On the second Lyapunov exponent of some multidimensional continued fraction algorithms

Wolfgang Steiner

(joint work with Valérie Berthé and Jörg Thuswaldner)

arXiv:1910.09386

IRIF, CNRS, Université de Paris

Lorentz Center workshop, March 2020 (online)

Multidimensional continued fraction algorithms

Write $\mathbf{x} \in \Delta \subseteq [0,1]^d$ as $\mathbf{x} = \lim_{n\to\infty} \frac{\mathbf{p}^{(n)}}{q^{(n)}}$ with small $\|q^{(n)}\mathbf{x} - \mathbf{p}^{(n)}\|$.

We consider MCF algorithms

$$A:\ \Delta \to \mathrm{GL}(d+1,\mathbb{Z})$$

with associated transformations

$$T:\ \Delta \to \Delta, \quad \mathbf{x} \mapsto \pi\big(\iota(\mathbf{x})\, A(\mathbf{x})^{-1}\big)$$

$$\iota(x_1,\ldots,x_d) = (1,x_1,\ldots,x_d), \quad \pi(x_0,x_1,\ldots,x_d) = \big(\tfrac{x_1}{x_0},\ldots,\tfrac{x_d}{x_0}\big)$$

Regular continued fractions, $d=1$: $A(x) = \begin{pmatrix} \lfloor \frac{1}{x} \rfloor & 1 \\ 1 & 0 \end{pmatrix}$,

$$T(x) = \pi\left((1,x)\begin{pmatrix} 0 & 1 \\ 1 & -\lfloor \frac{1}{x} \rfloor \end{pmatrix}\right) = \pi\big(x, 1 - \lfloor \tfrac{1}{x} \rfloor x\big) = \frac{1}{x} - \left\lfloor \frac{1}{x} \right\rfloor$$

Jacobi–Perron algorithm (1868, 1907)

$$A_{\mathrm{JP}}(x_1, \ldots, x_d) = \begin{pmatrix} \lfloor \frac{1}{x_1} \rfloor & 1 & \lfloor \frac{x_2}{x_1} \rfloor & \cdots & \lfloor \frac{x_d}{x_1} \rfloor \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & \vdots & & \ddots & 1 \\ 1 & 0 & \cdots & \cdots & 0 \end{pmatrix}$$

$$A_{\mathrm{JP}}(x_1, \ldots, x_d)^{-1} = \begin{pmatrix} 0 & 0 & \cdots & 0 & 1 \\ 1 & -\lfloor \frac{x_2}{x_1} \rfloor & \cdots & -\lfloor \frac{x_d}{x_1} \rfloor & -\lfloor \frac{1}{x_1} \rfloor \\ 0 & 1 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & 1 & 0 \end{pmatrix}$$

$$\begin{aligned} T_{\mathrm{JP}} &: [0,1]^d \to [0,1]^d, \\ &(x_1, \ldots, x_d) \mapsto \left(\frac{x_2}{x_1} - \left\lfloor \frac{x_2}{x_1} \right\rfloor, \ldots, \frac{x_d}{x_1} - \left\lfloor \frac{x_d}{x_1} \right\rfloor, \frac{1}{x_1} - \left\lfloor \frac{1}{x_1} \right\rfloor \right) \end{aligned}$$

Convergents, linear cocycle

The convergents $\frac{\mathbf{p}_i^{(n)}}{q_i^{(n)}} = \frac{(p_{i,1}^{(n)},\ldots,p_{i,d}^{(n)})}{q_i^{(n)}}$ of $\mathbf{x}$ are given by the cocycle

$$A^{(n)}(\mathbf{x}) = A(T^{n-1}\mathbf{x}) \cdots A(T\mathbf{x})\, A(\mathbf{x}) = \begin{pmatrix} q_0^{(n)} & p_{0,1}^{(n)} & \cdots & p_{0,d}^{(n)} \\ q_1^{(n)} & p_{1,1}^{(n)} & \cdots & p_{1,d}^{(n)} \\ \vdots & \vdots & \ddots & \vdots \\ q_d^{(n)} & p_{d,1}^{(n)} & \cdots & p_{d,d}^{(n)} \end{pmatrix}$$

Weak convergence: $\lim_{n\to\infty} \frac{\mathbf{p}_i^{(n)}}{q_i^{(n)}} = \mathbf{x}$ for all $0 \le i \le d$

