# Iterative Methods Documentation

- Prathamesh, Sai Krishna Chaitanya

This is the class and function reference of iterative methods.

---

# Packages in the project

- **generate**: provides **module generators** that helps generate matrix (A), system of linear equations (Ax = b) and store the generated system.
- **solve**: provides **module solvers** that helps execute iterative methods to solve the stored system.
- **visualize**: provides **module visualizers** that helps visualize performance of iterative methods.

---

# Module generate.generators [source]

## class generate.generators.MatrixGenerator() [source]

### Parameters

No parameters.

### Attributes

No attributes.

### Methods

| | |
|---|---|
| generate_random(dim, diagonally_dominant) | Generate a random matrix. |
| generate_nonnegative(dim, diagonally_dominant) | Generate a nonnegative matrix. |
| generate_positive(dim, diagonally_dominant) | Generate a positive matrix. |
| generate_Z(dim, diagonally_dominant) | Generate a Z matrix. |
| generate_Q(dim, diagonally_dominant) | Generate a Q matrix. |
| generate_tridiagonal(dim, diagonally_dominant) | Generate a tridiagonal matrix. |
| generate_triangular(dim, diagonally_dominant, upper) | Generate a triangular matrix. |
| generate(dim, kind, diagonally_dominant) | Generate a given kind of matrix. |

## class generate.generators.System(A, x_true, b, kind, diagonally_dominant, dim) [source]

### Parameters

| | | | |
|---|---|---|---|
| A | numpy.ndarray | default = None | Matrix A in the system Ax = b. |
| x_true | numpy.ndarray | default = None | True solution vector x in the system Ax = b. |
| b | numpy.ndarray | default = None | Vector b in the system Ax = b. |
| kind | str | default = None | Kind of a matrix A.<br>possible values: ‘random’, ‘nonnegative’, ‘positive’, ‘Z’, ‘Q’, ‘tridiagonal’, ‘triangular_U’, ‘triangular_L’ . |
| diagonally_dominant | bool | default = None | True ensures diagonal dominance. |
| dim | int | default = None | Dimension of matrix A. |

### Attributes

No attributes.

### Methods

No methods.

## class generate.generators.SystemGenerator() [source]

### Parameters

No parameters.

### Attributes

| | | |
|---|---|---|
| mg | generate.generators.MatrixGenerator | helps generate matrix A in the system Ax = b. |

### Methods

| | |
|---|---|
| load(file, kind, diagonally_dominant) | Load the matrix from .mtx file. |
| generate(dim, kind, diagonally_dominant) | Generate a system of linear equations. |

# Module solve.solvers [source]

## class solve.solvers.IterativeSolver(system, diagonal_list, x = None, use_modified_method=False, compute_spectral_radius=False, copy=True, warm_start=False) [source]

### Parameters

| | | | |
|---|---|---|---|
| system | generate.generators.System | default = None | System Ax = b that is to be solved. |
| diagonal_list | list | default = None | List of indices of diagonal. |
| x | numpy.ndarray | default = None | Initial guess of a solution vector. |
| use_modified_method | bool | default = False | If False, uses standard iterative method. If True, uses modified iterative method. |
| compute_spectral_radius | bool | default = False | If True, computes spectral radius of iteration matrix. |
| copy | bool | default = True | If True, copies the given system. |
| warm_start | bool | default = False | If True, reuses the solution of previous call as initial guess of a solution vector. |

### Attributes

| | | |
|---|---|---|
| A | numpy.ndarray | Matrix A in the system of linear equations Ax = b. |
| b | numpy.ndarray | Vector b in the system of linear equations Ax = b. |
| n | int | Dimension of a matrix A. |
| kind | str | Kind of a matrix A. |
| noi | int | Number of iterations taken for convergence. |
| x_true | numpy.ndarray | True solution vector. |
| zero_check_passed | bool | If True, diagonal of A contains 0. |
| splitted | bool | If True, iteration matrix T and vector c in the iteration equation have been computed. |
| spectral_radius | float | Spectral radius of iteration matrix T. |
| l_inf_values | list | List of $\|x-x^*\|_\infty$ values for every iteration until convergence. |
| converged | bool | If True, convergence has been achieved. |
| x | numpy.ndarray | Solution vector to which the iterations converged. |

### Methods

| | |
|---|---|
| zero_along_diagonal() | Check if there is a 0 along the diagonal. |
| all_ones_along_diagonal() | Check if all diagonal elements are 1. |
| make_diagonal_zero(diagonal_index) | Apply transformations to make diagonals represented by diagonal_index 0. |
| compute_spectral_radius() | Compute spectral radius of iteration matrix. |
| iterate(tol, max_iters) | Perform iterations of the iteration equation. |
| solve(tol, max_iters) | Solve the system of linear equations Ax = b. |

## class solve.solvers.Jacobi(system, x = None, use_modified_method=False, compute_spectral_radius=False, copy=True, warm_start=False, diagonal_list=[1]) [source]

Inherits solve.solvers.IterativeSolver.

