Carl von Ossietzky
Universität
Oldenburg

Examples for the Use of the Intlab and Cora Libraries in the Frame of Control and State Estimation Tasks

Prof. Dr.-Ing. habil. Andreas Rauh

Carl von Ossietzky Universität Oldenburg
Department of Computing Science –
Distributed Control in Interconnected Systems

Brest

September 21, 2023

Representation of Uncertainty in Intlab

- Resource: https://www.tuhh.de/ti3/rump/intlab/demos/

- Definition of interval variables & use of affine arithmetic

```
clc; close all;
format compact short infsup
phi = 30*pi/180;
Q = [ cos(phi) -sin(phi) ; sin(phi) cos(phi) ]
X = [ infsup(1,2) ; infsup(2,4) ]

Yint = Q*X;

Xa = affari(X)
Yaff = Q*affari(Xa);
plotintval(Yint,'r')

hold on;
plotaffari(Yaff)
shg
```

Representation of Uncertainty in Intlab

- Resource: https://www.tuhh.de/ti3/rump/intlab/demos/
- Fundamental difference between interval variables & use of affine arithmetic
  - Affine arithmetic uses internal memory to record dependencies

```
X = [ infsup(1,2) ; infsup(2,4) ];
% no inverse element to addition in classical interval arithmetic
X-X

% linear dependencies help to reduce dependency effect in affine
% arithmetic
Xa = affari(X);
Xa-Xa

% however
affari(X) - affari(X)
```

Representation of Uncertainty in Intlab

– Resource: https://www.tuhh.de/ti3/rump/intlab/demos/

– Definition of interval variables & use of affine arithmetic

```
clc; close all;
format compact short infsup
phi = affari(infsup(30,40))*pi/180;
Q = [ cos(phi) -sin(phi) ; sin(phi) cos(phi) ]
X = [ infsup(1,2) ; infsup(2,4) ]

Yint = intval(Q)*X;

Xa = affari(X)
Yaff = Q*affari(Xa);
plotintval(Yint,'r')

hold on;
plotaffari(Yaff)
shg
```

Matrix Multiplication in Intlab

– Interval matrix operations: https://www.tuhh.de/ti3/intlab/demos/html/dintval.html#30

```
clc; close all;
n = 1000;

c = randn(n);
C = intval(c);
C_ = midrad(c,.1);

intvalinit('SharpIVmult')

tic, scc = c*c; toc
tic, sCC = C*C; toc
tic, sCC = C*C_; toc
tic, sCC__ = C_*C_; toc
```

Matrix Multiplication in Intlab

– Interval matrix operations: https://www.tuhh.de/ti3/intlab/demos/html/dintval.html#30

```
clc; close all;
n = 1000;

c = randn(n);
C = intval(c);
C_ = midrad(c,.1);

intvalinit('FastIVmult')

tic, fcc = c*c; toc
tic, fCC = C*C; toc
tic, fCC = C*C_; toc
tic, fCC__ = C_*C_; toc

max(max(diam(fCC__)./diam(sCC__)))
```

Standard Functions and Solution of Algebraic Equations in Intlab

– Interval standard functions: https://www.tuhh.de/ti3/intlab/demos/html/dintval.html#40

```
clc; close all;
x = infsup(-1,1);


cos(x)
sin(x)
exp(x)
```

– Rigorous solution of linear systems:
https://www.tuhh.de/ti3/intlab/demos/html/dintval.html#51

– Enclosure of eigenvalues and eigenvectors:
https://www.tuhh.de/ti3/intlab/demos/html/dintval.html#73

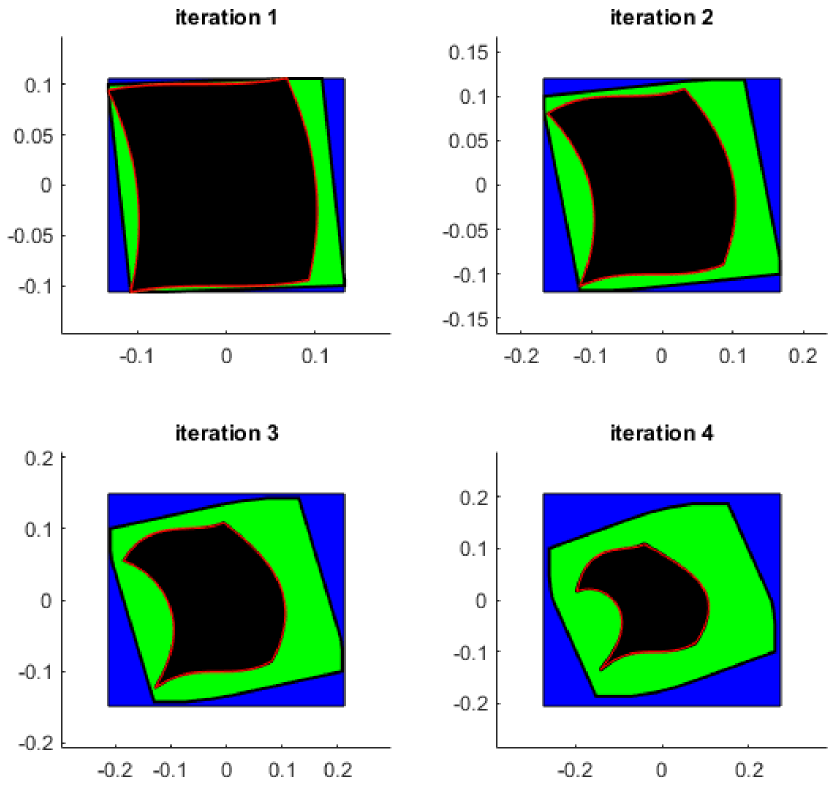
– verifylss (verified linear system solver):
https://www.tuhh.de/ti3/intlab/demos/html/dintval.html#44

