E

Code

> Computers help us to solve problems we never had before they came along.
>
> Anonymous

In this appendix, a brief description on how to use the code used along this work to obtain the numerical results is detailed. Almost all the code is written in standard `FORTRAN 90`, and make use of the library `afnl` (a free software `FORTRAN 90` library). The source code, and manual of this library can be found in the homepage of it's `sourceforge` project.

`http://sourceforge.net/projects/afnl`

All the software described below is `free software`, distributed under the terms of the `GPL`[1] license. This means, among other things that your are allowed to run the software, see the source code, copy it any number of times that you want, and even improve the software, and make this improvements available to the public. For more details take a look at the full license. You can download all the source code of the detailed software at

`http://lattice.ft.uam.es/perpag/alberto/codigo_en.php`

E.1 `omega`

E.1.1 Description

This code computes the sums

$$\mathcal{S}(n,n',\vec{k}) = \xi^{|n-n'|/2} \sum_{j=0}^{m} \frac{(-)^j \sqrt{n!n'!}\xi^j}{j!(m-j)!(j+|n-n'|)!} \tag{E.1}$$

where $m = \min(n,n')$, with arbitrary precision float point arithmetics. This code uses the `GNU gmp` library to perform arbitrary precision arithmetics.

[1] The full text of the license can be found at `http://www.gnu.org/copyleft/gpl.html`

E.1.2 Arguments

`n:` Minimum value of n and n'. Default 0.

`N:` Maximum value of n and n'. Default 100.

`k:` Minimum value of k_1 and k_2. Default 0.

`K:` Maximum value of k_1 and k_2. Default 10.

`t:` Value of the aspect ratio l_2/l_1. Default 1.0

`q:` The first chern class q (flux number). Default 1.

`p:` The precision. Default 1000.

E.1.3 Example

Using the command

```
[14] user@computer:~/prg/omega % omega.out -n 40 -N 42 -k 0 -K 1 -t 1.0 -q 1
```

we obtain the values $\mathcal{S}(n, n', \vec{k})$ for $40 \leq n, n' \leq 42$ and $0 \leq k_1, k_2 \leq 1$ in the following way:

```
40   40    0    0 1.0000000000000000000e+00
40   40    0    1 -7.7681347206489180032e-01
40   40    1    1 2.8840244766694697793e+00
40   41    0    0 0.0000000000000000000e+00
40   41    0    1 1.0833309826018148257e-01
40   41    1    1 -9.5764280031266865804e-01
40   42    0    0 0.0000000000000000000e+00
40   42    0    1 8.0656968655000805201e-01
40   42    1    1 -3.2788328895737988214e+00
41   41    0    0 1.0000000000000000000e+00
41   41    0    1 -8.0680124210834288140e-01
41   41    1    1 3.2589124893410681428e+00
41   42    0    0 0.0000000000000000000e+00
41   42    0    1 -1.1362091396207595007e-01
41   42    1    1 3.1431227688923508997e-01
42   42    0    0 1.0000000000000000000e+00
42   42    0    1 -7.7572642583850488570e-01
42   42    1    1 3.1373424286059177338e+00
```

E.2 `pert.out`

E.2.1 Description

This code computes the Fourier series of the functions h, solution of the vortex equation

$$\Delta h = \frac{1}{2}\left(1 - e^{-2h}|\chi|^2\right) \tag{E.2}$$

(for more details, see chapters 4 and 5). As input this program needs the flux number, the number of orders in the bradlow parameter expansion that you want to compute, the number of Fourier modes that you want to compute, a directory to save the computed data, and the positions of the zeros of the Higgs field. A flag is used to know if the computation has to be started or continued from some order.

E.2.2 Input

A typical input file for this has the following format

```
Nstart
q, Norders, Nfourier, Taui, dir
zero1_X, zero1_Y
zero2_X, zero2_Y
zero3_X, zero3_Y
.
.
.
```

where:

`Nstart:` Integer. Where the computation has to be started. If `Nstart` is 0 or 1, the computation is started from the beginning. In any other case the computation starts in this order. If the computation does not starts from the beginning, the program needs the result of the computed data up to order `Nstart`, *with the rest of the input exactly the same as in the previous computation.*

`q:` Integer. The flux number.

