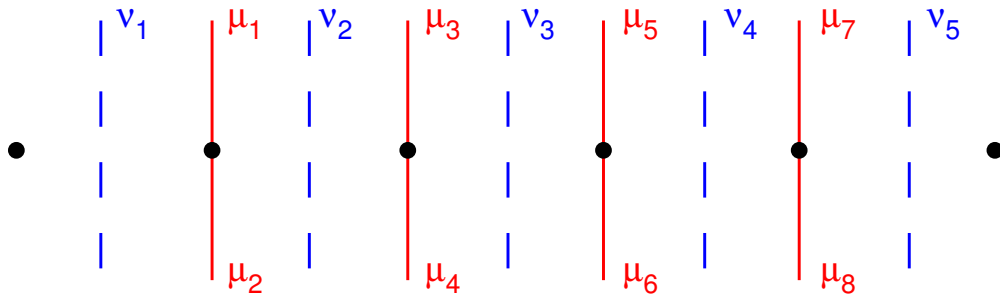


# Dynn program documentation (version 2.3)

June 2018

## 1 Dynnikov coordinate system

The Dynnikov coordinate system on the space of measured foliations on an  $n$ -punctured disk ( $n \geq 3$ ) is used. For example, for a 6-punctured disk the coordinates are  $(a_1, a_2, a_3, a_4, b_1, b_2, b_3, b_4) \in \mathbb{R}^8 \setminus \{0\}$ , and are defined by  $a_i = \frac{\mu_{2i} - \mu_{2i-1}}{2}$  and  $b_i = \frac{\nu_i - \nu_{i+1}}{2}$ , where the  $\mu_i$  and  $\nu_i$  are the measures of the corresponding arcs in the figure below.



When normalised, each coordinate is divided by the sum of the absolute values of all the coordinates. This therefore gives a real vector with entries whose absolute values sum to 1.

## 2 Program use

The program can be used in two ways:

`dynn`

which starts up an interactive session, and

`dynn filename`

which runs the commands in the given file, sending output to standard output.

The syntax of commands in both versions is the same. It is inflexible! All words in a given command must be on the same line and separated by spaces. In interactive mode, warnings are given about any commands which can't be interpreted. When processing a file, warnings are output about unknown commands to cerr, but the program continues, ignoring the command. On the other hand, known commands with incorrect syntax terminate the program.

Here are the commands:

**help** Displays a list of commands.

/ Comment. Note the comment symbol must be followed by a space.

/\* Multiline comment. Everything is ignored until the command \*/ is found. Note that these commands must appear at the start of lines and be followed by spaces.

**exit** Exits.

**read** The command `read infile outfile` processes the commands in the file `infile`, sending output to `outfile`. If `outfile` is omitted, output goes to the console in interactive mode, or to standard output in non-interactive mode. The files are in the current working directory unless full paths are specified.

**seed** The command `seed n` seeds the random number generator with  $n$  (to enable random braids to be repeated on each run of the file).

**set** Set a braid. (The command `braid` is synonymous.) There are several versions. Note that braid names can be any string not including spaces.

- `set b 4 1 2 -3`: sets  $b$  to be the 4-braid  $\sigma_1\sigma_2\sigma_3^{-1}$ .
- `set b horseshoe 10010`: sets  $b$  to be the braid corresponding to the given horseshoe orbit. `hs` is a permitted abbreviation.
- `set b predec m n w opt`: sets  $b$  to be the horseshoe braid of height  $m/n$  and decoration  $w$ , with symbol `opt` after the prefix. Note that this requires, but does not check, then  $m/n$  is *strictly* less than  $q_w$ . Use `e` for the empty decoration, and `NBT` for `NBT`.
- `set b beta m n`: sets  $b$  to be the braid  $\beta_{m,n}$ , e.g. with  $m = 3$  and  $n = 5$  will give  $\sigma_1\sigma_2\sigma_3\sigma_4^{-1}\sigma_5^{-1}\sigma_6^{-1}\sigma_7^{-1}\sigma_8^{-1}$ .
- `set b sigma m n`: sets  $b$  to be the braid  $\sigma_{m,n}$ , e.g. with  $m = 3$  and  $n = 5$  will give  $\sigma_1\sigma_2\sigma_3\sigma_3\sigma_2\sigma_1\sigma_1\sigma_2\sigma_3\sigma_4\sigma_5\sigma_6\sigma_7\sigma_8$ .
- `set b random 5 30`: sets  $b$  to a random 5-braid with 30 generators (no  $g -g$  pairs).
- `set b inverse c`: sets  $b$  to the inverse of braid  $c$ .
- `set b power c n`: sets  $b = c^n$ .
- `set b . c d` sets  $b$  to be the composition  $cd$ .
- `set b | c d`: sets  $b$  to the braid obtained by putting  $c$  and  $d$  disjointly side by side.
- `set b % c d`: sets  $b$  to  $d^{-1}cd$ .
- `set b ^ c d`: sets  $b$  to the stacked braid with  $c$  on the inside and  $d$  on the outside (if  $d$  isn't cyclic, then  $c$  only appears in the first tube).

