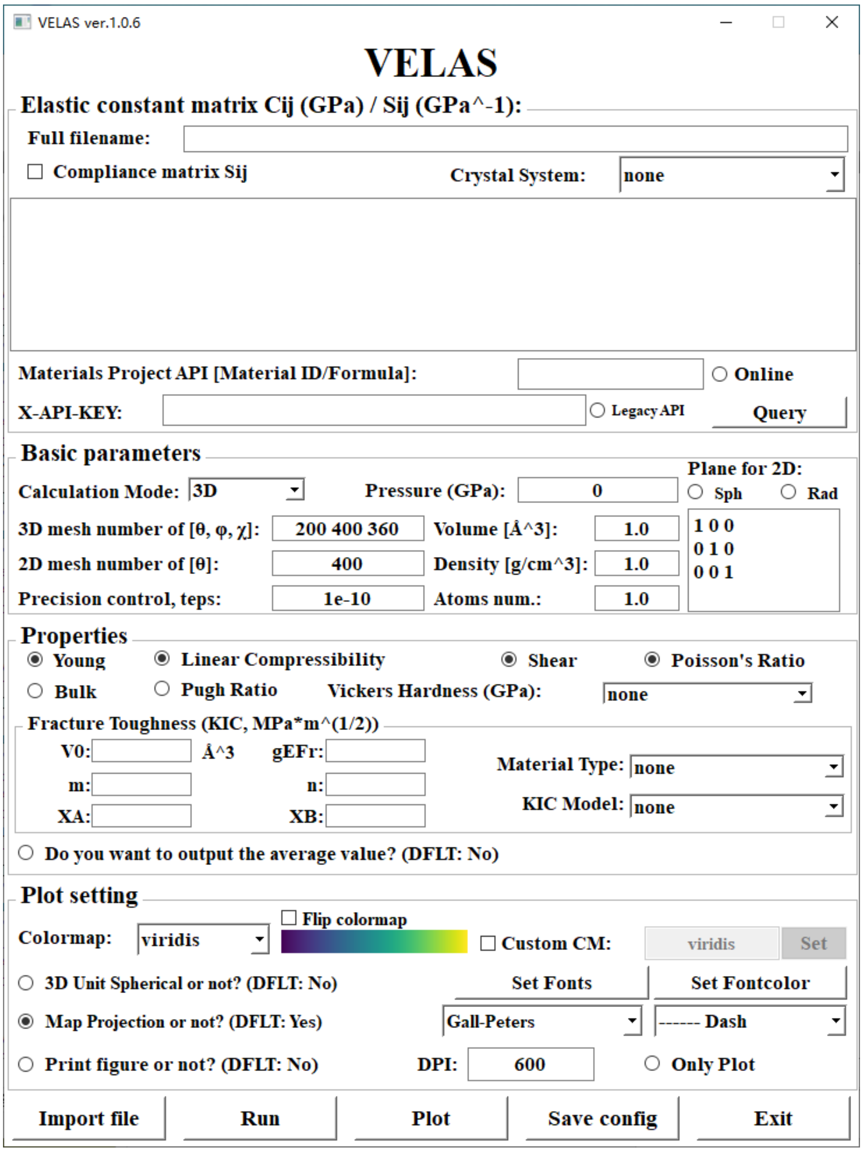


Fig. 1 VELAS GUI.

2.1 Program features

The VELAS code is divided into six sections: the Basic module, the Properties module, the GUI module, the Drawing module, the MPapi module, and the Doc module. The detailed code structure of the six sections is shown in **Fig. 2**.

