Termination Of Linear Programs

BTech Project Stage I Report

Submitted in partial fulfillment of the requirements
for the degree of

Bachelor of Technology

by

Aaditya Kumar Ramdas
Roll No: 05005027

under the guidance of

Prof. Supratik Chakraborty

Department of Computer Science and Engineering
Indian Institute of Technology, Bombay
Mumbai

Abstract

Many real-time applications have linear programs, which can get a variety of inputs in different situations. It is possible that some of these are “bad” inputs which cause unexpected failures like non-termination of the program. Such instances can lead to disastrous consequences and the programs must be verified to ensure that they either terminate or the inputs under which they don’t terminate are unreachable.

We will show that certain questions are decidable and algorithms have been developed for them. We will then proceed to analyse another type of program which has no decidable algorithm yet. We will break it down into solved cases and then pose an interesting open problem for future work.

1 Introduction

Interest has been growing in the termination of linear programs problem. This problem tries to see if there is an input for which the program does not terminate (or terminates only under certain conditions).

Ashish Tiwari's [1] introduces a decidable procedure to check for termination for a simple subset of programs with a single while loop. He does this by looking for a non-termination signature amongst the eigenvalues of the transformation matrix, and proves that it is sufficient to do so.

After discussing the above paper, we then consider a similar version of the problem, where we ask if the program will terminate for one particular input. This is also obviously closely related with the values of the eigenvalues, their signs, their realness (whether they are real or complex) and their multiplicities.

We then break the problem into parts and solve most of them in a later section and then mention some thoughts for future work (for the second stage of the BTech project) that might help to solve the remaining parts in our attempt to get a decidable procedure for checking termination.

2 Notations

We first define some notations. Whenever we refer to square matrices, we use uppercase bold symbols: $\mathbf{A}$. Column vectors are represented using lowercase bold symbols: $\mathbf{c}$. Single variables are simply represented using lowercase symbols: x, y. A transpose of a vector $\mathbf{c}$ is represented as $\mathbf{c}^T$.

A *piecewise linear program*, or a *linear program* in short, is one which has only linear assignments of the form $\mathbf{x} = \mathbf{Ax} + \mathbf{c}$, and loop and if conditions of the form $\mathbf{Bx} > \mathbf{c}$, $\mathbf{Bx} \geq \mathbf{c}$. A *homogeneous* program is one in which $\mathbf{c}$ in both the conditions or linear assignments is 0. A *non-homogeneous* program could have $\mathbf{c}$ as any constant vector.

We will also denote matrices by specifying the submatrices inside it. So, for instance, $diag(J_1, ..., J_K)$ would denote a matrix which has matrices $J_1, ..., J_K$ on its "diagonal" and 0 elsewhere. If $\mathbf{A}$ is a (N x N)-matrix and $\mathbf{c}$ is a vector such that $\mathbf{Ac} = \lambda\mathbf{c}$, then $\mathbf{c}$ is called an eigenvector of $\mathbf{A}$ corresponding to the eigenvalue λ. The effect of repeated linear assignments ($\mathbf{x}$:= $\mathbf{Ax}$) becomes much more explicit when we do a change of variables and let the new variables $\mathbf{y}$ be the eigenvectors of $\mathbf{A}$. In particular, we get transformed assignments of the form $\mathbf{y} := \lambda\mathbf{y}$. If there are N linearly independent eigenvectors, then $\mathbf{A}$ is said to be diagonalizable and the assignments on the new variables will be of the form $\mathbf{y} := diag(\lambda_1, ..., \lambda_N)\mathbf{y}$. But this is not possible always. However, instead of a diagonal matrix, we can always get an almost diagonal, the so-called Jordan form, matrix.

