



## User's Guide

VELAS version 1.0

July 11, 2022

### 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**

VELAS is distributed, used and modified under the GNU General Public License v3.0. Users of VELAS should abide by the relevant license terms and conditions of the license.

## **Contact Information**

**Zheng Ran**

**Email:** [ranzheng@outlook.com](mailto: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:

Zheng Ran et. al. VELAS: An open-source toolbox for visualization and analysis of elastic anisotropy. Submitted to computer physics communication (2022)

## **Update information**

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

## 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 "Visualization 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".

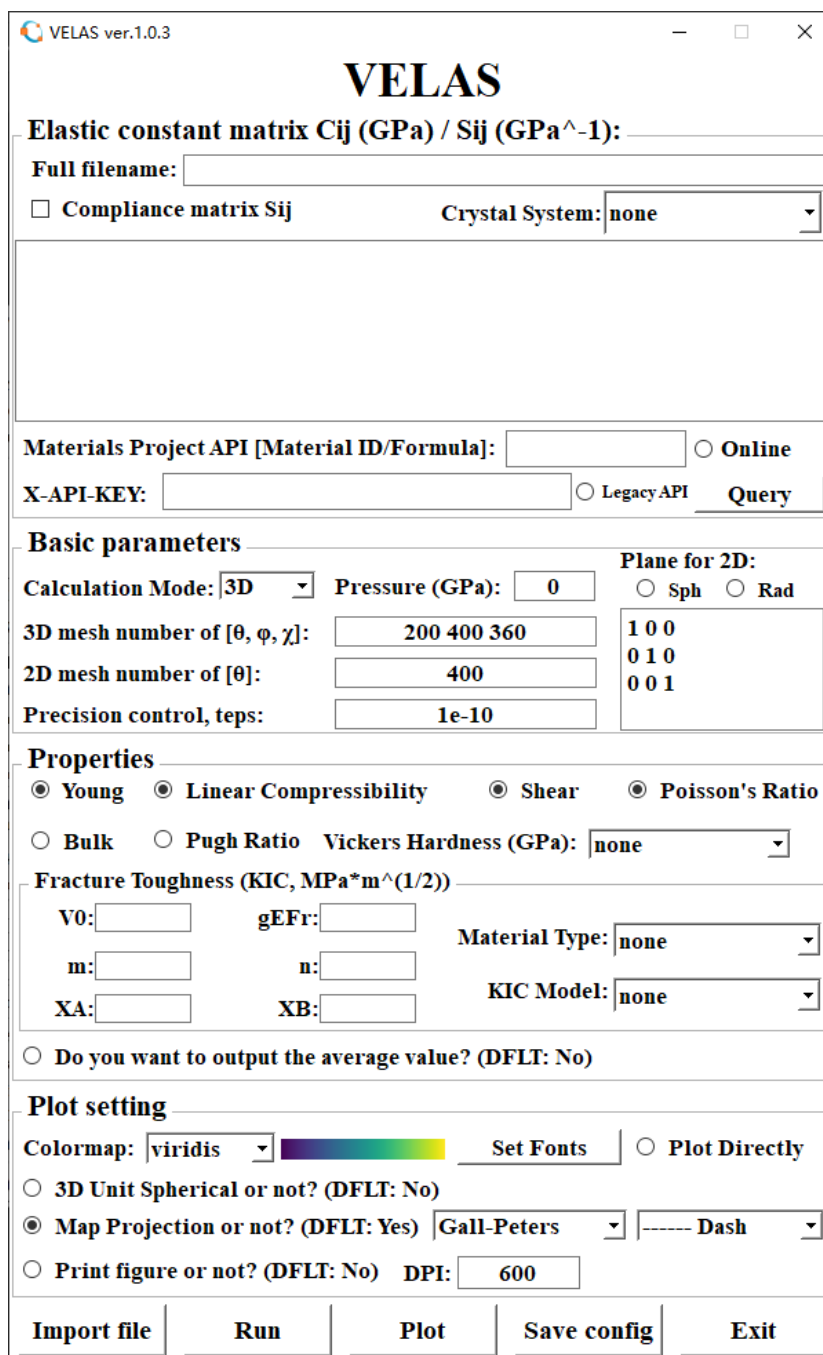


### 1.2 Why VELAS?

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.

