
elegant Documentation

Release 1.0b1

Eduardo Nunes & Fernando Dantas

Jun 09, 2020

CONTENTS:

1	Installation	3
2	Changelog	5
2.1	1.0 (2020-00-00)	5
3	Indices and tables	9
	Python Module Index	11
	Index	13

elegant is a Pure Python Power Systems Simulator. The latest development version can be found [here](#).

INSTALLATION

`elegant` requires `numpy` (for linear algebra), `PyQt5` (for the GUI), `pylatex` (to generate reports), `matplotlib` (for plotting), and `networkx` (for multigraph data structures)

Installing the most recent stable version of the package is as easy as:

```
python3 -m pip install elegant
```


CHANGELOG

2.1 1.0 (2020-00-00)

Initial beta release

2.1.1 Case Study

Inserting buses

Inserting lines

Inserting loads

2.1.2 API

This page details the methods and classes provided by the `elegant` module.

Top-Level classes

```
class elegant.core.Bus (bus_id, v=1.0, delta=0.0, pg=0.0, qg=0.0, pl=0.0, ql=0.0, xd=inf,  
                      iTPG=None, iSLG=None, iDLGb=None, iDLGc=None, iLL=None,  
                      gen_ground=False, load_ground=1)
```

Bases: `object`

property `P`

property `Q`

property `Z`

```
class elegant.core.Keys
```

Bases: `object`

```
add_keyobj (self, extremities, path, obj)
```

```
    get_keyobj (self, extremities, path)
    have_extremities (self, extremities)
class elegant.core.PowerSystem
    Bases: object
    property M
    property N
    property Y
    property Y0
    property Y1
    add_bus (self)
    add_line (self, line, path=None)
    add_trafo (self, trafo, path=None)
    property good_ids
    property hsh
    id2n (self, k)
    property masked_buses
    property masked_lines
    property masked_trafos
    remove_bus (self, n)
    remove_elements_linked_to (self, bus)
    remove_line (self, line, key=None)
    remove_trafo (self, trafo, key=None)
    sort_buses (self)
    update (self, Nmax=100)
    update_flow (self, Nmax=100)
    update_short (self)
class elegant.core.Transformer (orig, dest, snom=100000000.0, jx0=0.5, jx1=0.5, primary=0,
                                secondary=0, v1=0.0, v2=0.0)
    Bases: object
    property Ipu
    property S1
    property S2
    property Sper
    property Z0
    property Z1
```

```
class elegant.core.TransmissionLine(orig, dest, ell=10000.0, r=0.01, d12=1, d23=1, d31=1,  
                                     d=0.5, rho=1.78e-08, m=1, vbase=10000.0, imax=inf,  
                                     v1=0.0, v2=0.0, z=None, y=None)
```

Bases: `object`

property Ia

property Ipu

property Rb

property Rm

property S1

property S2

property Sper

property Tpu

property Y

property Ypu

property Z

property Zc

property Zcpu

property Zpu

property gamma

property param

`elegant.core.gmean` (*arr*)

Numerical Methods

`elegant.methods.gauss_seidel` (*Y, V0, S, eps=None, Niter=1, Nmax=1000*)

Gauss-Seidel Method

Parameters

Y: array, shape (N,N) Ybus matrix

V0: array, shape (N,) Complex initial guess

S: array, shape (N,2) Specified apparent power

eps: float, optional Tolerance

Niter: int, optional Minimum number of iterations (default=1)

Returns

V: array, shape (N,) Bus voltage approximations

`elegant.methods.newton_raphson` (*Y, V0, S, eps=None, Niter=1, Nmax=1000*)

Parameters

Y: admittance matrix

V0: array with initial estimates (1, N)

S: array with specified powers in each bar (N, 2)

eps: defined tolerance, default = None

Niter: max number of iterations, default = 1

Returns

V0: updated array with estimates to the node tensions (1, N)

`elegant.methods.short` (*Y1, Y0, V*)

Calculates three-phase short circuit current levels for each bus

Parameters

Y1: array, shape (N,N) Positive-sequence bus admittance matrix

Y0: array, shape (N,N) Zero-sequence bus admittance matrix

V: array, shape (N,) Pre-fault voltage levels for each bus

Returns

I: array, shape (N, 4, 3) Three-phase current levels for each of the N buses for each of the following fault types:

- Three-phase to ground (TPG);
- Single-line to ground (SLG);
- Double-line to ground (DLG);
- Line-to-line (LL)

