

Title: Lecture 4: Semidefinite program and the navigator function

Speakers: Ning Su

Collection: Mini-Course of Numerical Conformal Bootstrap

Date: April 27, 2023 - 10:00 AM

URL: <https://pirsa.org/23040145>

Lecture 4: SDP and the navigator function

Ning Su

University of Pisa

27/04/2023 Perimeter Bootstrap Minicourse



SIMONS
FOUNDATION

SDP : Lagrangian formalism

(Reference : arXiv:1502.02033, A Semidefinite Program Solver for the Conformal Bootstrap, David Simmons-Duffin)

Lagrangian : $L = b \cdot y + c \cdot x - x \cdot B \cdot y - \text{Tr}(Y(x A - X)) - \mu \log(\det(X))$

$B \in \mathbb{R}^{P \times N}$; $x, c \in \mathbb{R}^P$; $X, Y, A_1, \dots, A_P \in \text{Sym}_K$; $b, y \in \mathbb{R}^N$; $\mu \in \mathbb{R}^+$

Constants : $\sigma \in \{b, c, B, A\}$

Free variables : $\xi \in \{x, y, X, Y\}$

(1), SDP : find stationary solution of L (i.e. $\frac{\partial L}{\partial \xi} = 0$) at the limit $\mu \rightarrow 0$ while keeping $X, Y \geq 0$

In bootstrap : b, c, B, A encode the bootstrap constraints. y encode the linear functional α .

⊖

SDP: On-shell conditions

Lagrangian : $L = b \cdot y + c \cdot x - x \cdot B \cdot y - \text{Tr}(Y(x A - X)) - \mu \log(\det[X])$.

$$\begin{aligned} \frac{\partial L}{\partial \xi} = 0 \quad \implies \quad & b - x \cdot B = 0 \\ & x A - X = 0 \\ & c - B \cdot y - \text{Tr}(Y A) = 0 \\ & X Y = \mu I \end{aligned}$$

Exercise : Derive above equations from $\frac{\partial L}{\partial \xi} = 0$

⊥

SDP : Primal-Dual formulation

On-shell constraints : (1), $b - x.B = 0$; $x.A - X = 0$ (2), $c - B.y - \text{Tr}(A.Y) = 0$ (3), $X.Y = \mu I$

Dual version:

(2), Maximize “dual objective” $\equiv b.y$ over y and $Y \geq 0$, such that $\text{tr}(A.Y) + B.y = c$

Primal version:

(3), Minimize “primal objective” $\equiv c.x$ over x and $X \geq 0$, such that $X = A.x$; $B^T.x = b$

“duality gap” $= c.x - b.y = x.B.y - \text{Tr}(Y(x.A)) - x.B.y = \text{tr}(X.Y) \geq 0$

✕

SDP : connection with PMP and bootstrap

(See arXiv:1502.02033 for the precise procedure)

SDPB input : maximize objective = $b \cdot \alpha$, s.t.

$$\alpha_n M_j^n(x=0) = 1$$

$$\alpha_n M_j^n(x) \geq 0 \text{ for } x > 0, j = 1, 2, \dots$$

PMP : maximize objective = $b_0 + b \cdot y$

$$M_j^0(x) + y_n M_j^n(x) \geq 0, j = 1, 2, \dots$$

⌈

SDP : connection with PMP and bootstrap

$$w(x) = \frac{c b^x}{(x-p_1)(x-p_2)\dots(x-p_n)} \implies \int_0^\infty q_i(x) q_j(x) w(x) dx = \delta_{i,j}$$

Define matrix $Q_\delta = (q_1, \dots, q_\delta)^T \otimes (q_1, \dots, q_\delta)$

Any polynomial of degree $P(x)$ can be written as $P = \text{tr}(Y_1 Q_{\delta_1}) + x \text{tr}(Y_2 Q_{\delta_2})$ for some $Y_1, Y_2, \delta_1, \delta_2$

$$(M_j^0(x) + y_n M_j^n(x))_{k,l} = \text{tr}(Y_1 Q_{\delta_1}(x)) + x \text{tr}(Y_2 Q_{\delta_2}(x))$$

To determine Y_1, Y_2 , we demand it hold at various points:

I

$$(M_j^0(x) + y_n M_j^n(x))_{k,l} = \text{tr}(Y_1 Q_{\delta_1}(x_q)) + x \text{tr}(Y_2 Q_{\delta_2}(x_q))$$

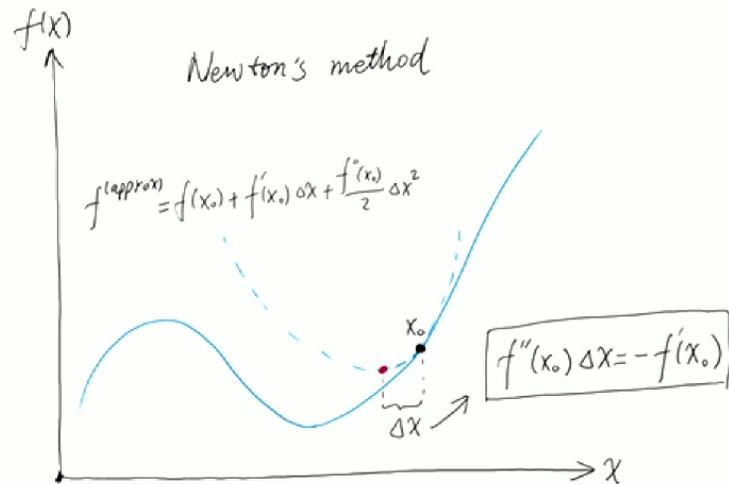
$$Y_1, Y_2 \longrightarrow Y \quad , \quad Q_{\delta_1}(x_q), Q_{\delta_2}(x_q) \longrightarrow A \quad , \quad M_j^n(x_q) \longrightarrow B$$

$$(M_j^0(x) + y_n M_j^n(x))_{k,l} = \text{tr}(Y_1 Q_{\delta_1}(x_q)) + x \text{tr}(Y_2 Q_{\delta_2}(x_q)) \longrightarrow \text{tr}(A, Y) + B y = c$$

Theorem : $Y \geq 0 \iff M_j^0(x) + y_n M_j^n(x) \geq 0$

Stationary solution : Newtonian method

Given $f(\vec{x})$, find \vec{x} satisfy $\partial f / \partial x = 0$



1, Starting at \vec{x}_0 , expand f around \vec{x}_0

2, Built a quadratic model at x_0 :

$$f^{(approx)}(\vec{x}) = f(\vec{x}_0) + \vec{g} \cdot (\vec{x} - \vec{x}_0) + \frac{1}{2} (\vec{x} - \vec{x}_0) \cdot H \cdot (\vec{x} - \vec{x}_0)$$

with $\vec{g} = \nabla f(\vec{x}_0)$, $H_{i,j} = \partial_i \partial_j f(\vec{x}_0)$

3, Find solution \vec{x} in $f^{(approx)}$: $H \cdot (\vec{x} - \vec{x}_0) = -\vec{g}$

4, Move to \vec{x} and repeat.

⌘

SDP : find stationary solution

$$f''(x_0) \Delta x = -f'(x) \xrightarrow{\text{equivalent}} \text{linearize } \frac{\partial f}{\partial x} = 0, \text{ i.e. } \frac{\partial^2 f}{\partial x^2} \Big|_{x_0} \Delta x + \frac{\partial f}{\partial x} \Big|_{x_0} = 0$$

On-shell constraints : (1), $b - x \cdot B = 0$; $x A - X = 0$ (2), $c - B \cdot y - \text{Tr}(Y \cdot A) = 0$ (3), $X Y = \mu I$

Linearized constraints :

$$\text{tr}(A \cdot (Y + dY)) + B \cdot (y + dy) = c$$

$$X + dX = A_p(x_p + dx_p)$$

$$B^T(x + dx) = b$$

$$XY + XdY + dXY = \mu I$$

⌘

SDP : Schur complement equation

I

On-shell constraints : (1), $b - x.B = 0$; $x.A - X = 0$ (2), $c - B.y - \text{Tr}(A, Y) = 0$ (3), $X Y = \mu I$

$$(4), \text{tr}(A, (Y + dY)) + B(y + dy) = c \quad \left(\begin{array}{cc} S_{ij} & -B \\ B^T & 0 \end{array} \right) \left(\begin{array}{c} dx \\ dy \end{array} \right) = \left(\begin{array}{c} -d - \text{Tr}(A, Z) \\ p \end{array} \right)$$

