Higher-order continuation for the determination of robot workspace boundaries

Presentation at the JJCR in Amiens

Gauthier Hentz
Isabelle Charpentier
Pierre Renaud

Control, Vision and Robotics Lab - ICube - Université de Strasbourg

October 20th 2015

Université de Strasbourg

ICUBE

Outline

Objective I

Context

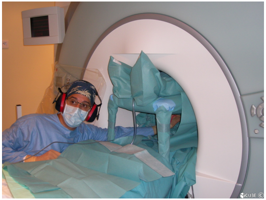
- Very specific applications in medical robotics, e.g. interventional radiology
- Need for compact architectures

Issue

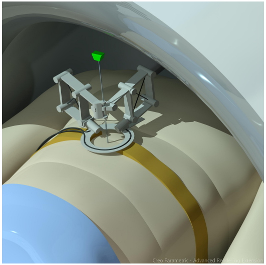
- Synthesis of **optimized architectures** remains a challenge

Proposition

- Development of numerical tools for the **systematic study** of mechanisms
- First property of interest: the **mechanism workspace**



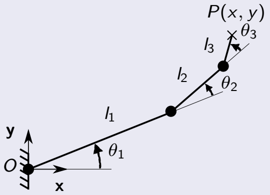
$\downarrow$?



Objective II

Given a mechanism and its non-linear loop-closure equations

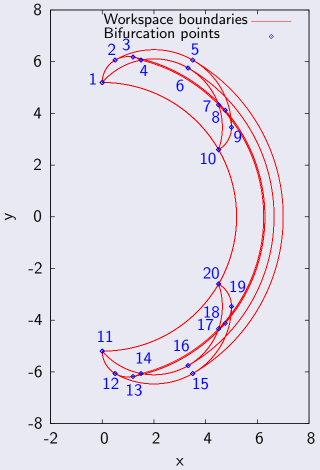
$$\begin{cases} x - (l_1 cos(\theta_1) + l_2 cos(\theta_1 + \theta_2) \\ \quad + l_3 cos(\theta_1 + \theta_2 + \theta_3)) = 0 \\ y - (l_1 sin(\theta_1) + l_2 sin(\theta_1 + \theta_2) \\ \quad + l_3 sin(\theta_1 + \theta_2 + \theta_3)) = 0 \end{cases}$$



- Joint constraints: $\theta_i \in [-\pi/3; \pi/3]$
- Variable change: $\theta_i = \pi/3 \sin(v_i)$

Fast, general, $\longrightarrow$ automatic method

Generate the mechanism reachable workspace "with one click"



Existing workspace analysis methods (overview)

- Purely geometric (Gosselin [1990]) or analytic (Abdel-Malek and Yeh [1997]) :
  - Precise and continuous solutions.
  - Specific to an architecture or a class of mechanisms
- Interval analysis (Merlet [1999], Bohigas et al. [2012]) :
  - Generic, and guaranteed solutions.
  - Difficult tradeoff accuracy/computation time, especially in high-dimensional workspaces
- Continuation on the workspace boundaries (Haug et al. [1996]) :
  - Fast, accurate, and general.
  - Computation of the Jacobian, 1D paths, and discrete results.

- To be improved: tradeoff accuracy - computation time - generality
- Most appropriate method for a systematic study : Haug et al. [1996]
- Its drawbacks have limited its dissemination

Equation system of the workspace boundaries (Haug et al. [1996])

- Operational coordinates $\mathbf{u}$, joint coordinates $\mathbf{v}$, constraint equations

$$\mathbf{R}^{\mathcal{W}}(\mathbf{u},\mathbf{v}) = \mathbf{0} \tag{1}$$

- Workspace definition:

$$\mathcal{W} = \left\{\mathbf{u} \mid \exists\ \mathbf{v} \text{ such that } \mathbf{R}^{\mathcal{W}}(\mathbf{u},\mathbf{v}) = 0\right\}$$

