n-th order, radial vector field with known ROA

Motivated by an example from Davison, Kurak, 1971, it is easy to create cubic vector fields with known ROA. With an arbitrarily chosen quadratic shape factor, it is also easy to compute the optimal Beta. The iteration can be tested by comparing the Beta obtained from iteration to the optimal Beta computed analytically.

Contents

Create matrices describing vector field and shape factor

2 positive-definite matrices for vector field and shape function

```
nX = 6;
[U,S,V] = svd(randn(nX,nX));  % Get "random" unitary matrices
lamB = diag(exp(2*randn(1,nX)));
lamR = diag(exp(2*randn(1,nX)));
B = U*lamB*U';  R = V*lamR*V';
```

Analytically compute BetaOpt; rescale data so BetaOpt = 1

```
betaOpt = 1/max(real(eig(B/R)));
B = B*betaOpt;
betaOpt = 1/max(real(eig(B/R)))
```

```
betaOpt =
    1.0000
```

Create Shape Function, and Vector field

```
x = mpvar('x',nX,1);
p = x'*R*x;
f = -x + x*(x'*B*x);
```

Create monomial basis for V (Vdeg = 2) and s1 and s2

```
zV = monomials(x,2:Vdeg);          % 'vec' form
```

```
z1maxd = ceil((Vdeg-p.maxdeg)/2);
z1 = monomials(x, 0:z1maxd );      % 'mat' form
z2 = monomials(x, 1:1);           % 'mat' form
L2 = 1e-6*(x'*x);
```

Run 3 steps of V-s iteration

```
Nsteps = 3;
opts = []; opts.gmin = 0; opts.gmax = 50; opts.L2 = L2;
for i1=1:Nsteps;
    if i1==1
        V = linstab(f,x);    % Initialize from Linearization
    else
        [V,c] = roavstep(f,p,x,zV,beta,gamma,s1,s2,opts);
        V = V/gamma;
    end
    [gbnds,s2] = pcontain(jacobian(V,x)*f+L2,V,z2,opts);
    gamma = gbnds(1);
    [bbnds,s1]  =pcontain(V-gamma,p,z1,opts);
    beta = bbnds(1);
    fprintf('i1 = %d  \t beta = %6.4f, betaOpt = %6.4f\n',i1,beta,betaOpt);
end
```

```
i1 = 1         beta = 0.3517, betaOpt = 1.0000
i1 = 2         beta = 0.9949, betaOpt = 1.0000
i1 = 3         beta = 0.9979, betaOpt = 1.0000
```

Published with MATLAB® 7.8