Fast Approximation of the Stability Radius and the H_∞ Norm for Large-Scale Linear Dynamical Systems with Output Feedback

Michael L. Overton, Courant Institute, N.Y.U.

Joint work with
Nicola Guglielmi (L'Aquila)
Mert Gürbüzbalaban (M.I.T.)
Tim Mitchell (Max Planck Institute, Magdeburg)

SIAM Conference on Applied Linear Algebra
Atlanta, October 2015

Thanks to my coauthors, the organizers, to ILAS and to NSF

SPECTRAL VALUE SETS AND THE TRANSFER MATRIX

Let σ denote spectrum, let $\varepsilon \in \mathbb{R}^+$ such that $\varepsilon\|D\| < 1$, let $\|\cdot\| = \|\cdot\|_2$, and define the spectral value set

$$\sigma_\varepsilon(A,B,C,D) := \bigcup \{\sigma(M(\Delta)) : \Delta \in \mathbb{C}^{p,m}, \|\Delta\| \leq \varepsilon\}$$

Key Lemma. (see e.g. Hinrichsen & Pritchard, *Mathematical Systems Theory I*, Springer, 2005) Let $\varepsilon \in \mathbb{R}^+$ such that $\varepsilon\|D\| < 1$. Then

$$\sigma_\varepsilon(A,B,C,D) \equiv \bigcup \{\sigma(M(\Delta)) : \Delta \in \mathbb{C}^{p,m}, \|\Delta\| \leq \varepsilon, \text{rank}(\Delta) \leq 1\}$$
$$\equiv \bigcup \{\lambda \in \mathbb{C} : \|G(\lambda)\| \geq \varepsilon^{-1}\},$$

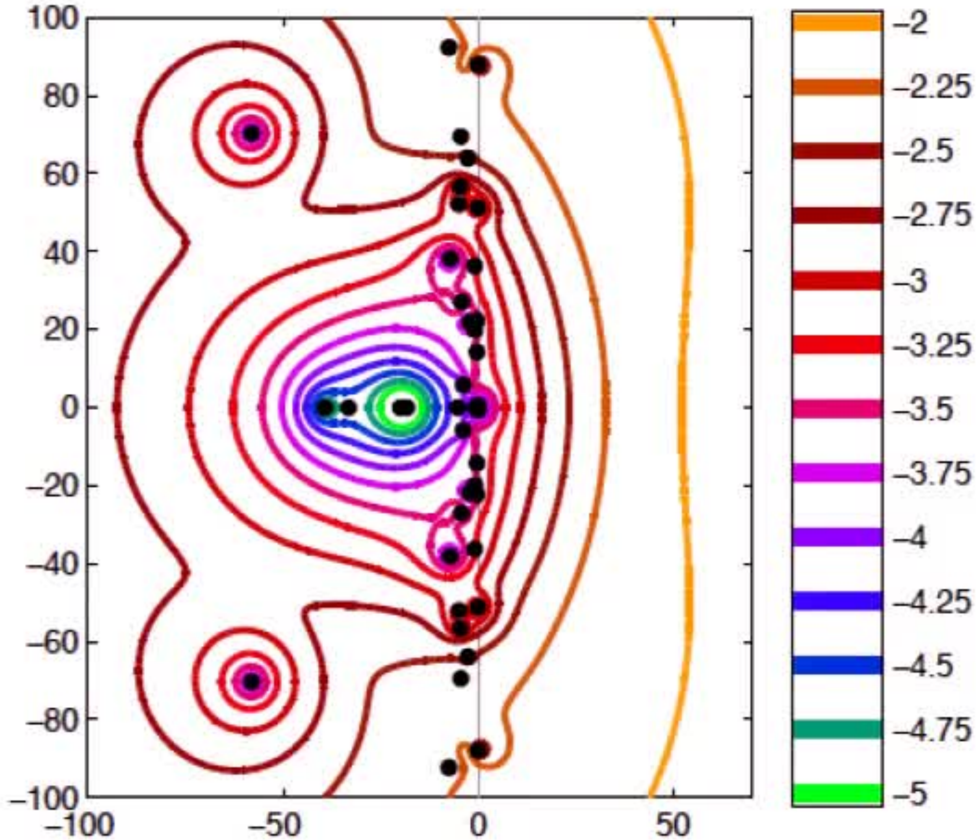
where $G(\lambda)$ is the transfer matrix function defined by

$$G(\lambda) := C(\lambda I - A)^{-1}B + D.$$

Furthermore, given $\lambda \in \mathbb{C}$ with $\|G(\lambda)\| = \varepsilon^{-1}$, we can obtain Δ with rank one such that $\|\Delta\| = \varepsilon$ and $\lambda \in \sigma(M(\Delta))$ by setting $\Delta = \varepsilon uv^*$, where u and v are respectively right and left singular vectors of $G(\lambda)$ corresponding to its largest singular value ε^{-1}.