$$(5), X + dX = A_p(x_p + dx_p) \quad \Rightarrow \quad dX = P + A_p dx_p$$

$$(6), B^T(x + dx) = b \quad dY = X^{-1}(R - dXY)$$

$$(7), XY + X(dY) + (dX) Y = \mu I$$

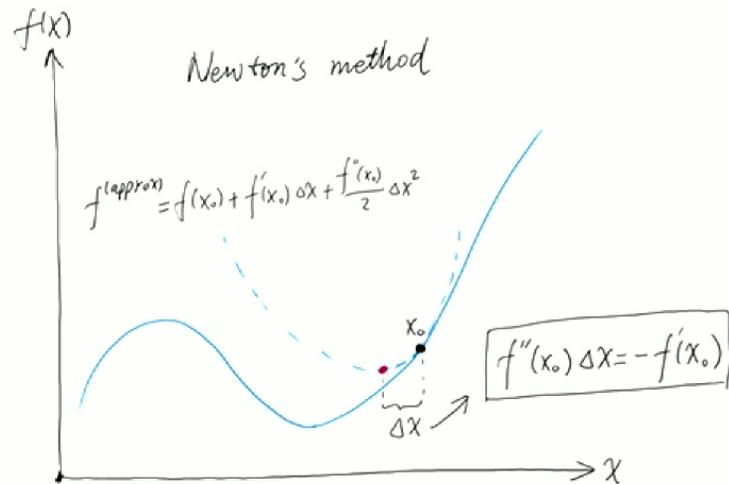
$$P \equiv A.p - X; \quad p \equiv b - B^T x; \quad d \equiv c - \text{Tr}(A, Y) - B.y; \quad R \equiv \mu I - XY \quad (\text{errors})$$

$$Z \equiv X^{-1}(P Y - R); \quad S_{ij} \equiv \text{Tr}(A_i X^{-1} A_j Y)$$

Exercise : derive Schur complement equation using (4)-(7). Strategy: Starting from (4), eliminate dY using (7), then eliminate dX using (5). Put the result and (6) in matrix form.

Stationary solution : Newtonian method

Given $f(\vec{x})$, find \vec{x} satisfy $\partial f / \partial x = 0$



1, Starting at \vec{x}_0 , expand f around \vec{x}_0

2, Built a quadratic model at x_0 :

$$f^{(approx)}(\vec{x}) = f(\vec{x}_0) + \vec{g} \cdot (\vec{x} - \vec{x}_0) + \frac{1}{2} (\vec{x} - \vec{x}_0) \cdot H \cdot (\vec{x} - \vec{x}_0)$$

with $\vec{g} = \nabla f(\vec{x}_0)$, $H_{i,j} = \partial_i \partial_j f(\vec{x}_0)$

3, Find solution \vec{x} in $f^{(approx)}$: $H \cdot (\vec{x} - \vec{x}_0) = -\vec{g}$

4, Move to \vec{x} and repeat.

SDP : naive Newtonian method

Step 1 : Taking $\xi_0 = (x_0, y_0, X_0, Y_0)$, computing P, p, d, R, Z, S_{ij}

Step 2 : Plug in to Schur complement equation RHS, solve for dx, dy, dX, dY

$$\begin{pmatrix} S & -B \\ B^T & 0 \end{pmatrix} \begin{pmatrix} dx \\ dy \end{pmatrix} = \begin{pmatrix} -d - \text{Tr}(A, Z) \\ p \end{pmatrix}$$

$$dX = P + A_p dx_p$$

$$dY = X^{-1}(R - dXY)$$

Step 3 : update $(x_0, y_0, X_0, Y_0) \rightarrow (x_0 + dx, y_0 + dy, X_0 + dX, Y_0 + dY)$. Repeat.

Caveat: (1), we want solution at $\mu \rightarrow 0$, not finite μ ; (2), we want $X, Y \geq 0$, but $X_0 + dX$ may ruins it.

⊖

SDP : Schur complement equation

On-shell constraints : (1), $b - x.B = 0$; $x.A - X = 0$ (2), $c - B.y - \text{Tr}(A, Y) = 0$ (3), $X Y = \mu I$

$$(4), \text{tr}(A, (Y + dY)) + B(y + dy) = c \quad \Rightarrow \quad \begin{pmatrix} S_{ij} & -B \\ B^T & 0 \end{pmatrix} \begin{pmatrix} dx \\ dy \end{pmatrix} = \begin{pmatrix} -d - \text{Tr}(A, Z) \\ p \end{pmatrix}$$

$$(5), X + dX = A_p(x_p + dx_p) \quad \Rightarrow \quad dX = P + A_p dx_p$$

$$(6), B^T(x + dx) = b \quad dY = X^{-1}(R - dXY)$$

$$(7), XY + X(dY) + (dX) Y = \mu I$$

$$P \equiv A.p - X; \quad p \equiv b - B^T x; \quad d \equiv c - \text{Tr}(A, Y) - B.y; \quad R \equiv \mu I - XY \quad (\text{errors})$$

$$Z \equiv X^{-1}(P Y - R); \quad S_{ij} \equiv \text{Tr}(A_i X^{-1} A_j Y)$$

Exercise : derive Schur complement equation using (4)-(7). Strategy: Starting from (4), eliminate dY using (7), then eliminate dX using (5). Put the result and (6) in matrix form.

SDP : Interior point method (SDPB)

1, Step length test : $\alpha(X, dX)$: largest α such that $X + \alpha dX \geq 0$

Update $(x_0, y_0, X_0, Y_0) \rightarrow (x_0 + s dx, y_0 + s dy, X_0 + s dX, Y_0 + s dY)$ with $s = \min(0.7 \alpha(X, dX), 1)$

2, When computing Schur complement : $\mu \rightarrow \beta \mu$ ($\beta = 0.3$)

Initial choice $x = 0, y = 0, X = 10^{20} I, Y = 10^{20} I, \mu = \text{tr}(X Y) / \text{dim}(X) = 10^{40}$

	time	mu	P obj	D obj	gap	P err	p err	D err	P step	D step	beta
1	27	1.0e+40	-1.53e-07	-1.53e-07	0.00	-1.00e+20	-3.62e+18	-5.03e+21	0.298	0.379	0.300
2	52	8.0e+39	-7.86e+18	-9.27e+20	0.983	-7.02e+19	-2.54e+18	-3.12e+21	0.347	0.338	0.300
3	76	6.4e+39	-1.94e+19	-3.15e+20	0.884	-4.58e+19	-1.66e+18	-2.07e+21	0.256	0.381	0.300
4	101	5.2e+39	-2.84e+19	-3.95e+20	1.00	-3.41e+19	-1.23e+18	-1.28e+21	0.278	0.331	0.300
5	125	4.2e+39	-4.06e+19	-7.30e+20	1.00	-2.46e+19	-8.92e+17	-8.56e+20	0.337	0.349	0.300
6	149	3.4e+39	-5.87e+19	-8.96e+20	1.00	-1.63e+19	-5.91e+17	-5.57e+20	0.263	0.336	0.300
7	174	2.7e+39	-7.55e+19	-8.87e+20	1.00	-1.20e+19	-4.36e+17	-3.70e+20	0.229	0.265	0.300
8	198	2.4e+39	-9.26e+19	-8.90e+20	1.00	-9.28e+18	-3.36e+17	-2.72e+20	0.198	0.290	0.300
9	222	2.0e+39	-1.10e+20	-9.05e+20	1.00	-7.44e+18	-2.70e+17	-1.93e+20	0.185	0.220	0.300
10	247	1.8e+39	-1.29e+20	-9.04e+20	1.00	-6.06e+18	-2.20e+17	-1.51e+20	0.177	0.284	0.300
11	272	1.6e+39	-1.51e+20	-9.02e+20	1.00	-4.99e+18	-1.81e+17	-1.08e+20	0.204	0.331	0.300
12	296	1.3e+39	-1.84e+20	-9.08e+20	1.00	-3.97e+18	-1.44e+17	-7.20e+19	0.294	0.327	0.300
13	320	1.1e+39	-2.45e+20	-9.15e+20	1.00	-2.80e+18	-1.02e+17	-4.85e+19	0.184	0.315	0.300

Hotstart : take (x, y, X, Y) of an old SDP as initial state for a new SDP.

SDP : naive Newtonian method

Step 1 : Taking $\xi_0 = (x_0, y_0, X_0, Y_0)$, computing P, p, d, R, Z, S_{ij}

Step 2 : Plug in to Schur complement equation RHS, solve for dx, dy, dX, dY

$$\begin{pmatrix} S & -B \\ B^T & 0 \end{pmatrix} \begin{pmatrix} dx \\ dy \end{pmatrix} = \begin{pmatrix} -d - \text{Tr}(A, Z) \\ p \end{pmatrix}$$

$$dX = P + A_p dx_p$$

$$dY = X^{-1}(R - dXY)$$

Step 3 : update $(x_0, y_0, X_0, Y_0) \rightarrow (x_0 + dx, y_0 + dy, X_0 + dX, Y_0 + dY)$. Repeat.

Caveat: (1), we want solution at $\mu \rightarrow 0$, not finite μ ; (2), we want $X, Y \geq 0$, but $X_0 + dX$ may ruins it.