- **Workspace boundaries** defined as configurations where the Jacobian $\mathbf{R}^{\mathcal{W}}_{\mathbf{v}}$ is rank deficient
- Solutions of the following **extended residual system**:

$$\mathbf{R}^{\partial\mathcal{W}}(\mathbf{u},\mathbf{v},\xi) = \begin{bmatrix} \mathbf{R}^{\mathcal{W}}(\mathbf{u},\mathbf{v}) \\ \left(\mathbf{R}^{\mathcal{W}}_{\mathbf{v}}\right)^{\mathsf{T}}(\mathbf{u},\mathbf{v}).\xi \\ \xi^{\mathsf{T}}.\xi - 1 \end{bmatrix} = \mathbf{0}. \tag{2}$$

Plan

Stage 1: Automatic generation of the boundary equations

User tasks

(1) Define $\mathbf{R}^{\mathcal{W}}(\mathbf{u},\mathbf{v})$ and $(\mathbf{u},\mathbf{v})$

(2) Implement the augmented system

Automated tasks

(3) Differentiate with respect to $\mathbf{v}$ using **Automatic Differentiation**

$$\mathrm{R}^{\mathcal{W}}(\mathbf{u},\mathbf{v}) \longrightarrow \begin{pmatrix} \mathbf{R}^{\mathcal{W}}(\mathbf{u},\mathbf{v}) \\ \left(\mathbf{R}^{\mathcal{W}}\right)^{\mathsf{T}}(\mathbf{u},\mathbf{v}).\xi \\ \xi^{\mathsf{T}}.\xi - 1 \end{pmatrix} \longrightarrow \begin{pmatrix} \mathbf{R}^{\mathcal{W}}(\mathbf{u},\mathbf{v}) \\ \mathbf{R}^{\mathcal{W}}_{\mathbf{v}}(\mathbf{u},\mathbf{v}) \\ \left(\mathbf{R}^{\mathcal{W}}\right)^{\mathsf{T}}(\mathbf{u},\mathbf{v}).\xi \\ \left(\mathbf{R}^{\mathcal{W}}_{\mathbf{v}}\right)^{\mathsf{T}}(\mathbf{u},\mathbf{v}).\xi \\ \xi^{\mathsf{T}}.\xi - 1 \end{pmatrix} \longrightarrow \mathbf{R}^{\partial\mathcal{W}}(\mathbf{u},\mathbf{v},\xi)$$

→ The Jacobian is computed systematically

Stage 1: Automatic generation of the boundary equations

User tasks

(1) Define $\mathbf{R}^{\mathcal{W}}(\mathbf{u},\mathbf{v})$ and $(\mathbf{u},\mathbf{v})$

(2) Implement the augmented system

Automated tasks

(3) Differentiate with respect to $\mathbf{v}$ using **Automatic Differentiation**

$$\mathbf{R}^{\mathcal{W}}(\mathbf{u},\mathbf{v}) \longrightarrow \begin{pmatrix} \mathbf{R}^{\mathcal{W}}(\mathbf{u},\mathbf{v}) \\ \left(\mathbf{R}^{\mathcal{W}}\right)^{\mathsf{T}}(\mathbf{u},\mathbf{v}).\xi \\ \xi^{\mathsf{T}}.\xi - 1 \end{pmatrix} \longrightarrow \begin{pmatrix} \mathbf{R}^{\mathcal{W}}(\mathbf{u},\mathbf{v}) \\ \mathbf{R}^{\mathcal{W}}_{\mathbf{v}}(\mathbf{u},\mathbf{v}) \\ \left(\mathbf{R}^{\mathcal{W}}\right)^{\mathsf{T}}(\mathbf{u},\mathbf{v}).\xi \\ \left(\mathbf{R}^{\mathcal{W}}_{\mathbf{v}}\right)^{\mathsf{T}}(\mathbf{u},\mathbf{v}).\xi \\ \xi^{\mathsf{T}}.\xi - 1 \end{pmatrix} \longrightarrow \mathbf{R}^{\partial\mathcal{W}}(\mathbf{u},\mathbf{v},\xi)$$

→ The Jacobian is computed systematically

Stage 1: Automatic generation of the boundary equations

User tasks

(1) Define $\mathbf{R}^{\mathcal{W}}(\mathbf{u},\mathbf{v})$ and $(\mathbf{u},\mathbf{v})$

