UNIVERSIDAD MIGUEL HERNÁNDEZ

CIENCIAS SOCIALES Y JURÍDICAS DE ELCHE

TRABAJO DE FIN DE GRADO

Distancia a la consistencia e inconsistencia de sistemas de desigualdades lineales: implemnetación en Matlab

Autor

Daniel Candela Murcia

Tutora

Maria Josefa Cánovas Cánovas

GRADO EN ESTADÍSTICA EMPRESARIAL

—

Curso 2020/2021

RESUMEN

El presente Trabajo de Fin de Grado versa de forma general sobre la teoría de sistemas de desigualdades lineales que rigen un problema de Programación Lineal.

Con el objetivo de entender el enfoque de este campo de la optimización, se contextualizan con breves notas históricas las investigaciones desarrolladas por los pioneros de la Programación Lineal.

Dentro del amplio campo de la Programación Lineal, el presente trabajo se ocupa de analizar las repercusiones que pueden ocasionar ligeras modificaciones en los coeficientes de los sistemas de desigualdades lineales y cómo pueden afectar a la solución del problema. Cabe destacar el hecho de que este trabajo se apoya, por un lado, en resultados clásicos de la Programación Lineal y, por otro lado, en resultados recientes de investigación (véanse las referencias bibliográficas comentadas en la introducción).

En este contexto, nuestro principal objetivo es conocer en qué cantidad hemos de modificar los coeficientes de los sistemas de desigualdades lineales de tal forma que consigamos transformar un sistema que no tiene solución en otro que sí la tiene, o bien saber en cuánto podemos modificar dichos coeficientes para poder seguir asegurando que tiene solución. Para comprender este análisis, comenzamos explicando algunos conceptos matemáticos básicos relacionados con los sistemas de desigualdades lineales. A continuación, incluimos varios teoremas imprescindibles acerca de la consistencia e inconsistencia de los sistemas. Por último, nos centramos en estudiar en que cantidad mínima hemos de modificar el sistema para llegar a dicha consistencia o inconsistencia, todo ello acompañado de sus respectivos ejemplos y gráficas.

En cuanto a nuestra contribución principal, es la elaboración e implementación en MATLAB de tres programas. El primero de ellos es capaz de calcular y mostrar genéricamente, mediante la introducción del usuario de los coeficientes y términos independientes del sistema de desigualdades lineales, la distancia a la inconsistencia (en norma euclídea). Respecto al segundo, también calcula la distancia a la inconsistencia, pero esta vez utiliza la norma del supremo. El tercero utiliza la norma del supremo y es capaz de calcular y mostrar genéricamente, mediante la introducción del usuario de los coeficientes y términos independientes del sistema de desigualdades lineales, la distancia a la consistencia o inconsistencia según el caso que nos encontremos. Seguidamente exponemos las conclusiones generales del proyecto. Además, mediante el empleo de los programas desarrollados en MATLAB, se exponen diferentes ejemplos que se pueden dar según el sistema de desigualdades lineales en el que nos encontremos.

Índice General

Capítulo 1

Introducción

El presente Trabajo de Fin de Grado se centra en la teoría de sistemas de desigualdades lineales que rigen un problema de Programación Lineal (PL, por brevedad). La PL se enmarca en el campo de la programación matemática y su objetivo es maximizar o minimizar una función lineal (conocida como función objetivo), la cual está sujeta a una serie de restricciones expresadas mediante un sistema de ecuaciones, inecuaciones o ambas, también lineales (véase Goberna [3] para más información). En cuanto al origen de la PL se puede situar a mediados del siglo XX ya que en la Segunda Guerra Mundial la emplearon con el fin de reducir los costes del ejército aliado y aumentar las pérdidas del enemigo.

Dentro del amplio campo de la PL, el presente trabajo versa sobre la teoría de sistemas de desigualdades lineales asociadas a problemas de Programación Lineal, y en concreto se ocupa de analizar las repercusiones que pueden ocasionar ligeras modificaciones en dichos coeficientes. Podemos encontrar mucha información al respecto gracias a numerosas monografías como Bazaraa *et al.* [1], Goberna y López [4], Hettich y Kortanek [5], Bertsimas y Tsitsiklis [2], Sierksma y Zwols [7].

Para concretar, nuestro objetivo principal es conocer en qué cantidad podemos modificar los coeficientes del sistema de tal forma que consigamos transformar un sistema que no tiene solución en otro que sí la tenga, o bien saber cuánto podemos modificar dichas restricciones para seguir asegurando que tiene solución.

Para poder entender el tema, el trabajo se desarrolla en dos capítulos:

- En el primer capítulo, comenzamos explicando algunos conceptos matemáticos básicos relacionados con los sistemas de desigualdades lineales y su clasificación junto con los ejemplos y gráficas correspondientes. También incluimos varios teoremas imprescindibles acerca de la consistencia e inconsistencia de los sistemas, que nos van a permitir seguir desarrollando el trabajo.

- En cuanto al segundo capítulo, nos centramos en estudiar en qué cantidad mínima hemos de modificar el sistema para llegar a la consistencia o inconsistencia. Para ello, hemos decidido dividir el capítulo en tres secciones. En la primera sección, empezamos describiendo conceptos relacionados con la Teoría de Conjuntos y seguimos con cálculo de distancias (todo ello con ejemplos y gráficas) y algunos conceptos relacionados con la convexidad, que nos permite exponer un teorema fundamental en este trabajo. En cuanto a la segunda sección, se trata de la implementación totalmente generalizada y sistemática en MATLAB de las distancias a la consistencia e inconsistencia, según el caso en el que nos encontremos. Para ello, hemos subdividido la sección en dos partes (una que utiliza la norma euclídea y otra con la norma del supremo). Por último, en la tercera sección se exponen ejemplos en los cuales no se conoce a priori de qué tipo de sistema se trata (consistente o inconsistente) y utilizando los programas desarrollados en MATLAB se calcula la distancia a la consistencia o bien a la inconsistencia, según corresponda.

