# INSTITUTO DE COMPUTAcCÃO
## UNIVERSIDADE ESTADUAL DE CAMPINAS

**Generating Asymptotically Non-Terminating Initial Values for Linear Programs**

*Rachid Rebiha*     *Arnaldo V. Moura*
*Nadir Matringe*

The contents of this report are the sole responsibility of the authors.
O conteúdo do presente relatório é de única responsabilidade dos autores.

# Generating Asymptotically Non-Terminating Initial Values for Linear Programs

Rachid Rebiha[*]     Arnaldo Vieira Moura[†]     Nadir Matringe [‡]

Tuesday 3rd June, 2014

## Abstract

We present the notion of *asymptotically non-terminating initial variable values* for linear loop programs. Those values are directly associated to initial variable values for which the corresponding program does not terminate. Our theoretical contributions provide us with powerful computational methods for automatically generating sets of asymptotically non-terminating initial variable values. Such sets are represented symbolically and exactly by a semi-linear space, *e.g.*, characterized by conjunctions and disjunctions of linear equalities and inequalities. Moreover, by taking their complements, we obtain a precise under-approximation of the set of inputs for which the program does terminate. We can then reduce the termination problem of linear programs to the emptiness check of a specific set of asymptotically non-terminating initial variable values. Our *static input data analysis* is not restricted only to programs where the variables are interpreted over the reals. We extend our approach and provide new decidability results for the termination problem of affine integer and rational programs.

# 1   Introduction

Proving termination of while loop programs is necessary for the verification of liveness properties that any well behaved and engineered system, or any safety critical embedded system, must guarantee. Also, generating input data that demonstrates critical

defects and vulnerabilities in programs allows for new looks at these properties. Such crucial *static input data analysis* can be seen as an important trend in the automated verification of loop programs, and is a cornerstone for modern software industry. We could list here many verification approaches that are only practical depending on the facility with which termination can be automatically determined.

The *halting problem* is equivalent to the problem of deciding whether a given program will eventually terminate when running with a given input. The *termination problem* can be stated as follows: given an arbitrary program, decide whether the program eventually halts for every possible input configuration. Both problems are known to be undecidable [1]. As it happens frequently, a program may terminate only for a specific set of input data configurations. The *conditional termination problem* [2] asks for preconditions representing input data that will cause the program to terminate when run with such input data.

Some recent work on automated termination analysis of imperative loop programs has focused on partial decision procedures based on the discovery and synthesis of ranking functions. Such functions map loop variables to well-defined domains where their values decrease further at each iteration of the loop [3, 4]. Several interesting approaches, based on the generation of *linear* ranking functions, have been proposed for loop programs where the guards and the instructions can be expressed in a logic supporting linear arithmetic [5, 6]. For the generation of such functions, there are effective heuristics [7, 4] and, in some cases, there are also complete methods for the synthesis of *linear* ranking functions [8]. On the other hand, there are simple linear terminating loop programs for which there is no linear ranking functions. Concerning decidability results, the work of Tiwari et al. [9] is often cited when treating linear programs over the reals. For linear programs over the rationals and integers, some of those theoretical results have been extended [10]. But the termination problem for *general affine* programs over the integers is left open in [10]. The framework presented in [11] is devoted to approches establishing termination by abstract interpretation of termination semantics. The approach exposed in [2] searches for non-terminating program executions. They first generates lasso-shaped candidate paths (i.e., a loop preceded by a finite program path), and then check each path for non-termination.

The recent literature on *conditional (non-)termination* narrows down to the works presented in [12, 2, 13].The methods proposed in [12] allow for the generation of non-linear preconditions. In [2], the authors derived termination preconditions for simple programs — with only one loop condition — by guessing a ranking function and inferring a supporting assertion. Those approaches are sound but not complete. Also, the interesting approach provided in [13] (focusing mostly on proofs of decidability), consider several systems and models but is restricted to two specific subclasses of linear relations. On the one hand, they consider octagonal relations which do not necessarily represent affine loop semantics. For such relation there method indicates the use of quantifier elimination techniques and computational steps running in exponential

time complexity on the number of variables. On the other hand, they treat restricted subclasses of linear affine relations which must satisfy several restrictions concerning the associated matrix, such as it being diagonalizable with all non-zero eigenvalues of multiplicity one. Using partial termination proofs, the technique proposed in [14] suggests an incremental proof reasonning on the programs. Most directly related work will be discussed in more details (see Sections 8 and 10).

Despite tremendous progress over the years [15, 16, 11, 17, 18, 12, 2, 21], the problem of finding a practical, sound and complete method, *i.e.*, an encoding leading deterministically to an algorithm, for determining (conditional) termination remains very challenging. In this article, we consider linear while loop programs where the loop condition is a conjunction of linear or affine inequalities and the assignments to each of the variables in the loop instruction block are affine or linear forms. In matrix notation, *linear or affine loop programs* will be represented as: while $(Fx > b)$, $\{x := Ax + c\}$, where $A$ and $F$ are matrices, $b$ and $c$ are vectors over the reals, rationals or integers, and $x$ is a vector of variables over $\mathbb{R}$, $\mathbb{Q}$, $\mathbb{N}$ or $\mathbb{Z}$. Automated verification for programs presented in a more complex form can often be reduced to the static analysis of a program expressed in this basic affine form.

We first address the problem of generating input variable values for which a program does not terminate and, conversely, the problem of obtaining the set of terminating inputs for the same program. Initial investigations were reported in [**?**, **?**] where we discussed termination analysis algorithms that ran in polynomial time complexity and the initial results on $ANT$ generation where presented in [**?**]. Subsequent studies considered the set of *asymptotically non-terminating initial variable values* ($ANT$, for short) whose elements are directly related to input values for which the loop does not terminate [22]. In that work we approached the problem of generating the $ANT$ set for a restricted class of linear programs over the reals, with only one loop condition, and where the associated linear forms of the loop lead to diagonalizable systems with no complex eigenvalues. Here, we remove these restrictions. We show how to handle complex eigenvalues, linear affine programs over $\mathbb{R}$, $\mathbb{Q}$, $\mathbb{N}$ or $\mathbb{Z}$, with conjunctions of several loop conditions, and where the system does not have to be diagonalizable. We thus drastically generalize the earlier results in [22]. Further, we introduce new static analysis methods that compute $ANT$ sets in polynomial time, and also yield a set of initial inputs values for which the program does terminate. This attests the innovation of our contributions, *i.e.*, none of the other mentioned works is capable of generating such critical information for non-terminating loops.

We summarize our contributions as follows, with all results rigorously stated and proved:

**Static input data analysis**:
• We introduce the important key concept of an $ANT$ set. Its elements are directly related to initial variables values for which the program does not terminate. Theorems 3.1, 5.1 and 5.2 show the importance of $ANT$ sets. Without loss of generality, we

show that the problem of generating $ANT$ sets for the class of affine programs can be reduced to the computation of $ANT$ sets for specific linear homogeneous programs whose transition matrix has a real spectrum.

• We provide efficient computational methods allowing for the exact computation of $ANT$ sets for linear loop programs. We automatically generate a set of linear equalities and inequalities describing a semi-linear space that symbolically and exactly represents such $ANT$ sets. See Theorem 7.2. Also, an $ANT$ complement set is a precise under-approximation of the set of terminating inputs for the same program. Even if these results are mathematical in nature, they are really easy to apply. In a practical static analysis scenario, one only needs to focus on ready-to-use generic formulas that represent the $ANT$ sets for affine programs. See Eqs. (1), (2), and (3). Such $ANT$ set representations allows for practical computational manipulations — like union, intersection, and emptiness check —, and implementations.

**Static termination analysis**:

• We obtain *necessary and sufficient conditions* for the termination of linear programs. Further, we reduce the problem of termination for linear programs to the emptiness check of the corresponding $ANT$ set. This characterization of terminating linear programs provide us with a deterministic computational procedure to check program termination. Such an algorithm is not present in previous works, such as [9], that discuss the decidability of the termination problem for linear programs over the reals

**Decidability results for the termination problem**:

• By extending our results to affine programs over $\mathbb{Q}$, $\mathbb{N}$ and $\mathbb{Z}$, we obtain new decidability results for the program termination problem. In [10], the termination problem for affine programs over the rationals has been proved to be decidable, and the termination of programs over the integers has been proved to be decidable only in the homogeneous case *i.e.*, with loops over the integers and with only one loop condition of the form while$(bx > 0)\{x := Ax\}$. Here, we successfully address the question left open in [10], namely, we settle the decidability problem for program termination in the case of affine programs over the integers.

**Example 1.1.** (Motivating Example) *Consider the program:*

```
while(x-1/2y-2z>0){
  x:=-20x-9y+75z;
  y:=-140x-55y+504z;
  z:=560x+228y-2011z;}
```

*The initial values of $x$, $y$ and $z$ are represented, respectively, by the parameters $u_1$, $u_2$ and $u_3$. Our prototype outputs the following ANT set:*

```
Locus of ANT:[[u1<-u2+3*u3]]OR[[u1==-u2+3*u3,-u3<u2]]OR
[[u1==4*u3,u2==-u3,0<u3]].
```

− Static input data analysis*: This semi-linear space represents symbolically all asymptotically initial values that are directly associated to initial values for which the program does not terminate. The complement of this set is a precise under-approximation*

*of the set of all initial values for which the program terminates.*
− Termination analysis: *The problem of termination is reduced to the emptiness check of this ANT set.* □

Section 2 introduces key notations, basic results from linear algebra, and formal programs. Section 3 presents the new notion of *asymptotically non-terminating initial values*, and important results for termination analysis. Sections 4 reduces the study of homogeneous programs to the case where the transition matrix has a real spectrum. Section 5 reduces the study of general affine loop programs, with several loop inequality conjunctions, to that of linear homogeneous programs with one loop condition. Section 6 provides new decidability results for the termination problem of integer and rational affine programs. Section 7 presents the ready-to-use formulas representing symbolically and exactly the *ANT* sets for linear homogeneous programs. Section 8 details our computational methods in practice, its algorithm and some experiments. We provide a complete discussion in Section 10. Finally, Section 11 states our conclusions. In the Appendix and in a companion Technical Report [**?**], we give proofs and details about the computational steps in the running examples.

# 2   Linear Algebra and Linear Loop Programs

We recall classical facts from linear algebra. Let $E$ be a real vector space and let $\mathbf{A}$ belong to $End_{\mathbb{R}}(E)$, the space of $\mathbb{R}$-linear maps from $E$ to itself. We denote by $\mathcal{M}(p, q, \mathbb{R})$ the space of $p \times q$ matrices. When $p = q$ we may write $\mathcal{M}(p, \mathbb{R})$. If $B$ is a basis of $E$, we denote by $A_B = Mat_B(\mathbf{A})$ the matrix of $\mathbf{A}$ in $B$ in the space $\mathcal{M}(n, \mathbb{R})$. Let $I_n$ be the identity matrix in $\mathcal{M}(n, \mathbb{R})$, and let $\mathbf{id\_E}$ the identity of $End_{\mathbb{R}}(E)$. We denote by $det(M)$ the determinant of a matrix in $\mathcal{M}(n, \mathbb{R})$. Since $det(A_B)$ is independent of the choice of a basis $B$, we also denote it by $det(\mathbf{A})$, where $A_B = Mat_B(\mathbf{A})$.

**Definition 2.1.** *The characteristic polynomial of $\boldsymbol{A}$ is $\chi_{\boldsymbol{A}}(T) = det(\boldsymbol{A} - T\boldsymbol{id\_E})$. It can be computed as $det(A_B - TI_n)$ for any basis $B$ of $E$. Let $Spec_{\mathbb{R}}(\boldsymbol{A})$ be the set of its real roots, which are the real eigenvalues of $\boldsymbol{A}$. If $\lambda \in Spec_{\mathbb{R}}(\boldsymbol{A})$ has multiplicity $d_\lambda$ as a root of $\chi_{\boldsymbol{A}}$, let $E_\lambda(\boldsymbol{A}) = Ker((\boldsymbol{A} - \lambda\boldsymbol{id\_E})^{d_\lambda})$ be the generalized eigenspace of $\lambda$. It contains the eigenspace $Ker(\boldsymbol{A} - \lambda\boldsymbol{id\_E})$, and its dimension is $d_\lambda$.*

We denote by $E^*$ the space of linear maps from $E$ to $\mathbb{R}$. In the following, we represent linear and affine loop programs in terms of linear forms and their matrix representation. We recall, as it is standard in static program analysis, that a primed symbol $x'$ refers to the next value of $x$ after a transition is taken. First, we present *transition systems* as representations of imperative programs, and *automata* as their computational models.

**Definition 2.2.** *A transition system is given by $\langle x, L, \mathcal{T}, l_0, \Theta \rangle$, where $x = (x_1, ..., x_n)$ is a set of variables, $L$ is a set of locations and $l_0 \in L$ is the initial location. A state is given by an interpretation of the variables in $x$. A transition $\tau \in \mathcal{T}$ is given by a tuple $\langle l_{pre}, l_{post}, q_\tau, \rho_\tau \rangle$, where $l_{pre}$ and $l_{post}$ designate the pre- and post- locations of $\tau$, respectively, and the transition relation $\rho_\tau$ is a first-order assertion over $x \cup x'$. The transition guard $q_\tau$ is a conjunction of inequalities over $x$. $\Theta$ is the initial condition, given as a first-order assertion over $x$. The transition system is said to be* linear *when $\rho_\tau$ is an affine form.*

We will use the following matrix notations to represent loop programs and their transition systems. We also use simple and efficient procedures to captures the effects of sequential linear assignments into simultaneous updates.

**Definition 2.3.** *Let $P = \langle x, l, \mathcal{T} = \langle l, l, q_\tau, \rho_\tau \rangle, l, \Theta \rangle$, with $x = (x_1, ..., x_n)$, be a loop program. We say that $P$ is a* linear loop program *if:*
*• Transition guards are conjunctions of linear inequalities. We represent the loop condition in matrix form as $Fx > b$ where $F \in \mathcal{M}(m, n, \mathbb{R})$, and $b \in \mathbb{R}^m$. By $Fx > b$ we mean that each coordinate of vector $Fx$ is greater than the corresponding coordinate of vector $b$.*
*• Transition relations are affine or linear forms. We represent the linear assignments in matrix form as $x := Ax + c$, where $A \in \mathcal{M}(n, \mathbb{R})$, and $c \in \mathbb{R}^n$. The most general linear loop program $P = P(A, F, b, c)$ is defined as* while $(Fx > b)$, $\{x := Ax + c\}$.

We will use the following classification.

**Definition 2.4.** *From the more specific to the more general form:*
*• Homogeneous: We denote by $P^{\mathbb{H}}$ the set of programs of the form $P(A, f)$ :* while $(f.x > 0)$, $\{x := Ax\}$*, where $f$ is a $1 \times n$ row matrix corresponding to the loop condition, and $A \in \mathcal{M}(n, \mathbb{R})$ corresponds to the list of assignments in the loop.*
*• Generalized Homogeneous: We denote by $P^{\mathbb{G}}$ the set of programs of the form $P(A, F)$ :* while $(Fx > 0)$, $\{x := Ax\}$ *where $F$ is a $(m \times n)$-matrix with rows corresponding to the loop conditions ($m$-loop conditions). We will sometimes write $P(A, F) = P(A, f_1, \ldots, f_m)$, where the $f_i$'s are the rows of $F$.*
*• Affine: We denote by $P^{\mathbb{A}}$ the set of programs of the form $P(A, F, b, c)$ :* while $(Fx > b)$, $\{x := Ax + c\}$*, for $A$ and $F$ as above, and $b$ and $c \in \mathbb{R}^n$.*

In Section 5, we show that the termination analysis for the general class $P^{\mathbb{A}}$ can be reduced to the problem of termination for programs in $P^{\mathbb{H}}$, when transition matrices have a real spectrum.

# 3   The ANT set

We present the new notion of *asymptotically non-terminating* ($ANT$) values of a loop program. It will be central in the analysis of non-termination. We start with the

definition of the *ANT* set and then give the first important result for homogeneous linear programs. We will extend these results in Section 5 to generalized linear homogeneous programs and then expand them further to general affine programs. The problem of termination analysis for the general class of linear programs will be reduced to the generation and the emptiness check of the *ANT* set for homogeneous linear programs.

Consider the program $P(A, f)$, where $A \in \mathcal{M}(n, \mathbb{R})$, $f \in \mathcal{M}(1, n, \mathbb{R})$. Alternatively, let $\mathbf{A} \in End_{\mathbb{R}}(E)$, $\mathbf{f} \in E^*$ and the program $P(\mathbf{A}, \mathbf{f}) : \mathsf{while}\, \mathbf{f}(\mathbf{x}) > 0, \{\mathbf{x} := \mathbf{A}\mathbf{x}\}$. Fixing a basis $B$ of $E$ we will write $A = Mat_B(\mathbf{A})$, $f = Mat_B(\mathbf{f})$, $x = Mat_B(\mathbf{x})$, and so on. We first give the definition of the termination for this class of programs.

