# Calcul des modes non linéaires avec Manlab-4.0

B. Cochelin and L. Guillot

*LMA, UMR 7051, Aix marseille univ, CNRS, Centrale Marseille*,

,

Ecole Thématique - Dynolin 2018

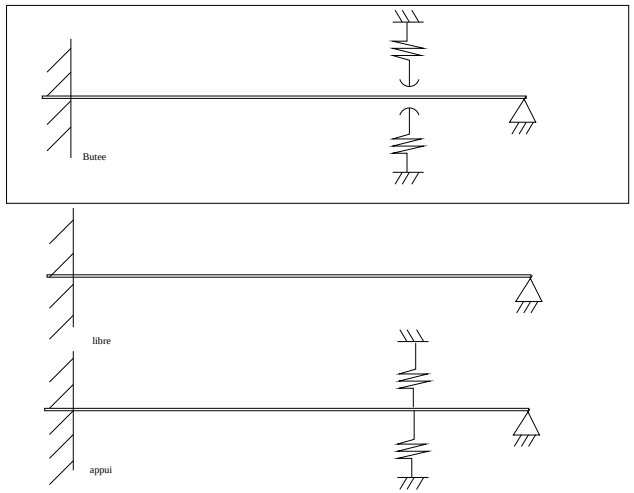
# Outline

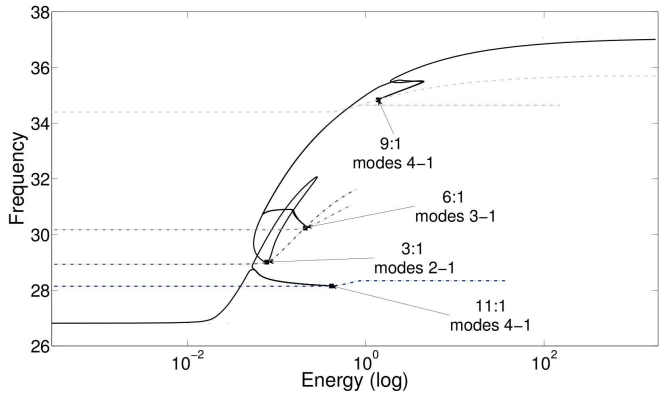
1 Overview

2 Continuation of algebraic system with Manlab-4.0
- User point of view
- More on the method used in Manlab-4.0

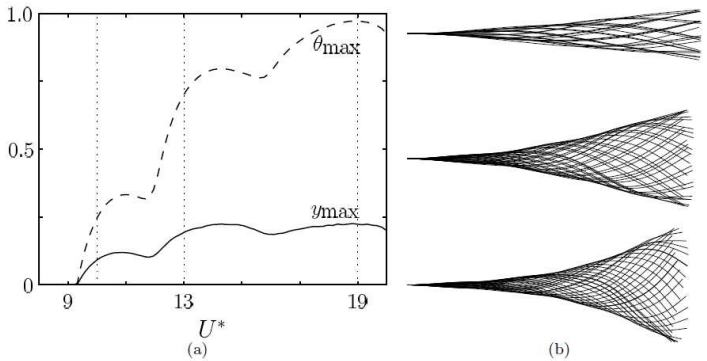
# Nonlinear mode : 1-D continunm of periodic solutions

- conservative systems




- auto-oscillations



# Aim of the talk :

Describe a method and a software to

- compute periodic solutions of dynamical systems using Harmonic Balance Method (HBM). Fourier series representation.
- do their continuation using Taylor series expansions, the so-called Asymptotic Numerical Method. (ANM)
- using the interactive software MANLAB

# Introduction to Harmonic Balance Method (HBM)

- We look for periodic solution to $\ddot{u} + 2\mu\dot{u} + u + u^3 = f\cos(\omega t)$ ($\mu$ and $f$ cst)
- HBM using one harmonic : $u(t) = u_c\cos(\omega t) + u_s\sin(\omega t)$