Strong convergence: $\lim_{n\to\infty} \|\mathbf{p}_i^{(n)} - q_i^{(n)}\mathbf{x}\| = 0$ for all $0 \le i \le d$

We assume that (Δ, T, μ) is ergodic for an invariant probability measure μ, the cocycle A is log-integrable, with Lyapunov exponents

$$\lambda_1(A) > \lambda_2(A) \ge \lambda_3(A) \ge \cdots \ge \lambda_{d+1}(A)$$

$\lambda_2(A) < 0 \Rightarrow$ almost everywhere exponential convergence

Approximation cocycle

$$D^{(n)}(\mathbf{x}) = \Pi\, A^{(n)}(\mathbf{x})\, H(\mathbf{x})$$
$$= \begin{pmatrix} p_{1,1}^{(n)} - q_1^{(n)} x_1 & \cdots & p_{1,d}^{(n)} - q_1^{(n)} x_d \\ \vdots & \ddots & \vdots \\ p_{d,1}^{(n)} - q_d^{(n)} x_1 & \cdots & p_{d,d}^{(n)} - q_d^{(n)} x_d \end{pmatrix} \in \mathbb{R}^{d\times d},$$

$$\Pi = \begin{pmatrix} 0 & 1 & 0 & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & 1 \end{pmatrix}, \quad H(x_1,\dots,x_d) = \begin{pmatrix} -x_1 & \cdots & -x_d \\ 1 & \cdots & 0 \\ 0 & \ddots & 0 \\ 0 & \cdots & 1 \end{pmatrix}.$$

Schweiger'00, Hardcastle–Khanin'02, ...: D is a cocycle of T

$$D^{(n)}(\mathbf{x}) = D(T^{n-1}\mathbf{x}) \cdots D(T\mathbf{x})\, D(\mathbf{x}), \quad \text{with } D(\mathbf{x}) = \Pi\, A(\mathbf{x})\, H(\mathbf{x})$$

$$\lim_{n\to\infty} \frac{1}{n} \log \|A^{(n)}(\mathbf{x})\,\mathbf{y}\| \le \lambda_2(A) \text{ for all } \mathbf{y} \in H(\mathbf{x})\mathbb{R}^d = \iota(\mathbf{x})^\perp \text{ (a.e. } \mathbf{x} \in \Delta)$$
$$\Rightarrow\ \lambda_2(A) = \lambda_1(D)$$

Determining $\lambda_2(A) = \lambda_1(D)$

$$|\det A(\mathbf{x})| = 1 \;\Rightarrow\; \sum_{k=1}^{d+1} \lambda_k(A) = 0$$

$$d = 2: \;\lambda_2(A) = -\lambda_1(A) - \lambda_3(A) = \lambda_1({}^tA^{-1}) - \lambda_1(A)$$

can be used to show $\lambda_2(A) < 0$ for Jacobi–Perron (Paley–Ursell'30), ...

$d \geq 2$: ergodicity, subadditivity $\Rightarrow$

$$\lambda_1(D) = \inf_{n\in\mathbb{N}} \frac{1}{n} \int_\Delta \log \|D^{(n)}(\mathbf{x})\| \,\mathrm{d}\mu(\mathbf{x})$$

gives upper bounds for $\lambda_2(A)$
Hardcastle'02: $\lambda_2(A) \leq -0.0053293$ for modified Jacobi–Perron (d-dimensional Gauss), $d = 3 \Rightarrow \lambda_2(A) < 0$ for Brun, $d = 3$

Lower bounds?

Adapt method of Pollicott'10 (approximation by periodic orbits) to matrices with negative entries?

Selmer algorithm (ordered)

$$\Delta = \{(x_1, \ldots, x_d) \in \mathbb{R}^d : 1 \geq x_1 \geq \cdots \geq x_d \geq 0,\ x_{d-1} + x_d \geq 1\}$$

$$T_{\mathrm{S}}(x_1, \ldots, x_d) = \pi\big(\mathrm{ord}(1-x_d, x_1, \ldots, x_d)\big) = \mathrm{ord}\left(\frac{x_2}{x_1}, \ldots, \frac{x_d}{x_1}, \frac{1-x_d}{x_1}\right)$$

$$A_{\mathrm{S}}(\mathbf{x}) = \begin{cases} S_a & \text{if } x_d > 1/2, \\ S_b & \text{if } x_d < 1/2, \end{cases}$$