## 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**.



The VELAS GUI (version 1.0.3) is a windowed application with a title bar. The main title "VELAS" is centered at the top. Below it, the section "Elastic constant matrix Cij (GPa) / Sij (GPa^-1):" contains a "Full filename:" text box, a checkbox for "Compliance matrix Sij", and a "Crystal System:" dropdown menu currently set to "none". A large empty rectangular box is positioned below these options. The "Materials Project API [Material ID/Formula]:" section includes a text box, radio buttons for "Online" and "Legacy API", and a "Query" button. The "Basic parameters" section features a "Calculation Mode:" dropdown set to "3D", a "Pressure (GPa):" text box set to "0", and a "Plane for 2D:" section with radio buttons for "Sph" and "Rad". Below these are text boxes for "3D mesh number of [θ, φ, χ]:" (set to "200 400 360"), "2D mesh number of [θ]:" (set to "400"), and "Precision control, teps:" (set to "1e-10"). A table for the "Plane for 2D:" shows values "1 0 0", "0 1 0", and "0 0 1". The "Properties" section has radio buttons for "Young", "Linear Compressibility", "Shear", and "Poisson's Ratio", along with "Bulk" and "Pugh Ratio" checkboxes. It includes a "Vickers Hardness (GPa):" dropdown set to "none" and a "Fracture Toughness (KIC, MPa\*m^(1/2))" section with text boxes for "V0:", "gEfr:", "m:", "n:", "XA:", and "XB:", and a "Material Type:" dropdown set to "none". A "KIC Model:" dropdown is also set to "none". A checkbox "Do you want to output the average value? (DFLT: No)" is present. The "Plot setting" section includes a "Colormap:" dropdown set to "viridis" with a color bar, a "Set Fonts" button, and a "Plot Directly" radio button. It also has checkboxes for "3D Unit Spherical or not? (DFLT: No)", "Map Projection or not? (DFLT: Yes)" with a "Gall-Peters" dropdown and a "Dash" dropdown, and "Print figure or not? (DFLT: No)" with a "DPI:" text box set to "600". At the bottom, there are five buttons: "Import file", "Run", "Plot", "Save config", and "Exit".

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**.