**Definition 3.1.** *The program $P(\mathbf{A}, \mathbf{f})$ terminates on input $\mathbf{x} \in E$ if and only if there exists $k \geq 0$ such that $\mathbf{f}(\mathbf{A}^k(\mathbf{x}))$ is not positive. Also, for $A \in \mathcal{M}_n(\mathbb{R})$, and $f \in \mathcal{M}_{1,n}(\mathbb{R})$, we say that $P(A, f)$ terminates on input $x \in \mathbb{R}^n$ if and only if there exists $k \geq 0$ such that $fA^kx$ is not positive. Thus, $P(\mathbf{A}, \mathbf{f})$ is non-terminating if and only if there exists an input $\mathbf{x} \in E$ such that $\mathbf{f}(\mathbf{A}^k(\mathbf{x})) > 0$ for all $k \geq 0$. In matrix terms, $P(A, f)$ is non-terminating on input $x \in \mathbb{R}^n$ if and only if $\langle A^k x, f \rangle > 0$ for all $k \geq 0$.* □

The following lemma is obvious.

**Lemma 3.1.** *$P(\mathbf{A}, \mathbf{f})$ terminates on $\mathbf{x}$ if and only if $P(A, f)$ terminates on $x$.* □

Next, we introduce the important notion of an asymptotically non-terminating value.
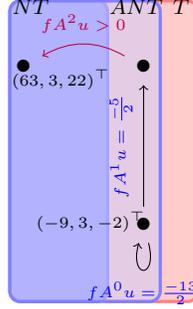
**Definition 3.2.** *We say that $\mathbf{x} \in E$ is an asymptotically non-terminating value for $P(\mathbf{A}, \mathbf{f})$ if there exists $k_{\mathbf{x}} \geq 0$ such that $P(\mathbf{A}, \mathbf{f})$ is non-terminating on $\mathbf{A}^{k_x}(\mathbf{x})$. We will also say that $\mathbf{x}$ is ANT for $P(\mathbf{A}, \mathbf{f})$. We will also say that $P(\mathbf{A}, \mathbf{f})$ is ANT on $\mathbf{x}$.* □

The definition of an ANT value for a program $P(A, f)$ with $A \in \mathcal{M}_n(\mathbb{R})$ and $f \in \mathcal{M}_{1,n}(\mathbb{R})$ is similar. It is again obvious that $\mathbf{x}$ is ANT for $P(\mathbf{A}, \mathbf{f})$ if and only if $x$ is ANT for $P(A, f)$. If $K$ is a subset of $E$, we will say that $P(\mathbf{A}, \mathbf{f})$ is ANT on $K$ if it is ANT on every $\mathbf{x}$ in $K$. Note that $P(\mathbf{A}, \mathbf{f})$ is non-terminating on $\mathbf{A}^{k_{\mathbf{x}}}(\mathbf{x})$ if and only if $\mathbf{f}(\mathbf{A}^k(\mathbf{x}))$ is $> 0$ for $k \geq k_{\mathbf{x}}$. The following example illustrates *ANT* sets and their properties.

**Example 3.1.** *Consider again Example1.1.*
*It is easy to check that the initial value $u = (-9, 3, -2)^{\top}$ belongs to the ANT set. But the program terminates on $u$ because with this initial value no loop iteration will be performed as $fA^0u = -13/2$. From Definition 3.2, there exists $k_u \geq 0$ such that $P(A, f)$ is non-terminating on $A^{k_u}u$. Also, $fA^1u = -5/2$, but $fA^2u > 0$ and the*

*program is non-terminating on $A^2u = (63, 3, 22)^\top$. So, the input $u = (-9, 3, -2)^\top$ is*



*ANT for $P(A, f)$ with $k_u = 2$.*                                                        □

We denote by $ANT(P(\mathbf{A}, \mathbf{f}))$ the set of $ANT$ values of $P(\mathbf{A}, \mathbf{f})$, and similarly for programs involving matrices. From now on, we give definitions and statements in terms of programs involving linear maps, and let the reader infer the obvious adaptation for programs involving matrices. If the set $ANT(P(\mathbf{A}, \mathbf{f}))$ is not empty, we say that the program $P(\mathbf{A}, \mathbf{f})$ is $ANT$. We will also write $NT$ for non terminating. The following theorem already shows the importance of $ANT$ sets: termination for linear programs is reduced to the emptiness check of the $ANT$ set.

**Theorem 3.1.** *The program $P(\boldsymbol{A}, \boldsymbol{f})$ in $P^{\mathbb{H}}$ is NT if and only if it is ANT (i.e., $ANT(P(\boldsymbol{A}, \boldsymbol{f})) \neq \emptyset$). More generally, if $K$ is an $\boldsymbol{A}$-stable subset of $E$, the program $P(\boldsymbol{A}, \boldsymbol{f})$ is NT on $K$ if and only if it is ANT on $K$.*                                                        □

We just saw that the set of $NT$ values is included in the $ANT$ set, but the most important property of an $ANT$ set resides in the fact that each of its elements gives an associated element in $NT$ for the corresponding program. That is, each element $\mathbf{x}$ in the $ANT$ set, even if it does not necessarily belong to the $NT$ set, refers directly to initial values $\mathbf{A}^{k_\mathbf{x}}(\mathbf{x})$ for which the program does not terminate. Hence there exists a number of loop iterations $k_\mathbf{x}$, departing from the initial value $\mathbf{x}$, such that $P(\mathbf{A}, \mathbf{f})$ does not terminate on $\mathbf{A}^{k_\mathbf{x}}(\mathbf{x})$. This does not imply that $\mathbf{x}$ is $NT$ for $P(A, f)$ because the program $P(A, f)$ could terminate on $\mathbf{x}$ by performing a number of loop iterations strictly smaller than $k_\mathbf{x}$. On the other hand, the $ANT$ set is more than an over-approximation of the $NT$ set, as it will provide us with a deterministic and efficient way to decide termination.

Let $ANT^c$ be the complement of the $ANT$ set. It gives us an under approximation for the set of all initial values for which the program terminates.

**Corollary 3.1.** *Let $P(\boldsymbol{A}, \boldsymbol{f})$ be in $P^{\mathbb{H}}$. Then $P(\boldsymbol{A}, \boldsymbol{f})$ terminates on the complementary set $ANT^c(P(\boldsymbol{A}, \boldsymbol{f}))$ of $ANT(P(\boldsymbol{A}, \boldsymbol{f}))$.*                                                        □

# 4   ANT set and Complex Eigenvalues

Let $Spec(\mathbf{A})$ be the set of all eigenvalues of $\mathbf{A}$. We show that the non-real eigenvalues in $Spec(\mathbf{A})$ do not affect the static termination analysis for a linear homogeneous program. We also show that the problem of checking the termination of linear programs can be reduced to verifying whether $Spec_{\mathbb{R}}(\mathbf{A}) \subset Spec(\mathbf{A})$. This will provide a complete and deterministic procedure to statically and automatically verify the termination of a linear loop program. We will reduce the problem of termination for a linear program $P(\mathbf{A}, \mathbf{f})$ to the emptiness check of $ANT^r(P(\mathbf{A}, \mathbf{f}))$, the set of $ANT$ values in the maximal $\mathbf{A}$-stable subspace $E^r$ such that the restriction $\mathbf{A}_{|E^r}$ has only real eigenvalues, that is, such that $Spec(\mathbf{A}_{|E^r}) = Spec_{\mathbb{R}}(\mathbf{A})$. We will also show how to compute the $ANT^r(P(\mathbf{A}, \mathbf{f}))$ sets. The complement of $ANT^r$ will turn out to be the set of initial values for which the program does terminate.

In $\mathbb{R}[X]$ the characteristic polynomial $\chi_{\mathbf{A}}$ factors uniquely as $\chi_{\mathbf{A}}^{nr} \prod_{\lambda \in Spec_{\mathbb{R}}(\mathbf{A})} (X - \lambda)^{d_\lambda}$, where $\chi_{\mathbf{A}}^{nr}$ has no real roots. We denote by $\chi_{\mathbf{A}}^{+}$ the product $\prod_{\lambda > 0 \in Spec_{\mathbb{R}}(\mathbf{A})} (X - \lambda)^{d_\lambda}$, and by $\chi_{\mathbf{A}}^{-}$ the product $\prod_{\lambda < 0 \in Spec_{\mathbb{R}}(\mathbf{A})} (X - \lambda)^{d_\lambda}$. We recall that for $P \in \mathbb{R}[X]$ the spaces $Ker(P(A))$ and $Im(P(A))$ are always $A$-stable.

**Definition 4.1.** *We denote by $E^+$ the space $Ker(\chi_{\mathbf{A}}^{+}(\mathbf{A}))$, by $E^-$ the space $Ker(\chi_{\mathbf{A}}^{-}(\mathbf{A}))$, by $E^r$ the space $E^+ \oplus E_0(\mathbf{A}) \oplus E^-$, and by $E^{nr}$ the space $Ker(\chi_{\mathbf{A}}^{nr}(\mathbf{A}))$. They are all $\mathbf{A}$-stable.*  □

The space $E^r$ is such that $\mathbf{A}_{|E^r}$ has only real eigenvalues, that is, $Spec(\mathbf{A}_{|E^r}) = Spec_{\mathbb{R}}(\mathbf{A})$. We recall the following proposition from basic linear algebra, which is a consequence of the fact that if the gcd $P \wedge Q$ of two polynomials in $\mathbb{R}[T]$ is equal to 1, then $Ker(PQ(A)) = Ker(P(A)) \oplus Ker(Q(A))$.

**Proposition 4.1.** *One has the decompositions:*
$E^+ = \oplus_{\lambda > 0 \in Spec(\mathbf{A})} E_\lambda(\mathbf{A})$, $E^- = \oplus_{\lambda < 0 \in Spec(\mathbf{A})} E_\lambda(\mathbf{A})$, *and* $E = E^r \oplus E^{nr}$.  □

We also recall a theorem from [23] (see Theorem 3.3 and 3.4 from [23]), which gives a necessary and sufficient condition for $P(\mathbf{A}, \mathbf{f})$ to be terminating. In [23], the result is rigorously stated in solid mathematical way (i.e, a mix of topological and algebraic arguments).

**Theorem 4.1.** *The program $P(\mathbf{A}, \mathbf{f})$ is non-terminating if and only if there is a $\lambda > 0$ in $Spec(\mathbf{A})$, such that $E_\lambda(\mathbf{A}) \not\subset Ker(\mathbf{f})$.*  □

Theorem 4.1 has the following important consequence.

**Proposition 4.2.** *If program $P(\mathbf{A}, \mathbf{f})$ is ANT on $\mathbf{x}$ in $E$, where $\mathbf{x} = \mathbf{x}^+ + \mathbf{x}'$ with $\mathbf{x}^+ \in E^+$ and $\mathbf{x}' \in E' = E^- \oplus E_0(\mathbf{A}) \oplus E^{nr}$, then it is asymptotically non-terminating on $\mathbf{x}^+$.*  □

We can refine Proposition 4.2 using the following result.

**Theorem 4.2.** *If program $P(\boldsymbol{A}, \boldsymbol{f})$ is asymptotically non-terminating on $\boldsymbol{x} = \boldsymbol{x}^r + \boldsymbol{x}^{nr}$, where $\boldsymbol{x}^r \in E^r$ and $\boldsymbol{x}^{nr} \in E^{nr}$, then it is asymptotically non-terminating on $\boldsymbol{x}^r$.*  □

We can now state the next result.

**Theorem 4.3.** *We write $\boldsymbol{A}^r$ for the restriction of $\boldsymbol{A}$ to $E^r$, $\boldsymbol{f}$ for the restriction of $\boldsymbol{f}$ to $E^r$, and let $ANT^r(P(\boldsymbol{A}, \boldsymbol{f})) = ANT(P(\boldsymbol{A}, \boldsymbol{f})) \cap E^r$. Then $ANT(P(\boldsymbol{A}, \boldsymbol{f}))$ is non empty if and only if $ANT^r(P(\boldsymbol{A}, \boldsymbol{f}))$ is non empty. In particular, $P(\boldsymbol{A}, \boldsymbol{f})$ terminates if and only if $P(\boldsymbol{A}^r, \boldsymbol{f})$ terminates. In any case, one has $ANT^r(P(\boldsymbol{A}, \boldsymbol{f})) = ANT(P(\boldsymbol{A}^r, \boldsymbol{f}))$.*  □

**Example 4.1.** *Consider the following program and the associated matrices:*

```
while(3t+7s+x-1/2y-2z>0){
  t:=t-s;
  s:=t+2s;
  x:=-20x-9y+75z;
  y:=-140x-55y+504z;
  z:=560x+228y-2011z;}
```

$$A' = \begin{pmatrix} \begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix} & \\ & \boxed{A} \end{pmatrix}$$

*and $f' = (3, 7, f)^{\top}$. Here the submatrices $A$ and $f$ are the one associated to Example 1.1.*

*The sumatrix $A = \begin{pmatrix} -20 & -9 & 75 \\ 7 & 8 & -21 \\ -7 & -3 & 26 \end{pmatrix}$ correspond to the simultaneous updates representing the the sequential loop assignments of Example 1.1. The submatrix $\begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix}$ encodes the effect of the two sequential instructions t:=t-s; and t:=t+2s; in terms of simultaneous updates. As this submatrix has only non real complex eigenvalues, Theorem 4.3 says that $ANT^r(P(A', f')) = ANT(P(A, f))$. If the initial values of $t, s, x, y$ and $z$ are represented, respectively, by the parameters $u_1, u_2, u_3, u_4$ and $u_5$, then we obtain conditions similar to that in Example 1.1. But now the ANT locus is described by the parameters $u_3, u_4$ and $u_5$, thus:*

```
[[u3 < -u4 + 3*u5]] OR [[u3 == -u4 + 3*u5, -u5 < u4]] OR
[[u3 == 4*u5, u4 == -u5, 0 < u5]]
```

□

# 5   The ANT set for affine programs

In the first subsection we express the *ANT* set of programs $P(\mathbf{A}, \mathbf{f}_1, \ldots, \mathbf{f}_m) \in P^{\mathbb{G}}$, involving several linear forms $\mathbf{f}_i$, in terms of the sets $ANT(P(\mathbf{A}, \mathbf{f}_i))$. In the next

subsection we reduce the computation of the $ANT$ sets and termination of affine programs $P(\mathbf{A}, (\mathbf{f}_i)_{i=1,\ldots,m}, \mathbf{b}, \mathbf{c}) \in P^{\mathbb{A}}$ to the corresponding problems for programs in $P^{\mathbb{G}}$. Hence, after Subsection 5.1, to corresponding problems for programs $P(\mathbf{A}, \mathbf{f})$ where $\mathbf{A}$ has a real spectrum.

## 5.1  Handling generalized homogeneous programs

Consider $P(\mathbf{A}, \mathbf{F}) = P(\mathbf{A}, (\mathbf{f}_i)_{i=1,\ldots,m})$ in $P^{\mathbb{G}}$ in the form while $(\forall i = 1, \ldots, m, \ \mathbf{f}_i \mathbf{x} > 0)$, $\{\mathbf{x} := \mathbf{A}\mathbf{x}\}$. We start with the following lemma.

**Lemma 5.1.** *The value $\boldsymbol{x}$ is NT for $P(\boldsymbol{A}, \mathbf{F})$ in $P^{\mathbb{G}}$ if and only if it is NT for all $P(\boldsymbol{A}, \boldsymbol{f}_i)$ with $i \in \{1, \ldots, m\}$.* $\square$

Now, we define $ANT$ values for such programs.

**Definition 5.1.** *We say that $\boldsymbol{x}$ is ANT for $P(\boldsymbol{A}, \mathbf{F})$ if there exists $k_{\boldsymbol{x}}$ such that for all $i \in \{1, \ldots, m\}$ we have $\boldsymbol{f}_i(\boldsymbol{A}^k(\boldsymbol{x})) > 0$ for $k > k_{\boldsymbol{x}}$, that is, if $\boldsymbol{x}$ is ANT for all programs $P(\boldsymbol{A}, \boldsymbol{f}_i)$.* $\square$

Again we have the following easy but important lemma.

**Lemma 5.2.** *Program $P(\boldsymbol{A}, \mathbf{F})$ is NT if and only if it is ANT, that is, $ANT(P(\boldsymbol{A}, \mathbf{F})) \neq \emptyset$.* $\square$

The next result will be crucial for the main result of this section.

**Proposition 5.1.** *If program $P(\boldsymbol{A}, \mathbf{F})$ is ANT then we must have $\bigcap_i ANT^r(P(\boldsymbol{A}, \boldsymbol{f}_i)) \neq \emptyset$.* $\square$

For the main result of this section, let $ANT^r(P(\mathbf{A}, \mathbf{F})) = \bigcap_i ANT^r(P(A, f_i))$.