$$S_a = \begin{pmatrix} 0 & 1 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & 1 & 0 \\ 1 & 0 & \cdots & 0 & 1 \\ 1 & 0 & \cdots & 0 & 0 \end{pmatrix}, \qquad S_b = \begin{pmatrix} 0 & 1 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & 1 & 0 \\ 1 & 0 & \cdots & 0 & 0 \\ 1 & 0 & \cdots & 0 & 1 \end{pmatrix}$$

invariant measure μ_{S} has density $\frac{c}{x_1 \cdots x_d}$

Selmer approximation cocycle

$$D_{\mathrm{S}}(\mathbf{x}) = \begin{pmatrix} 0 & 1 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & 1 & 0 \\ -x_1 & -x_2 & \cdots & -x_{d-1} & 1 - x_d \\ -x_1 & -x_2 & \cdots & -x_{d-1} & -x_d \end{pmatrix} \quad \text{if } x_d > 1/2$$

$$D_{\mathrm{S}}(\mathbf{x}) = \begin{pmatrix} 0 & 1 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & 1 & 0 \\ -x_1 & -x_2 & \cdots & -x_{d-1} & -x_d \\ -x_1 & -x_2 & \cdots & -x_{d-1} & 1 - x_d \end{pmatrix} \quad \text{if } x_d < 1/2$$

Selmer algorithm, $d = 2$

$$D_{\mathrm{S}}^{(2)}(\mathbf{x}) = \begin{cases} \begin{pmatrix} 1-x_1 & -x_2 \\ 1 & 0 \end{pmatrix} & \text{if } A_{\mathrm{S}}^{(2)}(\mathbf{x}) = S_a^2 = \begin{pmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 0 \end{pmatrix} \\ \begin{pmatrix} 1-x_1 & 1-x_2 \\ 1 & 0 \end{pmatrix} & \text{if } A_{\mathrm{S}}^{(2)}(\mathbf{x}) = S_a S_b = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{pmatrix} \\ \begin{pmatrix} 1 & 0 \\ 1-x_1 & -x_2 \end{pmatrix} & \text{if } A_{\mathrm{S}}^{(2)}(\mathbf{x}) = S_b S_a = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 0 \end{pmatrix} \\ \begin{pmatrix} 1 & 0 \\ 1-x_1 & 1-x_2 \end{pmatrix} & \text{if } A_{\mathrm{S}}^{(2)}(\mathbf{x}) = S_b^2 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix} \end{cases}$$

$$\|D_{\mathrm{S}}^{(2)}(\mathbf{x})\|_\infty = 1 \text{ for all } \mathbf{x} \in \Delta \ \Rightarrow\ \lambda_2(A_{\mathrm{S}}) \le 0$$

$$\|D_{\mathrm{S}}^{(4)}(\mathbf{x})\|_\infty < 1 \text{ for some } \mathbf{x} \in \Delta \ \Rightarrow\ \lambda_2(A_{\mathrm{S}}) < 0$$

$$\lambda_2(A_{\mathrm{S}}) \le \frac{1}{42} \int_\Delta \log \|D_{\mathrm{S}}^{(42)}(\mathbf{x})\| \mathrm{d}\mu(\mathbf{x}) < -0.050393$$

Selmer matrices

Theorem (Berthé–St-Thuswaldner)

Let $d = 2$ and $M \in \{S_a, S_b\}^n$ for some $n \geq 1$. The following are equivalent.

1. *M is a primitive matrix,*
2. *M is a Pisot matrix,*
3. *$M^2 \notin \{S_a S_b, S_b^2\}^n$.*

cf. results of Avila–Delecroix'15 (arXiv:1506.03692) for Brun matrices ($d = 2$) and Arnoux–Rauzy matrices ($d \geq 2$)

Selmer algorithm, computer simulations

Calculating $D_{\mathrm{S}}^{(n)}(\mathbf{x})$ and $A_{\mathrm{S}}^{(n)}(\mathbf{x})$ for $n = 2^{30}$ for randomly chosen points $\mathbf{x} \in \Delta$ gives the following estimates for $d \in \{2, 3, 4, 5\}$ (without guaranteed accuracy)

d	$\lambda_2(A_{\mathrm{S}})$	$1 - \frac{\lambda_2(A_{\mathrm{S}})}{\lambda_1(A_{\mathrm{S}})}$
2	-0.07072	1.3871
3	-0.02283	1.1444
4	$+0.00176$	0.9866
5	$+0.01594$	0.8577