3 The Basic Homogenous Program

We will first consider linear programs and theorems that were proved in Ashish Tiwari's [1] where the programs are of the following homogenous form :

`P1: while` $(\mathbf{c}^T\mathbf{x} > 0)$ $\{$ $\mathbf{x} := \mathbf{Ax}$ $\}$

Theorem 1. *If the linear loop program P1, defined by an (N* x *N)-matrix* $\boldsymbol{A}$ *and a nonzero (N* x *1) vector* $\mathbf{c}$*, is non-terminating, then there exists a real eigenvector* $\boldsymbol{v}$ *of* $\boldsymbol{A}$ *corresponding to a positive eigenvalue, such that* $\mathbf{c}^T\boldsymbol{v} \geq 0$*.*

Corollary 1. *If there is no real eigenvector* $\boldsymbol{v}$ *of* $\boldsymbol{A}$ *such that* $\mathbf{c}^T\boldsymbol{v} = 0$*, then the linear loop program defined by* $\boldsymbol{A}$ *and* $\mathbf{c}$ *is nonterminating iff there exists an eigenvector* $\boldsymbol{v}$ *on which the loop is nonterminating.*

We can also generalise the loop condition to allow for a conjunction of multiple linear inequalities. We continue to assume that all linear inequalities and linear assignments consist only of homogenous expressions. Let $\mathbf{B}$ be a (M x N)-matrix (with rational entries) and $\mathbf{A}$ be a (N x N)-matrix. We consider programs of the following form:

`P2: while` $(\mathbf{Bx} > 0)$ $\{$ $\mathbf{x} := \mathbf{Ax}$ $\}$

Theorem 1 and Corollary 1 immediately generalize to programs of the form P2.

Theorem 2. If Program P2, specified by matrices $\mathbf{A}$ and $\mathbf{B}$, is nonterminating, then there is a real eigenvector $\mathbf{v}$ of $\mathbf{A}$, corresponding to a positive real eigenvalue, such that $\mathbf{Bv} \geq 0$.

Corollary 2. Assume that for every real eigenvector $\mathbf{v}$ of $\mathbf{A}$, corresponding to a positive eigenvalue, whenever $\mathbf{Bv} \geq 0$, then it is actually the case that $\mathbf{Bv} > 0$. Then, the Program P2, defined by $\mathbf{A}$ and $\mathbf{B}$, is nonterminating iff there exists an eigenvector $\mathbf{v}$ on which the loop is nonterminating.

3.1 Reducing the Homogenous Case

Corollary 2 falls short of yielding decidability of termination of homogeneous linear programs. But it hints that the real eigenvalues and the corresponding eigenvectors are relevant for termination characteristics of such programs. We can show that the nonpositive eigenvalues (and the corresponding eigenspace) can be ignored and the termination problem can be reduced to only the eigenspace corresponding to positive real eigenvalues of the matrix $\mathbf{A}$. Note that Program P2 can be transformed by an invertible transformation, preserving its termination properties.

Proposition 1. Let P be an invertible linear transformation. The program
`P2: while` $(\mathbf{Bx} > 0)$ $\{$ $\mathbf{x} := \mathbf{Ax}$ $\}$
is terminating iff the program
`P3: while` $(\mathbf{BPy} > 0)$ $\{$ $\mathbf{y} := \mathbf{P}^{-1}\mathbf{APy}$ $\}$
is terminating.

Proof. If Program P2 does not terminate on input $\mathbf{x} := \mathbf{c}$, then Program P3 will not terminate on input $\mathbf{y} := \mathbf{P}^{-1}\mathbf{c}$. Conversely, if Program P3 does not terminate on input $\mathbf{y} := \mathbf{d}$, then Program P2 will not terminate on input $\mathbf{x} := \mathbf{Pd}$.

So, as in [3] let $\mathbf{P}$ be the real (N x N)-matrix such that $\mathbf{P}^{-1}\mathbf{AP} = diag(\mathbf{J}_1,...,\mathbf{J}_K)$. Thus, Program P2, specified by matrices $\mathbf{A}$ and $\mathbf{B}$ can be transformed into the new Program P3

`P3: while` $(\mathbf{BPy} > 0)$ $\{$ $\mathbf{y} := diag(\mathbf{J}_1,...,\mathbf{J}_K)\mathbf{y}$ $\}$.

