




1

Slide 2

Preliminaries

1. Make sure you have the latest update installed



Note that *PFC* will warn you when a new update is available (internet connection required). You can then visit our website to get the update file and find more information on the released features.

ITASCA[®] BasisSoft

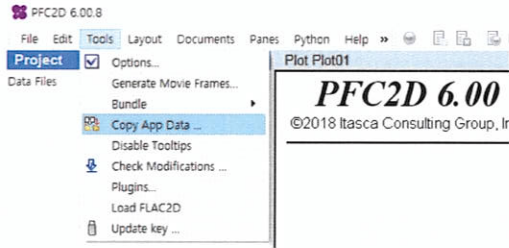
PFC 6.0 Short Course

2

Slide 3

Preliminaries

2. Create a local copy of the application data



PFC2D 6.00
©2018 Itasca Consulting Group, Inc.

ITASCA BasisSoft

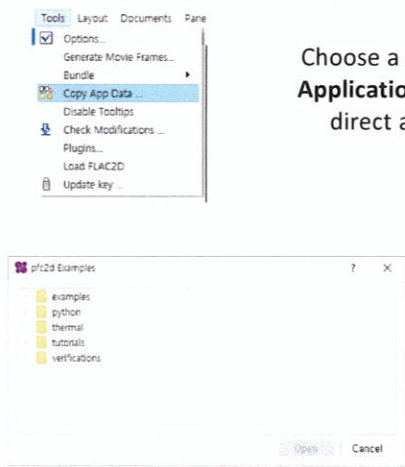
PFC 6.0 Short Course

3

Slide 4

PFC Code Overview - The hands on the software

Tools -> Copy App Data



Choose a local (modifiable) folder to copy
Application Data : examples, tutorials with
direct access from *PFC* user interface

All changes to Application Data
are **allowed** and **permanent**!

However, the default version of
tutorials and examples can be
recovered by repeating the
Copy App Data operation

ITASCA BasisSoft

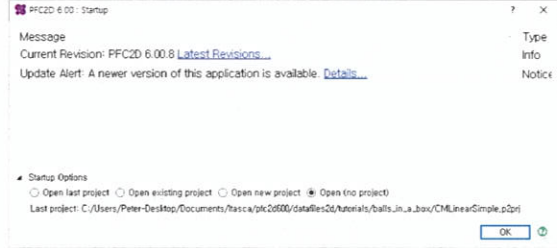
PFC 6.0 Short Course

4

Slide 5



PFC Code Overview - Hands on the software

The project file



A **project** file (*.p2prj) records and makes a link between all the components of a project. These components are:

- Data files (*.p2dat, *.p3dat)
- Save files (*.p2sav, *.p3sav)
- Plot views

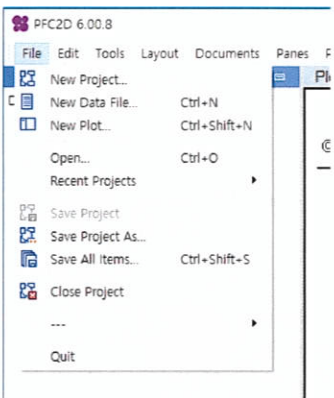
PFC 6.0 Short Course



5

Slide 6

PFC Code Overview - The hands on the software

The project file



PFC 6.0 Short Course

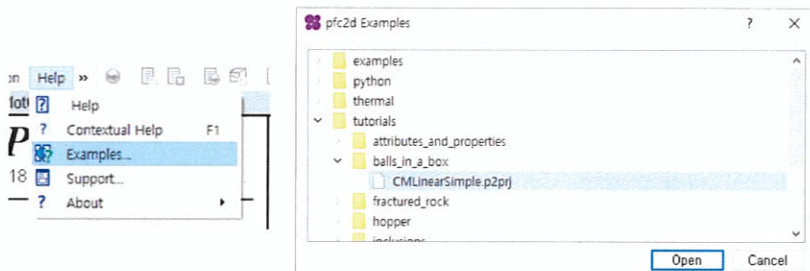
6



Slide 7

Exercise the user interface

Exercise

- Launch **PFC2D**
- If the startup dialog shows up, close it (**cancel**)
- Open the tutorial example: « Balls in a Box »



