Symposium on Theoretical Aspects of Computer Science 2008 (Bordeaux), pp. 289-300
www.stacs-conf.org

CONVERGENCE THRESHOLDS OF NEWTON'S METHOD FOR MONOTONE POLYNOMIAL EQUATIONS

JAVIER ESPARZA, STEFAN KIEFER, AND MICHAEL LUTTENBERGER

Institut für Informatik
Technische Universität München, Germany
E-mail address: {esparza,kiefer,luttenbe}@in.tum.de

Abstract. Monotone systems of polynomial equations (MSPEs) are systems of fixed-point equations $X_1 = f_1(X_1, \ldots, X_n), \ldots, X_n = f_n(X_1, \ldots, X_n)$ where each f_i is a polynomial with positive real coefficients. The question of computing the least non-negative solution of a given MSPE $\boldsymbol{X} = \boldsymbol{f}(\boldsymbol{X})$ arises naturally in the analysis of stochastic models such as stochastic context-free grammars, probabilistic pushdown automata, and back-button processes. Etessami and Yannakakis have recently adapted Newton's iterative method to MSPEs. In a previous paper we have proved the existence of a threshold $k_{\boldsymbol{f}}$ for strongly connected MSPEs, such that after $k_{\boldsymbol{f}}$ iterations of Newton's method each new iteration computes at least 1 new bit of the solution. However, the proof was purely existential. In this paper we give an upper bound for $k_{\boldsymbol{f}}$ as a function of the minimal component of the least fixed-point $\mu\boldsymbol{f}$ of $\boldsymbol{f}(\boldsymbol{X})$. Using this result we show that $k_{\boldsymbol{f}}$ is at most single exponential resp. linear for strongly connected MSPEs derived from probabilistic pushdown automata resp. from back-button processes. Further, we prove the existence of a threshold for arbitrary MSPEs after which each new iteration computes at least $1/w2^h$ new bits of the solution, where w and h are the width and height of the DAG of strongly connected components.

1. Introduction

A *monotone system of polynomial equations* (MSPE for short) has the form

$$\begin{aligned} X_1 &= f_1(X_1, \ldots, X_n) \\ &\vdots \\ X_n &= f_n(X_1, \ldots, X_n) \end{aligned}$$

where $f_1, \ldots, f_n$ are polynomials with *positive* real coefficients. In vector form we denote an MSPE by $\boldsymbol{X} = \boldsymbol{f}(\boldsymbol{X})$. We call MSPEs "monotone" because $\boldsymbol{x} \leq \boldsymbol{x'}$ implies $\boldsymbol{f}(\boldsymbol{x}) \leq \boldsymbol{f}(\boldsymbol{x'})$

1998 ACM Subject Classification: G.1.5, Mathematics of Computing, Numerical Analysis.
Key words and phrases: Newton's Method, Fixed-Point Equations, Formal Verification of Software, Probabilistic Pushdown Systems.
This work was supported by the project *Algorithms for Software Model Checking* of the *Deutsche Forschungsgemeinschaft (DFG)*. Part of this work was done at the Universität Stuttgart.

for every $\boldsymbol{x}, \boldsymbol{x}' \in \mathbb{R}^n_{\geq 0}$. MSPEs appear naturally in the analysis of many stochastic models, such as context-free grammars (with numerous applications to natural language processing [19, 15], and computational biology [21, 4, 3, 17]), probabilistic programs with procedures [6, 2, 10, 8, 7, 9, 11], and web-surfing models with back buttons [13, 14].

By Kleene's theorem, a feasible MSPE $\boldsymbol{X} = \boldsymbol{f}(\boldsymbol{X})$ (i.e., an MSPE with at least one solution) has a least solution $\mu\boldsymbol{f}$; this solution can be irrational and non-expressible by radicals. Given an MSPE and a vector $\boldsymbol{v}$ encoded in binary, the problem whether $\mu\boldsymbol{f} \leq \boldsymbol{v}$ holds is in PSPACE and at least as hard as the SQUARE-ROOT-SUM problem, a well-known problem of computational geometry (see [10, 12] for more details).

For the applications mentioned above the most important question is the efficient numerical approximation of the least solution. Finding the least solution of a feasible system $\boldsymbol{X} = \boldsymbol{f}(\boldsymbol{X})$ amounts to finding the least solution of $\boldsymbol{F}(\boldsymbol{X}) = \boldsymbol{0}$ for $\boldsymbol{F}(\boldsymbol{X}) = \boldsymbol{f}(\boldsymbol{X}) - \boldsymbol{X}$. For this we can apply (the multivariate version of) *Newton's method* [20]: starting at some $\boldsymbol{x}^{(0)} \in \mathbb{R}^n$ (we use uppercase to denote variables and lowercase to denote values), compute the sequence

$$\boldsymbol{x}^{(k+1)} := \boldsymbol{x}^{(k)} - (\boldsymbol{F}'(\boldsymbol{x}^{(k)}))^{-1}\boldsymbol{F}(\boldsymbol{x}^{(k)})$$

where $\boldsymbol{F}'(\boldsymbol{X})$ is the Jacobian matrix of partial derivatives.

While in general the method may not even be defined ($\boldsymbol{F}'(\boldsymbol{x}^{(k)})$ may be singular for some k), Etessami and Yannakakis proved in [10, 12] that this is not the case for the *Decomposed Newton's Method (DNM)*, that decomposes the MSPE into *strongly connected components* (SCCs) and applies Newton's method to them in a bottom-up fashion[1].

The results of [10, 12] provide no information on the number of iterations needed to compute i *valid bits* of $\mu\boldsymbol{f}$, i.e., to compute a vector $\boldsymbol{\nu}$ such that $\left|\mu\boldsymbol{f}_j - \nu_j\right| / \left|\mu\boldsymbol{f}_j\right| \leq 2^{-i}$ for every $1 \leq j \leq n$. In a former paper [16] we have obtained a first positive result on this problem. We have proved that for every strongly connected MSPE $\boldsymbol{X} = \boldsymbol{f}(\boldsymbol{X})$ there exists a threshold $k_{\boldsymbol{f}}$ such that for every $i \geq 0$ the $(k_{\boldsymbol{f}} + i)$-th iteration of Newton's method has at least i valid bits of $\mu\boldsymbol{f}$. So, loosely speaking, after $k_{\boldsymbol{f}}$ iterations DNM is guaranteed to compute at least 1 new bit of the solution per iteration; we say that DNM converges *linearly with rate 1.*

The problem with this result is that its proof provides no information on $k_{\boldsymbol{f}}$ other than its existence. In this paper we show that the threshold $k_{\boldsymbol{f}}$ can be chosen as

$$k_{\boldsymbol{f}} = 3n^2m + 2n^2\left|\log \mu_{\min}\right|$$

where n is the number of equations of the MSPE, m is such that all coefficients of the MSPE can be given as ratios of m-bit integers, and $\mu_{\min}$ is the minimal component of the least solution $\mu\boldsymbol{f}$.