Report files

`elegant.report.create_report` (*system, curves, grid, filename*)

INDICES AND TABLES

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

PYTHON MODULE INDEX

e

`elegant.core`, 5
`elegant.methods`, 7
`elegant.report`, 8

A

`add_bus()` (*elegant.core.PowerSystem* method), 6
`add_keyobj()` (*elegant.core.Keys* method), 5
`add_line()` (*elegant.core.PowerSystem* method), 6
`add_trafo()` (*elegant.core.PowerSystem* method), 6

B

`Bus` (class in *elegant.core*), 5

C

`create_report()` (in module *elegant.report*), 8

E

`elegant.core`
 module, 5
`elegant.methods`
 module, 7
`elegant.report`
 module, 8

G

`gamma()` (*elegant.core.TransmissionLine* property), 7
`gauss_seidel()` (in module *elegant.methods*), 7
`get_keyobj()` (*elegant.core.Keys* method), 5
`gmean()` (in module *elegant.core*), 7
`good_ids()` (*elegant.core.PowerSystem* property), 6

H

`have_extremities()` (*elegant.core.Keys* method), 6
`hsh()` (*elegant.core.PowerSystem* property), 6

I

`Ia()` (*elegant.core.TransmissionLine* property), 7
`id2n()` (*elegant.core.PowerSystem* method), 6
`Ipu()` (*elegant.core.Transformer* property), 6
`Ipu()` (*elegant.core.TransmissionLine* property), 7

K

`Keys` (class in *elegant.core*), 5

M

`M()` (*elegant.core.PowerSystem* property), 6

`masked_buses()` (*elegant.core.PowerSystem* property), 6

`masked_lines()` (*elegant.core.PowerSystem* property), 6

`masked_trafos()` (*elegant.core.PowerSystem* property), 6

module
 elegant.core, 5
 elegant.methods, 7
 elegant.report, 8

N

`N()` (*elegant.core.PowerSystem* property), 6
`newton_raphson()` (in module *elegant.methods*), 7

P

`P()` (*elegant.core.Bus* property), 5
`param()` (*elegant.core.TransmissionLine* property), 7
`PowerSystem` (class in *elegant.core*), 6

Q

`Q()` (*elegant.core.Bus* property), 5

R

`Rb()` (*elegant.core.TransmissionLine* property), 7
`remove_bus()` (*elegant.core.PowerSystem* method), 6
`remove_elements_linked_to()` (*elegant.core.PowerSystem* method), 6
`remove_line()` (*elegant.core.PowerSystem* method), 6
`remove_trafo()` (*elegant.core.PowerSystem* method), 6
`Rm()` (*elegant.core.TransmissionLine* property), 7

S

`S1()` (*elegant.core.Transformer* property), 6
`S1()` (*elegant.core.TransmissionLine* property), 7
`S2()` (*elegant.core.Transformer* property), 6
`S2()` (*elegant.core.TransmissionLine* property), 7
`short()` (in module *elegant.methods*), 8
`sort_buses()` (*elegant.core.PowerSystem* method), 6
`Sper()` (*elegant.core.Transformer* property), 6

`Sper()` (*elegant.core.TransmissionLine* property), 7

T

`Tpu()` (*elegant.core.TransmissionLine* property), 7

`Transformer` (class in *elegant.core*), 6

`TransmissionLine` (class in *elegant.core*), 6

U

`update()` (*elegant.core.PowerSystem* method), 6

`update_flow()` (*elegant.core.PowerSystem* method),
6

`update_short()` (*elegant.core.PowerSystem*
method), 6

Y

`Y()` (*elegant.core.PowerSystem* property), 6

`Y()` (*elegant.core.TransmissionLine* property), 7

`Y0()` (*elegant.core.PowerSystem* property), 6

`Y1()` (*elegant.core.PowerSystem* property), 6

`Ypu()` (*elegant.core.TransmissionLine* property), 7

Z

`Z()` (*elegant.core.Bus* property), 5

`Z()` (*elegant.core.TransmissionLine* property), 7

`Z0()` (*elegant.core.Transformer* property), 6

`Z1()` (*elegant.core.Transformer* property), 6

`Zc()` (*elegant.core.TransmissionLine* property), 7

`Zcpu()` (*elegant.core.TransmissionLine* property), 7

`Zpu()` (*elegant.core.TransmissionLine* property), 7