pyrw Documentation

Release 0.1

Alexander Blaessle

Nov 06, 2018

Contents

1 API: **1**

1.1 pyrw.RWdomain module . . . 1
1.2 pyrw.RWHC module . . . 3
1.3 pyrw.RWrun module . . . 3
1.4 pyrw.RWstep module . . . 3
1.5 pyrw.RWessentials module . . . 4
1.6 pyrw.RWmisc module . . . 6
1.7 pyrw.RWRW module . . . 7
1.8 pyrw.RWsuperposition module . . . 7
1.9 pyrw.RWBC module . . . 8
1.10 pyrw.RWgeometry module . . . 8
1.11 pyrw.RWwalker module . . . 12

2 Documentation: **15**

2.1 Need Help?: . . . 15

3 Indices and tables **17**

Python Module Index **19**

CHAPTER 1

API:

1.1 pyrw.RWdomain module

class pyrw.RWdomain.**domain**(*RW*, *typ='poly'*)

Domain class for defining random walk domain.

Parameters

- **RW** (*pyrw.RWRW*) – A pyrw RW.
- **typ** (*str*) – Type of domain

addArc(*vstart*, *vcenter=None*, *vend=None*, *angle=None*)

Adds arc from vstart to vend around vcenter to domain.

Parameters

- **vstart** (*pyrw.geometry.vertex*) – Vertex object.
- **vcenter** (*pyrw.geometry.vertex*) – Vertex object.
- **vend** (*pyrw.geometry.vertex*) – Vertex object.
- **angle** (*float*) – Angle.

Returns pyrw.geometry.arc – arc object

addCircle(*vcenter*, *radius*, *BC=''*)

Adds circle around vcenter with r=radius to domain.

Parameters

- **vcenter** (*pyrw.geometry.vertex*) – Vertex object.
- **radius** (*float*) – Radius of circle.

Returns pyrw.geometry.circle – circle object

addLine(*v1*, *v2*)

Adds line from v1 to v2 to domain.

Parameters

- **v1** (*pyrw.geometry.vertex*) – Vertex object.
- **v2** (*pyrw.geometry.vertex*) – Vertex object.

Returns pyrw.geometry.line – line object

addRectangle(*voffset*, *lenx*, *leny*)

Adds rectangle with offset voffset and sidelengths lenx,leny to domain.

Parameters

- **voffset** (*pyrw.geometry.vertex*) – Vertex object.
- **lenx** (*float*) – Sidelength in x direction.
- **leny** (*float*) – Sidelength in y direction.

Returns pyrw.geometry.rectangle – rectangle object

addVertex(*x*)

Adds vertex to domain.

Parameters **x** (*pyrw.RWRW*) – 2D coordinate.

Returns pyrw.geometry.vertex – vertex object

draw(*r=None*, *color=None*, *ann=False*)

Draw all geometrical elements associated with domain.

Parameters

- **r** (*pyrw.RWrun*) – pyrw run, if None, picks last run of main pyrw.RWRW object.
- **color** (*matplotlib color*) – color of elements in matplotlib syntax, e.g. 'r' or (0.1,1,0.5)
- **ann** (*bool*) – Annotation Flag

edgeByID(*ID*)

Returns edge given its ID

Parameters **ID** (*int*) – ID of edge.

Returns pyrw.geometry.edge – edge object

genRandomPoint()

Returns random coordinate inside domain.

Returns array – coordinate

getExtend()

Returns x-y-extend of domain.

Returns (float,float,float,float) – (minX,maxX,minY,maxY)

setRW(*rw*)

verticesCoordsToList()

Returns list with coordinates of all vertices of domain.