**Theorem 5.1.** *Program $P(\boldsymbol{A}, \mathbf{F})$ is non-terminating if and only if $ANT^r(P(\boldsymbol{A}, \mathbf{F})) \neq \emptyset$. In particular, this gives a deterministic procedure to check if $P(\boldsymbol{A}, \mathbf{F})$ is NT, as we can always compute $ANT^r(P(\boldsymbol{A}, \mathbf{F}))$. Moreover, if $P(\boldsymbol{A}, \mathbf{F})$ is non-terminating we obtain an under approximation of the set of terminating values for $P(\boldsymbol{A}, \mathbf{F})$, namely $E^r - ANT^r(P(\boldsymbol{A}, \mathbf{F}))$.* $\square$

## 5.2  Generalization to affine programs

We now reduce the affine case to the homogeneous case. First we define the notion of $ANT$ values for this class of programs.

**Definition 5.2.** *Let $P(A, F, b, c)$ be an affine program in $P^{\mathbb{A}}$. For $x = x_0 \in \mathbb{R}^n$, denote by $x_1$ the vector $Ax + c$, and recursively let $x_k = Ax_{k-1} + c$. We say that a vector $x$ is ANT for $P(A, F, b, c)$ if there is some $k_x$ such that $k \geq k_x$ implies $Fx_k > b$. We denote by $ANT(P(A, F, b, c))$ the set of ANT inputs of $P(A, F, b, c)$, and we set $ANT^r(P(A, F, b, c)) = ANT(P(A, F, b, c)) \cap E^r$.* $\qquad\square$

Let $A \in \mathcal{M}(n, \mathbb{R})$, $F \in \mathcal{M}(m, n, \mathbb{R})$, $b = (b_1, \ldots, b_m)^\top$ in $\mathcal{M}(1, m, \mathbb{R})$, and $c$ a vector in $\mathcal{M}(1, n, \mathbb{R})$. We denote by $P(A, F, b, c) \in P^{\mathbb{A}}$ the program which computes $x := Ax + c$ as long as $Fx > b$. Now we define $A' \in \mathcal{M}(n + 1, \mathbb{R})$ and $V' \in \mathcal{M}(m + 1, n + 1, \mathbb{R})$ as follows: $A' = \begin{bmatrix} A & c \\ 0 & 1 \end{bmatrix}$, $F' = \begin{bmatrix} F & -b \\ 0 & 1 \end{bmatrix}$. The following theorem shows that the generation of the $ANT$ set for a program in $P^{\mathbb{A}}$ reduces to the generation of the $ANT$ set for an associated program in $P^{\mathbb{H}}$.

**Theorem 5.2.** *Let $P(A, F, b, c)$ be an affine program in $P^{\mathbb{A}}$. Consider the matrices $A'$ and $F'$ as constructed just above. A vector $x$ is ANT for $P(A, F, b, c)$ if and only if the vector $\begin{bmatrix} x \\ 1 \end{bmatrix}$ is ANT for $P(A', F')$.* $\qquad\square$

# 6 New Decidability Results for Affine Rationals and Integers Programs

In this section, we provide new decidability results for the termination of affine programs over the rationals and the integers when the transition matrix has a real spectrum. This problem has been studied in [10], where the termination of programs over the rationals has been proved to be decidable in general, and over the integers it has been proved to be decidable only in the homogeneous case. With our restriction on the spectrum, we obtain termination decidability for general affine programs $P(A, F, b, c)$ over the rationals and the integers. Moreover, this comes with an exact algorithm to check termination. Such a simple and clear algorithm is not provided in [10]. In addition, we settle an open question in [10], namely the decidability of termination in the case of affine programs over the integers — under our assumption on $Spec(A)$. Further, our method works for a very large family of subsets of $\mathbb{R}^n$ — and not only of $\mathbb{Q}^n$, $\mathbb{Z}^n$, or $\mathbb{N}^n$, — namely, it works for all those stable subsets under $x \mapsto Ax + c$. Formally, we show that the computation of the $ANT$ set of an affine program $P(A, F, b, c)$ gives a very simple answer to the termination problem on any subset $K$ of $\mathbb{R}^n$ which is stable under $x \mapsto Ax + c$. However, in the previous sections, we only determine the set $ANT^r(P(A, F, b, c)) \subset ANT(P(A, F, b, c))$. While this is sufficient to answer the termination problem on the reals in general, it is not enough to answer the termination on stable subspaces like $K$, as in general, we do not have $K = K \cap E^r \oplus K \cap E^{nr}$.

For now, we will need to restrict ourselves to the case where $Spec(A)$ is real. We will remove this restriction and raise the general problem in another companion article where more technical details will be presented together with some experiments.

**Definition 6.1.** *If $P(A, F, b, c)$ is an affine program. Let $A'$ and $F'$ be the matrices defined in the previous section. Then $ANT^r(P(A, F, b, c))$ is the set of elements $x \in \mathbb{R}^n$ such that $x' = \begin{bmatrix} x \\ 1 \end{bmatrix} \in ANT^r(P(A', F'))$.* $\square$

**Proposition 6.1.** *If $A$ has a real spectrum, then we must have $ANT(P(A, F, b, c)) = ANT^r(P(A, F, b, c))$.* $\square$

Here is the core theoretical contribution of this section.

**Theorem 6.1.** *Let $K$ be a subspace of $\mathbb{R}^n$, stable under $x \mapsto Ax + c$. Then $P(A, F, b, c)$ is non-terminating on $K$ if and only if $ANT(P(A, F, b, c)) \cap K \neq \emptyset$. Moreover, the program terminates on $K - ANT(P(A, F, b, c)) \cap K$.* $\square$

When $Spec(A)$ is real we must have $ANT^r(P(A, F, b, c)) = ANT(P(A, F, b, c))$ by Proposition 6.1. Hence, we obtain the following corollary.

**Corollary 6.1.** *When $Spec(A)$ is real the termination problem on $K$ is decidable. Additionally we can always compute the set $ANT(P(A, F, b, c)) \cap K$.* $\square$

In [10], the termination of affine programs over the integers was left open. See Section 10 for a discussion. Our criterion for termination over stable subspaces allows us to answer this question when $A$ has a real spectrum. We have the following new characterization decidability for termination of affine programs over the rationals and the integers.

**Corollary 6.2.** *If $A$ and $c$ have rational, respectively integer, coefficients, and $Spec(A)$ is real, then $P(A, F, b, c)$ terminates over $\mathbb{Q}^n$, respectively $\mathbb{Z}^n$, if and only if $ANT(P(A, F, b, c)) \cap \mathbb{Q}^n = \emptyset$, respectively $ANT(P(A, F, b, c)) \cap \mathbb{Z}^n = \emptyset$.* $\square$

# 7    Automatic generation of ANT values

We show how to compute exactly $ANT$ loci. As we saw in the previous sections, it is enough to treat the case where the program belongs to $P^{\mathbb{H}}$, *i.e.*, can be written as $P(\mathbf{A}, \mathbf{f})$, with $Spec(\mathbf{A})$ a set of reals. In Subsection 7.1 we start with what we call the regular case: a condition on $\mathbf{A}$ and $\mathbf{f}$ which is satisfied most of the time. In Subsection 7.2 we explain how the general case reduces to the regular case. Given that we can generate the $ANT$ locus of any $P(\mathbf{A}, \mathbf{f})$ in $P^{\mathbb{H}}$ when $Spec(\mathbf{A})$ is a set of reals, we are in fact able to generate $ANT^r(P(A, F, b, c))$ for any $P(A, F, b, c)$ in $P^{\mathbb{A}}$, without any further hypothesis about $Spec(A)$. In this section, we drop the bold font when denoting elements in $E$.

## 7.1 The regular case

In this subsection we assume that $K(\mathbf{A}, \mathbf{f}) = \cap_{k \geq 0} Ker(\mathbf{f} \circ \mathbf{A}^k) = \cap_{k=0}^{n-1} Ker(\mathbf{f} \circ \mathbf{A}^k)$, and $E_0(\mathbf{A})$ are simply $\{0\}$. We first recall the following consequence of the existence of a Jordan form for $\mathbf{A}$.

**Lemma 7.1.** *Let $\lambda$ be a nonzero eigenvalue of $\mathbf{A}$. We can produce a basis $B_\lambda$ of $E_\lambda(\mathbf{A})$ such that $Mat_{B_\lambda}(\mathbf{A})$ is of the form $\lambda.diag(T_{\lambda,1}, \ldots, T_{\lambda,r_\lambda})$, where each $T_{\lambda,i}$ is*

*a matrix of size $n_{\lambda,i}$ of the form* $\begin{pmatrix} 1 & 1 & & & & \\ & 1 & 1 & & & \\ & & \ddots & \ddots & & \\ & & & 1 & 1 & \\ & & & & 1 & 1 \\ & & & & & 1 \end{pmatrix}$, *and with $n_{\lambda,i} \leq n_{\lambda,i+1}$ for $i$*

*between $1$ and $r_\lambda - 1$.*

In fact, we assumed $K(\mathbf{A}, \mathbf{f})$ to be null, there is only one block.

**Proposition 7.1.** *For every $\lambda$ in $Spec(\mathbf{A})$ with $\lambda \neq 0$, we have $Mat_{B_\lambda}(\mathbf{A}_{|E_\lambda(A)}) = \lambda.T_{\lambda,1}$. So, we simply write $T_\lambda = T_{\lambda,1}$.*

Now we compute the power of $T_\lambda$.

**Lemma 7.2.** $(T_\lambda)^k = \begin{pmatrix} 1 & \binom{k}{1} & \binom{k}{2} & \cdots & \cdots & \binom{k}{d_\lambda - 1} \\ & 1 & \binom{k}{1} & \binom{k}{2} & \ddots & \vdots \\ & & \ddots & \ddots & \ddots & \vdots \\ & & & 1 & \binom{k}{1} & \binom{k}{2} \\ & & & & 1 & \binom{k}{1} \\ & & & & & 1 \end{pmatrix}$.

We write $B = B_{\lambda_1} \cup \cdots \cup B_{\lambda_t}$. It is a basis of $E$. We write $Mat_B(f)$ as $(a_{\lambda_1,1}, a_{\lambda_1,2}, \ldots, a_{\lambda_1,d_{\lambda_1}}, \ldots, a_{\lambda_t,1}, \ldots, a_{\lambda_t,d_{\lambda_t}})$. The previous lemma has the following consequence.

**Proposition 7.2.** *For $\lambda$ in $Spec(\mathbf{A})$, there are well determined polynomials $P_{\lambda,j} \in \mathbb{R}[X]$, for $j$ between $1$ and $d_\lambda$, such that $Mat_B(\mathbf{f} \circ \mathbf{A}^k) = (P_{\lambda,1}(k), \ldots, P_{\lambda,d_\lambda}(k))$. In fact, $P_{\lambda,j}(k) = a_{\lambda,1}\binom{k}{j-1} + a_{\lambda,2}\binom{k}{j-2} + \cdots + a_{\lambda,j}$. In particular, $P_{\lambda,j}$, as a polynomial in $k$, is of degree at most $j - 1$. We thus write it as $P_{\lambda,j}(k) = b_{\lambda,j-1}^j k^{j-1} + \cdots + b_{\lambda,1}^j k + b_{\lambda,0}^j$, where we can compute each $b_{\lambda,l}^j$ explicitly as a linear combination of the $a_{\lambda,i}$'s.*

We now give a procedure, in several steps, to determine the set of $ANT$ values. Here, in order to lighten the notations, for $x = \sum_{\lambda \in Spec(\mathbf{A})} x_\lambda$, we write $P_\lambda(x_\lambda, k) = \sum_{j=1}^{d_\lambda} x_{\lambda,j} P_{\lambda,j}(k)$. By convention, if $\lambda$ is not an eigenvalue, the polynomial $P_{\lambda,j}$, so that $P_\lambda(x_\lambda) : k \mapsto P_\lambda(x_\lambda, k)$, is zero. We set $b_{\lambda,i}^j = 0$ as soon as $i > d_\lambda$. We obtain the expression $P_\lambda(x_\lambda, k) = \sum_{j=0}^{d_\lambda - 1} \phi_{\lambda,j}(x_\lambda)k^j$, where

$$\phi_{\lambda,j}(x_\lambda) = b_{\lambda,j}^{j+1} x_{\lambda,j+1} + b_{\lambda,j}^{j+2} x_{\lambda,j+2} x_{\lambda,j+2} + \ldots$$
$$+ b_{\lambda,j}^{d_\lambda} x_{\lambda,d_\lambda}.$$

We set $\phi_{\lambda,j}(x_\lambda) = 0$ as soon as $j \geq d_\lambda$. We also write $Q^+_{\pm\lambda}(x_{\pm\lambda}) = P_{|\lambda|}(x_{|\lambda|}) + P_{-|\lambda|}(x_{-|\lambda|})$, and $Q^-_{\pm\lambda}(x_{\pm\lambda}) = P_{|\lambda|}(x_{|\lambda|}) - P_{-|\lambda|}(x_{-|\lambda|})$. In particular, we have

$$Q^+_{\pm\lambda}(x_{\pm\lambda}, k) = \sum_{j=0}^{e_{|\lambda|}-1} \phi^+_{\pm\lambda,j}(x_{|\lambda|}, x_{-|\lambda|})k^j,$$

where $e_{|\lambda|} = max(d_{|\lambda|}, d_{-|\lambda|})$,

$$\phi^+_{\pm\lambda,j}(x_{|\lambda|}, x_{-|\lambda|}) = \phi_{|\lambda|,j}(x_{|\lambda|}) + \phi_{-|\lambda|,j}(x_{-|\lambda|}),$$

and

$$Q^-_{\pm\lambda}(x_{\pm\lambda}, k) = \sum_{j=0}^{e_{|\lambda|}-1} \phi^-_{\pm\lambda,j}(x_{|\lambda|}, x_{-|\lambda|})k^j,$$

where

$$\phi^-_{\pm\lambda,j}(x_{|\lambda|}, x_{-|\lambda|}) = \phi_{|\lambda|,j}(x_{|\lambda|}) - \phi_{-|\lambda|,j}(x_{-|\lambda|}).$$

As the first step of the procedure, we give a set of constraints giving birth to $ANT$ values.

**Proposition 7.3.** *Let $\lambda$ be a positive eigenvalue, such that both $Q^+_{\pm\lambda}(x_{|\lambda|})$ and $Q^-_{\pm\lambda}(x_{|\lambda|})$ have a positive dominant term, and such that $P_\mu(x_\mu)$ is zero whenever $|\mu| > \lambda$. Then $x$ is an ANT point of $P(\boldsymbol{A}, \boldsymbol{f})$.*

In terms of the linear forms $\phi_{|\lambda|,j}$ and $\phi^\pm_{|\lambda|,j}$ we have the following proposition.

**Proposition 7.4.** *For $\lambda > 0$ in $Spec(\boldsymbol{A})$, and two integers $k$ and $k'$ between $0$ and $d_\lambda - 1$, denote by $S^\lambda_{k,k'}$ the set of $x$ in $E$ which satisfy:*
*1) for all $\mu$ with $|\mu| > \lambda$, and all $j$ between $0$ and $d_\mu - 1$: $\phi_{\mu,j}(x_\mu) = 0$.*
*2) for all $e_\lambda - 1 \geq j > k$: $\phi^+_{\pm\lambda,j}(x_\lambda, x_{-\lambda}) = 0$.*
*3) for all $e_\lambda - 1 \geq j > k'$: $\phi^-_{\pm\lambda,j}(x_\lambda, x_{-\lambda}) = 0$.*
*4) $\phi^+_{\pm\lambda,k}(x_\lambda, x_{-\lambda}) > 0$.*
*5) $\phi^-_{\pm\lambda,k'}(x_\lambda, x_{-\lambda}) > 0$.*
*Consider the set $\Delta_S = \{(\lambda, k, k')|\lambda > 0 \in Spec(\boldsymbol{A}), k \in \{1, \ldots, d_\lambda - 1\}, k' \in \{1, \ldots, d_\lambda - 1\}\}$. If $x$ belongs to*

$$S = \bigvee_{(\lambda,k,k')\in\Delta_S} S^\lambda_{k,k'}, \tag{1}$$

*then $x$ is $ANT$.*

We illustrate how to generate the constraints of Proposition 7.4 on a running example.

**Example 7.1.** (Running example) *Let $T = Mat_B(\boldsymbol{A})$, and $f = Mat_B(\boldsymbol{f})$ be as fol-lows:* $T = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & -1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & -2 \end{pmatrix}$, *and* $f = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$. *Also,* $f = (a_{1,1}, a_{1,2}, a_{-1,1}, a_{-1,2}, a_{2,1}, a_{-2,1})$.

