MANLAB 4 theory and tutorial

June 2019

Olivier THOMAS

Arts et Métiers ParisTech,
Laboratoire d'Ingénierie des Systèmes Physiques et Numériques (LISPEN), EA 7515
Lille, France

olivier.thomas@ensam.eu

MANLAB

- An interactive path-following and bifurcation analysis software based on the Asymptotic numerical method;
- developed by the Bruno COCHELIN and Christophe VERGEZ team of Laboratoire de Mécanique et d'Acoustique (LMA) de Marseille since 2004: `http://manlab.lma.cnrs-mrs.fr/`
- addition of stability computation by Arnaud LAZARUS (Institut d'Alembert, Paris) and Olivier THOMAS (Arts et Métiers, Lille)

MANLAB versions

- Manlab 1 (2009, R. Arquier PhD) $\rightsquigarrow$ a continuation kernel for quadratic algebraic systems of equations;
- Manlab 2 (2010, S. Karkar PhD) $\rightsquigarrow$ addition of Harmonic balance method for the continuation of periodic orbits;
- Manlab 2 (2010, S. Karkar PhD) $\rightsquigarrow$ addition of Harmonic balance method for the continuation of periodic orbits + stability computation by the Hill method + Fortran acceleration;
- Manlab 3 (2011, 2014, S. Karkar) $\rightsquigarrow$ Tensor acceleration
- **Manlab 4 (2018, L. Guillot PhD) $\rightsquigarrow$ complete rewriting of the code with high acceleration + quasi-periodic solutions . . .**

Outline

General framework

We consider the N-dimensional algebraic system

$$\boldsymbol{R}(\boldsymbol{U},\lambda)=\boldsymbol{0},$$

with

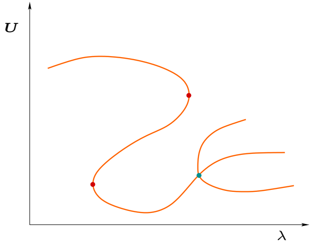
$$\boldsymbol{R}: \left| \begin{array}{ccc} \mathbb{R}^N\times\mathbb{R} & \longrightarrow & \mathbb{R}^N \\ (\boldsymbol{U},\lambda) & \longmapsto & \boldsymbol{R}(\boldsymbol{U},\lambda) \end{array} \right.$$

The implicit functions theorem says that if $\boldsymbol{R}$ is continuously differentiable with respect to $\boldsymbol{U}$ and λ and if $\partial\boldsymbol{R}/\partial\boldsymbol{U}$ is invertible, their exists a continuously differentiable function $\boldsymbol{g}$ such that:

$$\boldsymbol{g}: \left| \begin{array}{ccc} \mathbb{R} & \longrightarrow & \mathbb{R}^N \\ \lambda & \longmapsto & \boldsymbol{g}(\lambda)=\boldsymbol{U} \end{array} \right.$$

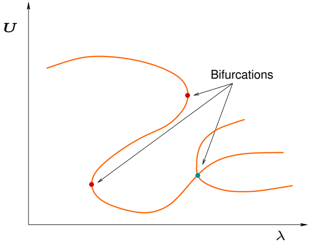
General framework

We can then wind a network of curves of $\boldsymbol{U}$ as a function of λ:



General framework

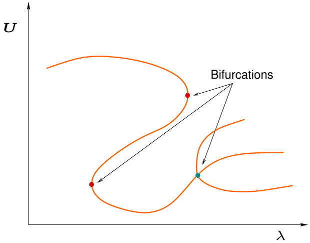
We can then wind a network of curves of $\boldsymbol{U}$ as a function of λ:



But in some **bifurcation points**, the curves can cross or have a vertical tangent with respect to λ.

General framework

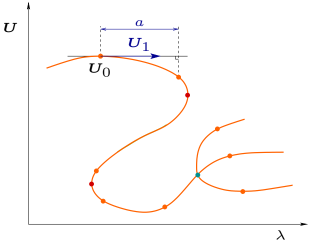
We can then wind a network of curves of $\boldsymbol{U}$ as a function of λ:



But in some **bifurcation points**, the curves can cross or have a vertical tangent with respect to λ.

A continuation (or path following) method is a numerical method that computes $\boldsymbol{U}$ for several values of λ in a given set and that manages the bifurcation points.