Returns list – list with coordinates

1.2 pyrw.RWHC module

class pyrw.RWHC.**HC**(*RW*, *w*, *typ*, *Id*, *suc=1.0*)

Bases: object

hitRW(*x1*, *x2*, *r1*)

class pyrw.RWHC.**none**(*RW*, *w*, *Id*, *suc=1.0*)

Bases: *pyrw.RWHC.HC*

hit()

class pyrw.RWHC.**stop**(*RW*, *w*, *Id*, *suc=1.0*)

Bases: *pyrw.RWHC.HC*

hit(*suc=1.0*, *debug=False*)

1.3 pyrw.RWrun module

class pyrw.RWrun.**run**(*RW*)

checkFig()

computeEndStastics()

doStep(*debug=False*)

getGroupOfInterest()

getRuntime()

getWalkerOfInterest()

plotStep(*color=None*)

plotTraj(*color=None*)

scalePlot()

setGroupOfInterest(*group*)

setWalkerOfInterest(*w*)

start(*plotStep=False*, *printProcess=False*)

stop(*stopped*)

1.4 pyrw.RWstep module

class pyrw.RWstep.**CCRWstep**(*w*, *r1*, *r2*, *gamma*, *kappa*)

Bases: *pyrw.RWstep.step*

scaleR(*rScale*)

setGamma(*gamma*)

setKappa(*kappa*)

setParms(*parms*)

setR1(*r*)

setR2(*r*)

class pyrw.RWstep.**CRWstep**(*w*, *r1*, *r2*, *gamma*)
Bases: *pyrw.RWstep.step*

class pyrw.RWstep.**CorRWstep**(*w*, *r*, *kappa*)
Bases: *pyrw.RWstep.step*

class pyrw.RWstep.**MRWstep**(*w*, *r*)
Bases: *pyrw.RWstep.step*

class pyrw.RWstep.**SCCRWstep**(*w*, *r1*, *r2*, *gamma*, *kappa*, *gammaSup=1*, *gammaStep=0.1*, *gammaMin=0.2*)
Bases: *pyrw.RWstep.CCRWstep*

performStep()
Overwrite performStep.

setBackGamma()

setGammaMin(*gammaMin*)

setGammaStep(*gammaStep*)

setOrigGamma(*gamma*)

updateGamma(*gamma*)

class pyrw.RWstep.**step**(*w*, *typ*)
Bases: object

addSuperposition(*r*, *gamma*, *kappa*)

checkGammas(*debug=False*)

getSuperpositionIds()

performStep()

updateGammaDist()

1.5 pyrw.RWessentials module

pyrw.RWessentials.**angle_from_vertices**(*vcenter*, *vstart*, *vend*, *samedist=False*)
Returns angle between two vertices around vcenter.

Parameters

- **vcenter** (*pyrw.geometry.vertex*) – Vertex object.
- **vstart** (*pyrw.geometry.vertex*) – Vertex object.
- **vend** (*pyrw.geometry.vertex*) – Vertex object.
- **samedist** (*bool*) – Flag if vectors need to have same length.

Returns float – angle

pyrw.RWessentials.**checkInsideCircle**(*x*, *center*, *radius*, *tol=1e-30*, *debug=False*)
Returns True if x is inside circle.

Parameters

- **x** (*array*) – Coordinate.

- **center** (*float*) – Coordinate of center.
- **radius** (*float*) – Radius of circle.
- **center** – Tolerance added to check to avoid rounding errors.
- **debug** (*bool*) – Debugging flag.

Returns bool – True if inside

`pyrw.RWessentials.checkInsidePoly`(*coord*, *poly*)

Returns True if coord is inside poly.

Taken from http://www.ariel.com.au/a/python-point-int-poly.html

Parameters

- **coord** (*array*) – Coordinate.
- **poly** (*list*) – Polygon corners.

Returns bool – True if inside

`pyrw.RWessentials.combinations`(*arrays*, *out=None*)

Returns cartesian product of arrays.

Parameters

- **arrays** (*list*) – List of arrays.
- **out** (*array*) – Result vector.

Returns array – Cartesian product

`pyrw.RWessentials.compute_angle`(*vec1*, *vec2*)

Returns angle between two vectors.

Parameters

- **vec1** (*array*) – Vector.
- **vec2** (*array*) – Vector.

Returns float – angle

`pyrw.RWessentials.create_arc_curve`(*vcenter*, *angle*, *angle_offset*, *radius*, *steps=100*)

Creates array with points describing the curve of an arc.

Parameters

- **vcenter** (*pyrw.geometry.vertex*) – Vertex object.
- **angle** (*float*) – Angle of arc.
- **angle_offset** (*float*) – Offset angle of arc.
- **radius** (*float*) – radius of arc.
- **steps** (*int*) – number of steps for arc.