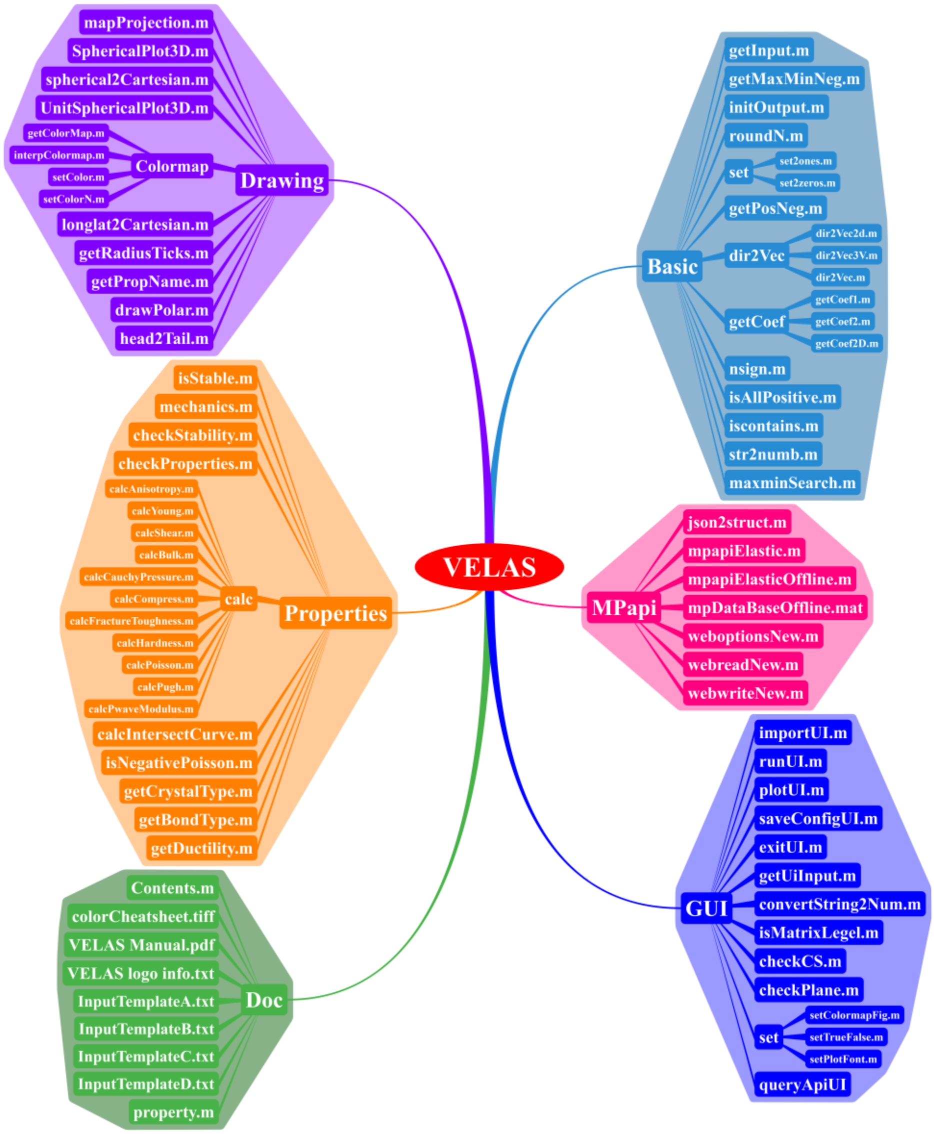


Fig. 2. The code structure of VELAS.

2.2 How to download VELAS

Any user can download any version of VELAS using the following link.
Download links: https://github.com/ranzhengcode/VELAS

If GitHub is not accessible in your country, or you do not have easy access to GitHub, you can request the source code directly from the author via email, which must include the purpose of use.

Author Email: ranzheng@outlook.com

2.3 How to install VELAS

There are three Schemes to install VELAS:

Installation Scheme A

(1) install VELAS using pkg command in GNU Octave command windows as following:

pkg install "https://github.com/ranzhengcode/VELAS/archive/refs/tags/v1.0.6.tar.gz"

(2) And then using pkg load velas to load VELAS.

(3) Type velasGUI to run VELAS.

(4) Enjoy!

Installation Scheme B

(1) Unzip the downloaded VELAS archive into any available path;

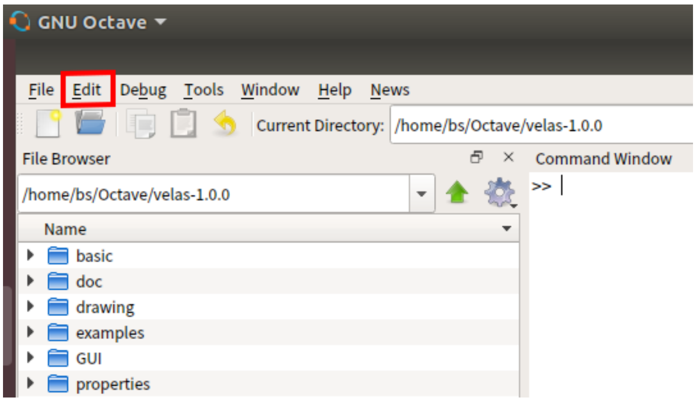
(2) Run install_VELAS or velasGUI, VELAS will be automatically installed.

