University Library

MINERVA ACCESS

A gateway to Melbourne's research publications

Minerva Access is the Institutional Repository of The University of Melbourne

Author/s:
Hautphenne, S;Massei, S

Title:
A LOW-RANK TECHNIQUE FOR COMPUTING THE QUASI-STATIONARY DISTRIBUTION OF SUBCRITICAL GALTON-WATSON PROCESSES

Date:
2020-01-01

Citation:
Hautphenne, S. & Massei, S. (2020). A LOW-RANK TECHNIQUE FOR COMPUTING THE QUASI-STATIONARY DISTRIBUTION OF SUBCRITICAL GALTON-WATSON PROCESSES. SIAM JOURNAL ON MATRIX ANALYSIS AND APPLICATIONS, 41 (1), pp.29-57. https://doi.org/10.1137/19M1241647.

Persistent Link:
https://hdl.handle.net/11343/249544

A LOW-RANK TECHNIQUE FOR COMPUTING THE QUASI-STATIONARY DISTRIBUTION OF SUBCRITICAL GALTON-WATSON PROCESSES

SOPHIE HAUTPHENNE* AND STEFANO MASSEI†

Abstract. We present a new algorithm for computing the quasi-stationary distribution of subcritical Galton–Watson branching processes. This algorithm is based on a particular discretization of a well-known functional equation that characterizes the quasi-stationary distribution of these processes. We provide a theoretical analysis of the approximate low-rank structure that stems from this discretization, and we extend the procedure to multitype branching processes. We use numerical examples to demonstrate that our algorithm is both more accurate and more efficient than other approaches.

Keywords: Galton-Watson processes, quasi-stationary distribution, Yaglom limit, low-rank matrices, low-rank approximation.

AMS subject classifications: 15B05, 65C40.

1. Introduction. Many biological populations are doomed to extinction due to low reproduction rates, the presence of predators, competition for limited resources, lack of suitable habitat, or other factors. However, before extinction eventually occurs the population size may fluctuate around some positive values for a long period of time. We are then interested in the long-term distribution of the size of the population; roughly speaking, this amounts to studying the *quasi-stationary distribution*. We illustrate this in Figure 1 for a stochastic process with logistic growth.

In this paper, we focus on the class of stochastic processes called the *Galton–Watson* (GW) *branching processes*. These processes are particular discrete-time Markov chains used to model randomly evolving populations in which individuals reproduce independently of each other. They have been successfully illuminating real-world problems that arise in diverse areas, such as biology, chemistry, particle physics, and computer science. Classical reference books on branching processes include Harris [19], Athreya and Ney [2], and Haccou, Jagers and Vatutin [17].

Quasi-stationary distributions of stochastic processes have been a focus of attention for many years. Their study started with the work of Yaglom in the late 1940's, who was the first to establish the existence of a particular quasi-stationary distribution, called the *Yaglom limit*, in the (subcritical) GW branching process [36]. The computation of quasi-stationary distributions of general Markov chains can generally be tackled from different angles. The most common approaches involve using simulation techniques, or solving for the left eigenvector of the transition matrix restricted to the positive states; for more details we refer to the excellent surveys of Méléard and Villemonais [26] and van Doorn and Pollet [34], and references therein. These methods have clear limitations, especially when the state space of the process is unbounded. Our motivation for considering subcritical GW processes here stems from the fact that their Yaglom limit has a specific characterisation: if $P(z) := \sum_{j\geqslant 0} p_j z^j$ denotes the (known) probability generating function of the offspring distribution, $m := P'(1) < 1$ its mean, and $G(z) := \sum_{j\geqslant 1} g_j z^j$ the unknown probability generating function of the Yaglom limit $(g_j)_{j\geqslant 1}$, then $G(z)$ solves the modified Schröder functional equation

$$G(0) = 0, \quad G(P(z)) = mG(z) + 1 - m, \qquad z \in [0,1]. \tag{1}$$

*University of Melbourne, Australia and EPF Lausanne, Switzerland sophiemh@unimelb.edu.au

†EPF Lausanne, Switzerland, stefano.massei@epfl.ch. The work of Stefano Massei has been supported by the SNSF research project *Fast algorithms from low-rank updates*, grant number: 200020_178806.

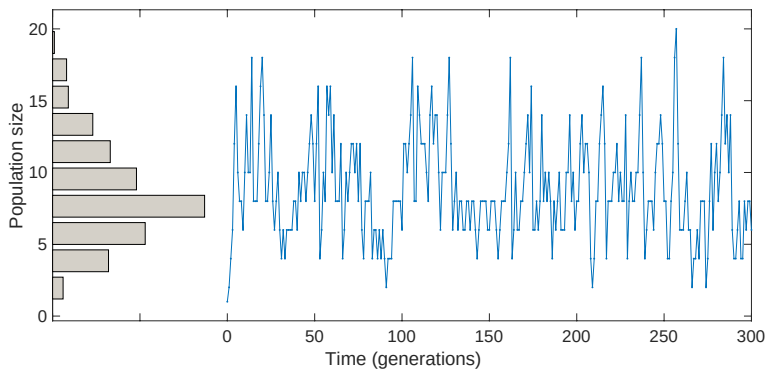


FIG. 1. *A trajectory of a discrete-time population-size-dependent branching process, starting with a single individual. An empirical estimate of the quasi-stationary distribution is superimposed.*

There exists a unique probability generating function $G(z)$ that solves (1) [20, Theorem 1]. To the best of our knowledge, no attention has been paid to the numerical solution of this equation.

In this paper, we propose an efficient algorithmic method to compute the coefficients g_j of $G(z)$ when the latter is analytic on a neighborhood of the unit disc. Our approach consists in using Cauchy's integral formula to rewrite (1) as

$$\int_\Gamma \frac{G(t)}{t - P(z)} dt = mG(z) + 1 - m,$$

where Γ is a circle of appropriate radius r. Discretizing the integral on the left-hand side by means of the trapezoidal rule leads to

$$\sum_{j=1}^{n} G(r\xi_j) \cdot \frac{r\xi_j}{n} \cdot (r\xi_j - P(z))^{-1} = mG(z) + 1 - m,$$

where $\{r\xi_j\}_{1 \leqslant j \leqslant n}$ are the scaled nth roots of unity. Evaluating the last equation in $z = r\xi_j$ for $1 \leqslant j \leqslant n$ leads to a linear system where the unknowns are the (approximate) quantities $G(r\xi_j)$. We then retrieve an interpolating polynomial of degree n for $G(z)$ by applying the Fast Fourier Transform (FFT). This particular discretization method provides highly structured data and allows us to deal with a large number n of integration nodes. In the second part of the paper we perform a theoretical analysis of the low-rank structure arising from the discretization scheme, and we discuss how to modify the algorithm in order to benefit from this property. The key idea is to approximate the coefficient matrix with a sum $mI - UV^*$ where U, V are tall and skinny matrices retrieved by means of the *Adaptive Cross Approximation* algorithm [9, Algorithm 1]. This yields a procedure with almost linear complexity of computational time and storage. In the final part of the paper we extend the technique to multitype branching processes. Here, the computational cost and the memory consumption suffer from the curse of dimensionality. The presence of the low-rank structure enables to partially mitigate this effect and to obtain satisfactory results in the two-dimensional case.

The paper is organized as follows; in Section 1.1 we recall some background notions. We dedicate Section 2 to the study of the regularity of $G(z)$ and the consequent decay of the coefficients g_j as $j \to \infty$. In particular, we provide results on the interplay between the regularity of $P(z)$ and $G(z)$.

In Section 3 we describe the numerical procedures for the computation of the coefficients g_j. In Section 3.3 we introduce a new method based on solving a discretized version of Equation (1), and we compare it with other techniques in Section 3.4. In Section 3.5 we perform an analysis of the rank structure stemming from the discretization process, and we provide a large scale version of the new algorithm in Section 3.6. In Section 4 we extend the procedure to multitype GW branching processes. Finally, we gather the proofs of the technical results of Section 2 in an appendix.

Throughout the paper, for $z_0 \in \mathbb{C}$ and $r > 0$, we let $\mathcal{B}(z_0, r) := \{z \in \mathbb{C} : \ |z - z_0| < r\}$, $\mathcal{D}(z_0, r) := \{z \in \mathbb{C} : \ |z - z_0| \leqslant r\}$ and $\mathcal{S}^1 := \{z \in \mathbb{C} : \ |z| = 1\}$; we let ∂ indicate the border of a set with respect to the Euclidean topology, e.g., $\mathcal{S}^1 = \partial\mathcal{B}(0, 1) = \partial\mathcal{D}(0, 1)$; for $h \geqslant 1$, we denote by $G^{(h)}(z_0)$ the h-th derivative of the function $G(z)$ evaluated at $z = z_0$; finally, we let $\mathbf{1}$ and $\mathbf{0}$ denote the column vectors of 1's and 0's, respectively, whose length will be determined by the context.

1.1. Background. A *Galton–Watson* (GW) *branching process* is a particular discrete-time Markov chain $\{Z_n\}_{n\geqslant 0}$ that takes its values in $\mathbb{N} := \{0, 1, 2, \ldots\}$, where 0 is an absorbing state. It describes the evolution of a population in which each individual lives for one unit of time (generation), at the end of which it gives birth to a random (nonnegative integer) number θ of children, independently of the rest of the population. The distribution of θ is called the *offspring distribution*, and is denoted by $\boldsymbol{p} := (p_j)_{j\in\mathbb{N}}$, where $p_j := \mathbb{P}(\theta = j) \geqslant 0$, $\sum_{j\geqslant 0} p_j = 1$; more explicitly, p_j represents the probability for an individual to have j offspring at the end of her/his lifetime. For each $n \geqslant 0$, the random variable Z_n represents the number of individuals in the population at generation n. We define the *probability generating function* of the offspring distribution as

$$P(z) := \sum_{j\geqslant 0} p_j z^j, \quad z \in [0, 1].$$

The mean offspring number per individual in the GW process is given by

$$m := \mathbb{E}(\theta) = \frac{dP(z)}{dz}\Big|_{z=1} = \sum_{j\geqslant 1} j p_j.$$

Given the initial population size Z_0, the size Z_n of the population at generation $n \geqslant 1$ evolves according to the recurrence formula

$$Z_n = \sum_{i=1}^{Z_{n-1}} \theta_i^{(n)},$$

where $\{\theta_i^{(n)}\}_{i,n}$ is a family of independent random variables taking values in $\mathbb{N}$ with the same distribution as θ, and $Z_n := 0$ if $Z_{n-1} = 0$ (hence 0 is absorbing). Indeed, the population size at generation n, Z_n, is made of the children of the Z_{n-1} individuals present in the population at generation $n - 1$ (the random variable $\theta_i^{(n)}$ represents the number of children of the ith individual present in the population at generation $n - 1$). Before extinction, the GW process takes its values in the space $\mathbb{N}_0 := \mathbb{N} \setminus \{0\}$.

In the sequel, we assume $0 < p_0 + p_1 < 1$. For any initial state $\ell \in \mathbb{N}_0$ and any initial probability distribution $\boldsymbol{\mu} := (\mu_\ell)_{\ell\in\mathbb{N}_0}$, we let $\mathbb{P}_\ell(\cdot) := \mathbb{P}(\cdot \,|\, Z_0 = \ell)$, and $\mathbb{P}_\mu(\cdot) := \sum_{\ell\geqslant 1} \mu_\ell \mathbb{P}_\ell(\cdot)$ be the probability measure conditional on the distribution of Z_0.

We distinguish between three cases:

- the *subcritical* case $m < 1$: the population becomes extinct with probability one, that is, for any initial probability distribution $\boldsymbol{\mu}$, $\mathbb{P}_\mu(\exists n < \infty : Z_n = 0) = 1$; the expected extinction time is finite.
- the *critical* case $m = 1$: the population becomes extinct with probability one, and the expected extinction time is infinite.
- the *supercritical* case $m > 1$: the population has a positive probability of surviving, and therefore the expected extinction time is infinite.

We refer to [2, 17, 19] for basic properties of GW processes.

We say that $\{Z_n\}$ has a *Yaglom limit* if there exists a probability distribution $\boldsymbol{g} := (g_j)_{j\in\mathbb{N}_0}$ (that is with $g_j \in [0,1]$ for all $j \geqslant 1$, and $\sum_{j\geqslant 1} g_j = 1$) such that, *for any initial population size* $\ell \in \mathbb{N}_0$ and any state $j \in \mathbb{N}_0$,

$$\lim_{n\to\infty} \mathbb{P}_\ell(Z_n = j \,|\, Z_n > 0) = g_j;$$

in other words, $\boldsymbol{g}$ is the *asymptotic* distribution of the population size at generation n, conditional on non-extinction by generation n. When it exists, the Yaglom limit $\boldsymbol{g}$ is a *quasi-stationary distribution*, that is, for all $n \geqslant 0$ and for any state $j \in \mathbb{N}_0$,

$$\mathbb{P}_g(Z_n = j \,|\, Z_n > 0) = g_j;$$

in other words, if the process starts with a number of individuals distributed according to $\boldsymbol{g}$, then the distribution of the population size at any subsequent generation $n \geqslant 1$ remains $\boldsymbol{g}$.

There is no quasi-stationary distribution in the critical and the supercritical case because conditioning on the event $\{Z_n > 0\}$ results in the process growing without bounds as $n \to \infty$. However, in the subcritical case there is a unique Yaglom limit; the next theorem states this formally, and we refer to [26, Theorem 6] for a proof.

THEOREM 1.1 (Yaglom [36]). *Let $\{Z_n\}$ be a GW process with offspring generating function $P(z)$ and mean offspring $m < 1$. There exists a unique probability distribution $\boldsymbol{g} = (g_j)_{j\in\mathbb{N}_0}$ such that, for any initial probability distribution $\boldsymbol{\mu}$ with finite mean on $\mathbb{N}_0$ (that is, $\sum_{j\geqslant 1} j\mu_j < \infty$), $\boldsymbol{g}$ satisfies*

$$\lim_{n\to\infty} \mathbb{P}_\mu(Z_n = j \,|\, Z_n > 0) = g_j. \tag{2}$$

The distribution $\boldsymbol{g}$ is a Yaglom limit for $\{Z_n\}$, and its generating function $G(z) := \sum_{j\geqslant 1} g_j z^j$, $z \in [0,1]$, satisfies Equation (1) on $[0,1]$.

Remark 1.2. Since the series that define $P(z)$ and $G(z)$ converge absolutely for all z in the unit disc $\mathcal{D}(0,1)$ then, by continuity, (1) holds $\forall z \in \mathcal{D}(0,1)$.

Remark 1.3. There exist quasi-stationary distributions which are not a Yaglom limit. In particular, for the subcritical GW process, there exists an infinite number of quasi-stationary distributions which are obtained as the limit in (2) for some initial probability distributions $\boldsymbol{\mu}$ *with infinite mean.* We refer to [26, Theorem 6] for more detail.

