BRIEF TUTORIAL ATOMISTIC POLYGEN

Correspondence email: crislpo@if.usp.br

May 20, 2016

Contents

List of Figures

Chapter 1

REFERENCE GUIDE

1.1 Modeling

PolygenDNA is a software that has as objective to model and to fit nano structures polyhedral. These models are based on nanocages of *DNA self-assembly*.

The software has a database with about 100 convex polyhedrons, Appendix A, that allows to model DNA nanocages using few parameters. To start the modelling, the user need fill/choose a convex polyhedral. If the user has a previous sequence is possible configure the sequence according as geometry, but if the user don't has sequence the program generate a random sequence. When the user don't has the information of scale, geometry/cage size, is recommended start the modelling setting the value of geometry Radius to 0, in 'R_expand: 0.0'. During the procedure of minimization, the program or the user can to change, little by little, this scale value ('R_expand:') until to find a optimum value, that adjust to all DNA ds and DNA ss in a specific geometry.

1.2 Fitting

To model the nanocages or find the optimum spatial configuration to all atoms, the PolygenDNA use simulated annealing algorithm. This algorithm, minimize the positions of all DNAs according to a correspondent geometry.

1.3 Procedures to build a atomistic model

The atomistic DNA cages, modeled in this work, are convex polyhedrons with DNA double strand (ds) in the edges, and DNA single strand (ss) linking the DNA ds. This procedure truncate the cage, figure 1.1. To build the atomistic model, two precedure are performed: geometry modeling (this procedure is based in a previous work (Alves et al., 2014) and atomistic modeling.

In this way, firstly the geometry is generated using a database (Andrew Humeâs Polyhedron Database; http://www.netlib.org/polyhedra/) or using random coordinates in a spherical surface, using in this last case, the algorithm *gift wraping*. Using this geometry model, the second procedure is modeling the DNAs and align it according with the respective geometry. The ds strands are generated individually, according with sequence, and are rotated and translated to align with the edges, in the polyhedron. To modeling the DNA, two approach were developed: i) one to the double strand (ds) and other to single strand (ss). The modeling used in DNA ds, is based in standards models, according with the sequence of nucleotides and respective spatial coordinates. After this procedure, each DNA ds axis is aligned parallel to respective edge in the polyhedron. In these sense, is necessary control the size of the edge of polyhedron, to position the DNA ds, and the relative distance between the helices. The positioning of DNA ds is realized by face, because the sequence nucleotides are planning by face.

In this stage, the polyhedron have shape, but the DNA ds strands not are connected, because the main difficulty to modeling polyhedron DNA cages is position the DNA ds in the edge, according to the position of linker (pink DNA in figure 1.1, between the helices. So, after the positioning of all double helices in polyhedron, the atomic coordinates, “O3” or “P”, of each nucleotide in tips of ds (the primer and terminus nucleotides of each ds), are save to represent the DNA ds model. After the previous step, is started the minimization of the atomic position “O3” and “P”, (rotation and translation of each ds) to obtain the same distance between ds (distance between the

respective strands). When all distances, between ds, is according with the desired, the minimization procedure is finalized.

Then, the last step is modeling the linker, or DNA ss, and position it correctly to coordinates relatively to each ds. The DNA ss modeling is performed building a "straight" linker. If the length of this linker (distance between the "P" atom in first nucleotide to the "O3" atom in last nucleotide) is different of the distance desired, then is applied a change in the backbone torsion angle. This last procedure, is finalized when the distance desired is achieved or when all the torsions, in DNA ss, are rotated to maximum or minimum distance possible.

1.4 Algorithm used to find the best positions of DNA ds

The objective of algorithm is find similar distances between tips of DNA double helices, because all linkers need has same length. The difference between the sum of distance desired and the sum of distance obtained with PolygenDNA is calculated, and the user can control this difference choosing a value, in file parm_modelage.in, "fit_restrition:". This parameter represent a measure calculated in each step of minimization. If the program calculate a sum that is lesser than the value in "fit_restrition:", the atomistic model will be built. Otherwise the program change the values of "parameters:" and perform the next step in minimization procedure.