Lagarias'93: uniform approximation exponent

$$\sup\left\{\delta : \left\|\mathbf{x} - \frac{\mathbf{p}_i^{(n)}}{q_i^{(n)}}\right\| = O\left(\frac{1}{\left(q_i^{(n)}\right)^{\delta}}\right), 0 \leq i \leq d\right\} = 1 - \frac{\lambda_2(A)}{\lambda_1(A)} \quad (\text{a.e. } \mathbf{x} \in \Delta)$$

Computer simulations

We calculate $D^{(n)}(\mathbf{x})$ for randomly chosen points $\mathbf{x} \in \Delta$ by a C program with double precision floating point arithmetic. We have to renormalize the matrices before they get too small or too large. For $k|n$, e.g., $k = 2^{10}$, $n = 2^{30}$, we have

$$\frac{1}{D^{(n)}_{1,1}(\mathbf{x})} D^{(n)}(\mathbf{x})$$
$$= \frac{D^{(n-k)}_{1,1}(\mathbf{x})}{D^{(n)}_{1,1}(\mathbf{x})} D^{(k)}(T^{n-k}\mathbf{x}) \cdots \frac{D^{(k)}_{1,1}(\mathbf{x})}{D^{(2k)}_{1,1}(\mathbf{x})} D^{(k)}(T^k\mathbf{x}) \frac{1}{D^{(k)}_{1,1}(\mathbf{x})} D^{(k)}(\mathbf{x})$$

$$\log\left|D^{(n)}_{1,1}(\mathbf{x})\right| = \log\frac{|D^{(n)}_{1,1}(\mathbf{x})|}{|D^{(n-k)}_{1,1}(\mathbf{x})|} + \cdots + \log\frac{|D^{(2k)}_{1,1}(\mathbf{x})|}{|D^{(k)}_{1,1}(\mathbf{x})|} + \log|D^{(k)}_{1,1}(\mathbf{x})|$$

($D^{(n)}_{1,1}(\mathbf{x})$ is the top left coefficient of $D^{(n)}(\mathbf{x})$)

Jacobi–Perron algorithm, computer simulations

$$A_{\mathrm{JP}}(x_1,\ldots,x_d) = \begin{pmatrix} \lfloor \frac{1}{x_1} \rfloor & 1 & \lfloor \frac{x_2}{x_1} \rfloor & \cdots & \lfloor \frac{x_d}{x_1} \rfloor \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & \vdots & & \ddots & 1 \\ 1 & 0 & \cdots & \cdots & 0 \end{pmatrix}$$

$$D_{\mathrm{JP}}(x_1,\ldots,x_d) = \begin{pmatrix} 0 & 1 & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & 1 \\ -x_1 & \cdots & -x_{d-1} & -x_d \end{pmatrix}$$

d	$\lambda_2(A_{\mathrm{JP}})$	$1-\frac{\lambda_2(A_{\mathrm{JP}})}{\lambda_1(A_{\mathrm{JP}})}$	d	$\lambda_2(A_{\mathrm{JP}})$	$1-\frac{\lambda_2(A_{\mathrm{JP}})}{\lambda_1(A_{\mathrm{JP}})}$
2	−0.44841	1.3735	7	−0.02819	1.0243
3	−0.22788	1.1922	8	−0.01470	1.0127
4	−0.13062	1.1114	9	−0.00505	1.0044
5	−0.07880	1.0676	10	+0.00217	0.9981
6	−0.04798	1.0413	11	+0.00776	0.9933

$$D_{d,j}^{(n)}(\mathbf{x}) = -\sum_{i=1}^{d} x_{d-i+1}^{(n)} D_{d,j}^{(n-i)}(\mathbf{x}) \quad \text{with } \left(x_1^{(n)},\ldots,x_d^{(n)}\right) = T^n(\mathbf{x})$$

Results for random recurrences or products of random matrices?