In the remainder of the paper we assume that the GW process is subcritical ($m < 1$) and we use the term "the quasi-stationary distribution of the GW process" when referring to its Yaglom limit $\boldsymbol{g}$. We refer the reader to [2, Section I. 8] and [26] for more details about quasi-stationary distributions of GW processes.

It is worth mentioning another characterization for the quasi-stationary distribution of a GW process. Let Q denote the truncated probability transition matrix of the process $\{Z_n\}$ corresponding

to the (transient) positive integer states. Then the quasi-stationary distribution satisfies

$$\boldsymbol{g}\, Q = m\, \boldsymbol{g}, \tag{3}$$

where $\boldsymbol{g}$ corresponds to the normalized Perron-Frobenius left eigenvector associated with the Perron-Frobenius eigenvalue m. The solution of (3) is unique up to a multiplicative constant. This characterization is not specific to the GW process and holds for any absorbing Markov chain. Note that in the case of the GW process, the matrix Q is dense and semi-infinite, hence solving (3) is a non trivial task. A classical approach to approximate the solution of (3) consists in considering the sequence of $N \times N$ northwest corner truncations of Q and computing their respective left Perron-Frobenius eigenvectors [30, Chapter 6]. In our context, this would require to solve eigenvector problems for non sparse and possibly large (as N grows) matrices. In this paper we introduce a new procedure that exploits more naturally the properties of the GW process.

We end this section by defining the *linear fractional branching processes*, which form a special class of GW processes amenable to explicit computation. In these processes, the offspring distribution is modified geometric, that is,

$$p_j = (1-p_0)(1-p)p^{j-1}, \quad j \geqslant 1,$$

fully characterized by just two parameters: $p_0 \in [0,1)$, and $p \in [0,1)$. Note here that $p_j \geqslant p_{j+1}$ for all $j \geqslant 1$. The mean offspring is given by $m = 1 - p_0/1 - p$, therefore the process is subcritical ($m < 1$) if and only if $p_0 > p$. The corresponding progeny generating function is given by

$$P(z) = p_0 + (1-p_0)\frac{(1-p)z}{1-pz}, \quad z \in [0, p^{-1}).$$

It is not difficult to verify that the quasi-stationary distribution of a linear-fractional GW process is geometric with parameter p/p_0, that is,

$$g_j = \left(1 - \frac{p}{p_0}\right)\left(\frac{p}{p_0}\right)^{j-1}, \; j \geqslant 1, \quad \text{and} \quad G(z) = \left(1 - \frac{p}{p_0}\right)\frac{z}{1 - \frac{p}{p_0}z}. \tag{4}$$

We shall use the linear fractional branching process in Section 3.4 as a benchmark tool to evaluate the quality of our numerical approximation methods for the computation of the quasi-stationary distribution.

2. Properties of $G(z)$. In this section we study the asymptotic behavior of the coefficients g_j, or in other words, the tail behavior of the quasi-stationary distribution $\boldsymbol{g}$. For computational purposes, we are interested in understanding the decay properties of these coefficients in order to ensure that a limited number of them is sufficient to describe $G(z)$ with arbitrary accuracy. For example, the existence of the h-th derivative $G^{(h)}(1)$ of $G(z)$ at $z = 1$, which corresponds to the h-th factorial moment of $\boldsymbol{g}$, provides an algebraic decay of (at least) order h, because $G^{(h)}(1) = \sum_{j \geqslant h} \frac{j!}{(j-h)!} g_j \approx \sum_{j \geqslant h} j^h g_j$. Exponential decay is directly linked to the radius of convergence of $G(z) = \sum_{j \geq 0} g_j z^j$ and, consequently, to the domain of analyticity of $G(z)$. Indeed, given $R > 0$, it is well known that a formal power series $\sum_{j \geq 0} g_j z^j$ defines an analytic function $G(z)$ on $\mathcal{B}(0,R)$ if and only if, for all $r \in (0,R)$ and $j \geqslant 0$, $|g_j| \leqslant \max_{|z|=r} |G(z)| \cdot r^{-j}$ [15, Proposition IV.1]. Since in our case the power series has real non-negative coefficients, $G(z)$ is analytic on $\mathcal{B}(0,R)$ if and only if

$$g_j \leqslant G(r) \cdot r^{-j} \qquad \forall r \in (0,R), j \geqslant 0. \tag{5}$$

From a computational perspective, we would like $G(z)$ to be analytic on a disc with radius bigger than 1. This would allow us to choose $r > 1$ in (5), ensuring that at most $\lceil \log(u^{-1}G(r))/\log(r) \rceil$ coefficients g_j are above the machine precision u. This property is equivalent to having $G(z)$ analytic at $z = 1$. The proof of the following proposition is provided in the appendix.

PROPOSITION 2.1. *Let $P(z)$ be analytic on $\mathcal{B}(0, r_P)$ with $r_P > 1$. Then, $G(z)$ is analytic at $z = 1$ if and only if there exists $r_G > 1$ such that $G(z)$ is analytic on $\mathcal{B}(0, r_G)$.*

In what follows we study the interplay between the regularity of the offspring distribution and that of the quasi-stationary distribution.

2.1. Derivatives of $G(z)$ at $z = 1$. We start by looking at the existence of the derivatives of $G(z)$ at $z = 1$. The next theorem gives a necessary and sufficient condition on the offspring distribution for the mean of the quasi-stationary distribution to exist, that is, for $G^{(1)}(1) < \infty$.

THEOREM 2.2 (Heathcote *et al.* [20]).

$$G^{(1)}(1) < \infty \quad \Leftrightarrow \quad \sum_{j=2}^{\infty} j \log(j) \cdot p_j < \infty.$$

Higher order moments of the quasi-stationary distribution are studied in [3], where the author shows that $G^{(h)}(1)$ is finite if and only if $P^{(h)}(1)$ is finite for $1 < h \in \mathbb{N}$. In the appendix we report a simpler and shorter proof of this fact, that relies on algebraic arguments only.

2.2. Domain of analyticity of $G(z)$. Motivated by the preceding results, we wonder if assuming the analyticity of $P(z)$ on an open disc of radius bigger than 1 is enough to ensure the same property for $G(z)$. The answer to this question is affirmative, as we formalize in the next theorem.

THEOREM 2.3. *If $P(z)$ has radius of convergence $r_P > 1$, then $G(z)$ has radius of convergence $r_G > 1$.*

In the appendix, we prove Theorem 2.3 by directly showing the convergence of the Taylor series of the probability generating function $G(z)$ centred at $z = 1$. This property can also be obtained by combining Proposition 2.1 with the *linearization theorem* of Koenigs [24]; see also [11, Chapter II].

The quantity r_G is related to the basin of attraction of the attractor 1 for the fixed point iteration $z_{k+1} = P(z_k)$ (see Section 3.2). The latter in turn is linked to the rightmost real solution of $z = P(z)$. Observe that the offspring generating function $P(z)$ and all its derivatives are real and positive on the interval $[0, r_P)$, where r_P denotes the radius of convergence of $P(z)$. In particular, the equation $z = P(z)$ can have either one or two solutions on the positive real semi-axis: 1 and possibly $\widehat{z} > 1$. We have that $z = P(z)$ only for $z = 1$ in the case $P(z) \equiv 1 - m + mz$ (the only degree 1 polynomial that satisfies the assumptions on $P(z)$), and in the case $P(r_P) < r_P$ (see the right part of Figure 2). Hence, we define

$$\psi_P := \begin{cases} \infty & \text{if } P(z) \equiv 1 - m + mz \\ r_P & \text{if } P(r_P) < r_P \\ \widehat{z} > 1 : \widehat{z} = P(\widehat{z}) & \text{otherwise,} \end{cases} \tag{6}$$

so that $P(z) < z \ \forall z \in (1, \psi_P)$.

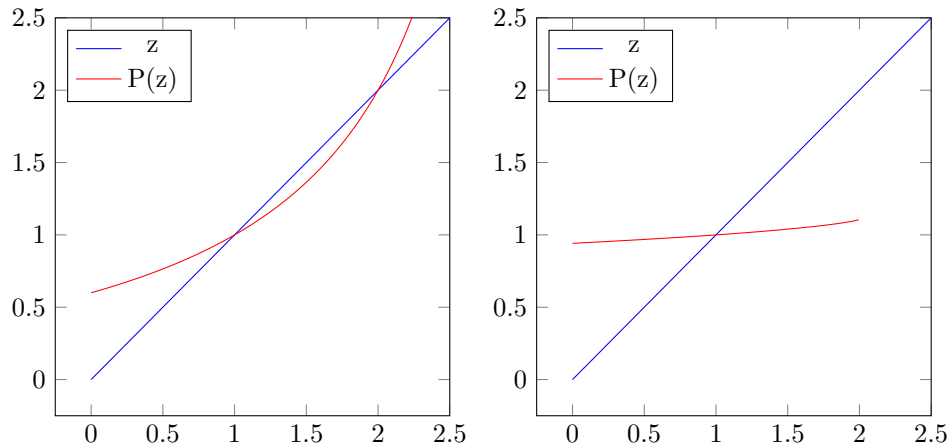


FIG. 2. *Intersections of $P(z)$ with the bisector of the first quadrant. On the left, $P(z) := 0.6 + 0.4\,(0.7z)(1-0.3z)^{-1}$. On the right, $P(z) := 1 + 0.1\,(\widetilde{P}(z) - \widetilde{P}(1))$ with $\widetilde{P}(z) := \sum_{j\geqslant 1} \frac{z^j}{2^j j^2}$. In both cases $\psi_P = 2$.*

COROLLARY 2.4. *Assume that $P(z)$ has radius of convergence $r_P > 1$. Then the following statements hold:*

(i) $r_G \geqslant \psi_P$ where ψ_P is defined in (6),

(ii) $G(P(z)) = m \cdot G(z) + 1 - m$, $\forall z \in \mathcal{D}(0, r_G)$.

Proof. By Theorem 2.3 we have $r_G > 1$. To show (i) we assume by contradiction that $1 < r_G < \psi_P$, and consider the set $P^{-1}([1, r_G)) := \{z \in \mathbb{R} : \ P(z) \in [1, r_G)\}$; $P(z) < z$ on $(1, \psi_P)$ implies that $P^{-1}([1, r_G)) = [1, y)$ with $y > r_G$. Rewriting (1), we can set $G(z) = (G(P(z)) - 1 + m)/m$ for every $z \in [r_G, y)$, extending analytically the function on $[0, y)$. Since $|P(z)| \leqslant P(|z|) < |z|$ for $1 < |z| < y$, $G(z)$ can be extended analytically on the disc of radius y, leading to a contradiction.

The claim in (ii) follows by continuity. ∎

Remark 2.5. The results in this section guarantee that we always have $r_P \geqslant r_G \geqslant \psi_P > 1$. In particular, this provides the upper bound $\mathcal{O}(\psi_P^{-j})$ for the asymptotic behavior of g_j.

Remark 2.6. Observe that, in the case of the linear fractional branching process, $r_P = \dfrac{1}{p} > r_G = \dfrac{p_0}{p} > 1$, and that, since $P(p_0/p) = p_0/p$, $r_G = \psi_P$; this is in accordance with Corollary 2.4.

3. Methods for computing $G(z)$. In this section, we first review a method known in the literature to compute the quasi-stationary distribution of a general transient Markov chain; we then discuss another natural approach, based on probabilistic arguments, which suffers from some numerical drawbacks; finally we present our new algorithm.

3.1. The returned process approach. This approach can be used to evaluate the quasi-stationary distribution $\boldsymbol{g}$ of a transient Markov chain $\{X_n\}_{n\geqslant 0}$ on $\mathbb{N}$, with the absorbing state 0 assumed to be reached in finite time with probability one, regardless the initial state. It relies on the idea that $\boldsymbol{g}$ can be approximated by the *stationary distribution* $\boldsymbol{\pi}^\mu$ of a positive recurrent *returned process* $\{X_n^\mu\}$, which is a Markov chain on the state space $\mathbb{N}_0$ that evolves exactly like the original process $\{X_n\}$ except at the times at which state 0 is visited, when it is instantly returned

to a random positive state, chosen according to a probability distribution $\boldsymbol{\mu} = (\mu_j)$ on $\mathbb{N}_0$. More precisely, if T and T^μ denote the probability transition matrices of $\{X_n\}$ and $\{X_n^\mu\}$ respectively, then T^μ is defined as $T_{ij}^\mu := T_{ij} + T_{i0}\,\mu_j$ for all $i, j \geqslant 1$, and $\boldsymbol{g} \approx \boldsymbol{\pi}^\mu$ where $\boldsymbol{\pi}^\mu$ solves $\boldsymbol{\pi}^\mu\, T^\mu = \boldsymbol{\pi}^\mu$, $\boldsymbol{\pi}^\mu\, \mathbf{1} = 1$; for more detail, see for instance [4, 8, 34].

The function $\boldsymbol{\mu} \to \boldsymbol{\pi}^\mu$ is contractive, and the quasi-stationary distribution $\boldsymbol{g}$ satisfies $\boldsymbol{g} = \boldsymbol{\pi}^g$. Therefore, in practice, instead of sampling from a fixed distribution $\boldsymbol{\mu}$ every time the process visits state 0, we sample the return state from the empirical distribution of the returned process up to that time; after a large enough time, the empirical distribution of the returned process will then be a good approximation of the quasi-stationary distribution. In summary, the returned process approach is a simulation-based method that works as follows:

(i) start the Markov chain $\{X_n\}$ in a non-absorbing state $i \geqslant 1$;
(ii) simulate a sequence of states visited by $\{X_n\}$ normally as long as it does not reach state 0;
(iii) if the Markov chain hits state 0, do not record that transition and re-sample a new state $j \geqslant 1$ at random according to the empirical estimate of the quasi-stationary distribution up until that time (that is, according to the proportion of time the chain has spent in each transient state since the start of the simulation), then go back to step *(ii)*;
(iv) after a large enough time, the samples of states will be drawn approximately from the quasi-stationary distribution $\boldsymbol{g}$.

In our setting where $\{X_n\}$ corresponds to a GW process, the simulation of $\{X_n\}$ requires the offspring of each individual to be simulated at each generation, which can be computationally demanding. In addition, a large number of generations generally need to be simulated in order to obtain a satisfactory approximation. This method is illustrated in Section 3.4.

3.2. A probabilistic interpolation approach. Here we discuss another method for computing $G(z)$ that has a probabilistic inspiration. This technique exploits Equation (1) in combination with the sequence $\{\widetilde{z}_k\}_{k\geqslant 0}$ recursively defined as

$$\widetilde{z}_{k+1} := P(\widetilde{z}_k),$$

with $\widetilde{z}_0 = 0$. This leads to the recursion $G(\widetilde{z}_{k+1}) = m \cdot G(\widetilde{z}_k) + 1 - m$, which, because $G(0) = 0$, can be solved explicitly:

$$G(\widetilde{z}_k) = 1 - m^k, \quad k \geqslant 0. \tag{7}$$

The sequence $\{\widetilde{z}_k\}_{k\in\mathbb{N}}$ has a probabilistic interpretation: $\widetilde{z}_k$ is the probability that the GW process becomes extinct by generation k, if it starts with a single individual in generation 0.

