USER MANUAL

SOFTWARE APPLICATION

PHYSICAL GROWTH MECHANISM AND CELLS' GROWTH

Yuri K. Shestopaloff[1]

TABLE OF CONTENTS

1. Introduction

This software application computes growth dependences for cells that have different geometrical forms, such as disk, sphere, two joined frustum cones, cylinder, cylinder with hemispheres at ends (similar to E. coli's shape). Frustum model can be changed from two cones joined by bases to a cylinder by changing the apex radius in the application's configuration file *CellGrConfig.txt* that is located in the application's *config* directory. The theory and some results can be found in the book by Yuri K. Shestopaloff "*Physics of growth and replication. Physical and geometrical perspectives on living organisms' developmen*t". "ISBN-13 9780980966756, ISBN-10 0980966752, Library of Congress Control Number: 2009943094. The book was published by AKVY Press, 2010, 174 p. [1, 2]. However, for the most up-to-date information, refer to the latest author's articles that can be found on the author's web site.

2. Project description

The project includes:

1. User manual
2. Four sample data files in the *Logs* directory
3. Sample configuration file in the *config* directory. Note that the supplied sample configuration file contains parameters that were used to generate sample data and sample Excel graphs.
4. Excel file with sample data for the sphere, disk, frustum, cylinder E. coli cells' models and graphs for these data in the *Docs* directory
5. VC++ project with source files
6. Executable *shesgr.exe* in the *Debug* directory that can be used on Windows platforms to compute growth dependencies for different cells. The user has a great deal of flexibility to define the cells' parameters changing the appropriate values in the configuration file.

3. License terms

This software application, executable, project files and supporting documentation can be freely used for any legitimate purpose except selling this application for profit without the author's prior consent. The users can use, change and distribute the code and executables under the terms of Free Software License, which means that no additions can restrict the originally granted freedom of using this application in software, textual, hardware and other applicable forms.

4. Modeling growth of cells using *shesgr.exe* executable

4.1 Compatibility issues

Although the author provides executable for Windows platforms, the project can be rebuilt for other platforms supporting C++ STL (standard template library). The code *is not* platform specific.

4.2. Getting help

Application help

Open command prompt. (You can type *cmd.exe* in the *Run* window to do that.) Using command *cd*, change directory to the project's *Debug* directory. For instance,

cd C:\Dev\CellGrowth\Debug

Once in this directory, type *shesgr.exe*

You will get the information on terms of using this application and following help message:

You should provide one valid argument from the list: disk, sphere, frustum, ecoli, ecolicylinder
Cell's parameters are configured in CellGrConfig.txt file located in the config directory. Calculation results will be shown on the screen and also written into the appropriate file located in Logs directory.

Project's source files

Project's source files are commented. Studying these comments and the code will help to understand the application and its logic as well.

Help documentation

Refer to User Manual and sample text and spreadsheet (Excel) files in the Docs directory.

4.3. Starting application

Once in the project's *Debug* directory, type the following and press *Enter*:

shesgr.exe sphere (1)

This way you compute the growth parameters for a sphere like cell. Substituting instead of a command argument *sphere* another one, such as *disk, frustum, ecoli, ecolicylinder*, you will accordingly compute growth parameters for the cells that have a disk form, a form composed of two frustums joined by bases, cylinder with hemispheres at ends and a cylinder like cell.

4.4. Computing growth parameters of a sphere like cell

Run command (1). Once the application starts, you will see the following output:

```
*SPHERE CELL
InitSphereVolume 3.0
EndSphereVolume 6.0
SphereGrowthTime 1.95
NumberOfTimePoints 20
InflowChangeDegree 3.0
MembInflowScaleFactor1 1.118
MembInflowScaleFactor2 1.0
TimeAxisScaleCoef 1.0
*DISK CELL
InitDiskVolume 1.0
EndDiskVolume 2.06
DiskHeightAsFractionOfDiameter 1.0
InflowChangeDegreeDisk 0.0
MembInflowScaleFactor1Disk 1.0
MembInflowScaleFactor2Disk 1.0
TimeAxisScaleCoefDisk 1.0
NumberOfRadiusPointsDisk 30
*FRUSTUM CELL
ApexDiameterAsFractionOfBaseDiam 0.5
InitialLengthInBaseDiameters 1.5
```