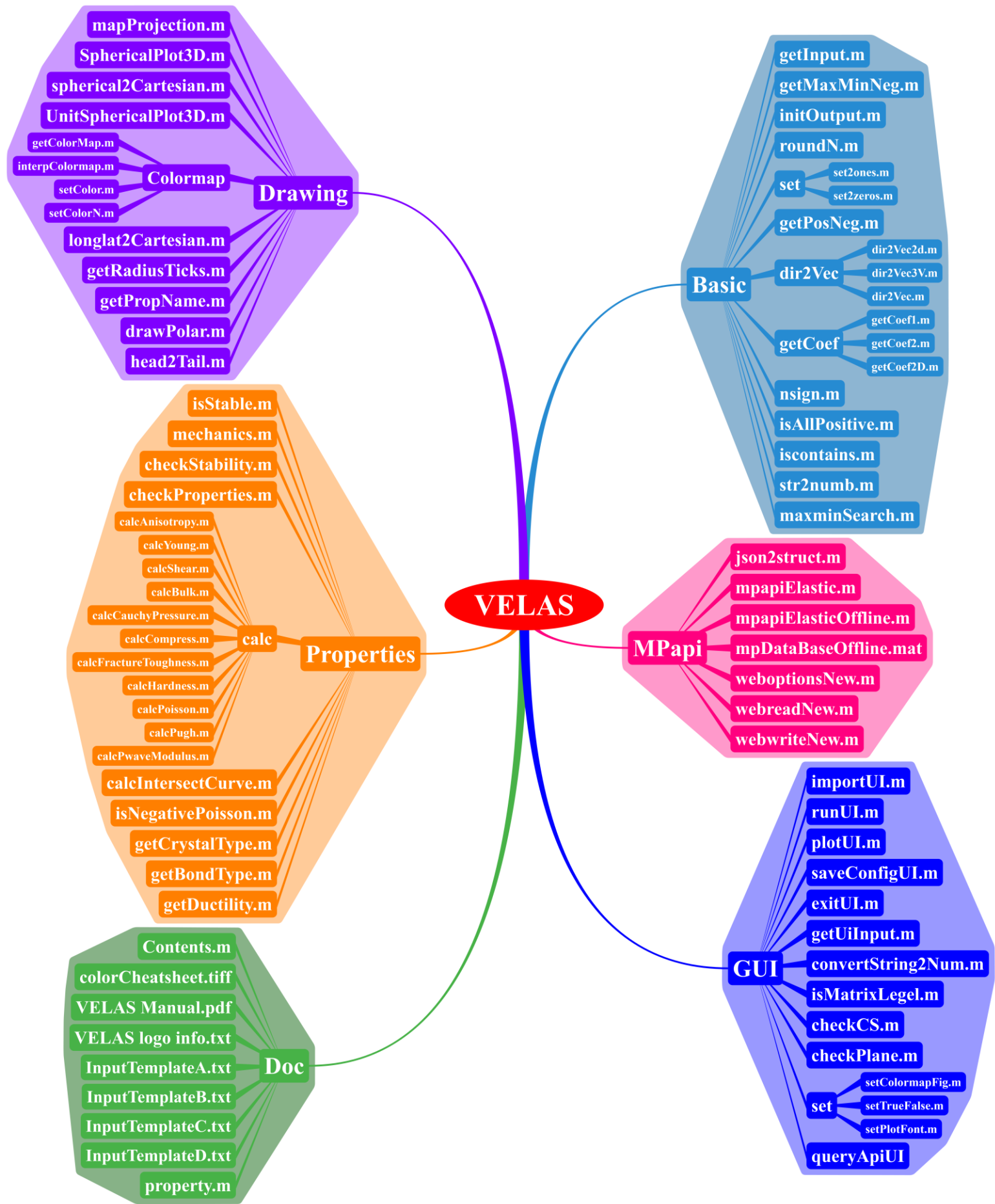


Fig. 2. The code structure of VELAS.