In view of (7), $G(\widetilde{z}_k)$ also has a probabilistic meaning: $G(\widetilde{z}_k) = \sum_{j\geqslant 1} g_j\, \widetilde{z}_k^j$ is the probability that a subcritical GW process observed in its quasi-stationary regime dies within the next k generations. So m^k is the probability that, if we observe a subcritical GW process which has been living for a long time, it is still going to survive for at least k generations. The mean offspring of a subcritical GW process can therefore be interpreted as the probability that the process survives one generation when it is in its quasi-stationary regime. A similar property is given in [26, Proposition 2].

From an algebraic perspective, we have at our disposal a sequence of nodes $\widetilde{z}_k \in [0, 1]$ with $\widetilde{z}_0 = 0, \widetilde{z}_1 = p_0, \widetilde{z}_2 = P(p_0), \ldots$, for interpolating the function $G(z)$. However, there are two main issues: first, the set of nodes accumulates near the point 1 and does not become dense in the interval $[0, 1]$. Second, since we are interested in the coefficients g_j we are forced to interpolate with respect to the monomial basis. This requires to solve linear systems (or linear least squares problem) with the Vandermonde matrix generated by the real nodes $\widetilde{z}_k$. It is well known that

the latter is exponentially ill-conditioned with respect to the degree of the interpolant [5,16]. The performance of this approach is tested in Section 3.4.

3.3. A new algorithm. In this section we propose an algorithm for approximating the unknown quasi-stationary distribution $\boldsymbol{g}$ characterized by $G(z)$ when the offspring distribution $P(z)$ has a radius of convergence $r_P > 1$.

The method described in Section 3.2 suffers from the bad quality of the set of nodes used for interpolation. Here, we propose an alternative strategy that performs an approximate interpolation of $G(z)$ on the roots of unity, which is the most suited set for interpolating with respect to the monomial basis.

We remark that the original functional equation $G(P(z)) = mG(z) + 1 - m$ admits an infinite number of solutions of the form $G(z) = 1 + t \cdot f(z)$, where $t \in \mathbb{C}$ is an arbitrary constant, and $f(z)$ satisfies

$$f(P(z)) - m \cdot f(z) \equiv 0. \tag{8}$$

Once $f(z)$ is known, $G(z)$ can be obtained by imposing the boundary condition $G(0) = 0$. Our strategy for computing $f(z)$ consists in discretizing the operator $f \to f \circ P - mf$ and looking for an eigenvector associated with its smallest eigenvalue.

Observe that, given $r \in (1, \psi_P)$, $\forall z \in r \cdot \mathcal{S}^1$ we have $|P(z)| < r$. Therefore, for $z \in r \cdot \mathcal{S}^1$ we can use the Cauchy integral formula and rewrite (8) as

$$\frac{1}{2\pi \mathbf{i}} \int_{r \cdot \mathcal{S}^1} f(t)(t - P(z))^{-1} dt - m \cdot f(z) = 0. \tag{9}$$

Then, we replace the left-hand side of (9) with its approximation obtained via the trapezoidal rule, choosing the scaled n-th roots of unity as nodes for integration. Even though other quadrature rules, e.g. Gauss-Legendre, have a higher order of convergence in general, for the particular case of the integration on a disk, the trapezoidal rule has exponential convergence as n increases [33]. This yields the integration scheme :

$$\frac{1}{2\pi \mathbf{i}} \int_{r \cdot \mathcal{S}^1} f(t)(t - P(z))^{-1} dt - mf(z) \approx \sum_{j=1}^{n} f(r\xi_j) \cdot \frac{r\xi_j}{n} \cdot (r\xi_j - P(z))^{-1} - mf(z), \tag{10}$$

where $\xi_j := \exp(2\pi j \mathbf{i}/n)$, $j = 1, 2, \ldots, n$. Evaluating the right-hand side of (10) in the scaled n-th roots of unity provides the system of n equations in the n unknowns $f(r\xi_j)$:

$$\sum_{h=1}^{n} f(r\xi_h) \cdot \frac{r\xi_h}{n} \cdot (r\xi_h - P(r\xi_j))^{-1} - m \cdot f(r\xi_j) \approx 0 \quad j = 1, \ldots, n. \tag{11}$$

Rewriting (11) in matrix form leads to the smallest-eigenpair problem:

$$A\mathbf{v_f} = \lambda_{min}\mathbf{v_f}, \qquad A = (a_{jh})_{j,h=1,\ldots,n}, \qquad a_{jh} = \begin{cases} \frac{r\xi_h}{n(r\xi_h - P(r\xi_j))} & h \neq j \\ \frac{r\xi_h}{n(r\xi_h - P(r\xi_h))} - m & h = j. \end{cases} \tag{12}$$

Indeed, when λ_{min} is the eigenvalue of smallest modulus of the matrix A, the vector $\mathbf{v_f}$ contains approximations of the quantities $\widetilde{f}(\xi_j) := f(r \cdot \xi_j)$, $j = 1, \ldots, n$, for a function f that verifies (8). Then, applying the *Inverse Fast Fourier Transform* (IFFT) to $\mathbf{v_f}$ provides the vector containing

the (approximate) coefficients of the interpolating polynomial $\sum_{j=0}^{n-1} \widetilde{f}_j z^j$ for $\widetilde{f}(z)$ at the nodes ξ_h, for $h = 1, \dots, n$. In order to retrieve the (approximate) interpolating polynomial for $f(z)$ we rescale the coefficients with the rule $f_j \leftarrow \widetilde{f}_j / r^j$. Finally, we impose the boundary condition $0 = G(0) = 1 + tf(0) = 1 + tf_0$, which implies $t = -1/f_0$. This yields the following (approximate) interpolating polynomial $\widehat{G}(z)$ for $G(z)$:

$$\widehat{G}(z) := \sum_{j=1}^{n-1} \widehat{g}_j z^j, \qquad \widehat{g}_j = -\frac{f_j}{f_0}.$$

The procedure is summarized in Algorithm 1; $\textsc{Eigs}(A)$ denotes any numerical method for computing the eigenvector of A associated with its smallest eigenvalue.

The construction of A and $\textsc{Eigs}(A)$ constitute the bottlenecks of the algorithm. In particular, memorizing the full matrix and running $\textsc{Eigs}(A)$ in dense arithmetic — for example using the MATLAB command `eigs(A, 1, 'SM')` — provides a quadratic cost in storage and a cubic time consumption, respectively. In this setting, we have to consider n less than 10^4 in order to carry on the computations on a standard laptop. In Section 3.5, we will show that it is possible to exploit the structure of the matrix A for achieving a cheaper storage and an efficient implementation of $\textsc{Eigs}$, allowing us to consider higher values for n.

Algorithm 1 Evaluation-Interpolation

procedure $\textsc{Compute_G}(P(z), n, r)$ $\quad \triangleright\ r > P(r) > 1$
  $m \leftarrow P^{(1)}(1)$
  $\xi \leftarrow \left(r \cdot e^{\frac{2\pi \mathrm{i} j}{n}}\right)_{j=1,\dots,n}$
  $A \leftarrow \left(\frac{\xi_h}{n(\xi_h - P(\xi_j))}\right)_{j,h=1,\dots,n}$
  $A \leftarrow A - m \cdot I_n$
  $\mathbf{v}_f \leftarrow \textsc{Eigs}(A)$
  $\mathbf{f} \leftarrow \mathrm{IFFT}(\mathbf{v}_f), \quad \mathbf{f} \leftarrow \left(\frac{f_j}{r^j}\right)_{j=0,\dots,n-1}$ $\quad \triangleright\ \sum_{j=0}^{n-1} f_j z^j$ interpolates $\widetilde{f}(z)$
  $\widehat{\mathbf{g}} \leftarrow -\frac{1}{f_0}\mathbf{f}, \quad \widehat{g}_0 \leftarrow 0$
  return $\widehat{\mathbf{g}}$
end procedure

3.4. Numerical tests. All the experiments reported in this paper have been performed on a Laptop with the dual-core Intel Core i7-7500U 2.70 GHz CPU, 256KB of level 2 cache, and 16 GB of RAM. The algorithms are implemented in MATLAB and tested under MATLAB2017a, with MKL BLAS version 11.2.3 utilizing both cores.

Example 3.1. As a first example, we consider the linear-fractional GW process with parameters $p_0 = 0.6$ and $p = 0.3$. In this case, we know that the quasi-stationary distribution is given by $g_j = 2^{-(j+1)}$, $j \geqslant 1$, see (4). This example allows us to evaluate the quality of the approximations resulting from the three algorithms that we have introduced.

In Figures 3 and 4 we denote with the label "Return map" the procedure described in Section 3.1, which we ran for 10^6 generations. With "Interpolation" we indicate the probabilistic interpolation approach of Section 3.2, that generates the data set $\{(\widetilde{z}_k, 1 - m^k)\}_{k=0,\dots,199}$ and computes the corresponding fitting polynomial of degree 12 using the `polyfit` function of MATLAB.

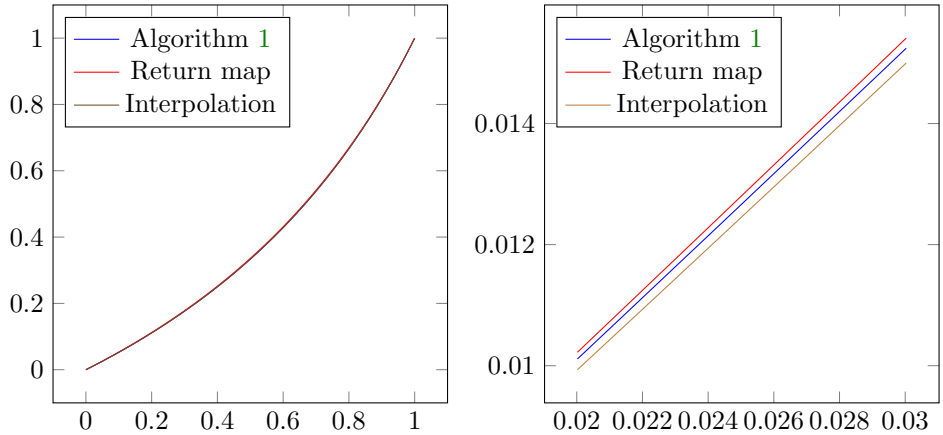


FIG. 3. *On the left; plots of the approximated solutions $\widehat{G}(z)$ returned by the three methods for the linear fraction branching process with $p_0 = 0.6$ and $p = 0.3$. On the right; zoom of the picture on the left for $z \in [0.02, 0.03]$.*

Clearly, this yields estimates only for g_j with $j = 1, \dots, 12$; however, experimentally we notice that using a higher degree for the fitting polynomial provides noisy results. Moreover, adding points to the data set brings no benefits due to the convergence of the sequence $\widetilde{z}_k$ to its limit point. Finally, we run Algorithm 1 using $n = 512$ integration nodes.

In Figure 3-left we plot the approximated solutions $\widehat{G}(z)$ returned by the three methods over the unit interval. Since the three graphs are indistinguishable on this scale, in Figure 3-right we zoom over the interval $[0.02, 0.03]$ where we finally observe some differences. In Figure 4-left we report the approximated coefficients $\widehat{g}_j$ returned by the three methods and the true g_j's. We let the index j vary in the range $[0, 52]$ because, for $j \geqslant 53$, g_j is below the machine precision. It is evident that the outcome of the interpolation method strongly differs from the ones of the other algorithms and from the true solution; in addition the coefficients of the fitting polynomial are sometimes negative. In Figure 4-right we report the relative errors $|(g_j - \widehat{g}_j)/g_j|$ of the three approaches; the results indicate that the accuracy of Algorithm 1 largely exceeds that of the others, as it returns relative errors of magnitudes close to the machine precision already for 512 integration nodes. We also remark that the return map method with 10^6 generations does not manage to provide non-zero estimates for the coefficients g_j with $j > 20$. Finally, we mention that the execution time of Return map was about 40 seconds while Algorithm 1 and Interpolation needed less than 0.1 seconds.

Example 3.2. We test Algorithm 1 on a randomly generated offspring distribution. More specifically, we set $P(z)$ equal to the polynomial of degree 8 with the following coefficients: $p_0 = 0.838$, $p_1 = 0.008$, $p_2 = 0.031$, $p_3 = 0.011$, $p_4 = 0.021$, $p_5 = 0.029$, $p_6 = 0.019$, $p_7 = 0.014$ and $p_8 = 0.029$. Consequently, we have $m = 0.776$ and $\psi_P \approx 1.101$. The latter is estimated numerically as the rightmost solution of $z = P(z)$. The parameter r is set equal to $\arg\min_{x \geqslant 1} P(x) - x$, which is obtained via the `fminsearch` function of MATLAB. This is because we want to keep the magnitude of the quantities $(P(r\xi_j) - r\xi_j)^{-1}$ (that are involved in the definition of the matrix A) under control.

In Figure 5 (top-left), we show the performances of Algorithm 1 and the features of the computed solution. In particular, we report the residual defined as

$$\text{Res} := \max_{j=1,\dots,n} |\widehat{G}(P(\xi_j)) - m\widehat{G}(\xi_j) - 1 + m|,$$

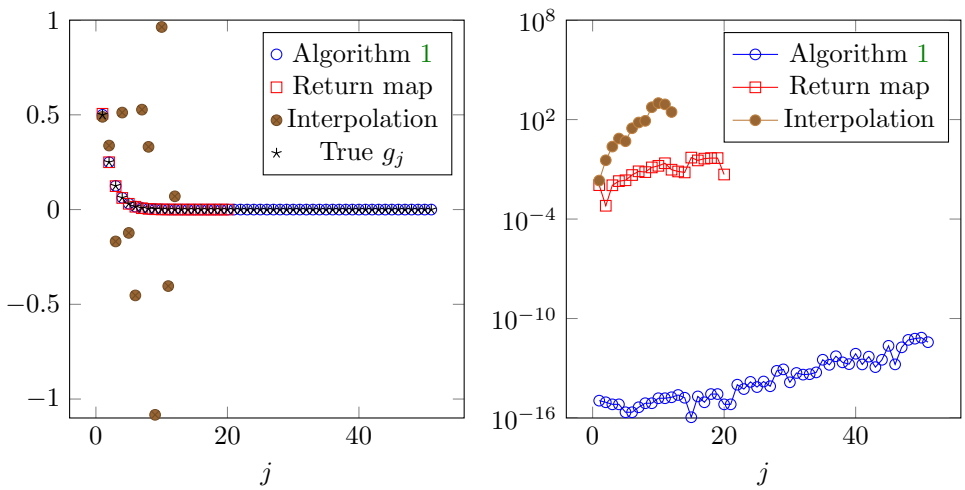


FIG. 4. *On the left: approximated coefficients $\widehat{g}_j$ computed by the three methods and true coefficients g_j of the linear fractional branching process with $p_0 = 0.6$ and $p = 0.3$. On the right: relative errors $\left|\frac{g_j - \widehat{g}_j}{g_j}\right|$ of the three approaches.*

and the sum of the coefficients $\widehat{g}_j$. We note that, in all our tests, the $\widehat{g}_j$'s are real and non-negative up to machine precision. As n increases, the execution times scale cubically; the residual decreases rapidly to 0 and the sum of the $\widehat{g}_j$'s converges to 1. In Figure 5 (top-right), we plot the first 300 coefficients $\widehat{g}_j$, computed in the case $n = 8192$, and we compare their distribution with the decay rate ψ_P^{-j}, suggested by Corollary 2.4. The outcome confirms the sharpness of the decay rate.

