

COCO basic metal & standard toolboxes

- COCO (toolbox): "Atlas" algorithm (continuation)
"corrector" (converges & linear solver)
"toolbox with constructors"

general problem formulation:

$$\Phi(u) = 0, \quad \Phi: \mathbb{R}^{n_u} \rightarrow \mathbb{R}^{n_\Phi}$$

$$\Psi(u) - \mu = 0 \quad \Psi: \mathbb{R}^{n_u} \rightarrow \mathbb{R}^{n_\Psi}$$

Φ zero problem

u variables

Ψ monitoring functions μ continuation parameters

distinction feature: one uses gradually add to each's
to compose problems step-by-step

=> demo_buildup_circle

typical format of Φ or Ψ (discrete)
at end of file

$$\Phi(u) = u_1^2 + u_2^2 - 1$$

Start with empty problem

coco-add-problem (updates problem), other arguments:

- name your choice for name of problem part
- identifier useful for re-hashing previously registered part
- function handle plan function $[data, g] = \text{plan}(\text{prob}, \text{data}, u)$ and (optional) derivatives
- initial data
- for zero problems 'zero', 'no', 'n0', 'idle', indices
initial if function depends
guess on previously defined variables

call to coco performs continuation

arguments: problem, max-name, [], dimension of manifold,
list of free continuation parameter values, ranges

add new equations:

$$\theta = u_1^2 + f_2 - 1 \leftarrow \underline{\Phi}_1$$

$$\theta = u_2 - 1 \leftarrow \underline{\Phi}_2(u) = u - 1, \dots, 'idle', 2$$

$$\theta = u_1^2 - (u_2 - 1)^2 - 1$$

$$\theta = u_2 + u_3 \in \Phi_2(u) = u_1, u_2, \dots, 'u\theta', -1, 'u\text{idr}', 2)$$

Monitoring functions

coco-add-func [prob, monitorname, fhas, data, as los zero fun]

'active'
'inactive'
'regular', unnames, 'hide', 'ad'
'discrete'
'singular'

'inactive'

$\chi(u_j) - \mu = 0$ fixed \rightarrow dimension deficit decreases
will be solved

χ must be differentiable

'active' parameters will be free

'regular'/'discrete' / after correction $\chi(u_j) = \mu$ is assigned

'singular'
(rest part of nonlinear problem)

e.g. number of unstable eigenvalues
can be used for events

Short cut for giving names to variables:

coco-add-pars (prob, names, indices, plnames)

demo: $(u_1, u_2) \sim \{x, y\}$ adds $\chi(u) - \begin{bmatrix} x \\ y \end{bmatrix}$

By default there are no machine => release when calling coco

Investigate problem composition: prob-fcninfo (prob)

Other things:-

- { o events
- o processing bd cell arrays (coco-Bd...)
- o reading of solutions
- o hic data for more complex fractions
- o save data as part of solution filter & reload
- o reacting to signals ('update', 'save-all', 'Qd-dat')

coco-demos