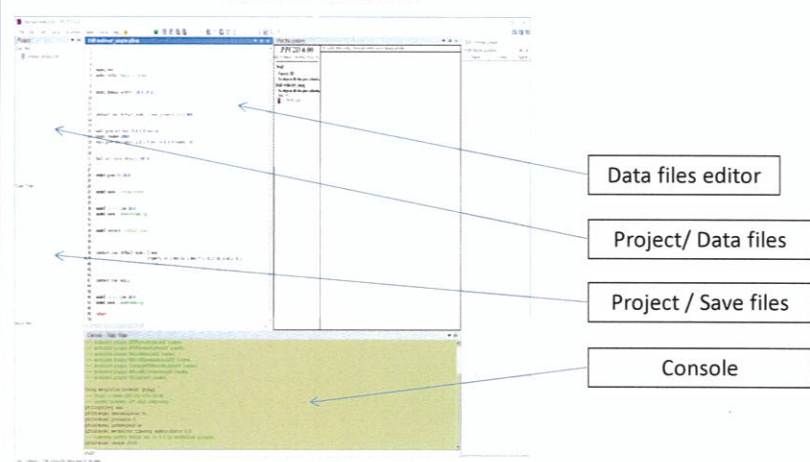
PFC 6.0 Short Course

7

Slide 8

PFC Code Overview - The hands on the software

The user interface





Data files editor

Project/ Data files

Project / Save files

Console

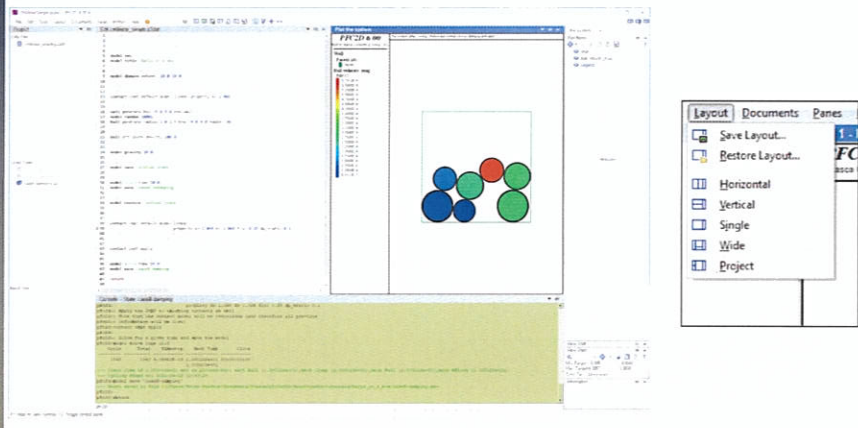
PFC 6.0 Short Course



8

Slide 9

PFC Code Overview - The hands on the software

Layout options



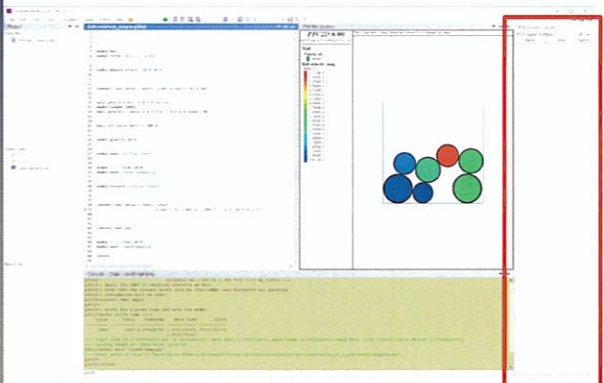
PFC 6.0 Short Course

9

Slide 10



PFC Code Overview - The hands on the software

Control panel



Control panel / 1

When the editor window is active, the control panel displays the current value of global variables

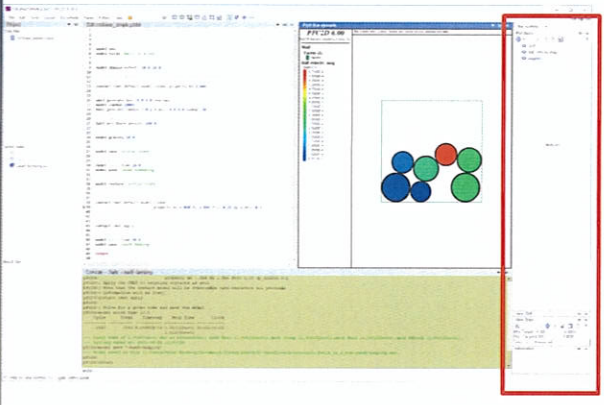
PFC 6.0 Short Course

10

Slide 11



PFC Code Overview - The hands on the software

Control panel



Control panel / 2

When the plot window is active, the control panel displays the **plot options**


PFC 6.0 Short Course

11



Slide 12

PFC Code Overview - The hands on the software

The data files



- **Data files** contains the necessary commands to build the model and introduce all model elements.
- **Data files** are written in **FISH** language
- **Data files** can be edited using the PFC editor, or any other text editor

PFC 6.0 Short Course

12

Slide 13

PFC Code Overview - The hands on the software

The data files – The PFC editor

Strings in green

Keywords and functions in bold

Comments in gray

Values are in blue

ITASCA

BasisSoft

PFC 6.0 Short Course

13

Slide 14

PFC Code Overview - The hands on the software

The data files – The PFC editor

Options

General License Fish Email Debug Startup Editor Model Pane Plots Console Listings Results

Font: Consolas Size: 10 Error Indicator

Syntax Type: Command Word Background: Underline

Tab size: 4

☒ Show line numbers
☒ Enable code folding for data files
☒ Show indentation guides
☒ Show syntax highlighting for data files
☒ Enable word wrapping
☐ Cut or copy line at cursor
 Menu Command: Find
 Show of Key: [C] Control [A] Alt [S] Shift [F]
☒ In the Replace dialog, suggest replacement of backward slashes with forward slashes (for file paths)
☒ Ask whether to do command conversion when opening a file

OK Cancel Apply

YOU CAN CUSTOMIZE IT !

ITASCA

BasisSoft

PFC 6.0 Short Course

14

Slide 15

PFC Code Overview - The hands on the software

PLOTS

Plot elements

Plot elements options

Main view

View controller

Legend

ITASCA

BasisSoft

PFC 6.0 Short Course

15

Slide 16

Exercise the user interface

Exercise (continued)

- Create a new Plot view
- Add a Ball plot item – color balls by position
- Show the contact force vectors

Full list of available plot items

quick access to select plot items (user modifiable)

Plot Items

- Wall
- Ball velocity_mag
- Legend

ITASCA

BasisSoft

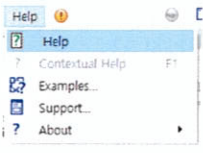
PFC 6.0 Short Course

16



Slide 17

PFC Code Overview - The hands on the software

HELP



- A **powerful tool** for many needs (description of the user interface, commands description, DEM formulation, contact laws, plot options)
- Many ways to access it:
 - Main menu -> help
 - From command prompt -> press **F1** to access command description)
 - From any line in the editor pane -> press **CTRL+SPACE** for inline help, and **F1** to access to FISH/Command reference page


PFC 6.0 Short Course



17

Slide 18

PFC Code Overview - The hands on the software

HELP



PFC 6.0 Short Course

18



Slide 19

PFC Code Overview - The hands on the software

HELP

- Program Guide
- PFC
 - Numerical Simulations with PFC
 - PFC Overview
 - PFC Model Formulation
 - How to ...
 - Release Notes
 - PFC Model Objects
 - Additional Features
 - PFC Commands Index
 - PFC FISH Index
 - FLAC3D
 - Coupling PFC and FLAC3D
 - Scripting
 - Examples Index
 - Command Index
 - FISH Index
 - Glossary

- **PFC**
 - Fundamental informations for the new user
 - Tutorial examples to get started using *PFC* and get familiar with its main elements

PFC 6.0 Short Course

19



Slide 20

PFC Code Overview - The hands on the software

HELP

- Program Guide
- PFC
- FLAC3D
- Coupling PFC and FLAC3D
- Scripting
- Examples Index
 - PFC Examples
 - Tutorials
 - Verification Problems
 - Example Applications
 - FLAC3D Examples
 - PFC-FLAC3D Coupling Examples
 - Command Index
 - FISH Index
 - Glossary

- **Example and Verification Problems**
 - **Verification Problems** : aim to demonstrate the consistency of *PFC* tools
 - **Example Applications** : aim to show the main/classic applications of *PFC*, and to get familiar with its simple/complex components

PFC 6.0 Short Course

20


Slide 21

PFC Code Overview - The hands on the software


HELP

- Program Guide
- ▶ PFC
 - Numerical Simulations with PFC
 - ▶ PFC Model Objects
 - Balls
 - Clumps
 - Rigid Blocks
 - Walls
 - Contacts and Contact Models
 - Additional Features
 - PFC Commands Index
 - PFC FISH Index
- FLAC3D
- Coupling PFC and FLAC3D
- Scripting
- Examples Index
- Command Index
- FISH Index
- Glossary