Installation Scheme C

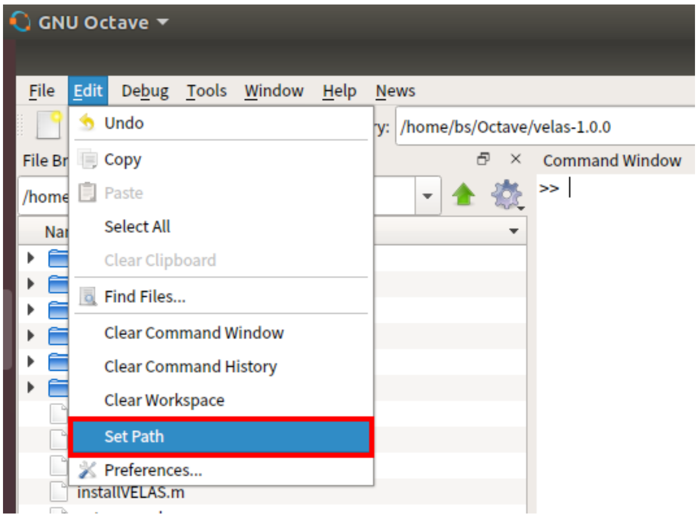
Installing VELAS is a very easy task, provided that you have GNU Octave 5.2.0 and higher or MATLAB 2010 and higher installed on your computer. The installation of VELAS on different operating systems (e.g. Windows, macOS, Linux ans BSD) is not very different, so GNU Octave 6.1.0 on Ubuntu 18.04 is used as a demonstration case.

(1) Unzip the downloaded VELAS archive into any available path.

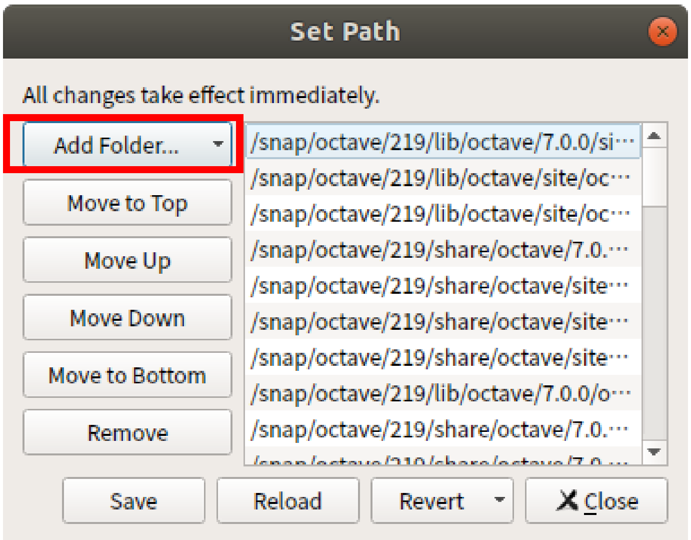
(2) Start GNU Octave 6.1.0 and click on the **Edit** option in the menu bar as shown in the image below.



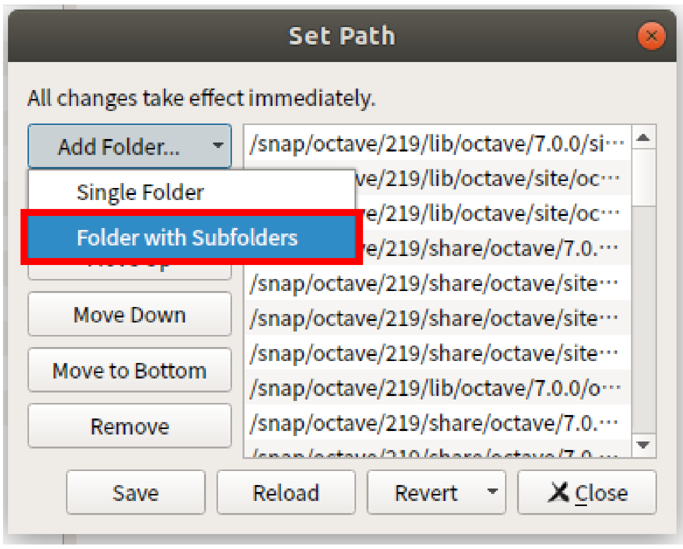
(2) Click on **Set Path** in the **Edit** drop-down box.



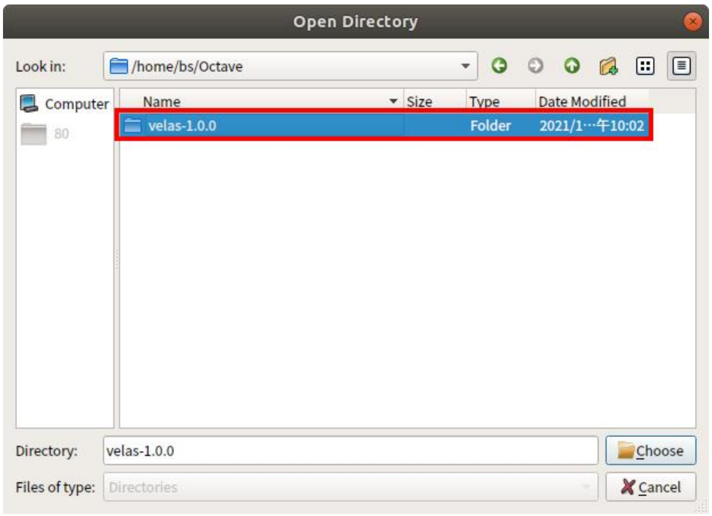
(3) Click **Add Folder** in the **Set Path** dialog box.