Capítulo 2

Teoría de sistemas de desigualdades lineales

2.1 Introducción

En esta sección se definen algunos elementos básicos de los sistemas de desigualdades lineales necesarios para desarrollar con rigor el tema a tratar.

Definición 2.1.1 *Semiespacios cerrados*

Un *semiespacio cerrado* es un conjunto de la forma S $= \{x \in \mathbb{R}^n \mid a'x \geq b\}$ con $a \neq 0_n$ (0_n denota el vector nulo en $\mathbb{R}^n$) , $a \in \mathbb{R}^n$, $b \in \mathbb{R}$.

Ejemplo 2.1.1 *Consideremos el conjunto* $S = \{x \in \mathbb{R}^2 \mid x_1 + 2x_2 \geq 4\}$.

En la siguiente figura vemos representado el semiespacio cerrado descrito por la desigualdad $x_1 + 2x_2 \geq 4$.

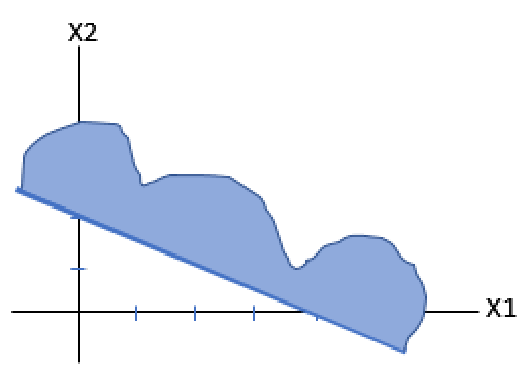


Figura 1: Elaboración propia

Definición 2.1.2 *Poliedro*

Se dice que $F \in \mathbb{R}^n$ es un *poliedro* si coincide con el conjunto de soluciones de un sistema finito de desigualdades lineales.

Formalmente: $F = \{x \in \mathbb{R}^n \mid a_i' x \geq b_i \ , \ con\ i = 1, ..., m\}$ donde $a_i \in \mathbb{R}^n$, $b_i \in \mathbb{R}$ para $i = 1, ..., m$.

Cuando un poliedro es acotado, se le conoce como *polítopo*.

Ejemplo 2.1.2 *:Consideremos el conjunto* $F = \{x \in \mathbb{R}^2 \mid x_1 + x_2 \geq 4, \quad x_1 - x_2 \geq 0\}$.

En la figura apreciamos a simple vista como se trata de un poliedro no acotado

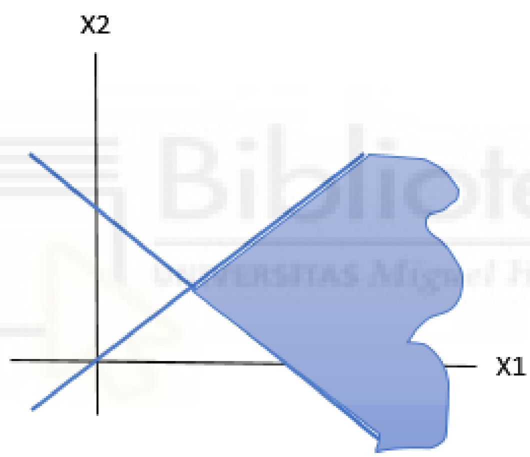


Figura 2: Elaboración propia

Ejemplo 2.1.3 *: Consideremos el conjunto* $F = \left\{x \in \mathbb{R}^2 \mid \frac{1}{2}x_1 + x_2 \leq 2, \quad x_1, x_2 \geq 0\right\}$.

En cambio, el poliedro representado en la siguiente figura es claramente acotado, por lo que se trata de un polítopo.

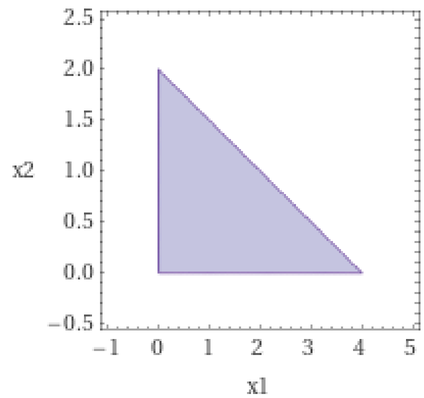


Figura 3: Elaboración mediante WolframAlpha

Definición 2.1.3 *Sistemas y conjuntos factibles*

A partir de ahora denotaremos por σ el *sistema de desigualdades lineales*, esto es, $\sigma := \{a_i'x \geq b_i \text{ , } con\ i = 1, ..., m\}$ y por F a su *conjunto factible*, esto es, $F := \{x \in \mathbb{R}^n \mid a_i'x \geq b_i \text{ , } con\ i = 1, ..., m\}$.

Definición 2.1.4 *Sistemas consistentes e inconsistentes*

Se dice que $\sigma := \{a_i'x \geq b_i \text{ , } con\ i = 1, ..., m\}$ es *consistente* si su conjunto factible F es no vacío.

Ejemplo 2.1.4 *Consideremos el sistema en* $\mathbb{R}^2$,

$$\sigma_1 = \{x_1 + x_2 \geq 1,\ x_1 - x_2 \geq 0,\ -2x_1 \geq 4\}.$$

En la siguiente imagen podemos apreciar visualmente que el sistema representado es inconsistente ya que su conjunto factible F es vacio.