SDP : Interior point method (SDPB)

1, Step length test : $\alpha(X, dX)$: largest α such that $X + \alpha dX \geq \emptyset$

Update $(x_0, y_0, X_0, Y_0) \rightarrow (x_0 + s dx, y_0 + s dy, X_0 + s dX, Y_0 + s dY)$ with $s = \min(0.7 \alpha(X, dX), 1)$

2, When computing Schur complement : $\mu \rightarrow \beta \mu$ ($\beta = 0.3$)

Initial choice $x = 0, y = 0, X = 10^{20} I, Y = 10^{20} I, \mu = \text{tr}(X Y) / \text{dim}(X) = 10^{40}$

	time	mu	P obj	D obj	gap	P err	p err	D err	P step	D step	beta
1	27	1.0e+40	-1.53e-07	-1.53e-07	0.00	-1.00e+20	-3.62e+18	-5.03e+21	0.298	0.379	0.300
2	52	8.0e+39	-7.86e+18	-9.27e+20	0.983	-7.02e+19	-2.54e+18	-3.12e+21	0.347	0.338	0.300
3	76	6.4e+39	-1.94e+19	-3.15e+20	0.884	-4.58e+19	-1.66e+18	-2.07e+21	0.256	0.381	0.300
4	101	5.2e+39	-2.84e+19	-3.95e+20	1.00	-3.41e+19	-1.23e+18	-1.28e+21	0.278	0.331	0.300
5	125	4.2e+39	-4.06e+19	-7.30e+20	1.00	-2.46e+19	-8.92e+17	-8.56e+20	0.337	0.349	0.300
6	149	3.4e+39	-5.87e+19	-8.96e+20	1.00	-1.63e+19	-5.91e+17	-5.57e+20	0.263	0.336	0.300
7	174	2.7e+39	-7.55e+19	-8.87e+20	1.00	-1.20e+19	-4.36e+17	-3.70e+20	0.229	0.265	0.300
8	198	2.4e+39	-9.26e+19	-8.90e+20	1.00	-9.28e+18	-3.36e+17	-2.72e+20	0.198	0.290	0.300
9	222	2.0e+39	-1.10e+20	-9.05e+20	1.00	-7.44e+18	-2.70e+17	-1.93e+20	0.185	0.220	0.300
10	247	1.8e+39	-1.29e+20	-9.04e+20	1.00	-6.06e+18	-2.20e+17	-1.51e+20	0.177	0.284	0.300
11	272	1.6e+39	-1.51e+20	-9.02e+20	1.00	-4.99e+18	-1.81e+17	-1.08e+20	0.204	0.331	0.300
12	296	1.3e+39	-1.84e+20	-9.08e+20	1.00	-3.97e+18	-1.44e+17	-7.20e+19	0.294	0.327	0.300
13	320	1.1e+39	-2.45e+20	-9.15e+20	1.00	-2.80e+18	-1.02e+17	-4.85e+19	0.184	0.315	0.300
..

Hotstart : take (x, y, X, Y) of an old SDP as initial state for a new SDP.

SDP : naive Newtonian method

Step 1 : Taking $\xi_0 = (x_0, y_0, X_0, Y_0)$, computing P, p, d, R, Z, S_{ij}

Step 2 : Plug in to Schur complement equation RHS, solve for dx, dy, dX, dY

$$\begin{pmatrix} S & -B \\ B^T & 0 \end{pmatrix} \begin{pmatrix} dx \\ dy \end{pmatrix} = \begin{pmatrix} -d - \text{Tr}(A, Z) \\ p \end{pmatrix}$$

$$dX = P + A_p dx_p$$

$$dY = X^{-1}(R - dXY)$$

Step 3 : update $(x_0, y_0, X_0, Y_0) \rightarrow (x_0 + dx, y_0 + dy, X_0 + dX, Y_0 + dY)$. Repeat.

Caveat: (1), we want solution at $\mu \rightarrow 0$, not finite μ ; (2), we want $X, Y \geq 0$, but $X_0 + dX$ may ruins it.



SDP : Interior point method (SDPB)

1, Step length test : $\alpha(X, dX)$: largest α such that $X + \alpha dX \geq 0$

Update $(x_0, y_0, X_0, Y_0) \rightarrow (x_0 + s dx, y_0 + s dy, X_0 + s dX, Y_0 + s dY)$ with $s = \min(0.7 \alpha(X, dX), 1)$

2, When computing Schur complement : $\mu \rightarrow \beta \mu$ ($\beta = 0.3$)

Initial choice $x = 0, y = 0, X = 10^{20} I, Y = 10^{20} I, \mu = \text{tr}(X Y) / \text{dim}(X) = 10^{40}$

	time	mu	P obj	D obj	gap	P err	p err	D err	P step	D step	beta
1	27	1.0e+40	-1.53e-07	-1.53e-07	0.00	-1.00e+20	-3.62e+18	-5.03e+21	0.298	0.379	0.300
2	52	8.0e+39	-7.86e+18	-9.27e+20	0.983	-7.02e+19	-2.54e+18	-3.12e+21	0.347	0.338	0.300
3	76	6.4e+39	-1.94e+19	-3.15e+20	0.884	-4.58e+19	-1.66e+18	-2.07e+21	0.256	0.381	0.300
4	101	5.2e+39	-2.84e+19	-3.95e+20	1.00	-3.41e+19	-1.23e+18	-1.28e+21	0.278	0.331	0.300
5	125	4.2e+39	-4.06e+19	-7.30e+20	1.00	-2.46e+19	-8.92e+17	-8.56e+20	0.337	0.349	0.300
6	149	3.4e+39	-5.87e+19	-8.96e+20	1.00	-1.63e+19	-5.91e+17	-5.57e+20	0.263	0.336	0.300
7	174	2.7e+39	-7.55e+19	-8.87e+20	1.00	-1.20e+19	-4.36e+17	-3.70e+20	0.229	0.265	0.300
8	198	2.4e+39	-9.26e+19	-8.90e+20	1.00	-9.28e+18	-3.36e+17	-2.72e+20	0.198	0.290	0.300
9	222	2.0e+39	-1.10e+20	-9.05e+20	1.00	-7.44e+18	-2.70e+17	-1.93e+20	0.185	0.220	0.300
10	247	1.8e+39	-1.29e+20	-9.04e+20	1.00	-6.06e+18	-2.20e+17	-1.51e+20	0.177	0.284	0.300
11	272	1.6e+39	-1.51e+20	-9.02e+20	1.00	-4.99e+18	-1.81e+17	-1.08e+20	0.204	0.331	0.300
12	296	1.3e+39	-1.84e+20	-9.08e+20	1.00	-3.97e+18	-1.44e+17	-7.20e+19	0.294	0.327	0.300
13	320	1.1e+39	-2.45e+20	-9.15e+20	1.00	-2.80e+18	-1.02e+17	-4.85e+19	0.184	0.315	0.300
..

Hotstart : take (x, y, X, Y) of an old SDP as initial state for a new SDP.

SDP : Schur complement equation

On-shell constraints : (1), $b - x.B = 0$; $x.A - X = 0$ (2), $c - B.y - \text{Tr}(A, Y) = 0$ (3), $X Y = \mu I$

$$(4), \text{tr}(A, (Y + dY)) + B(y + dy) = c \quad \left(\begin{array}{cc} S_{ij} & -B \\ B^T & 0 \end{array} \right) \begin{pmatrix} dx \\ dy \end{pmatrix} = \begin{pmatrix} -d - \text{Tr}(A, Z) \\ p \end{pmatrix}$$

$$(5), X + dX = A_p(x_p + dx_p) \quad \Rightarrow \quad dX = P + A_p dx_p$$

$$(6), B^T(x + dx) = b \quad dY = X^{-1}(R - dXY)$$

$$(7), XY + X(dY) + (dX) Y = \mu I$$

$$P \equiv A.p - X; \quad p \equiv b - B^T x; \quad d \equiv c - \text{Tr}(A, Y) - B.y; \quad R \equiv \mu I - XY \quad (\text{errors})$$

$$Z \equiv X^{-1}(P Y - R); \quad S_{ij} \equiv \text{Tr}(A_i X^{-1} A_j Y)$$

Exercise : derive Schur complement equation using (4)-(7). Strategy: Starting from (4), eliminate dY using (7), then eliminate dX using (5). Put the result and (6) in matrix form.