`Norders:` Integer. The number of orders that you want to compute. The maximum is 51.

`Nfourier:` Integer. The number of Fourier modes that you compute in the convolutions.

`Taui:` Real. The aspect ratio of the torus sizes (value of the quotient l_2/l_1).

`dir:` Character string. the name of an existing directory where the output will be saved.

Zeros of the Higgs field: Real. The position of $q-1$ zeros of the Higgs field follow, the position of the remaining zero is computed automatically, so that the centre of mass of the zeros is in the correct position. The position of the zeros is specified *relative* to the size of the torus, and the origin is situated in the centre of the torus.

E.2.3 Output

In the specified directory, we get the following output files:

`hf:O=<oo>:Nterm=<nn>:flux=<q>.dat:` The Fourier series of the function h. `<oo>` is a two digit integer specifying the order, `<nn>` is also a two digit integer that specifies the number of Fourier modes, and `<q>` is a one digit integer specifying the flux.

`deltaf:O=<oo>:Nterm=<nn>:flux=<q>.dat:` The Fourier series of the function Δh. `<oo>` is a two digit integer specifying the order, `<nn>` is also a two digit integer that specifies the number of Fourier modes, and `<q>` is a one digit integer specifying the flux.

`Pf:O=<oo>:Nterm=<nn>:flux=<q>.dat:` The Fourier series of the function e^{-2h}. `<oo>` is a two digit integer specifying the order, `<nn>` is also a two digit integer that specifies the number of Fourier modes, and `<q>` is a one digit integer specifying the flux.

`factor:O=<oo>:Nterm=<nn>:flux=<q>.dat:` The value of $e^{-2h_{00}}$, where h_{00} is the $\vec{0}$ Fourier mode of h. `<oo>` is a two digit integer specifying the order, `<nn>` is also a two digit integer that specifies the number of Fourier modes, and `<q>` is a one digit integer specifying the flux.

`summary.dat:` A text file with a small summary of the computed data: flux number, positions of the zeros, number of computed Fourier modes, time of the computation, etc...