Case $B = C = I$ and $D = 0$: $\sigma_\varepsilon(A,B,C,D)$ is the ε-pseudospectrum of A which contains λ iff $\|(\lambda I - A)^{-1}\| \geq \varepsilon^{-1}$ (Trefethen, Embree 2005)

PSEUDOSPECTRA FOR A BOEING 767 IN FLUTTER



Eigenvalues (black dots) are in left-half plane, so A is stable, but not stable under small perturbations! Curves are boundaries of $\sigma_\varepsilon(A)$, equivalently contours of the smallest singular value of $A - \lambda I$, with $\log_{10}$ scale for ϵ.
(EigTool, Trefethen & T. Wright, 2002)

THE STABILITY RADIUS AND THE H_∞ NORM

The stability radius for the system described by (A, B, C, D) is

$$\varepsilon_\star := \sup\{\varepsilon : \sigma_\varepsilon(A, B, C, D) \subset \mathbb{C}_-\}$$

where $\mathbb{C}_-$ is the open left half of the complex plane.

(Radius in the sense of perturbations, not the complex plane.)

Case $B = C = I$, and $D = 0$: stability radius for the dynamical system reduces to distance to instability of A (Van Loan 1985).

The H_∞ norm of the transfer matrix function G is

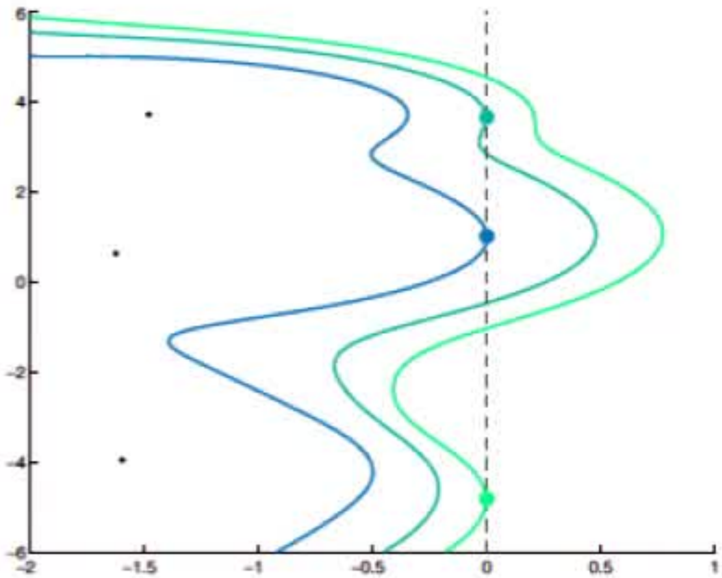
$$\|G\|_\infty := \sup_{\omega\in\mathbb{R}} \|G(\mathbf{i}\omega)\| \text{ if } \sigma(A) \in \mathbb{C}_- \text{ (or } +\infty \text{ otherwise).}$$

Case $B = C = I$, and $D = 0$: $\sup_{\omega\in\mathbb{R}} \|(\mathbf{i}\omega I - A)^{-1}\|$

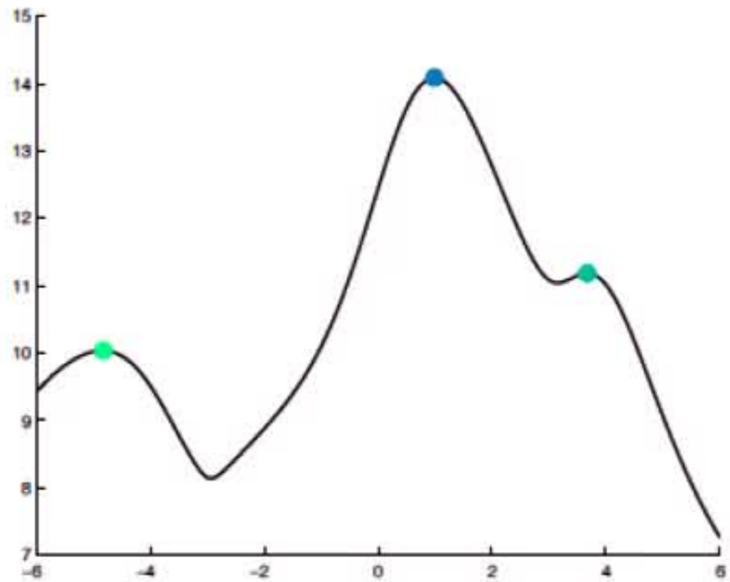
Because of the Key Lemma, the stability radius and the H_∞ norm are reciprocals of each other:

$$\varepsilon_\star = \|G\|_\infty^{-1}.$$

In Pictures: $\sigma_\varepsilon(A,B,C,D)$ and $\|G(\mathbf{i}\omega)\|$



$\sigma_\varepsilon(A,B,C,D)$
Complex plane plot
Black dots are eigenvalues of A
Blue contour is boundary of spectral value set for $\varepsilon_\star$
Green contours show spectral value sets for $\varepsilon > \varepsilon_\star$



$\|G(\mathbf{i}\omega)\|$
Horizontal axis is ω
Vertical axis is $\|G(\mathbf{i}\omega)\|$
Blue dot is global max $\varepsilon_\star^{-1}$
Green dots are local maxima with $\varepsilon^{-1} < \varepsilon_\star^{-1}$

Exploiting Rank One Perturbations

To compute $\alpha_\varepsilon(A, B, C, D)$, we need to find the rightmost point of $\sigma_\varepsilon(A, B, C, D)$. Since we can restrict $\Delta = \varepsilon u v^*$, we need only consider

$$\begin{aligned} M(\Delta) &= A + B\Delta(I - D\Delta)^{-1}C \\ &= A + B\frac{\varepsilon}{1 - \varepsilon v^* D u} u v^* C \\ &=: A + B\widetilde{\Delta}C \end{aligned}$$

where $\|u\| = \|v\| = 1$.

