SymRedInvCurves:
Symmetric Reducible Invariant Curves

À. Jorba [*], F.J. Muñoz-Almaraz [†], J.C. Tatjer [*]

June 14, 2016

This is a software to find 1–parameter families of symmetric invariant curves of a forced 1-D maps with Lyapunov exponent constant. These 1-D maps are also called skew–products and we study the following parametric systems

$$\begin{cases} x_{n+1} = f(a\,x_n) + b\,g(\theta_n) \\ \theta_{n+1} = \theta_n + \omega \end{cases} \tag{1}$$

where f is an odd function and g is periodic, being both of them analytical functions. The system is invariant with respect to the change of variable $(\theta, x) \mapsto (\theta + \pi, -x)$ and the symmetric curve is invariant w.r.t this change, i.e. $x(\theta + 1/2) = -x(\theta)$.

A full description of the algorithm used in this repository can be found in the preprint [1]. These algorithm use the Discrete Fourier Transform (DFT) and the library FFTW [2] has been employed and its installation is required.

The functions f and g are defined in the file `cont_complex.c` with homonymous name. Moreover, first and second derivatives of f have to be defined with the function names `df` and `ddf`. Other parameters have been defined in the file `parameter.h`

Newton's method is the main ingredient of continuation techniques which implies to solve linear systems in an iterative way. After reduction of a 1-D map around an approximated solution a system with the following framework must be solved

$$\begin{cases} h_y(\theta + \omega) = \lambda h_y(\theta) + h_a\,u_a(\theta) + h_b\,u_b(\theta), \\ 2\int_0^{1/2} d(\theta) h(\theta)\,d\theta = L. \end{cases} \tag{2}$$

where the unknown are h_y a symmetric function, h_a and h_b, and the data are the functions u_a, u_b, d and the number L. Following the procedure given by [1], it is possible to uncouple the system, getting a linear system whose unknowns are h_a and h_b and their coefficients are calculated according to the function in Section 3.

1 Description of parameters

The file `parameter.h` defines the folowing parameters:

*Universitat de Barcelona, Departament de Matemàtiques i Informàtica
†Universidad CEU-Cardenal Herrera, depto. Ciencias Físicas, Matemáticas y de la Computación

`OMEGA` This is the shift of the angular variable in the system. The default value is the conjugate of the golden ratio, $\Phi = .61803398874989484820458683436$5.

`EPS_FLOQUET` This is the maximum error during the reduction procedure.

`EPS` This is the maximum error solving the equation. Meanwhile, new iterations are calculated to reduce the error.

`LE` Lyapunov exponent constant for all the solutions in the continuation.

`INITIAL_A` Initial value of the parameter a to start the continuation.

`INITIAL_B` Initial value of the parameter b to start the continuation.

`INITIAL_CONTINUATION_STEP` Initial step of continuation, after that the continuation algorithm adapt the step to garantee a fast convergence to the solution.

`BD_FILE` The name of the file where data are saved for each calculated solution. The first column is the parameter a, the second column the parameter b, the third column is the length of the invariant curve, the number of Newton's method iteration to converge, continuation step and error. The default value is `bd.dat`

`TABU_FILE` The name of the file where every table values on a uniform mesh on $[0, 1/2]$ for the solution is stored, separated by an empty line. The default value is `tabu.dat`

`INITIAL_SOLUTION` It assigns the three first coefficients of the Fourier series of the starting solution.

`MODOS_MAX` It is the maximum number of Fourier coefficients allowed to perform the continuation. If more coefficients are needed to get a solution with the last 10% of coefficient less than EPS*EPS*EPS, the program is stopped.

Caviat: The value of `MODOS_MAX` must be small enough to avoid exceeding the computer memory.

2 Fourier Series (`fourier` structure)

Here we describe the file`symmetric_curves.c`. There is a header file associated with the same name. A structure is defined which contains the information about the function. This structure consists of the following attributes:

`mod` This is the number of terms in the Fourier series describing the function.

`type` There exist two types `SYMMETRIC` (1) and `HALF_PERIODIC` (0). Some methods have a different behaviour depending on this variable.

`coef` This is a complex vector containing the values of the Fourier series.

`tab` This is a vector with the values of the function on a uniform mesh in $[0, 1/2]$

`tabom` This is a vector with the values of the function on a uniform mesh in $[\omega, 1/2 + \omega]$. Please, notice that the inteval is considered in the torus modulus 1.

There are several methods which are applied to this structure:

`dimensiona` Init the vectors and increase the dimension if they were previously defined.

`tabulacion` Given the Fourier coefficients the values of the function on a uniform mesh in $[0, 1/2]$ are calculated.