Then, to find similar distances between tips of DNA ds, the PolygenDNA rotate the atoms "P" and "O3". The first value, in column 2 of "parameters:" (file parm_modelage.in), is the value of rotation (degrees) around of DNA ds axis. This procedure use a quaternions.

If the user configure the columns 3 and 4 of matrix of "parameters:", the PolygenDNA rotate the atoms "P" and "O3" around the axis 1 or axis 2, that are perpendicular to helical axis. The last column in the "parameters:" represent the value to translate the center of mass of DNA ds (CM-DNA). In this procedure, the PolygenDNA calculate a versor between CM of cage and CM-DNA. The versor is used to define the direction of

translation of atoms "P" and "O3", using the value chosen by user.

The algorithm basic is performed hereafter:

```
begin
  for step : 1 to n_interactions step 1 do
    comment: translate DNA ds, using versor CM cage and CM-DNA;
    Translate_all_DNA_tips(all_DNAs);
    comment: rotate DNA ds around helical axis, using quaternion;
    Rotate_all_DNA_tips_quartenion(all_DNAs);
    comment: rotate DNA axis 1 or axis 2, that are perpendicular to helical axis;
    Rotate_all_DNA_tips_M_rotation(all_DNAs);
    comment: calculate the sum of lengths, between DNAds tips: "P" and "O3";
    Slinkers = Calculate_sum_of_all_DNA_linkers(all_DNAs);
    if Slinkers < fit_restriction then
                                        Make_atomistic_model
                                        exit
    fi;
  end
```

Slinkers is the sum of lengths of all linkers, fit_restriction is a value defined by user.

1.5 Quaternion

A quaternion q, is a vector in 4D, where:

$$q = \langle w, x, y, z \rangle = w + xi + yj + zk. \tag{1.1}$$

A position, $\boldsymbol{p} = (p_x, p_y, p_z)$ in space three-dimensional, can be rotated to $\boldsymbol{p'}$ (using

quaternion) around a axis **u** by θ according with:

$$\boldsymbol{p'} = \boldsymbol{qpq^{-1}} \tag{1.2}$$

and **q** is:

$$\boldsymbol{q} = e^{\frac{\theta}{2}\boldsymbol{u}} \tag{1.3}$$

1.6 Rotation matrix

A position, $\boldsymbol{p} = (p_x, p_y, p_z)$ in space three-dimensional, can be rotated to $\boldsymbol{p'}$ (using rotation matrix, M) by Euler angles α, β, γ according with:

$$p' = Mp \tag{1.4}$$

and the rotation matrix, M, is:

$$M = \begin{pmatrix} cos\alpha cos\gamma - sin\alpha cos\beta sin\gamma & sin\alpha cos\gamma + cos\alpha cos\beta sin\gamma & sin\beta sin\gamma \\ -cos\alpha sin\gamma - sin\alpha cos\beta cos\gamma & -sin\alpha sin\gamma + cos\alpha cos\beta cos\gamma & sin\beta cos\gamma \\ sin\beta sin\gamma & -cos\alpha sin\beta & cos\beta \end{pmatrix} \tag{1.5}$$