Path parametrization

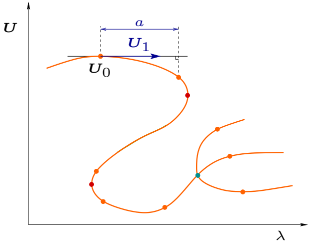


A way of dealing with the bifurcation points is to add a path parametrization. For instance, define a scalar a such that:

$$\begin{cases} \boldsymbol{U} = \boldsymbol{U}(a) \\ \lambda = \lambda(a) \\ f(\boldsymbol{U}, \lambda, a) = 0 \end{cases} \quad \Rightarrow \quad \begin{cases} \boldsymbol{R}(\boldsymbol{U}(a), \lambda(a)) = 0 \\ f(\boldsymbol{U}, \lambda, a) = 0 \end{cases}$$

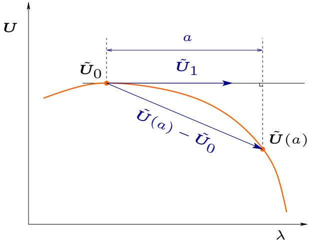
A new set of algebraic equations is defined, of size $\mathbb{R}^{N+1}$ which is not singular as a function of a.

Path parametrization



Depending on the definition $f(\boldsymbol{U}, \lambda, a) = 0$ of the path parameter a, several parametrization are available (arclength, pseudo-arclength, secant...) (see for instance [Seydel 2010]).

Path parametrization

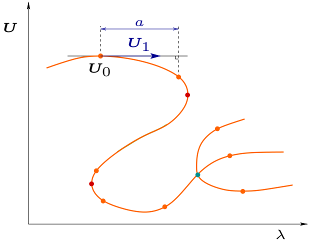

$\boldsymbol{U}$
a
$\tilde{\boldsymbol{U}}_0$
$\tilde{\boldsymbol{U}}_1$
$\tilde{\boldsymbol{U}}(a)-\tilde{\boldsymbol{U}}_0$
$\tilde{\boldsymbol{U}}(a)$
λ

In Manlab, a pseudo-arclength parametrization is used. Let us first define $\tilde{\boldsymbol{U}} = [\boldsymbol{U}^{\mathrm{t}}\ \lambda]^{\mathrm{t}}$ such that $\boldsymbol{R}(\boldsymbol{U},\lambda) = \boldsymbol{R}(\tilde{\boldsymbol{U}})$. Then, a is defined by:

$$f(\boldsymbol{U},\lambda,a) = 0 \quad \Rightarrow \quad \boxed{a = \left[\tilde{\boldsymbol{U}}(a) - \tilde{\boldsymbol{U}}_0\right]^{\mathrm{t}} \tilde{\boldsymbol{U}}_1}$$

with $\tilde{\boldsymbol{U}}_0 = \tilde{\boldsymbol{U}}(a=0)$ and $\tilde{\boldsymbol{U}}_1 = \partial\tilde{\boldsymbol{U}}/\partial a|_{a=0}$.

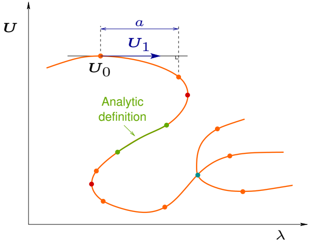
Numerical solving



▷ **Newton-Raphson method**

- the most used;
- enables the computation of a discrete set of solutions such that $\boldsymbol{R}(\boldsymbol{U}(a), \lambda(a)) = 0$ by sweeping the a parameter;
- one has to control the a stepping. . .

Numerical solving



▷ **The asymptotic numerical method (ANM)**

- Méthode Asymptotique Numérique (MAN) in french;
- Initiated by M. Potier-Ferry in Metz, France in the 1990';
- Power series expansions of the unknowns as a function of a: the branches are continuous function of a;
- Automatic stepping, very few control parameters;

Asymptotic Numerical Method

- The unknowns $\boldsymbol{U}$ and λ are expanded as power series of $a \in \mathbb{R}$ of order $n \in \mathbb{N}$:

$$\boldsymbol{U}(a) = \boldsymbol{U}_0 + a\boldsymbol{U}_1 + a^2\boldsymbol{U}_2 + \ldots + a^n\boldsymbol{U}_n,$$

$$\lambda(a) = \lambda_0 + a\lambda_1 + a^2\lambda_2 + \ldots + a^n\lambda_n.$$

- For **efficiency** of the numerical procedure, we write $\boldsymbol{R}(\tilde{\boldsymbol{U}})$ quadratically:

$$\boldsymbol{R}(\tilde{\boldsymbol{U}}) = \boldsymbol{C} + \boldsymbol{L}(\tilde{\boldsymbol{U}}) + \boldsymbol{Q}(\tilde{\boldsymbol{U}}, \tilde{\boldsymbol{U}})$$