## 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.

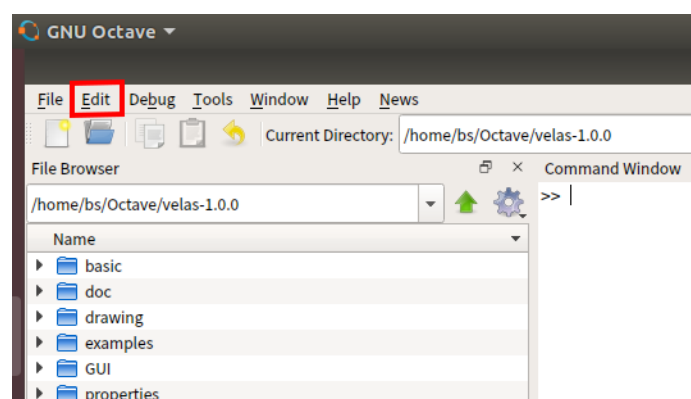
**Author Email:** ranzheng@outlook.com

## 2.3 How to install VELAS

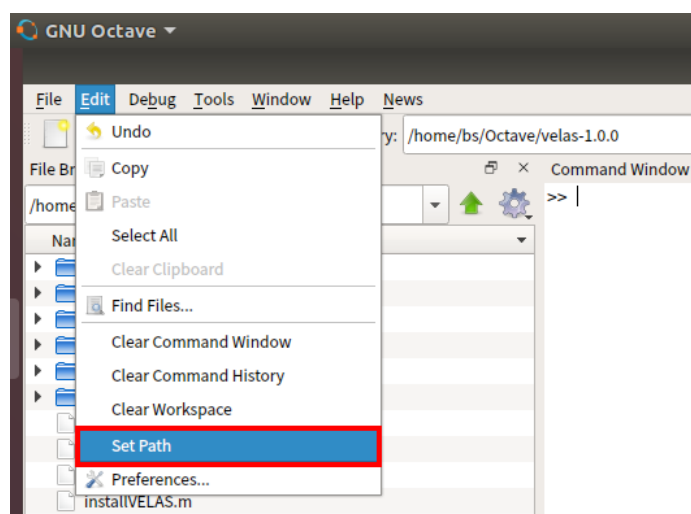
Installing VELAS is a very easy task, provided that you have GNU Octave 5.2.0 and higher or MATLAB 2010 and higher installed on your computer. The installation of VELAS on different operating systems (e.g. Windows, macOS, Linux and BSD) is not very different, so GNU Octave 6.1.0 on Ubuntu 18.04 is used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

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

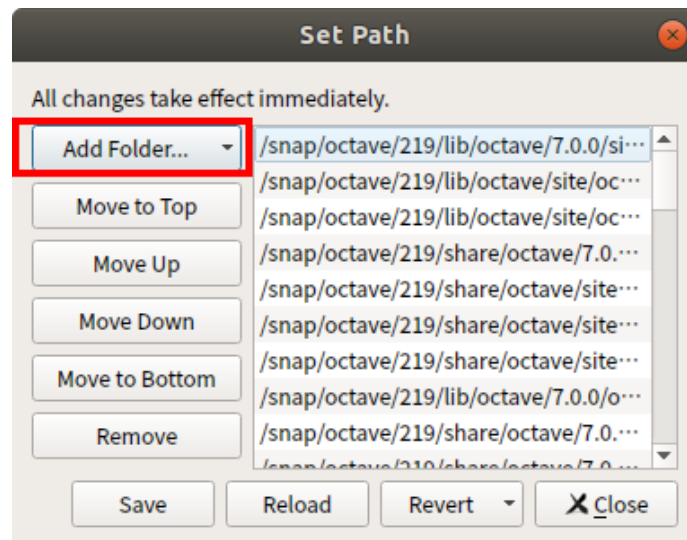
(2) Start GNU Octave 6.1.0 and click on the **Edit** option in the menu bar as shown in the image below.



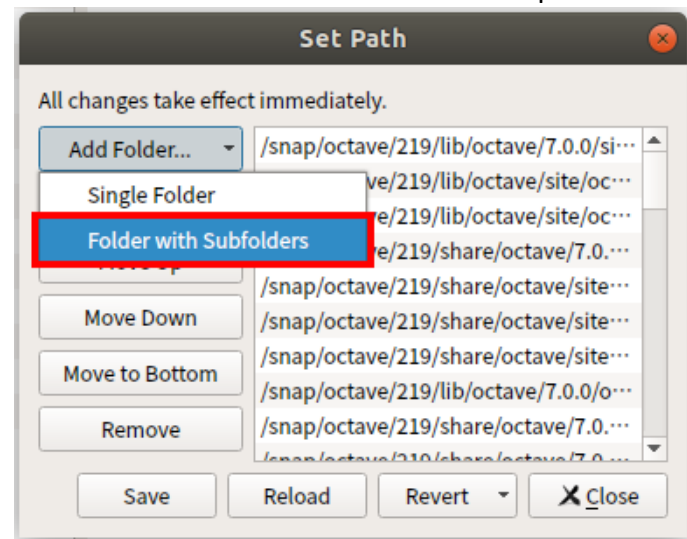
(2) Click on **Set Path** in the **Edit** drop-down box.