- **PFC Model Objects**
 - Description of all commands related to each element that compose a *PFC* model (**balls, clumps, walls**)
 - Description of all commands related to the definition of the mechanical interaction between the elements (**contacts and contact models**)



21



PFC 6.0 Short Course

21


Slide 22

PFC Code Overview - The hands on the software


HELP

- Program Guide
- PFC
- FLAC3D
- Coupling PFC and FLAC3D
- ▶ Scripting
 - ▶ FISH Scripting
 - FISH Rules and Usage
 - Lines
 - Data Types
 - Reserved Names for Functions and Variables
 - Scope of Variables
 - Functions: Structure, Evaluation, and Calling Scheme
 - Arithmetic Expressions and Type Conversions
 - Redefining FISH Functions
 - Execution of FISH Functions
 - Inline FISH or FISH Fragments
 - FISH Error Handling
 - FISH Callback Events
 - FISH Debugging
 - FISH Statements Index
 - FISH Functions
 - FISH Type Index
 - Python Scripting
- Examples Index
- Command Index
- FISH Index
- Glossary

- **FISH Scripting**
 - An indispensable guide to help the user with **FISH** programming language
 - Rules and Usage (fundamental aspects)
 - Statements (loop syntax, local/global variables)
 - Functions (FISH language intrinsics)
 - Type Index (useful for a full comprehension of documentation)



22



PFC 6.0 Short Course

22



Slide 23

PFC Code Overview - The hands on the software

HELP

- Program Guide
- ▲ PFC
 - Numerical Simulations with PFC
 - PFC Model Objects
 - ▲ Additional Features
 - Thermal Calculation
 - CFD module for PFC3D6.0
 - C++ Plugins
 - PFC Commands Index
 - PFC FISH Index
- FLAC3D
- Coupling PFC and FLAC3D
- Scripting
- Examples Index
- Command Index
- FISH Index
- Glossary

- **Additional features**
 - Thermal option: coupled thermo-mechanical simulations
 - CCFD option: coupled fluid-solid simulations
 - C++ plugins: description of the possibility to create customized FISH intrinsics and implement user-defined contact laws

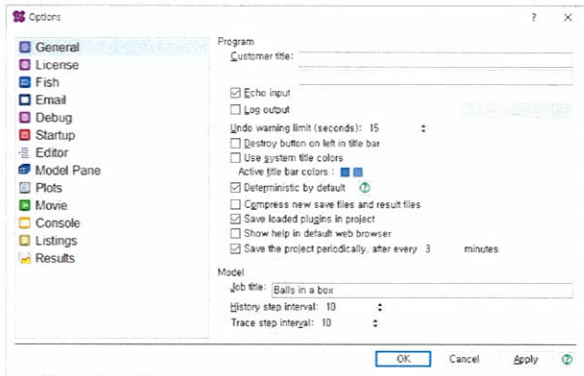
PFC 6.0 Short Course



23

Slide 24

PFC Code Overview - The hands on the software

PFC OPTIONS



PFC 6.0 Short Course

24

Slide 25

Exercise the main commands

ITASCA

BasisSoft

PFC 6.0 Short Course

25

Slide 26

Exercise the main commands

Exercise

- Create a new project and name it **test.p2prj**
- Add a new data file and name it **ball.p2dat**
- Let's model a single ball in a box

Model new

Clear program-state information. Most information is cleared, allowing one to begin a new problem.

How to create a ball?

- Press **F1** for context-sensitive help
- Press **Ctrl+Space** for inline help

Edit ball.p2dat

```
1 model new
2
3 ball create
```



Execute!

(CTRL + E to execute the whole file)
(CTRL + Shift + E to execute selected lines)

ITASCA

BasisSoft

PFC 6.0 Short Course

26

Slide 27

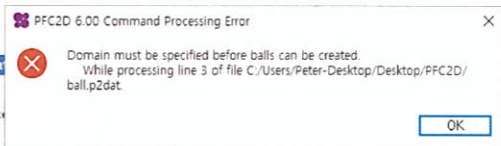
Exercise the main commands

Exercise

- Create a new project and name it **test.p3prj**
- Add a new data file and name it **ball.p3dat**
- Let's model a single ball in a box

new



Clear program-state information. Most information is cleared, allowing one to begin a new problem.



Edit ball.p2dat

```
1 model new
2
3 ball create
```

e!
the whole file)
ute selected lines)

PFC 6.0 Short Course

27

Slide 28


Exercise the main commands

Model domain



All PFC model components live inside the domain, an axis-aligned box that does not change size during the simulation. No model components can be created until the domain is specified. When model components hit the domain boundary, various mechanical responses may occur. The default condition is the **stop** condition. A domain is required for efficient contact detection.

Edit ball.p2dat

```
1 model new
2
3 model domain extent -1 1
4 model domain condition periodic
5
6 ball create
```



Execute!

PFC 6.0 Short Course

28

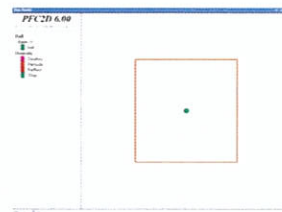
Slide 29

Exercise the main commands

Model domain

All PFC model components live inside the domain, an axis-aligned box that does not change size during the simulation. No model components can be created until the domain is specified. When model components hit the domain boundary, various mechanical responses may occur. The default condition is the **stop** condition. A domain is required for efficient contact detection.

```
Edit ball.p2dat
1 model new
2
3 model domain extent -1 1
4 model domain condition periodic
5
6 ball create
```



The logo for ITASCA, consisting of the word 'ITASCA' in a bold, sans-serif font.

The logo for BasisSoft, featuring a stylized 'B' with a rainbow gradient followed by the word 'BasisSoft'.

PFC 6.0 Short Course

29

Slide 30

Exercise the main commands

Model cycle / model solve

PFC uses a fully explicit time-marching solution scheme. The **cycle (step)** command may be used to perform a given number of steps; or a more complex target criterion can be specified with the **solve** command.

```
Edit ball.p2dat
1 model new
2
3 model domain extent -1 1
4 model domain condition periodic
5
6 ball create
7
8 model cycle 1000
```

Execute!

The logo for ITASCA, consisting of the word 'ITASCA' in a bold, sans-serif font.

The logo for BasisSoft, featuring a stylized 'B' with a rainbow gradient followed by the word 'BasisSoft'.

PFC 6.0 Short Course

30

Slide 31

Exercise the main commands

cycle / solve

PFC uses a fully explicit time-marching solution scheme. The **cycle (step)** command may be used to perform a given number of steps; or a more complex target criterion can be specified with the **solve** command.