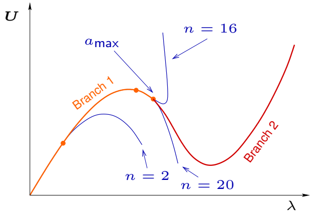
  where $\boldsymbol{C} \in \mathbb{R}^N$, $\boldsymbol{L} \in \mathbb{R}^N$ and $\boldsymbol{Q} \in \mathbb{R}^N$ are constant, linear and quadratic functions of $\tilde{\boldsymbol{U}}$.

- Introducing the above equations into $\boldsymbol{R}(\tilde{\boldsymbol{U}}) = 0$ **for all** $a \in \mathbb{R}$ leads to a cascade of successive problems:

$$\begin{array}{rll} \text{order } 0: & \boldsymbol{R}(\tilde{\boldsymbol{U}}_0) = 0 & \leftarrow \text{NL initial system} \\ \text{order } 1: & \boldsymbol{J}_0\tilde{\boldsymbol{U}}_1 = 0 & \leftarrow \text{Linear system} \\ \text{order } p = 2, 3, \ldots n: & \boldsymbol{J}_0\tilde{\boldsymbol{U}}_p = -\displaystyle\sum_{i=1}^{p-1} \boldsymbol{Q}(\tilde{\boldsymbol{U}}_i, \tilde{\boldsymbol{U}}_{p-i}) & \leftarrow \text{Linear system} \end{array}$$

  n linear systems with the same “stiffness” matrix $\boldsymbol{J}_0 = \partial\boldsymbol{R}/\partial\tilde{\boldsymbol{U}}|_{\tilde{\boldsymbol{U}}_0}$.

Asymptotic Numerical Method



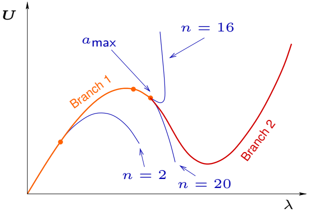
- The power series above defined have a radius of convergence such that the range of utility of a given series is defined by the value of $a_{\max}$ such that:

$$\forall a \in [0\, a_{\max}], \quad ||\boldsymbol{R}(\tilde{\boldsymbol{U}}(a)|| \leq \varepsilon$$

where ε is a user tolerance parameter.

- $a_{\max}$ is automatically computed.
- The complete branch of solutions is obtained by successive power series. The $k+1$ seris is initiated such that its first point equals the last point of the previous one: $\tilde{\boldsymbol{U}}_0^{k+1} = \tilde{\boldsymbol{U}}(a_{\max})^k$.

Asymptotic Numerical Method



- The tolerance ε does not grow from branches to branches.
- In Manlab, the first point of the first branch $\tilde{\boldsymbol{U}}_0$ is obtained by a Newton-Raphson method. Its accuracy conditions the one of the whole branche. Local Newton-Raphson corrections are possible with the interface.
- A bifurcation detector, special to the ANM, is also implemented in Matlab 4.

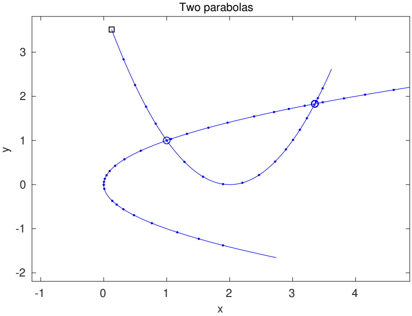
Outline

- **MANLAB**

- **The kernel: continuation of algebraic systems**
  - Theory
  - Examples

- **Continuation of periodic solutions**
  - Theory
  - Results visualizations

Two parabolas



- We consider two parabolas:

$$\begin{cases} x = y^2 \\ y = (x-2)^2 + a; \end{cases}$$

  where (x, y) are two variables and a a scalar paramater.