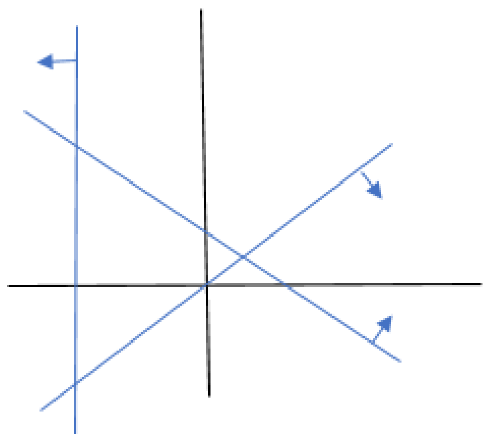


Figura 4: Elaboración propia

Ejemplo 2.1.5 *Consideremos el sistema en* $\mathbb{R}^2$,
$\sigma_2 = \{x_1 + x_2 \geq 1, \quad x_1 - x_2 \geq 0, \quad x_1 + 2x_2 \geq 4\}$.

Por el contrario, tal y como vemos en la siguiente imagen el conjunto factible F de este sistema es no vacío, por lo que se trata de un sistema consistente.

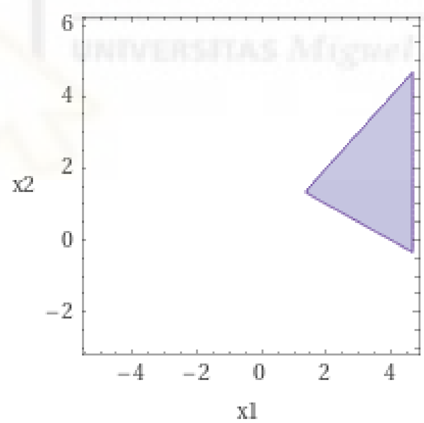


Figura 5: Elaboración mediante WolframAlpha

2.2 Lema de Farkas y caracterización de la consistencia

Antes de empezar con esta sección, es necesario hablar sobre lo que se conoce como los *conos asociados a un sistema de desigualdades lineales* y sobre el término *consecuencia*. En lo que sigue, dado un subconjunto $X \subset \mathbb{R}^n$, denotamos por

$cone\{X\}$ al llamado cono convexo generado por X, que está dado por: $coneX = \left\{\sum_{i=1}^{s} \lambda_i x_i \text{ con } \lambda i \geq 0,\ x_i \in X,\ i = 1, ..., s,\ s \in \mathbb{N}\right\}$.

Dado el sistema σ, definimos:

- El *primer cono de momentos de σ:*

$$M := cone\{a_i \text{ con } i = 1, ..., m\} = \left\{\sum_{i=1}^{m} \lambda_i a_i \text{ con } \lambda i \geq 0,\ i = 1, ..., m\right\}$$

- El *segundo cono de momentos de* σ:

$$\begin{aligned} N &:= cone\left\{\begin{pmatrix} a_i \\ b_i \end{pmatrix} \text{ con } i = 1, ..., m\right\} \\ &= \left\{\sum_{i=1}^{m} \lambda_i \begin{pmatrix} a_i \\ b_i \end{pmatrix} \text{ con } \lambda i \geq 0,\ i = 1, ..., m\right\} \end{aligned}$$

- El *cono característico de* σ:

$$K := cone\left\{\begin{pmatrix} a_i \\ b_i \end{pmatrix} \text{ con } i = 1, ..., m\ ;\ \begin{pmatrix} 0_n \\ -1 \end{pmatrix}\right\}$$

- Se dice que una desigualdad "$a'x \geq b$" es consecuencia de σ si para todo $\overline{x} \in F$ se tiene que $a'\overline{x} \geq b$.

A continuación, enunciamos el Lema de Farkas, uno de los pilares de la Programación Lineal. Farkas Gyula fue un matemático y físico húngaro cuya vida transcurrió entre 1847 y 1930. Estudió derecho y física y fue profesor de física, primero en un instituto y más tarde en la universidad, donde también fue galardonado como profesor asociado (el mayor título profesional conferido por algunas universidades europeas) de matemáticas.

Teorema 2.2.1 (Lema de Farkas (1901)) *Se tiene que una desigualdad "$a'x \geq b$" es consecuencia del sistema σ si y solo si* $\begin{pmatrix} a \\ b \end{pmatrix} \in K$.

Ejemplo 2.2.1 *Siendo el sistema en* $\mathbb{R}^2$, $\sigma = \{2x_1 + x_2 \geq 2,\ \ x_1 - x_2 \geq 0\}$ *¿Es la desigualdad "*$7x_1 + 2x_2 \geq 4$*" consecuencia de* σ*?*

Por el Lema de Farkas, se pregunta si

$$\begin{pmatrix} 7 \\ 2 \\ 4 \end{pmatrix} \in K = cone \left\{ \begin{pmatrix} 2 \\ 1 \\ 2 \end{pmatrix}, \begin{pmatrix} 1 \\ -1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ -1 \end{pmatrix} \right\},$$

es decir, si podemos escribir

$$\begin{pmatrix} 7 \\ 2 \\ 4 \end{pmatrix} = \lambda_1 \begin{pmatrix} 2 \\ 1 \\ 2 \end{pmatrix} + \lambda_2 \begin{pmatrix} 1 \\ -1 \\ 0 \end{pmatrix} + \lambda_3 \begin{pmatrix} 0 \\ 0 \\ -1 \end{pmatrix},\ para\ ciertos\ \lambda_1, \lambda_2, \lambda_3 \geq 0.$$

Si convertimos la ecuación matricial en su forma lineal obtenemos el siguiente sistema de igualdades:

$$\begin{aligned} 7 &= 2\lambda_1 + \lambda_2, \\ 2 &= \lambda_1 - \lambda_2, \\ 4 &= 2\lambda_1 - \lambda_3, \end{aligned}$$

que al resolverlo obtenemos los siguientes valores para nuestros λ_i:

$$\begin{aligned} \lambda_1 &= 3, \\ \lambda_2 &= 1, \\ \lambda_3 &= 2, \end{aligned}$$

por lo que al ser todos los $\lambda_i \geq 0$, la desigualdad "$7x_1 + 2x_2 \geq 4$" sí es consecuencia de σ.