Edit ball.p2dat

```

1 model new
2
3 model domain extent -1 1
4 model domain condition periodic
5
6 ball create
7
8 model cycle 1000
          
```

PFC2D 6.00 Command Processing Error

Ball 1 has zero inertial mass.
While processing line 8 of file C:/Users/Peter/Desktop/Desktop/PFC2D/ball.p2dat.

PFC 6.0 Short Course

31

Slide 32

Exercise the main commands

ball attribute ...

All PFC model components have a finite set of *attributes* (e.g. position, velocity, density...).

The *density* attribute must be specified for PFC to be able to solve equations of motion.

Edit ball.p2dat

```

1 model new
2
3 model domain extent -1 1
4 model domain condition periodic
5
6 ball create
7 ball attribute density 1.0
8
9 model cycle 1000
          
```

Execute!

PFC 6.0 Short Course

32

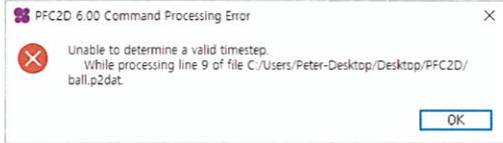
Slide 33

Exercise the main commands

ball attribute ...



All PFC model components have a finite set of *attributes* (e.g. position, velocity, density...).

The *density* attribute must be specified for PFC to be able to solve equations of motion.



```

Edit ball.p3dat
1 new
2
3 domain extent -1 1
4 domain condition periodic
5
6 ball create
7 ball attribute density 1.0
8
9 cycle 1000
10
11 return
  
```

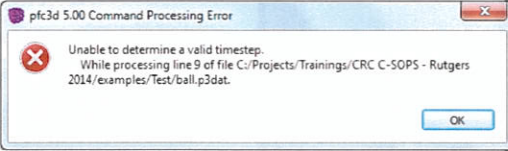
PFC 6.0 Short Course

33

Slide 34



Exercise the main commands

The solution procedure requires a finite timestep to operate. By default, *PFC* will attempt to automatically compute one. *PFC* was not able to compute a valid timestep in this case (no time scale has been defined).



```

Edit ball.p3dat
1 new
2
3 domain extent -1 1
4 domain condition periodic
5
6 ball create
7 ball attribute density 1.0
8
9 cycle 1000
10
11 return
  
```

PFC 6.0 Short Course

34

Slide 35

Exercise the main commands

Model gravity

The **model** command is used to control global parameters and computation modes (such as gravity, timestep calculation...).

Edit ball.p2dat*

```
1 model new
2
3 model domain extent -1 1
4 model domain condition periodic
5
6 ball create
7 ball attribute density 2000.0
8
9 model gravity 0 0 -9.81
10
11 model cycle 1000
```

Execute!

ITASCA

BasisSoft

PFC 6.0 Short Course

35

Slide 36

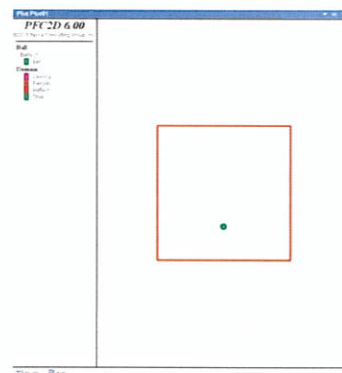
Exercise the main commands

Model gravity

The **model** command is used to control global parameters and computation modes (such as gravity, timestep calculation...).

Edit ball.p2dat*

```
1 model new
2
3 model domain extent -1 1
4 model domain condition periodic
5
6 ball create
7 ball attribute density 2000.0
8
9 model gravity 0 0 -9.81
10
11 model cycle 1000
```



ITASCA

BasisSoft

PFC 6.0 Short Course

36

Slide 37

Exercise the main commands

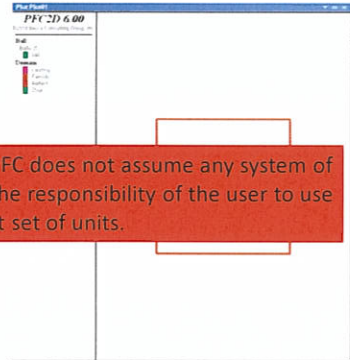
Model gravity



The **model** command is used to control global parameters and computation modes (such as gravity, timestep calculation...).

```

Edit ball.p2dat
1 model new
2
3 model domain extent -1 1
4 model domain condition periodic
5
6 ball create
7 ball attribute density 2000.0
8
9 model gravity 0 -9.81
10
11 model cycle 1000
        
```

Note that PFC does not assume any system of units. It is the responsibility of the user to use a consistent set of units.



PFC 6.0 Short Course

37

Slide 38


Exercise the main commands

Histories can be used to monitor the evolution of the system (PFC samples select data at a specified interval).



Particle traces are also available (PFC samples the position and velocity of the ball at a specified interval)

```

Edit ball.p2dat
1 model new
2
3 model domain extent -1 1
4 model domain condition periodic
5
6 ball create
7 ball attribute density 2000.0
8 ball attribute velocity -1.0 0.0
9 ball attribute velocity-x 0.5
10
11 model gravity 0 -9.81
12
13 ball history position-y id 1
14 ball trace id 1
15 model cycle 1000
16
17 return
        
```



Execute!

PFC 6.0 Short Course

38

Slide 39

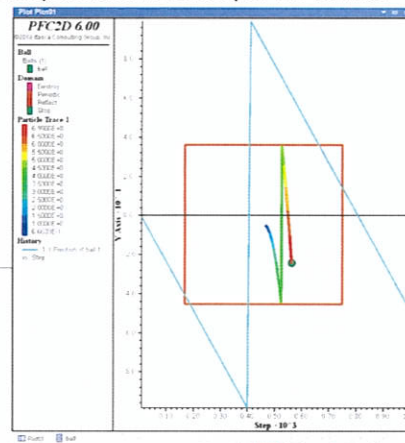
Exercise the main commands

Histories can be used to monitor the evolution of the system (PFC samples select data at a specified interval).

Particle traces are also available (PFC samples the position and velocity of the ball at a specified interval)

Edit ball.p2dat

```
1 model new
2
3 model domain extent -1 1
4 model domain condition periodic
5
6 ball create
7 ball attribute density 2000.0
8 ball attribute velocity -1.0 0.0
9 ball attribute velocity-x 0.5
10
11 model gravity 0 -9.81
12
13 ball history position-y id 1
14 ball trace id 1
15 model cycle 1000
16
17 return
```



ITASCA™

BasisSoft

PFC 6.0 Short Course

39

Slide 40

Exercise the main commands

wall

A wall is a manifold surface composed of {line segments in 2D; triangular facets in 3D} termed facets. The surface is defined by a mesh. The surface properties of a wall can be specified independently for each facet.