It can be objected that $k_{\boldsymbol{f}}$ depends on $\mu\boldsymbol{f}$, which is precisely what Newton's method should compute. However, for MSPEs coming from stochastic models, such as the ones listed above, we can do far better. The following observations and results help to deal with $\mu_{\min}$:

- We obtain a syntactic bound on $\mu_{\min}$ for probabilistic programs with procedures (having stochastic context-free grammars and back-button stochastic processes as special instances) and prove that in this case $k_{\boldsymbol{f}} \leq n2^{n+2}m$.

[1] A subset of variables and their associated equations form an SCC, if the value of any variable in the subset influences the value of all variables in the subset, see Section 2 for details.

- We show that if every procedure has a non-zero probability of terminating, then $k_{\boldsymbol{f}} \leq 3nm$. This condition always holds in the special case of back-button processes [13, 14]. Hence, our result shows that i valid bits can be computed in time $\mathcal{O}((nm+i)\cdot n^3)$ in the unit cost model of Blum, Shub and Smale [1], where each single arithmetic operation over the reals can be carried out exactly and in constant time. It was proved in [13, 14] by a reduction to a semidefinite programming problem that i valid bits can be computed in poly(i,n,m)-time in the classical (Turing-machine based) computation model. We do not improve this result, because we do not have a proof that round-off errors (which are inevitable on Turing-machine based models) do not crucially affect the convergence of Newton's method. But our result sheds light on the convergence of a practical method to compute $\mu\boldsymbol{f}$.
- Finally, since $\boldsymbol{x}^{(k)} \leq \boldsymbol{x}^{(k+1)} \leq \mu\boldsymbol{f}$ holds for every $k \geq 0$, as Newton's method proceeds it provides better and better lower bounds for $\mu_{\min}$ and thus for $k_{\boldsymbol{f}}$. In the paper we exhibit a MSPE for which, using this fact and our theorem, we can prove that no component of the solution reaches the value 1. This cannot be proved by just computing more iterations, no matter how many.

The paper contains two further results concerning non-strongly-connected MSPEs: Firstly, we show that DNM still converges linearly even if the MSPE has more than one SCC, albeit the convergence rate is poorer. Secondly, we prove that Newton's method is well-defined also for non-strongly-connected MSPEs. Thus, it is not necessary to decompose an MSPE into its SCCs – although decomposing the MSPE may be preferred for efficiency reasons.

The paper is structured as follows. In Section 2 we state preliminaries and give some background on Newton's method applied to MSPEs. Sections 3, 5, and 6 contain the three results of the paper. Section 4 contains applications of our main result. We conclude in Section 7. Missing proofs can be found in a technical report [5].

2. Preliminaries

In this section we introduce our notation and formalize the concepts mentioned in the introduction.

2.1. Notation

$\mathbb{R}$ and $\mathbb{N}$ denote the sets of real, respectively natural numbers. We assume $0 \in \mathbb{N}$. $\mathbb{R}^n$ denotes the set of n-dimensional real valued *column* vectors and $\mathbb{R}^n_{\geq 0}$ the subset of vectors with non-negative components. We use bold letters for vectors, e.g. $\boldsymbol{x} \in \mathbb{R}^n$, where we assume that $\boldsymbol{x}$ has the components $x_1, \ldots, x_n$. Similarly, the i^{th} component of a function $\boldsymbol{f} : \mathbb{R}^n \to \mathbb{R}^n$ is denoted by f_i.

$\mathbb{R}^{m\times n}$ denotes the set of matrices having m rows and n columns. The transpose of a vector or matrix is indicated by the superscript $^\top$. The identity matrix of $\mathbb{R}^{n\times n}$ is denoted by Id.

The *formal Neumann series* of $A \in \mathbb{R}^{n\times n}$ is defined by $A^* = \sum_{k\in\mathbb{N}} A^k$. It is well-known that A^* exists if and only if the spectral radius of A is less than 1, i.e. $\max\{|\lambda| \mid \mathbb{C} \ni \lambda \text{ is an eigenvalue of } A\} < 1$. If A^* exists, we have $A^* = (\mathrm{Id} - A)^{-1}$.

The partial order $\leq$ on $\mathbb{R}^n$ is defined as usual by setting $\boldsymbol{x} \leq \boldsymbol{y}$ if $x_i \leq y_i$ for all $1 \leq i \leq n$. By $\boldsymbol{x} < \boldsymbol{y}$ we mean $\boldsymbol{x} \leq \boldsymbol{y}$ and $\boldsymbol{x} \neq \boldsymbol{y}$. Finally, we write $\boldsymbol{x} \prec \boldsymbol{y}$ if $x_i < y_i$ in every component.

We use $X_1, \ldots, X_n$ as variable identifiers and arrange them into the vector $\boldsymbol{X}$. In the following n always denotes the number of variables, i.e. the dimension of $\boldsymbol{X}$. While $\boldsymbol{x}, \boldsymbol{y}, \ldots$ denote arbitrary elements in $\mathbb{R}^n$, resp. $\mathbb{R}^n_{\geq 0}$, we write $\boldsymbol{X}$ if we want to emphasize that a function is given w.r.t. these variables. Hence, $\boldsymbol{f}(\boldsymbol{X})$ represents the function itself, whereas $\boldsymbol{f}(\boldsymbol{x})$ denotes its value for $\boldsymbol{x} \in \mathbb{R}^n$.

If Y is a set of variables and $\boldsymbol{x}$ a vector, then by $\boldsymbol{x}_Y$ we mean the vector obtained by restricting $\boldsymbol{x}$ to the components in Y.

The *Jacobian* of a differentiable function $\boldsymbol{f}(\boldsymbol{X})$ with $\boldsymbol{f} : \mathbb{R}^n \to \mathbb{R}^m$ is the matrix $\boldsymbol{f}'(\boldsymbol{X})$ given by

$$\boldsymbol{f}'(\boldsymbol{X}) = \begin{pmatrix} \frac{\partial f_1}{\partial X_1} & \cdots & \frac{\partial f_1}{\partial X_n} \\ \vdots & & \vdots \\ \frac{\partial f_m}{\partial X_1} & \cdots & \frac{\partial f_m}{\partial X_n} \end{pmatrix} .$$

2.2. Monotone Systems of Polynomials

Definition 2.1. A function $\boldsymbol{f}(\boldsymbol{X})$ with $\boldsymbol{f} : \mathbb{R}^n_{\geq 0} \to \mathbb{R}^n_{\geq 0}$ is a *monotone system of polynomials (MSP)*, if every component $f_i(\boldsymbol{X})$ is a polynomial in the variables $X_1, \ldots, X_n$ with coefficients in $\mathbb{R}_{\geq 0}$. We call an MSP $\boldsymbol{f}(\boldsymbol{X})$ *feasible* if $\boldsymbol{y} = \boldsymbol{f}(\boldsymbol{y})$ for some $\boldsymbol{y} \in \mathbb{R}^n_{\geq 0}$.