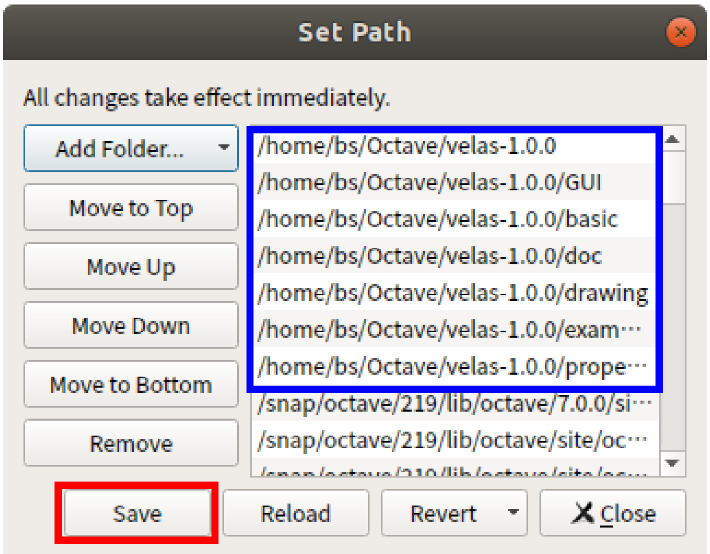
(4) Click on **Folder with Subfolders** in the **Add Folder** drop-down list box.



(5) In the pop-up dialog box, find the path of the **unpacked VELAS folder** in **(1)**, and select the **VELAS folder**, then click **Choose**.



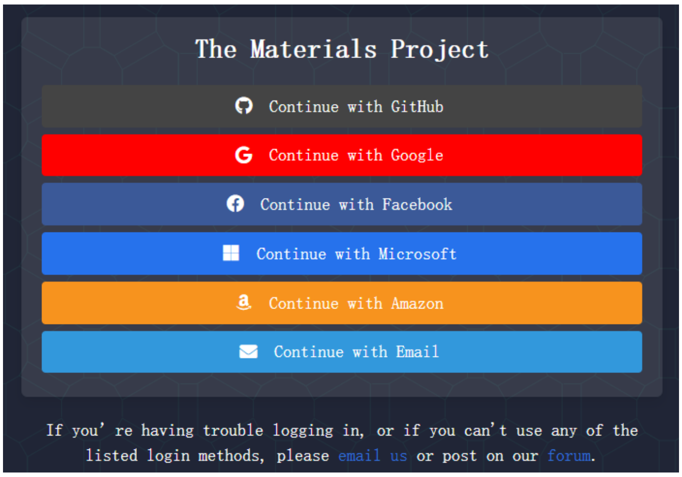
(5) Click **Save** in the **Set Path** dialog box to complete the installation.



After completing the installation, the VELAS folder and its sub-files will automatically be loaded into the GNU Octave working path the next time you start GNU Octave.

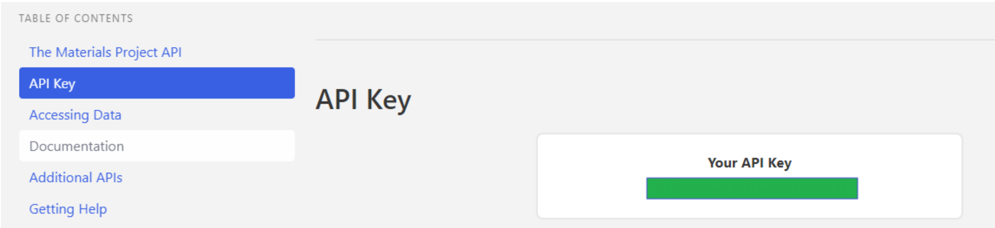
2.4 How to get an API Key of the Materials Project API

(1) New API Key:
Link: https://next-gen.materialsproject.org/api
Type the above link into your browser and then click on **Login** in the browser. Login is supported using the account shown below.