Ejemplo 2.2.2 *: Siendo el sistema en* $\mathbb{R}^2$, $\sigma = \left\{\frac{1}{2}x_1 + 2x_2 \geq 3,\ \ 4x_1 - 5x_2 \geq 1\right\}$. *¿Es la desigualdad "*$6x_1 + \frac{1}{3}x_2 \geq 11$*" consecuencia de* σ*?*

Por el Lema de Farkas, se pregunta si

$$\begin{pmatrix} 6 \\ \frac{1}{3} \\ 11 \end{pmatrix} \in K = cone \left\{ \begin{pmatrix} \frac{1}{2} \\ 2 \\ 3 \end{pmatrix}, \begin{pmatrix} 4 \\ -5 \\ 1 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ -1 \end{pmatrix} \right\},$$

es decir, si podemos escribir

$$\begin{pmatrix} 6 \\ \frac{1}{3} \\ 11 \end{pmatrix} = \lambda_1 \begin{pmatrix} \frac{1}{2} \\ 2 \\ 3 \end{pmatrix} + \lambda_2 \begin{pmatrix} 4 \\ -5 \\ 1 \end{pmatrix} + \lambda_3 \begin{pmatrix} 0 \\ 0 \\ -1 \end{pmatrix}, \text{ para ciertos } \lambda_1, \lambda_2, \lambda_3 \geq 0.$$

Si convertimos la ecuación matricial en su forma lineal obtenemos el siguiente sistema de igualdades

$$\begin{aligned} 6 &= \tfrac{1}{2}\lambda_1 + 4\lambda_2, \\ \tfrac{1}{3} &= 2\lambda_1 - 5\lambda_2, \\ 11 &= 3\lambda_1 + \lambda_2 - \lambda_3, \end{aligned}$$

que al resolverlo obtenemos los siguientes valores para nuestros λ_i:

$$\begin{aligned} \lambda_1 &= \tfrac{188}{63}, \\ \lambda_2 &= \tfrac{71}{63}, \\ \lambda_3 &= -\tfrac{58}{63}, \end{aligned}$$

por lo que al no ser todos los $\lambda_i \geq 0$, la desigualdad "$6x_1 + \frac{1}{3}x_2 \geq 11$" no es consecuencia de σ.

Teorema 2.2.2 (Caracterización de la consistencia) *Dado el sistema σ, son equivalentes:*

(i) σ es consistente;

(ii) $\begin{pmatrix} 0_n \\ 1 \end{pmatrix} \notin K$;

(iii) $\begin{pmatrix} 0_n \\ 1 \end{pmatrix} \notin N$.

Ejemplo 2.2.3 *Siendo el sistema en $\mathbb{R}^2$, $\sigma = \{x_1 + x_2 \leq 1,\ x_1 - x_2 \leq 0,\ 2x_1 \geq 4\}$, se desea comprobar su consistencia o inconsistencia.*

El primer paso es convertir las ecuaciones al formato de $\geq$ por lo que el sistema quedaría tal que así: $\sigma = \{-x_1 - x_2 \geq -1, \ \ -x_1 + x_2 \geq 0,\ 2x_1 \geq 4\}$.

Queremos saber si

$$\begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \in N = cone \left\{ \begin{pmatrix} -1 \\ -1 \\ -1 \end{pmatrix}, \begin{pmatrix} -1 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 2 \\ 0 \\ 4 \end{pmatrix} \right\},$$

es decir, si podemos escribir

$$\begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} = \lambda_1 \begin{pmatrix} -1 \\ -1 \\ -1 \end{pmatrix} + \lambda_2 \begin{pmatrix} -1 \\ 1 \\ 0 \end{pmatrix} + \lambda_3 \begin{pmatrix} 2 \\ 0 \\ 4 \end{pmatrix}, \text{ para ciertos } \lambda_1, \lambda_2, \lambda_3 \geq 0.$$

Si convertimos la ecuación matricial en su forma lineal obtenemos el siguiente sistema de igualdades

$$\begin{gathered} 0 = -\lambda_1 \ - \lambda_2 + 2\lambda_3, \\ 0 = -\lambda_1 \ + \lambda_2, \\ 1 = -\lambda_1 \ + 4 \ \lambda_3, \end{gathered}$$

que al resolverlo obtenemos los siguientes valores para nuestros λ_i:

$$\begin{gathered} \lambda_1 = \tfrac{1}{3}, \\ \lambda_2 = \tfrac{1}{3}, \\ \lambda_3 = \tfrac{1}{3}, \end{gathered}$$

por lo que al ser todos los $\lambda_i \geq 0$, hemos probado que el vector columna $\begin{pmatrix} 0_2 \\ 1 \end{pmatrix} \in N$ y por lo tanto el sistema σ es inconsistente.

Ejemplo 2.2.4 *Siendo el sistema en* $\mathbb{R}^2$,

$$\sigma = \left\{ 2x_1 - \frac{1}{2}x_2 \geq 1, \ \frac{1}{3}x_1 + 3x_2 \geq 3, \ 2x_1 + x_2 \geq 0 \right\}$$

se pretende comprobar su consistencia o inconsistencia.