Our algorithm will generate a sequence u_j, v_j for which we hope the rightmost eigenvalue of $M(u_j v_j^*)$ will converge to a rightmost point of $\sigma_\varepsilon(A, B, C, D)$.

COMPUTING RIGHTMOST EIGENVALUES

The cost at each iteration: two calls to `eigs` / ARPACK (Lehoucq, Sorensen and Yang) to compute the rightmost eigenvalue λ of a structured/sparse matrix, as well as the associated right eigenvector x and left eigenvector y, using matrix-vector products.

Generically, the eigenvalue is simple, so we use the normalization $\|x\| = \|y\| = 1$ and $y^*x \in \mathbb{R}^{++}$ (real and positive), and we call (λ, x, y) an RP-compatible eigentriple.

Although `eigs` is not absolutely guaranteed to find a rightmost eigenvalue, it is quite reliable as long as enough eigenvalues are requested.

Pushing Rightward via a Rank-1 Perturbation

For $t \in [0,1]$, consider the matrix-valued linear function:

$$K(t) = A + B\widetilde{\Delta}_{j-1}C + tB\left(\widetilde{\Delta}_j - \widetilde{\Delta}_{j-1}\right)C$$

so that $K(0) = A + B\widetilde{\Delta}_{j-1}C, \quad K(1) = A + B\widetilde{\Delta}_j C$, with

$$\widetilde{\Delta}_j = \frac{\varepsilon}{1 - \varepsilon v_j{}^* D u_j} u_j v_j^*$$

where unit-norm vectors u_j, v_j are free to be chosen and Δ_{j-1} is the previous iterate, with $M(\Delta_{j-1})$ having rightmost eigenvalue λ_{j-1} with RP-compatible eigentriple $(\lambda_{j-1}, x_{j-1}, y_{j-1})$.

COMPUTING RIGHTMOST EIGENVALUES

The cost at each iteration: two calls to `eigs` / ARPACK (Lehoucq, Sorensen and Yang) to compute the rightmost eigenvalue λ of a structured/sparse matrix, as well as the associated right eigenvector x and left eigenvector y, using matrix-vector products.

Generically, the eigenvalue is simple, so we use the normalization $\|x\| = \|y\| = 1$ and $y^*x \in \mathbb{R}^{++}$ (real and positive), and we call (λ, x, y) an RP-compatible eigentriple.

Although `eigs` is not absolutely guaranteed to find a rightmost eigenvalue, it is quite reliable as long as enough eigenvalues are requested.

Pushing Rightward via a Rank-1 Perturbation

For $t \in [0,1]$, consider the matrix-valued linear function:

$$K(t) = A + B\widetilde{\Delta}_{j-1}C + tB\left(\widetilde{\Delta}_j - \widetilde{\Delta}_{j-1}\right)C$$

so that $K(0) = A + B\widetilde{\Delta}_{j-1}C, \quad K(1) = A + B\widetilde{\Delta}_j C$, with

$$\widetilde{\Delta}_j = \frac{\varepsilon}{1 - \varepsilon v_j{}^* D u_j} u_j v_j^*$$

where unit-norm vectors u_j, v_j are free to be chosen and Δ_{j-1} is the previous iterate, with $M(\Delta_{j-1})$ having rightmost eigenvalue λ_{j-1} with RP-compatible eigentriple $(\lambda_{j-1}, x_{j-1}, y_{j-1})$.

Let $\kappa(t)$ be the eigenvalue of $K(t)$ that converges to λ_{j-1} as $t \to 0$, so

$$\kappa'(0) = \varepsilon \frac{y_{j-1}^* B\left(\frac{u_j v_j^*}{1-\varepsilon v_j{}^* D u_j}\right) C x_{j-1}}{y_{j-1}^* x_{j-1}} - \varepsilon \frac{y_{j-1}^* B\left(\frac{u_{j-1} v_{j-1}^*}{1-\varepsilon v_{j-1}^* D u_{j-1}}\right) C x_{j-1}}{y_{j-1}^* x_{j-1}}.$$

Note if $u_j := u_{j-1}$ and $v_j := v_{j-1}$ then $\kappa'(0) = 0$.