(Source screenshot from profile.materialsproject.org)

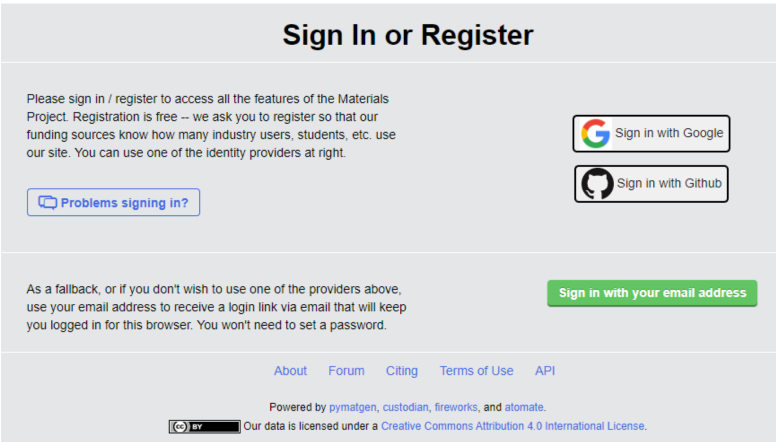
Log in and click on the **API Key** to go to your own API.



(Source screenshot from next-gen.materialsproject.org/api)

(2) Legacy API Key:

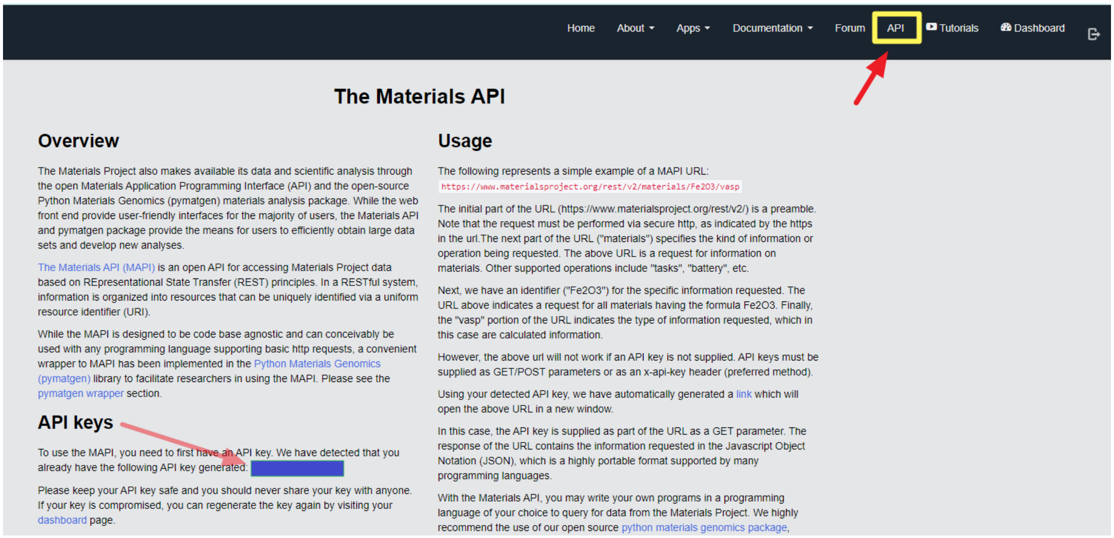
Link: https://legacy.materialsproject.org/open



(Source screenshot from legacy.materialsproject.org/janrain/loginpage)

Type the above link into your browser and then click on **Login** in the browser. The Legacy API supports logging in using Google account or GitHub account. Or use your email address to receive a login link via email that will keep you logged in on this browser.

Once logged in, click on **API** in the menu bar to find the API Key.