- Those two parabolas are solution of the implicit equation:

$$R(x, y) = (x - y^2)[y - (x-2)^2 - a] = 0$$

In Manlab

$$R(x,y) = (x - y^2)[y - (x-2)^2 - a] = 0$$

- Add some auxiliary variables to obtain a quadratic system:

$$\begin{cases} u_1 = x - y^2 \\ u_2 = y - (x-2)^2 - a \end{cases}$$

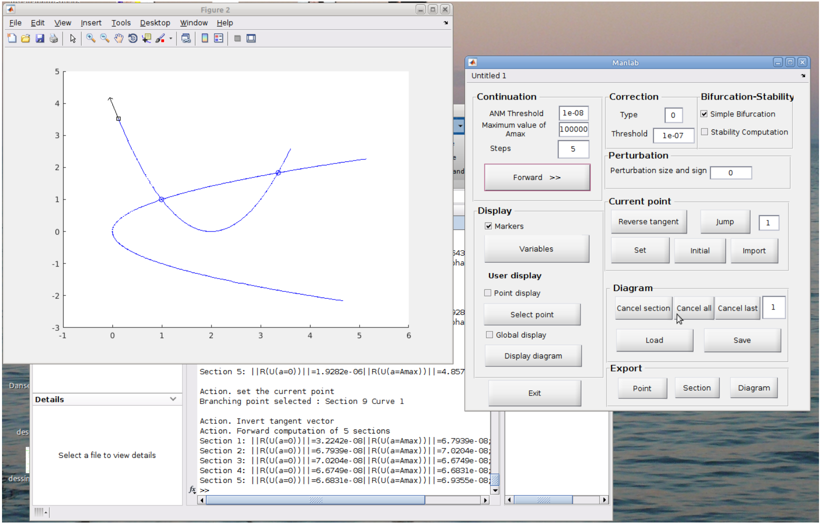
- Implement the following system, quadratic in the variables $\boldsymbol{U} = [x\ y\ u_1\ u_2]^{\mathrm{t}}$:

Primary system: $R_1 = u_1 u_2 = 0$

Auxiliary system: $\begin{cases} R_{1\mathsf{aux}} = u_1 - x + y^2 = 0 \\ R_{2\mathsf{aux}} = u_2 - y + (x-2)^2 + a = 0 \end{cases}$

- Four `.m` files:
  - the main file: `your_name.m`: contains all the parameters of the simulation; Its name can be chosen by the user; this file has to be executed to launch the interactive simulation;
  - the equation file: `equation.m`: where the algebraic system is coded;
  - two optional files for automatic display: `point_display.m` and `global_display.m`

Manlab screenshot



ANM Threshold 1e-08

Maximum value of Amax 100000

Steps 5

Type 0

Threshold 1e-07

☑ Simple Bifurcation

☐ Stability Computation

Perturbation size and sign 0

☑ Markers

☐ Point display

☐ Global display

Jump 1

Cancel last 1

Select a file to view details

```
Section 5: ||R(U(a=0))||=1.9282e-06||R(U(a=Amax))||=4.857

Action. set the current point
Branching point selected : Section 9 Curve 1

Action. Invert tangent vector
Action. Forward computation of 5 sections
Section 1: ||R(U(a=0))||=3.2242e-08||R(U(a=Amax))||=6.7939e-08;
Section 2: ||R(U(a=0))||=6.7939e-08||R(U(a=Amax))||=7.0204e-08;
Section 3: ||R(U(a=0))||=7.0204e-08||R(U(a=Amax))||=6.6749e-08;
Section 4: ||R(U(a=0))||=6.6749e-08||R(U(a=Amax))||=6.6831e-08;
Section 5: ||R(U(a=0))||=6.6831e-08||R(U(a=Amax))||=6.9355e-08;
>>
```

Outline

- ▷ **MANLAB**
- ▷ **The kernel: continuation of algebraic systems**
  - Theory
  - Examples
- ▷ **Continuation of periodic solutions**
  - Theory
  - Results visualizations

Framework

- We consider a N-dimensional first order dynamical system:

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \lambda, t),$$

with

$$\boldsymbol{f} : \left| \begin{array}{rcl} \mathbb{R}^N \times \mathbb{R} \times \mathbb{R} & \longrightarrow & \mathbb{R}^N \\ (\boldsymbol{x}, \lambda, t) & \longmapsto & \boldsymbol{f}(\boldsymbol{x}, \lambda, t) \end{array} \right.$$

where t is the time and λ a control parameter.