(2) Implement the augmented system

Automated tasks

(3) Differentiate with respect to $\mathbf{v}$ using **Automatic Differentiation**

$$\mathrm{R}^{\mathcal{W}}(\mathbf{u},\mathbf{v}) \longrightarrow \begin{pmatrix} \mathbf{R}^{\mathcal{W}}(\mathbf{u},\mathbf{v}) \\ \left(\mathbf{R}^{\mathcal{W}}\right)^{\mathsf{T}}(\mathbf{u},\mathbf{v}).\xi \\ \xi^{\mathsf{T}}.\xi - 1 \end{pmatrix} \longrightarrow \begin{pmatrix} \mathbf{R}^{\mathcal{W}}(\mathbf{u},\mathbf{v}) \\ \mathbf{R}^{\mathcal{W}}_{\mathbf{v}}(\mathbf{u},\mathbf{v}) \\ \left(\mathbf{R}^{\mathcal{W}}\right)^{\mathsf{T}}(\mathbf{u},\mathbf{v}).\xi \\ \left(\mathbf{R}^{\mathcal{W}}_{\mathbf{v}}\right)^{\mathsf{T}}(\mathbf{u},\mathbf{v}).\xi \\ \xi^{\mathsf{T}}.\xi - 1 \end{pmatrix} \longrightarrow \mathbf{R}^{\partial\mathcal{W}}(\mathbf{u},\mathbf{v},\xi)$$

→ The Jacobian is computed systematically

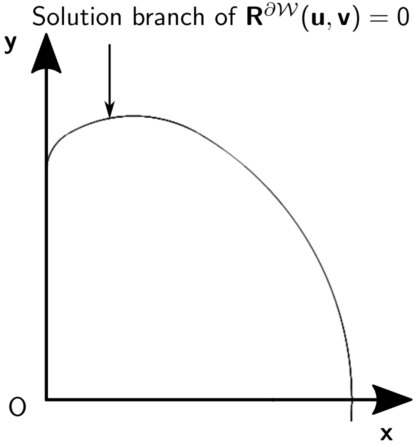
Stage 2: Higher-order continuation principle

First-order continuation (Haug 1996)

(1) Initial Point
(2) Newton-Raphson
(3) Solution Point
(4) Jacobian evaluation
(5) First-order continuation

Higher-order continuation

(4) Higher-Order differentiation
(5) Taylor polynomial of $\mathbf{R}^{\partial\mathcal{W}}$ in $a = 0$
(6) Solution (u(a),v(a))
(7) Validity domain computation $a \in [0, a_{max}]$



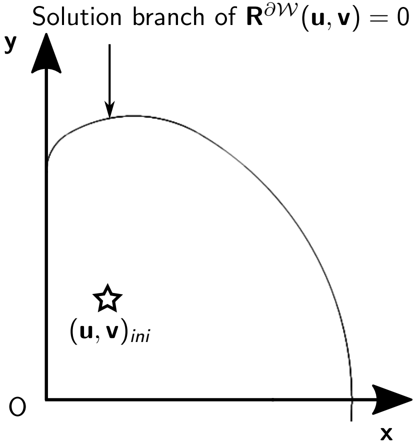
Stage 2: Higher-order continuation principle

First-order continuation (Haug 1996)

(1) Initial Point
(2) Newton-Raphson
(3) Solution Point
(4) Jacobian evaluation
(5) First-order continuation

Higher-order continuation

(4) Higher-Order differentiation
(5) Taylor polynomial of $\mathbf{R}^{\partial\mathcal{W}}$ in $a = 0$
(6) Solution (u(a),v(a))
(7) Validity domain computation $a \in [0, a_{max}]$



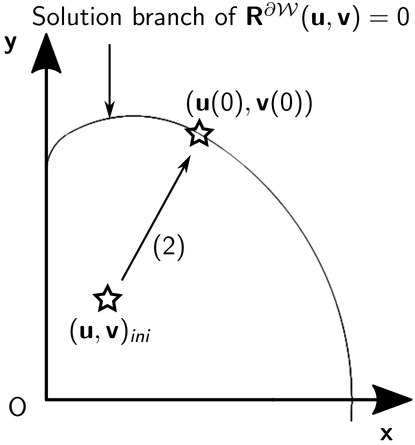
Stage 2: Higher-order continuation principle