Solving the Optimization Problem to Find u_j, v_j

To make $\mathrm{Re}\,(\kappa'(0))$ as large as possible, we need to solve:

$$\max_{\|u_j\|=\|v_j\|=1} \mathrm{Re}\,\frac{y_{j-1}^* B u_j v_j^* C x_{j-1}}{1-\varepsilon v_j^* D u_j} \qquad (1)$$

Case $B = C = I$ and $D = 0$: set $u_j := y_{j-1}$ and $v_j := x_{j-1}$.
If not, but $D = 0$, set

$$u_j := B^* y_{j-1}/\|B^* y_{j-1}\| \quad \text{and} \quad v_j := C^* x_{j-1}/\|C^* x_{j-1}\|$$

as long as the denominators are nonzero
(this is guaranteed if λ_{j-1} is "controllable and observable").
If $D \neq 0$, more complicated, but can still be solved explicitly.

ALGORITHM SVSA (SPECTRAL VALUE SET ABSCISSA)

Purpose: approximate $\sigma_\varepsilon(A, B, C, D)$ for fixed ε.

Input: ε with $0 < \varepsilon\|D\| < 1$ and u_0, v_0 with unit norm.

Compute rightmost RP-compatible eigentriple (λ_0, x_0, y_0) of

$$M(\varepsilon u_0 v_0^*) = A + B\frac{\varepsilon}{1 - \varepsilon v_0^* D u_0} u_0 v_0^* C$$

For $j = 1, 2, \ldots$

Set u_j, v_j to the u and v that explicitly maximize (1)
Compute rightmost RP-compatible eigentriple (λ_j, x_j, y_j) of

$$M(\varepsilon u_j v_j^*) = A + B\frac{\varepsilon}{1 - \varepsilon v_j^* D u_j} u_j v_j^* C.$$

If $\mathrm{Re}\,(\lambda_j) < \mathrm{Re}\,(\lambda_{j-1})$, use line search to enforce monotonicity.

ALGORITHM SVSA (SPECTRAL VALUE SET ABSCISSA)

Purpose: approximate $\sigma_\varepsilon(A, B, C, D)$ for fixed ε.

Input: ε with $0 < \varepsilon\|D\| < 1$ and u_0, v_0 with unit norm.

Compute rightmost RP-compatible eigentriple (λ_0, x_0, y_0) of

$$M(\varepsilon u_0 v_0^*) = A + B\frac{\varepsilon}{1 - \varepsilon v_0^* D u_0} u_0 v_0^* C$$

For $j = 1, 2, \ldots$

Set u_j, v_j to the u and v that explicitly maximize (1)

Compute rightmost RP-compatible eigentriple (λ_j, x_j, y_j) of

$$M(\varepsilon u_j v_j^*) = A + B\frac{\varepsilon}{1 - \varepsilon v_j^* D u_j} u_j v_j^* C.$$

If $\mathrm{Re}\,(\lambda_j) < \mathrm{Re}\,(\lambda_{j-1})$, use line search to enforce monotonicity.

By construction, $\lambda_j \in \sigma_\varepsilon(A, B, C, D)$ for all ε, but will this converge to a (locally) rightmost point of $\sigma_\varepsilon(A, B, C, D)$?

Convergence of SVSA in Theory and Practice

In theory: for ε sufficiently small, can prove that Algorithm SVSA has local convergence to locally rightmost points at a linear rate under reasonable assumptions.

In practice: for all ε, almost always get convergence to locally rightmost points, and often to globally rightmost points, at a linear rate which can be substantially accelerated using extrapolation techniques.

Globally rightmost points give the value $h(\varepsilon) = \alpha_\varepsilon(A, B, C, D)$ while locally rightmost points give lower bounds on $h(\varepsilon) = \alpha_\varepsilon(A, B, C, D)$.

A Newton-Bisection Method to Find $\varepsilon_\star$

- Use Algorithm SVSA to approximate $h(\varepsilon)$ inside a Newton-bisection outer iteration to find the zero of h, namely, the stability radius $\varepsilon_\star$.
- If $h(\varepsilon)$ is computed correctly, its derivative $h'(\varepsilon)$ can be cheaply computed from information returned by SVSA.
- Often, this converges to the stability radius $\varepsilon_\star$, which is also the reciprocal of the H_∞ norm (the global maximum of $\|G\|$ on the imaginary axis), with quadratic convergence.
- However, because SVSA is guaranteed to find only a lower bound on $h(\varepsilon)$, the Newton-bisection method is guaranteed to find only an upper bound on $\varepsilon_\star$.
- Usually, its reciprocal is at least a local maximizer of $\|G\|$ on the imaginary axis.
- But in some cases algorithm breaks down, even if SVSA always finds local maximizers of $\alpha_\varepsilon(A, B, C, D)$.

Hybrid Expansion-Contraction

Key Observation:

- Rightmost points found by SVSA in the right half-plane give upper bounds on $\varepsilon_\star$
- Rightmost points found by SVSA in the left half-plane do *not* give lower bounds on $\varepsilon_\star$

New Idea:

- Start by increasing ε until SVSA finds a rightmost point in the right half-plane, giving an upper bound on $\varepsilon_\star$.
- Then monotonically reduce ε, continuing to call SVSA to compute upper bounds until the rightmost point in the right half-plane is sufficiently close to the imaginary axis.