Root Finding and Global Optimization in Intlab

- Verified root finding: https://www.tuhh.de/ti3/rump/intlab/demos/html/dglobal.html#1
  - Univarite case
  - Multi-variate case
- Global minimization: https://www.tuhh.de/ti3/rump/intlab/demos/html/dglobal.html#14
- Constrained global minimization: https://www.tuhh.de/ti3/rump/intlab/demos/html/dglobal.html#41
- Parameter identification: https://www.tuhh.de/ti3/rump/intlab/demos/html/dglobal.html#56
  - Default setting (pure interval arithmetic)
  - Interval arithmetic using mean-value rule
  - Affine arithmetic

Dynamic System Simulation in Intlab

- AWA: Lohner’s method: https://www.tuhh.de/ti3/intlab/demos/html/dawa.html
- Taylor model toolbox: https://www.tuhh.de/ti3/intlab/demos/html/dtaylormodel.html
  - Re-implementation of Cosy-like solver



CORA: Available Uncertainty Representations

– Intervals

– Ellipsoids

– Zonotopes

– Matrix zonotopes

```
% set and matrix
S = zonotope([0 1 1 0; ...
              0 1 0 1]);
M = [1 0; -1 0.5];

% linear transformation
res = M * S;
figure; plot(res);
```

Matthias Althoff (2023). CORA (https://github.com/TUMcps/CORA)
https://fr.mathworks.com/matlabcentral/fileexchange/68551-cora

Carl von Ossietzky Universität Oldenburg

CORA: Available Uncertainty Representations

- Intervals
- Ellipsoids
- Zonotopes
- Matrix zonotopes

```
% set S1 and S2

S1 = zonotope([0 0.5 1; ...
               0 1 0]);

S2 = zonotope([0 1 0; ...
               0 0 1]);

% Minkowski sum
res = S1 + S2;
figure; hold on; plot(S1); plot(S2); plot(res); hold off
```

Carl von Ossietzky Universität Oldenburg

CORA: Available Uncertainty Representations

- Intervals
- Ellipsoids
- Zonotopes
- Matrix zonotopes

```
% set S1 and S2

S1 = zonotope([0 0.5 1; ...
               0 1 0]);
S2 = zonotope([0 1 0; ...
               0 0 1]);

% Minkowski sum
res = ellipsoid(S1) + ellipsoid(S2);
figure; hold on; plot(ellipsoid(S1)); plot(ellipsoid(S2));
plot(res); hold off
```

CORA: Available Uncertainty Representations

– Intervals

– Ellipsoids

– Zonotopes

– Matrix zonotopes

```
% set S1 and S2

S1 = conZonotope([1.5 1 0; ...
                  1.5 0 1]);

S2 = conZonotope([-1.5 1 0; ...
                 -1.5 0 1]);

% convex hull
res = convHull(S1,S2);
figure; hold on; plot(S1); plot(S2); plot(res); hold off
```

CORA: Concluding Example –

Linear Dynamic Systems and ReachSets

– Matlab demo

  – Matrix exponential

```
close all
G{1} = [0 0; 0 0];
G1{1} = [0.5 0.02; 0.02 0.2];
C = [-1 0.1; -0.1 -1];
mz = matZonotope(C,G);
mz1 = matZonotope(C,G1);

plot(expm(mz*matZonotope(0.5,{0.5}),5)*zonotope([1;1],[1 0;0 1]))
```

  – Recursive evaluation

  – Order reduction (of zonotopes and matrix zonotopes)

CORA: Concluding Example –

Linear Dynamic Systems and ReachSets

– Reachability analysis

```
sys = linParamSys(mz,matZonotope([0;1],{[0;0]}),'varParam‘)

params.tFinal = 5;
params.R0 = z_ini0;
params.U = zonotope(interval(0));
options.timeStep = 0.05;
options.zonotopeOrder = 10;
options.taylorTerms = 5;
options.intermediateTerms = 4;

R = reach(sys,params,options);

figure; han = plot(R)
figure; plotOverTime(R,[1]);
```

CORA: Concluding Example –

Linear Dynamic Systems and ReachSets

– Matlab demo

– Exploiting properties of cooperativity and (mixed) monotonicity is always advantageous