SDP : Interior point method (SDPB) I

1, Step length test : $\alpha(X, dX)$: largest α such that $X + \alpha dX \geq 0$

Update $(x_0, y_0, X_0, Y_0) \rightarrow (x_0 + s dx, y_0 + s dy, X_0 + s dX, Y_0 + s dY)$ with $s = \min(0.7 \alpha(X, dX), 1)$

2, When computing Schur complement : $\mu \rightarrow \beta \mu$ ($\beta = 0.3$)

Initial choice $x = 0, y = 0, X = 10^{20} I, Y = 10^{20} I, \mu = \text{tr}(X Y) / \text{dim}(X) = 10^{40}$

	time	mu	P obj	D obj	gap	P err	p err	D err	P step	D step	beta
1	27	1.0e+40	-1.53e-07	-1.53e-07	0.00	-1.00e+20	-3.62e+18	-5.03e+21	0.298	0.379	0.300
2	52	8.0e+39	-7.86e+18	-9.27e+20	0.983	-7.02e+19	-2.54e+18	-3.12e+21	0.347	0.338	0.300
3	76	6.4e+39	-1.94e+19	-3.15e+20	0.884	-4.58e+19	-1.66e+18	-2.07e+21	0.256	0.381	0.300
4	101	5.2e+39	-2.84e+19	-3.95e+20	1.00	-3.41e+19	-1.23e+18	-1.28e+21	0.278	0.331	0.300
5	125	4.2e+39	-4.06e+19	-7.30e+20	1.00	-2.46e+19	-8.92e+17	-8.56e+20	0.337	0.349	0.300
6	149	3.4e+39	-5.87e+19	-8.96e+20	1.00	-1.63e+19	-5.91e+17	-5.57e+20	0.263	0.336	0.300
7	174	2.7e+39	-7.55e+19	-8.87e+20	1.00	-1.20e+19	-4.36e+17	-3.70e+20	0.229	0.265	0.300
8	198	2.4e+39	-9.26e+19	-8.90e+20	1.00	-9.28e+18	-3.36e+17	-2.72e+20	0.198	0.290	0.300
9	222	2.0e+39	-1.10e+20	-9.05e+20	1.00	-7.44e+18	-2.70e+17	-1.93e+20	0.185	0.220	0.300
10	247	1.8e+39	-1.29e+20	-9.04e+20	1.00	-6.06e+18	-2.20e+17	-1.51e+20	0.177	0.284	0.300
11	272	1.6e+39	-1.51e+20	-9.02e+20	1.00	-4.99e+18	-1.81e+17	-1.08e+20	0.204	0.331	0.300
12	296	1.3e+39	-1.84e+20	-9.08e+20	1.00	-3.97e+18	-1.44e+17	-7.20e+19	0.294	0.327	0.300
13	320	1.1e+39	-2.45e+20	-9.15e+20	1.00	-2.80e+18	-1.02e+17	-4.85e+19	0.184	0.315	0.300
..

Hotstart : take (x, y, X, Y) of an old SDP as initial state for a new SDP.

Slater condition

- (1), Find stationary solution of L at the limit $\mu \rightarrow 0$ while keeping $X, Y \geq 0$
- (2) (Dual version), Maximize “dual objective” $\equiv b \cdot y$ over y and $Y \geq 0$, such that $\text{tr}(A, Y) + B y = c$
- (3) (Primal version), Minimize “primal objective” $\equiv c \cdot x$ over x and $X \geq 0$, such that $X = A x; B^T \cdot x = b$

If $\text{tr}(A, Y) + B y = c, X = A x, B^T \cdot x = b$ and has a solution with $X > 0, Y > 0$ (no equal sign), then (1) has solution in at $\mu \rightarrow 0$

⌘

SDPB : primal / dual feasible jump

“primal jump detected” : find x such that $X = A x ; B^T x = b$ and $X > 0$ (no equal sign)

“dual jump detected” : find y such that $c - B \cdot y - \text{Tr}(A, Y) = 0$ and $Y > 0$ (no equal sign)

“dual jump detected” \rightarrow find $\alpha \rightarrow$ bootstrap equation has no solution

Why in bootstrap “primal jump detected” = allowed point?

✕

SDP in feasibility mode

Set $b = 0$. Ask if solution (x, y, X, Y) exist for

$$X = A x, B^T x = b = 0, X > 0$$

$$c - B.y - \text{Tr}(A, Y) = 0, Y > 0$$

“dual jump detected” : infeasible

“primal jump detected” and $c.x < 0$: feasible

“primal jump detected” and $c.x < 0$ AND we find a “dual jump detected” later :

$$c.x - b.y = c.x = \text{tr}(X Y) \geq 0$$

I

SDP in feasibility mode

Set $b = 0$. Ask if solution (x, y, X, Y) exist for

$$X = A x, B^T x = b = 0, X \succ 0$$

$$c - B^T y - \text{Tr}(A, Y) = 0, Y \succ 0$$

“dual jump detected” : infeasible

“primal jump detected” and $c \cdot x < 0$: feasible

$x = 0$ is a solution to “ $X = A x, B^T x = b = 0, X \succeq 0$ ” : $c \cdot x \leq 0$

Conclusion : “primal jump detected” : feasible



Goals for today

• *Semidefinite program (SDP)*

• SDP definition

• SDP solver algorithm

• Primal / dual jump and SDP in feasibility mode

• *Navigator function*

Reference : [Reehorst, Rychkov, Simmons-Duffin, Sirois, SN, van Rees 2021]

• *Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm*

• *Afternoon : tutorials for using the navigator function*

Result from Wednesday tutorial

```
{0.51, 1.}    primal -6.2301 × 1025
{0.515, 1.}   primal -7.8617 × 1025
{0.515, 1.5}  primal -3.5328 × 108
{0.52, 1.}    primal -1.1536 × 1026
{0.52, 1.5}   primal -3.6487 × 108
{0.51333, 1.3333} primal -8.4373 × 108
{0.51111, 1.2778} primal -2.0032 × 109
{0.51352, 1.4629} primal -6.5707 × 108
{0.51759, 1.537} primal -4.8335 × 107
{0.51037, 1.2592} primal -3.3872 × 109
{0.51111, 1.4444} primal -3.054 × 109
{0.51599, 1.5617} primal -2.6262 × 108
{0.51889, 1.5555} primal -5.5408 × 106
{0.51228, 1.4876} primal -1.222 × 109
{0.51049, 1.4012} primal -3.4749 × 109
{0.51012, 1.2531} primal -5.5324 × 109
{0.51407, 1.5061} primal -9.1704 × 108
{0.51951, 1.5617} primal -2.798 × 107
{0.51317, 1.502}  primal -1.5068 × 109
{0.51021, 1.3847} primal -5.1195 × 109
{0.51858, 1.5987} primal -2.1818 × 108
{0.51545, 1.5514} primal -3.8794 × 107
_
```

Slater condition

- (1), Find stationary solution of L at the limit $\mu \rightarrow 0$ while keeping $X, Y \geq 0$
- (2) (Dual version), Maximize “dual objective” $\equiv b \cdot y$ over y and $Y \geq 0$, such that $\text{tr}(A, Y) + B y = c$
- (3) (Primal version), Minimize “primal objective” $\equiv c \cdot x$ over x and $X \geq 0$, such that $X = A x; B^T \cdot x = b$

If $\text{tr}(A, Y) + B y = c, X = A x, B^T \cdot x = b$ and has a solution with $X > 0, Y > 0$ (no equal sign), then (1) has solution in at $\mu \rightarrow 0$

Result from Wednesday tutorial

```
{0.51, 1.}    primal -6.2301 × 1025
{0.515, 1.}  primal -7.8617 × 1025
{0.515, 1.5} primal -3.5328 × 108
{0.52, 1.}   primal -1.1536 × 1026
{0.52, 1.5}  primal -3.6487 × 108
{0.51333, 1.3333} primal -8.4373 × 108
{0.51111, 1.2778} primal -2.0032 × 109
{0.51352, 1.4629} primal -6.5707 × 108
{0.51759, 1.537}  primal -4.8335 × 107
{0.51037, 1.2592} primal -3.3872 × 109
{0.51111, 1.4444} primal -3.054 × 109
{0.51599, 1.5617} primal -2.6262 × 108
{0.51889, 1.5555} primal -5.5408 × 106
{0.51228, 1.4876} primal -1.222 × 109
{0.51049, 1.4012} primal -3.4749 × 109
{0.51012, 1.2531} primal -5.5324 × 109
{0.51407, 1.5061} primal -9.1704 × 108
{0.51951, 1.5617} primal -2.798 × 107
{0.51317, 1.502}  primal -1.5068 × 109
{0.51021, 1.3847} primal -5.1195 × 109
{0.51858, 1.5987} primal -2.1818 × 108
{0.51545, 1.5514} primal -3.8794 × 107
_
```

Slater condition

Next Slide

- (1), Find stationary solution of L at the limit $\mu \rightarrow 0$ while keeping $X, Y \geq 0$
- (2) (Dual version), Maximize “dual objective” $\equiv b \cdot y$ over y and $Y \geq 0$, such that $\text{tr}(A, Y) + B y = c$
- (3) (Primal version), Minimize “primal objective” $\equiv c \cdot x$ over x and $X \geq 0$, such that $X = A x; B^T \cdot x = b$

If $\text{tr}(A, Y) + B y = c, X = A x, B^T \cdot x = b$ and has a solution with $X > 0, Y > 0$ (no equal sign), then (1) has solution in at $\mu \rightarrow 0$

SDPB : primal / dual feasible jump

“primal jump detected” : find x such that $X = A x ; B^T x = b$ and $X > 0$ (no equal sign)

“dual jump detected” : find y such that $c - B \cdot y - \text{Tr}(A, Y) = 0$ and $Y > 0$ (no equal sign)

“dual jump detected” \rightarrow find $\alpha \rightarrow$ bootstrap equation has no solution

Why in bootstrap “primal jump detected” = allowed point?