CONTRACTING UPPER BOUNDS ON $\varepsilon_\star$

Given vectors u, v and scalar ε_0 such that $M(\varepsilon_0 uv^*)$ has an eigenvalue in the right-half plane, define

$$M_{uv}(\varepsilon) := A + B\widetilde{\Delta}_{uv}(\varepsilon)C \quad \text{where} \quad \widetilde{\Delta}_{uv}(\varepsilon) := \frac{\varepsilon}{1 - \varepsilon v^* D u} uv^*$$

where u,v are now fixed. Also define

$$h_{uv}(\varepsilon) := \max\{\mathrm{Re}\,(\lambda) : \lambda \in \sigma(M_{uv}(\varepsilon)\}$$

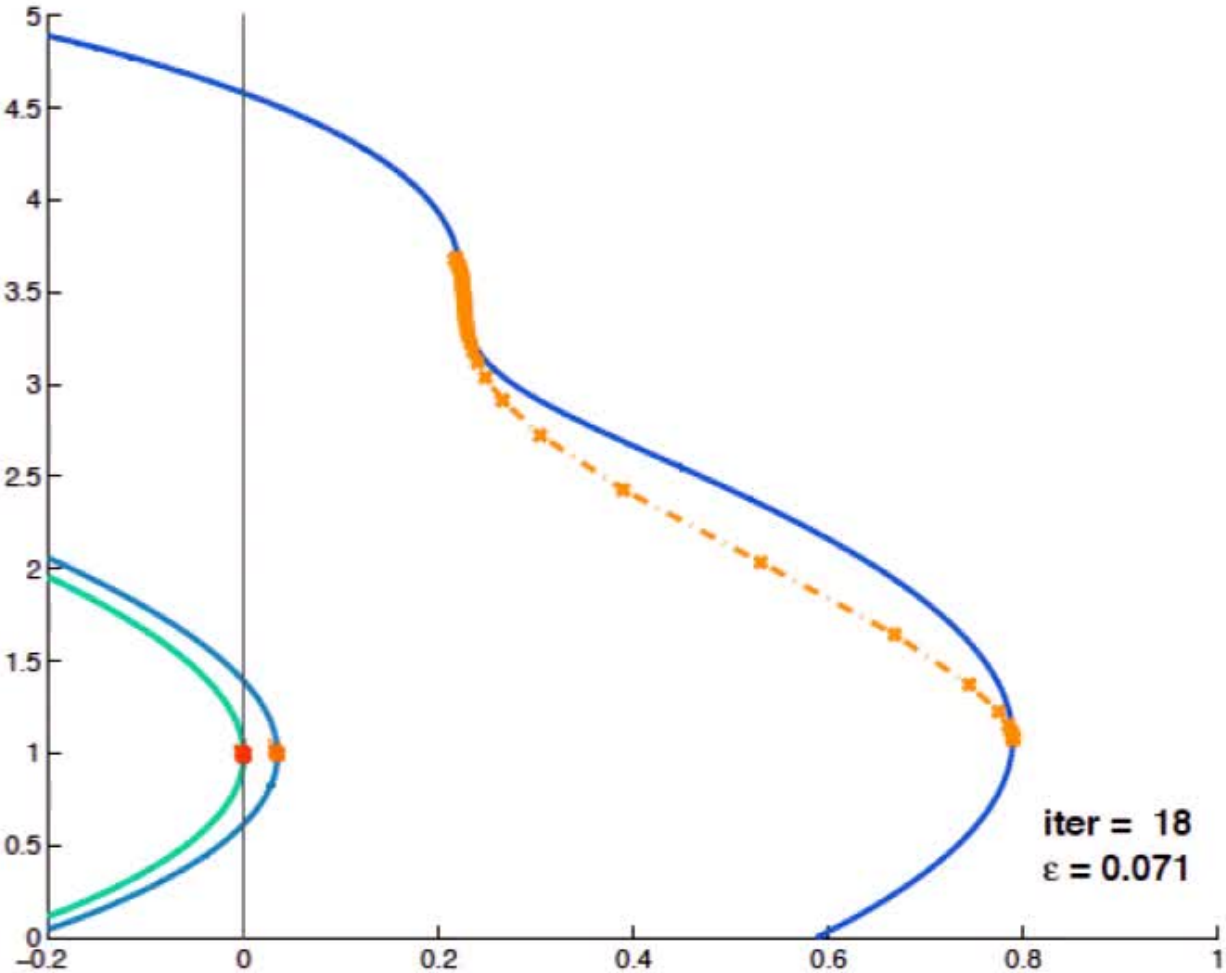
the spectral abscissa of $M_{uv}(\varepsilon)$. We know that $h_{uv}(\varepsilon_0) > 0$ and $h_{uv}(0) < 0$ assuming $\sigma(A) \in \mathbb{C}_-$, and since h_{uv} is continuous and easy to evaluate using `eigs`, we can easily find a zero of it using Newton-bisection.