2.5 How to run VELAS

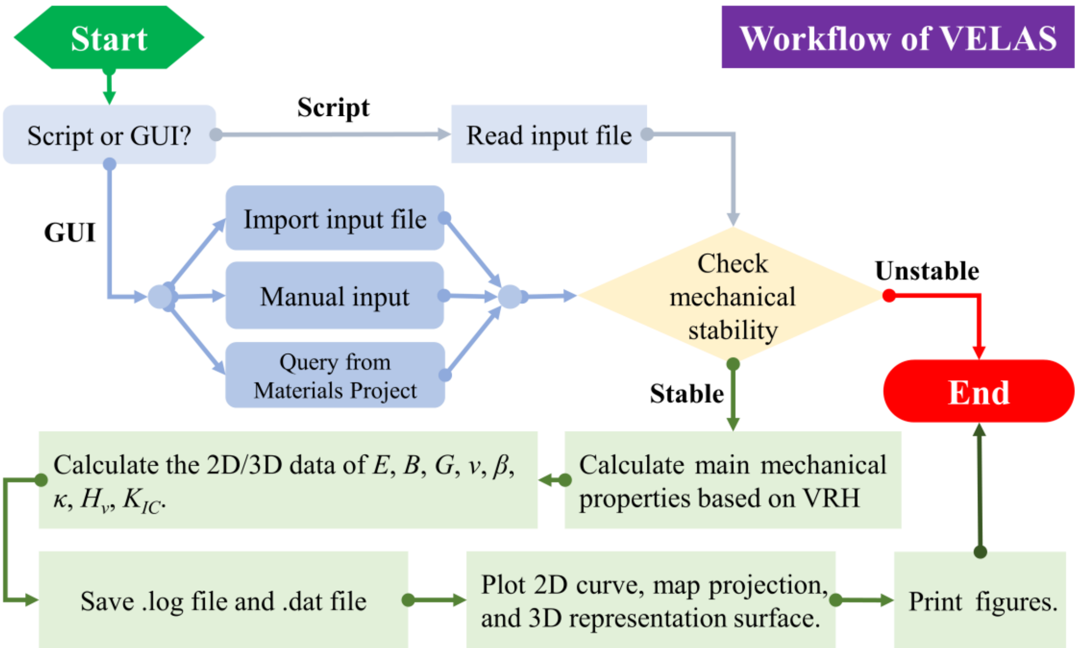


Fig. 3. Workflow of VELAS.

Option 1: Run VELAS using **velasScript**.

(1) The user has to prepare the Doc file according to Template A, Template B, Template C or Template D. A detailed explanation of the keywords in the templates is given later in **section 2.8.**

(2) Type **velasScript** in the command window of GNU Octave and press **Enter** to run.

(3) At the end of the run, the .log file, .mat file and .dat file will be automatically saved in the folder where the input file is located.

Option 2: Run VELAS using **velasGUI**.
(1) Type **velasGUI** in the command window of GNU Octave and press enter to bring up the **VELAS GUI**.
(2) There are three different ways to import data in VELAS GUI:
(a) importing via input files. The user needs to prepare the input file as required by **Option 1 (1)**;
(b) entering and setting using GUI interface. Firstly, users need to fill in the **Full Filename** editbox with a custom filename containing the path. Secondly, in the **Elastic constants** editbox, paste either Cij or Sij, if Sij is pasted, users need to check the **Compliance matrix Sij**. Thirdly, in the **Crystal System** drop-down listbox, select the name of the corresponding crystal system. Then, set the desired options in the interactive screen later.
(c) importing by calling Materials Project's elastic database. The online mode supports calls to both new (default) and legacy API calls, but the user needs to provide a different API Key. currently the online mode only supports queries using the material ID. When using the offline mode, users only need to provide the material ID (e.g. mp-2593) or pretty formula (e.g. SiC) to query data. Offline mode is used by default. Only Cij and crystal system types are imported from the database, other options need to be set by the user as required.
(3) After completing the setup in step (2), click the **Run** button in the GUI to start the calculation.
(4) Once the calculation is complete, click on the **Plot** button in the GUI to plot. If **Print figure or not** is checked, the images will be saved automatically according to the settings in the **Plot setting**.
(5) Click on the **Save config** button to save all the settings in the current GUI. **Save config** is an optional option.
(6) Click on the **Exit** button to exit VELAS.

2.6 Plot and save images

Once the calculation has been completed, users can plot and save the image using the **velasPlot** script or the **Plot** button in the VELAS GUI. Note that if users only uses the VELAS GUI to plot and save images, the **Plot Directly** radio button must be checked.

Support 75 kinds of colormap:

'viridis' (default), 'inferno', 'plasma', 'magma', 'rocket', 'mako', 'flare', 'crest', 'vlag', 'icefire','seismic' 'cool', 'summer', 'copper', 'hot', 'ocean', 'gray', 'bone', 'Spectral', 'coolwarm', 'pink', 'spring', 'autumn', 'winter', 'thermal', 'haline', 'solar', 'ice', 'deep', 'dense', 'algae', 'matter', 'turbid', 'speed', 'amp', 'tempo', 'rain', 'phase', 'balance', 'delta', 'curl', 'diff', 'tarn', 'cubehelix', 'turbo','Blues', 'BuGn', 'BuPu', 'GnBu', 'Greens', 'Greys', 'Oranges', 'OrRd', 'PuBu', 'PuBuGn', 'PuRd', 'Purples', 'RdPu', 'Reds', 'YlGn', 'YlGnBu', 'YlOrBr', 'YlOrRd', 'afmhot', 'gistheat', 'BrBG', 'bwr', 'coolwarmC', 'PiYG', 'PRGn', 'PuOr', 'RdBu', 'RdGy', 'RdYlBu', 'RdYlGn'

Note: If the colormap above is not in the drop-down list box, check the custom colormap checkbox and enter the name of one of the above colormaps.