  $\cos^3(x) = \frac{3}{4}\cos(x) + \frac{1}{4}\cos(3x)$

$$\rightarrow \begin{array}{l} \{(1-\omega^2)u_c + 2\mu\omega u_s + \frac{3}{4}u_c^3 + \frac{3}{4}u_c u_s^2\}\cos(\omega t) \\ +\{(1-\omega^2)u_s - 2\mu\omega u_c + \frac{3}{4}u_s^3 + \frac{3}{4}u_s u_c^2\}\sin(\omega t) \\ +\{\ldots\}\cos(3\omega t) \\ +\{\ldots\}\sin(3\omega t) = f\cos(\omega t) \end{array}$$

- balancing of the harmonics yields an algebraic system with two equations

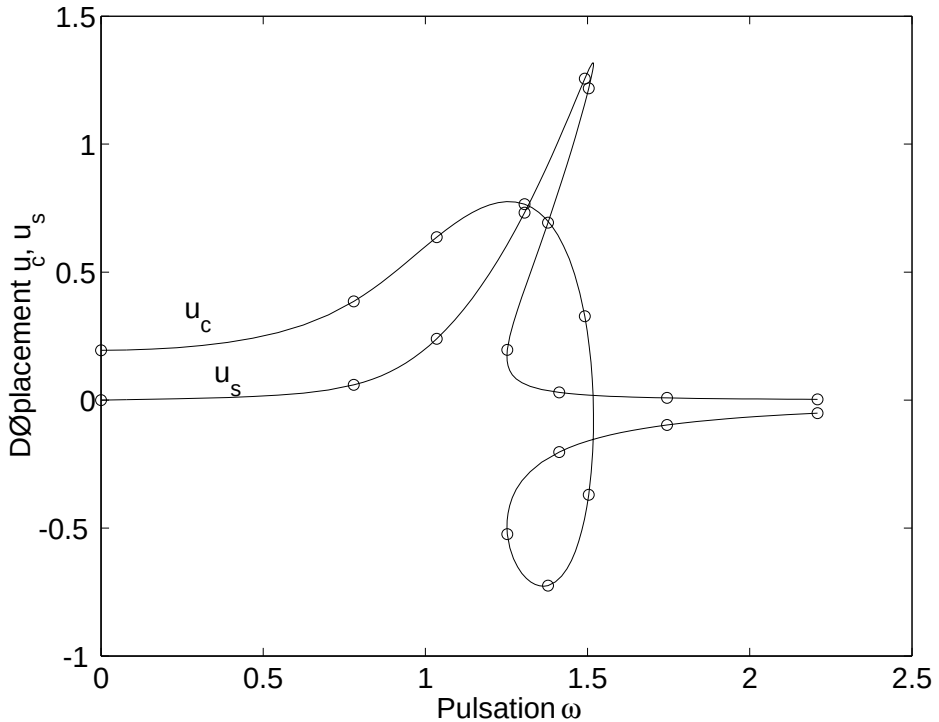
$$\begin{aligned} r_1(u_c, u_s, \omega) &:= (1-\omega^2)u_c + 2\mu\omega u_s + \frac{3}{4}u_c^3 + \frac{3}{4}u_c u_s^2 - f = 0 \\ r_2(u_c, u_s, \omega) &:= (1-\omega^2)u_s - 2\mu\omega u_c + \frac{3}{4}u_s^3 + \frac{3}{4}u_s u_c^2 = 0 \end{aligned}$$

for the two unknowns $u_c$, $u_s$ and one parameters, $\omega$.