First-order continuation (Haug 1996)

(1) Initial Point
(2) Newton-Raphson
(3) Solution Point
(4) Jacobian evaluation
(5) First-order continuation

Higher-order continuation

(4) Higher-Order differentiation
(5) Taylor polynomial of $\mathbf{R}^{\partial \mathcal{W}}$ in $a = 0$
(6) Solution (u(a),v(a))
(7) Validity domain computation $a \in [0, a_{max}]$



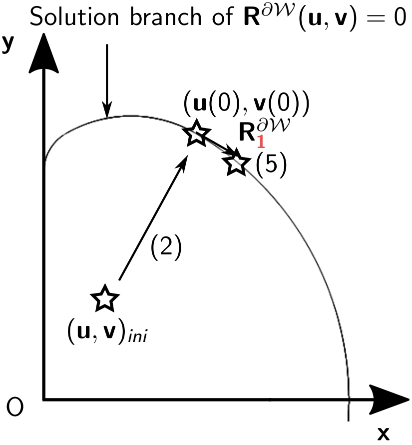
Stage 2: Higher-order continuation principle

First-order continuation (Haug 1996)

(1) Initial Point
(2) Newton-Raphson
(3) Solution Point
(4) Jacobian evaluation
(5) First-order continuation

Higher-order continuation

(4) Higher-Order differentiation
(5) Taylor polynomial of $\mathbf{R}^{\partial \mathcal{W}}$ in $a = 0$
(6) Solution (u(a),v(a))
(7) Validity domain computation $a \in [0, a_{max}]$



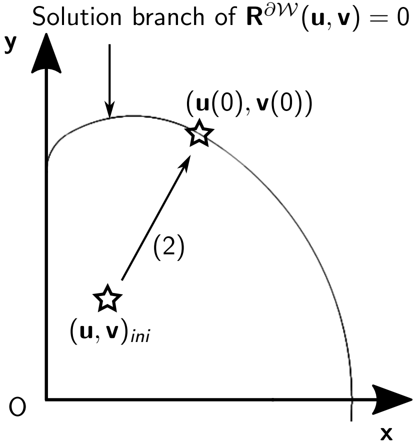
Stage 2: Higher-order continuation principle

First-order continuation (Haug 1996)

(1) Initial Point
(2) Newton-Raphson
(3) Solution Point
(4) Jacobian evaluation
(5) First-order continuation

Higher-order continuation

(4) Higher-Order differentiation
(5) Taylor polynomial of $\mathbf{R}^{\partial \mathcal{W}}$ in $a = 0$
(6) Solution (u(a),v(a))
(7) Validity domain computation $a \in [0, a_{max}]$



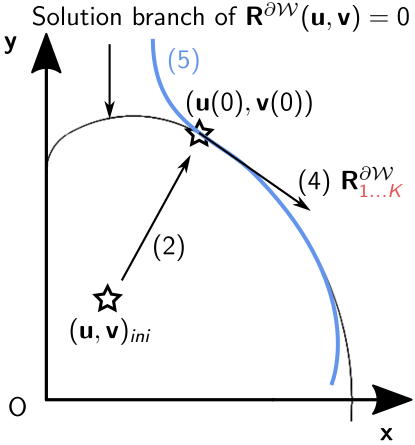
Stage 2: Higher-order continuation principle

First-order continuation (Haug 1996)

(1) Initial Point
(2) Newton-Raphson
(3) Solution Point
(4) Jacobian evaluation
(5) First-order continuation

Higher-order continuation

(4) Higher-Order differentiation
(5) Taylor polynomial of $\mathbf{R}^{\partial\mathcal{W}}$ in $a = 0$
(6) Solution (u(a),v(a))
(7) Validity domain computation $a \in [0, a_{max}]$