Returns (array,array) – vectors describing x/y-coordinates

`pyrw.RWessentials.cross2d`(*x1*, *x2*)

Returns reduced 2D cross product.

Parameters

- **x1** (*array*) – Coordinate 1.
- **x2** (*array*) – Coordinate 2.

Returns float – cross product

`pyrw.RWessentials.direc_angle(v1, v2)`
Returns directional angle.

Parameters

- **v1** (*array*) – Vector.
- **v2** (*array*) – Vector.

Returns array – directional angle

`pyrw.RWessentials.dist(p1, p2)`
Returns euclidean distance between two points.

Parameters

- **p1** (*array*) – Point 1.
- **p2** (*array*) – Point 2.

Returns float – distance

`pyrw.RWessentials.norm(vec)`
Returns L2-norm of vector.

Parameters **vec** (*array*) – Vector.

Returns float – length of vector

`pyrw.RWessentials.outwPerp(x, c)`
Returns outward perpendicular vector.

Parameters

- **x** (*array*) – Vector.
- **c** (*array*) – Center coordinate.

Returns array – perpendicular vector

`pyrw.RWessentials.perp(x)`
Returns perpendicular vector.

Parameters **x** (*array*) – Vector.

Returns array – perpendicular vector

`pyrw.RWessentials.unit_vector(vector)`
Returns unit vector.

Parameters **vector** (*array*) – Vector.

Returns array – unit vector

1.6 pyrw.RWmisc module

`pyrw.RWmisc.listToCSV(l, fn=None)`

`pyrw.RWmisc.loadFromPickle(fn)`

`pyrw.RWmisc.printVariable(var, obj)`

`pyrw.RWmisc.saveToPickle(obj, fn=None)`

`pyrw.RWmisc.setVariable`(*var*, *obj*, *val*)

1.7 pyrw.RWRW module

class `pyrw.RWRW.RW`(*tstart=0.0*, *tend=1000.0*, *deltat=1.0*, *N=5*, *P=1*, *name='RW'*)

- **addNRuns**(*N*)
- **addRun**()
- **addWalker**(*x0=<Mock name='mock.array()' id='139686300908560'>*, *RWtyp='MRW'*, *color='b'*, *BCtyp='sticky'*, *HCtyp=None*, *hitGroup=[]*, *typ='typ0'*, *hitTypes=None*, *detectRadius=1.0*, *successRate=1.0*)
- **computeStatistics**()
- **getDomain**()
- **getMeanRuntime**()
- **getTotalRuntime**()
- **getWalkerById**(*Id*)
- **printStatistics**()
- **printVariable**(*var*)
- **runAll**(*printProcess=True*, *printRunProcess=False*, *plotStep=False*)
- **saveToFile**(*fn=None*)
- **setDomain**(*d*)
- **setVarForAllRuns**(*var*, *val*)
- **statsToCSV**(*fn=None*)

1.8 pyrw.RWsuperposition module

class `pyrw.RWsuperposition.superposition`(*w*, *r*, *gamma*, *kappa*, *Id*)

- **doStep**()
- **getGamma**()
- **getKappa**()
- **getR**()
- **setGamma**(*gamma*)
- **setKappa**(*kappa*)
- **setR**(*r*)

1.9 pyrw.RWBC module

class pyrw.RWBC.**BC**(*w*, *typ*, *Id*, *edge*, *direcBehaviour*)

Bases: object

arcIntersect(*x1*, *x2*, *arc*, *tol=1e-30*, *spacer=1e-05*, *breakAtProblem=False*, *debug=False*)

arcPreChecks(*x1*, *x2*, *arc*, *tol=1e-30*, *breakAtProblem=False*, *debug=False*)

checkArcIntersect(*x1*, *x2*, *arc*, *tol=1e-30*, *spacer=1e-05*, *breakAtProblem=False*, *preCheck=True*, *debug=False*)

circCheck(*x*, *center*, *radius*, *tol=1e-30*, *debug=False*)

computeArcIntersect(*x1*, *x2*, *arc*, *a*, *b*, *discriminant*, *spacer=1e-05*, *debug=False*)

cross2d(*x1*, *x2*)

findIntersectPoly(*xold*, *xnew*, *poly*)

perp(*x*)

segIntersect(*x1*, *x2*, *y1*, *y2*, *debug=False*)

class pyrw.RWBC.**setBack**(*w*, *Id*, *edge*, *direcBehaviour=1*)