*In [?] we give values of all the needed terms appearing in the computation of the ANT set. We have four eigenvalues: $1$ of multiplicity $2$, $-1$ of multiplicity $2$, $2$ of multiplicity $1$ and $-2$ of multiplicity $1$. For Proposition 7.4, we take only the positive eigenvalues $2$ and $1$. From the multiplicity of these eigenvalues, we know that we have to consider $k \in \{0,1\}$ and $k' \in \{0,1\}$. From Proposition 7.4 we generate:*

$$S^2_{0,0} \equiv (x_{2,1} + x_{-2,1} > 0) \wedge (x_{2,1} - x_{-2,1} > 0)$$

$$S^1_{0,0} \equiv (x_{2,1} = 0) \wedge (x_{-2,1} = 0) \wedge (x_{1,2} + x_{-1,2} = 0)$$
$$\wedge (x_{1,2} - x_{-1,2} = 0) \wedge (x_{1,1} + x_{1,2} + x_{-1,1} + x_{-1,2} > 0)$$
$$\wedge (x_{1,1} + x_{1,2} - x_{-1,1} - x_{-1,2} > 0)$$

$$S^1_{0,1} \equiv (x_{2,1} = 0) \wedge (x_{-2,1} = 0) \wedge (x_{1,2} + x_{-1,2} = 0)$$
$$\wedge (x_{1,1} + x_{1,2} + x_{-1,1} + x_{-1,2} > 0) \wedge (x_{1,2} - x_{-1,2} > 0)$$

$$S^1_{1,0} \equiv (x_{2,1} = 0) \wedge (x_{-2,1} = 0) \wedge (x_{1,2} + x_{-1,2} > 0)$$
$$\wedge (x_{1,1} + x_{1,2} - x_{-1,1} - x_{-1,2} > 0).$$

*Proposition 7.4 gives the set $S = S^2_{0,0} \vee S^1_{0,0} \vee S^1_{0,1} \vee S^1_{1,0} \vee S^1_{1,1}$.*

The next result gives other set of constraints for *ANT* values.

**Proposition 7.5.** *Let $\lambda$ be a positive eigenvalue and $|\lambda'| < \lambda$ in $|Spec(\boldsymbol{A})|$ such that:*
- *$P_\mu(x_\mu)$ is zero whenever $|\mu| > \lambda$,*
- *$Q^-_{\pm\lambda}(x_{\pm\lambda}) = 0$,*
- *$Q^+_{\pm\lambda}(x_{\pm\lambda}) = 2P_\lambda(x_\lambda)$ has a positive dominant term,*
- *$Q^-_{\pm\mu}(x_{\pm\mu})$ is zero whenever $|\lambda'| < |\mu| < \lambda$,*
- *$Q^-_{\pm\lambda'}(x_{\pm\lambda'})$ has a positive dominant term*

*Then $P(\boldsymbol{A}, f)$ is ANT on $x$.*

In terms of the linear forms $\phi_{\lambda,j}$, $\phi^+_{\pm\lambda,j}$ and $\phi^-_{\pm\lambda,j}$:

**Proposition 7.6.** *For $\lambda > 0$ and $\lambda'$ in $Spec(\boldsymbol{A})$, with $|\lambda'| < \lambda$, an integer $k$ between $1$ and $d_\lambda - 1$, and an integer $k'$ between $1$ and $d_{\lambda'} - 1$, we denote by $U^{\lambda,|\lambda'|}_{k,k'}$ the set of $x$ in $E$ which satisfy:*
*1) for all $\mu$ with $|\mu| > \lambda$, and all $j$ between $0$ and $d_\mu - 1$: $\phi_{\mu,j}(x_\mu) = 0$.*
*2) for all $0 \leq j \leq e_\lambda - 1$: $\phi^-_{\pm\lambda,j}(x_{|\lambda|}, x_{-|\lambda|}) = 0$.*

*3) for all $|\mu|$ with $\lambda' < |\mu| < \lambda$, and all $0 \leq j \leq e_{|\mu|} - 1$: $\phi^-_{\pm\mu,j}(x_{|\mu|}, x_{-|\mu|}) = 0$.*
*4) for all $d_\lambda - 1 \geq j > k$: $\phi_{\lambda,j}(x_\lambda) = 0$.*
*5) for all $e_{|\lambda'|} - 1 \geq j > k'$: $\phi^-_{\pm\lambda',j}(x_{|\lambda'|}, x_{-|\lambda'|}) = 0$.*
*6) $\phi_{\lambda,k}(x_\lambda) > 0$.*
*7) $\phi^-_{\pm\lambda',k'}(x_{|\lambda'|}, x_{-|\lambda'|}) > 0$.*
*Consider the set $\Delta_U = \{(\lambda, \lambda', k, k') | \lambda > 0 \in Spec(\boldsymbol{A}), |\lambda'| < \lambda \in |Spec(\boldsymbol{A})|, k \in \{1, \ldots, d_\lambda - 1\}, k' \in \{1, \ldots, e_{|\lambda'|} - 1\}\}$. If $x$ belongs to*

$$U = \bigvee_{(\lambda,\lambda',k,k')\in\Delta_U} U^{\lambda,|\lambda'|}_{k,k'}, \tag{2}$$

*then $x$ is ANT.*

**Example 7.2.** (Running example)*: Again, we illustrate how the conditions are generated for our running Example 7.1. There are only two possible cases to consider.*

$$U^{2,1}_{0,0} \equiv (x_{2,1} - x_{-2,1} = 0) \wedge (x_{1,2} - x_{-1,2} = 0)$$
$$\wedge (x_{2,1} > 0) \wedge (x_{1,1} + x_{1,2} - x_{-1,1} - x_{-1,2} > 0)$$

$$U^{2,1}_{0,1} \equiv (x_{2,1} - x_{-2,1} = 0) \wedge (x_{2,1} > 0) \wedge (x_{1,2} - x_{-1,2} > 0)$$

*We obtain $U = U^{2,1}_{0,0} \vee U^{2,1}_{0,1}$.*

Finally, we give a last set of constraints, giving the remaining *ANT* values.

**Proposition 7.7.** *Let $\lambda$ be a positive eigenvalue and $|\lambda'| < \lambda$ in $|Spec(\boldsymbol{A})|$ such that:*
*1) $P_\mu(x_\mu)$ is zero whenever $|\mu| > \lambda$,*
*2) $Q^+_{\pm\lambda}(x_{\pm\lambda}) = 0$ and $Q^-_{\pm\lambda}(x_{\pm\lambda}) = 2P_\lambda(x_\lambda)$ has a positive dominant term,*
*3) $Q^{\mp}_{\pm\mu}(x_{\pm\mu})$ is zero whenever $|\lambda'| < |\mu| < \lambda$,*
*4( $Q^+_{\pm\lambda'}(x_{\pm\lambda'})$ has a positive dominant term.*
*  Then $P(\boldsymbol{A}, \boldsymbol{f})$ is ANT on $x$.*

We now write the preceding proposition in terms of the linear forms $\phi_{\lambda,j}$, $\phi^+_{\pm\lambda,j}$ and $\phi^-_{\pm\lambda,j}$.

**Proposition 7.8.** *For $\lambda > 0$ and $\lambda'$ in $Spec(\boldsymbol{A})$, with $|\lambda'| < \lambda$, an integer $k$ between 1 and $d_\lambda - 1$, and an integer $k'$ between 1 and $d_{\lambda'} - 1$, we denote by $V^{\lambda,|\lambda'|}_{k,k'}$ the set of $x$ in $E$ which satisfy:*
*1) for all $\mu$ with $|\mu| > \lambda$, and all $j$ between 0 and $d_\mu - 1$: $\phi_{\mu,j}(x_\mu) = 0$.*
*2) for all $0 \leq j \leq e_{|\lambda|} - 1$: $\phi^+_{\pm\lambda,j}(x_\lambda, x_{-\lambda}) = 0$.*
*3) for all $|\mu|$ with $\lambda' < |\mu| < \lambda$, and all $0 \leq j \leq e_{|\mu|} - 1$: $\phi^+_{\pm\mu,j}(x_{|\mu|}, x_{-|\mu|}) = 0$.*
*4) for all $d_\lambda - 1 \geq j > k$: $\phi_{\lambda,j}(x_\lambda) = 0$.*
*5) for all $e_{|\lambda'|} - 1 \geq j > k'$: $\phi^+_{\pm\lambda',j}(x_{|\lambda'|}, x_{-|\lambda'|}) = 0$.*

6) $\phi_{\lambda,k}(x_\lambda) > 0$.

7) $\phi^+_{|\lambda'|,k'}(x_{|\lambda'|}, x_{-|\lambda'|}) > 0$.

Consider the set $\Delta_V = \{(\lambda, \lambda', k, k') | \lambda > 0 \in Spec(\boldsymbol{A}), |\lambda'| < \lambda \in |Spec(\boldsymbol{A})|, k \in \{1, \ldots, d_\lambda - 1\}, k' \in \{1, \ldots, e_{|\lambda|} - 1\}\}$. If $x$ belongs to

$$V = \bigvee_{(\lambda,\lambda',k,k')\in\Delta_V} V^{\lambda,|\lambda'|}_{k,k'}, \tag{3}$$

then $x$ is ANT.

**Example 7.3.** (Running example): On our running example, this gives the conditions:

$$V^{2,1}_{0,0} \equiv (x_{2,1} + x_{-2,1} = 0) \wedge (x_{1,2} + x_{-1,2} = 0) \wedge (x_{2,1} > 0)$$
$$\wedge (x_{1,1} + x_{-1,2} + x_{-1,1} + x_{-1,2} > 0)$$
$$V^{2,1}_{0,0} \equiv (x_{2,1} + x_{-2,1} = 0) \wedge (x_{1,2} + x_{-1,2} = 0) \wedge (x_{2,1} > 0)$$
$$\wedge (x_{1,1} + x_{-1,2} + x_{-1,1} + x_{-1,2} > 0)$$

By Proposition 7.8, the set $V$ of ANT values is $V = V^{2,1}_{0,0} \cup V^{2,1}_{0,1}$.

We now state the main theorem.

**Theorem 7.1.** An element $x$ in $E$ is ANT for $P(\boldsymbol{A}, \boldsymbol{f})$ if and only if we are in the situation of Propositions 4, 5, or 6.

We can rewrite the preceding theorem.

**Theorem 7.2.** The set $ANT(P(\boldsymbol{A}, \boldsymbol{f}))$, of ANT points of $P(\boldsymbol{A}, \boldsymbol{f})$, is equal to the disjoint union $S \cup U \cup V$, where $S$, $U$ and $V$ are defined in Propositions 7.4, 7.6, and 7.8.

We illustrate the conclusion of Theorem 7.2 next.

**Example 7.4.** (Running example): Using to Theorem 7.2 the ANT set of our running example is $S \vee U \vee V$ where the sets $S$, $U$ and $V$ have already been explicitly computed. See Examples 7.1, 7.2, and 7.3. Our prototype generates the equivalent semi-linear system:

```
Locus of ANT
[[[max(-X[-2,1], X[-2,1]) < X[2,1]]]
OR[[X[1,2] == 0, X[-1,2] == 0, X[2,1] == 0,
X[-2,1] == 0, max(-X[-1,1], X[-1,1]) < X[1,1]]]
OR[[X[1,2] == -X[-1,2], X[2,1] == 0, X[-2,1] == 0,
-X[-1,1] < X[1,1], X[-1,2] < 0]]
OR[[X[2,1] == 0, X[-2,1] == 0,
X[-1,1] - X[1,2] + X[-1,2] < X[1,1], -X[-1,2] < X[1,2]]]
OR[[X[2,1] == 0, X[-2,1] == 0,
max(-X[-1,2], X[-1,2]) < X[1,2]]]]
OR[[[X[1,2] == X[-1,2], X[2,1] == X[-2,1],
X[-1,1] < X[1,1], 0 < X[-2,1]]]
OR[[X[2,1] == X[-2,1], X[-1,2] < X[1,2], 0 < X[-2,1]]]]
OR[[[X[1,2] == X[-1,2], X[2,1] == -X[-2,1],
-X[-1,1] - 2*X[-1,2] < X[1,1], X[-2,1] < 0]]
OR[[X[2,1] == -X[-2,1], -X[-1,2] < X[1,2], X[-2,1] < 0]]]
```

## 7.2 The general case

Here we do not suppose that the spaces $K(\mathbf{A}, \mathbf{f})$ and $E_0(\mathbf{a})$ are reduced to zero anymore, but $Spec(\mathbf{A})$ is still assumed to be real. We first make the following definition.

**Definition 7.1.** *For $x$ in $E$, we denote by $E(\boldsymbol{A}, x)$ the subspace of $E$ generated by the family $(\boldsymbol{A}^k(x))_{k \geq 0}$. It is an $\boldsymbol{A}$-stable subspace.*

We next give the asymptotic behavior of $\mathbf{f}(\mathbf{A}^k(x))$ for $k$ large, $x \in E_\lambda(\mathbf{A})$, and $\lambda \in Spec(\mathbf{A}) - \{0\}$, of which Proposition 7.2 was a special precise case.

**Proposition 7.9.** *For $\lambda \in Spec(\boldsymbol{A}) - \{0\}$, and $x$ in $E_\lambda(\boldsymbol{A})$, there exists $P_\lambda(\boldsymbol{f}, x) \in \mathbb{R}[T]$ such that $\boldsymbol{f}(\boldsymbol{A}^k(x)) = \lambda^k P(\boldsymbol{f}, x)(k)$.*

In this situation, we will write $P_\lambda(v, x, k)$ for $P_\lambda(v, x)(k)$, so that $P_\lambda$ is a map from $\mathbb{R}^n \times E_\lambda(\mathbf{A}) \times \mathbb{N}$ to $\mathbb{R}$, linear in the first two variables, and polynomial in the last. We have the following:

**Proposition 7.10.** *If $\lambda \neq 0$ is a real eigenvalue of $\boldsymbol{A}$, $\boldsymbol{f}$ belongs to $E^*$, and $x$ belongs to $E_\lambda(\boldsymbol{A})$, then $P_\lambda(\boldsymbol{f}, x, .)$ is nonzero if and only if $x \notin K(\boldsymbol{f}, \boldsymbol{A})$.*

When studying the locus of $ANT$ values in $E$, or for any question related to the termination of program $P(\mathbf{A}, \mathbf{f})$, the subspace $K(\mathbf{f}, \mathbf{A})$ is not important since we have the following:.

**Proposition 7.11.** *For any $k \geq 0$, the linear form $\boldsymbol{f} \circ \boldsymbol{A}^k$ factors through the quotient $E/K(\boldsymbol{A}, \boldsymbol{f})$, i.e., for any $x \in E$, the value of $\boldsymbol{f}(\boldsymbol{A}^k(x))$ depends only on the class $x + K(\boldsymbol{A}, \boldsymbol{f})$.*

We denote by $\overline{\mathbf{A}}$ the endomorphism of $\overline{E} = E/K(\mathbf{A}, \mathbf{f})$ induced by $\mathbf{A}$, and by $\overline{\mathbf{f}}$ the linear form on $E/K(\mathbf{A}, \mathbf{f})$ induced by $\mathbf{f}$. Then, we write $\overline{E} = \overline{E}_0(\overline{\mathbf{A}}) \oplus \overline{E}^a$, where $\overline{E}^a = \oplus_{\lambda \in Spec(\overline{\mathbf{A}}) - \{0\}} \overline{E}_\lambda(\overline{\mathbf{A}})$. Consider the restriction $\overline{\mathbf{A}}^a$ of $\overline{\mathbf{A}}$ to $\overline{E}^a$, as well as the restriction $\overline{\mathbf{f}}^a$ of $\overline{\mathbf{f}}$ to $\overline{E}^a$. Program $P(\overline{\mathbf{A}}^a, \overline{\mathbf{f}}^a)$ is of the form studied in the previous section, that is, we have $\overline{E}_0^a(\overline{\mathbf{A}}^a) = K(\overline{\mathbf{A}}^a, \overline{\mathbf{f}}^a) = \{0\}$. Hence, we know how to compute $ANT(P(\overline{\mathbf{A}}^a, \overline{\mathbf{f}}^a))$. The main theorem of this section reduces the $ANT$ computation of $P(\mathbf{A}, \mathbf{f})$ to that of $ANT(P(\overline{\mathbf{A}}^a, \overline{\mathbf{f}}^a))$.