Setting your own colormap:

You can define your own colormap and store it in the valesColormap.mat file, and then import your colormap via custom colormap.

Note: the name of the colormap you define cannot be the same as the existing colormap in valesColormap.mat, otherwise it will overwrite the existing colormap.

2.7 Inputs and outputs of VELAS

Users can prepare input files by modelling them on the four template input files in the **Doc** module and personalise them with the keywords listed in **section 2.8**.

The VELAS output files include .log file, .mat files and .dat files, as listed in Table 2.1.

Table 2.1 List of the output file for VELAS.

Output file	Property
demoname.log	A log file containing information on input parameters, VRH model results, and the maximum and minimum information of properties.
demoname_3D_E.data	3D representation data of Young's modulus.
demoname_3D_Eneg.dat	The negative values of 3D representation data of Young's modulus.
demoname_3D_B.data	3D representation data of bulk modulus.
demoname_3D_Bneg.dat	The negative values of 3D representation data of bulk modulus.
demoname_3D_beta.data	3D representation data of linear compressibility.
demoname_3D_betaneg.dat	The negative values of 3D representation data of linear compressibility.
demoname_3D_Gmax.dat	The maximum values of 3D representation data of shear modulus.
demoname_3D_Gminp.dat	The minimum positive values of 3D representation data of shear modulus

demoname_3D_Gminn.dat	The minimum negative values of 3D representation data of shear modulus
demoname_3D_Gavg.dat	The average values of 3D representation data of shear modulus
demoname_3D_Pmax.dat	The maximum values of 3D representation data of Poisson's ratio
demoname_3D_Pminp.dat	The minimum positive values of 3D representation data of Poisson's ratio
demoname_3D_Pminn.dat	The minimum negative values of 3D representation data of Poisson's ratio
demoname_3D_Pavg.dat	The average values of 3D representation data of Poisson's ratio
demoname_3D_Prmax.dat	The maximum values of 3D representation data of Pugh ratio
demoname_3D_Prmin.dat	The minimum values of 3D representation data of Pugh ratio
demoname_3D_Pravg.dat	The average values of 3D representation data of Pugh ratio
demoname_3D_Hvmax.dat	The maximum values of 3D representation data of hardness
demoname_3D_Hvmin.dat	The minimum values of 3D representation data of hardness
demoname_3D_Hvavg.dat	The average values of 3D representation data of hardness
demoname_3D_Kicmax.dat	The maximum values of 3D representation data of fracture toughness
demoname_3D_Kicmin.dat	The minimum values of 3D representation data of fracture toughness
demoname_3D_Kicavg.dat	The average values of 3D representation data of fracture toughness
demoname_2D_E(hkl).dat	2D plane data of Young's modulus
demoname_2D_B(hkl).dat	2D plane data of bulk modulus
demoname_2D_beta(hkl).dat	2D plane data of linear compressibility
demoname_2D_G(hkl).dat	2D plane data of shear modulus
demoname_2D_P(hkl).dat	2D plane data of Poisson's ratio
demoname_2D_Pr(hkl).dat	2D plane data of Pugh ratio
demoname_2D_Hv(hkl).dat	2D plane data of hardness
demoname_2D_Kic(hkl).dat	2D plane data of fracture toughness

Notes:

a. demoname – the filename of the input file.

b. 2D/3D – 2D plane/3D space.

c. hkl – Miller index.

2.8 List of all available keywords for VELAS

Table 2.2 List of all available keywords for VELAS and its explanations.

Keyword	Comments	Default value
cfull	Full 6×6 stiffness matrix	N/A
sfull	Full 6×6 compliance matrix	N/A
isotropic[a]	Isotropic crystal system (C_{11}, C_{12})	N/A