Bases: *pyrw.RWBC.BC*

circleHit(*xold*, *xnew*, *breakAtProblem=False*, *debug=False*)

hit(*xold*, *xnew*, *breakAtProblem=False*, *debug=False*)

class pyrw.RWBC.**sticky**(*w*, *Id*, *edge*, *direcBehaviour=1*)

Bases: *pyrw.RWBC.BC*

hit(*xold*, *xnew*, *debug=False*)

1.10 pyrw.RWgeometry module

class pyrw.RWgeometry.**arc**(*domain*, *vstart*, *vcenter*, *Id*, *vend=None*, *angle=None*)

Bases: *pyrw.RWgeometry.edge*

Arc class for defining domain geometry.

Parameters

- **domain** (*pyrw.RWdomain*) – A pyrw domain.
- **vstart** (pyrw.RWgeometry.vertex) – Start vertex of the arc.
- **vcenter** (pyrw.RWgeometry.vertex) – Center vertex of the arc.
- **Id** (*int*) – ID of arc.
- **vend** (pyrw.RWgeometry.vertex) – Second vertex of the arc.
- **angle** (*float*) – Angle of arc.

computeAngle(*debug=False*)

Computes both angle and angle_offset of arc

Parameters **debug** (*bool*) – Debugging flag.

Returns (float,float) – angle, angle_offset

computePoint (*angle*, *radius*)
Computes point on arc

Parameters

- **angle** (*float*) – Point's angle.
- **radius** (*float*) – Point's radius.

computeRadius ()

computeVend ()

computeVstart ()

draw (*r=None*, *color=None*, *ann=False*)
Draw arc

Parameters

- **r** (*pyrw.RWrun*) – pyrw run, if None, picks last run of main pyrw.RWRW object.
- **color** (*matplotlib color*) – color of arc in matplotlib syntax, e.g. 'r' or (0.1,1,0.5)
- **ann** (*bool*) – Annotation Flag

genFromAngle (*vcenter*, *vstart*, *angle*)
Initializes arc if angle was given at initialization

Parameters

- **vstart** (pyrw.RWgeometry.vertex) – Start vertex of the arc.
- **vcenter** (pyrw.RWgeometry.vertex) – Center vertex of the arc.
- **angle** (*float*) – Angle of arc.

genFromPoints (*vcenter*, *vstart*, *vend*)
Initializes arc if vend was given at initialization

Parameters

- **vstart** (pyrw.RWgeometry.vertex) – Start vertex of the arc.
- **vcenter** (pyrw.RWgeometry.vertex) – Center vertex of the arc.
- **vend** (pyrw.RWgeometry.vertex) – Second vertex of the arc.

getAngle ()

getAngleOffset ()

getRadius ()

getVcenter ()

getVend ()

getVstart ()

getXcenter ()

getXend ()

getXstart ()

inArc (*x*, *debug=False*)
Checks if points x lies on arc

Parameters

- **x** (*array*) – coordinate.
- **debug** (*bool*) – Debugging flag.

setAngle(*angle*)

setRadius(*radius*)

class pyrw.RWgeometry.**circle**(*domain*, *vcenter*, *radius*)

Circle class for defining domain geometry.

Parameters

- **domain** (*pyrw.RWdomain*) – A pyrw domain.
- **vcenter** (pyrw.RWgeometry.vertex) – Center vertex of the arc.
- **radius** (*float*) – Radius.

draw()

getArcs()

getDomain()

getRadius()

getVcenter()

getVertices()

getXcenter()

setRadius(*radius*)

setVcenter(*v*)

setXcenter(*x*)

class pyrw.RWgeometry.**edge**(*domain*, *Id*, *typ*)

Edge class for defining domain geometry.

Parameters

- **domain** (*pyrw.RWdomain*) – A pyrw domain.
- **Id** (*int*) – ID of edge.
- **typ** (*int*) – Edge type

decodeTyp()

Returns type of edge in words.