**printbraid b** Prints the braid  $b$ .

**length b** Prints the number of generators in  $b$ .

**write whatever you like** Writes whatever you like to standard output.

**writeerr whatever you like** Writes whatever you like to standard error (useful for showing progress when standard output is redirected to a file).

**separator n** Writes a separating line consisting of n - signs. The default value of n is 80.

**do b** Processes the braid b according to the various options which have been set.

**dobraid 4 1 2 -3** sets a temporary braid and processes it as with **do**. Any of the above means of setting a braid can be used, for example **dobraid hs 10010**.

The options are of two types: boolean and integer. A boolean option can be set or unset by using (for example) **birman on** or **birman off**. An integer option is set using (for example) **accuracy 30**.

In fact there's one option of another type:

**convergeby** Specifies how the algorithm should decide that convergence should take place. There are 2 options, **convergeby coordinates**, and **convergeby cycle**. Convergence by coordinates iterates until the (normalized) coordinates of the measured foliations agree to the given accuracy. Convergence by cycle iterates until the approximate growth rates of successive iterates are very close to a periodic cycle of length less than the number of punctures. By default, having found such a cycle it takes the appropriate power of the braid and processes that (see the **dopower** option below). **Warning: the growth rate/entropy reported in this case will be of the appropriate power of the braid.**

In either case, behaviour is erratic for non-pA braids.

Here are the boolean options:

**birman** Use Birman braid convention. Default is off.

**regions** After a **do** command, the program attempts to find and describe all of the Dynnikov regions around the unstable foliation, together with the associated matrices.

**signs** Displays the signs of the Dynnikov coordinates at the end of each **do** command. Default is off.

**time** Displays the time in seconds taken to execute each **do** command. Timing includes any optional additions (e.g. calculating regions).

**onlysigns** In a **do** command, show *only* the signs of the coordinates. May interfere with other options which request display of certain things. Default is off.

**showinteger** At convergence, show the integer as well as normalized coordinates. Default is off.

**nomatrix** Don't show (or calculate) any matrix at the end of a **do** command. Default is off. This can save a great deal of time.

**nozero** Do not set coordinates within accuracy decimal places of 0 to 0. Default is off.

**showgrowths** Show a list of growth rates throughout the algorithm. Only valid for cycle convergence. Default is off.

**dopower** When cycle convergence has been found, repeat the algorithm with the relevant power of the braid. Default is on. (When this option is turned off, growth rates output will be pretty much meaningless.)

**silent** When this option is on (and the **verbose** option is off), running the **step** command will produce no output at all. When it is off (and **verbose** is also off), it will report each time a puncture or an annulus is erased. Default is off.

**verbose** When this option is on, the **step** command will produce output after *every* step of the relaxation algorithm. Default is off.

**rescale** When this option is on, the foliation will be rescaled (so that sum of squares of Dynnikov coordinates is 1) after every step of the relaxation algorithm when the **step** command is run. Default is on.

Here are the integer options:

**accuracy** Number of decimal places required for convergence. Default 15.

**precision** Number of displayed decimal places. Default 8.

**bailout** Number of iterations after which to give up. Default 10000.

**discrepancy** If this value is positive, then once convergence has taken place, the eigenvector is computed to much greater precision, and the current vector is repeatedly hit with the transfer matrix, projected orthogonal to the eigenvector, and normalized (the given number of times). Each time the vector is displayed together with its growth under the previous application of the matrix. I don't remember why this might be useful. . .

Finally, there are some commands to do with curves, matrices, vectors, and foliations:

**curve** Set a curve. There are several versions. Curve names can be any string not including spaces: they are independent of braid names (so the same string can be used to describe both a braid and a curve).