**Theorem 7.3.** *Program $P(\boldsymbol{A}, \boldsymbol{f})$ terminates if and only if program $P(\overline{\boldsymbol{A}}^a, \overline{\boldsymbol{f}}^a)$ terminates. Moreover, if we write the canonical projection $p : E \to \overline{E}$, we have the relation $ANT(P(\boldsymbol{A}, \boldsymbol{f})) = p^{-1}(ANT(P(\overline{\boldsymbol{A}}^a, \overline{\boldsymbol{f}}^a)) + \overline{E}_0(\overline{\boldsymbol{A}}))$.*

It might not be obvious how to apply this in a concrete situation, where we are given a program $P(A, f)$, corresponding to a matrix $A \in M(n, \mathbb{R})$, and a row vector $f$ in $\mathbb{R}^n$. We explain how to proceed. First, compute a basis $B_{A,f}$ of $K(A, f) = \cap_{k=0}^{n-1} Ker(fA^k)$, which is the kernel of the matrix of $M(n, \mathbb{R})$ with its $i$-th row equal

to $fA^{i-1}$. Then, take any family $B_1$ where $B' = B_{A,v} \cup B_1$ is a basis of $\mathbb{R}^n$, and let $P$ be the matrix whose columns are the vectors of $B'$. We have $P^{-1}AP = \begin{pmatrix} X & Y \\ 0 & A_1 \end{pmatrix}$, for $A_1$ the size of $B_1$. Now consider the matrix $A_1 \in M(n_1, \mathbb{R})$, and take the modified Jordan basis $B_J$ where the first vectors of $B_J$ are a Jordan basis $B_0$ of $E_0(A_1)$, and the next vectors in $B_J$ are ordered as a union of basis $B_\lambda$ for each $E_\lambda(A_1)$, with $\lambda \neq 0$ in $Spec(A_1)$, where $B_\lambda$ is the modified Jordan basis defined in Lemma 7.1. If $P_1$ is the matrix in $M(n_1, \mathbb{R})$, whose columns are the vectors of $B_J$, then $T = P_1^{-1}A_1P_1$ is of the form $T = diag(T_0, \lambda_1 T_{\lambda_1}, \ldots, \lambda_{t-1} T_{\lambda_{t-1}}, \lambda_t T_{\lambda_t})$ where $\lambda_1, \ldots, \lambda_t$ are the nonzero eigenvalues of $A_1$, $T_{\lambda_i}$ is of the form described in Lemma 7.1, and

$T_0$ is of the form $\begin{pmatrix} 0 & 1 & & & \\ & 0 & 1 & & \\ & & \ddots & \ddots & \\ & & & 0 & 1 \\ & & & & 0 \end{pmatrix}$. Write $T^a = diag(\lambda_1 T_{\lambda_1}, \ldots, \lambda_t T_{\lambda_t})$, so that $T = diag(T_0, T^a)$ in $M(n_a, \mathbb{R})$, where $n_a = \sum_{\lambda \neq 0 \in Spec(A_1)} dim(E_\lambda(A_1))$. If we write $Q = diag(I_{n-n_1}, P_1)$, and $R = PQ$, we get $B = R^{-1}AR = \begin{pmatrix} X & Y \\ 0 & T \end{pmatrix} = \begin{pmatrix} X & Y \\ 0 & diag(T_0, T^a) \end{pmatrix}$.

For $x = (x_1, \ldots, x_n)^T \in \mathbb{R}^n$, we write $x^a = (x_{n-n_a+1}, \ldots, x_n)^T$. Then, we set $w = fR$ in $M(1, n, \mathbb{R})$, and write $w^a = (w_{n-n_a+1}, \ldots, w_n)$. We know how to compute the set $ANT(P(T^a, w^a))$ using the results of the previous section. We finally obtain the following theorem.

**Theorem 7.4.** *Vector $x$ is in $ANT(P(B, w))$ if and only if $x^a$ is in $ANT(P(T^a, w^a))$. Vector $y$ is in $ANT(A, f)$ if and only if $R^{-1}y$ is in $ANT(P(B, w))$, i.e., $ANT(P(A, v)) = R(ANT(P(B, w)))$. In particular, $P(A, f)$ terminates if and only if $P(T^a, w^a)$ does.*

**Example 7.5.** *Take a program $P(\boldsymbol{A}, \boldsymbol{f})$ and matrices $Mat_C(\boldsymbol{A}) = A$ and $Mat_C(\boldsymbol{f}) = v$ corresponding, respectively, to the linear forms $\boldsymbol{A}$ and $\boldsymbol{f}$, expressed in the canonical basis $C$ of $\mathbb{R}^n$:*

$$A = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 6 & -2 & -1 & 1 & 0 & 0 & 0 & 0 \\ 10 & -3 & -4 & 3 & 0 & 0 & 0 & 0 \\ 30 & -15 & -6 & 7 & 0 & -1 & 0 & 0 \\ 44 & -28 & -6 & 9 & 1 & -2 & 0 & 0 \\ 90 & -55 & -9 & 12 & 4 & -7 & 2 & 0 \\ 57 & -19 & -9 & 1 & 5 & -8 & 4 & -2 \end{pmatrix}$$

*and $v = (-1, -2, 1, 0, 0, 0, 0, 1)$.*

The main step is construction of a basis $B_{E_0, K}$ in which the matrices of $\boldsymbol{A}$ and $\boldsymbol{f}$ are the form $B$ and $w$. Follow the steps described above, we obtain the following matrices $R$, $Mat_{B_{E_0, K}}(\boldsymbol{A}) = B$, and $Mat_{B_{E_0, K}}(\boldsymbol{f}) = w$:

$$R = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 3 & 3 & 1 & 0 & 0 & 0 & 0 & 0 \\ 4 & 5 & 2 & 1 & 0 & 0 & 0 & 0 \\ 6 & 5 & 3 & 2 & 1 & 0 & 0 & 0 \\ 4 & 2 & 5 & 2 & 1 & 1 & 0 & 0 \\ 3 & 8 & 8 & 2 & 1 & 2 & 1 & 0 \\ 2 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{pmatrix},$$

$$B = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -2 \end{pmatrix}, \text{ and } w = \begin{pmatrix} 0 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}.$$

*Matrix $R$ is such that $B = R^{-1}AR = \begin{pmatrix} X & Y \\ 0 & \begin{pmatrix} T_0 & \\ & T^a \end{pmatrix} \end{pmatrix}$. In this case, $B$ has the expected form with $X = (1)$, $Y = (0\ 0\ 0\ 0\ 0\ 0)$, $T_0 = (0)$, and $T^a$ being the matrix $T$ depicted in Example 7.1. Also, if we denote by $e_1$ and $e_2$ the two first elements of the canonical basis $C$, we obtain $K(A, f) = R(K(B, w)) = Vect(R(e_1))$, i.e., $Vect(R(e_1))$ is the space spanned by the first column of $R$, and $E_0(B) = Vect(e_2)$. By construction $w = vR = (0, 1, 1, 1, 1, 1, 1, 1)$, and thus $w^a = (1, 1, 1, 1, 1, 1)$. Finally, we apply Theorem 7.4, which claims the following equivalence: ($y$ is ANT for $P(A, v)$) $\Leftrightarrow$ ($x = (x_1, \ldots x_8)^\top = R^{-1}y$ is ANT for $P(B, w)$) $\Leftrightarrow$ ($x^a = (x_3, \ldots, x_8)^\top$ is ANT for $P(T^a, w^a)$). The analysis of the ANT set for $P(A, f)$ is reduced to the generation of $ANT(P(T^a, w^a))$. As $T^a$ and $w^a$ describe the same system as the one obtained in Example 7.4, we already have the symbolic representation of $ANT(P(T^a, w^a))$. To generate the ANT set for $P(A, v)$, one just needs to rewrite the semi-linear space obtained in Example 7.4, now considering the variables $(x_3, \ldots, x_8)$.*

# 8 ANT sets generation in practice

We provide more practical details on our computational method. We show that in practice one can obtain the *ANT* set using only a few specific and concise formulas *e.g.*, Theorem 8.3 and Equations 4, 5, and 6. First we identify a useful characteristic of almost all affine programs. Previously, we identified and treated separately and completely the degenerate cases where the spaces $K(\mathbf{A}, \mathbf{f}) = \cap_{k \geq 0} Ker(\mathbf{f} \circ \mathbf{A}^k) = \cap_{k=0}^{n-1} Ker(\mathbf{f} \circ \mathbf{A}^k)$ and $E_0(\mathbf{A})$ are not reduced to $\{0\}$. That is why we made the assumption that $K(\mathbf{A}, \mathbf{f}) = \{0\}$ and $E_0(\mathbf{A}) = \{0\}$ for the remainder of the paper.

**Definition 8.1.** *Let $(A, f)$ belong to $\mathcal{M}(n, \mathbb{R}) \times \mathbb{R}^n$. We say that a program $P(A, f)$ is* normal *if $Spec(A)$ does not contain a real eigenvalue and its additive inverse. In other words, if the $\lambda \in Spec(A)$ then $-\lambda$ is not an eigenvalue of $A$.*

We are going to show that almost all programs are normal. We recall the definition

of a Zariski open subset of a real vector space.

**Definition 8.2.** *let $V$ be a finite-dimensional vector space over $\mathbb{R}$, and let $P$ in $\mathbb{R}[V]$, that is, a polynomial map from $V$ to $K$. If $P_1, \ldots, P_t$ are in $K[V]$, we denote by $D_{P_1, \ldots, P_t} = \{v \in V, \exists i \in [1, \ldots, t], P_i(v) \neq 0\}$. A Zariski open subset of $V$ is a finite intersection of sets of the form $D_{P_1, \ldots, P_t}$.*    $\square$

The following lemma is standard.

**Lemma 8.1.** *A non-empty Zariski open subset of $V$ is open, dense, and its complementary set in $V$ has zero Lebesgue measure. Two non-empty Zariski open sets have a non-empty Zariski open intersection.*

We can now state and prove the main theorem of this section.

**Theorem 8.1.** *The set $R(A, v)$ of pairs $(A, v)$ with $A$ in $\mathcal{M}(n, \mathbb{R})$ and $v$ in $\mathbb{R}^n$, where $A$ has no real eigenvalue $\lambda$ with $-\lambda$ is also an eigenvalue, and such that the space $K(A, v)$ is reduced to zero, contains a non empty Zariski open subset of $\mathcal{M}(n,_R) \times \mathbb{R}^n$. In particular, its complementary set in $\mathcal{M}(n, \mathbb{R}) \times \mathbb{R}^n$ has zero Lebesgue measure. This basically says, that a program $P(A, v)$ is almost always regular.*    $\square$

We now present the practical details of our procedure for generating the formulas that compose the symbolic representations of the $ANT$ set for normal programs. We first recall the fact that one can produce a basis $B_\lambda$ of $E_\lambda(\mathbf{A})$ such that $Mat_{B_\lambda}(\mathbf{A})$ is of the form $\lambda.diag(T_{\lambda,1}, \ldots, T_{\lambda,r_\lambda})$, where each $T_{\lambda,i}$ is a matrix of the form depicted in Lemma 7.1. The power of $T_\lambda$ is indicated in Lemma 7.2. Again, we write $B = B_{\lambda_1} \cup \cdots \cup B_{\lambda_t}$ a basis of $E$, and $Mat_B(f)$ as $(a_{\lambda_1,1}, a_{\lambda_1,2}, \ldots, a_{\lambda_1,d_{\lambda_1}}, \ldots, a_{\lambda_t,1}, \ldots, a_{\lambda_t,d_{\lambda_t}})$. For $\lambda$ in $Spec(\mathbf{A})$ let $P_{\lambda,j} = a_{\lambda,1}\binom{k}{j-1} + a_{\lambda,2}\binom{k}{j-2} + \cdots + a_{\lambda,j}$, as in Proposition 7.2, for $j$ between 1 and $d_\lambda$, and such that $Mat_B(\mathbf{f} \circ \mathbf{A}^k) = (P_{\lambda,1}(k), \ldots, P_{\lambda,d_\lambda}(k))$. When $x = \sum_{\lambda \in Spec(\mathbf{A})} x_\lambda$, we write $P_\lambda(x_\lambda, k) = \sum_{j=1}^{d_\lambda} x_{\lambda,j} P_{\lambda,j}(k)$.

**Theorem 8.2.** *Suppose that we are in the common situation where $\mathbf{A}$ has no eigenvalue $\lambda$ with $-\lambda$ is also an eigenvalue. Then $x$ is $ANT^r$ if and only if the following two conditions hold:*

1. *The $\lambda$ of highest absolute value satisfying : $\exists j \in \{1, \ldots d_\lambda\}$ such that $a_{\lambda,j} x_{\lambda,j} \neq 0$ is strictly positive.*

2. *For this $\lambda$, the highest $j_0 \in \{1, \ldots d_\lambda\}$ such that $a_{\lambda,j_0} x_{\lambda,j_0} \neq 0$ satisfies $a_{\lambda,j_0} x_{\lambda,j_0} > 0$.*    $\square$

Now we use theorem 8.2 to get the generic formula that, once instantiated, represent the $ANT$ set symbolically and exactly.

**Theorem 8.3.** *For $\lambda > 0$ in $Spec(A)$ and an integer $k$ between $1$ and $d_\lambda$, denote by $S_{\lambda,k}$ the set of $x$ in $E$ which satisfy:*

*If $\mu \in Spec(\boldsymbol{A})$ is such that $|\mu| > \lambda$ then*

$$\forall h \in \{1, \ldots, d_\mu\} : a_{\mu,h} x_{\mu,h} = 0. \quad (4)$$

$$\forall h \in \{k+1, \ldots, d_\lambda\} : \quad a_{\lambda,h} x_{\lambda,h} = 0. \quad (5)$$

$$a_{\lambda,k} x_{\lambda,k} > 0. \quad (6)$$

*Consider the set $\Delta_S = \{(\lambda, k) | \lambda > 0 \in Spec(\boldsymbol{A}), k \in \{1, \ldots, d_\lambda\}\}$. Then we have*

$$ANT(P(A, f)) = \bigvee_{(\lambda,k) \in \Delta_S} S_{\lambda,k}. \quad \square \quad (7)$$

When using Theorem 8.3, one needs first to evaluate the terms $a_{\lambda,k}$ and $x_{\lambda,k}$. As we obtain our results using the decomposition of $x$ and $f$ in $B$, we recall in the following lemma how one obtains it from the decomposition of $x$ and $f$ in $B_c$, the canonical basis.

**Lemma 8.2.** *Let $P$ be the transformation matrix corresponding to $B$, and $x \in E$. If $x = \sum_{i=1}^{n} x_i e_i = (x_1, \ldots, x_n)^\top \in B_c$, and $x$ decomposes as $\sum_{j=1}^{t}(\sum_{i=1}^{d_j} x_{\lambda_j,i} e_{\lambda_j,i})$ in $B$, then the coefficients $x_{\lambda_j,i}$ are those of the column vector $P^{-1}x$ in $B_c$.* $\square$

This lemma is illustrated in the first computational step presented next. We first provide an example showing the main steps when computing $ANT$ sets for normal programs.

**Example 8.1.** *Consider the program $P(A, f)$ depicted as follows:* $A = \begin{pmatrix} 26 & 2 & -15 & -6 & 30 \\ 24 & 3 & -12 & -6 & 48 \\ 32 & 0 & -9 & 2 & 66 \\ -12 & 1 & 6 & 8 & -24 \\ -4 & -1 & 3 & 0 & 0 \end{pmatrix}$,

*and* $f = \begin{pmatrix} -2 \\ 0 \\ -1 \\ 0 \\ -1/2 \end{pmatrix}$.

**Step** 1: *In the triangularization of matrix $A$ we get:*

$P = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 \\ 4/5 & -3/2 & 6/5 & 0 & 1 \\ 1 & 0 & 8/5 & 2/3 & 2/3 \\ -2/5 & 1/2 & -3/5 & 0 & -1/2 \\ -1/5 & -1/2 & -1/5 & -1/3 & 1/6 \end{pmatrix}$,

$$D = \begin{pmatrix} 9 & 0 & 0 & 0 & 0 \\ 0 & 5 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 6 & 0 \\ 0 & 0 & 0 & 0 & 6 \end{pmatrix}, \ and$$

$$P^{-1} = \begin{pmatrix} 0 & 0 & -5 & -10 & -10 \\ 0 & -2 & 0 & -4 & 0 \\ -5 & 0 & 0 & -5 & -15 \\ 6 & 2 & 5 & 19 & 25 \\ 6 & -2 & 4 & 8 & 26 \end{pmatrix}.$$

We obtain the following eigenvectors given by the column of $P$, written using our notation:

$e_{9,1} = (1, 4/5, 1, -2/5, -1/5)^\top$,

$e_{5,1} = (1, -3/2, 0, 1/2, -1/2)^\top$,

$e_{2,1} = (1, 6/5, 8/5, -3/5, -1/5)^\top$,

$e_{6,1} = (1, 0, 2/3, 0, -1/3)^\top$ and

$e_{6,2} = (0, 1, 2/3, -1/2, 1/6)^\top$.