Queremos saber si

$$\begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \in N = cone \left\{ \begin{pmatrix} 2 \\ -\frac{1}{2} \\ 1 \end{pmatrix}, \begin{pmatrix} \frac{1}{3} \\ 3 \\ 3 \end{pmatrix}, \begin{pmatrix} 2 \\ 1 \\ 0 \end{pmatrix} \right\},$$

es decir, si podemos escribir

$$\begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} = \lambda_1 \begin{pmatrix} 2 \\ -\frac{1}{2} \\ 1 \end{pmatrix} + \lambda_2 \begin{pmatrix} \frac{1}{3} \\ 3 \\ 3 \end{pmatrix} + \lambda_3 \begin{pmatrix} 2 \\ 1 \\ 0 \end{pmatrix}, \text{ para ciertos } \lambda_1, \lambda_2, \lambda_3 \geq 0.$$

Si convertimos la ecuación matricial en su forma lineal obtenemos el siguiente sistema de igualdades

$$\begin{gathered} 0 = 2\lambda_1 + \tfrac{1}{3}\lambda_2 + 2\lambda_3, \\ 0 = -\tfrac{1}{2}\lambda_1 + 3\lambda_2 + \lambda_3, \\ 1 = \lambda_1 + 3\ \lambda_2, \end{gathered}$$

que al resolverlo obtenemos los siguientes valores para nuestros λ_i:

$$\begin{gathered} \lambda_1 = \tfrac{17}{44}, \\ \lambda_2 = \tfrac{9}{44}, \\ \lambda_3 = -\tfrac{37}{88}, \end{gathered}$$

por lo que al no ser todos los $\lambda_i \geq 0$, hemos probado que el vector columna $\begin{pmatrix} 0_n \\ 1 \end{pmatrix} \notin N$ y por lo tanto el sistema σ es consistente.

Nota: Por simplicidad, los ejemplos ilustrados consideran sistemas de ecuaciones compatibles determinados (con solución única). Sin embargo, en general, estos sistemas podrán no tener solución o tener infinitas soluciones. En el último caso la consistencia se caracteriza por el hecho de que todas las soluciones tengan algún λ_i negativo.

Capítulo 3

Distancia a la consistencia/inconsistencia

3.1 Definiciones y resultados fundamentales

Una vez sabemos identificar si un sistema es consistente o inconsistente, el objetivo de esta sección es estudiar en que cantidad mínima hemos de modificar σ para llegar a la consistencia/inconsistencia respectivamente.

Notación: llamamos Θ al conjunto de sistemas de desigualdades lineales de la forma $\sigma = \{a_i'x \geq b_i \text{ , } con \ i = 1, ..., m\}$. Formalmente, consideramos $\Theta = (\mathbb{R}^{n+1})^m$, que podemos identificar con el conjunto de matrices de coeficientes reales de tamaño $m * (n+1)$, el cual denotaremos por $M_{m \, * \, (n+1)}$.

Ejemplo 3.1.1 *Consideremos* $\sigma = \{x_1 + x_2 \geq 1, \ \ x_1 - x_2 \geq 0, \ \ x_1 \leq 0\}$ *que se encuentra en el espacio* $\Theta = (\mathbb{R}^3)^3 = \mathbb{R}^9$*, esto es, estamos considerando m = 3 restricciones y n = 2 variables.*

Los vectores de coeficientes del sistema son:

$$\begin{pmatrix} a_1 \\ b_1 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}, \begin{pmatrix} a_2 \\ b_2 \end{pmatrix} = \begin{pmatrix} 1 \\ -1 \\ 0 \end{pmatrix}, \begin{pmatrix} a_3 \\ b_3 \end{pmatrix} = \begin{pmatrix} -1 \\ 0 \\ 0 \end{pmatrix}.$$

Alternativamente podemos identificar el sistema σ con la matriz

$$\sigma \equiv \begin{pmatrix} 1 & 1 & 1 \\ 1 & -1 & 0 \\ -1 & 0 & 0 \end{pmatrix} \in M_{3x3}.$$

Representamos las restricciones del sistema en la siguiente figura:

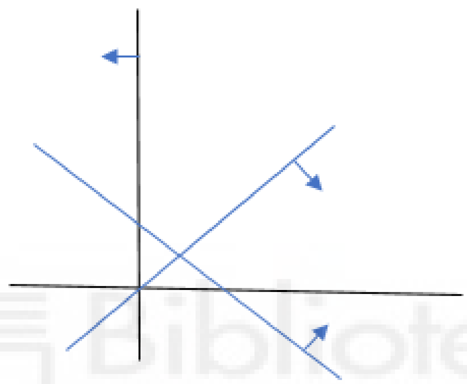


Figura 6: Elaboración propia

Vemos intuitivamente que el sistema es inconsistente (aunque también podríamos comprobarlo mediante el teorema 2.2.2). Uno de los objetivos de esta sección es responder a la siguiente pregunta ¿en que cantidad mínima hemos de modificar σ para llegar a la consistencia? Con este fin introducimos a continuación los conceptos y herramientas necesarias.

El conjunto de sistemas de desigualdades lineales Θ está formado por la unión de 2 subconjuntos, Θ_c que denota al subconjunto de sistemas consistentes y Θ_i que denota al subconjunto de sistemas inconsistentes. De este modo se tiene que : $\Theta = \Theta_c \cup \Theta_i$. De hecho, es una partición de Θ, es decir, $\Theta_c \cap \Theta_i = \emptyset$.

Seguidamente, se ilustra la situación de estos conjuntos y un sistema $\sigma \in \Theta_i$. Además, se pretende ilustrar la perturbación que se debe efectuar sobre σ para alcanzar la consistencia, formalmente para llegar a la frontera de Θ_i. Tras la figura recordamos los conceptos necesarios.