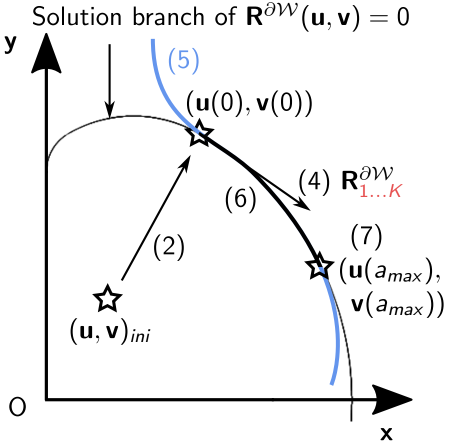
Stage 2: Higher-order continuation principle

First-order continuation (Haug 1996)

(1) Initial Point
(2) Newton-Raphson
(3) Solution Point
(4) Jacobian evaluation
(5) First-order continuation

Higher-order continuation

(4) Higher-Order differentiation
(5) Taylor polynomial of $\mathbf{R}^{\partial\mathcal{W}}$ in $a = 0$
(6) Solution (u(a),v(a))
(7) Validity domain computation $a \in [0, a_{max}]$



Plan

RRR workspace boundaries I

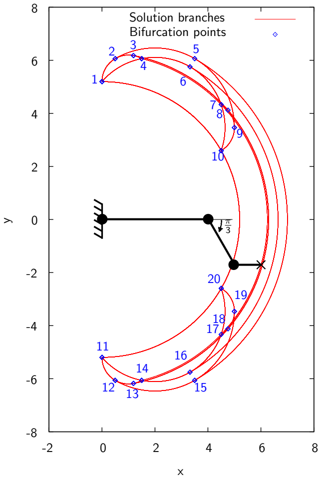
Simulation parameters

l_1	4
l_2	2
l_3	1
θ_i	$\in [-\pi/3; \pi/3]$

$\mathbf{u}_0$	$(7,0)$
$\mathbf{v}_0$	$(0,0,0)$
ξ_0	$(1,0)$
ϵ_{series}	$1e-6$
ϵ_{N-R}	$1e-5$

Results

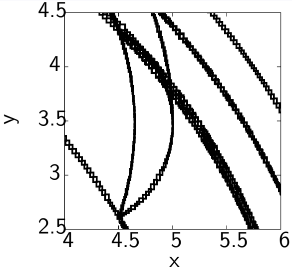
- 20 **solution branches** connected by **bifurcation points**
- Automatic detection of bifurcation points and branch switching
- Branches related to different Inverse Kinematics Solutions may overlap

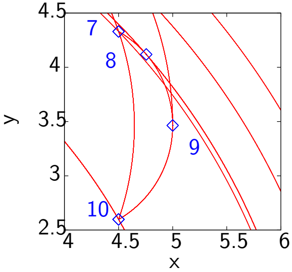


RRR workspace boundaries II

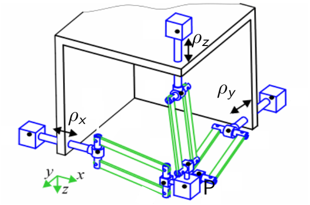
- Compared to Haug and Bohigas, solution branches are continuous thanks to the Taylor series
- Even close boundaries can be identified easily
- Very good compromise between computation time and accuracy

Method	Computation time	Accuracy ϵ
Branch-and-prune	1m15s	$5e-2$
Branch-and-prune	196m42s	$1e-4$
Diamant	**7s**	$\mathbf{1e-6}$

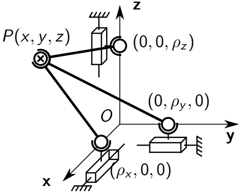




Orthoglide presentation (Pashkevich et al. [2006])



↓



- Simplified kinematics
- Isotropic configuration $(x, y, z) = (0, 0, 0)$, $(\rho_x, \rho_y, \rho_z) = (l, l, l)$