Proposition 1 means that we can focus on termination of Program P3. Partition the variables in $\mathbf{y}$ into $\mathbf{y}_1$, $\mathbf{y}_2$,...,$\mathbf{y}_K$ and partition matrix $\mathbf{BP}$ into $\mathbf{B}_i$s and rewrite the Program P3 as

`P3': while` $(\mathbf{B}_1\mathbf{y}_1 +...+ \mathbf{B}_K\mathbf{y}_K > 0)$ $\{$ $\mathbf{y}_1 := \mathbf{J}_1\mathbf{y}_1; \ldots ; \mathbf{y}_K := \mathbf{J}_K\mathbf{y}_K$ $\}$

Let S_+ be the set of indices of all those $\mathbf{J}_i$ which correspond to real and strictly positive eigenvalues. Then, we have the following lemma which says that we can ignore the state space corresponding to negative and complex eigenvalues while still preservind the termination behaviour of program P3.

`P4: while` $(\Sigma j \in S_+ \ \mathbf{B}_j\mathbf{y}_j > 0)$ $\{$ $\mathbf{y}_j = \mathbf{J}_j\mathbf{y}_j$; `for` $j \in S_+$ $\}$

Lemma 1. Program P3 is terminating iff the program P4 terminates.

Lemma 1 reduces the termination of Program P2 to testing termination of Program P4. Tiwari [1] gives a nondeterministic procedure *NP* which returns "nonterminating" iff the program P4 is nonterminating. This helps us complete the algorithm for decidability of P2.

3.2 Decidability of P2

Theorem 4. The termination of a homogenous linear program of the form is decidable
`P2: while` $(\mathbf{Bx} > 0)$ $\{$ $\mathbf{x} := \mathbf{Ax}$ $\}$

Proof. We decide termination of the Program P2 as follows: If $\mathbf{A}$ has no positive real eigenvalues, then return "terminating" (Corollary 2). If every real eigenvector $\mathbf{v}$ corresponding to a positive real eigenvalue of A satisfies $\mathbf{Bv} < 0$, then return "terminating" (Theorem 2). If there is a real eigenvector $\mathbf{v}$ corresponding to a positive real eigenvalue of $\mathbf{A}$ such that $\mathbf{Bv} > 0$, then return "nonterminating" (Corollary 2). If none of the above cases is true, then clearly $\mathbf{A}$ has positive real eigenvalues. Compute the Jordan blocks and the generalized eigenvectors only corresponding to the positive real eigenvalues of $\mathbf{A}$. Generate a transformation P by extending the computed set of generalized eigenvectors with any set of vectors in space orthogonal to that of the generated eigenvectors. Transform Program P2 by P as in Proposition 1. It is an easy exercise to note that we can apply Lemma 1 and reduce the termination problem to that for Program P4 of Lemma 1. Finally, we decide termination of Program P4 using the nondeterministic procedure *NP* and Lemma 2.

4 The Initialized Homogenous Program

We continue to follow the notation and terminology used in the previous section. Consider the problem of termination of a single-update (assuming simultaneous updates) single-loop-condition while loop with initialization:

$\mathbf{x}$=$\mathbf{b}$; `while` ($\mathbf{c}^T\mathbf{x} < 0$) { $\mathbf{x}$=$\mathbf{Ax}$ }

When there are multiple inequalities in the loop condition, if any of the inequalities is violated then the program terminates, so our analysis can be easily carried forward to multiple inequalities, since we will anyway consider one inequality at a time over several iterations of the loop.

P_0: $\mathbf{x}$=$\mathbf{b}$; `while` ($\mathbf{Bx} < 0$) { $\mathbf{x} := \mathbf{Ax}$ }

P_1: $\mathbf{y}$=$\mathbf{P}^{-1}\mathbf{b}$; `while` ($\mathbf{BPy} < 0$) { $\mathbf{y} := \mathbf{P}^{-1}\mathbf{APy}$ }

Theorem A. *Let P be an invertible linear transformation. The program P_0 is terminating iff the program P_1 is terminating.*

As before, using [3], let $\mathbf{P}$ be the real (N x N)-matrix such that $\mathbf{P}^{-1}\mathbf{AP} = diag(\mathbf{J}_1,...,\mathbf{J}_K)$.