Edit ball.p3dat

```
1 new
2
3 domain extent -1 1
4 domain condition periodic
5
6 ball create
7 ball attribute density 2000.0
8 ball attribute velocity -1.0 0.0 0.0
9 ball attribute yvelocity 0.5
10
11 wall generate box -0.5 0.5
12
13 set gravity 0 0 -9.81
14
15 ball history zposition id 1
16 ball trace id 1
17 cycle 1000
18
19 return
```

Execute!

ITASCA™

BasisSoft

PFC 6.0 Short Course

40

Slide 41

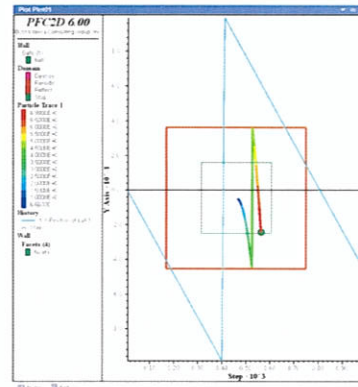
Exercise the main commands

wall

A wall is a manifold surface composed of {line segments in 2D; triangular facets in 3D} termed facets. The surface is defined by a mesh. The surface properties of a wall can be specified independently for each facet.

```

1 new
2
3 domain extent -1 1
4 domain condition periodic
5
6 ball create
7 ball attribute density 2000.0
8 ball attribute velocity -1.0 0.0 0.0
9 ball attribute yvelocity 0.5
10
11 wall generate box -0.5 0.5
12
13 set gravity 0 0 -9.81
14
15 ball history zposition id 1
16 ball trace id 1
17 cycle 1000
18
19 return
  
```



ITASCA

BasisSoft

PFC 6.0 Short Course

41

Slide 42

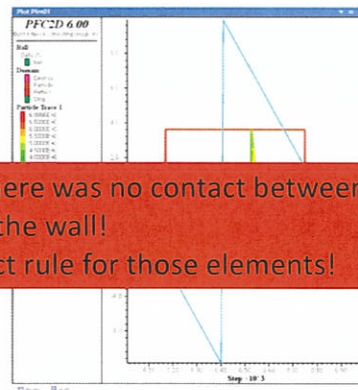
Exercise the main commands

wall

A wall is a manifold surface composed of {line segments in 2D; triangular facets in 3D} termed facets. The surface is defined by a mesh. The surface properties of a wall can be specified independently for each facet.

```

1 new
2
3 domain extent -1 1
4 domain condition periodic
5
6 ball create
7 ball attribute density 2000.0
8 ball attribute velocity -1.0 0.0 0.0
  
```



The system behaves as though there was no contact between the ball and the wall!

We have to define a contact rule for those elements!

```

16 ball trace id 1
17 cycle 1000
18
19 return
  
```

ITASCA

BasisSoft

PFC 6.0 Short Course

42

Slide 43

Exercise the main commands

Contact cmat

The **Contact Model Assignment Table (CMAT)** controls the assignment of contact models and their associated properties to newly created contacts, and also provides the detection distances used by the contact-creation procedure. The CMAT consists of an ordered set of optional slots (populated with **cmat add** command) along with a default slot for each contact type (set with the **cmat default** command).

contact **cmat** default model linear property kn **1.0e6**

ITASCA

BasisSoft

PFC 6.0 Short Course

43

Slide 44

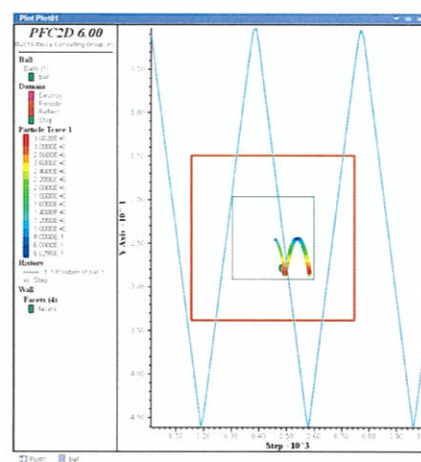
Exercise the main commands

Edit ball.p2dat

```

1 model new
2
3 model domain extent -1 1
4 model domain condition periodic
5
6 contact cmat default model linear ...
7           property kn 1.0e6
8
9 ball create
10 ball attribute density 2000.0
11 ball attribute velocity -1.0 0.0
12 ball attribute velocity-x 0.5
13
14 wall generate box -0.5 0.5
15
16 model gravity 0 -9.81
17
18 ball history position-y id 1
19 ball trace id 1
20 model cycle 1000
21
22 return

```



ITASCA

BasisSoft

PFC 6.0 Short Course

44

Slide 45

Exercise the main commands

```

1 model new
2
3 model domain extent -1 1
4 model domain condition periodic
5
6 contact cmat default model linear ...
7           property kn 1.0e6
8
9 ball create
10 ball attribute density 2000.0
11 ball attribute velocity -1.0 0.0
12 ball attribute velocity-x 0.5
13
14 wall generate box -0.5 0.5
15
16 model gravity 0 -9.81
17
18 ball history position-y id 1
19 ball trace id 1
20 model cycle 1000
21
22 return

```

line continuation

ITASCA BasisSoft

PFC 6.0 Short Course

45

Slide 46

Exercise the main commands

```

1 model new
2
3 model domain extent -1 1
4 model domain condition periodic
5
6 contact cmat default model linear ...
7           property kn 1.0e6
8
9
10 ball create
11 ball attribute density 2000.0
12 ball attribute velocity -1.0 0.0
13 ball attribute velocity-x 0.5
14
15 generate a box wall
16 wall generate box -0.5 0.5
17
18 model gravity 0 -9.81
19
20 ball history position-y id 1
21 ball trace id 1
22 model cycle 1000
23
24 return

```

line continuation

comment

ITASCA BasisSoft

PFC 6.0 Short Course

46

Slide 47

Exercise the main commands

PFC does not damp the equations of motion by default. Some damping mechanism must be specified by the user.

- Damping may occur at contacts, depending on the specifics of the contact model(friction, stiffness hysteresis, viscous damping,...)
- For compact assemblies, **local damping** may be used to converge more rapidly to equilibrium or for quasi-static simulations

```

Edit ball.p2dat
1 model new
2
3 model domain extent -1 1
4 model domain condition periodic
5
6 contact cmat default model linear ...
7 property kn 1.0e6 ...
8 dp_ratio 0.5
9
10 ball create
11 ball attribute density 2000.0
12 ball attribute velocity -1.0 0.0
13 ball attribute velocity-x 0.5
14
15
16 wall generate box -0.5 0.5
17
18 model gravity @ -9.81
19
20 ball history position-y id 1
21 ball trace id 1
22 model cycle 1000
23
24 return
  
```

Execute!

