

# Oaktree Manual



April 22, 2012

# Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Installation</b>	<b>3</b>
<b>3</b>	<b>Running</b>	<b>4</b>
<b>4</b>	<b>Input</b>	<b>5</b>
4.1	SIMULATION . . . . .	5
4.2	SPHERE . . . . .	5
4.3	CUBE . . . . .	6
4.4	UNION . . . . .	6
4.5	INTERSECTION . . . . .	6
4.6	DIFFERENCE . . . . .	6
4.7	MOVE . . . . .	7
4.8	ROTATE . . . . .	7
4.9	SOLID . . . . .	7
<b>5</b>	<b>Output</b>	<b>8</b>
<b>6</b>	<b>Viewer</b>	<b>9</b>
<b>7</b>	<b>Tutorials</b>	<b>10</b>
<b>8</b>	<b>Theory</b>	<b>11</b>

# Chapter 1

## Introduction

## Chapter 2

# Installation

## Chapter 3

# Running

# Chapter 4

## Input

Oaktree input language extends Python. Subroutines and objects related to input processing are listed below.

### 4.1 SIMULATION

SIMULATION object stores data specific to one distinct analysis.

**obj = SIMULATION (outpath, duration, step, grid, cutoff, extents)**

- **obj** - SIMULATION object
- **outpath** - output directory path
- **duration** - simulation duration
- **step** - time step
- **grid** - regular size of octree grid
- **cutoff** - cutoff length below which geometrical details are not resolved
- **extents** - tuple  $(x_{min}, y_{min}, z_{min}, x_{max}, y_{max}, z_{max})$  of domain extents beyond which bodies are deleted

### 4.2 SPHERE

A sphere shape.

**obj = SPHERE (center, r, vcolor, scolor)**

- **obj** - SHAPE object
- **center** - tuple  $(x, y, z)$  defining the center
- **r** - radius
- **vcolor** - integer volume color
- **scolor** - integer surface color

### 4.3 CUBE

A cube shape.

**obj** = CUBE (**corner**, **u**, **v**, **w**, **vcolor**, **scolor**)

- **obj** - SHAPE object
- **corner** - tuple  $(x, y, z)$  defining the minimum coordinate corner
- **u** - length along  $x$
- **v** - length along  $y$
- **w** - length along  $z$
- **vcolor** - integer volume color
- **scolor** - integer tuple  $(s_{xmin}, s_{ymax}, s_{zmin}, s_{xmax}, s_{ymax}, s_{zmax})$  of surface colors

### 4.4 UNION

Set theoretic union of two shapes.

**obj** = UNION (**shape1**, **shape2**)

- **obj** - SHAPE object
- **shape1** - first input SHAPE object
- **shape2** - second input SHAPE object

### 4.5 INTERSECTION

Set theoretic intersection of two shapes.

**obj** = INTERSECTION (**shape1**, **shape2**)

- **obj** - SHAPE object
- **shape1** - first input SHAPE object
- **shape2** - second input SHAPE object

### 4.6 DIFFERENCE

Set theoretic difference of two shapes.

**obj** = DIFFERENCE (**shape1**, **shape2**)

- **obj** - SHAPE object
- **shape1** - first input SHAPE object
- **shape2** - second input SHAPE object

## 4.7 MOVE

Move shape linearly.

### MOVE (shape, vector)

- **shape** - input SHAPE object
- **vector** - tuple  $(u, v, w)$  defining the translation

## 4.8 ROTATE

Rotate shape about an axis.

### MOVE (shape, point, vector, angle)

- **shape** - input SHAPE object
- **point** - tuple  $(x, y, z)$  defining axis point
- **vector** - tuple  $(u, v, w)$  defining axis direction
- **angle** - oriented angle in degrees

## 4.9 SOLID

A solid is created in a simulation.

### obj = SOLID (simu, shape, label)

- **obj** - SOLID object
- **simu** - simulation in which the solid is created
- **shape** - solid shape
- **label** - solid label



## Chapter 5

# Output

## Chapter 6

# Viewer

## Chapter 7

# Tutorials

## Chapter 8

# Theory