

CC3D Parameter Scan Setup

Juliano Gianlupi
(Maciej Swat, Julio Belmonte)

Online CompuCell3D Manual 2017 - Revision f13c8f22

Outline

- 1 What is it?
- 2 Setting up Parameter Scan Using Twedit++
- 3 Running Parameter Scan
- 4 Getting rid of parameter scan
- 5 Note on saving files
- 6 Source

What is it?

- You may want to explore your parameter space in your simulations.
- In the past researchers have used (or abused) Python to run multiple replicas of the same simulation with different parameter set for each run.
- Fortunately you don't need to do that anymore.

What is it?

An xml file

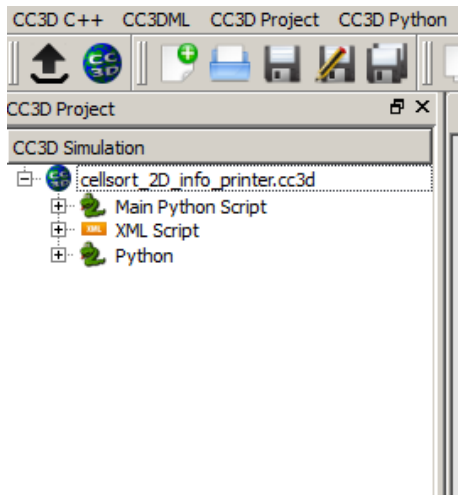
- You can generate a Parameter Scan directly from Twedit++.
- It **won't** change the values in the xml/python files but it **will** change the values the CC3D player uses.

Outline

- 1 What is it?
- 2 Setting up Parameter Scan Using Twedit++**
- 3 Running Parameter Scan
- 4 Getting rid of parameter scan
- 5 Note on saving files
- 6 Source

Step 1

- To setup up parameter scan we open any valid .cc3d project file in Twedit++



Step 2

- Right-click on the project and select Add Parameter Scan option:

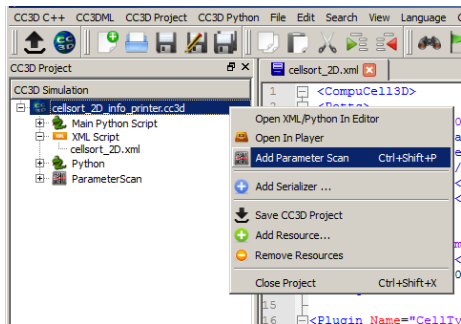


Figure: Step 2

- Notice it'll have added a new file to the project on the left

Step 3

- To add a parameter from the CC3DML to the parameter scan we right-click on the CC3DML file and select Open Scan Editor

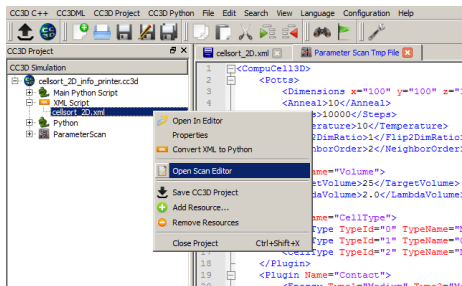
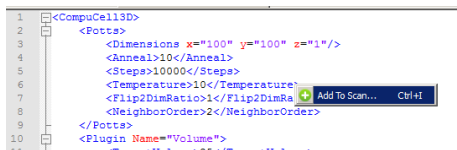


Figure: Step 3

- Notice that Twedit++ opens a new tab called Parameter Scan Tmp File. This is read-only file that you use to select parameters for scanning purposes.

Step 4

- To set the parameter that, you click in the desired place of this file.
- For example if you want to to run simulation with different *Temperature* parameters you click in the line with *Temperature* parameter and then right-click to get access to *Add To Scan...* option:



```
1 <CompuCell3D>
2   <Potts>
3     <Dimensions x="100" y="100" z="1"/>
4     <Anneal>10</Anneal>
5     <Steps>10000</Steps>
6     <Temperature>10</Temperature>
7     <Flip2DimRatio>1</Flip2DimRa
8     <NeighborOrder>2</NeighborOrder>
9   </Potts>
10  <Plugin Name="Volume">
11    <Type>Volume</Type>
```

The screenshot shows a code editor with an XML document. A right-click context menu is open over the line containing the `<Temperature>10</Temperature>` tag. The menu has a green plus icon and the text "Add To Scan..." followed by "Ctrl+I".

Figure: Step 4

Step 5

- After you choose this option Twedit++ displays parameter scan configuration dialog:

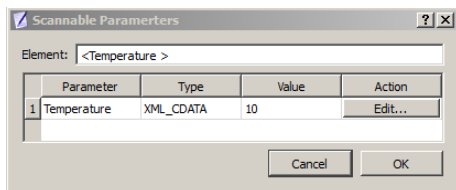


Figure: Step 5

- This dialog displays all parameters in the given XML element which can change. In our case of a simple XML element we have only oneway to change the element and it is through its value (10).

