pyCollocation Documentation

Release 0.3.0-alpha

David R. Pugh

July 20, 2016

Contents

Contents:

CHAPTER 1

License

The MIT License (MIT)

CHAPTER 2

pycollocation

2.1 pycollocation package

2.1.1 Submodules

2.1.2 pycollocation.boundary_value_problems module

2.1.3 pycollocation.differential_equations module

2.1.4 pycollocation.models module

2.1.5 pycollocation.orthogonal_polynomials module

2.1.6 pycollocation.solvers module

Objects imported here will live in the *pycollocation.solvers* namespace

class `pycollocation.solvers.`**`LeastSquaresSolver`**(*basis_functions*)
Bases: `pycollocation.solvers.solvers.Solver`

Attributes

`basis_functions`	Functions used to approximate the solution to a boundary value problem.

Methods

`solve`(basis_kwargs, boundary_points, ...)	Solve a boundary value problem using the collocation method.

`solve`(*basis_kwargs*, *boundary_points*, *coefs_array*, *nodes*, *problem*, ***solver_options*)
Solve a boundary value problem using the collocation method.

Parameters **basis_kwargs** : dict

Dictionary of keyword arguments used to build basis functions.

coefs_array : numpy.ndarray

Array of coefficients for basis functions defining the initial condition.

problem : bvp.TwoPointBVPLike

A two-point boundary value problem (BVP) to solve.

solver_options : dict

Dictionary of options to pass to the non-linear equation solver.

class pycollocation.solvers.**Solver**(*basis_functions*)

Bases: pycollocation.solvers.solvers.SolverLike

Attributes

basis_functions	Functions used to approximate the solution to a boundary value problem.

Methods

solve(basis_kwargs, boundary_points, ...)	Solve a boundary value problem using the collocation method.

solve(*basis_kwargs*, *boundary_points*, *coefs_array*, *nodes*, *problem*, ***solver_options*)

Solve a boundary value problem using the collocation method.

Parameters **basis_kwargs** : dict

Dictionary of keyword arguments used to build basis functions.

coefs_array : numpy.ndarray

Array of coefficients for basis functions defining the initial condition.

problem : bvp.TwoPointBVPLike

A two-point boundary value problem (BVP) to solve.

solver_options : dict

Dictionary of options to pass to the non-linear equation solver.

class pycollocation.solvers.**Solution**(*basis_kwargs*, *functions*, *nodes*, *problem*, *residual_function*, *result*)

Bases: pycollocation.solvers.solutions.SolutionLike

Class representing the solution to a Boundary Value Problem (BVP).

Attributes

basis_kwargs
functions
nodes
problem
residual_function
result

Methods

evaluate_residual(points)	
evaluate_solution(points)	
normalize_residuals(points)	Normalize residuals by the level of the variable.

evaluate_residual(*points*)

evaluate_solution(*points*)

normalize_residuals(*points*)
Normalize residuals by the level of the variable.

class pycollocation.solvers.**SolutionLike**
Bases: object

Attributes

basis_kwargs
functions
nodes
problem
residual_function
result

basis_kwargs

functions

nodes

problem

residual_function

result

class pycollocation.solvers.**SolverLike**
Bases: object

Class describing the protocol the all SolverLike objects should satisfy.

Notes

Subclasses should implement *solve* method as described below.

Attributes

basis_functions	Functions used to approximate the solution to a boundary value problem.

Methods

solve(basis_kwargs, boundary_points, ...)	Solve a boundary value problem using the collocation method.

basis_functions
Functions used to approximate the solution to a boundary value problem.

Getter Return the current basis functions.

Type *basis_functions.BasisFunctions*

solve(*basis_kwargs*, *boundary_points*, *coefs_array*, *nodes*, *problem*, ***solver_options*)
Solve a boundary value problem using the collocation method.

Parameters **basis_kwargs** : dict

Dictionary of keyword arguments used to build basis functions.

coefs_array : numpy.ndarray

Array of coefficients for basis functions defining the initial condition.

problem : bvp.TwoPointBVPLike

A two-point boundary value problem (BVP) to solve.

solver_options : dict

Dictionary of options to pass to the non-linear equation solver.

2.1.7 pycollocation.symbolics module

2.1.8 pycollocation.version module

2.1.9 pycollocation.visualizers module

2.1.10 Module contents

Objects imported here will live in the *pycollocation* namespace

CHAPTER 3

Indices and tables

- genindex
- modindex
- search

p

B

E

F

L

N

P

R

S