EndLengthInBaseDiameters 3.0
TimeAxisScaleCoefFrustum 1.0
NumberOfLengthPointsFrustum 30
InflowChangeDegreeFrustum 2.0
MembInflowScaleFactor1Frustum 1.0
MembInflowScaleFactor2Frustum 1.0
*ECOLI CELL
InitLengthInDiametersEcoli 3.0
EndLengthInDiametersEcoli 8.1
NumberOfLengthPointsEcoli 35
InflowChangeDegreeEcoli 2.0
MembInflowScaleFactor1Ecoli 1.0
MembInflowScaleFactor2Ecoli 1.0
TimeAxisScaleCoefEcoli 1.0
*ECOLICYLINDER CELL
TimeAxisScaleCoefEcoliFlat 1.0

Time=0 Radius=0.8947 VolumeIncrease=1 G-1=0.25992 k=1.0008
Time=0.0975 Radius=0.92038 VolumeIncrease=1.0886 G-1=0.22477 k=1.0895
Time=0.195 Radius=0.94623 VolumeIncrease=1.1829 G-1=0.1913 k=1.1839
Time=0.2925 Radius=0.97145 VolumeIncrease=1.28 G-1=0.16039 k=1.2811
Time=0.39 Radius=0.99516 VolumeIncrease=1.3761 G-1=0.13274 k=1.3772
Time=0.4875 Radius=1.0167 VolumeIncrease=1.4672 G-1=0.10878 k=1.4684
Time=0.585 Radius=1.0355 VolumeIncrease=1.5504 G-1=0.088592 k=1.5516
Time=0.6825 Radius=1.0516 VolumeIncrease=1.6238 G-1=0.071926 k=1.6251
Time=0.78 Radius=1.0651 VolumeIncrease=1.6871 G-1=0.058356 k=1.6885
Time=0.8775 Radius=1.0762 VolumeIncrease=1.7406 G-1=0.047395 k=1.742
Time=0.975 Radius=1.0854 VolumeIncrease=1.7853 G-1=0.038573 k=1.7868
Time=1.0725 Radius=1.0929 VolumeIncrease=1.8224 G-1=0.031474 k=1.8239
Time=1.17 Radius=1.0989 VolumeIncrease=1.8531 G-1=0.025754 k=1.8546
Time=1.2675 Radius=1.1039 VolumeIncrease=1.8784 G-1=0.021132 k=1.8799
Time=1.365 Radius=1.108 VolumeIncrease=1.8992 G-1=0.017386 k=1.9008
Time=1.4625 Radius=1.1113 VolumeIncrease=1.9164 G-1=0.01434 k=1.918
Time=1.56 Radius=1.114 VolumeIncrease=1.9305 G-1=0.011854 k=1.9321
Time=1.6575 Radius=1.1163 VolumeIncrease=1.9422 G-1=0.0098193 k=1.9438
Time=1.755 Radius=1.1181 VolumeIncrease=1.9519 G-1=0.0081486 k=1.9535
Time=1.8525 Radius=1.1197 VolumeIncrease=1.9599 G-1=0.0067734 k=1.9615
Time=1.95 Radius=1.1209 VolumeIncrease=1.9665 G-1=0.0056382 k=1.9682

The first print is the content of the configuration file. The part

```
*SPHERE CELL
InitSphereVolume 3.0
EndSphereVolume 6.0
SphereGrowthTime 1.95
NumberOfTimePoints 20
InflowChangeDegree 3.0
MembInflowScaleFactor1 1.118
MembInflowScaleFactor2 1.0
TimeAxisScaleCoef 1.0
```

relates to computing of growth parameters of a sphere like cell. You can configure the beginning volume, the end growth volume, time interval of growth, number of points to compute on the time axis and a time axis's scale coefficient. The last parameter is needed to adjust the location of theoretical growth curves to experimental data.

