
Seismic Documentation

Release 1

Igor Morgado

dez 18, 2017

Contents:

1	README	1
2	CHANGELOG	3
3	Seismic	5
3.1	Base functions	5
3.2	Wave Propagation	9
3.3	Seismic Source	10
3.4	Stability Analysis	10
4	Indices and tables	13
	Índice de Módulos Python	15
	Índice	17

CAPÍTULO 1

README

A readme

CAPÍTULO 2

CHANGELOG

A changlog

3.1 Base functions

3.1.1 Module contents

This code implements all algorithms described on SILVA2015 - Introducao ao metodo sismico: Modelagem Computacional

Serao utilizadas a principio as mesmas variaves e nomeclaturas usadas no artigo

`base.Vp` (*density=None, young=None, poisson=None, shear=None, deformation=None, distance=None, time=None*)

Retorna a velocidade da onda compressional (P) pelas propriedades dadas.

Para computacao, deve se entrar uma das seguintes combinacoes de parametros: * density, young, poisson; * density, shear, deformation; * distance, time

Parâmetros

- **density** (*float*) – Densidade do material (ρ).
- **young** (*float*) – Coeficiente de Young (E).
- **poisson** (*float*) – Coeficiente de poisson (σ)
- **shear** (*float*) – Coeficiente compressional (μ)
- **deformation** (*float*) – Deformacao do material (k) (Bulk's)
- **distance** (*float*) – Distancia em metros (x)
- **time** (*float*) – tempo percorrido (t).

Retorna Velocidade da onda P.

Tipo de retorno float

`base.Vs` (*density, shear=None, young=None, poisson=None*)

Returns shear velocity based on physical coefficients.

You should supply one of the two parameter combinations:

- density and shear;
- density, Young and Poisson coefficient

This function verifies the values, therefore is slower, if you need faster function, call directly:

- `velocity_s_by_shear`
- `velocity_s_by_young`

Parâmetros

- **density** (*float*) – Density (non negative) (ρ).
- **shear** (*float*) – Shear (non negative) (μ).
- **young** (*float*) – Young coefficient (E).
- **poisson** (*float*) – Poisson coefficient. $0 < \sigma < 1/2$.

Retorna Shear velocity

Tipo de retorno float

`base.acoustic_impedance` (*density, velocity*)

Returns acoustic impedance

Parâmetros

- **density** (*float*) – Density (rho).
- **velocity** (*float*) – Wave velocity (V)

Retorna Acoustic impedance

Tipo de retorno float

`base.frequency` (*initial_time, end_time*)

Retorna a frequencia baseado no intervalo de tempo.

Parâmetros

- **t0** (*float*) – Inicio do intervalo.
- **t1** (*float*) – Fim do intervalo.

Retorna Frequencia.

Tipo de retorno float

`base.period` (*initial_time, final_time*)

Retorna o periodo do intervalo.

Parâmetros

- **initial_time** (*float*) – Inicio do intervalo.
- **final_time** (*float*) – Fim do intervalo.

Retorna Tamanho do intervalo.

Tipo de retorno float

`base.reflection_coefficient` (*z1, z2*)

Returns the reflection coefficient.

Parâmetros

- **z1** (*float*) – Acoustic impedance of upper layer.
- **z2** (*float*) – Acoustic impedance of lower layer.

Retorna Reflection coefficient.

Tipo de retorno float

`base.velocity_p_by_distance(x, t)`

Retorna a velocidade da onda compressional dados distancia e tempo

Parâmetros

- **x** (*float*) – Distancia percorrida.
- **t** (*float*) – Tempo percorrido.

Retorna velocidade compressional

Tipo de retorno float

`base.velocity_p_by_shear(rho, mu, k)`

Retorna a velocidade da onda P (compressional) pelas propriedades da rocha.

Parâmetros

- **rho** (*float*) – Densidade do material.
- **mu** (*float*) – Modulo de cisalhamento (mede deformacao sem mudanca de
- **volume**) –
- **k** (*float*) – Modulo da deformacao volumetrica (Bulk).

Retorna velocidade compressional.

Tipo de retorno float

`base.velocity_p_by_young(rho, E, sigma)`

Retorna a velocidade da onda P (compressional) pelas propriedades da rocha.

Parâmetros

- **rho** (*float*) – densidade do material.
- **E** (*float*) – Modulo de Young (proporcionalidade entre a tensao e a
- **deformacao**) –
- **sigma** (*float*) – Coeficiente de Poisson (razao da contracao transversal
- **extensao longitudinal.** (e) –

Retorna Velocidade compressional.

Tipo de retorno float

`base.velocity_ratio(deformation=None, shear=None, poisson=None)`

Returns velocity ratio P/S

The function requires “deformation and shear” OR poisson coefficients. If all three are supplied Poisson is ignored.

This function is slower since it does verifications but safer. If you want to call without value verification use: *
velocity_ratio_by_deformation * velocity_ratio_by_poisson

Parâmetros

- **deformation** (*float*) – Deformation factor (k), must be positive.

- **shear** (*float*) – Shear factor (μ), must be positive.
- **poisson** (*float*) – Poisson coeficient (σ), $0 < \sigma < 1/2$.

Retorna Velocity ratio P/S. Always a number bigger than 1.

Tipo de retorno float

`base.velocity_ratio_by_deformation(k, mu)`

Retorna a razao entre a velocidade P e S.

Parâmetros

- **deformation** (*float*) – Coeficiente de deformacao (de Bulk)
- **shear** (*float*) – Coeficiente de cisalhamento

Retorna Razao da velocidade Vp/Vs.