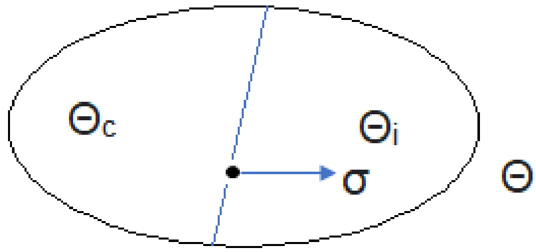


Figura 7: Elaboración propia

En la gráfica anterior la línea que divide los 2 subconjuntos se conoce como *frontera* o *boundary* en inglés (a partir de ahora '*bd*') . En rigor, que un elemento se encuentre en la frontera de un conjunto $X \subset \mathbb{R}^k$, con $k \in \mathbb{N}$, significa que:

$$x \in bd(X) \;:\Leftrightarrow\; \exists\ \{x^r\} \subset X \text{ tal que } \lim\ x^r = x\ ;\ \exists\ \{y^r\} \subset\ \mathbb{R}^k \diagdown X \text{ tal que } \lim\ y^r = x. \tag{3.1}$$

Observación: Se tiene que $bd(X) = bd(\ \mathbb{R}^k \diagdown X\ )$. En nuestro caso, se tiene que $bd(\Theta_c) = bd(\Theta_i)$ dado que $\Theta_i = \Theta \diagdown \Theta_c$.

Ahora nuestro propósito es, dado un sistema $\sigma \in \Theta$, hallar la siguiente distancia:

$$d(\sigma, bd(\Theta_c)) = d(\sigma, bd(\Theta_i)).$$

Para una mejor interpretación, se distinguen dos casos:

- Por un lado, si el sistema $\sigma \in \Theta_c$, se tiene que
$$d(\sigma, bd(\Theta_c)) = d(\sigma, \Theta_i) = \inf \{d(\sigma, \sigma_1) \text{ tal que } \sigma_1 \in \Theta_i\}.$$
- Por otro lado, si el sistema $\sigma \in \Theta_i$, se tiene que
$$d(\sigma, bd(\Theta_c)) = d(\sigma, \Theta_c) = \inf \{d(\sigma, \sigma_1) \text{ tal que } \sigma_1 \in \Theta_c\}.$$

Para poder utilizar dicha fórmula, debemos mencionar que la distancia entre dos sistemas queda definida de la siguiente manera : dados $\sigma = \{a_i' x \geq b_i\ ,\ con\ i = 1, ..., m\}$ y $\sigma_1 = \left\{(a_i^1)' x \geq b_i^1\ ,\ con\ i = 1, ..., m\right\}$

$$d(\sigma, \sigma_1) = \max_{i=1,\dots,m} \left\{ \left\| \begin{pmatrix} a_i \\ b_i \end{pmatrix} - \begin{pmatrix} a_i^1 \\ b_i^1 \end{pmatrix} \right\| \right\}, \tag{3.2}$$

donde $\|\cdot\|$ representa una norma en $\mathbb{R}^{n+1}$.

Ejemplo 3.1.2 *Considerando* $\sigma = \{x_1 + x_2 \geq 1,\ x_1 - x_2 \geq 0,\ x_1 \leq 0\}$ *y* $\sigma_1 = \{x_1 + 2x_2 \geq 3,\ 2x_1 + x_2 \geq 1,\ x_1 + x_2 \geq 1\}$, *vamos a calcular la distancia* $d(\sigma, \sigma_1)$ *utilizando la norma euclídea dada en 3.2*

Si cogemos los coeficientes de los sistemas y los representamos matricialmente (teniendo en cuenta que todas las desigualdades deben estar en formato de $\geq$) quedaría tal que así:

$$\sigma = \begin{pmatrix} 1 & 1 & 1 \\ 1 & -1 & 0 \\ -1 & 0 & 0 \end{pmatrix}, \quad \sigma_1 = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

Si hacemos las diferencias $\binom{a_i}{b_i} - \binom{a_i^1}{b_i^1}$ para $i = 1, 2, 3$, respectivamente, obtenemos:

$$\begin{pmatrix} 1-1 \\ 1-2 \\ 1-3 \end{pmatrix} = \begin{pmatrix} 0 \\ -1 \\ -2 \end{pmatrix}, \begin{pmatrix} 1-2 \\ -1-1 \\ 0-1 \end{pmatrix} = \begin{pmatrix} -1 \\ -2 \\ -1 \end{pmatrix}, \begin{pmatrix} -1-1 \\ 0-1 \\ 0-1 \end{pmatrix} = \begin{pmatrix} -2 \\ -1 \\ -1 \end{pmatrix}.$$

Aplicando la fórmula (3.2) tenemos que

$$\begin{aligned} d(\sigma, \sigma_1) &= \max \left\{ \left\| \begin{pmatrix} 0 \\ -1 \\ -2 \end{pmatrix} \right\|_2, \quad \left\| \begin{pmatrix} -1 \\ -2 \\ -1 \end{pmatrix} \right\|_2, \quad \left\| \begin{pmatrix} -2 \\ -1 \\ -1 \end{pmatrix} \right\|_2 \right\} \\ &= \max \left\{ \sqrt{5},\ \sqrt{6},\ \sqrt{6} \right\} = \sqrt{6} \end{aligned}$$

Por lo tanto, la distancia entre ambos sistemas es de $\sqrt{6}$ unidades.

Ejemplo 3.1.3 *Dado el sistema* $\sigma = \{0x_1 + 0x_2 \geq 0,\ x_1 + x_2 \geq 1\}$, *¿se encuentra* σ *en la frontera de* Θ_c *? Es decir, ¿*$\sigma \in bd(\Theta_c) = bd(\Theta_i)$*?*

Nótese que el sistema es consistente puesto que el punto $(\frac{1}{2}, \frac{1}{2})$ es factible.

Para utilizar la definición dada en (3.1), con el fin de probar que $\sigma \in bd(\Theta_c)$ podemos por un lado considerar la sucesión constante dada por $\sigma_r = \sigma$, $\forall r$ y por otro lado, hemos de encontrar otra sucesión de sistemas inconsistentes que converja hacia el sistema consistente, es decir, $\{\widehat{\sigma}_r\} \subset \Theta_i$ con $\widehat{\sigma}_r \longrightarrow \sigma$.