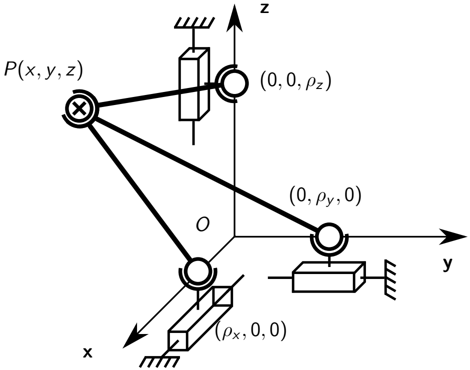
Orthoglide workspace boundaries

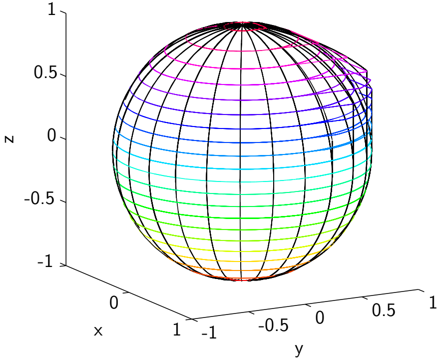
User tasks

- 3D WS $\rightarrow$ intersection with a plan
- Solve discretized system as done for the RRR
- Assembly of curves to build the workspace

Results

- Sets of isoaltitude and isoorientation slices of the mechanism workspace boundaries
- Convex surface corresponding to a leg orthogonal to its axis of actuation





Method

Contribution

- Systematic generation of the extended equation system characterizing the workspace boundaries of any mechanism and related sets of equations
- Good results with the RRR (2D workspace) and the Orthoglide (3D workspace)
- Good ratio accuracy/computational load

Limits

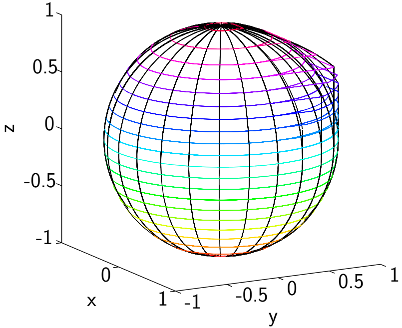
- 1D path
- Need of a strategy to ensure the exhaustive detection of workspaces with multiple components that are not connected

Future works

- Further validation of the method on higher-dimensional workspaces
- Reconstruct surfaces using the Taylor series
- Integration of the method in an optimization scheme

Thanks for your attention

Do you have Questions?



References

Karim Abdel-Malek and Harn-Jou Yeh. Analytical boundary of the workspace for general 3-DOF mechanisms. *International Journal of Robotics Research*, 16(2):198–213, 1997. ISSN 0278-3649.

O. Bohigas, M. Manubens, and L. Ros. A complete method for workspace boundary determination on general structure manipulators. *IEEE Transactions on Robotics*, 28(5):993–1006, October 2012. ISSN 1552-3098. doi: 10.1109/TRO.2012.2196311.

I. Charpentier. On higher-order differentiation in nonlinear mechanics. *Optimization Methods and Software*, 27(2):221–232, 2012. ISSN 1055-6788. doi: 10.1080/10556788.2011.577775.

C. Gosselin. Determination of the workspace of 6-DOF parallel manipulators. *Journal of Mechanical Design*, 112(3):331–336, September 1990. ISSN 1050-0472. doi: 10.1115/1.2912612. URL http://dx.doi.org/10.1115/1.2912612.

L. Hascoet and V. Pascual. The tapenade automatic differentiation tool: Principles, model, and specification. *ACM Transactions on Mathematical Software*, 39(3), 2013. ISSN 0098-3500. doi: 10.1145/2450153.2450158.

E.J. Haug, C.-M. Luh, F.A. Adkins, and J.-Y. Wang. Numerical algorithms for mapping boundaries of manipulator workspaces. *Journal of Mechanical Design, Transactions of the ASME*, 118(2):228–234, 1996. ISSN 1050-0472.

J.-P. Merlet. Determination of 6D-workspaces of gough-type parallel manipulator and comparison between different geometries. *The International Journal of Robotics Research*, 18(9):902–916, 1999. ISSN 0278-3649.