# Introduction to Taylor series continuation

- The branches of solution of an algenraic system $\mathbf{R}(\mathbf{U}) = 0$ with $\mathbf{R} \in \mathbb{R}^n$ and $\mathbf{U} = [u, \lambda] \in \mathbb{R}^{n+1}$ is determine as a succession of local Taylor series expansions with respect to a path parameter $a$.

$$\mathbf{U}(a) = \mathbf{U}_0 + a\mathbf{U}_1 + a^2\mathbf{U}_2 + \cdots + a^N\mathbf{U}_N$$



# Summary of the method :

- Find the periodic solution of an ODE system

$$\dot{Y} = f(Y, \lambda) \qquad Y(t) \in \mathbb{R}^n$$

by using a high order Fourrier serie expansion (HBM)

$$Y(t) = Y_0 + \sum_{k=1}^{H} Y_{c,k} \cos(k\omega t) + \sum_{k=1}^{H} Y_{s,k} \sin(k\omega t)$$

- The resulting (nonlinear) algebraic system on the Fourrier coefficients reads :

$$R(U) = 0 \qquad \text{with} \qquad U = [Y_0, Y_{c,k}, Y_{s,k}, \omega, \lambda]$$

- Pathfollowing of the branches using high order Taylor series expansions with respest to a path parameter $a$ .

$$U(a) = U_0 + a\, U_1 + a^2 \ + \cdots + a^N\, U_N$$

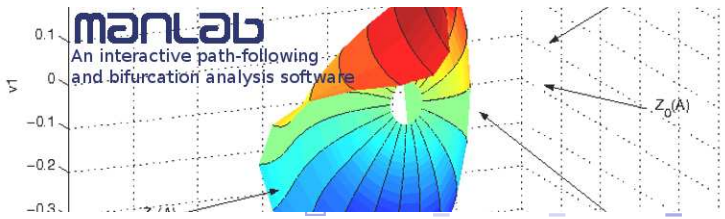
- The resulting linear algebraic system on the $U_i$ reads :

$$\frac{\partial \mathbf{R}}{\partial \mathbf{U}} \mathbf{U}_p = F_p^{nl}(\mathbf{U}_1, \ldots, \mathbf{U}_{p-1})$$

# Manlab software History

- Very first version : FORTRAN95 (2000), Matlab(2004) , no GUI !
- Manlab-1.0 on line April 2009 (written by R. Arquier).
  - first implementation of HBM in MANLAB (2009).
- Manlab-2.0 on line November 2010.
  - improved GUI, stability analysis by A. Lazarus and O. Thomas
- Diamanlab-1.0 2012
  - Automatic Differentiation with I. Charpentier
- Manlab-3.0 on line october 2014 (S. Karkar)
  - HBM,orthogonal collocation,Stability, tensor approach initiation.
- Manlab-4.0 on line may 2018 (L. Guillot)
  - Explicit definition of auxiliary variables
  - New sparce tensor formalism, condensation -> efficiency (CPU/50 to /1000)
  - treatement of elementary function more transparent

http ://manlab.lma.cnrs-mrs.fr



# The conerstone

- In HBM : insert $\theta(t) = \theta_0 + \sum_{k=1}^{H} \theta_{c,k} \cos(k\omega t) + \sum_{k=1}^{H} \theta_{s,k} \sin(k\omega t)$ into, for example,

$$R(\theta(t), \lambda) := \ddot{\theta} + \lambda\dot{\theta} + \theta^3 + \sin(\theta) = 0$$

  and "balance" the harmonic !

- in ANM : insert $u(a) = u_0 + a\, u_1 + a^2 \ + \cdots + a^N\, u_N$ into

$$R(u, \lambda) := u + u^2 + \frac{tan(u)}{1+u} - \lambda = 0$$

  and collect term with the same powers !

How to :

- use Automatic Differentiation to do the job : nice for the user but poor efficiency.
- do a quadratic recast of the equation : then the job become easy and efficient

# Outline

1 Overview

2 Continuation of algebraic system with Manlab-4.0
- User point of view
- More on the method used in Manlab-4.0

# Quadratic recast of algebraic systems

Exemple 1 : basique, sans fonction
Equations principales :

$$r(u, \lambda) := u + u^3 - \lambda = 0$$

Definition des variables auxiliaires :

$$v = u^2$$

Ecriture quadratique des équations principales

$$R_1 := u + u * v - \lambda = 0$$

Ecriture quadratique des équations définissant les variables auxiliaires

$$R_{aux1} := v - u^2 = 0$$

Finalement, le vecteur d'inconnues et le vecteur d'équation sont :