Step 6

- When we click edit we get to next dialog

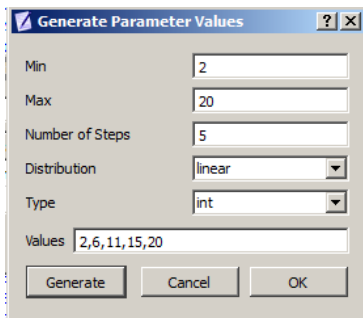
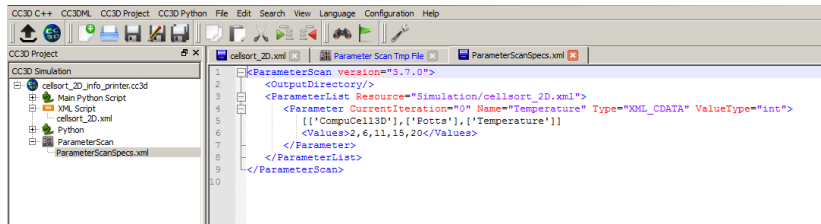


Figure: Step 6

- Fill minimum, maximum value, select value type (integer/float) and generate values (given by the Number Of Steps) according to given distribution.

You're done!

- When you open parameter scan XML file we see that our choices made in the GUI got translated into XML format:



```
1 <ParameterScan version="3.7.0">
2   <OutputDirectory/>
3   <ParameterList Resource="Simulation/cellsort_2D.xml">
4     <Parameter CurrentIteration="0" Name="Temperature" Type="XML_CDATA" ValueType="int">
5       [ ['CompuCell3D'], ['Potts'], ['Temperature']]
6       <Values>2,6,11,15,20</Values>
7     </Parameter>
8   </ParameterList>
9 </ParameterScan>
10
```

Figure: How the ParameterScanSpecs.xml looks like now.

Adding more Parameters

- To add more parameters we position the cursor in the desired location in the *Parameter Scan Tmp* File tab, right click to select Add To Scan... option and follow steps outlined above.

More complicated element

- When we select more complicated element e.g. `<Dimensions>` we will get the following parameter configuration dialog:

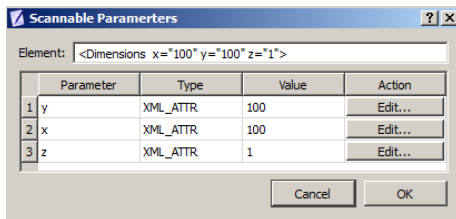


Figure: More complicated element

More complicated element

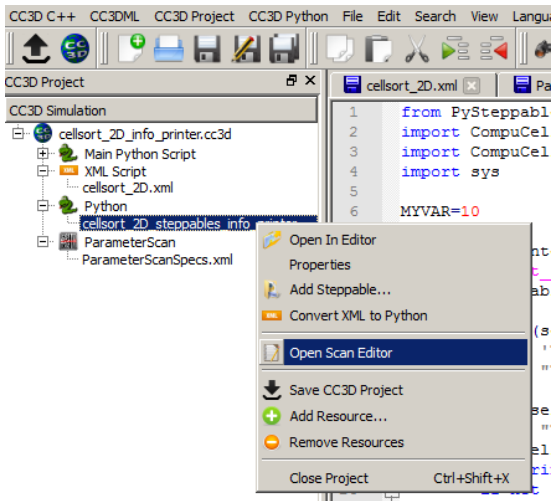
- Notice that for this element there are 3 values which can change when we pick.
- Say you pick x, and generate values in the next pop-up widget we will end up with the following parameter scan XML file:

```
<ParameterScan version="3.7.0">
  <OutputDirectory/>
  <ParameterList Resource="Simulation/cellsort_2D.xml">
    <Parameter CurrentIteration="0" Name="Temperature" Type="XML_CDATA" ValueType="int">
      [[ 'CompuCell3D' ], [ 'Potts' ], [ 'Temperature' ]]
      <Values>2, 6, 11, 15, 20</Values>
    </Parameter>
    <Parameter CurrentIteration="0" Name="x" Type="XML_ATTR" ValueType="int">
      [[ 'CompuCell3D' ], [ 'Potts' ], [ 'Dimensions', 'x', '100', 'y', '100', 'z', '1' ]]
      <Values>20, 65, 110, 155, 200</Values>
    </Parameter>
  </ParameterList>
</ParameterScan>
```

Figure: More complicated element

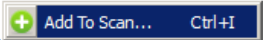
- As you can tell this XML file gets updated automatically so users do not need to type XML file.