- We look for T-periodic solutions (at frequency $\omega = 2\pi/T$) of this system:

$$\forall T \in \mathbb{R} \qquad \boldsymbol{x}(t+T) = \boldsymbol{x}(t).$$

- For this, we expand $\boldsymbol{x}$ in Fourier series:

$$\boldsymbol{x}(t) = \boldsymbol{x}^{(0)} + \sum_{h=1}^{H} \left(\boldsymbol{x}^{(hc)} \cos \omega t + \boldsymbol{x}^{(hs)} \sin \omega t \right),$$

and we use the the harmonic balance method (HBM), which leads to replace the initial dynamical system by an algébraic dynamical system $\boldsymbol{R}(\boldsymbol{X}, \lambda, \omega) = \boldsymbol{0}$ where $\boldsymbol{X}$ collects all the Fourier series coefficients $\boldsymbol{x}^{(0)}$, $\boldsymbol{x}^{(hc)}$, $\boldsymbol{x}^{(hs)}$ of $\boldsymbol{x}$.

In MANLAB, four cases

- Autonomous system (ex: Van der Pol)

$$\ddot{u} - \lambda(1-u^2)\dot{u} + u = 0$$

  - N primary unknowns $+ \lambda + \omega \Rightarrow N+2$ unknowns
  - N primary equations + 1 phase condition $\Rightarrow N+1$ equations

```
sys=SystHBQ(nz,nz_aux,H,@equations,@point_display,@global_display,
    parameters,'autonomous','standard');
sys.zi_phase = 1; % Indice of component for the phase condition
```

- Autonomous conservative system (Hamiltonian, ex: free Duffing)

$$\ddot{u} + \lambda\dot{u} + u + u^3 = 0$$

  - N primary unknowns $+ \lambda + \omega \Rightarrow N+2$ unknowns
  - N primary equations + phase condition $\Rightarrow N+1$ equations

```
sys=SystHBQ(nz,nz_aux,H,@equations,@point_display,@global_display,
    parameters,'autonomous','standard');
sys.zi_phase = 1; % Indice of component for the phase condition
```

In MANLAB, four cases

- Forced system at Ω and Ω is the bifurcation parameter (ex: Duffing)

$$\ddot{u} + 2\mu\dot{u} + u + u^3 = F\cos\Omega t$$

  - N primary unknowns $+ \lambda = \Omega \Rightarrow N + 1$ unknowns
  - N primary equations $\Rightarrow N$ equations

```
parameters.angfreq = 'omega';
sys=SystHBQ(nz,nz_aux,H,@equations,@point_display,@global_display,
    parameters,'forced','standard')
```

- Forced system at fixed Ω with a bifurcation parameter (ex: Duffing)

$$\ddot{u} + 2\mu\dot{u} + u + u^3 = \lambda\cos\Omega t$$

  - N primary unknowns $+ \lambda \Rightarrow N + 1$ unknowns
  - N primary equations $\Rightarrow N$ equations

```
parameters.angfreq = 2; % fixed value of the angular frequency
sys=SystHBQ(nz,nz_aux,H,@equations,@point_display,@global_display,
    parameters,'forced','standard')
```

The phase condition in Manlab

- For forced systems, the time reference is given by the forcing signal, so that the phase of the periodic solution $\boldsymbol{x}(t)$ is referenced with respect to the phase of the forcing signal. No phase condition is needed.
- For autonomous systems, there is no reference for the phase of the periodic solution of the system and if $\boldsymbol{x}(t)$ is solution, $\boldsymbol{x}(t+\tau)$ for any $\tau < T$ is also solution. One has then to impose the phase of the seeked periodic solution to make it unique.
- In harmonic balance methods, a way is to set to zero a given harmonic of a given component of $\boldsymbol{x}(t)$ (for instance, the sine component of the first harmonics of the i-th. component: $x_i^{(1s)} = 0$. This is the phase condition.
- In Manlab 4.1.5, it is the time derivative at $t = 0$ of the i-th. component of $\boldsymbol{x}(t)$ which is set to zero:

$$\frac{\mathrm{d}\, x_i}{\mathrm{d}t}(t=0) = 0 \quad \Rightarrow \quad \sum_{h=1}^{H} h\, x_i^{(h\mathrm{s})} = 0$$

- The component number i is set by `sys.zi_phase = i;` in Manlab.