### Parameters

| | | | |
|---|---|---|---|
| system | generate.generators.System | default = None | System Ax = b that is to be solved. |
| x | numpy.ndarray | default = None | Initial guess of a solution vector. |
| use_modified_method | bool | default = False | If False, uses standard jacobi method.<br>If True, uses modified jacobi method. |
| compute_spectral_radius | bool | default = False | If True, computes spectral radius of iteration matrix. |
| copy | bool | default = True | If True, copies the given system. |
| warm_start | bool | default = False | If True, reuses the solution of previous call as initial guess of a solution vector. |
| diagonal_list | list | default = [1] | List of indices of diagonal. |

### Attributes

| | | |
|---|---|---|
| T | numpy.ndarray | Iteration matrix T in the iteration equation. |
| c | numpy.ndarray | Vector c in the iteration equation. |

### Methods

| | |
|---|---|
| split() | Perform the usual splitting and compute iteration matrix T and vector c in the iteration equation $x^{(k+1)} = Tx^{(k)} + c$ |

## class solve.solvers.GaussSeidel(system, x = None, use_modified_method=False, compute_spectral_radius=False, copy=True, warm_start=False, diagonal_list=[1]) [source]

Inherits solve.solvers.IterativeSolver.

### Parameters

| | | | |
|---|---|---|---|
| system | generate.generators.System | default = None | System Ax = b that is to be solved. |
| x | numpy.ndarray | default = None | Initial guess of a solution vector. |
| use_modified_method | bool | default = False | If False, uses standard jacobi method.<br>If True, uses modified jacobi method. |
| compute_spectral_radius | bool | default = False | If True, computes spectral radius of iteration matrix. |
| copy | bool | default = True | If True, copies the given system. |

| | | | |
|---|---|---|---|
| warm_start | bool | default = False | If True, reuses the solution of previous call as initial guess of a solution vector. |
| diagonal_list | list | default = [1] | List of indices of diagonal. |

### Attributes

| | | |
|---|---|---|
| T | numpy.ndarray | Iteration matrix T in the iteration equation. |
| c | numpy.ndarray | Vector c in the iteration equation. |

### Methods

| | |
|---|---|
| split() | Perform the usual splitting and compute iteration matrix T and vector c in the iteration equation $x^{(k+1)} = Tx^{(k)} + c$ |

## class solve.solvers.Milaszewicz(system, k, method='jacobi', x = None, use_modified_method=False, compute_spectral_radius=False, copy=True, warm_start=False, diagonal_list=[1]) [source]

### Parameters

| | | | |
|---|---|---|---|
| system | generate.generators.System | default = None | System Ax = b that is to be solved. |
| k | int | default = None | Value of k in milaszewicz method. |
| method | str | default = ‘jacobi’ | Method flag that decides the usual splitting and iteration matrix T as well as vector c in the iteration equation $x^{(k+1)} = Tx^{(k)} + c$ |
| x | numpy.ndarray | default = None | Initial guess of a solution vector. |
| use_modified_method | bool | default = False | If False, uses standard jacobi method. If True, uses modified jacobi method. |
| compute_spectral_radius | bool | default = False | If True, computes spectral radius of iteration matrix. |
| copy | bool | default = True | If True, copies the given system. |
| warm_start | bool | default = False | If True, reuses the solution of previous call as initial guess of a solution vector. |
| diagonal_list | list | default = [1] | List of indices of diagonal. |

### Attributes

| | | |
|---|---|---|
| solver | solve.solvers.Jacobi if method = ‘jacobi’. solve.solvers.GaussSeidel if method = ‘gauss_seidel’ | Solver that performs usual splitting and executes iterations. |
| solver_name | str | Name of the solver. |

### Methods

| | |
|---|---|
| perform_elimination() | Perform the gaussian elimination steps of milaszewicz method. |
| solve(tol, max_iters) | Solve the system of linear equations Ax = b. |

# Module visualize.visualizers [source]

Functions

| | |
|---|---|
| show_l_inf_plot(kind, dim, method, spectral_radius, l_inf_values, s, folder) | Show a scatter plot with iteration number on x-axis and $\|x-x^*\|_\infty$ on y-axis. |
| show_iterations_plot(kind, dim, y, iteration_values, s, color, folder) | Show a horizontal bar chart with number of iterations on x-axis and iterative method on y-axis. |
| show_spectral_radius_plot(kind, dim, y, spectral_radius_values, s, color, folder) | Show a horizontal bar chart with spectral radius on x-axis and iterative method on y-axis. |

# Dependencies

Iterative methods package uses following standard libraries.

1. operator

Iterative methods package uses following third-party libraries.

1. matplotlib
2. numpy
3. scipy
4. tabulate