Slater condition

- (1), Find stationary solution of L at the limit $\mu \rightarrow 0$ while keeping $X, Y \geq 0$
- (2) (Dual version), Maximize “dual objective” $\equiv b \cdot y$ over y and $Y \geq 0$, such that $\text{tr}(A, Y) + B y = c$
- (3) (Primal version), Minimize “primal objective” $\equiv c \cdot x$ over x and $X \geq 0$, such that $X = A x; B^T \cdot x = b$

If $\text{tr}(A, Y) + B y = c, X = A x, B^T \cdot x = b$ and has a solution with $X > 0, Y > 0$ (no equal sign), then (1) has solution in at $\mu \rightarrow 0$

SDP in feasibility mode

Set $b = 0$. Ask if solution (x, y, X, Y) exist for

$$X = A x, B^T x = b = 0, X \succ 0$$

$$c - B^T y - \text{Tr}(A, Y) = 0, Y \succ 0$$

“dual jump detected” : infeasible

“primal jump detected” and $c \cdot x < 0$: feasible

$x = 0$ is a solution to “ $X = A x, B^T x = b = 0, X \succeq 0$ ” : $c \cdot x \leq 0$

Conclusion : “primal jump detected” : feasible

⌘

SDP in feasibility mode

Set $b = 0$. Ask if solution (x, y, X, Y) exist for

$$X = A x, B^T x = b = 0, X \succ 0$$

$$c - B \cdot y - \text{Tr}(A, Y) = 0, Y \succ 0$$

“dual jump detected” : infeasible

“primal jump detected” and $c \cdot x < 0$: feasible

“primal jump detected” and $c \cdot x < 0$ AND we find a “dual jump detected” later :

$$c \cdot x - b \cdot y = c \cdot x = \text{tr}(X Y) \geq 0$$

Review of feasibility mode

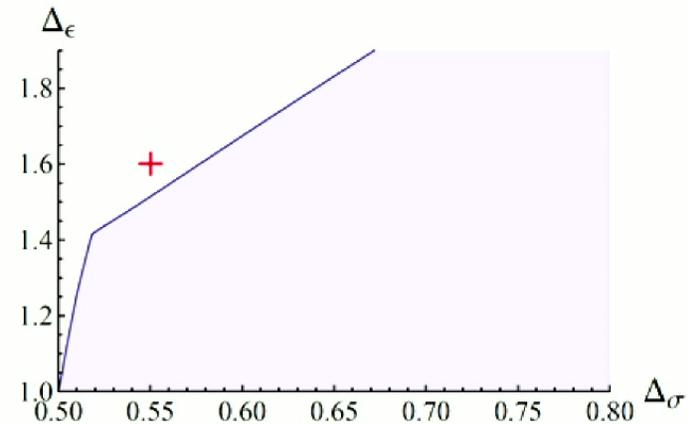
$$\sum_{\mathcal{O} \in \phi \times \phi} \lambda_{\phi \phi \mathcal{O}}^2 F_{\Delta, \ell}^{\sigma \sigma \sigma \sigma}(u, v) = 0$$

Find a linear functional α satisfy

$$\alpha(F_{\Delta=0, \ell=0}^{\sigma \sigma \sigma \sigma}(u, v)) > 0$$

$$\alpha(F_{\Delta, \ell=0}^{\sigma \sigma \sigma \sigma}(u, v)) \geq 0 \text{ for } \Delta \geq \Delta_{\epsilon}$$

$$\alpha(F_{\Delta, \ell}^{\sigma \sigma \sigma \sigma}(u, v)) \geq 0 \text{ for } \ell > 0, \Delta \geq \Delta_{\text{unitary}}$$



“Find α at +” $\Rightarrow \alpha \cdot (\text{bootstrap equation}) \Rightarrow \sum \text{positive number} = 0 \Rightarrow$ “+ is excluded”

⊘

Examples of $\alpha \cdot F_{\Delta=0, \ell=0}^{\sigma\sigma\sigma\sigma}$

$$\alpha = \sum_{n+m \leq \Lambda=11} C_{n m} \partial_z^n \partial_{\bar{z}}^m \Big|_{z=\bar{z}=\frac{1}{2}}, \quad \alpha \cdot \alpha = 1 \text{ i.e. } \sum_{n m} C_{n m} C_{n m} = 1$$

At disallowed points :

run	$\alpha \cdot F_{\Delta=0, \ell=0}^{\sigma\sigma\sigma\sigma}$	$\min(\alpha \cdot F_{\Delta \geq \Delta_{\ell}, \ell=0}^{\sigma\sigma\sigma\sigma})$	$\min(\alpha \cdot F_{\Delta, \ell=2}^{\sigma\sigma\sigma\sigma})$	$\min(\alpha \cdot F_{\Delta, \ell=4}^{\sigma\sigma\sigma\sigma})$	$\min(\alpha \cdot F_{\Delta, \ell=6}^{\sigma\sigma\sigma\sigma})$
1	1.7581×10^{-23}	0.000084458	0.00018025	0.021167	0.011633
2	3.5297×10^{-17}	0.000022712	0.000022478	0.0052397	0.025652
3	1.6527×10^{-16}	0.000046844	0.000097581	0.018461	0.017934

At allowed points :

run	$\alpha \cdot F_{\Delta=0, \ell=0}^{\sigma\sigma\sigma\sigma}$	$\min(\alpha \cdot F_{\Delta \geq \Delta_{\ell}, \ell=0}^{\sigma\sigma\sigma\sigma})$	$\min(\alpha \cdot F_{\Delta, \ell=2}^{\sigma\sigma\sigma\sigma})$	$\min(\alpha \cdot F_{\Delta, \ell=4}^{\sigma\sigma\sigma\sigma})$	$\min(\alpha \cdot F_{\Delta, \ell=6}^{\sigma\sigma\sigma\sigma})$
1	-2.3514×10^{-15}	-0.000084458	-0.00004048	0.010235	0.039432
2	4.39345×10^{-10}	0.00013043	-0.00090309	0.044526	0.194500

Fact : We can always find α s.t. $\alpha \cdot F_{\Delta, \ell} \geq 0$ for all $\Delta \geq \Delta_{\text{unitary}}$ and ℓ

Strategy : ensure $\alpha \cdot F_{\Delta, \ell} \geq 0$ except $\alpha \cdot F_{0,0}$, then maximizing $\alpha \cdot F_{\Delta=0, \ell=0}^{\sigma\sigma\sigma\sigma}$

↔

Navigator function

$$F_{\Delta=0, \ell=0}^{\sigma\sigma\sigma\sigma}(u, v) + \sum_{\Delta, \ell \geq \Delta_\epsilon, \ell=0} \lambda_{\phi\phi}^2 F_{\Delta, \ell}^{\sigma\sigma\sigma\sigma}(u, v) + \sum_{\Delta, \ell \geq \Delta_{\text{unitary}}, \ell=0} \lambda_{\phi\phi}^2 F_{\Delta, \ell}^{\sigma\sigma\sigma\sigma}(u, v) = 0$$

Maximize objective = $\alpha \cdot (F_{\Delta=0, \ell=0}^{\sigma\sigma\sigma\sigma}(u, v))$, such that

$$\alpha \cdot (F_{\Delta, \ell=0}^{\sigma\sigma\sigma\sigma}(u, v)) \geq 0 \text{ for } \Delta \geq \Delta_\epsilon$$

$$\alpha \cdot (F_{\Delta, \ell}^{\sigma\sigma\sigma\sigma}(u, v)) \geq 0 \text{ for } \ell > 0, \Delta \geq \Delta_{\text{unitary}}$$

$$\alpha \cdot (\text{"a good normalization vector"}) = 1$$

Navigator function : objective as a function depends on the parameters

⋈

Geometric picture

$$F_{\Delta=0, \ell=0}^{\sigma\sigma\sigma\sigma}(u, v) + \sum_{O, \Delta_0 \geq \Delta_\epsilon, \ell=0} \lambda_{\phi\phi} O^2 F_{\Delta, \ell}^{\sigma\sigma\sigma\sigma}(u, v) + \sum_{O, \Delta_0 \geq \Delta_{\text{unitary}}, \ell \geq 0} \lambda_{\phi\phi} O^2 F_{\Delta, \ell}^{\sigma\sigma\sigma\sigma}(u, v) = 0$$

maximize objective = $\alpha(F_{\Delta=0, \ell=0}^{\sigma\sigma\sigma\sigma}(u, v))$, s.t.