Example 3.3. We consider a test analogous to the one in Example 3.2, but with a mean offspring m closer to 1. We set the coefficients of $P(z)$ as $p_0 = 0.782$, $p_1 = 0.016$, $p_2 = 0.045$, $p_3 = 0.038$, $p_4 = 0.037$, $p_5 = 0.008$, $p_6 = 0.009$, $p_7 = 0.04$ and $p_8 = 0.025$, which yields $m = 0.942$ and $\psi_P \approx 1.026$. Apart from the case $n = 8192$, we note the presence of negative coefficients $\widehat{g}_j$ whose order of magnitude ranges from 10^{-3} (for $n = 256$) to 10^{-5} (for $n = 4096$). The results reported in the bottom of Figure 5 highlight the fact that we are still far from convergence; indeed the residual is much higher than in Example 3.2, and the sum of the $\widehat{g}_j$'s is not close to 1. Finally, the slope of the decay of the coefficients is further from the theoretical estimate.

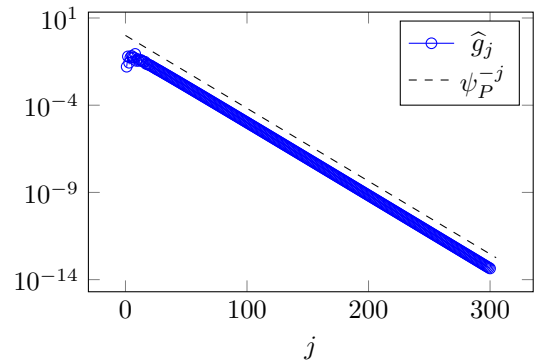
Example 3.3 suggests that for m close to 1, higher values of n are needed in order to reach a satisfactory accuracy. This leads us to deal with a large scale matrix A. In the next sections we propose an improvement of Algorithm 1 for treating this case.

3.5. Rank structure in the matrix A. We now take a closer look at the matrix A in (12). This matrix can be written as

$$A = C_{P,r}^{(n)} \cdot \operatorname{diag}\left(\frac{r\xi_1}{n}, \dots, \frac{r\xi_n}{n}\right) - mI_n,$$

where the $n \times n$ matrix $C_{P,r}^{(n)} = (c_{hj})$ is defined as $c_{hj} := (r\xi_h - P(r\xi_j))^{-1}$ with $\xi_h = \exp(2\pi \mathbf{i} h/n)$. The aim of this subsection is to show that the matrix A is well-approximated by a multiple of the identity matrix plus a low-rank matrix. In view of the Eckart-Young theorem [22, Chapter 3], this

n	Time (s)	Res	$\sum \widehat{g}_j$
256	0.05	$8.67 \cdot 10^{-2}$	0.72
512	0.11	$1.40 \cdot 10^{-2}$	0.96
1,024	0.44	$4.96 \cdot 10^{-4}$	1
2,048	1.62	$8.85 \cdot 10^{-7}$	1
4,096	9.92	$3.80 \cdot 10^{-12}$	1
8,192	78.21	$2.04 \cdot 10^{-15}$	1



n	Time (s)	Res	$\sum \widehat{g}_j$
256	0.06	$9.94 \cdot 10^{-2}$	0.73
512	0.11	$7.53 \cdot 10^{-2}$	0.74
1,024	0.39	$4.33 \cdot 10^{-2}$	0.97
2,048	1.55	$2.14 \cdot 10^{-2}$	0.67
4,096	9.94	$2.49 \cdot 10^{-2}$	0.65
8,192	79.88	$1.33 \cdot 10^{-2}$	0.86

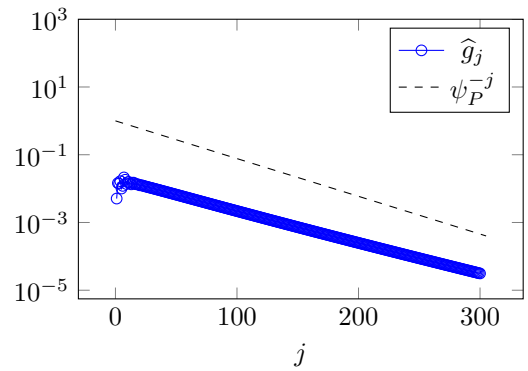


FIG. 5. *Example 3.2 (top) and Example 3.3 (bottom). On the left, performances of Algorithm 1 as n increases. On the right, comparison between the estimated coefficicients of $G(z)$, in the case $n = 8192$, and the decay suggested by Corollary 2.4.*

is equivalent to showing that the singular values σ_k of $C_{P,r}^{(n)} \cdot \operatorname{diag}\left(\frac{r\xi_1}{n}, \dots, \frac{r\xi_n}{n}\right)$ rapidly become negligible, with respect to σ_1, as k increases. We note that the analysis is reduced to studying the $n \times n$ matrix $C_{P,r}^{(n)}$, because the multiplication by the diagonal matrix does not alter the ratio σ_k/σ_1.

The matrix $C_{P,r}^{(n)}$ belongs to a well-studied class of structured matrices that we introduce within the next definition.

DEFINITION 3.4. *A matrix $(c_{hj}) \in \mathbb{C}^{m\times n}$ is called a* Cauchy matrix *if there exist two vectors $\mathbf{x} \in \mathbb{C}^m$ and $\mathbf{y} \in \mathbb{C}^n$ such that $c_{hj} = (x_h - y_j)^{-1}$. We call $\mathbf{x}, \mathbf{y}$ the* generators *and we denote the matrix (c_{hj}) by $C(\mathbf{x}, \mathbf{y})$.*

The behavior of the singular values of $C(\mathbf{x}, \mathbf{y})$ can be linked to the configurations of the two sets $\mathcal{X} := \{x_h\}_{h=1,\dots,m}$ and $\mathcal{Y} := \{y_j\}_{j=1,\dots,n}$ in the complex plane. There are many results in the literature on the singular value decay of Cauchy matrices whose corresponding sets $\mathcal{X}$ and $\mathcal{Y}$ are separated in some sense [13, 28, 29].

3.5.1. Bounds linked to polynomial approximation. The existence of accurate low-rank approximations of $C(\mathbf{x}, \mathbf{y})$ is implied by the existence of low-degree *separable approximations*

$\widetilde{a}(x,y) = \sum_{j=1}^{k} g_j(x)h_j(y)$ of the function $a(x,y) := (x-y)^{-1}$, over the set $\mathcal{X} \times \mathcal{Y}$, see [18, Section 4]. A possible way to determine a separable approximation of $a(x,y)$ consists in considering its truncated Taylor expansion with respect to one of the two variables. Intuitively, for using the Taylor expansion we need the set $\mathcal{X} \times \mathcal{Y}$ to be well separated from the singularities of $a(x,y)$. This is encoded in the following definition.

Definition 3.5. *Given $\theta \in (0,1)$ and $c \in \mathbb{C}$ we say that two sets $\mathcal{X}, \mathcal{Y} \subset \mathbb{C}$ are (θ, c)-separated if for every $x \in \mathcal{X}$ and $y \in \mathcal{Y}$ we have $|y-c| \leqslant \theta|x-c|$.*

The property in Definition 3.5 provides an explicit exponential decay in the singular values of $C(\mathbf{x},\mathbf{y})$, as stated in the next result.

Theorem 3.6 (Chandrasekaran *et al.* [13], Section 2.2). *Let $\{x_h\}_{h=1,\dots,m}, \{y_j\}_{j=1,\dots,n} \subset \mathbb{C}$ be (θ, c)-separated for a certain $\theta \in (0,1)$ and a complex center c. Then, $\forall k \in \mathbb{N}$ there exist $U \in \mathbb{C}^{m\times k}$ and $V \in \mathbb{C}^{n\times k}$ such that*

$$\|C(\mathbf{x},\mathbf{y}) - UV^*\|_2 \leqslant \frac{\theta^k}{(1-\theta)\delta}\sqrt{mn},$$

where $\delta := \min_{h=1,\dots,m} |c - x_h|$.

Remark 3.7. By the Eckart-Young theorem, Theorem 3.6 provides the bound

$$\sigma_{k+1} \leqslant \frac{\theta^k}{(1-\theta)\delta}\sqrt{mn}, \qquad k = 1, 2, \dots,$$

for the singular values of $C(\mathbf{x},\mathbf{y})$.

In order to apply Theorem 3.6 in our framework, we need to understand better where the function $P(r \cdot z)$ maps the unit circle. Relying on the stochasticity properties of the coefficients p_j, $\forall z \in \mathcal{S}^1$ we have

$$|P(r\cdot z) - p_0| = \left|\sum_{j=1}^{\infty} p_j (rz)^j\right| \leqslant \sum_{j=1}^{\infty} p_j r^j = P(r) - p_0, \tag{13}$$

i.e., $P(r \cdot \mathcal{S}^1)$ can be enclosed into the circle $p_0 + \alpha(r)\mathcal{S}^1$, with $\alpha(r) := P(r) - p_0$. This is at the basis of the next lemma.

Lemma 3.8. *Let $n \in \mathbb{N}$, $P(z) = \sum_{j\geqslant 0} p_j z^j$, $p_j \geq 0$ $\forall j \geq 0$, $P(1) = 1$, $P^{(1)}(1) \in (0,1)$, and let $r > 1$ be such that $r > P(r)$. Then $\forall k = 1, \dots, n-1$ we have*

$$\sigma_{k+1}(C_{P,r}^{(n)}) \leqslant \frac{\theta^k n}{(1-\theta)(r-p_0)}, \tag{14}$$

where $\theta := \frac{P(r)-p_0}{r-p_0} \in (0,1)$.

Proof. Let us consider $\mathbf{x} = (r\xi_j)_{j=1,\dots,n}$ and $\mathbf{y} = (P(r\xi_j))_{j=1,\dots,n}$, so that $C_{P,r}^{(n)} = C(\mathbf{x},\mathbf{y})$. In light of (13), we have

$$\frac{|y_j - p_0|}{|x_h - p_0|} = \frac{|P(r\xi_j) - p_0|}{|r\xi_j - p_0|} \leqslant \frac{P(r) - p_0}{r - p_0},$$

that is $\mathcal{X}$ and $\mathcal{Y}$ are $\left(\frac{P(r)-p_0}{r-p_0}, p_0\right)$-separated. Then, the claim follows by applying Theorem 3.6 and Remark 3.7 to $C(\mathbf{x},\mathbf{y})$. ∎

Remark 3.9. The decay rate θ in Lemma 3.8 depends on the parameter r. We note that the strategy of minimizing the difference $P(r) - r$ on the interval $(1, \psi_P)$ (when we choose r) also minimizes the quantity θ.

Example 3.10. We proceed to test the quality of the bound (14) by considering the linear fractional family of progeny distributions with $p = 1/2$:

$$P(z) = p_0 + (1 - p_0) \sum_{j \geqslant 1} \left(\frac{z}{2}\right)^j = p_0 + \frac{z(1 - p_0)}{2 - z}, \qquad p_0 \in (0, 1).$$

For these distributions, we have $m = P^{(1)}(1) \in (0, 1)$ if and only if $p_0 \in (1/2, 1)$. In particular, when p_0 is close to $1/2$, m is close to 1 and the interval $(1, \psi_P)$ shrinks drastically. This yields a ratio $(P(r) - p_0)/(r - p_0)$ close to 1 and consequently a slow decay. The opposite behavior is obtained when p_0 tends to 1.

In the left panels of Figures 6 we report the singular values of the matrix $C_{P,r}^{(n)}$ together with the bound (14), for $n = 1000$, in the cases $p_0 = 0.55$ and $p_0 = 0.95$. In the right panels of these figures we plot the three curves: $r\mathcal{S}^1$, $P(r\mathcal{S}^1)$ and $\alpha(r)\mathcal{S}^1 + p_0$. The provided bound is quite informative in the case $p = 0.95$ but it is useless when $p = 0.55$. Indeed, whenever the distance between $\mathcal{X}$ and $\mathcal{Y}$ tends to 0, e.g. in the case $p_0 = 0.55$, the Taylor expansion of $(x - y)^{-1}$ converges slowly and this translates in slowly decaying bounds for the singular values of $C(\mathbf{x}, \mathbf{y})$.

3.5.2. Bounds linked to rational approximations. A link between the quantities $\sigma_k(C_{P,r}^{(n)})$ and certain rational approximation problems has been described in [7]. This motivates the presence of a fast decay in a wider class of configurations for $\mathcal{X}$ and $\mathcal{Y}$.

THEOREM 3.11 (Beckermann and Townsend [7], Section 4). *Let $\mathcal{R}_{k,k}$ denote the set of rational functions of the form $r(z) = p(z)/q(z)$, where $p(z)$ and $q(z)$ are polynomials of degree at most k. Then*

$$\frac{\sigma_{k+1}(C(\mathbf{x}, \mathbf{y}))}{\|C(\mathbf{x}, \mathbf{y})\|_2} \leqslant Z_k(\mathcal{X}, \mathcal{Y}) := \min_{r \in \mathcal{R}_{k,k}} \frac{\max_{\mathcal{X}} |r(z)|}{\min_{\mathcal{Y}} |r(z)|}. \tag{15}$$

Relation (15) bounds the relative singular values with $Z_k(\mathcal{X}, \mathcal{Y})$, usually called the k-th *Zolotarev number*. Intuitively, these quantities become small when $\mathcal{X}$ and $\mathcal{Y}$ are well separated. For example, if $\mathcal{X} = \{|z| \geqslant r_1\}$ and $\mathcal{Y} = \{|z| \leqslant r_2\}$ with $r_1 > r_2$ then

$$Z_k(\mathcal{X}, \mathcal{Y}) \leqslant \frac{\max_{\mathcal{X}} |z^{-k}|}{\min_{\mathcal{Y}} |z^{-k}|} = \left(\frac{r_2}{r_1}\right)^k. \tag{16}$$

It can be shown that the inequality in (16) can be replaced by an equality, see [31, Theorem 2.1]. In light of (16), the bound in Theorem 3.6 can be retrieved from Theorem 3.11 by including all the points $P(z_j)$ in a disc centered at 0.

In addition, Zolotarev numbers enjoy the following properties.