HYBRID EXPANSION-CONTRACTION

Algorithm HEC (Hybrid Expansion-Contraction)
Purpose: approximate the stability radius $\varepsilon_\star$ for A, B, C, D
Input: $\varepsilon_0 > 0$ with $\varepsilon_0\|D\| < 1$ and unit norm vectors u_0, v_0 such that λ_0, a rightmost eigenvalue of $M(\varepsilon_0 u_0 v_0^*)$, is in the right half-plane.
For $k = 0, 1, 2, \ldots$

1. **Contraction:** call a Newton-bisection zero finding algorithm to compute $\hat{\varepsilon}_k \in (0, \varepsilon_k]$ so that $h_{u_k v_k}(\hat{\varepsilon}_k) = 0$, along with $\hat{\lambda}_k$, a rightmost eigenvalue of $M_{u_k v_k}(\hat{\varepsilon}_k)$ on the imaginary axis.
2. **Expansion:** call Algorithm SVSA with input $\hat{\varepsilon}_k, u_k, v_k, \hat{\lambda}_k$, returning as output $u_{k+1}, v_{k+1}, \lambda_{k+1}$.
3. Set $\varepsilon_{k+1} = \hat{\varepsilon}_k$.

Hybrid Expansion-Contraction: Demo



SIMPLICITY CONDITIONS

Suppose that λ is given with $\|G(\lambda)\| = \varepsilon^{-1}$. We say that the simplicity conditions hold at λ with respect to ε if

1. the largest singular value ε^{-1} of $G(\lambda)$ is simple.
2. letting u and v be corresponding right and left singular vectors and setting $\Delta = \varepsilon uv^*$, the eigenvalue λ of $M(\Delta)$ is simple.

Then it is straightforward to write down a first-order necessary condition for λ to be a rightmost point of $\sigma_\varepsilon(A, B, C, D)$.

ALGORITHM HEC CONVERGES

Assume that for all ε, Algorithm SVSA delivers points λ satisfying the simplicity conditions and the first-order necessary condition for λ to be a locally rightmost point of $\sigma_\varepsilon(A, B, C, D)$.

Theorem

Algorithm HEC generates $\{\varepsilon_k\}$ *converging monotonically to a limit* $\tilde{\varepsilon} \geq \varepsilon_\star$ *and with* $\{\lambda_k\}$ *having at least one cluster point* $\tilde{\lambda}$*, with* $\|G(\tilde{\lambda})\| = \tilde{\varepsilon}^{-1}$*, with* $\mathrm{Re}\,(\tilde{\lambda}) = 0$*. Assuming that the simplicity conditions hold at* $\tilde{\lambda}$ *with respect to* $\tilde{\varepsilon}$*, then*

1. $\tilde{\lambda}$ *satisfies the first-order necessary condition to be a locally rightmost point of* $\alpha_\varepsilon(A, B, C, D)$ *for* $\varepsilon = \tilde{\varepsilon}$
2. $\mathrm{Im}\,(\tilde{\lambda})$ *is a stationary point of* $\|G(\mathbf{i}\omega)\|$ *with value* $\tilde{\varepsilon}^{-1}$*.*

Furthermore, if $\tilde{\lambda}$ *is indeed a locally rightmost point of* $\sigma_{\tilde{\varepsilon}}(A, B, C, D)$*, then* $\mathrm{Im}\,(\tilde{\lambda})$ *is a local maximizer of* $\|G(\mathbf{i}\omega)\|$ *with locally maximal value* $\tilde{\varepsilon}^{-1}$*.*

ALGORITHM HEC CONVERGES QUADRATICALLY

In fact, provided SVSA produced a rightmost point of $\sigma_{\varepsilon_k}(A, B, C, D)$, it turns out that

$$h'_{u_k v_k}(\varepsilon_k) = h'(\varepsilon_k)$$

So the first Newton step in the contraction phase to find a zero of $h_{u_k v_k}$ is equivalent to the Newton step to find a zero of h (which we cannot evaluate exactly).

The additional work done in the contraction phase can only improve the step, and so we can prove that Algorithm HEC converges quadratically under a regularity condition.

HEC IN PRACTICE: 33 SMALL-SCALE PROBLEMS

taken from EigTool

HEC Overall Performance: Small-scale

	Totals		# Rel Diff to $\|G\|_\infty$			
Alg + Opts	λ_{RP}	sec	10^{-8}	10^{-6}	10^{-4}	S
NB	32112	465.49	18	22	25	30
HEC	16665	199.99	21	25	29	33
HEC + E	9708	168.71	19	23	28	33
HEC + RS	10565	99.70	21	25	28	33
HEC + E,RS	6767	89.64	21	25	28	33

NB	Newton-Bisection - `hinfnorm` v1.02
HEC	Hybrid-Expansion-Contraction, with options ...
E	Vector Extrapolation (5 vectors)
RS	Relative Step Size Termination (0.01)
λ_{RP}	Number of Computed Eigentriples
$\|G\|_\infty$	from `getPeakGain` (BBBS algorithm, MATLAB)
S	"success"

HEC IN PRACTICE: 14 LARGE-SCALE PROBLEMS

taken from EigTool

HEC Overall Performance: Large-scale

	Totals		# Rel Diff to Best			
Alg + Opts	λ_{RP}	sec	10^{-8}	10^{-6}	10^{-4}	S
NB	4196	20920	11	11	11	13
HEC	2338	3756	9	10	12	14
HEC + V	2336	2362	10	11	13	14
HEC + E	636	1504	10	12	13	14
HEC + E,V	690	1110	10	11	13	14
HEC + RS	861	1046	9	10	11	14
HEC + RS,V	849	919	10	11	12	14
HEC + E,RS	700	960	9	10	11	14
HEC + E,RS,V	794	841	11	12	13	14