Partition the variables in $\mathbf{y}$ into $\mathbf{y}_1$, $\mathbf{y}_2$,...,$\mathbf{y}_K$, and partition the matrix $\mathbf{BP}$ into $\mathbf{B}_i$s, and rewrite the Program P_1 as

P_1': `while` ($\mathbf{B}_1\mathbf{y}_1$ +...+ $\mathbf{B}_K\mathbf{y}_K < 0$) { $\mathbf{y}_1 := \mathbf{J}_1\mathbf{y}_1$; ... ; $\mathbf{y}_K := \mathbf{J}_K\mathbf{y}_K$ }

Now, consider the following program (S_+ is the set of indices that represent positive eigenvalues).

P_2: `while` ($\Sigma j \in S_+\ \mathbf{B}_j\mathbf{y}_j < 0$) { $\mathbf{y}_j = \mathbf{J}_j\mathbf{y}_j$; `for` $j \in S_+$ }

The Jordan blocks corresponding to the negative and complex eigenvalues are ignored in the condition within the while loop. The following result holds :

Theorem B. *If P_2 terminates on $\boldsymbol{b}$, then P_1 terminates on $\boldsymbol{b}$.*

Since the $\mathbf{y}_i$ corresponding to negative and complex eigenvalues do not matter, the updates within the while loop corresponding to them can be removed without affecting the termination of the program. If P_2 terminates, then we can be certain that the original P_1, and therefore P_0 terminates. However, if P_2 does not terminate, termination or non-termination of P_1 is unknown.

4.1 Termination of P_1

Note that $\mathbf{y}_i$ after k iterations is equal to $\mathbf{J}_i^k\ \mathbf{y}_{i0}$. It can be shown that the highest power of λ_i that will be present in any of the components of $\mathbf{y}_i$ is λ_i^k, and the powers of k range from k^{m_i-1}

to k^0, where (m_i x m_i) is the size of the Jordan block matrix corresponding to λ_i.

We take λ_i^k common (since the size of the Jordan block matrix is constant) and see that each component of $\mathbf{y}_i$ equals $\lambda_i^k(a_{i1}k^{m_i-1} + a_{i2}k^{m_i-2} + ... + a_{im_i})$, where k is the number of iterations and a_{ij} are constants. None of the a_{ij} depend on k; they depend only on m_i, which are known.

Each inequality in the while condition can be expressed as a linear combination of $\mathbf{y}_i$, i.e. as

$$\Sigma_{i=1}^{K} T_i < 0$$

$$.$$

$$T_i = \lambda_i^k(a_{i1}k^{m_i-1} + a_{i2}k^{m_i-2} + ... + a_{im_i})$$

Even if one of these inequalities is violated, the while loop terminates. Thus, to solve the problem of termination, we are checking if $I = \Sigma_{i=1}^{K} T_i$ can ever be greater than or equal to 0.

4.2 Analysis of Single Inequality $\Sigma_{i=1}^{K} T_i < 0$

A procedure to analyse I given that all the λ_i are positive reals and a_i are real, is also assumed. This procedure is called `PosTerm` [3].

Consider the case of T_i with complex λ_i. Each such λ_i can be expressed as $r_i(cos\theta_i + isin\theta_i)$. Since $\Sigma_{i=1}^{K} T_i$ evaluates to a real number, the complex parts of each T_i are ignored. The polynomial to which λ_i^k is multiplied can be split up into two parts, one corresponding to $sin(k\theta_i)$, and one to $cos(k\theta_i)$, i.e. $real(T_i) = r_k(cos(k\theta_i)g_i(k) + sin(k\theta_i)h_i(k))$, where both h and g are real polynomials.

Assume that $b = argmax_{i=1}^{K}\lambda_i$. b is nothing but the index of the λ term which dominates.

b is unique We can divide this into 3 cases : $\lambda_b < 0$, λ_b is complex and $\lambda_b > 0$.

If $\lambda_b < 0$, then it is clear that the term $\|\lambda_b\|^k a_{b1}k^{m_b-1}$, will dominate the expression $\Sigma_{i=1}^{K} T_i$ after some k. Since this term keeps flipping signs based on whether k is even or odd, overall, the inequality $I < 0$ will always be violated.