Returns str – type of the edge

getDomain()

getId()

getTyp()

class pyrw.RWgeometry.**line**(*domain*, *v1*, *v2*, *Id*)

Bases: *pyrw.RWgeometry.edge*

Line class for defining domain geometry.

Parameters

- **domain** (*pyrw.RWdomain*) – A pyrw domain.

- **Id** (*int*) – ID of line.
- **v1** (pyrw.RWgeometry.vertex) – Start vertex of the line.
- **v2** (pyrw.RWgeometry.vertex) – Second vertex of the line.

draw(*r=None*, *color=None*, *ann=False*)
Draw line

Parameters

- **r** (*pyrw.RWrun*) – pyrw run, if None, picks last run of main pyrw.RWRW object.
- **color** (*matplotlib color*) – color of line in matplotlib syntax, e.g. 'r' or (0.1,1,0.5)
- **ann** (*bool*) – Annotation Flag

class pyrw.RWgeometry.**rectangle**(*domain*, *voffset*, *lenx*, *leny*)
Rectangle class for defining domain geometry.

Parameters

- **domain** (*pyrw.RWdomain*) – A pyrw domain.
- **voffset** (pyrw.RWgeometry.vertex) – Offset vertex of the arc.
- **lenx** (*float*) – Sidelength in x direction.
- **leny** (*float*) – Sidelength in y direction.

draw(*r=None*, *color=None*, *ann=None*)

getDomain()

getEdges()

getLenx()

getLeny()

getVertices()

getVoffset()

getXoffset()

setLenx(*lenx*)

setLeny(*leny*)

setXoffset(*xoffset*)

class pyrw.RWgeometry.**vertex**(*domain*, *x*, *Id*)
Vertex class for defining domain geometry.

Parameters

- **domain** (*pyrw.RWdomain*) – A pyrw domain.
- **x** (*array*) – A coordinate, e.g. [0,1].
- **Id** (*int*) – ID of vertex.

draw(*r=None*, *color=None*, *ann=False*)
Draw vertex

Parameters

- **r** (*pyrw.RWrun object*) – pyrw run, if None, picks last run of main pyrw.RWRW object.

- **color** (*matplotlib color*) – color of vertex in matplotlib syntax, e.g. 'r' or (0.1,1,0.5)
- **ann** (*bool*) – Annotation Flag

setX(*x*)

Define vertex location

Parameters **x** (*array*) – A coordinate, e.g. [0,1]

1.11 pyrw.RWwalker module

class pyrw.RWwalker.**walker**(*RW, x0=<Mock name='mock.array()' id='139686300908560'>, RWtyp='MRW', color='r', wid=None, BCs=[], BCtyp='sticky', HCtyp=None, hitGroup=[], typ='typ0', hitTypes=None, detectRadius=1.0, successRate=1.0*)

allBCsSame()

checkBC()

checkHC()

computeSquaredDisplacement()

computeStatistics()

computeTrajLength()

genBC(*typ, edge*)

genHC(*typ, hitGroup*)

genHitGroup()

genRandomX0()

genStep(*typ, rs=[1.0, 3.0], gammas=[0.5, 0.5], kappas=[0.0, 3.0], gammaSup=1, gammaStep=0.1, gammaMin=0.2*)

getBCIds()

getDetectRadius()

getHitGroupLoc()

getSquaredDisplacement()

getTrajLength()

getX0()

plotTraj(*draw=True, color=None*)

setAllBCs(*typ*)

setBC(*typ, ID*)

setCurrRun(*r*)

setDetectRadius(*radius*)

setToStart()

setX0(*x0*)

toTraj()

updateBCIds()

updateBCTyp()

CHAPTER 2

Documentation:

2.1 Need Help?:

Contact alexander.blaessle[at]gmail.com.