$$\alpha \cdot \alpha = 1$$

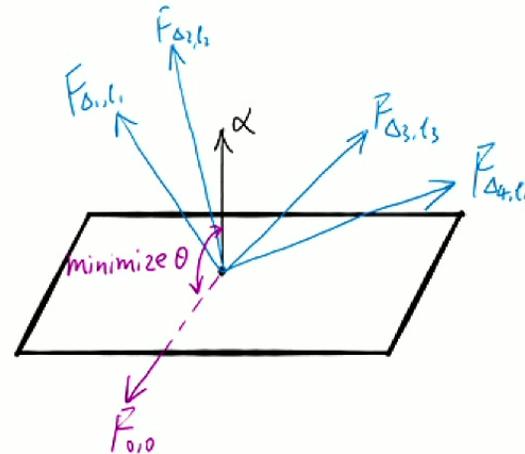
$$\alpha(F_{\Delta, \ell=0}^{\sigma\sigma\sigma\sigma}(u, v)) \geq 0 \text{ for } \Delta \geq \Delta_\epsilon$$

$$\alpha(F_{\Delta, \ell}^{\sigma\sigma\sigma\sigma}(u, v)) \geq 0 \text{ for } \ell > 0, \Delta \geq \Delta_{\text{unitary}}$$

$$\cos(\theta) = \frac{\alpha \cdot F_{0,0}}{|\alpha| |F_{0,0}|}$$

$\theta \leq 90^\circ$: disallowed

$\theta > 90^\circ$: allowed

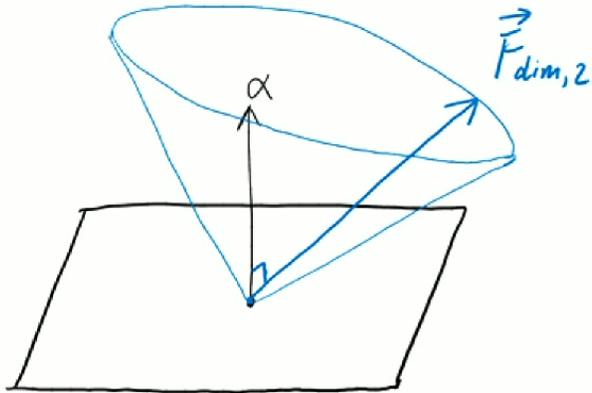


Unfortunately $\alpha \cdot \alpha = 1$ can not translate to standard SDP

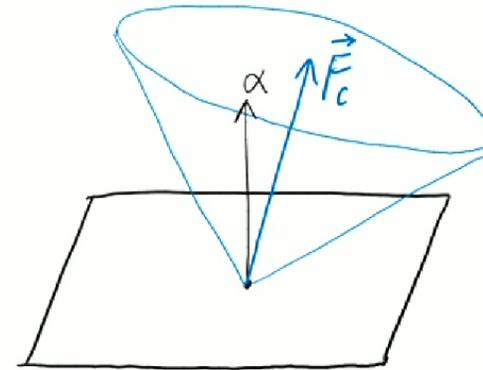
⊥

A good normalization vector

$\alpha(F_{\Delta=\dim, \ell=2}(u, v)) = 1$ **doesn't work**: $\alpha(F_{\Delta=0, \ell=0}(u, v)) \rightarrow +\infty$ in disallowed region ($\alpha \cdot \alpha = \infty$)



v.s.



For F_b on the **boundary** of the cone:

$\exists \alpha$ s.t. $\alpha \perp F_b$ i.e. $\alpha \cdot F_b = \epsilon \sim 0$

$\xrightarrow{\alpha' = \alpha/\epsilon} \exists \alpha'$ s.t. $\alpha' \cdot F_b = 1$, $\alpha' \cdot \alpha' = \frac{1}{\epsilon} \sim \infty$

For F_c **inside** the cone: $\nexists \alpha$ s.t. $\alpha \perp F_c$

$\alpha \cdot F_c = 1 \implies \alpha \cdot \alpha < \infty$

(otherwise $\alpha' = \frac{\alpha}{\infty} \implies \alpha' \cdot F_c = 0$)

⊘

A good normalization vector

Condition for the normalization vector to guarantee **finite navigator function** : any vector inside the cone (not on the boundary of the cone).

In numerics : any vectors sufficiently away from the boundary of cone.

Vector of physical operators : in general, they are on the boundary of the cone.

(For example, if EMT is not on the boundary, we would have $F_{\Delta-\text{dim},\ell-2} = \sum_i x_i F_{\Delta_i,\ell_i}$ with positive x_i)

⌘

The Σ -Navigator

Maximize objective = $\alpha(F_{\Delta=0, \ell=0}(u, v))$, such that

$$\alpha(F_{\Delta, \ell=0}(u, v)) \geq 0 \text{ for } \Delta \geq \Delta_\epsilon$$

$$\alpha(F_{\Delta, \ell}(u, v)) \geq 0 \text{ for } \ell > 0, \Delta \geq \Delta_{\text{unitary}}$$

$$\alpha \cdot \left(\sum_i F_{\Delta_i, \ell_i}(u, v) \right) = 1 \text{ for a finite set of } \{(\Delta_i, \ell_i)\}$$

Guarantee $\alpha(F_{\Delta=0, \ell=0}(u, v))$ is bounded both above and below. (otherwise $\alpha \cdot \alpha = \infty$)



The GFF-Navigator

Take $\{\Delta_\sigma, \Delta_\epsilon\} = \{0.55, 1.6\}$

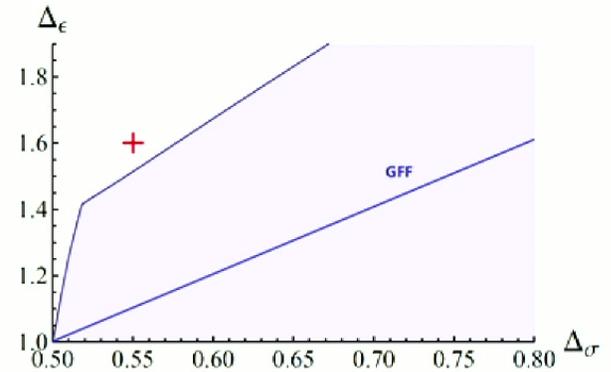
No solution for

$$F_{\Delta=0, \ell=0}(u, v) + \sum_{\Delta, \ell} p_{\Delta, \ell} F_{\Delta, \ell}(u, v) = 0 \text{ for } \Delta_\epsilon > 1.6$$



Solution exist for

$$F_{\Delta=0, \ell=0}(u, v) + p_0 (F_{\Delta=2\Delta_\sigma, \ell=0}(u, v)) + \sum_{\Delta, \ell} p_{\Delta, \ell} F_{\Delta, \ell}(u, v) = 0 \text{ for } \Delta_\epsilon > 1.6$$



Minimize p_0 : maximize objective = $\alpha(F_{\Delta=0, \ell=0}(u, v))$ with $\alpha \cdot (F_{\Delta=2\Delta_\sigma, \ell=0}^{\sigma\sigma\sigma\sigma}(u, v)) = -1$

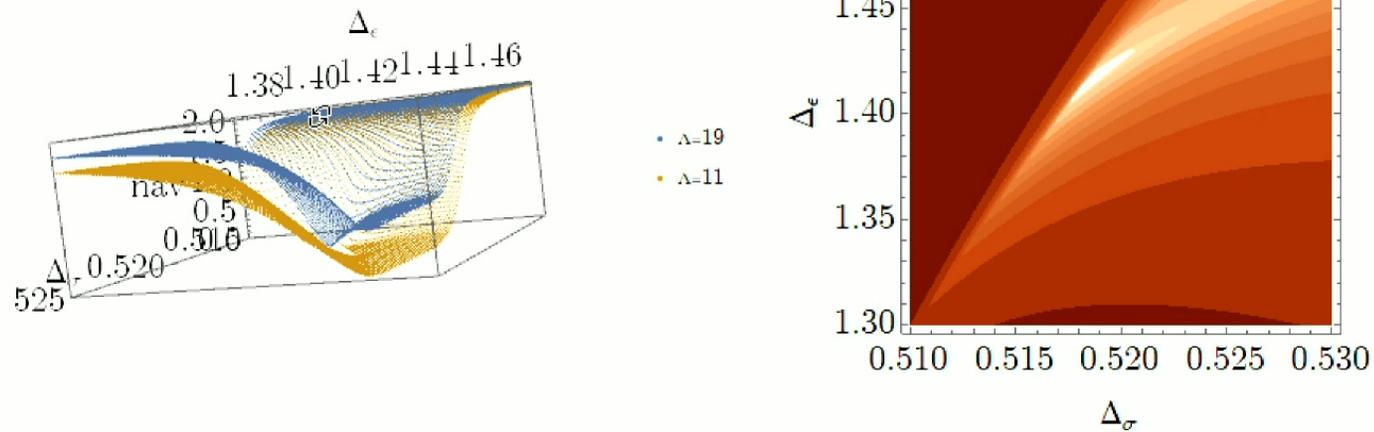
Guarantee $\alpha(F_{\Delta=2\Delta_\sigma, \ell=0}^{\sigma\sigma\sigma\sigma}(u, v))$ is bounded above by p_{GFF} .