PROPOSITION 3.12 (Akhiezer [1]). *Let $\mathcal{X}, \mathcal{Y}$ be disjoint subsets of $\mathbb{C}$ and assume $Z_k(\mathcal{X}, \mathcal{Y})$ is defined as in (15). Then the following properties hold:*

(i) Let $\mathcal{W}, \mathcal{Z} \subset \mathbb{C}$ and assume $\mathcal{X} \subseteq \mathcal{W}$ and $\mathcal{Y} \subseteq \mathcal{Z}$. Then $Z_k(\mathcal{X}, \mathcal{Y}) \leqslant Z_k(\mathcal{W}, \mathcal{Z})$, $\forall k \in \mathbb{N}$.

(ii) Let $T(z)$ be any Möbius transform, then $Z_k(\mathcal{X}, \mathcal{Y}) = Z_k(T(\mathcal{X}), T(\mathcal{Y}))$, $\forall k \in \mathbb{N}$.

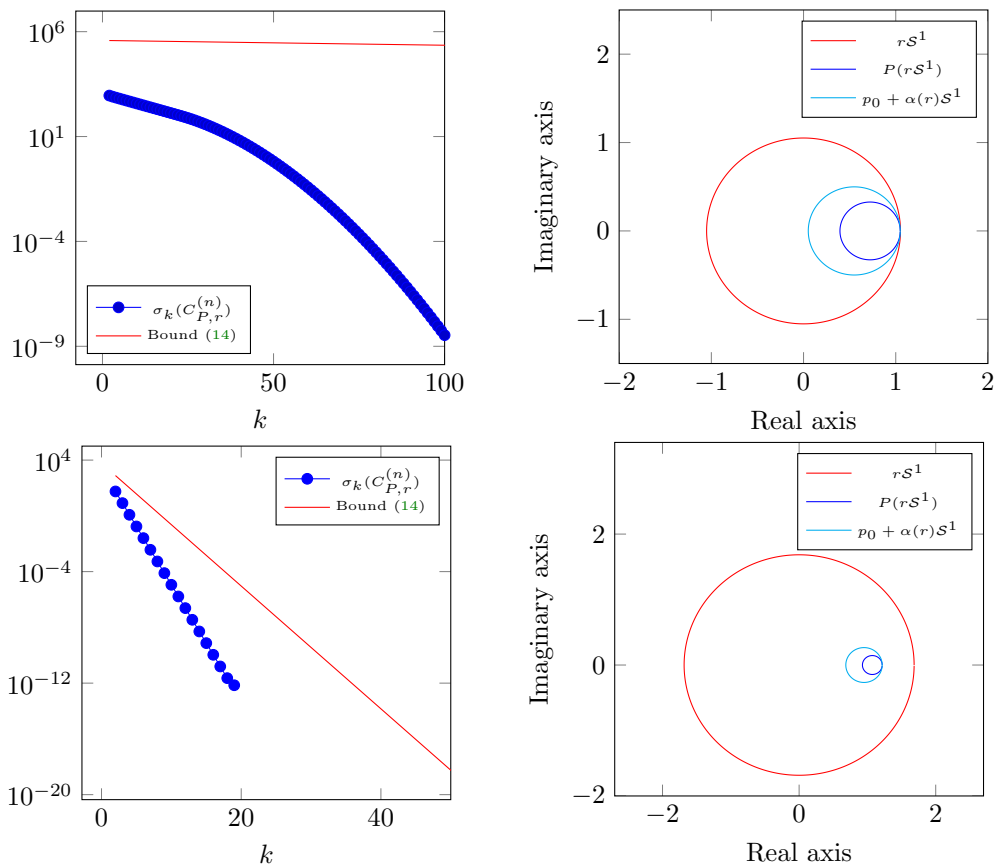


FIG. 6. *Case $p_0 = 0.55$ (top) and case $p_0 = 0.95$ (bottom). On the left, the first singular values of the matrix $C^{(n)}_{P,r}$, with $n = 1000$, compared with the bound in (14). On the right, the regions $r\mathcal{S}^1$ (in red) and $P(r\mathcal{S}^1)$ (in blue) containing the sets $\mathcal{X}$ and $\mathcal{Y}$, respectively. In light blue the curve $p_0 + \alpha(r)\mathcal{S}^1$ that encloses $\mathcal{Y}$, by (13)*

For generic complex sets, it appears to be difficult to derive explicit bounds for $Z_k(\mathcal{X}, \mathcal{Y})$. However, our situation can be re-casted to the case where $\mathcal{X}, \mathcal{Y}$ are the two connected components of the complement of an open annulus.

LEMMA 3.13. *Under the assumptions of Lemma 3.8, we have*

$$\frac{\sigma_{k+1}(C^{(n)}_{P,r})}{\|C^{(n)}_{P,r}\|_2} \leqslant \theta^k \qquad \forall k \geqslant 0, \tag{17}$$

where $\theta = \frac{(r-\beta)(P(r)-\alpha)}{(r-\alpha)(P(r)-\beta)}$, $\alpha = \frac{2p_0P(r)-P(r)^2+r^2+\sqrt{(2p_0P(r)-P(r)^2+r^2)^2-4p_0^2r^2}}{2p_0}$ *and* $\beta = \frac{r^2}{\alpha}$.

Proof. By Theorem 3.11 and Proposition 3.12 (i) we have $\sigma_{k+1}(C^{(n)}_{P,r})/\|C^{(n)}_{P,r}\|_2 \leqslant Z_k(\mathcal{W}, \mathcal{Z})$ where $\mathcal{W} = r \cdot \mathcal{S}^1$ and $\mathcal{Z} = \{|z - p_0| \leqslant P(r) - p_0\}$. The idea is to consider a Möbius transformation that maps $\mathcal{W}$ and $\partial\mathcal{Z}$ into two concentric circles centred in the origin and that maps the inner

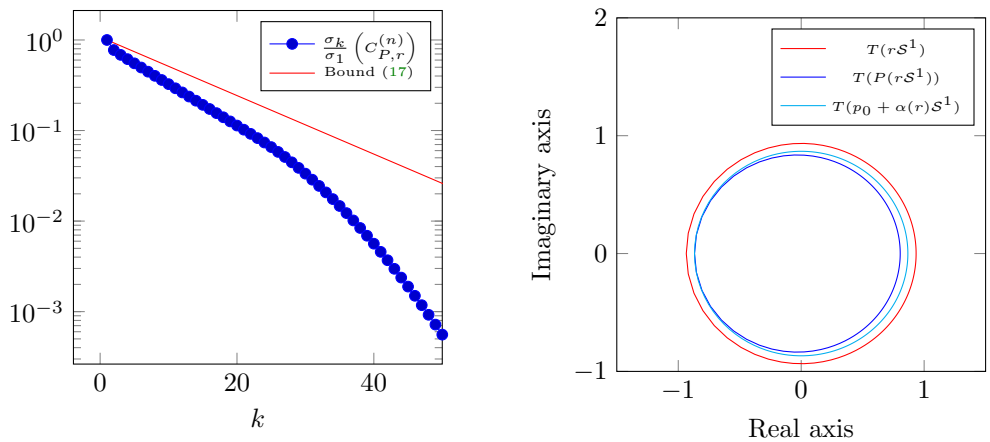


FIG. 7. *Case $p_0 = 0.55$. On the left, the first relative singular values of $C^{(n)}_{P,r}$, with $n = 1000$, compared with the bound in (17). On the right, the image of the regions $r\mathcal{S}^1$, $P(r\mathcal{S}^1)$ and $p_0+\alpha(r)\mathcal{S}^1$ under the Möebius transformation $T(z) = (z-\alpha)/(z-\beta)$*

part of $\mathcal{Z}$ into the inner part of the smaller disc. The Möbius transformation that satisfies these requirements is given by $T(z) = (z-\alpha)/(z-\beta)$ where the coefficients α, β are common *inverse points for the circles $\mathcal{W}$ and $\partial\mathcal{Z}$* [21, Section 4.2]. Algebraically, α and β solve the system

$$\begin{cases} \alpha\beta = r^2 \\ (\alpha - p_0)(\beta - p_0) = (P(r) - p_0)^2, \end{cases}$$

and $T(z)$ maps $\mathcal{W}$ into $(r-\alpha)/(r-\beta)\cdot\mathcal{S}^1$ and $\mathcal{Z}$ into $\mathcal{D}\left(0, (P(r)-\alpha)/(P(r)-\beta)\right)$. Hence, the claim follows by applying Proposition 3.12 (ii) with $T(z)$ and the bound in (16). ∎

Example 3.14. The qualitative behavior of the bound in Lemma 3.13 on the case considered in Example 3.10 is shown in Figures 7–8. We also plot the action of the Möebius transform $(z-\alpha)/(z-\beta)$ on the sets $r\mathcal{S}^1$, $P(r\mathcal{S}^1)$ and $p_0 + \alpha(r)\mathcal{S}^1$. Inequality (17) achieves a sharper description of the slope of the decay, especially for what concerns the first singular values. We expect that a complete sharpness is not attainable due to the fact that we are using an estimate for $P(r\cdot\mathcal{S}^1)$. Moreover, in order to capture the superlinear behavior that appears in the case $p_0 = 0.55$ one might consider the Zolotarev numbers on the discrete sets $\mathcal{X}, \mathcal{Y}$, see for example [6]. The latter is beyond the scope of this paper.

3.6. Exploiting the structure in Algorithm 1. The results in Section 3.5 ensure that the matrix A in (12) can be well approximated by the sum $UV^* - mI_n$ where $U, V \in \mathbb{C}^{n\times k}$ are tall and skinny matrices (i.e., $k \ll n$) that constitute a low-rank approximation of $C^{(n)}_{P,r}\cdot\mathrm{diag}\left(\frac{r\xi_1}{n}, \ldots, \frac{r\xi_n}{n}\right)$. More specifically, we just need to store two $n\times k$ matrices and one scalar in order to represent A, making the memory consumption linear with respect to the number of integration nodes.

The factors U and V are computed by means of the *Adaptive Cross Approximation with partial pivoting* (ACA) [9, Algorithm 1] whose pseudocode is reported in Algorithm 2. This procedure is heuristic but experimentally effective for the case studies reported in this paper. Note that Algorithm 2 only needs to have a cheap access to the entries of its argument, so there is no need to

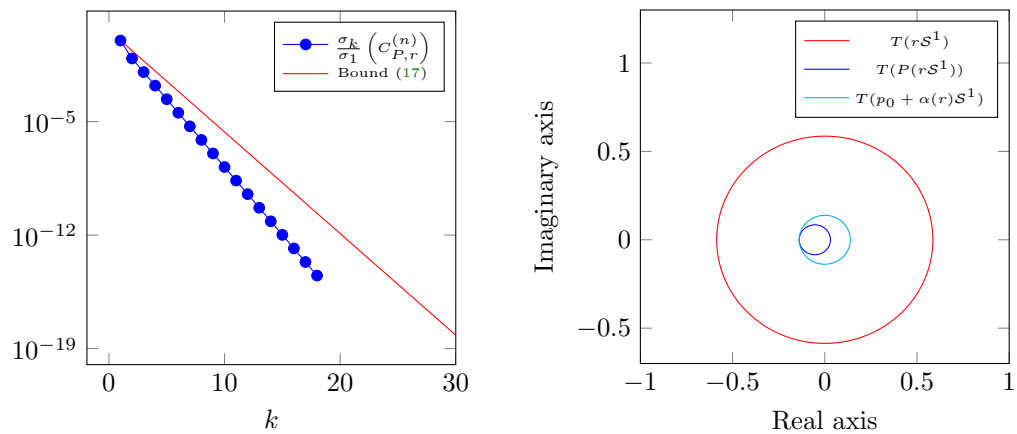


FIG. 8. *Case $p_0 = 0.95$. On the left, the first relative singular values of $C_{P,r}^{(n)}$, with $n = 1000$, compared with the bound in (17). On the right, the image of the regions $r\mathcal{S}^1$, $P(r\mathcal{S}^1)$ and $p_0+\alpha(r)\mathcal{S}^1$ under the Möebius transformation $T(z) = (z-\alpha)/(z-\beta)$*

form the full matrix $C_{P,r}^{(n)}$ in order to compress it. In all the numerical tests that call Algorithm 2, we have set $\tau = 10^{-10}$.

In order to keep the rank k of the approximation as low as possible one might apply a recompression technique — e.g. [18, Algorithm 2.17] — to the factors U and V returned by Algorithm 2. Experimentally, we notice that this strategy does not bring any advantage in term of computational time, hence we do not apply any recompression method.

We also use the structure of A in the computation of the eigenvector associated with its smallest eigenvalue. Indicating with $\lambda_{min}(\cdot)$ the eigenvalue of smallest magnitude of the (matrix) argument, we notice that if $\lambda_{min}(UV^* - mI_n) \neq -m$, then $\lambda_{min}(UV^* - mI_n) = \lambda_{min}(V^*U - mI_k) =: \widetilde{\lambda}$. Moreover, if $\mathbf{v}_{min}$ is such that $(V^*U - mI_k)\mathbf{v}_{min} = \widetilde{\lambda}\mathbf{v}_{min}$ then $U\mathbf{v}_{min}$ satisfies $(UV^* - mI_n)U\mathbf{v}_{min} = \widetilde{\lambda}U\mathbf{v}_{min}$. This suggests the procedure outlined in Algorithm 3, that has a $\mathcal{O}(nk^2 + k^3)$ cost. Replacing `eigs` in Line 6 of Algorithm 1 with Algorithm 3, we get the structured procedure for computing the coefficients of $G(z)$ that is summarized in Algorithm 4.

The method is tested on the problematic Example 3.3, where we considered larger values of n. We observe that the computed coefficients $\widehat{g}_j$ are positive, up to machine precision, and they sum up to 1 in all cases. A complete picture of this test is shown in Figure 9. The rank of the approximation of $C_{P,r}^{(n)}$ (returned by Algorithm 2) is reported in the column with the label "rank". This quantity seems to stabilize around a value less than 500. When the rank growth is limited (as in the last two numerical tests) the computational times confirm the almost linear complexity with respect to n. The residual error remains around the value 10^{-10}, this quantity being a reasonable estimate of the approximation error which affects the outcome of ACA. By decreasing the value of τ when running Algorithm 2, the method provides more accurate results for $n = 1.31 \cdot 10^5, 2.62 \cdot 10^5$.