cubic[a]	Cubic crystal system (C_{11}, C_{44}, C_{12})	N/A
hexagonal[a]	Hexagonal crystal system (C_{11}, C_{33}, C_{44}, C_{12}, C_{13})	N/A
tetragonal[a]	Tetragonal Type Ⅰ (C_{11}, C_{33}, C_{44}, C_{66}, C_{12}, C_{13})	N/A
	Tetragonal Type Ⅱ (C_{11}, C_{33}, C_{44}, C_{66}, C_{12}, C_{13}, C_{16})	
trigonal[a]	Trigonal Type Ⅰ (C_{11}, C_{33}, C_{44}, C_{12}, C_{13}, C_{14})	N/A
	Trigonal Type Ⅱ (C_{11}, C_{33}, C_{44}, C_{12}, C_{13}, C_{14}, C_{15})	
orthorhombic[a]	Orthorhombic crystal system (C_{11}, C_{22}, C_{33}, C_{44}, C_{55}, C_{66}, C_{12}, C_{13}, C_{23})	N/A
monoclinic[a]	Monoclinic Type Ⅰ (C_{11}, C_{22}, C_{33}, C_{44}, C_{55}, C_{66}, C_{12}, C_{13}, C_{15}, C_{23}, C_{25}, C_{35}, C_{46})	N/A
	Monoclinic Type Ⅱ (C_{11}, C_{22}, C_{33}, C_{44}, C_{55}, C_{66}, C_{12}, C_{13}, C_{16}, C_{23}, C_{26}, C_{36}, C_{45})	
triclinic[a]	Triclinic crystal system (C_{11}, C_{12}, C_{13}, C_{14}, C_{15}, C_{16}, C_{22}, C_{23}, C_{24}, C_{25}, C_{26}, C_{33}, C_{34}, C_{35}, C_{36}, C_{44}, C_{45}, C_{46}, C_{55}, C_{56}, C_{66})	N/A
nmesh2d	2D mesh number of theta(θ)	400
nmesh3d	3D mesh number of theta(θ), phi(φ), chi (χ)	200 400 360
teps	Precision control	1.0e-6
planexy	Plane in the Cartesian coordinate system, 3 coordinates of a vector, select the plane in which a 2D cut is performed	1 0 0 0 1 0 0 0 1
planesph	Plane in the spherical coordinate system	N/A
rad/radian	Indicating ***planesph*** command with radian values	default
ang/angle	Indicating ***planesph*** command with angle values	N/A
mponline	Offline mode or Online mode to call Materials Project	no
mpid	The Material ID for Materials Project	none
xapikey	The X-API-Key for Materials Project to call API	none
mpapiver	Indicating which version of API to use	new
pressure	Pressure (GPa)	0.0
molarmass	Molar mass of crystal cell (g/mol)	1.0
volume	Volume of crystal cell (Å^3, Ångstrom)	1.0
density	Density of crystal cell (g/cm^3)	1.0
atomnum	total number of atoms in crystal cell	1
properties	Properties in 3D/2D space to be calculated, the first row is 3D, and the second row is 2D. 1: to be calculated, 0: not to be calculated.	1 1 1 1 1 0 0 0 1 1 1 1 1 0

		0 0
hv/hardness	Hardness	M
KIC	Fracture Toughness	N/A
avg/average	Output the average value or not?	no
cmap/colormap	Colormap,	hot
unitsph/unitspherical	Draw the 3D unit spherical or not?	no
mapproj/mapprojection	Map projection	yes
lstyle/linestyle	Line style, '-' solid line, '--' dashed line, ':' dotted line, '-.' dash-dotted line	--
print	Print or not?	no
dpi	The resolution size of the output image	600
y/yes	Logical value: true	N/A
n/no	Logical value: false	N/A
#	Comment symbols	N/A

a: Only the first four or more letters of the keywords isotropic, cubic, hexagonal, tetragonal, trigonal, orthorhombic, monoclinic, triclinic can be identified without using the full keyword.

Table 2.3 Use cases of keywords.

Use cases	Format
# Elements of stiffness matrix C, non-zero elements, or all elements	**Comment** Case 1
nmesh3d 200.00 400.00 360.00 # 3D mesh number of theta(θ), phi(φ), chi (χ)	**Comment** Case 2
tetragonal 59.4 42.4 67.2 25.7 3.8 -4.4	**Keyword** $data_1$ ... $data_n$
properties	**Keyword**
1 1 1 1 1 1 1 1	$data_{11}$ ... $data_{1n}$
1 1 1 1 1 1 1 1	… … …
	$data_{n1}$ ... $data_{nn}$
mponline no	**Keyword** yes/no
mpapiver new	**Keyword** $word_1$
map yes M	**Keyword** $word_1$ … $word_n$
KIC M IC 0.00 0.00 0.00 0.00 0.00 0.00	**Keyword** $word_1$ ... $data_1$ ...
lstyle --	**Keyword** symbols

2.9 Bug report

Users are encouraged to report issues, bugs and requests for new features, as well as any potentially useful changes or improvements you may make. Feel free to

communicate with the author via GitHub or email.
GitHub Link: https://github.com/ranzhengcode/VELAS
Author Email: ranzheng@outlook.com

MANY THANKS for your approval and use of VELAS!