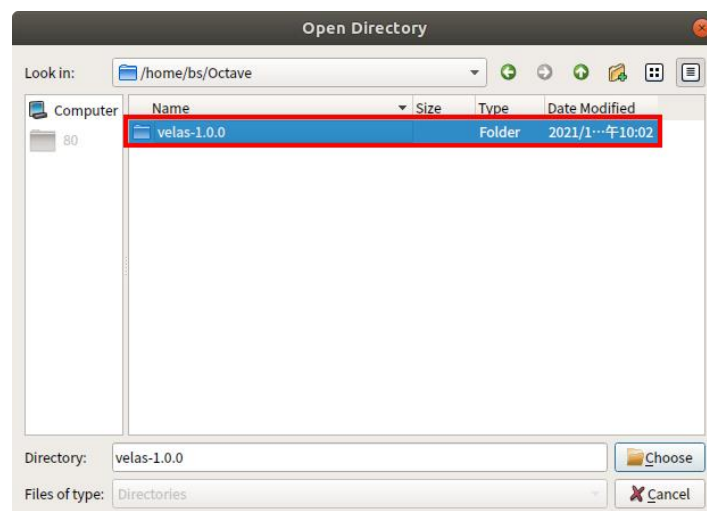
(3) Click **Add Folder** in the **Set Path** dialog box.



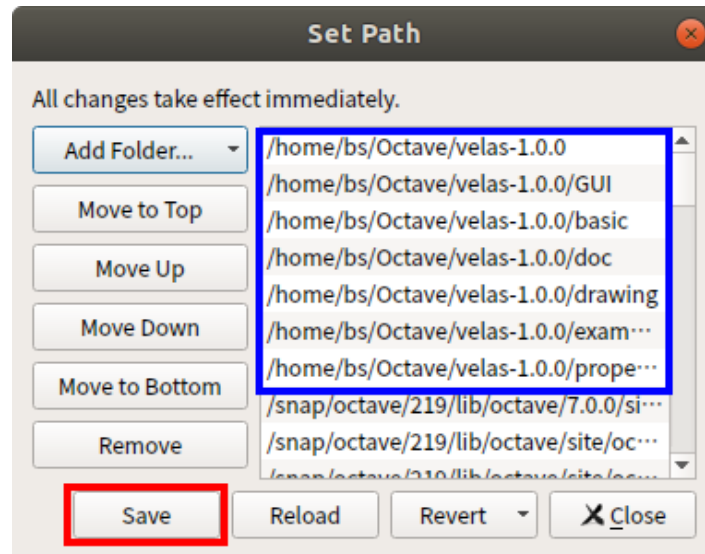
(4) Click on **Folder with Subfolders** in the **Add Folder** drop-down list box.



(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.



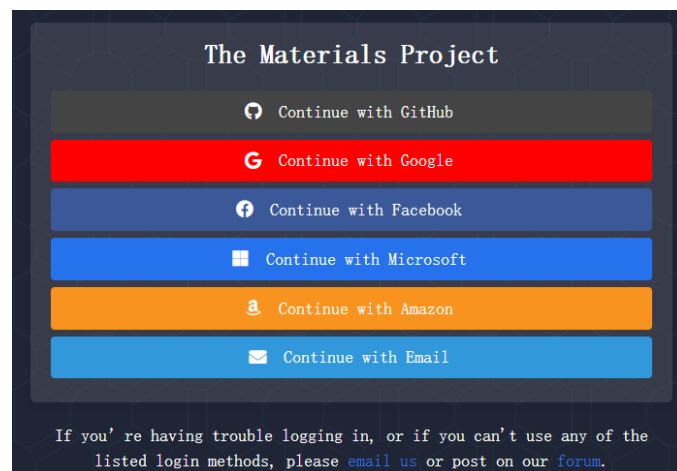
After completing the installation, the VELAS folder and its sub-files will automatically be loaded into the GNU Octave working path the next time you start GNU Octave.