Fact 2.2. *Every MSP $\boldsymbol{f}$ is monotone on $\mathbb{R}^n_{\geq 0}$, i.e. for $\boldsymbol{0} \leq \boldsymbol{x} \leq \boldsymbol{y}$ we have $\boldsymbol{f}(\boldsymbol{x}) \leq \boldsymbol{f}(\boldsymbol{y})$.*

Since every MSP is continuous, Kleene's fixed-point theorem (see e.g. [18]) applies.

Theorem 2.3 (Kleene's fixed-point theorem)**.** *Every feasible MSP $\boldsymbol{f}(\boldsymbol{X})$ has a least fixed point $\mu\boldsymbol{f}$ in $\mathbb{R}^n_{\geq 0}$ i.e., $\mu\boldsymbol{f} = \boldsymbol{f}(\mu\boldsymbol{f})$ and, in addition, $\boldsymbol{y} = \boldsymbol{f}(\boldsymbol{y})$ implies $\mu\boldsymbol{f} \leq \boldsymbol{y}$. Moreover, the sequence $(\boldsymbol{\kappa}^{(k)}_{\boldsymbol{f}})_{k\in\mathbb{N}}$ with $\boldsymbol{\kappa}^{(0)}_{\boldsymbol{f}} := \boldsymbol{0}$, and $\boldsymbol{\kappa}^{(k+1)}_{\boldsymbol{f}} := \boldsymbol{f}(\boldsymbol{\kappa}^{(k)}_{\boldsymbol{f}}) = \boldsymbol{f}^{k+1}(\boldsymbol{0})$ is monotonically increasing with respect to $\leq$ (i.e. $\boldsymbol{\kappa}^{(k)}_{\boldsymbol{f}} \leq \boldsymbol{\kappa}^{(k+1)}_{\boldsymbol{f}}$) and converges to $\mu\boldsymbol{f}$.*

In the following we call $(\boldsymbol{\kappa}^{(k)}_{\boldsymbol{f}})_{k\in\mathbb{N}}$ the *Kleene sequence* of $\boldsymbol{f}(\boldsymbol{X})$, and drop the subscript whenever $\boldsymbol{f}$ is clear from the context. Similarly, we sometimes write $\boldsymbol{\mu}$ instead of $\mu\boldsymbol{f}$.

A variable X_i of an MSP $\boldsymbol{f}(\boldsymbol{X})$ is *productive* if $\kappa^{(k)}_i > 0$ for some $k \in \mathbb{N}$. An MSP is *clean* if all its variables are productive. It is easy to see that $\kappa^{(n)}_i = 0$ implies $\kappa^{(k)}_i = 0$ for all $k \in \mathbb{N}$. As for context-free grammars we can determine all productive variables in time linear in the size of $\boldsymbol{f}$.

Notation 2.4. In the following, we always assume that an MSP $\boldsymbol{f}$ is clean and feasible. I.e., whenever we write "MSP", we mean "clean and feasible MSP", unless explicitly stated otherwise.

For the formal definition of the *Decomposed Newton's Method (DNM)* (see also Section 1) we need the notion of *dependence* between variables.

Definition 2.5. Let $\boldsymbol{f}(\boldsymbol{X})$ be an MSP. X_i *depends directly* on X_k, denoted by $X_i \trianglelefteq X_k$, if $\frac{\partial f_i}{\partial X_k}(\boldsymbol{X})$ is not the zero-polynomial. X_i *depends* on X_k if $X_i \trianglelefteq^* X_k$, where $\trianglelefteq^*$ is the reflexive transitive closure of $\trianglelefteq$. An MSP is *strongly connected* (short: an *scMSP*) if all its variables depend on each other.

Any MSP can be decomposed into strongly connected components (SCCs), where an SCC S is a maximal set of variables such that each variable in S depends on each other variable in S. The following result for strongly connected MSPs was proved in [10, 12]:

Theorem 2.6. *Let $\boldsymbol{f}(\boldsymbol{X})$ be an scMSP and define the Newton operator $\mathcal{N}_{\boldsymbol{f}}$ as follows*

$$\mathcal{N}_{\boldsymbol{f}}(\boldsymbol{X}) = \boldsymbol{X} + (\mathrm{Id} - \boldsymbol{f}'(\boldsymbol{X}))^{-1}(\boldsymbol{f}(\boldsymbol{X}) - \boldsymbol{X}) \ .$$

We have: (1) $\mathcal{N}_{\boldsymbol{f}}(\boldsymbol{x})$ is defined for all $\boldsymbol{0} \leq \boldsymbol{x} \prec \mu\boldsymbol{f}$ (i.e., $(\mathrm{Id} - \boldsymbol{f}'(\boldsymbol{x}))^{-1}$ exists). Moreover, $\boldsymbol{f}'(\boldsymbol{x})^* = \sum_{k\in\mathbb{N}} \boldsymbol{f}'(\boldsymbol{x})^k$ exists for all $\boldsymbol{0} \leq \boldsymbol{x} \prec \mu\boldsymbol{f}$, and so $\mathcal{N}_{\boldsymbol{f}}(\boldsymbol{X}) = \boldsymbol{X} + \boldsymbol{f}'(\boldsymbol{X})^*(\boldsymbol{f}(\boldsymbol{X}) - \boldsymbol{X})$. (2) The Newton sequence $(\boldsymbol{\nu}_{\boldsymbol{f}}^{(k)})_{k\in\mathbb{N}}$ with $\boldsymbol{\nu}^{(k)} = \mathcal{N}_{\boldsymbol{f}}^k(\boldsymbol{0})$ is monotonically increasing, bounded from above by $\mu\boldsymbol{f}$ (i.e. $\boldsymbol{\nu}^{(k)} \leq \boldsymbol{f}(\boldsymbol{\nu}^{(k)}) \leq \boldsymbol{\nu}^{(k+1)} \prec \mu\boldsymbol{f}$), and converges to $\mu\boldsymbol{f}$.

DNM works by substituting the variables of lower SCCs by corresponding Newton approximations that were obtained earlier.

3. A Threshold for scMSPs

In this section we obtain a threshold after which DNM is guaranteed to converge linearly with rate 1.

We showed in [16] that for worst-case results on the convergence of Newton's method it is enough to consider *quadratic* MSPs, i.e., MSPs whose monomials have degree at most 2. The reason is that any MSP (resp. scMSP) $\boldsymbol{f}$ can be transformed into a quadratic MSP (resp. scMSP) $\widetilde{\boldsymbol{f}}$ by introducing auxiliary variables. This transformation is very similar to the transformation of a context-free grammar into Chomsky normal form. The transformation does not accelerate DNM, i.e., DNM on $\boldsymbol{f}$ is at least as fast (in a formal sense) as DNM on $\widetilde{\boldsymbol{f}}$, and so for a worst-case analysis, it suffices to consider quadratic systems. We refer the reader to [16] for details.

We start by defining the notion of "valid bits".

