

# DynamO Workshop

## Tutorial Worksheet 2

The aim of this tutorial is to introduce how to run more complex systems, familiarise you with the file format of DynamO, and to demonstrate how you might simulate your own systems.

### 1 Online Tutorial Instructions

The first part of the tutorial will lead you through manually running a square-well simulation with thermostats and collecting transport properties. This is available on-line with links to the reference documentation [here](#):

<http://dynamomd.org/index.php/tutorial4>

The second part to the tutorial is on manually setting up a multi-component simulation and using compression dynamics to compact it to a higher density. Again, this is available on-line with links to the reference documentation.

<http://dynamomd.org/index.php/tutorial5>

### 2 Automating setting up a simulation

This advanced section is to show you how to set your own simulations up using python to manipulate the configuration file. We will repeat the set up of on-line tutorial 5 but using an automated script. Ultimately you may need to write your own configuration generators from scratch, but here we will use **dynamod** to boot-strap our simulation and perform some modifications.

Our goal is to take a single component square-well system, and modify it to be a multicomponent system:

```
#!/usr/bin/python
#Load the xml library
import xml.etree.ElementTree as ET
import os

#Create the starting configuration
os.system("dynamod -m 1 -C 10 -d 0.5 --il 0 -r 1 -o config.start.xml")

#Open the XML file ready for editing
xmldoc = ET.parse("config.start.xml")

#Remove the current species tags
#Grab the empty Genus tag
GenusTag = xmldoc.find('./Simulation/Genus')
for speciesTag in GenusTag.findall('./Species'):
    GenusTag.remove(speciesTag)

#Add two new Species tags to the Genus tag
species1 = ET.SubElement(GenusTag, "Species",
                          attrib={'Mass': "1", 'Name': 'A', 'Type': 'Point'})
```

```
idrangel = ET.SubElement(species1, "IDRange",
                          attrib={'Type':'Ranged', 'Start':'0', 'End':'99'})

species2 = ET.SubElement(GenusTag, "Species",
                          attrib={'Mass':"0.125", 'Name':'B', 'Type':'Point'})
idrangep2 = ET.SubElement(species2, "IDRange",
                          attrib={'Type':'Ranged', 'Start':'100', 'End':'3999'})

#Remove the current interaction tags
interactionsTag = xmldoc.find('./Simulation/Interactions')
for interactionTag in interactionsTag.findall('./Interaction'):
    interactionsTag.remove(interactionTag)

#Add the three new interactions
for name, diameter in [{"AAInteraction", "1"},
                      ["ABInteraction", "0.75"],
                      ["BBInteraction", "0.5"]]:
    interaction = ET.SubElement(interactionsTag, "Interaction",
                                attrib={'Type':'SquareWell', 'Diameter':diameter, 'Name':name,
                                        'Elasticity':'1', 'Lambda':'1.5', 'WellDepth':'1'})

#Set up the IDPairRanges for each Interaction
interactions = xmldoc.findall('./Simulation/Interactions/Interaction')

pairidrangep1 = ET.SubElement(interactions[0], "IDPairRange",
                              attrib = {'Type':'Single'})
idrangep1 = ET.SubElement(pairidrangep1, "IDRange",
                          attrib={'Type':'Ranged', 'Start':'0', 'End':'99'})

pairidrangep2 = ET.SubElement(interactions[1], "IDPairRange", attrib={'Type':'Pair'})
idrangep1 = ET.SubElement(pairidrangep2, "IDRange",
                          attrib={'Type':'Ranged', 'Start':'0', 'End':'99'})
idrangep2 = ET.SubElement(pairidrangep2, "IDRange",
                          attrib={'Type':'Ranged', 'Start':'100', 'End':'3999'})

pairidrangep3 = ET.SubElement(interactions[2], "IDPairRange",
                              attrib={'Type':'All'})

#Finally, write it out
xmldoc.write("config.start.xml")
```