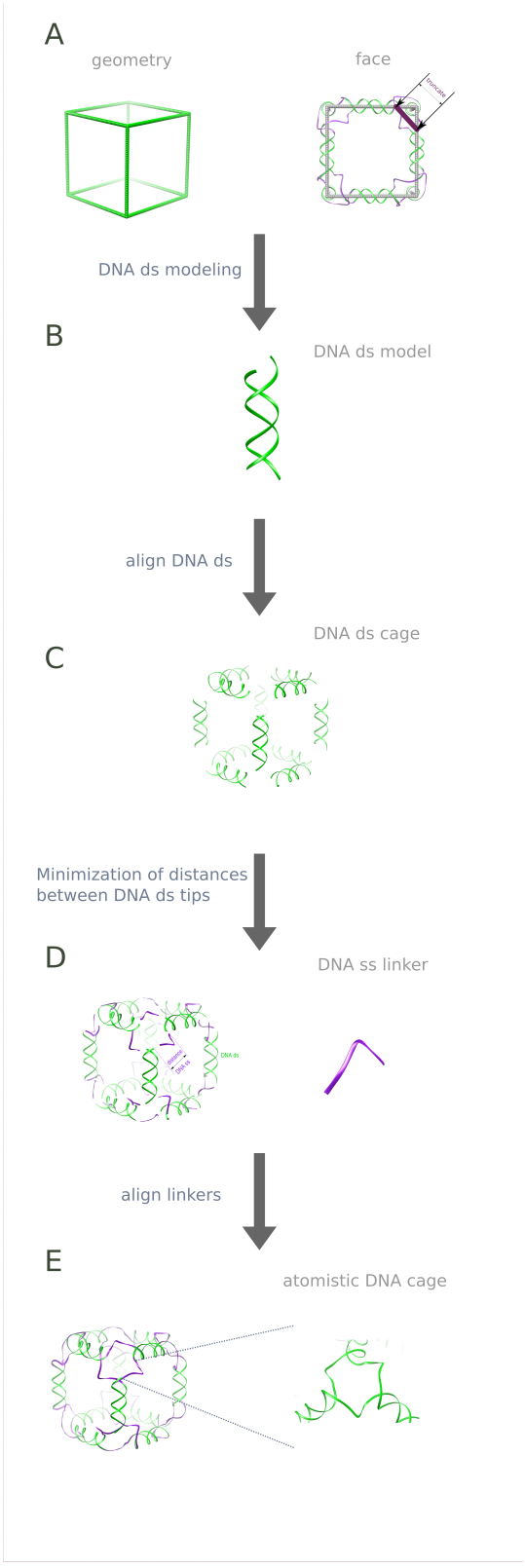

Figure 1.1: *Procedure to model the nanocages. A) Initial geometry. B) Creation of the DNA ds using known atomic resolution structures. C) DNA ds are placed and aligned to the polyhedron edges. In this step an optimization is performed, to find similar distances between the DNA ds tips. D) Creation of the DNAs single strands. E) All the linkers, DNA ss, are inserted in the model to connect the DNA ds.*

Chapter 2

USER GUIDE ATOMISTIC

The PolygenDNA is a program to generate atomistic models nanocages. To start modelling is necessary has in the same folder all these 4 files:

- *PolygenDNA*: the executable file
- *linker_input.pdb*: linker data-base. If the user change the content of this file, PolygenDNA will build the nanocages using this linker model.
- *configure.dat*: configuration file. This file set the basic information to start modelling. The content of this file enable modify the parameters as kind and size of geometry and linker, and also allow modify the basic information to choose of DNA sequence parameters.
- *parm_modelage.in*: file necessary to start procedure minimization of parameters of geometries and of atomistic structures. The file content is used to load the parameter about each DNA ds and enable modify its positions.

Inside of the folder where is the executable, type "./PolygenDNA" to start the modeling. Pay attention, before start modelling choose/fill a small set of parameters in files: "configure.dat" and "parm_modelage.in".

In configure.dat file, is necessary choose/fill 3 parameters related with geometric model, 2 parameters related with linker model, and 2 parameters related to DNA sequence.

To use the file *configure.dat*. Part 1 - Geometries:

- *number*: number of the geometry, see list of geometries in appendix A. (Integer number > 0)
- *len_linkers*: length linker, number of single strand nucleotides, necessary to connect two DNA double helices. This parameter is useful only if the user intend use PolygenDNA linker. If the user choose the value '116' in this parameter and fill a set of spatial coordinates inside of file 'positions.dat', is possible use this spatial coordinates to build a nanocage. This last procedure use gift wrapping algorithm. (Integer number ≥ 0 and ≤ 116)
- *kind_linker_model*: kind of linker model, with 2 options: 1 PolygenDNA standard or 2 external (linker_input.pdb) structures. In option 1 the PolygenDNA build the linker structure to connect DNAs ds and option 2 PolygenDNA use the structure contained in file linker_input.pdb, to connect DNAs ds.(Integer number, 1 or 2)
- *min_dist_steric:*: this parameter configure the minimum distance used to find steric clashes between atoms, in different residues. (Real number ≥ 0.0)
- *R_expand*: expansion/scale radius in Angstroms. This is a radius that contains geometry, radius of a circumscribed sphere. If the user choose a value equal 0.0, the program find the best value to R_expand and after minimize the structure. But, if the user choose a value > 0.0 the program minimize the structure using this value. (Real number ≥ 0.0)

The next parameters are referent to DNA sequence related to same geometry, but if the user don't have idea about sequence or number of parameter necessary to elaborate

sequence. In this case, is possible modelling a standard nanocage setting the parameter "seq_automatic:" to 1 (true), as "seq_automatic: 1". In other hands, if this parameter is 0 (false) is necessary fill all parameters in this file.

Firstly, is recommended to choose the option 1 and after set it to 0 and modify or rewrite the set sequence, according to intent.

File *configure.dat*, part 2 - FACES_SEQUENCES:

- *seq_automatic*: This parameter set the DNAs sequence automatically (1) or enable the user set its manually (0). If the user not have a previous information about the correct parameters to fill sequence, is recommended change this parameter to 1 (true), as "seq_automatic: 1". Then, run PolygenDNA and after, change this parameter to 0 (false) and fill the bellow parameters. (Integer number, 0 or 1).

- *n_sequences*: Number of the sequence or faces in each polyhedron. The nanocages are modelled setting the sequence according with number of faces, so the number of the faces is equal to number of sequences. Example: the cube has 6 faces so require 6 oligonucleotides sequences. (Integer number > 0)

- *max_residues*: The maximum number of nucleotides in a face, or the maximum number of residues present in a sequence. Example: if the biggest oligonucleotides sequence has 23 nucleotides this parameter is "max_residues: 23". (Integer number > 0)

- *face*: The number of this parameter change according with of the geometry, or with the number of the faces in geometry. So to build a cube is necessary set 6 faces or 6 oligonucleotides sequences. Each sequence start after first "/" and finish in the last "/", and the nucleotides are filled between this two slashes, without blank spaces. Example:
  face: 1: /CTTAGAGTTGCCACCAGGTTTTTTT/.

In this sequence are present DNA ds and DNA ss (linkers), as /DOUBLEHELICE*LINKER*DOUBLEHELICE*LINKER*/.

The above parameters are characteristics related the sequence and the correspondent geometry. The next parameters are a mask, as stencil, that unify the informations about sequence and geometry .

file *configure.dat*, part 3 - FACES_STENCIL_SEQUENCES::

- *face*: This parameter change according or with the number of faces of the geometry. So, to build a cube is necessary set 6 faces or oligonucleotides sequences. These parameters relate DNA ds, DNA ss (linkers) with the respective edges in each face of geometry. Example:
  face: 1 DNA / DS001A: 1- 18, SS001L: 19- 23 /
  Then, "DS001A: 1- 18" means: *'DS'* = DNA ds, *'001'* = number of edge and *'A'* = side of DNA ds, the numbers *'1- 18'* is the sequential number correspondent to sequence of DNA written in part 2, above. In this example the DNA ds start in nucleotide 1 "C" and finish in nucleotide 18 "G". After this, "SS001L: 19- 23", means: *'SS'* = DNA ss or linker, *'001'* represent number of linker and *'L'* means linker, the numbers *'19- 23'* represent the sequential number correspondent to sequence of DNA. In this case, the linker start in nucleotide 19 "T" and finish in nucleotide 23 "T".