Definition 3.1. Let $\boldsymbol{f}(\boldsymbol{X})$ be an MSP. A vector $\boldsymbol{\nu}$ has i *valid bits* of the least fixed point $\mu\boldsymbol{f}$ if $\left|\mu\boldsymbol{f}_j - \nu_j\right| / \left|\mu\boldsymbol{f}_j\right| \leq 2^{-i}$ for every $1 \leq j \leq n$.

In the rest of the section we prove the following:

Theorem 3.2. *Let $\boldsymbol{f}(\boldsymbol{X})$ be a quadratic scMSP. Let c_{min} be the smallest nonzero coefficient of $\boldsymbol{f}$ and let μ_{min} and μ_{max} be the minimal and maximal component of $\mu\boldsymbol{f}$, respectively. Let*

$$k_{\boldsymbol{f}} = n \cdot \log\left(\frac{\mu_{max}}{c_{min} \cdot \mu_{min} \cdot \min\{\mu_{min}, 1\}}\right) .$$

Then $\boldsymbol{\nu}^{(\lceil k_{\boldsymbol{f}} \rceil + i)}$ has i valid bits of $\mu\boldsymbol{f}$ for every $i \geq 0$.

Loosely speaking, the theorem states that after $k_{\boldsymbol{f}}$ iterations of Newton's method, every subsequent iteration guarantees at least one more valid bit. It may be objected that $k_{\boldsymbol{f}}$ depends on the least fixed point $\mu\boldsymbol{f}$, which is precisely what Newton's method should compute. However, in the next section we show that there are important classes of MSPs (in fact, those which motivated our investigation), for which bounds on $\mu_{\min}$ can be easily obtained.

The following corollary is weaker than Theorem 3.2, but less technical in that it avoids a dependence on $\mu_{\max}$ and $c_{\min}$.

Corollary 3.3. *Let $\boldsymbol{f}(\boldsymbol{X})$ be a quadratic scMSP of dimension n whose coefficients are given as ratios of m-bit integers. Let μ_{min} be the minimal component of $\mu\boldsymbol{f}$. Let $k_{\boldsymbol{f}} = 3n^2m + 2n^2\,|\log \mu_{min}|$. Then $\boldsymbol{\nu}^{(\lceil k_{\boldsymbol{f}} \rceil + i)}$ has at least i valid bits of $\mu\boldsymbol{f}$ for every $i \geq 0$.*

Corollary 3.3 follows from Theorem 3.2 by a suitable bound on $\mu_{\max}$ in terms of $c_{\min}$ and $\mu_{\min}$ [5] (notice that, since $c_{\min}$ is the quotient of two m-bit integers, we have $c_{\min} \geq 1/2^m$).

In the rest of the section we sketch the proof of Theorem 3.2. The proof makes crucial use of vectors $\boldsymbol{d} \succ \boldsymbol{0}$ such that $\boldsymbol{d} \geq \boldsymbol{f}'(\mu\boldsymbol{f})\boldsymbol{d}$. We call a vector satisfying these two conditions a *cone vector of* $\boldsymbol{f}$ or, when $\boldsymbol{f}$ is clear from the context, just a cone vector.

In a previous paper we have shown that if the matrix $(\mathrm{Id} - \boldsymbol{f}'(\mu\boldsymbol{f}))$ is singular, then $\boldsymbol{f}$ has a cone vector ([16], Lemmata 4 and 8). As a first step towards the proof of Theorem 3.2 we show the following stronger proposition.

Proposition 3.4. *Any scMSP has a cone vector.*

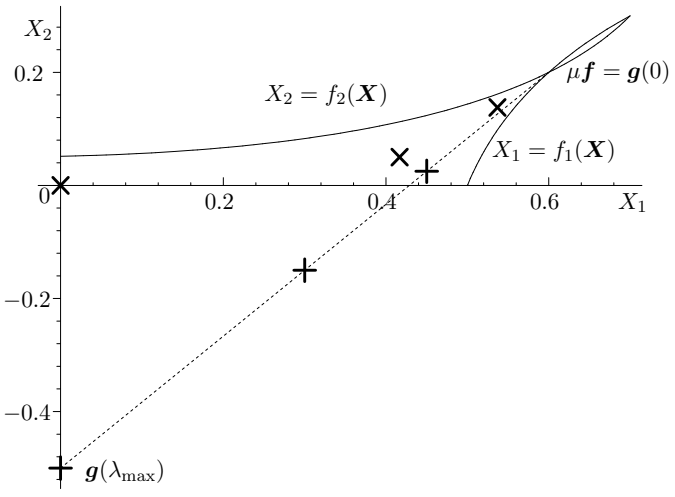
To a cone vector $\boldsymbol{d} = (d_1, \ldots, d_n)$ we associate two parameters, namely the maximum and the minimum of the ratios $\mu\boldsymbol{f}_1/d_1, \mu\boldsymbol{f}_2/d_2, \ldots, \mu\boldsymbol{f}_n/d_n$, which we denote by $\lambda_{\max}$ and $\lambda_{\min}$, respectively. The second step consists of showing (Proposition 3.6) that given a cone vector $\boldsymbol{d}$, the threshold $k_{\boldsymbol{f},\boldsymbol{d}} = \log(\lambda_{\max}/\lambda_{\min})$ satisfies the same property as $k_{\boldsymbol{f}}$ in Theorem 3.2, i.e., $\boldsymbol{\nu}^{(\lceil k_{\boldsymbol{f},\boldsymbol{d}} \rceil + i)}$ has i valid bits of $\mu\boldsymbol{f}$ for every $i \geq 0$. This follows rather easily from the following fundamental property of cone vectors: a cone vector leads to an upper bound on the error of Newton's method.

Lemma 3.5. *Let $\boldsymbol{d}$ be a cone vector of an MSP $\boldsymbol{f}$ and let $\lambda_{max} = \max\{\frac{\mu\boldsymbol{f}_i}{d_i}\}$. Then*

$$\mu\boldsymbol{f} - \boldsymbol{\nu}^{(k)} \leq 2^{-k}\lambda_{max}\,\boldsymbol{d}.$$

Proof Idea. Consider the ray $\boldsymbol{g}(t) = \mu\boldsymbol{f} - t\boldsymbol{d}$ starting in $\mu\boldsymbol{f}$ and headed in the direction $-\boldsymbol{d}$ (the dashed line in the picture below). It is easy to see that $\boldsymbol{g}(\lambda_{\max})$ is the intersection of $\boldsymbol{g}$ with an axis which is located farthest from $\mu\boldsymbol{f}$. One can then prove $\boldsymbol{g}(\frac{1}{2}\lambda_{\max}) \leq \boldsymbol{\nu}^{(1)}$, where $\boldsymbol{g}(\frac{1}{2}\lambda_{\max})$ is the point of the ray equidistant from $\boldsymbol{g}(\lambda_{\max})$ and $\mu\boldsymbol{f}$. By repeated application of this argument one obtains $\boldsymbol{g}(2^{-k}\lambda_{\max}) \leq \boldsymbol{\nu}^{(k)}$ for all $k \in \mathbb{N}$.