$$U_{tot} = [u\ \lambda\ v]^T$$

$$R_{tot} = [R_1\ R_{aux1}]^T$$

# Quadratic recast of algebraic systems

Exemple 2 : simple, sans fonction
Equations principales ($\mu = cst$) :

$$r_1(u_1, u_2, \lambda) := 2u_1 - u_2 + 100\,\frac{u_1}{1+u_1+u_1^2} - \lambda = 0$$
$$r_2(u_1, u_2, \lambda) := 2u_2 - u_1 + 100\,\frac{u_2}{1+u_2+u_2^2} - \lambda - \mu = 0$$

Definition des variables auxiliaires :

$v_1 =$

$v_2 =$

$v_3 =$

$v_4 =$

# Quadratic recast of algebraic systems

Exemple 2 : simple, sans fonction
Equations principales ($\mu = cst$) :

$$r_1(u_1, u_2, \lambda) := 2u_1 - u_2 + 100\ \frac{u_1}{1+u_1+u_1^2} - \lambda = 0$$
$$r_2(u_1, u_2, \lambda) := 2u_2 - u_1 + 100\ \frac{u_2}{1+u_2+u_2^2} - \lambda - \mu = 0$$

Definition des variables auxiliaires :

$$v_1 = 1 + u_1 + u_1^2$$

$$v_2 = 1 + u_2 + u_2^2$$

$$v_3 = \frac{1}{v_1}$$

$$v_4 = \frac{1}{v_2}$$

- All these expression are quadratic, or easily made quadratic
- "linear declaration rule" : an auxiliary variable $v_i$ cannot appear on the left hand side before it has been explicitely defined as $v_i = f(\mathbf{U}, v_1, v_2, \ldots, v_{i-1})$.

# Quadratic recast of algebraic systems

Exemple 2 : simple, sans fonction
Equations principales ($\mu = cst$) :

$$r_1(u_1, u_2, \lambda) := 2u_1 - u_2 + 100\ \frac{u_1}{1+u_1+u_1^2} - \lambda = 0$$
$$r_2(u_1, u_2, \lambda) := 2u_2 - u_1 + 100\ \frac{u_2}{1+u_2+u_2^2} - \lambda - \mu = 0$$

Definition des variables auxiliaires :

$$v_1 = 1 + u_1 + u_1^2$$

$$v_2 = 1 + u_2 + u_2^2$$

$$v_3 = \frac{1}{v_1}$$

$$v_4 = \frac{1}{v_2}$$

Ecriture quadratique des équations principales

$$R_1 := 2u_1 - u_2 + 100\ u_1 * v_3 - \lambda = 0$$

$$R_2 := 2u_2 - u_1 + 100; u_2 * v_4 - \lambda - \mu = 0$$

# Quadratic recast of algebraic systems

Exemple 2 : simple, sans fonction
Definition des variables auxiliaires :

$$v_1 = 1 + u_1 + u_1^2$$

$$v_2 = 1 + u_2 + u_2^2$$

$$v_3 = \frac{1}{v_1}$$

$$v_4 = \frac{1}{v_2}$$

Ecriture quadratique des équations définissant les variables auxiliaires

$$R_{aux1} := v_1 - 1 + u_1 + u_1^2 = 0$$

$$R_{aux2} := v_2 - 1 + u_2 + u_2^2 = 0$$

$$R_{aux3} := v_3 * v_1 - 1 = 0$$

$$R_{aux4} := v_4 * v_2 - 1 = 0$$

# Quadratic recast of algebraic systems

Exemple 3 : un exemple mécanique simple
Equations principales :

$$r(u,\lambda) := \varepsilon * \frac{\partial \varepsilon}{\partial u} - \lambda = 0$$