If λ_b is complex, then $\theta \neq 0$ and $\theta \neq \pi/2$. If θ is a rational angle in degrees, ie $\theta = p/q$. Then there is a k, occurring infinitely often such that $k\theta$ is a multiple of 360. We can prove [2] that T_b dominates the sum I beyond a point, infinitely often (repetitively) and even when T_b dominates the sum, it keeps flipping signs infinitely often and hence violates the inequality. If θ is not rational, then by multiplying by suitable integers, we can get it as close to 90, 180, 270, 360 as we want. Hence, again when the λ_b term starts dominating, the term also keeps flipping signs infinitely often, and hence has to violate the inequality.

If $\lambda_b > 0$, and the coefficient of the leading term in the k-polynomial is positive, then since the term corresponding to λ_b multiplied with this leading term has to dominate, the inequality will always be violated. If the coefficient is negative, we can use approximations [2] to bound the number of iterations to be checked for the violation of the inequality, and also use `PosTerm` to return an upper-bound on number of iterations.

No unique largest magnitude b For all such $j = argmax_{i=1}^{K}\lambda_i$, compare the largest power of the k that they are multiplied with, if there is a unique j such that the largest power of k is the maximum over all js, let $b = j$. Clearly, the term corresponding to b dominates. The algorithm in the previous section can be used once again to determine violation.

However, if this is not the case, a definite algorithm to decide termination has not yet been figured out. This has been further broken down into cases [2], some of which have not been solved, and are of research interest.

4.3 Overview of `Posterm`

Let $b = argmax_{i=1}^{K}\lambda_i$ and let $f(i)$ = sign of the leading coefficient in T_i (1 if positive, -1 if negative).

If $a_{b1} > 0$, the leading term of T_b is positive. Since exponential growth is much faster than polynomial growth, T_b will eventually be positive, and will dominate over all other T_i, irrespective of the m_i. Thus I will eventually be positive.

If $a_{b1} < 0$, the dominant term has a negative coefficient, and inequality I will eventually always be negative. In this case, we need to bound the number of iterations k_0 uptil which we need to check for termination. Since this equation is heavily non-linear, approximate methods are used to figure out k_0 such that $\Sigma_{i=1}^{K}T_i < 0$ after all k beyond k_0. The details are available in [2].

4.4 Description of `Poly`

Given a polynomial with real coefficients, $\Sigma a_i x_i$, a procedure to compute an integer x_0 such $a_i x_i$ evaluates to the same sign (either always positive, or always that for all integer $x > x_0$, negative) is described. This procedure is called `Poly`, and it returns x_0.

Assume without loss of generality that the polynomial is always greater than 0 after some x_0 since $a_n > 0$. Then, for large $x > x_0$, we have

$$\Sigma_{i=0}^{n} a_i x^i > a_n x^n + (\Sigma_{i:a_i<0}\ a_i)x^{n-1} > 0$$

Hence, `Poly` will return the value $x_0 = -\frac{\Sigma_{i:a_i<0} a_i}{a_n}$.

The problem of termination is unsolvable only in the case when *the maximum-magnitude λ is not unique, it is complex, the associated polar angle is irrational in degrees, and the associated Jordan matrices have the same size m_b.* In this case, the dominating term becomes equal to $\lambda_1^k a_1 k^{m_b-1} + \lambda_2^k a_2 k^{m_b-1}$ where $\lambda_1 = (r, \theta_1)$ and $\lambda_2 = (r, \theta_2)$

$$f(k) = A_1 cos(k\theta_1) + A_2 cos(k\theta_2)$$

This seems to be a fairly simple expression, but it has some interesting characteristics. A_1 and A_2 are any constants, while θ_1 and θ_2 are different irrational angles in degrees.

Is $k\theta$ dense?

First, let us look at a result which states that if θ_1 is irrational, then the set $\{k\theta\}$ (for integer k) is dense in the circle.