ITASCA

BasisSoft

PFC 6.0 Short Course

47

Slide 48

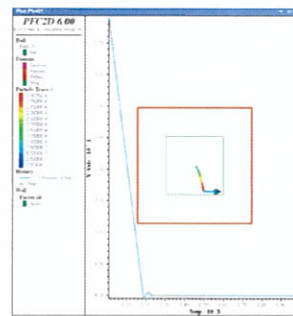
Exercise the main commands

PFC does not damp the equations of motion by default. Some damping mechanism must be specified by the user.

- Damping may occur at contacts, depending on the specifics of the contact model(friction, stiffness hysteresis, viscous damping,...)
- For compact assemblies, **local damping** may be used to converge more rapidly to equilibrium or for quasi-static simulations

```

Edit ball.p2dat
1 model new
2
3 model domain extent -1 1
4 model domain condition periodic
5
6 contact cmat default model linear ...
7 property kn 1.0e6 ...
8 dp_ratio 0.5
9
10 ball create
11 ball attribute density 2000.0
12 ball attribute velocity -1.0 0.0
13 ball attribute velocity-x 0.5
14
15
16 wall generate box -0.5 0.5
17
18 model gravity @ -9.81
19
20 ball history position-y id 1
21 ball trace id 1
22 model cycle 1000
23
24 return
  
```



ITASCA

BasisSoft

PFC 6.0 Short Course

48

Slide 49

Exercise the main commands

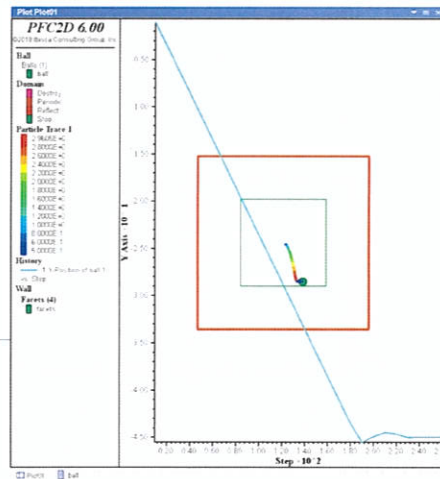
The model state can be saved and restored at any time.

Edit ball.p2dat

```

1 model new
2
3 model domain extent -1 1
4 model domain condition periodic
5
6 contact cmat default model linear ...
7
8 ball create
9
10 ball attribute density 2000.0
11 ball attribute velocity -1.0 0.0
12 ball attribute velocity-x 0.5
13
14
15 wall generate box -0.5 0.5
16
17 model gravity 0 -9.81
18
19 ball history position-y id 1
20 ball trace id 1
21
22
23 model save 'initial'
24 model cycle 1000
25 model save 'after-cycle'
26
27 model restore 'initial'
28 model solve
29 model save 'after-solve'
30
31 return

```



ITASCA

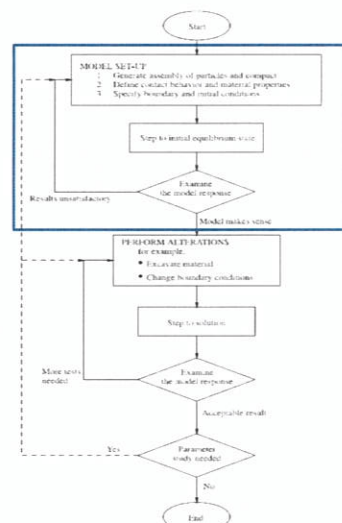
BasisSoft

PFC 6.0 Short Course

49

Slide 50

PROBLEM SOLVING WITH PFC



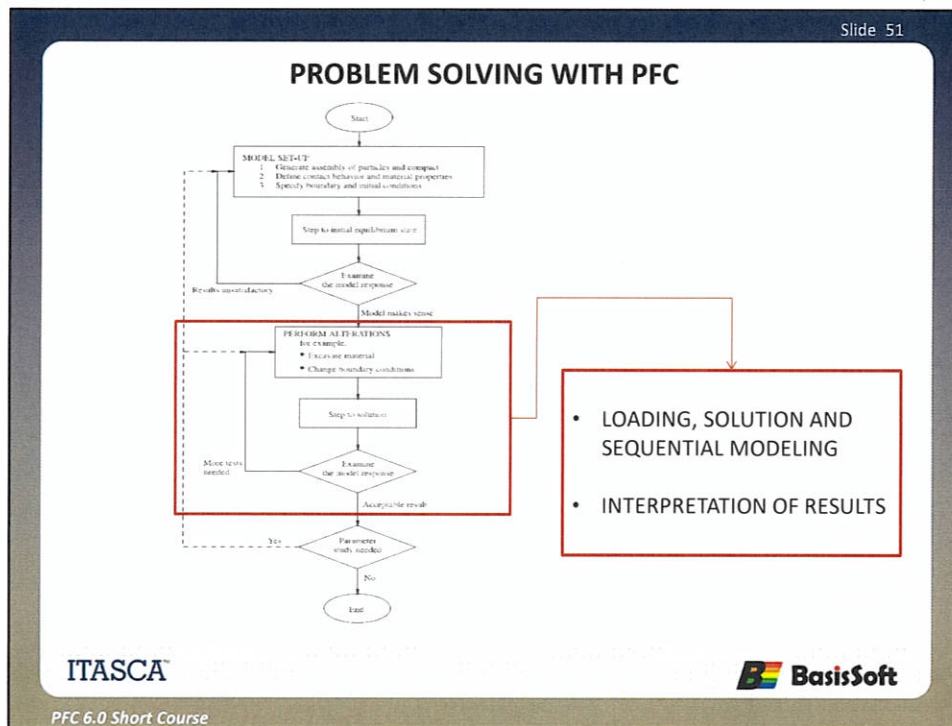
- PARTICLE GENERATION
- BOUNDARY AND INITIAL CONDITIONS
- CHOICE OF CONTACT MODEL (definition of the CMAT)
- DEFINITION OF MATERIAL PROPERTIES