The file parm_modelage.in is used to configure the minimization procedure of nanostructures in a nanocage/geometry, and consequently the parameters correspond to rotation and translation of helices/edges. If the user don't have idea about the parameters necessary to start minimization procedure or the number of parameter necessary to build a sequence related to a specific geometry, is possible modelling a standard nanocage setting the parameter "opt_automatic:" to 1 (true), as "opt_automatic: 1". In other hand, if this parameter is 0 (false) is necessary fill all parameters in this file.

To use automatic procedure is necessary fill the firsts parameters, n_interactions and fit_restriction.

Is recommended, to choose firstly the option 1 and after set it to 0 and modify or rewrite according to intent.

- *n_interactions*: this parameter set the number of interactions to realize the minimization procedure. (Integer number)

- *fit_restriction*: This value is used to restrict the fit minimization or stop the minimization. This value is generated by a cost function which measure total distance of all linker, DNA ss. (Real number > 0.0)

- *opt_automatic*: Use a set parameters predefined automatically (1) or manually (0). If the user not have a previous information about the correct parameters, change this parameter to 1 (true), as "opt_automatic: 1", then run PolygenDNA. After, is possible, change this parameter to 0 (false) and fill the bellow parameters. (Integer number, 0 or 1).

- *parameters*: The number of this parameter change according with of the geometry or number of edges. The column 1 correspond the number of edges, columns 2 to 4 correspond to rotation of each helix (DNA ds) around the correspondent edge in a specific face on geometry (each columns represent a dimension), the values are in degree. The column 5 correspond to values used to translate the center of mass of each DNA ds referent the initial position. This value is in Angstroms. (Real value).

- *restriction_min*: Is the minimum value allowed to use in *parameters*, each column and row correspond exactly at a *parameter*. (Real number)

- *restriction_max*: Is the same that *restriction_min* but correspond to maximum. (Real number)

- *maska*: Is the logical value that enable to use a specific *parameters*. Each column and row correspond exactly at a *parameter*. If the value is T (true) the parameter will minimized but if is F (false) the parameter value not will change. (Logical)

- *temp_min*: Is final temperature of cooling. Is a number used in minimization procedure, simulated annealing, to determine the final level of randomization of each column in *parameter*. (Real number)

- *temp_max*: Is start temperature of cooling. Is a number used in minimization procedure, simulated annealing, to determine the start level of randomization of each column in *parameter*. (Real number)

Chapter 3

EXAMPLES

3.1 Octahedron

File *configure.dat*

Geometries: **********

number: 2

len_linkers: 5

kind_linker_model: 1 #1 Polygen standard or #2 external (linker_input.pdb)

min_dist_steric: 0.6 R_expand: 00

FACES_SEQUENCES:

seq_automatic: 0

n_sequences: 8

max_residues: 69

face: 1: /CTTAGAGTTGCCACCAGGTTTTTCGATGTCTAAGCTGACCGTTTTT
GGACCGTGATTCCATGACTTTTT/

face: 2: /CCTGGCACTAAGGTACTGTTTTTCGGTCAGCTTAGACATCGTTTTT
GAATCCTATGCTCGGACGTTTTT/

face: 3: /GCCAGTCGAATCTGTAGCTTTTTCTATCCGATCGAGGCATGTTTTT

CATACTGAGAGCGTTCCGTTTTT/

face: 4: /GCTACAGATTCGACTGGCTTTTTCATGCCTCGATCGGATAGTTTTT
GATAGGCTAGCTCCGTACTTTTT/

face: 5: /CTTAGGATACGAGCCTGCTTTTTCGGTTACGGTACAATGCCTTTTT
CGCAAGACGTTAGTGTCCTTTTT/

face: 6: /GTATGACTCTCGCAAGGCTTTTTGGCATTGTACCGTAACCGTTTTT
GCCAATGCCATGTTACGGTTTTT/

face: 7: /CCGTAACATGGCATTGGCTTTTTGTACGGAGCTAGCCTATCTTTTT
CCTGGTGGCAACTCTAAGTTTTT/

face: 8: /GGACCACCGTTGAGATTCTTTTTGAATCTCAACGGTGGTCCTTTTT
GCGTTCTGCAATCACAGGTTTTT/

FACES_STENCIL_SEQUENCES:

face: 1 DNA / DS001A: 1- 18, SS001L: 19- 23, DS002A: 24- 41, SS002L: 42- 46, DS003A: 47- 64, SS003L: 65- 69 /

face: 2 DNA / DS003B: 70- 87, SS004L: 88- 92, DS004A: 93- 110, SS005L: 111- 115, DS005A: 116- 133, SS006L: 134- 138 /

face: 3 DNA / DS004B: 139- 156, SS007L: 157- 161, DS006A: 162- 179, SS008L: 180- 184, DS007A: 185- 202, SS009L: 203- 207 /

face: 4 DNA / DS002B: 208- 225, SS010L: 226- 230, DS008A: 231- 248, SS011L: 249- 253, DS006B: 254- 271, SS012L: 272- 276 /

face: 5 DNA / DS005B: 277- 294, SS013L: 295- 299, DS009A: 300- 317, SS014L: 318- 322, DS010A: 323- 340, SS015L: 341- 345 /

face: 6 DNA / DS007B: 346- 363, SS016L: 364- 368, DS011A: 369- 386, SS017L: 387- 391, DS009B: 392- 409, SS018L: 410- 414 /

face: 7 DNA / DS011B: 415- 432, SS019L: 433- 437, DS008B: 438- 455, SS020L: 456- 460, DS012A: 461- 478, SS021L: 479- 483 /

face: 8 DNA / DS012B: 484- 501, SS022L: 502- 506, DS001B: 507- 524, SS023L: 525- 529, DS010B: 530- 547, SS024L: 548- 552 /

File *parm_ modelage.in*

n_interactions: 300

fit_restriction: 1.00

opt_automatic: 1

n_parameters: 12

parameters:

1 0.0000000E+000 0.0000000E+000 0.0000000E+000 0.0000000E+000

2 0.0000000E+000 0.0000000E+000 0.0000000E+000 0.0000000E+000

3 0.0000000E+000 0.0000000E+000 0.0000000E+000 0.0000000E+000

4 0.0000000E+000 0.0000000E+000 0.0000000E+000 0.0000000E+000

5 0.0000000E+000 0.0000000E+000 0.0000000E+000 0.0000000E+000

6 0.0000000E+000 0.0000000E+000 0.0000000E+000 0.0000000E+000

7 0.0000000E+000 0.0000000E+000 0.0000000E+000 0.0000000E+000

8 0.0000000E+000 0.0000000E+000 0.0000000E+000 0.0000000E+000

9 0.0000000E+000 0.0000000E+000 0.0000000E+000 0.0000000E+000

10 0.0000000E+000 0.0000000E+000 0.0000000E+000 0.0000000E+000

11 0.0000000E+000 0.0000000E+000 0.0000000E+000 0.0000000E+000

12 0.0000000E+000 0.0000000E+000 0.0000000E+000 0.0000000E+000

restriction_min:

1 -9.0000000E+001 -1.1000001E+001 -1.1000001E+001 -1.0000000E+001

2 -9.0000000E+001 -1.1000001E+001 -1.1000001E+001 -1.0000000E+001

3 -9.0000000E+001 -1.1000001E+001 -1.1000001E+001 -1.0000000E+001

4 -9.0000000E+001 -1.1000001E+001 -1.1000001E+001 -1.0000000E+001

5 -9.0000000E+001 -1.1000001E+001 -1.1000001E+001 -1.0000000E+001

6 -9.0000000E+001 -1.1000001E+001 -1.1000001E+001 -1.0000000E+001

7 -9.0000000E+001 -1.1000001E+001 -1.1000001E+001 -1.0000000E+001

8 -9.0000000E+001 -1.1000001E+001 -1.1000001E+001 -1.0000000E+001

9 -9.0000000E+001 -1.1000001E+001 -1.1000001E+001 -1.0000000E+001

10 -9.0000000E+001 -1.1000001E+001 -1.1000001E+001 -1.0000000E+001

11 -9.0000000E+001 -1.1000001E+001 -1.1000001E+001 -1.0000000E+001

12 -9.0000000E+001 -1.1000001E+001 -1.1000001E+001 -1.0000000E+001

restriction_max:

1 9.0000000E+001 1.1000001E+001 1.1000001E+001 1.0000000E+001

2 9.0000000E+001 1.1000001E+001 1.1000001E+001 1.0000000E+001

3 9.0000000E+001 1.1000001E+001 1.1000001E+001 1.0000000E+001

4 9.0000000E+001 1.1000001E+001 1.1000001E+001 1.0000000E+001

5 9.0000000E+001 1.1000001E+001 1.1000001E+001 1.0000000E+001

6 9.0000000E+001 1.1000001E+001 1.1000001E+001 1.0000000E+001

7 9.0000000E+001 1.1000001E+001 1.1000001E+001 1.0000000E+001

8 9.0000000E+001 1.1000001E+001 1.1000001E+001 1.0000000E+001

9 9.0000000E+001 1.1000001E+001 1.1000001E+001 1.0000000E+001

10 9.0000000E+001 1.1000001E+001 1.1000001E+001 1.0000000E+001

11 9.0000000E+001 1.1000001E+001 1.1000001E+001 1.0000000E+001

12 9.0000000E+001 1.1000001E+001 1.1000001E+001 1.0000000E+001

maska:

1 T T T T

2 T T T T

3 T T T T

4 T T T T

5 T T T T

6 T T T T

7 T T T T

8 T T T T

9 T T T T

10 T T T T

11 T T T T

12 T T T T

temp_min:

1 1.0000000E-001 1.0000000E-002 1.0000000E-001 1.0000000E+000

temp_max:

1 1.0000000E+000 1.0000000E-001 1.0000000E+000 1.0000000E+001

3.2 Build a Nanocage using spatial coordinates, Gift Wrapping algorithm

To make this kind of modelling the user need fill the file *positions.dat* with spatial coordinates. To use this file is necessary set, in file *configure.dat*, the parameters 'number: 116' and 'seq_automatic: 1' (if the user don't has idea of how configure the sequence, use 'seq_automatic: 1').

File *positions.dat*

50.000 , 50.000 , 50.000

50.000 , 50.000 , -50.000

50.000 , -50.000 , 50.000

50.000 , -50.000 , -50.000

-50.000 , 50.000 , 50.000

-50.000 , 50.000 , -50.000

-50.000 , -50.000 , 50.000

-50.000 , -50.000 , -50.000

Appendix A

Database structures of high symmetry

```
choose the number of the shape
 0  - tetrahedron
 1  - cube
 2  - octahedron
 3  - dodecahedron
 4  - icosahedron
 5  - triangular prism
 6  - pentagonal prism
 7  - hexagonal prism
 8  - octagonal prism
 9  - decagonal prism
 10  - square antiprism
 11  - pentagonal antiprism
 12  - hexagonal antiprism
 13  - octagonal antiprism
 14  - decagonal antiprism
```

```
15  - rhombic dodecahedron
16  - triakis octahedron
17  - tetrakis hexahedron
18  - trapezoidal icositetrahedron
19  - hexakis octahedron
20  - pentagonal icositetrahedron (dextro)
21  - rhombic triacontahedron
22  - triakis icosahedron
23  - pentakis dodecahedron
24  - trapezoidal hexecontahedron
25  - hexakis icosahedron
26  - pentagonal hexecontahedron (dextro)
27  - square pyramid (J1)
28  - pentagonal pyramid (J2)
29  - triangular cupola (J3)
30  - square cupola (J4)
31  - pentagonal cupola (J5)
32  - pentagonal rotunda (J6)
33  - elongated triangular pyramid (J7)
34  - elongated square pyramid (J8)
35  - elongated pentagonal pyramid (J9)
36  - gyroelongated square pyramid (J10)
37  - gyroelongated pentagonal pyramid (J11)
38  - triangular dipyramid (J12)
39  - pentagonal dipyramid (J13)
40  - elongated triangular dipyramid (J14)
41  - elongated square dipyramid (J15)
```