`coeficiente` Given the value of the function on a mesh, the Fourier coefficients are described.

`coef_integral` Procedure to calculate the values of the coefficients of Trapezoidal rule.

2.1 Type `SYMMETRIC`

We consider $x : \mathbb{R}/\mathbb{Z} \to \mathbb{R}$ such that $x(\theta + \frac{1}{2}) = -x(\theta)$ The function $\theta \mapsto e^{i\pi\theta}x(\theta/2)$ is 1-periodic. Indeed,

$$e^{i\pi(\theta+1)}x(\frac{\theta+1}{2}) = -e^{i\pi\theta}x(\frac{\theta}{2} + \frac{1}{2}) = e^{i\pi\theta}x(\theta/2). \tag{3}$$

If $\tilde{\theta}_k$ is a uniform mesh of $[0, 1]$, with $\tilde{\theta}_k = k/n$ There exists $\{Y_j\}_{j=0}^{n-1}$ whose FFT interpolates the function on the points $\tilde{\theta}_k$, i.e $\{Y_j\}_{j=0}^{n-1}$ verifies

$$e^{i\pi\tilde{\theta}_k}x(\tilde{\theta}_k/2) = \sum_{j=0}^{n-1} Y_j e^{-2\pi j\tilde{\theta}_k i} = \sum_{j=0}^{n-1} Y_j e^{-i2\pi jk/n} = \mathrm{DFT}(\{Y_j\}_{j=0}^{n-1}) \tag{4}$$

2.1.1 Calculation of the function on a mesh $[0, 1/2]$

An uniform mesh of $[0, 1/2]$ is given by $\theta_k = \tilde{\theta}_k/2$. So, $e^{2i\pi\theta_k}x(\theta_k) = \mathrm{DFT}(\{Y_j\}_{j=0}^{n-1})$.

Description of calculation of `fourier.tab`: Given the coefficients of the Fourier series $\{Y_j\}_{j=0}^{n-1}$, if X_k is the output of DFT, i.e. $\{X_k\}_{k=0}^{n-1} = DFT(\{Y_j\}_{j=0}^{n-1})$. The value of the function on θ_k is given by $e^{i\pi k/n}X_k$.

The points $\{x(\theta_k)\}_{k=0}^{n-1}$ are given by

$$x(\theta_k) = \sum_{j=0}^{n-1} Y_j e^{-i2\pi(2j+1)\theta_k} \tag{5}$$

Calculation of coefficients `fourier.coef`: Given the values of the function $\{X_j\}_{j=0}^{n-1}$ on a mesh $[0, 1/2]$, we calculate the inverse DFT of $e^{2i\pi\, k/n}X_k$, i.e $\{Y_j\}_{j=0}^{n-1} = DFT^{-1}(e^{2i\pi\, k/n})$.

2.1.2 Calculation of tabom

The function $\tilde{\theta} \mapsto x(\tilde{\theta}/2)$ is approximate to

$$e^{i\pi\tilde{\theta}}x(\tilde{\theta}/2) \sim \sum_{j=0}^{n/2-1} Y_j e^{-i2\pi j\tilde{\theta}} + \sum_{j=0}^{n/2-1} Y_{n-1-j}e^{i2\pi(j+1)\tilde{\theta}} \tag{6}$$

With the change $\theta = \frac{\tilde{\theta}}{2}$

$$x(\theta) \sim \sum_{j=0}^{n/2-1} Y_j e^{-i2\pi(2j+1)\theta} + \sum_{j=0}^{n/2-1} Y_{n-1-j} e^{i2\pi(2j+1)\theta} \tag{7}$$

Obviously, if $x(\theta)$ is well approximated, then $\overline{Y_j} = Y_{n-1-j}$. So, $x(\theta_k + \omega)$ is approximate to

$$x(\theta_k + \omega) \sim \sum_{j=0}^{n/2-1} Y_j e^{-i2\pi(2j+1)\omega} e^{-i\pi(2j+1)k/n} + \sum_{j=0}^{n/2-1} Y_{n-1-j} e^{i2\pi(2j+1)\omega} e^{i\pi(2j+1)k/n} \tag{8}$$

We define $Z_j := Y_j e^{-i2\pi(2j+1)\omega}$ for $j = 0, ..., n/2-1$ and $Z_j = \overline{Z_{n-1-j}}$ for $j = n/2, ..., n-1$. So,

$$\sum_{j=0}^{n/2-1} Y_{n-1-j} e^{i2\pi(2j+1)\omega} e^{i\pi(2j+1)k/n} = \sum_{j=0}^{n/2-1} \overline{Z_j} e^{i\pi(2j+1)k/n} = \sum_{j=n/2}^{n-1} Z_j e^{i\pi(2n-2j-1)k/n} = \sum_{j=n/2}^{n-1} Z_j e^{-i\pi(2j+1)k/n} \tag{9}$$