ITASCA

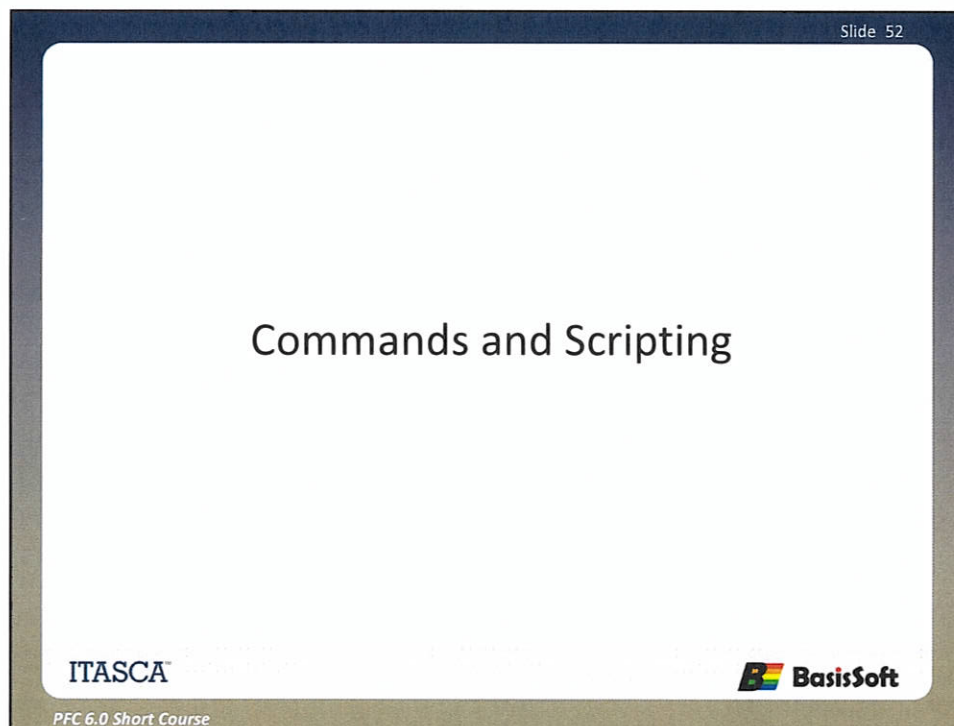
BasisSoft

PFC 6.0 Short Course

50



51





52

Slide 53

Important and widely used commands

Model domain	specify domain extent and boundary conditions
ball	create and manipulate balls
wall	create and manipulate wall
Model cmat	define contact model assignment



PFC 6.0 Short Course

53

Slide 54

Important and widely used commands

Model domain	specify domain extent and boundary conditions
ball	create and manipulate balls
wall	create and manipulate wall
Model cmat	define contact model assignment
clump	create and manipulate clumps
Model cycle	perform simulation steps
Model solve	
Model list	list information

PFC 6.0 Short Course

54

Slide 55

Important and widely used commands

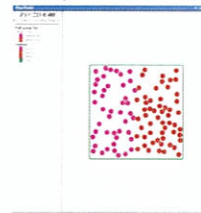
Range

A PFC command that *does not* use a [range](#) will operate on all possible target objects of the given command (balls, contacts, clumps, etc.). The [range](#) statement, when applied to a command, will provide the means to restrict the operation of the command to a subset of the target objects. The subset is defined by *range elements*, which are keywords following the [range](#) statement.

```

1 model new
2
3 model domain extent -10.0 10.0
4 model random 10001
5 ball generate id 1 100 box -10.0 10.0 -10.0 10.0
6
7 [half_model_x = 0.5*(domain.max.x + domain.min.x)]
8
9
10 ball group 'left' range position-x [domain.min.x] @half_model_x
11 ball group 'right' range position-x @half_model_x [domain.max.x]
12
13

```



ITASCA

BasisSoft

PFC 6.0 Short Course

55

Slide 56

Important and widely used commands

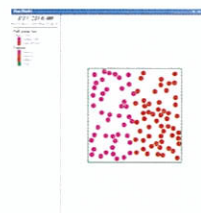
Group

Each model object provides a group keyword (e.g., [ball group](#) or [wall thermal group](#)) that may be used to create groups of that object type. The group logic in PFC is supplied, therefore, nearly exclusively through use of the *group keyword*.

```

1 model new
2
3 model domain extent -10.0 10.0
4 model random 10001
5 ball generate id 1 100 box -10.0 10.0 -10.0 10.0
6
7 [half_model_x = 0.5*(domain.max.x + domain.min.x)]
8
9
10 ball group 'left' range position-x [domain.min.x] @half_model_x
11 ball group 'right' range position-x @half_model_x [domain.max.x]
12
13

```



ITASCA

BasisSoft

PFC 6.0 Short Course

56

Slide 57

Important and widely used commands

Range

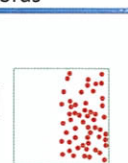
A PFC command that *does not* use a [range](#) will operate on all possible target objects of the given command (balls, contacts, clumps, etc.). The [range](#), when applied to a command, will provide the means to restrict the operation of the command to a subset of target objects. The subset is defined by *range elements*, which are keywords following the [range](#) keyword.

```

1 model new
2
3 model domain extent -10.0 10.0
4 model random 10001
5 ball generate id 1 100 box -10.0 10.0 -10.0
6
7 [half_model_x = 0.5*(domain.max.x + domain.min.x)]
8
9
10 ball group 'left' range position-x [domain.min.x] @half_model_x
11 ball group 'right' range position-x @half_model_x [domain.max.x]
12
13

```

ball delete range group 'left'



ITASCA

BasisSoft

PFC 6.0 Short Course

57

Slide 58

Important and widely used commands

History

The values of a set of variables can be sampled and stored during a model run by using the history command. These variables can then be plotted versus step number or versus other histories (in a View pane in the user interface or with the [plot history](#) command). Histories can also be written to a file (with the write keyword). Only one variable may be given per history command. History variables may be added at any time. The contents of all histories can be erased with the purge keyword, and all histories can be deleted with the delete keyword. A summary of all histories is printed by the [list history](#) command.

ITASCA

BasisSoft

PFC 6.0 Short Course

58

Slide 59

Generating Complex Assemblies

ITASCA

BasisSoft

PFC 6.0 Short Course

59

Slide 60

Wall commands

wall active-sides	Specify which side(s) of facets are
wall addfacet	Add a facet a wall.
wall attribute	Set the value of wall attributes.
wall create	Create a wall from vertices.
wall delete	Delete walls and/or facets.
wall export	Exports walls.
wall extra	Set wall or facet extra variables.
wall generate	Generate walls with specified shape.
wall group	Specify wall or facet group names.
wall history	Adds a history of a wall value.
wall import	Import a wall.
wall initialize	Modify wall attributes.
wall property	Assign facet surface properties.
wall resolution	Modify the contact-resolution strategy.
wall results	Modify the usage of the wall result.
wall rotate	Rotate walls.
wall servo	The wall servo provides the ability
wall tolerance	Set contact detection tolerances.
wall velocity-conveyor	Assign a rotational conveyor velocity.