E.2.4 Example

A small example of the usage of the program:

```
[14] user@computer:~/prg/pert % cat input.dat
0
2, 2, 12, 1.0, data
0.0, 0.25
[15] user@computer:~/prg/pert % cat input.dat| ./pert.out

 Stage 1: Reading input and needed data:
 =======================================
 Flux (a.k.a q, first chern class):    2
 Orders to compute:       2
 Number of terms used for the Fourier series:   12
 Aspect ratio of the Torus (l2/l1): 1.0000E+00

 Positions of the zeros (relative to the center of the Torus):
  0.00000000000000000000000000E+00  2.50000000000000000000000000E-01
  0.00000000000000000000000000E+00 -2.50000000000000000000000000E-01


 Stage 2: Computing h, Factor:
 ==============================
Order:     2 done
[16] user@computer:~/prg/pert % ls data
deltaf:O=01:Nterm=12:flux=2.dat  hf:O=01:Nterm=12:flux=2.dat
deltaf:O=02:Nterm=12:flux=2.dat  hf:O=02:Nterm=12:flux=2.dat
factor:O=01:Nterm=12:flux=2.dat  Pf:O=02:Nterm=12:flux=2.dat
factor:O=02:Nterm=12:flux=2.dat  summary.dat
[19] alberto@computer:~/prg/pert % cat input2.dat
2
```

```
2, 5, 12, 1.0, data
0.0, 0.25
[20] user@computer:~/prg/pert % cat input2.dat | ./pert.out

 Stage 1: Reading input and needed data:
 =======================================
 Flux (a.k.a q, first chern class):    2
 Orders to compute:       5
 Number of terms used for the Fourier series:   12
 Aspect ratio of the Torus (l2/l1): 1.0000E+00

 Positions of the zeros (relative to the center of the Torus):
  0.00000000000000000000000000E+00  2.50000000000000000000000000E-01
  0.00000000000000000000000000E+00 -2.50000000000000000000000000E-01

Reading order:     2

 Stage 2: Computing h, Factor:
 ==============================
Order:     3 done
Order:     4 done
Order:     5 done
[21] user@computer:~/prg/pert % ls data
deltaf:O=01:Nterm=12:flux=2.dat  hf:O=01:Nterm=12:flux=2.dat
deltaf:O=02:Nterm=12:flux=2.dat  hf:O=02:Nterm=12:flux=2.dat
deltaf:O=03:Nterm=12:flux=2.dat  hf:O=03:Nterm=12:flux=2.dat
deltaf:O=04:Nterm=12:flux=2.dat  hf:O=04:Nterm=12:flux=2.dat
deltaf:O=05:Nterm=12:flux=2.dat  hf:O=05:Nterm=12:flux=2.dat
factor:O=01:Nterm=12:flux=2.dat  Pf:O=02:Nterm=12:flux=2.dat
factor:O=02:Nterm=12:flux=2.dat  Pf:O=03:Nterm=12:flux=2.dat
factor:O=03:Nterm=12:flux=2.dat  Pf:O=04:Nterm=12:flux=2.dat
factor:O=04:Nterm=12:flux=2.dat  Pf:O=05:Nterm=12:flux=2.dat
factor:O=05:Nterm=12:flux=2.dat  summary.dat
[22] user@computer:~/prg/pert % cat data/summary.dat
#####################################################
# SUMMARY of computed data. Mon Jun 23 16:47:14 2007
#
# Flux (a.k.a q, first chern class):    2
# Orders computed:       2
# Number of terms used for the Fourier series:   12
# Aspect ratio of the Torus (l2/l1): 1.0000E+00

# Positions of the zeros (relative to the center of the Torus):
  0.00000000000000000000000000E+00  2.50000000000000000000000000E-01
  0.00000000000000000000000000E+00 -2.50000000000000000000000000E-01

# Values of C (position in moduli space in the coordinates of CP):
```

```
#  -7.0710678118655523327618084E-01 -9.3755900023106702411485953E-17
#  -7.0710678118653991219844102E-01 -1.6603091684753279075317654E-16
#####################################################
#
# Times of the computation:
#
Order:     2 done (1.80E-02sec)
#
# END of run. Mon Jun 23 16:47:14 2007
#
#####################################################
# SUMMARY of computed data. Mon Jun 23 16:48:34 2007
#
# Flux (a.k.a q, first chern class):    2
# Orders computed:       5
# Number of terms used for the Fourier series:   12
# Aspect ratio of the Torus (l2/l1): 1.0000E+00

# Positions of the zeros (relative to the center of the Torus):
  0.000000000000000000000000E+00  2.500000000000000000000000E-01
  0.000000000000000000000000E+00 -2.500000000000000000000000E-01

# Values of C (position in moduli space in the coordinates of CP):
#  -7.0710678118655523327618084E-01 -9.3755900023106702411485953E-17
#  -7.0710678118653991219844102E-01 -1.6603091684753279075317654E-16
#####################################################
#
# Times of the computation:
#
Reading order:     2(1.00E-02sec)
Order:     3 done (2.20E-02sec)
Order:     4 done (3.20E-02sec)
Order:     5 done (4.70E-02sec)
#
# END of run. Mon Jun 23 16:48:34 2007
#
```

E.3 metric.out

E.3.1 Description

This code computes the metric in the space of solutions of the Bogomolny equations. For more details, see chapter 7. As input this program needs the flux number, the number of orders in the bradlow parameter expansion that you want to compute, the number of Fourier modes that you want to compute, a directory to save the computed data, and the positions of the zeros of the Higgs field. The computation of the metric has two different parts, and the first one requires the computation of exactly the same quantities than to solve the Bogomolny

equations, that is what the program `pert` (E.2) does. This code can compute all the data from scratch of use previously computed data. Two flags are used to know if the computation has to be started or continued.