The following picture shows the Newton iterates $\boldsymbol{\nu}^{(k)}$ for $0 \leq k \leq 2$ (shape: $\times$) and the corresponding points $\boldsymbol{g}(2^{-k}\lambda_{\max})$ (shape: $+$) located on the ray $\boldsymbol{g}$. Notice that $\boldsymbol{\nu}^{(k)} \geq \boldsymbol{g}(2^{-k}\lambda_{\max})$. ∎



Now we easily obtain:

Proposition 3.6. *Let $\boldsymbol{f}(\boldsymbol{X})$ be an scMSP and let $\boldsymbol{d}$ be a cone vector of $\boldsymbol{f}$. Let $k_{\boldsymbol{f},\boldsymbol{d}} = \log \frac{\lambda_{max}}{\lambda_{min}}$, where $\lambda_{max} = \max_j \frac{\mu \boldsymbol{f}_j}{d_j}$ and $\lambda_{min} = \min_j \frac{\mu \boldsymbol{f}_j}{d_j}$. Then $\boldsymbol{\nu}^{(\lceil k_{\boldsymbol{f},\boldsymbol{d}} \rceil + i)}$ has at least i valid bits of $\mu \boldsymbol{f}$ for every $i \geq 0$.*

We now proceed to the third and final step. We have the problem that $k_{\boldsymbol{f},\boldsymbol{d}}$ depends on the cone vector $\boldsymbol{d}$, about which we only know that it exists (Proposition 3.4). We now sketch how to obtain the threshold $k_{\boldsymbol{f}}$ claimed in Theorem 3.2, which is independent of any cone vectors.

Consider Proposition 3.6 and let $\lambda_{\max} = \frac{\mu \boldsymbol{f}_i}{d_i}$ and $\lambda_{\min} = \frac{\mu \boldsymbol{f}_j}{d_j}$. We have $k_{\boldsymbol{f},\boldsymbol{d}} = \log\left(\frac{d_j}{d_i} \cdot \frac{\mu \boldsymbol{f}_i}{\mu \boldsymbol{f}_j}\right)$. The idea is to bound $k_{\boldsymbol{f},\boldsymbol{d}}$ in terms of $c_{\min}$. We show that if $k_{\boldsymbol{f},\boldsymbol{d}}$ is very large, then there must be variables X, Y such that X depends on Y only via a monomial that has a very small coefficient, which implies that $c_{\min}$ is very small.

4. Stochastic Models

As mentioned in the introduction, several problems concerning stochastic models can be reduced to problems about the least solution $\mu \boldsymbol{f}$ of an MSPE $\boldsymbol{f}$. In these cases, $\mu \boldsymbol{f}$ is a vector of probabilities, and so $\mu_{\max} \leq 1$. Moreover, we can obtain information on $\mu_{\min}$, which leads to bounds on the threshold $k_{\boldsymbol{f}}$.

4.1. Probabilistic Pushdown Automata

Our study of MSPs was initially motivated by the verification of probabilistic pushdown automata. A *probabilistic pushdown automaton (pPDA)* is a tuple $\mathcal{P} = (Q, \Gamma, \delta, Prob)$ where Q is a finite set of *control states*, Γ is a finite *stack alphabet*, $\delta \subseteq Q \times \Gamma \times Q \times \Gamma^*$ is a finite *transition relation* (we write $pX \hookrightarrow q\alpha$ instead of $(p, X, q, \alpha) \in \delta$), and *Prob* is a function which to each transition $pX \hookrightarrow q\alpha$ assigns its probability $Prob(pX \hookrightarrow q\alpha) \in (0, 1]$ so that for all $p \in Q$ and $X \in \Gamma$ we have $\sum_{pX \hookrightarrow q\alpha} Prob(pX \hookrightarrow q\alpha) = 1$. We write $pX \overset{x}{\hookrightarrow} q\alpha$ instead of $Prob(pX \hookrightarrow q\alpha) = x$. A *configuration* of $\mathcal{P}$ is a pair qw, where q is a control state and $w \in \Gamma^*$ is a *stack content*. A probabilistic pushdown automaton $\mathcal{P}$ naturally induces a possibly infinite Markov chain with the configurations as states and transitions given by: $pX\beta \overset{x}{\hookrightarrow} q\alpha\beta$ for every $\beta \in \Gamma^*$ iff $pX \overset{x}{\hookrightarrow} q\alpha$. We assume w.l.o.g. that if $pX \overset{x}{\hookrightarrow} q\alpha$ is a transition then $|\alpha| \leq 2$.

pPDAs and the equivalent model of recursive Markov chains have been very thoroughly studied [6, 2, 10, 8, 7, 9, 11]. These papers have shown that the key to the analysis of pPDAs are the *termination probabilities* $[pXq]$, where p and q are states, and X is a stack letter, defined as follows (see e.g. [6] for a more formal definition): $[pXq]$ is the probability that, starting at the configuration pX, the pPDA eventually reaches the configuration $q\varepsilon$ (empty stack). It is not difficult to show that the vector of termination probabilities is the least fixed point of the MSPE containing the equation

$$[pXq] = \sum_{pX \overset{x}{\hookrightarrow} rYZ} x \cdot \sum_{t \in Q} [rYt] \cdot [tZq] \quad + \quad \sum_{pX \overset{x}{\hookrightarrow} rY} x \cdot [rYq] \quad + \quad \sum_{pX \overset{x}{\hookrightarrow} q\varepsilon} x$$

for each triple (p, X, q). Call this quadratic MSPE the *termination MSPE* of the pPDA (we assume that termination MSPEs are clean, and it is easy to see that they are always

feasible). We immediately have that if $\boldsymbol{X} = \boldsymbol{f}(\boldsymbol{X})$ is a termination MSP, then $\mu_{\max} \leq 1$. We also obtain a lower bound on $\mu_{\min}$:

Lemma 4.1. *Let $\boldsymbol{X} = \boldsymbol{f}(\boldsymbol{X})$ be a termination MSPE with n variables. Then $\mu_{min} \geq c_{min}^{(2^{n+1}-1)}$.*

Together with Theorem 3.2 we get the following exponential bound for $k_{\boldsymbol{f}}$.

Proposition 4.2. *Let $\boldsymbol{f}$ be a strongly connected termination MSP with n variables and whose coefficients are expressed as ratios of m-bit numbers. Then $k_{\boldsymbol{f}} \leq n2^{n+2}m$.*

We conjecture that there is a lower bound on $k_{\boldsymbol{f}}$ which is exponential in n for the following reason. We know a family $(\boldsymbol{f}^{(n)})_{n=1,3,5,\ldots}$ of strongly connected MSPs with n variables and irrational coefficients such that $c_{\min}^{(n)} = \frac{1}{4}$ for all n and $\mu_{\min}^{(n)}$ is double-exponentially small in n. Experiments suggest that $\Theta(2^n)$ iterations are needed for the first bit of $\mu\boldsymbol{f}^{(n)}$, but we do not have a proof.