Tipo de retorno float

`base.velocity_ratio_by_poisson(sigma)`

Retorna a razao entre a Vp e Vs

Parâmetros **poisson** (*float*) – Coeficiente de poisson

Retorna Razao da velocidade Vp/Vs

Tipo de retorno float

`base.velocity_s_by_shear(rho, mu)`

Retorna a velocidade da onda S (cisalhante) dados mu e rho.

Parâmetros

- **rho** (*float*) – Coeficiente de Cisalhamento
- **mu** (*float*) – Coeficiente de densidade.

Retorna Velocidade cisalhante

Tipo de retorno float

`base.velocity_s_by_young(rho, E, sigma)`

Retorna a velocidade da onda S dados sigma e rho.

Parâmetros

- **rho** (*float*) – Coeficiente de densidade (rho)
- **E** (*float*) – Young's soefficient
- **sigma** (*float*) – Coeficiente de Poisson (sigma)

Retorna Velocidade cisalhante

Tipo de retorno float

`base.wave_velocity(initial_position, initial_time, final_position, final_time)`

Retorna a velocidade da onda como a razao entre duas cristas de onda.

Parâmetros

- **initial_position** (*float*) – Inicio do Intervalo.
- **initial_time** (*float*) – Inicio do intervalo.
- **final_position** (*float*) – Final do intervalo.
- **final_time** (*float*) – Fim do intervalo.

Retorna Velocidade da onda

Tipo de retorno float

`base.wavelength(initial_position, final_position)`

Retorna 'lambda' (comprimento de onda), dados dois picos `r1`, `r0`.

Parâmetros

- **initial_position** (*float*) – Início do Intervalo.
- **final_position** (*float*) – Final do intervalo.

Retorna Comprimento de onda (`lambda`)

Tipo de retorno float

3.2 Wave Propagation

3.2.1 Module contents

`wave.discrete_wave_step_3d(wave_field, velocity_field, dx, dy, dz, dt)`

Executes the discrete wave timestep forward in time

`wave.fp_second_order(values, interval, axis=-1)`

Returns the second derivative of second order

values must have (at least) 3 points equally spaced.

Parâmetros

- **values** (*nparray*) – Values of points in grid
- **interval** (*float*) – Interval

Retorna List of second order derivatives

Tipo de retorno float

`wave.fp_second_order_explicit_terms(U, axis=-1)`

Returns the terms of explicit second order first derivative

values must have (at least) 3 points equally spaced.

Parâmetros `U` (*nparray*) – Values of points in grid

Retorna List of second order derivatives

Tipo de retorno float

`wave.fp_second_order_term(U, axis=-1)`

Returns the second derivative of second order without the step

values must have (at least) 3 points equally spaced.

Parâmetros `U` (*nparray*) – Value points in grid

Retorna List of second order derivatives

Tipo de retorno float

`wave.fpp_fourth_order(U, interval, axis=-1)`

Returns the second derivative of fourth order

values must have (at least) 5 points equally spaced.

Parâmetros

- **values** (*nparray*) – Values of points in grid
- **interval** (*float*) – Interval

Retorna List of fourth order derivatives

Tipo de retorno float

`wave.fpp_fourth_order_term(U, axis=-1)`

Returns the second derivative of fourth order term without the step values must have (at least) 5 points equally spaced.

Parâmetros **values** (*nparray*) – Values of points in grid

Retorna List of fourth order derivatives

Tipo de retorno float

3.3 Seismic Source

3.3.1 Module contents

`source.source()`

A seismic source

3.4 Stability Analysis

3.4.1 Module contents

This module will evaluate model stability

`stability.dispersion(V, dx, dy, dz, f, k=5)`

Evaluate the model dispersion.

Parâmetros

- **V** (*nparray*) – is the velocity model
- **dx** (*float*) – is the x interval
- **dy** (*float*) – is the y interval
- **dz** (*float*) – is the z interval
- **f** (*float*) – is the cutdown frequency
- **k** (*float*) – maximum number of samples in a wavelength.

Retorna If the model is stable

Tipo de retorno bool

`stability.stability(V, dx, dy, dz, dt, mu=5)`

Evaluate the model stability.

Parâmetros

- **V** (*nparray*) – is the velocity model

- **dx** (*float*) – is the x interval
- **dy** (*float*) – is the y interval
- **dz** (*float*) – is the z interval
- **dt** (*float*) – is the t interval (timestep)
- **mu** (*float*) – is stability constant

Retorna If the model is stable

Tipo de retorno bool

Indices and tables

- `genindex`
- `modindex`
- `search`

b

base, 5

s

source, 10

stability, 10

w

wave, 9

A

acoustic_impedance() (no módulo base), 6

B

base (módulo), 5

D

discrete_wave_step_3d() (no módulo wave), 9

dispersion() (no módulo stability), 10

F

fp_second_order() (no módulo wave), 9

fp_second_order_explicit_terms() (no módulo wave), 9

fp_second_order_term() (no módulo wave), 9

fpp_fourth_order() (no módulo wave), 9

fpp_fourth_order_term() (no módulo wave), 10

frequency() (no módulo base), 6

P

period() (no módulo base), 6

R

reflection_coefficient() (no módulo base), 6

S

source (módulo), 10

source() (no módulo source), 10

stability (módulo), 10

stability() (no módulo stability), 10

V

velocity_p_by_distance() (no módulo base), 7

velocity_p_by_shear() (no módulo base), 7

velocity_p_by_young() (no módulo base), 7

velocity_ratio() (no módulo base), 7

velocity_ratio_by_deformation() (no módulo base), 8

velocity_ratio_by_poisson() (no módulo base), 8

velocity_s_by_shear() (no módulo base), 8

velocity_s_by_young() (no módulo base), 8

Vp() (no módulo base), 5

Vs() (no módulo base), 5

W

wave (módulo), 9

wave_velocity() (no módulo base), 8

wavelength() (no módulo base), 9