E.3.2 Input

A typical input file for this has the following format

```
Nstart1, Nstart2
q, Norders, Nfourier, dir
zero1_X, zero1_Y
zero2_X, zero2_Y
zero3_X, zero3_Y
.
.
.
```

where:

`Nstart1:` Integer. Where the first part of the computation has to be started. If `Nstart1` is 0 or 1, the computation is started from the beginning. In any other case the computation starts in this order. If the computation does not starts from the beginning, the program needs the result of the computed data up to order `Nstart`, *with the rest of the input exactly the same as in the previous computation.* In this case these previously computed data can be the ones computed with the program `pert`.

`Nstart2:` Integer. Where the second part of the computation has to be started. If `Nstart2` is 0 or 1, the computation is started from the beginning. In any other case the computation starts in this order. If the computation does not starts from the beginning, the program needs the result of the computed data up to order `Nstart`, *with the rest of the input exactly the same as in the previous computation.*

`q:` Integer. The flux number.

`Norders:` Integer. The number of orders that you want to compute. The maximum is 51.

`Nfourier:` Integer. The number of Fourier modes that you compute in the convolutions.

`dir:` Character string. the name of an existing directory where the output will be saved.

Zeros of the Higgs field: Real. The position of $q-1$ zeros of the Higgs field follow, the position of the remaining zero is computed automatically, so that the centre of mass of the zeros is in the correct position. The position of the zeros is specified *relative* to the size of the torus, and the origin is situated in the centre of the torus.

E.3.3 Output

In the specified directory, we get the following output files:

`hf:O=<oo>:Nterm=<nn>:flux=<q>.dat:` The Fourier series of the function h. `<oo>` is a two digit integer specifying the order, `<nn>` is also a two digit integer that specifies the number of Fourier modes, and `<q>` is a one digit integer specifying the flux.

`deltaf:O=<oo>:Nterm=<nn>:flux=<q>.dat:` The Fourier series of the function Δh. `<oo>` is a two digit integer specifying the order, `<nn>` is also a two digit integer that specifies the number of Fourier modes, and `<q>` is a one digit integer specifying the flux.

`Pf:O=<oo>:Nterm=<nn>:flux=<q>.dat:` The Fourier series of the function e^{-2h}. `<oo>` is a two digit integer specifying the order, `<nn>` is also a two digit integer that specifies the number of Fourier modes, and `<q>` is a one digit integer specifying the flux.

`hdotf:<i>:O=<oo>:Nterm=<nn>:flux=<q>.dat:` The Fourier series of the function $\mathcal{H}_i$. `<i>` is a one digit integer specifying which function, `<oo>` is a two digit integer specifying the order, `<nn>` is also a two digit integer that specifies the number of Fourier modes, and `<q>` is a one digit integer specifying the flux.

`factor:O=<oo>:Nterm=<nn>:flux=<q>.dat:` The value of $e^{-2h_{00}}$, where h_{00} is the $\vec{0}$ Fourier mode of h. `<oo>` is a two digit integer specifying the order, `<nn>` is also a two digit integer that specifies the number of Fourier modes, and `<q>` is a one digit integer specifying the flux.

`Bf:<i>:O=<oo>:Nterm=<nn>:flux=<q>.dat:` The value of $\mathcal{H}_i(\vec{0})$, the $\vec{0}$ Fourier mode of $\mathcal{H}_i$. `<i>` is a one digit integer specifying which function, `<oo>` is a two digit integer specifying the order, `<nn>` is also a two digit integer that specifies the number of Fourier modes, and `<q>` is a one digit integer specifying the flux.