Virtualization of navigator function

From Ising $\langle \sigma\sigma\sigma\sigma \rangle$, $\langle \epsilon\epsilon\epsilon\epsilon \rangle$, $\langle \sigma\sigma\epsilon\epsilon \rangle$. Assuming $\Delta_\sigma > 3$, $\Delta_\epsilon > 3$. $\theta_{\sigma\epsilon} = \lambda_{\epsilon\epsilon\epsilon} / \lambda_{\sigma\sigma\epsilon}$ unspecified

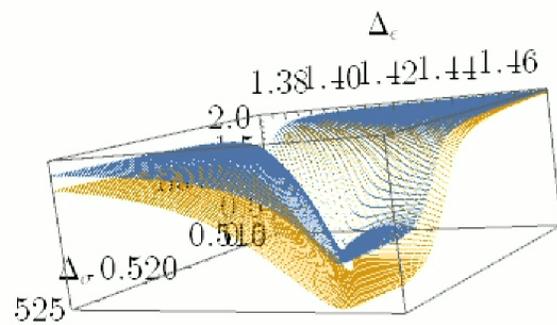
Unspecified $\theta_{\sigma\epsilon}$, $\Lambda=19$ vs $\Lambda=11$



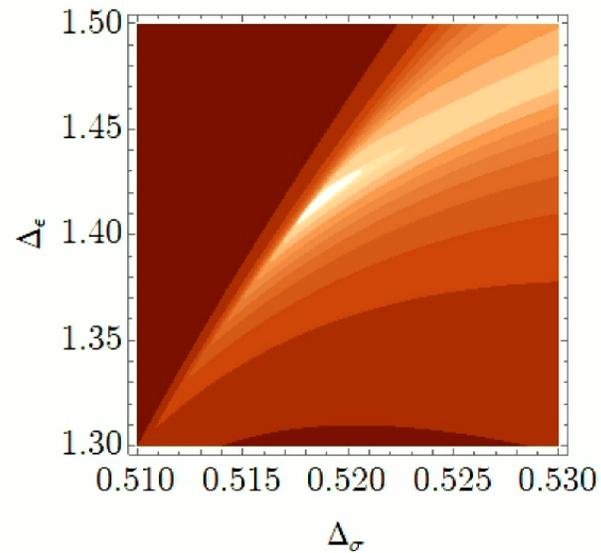
Virtualization of navigator function

From Ising $\langle \sigma\sigma\sigma\sigma \rangle$, $\langle \epsilon\epsilon\epsilon\epsilon \rangle$, $\langle \sigma\sigma\epsilon\epsilon \rangle$. Assuming $\Delta_\sigma > 3$, $\Delta_\epsilon > 3$. $\theta_{\sigma\epsilon} = \lambda_{\epsilon\epsilon\epsilon} / \lambda_{\sigma\sigma\epsilon}$ unspecified

Unspecified $\theta_{\sigma\epsilon}$, $\Lambda=19$ vs $\Lambda=11$

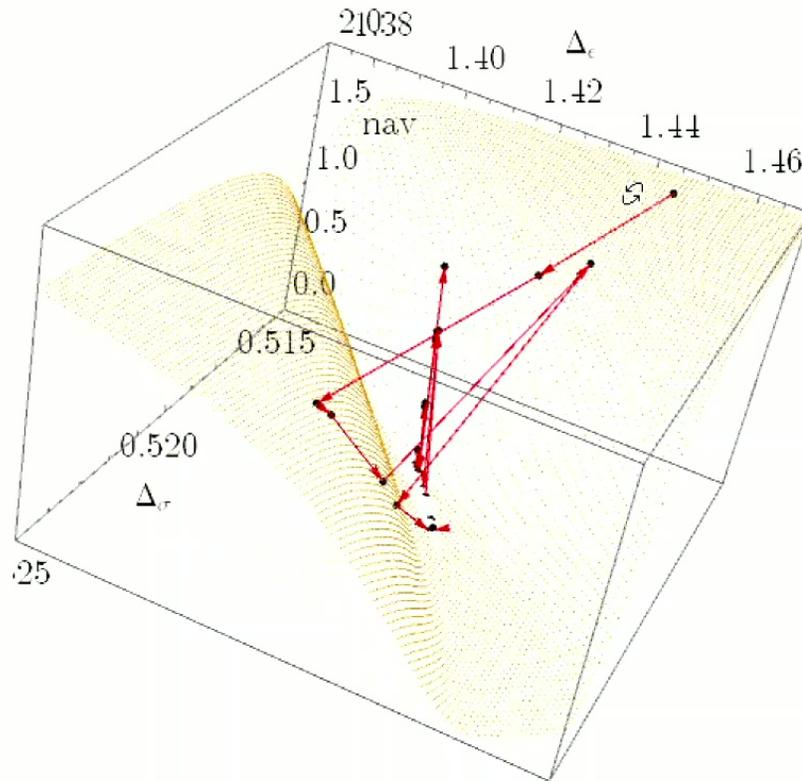


- $\Lambda=19$
- $\Lambda=11$



Minimize the navigator function

Unspecified θ_{rc} , $\Lambda=11$



Navigator function : gradient formula

$$L = b.y + c.x - x.B.y - \text{Tr}(Y(xA - X)) - \mu \log(\det(X))$$

In the limit $\mu \rightarrow 0$ and $b - x.B = 0$; $xA - X = 0$; $c - B.y - \text{Tr}(Y.A) = 0$

$$L = b.y = c.x \text{ (primal dual objective)}$$

Define navigator function $N(\vec{p}) = L[\sigma[p], \xi^*]$, where ξ^* is solution to $\frac{\partial L}{\partial \xi} = 0$

If $b \rightarrow b + db$, $c \rightarrow c + dc$, $B \rightarrow B + dB$: (1), $\xi \rightarrow \xi + d\xi$ (2), $L \rightarrow L + dL$.

$$\frac{\partial N}{\partial p} = \frac{\partial L}{\partial \sigma} \frac{\partial \sigma}{\partial p} + \frac{\partial L}{\partial \xi} \frac{\partial \xi}{\partial p}$$

$$\frac{\partial L}{\partial \xi} d\xi = (b - x.B) dy + dx.(c - B.y - \text{Tr}(Y.A)) - \text{Tr}(dY(xA - X)) + \text{Tr}((Y - \mu X^{-1}) dX) = 0 \text{ for } \xi^*$$

$$d_i N = d_i b.y + d_i c.x - x.d_i B.y$$



Navigator function : gradient formula

$$L = b.y + c.x - x.B.y - \text{Tr}(Y(xA - X)) - \mu \log(\det(X))$$

In the limit $\mu \rightarrow 0$ and $b - x.B = 0$; $xA - X = 0$; $c - B.y - \text{Tr}(Y.A) = 0$

$$L = b.y = c.x \text{ (primal dual objective)}$$

Define navigator function $N(\vec{p}) = L[\sigma[p], \xi^*]$, where ξ^* is solution to $\frac{\partial L}{\partial \xi} = 0$

If $b \rightarrow b + db$, $c \rightarrow c + dc$, $B \rightarrow B + dB$: (1), $\xi \rightarrow \xi + d\xi$ (2), $L \rightarrow L + dL$.

$$\frac{\partial N}{\partial p} = \frac{\partial L}{\partial \sigma} \frac{\partial \sigma}{\partial p} + \frac{\partial L}{\partial \xi} \frac{\partial \xi}{\partial p}$$

$$\frac{\partial L}{\partial \xi} d\xi = (b - x.B) dy + dx.(c - B.y - \text{Tr}(Y.A)) - \text{Tr}(dY(xA - X)) + \text{Tr}((Y - \mu X^{-1}) dX) = 0 \text{ for } \xi^*$$

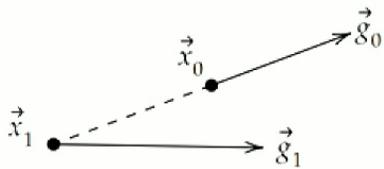
$$d_i N = d_i b.y + d_i c.x - x.d_i B.y$$



Quasi-Newton method

Assuming we can compute gradient cheaply, but not hessian

Minimize $f(\vec{x})$, given $g(\vec{x}) = \nabla f(\vec{x})$



Given gradient \vec{g}_0 at \vec{x}_0 and \vec{g}_1 at \vec{x}_1 :

We know certain component in Hessian (approximately)!

$$H \cdot (\vec{x}_1 - \vec{x}_0) = (\vec{g}_1 - \vec{g}_0) \text{ (secant equation)}$$

Basic idea : Take initial Hessian $H^{(0)} = \text{Id}$, minimize $f^{(\text{approx})}(\vec{x}) = f_0 + \vec{g} \cdot \vec{x} + \frac{1}{2} \vec{x} \cdot H^{(0)} \cdot \vec{x}$.

In the subsequent steps, update $H^{(k)} \rightarrow H^{(k+1)}$ such that $H^{(k+1)}(\vec{x}_{k+1} - \vec{x}_k) = (\vec{g}_{k+1} - \vec{g}_k)$

⋮

Navigator function : Hessian formula

Similarly $d_i d_j N(\vec{p}) = d_i b_j d_j y + d_i c_j d_j x - d_j x d_i B_j y - x d_i B_j d_j y + (d_i d_j b_j) y + (d_i d_j c_j) x - x (d_i d_j B_j) y$

$d_j x, d_j y$: assuming $\xi = (x, y, X, Y)$ are stationary. Slightly perturb $b' = b + db, c' = c + dc, B' = B + dB$. How ξ changes?

