Continuation with COCO – Basics
Bare metal COCO & Standard toolboxes

COCO: combines (choices of) "atlas algorithm" (continuation)
"corrector" (nonlinear solver, incl. linear solver)
"toolboxes" with "constructors" (defining equations)

[will use modified demos from "Recipes of Continuation"]

type of problems: $\Phi(u) = 0$, $\Phi : \mathbb{R}^n \to \mathbb{R}^m$, $n \geq m$

$\Psi(u) - \mu = 0$, $\Psi : \mathbb{R}^n \to \mathbb{R}^r$

Φ: zero problem, u: variables

Ψ: monitoring functions, μ: continuation parameters

distinctive feature: One may sequentially add to each of the above to compose a problem step-by-step.

demo_buildup_circle

typical/basic format of Φ or Ψ component

→ bottom of demo

Start empty prob(lem)

coco_add_func updates problem

other args 2–4:

- **name** (your choice) useful for extracting info previously stored
  "identifier"
  coco_get_func_data(prob, 'Phi', 'data')
  'uidx' ← which indices in u does its argument occupy
- **function handle** fhan [data, y] = fhan(prob, data, u)
- **initial data**

- for zero problems: 'zero', 'u0', u0, 'uidx', indices)
  "initial guess" also tells coco dimension of input for fhan
  'uidx', indices: use if function depends on previous variables
  (important optional inputs for derivatives of fhan)
  'uidx', uidx or 'u0', u0 can be dropped

Call of coco performs continuation
args: problem, name, [ ], dimension of mf, free continuation parameters, ranges

Screen output:
- tries to find consistent solution
- watch convergence
- progress report: labels, types (EP)
- check output bd ← bd is only a summary
  bd = 'bifurcation diagram'
  at the moment rudimentary
- everything is stored in data/runname

Helpful file: coco_ShortRef (for options) & tutorials

Adding to Ψ - monitoring functions and continuation parameters

prob = coco_add_func(prob, monitorname, fcn, data, 'active'/'inactive'/'regular'/..., μnames, 'uidx', indices, ...)

- fcn: Ψ
- prob, monitorname, fcn, data: as for zero functions
- 'active'/'inactive'/'regular'/...: instead of 'zero'
- μnames: μ
- indices: e.g. [2,3]
- 'uidx', indices: typically: which components of u does fcn depend on?

'active': parameters will be free → dimension deficit unchanged

'inactive': parameters will be fixed → dimension deficit decreases
equation $\Psi(u_J) - \mu = 0$ will be solved using corrector algorithm

'regular': after corrector is successful $\mu = \Psi(u_J)$ is assigned ← not part of nonlinear problem (e.g. # unstable eigenvalues)

show demo: $\Psi_1([u_1;u_2]) = u_1^2 + u_2^2$, call parameter 'p'
watch output & bd

Shortcut for "giving names to variables"

prob = coco_add_pars(prob, name, indices, pnames)

demo: call $[u_1, u_2] = \{$'x', 'y'$\}$ → this adds $\Psi([u_1;u_2]) - \begin{bmatrix}\text{'x'}\\ \text{'y'}\end{bmatrix}$

watch output & bd

Investigate problem composition prob_fcn_info(prob)

Other things:
- events
- processing of bd cell arrays (coco_bd_....)
- reading of solutions (coco_read_solution)
- use data for more complex functions
- save data as part of solution files & reload
- reacting to signals (e.g. 'update', 'save_full', 'bd_data', 'print')
- update data during computation and store (e.g. mesh for BVPs)

all in cmds_demos

Cusp very basic example for 2d continuation
important: introduce sufficiently many u variables as continuation parameters
atlas_kd will embed manifold only in continuation parameter space

---

Standard toolboxes - demo tor has many standard bifurcations
uses toolboxes ep (= equilibrium)
po (= periodic orbit, depending on coll)
- investigate problem composition,
- spot naming conventions
- note that toolboxes use and save data (incl. r.h.s.) in solution points

Recipes notes

zero fcn $\Phi : \mathbb{R}^n \to \mathbb{R}^m$ (diff) $\quad \Phi(u) = 0$

monitoring fcn $\Psi : \mathbb{R}^n \to \mathbb{R}^r$ $\quad \Psi(u) - \mu = 0$ $\quad = \Big\{ F(u,\mu)$

$I \cup J = \{1,\dots,r\}$

Ψ monitoring fcns, μ = cont. parameters

initial guess: u_0, restricted $G : (u, \mu_J) := F(u,\mu)\big|_{\mu_I = \mu_I^0}$

(ont. prob $F_I^{u}(u,\mu) := \begin{cases} F(u,\mu) = 0 \\ \mu_I - \mu_I^* = 0 \end{cases}$

F_I: F with monitors in I fixed

dimension of manifold: $d = n + r - (m + r) - |I| = n - m - |I|$

$u_0, \mu_I^*, \mu_J^0 \leadsto \mathcal{I}(u_0, \mu_J^0) \mapsto \begin{bmatrix} \Phi(u_0) \\ \Psi(u_0) - \begin{bmatrix} \mu_I^* \\ \mu_J^0 \end{bmatrix} \end{bmatrix}$ $\quad \partial \mathcal{I} = \begin{bmatrix} \partial \Phi, & 0 \\ \partial \Psi, & -I_J \end{bmatrix}$

let (u^*, μ^*) be s.t. $\Phi(u^*) = 0$, $\Psi(u^*) = \mu^*$ $\quad$ Jacobian $\begin{pmatrix} \partial_u \Phi(u_0) & 0 \\ \partial_u \Psi(u_0) & -I \end{pmatrix}$ of $F(u,\mu) = 0$

rank deficit $= \min\{n,m\} - \mathrm{rk}\,\Phi$

if u_0 regular $\to$ solution mf of F has dim $d = n - m$

extended CP = $F(u,\mu)$ with some free μ (active) $\quad$ extend a C.P.

restricted CP = $F(u,\mu)$ with some fixed μ (inactive) $\quad$ restrict a C.P.

$\hat{E}$ embedded in E $\quad \hat{E} \subseteq E$ if $\quad \Phi_{L_\Phi}(u) = \hat{\Phi}(u_K)$

$\hat{\Psi}, \hat{\Phi}$ are independent of $u_{\bar{K}}$ $\quad \Psi_{L_\Psi}(u) = \hat{\Psi}(u_K)$

$u_{0K} = \hat{u}_0$

$0 = f_1(v_1, v_3)$ $\quad \hat{E}_1 = [f_1, \emptyset, (v_1, v_3)]$

$0 = f_2(v_2, v_3)$ $\quad \hat{E}_2 = [f_2, \emptyset, (v_2, v_3)]$

$\hat{E} \subseteq E$ canonical embedding if subproblem covers first components of index sets ($\hat{u} = u|_{[1..n]}$, $\hat{\Phi}, \hat{\Psi} = \Phi, \Psi|_{[1..r]}$)

nontrivial if at least a real increase