**Step** 2: *Computing $S_{\lambda,k}$ for all positive $\lambda \in Spec(A)^*$ and $k \in \{1, \ldots, d_\lambda\}$:*

- *Our algorithm first computes the coefficients $a_{\lambda,i}$:*

$$a_{9,1} = <f, e_{9,1}>$$
$$= < (-2, 0, -1, 0, -1/2)^\top, (1, 4/5, 1, -2/5, -1/5)^\top$$
$$>= -29/10, \quad (8)$$

$$a_{5,1} = <f, e_{5,1}>$$
$$= < (-2, 0, -1, 0, -1/2)^\top, (1, -3/2, 0, 1/2, -1/2)^\top$$
$$>= -7/4, \quad (9)$$

$$a_{2,1} = <f, e_{2,1}>$$
$$= < (-2, 0, -1, 0, -1/2)^\top, (1, 6/5, 8/5, -3/5, -1/5)^\top$$
$$>= -7/2, \quad (10)$$

$$a_{6,1} = <f, e_{6,1}>$$
$$= < (-2, 0, -1, 0, -1/2)^\top, (1, 0, 2/3, 0, -1/3)^\top$$
$$>= -5/2, \quad (11)$$

$$a_{6,2} = <f, e_{6,2}>$$
$$= < (-2, 0, -1, 0, -1/2)^\top, (0, 1, 2/3, -1/2, 1/6)^\top >$$
$$= -3/4. \quad (12)$$

- *Now our algorithm computes the coefficients of the decomposition of the initial variable values in B. They are the column vector $P^{-1} \cdot u$ in $B_c$ where*

$u = (u_1, u_2, u_3, u_4, u_5)^\top$ *is the vector encoding the initial variable values.*

$$P^{-1} \cdot \begin{pmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \end{pmatrix} = \begin{pmatrix} -5*u3 - 10*u4 - 10*u5 \\ -2*u2 - 4*u4 \\ -5*u1 - 5*u4 - 15*u5 \\ 6*u1 + 2*u2 + 5*u3 + 19*u4 + 25*u5 \\ 6*u1 - 2*u2 + 4*u3 + 8*u4 + 26*u5] \end{pmatrix} = \begin{pmatrix} x_{9,1} \\ x_{5,1} \\ x_{2,1} \\ x_{6,1} \\ x_{6,2} \end{pmatrix}.$$

**Step** 3: *We apply Theorem 8.3 to generate the ANT Locus. In order to computes the sets $S_{\lambda,i}$ one needs to first generate the formulas appearing in Theorem 8.3 for each positive eigenvalue. Then one needs to instantiate the generic formulas according to the computed terms $a_{\lambda,i}s$ and $x_{\lambda,j}s$. In the following, we show all computational operations in this example. For each positive eigenvalue $\lambda$ and integer $k \in \{1,...,d_\lambda\}$, we directly apply Theorem 8.3, using its three generic formulas.*

- *Case $\lambda = 9$ and $k = 1$: Eqs. 4 and 5, from Theorem 8.3, induce no constraint in this case. Firstly because there is no $\mu \in Spec(\boldsymbol{A})$ such that $|\mu| > \lambda$, and secondly because $\{k+1,...,d_\lambda\} = \emptyset$ since $k = d_\lambda = 1$. Eq, 6, from Theorem 8.3, generates the constraint $S_{9,1} = (a_{9,1}x_{9,1} > 0)$.*

- *Case $\lambda = 5$ and $k = 1$: Eq. 5 from Theorem 8.3 induce no formula since $\{k+1,...,d_\lambda\} = \emptyset$, as we are in the case where $k = d_\lambda = 1$. Considering Eq. 4, one needs to treat the two sub-cases with $\mu = 9$ and $\mu = 6$. When $\mu = 9$, we get $a_{9,1}x_{9,1} = 0$, and when $\mu = 6$, we obtain $(a_{6,1}x_{6,1} = 0) \wedge (a_{6,2}x_{6,2} = 0)$. Eq. 3 induces the formula $a_{5,1}x_{5,1} > 0$, and we generate the following equation associated to this eigenvalue:*

$$S_{5,1} = (a_{9,1}x_{9,1} = 0) \wedge (a_{6,1}x_{6,1} = 0) \wedge (a_{6,2}x_{6,2} = 0)$$
$$\wedge (a_{5,1}x_{5,1} > 0). \quad (13)$$

- *Case $\lambda = 2$ and $k = 1$: With Eq. 4, we have three sub-cases when $\mu = 9$, $\mu = 6$, and when $\mu = 5$. Starting with $\mu = 9$, we generate the constraint $a_{9,1}x_{9,1} = 0$. With $\mu = 6$, we obtain $(a_{6,1}x_{6,1} = 0) \wedge (a_{6,2}x_{6,2} = 0)$, and when $\mu = 5$ we have $a_{5,1}x_{5,1} = 0$. Here, Eq. 5 generates no further formulas since $\{k+1,...,d_\lambda\} = \emptyset$, as we have $k = d_\lambda = 1$. With Eq. 6, we obtain the constraint $(a_{2,1}x_{2,1} > 0)$ and the formula*

$$S_{2,1} = (a_{9,1}x_{9,1} = 0) \wedge (a_{6,1}x_{6,1} = 0) \wedge (a_{6,2}x_{6,2} = 0)$$
$$\wedge (a_{5,1}x_{5,1} = 0) \wedge (a_{2,1}x_{2,1} > 0). \quad (14)$$

- *Case $\lambda = 6$ and $k = 1$: With Eq. 4, one needs to consider $\mu = 9$, which gives the formula $(a_{9,1}x_{9,1} = 0)$. Now, looking at Eq. 5, we have $h = 2$ and we obtain*

*the constraint $(a_{2,2}x_{2,2} = 0)$. With Eq. 6 we obtain the constraint $(a_{2,1}x_{2,1} > 0)$, and generate the formula*

$$S_{6,1} = (a_{9,1}x_{9,1} = 0) \wedge (a_{2,2}x_{2,2} = 0) \wedge (a_{2,1}x_{2,1} > 0).$$

- *Case $\lambda = 6$ and $k = 2$: Again, with Eq. 4 one needs to consider $\mu = 9$ which gives the formula $(a_{9,1}x_{9,1} = 0)$. Also, Eq. 5 induces no formula since $\{k + 1, ..., d_\lambda\} = \emptyset$ because we have $k = d_\lambda = 2$. Using Eq. 6, we get the constraint $(a_{2,2}x_{2,2} > 0)$, and generate the formula:*

$$S_{6,2} = (a_{9,1}x_{9,1} = 0) \wedge (a_{2,2}x_{2,2} > 0).$$

*According to Theorem 8.3, Eq. 8.3, the ANT locus $S$ reduces to the following semi-linear space:*

$$S = S_{9,1} \vee S_{5,1} \vee S_{2,1} \vee S_{6,1} \vee S_{6,2}.$$

*The initial values of $x, y, z, s, t$ are represented, respectively, by the parameters $u_1, u_2, u_3, u_4, u_5$. Now, we can express the results in the canonical basis using Lemma 8.2 if we want, as all the terms $a_{\lambda,i}$ and $x_{\lambda,j}$ have been already computed in Step 1.* $\square$

The pseudo code depicted in Algorithm 1 illustrates the strategy. Our algorithm takes as input the number of variables, the chosen field where the variables are interpreted, the assignment matrix $A$ and the vector $f$ encoding the loop condition. We first compute the list of positive eigenvalues. See lines 1 and 2 in 1. Then, we just need to directly encode the statements and formulas provided in Theorem 8.3. We proceed considering each positive eigenvalues $e'[i]$ at a time, ee line 3, for each $k$ in $\{1, ..., d_\lambda\}$, see line 5.

- Then, we generate the constraint given by equation 4. We look for $\mu \in Spec(\mathbf{A})$ such that $|\mu| > \lambda$, see line 7. Then, for all $h \in \{1, \ldots, d_\mu\}$, see line 10, we add the constraint $(a_{\mu,h}x_{\mu,h} = 0)$ indicated in line 11.

- Next, we consider Eq. 5, in line 12, and we add the constraint $a_{\lambda,h}x_{\lambda,h} = 0$, as in line 13.

- Proceeding, we consider Eq. 6 and we add the constraint $(a_{\lambda,k}x_{\lambda,k} > 0)$, by line 14.

Finally, we progressively compute the disjunction $\bigvee_{(\lambda,k) \in \Delta_S} S_{\lambda,k}$, as in line 15.

---

**Algorithm 1: ANT_linear_Loop** $(n, \mathbb{K}, A, f)$

---

/\***Generating the** $ANT$ **set.**\*/;

**Data**: $n$ the number of program variables, $\mathbb{K}$ the field, $P(A, f) \in P^{\mathbb{H}}$ where
$\qquad A \in \mathcal{M}(n, \mathbb{K})$ and $f \in \mathcal{M}(n, 1, \mathbb{K})$

**Result**: $ANT^r(P(A, f))$

**begin**

1    $\{e[1], ..., e[r]\} \longleftarrow$ **eigenvalues**$(A)$;

2    $\{e'[1], ..., e'[s]\} \longleftarrow$ **strictly_positives**$(\{e[1], ..., e[r]\})$;

3    **for** $i = 1$ **to** $s$ **do**

4      $d_\lambda \longleftarrow$ **multiplicity**$(e'[i])$;

5      **for** $k = 1$ **to** $d_\lambda$ **do**

6        **for** $p = 1$ **to** $r$ **do**

7          **if** $|e[p]| > e'[i]$ **then**

8            $\mu \longleftarrow e[p]$;

9            $d_\mu \longleftarrow$ **multiplicity**$(e[p])$;

10            **for** $h = 1$ **to** $d_\mu$ **do**

11              $Ant[i] \longleftarrow Ant[i] \wedge (a_{\mu,h} x_{\mu,h} = 0)$;

12        **for** $l = k + 1$ **to** $d_\lambda$ **do**

13          $Ant[i] \longleftarrow Ant[i] \wedge (a_{e'[i],l} x_{e'[i],l} = 0)$;

14        $Ant[i] \longleftarrow Ant[i] \wedge (a_{e'[i],k} x_{e'[i],k} > 0)$;

15        $ANT \longleftarrow ANT \vee Ant[i]$;

16    **return** $ANT$;

---

Table 1: Experiments on randomly generated linear loop programs

| #Loops | Class | #Cond | #Var | #T | #NT | CPU/s[ANT] |
|--------|-------|-------|------|-----|-----|------------|
| 1000 | $P^{\mathbb{H}}$ | 1 | $[3, 4]$ | 130 | 870 | $19, 91$ |
| 1000 | $P^{\mathbb{G}}$ | $[2, 4]$ | $[3, 4]$ | 125 | 875 | $23, 72$ |
| 1000 | $P^{\mathbf{A}}$ | $[2, 4]$ | $[3, 4]$ | 117 | 883 | $24, 57$ |
| 1000 | $P^{\mathbb{H}}$ | 1 | $[5, 6]$ | 58 | 942 | $39.03$ |
| 1000 | $P^{\mathbb{G}}$ | $[2, 4]$ | $[4, 6]$ | 55 | 945 | $45, 45$ |
| 1000 | $P^{\mathbf{A}}$ | $[2, 4]$ | $[4, 6]$ | 52 | 948 | $49.79$ |
| 1000 | $P^{\mathbb{H}}$ | 1 | $[7, 15]$ | 26 | 974 | $107.92$ |
| 1000 | $P^{\mathbb{G}}$ | 2 | $[7, 15]$ | 44 | 956 | $178, 08$ |
| 1000 | $P^{\mathbf{A}}$ | 2 | $[7, 15]$ | 21 | 979 | $187, 78$ |

# 9    ANT Algorithm and Experiments

In Table 1 we list some experimental results. The column **Set-i** refers to a set of loops generated randomly. As expected, the probability to produce terminating programs tends to zero when the number of variables grows. The column **#Loops** gives the number of loops treated, where each set includes the analysis of 500 loops. The column **Class** gives the class of the linear loop programs either. $P^{\mathbb{H}}$, $P^{\mathbb{G}}$ or $P^{\mathbb{A}}$. The column **#Cond** gives the number of conjunctions in the loop condition for each program, and **#Var** refers to the numbers of program variables. The column **#T** returns the number of terminating programs, and the column **#NT** gives the number of non-terminating programs. Finally, column **CPU/s[ANT]** gives the cpu time for deciding on termination and the computation of the *ANT* loci. We have implemented our prototype using **Sage** [24] using interfaces written in python. This way we had access to several mathematical packages that were used to guarantee that all random loops were triangulable in the corresponding field, even when we had lots of variables.

**Example 9.1.** *Here we show an example of ANT computations taken from the long output results refering to line* 7 *(with* 10 *variables):*

```
=====Vector F====
[  1    0    1    0 1/2 1/2   0    2   -1    2]
=====Matrix A====
[ -67   55   -1   19   15   -5   12   -4 -340  -81]
[ -15 -132  -27  -15  -15  -15  -12  -12  357  462]
[ -36  -34  -13    2    0  -10    0   -8   10  170]
[-124   28  -20   37   20  -20   16  -16 -364   72]
[ 111   -4   22  -22   -7   20   -8   16  271 -127]
[  -2 -147  -29  -21  -20  -12  -16  -12  423  485]
[  36   34   14   -2    0   10    4    8  -10 -164]
[  20  164   36   20   20   20   16   20 -424 -564]
[  13  -24   -2   -6   -5    0   -4    0  105   59]
[ -18  -17   -7    1    0   -5    0   -4    5   86]
================
Locus of ANT:[-20*u10 + 5*u4 + 20*u9 > 0, 6*u1 - 6*u10 + 18*u9 == 0]
OR[22*u10 - 11*u3 > 0, -20*u10 + 5*u4 + 20*u9 == 0, 123/2*u10 - 41/2*u2 + 123/2*u9 == 0,
```

```
6*u1 - 6*u10 + 18*u9 == 0, 7/2*u1 + 19*u10 + 7/2*u2 + 7/2*u3 + 7/2*u4 + 9*u5 + 7/2*u6 + 9*u7 + 7/2*u8 - 13*u9 == 0,
-24*u10 - 9*u7 - 3/2*u8 - 6*u9 == 0, -17/2*u1 - 101/2*u10 + 17*u2 + 17/2*u3 - 17/2*u4 - 17/2*u5 - 3*u6 - 163/2*u9 == 0]
OR[7/2*u1 + 19*u10 + 7/2*u2 + 7/2*u3 + 7/2*u4 + 9*u5 + 7/2*u6 + 9*u7 + 7/2*u8 - 13*u9 > 0,
-20*u10 + 5*u4 + 20*u9 == 0, 123/2*u10 - 41/2*u2 + 123/2*u9 == 0, 6*u1 - 6*u10 + 18*u9 == 0]OR
[-24*u10 - 9*u7 - 3/2*u8 - 6*u9 > 0, -20*u10 + 5*u4 + 20*u9 == 0, 123/2*u10 - 41/2*u2 + 123/2*u9 == 0,
6*u1 - 6*u10 + 18*u9 == 0, 7/2*u1 + 19*u10 + 7/2*u2 + 7/2*u3 + 7/2*u4 + 9*u5 + 7/2*u6 + 9*u7 + 7/2*u8 - 13*u9 == 0]
OR[-17/2*u1 - 101/2*u10 + 17*u2 + 17/2*u3 - 17/2*u4 - 17/2*u5 - 3*u6 - 163/2*u9 > 0,
-20*u10 + 5*u4 + 20*u9 == 0, 123/2*u10 - 41/2*u2 + 123/2*u9 == 0, 6*u1 - 6*u10 + 18*u9 == 0,
7/2*u1 + 19*u10 + 7/2*u2 + 7/2*u3 + 7/2*u4 + 9*u5 + 7/2*u6 + 9*u7 + 7/2*u8 - 13*u9 == 0,
-24*u10 - 9*u7 - 3/2*u8 - 6*u9 == 0]
0.05
=====Vector V====
```

Table 2 presents more comparisons with some existing methods. The first line refers to a program drawn from an industrial audio compression module in [2], Ex.1 and Fig.3. It is depicted below on the right. We concentrate on the while loop staring at line. Following Section 5.2 and its Theorem 5.2, we have the matrices:

```
y=0;
if (x>=0){
  while(-x >
     -2^(30)){
  x:=x << 1;
  y++;}
}
```

$$A' = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix} \text{ and } F' = \begin{bmatrix} -1 & 0 & 2^{30} \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \text{ Matrix } A' \text{ is already in}$$

Jordan normal form with eigenvalues 2 and 1 of multiplicity 1 and 2, respectively. One needs to treat lines $f_1, f_2$ and $f_3$ of $F'$. Using Theorem 8.3 and its ready-to-use formulas 4, 5 and 6 one computes $ANT(P(A', f_i))$ for $1 \leq i \leq 3$. Using the same notations introduced in Section 8, we obtain $S_{2,1} \equiv (x_{2,1} < 0)$, $S_{1,1} \equiv (x_{2,1} = 0)$ and $S_{1,2} \equiv (x_{2,1} = 0)$, and the computed $ANT$ set is $S_{2,1} \vee S_{1,1} \vee S_{1,2}$. In other words, in the canonical basis if $u_1$ and $u_2$ are the parameters used as initial values for the variables $x$ and $y$ we obtain the precise $ANT$ set $(u_1 < 0) \vee (u_1 = 0)$. In [2], the computation of there precondition for termination took 22 seconds. Our algorithm took 0.03 seconds to compute our precondition for termination. Moreover, the computed $ANT$ set is exactly the set of non-terminating inputs and thus its complementary set is the exact set of terminating inputs. The second line in Table 2 refers to a program from [13] and [2], depicted below.