4.2. Strict pPDAs and Back-Button Processes

A pPDA is *strict* if for every $pX \in Q \times \Gamma$ the transition relation contains a pop-rule $pX \overset{x}{\hookrightarrow} q\epsilon$ for some $q \in Q$ and some $x > 0$. Essentially, strict pPDAs model programs in which every procedure has at least one terminating execution that does not call any other procedure. The termination MSP of a strict pPDA is of the form $\boldsymbol{b}(\boldsymbol{X}, \boldsymbol{X}) + \boldsymbol{l}\boldsymbol{X} + \boldsymbol{c}$ for $\boldsymbol{c} \succ \boldsymbol{0}$. So we have $\mu\boldsymbol{f} \geq \boldsymbol{c}$, which implies $\mu_{\min} \geq c_{\min}$. Together with Theorem 3.2 we get:

Proposition 4.3. *Let $\boldsymbol{f}$ be a strongly connected termination MSP with n variables and whose coefficients are expressed as ratios of m-bit numbers. If $\boldsymbol{f}$ is derived from a strict pPDA, then $k_{\boldsymbol{f}} \leq 3nm$.*

Since in most applications m is small, we obtain an excellent convergence threshold.

In [13, 14] Fagin et al. introduce a special class of strict pPDAs called *back-button processes*: in a back-button process there is only one control state p , and any rule is of the form $pA \overset{b_A}{\hookrightarrow} p\varepsilon$ or $pA \overset{l_{AB}}{\hookrightarrow} pBA$. So the stack corresponds to a path through a finite graph with Γ as set of nodes and edges $A \to B$ for $pA \overset{l_{AB}}{\hookrightarrow} pBA$.

In [13, 14] back-button processes are used to model the behaviour of web-surfers: Γ is the set of web-pages, l_{AB} is the probability that a web-surfer uses a link from page A to page B, and b_A is the probability that the surfer pushes the "back"-button of the web-browser while visiting A. Thus, the termination probability $[pAp]$ is simply the probability that, if A is on top of the stack, A is eventually popped from the stack. The termination probabilities are the least solution of the MSPE consisting of the equations

$$[pAp] \quad = \quad b_A + \sum_{pA \overset{l_{AB}}{\hookrightarrow} pBA} l_{AB}[pBp][pAp] \quad = \quad b_A + [pAp] \sum_{pA \overset{l_{AB}}{\hookrightarrow} pBA} l_{AB}[pBp].$$

4.3. An Example

As an example of application of Theorem 3.2 consider the following scMSPE $\boldsymbol{X} = \boldsymbol{f}(\boldsymbol{X})$.

$$\begin{pmatrix} X_1 \\ X_2 \\ X_3 \end{pmatrix} = \begin{pmatrix} 0.4X_2X_1 + 0.6 \\ 0.3X_1X_2 + 0.4X_3X_2 + 0.3 \\ 0.3X_1X_3 + 0.7 \end{pmatrix}$$

The least solution of the system gives the revocation probabilities of a back-button process with three web-pages. For instance, if the surfer is at page 2 it can choose between following links to pages 1 and 3 with probabilities 0.3 and 0.4, respectively, or pressing the back button with probability 0.3.

We wish to know if any of the revocation probabilities is equal to 1. Performing 14 Newton steps (e.g. with Maple) yields an approximation $\boldsymbol{\nu}^{(14)}$ to the termination probabilities with

$$\begin{pmatrix} 0.98 \\ 0.97 \\ 0.992 \end{pmatrix} \leq \boldsymbol{\nu}^{(14)} \leq \begin{pmatrix} 0.99 \\ 0.98 \\ 0.993 \end{pmatrix} .$$

We have $c_{\min} = 0.3$. In addition, since Newton's method converges to $\mu\boldsymbol{f}$ from below, we know $\mu_{\min} \geq 0.97$. Moreover, $\mu_{\max} \leq 1$, as $\mathbf{1} = \boldsymbol{f}(\mathbf{1})$ and so $\mu\boldsymbol{f} \leq \mathbf{1}$. Hence $k_{\boldsymbol{f}} \leq 3 \cdot \log \frac{1}{0.97 \cdot 0.3 \cdot 0.97} \leq 6$. Theorem 3.2 then implies that $\boldsymbol{\nu}^{(14)}$ has (at least) 8 valid bits of $\mu\boldsymbol{f}$. As $\mu\boldsymbol{f} \leq \mathbf{1}$, the absolute errors are bounded by the relative errors, and since $2^{-8} \leq 0.004$ we know:

$$\mu\boldsymbol{f} \prec \boldsymbol{\nu}^{(14)} + \begin{pmatrix} 2^{-8} \\ 2^{-8} \\ 2^{-8} \end{pmatrix} \prec \begin{pmatrix} 0.994 \\ 0.984 \\ 0.997 \end{pmatrix} \prec \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$$

So Theorem 3.2 gives a proof that all 3 revocation probabilities are strictly smaller than 1.

5. Linear Convergence of the Decomposed Newton's Method

Given a strongly connected MSP $\boldsymbol{f}$, Theorem 3.2 states that, if we have computed $k_{\boldsymbol{f}}$ preparatory iterations of Newton's method, then after i additional iterations we can be sure to have computed at least i bits of $\mu\boldsymbol{f}$. We call this linear convergence with rate 1. Now we show that DNM, which handles non-strongly-connected MSPs, converges linearly as well. We also give an explicit convergence rate.

Let $\boldsymbol{f}(\boldsymbol{X})$ be any quadratic MSP (again we assume *quadratic* MSPs throughout this section), and let $h(\boldsymbol{f})$ denote the height of the DAG of strongly connected components (SCCs). The convergence rate of DNM crucially depends on this height: In the worst case one needs asymptotically $\Theta(2^{h(\boldsymbol{f})})$ iterations in each component per bit, assuming one performs the same number of iterations in each component.

To get a sharper result, we suggest to perform a different number of iterations in each SCC, depending on its *depth*. The depth of an SCC S is the length of the longest path in the DAG of SCCs from S to a top SCC.

In addition, we use the following notation. For a depth t, we denote by $comp(t)$ the set of SCCs of depth t. Furthermore we define $C(t) := \bigcup comp(t)$ and $C_>(t) := \bigcup_{t'>t} C(t')$ and, analogously, $C_<(t)$. We will sometimes write $\boldsymbol{v}_t$ for $\boldsymbol{v}_{C(t)}$ and $\boldsymbol{v}_{>t}$ for $\boldsymbol{v}_{C_>(t)}$ and $\boldsymbol{v}_{<t}$ for $\boldsymbol{v}_{C_<(t)}$, where $\boldsymbol{v}$ is any vector.