Definition des variables auxiliaires :

$$eps = -\frac{u}{2} + \frac{u^2}{4}$$

Ecriture quadratique des équations principales

$$R_1 := eps * (-\frac{1}{2} + \frac{u}{2}) - \lambda = 0$$

Ecriture quadratique des équations définissant les variables auxiliaires

$$R_{aux1} := eps + \frac{u}{2} - \frac{u^2}{4} = 0$$

Finalement, le vecteur d'inconnues et le vecteur d'équation sont :

$$U_{tot} = [u\ \lambda\ eps]^T$$

$$R_{tot} = [R_1\ R_{aux1}]^T$$

# Quadratic recast with elementary function

let $u$ and $v$ related by

$$v = exp(u)$$

Assume we know

$$u(a) = u_0 + a\, u_1 + a^2 u_2 + \cdots + a^N u_N$$

and want to compute

$$v(a) = v_0 + a\, v_1 + a^2 v_2 + \cdots + a^N v_N$$

this can be done easily by considering the differential

$$dv = exp(u)du \qquad \rightarrow \qquad dv = v * du \qquad \text{(quadratic)}$$

Hence, elementary function should appears isolated in the list of auxiliary variable, and a quadratic differential form shoud be given also. See example 4.

# Quadratic recast of algebraic systems

Exemple 6 : pour récapituler

$$r_1(u_1, u_2, \lambda) = u_1 + \lambda \frac{\exp(u_2)}{1+u_1}$$
$$r_2(u_1, u_2, \lambda) = u_2 + u_1 \tanh(\frac{-5u_1}{1+u_1 u_2})$$

Definition des variables auxiliaires :

$$\begin{aligned} v_1 &= \exp(u_2) \\ v_2 &= \frac{v_1}{1+u_1} \\ v_3 &= 1 + u_1 u_2 \\ v_4 &= -\frac{5u_1}{v_3} \\ v_5 &= \tanh(v_4) \\ v_6 &= 1 + v_5^2 \end{aligned}$$

Ecriture quadratique des équations principales

$$\begin{aligned} r_1 &= u_1 + \lambda v_2 \\ r_2 &= u_2 + u_1 v_5 \end{aligned}$$

$$\begin{aligned} R_{aux1} &:= v_1 - exp(u_2) = 0 \\ R_{aux2} &:= v_1 - v_2 - v_2 u_1 = 0 \\ R_{aux3} &:= 1 + u_1 u_2 - v - 3 = 0 \\ R_{aux4} &:= -5u_1 - v_4 v - 3 = 0 \\ R_{aux5} &:= v_5 - tanh(v4) = 0 \\ R_{aux6} &:= 1 + v_5^2 - v_6 \end{aligned}$$

$$dR_{aux1} = dv_1 - v_1 dv_2 = 0$$

$$dR_{aux5} = dv_5 - v_6 dv_4 = 0$$

# Demo Manlab-4.0 on algebraic system

- ECrea
- Elasticite 2D nlgeom

# TP Manlab-4.0 on algebraic system

## It's your turn

- exemple 3 : jambière
- exemple 5 : instabilité d'une barre en rotation

# Taylor serie based continuation

Goal : determine solution branches of

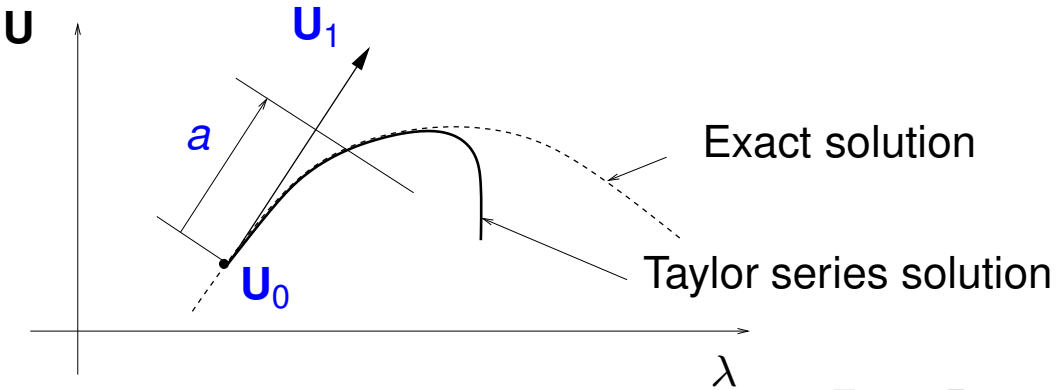
$$\mathbf{R}(\mathbf{U}) = 0 \quad \text{with} \quad \mathbf{R} \in \mathbb{R}^n \quad \text{and} \quad \mathbf{U} = [u, \lambda] \in \mathbb{R}^{n+1}$$