Para ello, planteamos el siguiente sistema: $\widehat{\sigma}_r = \left\{0x_1 + 0x_2 \geq \frac{1}{r},\ x_1 + x_2 \geq 1\right\}$.
Queremos probar si $\lim \widehat{\sigma}_r = \sigma$, lo que significa que $\lim_{r \longrightarrow \infty} d(\widehat{\sigma}_r, \sigma) = 0$.
Debemos calcular en primer lugar la distancia entre ambos sistemas, es decir:

$$d(\widehat{\sigma}_r, \sigma) = \max\left\{\left\|\begin{pmatrix} 0 \\ 0 \\ \frac{1}{r} \end{pmatrix}\right\|_2, \left\|\begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}\right\|_2\right\} =$$
$$\max\left\{\sqrt[2]{0^2 + 0^2 + \left(\frac{1}{r}\right)^2}, 0\right\} = \max\left\{\frac{1}{r}, 0\right\} = \frac{1}{r}$$

Por lo tanto, $\lim_{r \longrightarrow \infty} d(\widehat{\sigma}_r, \sigma) = \lim_{r \longrightarrow \infty} \left(\frac{1}{r}\right) = 0$.
Como conclusión, podemos decir que el sistema σ se encuentra en la frontera porque es consistente y hemos probado que existe una sucesión de sistemas inconsistentes que converge hacia él.

Ejemplo 3.1.4 *Dado el sistema* $\sigma = \{x_1 + x_2 \geq 1,\ -x_1 - x_2 \geq 1\}$, *¿se encuentra* σ *en la frontera de* Θ_c *? Es decir, ¿*$\sigma \in bd(\Theta_c) = bd(\Theta_i)$*?*

Si representamos el sistema gráficamente

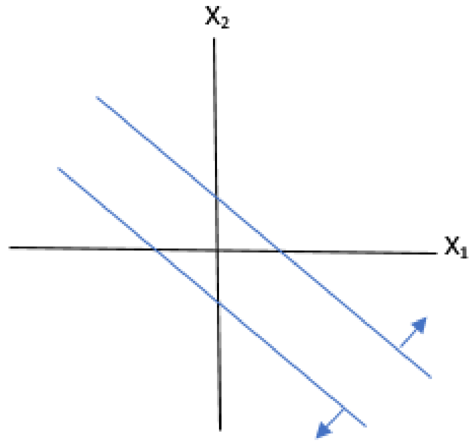


Figura 8: Elaboración propia

podemos observar que se trata de un sistema inconsistente, es decir, $\sigma \in \Theta_i$.

Para utilizar la definición dada en (3.1), con el fin de probar que $\sigma \in bd(\Theta_i)$ podemos por un lado considerar la sucesión constante dada por $\sigma_r = \sigma$, $\forall r$ y por otro lado, hemos de encontrar otra sucesión de sistemas consistentes que converja hacia el sistema inconsistente, es decir, $\{\widehat{\sigma}_r\} \subset \Theta_c$ con $\widehat{\sigma}_r \longrightarrow \sigma$.

Para ello, planteamos el siguiente sistema:

$$\widehat{\sigma}_r = \left\{ x_1 + \left(1 - \frac{1}{r}\right) x_2 \geq 1, \quad \left(-1 + \frac{1}{r}\right) x_1 - x_2 \geq 1 \right\}.$$

Queremos probar que σ_r es consistente para todo r y que $\lim \widehat{\sigma}_r = \sigma$, lo que significa que $\lim_{r \longrightarrow \infty} d(\widehat{\sigma}_r, \sigma) = 0$.

Debemos calcular en primer lugar la distancia entre ambos sistemas, es decir:

$$d(\widehat{\sigma}_r, \sigma) = \max \left\{ \left\| \begin{pmatrix} 0 \\ \frac{1}{r} \\ 0 \end{pmatrix} \right\|_2, \quad \left\| \begin{pmatrix} \frac{1}{r} \\ 0 \\ 0 \end{pmatrix} \right\|_2 \right\} =$$

$$\max \left\{ \sqrt[2]{0^2 + \left(\frac{1}{r}\right)^2 + 0^2}, \ \sqrt[2]{\left(\frac{1}{r}\right)^2 + 0^2 + 0^2} \right\} = \max \left\{ \frac{1}{r}, \ \frac{1}{r} \right\} = \frac{1}{r}$$

Por lo tanto, $\lim_{r \longrightarrow \infty} d(\widehat{\sigma}_r, \sigma) = \lim_{r \longrightarrow \infty} \left(\frac{1}{r}\right) = 0$.

Como conclusión, podemos decir que el sistema σ se encuentra en la frontera porque es inconsistente y hemos probado que existe una sucesión de sistemas consistentes que converge hacia él.

Siguiendo con la notación, un sistema $\sigma \in bd\,(\Theta_c) = bd\,(\Theta_i)$ se suele denominar '*mal planteado*' con respecto a la consistencia, o en inglés '*ill-possed*'. Así pues, dado $\sigma \in \Theta$, $d(\sigma, bd\,(\Theta_c))$ se conoce como distancia al mal planteamiento.

Antes de introducir el siguiente teorema, recordamos la definición de envoltura convexa,expresada en términos de combinaciones convexas.