Outline

- **MANLAB**
- **The kernel: continuation of algebraic systems**
  - Theory
  - Examples
- **Continuation of periodic solutions**
  - Theory
  - Results visualizations

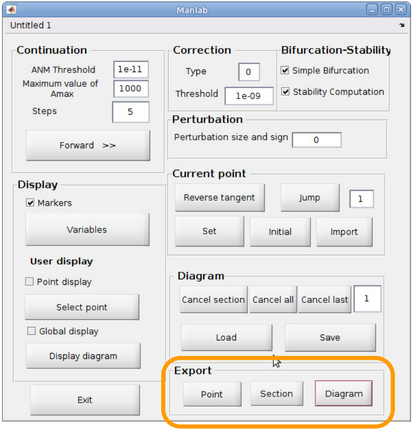
Storage of data in Manlab

- In Manlab, the basic object is a power series of $\tilde{\boldsymbol{U}}$:

  $$\tilde{\boldsymbol{U}}(a) = \tilde{\boldsymbol{U}}_0 + a\tilde{\boldsymbol{U}}_1 + a^2\tilde{\boldsymbol{U}}_2 + \ldots + a^n\tilde{\boldsymbol{U}}_n,$$

  valid for $a \in [0\ a_{\text{max}}]$. For graphical purpose, it is computed at several points between $a = 0$ and $a = a_{\text{max}}$. All those data ($\boldsymbol{U}_0$, $\tilde{\boldsymbol{U}}_1$, a_{max} etc.) are stored in a `section` object. An ensemble of sections is a `diagram` object.

- For a given simulation, you have access to a particular `section` or the whole `diagram` by clicking on the buttons of the Export section of the Manlab interface.



Manlab

Untitled 1

Continuation

ANM Threshold 1e-11

Maximum value of Amax 1000

Steps 5

Correction

Type 0

Threshold 1e-09

Bifurcation-Stability

☑ Simple Bifurcation

☑ Stability Computation

Perturbation

Perturbation size and sign 0

Display

☑ Markers

User display

☐ Point display

☐ Global display

Current point

Jump 1

Diagram

Cancel last 1

Basic export

- If you click on the **Point** button, and you select a given point of the diagram, you simply obtain a vector `U` that contains all the components of $\tilde{U}$.
- If you click on the **Section** button, and you select a given section of the diagram, you obtain an object `Section` that contains all the data of the section. In particular, `Section.Upp` contains the components of $\tilde{U}$ for all the points of the section.
- If you click on the **Diagram** button, you obtain a `cell array` of section objects. In particular, `Diagram{3}.Upp` contains the components of $\tilde{U}$ for all the points of the 3rd. section of the Diagram.

$$\mathtt{U} = \begin{bmatrix} \boldsymbol{U} \\ \lambda \\ \boldsymbol{U}_{\text{aux}} \end{bmatrix} \qquad \mathtt{Section.Upp} = \underbrace{\begin{bmatrix} \dots & \boldsymbol{U} & \dots \\ \dots & \lambda & \dots \\ \dots & \boldsymbol{U}_{\text{aux}} & \dots \end{bmatrix}}_{\substack{\text{number of computation} \\ \text{points in the section}}}$$

A convenient object for Diagrams

- `Diagram` objects can be converted to a simplier object:

  `[Diag] = calcdiagUpp(sys,Diagram)`

$$\mathtt{Diag.DiagUpp} = \underbrace{\begin{bmatrix} \dots & \boldsymbol{U} & \dots \\ \dots & \lambda & \dots \\ \dots & \boldsymbol{U}_{\mathsf{aux}} & \dots \end{bmatrix}}_{\substack{\text{number of computation} \\ \text{points in the diagram}}}$$

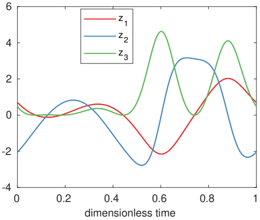
  where `Diag.DiagUpp` contains $\tilde{\boldsymbol{U}}$ for all the points of the diagram in a single matrix. `Diag.Stabinfo` and `Diag.change` contains data relative to the stability of the branches.