CHAPTER 3

Indices and tables

- genindex
- modindex
- search

Python Module Index

p

pyrw.RWBC, 8
pyrw.RWdomain, 1
pyrw.RWessentials, 4
pyrw.RWgeometry, 8
pyrw.RWHC, 3
pyrw.RWmisc, 6
pyrw.RWrun, 3
pyrw.RWRW, 7
pyrw.RWstep, 3
pyrw.RWsuperposition, 7
pyrw.RWwalker, 12

Index

A

addArc() (pyrw.RWdomain.domain method), 1
addCircle() (pyrw.RWdomain.domain method), 1
addLine() (pyrw.RWdomain.domain method), 1
addNRuns() (pyrw.RWRW.RW method), 7
addRectangle() (pyrw.RWdomain.domain method), 2
addRun() (pyrw.RWRW.RW method), 7
addSuperposition() (pyrw.RWstep.step method), 4
addVertex() (pyrw.RWdomain.domain method), 2
addWalker() (pyrw.RWRW.RW method), 7
allBCsSame() (pyrw.RWwalker.walker method), 12
angle_from_vertices() (in module pyrw.RWessentials), 4
arc (class in pyrw.RWgeometry), 8
arcIntersect() (pyrw.RWBC.BC method), 8
arcPreChecks() (pyrw.RWBC.BC method), 8

B

BC (class in pyrw.RWBC), 8

C

CCRWstep (class in pyrw.RWstep), 3
checkArcIntersect() (pyrw.RWBC.BC method), 8
checkBC() (pyrw.RWwalker.walker method), 12
checkFig() (pyrw.RWrun.run method), 3
checkGammas() (pyrw.RWstep.step method), 4
checkHC() (pyrw.RWwalker.walker method), 12
checkInsideCircle() (in module pyrw.RWessentials), 4
checkInsidePoly() (in module pyrw.RWessentials), 5
circCheck() (pyrw.RWBC.BC method), 8
circle (class in pyrw.RWgeometry), 10
circleHit() (pyrw.RWBC.setBack method), 8
combinations() (in module pyrw.RWessentials), 5
compute_angle() (in module pyrw.RWessentials), 5
computeAngle() (pyrw.RWgeometry.arc method), 8
computeArcIntersect() (pyrw.RWBC.BC method), 8
computeEndStastics() (pyrw.RWrun.run method), 3
computePoint() (pyrw.RWgeometry.arc method), 8
computeRadius() (pyrw.RWgeometry.arc method), 9
computeSquaredDisplacement() (pyrw.RWwalker.walker method), 12
computeStatistics() (pyrw.RWRW.RW method), 7
computeStatistics() (pyrw.RWwalker.walker method), 12
computeTrajLength() (pyrw.RWwalker.walker method), 12
computeVend() (pyrw.RWgeometry.arc method), 9
computeVstart() (pyrw.RWgeometry.arc method), 9
CorRWstep (class in pyrw.RWstep), 4
create_arc_curve() (in module pyrw.RWessentials), 5
cross2d() (in module pyrw.RWessentials), 5
cross2d() (pyrw.RWBC.BC method), 8
CRWstep (class in pyrw.RWstep), 4

D

decodeTyp() (pyrw.RWgeometry.edge method), 10
direc_angle() (in module pyrw.RWessentials), 6
dist() (in module pyrw.RWessentials), 6
domain (class in pyrw.RWdomain), 1
doStep() (pyrw.RWrun.run method), 3
doStep() (pyrw.RWsuperposition.superposition method), 7
draw() (pyrw.RWdomain.domain method), 2
draw() (pyrw.RWgeometry.arc method), 9
draw() (pyrw.RWgeometry.circle method), 10
draw() (pyrw.RWgeometry.line method), 11
draw() (pyrw.RWgeometry.rectangle method), 11
draw() (pyrw.RWgeometry.vertex method), 11

E

edge (class in pyrw.RWgeometry), 10
edgeByID() (pyrw.RWdomain.domain method), 2

F

findIntersectPoly() (pyrw.RWBC.BC method), 8

G

genBC() (pyrw.RWwalker.walker method), 12
genFromAngle() (pyrw.RWgeometry.arc method), 9