ITASCA

BasisSoft

PFC 6.0 Short Course

60

Slide 61

Exercise

- Create a new project **WallCreate.p2prj**
- Add new data file wallcreate.p2dat
- Create the walls below

Diagram 1: A blue polygon with vertices at (0.0,0.0), (5.0,0.0), (4.0,2.0), (5.0,4.0), and (0.0,4.0).

Diagram 2: A blue polygon with vertices at (0.0,0.0), (5.0,0.0), (6.0,2.0), (5.0,4.0), and (0.0,4.0).

ITASCA
 BasisSoft

PFC 6.0 Short Course

61

Slide 62

Exercise (continued)

- Create a new project **WallCreate.p2prj**
- Add new data file wallcreate.p2dat
- Create the walls below

Diagram 1: A blue polygon with vertices at (0.0,0.0), (5.0,0.0), (4.0,2.0), (5.0,4.0), and (0.0,4.0). A diagonal line is drawn from (0.0,4.0) to (4.0,2.0).

Diagram 2: A blue polygon with vertices at (0.0,0.0), (5.0,0.0), (6.0,2.0), (5.0,4.0), and (0.0,4.0).


ITASCA
 BasisSoft


PFC 6.0 Short Course


62


Slide 63



Clumps










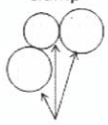


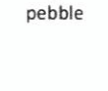




PFC 6.0 Short Course

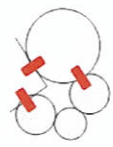
63

Slide 64



PFC Model Components

Bodies:	ball	clump	wall
			
Pieces:	ball	pebble	facet
			

Bodies and pieces have *attributes*.
 Pieces may have *surface properties*,
 which can be used for contact model
 property inheritance.



**contacts occur
between pieces !**



PFC 6.0 Short Course

64

Slide 65

Clump commands

- Clumps
 - Commands
 - clump attribute command
 - clump create command
 - clump delete command
 - clump distribute command
 - clump export command
 - clump extra command
 - clump fix command
 - clump free command
 - clump generate command
 - clump group command
 - clump history command
 - clump initialize command
 - clump list command
 - clump order command
 - clump property command
 - clump replicate command
 - clump results command
 - clump rotate command
 - clump scale command
 - clump template command
 - clump tolerance command
 - clump trace command
 - FISH Functions


PFC 6.0 Short Course

65

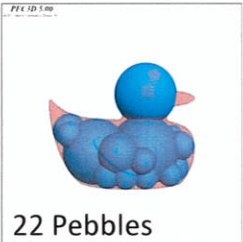
Slide 66

BubblePack

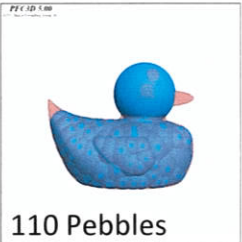
- Built-in BubblePack algorithm to fill clump surface descriptions with pebbles to a desired fidelity via an approximation to the medial axis




PFC ID 5.00





22 Pebbles



110 Pebbles



573 Pebbles

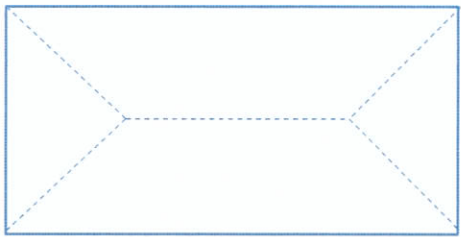
PFC 6.0 Short Course

66

Slide 67

BubblePack

- Built-in BubblePack algorithm to fill clump surface descriptions with pebbles to a desired fidelity via an approximation to the medial axis



ITASCA™ BasisSoft

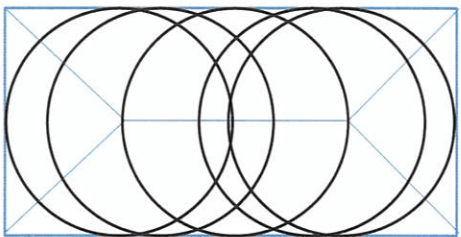
PFC 6.0 Short Course

67

Slide 68

BubblePack

- Built-in BubblePack algorithm to fill clump surface descriptions with pebbles to a desired fidelity via an approximation to the medial axis



ITASCA™ BasisSoft

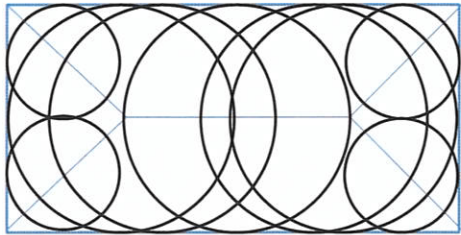
PFC 6.0 Short Course



68

Slide 69

BubblePack

- Built-in BubblePack algorithm to fill clump surface descriptions with pebbles to a desired fidelity via an approximation to the medial axis



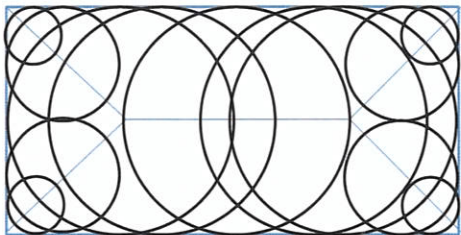
PFC 6.0 Short Course



69

Slide 70

BubblePack

- Built-in BubblePack algorithm to fill clump surface descriptions with pebbles to a desired fidelity via an approximation to the medial axis



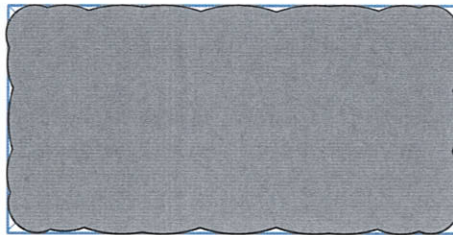
PFC 6.0 Short Course

70

Slide 71

BubblePack

- Built-in BubblePack algorithm to fill clump surface descriptions with pebbles to a desired fidelity via an approximation to the medial axis



ITASCA™

BasisSoft

PFC 6.0 Short Course

71

Slide 72

Scripting with FISH

FISH (FLAC-ish) is a built-in programming language for Itasca software



ITASCA™

BasisSoft

PFC 6.0 Short Course

72

Slide 73

Scripting with FISH

FISH variables and FISH Functions

```
define output_string(s)
  status = io.out(s)
end

[my_word = 'hello !']
@output_string(@my_word)
```

Code usually indented

Function name

Parameters (optional)

Calling function

ITASCA

PFC 6.0 Short Course

BasisSoft