Figure 1 shows the Decomposed Newton's Method (DNM) for computing an approximation $\boldsymbol{\nu}$ for $\mu\boldsymbol{f}$, where $\boldsymbol{f}(\boldsymbol{X})$ is any quadratic MSP. The authors of [10] recommend to

run Newton's Method in each SCC S until "approximate solutions for S are considered 'good enough' ". Here we suggest to run Newton's Method in each SCC S for a number of steps that depends (exponentially) on the depth of S and (linearly) on a parameter j that controls the number of iterations (see Figure 1).

function DNM $(\boldsymbol{f}, j)$
/* *The parameter j controls the number of iterations.* */
for t **from** $h(\boldsymbol{f})$ **downto** 0
  forall $S \in comp(t)$ /* *all SCCs S of depth t* */
    $\boldsymbol{\nu}_S := \mathcal{N}^{j\cdot 2^t}_{\boldsymbol{f}_S}(\mathbf{0})$ /* *$j \cdot 2^t$ iterations* */
    /* *apply $\boldsymbol{\nu}_S$ in the depending SCCs* */
    $\boldsymbol{f}_{<t}(\boldsymbol{X}) := \boldsymbol{f}_{<t}(\boldsymbol{X})[\boldsymbol{X}_S/\boldsymbol{\nu}_S]$
return $\boldsymbol{\nu}$

Figure 1: Decomposed Newton's Method (DNM) for computing an approximation $\boldsymbol{\nu}$ of $\mu\boldsymbol{f}$

Recall that $h(\boldsymbol{f})$ was defined as the height of the DAG of SCCs. Similarly we define the width $w(\boldsymbol{f})$ to be $\max_t |comp(t)|$. Notice that $\boldsymbol{f}$ has at most $(h(\boldsymbol{f})+1)\cdot w(\boldsymbol{f})$ SCCs. We have the following bound on the number of iterations run by DNM.

Proposition 5.1. *The function* DNM$(\boldsymbol{f}, j)$ *of Fig. 1 runs at most $j\cdot w(\boldsymbol{f})\cdot 2^{h(\boldsymbol{f})+1}$ iterations of Newton's method.*

We will now analyze the convergence behavior of DNM asymptotically (for large j). Let $\boldsymbol{\Delta}^{(j)}_S$ denote the error in S when running DNM with parameter j, i.e., $\boldsymbol{\Delta}^{(j)}_S := \boldsymbol{\mu}_S - \boldsymbol{\nu}^{(j)}_S$. Observe that the error $\boldsymbol{\Delta}^{(j)}_t$ can be understood as the sum of two errors:

$$\boldsymbol{\Delta}^{(j)}_t = \boldsymbol{\mu}_t - \boldsymbol{\nu}^{(j)}_t = (\boldsymbol{\mu}_t - \widetilde{\boldsymbol{\mu}}^{(j)}_t) + (\widetilde{\boldsymbol{\mu}}^{(j)}_t - \boldsymbol{\nu}^{(j)}_t)\,,$$

where $\widetilde{\boldsymbol{\mu}}^{(j)}_t := \mu\big(\boldsymbol{f}_t(\boldsymbol{X})[\boldsymbol{X}_{>t}/\boldsymbol{\nu}^{(j)}_{>t}]\big)$, i.e., $\widetilde{\boldsymbol{\mu}}^{(j)}_t$ is the least fixed point of $\boldsymbol{f}_t$ after the approximations from the lower SCCs have been applied. So, $\boldsymbol{\Delta}^{(j)}_t$ consists of the *propagation error* $(\boldsymbol{\mu}_t - \widetilde{\boldsymbol{\mu}}^{(j)}_t)$ and the newly inflicted *approximation error* $(\widetilde{\boldsymbol{\mu}}^{(j)}_t - \boldsymbol{\nu}^{(j)}_t)$.

The following lemma, technically non-trivial to prove, gives a bound on the propagation error.

Lemma 5.2 (Propagation error)**.** *Let $\boldsymbol{\nu}_{>t}$ be some approximation of $\boldsymbol{\mu}_{>t}$, i.e., $\mathbf{0} \le \boldsymbol{\nu}_{>t} \le \boldsymbol{\mu}_{>t}$. Let $\widetilde{\boldsymbol{\mu}_t} = \mu\big(\boldsymbol{f}_t(\boldsymbol{X})[\boldsymbol{X}_{>t}/\boldsymbol{\nu}_{>t}]\big)$. Then there is a constant $c > 0$ such that*

$$\|\boldsymbol{\mu}_t - \widetilde{\boldsymbol{\mu}_t}\| \le c \cdot \sqrt{\|\boldsymbol{\mu}_{>t} - \boldsymbol{\nu}_{>t}\|}\,.$$

Intuitively, Lemma 5.2 states that if $\boldsymbol{\nu}_{>t}$ has k valid bits of $\boldsymbol{\mu}_{>t}$, then $\widetilde{\boldsymbol{\mu}_t}$ has roughly $k/2$ valid bits of $\boldsymbol{\mu}_t$. In other words, (at most) one half of the valid bits are lost on each level of the DAG due to the propagation error.

The following theorem assures that after combining the propagation error and the approximation error, DNM still converges linearly.

Theorem 5.3. *Let $\boldsymbol{f}$ be a quadratic MSP. Let $\boldsymbol{\nu}^{(j)}$ denote the result of calling* DNM$(\boldsymbol{f}, j)$ *(see Figure 1). Then there is a $k_{\boldsymbol{f}} \in \mathbb{N}$ such that $\boldsymbol{\nu}^{(k_{\boldsymbol{f}}+i)}$ has at least i valid bits of $\mu\boldsymbol{f}$ for every $i \ge 0$.*

We conclude that increasing i by one gives us asymptotically at least one additional bit in each component and, by Proposition 5.1, costs $w(\boldsymbol{f}) \cdot 2^{h(\boldsymbol{f})+1}$ additional Newton iterations.

In the technical report [5] we give an example that shows that the bound above is essentially optimal in the sense that an exponential (in $h(\boldsymbol{f})$) number of iterations is in general needed to obtain an additional bit.

6. Newton's Method for General MSPs

Etessami and Yannakakis [10] introduced DNM because they could show that the matrix inverses used by Newton's method exist if Newton's method is run on each SCC separately (see Theorem 2.6).

It may be surprising that the matrix inverses used by Newton's method exist even if the MSP is *not* decomposed. More precisely one can show the following theorem, see [5].

Theorem 6.1. *Let $\boldsymbol{f}(\boldsymbol{X})$ be any MSP, not necessarily strongly connected. Let the Newton operator $\mathcal{N}_{\boldsymbol{f}}$ be defined as before:*

$$\mathcal{N}_{\boldsymbol{f}}(\boldsymbol{X}) = \boldsymbol{X} + (\mathrm{Id} - \boldsymbol{f}'(\boldsymbol{X}))^{-1}(\boldsymbol{f}(\boldsymbol{X}) - \boldsymbol{X})$$

Then the Newton sequence $(\boldsymbol{\nu}_{\boldsymbol{f}}^{(k)})_{k\in\mathbb{N}}$ with $\boldsymbol{\nu}^{(k)} = \mathcal{N}_{\boldsymbol{f}}^{k}(\mathbf{0})$ is well-defined (i.e., the matrix inverses exist), monotonically increasing, bounded from above by $\mu\boldsymbol{f}$ (i.e. $\boldsymbol{\nu}^{(k)} \leq \boldsymbol{\nu}^{(k+1)} \prec \mu\boldsymbol{f}$), and converges to $\mu\boldsymbol{f}$.