NB Newton-Bisection - `hinfnorm` v1.02
HEC Hybrid-Expansion-Contraction, with options ...
E Vector Extrapolation (5 vectors)
RS Relative Step Size Termination (0.01)
V Eigenvector Recycling for `eigs`
λ_{RP} Number of Computed Eigentriples

RELATED WORK

- Guglielmi and M.L.O., 2011
  (first large-scale method for the case $B = C = I, D = 0$)
- Kressner and Vandereycken, 2014
  (substantial improvement to G & O, but apparently does not extend beyond $B = C = I, D = 0$)
- Benner and Voigt, 2014
  (method closely related to our SVSA when $D = 0$, applicable to descriptor systems $E\dot{x} = Ax + Bu \ldots$)
- Freitag, Spence and Van Dooren, 2014
  (completely different approach, very efficient in medium-scale case)

The Real Spectral Value Set Abscissa

For $\varepsilon \geq 0$, $\varepsilon\|D\| < 1$, the real structured spectral value set abscissa is

$$\alpha_\varepsilon^{\mathbb{R},\|\cdot\|}(A,B,C,D) := \max\left\{\operatorname{Re}(\lambda) : \lambda \in \sigma_\varepsilon^{\mathbb{R},\|\cdot\|}(A,B,C,D)\right\}$$

By definition, the stability radius $\varepsilon_\star^{\mathbb{R},\|\cdot\|}$ is the zero of the monotonic function $h^{\mathbb{R}} : \mathbb{R} \to \mathbb{R}$ defined by

$$h^{\mathbb{R}}(\varepsilon) = \alpha_\varepsilon^{\mathbb{R},\|\cdot\|}(A,B,C,D)$$

We have developed Algorithm RSVSA to approximate $\varepsilon_\star^{\mathbb{R},\|\cdot\|_F}$, extending Algorithm SVSA to the case of real structure using the Frobenius norm. Key point: iterate with rank-two perturbations instead of rank-one. Based on an ODE approach originating with (Guglielmi-Lubich 2013, Guglielmi-Manetta 2014).

Reason for rank-two in a nutshell: when $u = u_R + \mathbf{i}u_I$ and $v = v_R + \mathbf{i}v_I$ are complex, which will happen even for real A, B, C, D when the rightmost eigenvalue λ is part of a complex conjugate pair,

$$\operatorname{Re}(uv^*) = u_R v_R^T + u_I v_I^T$$

which has rank two.

CONTROLLER SYNTHESIS

- Where does the (A, B, C, D) arise from?
- Typically from an open-loop state-space system:

$$\left[\begin{array}{c}\dot{x}\\ \hline z\\ y\end{array}\right] = \left[\begin{array}{c|cc}A_1 & B_1 & B_2\\ \hline C_1 & D_{11} & D_{12}\\ C_2 & D_{21} & D_{22}\end{array}\right]\left[\begin{array}{c}x\\ \hline w\\ u\end{array}\right]$$

where:

x - states (length: n_x)
u - physical (control) inputs w - performance inputs
y - physical (measured) outputs z - performance outputs

- Combined with a controller:

$$\begin{bmatrix}\dot{x}_K\\ u\end{bmatrix} = K\begin{bmatrix}x_K\\ y\end{bmatrix} = \begin{bmatrix}A_K & B_K\\ C_K & D_K\end{bmatrix}\begin{bmatrix}x_K\\ y\end{bmatrix}$$

where:

$x_K \in \mathbb{R}^{\hat{n}}$ is the controller state
$\hat{n}$ is the order of the controller

The Closed Loop System

- Combining the open-loop system system and the controller yields the *closed-loop system*, which is our (A, B, C, D):

$$\begin{bmatrix} \dot{x} \\ \dot{x}_K \\ \hline z \end{bmatrix} = \left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right] \begin{bmatrix} x \\ x_K \\ \hline w \end{bmatrix}$$

- assume $D_{22} = 0$ for brevity:

$$A = \begin{bmatrix} A_1 + B_2 D_K C_2 & B_2 C_K \\ B_K C_2 & A_K \end{bmatrix} \quad B = \begin{bmatrix} B_1 + B_2 D_K D_{21} \\ B_K D_{21} \end{bmatrix}$$

$$C = \begin{bmatrix} C_1 + D_{12} D_K C_2 & D_{12} C_K \end{bmatrix} \quad D = \begin{bmatrix} D_{11} + D_{12} D_K D_{21} \end{bmatrix}$$

- A has order $n = n_x + \hat{n}$
- How to choose the controller matrices A_K, B_K, C_K, D_K?

