The Journal of Open Source Software

continuum_mechanics: A Python package for Continuum Mechanics

Nicolás Guarín-Zapata[1]

1 Departamento de Ingeniería Civil, Universidad EAFIT, Medellín-Colombia

DOI: 10.21105/joss.02047

Software

- Review
- Repository
- Archive

Editor: Kevin M. Moerman

Submitted: 29 January 2020
Published: 29 January 2020

Summary

`continuum_mechanics` is a Python package built on top of SymPy (Meurer et al. (2017)) to aid with calculations in Continuum Mechanics that are commonly lengthy and tedious if done by hand. It also provides visualization capabilities for second-order tensors such as Mohr's circle to help in stress analyses.

The package can be used by:

- researchers that need to double-check analytic calculations;
- researchers implementing numerical methods, such as the Finite element Method, that need to verify the solutions using techniques such as the Method of Manufactured Solutions (Roache (2001), Aycock, Rebelo, & Craven (2020));
- analysts that need to *calibrate* computational models related structural analysis in Civil or Mechanical Engineering;
- teachers who want to automate the creation of problem sets with solutions;
- students who want to verify their solutions to problem sets.

The `continuum_mechanics` package is ready for installation using `pip` or can be tested online using the provided Jupyter Notebooks.

Statement of Need

`continuum_mechanics` was designed to be used by researchers and instructors in the field of Continuum Mechanics. The package helps with tedious calculations such as vector identities or the application of differential operators to scalar, vector or tensor fields.

Some features of `continuum_mechanics` are:

- It is based on an open-source environment.
- It is easy to use.
- It provides the following curvilinear (orthogonal) coordinate systems:
  - Cartesian;
  - Cylindrical;

  - Spherical;
  - Parabolic cylindrical;
  - Parabolic;
  - Paraboloidal;
  - Elliptic cylindrical;
  - Oblate spheroidal;
  - Prolate spheroidal;
  - Ellipsoidal;
  - Bipolar cylindrical;
  - Toroidal;
  - Bispherical; and
  - Conical.

- It supports major vector operators such as:
  - gradient of a scalar function;
  - divergence of a vector function;
  - curl of a vector function;
  - gradient of a vector function;
  - divergence of a tensor;
  - Laplace operator of a scalar function;
  - Laplace operator of a vector function; and
  - Biharmonic operator of a scalar function.

Examples of use

Gradient of a scalar function

By default Cartesian coordinates are given by x, y and z. If these coordinates are used there is not necessary to specify them when calling the vector operators

```
from sympy import *
from continuum_mechanics import vector
x, y, z = symbols("x y z")
```

The gradient takes as input a scalar and returns a vector, represented by a 3 by 1 matrix.

```
f = 2*x + 3*y**2 - sin(z)
f
```

$$2x + 3y^2 - \sin(z)$$

```
vector.grad(f)
```

$$\begin{bmatrix} 2 \\ 6y \\ -cos(z) \end{bmatrix}$$

Visualization of a second-order symmetric tensor

The Mohr's circle is a two-dimensional graphical depiction of the transformation law for the Cauchy stress tensor, where the abcissa represents the normal stress component and the ordinate the shear stress component. This representation is commonly used to represent stress states.

We can visualize

$$\begin{bmatrix} 1 & 2 & 4 \\ 2 & 2 & 1 \\ 4 & 1 & 3 \end{bmatrix}.$$

using the following snippet.

```
from sympy import Matrix
from continuum_mechanics.visualization import mohr3d

mohr3d(Matrix([
    [1, 2, 4],
    [2, 2, 1],
    [4, 1, 3]]))
```

The Mohr circle for this tensor is presented in Figure 1.

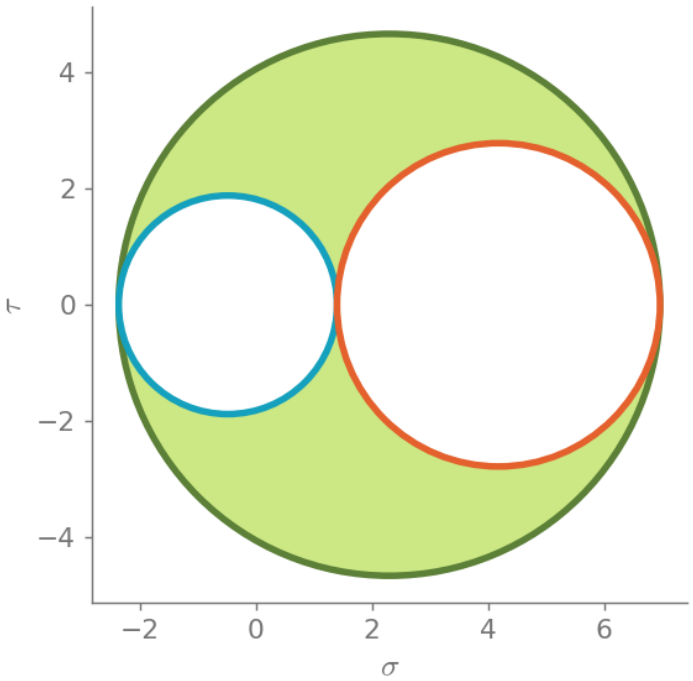

Figure 1: Mohr circle for a 3D symmetric tensor.

Recent Uses

The `continuum_mechanics` package was developed as a research aid in the Applied Mechanics group at Universidad EAFIT, Colombia. Particularly, it has been helpful during the development of Finite Element Methods for generalized continua (Nowacki (1972), Hadjesfandiari & Dargush (2011)). Some of the calculations related to Guarín-Zapata, Gomez, Valencia, Dargush, & Hadjesfandiari (2020) are presented in one of the tutorials available in the documentation.

References

Aycock, K. I., Rebelo, N., & Craven, B. A. (2020). Method of manufactured solutions code verification of elastostatic solid mechanics problems in a commercial finite element solver. *Computers & Structures*, *229*, 106175. doi:10.1016/j.compstruc.2019.106175

Guarín-Zapata, N., Gomez, J., Valencia, C., Dargush, G. F., & Hadjesfandiari, A. R. (2020). Finite element modeling of micropolar-based phononic crystals. *Wave Motion*, *92*, 102406. doi:10.1016/j.wavemoti.2019.102406

Hadjesfandiari, A. R., & Dargush, G. F. (2011). Couple stress theory for solids. *International Journal of Solids and Structures*, *48*(18), 2496–2510. doi:10.1016/j.ijsolstr.2011.05.002

Meurer, A., Smith, C. P., Paprocki, M., Čertík, O., Kirpichev, S. B., Rocklin, M., Kumar, A., et al. (2017). SymPy: symbolic computing in Python. *PeerJ Computer Science*, *3*, 103. doi:10.7717/peerj-cs.103

Nowacki, W. (1972). *Theory of Micropolar Elasticity*. International centre for mechanical sciences. Springer. doi:10.1007/978-3-7091-2720-9

Roache, P. J. (2001). Code verification by the method of manufactured solutions. *Journal of Fluids Engineering*, *124*(1), 4–10. doi:10.1115/1.1436090