`metric:O=<oo>:Nterm=<nn>:flux=<q>.dat:` Complex. The metric. The metric is as q complex numbers with format `ES33.25`. An example to read the stored data and save it in the two dimensional array `g(:,:)` is:

```
Open (Unit = 99, File = 'metric:O=34:Nterm=16:flux=2.dat')
Do I = 1, q
  Read(99,'(100ES33.25)')(g(I,J), J = 1, q)
End Do
Close (99)
```

The name conventions for the files are the same as before: `<oo>` is a two digit integer specifying the order, `<nn>` is also a two digit integer that specifies the number of Fourier modes, and `<q>` is a one digit integer specifying the flux.

`summary_metric.dat:` A text file with a small summary of the computed data: flux number, positions of the zeros, number of computed Fourier modes, time of the computation, etc...

E.3.4 Example

A small example that shows some possibilities of the program. It uses the output of the example detailed in (E.2):

```
[46] user@computer:~/prg/metric % cat point:0000.dat
5, 0
2, 7, 12,  data
0.0, 0.25
```

```
[47] user@computer:~/prg/metric % cat point:0000.dat | ./metric.out

 Stage 1: Reading input and needed data:
 ======================================
 Flux (a.k.a q, first chern class):    2
 Orders to compute:       7
 Number of terms used for the Fourier series:   12

 Positions of the zeros (relative to the center of the Torus):
  0.00000000000000000000000000E+00  2.50000000000000000000000000E-01
  0.00000000000000000000000000E+00 -2.50000000000000000000000000E-01

Reading order:     2
Reading order:     3
Reading order:     4
Reading order:     5

 Stage 2: Computing h, Factor:
 ============================
Order:     6 done
Order:     7 done

 Stage 3: Computing \dot H_i, B_i:
 ================================
Order:     2 done
Order:     3 done
Order:     4 done
Order:     5 done
Order:     6 done
Order:     7 done

 Stage 4: Computing the metric:
 =============================
Order:     1 done
Order:     2 done
Order:     3 done
Order:     4 done
Order:     5 done
Order:     6 done
Order:     7 done
[48] user@computer:~/prg/metric % cat point2:0000.dat
7, 7
2, 10, 12,  data
0.0, 0.25
[49] user@computer:~/prg/metric % cat point2:0000.dat | ./metric.out

 Stage 1: Reading input and needed data:
```

```
 ======================================
 Flux (a.k.a q, first chern class):    2
 Orders to compute:      10
 Number of terms used for the Fourier series:   12

 Positions of the zeros (relative to the center of the Torus):
  0.000000000000000000000000E+00  2.500000000000000000000000E-01
  0.000000000000000000000000E+00 -2.500000000000000000000000E-01

Reading order:     2
Reading order:     3
Reading order:     4
Reading order:     5
Reading order:     6
Reading order:     7

 Stage 2: Computing h, Factor:
 ==============================
Order:     8 done
Order:     9 done
Order:    10 done

 Stage 3: Computing \dot H_i, B_i:
 ==================================
Reading order:     2
Reading order:     3
Reading order:     4
Reading order:     5
Reading order:     6
Reading order:     7
Order:     7 done
Order:     8 done
Order:     9 done
Order:    10 done

 Stage 4: Computing the metric:
 ===============================
Order:     7 done
Order:     8 done
Order:     9 done
Order:    10 done
[51] user@computer:~/prg/metric % cat data/summary_metric.dat
#
# SUMMARY of computed data. Mon Jan 15 12:20:55 2007
#
# Flux (a.k.a q, first chern class):    2
# Orders computed:      10
```

```
# Number of terms used for the Fourier series:   12

# Positions of the zeros (relative to the center of the Torus):
  0.000000000000000000000000E+00  2.500000000000000000000000E-01
  0.000000000000000000000000E+00 -2.500000000000000000000000E-01

# Values of C (position in moduli space in the coordinates of CP):
#  -7.071067811865552332761808 4E-01 -9.375590002310670241148595 3E-17
#  -7.071067811865399121984410 2E-01 -1.660309168475327907531765 4E-16
#
# Times of the computation (PART I):
#
Reading order:     2(2.80E-02sec)
Reading order:     3(2.20E-02sec)
Reading order:     4(1.40E-02sec)
Reading order:     5(8.00E-03sec)
Reading order:     6(7.00E-03sec)
Reading order:     7(9.00E-03sec)
Order:     8 done (2.35E-01sec)
Order:     9 done (3.43E-01sec)
Order:    10 done (5.04E-01sec)
#
# Times of the computation (PART II):
#
Reading order:     2(4.00E-03sec)
Reading order:     3(5.00E-03sec)
Reading order:     4(5.00E-03sec)
Reading order:     5(5.00E-03sec)
Reading order:     6(5.00E-03sec)
Reading order:     7(5.00E-03sec)
Order:     7 done (1.24E-01sec)
Order:     8 done (1.37E-01sec)
Order:     9 done (1.52E-01sec)
Order:    10 done (1.68E-01sec)
#
# Times of the computation (PART III):
#
Order:     7 done (5.70E-01sec)
Order:     8 done (6.57E-01sec)
Order:     9 done (7.42E-01sec)
Order:    10 done (8.28E-01sec)
```