With the Equation (8) we have,

$$x(\theta_k + \omega) \sim \sum_{j=0}^{n-1} Z_j e^{-i\pi(2j+1)k/n} = e^{-i\pi k/n} \sum_{j=0}^{n-1} Z_j e^{-i\pi(2j+1)k/n} = \mathrm{DFT}(\{Z_j\}_{j=0}^{n-1}) \tag{10}$$

Description of calculation of `fourier.tabom`: Given the coefficients $\{Y_j\}_{j=0}^{n-1}$ of the Fourier series, we apply the formula to define Z_j and we apply discrete Fourier transform (DFT). The output is the values of the function on the mesh $[\omega, 1/2 + \omega]$

2.2 Type `HALF SYMMETRIC`

Use the Discrete Fourier Transform for a real function.

3 Coefficient of a reduced system

Given the function r

$$y(\theta + \omega) = \lambda y(\theta) + r(\theta)$$

We calculate an approximation of the integral $2\int_0^{1/2} d(\theta) y(\theta)\, d\theta$

This function is implemented in the function `coefficient_parameter_system` the first parameter are the series corresponding to $d(\theta)$ and the second parameter the series of the independent term $r(\theta)$, the third parameter λ and the fourth parameter is number of modes n.

```
double coefficient_parameter_system(fourier *tint,fourier *rf,double lambda,double dmod)
```

We consider the following series for every function

$$y(\theta) = \sum_{j=0}^{n/2-1} Y_j e^{-i2\pi(2j+1)\theta} + \sum_{j=0}^{n/2-1} \overline{Y_j} e^{i2\pi(2j+1)\theta} \tag{11}$$

$$r(\theta) = \sum_{j=0}^{n/2-1} R_j e^{-i2\pi(2j+1)\theta} + \sum_{j=0}^{n/2-1} \overline{R_j} e^{i2\pi(2j+1)\theta} \tag{12}$$

We substitute in (3) and we get

$$Y_j e^{-i2\pi(2j+1)\omega} = \lambda Y_j + R_j, \qquad \forall j = 0, \dots, \frac{n}{2} - 1. \tag{13}$$

So, the value of Y_j is

$$Y_j = \frac{R_j}{e^{-i2\pi(2j+1)\omega} - \lambda} \tag{14}$$

We estimate the second equation in (3) with Composite Trapezoidal Rule

$$\frac{1}{n} \sum_{k=0}^{n-1} d(\theta_k) y(\theta_k) = \gamma \tag{15}$$

The function y evaluated on θ_k is $y(\theta_k) = e^{-i\pi k/n} \sum_{j=0}^{n-1} Y_j e^{-i2\pi(2j+1)k/n}$.So,

$$\frac{1}{n} \sum_{j=0}^{n-1} \left(\sum_{k=0}^{n-1} e^{-i\pi k/n} d(\theta_k) e^{-i2\pi jk/n} \right) Y_j = \gamma \tag{16}$$

We define

$$L_j := \sum_{k=0}^{n-1} e^{-i\pi k/n} d(\theta_k) e^{-i2\pi jk/n} \tag{17}$$

Obviously, $\mathrm{DFT}(\{e^{-i\pi k/n} d(\theta_k)\}_{k=0}^{n-1}) = \{L_j\}_{j=0}^{n-1}$. We check that $L_{n-1-j} = \overline{L_j}$

$$L_{n-1-j} = \sum_{k=0}^{n-1} e^{-i\pi k/n} d(\theta_k) e^{-i2\pi(n-1-j)k/n} = \sum_{k=0}^{n-1} d(\theta_k) e^{-i\pi k/n} e^{i2\pi k/n} e^{i2\pi jk/n} = \overline{L_j} \tag{18}$$

With the above notation

$$2 \int_0^{1/2} d(\theta) y(\theta)\, d\theta = \frac{1}{n} \sum_{j=0}^{n-1} L_j Y_j = \frac{2}{n} \mathrm{Re}(\sum_{j=0}^{n/2-1} L_j Y_j) = \frac{2}{n} \mathrm{Re} \left(\sum_{j=0}^{n/2-1} \frac{L_j R_j}{e^{-i2\pi(2j+1)\omega} - \lambda} \right)$$

References

[1] A. Jorba, F.J. Muñoz, J.C. Tatjer *On non-smooth pitchfork bifurcations in invertible queasi-periodically forced 1-D maps.* Preprint

[2] M. Friggo, S.G. Johnson *FFTW* Available in `http://www.fftw.org/`