Nearest integer Jacobi–Perron algorithm

$\Delta = [-1/2, 1/2]^d$

$$A_{\mathrm{NIJP}}(x_1,\dots,x_d) = \begin{pmatrix} \lfloor \frac{1}{x_1} + \frac{1}{2} \rfloor & 1 & \lfloor \frac{x_2}{x_1} + \frac{1}{2} \rfloor & \cdots & \lfloor \frac{x_d}{x_1} + \frac{1}{2} \rfloor \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & \vdots & & \ddots & 1 \\ 1 & 0 & \cdots & \cdots & 0 \end{pmatrix}$$

(coefficients can be negative)

$$D_{\mathrm{NIJP}}(x_1,\dots,x_d) = \begin{pmatrix} 0 & 1 & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & 1 \\ -x_1 & \cdots & -x_{d-1} & -x_d \end{pmatrix}$$

d	$1 - \frac{\lambda_2(A)}{\lambda_1(A)}$	d	$1 - \frac{\lambda_2(A)}{\lambda_1(A)}$	d	$1 - \frac{\lambda_2(A)}{\lambda_1(A)}$	d	$1 - \frac{\lambda_2(A)}{\lambda_1(A)}$
2	1.40145	6	1.06898	10	1.01737	14	0.99924
3	1.22519	7	1.04944	11	1.01125	15	0.99657
4	1.14373	8	1.03551	12	1.00639		
5	1.09786	9	1.02521	13	1.00246		

Brun and modified Jacobi–Perron algorithms (ordered)

$$\Delta = \{(x_1,\dots,x_d) \in \mathbb{R}^d : 1 \ge x_1 \ge \cdots \ge x_d \ge 0\}$$

$$T_{\mathrm{B}}(x_1,\dots,x_d) = \pi\big(\mathrm{ord}(1-x_1, x_2, \dots, x_d)\big)$$

$$A_{\mathrm{B}}(\mathbf{x}) = B_k \quad \text{if } x_k > 1-x_1 > x_{k+1} \qquad (0 \le k \le d,\ x_0 = 1,\ x_{d+1} = 0)$$

$$B_0 = \begin{pmatrix} 1 & 0 & \cdots & \cdots & 0 \\ 1 & 1 & \ddots & & \vdots \\ 0 & 0 & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & 1 & 0 \\ 0 & 0 & \cdots & 0 & 1 \end{pmatrix}, \quad B_k = \left(\begin{array}{r|cccccccc} & 1 & 1 & 0 & \cdots & \cdots & \cdots & \cdots & 0 \\ \multirow{3}{*}{$k-1\{$} & 0 & 0 & 1 & \ddots & & & & \vdots \\ & \vdots & 0 & \ddots & \ddots & \ddots & & & \vdots \\ & 0 & \vdots & \ddots & \ddots & 1 & \ddots & & \vdots \\ & 1 & \vdots & & \ddots & 0 & 0 & \ddots & \vdots \\ \multirow{3}{*}{$d-k\{$} & 0 & \vdots & & & \ddots & 1 & \ddots & 0 \\ & \vdots & \vdots & & & & \ddots & \ddots & 0 \\ & 0 & 0 & \cdots & \cdots & \cdots & \cdots & 0 & 1 \end{array}\right)$$

$$A_{\mathrm{MJP}}(\mathbf{x}) = A_{\mathrm{B}}^{(n)}(\mathbf{x}) \quad \text{with } n = \min\{k \ge 1 : A_{\mathrm{B}}(T^{k-1}\mathbf{x}) \ne B_0\}$$

d	$\lambda_2(A_{\mathrm{B}})$	$1 - \frac{\lambda_2(A_{\mathrm{B}})}{\lambda_1(A_{\mathrm{B}})}$	d	$\lambda_2(A_{\mathrm{B}})$	$1 - \frac{\lambda_2(A_{\mathrm{B}})}{\lambda_1(A_{\mathrm{B}})}$
2	-0.11216	1.3683	7	-0.01210	1.0493
3	-0.07189	1.2203	8	-0.00647	1.0283
4	-0.04651	1.1504	9	-0.00218	1.0102
5	-0.03051	1.1065	10	$+0.00115$	0.9943
6	-0.01974	1.0746	11	$+0.00381$	0.9799