THe $ANT$ complementary set obtained by our prototype gives a more precise preconditions for termination than the one previously proposed. Our algorithm took 0.03 seconds to compute our precondition for termination. Moreover, the computed $ANT$ set is exactly the set of non-terminating inputs and thus its complementary set is the exact set of terminating inputs. The second line in Table 2 refers to a program from [13] and [2].For this program, we generate a more precise precondition, representing a larger set of inputs. The experiments in [2], involving industrial examples and handwritten programs, indicate that there techniques took 24 seconds to output there preconditions. The sixth entry in Table 2 deals with a simple program with a no existing linear ranking function. Also the eighth example deals with a program terminating on $\mathbb{Z}$ but not on $\mathbb{Q}$. The last line of 2 shows that our algorithm took 1.028 seconds

```
while(x>0)
  {
  x:=x+y;
  y:=y+z;
}
```

Table 2: $ANT^c$ for linear programs from related works

| Programs | Class | #Cond | #Var | Cpu/s:$ANT^c$ |
|---|---|---|---|---|
| [2] Ex.1 Fig.3 | $P^{\mathbf{A}}$ | 1 | 2 | 0.03 |
| [13] and [2] Ex.1 Fig.4 | $P^{\mathbb{H}}$ | 1 | 3 | 0.02 |
| [9] Ex.3 | $P^{\mathbb{H}}$ | 1 | 2 | 0.02 |
| [8] and [9] Ex.4 | $P^{\mathbb{G}}$ | 2 | 2 | 0.03 |
| [9] Ex.5 | $P^{\mathbb{G}}$ | 2 | 2 | 0.03 |
| [8] Ex.2 | $P^{\mathbf{A}}$ | 1 | 1 | 0.02 |
| [10] Ex.2 | $P^{\mathbb{H}}$ | 1 | 2 | 0.04 |
| [21] (1) | $P^{\mathbf{A}}$ | 2 | 2 | 0.05 |
| [21] Ex.4.15 | $P^{\mathbf{A}}$ | 4 | 3 | 0.06 |
| 32 loops from [14]: Tab1.#10 to #41 | $P^{\mathbb{H}}, P^{\mathbb{G}}, P^{\mathbf{A}}$ | $[1,2]$ | $[1,4]$ | 1.28 |

to handle 32 loops taken from [14].

# 10  Discussion

Concerning *termination analysis* for affine programs over the reals, rationals and the integers, we reduced the problem to the emptiness check of the generated $ANT$ sets. By so doing, we obtained a characterization of terminating linear programs which allows for a practical and complete polynomial time computational procedure. In [10, 9], the authors focused on the decidability of the termination problem for linear loop programs. Also [10] is based on the approach in [9], but now considering termination analysis over the rationals and integers for *homogeneous programs*. But the termination problem for *general affine* programs over the integers is left open in [10]. Our criteria for termination over stable subspaces allowed us to address this question, and lead to new decidability results. See Section 6.

In this work, although we also considered the termination problem, we addressed a more general problem, namely, the *conditional termination* problem of generating static sets of terminating and non-terminating inputs for the program. We provided efficient computational methods allowing for the exact computation and symbolic representation of the $ANT$ sets for affine loop programs over $\mathbb{R}$, $\mathbb{Q}$, $\mathbb{Z}$, and $\mathbb{N}$. The $ANT$ sets generated by our approach can be seen as a precise over-approximation for the set of non-terminating inputs for the program. Here, we use "precise" in the sense that $NT \subseteq ANT$ and all elements in $ANT$, even those not in $NT$, are directly associated with non-terminating values modulo a finite numbers of loop iterations. The, possibly infinite, complement of an $ANT$ set is also a "precise" under-approximation of the set of terminating inputs, as it provides terminating input data entering the loop at least once. Our methods differs from [2] as we do not use the synthesis of ranking functions. The methods proposed in [12] can provide non-linear preconditions, but

we always generate semi-linear sets as precondition for termination, which facilitates the static analysis of liveness properties. The approaches proposed in [13] considers first octagonal relations and suggests the use of quantifier elimination techniques and algorithms, which would running in exponential time complexity $O(n^3 \cdot 5^n)$. They also consider the conditional termination problem for restricted subclasses of linear affine relations where the associated matrix has to be diagonalizable, and with all non-zero eigenvalues of multiplicity one. The experiments in [14], involving hand-written programs, are handled succesfully by our algorithm. The strength and the practical efficiency of the approach is shown by our experiments dealing with a large number of larger linear loops.

Our main results, Theorems 3.1, 4.3 , 5.1, 5.2, 6.1, 7.2, 7.4, 8.3 and their associated corollaries and direct encodings, are evidences of the novelty of our approach.

# 11    Conclusion

We presented the new notion of *asymptotically non-terminating initial variable values* for linear programs. Our theoretical results provided us with powerful computational methods allowing for the automated generation of the sets of all asymptotically non-terminating initial variable values, represented symbolically and exactly by a semi-linear space, *e.g.*, conjunctions and disjunctions of linear equalities and inequalities. We reduced the termination/non-termination problem of linear, affine programs to the emptiness check of the *ANT* set of specific homogeneous linear programs. Moreover, by taking the complement of the semi-linear set of ANT initial variable values, we obtained a precise under-approximation of the set of terminating initial values for such programs.

These theoretical contributions are mathematical in nature with proofs that are quite technical. We showed, however, that these results can be directly applied in practical ways: one can use the ready-to-use formulas representing the ANT set provided in this article. Any static program analysis could incorporate, by a simple and direct instantiation techniques illustrated in our examples, the generic ready-to-use formulas representing precondition for (non-)termination, which were provided. This method was also used to tackle the termination and non-termination problem of linear/affine programs on rational or integer initial values, leading to new decidability results for these classes of programs.

# References

[1] Turing, A.M.: On computable numbers, with an application to the Entscheidungsproblem. Proceedings of the London Mathematical Society **2**(42) (1936) 230–265

[2] Cook, B., Gulwani, S., Lev-Ami, T., Rybalchenko, A., Sagiv, M.: Proving conditional termination. In: Proc. CAV. LNCS, Springer (2008) 328–340

[3] Colón, M., Sipma, H.: Synthesis of linear ranking functions. In: Proc. TACAS. LNCS, London, UK, Springer (2001) 67–81

[4] Colón, M.A., Sipma, H.B.: Practical methods for proving program termination. In: Proc. CAV. Volume 2404 of LNCS., Springer (2002) 442–454

[5] Bradley, A.R., Manna, Z., Sipma, H.B.: Linear ranking with reachability. In: Proc. CAV, Springer (2005) 491–504

[6] Bradley, A.R., Manna, Z., Sipma, H.B.: Termination analysis of integer linear loops. In: CONCUR, Springer (2005) 488–502

[7] Dams, D., Gerth, R., Grumberg, O.: A heuristic for the automatic generation of ranking functions. In: Workshop on Advances in Verification. (2000) 1–8

[8] Podelski, A., Rybalchenko, A.: A complete method for the synthesis of linear ranking functions. In: Proc. VMCAI. (2004) 239–251

[9] Tiwari, A.: Termination of linear programs. In: Proc. CAV. Volume 3114 of LNCS., Springer (2004) 70–82

[10] Braverman, M.: Termination of integer linear programs. In: Proc. CAV. Volume 4144 of LNCS., Springer (2006) 372–385

[11] Cousot, P., Cousot, R.: An abstract interpretation framework for termination. SIGPLAN Not. **47**(1) (January 2012) 245–258

[12] Gulwani, S., Srivastava, S., Venkatesan, R.: Program analysis as constraint solving. In: Proc. PLDI. ACM, ACM (2008) 281–292

[13] Bozga, M., Iosif, R., Konecený, F.: Deciding conditional termination. In: Proc. TACAS. LNCS, Springer (2012) 252–266

[14] Ganty, P., Genaim, S.: Proving termination starting from the end. In: CAV. (2013) 397–412

[15] Bradley, A.R., Manna, Z., Sipma, H.B.: Termination of polynomial programs. In: Proc. VMCAI. Volume 3385 of LNCS., Springer (2005) 113–129

[16] Chen, H.Y., Flur, S., Mukhopadhyay, S.: Termination proofs for linear simple loops. In: Proc. SAS. LNCS, Springer (2012) 422–438

[17] Cook, B., Podelski, A., Rybalchenko, A.: Termination proofs for systems code. SIGPLAN Not. **41**(6) (June 2006) 415–426

[18] Ben-Amram, A.M., Genaim, S., Masud, A.N.: On the termination of integer loops. In: Proc. VMCAI. LNCS (2012) 72–87

[19] Gupta, A., Henzinger, T.A., Majumdar, R., Rybalchenko, A., Xu, R.G.: Proving non-termination. In: Proc. POPL, ACM (2008) 147–158

[20] Alias, C., Darte, A., Feautrier, P., Gonnord, L.: Multi-dimensional rankings, program termination, and complexity bounds of flowchart programs. In: Proc. SAS. LNCS, Springer (2010) 117–133

[21] Ben-Amram, A.M., Genaim, S.: On the linear ranking problem for integer linear-constraint loops. In: Proc. POPL, ACM (2013) 51–62

[22] Rebiha, R., Matringe, N., Moura, A.V.: Generating asymptotically non-terminant initial variable values for linear diagonalizable programs. In: Proc. SCSS. Volume 15 of EPiC. (2013) 81–92

[23] Rebiha, R., Matringe, N., Moura, A.V.: Characterization of termination for linear homogeneous programs. Technical Report IC-14-08, Institute of Computing, University of Campinas (March 2014)

[24] Stein, W., Joyner, D.: SAGE: System for Algebra and Geometry Experimentation. ACM SIGSAM Bulletin, volume 39, number 2, pages 61–64 (2005)

# A   Appendix

## A.1   Proofs of Section 3

*Proof.* [**Theorem 3.1**] It is clear that if $P(\mathbf{A}, \mathbf{f})$ is $NT$, it is $ANT$ as a $NT$ value of $P(\mathbf{A}, \mathbf{f})$ is of course $ANT$ (with $k_\mathbf{x} = 0$). Conversely, if $P(\mathbf{A}, \mathbf{f})$ is $ANT$, call $\mathbf{x}$ an $ANT$ value, then $A^{k_\mathbf{x}}(\mathbf{x})$ is a $NT$ value of $P(\mathbf{A}, \mathbf{f})$, so $P(\mathbf{A}, \mathbf{f})$ is $NT$. The assertion for $K$-stable subspaces of $E$ is obvious, the proof being the same, as if $x \in K$ is $ANT$, we have $A^{k_\mathbf{x}}(\mathbf{x}) \in K$. □

*Proof.* [**Corollary 3.1**] As $NT(P(\mathbf{A}, \mathbf{f})) \subseteq ANT(P(\mathbf{A}, \mathbf{f}))$, passing to complementary sets gives the result. □

## A.2 Proofs of Section 4

*Proof.* [**Proposition 4.2**] We write $\mathbf{x}^+ = \mathbf{x}_1 + \cdots + \mathbf{x}_t$, with $\mathbf{x}_i \in E_{\lambda_i}(\mathbf{A})$. There are polynomials $P_1, \ldots, P_t$ in $\mathbb{R}[T]$, such that $\mathbf{f}(\mathbf{A}^k(\mathbf{x})) = \lambda_1^k P_1(k) + \cdots + \lambda_t^k P_t(k)$. Let $k_0$ be the smallest integer $i$ such that $P_i$ is nonzero, and set $\lambda = \lambda_i$. Then $\mathbf{f}(\mathbf{A}^k(x))$ becomes equivalent for $k$ large, to $a\lambda^k k^m$ for some $a$ the leading coefficient of $P_{k_0}$, and $m$ the degree of $P_{k_0}$. On the other hand, according to Theorem 4.1, the program $P(\mathbf{A}, \mathbf{f})$ is terminating on $E'$, hence on $\mathbf{x}'$, and more generally on any $\mathbf{A}^l(x')$ for $l \geq 0$ because $E'$ is $\mathbf{A}$-stable. In particular, there is an infinity of $l \geq 0$, such that $\mathbf{A}^l(x^-)$ is $\leq 0$. This implies that if $P(\mathbf{A}, \mathbf{f})$ is asymptotically non terminating on $\mathbf{x} = \mathbf{x}^+ + \mathbf{x}'$, then we have $a > 0$, and thus $P(\mathbf{A}, \mathbf{f})$ is asymptotically non terminating on $x^+$. $\square$

*Proof.* [**Theorem 4.2**] Indeed, if one considers $\mathbf{A}^2$ instead of $\mathbf{A}$, then $\mathbf{x}^r$ is equal to $\mathbf{x}^+$ for $\mathbf{A}^2$, and $\mathbf{x}^{nr}$ corresponds to $\mathbf{x}'$ for $\mathbf{A}^2$. As $P(\mathbf{A}, \mathbf{f})$ is ANT on $\mathbf{x}$, so is $P(\mathbf{A}^2, \mathbf{f})$. But according to the previous proposition, this means that $P(\mathbf{A}^2, \mathbf{f})$ is ANT on $\mathbf{x}^+$. Similarly, $\mathbf{A}(\mathbf{x}^r) = \mathbf{A}(\mathbf{x})^r$, is equal to $\mathbf{A}(\mathbf{x})^+$ for $\mathbf{A}^2$, and $\mathbf{A}(\mathbf{x}^{nr}) = \mathbf{A}(\mathbf{x})^{nr}$, is equal to $\mathbf{A}(\mathbf{x})'$ for $\mathbf{A}^2$. Again, $P(\mathbf{A}^2, \mathbf{f})$ is *ANT* on $\mathbf{A}(\mathbf{x})$, hence by the same argument, it is ANT on $\mathbf{A}(\mathbf{x}^r)$. We thus conclude that $P(\mathbf{A}, \mathbf{f})$ is ANT on $\mathbf{x}^r$. $\square$

*Proof.* [**Theorem 4.3**] It is clear that if $ANT^r(P(\mathbf{A}, \mathbf{f}))$ is non empty, then $ANT(P(\mathbf{A}, \mathbf{f}))$ is not. The converse is a consequence of Theorem 4.2. The last equality is almost by definition. $\square$

## A.3 Proofs of Section 5

*Proof.* [**Proposition 5.1**] Suppose that $P(\mathbf{A}, \mathbf{F})$ is *ANT*. Then there is $\mathbf{x}$ in $\bigcap_i ANT(P(\mathbf{A}, \mathbf{f}_i))$. Now we write $x = x^r + x^{nr}$ in a unique way. According to Theorem 4.2, we know that $x^r$ is is *ANT* for every $P(\mathbf{A}, \mathbf{f}_i)$, hence it belongs to $\bigcap_i ANT^r(P(A, \mathbf{f}_i))$ which can't be empty. $\square$

*Proof.* [**Theorem 5.1**] We recall that $P(\mathbf{A}, \mathbf{F})$ is *NT* if and only if it is *ANT* thanks to Lemma 5.2. The proof then follows from Proposition 5.1 and the inclusion $ANT^r(P(\mathbf{A}, \mathbf{F})) \subset ANT(P(\mathbf{A}, \mathbf{F}))$. $\square$

*Proof.* [**Theorem 5.2**] To say that $(x, 1)^\top$ is *ANT* for $P(A', F')$ means that there exists $k_x$, such that if $k \geq k_x$, then $F'A'^k \cdot (x, 1)^\top$ is $> 0$. But as $A' \cdot (x, 1)^\top = \begin{pmatrix} Ax + c \\ 1 \end{pmatrix} = (x_1, 1)^\top$, by induction, we obtain $A'^k \cdot (x, 1)^\top = (x_k, 1)^\top$. Now, we obtain the relation $F'A'^k \cdot (x, 1)^\top = F' \cdot (x_k, 1)^\top = \begin{pmatrix} Bx_k - b \\ 1 \end{pmatrix}$. Hence $F'A'^k \cdot (x, 1)^\top > 0$ is equivalent to $Fx_k > b$, and the result follows. $\square$

## A.4    Proofs of Section 6

*Proof.* [**Proposition 6.1**] If $A$ has a real spectrum, then $Spec(A') = Spec(A) \cup \{1\}$ is also real. In particular $ANT(P(A', F')) = ANT^r(P(A', F'))$. The result follows. □

*Proof.* [**Theorem 6.1**] We saw that $x \in K$ is ANT (resp. NT) for $P(A, F, b, c)$ if and only if $x' = (x, 1)^\top$ is ANT (resp. NT) for $P(A', F')$, with $A' = \begin{bmatrix} A & c \\ & 1 \end{bmatrix}$, and $F' = \begin{bmatrix} F & -b \\ 0 & 1 \end{bmatrix}$. We then apply Theorem 3.1, to the subset $K' = \{x', x \in K\}$ of $\mathbb{R}^{n+1}$, which is $A'$-stable. □