genFromPoints() (pyrw.RWgeometry.arc method), 9
genHC() (pyrw.RWwalker.walker method), 12
genHitGroup() (pyrw.RWwalker.walker method), 12
genRandomPoint() (pyrw.RWdomain.domain method), 2
genRandomX0() (pyrw.RWwalker.walker method), 12
genStep() (pyrw.RWwalker.walker method), 12
getAngle() (pyrw.RWgeometry.arc method), 9
getAngleOffset() (pyrw.RWgeometry.arc method), 9
getArcs() (pyrw.RWgeometry.circle method), 10
getBCIds() (pyrw.RWwalker.walker method), 12
getDetectRadius() (pyrw.RWwalker.walker method), 12
getDomain() (pyrw.RWgeometry.circle method), 10
getDomain() (pyrw.RWgeometry.edge method), 10
getDomain() (pyrw.RWgeometry.rectangle method), 11
getDomain() (pyrw.RWRW.RW method), 7
getEdges() (pyrw.RWgeometry.rectangle method), 11
getExtend() (pyrw.RWdomain.domain method), 2
getGamma() (pyrw.RWsuperposition.superposition method), 7
getGroupOfInterest() (pyrw.RWrun.run method), 3
getHitGroupLoc() (pyrw.RWwalker.walker method), 12
getId() (pyrw.RWgeometry.edge method), 10
getKappa() (pyrw.RWsuperposition.superposition method), 7
getLenx() (pyrw.RWgeometry.rectangle method), 11
getLeny() (pyrw.RWgeometry.rectangle method), 11
getMeanRuntime() (pyrw.RWRW.RW method), 7
getR() (pyrw.RWsuperposition.superposition method), 7
getRadius() (pyrw.RWgeometry.arc method), 9
getRadius() (pyrw.RWgeometry.circle method), 10
getRuntime() (pyrw.RWrun.run method), 3
getSquaredDisplacement() (pyrw.RWwalker.walker method), 12
getSuperpositionIds() (pyrw.RWstep.step method), 4
getTotalRuntime() (pyrw.RWRW.RW method), 7
getTrajLength() (pyrw.RWwalker.walker method), 12
getTyp() (pyrw.RWgeometry.edge method), 10
getVcenter() (pyrw.RWgeometry.arc method), 9
getVcenter() (pyrw.RWgeometry.circle method), 10
getVend() (pyrw.RWgeometry.arc method), 9
getVertices() (pyrw.RWgeometry.circle method), 10
getVertices() (pyrw.RWgeometry.rectangle method), 11
getVoffset() (pyrw.RWgeometry.rectangle method), 11
getVstart() (pyrw.RWgeometry.arc method), 9
getWalkerById() (pyrw.RWRW.RW method), 7
getWalkerOfInterest() (pyrw.RWrun.run method), 3
getX0() (pyrw.RWwalker.walker method), 12
getXcenter() (pyrw.RWgeometry.arc method), 9
getXcenter() (pyrw.RWgeometry.circle method), 10
getXend() (pyrw.RWgeometry.arc method), 9
getXoffset() (pyrw.RWgeometry.rectangle method), 11
getXstart() (pyrw.RWgeometry.arc method), 9

H

HC (class in pyrw.RWHC), 3
hit() (pyrw.RWBC.setBack method), 8
hit() (pyrw.RWBC.sticky method), 8
hit() (pyrw.RWHC.none method), 3
hit() (pyrw.RWHC.stop method), 3
hitRW() (pyrw.RWHC.HC method), 3

I

inArc() (pyrw.RWgeometry.arc method), 9

L

line (class in pyrw.RWgeometry), 10
listToCSV() (in module pyrw.RWmisc), 6
loadFromPickle() (in module pyrw.RWmisc), 6

M

MRWstep (class in pyrw.RWstep), 4

N

none (class in pyrw.RWHC), 3
norm() (in module pyrw.RWessentials), 6

O

outwPerp() (in module pyrw.RWessentials), 6

P

performStep() (pyrw.RWstep.SCCRWstep method), 4
performStep() (pyrw.RWstep.step method), 4
perp() (in module pyrw.RWessentials), 6
perp() (pyrw.RWBC.BC method), 8
plotStep() (pyrw.RWrun.run method), 3
plotTraj() (pyrw.RWrun.run method), 3
plotTraj() (pyrw.RWwalker.walker method), 12
printStatistics() (pyrw.RWRW.RW method), 7
printVariable() (in module pyrw.RWmisc), 6
printVariable() (pyrw.RWRW.RW method), 7
pyrw.RWBC (module), 8
pyrw.RWdomain (module), 1
pyrw.RWessentials (module), 4
pyrw.RWgeometry (module), 8
pyrw.RWHC (module), 3
pyrw.RWmisc (module), 6
pyrw.RWrun (module), 3
pyrw.RWRW (module), 7
pyrw.RWstep (module), 3
pyrw.RWsuperposition (module), 7
pyrw.RWwalker (module), 12