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

(1) New API Key:

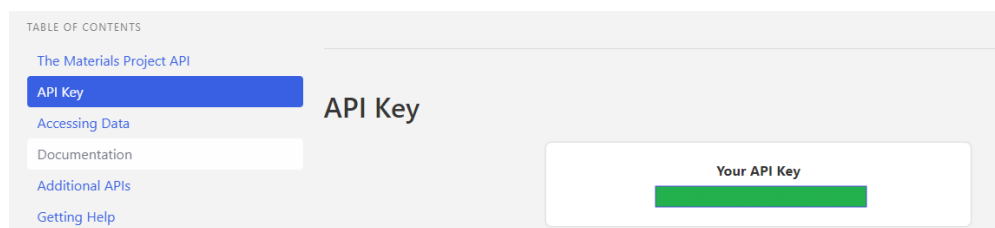
Link: <https://next-gen.materialsproject.org/api>

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>

# Sign In or Register

Please sign in / register to access all the features of the Materials Project. Registration is free – we ask you to register so that our funding sources know how many industry users, students, etc. use our site. You can use one of the identity providers at right.

[Problems signing in?](#)

 Sign in with Google

 Sign in with GitHub

As a fallback, or if you don't wish to use one of the providers above, use your email address to receive a login link via email that will keep you logged in for this browser. You won't need to set a password.

Sign in with your email address

[About](#) [Forum](#) [Citing](#) [Terms of Use](#) [API](#)

Powered by [pymatgen](#), [custodian](#), [fireworks](#), and [atomate](#).

 Our data is licensed under a [Creative Commons Attribution 4.0 International License](#).

(Source screenshot from [legacy.materialsproject.org/janrain/loginpage](https://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.

Home

About >

Apps >

Documentation >

Forum

**API**

Tutorials

Dashboard

The Materials API

Overview

Usage

The Materials Project also makes available its data and scientific analysis through the open Materials Application Programming Interface (API) and the open-source Python Materials Genomics (`pymatgen`) materials analysis package. While the web front end provide user-friendly interfaces for the majority of users, the Materials API and `pymatgen` package provide the means for users to efficiently obtain large data sets and develop new analyses.

The Materials API (MAPI) is an open API for accessing Materials Project data based on Representational State Transfer (REST) principles. In a RESTful system, information is organized into resources that can be uniquely identified via a uniform resource identifier (URI).

While the MAPI is designed to be code base agnostic; and can conceivably be used with any programming language supporting basic http requests, a convenient wrapper to MAPI has been implemented in the [Python Materials Genomics \(`pymatgen`\) library](#) to facilitate researchers in using the MAPI. Please see the [pymatgen wrapper](#) section.

### API keys

To use the MAPI, you need to first have an API key. We have detected that you already have the following API key generated: [REDACTED]

Please keep your API key safe and you should never share your key with anyone. If your key is compromised, you can regenerate the key again by visiting your [dashboard page](#).

The following represents a simple example of a MAPI URL:

```
https://www.materialsproject.org/rest/v2/materials/?Fe2O3&vasp
```

The initial part of the URL (<https://www.materialsproject.org/rest/v2/>) is a preamble. Note that the request must be performed via secure http, as indicated by the https in the url. The next part of the URL ("`/materials/`") specifies the kind of information or operation being requested. The above URL is a request for information on materials. Other supported operations include "`tasks`", "`battery`", etc.

Next, we have an identifier ("`?Fe2O3`") for the specific information requested. The URL above indicates a request for all materials having the formula Fe<sub>2</sub>O<sub>3</sub>. Finally, the "`&vasp`" portion of the URL indicates the type of information requested, which in this case are calculated information.