E.4 `ext.out`

E.4.1 Description

This code computes the solutions of the extended abelian Higgs model. For more details, see section 5.3 for more details. As input this program needs the flux number, the number of components of the Higgs field in internal space, the number of orders in the bradlow parameter expansion that you want to compute, the number of Fourier modes that you want to compute, a directory to save the computed data, and the position in the moduli space of the solution that you want to see. This code is able to continue a previous computation. A flag is used to determine if the computation has to be started or continued.

E.4.2 Input

A typical input file for this has the following format

```
Nstart
q, Ncolors, Norders, Nfourier, tau, dir
(c^(1)_1), (c^(1)_2), ...
(c^(2)_1), (c^(2)_2), ...
.
.
.
```

where:

`Nstart`: Integer. Where the computation has to be started. If `Nstart` is 0 or 1, the computation is started from the beginning. In any other case the computation starts in this order. If the computation does not starts from the beginning, the program needs the result of the computed data up to order `Nstart`, *with the rest of the input exactly the same as in the previous computation.*

`q`: Integer. The flux number.

`Ncolors`: Integer. The number of components of the Higgs field (in internal space).

`Norders`: Integer. The number of orders that you want to compute. The maximum is 51.

`Nfourier`: Integer. The number of Fourier modes that you compute in the convolutions.

`tau`: Real. The aspect ratio of the Torus (l_2/l_1).

`dir`: Character string. the name of an existing directory where the output will be saved.

$c_i^{(a)}$: Complex. The matrix that determines the position in the moduli space. See equation (4.52).

E.4.3 Output

In the specified directory, we get the following output files:

`hf:O=<oo>:Nterm=<nn>:flux=<q>.dat:` The Fourier series of the function h. `<oo>` is a two digit integer specifying the order, `<nn>` is also a two digit integer that specifies the number of Fourier modes, and `<q>` is a one digit integer specifying the flux.

`deltaf:O=<oo>:Nterm=<nn>:flux=<q>.dat:` The Fourier series of the function Δh. `<oo>` is a two digit integer specifying the order, `<nn>` is also a two digit integer that specifies the number of Fourier modes, and `<q>` is a one digit integer specifying the flux.

`Pf:O=<oo>:Nterm=<nn>:flux=<q>.dat:` The Fourier series of the function e^{-2h}. `<oo>` is a two digit integer specifying the order, `<nn>` is also a two digit integer that specifies the number of Fourier modes, and `<q>` is a one digit integer specifying the flux.

`hdotf:<i>:O=<oo>:Nterm=<nn>:flux=<q>.dat:` The Fourier series of the function $\mathcal{H}_i$. `<i>` is a one digit integer specifying which function, `<oo>` is a two digit integer specifying the order, `<nn>` is also a two digit integer that specifies the number of Fourier modes, and `<q>` is a one digit integer specifying the flux.

`factor:O=<oo>:Nterm=<nn>:flux=<q>.dat:` The value of $e^{-2h_{00}}$, where h_{00} is the $\vec{0}$ Fourier mode of h. `<oo>` is a two digit integer specifying the order, `<nn>` is also a two digit integer that specifies the number of Fourier modes, and `<q>` is a one digit integer specifying the flux.