Garrity's triangle algorithm (ordered)

$$\Delta = \{(x_1, \ldots, x_d) \in \mathbb{R}^d : 1 \geq x_1 \geq \cdots \geq x_d \geq 0\}$$

$$T_{\mathrm{G}}(x_1, \ldots, x_d) = \begin{cases} \pi\big(\mathrm{ord}\big(1 - \sum_{j=1}^{k} x_j, x_1, \ldots, x_d\big)\big) \\ \qquad \text{if } 0 < 1 - \sum_{j=1}^{k} x_j < x_{k+1},\ 1 \leq k \leq d-2 \\ \frac{1}{x_1}\big(x_2, \ldots, x_d, 1 - \sum_{j=1}^{d-1} x_j - \ell x_d\big) \\ \qquad \text{if } 0 < 1 - \sum_{j=1}^{d-1} x_j - \ell x_d < x_d,\ \ell \geq 0 \end{cases}$$

$d = 2$:

$$A_{\mathrm{G}}(\mathbf{x}) = \begin{pmatrix} 1 & 1 & 0 \\ \lfloor \frac{1-x_1}{x_2} \rfloor & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix},\ D_{\mathrm{G}}(\mathbf{x}) = \begin{pmatrix} -\lfloor \frac{1-x_1}{x_2} \rfloor x_1 & 1 - \lfloor \frac{1-x_1}{x_2} \rfloor x_2 \\ -x_1 & -x_2 \end{pmatrix}$$

Problem: $\lfloor \frac{1-x_1}{x_2} \rfloor x_1$ can be large

d	$\lambda_2(A_G)$	$1 - \frac{\lambda_2(A_G)}{\lambda_1(A_G)}$	d	$\lambda_2(A_G)$	$1 - \frac{\lambda_2(A_G)}{\lambda_1(A_G)}$
2	+0.34434	0.6859	7	−0.00644	1.0225
3	+0.37673	0.5798	8	−0.00768	1.0304
4	+0.25232	0.6286	9	−0.00435	1.0189
5	+0.10677	0.7778	10	−0.00074	1.0035
6	+0.01859	0.9468	11	+0.00237	0.9880

Intermediate algorithm between Arnoux–Rauzy and Brun

$$\Delta = \{(x_1, \ldots, x_d) \in \mathbb{R}^d : 1 \geq x_1 \geq \cdots \geq x_d \geq 0\}$$

$$T_{\mathrm{BST}}(x_1, \ldots, x_d) = \pi\big(\mathrm{ord}\big(1 - \textstyle\sum_{j=1}^{k} x_j, x_1, \ldots, x_d\big)\big)$$
$$\text{if } 0 < 1 - \textstyle\sum_{j=1}^{k} x_j < x_{k+1} \quad (1 \leq k \leq d,\ x_{d+1} = 1)$$

d	$\lambda_2(A_{\mathrm{BST}})$	$1 - \frac{\lambda_2(A_{\mathrm{BST}})}{\lambda_1(A_{\mathrm{BST}})}$	d	$\lambda_2(A_{\mathrm{BST}})$	$1 - \frac{\lambda_2(A_{\mathrm{BST}})}{\lambda_1(A_{\mathrm{BST}})}$
2	-0.13648	1.3606	7	-0.02033	1.0729
3	-0.10803	1.2430	8	-0.01175	1.0468
4	-0.07540	1.1817	9	-0.00563	1.0246
5	-0.05035	1.1388	10	-0.00114	1.0054
6	-0.03263	1.1034	11	$+0.00224$	0.9886

Comparison between (simulations of) algorithms

uniform approximation coefficients $1 - \frac{\lambda_2(A)}{\lambda_1(A)} \leq 1 + \frac{1}{d}$

d	Selmer	Brun	JP	BST	Garrity	NIJP
2	**1.3871**	1.3683	1.3735	1.3606	0.6859	**1.40145**
3	1.1444	1.2203	1.1922	**1.2430**	0.5798	1.22519
4	0.9866	1.1504	1.1114	**1.1817**	0.6286	1.14373
5	0.8577	1.1065	1.0676	**1.1388**	0.7778	1.09786
6	0.7442	1.0746	1.0413	**1.1034**	0.9468	1.06898
7	0.6437	1.0493	1.0243	**1.0729**	1.0225	1.04944
8	0.5561	1.0283	1.0127	**1.0468**	1.0304	1.03551
9	0.4810	1.0102	1.0044	**1.0246**	1.0189	**1.02521**
10	0.4173	0.9943	0.9981	**1.0054**	1.0035	**1.01737**
11	0.3636	0.9799	**0.9933**	0.9886	0.9880	**1.01125**
12						**1.00639**
13						**1.00246**
14						**0.99924**

bold face: best algorithm or best positive algorithm in dimension d (among those that we considered)