3.7. Taking advantage of self-similarity when $m \approx 1$. The closer m to 1, the higher the rank k of the approximation of $C_{P,r}^{(n)} \cdot \mathrm{diag}\left(\frac{r\xi_1}{n}, \ldots, \frac{r\xi_n}{n}\right)$. Therefore, to limit resource consumption, when $m \approx 1$, one can think about exploiting self-similarity of Cauchy matrices. Indeed, every

Algorithm 2 Adaptive cross approximation with partial pivoting

procedure ACA(C, τ) ▷ Computes the low-rank approximation $C \approx UV^*$
  Choose a starting i_1^*
  Set $k \leftarrow 1$, $U \leftarrow [\,]$, $V \leftarrow [\,]$
  for $k = 1, 2, \dots$ **do**
    $v \leftarrow C_{i_k^*,:} - U_{i_k^*,:}\, V^*$
    $j_k^* \leftarrow \arg\max_j |v_j|$
    $u \leftarrow \left(C_{:,j_k^*} - U\, V^*_{j_k^*,:}\right) / v_{j_k^*}$
    $U \leftarrow [U, u]$
    $V \leftarrow [V, v^*]$
    if $\|u\|_2 \|v\|_2 < \tau$ **then**
      break
    end if
    $i_k^* \leftarrow \arg\max_{i \neq i_k^*} |u_j|$
  end for
  return U, V
end procedure

Algorithm 3

procedure EIGS_LR(U, V, m) ▷ Computes the smallest eigenvector of $UV^* - mI$
  $v = \text{EIGS}(V^*U - mI_k)$
  return Uv
end procedure

sub matrix of $C(\mathbf{x}, \mathbf{y})$ is again a Cauchy matrix whose generators are subvectors $\tilde{\mathbf{x}} := (x_j)_{j \in J_1}$, $\tilde{\mathbf{y}} := (y_j)_{j \in J_2}$. In our setting, $\mathbf{x}, \mathbf{y}$ represent samplings of closed curves that rotate counterclockwise, so, intuitively, selecting disjointed subsets J_1, J_2 of $\{1, \dots, n\}$ provides well separated sets of nodes. This translates in saying that the rank of the off-diagonal blocks is smaller than the rank of $C(\mathbf{x}, \mathbf{y})$ and sometimes the difference is substantial; see the example reported in Figure 10. In these cases, it is advisable to rely on representations like $\mathcal{H}^2$ [9] and HSS [12] that aim at compressing the off-diagonal submatrices while keeping the small diagonal blocks in the dense format. Adopting this strategy still allows us to store and operate with matrices with a $\mathcal{O}(n)$ complexity. The $\mathcal{H}^2$ and HSS representations of the matrix A can be obtained by applying the algorithms described in [10,25]. The use of these more sophisticated formats is beyond the scope of this paper and might be the subject of future investigations.

4. Multitype processes. In a multitype GW process, individuals of each type can give birth to children of various types according to a progeny distribution specific to the parental type. For the sake of clarity, in this section we consider the two-type case; analogous arguments can be applied to the case of an arbitrary (yet finite) number of types.

For $j = 1, 2$ and $h, k \in \mathbb{N}$, we denote by $p_{h,k}^{(j)}$ the probability that a type-j individual produces h children of type 1 and k children of type 2; we let $P_j(x, y) := \sum_{(h,k) \in \mathbb{N}^2} p_{h,k}^{(j)}\, x^h y^k$ denote the offspring generating function of a type-j individual, and $\boldsymbol{P}(x, y) := (P_1(x, y), P_2(x, y))^\top$. The mean

Algorithm 4 Low-rank Evaluation-Interpolation

procedure COMPUTE_G_LR$(P(z), n, r)$ $\triangleright\ r > P(r) > 1$
  $m \leftarrow P^{(1)}(1)$
  $\xi \leftarrow \left(r \cdot e^{\frac{2\pi \mathbf{i} j}{n}}\right)_{j=1,\dots,n}$
  $[U, V] \leftarrow \mathrm{ACA}(C^{(n)}_{P,r})$
  $V \leftarrow \frac{1}{n} V \cdot \mathrm{diag}\,(\xi)$
  $\mathbf{v}_f \leftarrow$ EIGS_LR(U, V, m)
  $\mathbf{f} \leftarrow \mathrm{IFFT}(\mathbf{v}_f), \quad \mathbf{f} \leftarrow \left(\frac{f_j}{r^j}\right)_{j=0,\dots,n-1}$
  $\widehat{\mathbf{g}} \leftarrow -\frac{1}{f_0}\mathbf{f}, \quad \widehat{g}_0 \leftarrow 0$
  return $\widehat{\mathbf{g}}$
end procedure

n	Time (s)	Res	$\sum \widehat{g}_j$	rank
16,384	2.28	$4.10 \cdot 10^{-4}$	1	264
32,768	6.88	$5.80 \cdot 10^{-7}$	1	366
65,536	21.67	$1.15 \cdot 10^{-10}$	1	465
$1.31 \cdot 10^5$	49.89	$4.33 \cdot 10^{-10}$	1	471
$2.62 \cdot 10^5$	113.19	$5.94 \cdot 10^{-10}$	1	475

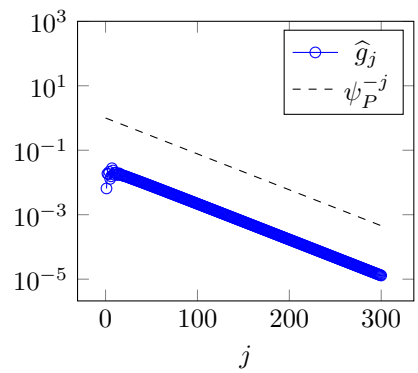


FIG. 9. *Example 3.3. On the left, performances of Algorithm 4 as n increases. On the right, comparison between the estimated coefficients of $G(z)$, in the case $n = 262144$, and the decay suggested by Corollary 2.4.*

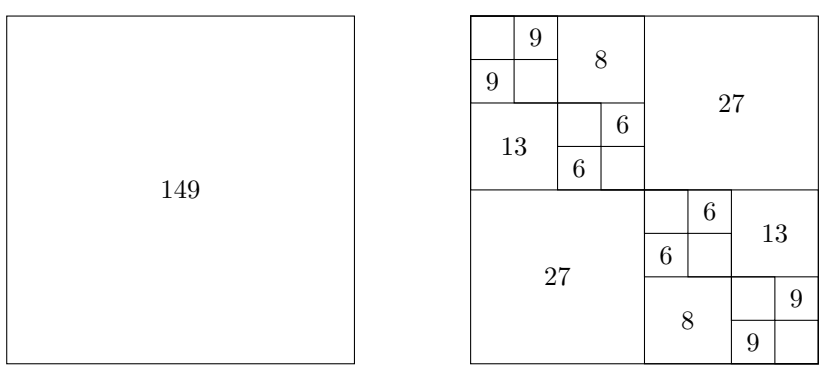


FIG. 10. *Rank (left) and off-diagonal rank distribution (right) of the matrix $C^{(n)}_{P,r}$. We set $n = 4096$, $r \approx 1.0078$ and $P(z)$ equal to the polynomial of degree 7 with coefficients $p_0 = 0.765$, $p_1 = 0.016$, $p_2 = 0.039$, $p_3 = 0.034$, $p_4 = 0.049$, $p_5 = 0.043$, $p_6 = 0.005$ and $p_7 = 0.049$ which yields $m = 0.98$.*

progeny matrix is defined as

$$M := \left.\begin{bmatrix} \frac{\partial P_1(x,y)}{\partial x} & \frac{\partial P_1(x,y)}{\partial y} \\ \frac{\partial P_2(x,y)}{\partial x} & \frac{\partial P_2(x,y)}{\partial y} \end{bmatrix}\right|_{(x,y)=(1,1)}, \tag{18}$$

and is assumed throughout the section to be positive regular, that is, $\exists n \geqslant 1$ such that $(M^n)_{ij} > 0$ for all $i, j = 1, 2$. Analogue to the mean offspring m in the single-type case, the Perron-Frobenius eigenvalue ρ of M determines the criticality of the process. Here we consider the subcritical case $\rho < 1$ with almost sure extinction regardless the type of the initial individual [19, Theorem 7.1].

Let $\boldsymbol{Z}_n := (Z_{n,1}, Z_{n,2})$ denote the population size in both types at generation $n \geqslant 0$. We assume that the second moments of the offspring distribution are finite, that is, $\mathbb{E}(Z_{n,i}Z_{n,j} \mid \boldsymbol{Z}_0 = \boldsymbol{e}_k) < \infty$ for all $i, j, k = 1, 2$. If the process starts with a single individual of type j, then the conditional probability distribution of $\boldsymbol{Z}_n$, given $\boldsymbol{Z}_n \neq 0$, converges as $n \to \infty$ to a limiting bivariate quasi-stationary distribution $\boldsymbol{g}^{(j)} := (g^{(j)}_{h,k})_{(h,k)\in\mathbb{N}^2}$, whose probability generating function

$$G_j(x, y) := \sum_{(h,k)\in\mathbb{N}^2} g^{(j)}_{h,k}\, x^h y^k, \quad x, y \in [0, 1],$$

satisfies

$$G_j(\boldsymbol{P}(x, y)) = \rho \cdot G_j(x, y) + 1 - \rho, \quad j = 1, 2; \tag{19}$$

see for instance [19, Theorem 9.1] and [2, Chapter 4, Theorem 2] for a stronger result without the second moment assumption. Here we aim at computing $G_j(x, y)$ with $g^{(j)}_{h,k} \geqslant 0$ and $G_j(0, 0) = 0$ for $j = 1, 2$. Note that $G_1(x, y) = G_2(x, y)$, therefore we remove the subscript of $G(x, y)$ and the superscript of the coefficients $g_{h,k}$.

4.1. The bivariate linear fractional case. In the two-type case, the progeny distributions of a linear fractional GW process take the form

$$P_1(x, y) = \frac{s_{11}x + s_{12}y + b_1}{c_1x + c_2y + d}, \qquad P_2(x, y) = \frac{s_{21}x + s_{22}y + b_2}{c_1x + c_2y + d}$$

for some real parameters s_{ij}, b_i, c_i, d, $i = 1, 2$. Defining the matrix $S := (s_{ij})_{i,j=1,2}$ and the vector $\boldsymbol{c} := (c_1, c_2)$, the mean progeny matrix is given by $M = (S - \mathbf{1} \otimes \boldsymbol{c})/(c_1 + c_2 + d)$. Let $\boldsymbol{\nu}$ denote its left Perron-Frobenius eigenvector, normalised such that $\nu_1 + \nu_2 = 1$. Finally, let $\boldsymbol{t} := (t_1, t_2) = -\boldsymbol{c}/d$, $t_0 = 1 - (t_1 + t_2)$, $\boldsymbol{w} := (w_1, w_2) = \boldsymbol{t}/t_0$, and $\boldsymbol{\mu} := \boldsymbol{w}(I - M)^{-1}/(1 + \boldsymbol{w}(I - M)^{-1}\mathbf{1})$. Then, the generating function of the quasi-stationary distribution is given by

$$G(x, y) = \frac{(\nu_1 - \mu_1)x + (\nu_2 - \mu_2)y}{1 - \mu_1 x - \mu_2 y},$$

see [23, Theorem 1]. This provides the explicit expression

$$g_{h,k} = (\nu_1 - \mu_1)\binom{h + k - 1}{k}\mu_1^{h-1}\mu_2^k + (\nu_2 - \mu_2)\binom{h + k - 1}{h}\mu_1^h\mu_2^{k-1}, \tag{20}$$

where the binomial coefficient $\binom{a}{b}$ is assumed to be equal to 0 whenever $b > a$.

4.2. Extension of Algorithm 1 to two dimensions. Similar to the one-dimensional case, we define for $j = 1, 2$

$$\psi_{P_j} = \begin{cases} \infty & \text{if } P_j(x,x) \text{ is of degree } 1 \\ r_{P_j} & \text{if } P_j(r_{P_j}, r_{P_j}) < r_{P_j} \\ \widehat{z} \in (1,\infty) : \ \widehat{z} = P_j(\widehat{z},\widehat{z}) & \text{otherwise,} \end{cases}$$

where r_{P_j} denotes the radius of convergence of $P_j(x,x)$. In particular, given $r_1 \in (1, \psi_{P_1})$ and $r_2 \in (1, \psi_{P_2})$, the function $P(x,y)$ is holomorphic on a open neighborhood of the polydisc $\mathcal{D}(0,r_1) \times \mathcal{D}(0,r_2)$.

As in the previous case, every function of the form $G(x,y) = 1 + t \cdot f(x,y)$ such that $t \in \mathbb{C}$ and

$$f(\boldsymbol{P}(x,y)) - \rho \cdot f(x,y) \equiv 0, \tag{21}$$

solves (19). By construction, $\forall (x,y) \in (r_1\mathcal{S}^1 \times r_2\mathcal{S}^1)$ we have $|P_j(x,y)| < r_j$, $j = 1,2$, hence applying the multivariate Cauchy integral formula to (21) provides

$$\frac{1}{(2\pi \boldsymbol{i})^2} \int_{r_1\mathcal{S}^1} \int_{r_2\mathcal{S}^1} \frac{f(\tilde{x},\tilde{y})}{(\tilde{x} - P_1(x,y))(\tilde{y} - P_2(x,y))} \, d\tilde{x} \, d\tilde{y} - \rho \ f(x,y) = 0. \tag{22}$$

Then, we approximate (22) by means of the composite trapezoidal rule, i.e., for both integrals we select as nodes of integration the scaled n-th roots of unity:

$$\frac{1}{(2\pi \boldsymbol{i})^2} \int_{r_1\mathcal{S}^1} \int_{r_2\mathcal{S}^1} \frac{f(\tilde{x},\tilde{y})}{(\tilde{x} - P_1(x,y))(\tilde{y} - P_2(x,y))} \, d\tilde{x} \, d\tilde{y} \approx \sum_{h,k=0}^{n-1} \frac{f(r_1\xi_h, r_2\xi_k) \cdot r_1 r_2 \xi_{h+k}}{n^2 (r_1\xi_h - P_1(x,y))(r_2\xi_k - P_2(x,y))}. \tag{23}$$

Evaluating (23) in all the pairs of scaled n-th roots of unity yields

$$\sum_{h,k=0}^{n-1} \frac{f(r_1\xi_h, r_2\xi_k) \cdot r_1 r_2 \xi_{h+k}}{n^2 \cdot (r_1\xi_h - P_1(r_1\xi_s, r_2\xi_t))(r_2\xi_k - P_2(r_1\xi_s, r_2\xi_t))} - \rho f(r_1\xi_s, r_2\xi_t) \approx 0, \quad s,t = 1,\dots,n.$$

Rearranging the (approximate) evaluations of f into the vector $\boldsymbol{v}_f \in \mathbb{C}^{n^2}$, $(\boldsymbol{v}_f)_{h+nk} \approx f(r_1\xi_h, r_2\xi_k)$, leads us to the eigenvalue problem

$$A\boldsymbol{v_f} = \lambda_{min}\boldsymbol{v_f}, \qquad A = C^{(n^2)}_{P_1,P_2,r_1,r_2} D - \rho I_{n^2} \in \mathbb{C}^{n^2 \times n^2}, \tag{24}$$

where

$$\left(C^{(n^2)}_{P_1,P_2,r_1,r_2}\right)_{s+n(t-1),h+n(k-1)} := \frac{1}{(r_1\xi_h - P_1(r_1\xi_s, r_2\xi_t))(r_2\xi_k - P_2(r_1\xi_s, r_2\xi_t))},$$

$D = n^{-2}\,\mathrm{diag}(\boldsymbol{\xi}_{(1)} \otimes \boldsymbol{\xi}_{(2)})$, and $\boldsymbol{\xi}_{(j)} \in \mathbb{C}^n$ is the vector containing the n-th roots of unity in the counterclockwise order multiplied by the constant r_j.