- let $\mathbf{U}_0$ be a regular point with $||\mathbf{R}(\mathbf{U}_0)|| < \varepsilon_R$ (tolerance).
- let $\mathbf{U}_1$ be the tangent at $\mathbf{U}_0$. ( $\frac{\partial \mathbf{R}}{\partial \mathbf{U}}\mathbf{U}_1 = 0$ size : $n \times (n+1)$).
- let $a$ be the pseudo arc length parameter $a = \mathbf{U}_1^T.(\mathbf{U} - \mathbf{U}_0)$.

IFT theorem : The solution branch passing through $\mathbf{U}_0$ may be represented as a (truncated) Taylor series with respect to the pseudo-arclength parameter $a$.

$$\mathbf{U}(a) = \mathbf{U}_0 + a\mathbf{U}_1 + a^2\mathbf{U}_2 + \cdots + a^N\mathbf{U}_N \qquad \text{with } \mathbf{U} = [U, \lambda] \qquad \text{and } N = 20 \text{ or } 30$$

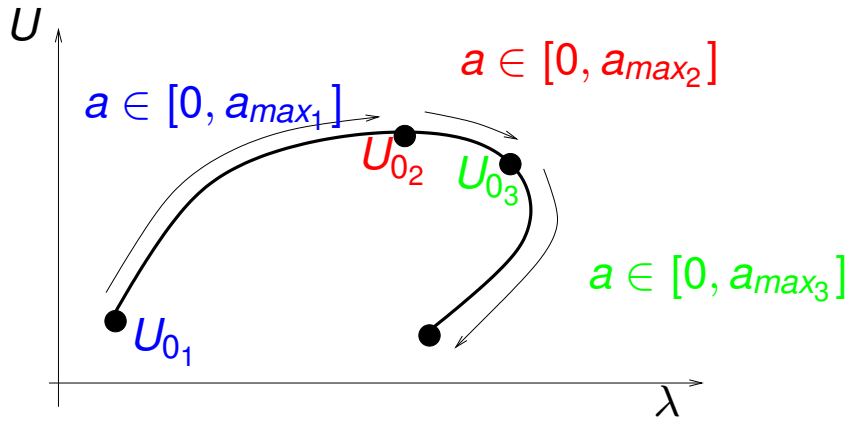
Series computation : solve a succession of linear systems that share the same stiffness matrix $\frac{\partial \mathbf{R}}{\partial \mathbf{U}}\mathbf{U}_p = F_p^{nl}(\mathbf{U}_1, \ldots, \mathbf{U}_{p-1})$



The Domain of utility of the series is the interval $[0, a_{max}]$ for which $||\mathbf{R}(\mathbf{U}(a))|| < \varepsilon_R$

A good approximation : $\mathbf{R}(\mathbf{U}_0 + \cdots + a^N\mathbf{U}_N) \simeq \mathbf{R}(\mathbf{U}_0) + a^{N+1}\mathbf{R}^{N+1}$,

So, if one requires $||a^{N+1}\mathbf{R}^{N+1}|| < \varepsilon_R$ then $a_{max} = \left(\frac{\varepsilon_R}{||\mathbf{R}^{N+1}||}\right)^{\frac{1}{N+1}}$




The complete solution branch is obtained as a succession of local Taylor series

$$\mathbf{U}(a) = \mathbf{U}_0 + a\mathbf{U}_1 + a^2\mathbf{U}_2 + \cdots + a^N\mathbf{U}_N \quad \text{avec} \quad a \in [0, a_{max}]$$

.