Three parameters

```
InflowChangeDegree 3.0
MembInflowScaleFactor1 1.118
MembInflowScaleFactor2 1.0
```

define the dependence of the membrane's inflow on the size of the cell. For the sphere, it depends on radius as follows.

$$k = F_2 (F_1 R)^D \qquad (2)$$

Here, D, F_1, F_2 are the listed above configuration parameters. This way, the membrane's inflow can be changed in a wide range. For instance, the value $D = 0$ makes the membrane's inflow constant, while the value of $D = 1$ provides linear dependence. Value of $D = 3$ makes the membrane's inflow proportional to volume.

Excel file *CellGrowth.xls* in the Docs directory presents sample results of calculation and appropriate graphs for two growth scenarios with different end volumes: 6.0 and 8.13 (switch to the *SphereGrowth* tab in Excel spreadsheet). These two scenarios present two evolutional types of growth described in the articles and books, when the cell growth proceeds through the whole growth cycle defined by the physical growth mechanism (value of 6.0), and when the growth stops once the growth rate provided by the physical growth mechanism begins to decelerate.

Note that we use a direct analytical solution of the growth equation as a computational algorithm. Implementation details can be found in the project's source files.

The second print presents output data with parameters names. G-1 denotes the value of the growth ratio minus one; *k* is the membrane's inflow. Similar output will be shown for all other growth scenarios. The same data are written into output files in the *Logs* directory. However, there are no parameters' names in order to make these outputs convenient to use in spreadsheets or storing in the databases. If in doubt what column means what, refer to the command window output.

4.5. Computing growth parameters of a disk like cell

Run a command

shesgr.exe disk

Once the application starts, you will see the content of the configuration file. The part

```
*DISK CELL
InitDiskVolume 1.0
EndDiskVolume 2.06
DiskHeightAsFractionOfDiameter 1.0
InflowChangeDegreeDisk 0.0
MembInflowScaleFactor1Disk 1.0
MembInflowScaleFactor2Disk 1.0
TimeAxisScaleCoefDisk 1.0
NumberOfRadiusPointsDisk 30
```

relates to configuration of a disk cell. The growth equation is solved numerically in this computational implementation, although there is analytical solution for the disk growth scenario with respect to the growth time (same as for a sphere).

Membrane's inflow is configured in the same way (2) as for a sphere. Unlike in the case of a sphere, this time, we define the computational points for discrete values of disk radius (*NumberOfRadiusPointsDisk* parameter).

Default configuration parameters produce output shown in *CellGrowth.xls* file, tab *Disk*. In this case, the membrane's inflow is constant. We used this model to compute amoeba's

growth curves (membrane's inflow of amoeba remains approximately constant during the growth).

4.6. Computing growth parameters of a cell formed by two joined frustum cones

Run a command

shesgr.exe frustum

Once the application starts, you will see the content of the configuration file. The part

```
*FRUSTUM CELL
ApexDiameterAsFractionOfBaseDiam 0.5
InitialLengthInBaseDiameters 1.5
EndLengthInBaseDiameters 3.0
TimeAxisScaleCoefFrustum 1.0
NumberOfLengthPointsFrustum 30
InflowChangeDegreeFrustum 2.0
MembInflowScaleFactor1Frustum 1.0
MembInflowScaleFactor2Frustum 1.0
```

relates to configuration of a cell composed of two frustum cones. Fig. 1 below shows the form of such a cell.

Default configuration parameters produce sample output and the appropriate growth curve presented in *CellGrowth.xls file*, tab *Frustum*.