CONTROLLER SYNTHESIS

- Controllers are designed/built to ensure stability of the closed-loop system and increase robustness wr.t. disturbances, often by minimizing the H_∞ norm (maximizing the stability radius).
- Full-order controller ($\hat{n} = n$): can be computed explicitly, but too expensive for large n, and too complicated for engineers' tastes.
- Low-order controller ($\hat{n} \ll n$): goal can be expressed as a nonsmooth, nonconvex optimization problem in the controller variables.

HIFOO: H_∞ FIXED-ORDER OPTIMIZATION

Open source MATLAB toolbox for low-order controller design for small-scale systems (Henrion, M.L.O., et al, 2006 —).
Has been used for e.g.

- "tire damping on actively controlled quarter-car suspensions"
- "vibration control of fluid/plate"
- "lateral control for flexible BWB ... aircraft"
- "nose landing gear steering system"
- "bilateral teleoperation ... minimally invasive surgery"
- "proton exchange membrane fuel cell"
- "gust load alleviation ... direct lift control flaps"

Competitor: HINFSTRUCT, part of MATLAB's Control Systems Toolbox since 2010. Faster but closed-source. Based on (Apkarian-Noll 2006).

Both use methods for nonsmooth, nonconvex optimization to minimize the H_∞ norm of the closed loop system defined by the controller variables, as computed by implementation of BBBS algorithm in MATLAB or SLICOT.

CONTROLLER DESIGN FOR LARGE-SCALE SYSTEMS

- Usual approach: use model order reduction to reduce size of system first: much work done in this community (Antoulas, Benner, Gugercin, Mehrmann, Sorensen, etc.)
- HEC is a scalable and robust method to approximate H_∞ norm directly in the large-scale case.
- So, can it be used to design low-order controllers for large-scale dynamical systems directly?
- HIFOO + HEC: an initial evaluation
  - Compare order 10 controllers designed by:
  - original HIFOO using reduced-order models (ROM)
  - new HIFOOS (HIFOO-Sparse using HEC) using full-order models (FOM)
  - test set: 12 Problems (FOM + ROM) from *COMPl$_e$ib* (Leibfritz, 2004)
  - 144 to 168 controller variables per problem

ROM vs FOM Controller Performance

	H_∞ (ROM)	H_∞^{lb} (FOM)		
Problem	HIFOO	HIFOO	HIFOOS-W	HIFOOS-C
HF2D1	6.73×10^3	3.43×10^3	5.52×10^3	5.08×10^3
HF2D2	5.64×10^3	2.68×10^3	3.10×10^3	2.89×10^3
HF2D5	1.94×10^4	2.83×10^4	2.38×10^6	6.15×10^5
HF2D6	7.86×10^3	1.08×10^4	2.63×10^5	2.37×10^5
HF2D9	7.42×10^1	2.95×10^1	2.95×10^1	2.95×10^1
HF2D_CD1	4.62×10^0	∞	6.23×10^2	2.55×10^2
HF2D_CD2	7.01×10^0	∞	6.18×10^1	1.64×10^1
HF2D_CD3	4.30×10^0	∞	9.84×10^2	3.54×10^2
HF2D_IS1	7.57×10^4	3.32×10^5	4.21×10^6	3.54×10^6
HF2D_IS2	1.17×10^4	4.68×10^3	6.05×10^5	5.97×10^5
HF2D_IS3	8.49×10^0	∞	1.31×10^3	4.12×10^2
HF2D_IS4	6.92×10^0	∞	3.88×10^3	8.36×10^3

Column 2: controller design by HIFOO on the reduced-order model
Column 3: using this controller in the full-order model: sometimes unstable.
Columns 4 and 5: controller design by HIFOO-Sparse on the full-order model
(Warm and Cold-started versions)

CPU-TIME TO DESIGN 12 CONTROLLERS

	Time (ROM)	Time (FOM)	
Problem	HIFOO	HIFOOS-W	HIFOOS-C
HF2D1	419.37	631.70	875.76
HF2D2	714.72	967.42	994.80
HF2D5	314.47	141.75	134.42
HF2D6	316.17	41.23	12.98
HF2D9	68.75	21.23	37.52
HF2D_CD1	170.90	151.47	88.75
HF2D_CD2	175.78	95.80	46.19
HF2D_CD3	418.28	255.25	124.47
HF2D_IS1	625.17	170.09	66.62
HF2D_IS2	597.66	856.67	617.38
HF2D_IS3	164.00	249.72	275.65
HF2D_IS4	193.03	153.63	96.11
TOTAL	4178.30	3735.96	3370.65

Summary

- We have presented a fast algorithm to obtain upper bounds on the stability radius $\varepsilon_\star$ of a large-scale linear dynamical system with input and output (lower bounds on its H_∞ norm).
- It works by repeatedly expanding out in the complex plane to find the rightmost point of a spectral value set defined by a fixed ε, and then contracting back to the imaginary axis by reducing ε.
- The cost is repeated calls to `eigs` to compute the rightmost eigenvalue of a large sparse matrix plus a rank-one correction.
- The algorithm is quadratically convergent in ε.
- It extends to real structured perturbations, using rank-two corrections instead of rank-one.
- It can potentially be used to design low-order controllers for large-scale systems without model order reduction.
- Everything extends to the discrete-time case (not discussed).