`summary.dat:` A text file with a small summary of the computed data: flux number, positions of the zeros, number of computed Fourier modes, time of the computation, etc...

E.4.4 Example

A small example that shows some possibilities of the program.

```
[46] user@computer:~/prg/ext % cat input.dat
0
2, 2, 2, 16, 1.0, data2
(1.0, 0.0), (0.0,0.0)
(0.0, 0.0), (1.0,0.0)
[47] user@computer:~/prg/ext % cat input.dat | ./ext.out

 Stage 1: Reading input and needed data:
 =======================================
 Flux (a.k.a q, first chern class):    2
 Orders to compute:       2
 Number of terms used for the Fourier series:   16
 Value of alpha (Units of pi): 0.000000E+00
 Value of tau: 1.000000E+00
 Positions of the zeros (relative to the center of the Torus):
  5.0000000000000000000000000E-01  0.0000000000000000000000000E+00  0.00000000000000000000000
  0.0000000000000000000000000E+00  0.0000000000000000000000000E+00  5.00000000000000000000000
```

```
 Stage 2: Computing h, Factor:
 ==============================
Order:     2 done
[48] user@computer:~/prg/ext % cat input.dat
2
2, 2, 5, 16, 1.0, data2
(1.0, 0.0), (0.0,0.0)
(0.0, 0.0), (1.0,0.0)
[49] user@computer:~/prg/ext % cat input.dat | ./ext.out

 Stage 1: Reading input and needed data:
 ======================================
 Flux (a.k.a q, first chern class):    2
 Orders to compute:       5
 Number of terms used for the Fourier series:   16
 Value of alpha (Units of pi): 0.000000E+00
 Value of tau: 1.000000E+00
 Positions of the zeros (relative to the center of the Torus):
  5.000000000000000000000000E-01  0.000000000000000000000000E+00  0.0000000000000000000
  0.000000000000000000000000E+00  0.000000000000000000000000E+00  5.0000000000000000000

Reading order:     2

 Stage 2: Computing h, Factor:
 ==============================
Order:     3 done
Order:     4 done
Order:     5 done
[51] user@computer:~/prg/ext % cat data/summary.dat
######################################################
# SUMMARY of computed data. Mon Jun 23 18:43:52 2007
#
# Flux (a.k.a q, first chern class):    2
# Orders computed:       2
# Number of terms used for the Fourier series:   16
# Value of alpha (Units of pi): 0.000000E+00
# Value of tau: 1.000000E+00
#
 Positions of the zeros (relative to the center of the Torus):
  5.000000000000000000000000E-01  0.000000000000000000000000E+00  0.0000000000000000000
00000000000000000000E+00
  0.000000000000000000000000E+00  0.000000000000000000000000E+00  5.0000000000000000000
00000000000000000000E+00
######################################################
#
```

```
# Times of the computation:
#
Order:     2 done (4.00E-02sec)
#
# END of run. Mon Jun 23 18:43:52 2007
#
#####################################################
# SUMMARY of computed data. Mon Jun 23 18:45:07 2007
#
# Flux (a.k.a q, first chern class):    2
# Orders computed:       5
# Number of terms used for the Fourier series:   16
# Value of alpha (Units of pi): 0.000000E+00
# Value of tau: 1.000000E+00
#
 Positions of the zeros (relative to the center of the Torus):
  5.000000000000000000000000E-01  0.000000000000000000000000E+00  0.000000000000000000000000
000000000000000000E+00
  0.000000000000000000000000E+00  0.000000000000000000000000E+00  5.000000000000000000000000
000000000000000000E+00
#####################################################
#
# Times of the computation:
#
Reading order:     2(2.10E-02sec)
Order:     3 done (5.40E-02sec)
Order:     4 done (8.50E-02sec)
Order:     5 done (1.27E-01sec)
#
# END of run. Mon Jun 23 18:45:08 2007
#
```