- **curve C 4 1 -2 3 -5**: sets the curve using Dynnikov coordinates. In the above, the curve is on the 4-punctured disk. For  $n$  braid strings ( $n = 4$  above), there should be  $2n-4$  coordinates, which are in the order  $a_1, \dots, a_n, b_1, \dots, b_n$ .
- **curve C L n**: sets the curve to be the standard curve  $L_n$  on the  $n$ -punctured disk: that is, all of the  $a$  coordinates are 0, and all of the  $b$  coordinates are 1.
- **curve C u p i j**: sets the curve to be a “u” shaped curve on the  $p$  times punctured disk enclosing punctures  $i$  and  $j$  only. Must have  $1 \leq i \leq j - 2$  and  $j \leq p$ .

- **curve C n p i j**: sets the curve to be an “n” shaped curve on the p times punctured disk enclosing punctures i and j only. Must have  $1 \leq i \leq j - 2$  and  $j \leq p$ .
- **curve C act D b n**: sets C to be the curve obtained by acting n times on the curve D with the braid b. n defaults to 1 if omitted.
- **curve C random 5 10**: sets C to be a random curve on the 5-punctured disk with all coordinates between -10 and 10.
- **curve C randomsum 5 100**: sets C to be a random curve on the 5-punctured disk with sum of absolute values of the coordinates close to 100.

**printcurve printcurve C** displays the coordinates of C.

**norm norm C** displays the sum of the absolute values of the coordinates of C.

**count count C** displays the number of connected components of C.

**docount docount 4 1 -2 3 -5** sets a temporary curve and counts its components.

Any of the above means of setting a curve can be used, e.g. **docount random 100 1000**.

**multicount multicount 1000 random 100 100000** calculates the number of components of 1000 curves on the 100-punctured disk with random coordinates between -100000 and 100000. Can be used with any of the other curve setting commands, though it’s hard to see why you’d want to...

**propone propone 100000 5 1000** calculates the proportion of 100000 curves on the 5-punctured disk with random coordinates between -1000 and 1000 which have exactly 1 component.

**relax relax C D B** carries out Cumplido’s relaxation algorithm on the curve C. After running, D is the curve obtained by relaxing C, and B is the positive braid which carries out the relaxation. Note that the meaning of positive is determined by whether or not the Birman convention is active. B, or both B and D, can be omitted, in which case they are displayed but not saved. If the **silent** option is on, there is no output.

**relaxstep** works like relax, but only applies a the first braid generator.

**intersect intersect C D** displays the minimum intersection number of C and D.

**multiintersect multiintersect 1000 random 100 100000** calculates the intersection numbers of 1000 pairs of curves on the 100-punctured disk, each with random coordinates between -100000 and 100000. Can be used with any of the other curve setting commands, though it’s hard to see why you’d want to...

**draw draw C filename** puts a picture of the curve in filename (after adding .eps extension). It throws if the curve is too big. (For curves which aren’t that big, but which are quite big, the output will be useless (will look like a straight line segment).)

**using** `using C` tells the program that future `do` commands should use the curve `C` as the initial curve. The default is `'L'`. Can reset to the default by issuing `using`.

**matrix** `matrix M C b` sets  $M$  to be the Dynnikov matrix for the action of braid `b` on curve `C`.

**printmatrix** `printmatrix M` prints the matrix.

**vector** sets a vector. There are two options:

- `vector v curve C` sets  $v$  to be the coordinates of the curve `C`.
- `vector v mult M u` sets  $v$  to be the product of the matrix `M` and the vector `u`.

**printvector** `printvector v` prints the vector.

**foliation** sets a foliation. There are two options:

- `foliation f 4 -0.3 1.7 -2.215 -0.001` sets  $f$  to be the foliation on the 4-punctured disk with the given Dynnikov coordinates.
- `foliation f last` sets  $f$  to be the foliation resulting from the last `do` or `step` command.

**step** `step f n` carries out  $n$  steps of the relaxation algorithm on the foliation  $f$ . Note the effect of the boolean options `verbose`, `silent` and `rescale`. This command does not change the foliation  $f$ : to recover the resulting foliation, use the `foliation last` command. Note that `accuracy` is used in determining equality of Dynnikov coordinates.

**cstep** `cstep f n` as `step f n`, but uses the Cumplido relaxation algorithm.

**printfoliation** `f` Prints the foliation  $f$ .