However, the above url will not work if an API key is not supplied. API keys must be supplied as GET/POST parameters or as an x-api-key header (preferred method).

Using your detected API key, we have automatically generated a [link](#) which will open the above URL in a new window.

In this case, the API key is supplied as part of the URL as a GET parameter. The response of the URL contains the information requested in the Javascript Object Notation (JSON), which is a highly portable format supported by many programming languages.

With the Materials API, you may write your own programs in a programming language of your choice to query for data from the Materials Project. We highly recommend the use of our [open source python tools](#) to interact with the API.

## 2.5 How to run VELAS

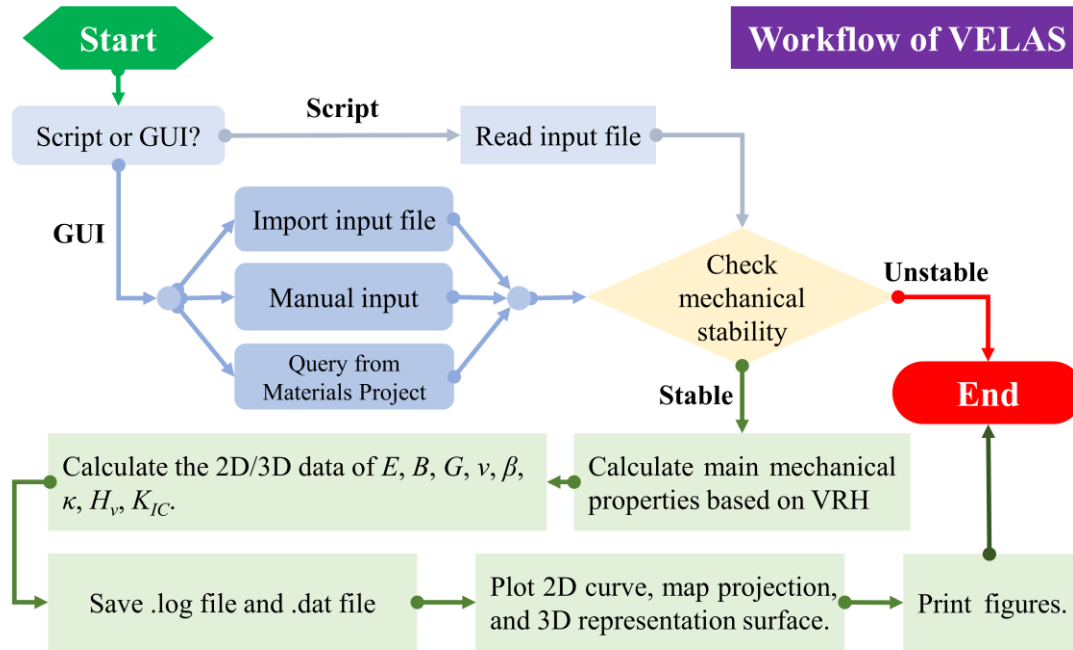


Fig. 3. Workflow of VELAS.