I

$$\text{tr}(A_* (Y + dY)) + B' (y + dy) = c'$$

$$X + dX = A_p(x_p + dx_p) \quad \Rightarrow \quad \begin{pmatrix} S & -B \\ B^T & 0 \end{pmatrix} \begin{pmatrix} dx \\ dy \end{pmatrix} = \begin{pmatrix} -d - \text{Tr}(A_* Z) \\ p \end{pmatrix}$$

$$B' (x + dx) = b'$$

$$XY + XdY + dXY = \mu I$$

$$P \equiv A_* x - X = 0; \quad p \equiv b' - B' x = db - (dB) x;$$

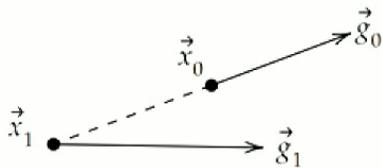
$$d \equiv c' - \text{Tr}(A_* Y) - B' y = dc - \text{Tr}(A_* Y) - dB y; \quad R \equiv \mu I - XY = 0$$

$$Z \equiv X^{-1}(P Y - R) = 0; \quad S_{ij} \equiv \text{Tr}(A_i X^{-1} A_j Y)$$

Quasi-Newton method

Assuming we can compute gradient cheaply, but not hessian

Minimize $f(\vec{x})$, given $g(\vec{x}) = \nabla f(\vec{x})$



Given gradient \vec{g}_0 at \vec{x}_0 and \vec{g}_1 at \vec{x}_1 :

We know certain component in Hessian (approximately)!

$$H \cdot (\vec{x}_1 - \vec{x}_0) = (\vec{g}_1 - \vec{g}_0) \quad (\text{secant equation})$$

Basic idea : Take initial Hessian $H^{(0)} = \text{Id}$, minimize $f^{(\text{approx})}(\vec{x}) = f_0 + \vec{g} \cdot \vec{x} + \frac{1}{2} \vec{x} \cdot H^{(0)} \cdot \vec{x}$.

In the subsequent steps, update $H^{(k)} \rightarrow H^{(k+1)}$ such that $H^{(k+1)}(\vec{x}_{k+1} - \vec{x}_k) = (\vec{g}_{k+1} - \vec{g}_k)$

⋮

Broyden–Fletcher–Goldfarb–Shanno formula

$$\vec{y}_k := \vec{g}_{k+1} - \vec{g}_k, \quad \vec{s}_k := \vec{x}_{k+1} - \vec{x}_k$$

Ansatz : $H_{k+1} = H_k + \alpha u u^T + \beta v v^T$ for some numbers α, β and vectors v, u

Secant equation : $H_{k+1} s_k = H_k s_k + \alpha u (u^T \cdot s_k) + \beta v (v^T \cdot s_k) = y_k$

Solution : $u = y_k, v = H_k s_k, \alpha = \frac{1}{y_k^T \cdot s_k}, \beta = -\frac{1}{s_k^T \cdot B_k \cdot s_k}$

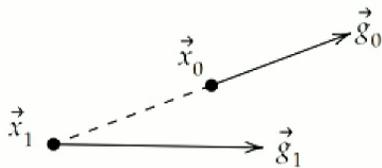
BFGS formula : $H_{k+1} = H_k + \frac{1}{y_k^T \cdot s_k} y_k y_k^T - \frac{1}{s_k^T \cdot B_k \cdot s_k} H_k s_k s_k^T H_k^T$

Y

Quasi-Newton method

Assuming we can compute gradient cheaply, but not hessian

Minimize $f(\vec{x})$, given $g(\vec{x}) = \nabla f(\vec{x})$



Given gradient \vec{g}_0 at \vec{x}_0 and \vec{g}_1 at \vec{x}_1 :

We know certain component in Hessian (approximately)!

$$H \cdot (\vec{x}_1 - \vec{x}_0) = (\vec{g}_1 - \vec{g}_0) \quad (\text{secant equation})$$

Basic idea : Take initial Hessian $H^{(0)} = \text{Id}$, minimize $f^{(\text{approx})}(\vec{x}) = f_0 + \vec{g} \cdot \vec{x} + \frac{1}{2} \vec{x} \cdot H^{(0)} \cdot \vec{x}$.

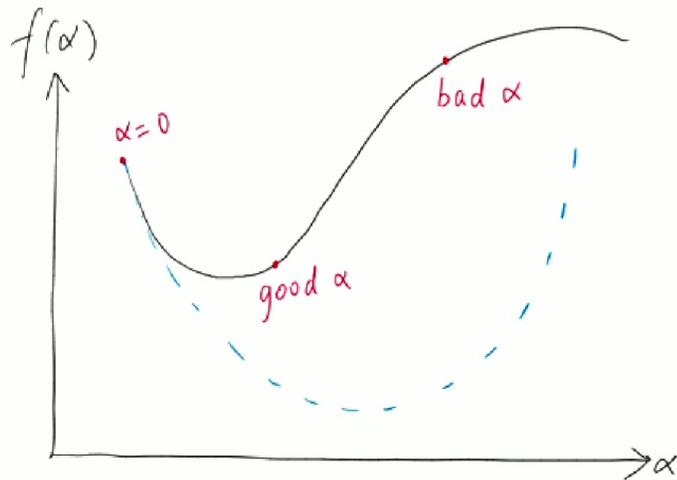
In the subsequent steps, update $H^{(k)} \rightarrow H^{(k+1)}$ such that $H^{(k+1)}(\vec{x}_{k+1} - \vec{x}_k) = (\vec{g}_{k+1} - \vec{g}_k)$

Line search

Given H_{k+1} , quadratic model : $f^{(\text{approx})}(\vec{x}) = f(\vec{x}_0) + \vec{g}_k \cdot (\vec{x} - \vec{x}_k) + \frac{1}{2} (\vec{x} - \vec{x}_k) \cdot H_{k+1} \cdot (\vec{x} - \vec{x}_k)$

Naive step : $H_{k+1} \cdot \Delta \vec{x} = -\vec{g}_k$

Line search : find a good α for $\vec{x}_{k+1} = \vec{x}_k + \alpha \Delta \vec{x}$, $\alpha > 0$



“good α ” :

- (1), $f(\vec{x} + \alpha \Delta \vec{x}) \leq f(\vec{x})$
 - (2), $|\partial_\alpha f(\vec{x} + \alpha \Delta \vec{x})| < |\partial_\alpha f(\vec{x})|$
- (Guarantee $H_k > 0$)

BFGS : find boundary of feasible region

Same as minimization, except we solve for the boundary in $N^{(\text{approx})}(\vec{p})$

Exercise : Given $N = \frac{1}{2} x.H.x + g.x + f$ and $N \leq 0$, show the boundary point in the direction \hat{e} is given by

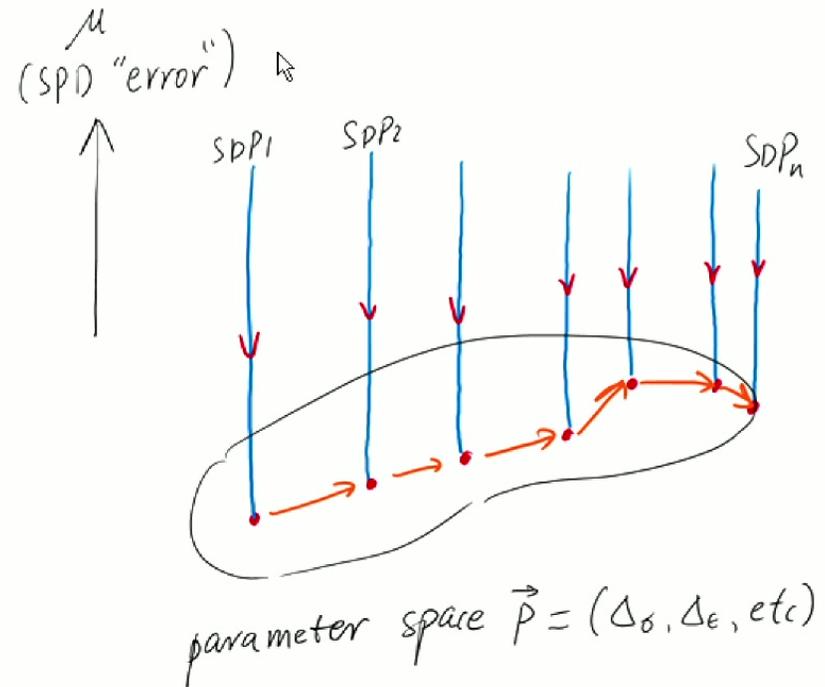
$$x = H^{-1}(-g + \lambda \hat{e}) \text{ with } \lambda = \sqrt{(g.H^{-1}.g - 2f) / \hat{e}.H^{-1}.\hat{e}}$$

Subtleties : in the line search, if the solver goes too far into the $N > 0$ region, we should be cautious and back track.

⌘

Summary

A section of SDPs over a parameter space. We solve SDPs one by one.



Maximize Δ_σ