- Adding Python parameters to parameter scan is as easy. We open up Python file using Open Scan Editor option:



- Go to the *Parameter Scan Tmp* File tab which now has Python code open in the read-only mode and right click on any **global** variable to add it to the parameter scan.

```
1 from PySteppables import *
2 import CompuCell
3 import CompuCellSetup
4 import sys
5
6 MYVAR=10
7
8 class InfoPrinterSteppable(SteppableBase):
9     def init (self, simulator, freq
```



Figure

Changing Output Directory

- Finally we can also edit the output directory of the parameter scan results by manually editing the parameter scan XML. Here we set it to `<OutputDirectory>InfoPrinter_ParameterScan</OutputDirectory>`

```
<ParameterScan version="3.7.0">
  <OutputDirectory>InfoPrinter_ParameterScan</OutputDirectory>
  <ParameterList Resource="Simulation/cellsort_2D.xml">
    <Parameter CurrentIteration="0" Name="Temperature" Type="XML_CDATA" ValueType="int">
      [[ 'CompuCell3D', ['Potts'], ['Temperature']]
      <Values>2,6,11,15,20</Values>
    </Parameter>
    <Parameter CurrentIteration="0" Name="x" Type="XML_ATTR" ValueType="int">
      [[ 'CompuCell3D', ['Potts'], ['Dimensions', 'x', '100', 'y', '100', 'z', '1']]
      <Values>20,65,110,155,200</Values>
    </Parameter>
  </ParameterList>
  <ParameterList Resource="Simulation/cellsort_2D_steppables_info_printer.py">
    <Parameter CurrentIteration="0" Name="MYVAR" Type="PYTHON_GLOBAL" ValueType="int">
      <Values>2,6,11,15,20</Values>
    </Parameter>
  </ParameterList>
</ParameterScan>
```

Figure: Output Directory

Outline

- 1 What is it?
- 2 Setting up Parameter Scan Using Twedit++
- 3 Running Parameter Scan**
- 4 Getting rid of parameter scan
- 5 Note on saving files
- 6 Source

Running Parameter Scan

- As you can see defining parameters scans is quite easy.
- Running simulation is easy as well and you have two options:
 - The recommended way of running parameter scan is via script, this will come latter.
 - The most straightforward is to open up the project in the Player and run it.

Running Parameter Scan

From the Player

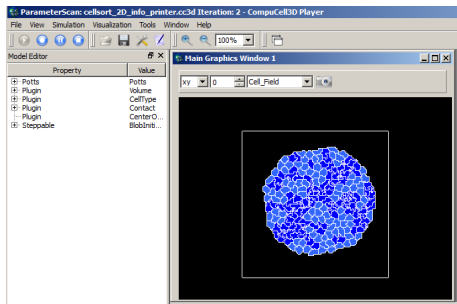


Figure: Running from the Player

- Notice that the title bar of the player informs you that you are running parameter scan .
- It also displays number of the parameter simulation that is currently being executed (*Iteration: 2*).

- The results of the parameter scans are written to the *Simulation Output* directory :

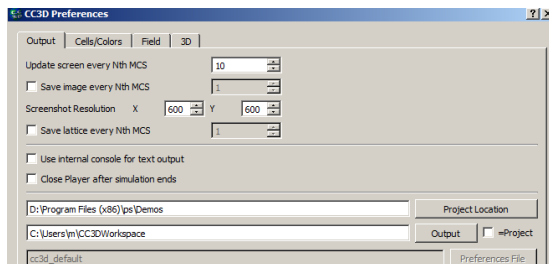
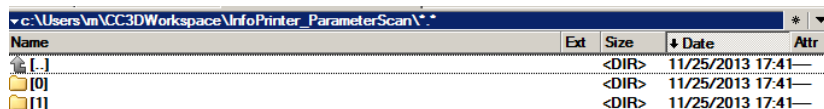


Figure: Output

Looking inside the directory

- When we look into this directory we will see that it contains subdirectories corresponding to the consecutive simulation runs – each such simulation corresponds to a different set of parameters:



Name	Ext	Size	Date	Attr
[..]		<DIR>	11/25/2013 17:41	—
[0]		<DIR>	11/25/2013 17:41	—
[1]		<DIR>	11/25/2013 17:41	—