Theorem *Let ϕ be an irrational angle. Then $\{n\phi \mid n \in N\}$ is a dense subset of the set $[0, 360)$ of all angles. This means, for every $\delta \in [0, 360)$ and every $N \in N$ there exists some $n \in N$ such that $|n\phi \ - \ \delta| < \frac{1}{N}360$.*

Proof. Consider the $N + 1$ angles ϕ, 2ϕ, ..., $(N + 1)\phi$. As they are all irrational, each one of them is located in one of the N segments $(0, \frac{1}{N}360)$, $(\frac{1}{N}360, \frac{2}{N}360)$, ..., $(\frac{N-2}{N}360, \frac{N-1}{N}360)$ and $(\frac{N-1}{N}360, 360)$.

By Dirichlet's box principle or the pigeonhole principle, there are two angles $a\phi$, $b\phi$ $(a < b)$ within the same segment. It follows that $0 < s\phi < \frac{1}{N}360$ where $s := (b - a)$.

Put $M := [360/s\phi]$. Clearly, $M > N$. Now, let $\delta \in (0, 360)$ be an arbitrary angle. We put $R := max\{r \in \{0, 1, ..., M\} \mid rs\phi \leq \delta\}$.

Then, $Rs\phi \leq \delta \leq (R + 1)s\phi$. Which implies that $|\delta - (R + 1)s\phi| \ \leq s\phi < \frac{1}{N}360$. As $N \in N$ may still be chosen freely, we see that there are multiples of ϕ arbitrarily close to δ.

If $A_1 \neq A_2$

We are looking at $f(k) = A_1 cos(k\theta_1) + A_2 cos(k\theta_2)$. Assume without loss of generality that $|A_1| > |A_2|$. We can apply a similar argument in the other case.

Now, since $k\theta$ is dense, $k\theta_1$ will take values arbitrarily close to 0 infinite number of times. Consider all these ks. Whatever the value of $cos(k\theta_2)$, the value of $f(k)$ will fluctuate between $A_1 - A_2$ and $A_1 + A_2$, which are all of the same sign as A_1.

Similarly, $k\theta_1$ will take values arbitrarily close to 180 infinite number of times. Consider all these ks. Whatever the value of $cos(k\theta_2)$, the value of $f(k)$ will fluctuate between $-A_1 - A_2$ and $-A_1 + A_2$, which are all of the same sign as $-A_1$.

Hence, $f(k)$ takes both positive and negative signs, as long as there is a finite non-zero difference between A_1 and A_2.

So, now we only consider the case $A_1 = A_2$ and hence look at the equation $f(k) = cos(k\theta_1) + cos(k\theta_2)$.

Is $k(\theta_1, \theta_2)$ dense?

The reason we consider this question, is because then we can make both angles simultaneously close to 0 or 180 and hence make their cosines take values arbitrarily close to 1 and -1 respectively.

We can apply a similar logic to the proof mentioned above. Divide the circle into N sectors of size $\frac{360}{N}$ each. There are only $N^2 - N$ combinations of different sectors that they can lie in. After letting k run from 1 till $N^2 - N + 1$, one of the pairs of sectors will repeat, ie. $l\theta_1$ and $m\theta_1$ are in the same sector, and simultaneously $l\theta_2$ and $m\theta_2$ are in the same sector.

In other words, $0 < |(l-m)\theta_1| < \frac{360}{N}$ and $0 < |(l-m)\theta_2| < \frac{360}{N}$

This means that *whatever* θ_1 and θ_2 are, whichever sectors they may have start off in, they will surely be simultaneously together in the first sector left and right of zero.

Note that this is true independent of what N is chosen to be, and hence N can be made large enough, so that both $cos(l-m)\theta_1$ and $cos(l-m)\theta_2$ get arbitrarily close to 1 (noting that cos is an even function).

Hence we are sure that this function $f(k)$ definitely comes arbitrarily close to the value $A_1 + A_2$.

Let us look at something more interesting. Let us say that
$min\{cos(l-m)\theta_1, cos(l-m)\theta_2\} = \delta$.

Now, we can take $\frac{360}{N} < \delta$, and run our above algorithm. This will yield l', m' such that $min\{cos(l'-m')\theta_1, cos(l'-m')\theta_2\} < \delta$.

This shows that we can bring this function $f(k)$ arbitrarily close to the value $A_1 + A_2$ infinite number of times.