A. Pashkevich, D. Chablat, and P. Wenger. Kinematics and workspace analysis of a three-axis parallel manipulator: the orthoglide. *Robotica*, 24(01):39–49, January 2006. ISSN 1469-8668. doi: 10.1017/S0263574704000347. URL http://journals.cambridge.org/article_S0263574704000347.

Introducing Automatic Differentiation

Existing differentiation methods

- Hand-written: Differentiation rules
- Numerical (finite differences): $f'(a) = \lim_{h \to 0} \frac{f(a+h) - f(a)}{h}$
- Symbolic: Maple

Automatic differentiation of a computer code

- Decomposition of an expression according to the chain rule $(g \circ f)'(u) = g'(f(u)) \cdot f'(u)$
- Differentiation using basic $(sin(x))' = x' cos(x)$ and combined $(fg)' = f'g + fg'$ rules

Differentiation of a source code with Tapenade Hascoet and Pascual [2013] http://www-tapenade.inria.fr:8080/tapenade/index.jsp

Implemented code $\xrightarrow[\text{differentiation}]{\text{forward (tangent)}}$ Differentiated code

Stage 2: II - Higher-order continuation principle

- Under-determined residual equations:

$$\mathbf{R}^{\partial \mathcal{W}}(\mathbf{U}) = 0$$

with $\mathbf{U} = (\mathbf{u}, \mathbf{v}, \xi)$

- Path equation:

$$a = \left\langle \mathbf{U}(a) - \mathbf{u}(0), \frac{\partial \mathbf{U}}{\partial a}(0) \right\rangle$$

- **Solutions as Taylor series**:

$$\mathbf{U}(a) = \sum_{k=0}^{K} a^k \mathbf{U}_k \text{ with } \textbf{Taylor coefficients } \mathbf{U}_k = \frac{1}{k!} \frac{\partial^k \mathbf{U}}{\partial a^k}(0)$$

Stage 2: III - Diamant method (Charpentier [2012])

- Taylor series introduced in the system to solve:

$$\mathbf{R}^{\partial\mathcal{W}}(\mathbf{U}(a)) = \sum_{k=0}^{K} a^k \mathbf{R}_k^{\partial\mathcal{W}} = \mathbf{0}$$

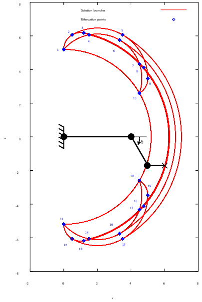
- System decomposed in an iterative sequence of K linear systems:

$$\begin{cases} \{\mathcal{R}_1\}\,\mathbf{U}_1 = -\left\{\mathcal{R}_{1|\mathbf{U}_1=\mathbf{0}}\right\}, & \langle \mathbf{U}_1, \mathbf{U}_1 \rangle = 1, \\ \{\mathcal{R}_1\}\,\mathbf{U}_k = -\left\{\mathcal{R}_{k|\mathbf{U}_k=\mathbf{0}}\right\}, & \langle \mathbf{U}_k, \mathbf{U}_1 \rangle = 0, \\ \quad\vdots & \text{for } k = 2, .., K. \end{cases}$$

- Requires higher-order differentiation of $\mathbf{R}^{\partial\mathcal{W}}(\mathbf{U}(a))$
- Can be computed efficiently with **Automatic Differentiation** (operator overloading)
- The continuous solution branches are computed systematically and faster

Stage 2: IV - Diamanlab implementation (Charpentier [2012])

- Screencast of interactive continuation on the workspace boundaries of the planar RRR



- *Diamanlab v1.0* is freely available at the bottom of the following download page:
  `http://manlab.lma.cnrs-mrs.fr/spip/spip.php?rubrique1`

Higher-order differentiation by operator overloading

Definition of a Taylor type comprising the variable derivatives (Taylor coefficients)

Taylor
Order Value Coefficient

Operator overloading through recurrence formula

function	recurrence formula k
$w = u + v$	$w_k = u_k + v_k$
$w = u \cdot v$	$w_k = \sum_{j=0}^{k} u_j v_{k-j}$
$w = \frac{u}{v}$	$w_k = \frac{1}{v_0} \left[u_k - \sum_{j=0}^{k} w_j v_{k-j} \right]$