- piece-wize continuous representation
- auto-adaptative step length → Robustness (but small steps accumulation drawback near bifurcation)
- no algorithmic parameter

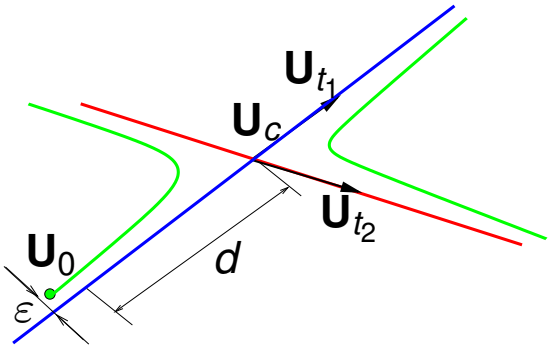
# Bifurcation detection using series analysis

Numerical evidence : near a simple bifurcation, a geometric series emerge in the Taylor series.

$$\mathbf{U}(a) = \mathbf{U}_0 + a\,\mathbf{U}_1 + a^2\,\mathbf{U}_2 + a^3\,\mathbf{U}_3 + \dots$$

First order analysis : expression of a perturbed branches near a simple bifurcation [Cochelin & Médale, 2013]

$$\mathbf{U}(a) = \mathbf{U}_0 + a\,\mathbf{U}_{t_1} - \varepsilon\,\frac{\frac{a}{d}}{(1-\frac{a}{d})}\mathbf{U}_{t_2}$$




- $\frac{\frac{a}{d}}{1-\frac{a}{d}} = \frac{a}{d} + (\frac{a}{d})^2 + (\frac{a}{d})^3 + \cdots$, a geometric serie with common ratio $\frac{1}{d}$
- After each Taylor series computation, we look for an emerging geometric series. When detected , it is extracted, completed to infinity and replaced by a fraction

$$\mathbf{U}(a) = \underbrace{\mathbf{U}_0 + a\,\hat{\mathbf{U}}_1 + a^2\,\hat{\mathbf{U}}_2 + \cdots + a^{n-1}\,\hat{U}_{n-1}}_{\hat{\mathbf{U}}(a)\ \text{cleanned series}} + \frac{a}{d}\left(\frac{1}{1-\frac{a}{d}}\right)\mathbf{U}_{scale}$$

We get $d$, $\mathbf{U}_{t2}$ and can go further the bifurcation thanks to the cleanned series.

# Manlab 4.0

Let $R(\mathbf{U}) = 0$ be

$$\begin{aligned} r_1(u_1, u_2, \lambda) = &\quad 2u_1 - u_2 + 100\frac{u_1}{1+u_1+u_1^2} - \lambda &= 0 \\ r_2(u_1, u_2, \lambda) = &\quad 2u_2 - u_1 + 100\frac{u_2}{1+u_2+u_2^2} - (\lambda + \mu) &= 0 \end{aligned}$$

Definition of the auxiliary variables

$$\begin{aligned} v_1 &= 1 + u_1 + u_1 u_1 \\ v_2 &= 1 + u_2 + u_2 u_2 \\ v_3 &= 1/v_1 \\ v_4 &= 1/v_2 \end{aligned}$$

- All these expression are quadratic, or easily made quadratic
- "linear declaration rule" : an auxiliary variable $v_i$ cannot appear on the left hand side before it has been explicitely defined as $v_i = f(\mathbf{U}, v_1, v_2, \ldots, v_{i-1})$.