By exploiting Theorem 5.3 and Theorem 6.1 one can show the following theorem which addresses the convergence *speed* of Newton's Method in general.

Theorem 6.2. *Let $\boldsymbol{f}$ be any quadratic MSP. Then the Newton sequence $(\boldsymbol{\nu}^{(k)})_{k\in\mathbb{N}}$ is well-defined and converges linearly to $\mu\boldsymbol{f}$. More precisely, there is a $k_{\boldsymbol{f}} \in \mathbb{N}$ such that $\boldsymbol{\nu}^{(k_{\boldsymbol{f}}+i\cdot(h(\boldsymbol{f})+1)\cdot 2^{h(\boldsymbol{f})})}$ has at least i valid bits of $\mu\boldsymbol{f}$ for every $i \geq 0$.*

Again, the $2^{h(\boldsymbol{f})}$ factor cannot be avoided in general as shown by an example in [5].

7. Conclusions

We have proved a threshold $k_{\boldsymbol{f}}$ for strongly connected MSPEs. After $k_{\boldsymbol{f}}+i$ Newton iterations we have i bits of accuracy. The threshold $k_{\boldsymbol{f}}$ depends on the representation size of $\boldsymbol{f}$ and on the least solution $\mu\boldsymbol{f}$. Although this latter dependence might seem to be a problem, lower and upper bounds on $\mu\boldsymbol{f}$ can be easily derived for stochastic models (probabilistic programs with procedures, stochastic context-free grammars and back-button processes). In particular, this allows us to show that $k_{\boldsymbol{f}}$ depends linearly on the representation size for back-button processes. We have also shown by means of an example that the threshold $k_{\boldsymbol{f}}$ improves when the number of iterations increases.

In [16] we left the problem whether DNM converges linearly for non-strongly-connected MSPEs open. We have proven that this is the case, although the convergence rate is poorer: if h and w are the height and width of the graph of SCCs of $\boldsymbol{f}$, then there is a threshold $\widetilde{k}_{\boldsymbol{f}}$ such that $\widetilde{k}_{\boldsymbol{f}} + i \cdot w \cdot 2^{h+1}$ iterations of DNM compute at least i valid bits of $\mu\boldsymbol{f}$, where the exponential factor cannot be avoided in general.

Finally, we have shown that the Jacobian of the whole MSPE is guaranteed to exist, whether the MSPE is strongly connected or not.

Acknowledgments.

The authors wish to thank Kousha Etessami and anonymous referees for very valuable comments.

References

[1] L. Blum, M. Shub, and S. Smale. On a theory of computation and complexity over the real numbers: NP-completeness, recursive functions and universal machines. *Bulletin of the Amer. Math. Society*, 21(1):1–46, 1989.

[2] T. Brázdil, A. Kučera, and O. Stražovský. On the decidability of temporal properties of probabilistic pushdown automata. In *Proceedings of STACS'2005*, volume 3404 of *LNCS*, pages 145–157. Springer, 2005.

[3] R.D. Dowell and S.R. Eddy. Evaluation of several lightweight stochastic context-free grammars for RNA secondary structure prediction. *BMC Bioinformatics*, 5(71), 2004.

[4] R. Durbin, S.R. Eddy, A. Krogh, and G.J. Michison. *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press, 1998.

[5] J. Esparza, S. Kiefer, and M. Luttenberger. Convergence thresholds of Newton's method for monotone polynomial equations. Technical report, Technische Universität München, 2007.

[6] J. Esparza, A. Kučera, and R. Mayr. Model-checking probabilistic pushdown automata. In *Proceedings of LICS 2004*, pages 12–21, 2004.

[7] J. Esparza, A. Kučera, and R. Mayr. Quantitative analysis of probabilistic pushdown automata: Expectations and variances. In *Proceedings of LICS 2005*, pages 117–126. IEEE Computer Society Press, 2005.

[8] K. Etessami and M. Yannakakis. Algorithmic verification of recursive probabilistic systems. In *Proceedings of TACAS 2005*, LNCS 3440, pages 253–270. Springer, 2005.

[9] K. Etessami and M. Yannakakis. Checking LTL properties of recursive Markov chains. In *Proceedings of 2nd Int. Conf. on Quantitative Evaluation of Systems (QEST'05)*, 2005.

[10] K. Etessami and M. Yannakakis. Recursive Markov chains, stochastic grammars, and monotone systems of nonlinear equations. In *STACS*, pages 340–352, 2005.

[11] K. Etessami and M. Yannakakis. Recursive Markov decision processes and recursive stochastic games. In *Proceedings of ICALP 2005*, volume 3580 of *LNCS*, pages 891–903. Springer, 2005.

[12] K. Etessami and M. Yannakakis. Recursive Markov chains, stochastic grammars, and monotone systems of nonlinear equations, 2006. Draft journal submission, `http://homepages.inf.ed.ac.uk/kousha/bib_index.html`.

[13] R. Fagin, A.R. Karlin, J. Kleinberg, P. Raghavan, S. Rajagopalan, R. Rubinfeld, M. Sudan, and A. Tomkins. Random walks with "back buttons" (extended abstract). In *STOC*, pages 484–493, 2000.

[14] R. Fagin, A.R. Karlin, J. Kleinberg, P. Raghavan, S. Rajagopalan, R. Rubinfeld, M. Sudan, and A. Tomkins. Random walks with "back buttons". *Annals of Applied Probability*, 11(3):810–862, 2001.

[15] S. Geman and M. Johnson. Probabilistic grammars and their applications, 2002.

[16] S. Kiefer, M. Luttenberger, and J. Esparza. On the convergence of Newton's method for monotone systems of polynomial equations. In *Proceedings of STOC*, pages 217–226. ACM, 2007.

[17] B. Knudsen and J. Hein. Pfold: RNA secondary structure prediction using stochastic context-free grammars. *Nucleic Acids Research*, 31(13):3423–3428, 2003.

[18] W. Kuich. *Handbook of Formal Languages*, volume 1, chapter 9: Semirings and Formal Power Series: Their Relevance to Formal Languages and Automata, pages 609 – 677. Springer, 1997.

[19] C. Manning and H. Schütze. *Foundations of Statistical Natural Language Processing*. MIT Press, 1999.

[20] J.M. Ortega and W.C. Rheinboldt. *Iterative solution of nonlinear equations in several variables*. Academic Press, 1970.

[21] Y. Sakabikara, M. Brown, R. Hughey, I.S. Mian, K. Sjolander, R.C. Underwood, and D. Haussler. Stochastic context-free grammars for tRNA. *Nucleic Acids Research*, 22:5112–5120, 1994.