Figure: Directory

Looking inside the directory

How the folders are ordered

- Parameter scan will cycle through the parameters in a particular order.
- It will cycle from top to bottom in the lists of parameters, in other words:
 - For every cycle of the first parameter the second will be incremented by one.
 - For every cycle of the second parameter the third will be incremented by one.
 - And so forth...

```
<ParameterScan version="3.7.0">
  <OutputDirectory>InfoPrinter_ParameterScan</OutputDirectory>
  <ParameterList Resource="Simulation/cellsort_2D.xml">
    <Parameter CurrentIteration="0" Name="Temperature" Type="XMLCDATA" ValueType="int">
      [['CompuCell3D'], ['Potts'], ['Temperature']]
      <Values>2, 6, 11, 15, 20</Values>
    </Parameter>
    <Parameter CurrentIteration="0" Name="x" Type="XML_ATTR" ValueType="int">
      [['CompuCell3D'], ['Potts'], ['Dimensions', 'x', '100', 'y', '100', 'z', '1']]
      <Values>20, 65, 110, 155, 200</Values>
    </Parameter>
  </ParameterList>
  <ParameterList Resource="Simulation/cellsort_2D_steppables_info_printer.py">
    <Parameter CurrentIteration="0" Name="MYVAR" Type="PYTHON_GLOBAL" ValueType="int">
      <Values>2, 6, 11, 15, 20</Values>
    </Parameter>
  </ParameterList>
</ParameterScan>
```

Figure: Order of folders

Outline

- 1 What is it?
- 2 Setting up Parameter Scan Using Twedit++
- 3 Running Parameter Scan
- 4 Getting rid of parameter scan**
- 5 Note on saving files
- 6 Source

Getting rid of parameter scan

- If you ever want get rid of parameter scan it is very easy.
- You first need to open the **actual** .cc3d file

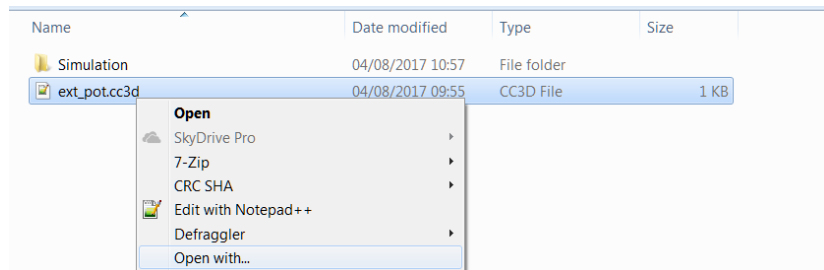
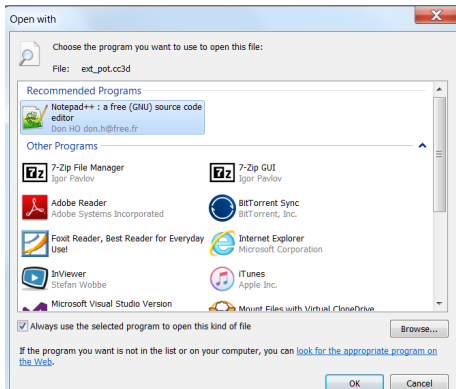


Figure: Right click the .cc3d and select *Open with*

Getting rid of parameter scan

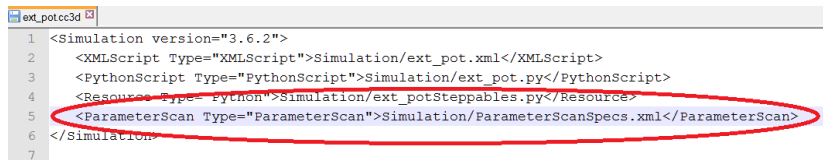
- You can select any text editor, I recommend either the normal *notepad* or *Notepad++*



Figure

Getting rid of parameter scan

- Now you only need to delete the line that adds the parameter scan as a resource



```
ext_potcc3d x3
1 <Simulation version="3.6.2">
2   <XMLScript Type="XMLScript">Simulation/ext_pot.xml</XMLScript>
3   <PythonScript Type="PythonScript">Simulation/ext_pot.py</PythonScript>
4   <Resource Type="Python">Simulation/ext_potSteppables.py</Resource>
5   <ParameterScan Type="ParameterScan">Simulation/ParameterScanSpecs.xml</ParameterScan>
6 </Simulation>
7
```

Figure

- You're done!

Outline

- 1 What is it?
- 2 Setting up Parameter Scan Using Twedit++
- 3 Running Parameter Scan
- 4 Getting rid of parameter scan
- 5 Note on saving files**
- 6 Source

Note on saving files

- Sometimes Parameter Scan will not behave properly and save your files in the wrong place.
- In other to counter that you can use a combination of python commands (it's long).
- You'll need to use the following spinet of code:

```
import os
import os.path as path
import shutil
import inspect

global saving_directory
saving_directory = path.dirname(path.abspath(
inspect.getfile(inspect.currentframe())))
```

Outline

- 1 What is it?
- 2 Setting up Parameter Scan Using Twedit++
- 3 Running Parameter Scan
- 4 Getting rid of parameter scan
- 5 Note on saving files
- 6 Source**

Source

(It has more information)

- It is the online CC3D manual
- `http://pythonscriptingmanual.readthedocs.io/en/latest/parameter_scans.html`