Let $\mathbf{U}_{aux} = [v_1, v_2, v_3, v_4]$ be the vector of auxiliary variables
Let $\mathbf{U}_{tot} = [\mathbf{U}, \mathbf{U}_{aux}]$

# Manlab 4.0

The original system $R(\mathbf{U}) = 0$ is replaced by the equivalent quadratic one $\mathbf{R}(\mathbf{U}_{tot})$

$$\begin{array}{lll} r_{aux1} := & v_1 - 1 + u_1 + u_1 * u_1 & = 0 \\ r_{aux2} := & v_2 - 1 + u_2 + u_2 * u_2 & = 0 \\ r_{aux3} := & v_3 * v_1 - 1 & = 0 \\ r_{aux4} := & v_4 * v_2 - 1 & = 0 \\ r_1 := & 2u_1 - u_2 + 100u_1 v_3 - \lambda & = 0 \\ r_2 := & 2u_2 - u_1 + 100u_2 v_3 - (\lambda + \mu) & = 0 \end{array}$$

Tensor formalism : this quadratic system may be written

$$R_i = C_i + L_{ij} U_j + Q_{ijk} U_j U_k \qquad i, j, k = 1, 2, \ldots n$$

with $C$,$L$,$Q$ being tensors of order 1, 2 and 3
Here, we have 7 components $C_i$, 49 components $L_{ij}$ and 343 components $Q_{ijk}$.

But most of them are zero !

# Manlab 4.0 : Sparse tensor formalism

The sparse tensor $C$, $L$ and $Q$ are defined by the following lists (as in Matlab for a sparse matrix)

- order 1 tensor $C$

```
iC= [ 2   5   6 ]
vC= [-μ  -1   -1 ]
```

- order 2 tensor $L$

```
iL= [1   1   1   2   2   2   3   3   4   4   5   6 ]
jL= [1   2   7   1   2   7   1   3   2   4   5   6 ]
vL= [2 -1  -1 -1   2   -1 -1   1 -1   1   1   1 ]
```

- order 3 tensor $Q$

```
iQ= [ 1   2   3   4   5   6 ]
jQ= [ 1   2   1   2   3   4 ]
kQ= [ 5   6   1   2   5   6 ]
vQ= [100 100 -1 -1   1   1 ]
```

In Manlab 4.0, these lists are automatically generated from the quadratic system.

# Manlab 4.0 : Sparse tensor formalism

How to get the lists defining the sparse tensor, from the quadratic expression $R(X) := 0$ ?

Polarization formula :

$$C = R(0)$$

$$L(X) = \frac{1}{2}\left(R(X) - R(-X)\right)$$

$$Q(X, Y) = \frac{1}{4}\left(R(X + Y) - R(X - Y) - R(Y) + R(-Y)\right)$$

# Manlab 4.0 : Sparse tensor formalism

Using these lists, the computation of the residual vector $R(\mathbf{U}) = C + L(\mathbf{U}) + Q(\mathbf{U}, \mathbf{U})$ stand in one (Matlab) line.

```
R =sparse(iC,ones(1,size(iC,2)),vC',neq,1)
+ sparse(iL,ones(1,size(iL,2)),vL'.*U(jL),neq,1)
+ sparse(iQ,ones(1,size(iQ,2)),vQ'.*(U(jQ).*U(kQ)),neq,1)
```

For the jacobian matrix $dRdU = L(.) + Q(\mathbf{U}, .) + Q(., \mathbf{U})$

```
dRdU = sparse(iL,jL,sys.vL,neq,ninc)
+ sparse(iQ,kQ,vQ'.*U(jQ),neq,ninc)
+ sparse(iQ,jQ,vQ'.*U(kQ),neq,ninc)
```

# Manlab 4.0 : condensation

The linear problem to be solved at each order $p$ reads :

$$\begin{bmatrix} B & A_{aux} \\ A & C \end{bmatrix} \begin{bmatrix} \mathbf{U} \\ \mathbf{U}_{aux} \end{bmatrix} = \begin{bmatrix} F_{aux\ p} \\ F_p \end{bmatrix} \qquad \begin{matrix} \leftarrow R_{aux} \\ \leftarrow R \end{matrix}$$

Thanks to the "linear declaration rule", the matrix $A_{aux}$ is triangular which allows an easy and cheap block solving

We first solve

$$\left[A - C\ A_{aux}^{-1}\ B\right]\ [U] = \left[F_p - C\ A_{aux}^{-1} F_{aux\ p}\right]$$

where $\left[A - C\ A_{aux}^{-1}\ B\right]$ is the jacobian matrix of the original (non quadratic) system.