42 - elongated pentagonal dipyramid (J16)

43 - gyroelongated square dipyramid (J17)

44 - elongated triangular cupola (J18)

45 - elongated square cupola (J19)

46 - elongated pentagonal cupola (J20)

47 - elongated pentagonal rotunds (J21)

48 - gyroelongated triangular cupola (J22)

49 - gyrobifastigium (J26)

50 - triangular orthobicupola (J27)

51 - square orthobicupola (J28)

52 - square gyrobicupola (J29)

53 - pentagonal orthobicupola (J30)

54 - pentagonal gyrobicupola (J31)

55 - pentagonal orthocupolarontunda (J32)

56 - pentagonal gyrocupolarotunda (J33)

57 - pentagonal orthobirotunda (J34)

58 - elongated triangular orthobicupola (J35)

59 - elongated triangular gyrobicupola (J36)

60 - elongated square gyrobicupola (J37)

61 - elongated pentagonal orthobicupola (J38)

62 - elongated pentagonal gyrobicupola (J39)

63 - elongated pentagonal orthocupolarotunda (J40)

64 - elongated pentagonal gyrocupolarotunda (J41)

65 - elongated pentagonal orthobirotunda (J42)

66 - elongated pentagonal gyrobirotunda (J43)

67 - gyroelongated triangular bicupola (J44)

68 - gyroelongated square bicupola (J45)

```
69  - gyroelongated pentagonal bicupola (J46)
70  - gyroelongated pentagonal cupolarotunda (J47)
71  - gyroelongated pentagonal birotunda (J48)
72  - augmented triangular prism (J49)
73  - biaugmented triangular prism (J50)
74  - triaugmented triangular prism (J51)
75  - augmented pentagonal prism (J52)
76  - biaugmented pentagonal prism (J53)
77  - augmented hexagonal prism (J54)
78  - parabiaugmented hexagonal prism (J55)
79  - metabiaugmented hexagonal prism (J56)
80  - triaugmented hexagonal prism (J57)
81  - augmented dodecahedron (J58)
82  - parabiaugmented dodecahedron (J59)
83  - metabiaugmented dodecahedron (J60)
84  - triaugmented dodecahedron (J61)
85  - metabidiminished icosahedron (J62)
86  - tridiminished icosahedron (J63)
87  - augmented tridiminished icosahedron (J64)
88  - augmented truncated tetrahedron (J65)
89  - augmented truncated cube (J66)
90  - biaugmented truncated cube (J67)
91  - augmented truncated dodecahedron (J68)
92  - parabiaugmented truncated dodecahedron (J69)
93  - metabiaugmented truncated dodecahedron (J70)
94  - triaugmented truncated dodecahedron (J71)
95  - gyrate rhombicosidodecahedron (J72)
```

```
96  - parabigyrate rhombicosidodecahedron (J73)
97  - metabigyrate rhombicosidodecahedron (J74)
98  - trigyrate rhombicosidodecahedron (J75)
99  - diminished rhombicosidodecahedron (J76)
100  - paragyrate diminished rhombicosidodecahedron (J77)
101  - metagyrate diminished rhombicosidodecahedron (J78)
102  - bigyrate diminished rhombicosidodecahedron (J79)
103  - parabidiminished rhombicosidodecahedron (J80)
104  - metabidiminished rhombicosidodecahedron (J81)
105  - gyrate bidiminished rhombicosidodecahedron (J82)
106  - tridiminished rhombicosidodecahedron (J83)
107  - snub disphenoid (J84)
108  - snub square antiprism (J85)
109  - sphenocorona (J86)
110  - augmented sphenocorona (J87)
111  - sphenomegacorona (J88)
112  - hebesphenomegacorona (J89)
113  - disphenocingulum (J90)
114  - bilunabirotunda (J91)
115  - triangular hebesphenorotunda (J92)
116  - Gift wrapping algorithm. To use with 'file positions.dat'
```