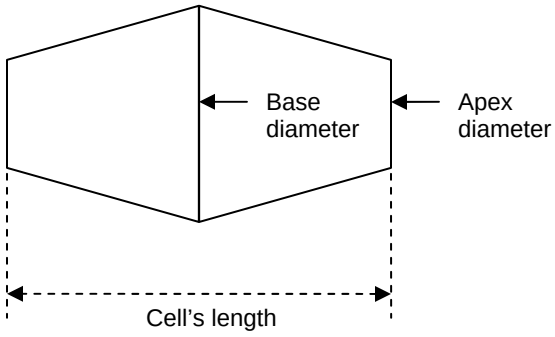


Fig. 1. A cell composed of two joined frustum cones.

The cell's shape is characterized by diameters of the base and apex circles and the length. All these values are measured as fractions of the base's diameter.

The convenience of this model is its universality. When the apex diameter is zero, then the frustum cone becomes a cone, while when the apex's and base's diameters are the same, the shape transforms to a cylinder. For this form, volume is proportional to frustum cone's length, so that the differences in the beginning and end growth length present also the difference between the volumes of initial and fully grown cell.

Membrane's inflow is defined by the following parameters:

InflowChangeDegreeFrustum 2.0
MembInflowScaleFactor1Frustum 1.0
MembInflowScaleFactor2Frustum 1.0

The functional dependence is as follows.

$$k(L) = F_2\left(F_1 \frac{L}{L_{init}}\right)^D \qquad (3)$$

As a side note, we computed numerically the form of the membrane's inflow dependence for known experimental data. These are monotonically increasing convex functions. Formulas (2), (3) reflect on this specific, although they can also provide concave and constant functions. In all growth scenarios, the membrane's inflow is constant when $D = 0$. Cell's density is assumed constant in all considered growth scwenarios.

4.7. Computing growth parameters of E. coli cell

Run a command

shesgr.exe ecoli

Once the application starts, you will see the content of the configuration file. The part

*ECOLI CELL
InitLengthInDiametersEcoli 3.0
EndLengthInDiametersEcoli 8.1
NumberOfLengthPointsEcoli 35
InflowChangeDegreeEcoli 2.0
MembInflowScaleFactor1Ecoli 1.0
MembInflowScaleFactor2Ecoli 1.0
TimeAxisScaleCoefEcoli 1.0

relates to configuration of a cell composed of a cylinder with two hemispheres at ends, as it is shown in Fig. 2.

Fig. 2. E. coli cell modeled by hemispheres and a cylinder.

The configuration parameters are self explanatory. Membrane's inflow is configured as for a frustum like cell, equation (3). Note that this model does not provide proportionality between the length and volume because of the presence of hemispheres. So, if one needs to increase the grown volume twice, he should increase the length more than that. The actual number can be found from the basic geometrical considerations. In units of diameter, if the grown volume is *M* times larger, then the length of a cylinder for the grown cell is defined as follows.

$$L_{grown} = M(2/3 + L_{init}) - 2/3 \qquad (4)$$

E. coli can be modeled by a cylinder as well. To compute this growth scenario, run command

shesgr.exe ecolicylinder

Configuration parameters produce output presented in the *CellGrowth.xls* file, open tab *E. coli cylinder*.

The part of configuration file related to cylindrical e. coli growth is this.

```
*ECOLICYLINDER CELL
TimeAxisScaleCoefEcoliFlat 1.0
```

The rest of model's parameters is defined in the *ECOLI CELL section that we described above. So, the only new parameter is the time scaling coefficient. For this model, the volume is proportional to cylinder length. So, if the grown cell is twice as bigger than the initial cell, then their lengths differ two times as well.