**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**, as shown in the image below.

VELAS ver.1.0.0

# VELAS

**Elastic constant matrix Cij (GPa) / Sij (GPa<sup>-1</sup>):** \_\_\_\_\_

**Full filename:** \_\_\_\_\_

☐ Compliance matrix Sij      **Crystal System:** none ▾

**Materials Project API [Material ID/Formula]:** \_\_\_\_\_ ☐ Online

**X-API-KEY:** \_\_\_\_\_ ☐ Legacy API    Query

**Basic parameters**

**Calculation Mode:** 3D ▾    **Pressure (GPa):** 0    **Plane for 2D:** ☐ Sph ☐ Rad

**3D mesh number of [θ, φ, χ]:** 200 400 360    

1	0	0
0	1	0
0	0	1

**2D mesh number of [θ]:** 400

**Precision control, teps:** 1e-10

**Properties**

☒ Young    ☒ Linear Compressibility    ☒ Shear    ☒ Poisson's Ratio

☐ Bulk    ☐ Pugh Ratio    **Vickers Hardness (GPa):** none ▾

**Fracture Toughness (KIC, MPa\*m<sup>1/2</sup>):**

**V0:** \_\_\_\_\_    **gEfr:** \_\_\_\_\_    **Material Type:** none ▾

**m:** \_\_\_\_\_    **n:** \_\_\_\_\_    **KIC Model:** none ▾

**XA:** \_\_\_\_\_    **XB:** \_\_\_\_\_

☐ Do you want to output the average value? (DFLT: No)

**Plot setting**

**Colormap:** jet ▾ **Set Fonts** ☐ Plot Directly

☐ 3D Unit Spherical or not? (DFLT: No)

☒ Map Projection or not? (DFLT: Yes) Gall-Peters ▾    ----- Dash ▾

☐ Print figure or not? (DFLT: No)    **DPI:** 600

**Import file**    **Run**    **Plot**    **Save config**    **Exit**

(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.

## 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 2.7**.

The VELAS output files include .log file, .mat files and .dat files, .

## 2.8 List of all available keywords for VELAS

Table 1. 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 <sup>a</sup>	Isotropic crystal system ( $C_{11}$ , $C_{12}$ )	N/A
cubic <sup>a</sup>	Cubic crystal system ( $C_{11}$ , $C_{44}$ , $C_{12}$ )	N/A
hexagonal <sup>a</sup>	Hexagonal crystal system ( $C_{11}$ , $C_{33}$ , $C_{44}$ , $C_{12}$ , $C_{13}$ )	N/A
tetragonal <sup>a</sup>	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 <sup>a</sup>	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 <sup>a</sup>	Orthorhombic crystal system ( $C_{11}$ , $C_{22}$ , $C_{33}$ , $C_{44}$ , $C_{55}$ , $C_{66}$ , $C_{12}$ ,	N/A

	C <sub>13</sub> , C <sub>23</sub> )	
monoclinic <sup>a</sup>	Monoclinic Type I (C <sub>11</sub> , C <sub>22</sub> , C <sub>33</sub> , C <sub>44</sub> , C <sub>55</sub> , C <sub>66</sub> , C <sub>12</sub> , C <sub>13</sub> , C <sub>15</sub> , C <sub>23</sub> , C <sub>25</sub> , C <sub>35</sub> , C <sub>46</sub> )	N/A
	Monoclinic Type II (C <sub>11</sub> , C <sub>22</sub> , C <sub>33</sub> , C <sub>44</sub> , C <sub>55</sub> , C <sub>66</sub> , C <sub>12</sub> , C <sub>13</sub> , C <sub>16</sub> , C <sub>23</sub> , C <sub>26</sub> , C <sub>36</sub> , C <sub>45</sub> )	
triclinic <sup>a</sup>	Triclinic crystal system (C <sub>11</sub> , C <sub>12</sub> , C <sub>13</sub> , C <sub>14</sub> , C <sub>15</sub> , C <sub>16</sub> , C <sub>22</sub> , C <sub>23</sub> , C <sub>24</sub> , C <sub>25</sub> , C <sub>26</sub> , C <sub>33</sub> , C <sub>34</sub> , C <sub>35</sub> , C <sub>36</sub> , C <sub>44</sub> , C <sub>45</sub> , C <sub>46</sub> , C <sub>55</sub> , C <sub>56</sub> , C <sub>66</sub> )	N/A
nmesh2d	2D mesh number of theta( $\theta$ )	400
nmesh3d	3D mesh number of theta( $\theta$ ), phi( $\phi$ ), chi ( $\chi$ )	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 <b>planesph</b> command with radian values	default
ang/angle	Indicating <b>planesph</b> 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	0.0
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.

---

## 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!**