After computing a vector $\boldsymbol{v_f}$ that verifies (24), we apply the two-dimensional FFT on it; for example, this task is performed by the MATLAB command `ifft2(reshape(`$\boldsymbol{v_f}$`, n, n))`. This returns the (approximate) coefficients of the interpolating bivariate polynomial $\sum_{h,k=0}^{n-1} \widetilde{f}_{h,k} x^h y^k$ for $\widetilde{f}(x,y) := f(r_1 x, r_2 y)$. In order to obtain those for $f(x,y)$ we rescale them with the rule

$f_{h,k} \leftarrow \widetilde{f}_{h,k}/(r_1^h r_2^k)$. Once again, we impose the boundary condition $0 = G(0,0) = 1 + tf(0,0) = 1 + tf_{0,0}$, which implies $t = -1/f_{0,0}$. This yields the following (approximate) interpolating bivariate polynomial $\widehat{G}(x,y)$ for $G(x,y)$:

$$\widehat{G}(x,y) := \sum_{h,k=0}^{n-1} \widehat{g}_{h,k} x^h y^k, \qquad \begin{cases} \widehat{g}_{0,0} = 0 \\ \widehat{g}_{h,k} = -\frac{f_{h,k}}{f_{0,0}}, \quad (h,k) \neq (0,0). \end{cases}$$

The whole procedure is summarized in Algorithm 5.

In our implementation, the parameters r_1 and r_2 are set as $r_j = \arg\min_{x \geqslant 1} P_j(x,x) - x$, $j = 1, 2$. A significant difference in the parameters r_1 and r_2 suggests the use of different levels of discretization on the two integrals. This requires to slightly modify Algorithm 5 in order to consider n_1 and n_2 integration nodes for the two integrals in (23). In the numerical experiments of Section 4.1 we always use $n_1 = n_2 = n$.

Algorithm 5 Evaluation-Interpolation in the 2D case

procedure COMPUTE_G_2D$(P_1(x,y), r_1, P_2(x,y), r_2, n)$ $\triangleright$ $r_j \in (1, \psi_{P_j})$
  $\rho \leftarrow$ spectral radius of M given by (18)
  $\boldsymbol{\xi} \leftarrow \left(e^{\frac{2\pi i j}{n}}\right)_{j=1,\dots,n}$
  $A \leftarrow \left(\frac{1}{(r_1\xi_h - P_1(r_1\xi_s, r_2\xi_t))(r_2\xi_k - P_2(r_1\xi_s, r_2\xi_t))}\right)_{s+n(t-1),h+n(k-1)}, \quad h,k,s,t, = 1,\dots,n$
  $D \leftarrow \frac{r_1 r_2}{n^2} \operatorname{diag}(\xi \otimes \xi)$
  $A \leftarrow A \cdot D - \rho \cdot I_{n^2}$
  $\boldsymbol{v}_f \leftarrow$ EIGS(A), $\quad V_f \leftarrow$ RESHAPE$(\boldsymbol{v}_f, n, n)$
  $\widehat{G} \leftarrow$ IFFT2(V_f), $\quad \widehat{G} \leftarrow \left(\frac{\widehat{g}_{h,k}}{r_1^h r_2^k}\right)_{h,k=0,\dots,n-1}$ $\triangleright$ $\sum_{h,k=0}^{n-1} \widehat{g}_{h,k} x^h y^k$ interpolates $f(r_1 x, r_2 y)$
  $\widehat{G} \leftarrow -\frac{1}{\widehat{g}_{0,0}} \widehat{G}, \quad \widehat{g}_{0,0} \leftarrow 0$
  return $\widehat{G}$
end procedure

4.3. Rank structure in the matrix A. The size of the linear system (24) depends quadratically on the number of nodes n that we use for discretizing each integral in (22). In particular, the execution — in dense arithmetic — of Algorithm 5 rapidly become computationally not feasible as n increases. We here see that, similar to the single-type scenario, the matrix A exhibits a rank structure and we discuss how to modify Algorithm 5 in order to consider larger values of n.

Let us denote with $\mathbf{1}_n$ the vector of all ones of length n; then we can write

$$C^{(n^2)}_{P_1,P_2,r_1,r_2} = C(\mathbf{1}_n \otimes \boldsymbol{\xi}_{(1)}, P_1(\boldsymbol{\xi}_{(1)}, \boldsymbol{\xi}_{(2)})) \circ C(\boldsymbol{\xi}_{(1)} \otimes \mathbf{1}_n, P_1(\boldsymbol{\xi}_{(1)}, \boldsymbol{\xi}_{(2)})),$$

where $\circ$ denotes the Hadamard product. In light of (24), A is obtained by applying a column scaling and a diagonal shift to the Hadamard product of two Cauchy matrices. Intuitively, if the latter are both numerically low-rank, then we expect the numerical rank of $C^{(n^2)}_{P_1,P_2,r_1,r_2}$ to be much smaller than n^2. More formally, as pointed out in [32, Section 4.2], Hadamard products of Cauchy matrices solve certain rank structured linear matrix equations and this enables to state decaying bounds for their singular values; see Theorem 2 in [32].

Since we have a cheap access to the entries of $C^{(n^2)}_{P_1,P_2,r_1,r_2}$, we compress it using Algorithm 2 in place of forming the full matrix A in Line 4 of Algorithm 5. Once again, this provides two tall and skinny matrices U, V whose storage consumption is $\mathcal{O}(n^2)$. Finally, we apply the diagonal scaling to the matrix V and we replace the call to EIGS, in Line 7, with a call to Algorithm 3. The modified procedure is reported in Algorithm 6 and tested in the next example.

Algorithm 6 Low-rank Evaluation-Interpolation in the 2D case

1: **procedure** COMPUTE_G_2D_LR($P_1(x,y), r_1, P_2(x,y), r_2, n$)
2: $\rho \leftarrow$ spectral radius of M given by (18)
3: $\boldsymbol{\xi} \leftarrow \left(e^{\frac{2\pi \boldsymbol{i} j}{n}}\right)_{j=1,\dots,n}$
4: $[U,V] \leftarrow \mathrm{ACA}(C^{(n^2)}_{P_1,P_2,r_1,r_2})$
5: $D \leftarrow \frac{r_1 r_2}{n^2}\,\mathrm{diag}(\xi \otimes \xi)$
6: $V \leftarrow V \cdot D$
7: $\boldsymbol{v}_f \leftarrow$ EIGS_LR(U,V,ρ), $\quad V_f \leftarrow$ RESHAPE$(\boldsymbol{v}_f, n, n)$
8: $\widehat{G} \leftarrow \mathrm{IFFT2}(V_f)$, $\quad \widehat{G} \leftarrow \left(\frac{\widehat{g}_{h,k}}{r_1^h r_2^k}\right)_{h,k=0,\dots,n-1}$
9: $\widehat{G} \leftarrow -\frac{1}{\widehat{g}_{0,0}}\widehat{G}$, $\quad \widehat{g}_{0,0} \leftarrow 0$
10: **return** $\widehat{G}$
11: **end procedure**

4.4. Numerical tests.

Example 4.1. We first consider a two-type linear fractional GW process whose offspring distribution is defined in Section 4.1, with the following parameters:

$$S = \begin{bmatrix} 0.3 & 0.2 \\ 0.2 & 0.3 \end{bmatrix}, \qquad \boldsymbol{c} = \begin{bmatrix} -0.05 \\ -0.05 \end{bmatrix}, \qquad \boldsymbol{b} = \begin{bmatrix} 0.4 \\ 0.4 \end{bmatrix}, \qquad d = 1.$$

Via direct computation we find that $\rho = \frac{2}{3}$, and the explicit expression of the coefficients $g_{h,k}$ is given in (20) with $\nu_1 = \nu_2 = \frac{1}{2}$ and $\mu_1 = \mu_2 = \frac{1}{8}$.

We run Algorithm 6 for this example with $n = 256$, and we compute the relative error $|(\widehat{g}_{h,k} - g_{h,k})/g_{h,k}|$ of the approximate coefficients $\widehat{g}_{h,k}$ returned by our method. The results are shown in Figure 11 where we let the indices h, k vary in $[0, 20]$ (outside this range, the coefficients $g_{h,k}$ are below the machine precision). We see that the most accurate quantities are those with highest magnitude, i.e., the coefficients $\widehat{g}_{h,k}$ whose index (h,k) is close to $(0,0)$. The relative error increases progressively as h, k increase, reaching about 10^{-4} for the quantities that are at the level of the machine precision.

Example 4.2. Here we test the scalability of Algorithm 6 on a randomly generated example. We consider $P_1(x,y)$ and $P_2(x,y)$ equal to bivariate polynomials of degree $(2,2)$ with coefficients reported in Table 1. This yields $\rho \approx 0.5884$, $r_1 = 1.2462$ and $r_2 = 1.4104$. The radii r_j are estimated using the MATLAB function `fminsearch`.

In Figure 12 left, we show the performances of Algorithm 6 and the features of the computed solution. For all values of n the computed coefficients $\widehat{g}_{h,k}$ are non negative up to machine precision. The reported residual is defined as

$$\mathrm{Res} := \max_{i,j=1,\dots,n} |\widehat{G}(P_1(\xi_i,\xi_j), P_2(\xi_i,\xi_j)) - \rho\widehat{G}(\xi_i,\xi_j) - 1 + \rho|, \qquad \xi_j = \exp(2\pi \boldsymbol{i} j/n).$$

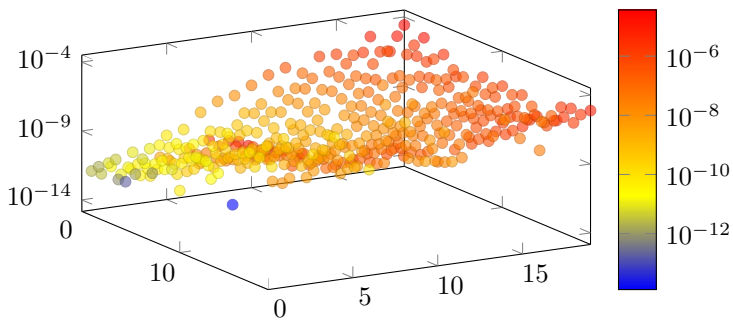


FIG. 11. *Example 4.1. Relative error of the computed coefficients $\widehat{g}_{h,k}$ in the bivariate linear fractional example.*

	$p^{(j)}_{0,0}$	$p^{(j)}_{0,1}$	$p^{(j)}_{0,2}$	$p^{(j)}_{1,0}$	$p^{(j)}_{1,1}$	$p^{(j)}_{1,2}$	$p^{(j)}_{2,0}$	$p^{(j)}_{2,1}$	$p^{(j)}_{2,2}$
$P_1(x,y)$	0.798	0.029	0.009	0.015	0.010	0.022	0.052	0.020	0.045
$P_2(x,y)$	0.694	0.041	0.057	0.035	0.027	0.043	0.024	0.051	0.028

TABLE 1
Example 4.2. Coefficients of the bivariate polynomials $P_j(x,y)$.

In the last column we also report the rank of the approximation of the matrix $C^{(n^2)}_{P_1,P_2,r_1,r_2}$ returned by Algorithm 2. The growth of this quantity — as the number of nodes increases — makes the time consumption slightly super-quadratic with respect to n. In Figure 12-right, we plot the coefficients $\widehat{g}_{h,k}$ up to degree $(63,63)$, computed in the case $n = 512$. Experimentally, we observe that if h or k is not in the range $[0,63]$ then $\widehat{g}_{h,k} \leqslant 10^{-31}$. This confirms that a small number of coefficients is sufficient to describe the quasi-stationary distribution with high accuracy.

5. Conclusions. We provided a fully algebraic analysis of the interplay between the regularity of the offspring distribution and that of the quasi-stationary distribution of a subcritical GW process. We proposed a new numerical method for computing the quasi-stationary distribution. We showed that our approach can significantly outperform the accuracy of other techniques based on simulations or on interpolation.

Moreover, we provided a theoretical analysis of the low-rank structure stemming from the discretization of the problem. This enabled our algorithm to be slightly modified in order to scale well with the fineness of the discretization. The reported numerical tests confirm the scalability of computational time.

6. Appendix. Here we report the proofs of some of the results in Section 2.

Proof of Proposition 2.1. First, note that $G(z)$ analytic on $\mathcal{B}(0, r_G)$ with $r_G > 1$ implies $G(z)$ analytic at 1.

Now, let us assume $G(z)$ analytic on an open neighborhood A_1 of 1. We proceed by proving that $G(z)$ is analytic at every point of $\mathcal{S}^1$. Given $z \in \mathcal{S}^1$ we distinguish between two cases: $|P(z)| < 1$ and $|P(z)| = 1$. If $|P(z)| < 1$, then there exists an open neighborhood A_z of z such that $|P(\widetilde{z})| < 1$ $\forall \widetilde{z} \in A_z$. Since $G(z)$ verifies (1), the expression

$$G(z) = m^{-1}(G(P(z)) + m - 1) \tag{25}$$

n	Time (s)	Res	$\sum \widehat{g}_{h,k}$	rank
16	0.12	0.27	1.17	107
32	0.18	$6.59 \cdot 10^{-2}$	0.9	192
64	0.86	$2.26 \cdot 10^{-3}$	1	306
128	5.66	$1.45 \cdot 10^{-5}$	1	437
256	33.45	$9.53 \cdot 10^{-10}$	1	572
512	218.49	$5.95 \cdot 10^{-12}$	1	673

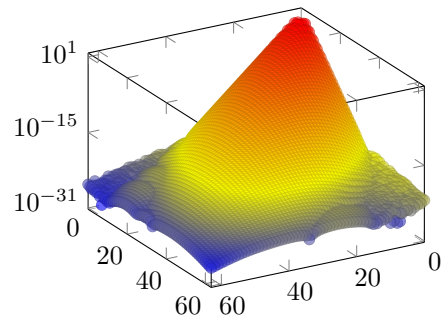


FIG. 12. *Example 4.2. On the left, performances of Algorithm 5 as n increases. On the right, 2D plot of the coefficients $\widehat{g}_{h,k}$, in the case $n = 512$.*

provides an analytic continuation of $G(z)$ on A_z. If $|P(z)| = 1$, then $1 = |p_0 + w|$ where $w := \sum_{j=1}^{\infty} p_j z^j$. Since $p_0 \in (0,1)$ and $|w| \leqslant \sum_{j=1}^{\infty} p_j = 1 - p_0$, the sum $p_0 + w$ has modulus 1 if and only if $w = 1 - p_0$, i.e., $P(z) = 1 \in A_1$. In particular, there exists an open neighborhood A_z of z such that $P(\widetilde{z}) \in A_1$ $\forall \widetilde{z} \in A_z$. Once again, (25) defines an analytic continuation of $G(z)$ on A_z.