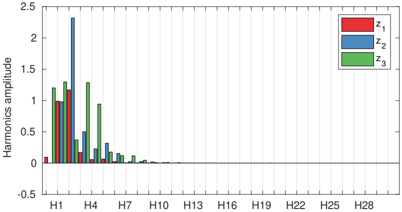
Structure of `Upp` for HBM simulations

For HBM simulations, $\boldsymbol{U}$ and $\boldsymbol{U}_{\text{aux}}$ contains all the Fourier series coefficients of $\boldsymbol{X}(t)$ as well as ω, ω^2 and λ^2 in the following way:

$$\boldsymbol{X}_i = \begin{bmatrix} x_i^{(0)} \\ x_i^{(1\text{c})} \\ \vdots \\ x_i^{(H\text{c})} \\ x_i^{(1\text{s})} \\ \vdots \\ x_i^{(H\text{s})} \end{bmatrix} \qquad \texttt{Diag.DiagUpp} = \underbrace{\begin{bmatrix} \dots & \boldsymbol{X}_1 & \dots \\ \dots & \vdots & \dots \\ \dots & \boldsymbol{X}_N & \dots \\ \dots & \omega & \dots \\ \dots & \lambda & \dots \\ \dots & \boldsymbol{X}_{1\text{aux}} & \dots \\ \dots & \vdots & \dots \\ \dots & \boldsymbol{X}_{N\text{aux}} & \dots \\ \dots & \omega^2 & \dots \\ \dots & \lambda^2 & \dots \end{bmatrix}}_{\substack{\text{number of computation} \\ \text{points in the diagram}}}$$

Display / computation functions for single point `U`





Let `U` be the result of a single point export.

- `Utime = calcperiodHBM(sys,U,Icalc,time)`

  `plotperiodHBM(sys,U,Idisp)`

  computes / plots the periodic time evolution of several variables.
- `plotbarHBM(sys,U,Idisp)`

  `plotbarsincosHBM(sys,U,Idisp)`

  plots the harmonics content of several variables as a bargraph

Display / computation functions for Sections / Diagram

Let `Diag` be a Section or a Diagram / Diag object.

- `plotdiagnormHBMbif(sys,Diag,Idisp,bifpara_str)`
  `plotdiagHBMbif(sys,Diag,Idisp,Hdisp,bifpara_str)`
  `plotdiagHBM(sys,Diag,Idisp,Hdisp,bifpara_str)`
  plots the amplitude or the L_2 norm of the variables in `Idisp` as a function of ω or λ (`bifpara_str='omega'` or `bifpara_str='lambda'`). For the first two functions, the type of bifurcations are specified by letters ('B': branch point; 'PD': period doubling; 'NS': Neimark-Sacker);
- `plotdiagYHBM(sys,Diag,Y,bifpara_str)`
  `plotdiagXYHBM(sys,Diag,X,Y)`
  The same as before with y-axis (or both axis) ploted with the stability. `X` and `Y` can be ontained by
  `calcdiagHBM(sys,Diag,Icalc,Hcalc)`
  `calcdiagUpp(sys,Diagram)`
- Other functions are under addition, for Manlab 4.1.6...

During simulation plots

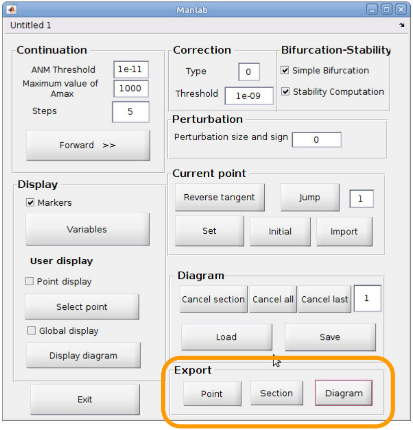
Don't hesitate to include the previous defined functions in:

- `point_display.m` → applies to a single point `U`
- `global_display.m` → applies to a section or the diagram.

Doing plots and post-treatments

- do your Manlab simulations. Check them during the simulation by programming `point_display.m` and `global_display.m`;
- when you know your bifurcation diagram, compute and export using the "Diagram" button of the interface. It creates a global variable called `Diagram` that contains all the results of the simulation.
- Save **all** the variables (`Diagram`, `sys` and the others).

  `> save simulation.mat`
- Apply all the previous defined display functions.



ANM Threshold 1e-11

Maximum value of Amax 1000

Steps 5

Type 0

Threshold 1e-09

☑ Simple Bifurcation

☑ Stability Computation

Perturbation size and sign 0

Jump 1

☑ Markers

☐ Point display

☐ Global display

Cancel last 1