$k(\theta_1, \theta_2)$ is dense when θ_1/θ_2 is irrational

Assume that you want to show that $k(\theta_1, \theta_2)$ comes arbitrarily close to any (α, β), say by less than an amount, ϵ. Assume that all angles are being considered $mod360$. That is to say $|\alpha - \theta_1| < \epsilon$ and $|\beta - \theta_2| < \epsilon$

Consider a square S_1 of size 360. Plot the points (θ_1, θ_2) and (α, β). The points $k(\theta_1, \theta_2)$ will be along the line joining the origin to (θ_1, θ_2) and wrapping itself around, since we are taking $mod 360$. We want to show that these points do fall inside the circle of radius ϵ around (α, β).

Fact. If θ_1/θ_2 is irrational then the slope of the above line is irrational, and the line is dense in the square. Infact, if the line is viewed as a continued circle and the square is viewed as a torus, then this continued circle is dense in the torus. It keeps filling the square with infinite parallel lines of that slope.

Choose $s := \frac{360}{D}$ for integer D such that a square of that size can completely fit inside a circle of radius ϵ. Divide the square S_1 into subsquares of size s. Assume that (α, β) is in a special square $s*$.

Using the argument above, we can find r such that
$0 < r\theta_1 < \frac{360}{2N}$ and $0 < r\theta_2 < \frac{360}{2N}$

Using this point and its multiples only (which all lie along a line joining origin and $(r\theta_1, r\theta_2)$), we see that this line will definitely cross the square $s*$ infinitely often, and arbitrarily close to its left bottom corner as well. Given that the square is of size at least twice that of the jumps of $(r\theta_1, r\theta_2)$, it is clear that at least one of these multiples lies inside the square $s*$, say the pth multiple.

So we have $pr(\theta_1, \theta_2)$ as the required point which gets arbitrarily close to (α, β). Note that here we assumed the pair $(r\theta_1, r\theta_2)$ was actually in the bottom left square. If, instead, it was in the bottom right, top left, or top right square, we would consider using the lines from those appropriate corners - the same calculations regarding slope and density still hold. Hence proved.

When θ_1/θ_2 is rational

Let $\theta_1/\theta_2 := p$ where p is rational.
So, $\theta_1 = p\theta_2$.
We hence get $\frac{\theta_1+\theta_2}{\theta_1-\theta_2} = \frac{p+1}{p-1} = q$.
Also, note that $\phi_1 = \theta_1 - \theta_2$ is irrational, and $\phi_2 = \theta_1 + \theta_2$ is also irrational. We have $\phi_2 = q\phi_1$

We use the formula $cosA + cosB = 2cos(\frac{A+B}{2})cos(\frac{A-B}{2})$.
So $f(k) = 2cos(k\frac{\theta_1+\theta_2}{2})cos(k\frac{\theta_1-\theta_2}{2})$.
Or, we can say $f(k) = cos(k\phi)cos(kq\phi)$ where q is rational and ϕ is irrational.

Now, choose θ in the first quadrant such that $q\theta$ is in second quadrant. Since ϕ is irrational, multiples of it can be made arbitrarily close to θ. In that case $q\phi$ will be arbitrarily close to $q\theta$ as well. Since they are in the first and second quadrants, their cosines will be of different signs, and hence their product is negative.

We have already shown that $f(k)$ will come close the value $A_1 + A_2$ several times. Here that value is 2. Hence, $f(k)$ takes positive and negative signs.

6 References

[1] Tiwari A. : Termination of Linear Programs, In Alur R. and Peled D. A. (Editors), *Computer Aided Verification (CAV), LNCS 3114*, pages 70-82, 2004. Springer Verlag.

[2] Aditya G. Parameswaran's BTech Thesis, 2006.

[3] Braverman M. : Termination of Integer Linear Programs, In *Proc. of 18th International Conference on Computer Aided Verification (CAV 2006)*, 2006, Springer-Verlag.

[4] Bradley A. R., Manna Z., Sipma H. B. : Termination Analysis of Integer Linear Loops, In *Proc. of the 16th International Conference on Concurrency Theory (CONCUR'05), San Francisco*, 2005, Springer Verlag.