By construction, $G(z)$ is analytic on $\mathcal{B}(0,1)$ and the union $\mathcal{B}(0,1) \bigcup \{A_z\}_{z \in \mathcal{S}^1}$ yields an open set $\widehat{A}$ that contains $\mathcal{D}(0,1)$ where $G(z)$ is analytic. This implies that there exists $r_G > 1$ such that $B(0, r_G) \subseteq \widehat{A}$ and $G(z)$ is analytic on $\mathcal{B}(0, r_G)$. ∎

6.1. Differentiability of $G(z)$. First we establish the relationship between the higher order derivatives of $G(z)$ and those of $P(z)$. Differentiating (1) h times ($h \geqslant 1$) leads to

$$m\, G^{(h)}(z) = (G \circ P)^{(h)}(z), \tag{26}$$

where $(G \circ P)(z) := G(P(z))$. The derivative of the composition is expressed in closed form with the *Faà di Bruno's formula* [14],

$$(G \circ P)^{(h)}(z) = \sum_{j=1}^{h} G^{(j)}(P(z)) \cdot B_{h,j}\left(P^{(1)}(z), \ldots, P^{(h-j+1)}(z)\right), \tag{27}$$

which involves the so-called *Bell polynomials* $B_{h,j}$ [27], defined as

$$B_{h,j}(x_1, \ldots, x_{h-j+1}) := \sum \frac{h!}{j_1! \ldots j_{h-j+1}!} \prod_{s=1}^{h-j+1} \left(\frac{x_s}{s!}\right)^{j_s},$$

where the sum is taken over all sequences $j_1, j_2, \ldots, j_{h-j+1}$ of non-negative integers such that $\sum_{s=1}^{h-j+1} j_s = j$ and $\sum_{s=1}^{h-j+1} s \cdot j_s = h$. In particular we have $B_{h,h}(x_1) = x_1^h$. Plugging (27) into (26) and evaluating at $z = 1$ yields the relation

$$(m - m^h) G^{(h)}(1) = \sum_{j=1}^{h-1} G^{(j)}(1) \cdot B_{h,j}\left(P^{(1)}(1), \ldots, P^{(h-j+1)}(1)\right), \tag{28}$$

which is only informative for $h > 1$. Equation (28) highlights the connection between the existence of higher order derivatives of $P(z)$ and those of $G(z)$. In probabilistic terms, it relates the factorial

moments of the quasi-stationary distribution to those of the offspring distribution. We are now ready to prove the following lemma.

LEMMA 6.1. *For any $1 < h \in \mathbb{N}$, $P^{(h)}(1)$ is finite if and only if $G^{(h)}(1)$ is finite.*

Proof. First, assume that $P^{(h)}(1) < \infty$; observe that this implies $\sum_{j=2}^{\infty} j \log(j) p_j < \infty$, or equivalently, $G^{(1)}(1) < \infty$, in light of Theorem 2.2. Moreover, (28) expresses $G^{(h)}(1)$ as a linear combination of $G^{(1)}(1), \dots, G^{(h-1)}(1)$, whose coefficients are polynomial functions of $P^{(1)}(1), \dots, P^{(h)}(1)$. The claim then follows using an inductive argument.

Next, assume $G^{(h)}(1) < \infty$. The only term which involves the h-th derivative of $P(z)$ in the right-hand side of (28) is $G^{(1)}(1)P^{(h)}(1)$, which is obtained by choosing $j = 1$, $j_1 = \dots, j_{h-1} = 0$ and $j_h = 1$ in the series expansion. Since $G^{(1)}(1) \neq 0$, this allows us to express $P^{(h)}(1)$ as a well-defined function of $G^{(1)}(1), \dots, G^{(h)}(1)$ and $P^{(1)}(1), \dots, P^{(h-1)}(1)$. The claim then again follows by induction. ∎

6.2. Proof of Theorem 2.3. We recall an identity regarding Bell polynomials.

LEMMA 6.2 (Wang and Wang [35], Lemma 2.6). *Let $f(z) := \sum_{j=1}^{\infty} \frac{f_j}{j!} z^j$, then $\forall h, k \in \mathbb{N}$,*

$$B_{h,k}(f_1, \dots, f_{h-k+1}) = \frac{h!}{k!} \cdot [z^h](f(z)^k),$$

where $[z^h](\cdot)$ denotes the operator that extracts the h-th coefficient from the power series expansion of the argument around zero.

Proof of Theorem 2.3. The function $P(z)$ being analytic at $z = 1$ by assumption, we consider the power series expansion of $\widetilde{P}(z) := P(1+z) = \sum_{j \geqslant 0} \tilde{p}_j z^j$ that has radius of convergence $r_P - 1$.

By Proposition 2.1, the claim is equivalent to having $G(z)$ analytic at 1. Therefore, we proceed by considering the (left looking) Taylor expansion of $G(z)$ at 1 and by showing that its radius of convergence is non-zero. In view of (5) this is equivalent to showing that $\exists \rho, \theta_G > 0$ such that

$$G^{(h)}(1) \leqslant \theta_G \cdot \rho^{-h} \cdot h!, \qquad \forall h \in \mathbb{N}.$$

Choosing $\theta_G = \max\{1, G^{(1)}(1) \cdot \rho\}$ provides the claim for $h = 0$ and $h = 1$ without limiting the parameter ρ. For $h > 1$ we use an inductive argument; from (28) we get

$$\begin{aligned} G^{(h)}(1) &= \frac{1}{m - m^h} \sum_{j=1}^{h-1} G^{(j)}(1) \cdot B_{h,j}\left(P^{(1)}(1), \dots, P^{(h-j+1)}(1)\right) \\ &\leqslant \frac{\theta_G}{m - m^h} \sum_{j=1}^{h-1} \rho^{-j} \cdot j! \cdot B_{h,j}\left(P^{(1)}(1), \dots, P^{(h-j+1)}(1)\right). \end{aligned}$$

Observe that $P^{(j)}(1) = j! \cdot \tilde{p}_j$, therefore, Lemma 6.2 implies

$$B_{h,j}\left(P^{(1)}(1), \dots, P^{(h-j+1)}(1)\right) = \frac{h!}{j!} \cdot [z^h]\left((\widetilde{P}(z) - 1)^j\right).$$

Since $(\widetilde{P}(z) - 1)^j$ also has radius of convergence $r_P - 1$ and its expansion involves non-negative coefficients, the h-th coefficient of the latter satisfies

$$[z^h]\left((\widetilde{P}(z) - 1)^j\right) \leqslant (\widetilde{P}(r) - 1)^j r^{-h}, \qquad \forall r \in (0, r_P - 1).$$

In particular, selecting $\widetilde{r} \in (0, \psi_P - 1)$, where ψ_P is given in (6), provides $P(1+\widetilde{r}) \leqslant 1+\widetilde{r}$ and $[z^h]\left((\widetilde{P}(z)-1)^j\right) \leqslant \widetilde{r}^{(j-h)}$. Coming back to $G^{(h)}(1)$, we then have

$$G^{(h)}(1) \leqslant \frac{\theta_G \cdot h!}{\widetilde{r}^h (m-m^h)} \sum_{j=1}^{h-1} \left(\frac{\widetilde{r}}{\rho}\right)^j = \theta_G \cdot \rho^{-h} \cdot h! \cdot \frac{\left(\frac{\widetilde{r}}{\rho}\right)^{1-h} - 1}{\left(1-\frac{\widetilde{r}}{\rho}\right)(m-m^h)}.$$

Choosing ρ small enough we can ensure $\frac{\left(\frac{\widetilde{r}}{\rho}\right)^{1-h}-1}{\left(1-\frac{\widetilde{r}}{\rho}\right)(m-m^h)} < 1$ independently of $h > 1$. This completes the proof. ∎

7. Acknowledgements. Sophie Hautphenne thanks the Australian Research Council for support through Discovery Early Career Researcher Award DE150101044. The authors also thank Daniel Kressner, Beatrice Meini and Phil Pollett for fruitful discussions, as well as three anonymous referees for valuable comments and suggestions.

REFERENCES

[1] N. I. Akhiezer, *Elements of the theory of elliptic functions*, vol. 79, American Mathematical Soc., 1990.
[2] K. B. Athreya, P. E. Ney, and P. Ney, *Branching processes*, Courier Corporation, 2004.
[3] J. Bagley, *Asymptotic properties of subcritical Galton-Watson processes*, Journal of Applied Probability, 19 (1982), pp. 510–517.
[4] A. Barbour and P. Pollett, *Total variation approximation for quasi-equilibrium distributions, ii*, Stochastic Processes and their Applications, 122 (2012), pp. 3740–3756.
[5] B. Beckermann, *The condition number of real Vandermonde, Krylov and positive definite Hankel matrices*, Numer. Math., 85 (2000), pp. 553–577, doi:10.1007/PL00005392, https://doi.org/10.1007/PL00005392.
[6] B. Beckermann and A. Gryson, *Extremal rational functions on symmetric discrete sets and superlinear convergence of the ADI method*, Constr. Approx., 32 (2010), pp. 393–428, doi:10.1007/s00365-010-9087-6, https://doi.org/10.1007/s00365-010-9087-6.
[7] B. Beckermann and A. Townsend, *On the singular values of matrices with displacement structure*, SIAM J. Matrix Anal. Appl., 38 (2017), pp. 1227–1248, doi:10.1137/16M1096426, https://doi.org/10.1137/16M1096426.
[8] J. Blanchet, P. Glynn, and S. Zheng, *Empirical analysis of a stochastic approximation approach for computing quasi-stationary distributions*, in EVOLVE-A Bridge between Probability, Set Oriented Numerics, and Evolutionary Computation II, Springer, 2013, pp. 19–37.
[9] S. Börm, *$\mathcal{H}^2$-matrices – an efficient tool for the treatment of dense matrices*. Habilitationsschrift, Christian-Albrechts-Universität zu Kiel, 2006.
[10] D. Cai, E. Chow, Y. Saad, and Y. Xi, *Smash: Structured matrix approximation by separation and hierarchy*, arXiv preprint arXiv:1705.05443, (2017).
[11] L. Carleson and T. W. Gamelin, *Complex dynamics*, Universitext: Tracts in Mathematics, Springer-Verlag, New York, 1993, doi:10.1007/978-1-4612-4364-9, https://doi.org/10.1007/978-1-4612-4364-9.
[12] S. Chandrasekaran, M. Gu, and T. Pals, *A fast ULV decomposition solver for hierarchically semiseparable representations*, SIAM Journal on Matrix Analysis and Applications, 28 (2006), pp. 603–622.
[13] S. Chandrasekaran, M. Gu, X. Sun, J. Xia, and J. Zhu, *A superfast algorithm for toeplitz systems of linear equations*, SIAM Journal on Matrix Analysis and Applications, 29 (2007), pp. 1247–1266.
[14] L. Comtet, *Advanced combinatorics*, D. Reidel Publishing Co., Dordrecht, enlarged ed., 1974. The art of finite and infinite expansions.
[15] P. Flajolet and R. Sedgewick, *Analytic combinatorics*, Cambridge University Press, Cambridge, 2009, doi:10.1017/CBO9780511801655, https://doi.org/10.1017/CBO9780511801655.
[16] W. Gautschi and G. Inglese, *Lower bounds for the condition number of Vandermonde matrices*, Numer. Math., 52 (1988), pp. 241–250, doi:10.1007/BF01398878, https://doi.org/10.1007/BF01398878.
[17] P. Haccou, P. Jagers, and V. A. Vatutin, *Branching processes: variation, growth, and extinction of populations*, no. 5, Cambridge university press, 2005.
[18] W. Hackbusch, *Hierarchical matrices: Algorithms and analysis*, Springer, 2015.

[19] T. E. Harris, *The theory of branching processes*, Courier Corporation, 2002.

[20] C. R. Heathcote, E. Seneta, and D. Vere-Jones, *A refinement of two theorems in the theory of branching processes*, Theory of Probability & Its Applications, 12 (1967), pp. 297–301.

[21] P. Henrici, *Applied and computational complex analysis. Vol. 1*, Wiley Classics Library, John Wiley & Sons, Inc., New York, 1988. Power series—integration—conformal mapping—location of zeros, Reprint of the 1974 original, A Wiley-Interscience Publication.

[22] R. A. Horn and C. R. Johnson, *Topics in matrix analysis*, Cambridge University Press, Cambridge, 1994. Corrected reprint of the 1991 original.

[23] A. Joffe and G. Letac, *Multitype linear fractional branching processes*, Journal of applied probability, 43 (2006), pp. 1091–1106.

[24] G. Koenigs, *Recherches sur les intégrales de certaines équations fonctionnelles*, Ann. Sci. École Norm. Sup. (3), 1 (1884), pp. 3–41, http://www.numdam.org/item?id=ASENS_1884_3_1__S3_0.

[25] P. G. Martinsson, *A fast randomized algorithm for computing a hierarchically semiseparable representation of a matrix*, SIAM J. Matrix Anal. Appl., 32 (2011), pp. 1251–1274, doi:10.1137/100786617, https://doi.org/10.1137/100786617.

[26] S. Méléard, D. Villemonais, et al., *Quasi-stationary distributions and population processes*, Probability Surveys, 9 (2012), pp. 340–410.

[27] M. Mihoubi, *Bell polynomials and binomial type sequences*, Discrete Math., 308 (2008), pp. 2450–2459, doi:10.1016/j.disc.2007.05.010, https://doi.org/10.1016/j.disc.2007.05.010.

[28] V. Y. Pan, *Fast approximate computations with cauchy matrices, polynomials and rational functions*, in International Computer Science Symposium in Russia, Springer, 2014, pp. 287–299.

[29] V. Rokhlin, *Rapid solution of integral equations of classical potential theory*, Journal of computational physics, 60 (1985), pp. 187–207.

[30] E. Seneta, *Non-negative matrices and Markov chains*, Springer Series in Statistics, Springer, New York, 2006. Revised reprint of the second (1981) edition [Springer-Verlag, New York; MR0719544].

[31] G. Starke, *Near-circularity for the rational Zolotarev problem in the complex plane*, J. Approx. Theory, 70 (1992), pp. 115–130, doi:10.1016/0021-9045(92)90059-W, https://doi.org/10.1016/0021-9045(92)90059-W.

[32] A. Townsend and H. Wilber, *On the singular values of matrices with high displacement rank*, Linear Algebra and its Applications, 548 (2018), pp. 19–41.

[33] L. N. Trefethen and J. A. C. Weideman, *The exponentially convergent trapezoidal rule*, SIAM Rev., 56 (2014), pp. 385–458, doi:10.1137/130932132, http://dx.doi.org/10.1137/130932132.

[34] E. A. van Doorn and P. K. Pollett, *Quasi-stationary distributions for discrete-state models*, European Journal of Operational Research, 230 (2013), pp. 1–14.

[35] W. Wang and T. Wang, *General identities on Bell polynomials*, Comput. Math. Appl., 58 (2009), pp. 104–118, doi:10.1016/j.camwa.2009.03.093, https://doi.org/10.1016/j.camwa.2009.03.093.

[36] A. M. Yaglom, *Certain limit theorems of the theory of branching random processes*, in Doklady Akad. Nauk SSSR (NS), vol. 56, 1947, p. 3.