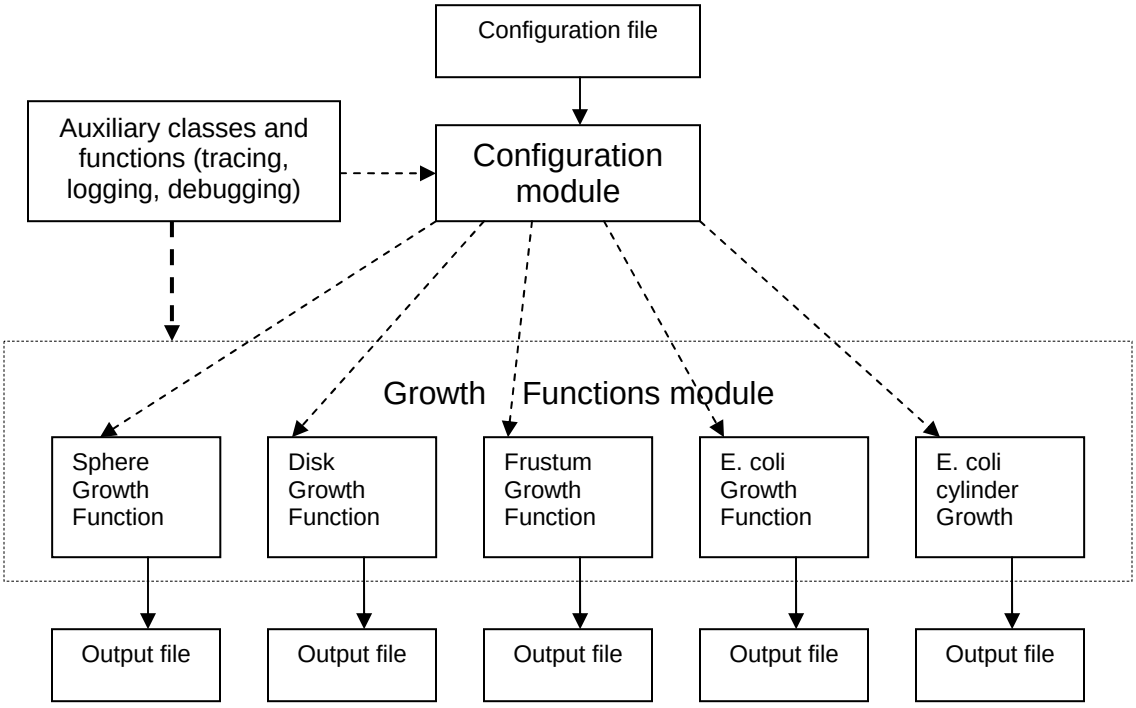
5. Software implementation details

5.1. Programming language. Application's portability

Software application is written in C++ programming language with usage of STL (Standard Template Library). No platform specific code was used, so that the application should be easily portable to any platform supporting C++. Executable was built for Windows platforms.

5.2. Application design

Application design is straightforward. We intentionally isolated and made growth functions global in order to make them self-sufficient. This allows making changes easily, while each function has a well defined interface. Auxiliary functions support logging and debugging functionalities. In-memory configuration information is stored in the STL container map. The configuration data are retrieved based on the name of key identical to one in the configuration file. If one needs additional growth function, he can continue adding them in the same manner without changing the rest of the code.



Fog. 3. Growth application's system design.

5.3. Application's interface

This is a command line application. However, if one wants to add a graphical interface, the author would not recommend to embed the code into a graphical application, but rather to build a GUI to manipulate the content of the configuration file. Then, from the same GUI application, one should start the application's executable in order to compute the growth data. The content of log files can be viewed from the same GUI application by opening them in some text editor. For ergonomic reasons, the author recommends to put the list of growth scenarios into combobox instead of adding tabs. The author used this design for other applications, and it worked well. Divide and conquer – this principle well complies with the suggested GUI design.

For the frustum cell, the integration steps are chosen very small to make calculations more accurate. However, if one needs better performance, increase the integration step.

References

1. Yuri K. Shestopaloff, “Physics of growth and replication. Physical and geometrical perspectives on living organisms’ development”, 2010, AKVY Press, 174 p.
2. Yuri K. Shestopaloff, “Growth and replication of cells and other living organisms. Physical mechanisms that govern Nature’s evolvement”, 2009, AKVY Press, 84 p.