R

rectangle (class in pyrw.RWgeometry), 11
run (class in pyrw.RWrun), 3
runAll() (pyrw.RWRW.RW method), 7

RW (class in pyrw.RWRW), 7

S

saveToFile() (pyrw.RWRW.RW method), 7
saveToPickle() (in module pyrw.RWmisc), 6
scalePlot() (pyrw.RWrun.run method), 3
scaleR() (pyrw.RWstep.CCRWstep method), 3
SCCRWstep (class in pyrw.RWstep), 4
segIntersect() (pyrw.RWBC.BC method), 8
setAllBCs() (pyrw.RWwalker.walker method), 12
setAngle() (pyrw.RWgeometry.arc method), 10
setBack (class in pyrw.RWBC), 8
setBackGamma() (pyrw.RWstep.SCCRWstep method), 4
setBC() (pyrw.RWwalker.walker method), 12
setCurrRun() (pyrw.RWwalker.walker method), 12
setDetectRadius() (pyrw.RWwalker.walker method), 12
setDomain() (pyrw.RWRW.RW method), 7
setGamma() (pyrw.RWstep.CCRWstep method), 3
setGamma() (pyrw.RWsuperposition.superposition method), 7
setGammaMin() (pyrw.RWstep.SCCRWstep method), 4
setGammaStep() (pyrw.RWstep.SCCRWstep method), 4
setGroupOfInterest() (pyrw.RWrun.run method), 3
setKappa() (pyrw.RWstep.CCRWstep method), 3
setKappa() (pyrw.RWsuperposition.superposition method), 7
setLenx() (pyrw.RWgeometry.rectangle method), 11
setLeny() (pyrw.RWgeometry.rectangle method), 11
setOrigGamma() (pyrw.RWstep.SCCRWstep method), 4
setParms() (pyrw.RWstep.CCRWstep method), 3
setR() (pyrw.RWsuperposition.superposition method), 7
setR1() (pyrw.RWstep.CCRWstep method), 3
setR2() (pyrw.RWstep.CCRWstep method), 4
setRadius() (pyrw.RWgeometry.arc method), 10
setRadius() (pyrw.RWgeometry.circle method), 10
setRW() (pyrw.RWdomain.domain method), 2
setToStart() (pyrw.RWwalker.walker method), 12
setVarForAllRuns() (pyrw.RWRW.RW method), 7
setVariable() (in module pyrw.RWmisc), 6
setVcenter() (pyrw.RWgeometry.circle method), 10
setWalkerOfInterest() (pyrw.RWrun.run method), 3
setX() (pyrw.RWgeometry.vertex method), 12
setX0() (pyrw.RWwalker.walker method), 12
setXcenter() (pyrw.RWgeometry.circle method), 10
setXoffset() (pyrw.RWgeometry.rectangle method), 11
start() (pyrw.RWrun.run method), 3
statsToCSV() (pyrw.RWRW.RW method), 7
step (class in pyrw.RWstep), 4
sticky (class in pyrw.RWBC), 8
stop (class in pyrw.RWHC), 3
stop() (pyrw.RWrun.run method), 3
superposition (class in pyrw.RWsuperposition), 7

T

toTraj() (pyrw.RWwalker.walker method), 12

U

unit_vector() (in module pyrw.RWessentials), 6
updateBCIds() (pyrw.RWwalker.walker method), 13
updateBCTyp() (pyrw.RWwalker.walker method), 13
updateGamma() (pyrw.RWstep.SCCRWstep method), 4
updateGammaDist() (pyrw.RWstep.step method), 4

V

vertex (class in pyrw.RWgeometry), 11
verticesCoordsToList() (pyrw.RWdomain.domain method), 2

W

walker (class in pyrw.RWwalker), 12