## A.5    Proofs of Section 7

*Proof.* [**Lemma 7.1**] Let $(e_1, \ldots, e_{d_\lambda})$ is a Jordan basis of $E_\lambda(\mathbf{A})$, i.e. a basis $J_\lambda$ (which exists by classical linear algebra) of $E_\lambda(\mathbf{A})$ such that $Mat_{J_\lambda}(\mathbf{A}) = diag(U_{\lambda,1}, \ldots, U_{\lambda,r_\lambda})$, where each $U_{\lambda,i}$ is of the form $\begin{pmatrix} \lambda & 1 & & & \\ & \lambda & 1 & & \\ & & \ddots & \ddots & \\ & & & \lambda & 1 \\ & & & & \lambda & 1 \\ & & & & & \lambda \end{pmatrix}$. We simply take $B_\lambda = (e_1, \lambda^{-1}e_2, \ldots, \lambda^{1-d_\lambda}e_{d_\lambda})$. □

*Proof.* [**Proposition 7.1**] If it was not the case, $\mathbf{A}$ would have two linearly independent eigenvectors $v$ and $w$ associated to $\lambda$. But as $K(\mathbf{A}, \mathbf{f})$ is zero, $\mathbf{f}(\mathbf{A}^k(v))$ is non constantly zero. As it is equal to $\lambda^k \mathbf{f}(v)$, we obtain $\mathbf{f}(v) \neq 0$, hence we can actually normalize $v$ so that $\mathbf{f}(v) = 1$. Similarly, we can suppose that $\mathbf{f}(w) = 1$. But then, $\mathbf{f}(\mathbf{A}^k(v - w))$ is constantly 0, i.e. $v - w \in K(\mathbf{A}, \mathbf{f})$, which contradicts $K(\mathbf{A}, \mathbf{f}) = \{0\}$. □

*Proof.* [**Lemma 7.2**] For $k = 1$, it is by definition of $T_\lambda$. We now do the induction step from $k$ to $k + 1$, for $k \geq 1$. We have $(T_\lambda)^{k+1} = T_\lambda(T_\lambda)^k$. But multiplying on the left by $T_\lambda$ a matrix $A$, amounts to replace every row of $A$, by this row added with the same one. But by Pascal's triangle equality, we have $\binom{k}{i} + \binom{k}{i+1} = \binom{k+1}{i+1}$, and this shows the induction step. □

*Proof.* [**Proposition 7.2**] It is a consequence of Lemma 7.2, as $Mat_B(\mathbf{f} \circ \mathbf{A}^k) = Mat_B(\mathbf{f})Mat_B(\mathbf{A}^k)$. □

*Proof.* [**Proposition 4**] As $P_\mu$ is zero as soon as $|\mu| > \lambda$ we know that, asymptotically, $f(\mathbf{A}^k(x))$ is equivalent to $\lambda^k(P_{|\lambda|}(x_{|\lambda|}, k) + (-1)^k P_{-|\lambda|}(x_{-|\lambda|}, k))$. So we have that $f(\mathbf{A}^{2k}(x))$ is equivalent to $\lambda^{2k}Q_{\pm\lambda}^+(x_{\pm\lambda}, 2k)$, and $f(\mathbf{A}^{2k+1}(x))$ is equivalent to $\lambda^{2k+1}Q_{\pm\lambda}^-(x_{\pm\lambda}, 2k + 1)$. In particular, for $k$ large enough, both quantities $\lambda^{2k}Q_{\pm\lambda}^+(x_{\pm\lambda}, 2k)$ and $\lambda^{2k+1}Q_{\pm\lambda}^-(x_{\pm\lambda}, 2k+1)$ will be positive by our assumption on the dominant terms of $Q_{\pm\lambda}^+$ and $Q_{\pm\lambda}^-$. This means that $x$ is $ANT$. □

*Proof.* [**Proposition 7.4**] One just needs to expand the polynomials $Q^+_{\pm\lambda}(x_{|\lambda|})$, $Q^-_{\pm\lambda}(x_{|\lambda|})$, and $P_\mu(x_\mu)$. Their coefficients are the linear forms $\phi_{\pm\lambda,i}(x_\lambda, x_{-\lambda})$ and $\phi_{\mu,j}(x_\mu)$ involved in the statement. Now we simply express the fact that a polynomial is zero if and only if its coefficients are zero, and that $Q^+_{\pm\lambda}(x_{|\lambda|})$ and $Q^-_{\pm\lambda}(x_{|\lambda|})$ have positive dominant coefficient if and only if their first coefficients are zero, and the first nonzero one occurring is positive.                                                                    $\square$

*Proof.* [**Proposition 5**] Because of our first condition, the quantity $\mathbf{f}(\mathbf{A}^{2k}(x))$ is asymptotically equivalent to $\lambda^{2k}Q^+_{\pm\lambda}(x_{\pm\lambda}, 2k)$ for $k$ large. Thanks to our second condition, the quantity $f(\mathbf{A}^{2k+1}(x)$ is asymptotically equivalent to $|\lambda'|^{2k+1}Q^-_{\pm\lambda'}(x_{\pm\lambda'}, 2k+1)$. In both cases, as $Q^+_{\pm\lambda}(x_{\pm\lambda})$ and $Q^-_{\pm\lambda'}(x_{\pm\lambda'})$ both have positive dominant term, we conclude that $\mathbf{f}(\mathbf{A}^{2k}(x))$ and $\mathbf{f}(\mathbf{A}^{2k+1}(x))$ are both positive when $k$ is large.              $\square$

*Proof.* [**Proposition 7.6**] Similar Proposition 7.4's proof.                                   $\square$

*Proof.* [**Proposition 6**] Because of our first condition, the quantity $f(\mathbf{A}^{2k+1}(x))$ is asymptotically equivalent to $\lambda^{2k+1}Q^-_{\pm\lambda}(x_{\pm\lambda}, 2k+1)$ for $k$ large. Thanks to our second condition, $f(\mathbf{A}^{2k}(x))$ is asymptotically equivalent to $|\lambda'|^{2k}Q^+_{\pm\lambda'}(x_{\pm\lambda'}, 2k)$. In both case, as $Q^-_{\pm\lambda}(x_{\pm\lambda})$ and $Q^+_{\pm\lambda'}(x_{\pm\lambda'})$ both have positive dominant term, we conclude that $f(\mathbf{A}^{2k}(x))$ and $f(\mathbf{A}^{2k+1}(x))$ are both positive when $k$ is large.              $\square$

*Proof.* [**Proposition 7.8**] Similar to Proposition 7.4's proof.                               $\square$

*Proof.* [**Theorem 7.2**] Let $x$ be an *ANT* point. If the eigenvalue $\lambda$ of largest absolute value such that $P_\lambda(x_\lambda)$ is nonzero, was negative, and $P_{-\lambda}(x_{-\lambda})$ was equal to zero, then $\mathbf{f}(\mathbf{A}^k(x)$ would be asymptotically equivalent to $\lambda^k P_\lambda(x_\lambda, k)$. As $P_\lambda(x_\lambda, k)$ is asymptotically of the sign of its dominant term, and $\lambda^k$ is alternatively positive and negative, the program $P(\mathbf{A}, \mathbf{f})$ would terminate on $x$. Hence, if $\lambda$ is of largest absolute value such that $P_\lambda(x_\lambda)$ is nonzero, then $P_{|\lambda|}(x_{|\lambda|})$ is nonzero, and we can actually suppose that $\lambda$ is positive. If $Q^+_{\pm\lambda}(x_{\pm\lambda})$ and $Q^-_{\pm\lambda}(x_{\pm\lambda})$ are nonzero, as $\mathbf{f}(\mathbf{A}^{2k}(x)) \sim \lambda^{2k}Q^+_{\pm\lambda}(x_{\pm\lambda}, 2k)$ and $\mathbf{f}(\mathbf{A}^{2k+1}(x)) \sim \lambda^{2k+1}Q^-_{\pm\lambda}(x_{\pm\lambda}, 2k+1)$, they will both be positive for $k$ large if and only if $Q^+_{\pm\lambda}(x_{\pm\lambda})$ and $Q^-_{\pm\lambda}(x_{\pm\lambda})$ have a positive dominant term. In this case, we are in the situation of Proposition 4. If $Q^-_{\pm\lambda}(x_{\pm\lambda})$ is equal to zero, then $Q^+_{\pm\lambda}(x_{\pm\lambda})$ is not (otherwise $P_\lambda(x_\lambda)$ and $P_{-\lambda}(x_{-\lambda})$ would both be zero), so $\mathbf{f}(\mathbf{A}^{2k}(x)) \sim \lambda^{2k}Q^+_{\pm\lambda}(x_{\pm\lambda}, 2k)$, and $Q^+_{\pm\lambda}(x_{\pm\lambda})$ must have a positive dominant term. If $Q^-_{\pm\mu}(x_{\pm\mu})$ was zero for all eigenvalue $\mu$, we would have $\mathbf{f}(\mathbf{A}^{2k+1}(x)) = 0$ for all $k$, which is absurd because $x$ is ANT. Hence there is $\lambda'$ of absolute value as large as possible (with $|\lambda'| < \lambda$ necessarily), such that $Q^-_{\pm\lambda'}(x_{\pm\lambda'})$ is nonzero. In this case, $\mathbf{f}(\mathbf{A}^{2k+1}(x))$ is equivalent to $|\lambda'|^{2k+1}Q^-_{\pm\lambda'}(x_{\pm\lambda'})$, and as $x$ is ANT, this forces $Q^-_{\pm\lambda'}(x_{\pm\lambda'})$ to have a positive dominant term, and we are in the situation of Proposition 5. Finally, in the last case, $Q^+_{\pm\lambda}(x_{\pm\lambda})$ is equal to zero, and $Q^-_{\pm\lambda}(x_{\pm\lambda})$ is necessarily nonzero. As $\mathbf{f}(\mathbf{A}^{2k+1}(x)) \sim \lambda^{2k+1}Q^-_{\pm\lambda}(x_{\pm\lambda}, 2k+1)$, this implies that $Q^-_{\pm\lambda}(x_{\pm\lambda}, 2k+1)$ has a positive

dominant term. Again, if $Q_{\pm\mu}^+(x_{\pm\mu})$ was equal to zero, for all $k \geq 0$, we would have $\mathbf{f}(\mathbf{A}^{2k}(x)) = 0$, which is absurd as $x$ is ANT. Hence there is $\lambda'$ of absolute value as large as possible (with $|\lambda'| < \lambda$ necessarily), such that $Q_{\pm\lambda'}^+(x_{\pm\lambda'})$ is nonzero. In this case, $\mathbf{f}(\mathbf{A}^{2k}(x))$ is equivalent to $|\lambda'|^{2k}Q_{\pm\lambda'}^+(x_{\pm\lambda'})$, and as $x$ is ANT, this forces $Q_{\pm\lambda'}^+(x_{\pm\lambda'})$ to have a positive dominant term, and we are in the situation of Proposition 6. $\qquad\square$

## A.6 Proofs of Section 7.2

*Proof.* [**Proposition 7.9**] The restriction of $\mathbf{A}$ to $E_\lambda(\mathbf{A})$ admits a matrix of the form $T = \begin{pmatrix} \lambda & \cdots & \\ & \ddots & \vdots \\ & & \lambda \end{pmatrix}$ in a Jordan basis of $E_\lambda(\mathbf{A})$. It is easy to check, by induction on $d_\lambda$, using the theory of Bernoulli polynomials, that $T^k$ is upper triangular, with diagonal entries equal to $\lambda^k$, and non diagonal nonzero entries of the form $\lambda^k Q(k)$, for $Q \in \mathbb{R}[X]$ of degree $\leq d_\lambda$. The result follows. $\qquad\square$

*Proof.* [**Theorem 7.3**] As $\mathbf{f}(\mathbf{A}^k(x)) = \bar{\mathbf{f}}(\overline{\mathbf{A}}^k(\overline{x}))$, it is obvious that $x$ is *ANT* for $P(\mathbf{A}, \mathbf{f})$ if and only if $\overline{x}$ is *ANT* for $P(\overline{\mathbf{A}}, \overline{\mathbf{f}})$. Now if we write $\overline{x} = \overline{x}_0 + \overline{x}^a$, with $\overline{x}_0$ in $\overline{E}_0(\overline{\mathbf{A}})$, and $\overline{x}^a$ in $\overline{E}^a$, then for $k$ large, we have $\bar{\mathbf{f}}(\overline{\mathbf{A}}^k(\overline{x})) = \bar{\mathbf{f}}(\overline{\mathbf{A}}^k(\overline{x}^a))$. This means that $\overline{x}$ is *ANT* for $P(\overline{\mathbf{A}}, \overline{\mathbf{f}})$ if and only if $\overline{x}^a$ is *ANT* for $P(\overline{\mathbf{A}}^a, \overline{\mathbf{f}}^a)$. This ends the proof. $\qquad\square$

## A.7 Proofs of Section 8

*Proof.* [**Theorem 8.1**] For $A$ in $\mathcal{M}(n, \mathbb{R})$, let $\tilde{A}$ be the block diagonal matrix $diag(A, -A)$ of $\mathcal{M}(2n, \mathbb{R})$. The spectrum of $\tilde{A}$ is $Spec(A) \cup -Spec(A)$, hence if it contains $2n$ different elements, certainly, $A$ will not admit $\lambda$ and $-\lambda$ as simultaneous eigenvalues. But $Spec(\tilde{A})$ is of cardinality $2n$ if and only $P(A) = Disc(\chi_{\tilde{A}}) \neq 0$, where $Disc$ is the discriminant, and $\chi_{\tilde{A}}$ the characteristic polynomial of $\tilde{A}$. As the map $A \mapsto P(A)$ is polynomial, we see that $R(A) = \{A \in \mathcal{M}(n, \mathbb{R}), P(A) \neq 0\} = \{A \in \mathcal{M}(n, \mathbb{R}), |Spec(\tilde{A})| = 2n\}$ is a Zariski open subset, obviously non empty (take $A = diag(1, \ldots, n)$), hence $R(A) \times \mathbb{R}^n$. Now we are going to show that the set $R'(A) = \{(A, v) \in \mathcal{M}(n, \mathbb{R}) \times \mathbb{R}^n, K(A, v) = \{0\}\}$ is also a Zariski open and nonempty in $\mathcal{M}(n, \mathbb{R})$. Write $B(A, v) \in \mathcal{M}(n, \mathbb{R})$ the matrix the rows of which are $^t v, ^t vA, \ldots, ^t vA^{n-1}$. Then $R'(A) = \{(A, v) \in \mathcal{M}(n, \mathbb{R}) \times \mathbb{R}^n, det(B(A, v)) \neq 0\}$, hence is Zariski open in $\mathcal{M}(n, \mathbb{R}) \times \mathbb{R}^n$. To see that it is non empty, take $v$ the first vector of the canonical basis of $\mathbb{R}^n$, and $A$ the permutation matrix representing the cycle $(1, \ldots, n)$. Finally, the set $R(A, v)$ we are interested contains the non empty Zariski open set $R(A) \times \mathbb{R}^n \cap R'(A)$, which proves the statement. $\qquad\square$

*Proof.* [**Theorem 8.2**] To begin, one need to recall the following observation: if $P_{\lambda,j}$ is nonzero, its dominant term is the first nonzero $a_{\lambda,m}$, for $m$ between 1 and $j$. Sup-

pose that the $\lambda$ of largest absolute value such that a coefficient $a_{\lambda,j}x_{\lambda,j}$ is nonzero is positive, and that $a_{\lambda,j}x_{\lambda,j}$ is positive as well. In this case, $\mathbf{f}(\mathbf{A}^k(x))$ is equivalent to $\lambda^k P_\lambda(x_\lambda, k)$. We then recall that $P_\lambda(x_\lambda)$ is equal to $\sum_{j=1}^{d_\lambda} x_{\lambda,j} P_{\lambda,j}$. Finally, thanks to Proposition 7.2, we see that the dominant term of $P_\lambda(x_\lambda)$ is equal to $a_{\lambda,j_0}x_{\lambda,j_0}$, for the largest $j_0$ such that $a_{\lambda,j_0}x_{\lambda,j_0}$ is nonzero, and as it is positive, $x$ is $ANT$.

Conversely, if $x$ is $ANT$, all $P_\lambda(x_\lambda)$ can't be zero, otherwise $\mathbf{f}(\mathbf{A}^k(x))$ would be constantly zero, which is absurd. Let $\lambda$ be the eigenvalue of largest absolute value, such that $P_\lambda(x_\lambda)$ is nonzero. Then again, $\mathbf{f}(\mathbf{A}^k(x))$ is equivalent to $\lambda^k P_\lambda(x_\lambda, k)$, and as $P_\lambda(x_\lambda, k)$'s dominant term is equal to $a_{\lambda,j_0}x_{\lambda,j_0}$, for the largest $j_0$ such that $a_{\lambda,j_0}x_{\lambda,j_0}$ is nonzero, $\mathbf{f}(\mathbf{A}^k(x))$ is equivalent to $\lambda^k a_{\lambda,j_0}x_{\lambda,j_0}k^{j_0}$ for $k$ large. As $x$ is $ANT$, $\lambda$ must be positive, because otherwise $\mathbf{f}(\mathbf{A}^k(x))$ would alternatively change sign for $k$ large. Moreover, $a_{\lambda,j_0}x_{\lambda,j_0}$ must be positive. $\qquad\square$