- Sean $x^1, ..., x^r \subset \mathbb{R}^n$. Se dice que $x \in \mathbb{R}^n$ es una *combinación convexa* de $x^1, ..., x^r$ si existen escalares $\alpha_1, ..., \alpha_r$ tales que

$$x = \sum_{i=1}^{r} \alpha_i x^i, \text{ con } \sum_{i=1}^{r} \alpha_i = 1 \text{ y } \alpha_1, ..., \alpha_r \geq 0.$$

- Dado $X \in \mathbb{R}^n$, llamamos *envoltura convexa de X*, denotada por *convX*, al conjunto de todas las combinaciones convexas finitas de elementos de X, es decir,

$$convX = \left\{ \sum_{i=1}^{r} \alpha_i x^i \mid x^i \in X,\ \alpha_i \geq 0,\ i = 1, ..., r,\ \sum_{i=1}^{r} \alpha_i = 1,\ r \in \mathbb{N} \right\}.$$

Teorema 3.1.1 *Dado un sistema $\sigma \in \Theta$, se tiene que:*

$$d(\sigma, bd\,(\Theta_c)) = d(0_{n+1}, bd\,(H)),$$

donde

$$\begin{aligned} H &= conv\left\{ \begin{pmatrix} a_i \\ b_i \end{pmatrix} \text{ con } i = 1, ..., m \right\} + \mathbb{R}_+ \begin{pmatrix} 0_n \\ -1 \end{pmatrix} = \\ & \left\{ \sum_{i=1}^{m} \lambda_i \begin{pmatrix} a_i \\ b_i \end{pmatrix} + \mu \begin{pmatrix} 0_n \\ -1 \end{pmatrix}, \text{ con } \lambda_i \geq 0,\ i = 1, ..., m, \quad \sum \lambda_i = 1,\ \mu \geq 0 \right\} \end{aligned}$$

siendo $\mathbb{R}_+$ el conjunto de números reales no negativos.

Proposición 3.1.1 *Dado un sistema $\sigma \in \Theta$, se tiene que:*

(i) *Si $0_{n+1} \notin H$, entonces $\sigma \in \Theta_c$; en consecuencia $d(\sigma, bd\,(\Theta_c)) = d(\sigma, \Theta_i) = d(0_{n+1}, H) > 0$.*

(ii) *Si $0_{n+1} \in int(H)$, entonces $\sigma \in \Theta_i$; en consecuencia $d(\sigma, bd\,(\Theta_c)) = d(\sigma, \Theta_c) = d(0_{n+1}, bd(H)) > 0$.*

(iii) *Si $0_{n+1} \in bd(H)$ el sistema puede ser consistente o inconsistente, y $d(\sigma, bd\,(\Theta_c)) = d(\sigma, bd\,(\Theta_i)) = 0$.*

Ejemplo 3.1.5 *Dado el sistema $\sigma = \{x_1 \geq 1, \ -x_1 \geq -2\}$, se pretende calcular la distancia del sistema a la frontera de la consistencia.*

Mediante el teorema 3.1.1 sabemos que $d(\sigma, bd\,(\Theta_c)) = d(0_{n+1}, bd\,(H))$, o en nuestro caso tenemos que $d(\sigma, bd\,(\Theta_c)) = d(0_2, bd\,(H))$, donde

$$H = conv\left\{ \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \ \begin{pmatrix} -1 \\ -2 \end{pmatrix} \right\} + \mathbb{R}_+ \begin{pmatrix} 0 \\ -1 \end{pmatrix}.$$

A continuación, representamos gráficamente el problema para una mejor interpretación.

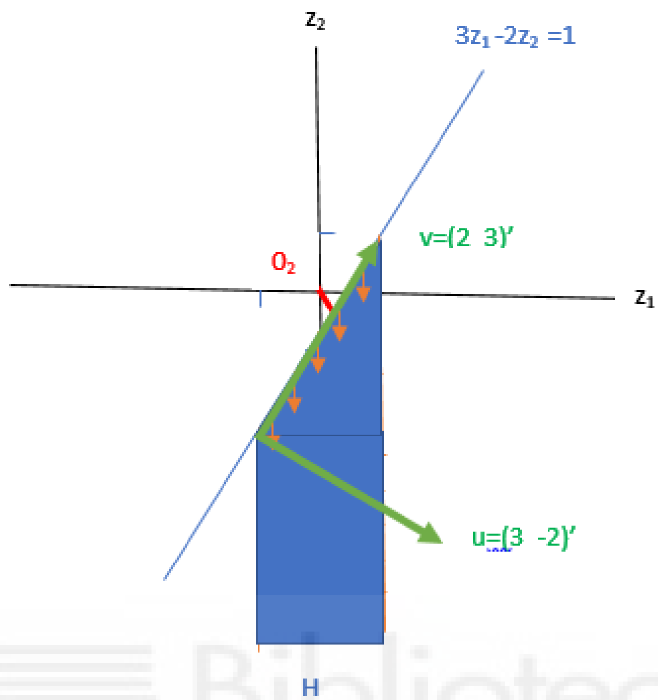


Figura 9: Elaboración propia

En la representación, podemos observar en primer lugar que la distancia que buscamos (la distancia a la frontera de la consistencia) es la que aparece representada en rojo, que en este caso corresponde con la distancia de un punto a una recta.

También podemos apreciar en color verde el vector v, el cuál se obtiene de la diferencia entre las matrices de las inecuaciones del sistema. Así pues, sería:

$$v = \begin{pmatrix} 1 \\ 1 \end{pmatrix} - \begin{pmatrix} -1 \\ -2 \end{pmatrix} = \begin{pmatrix} 2 \\ 3 \end{pmatrix}.$$

Respecto al vector u representado también en verde, se trata de un vector perpendicular al vector v por lo que basta con permutar las filas y cambiar una de ellas de signo, por lo que quedaría tal que así:

$$u = \begin{pmatrix} 3 \\ -2 \end{pmatrix}.$$

Con este vector u ya tenemos los coeficientes de la recta r representada en azul, es decir, $r : 3x_1 - 2x_2 = C$. El siguiente paso es obtener el valor del término independiente de la recta, es decir, el valor de C. Para ello, sustituimos un punto de la recta, por ejemplo el (1,1). De este modo, se tiene que

$$3 \times 1 - 2 \times 1 = C; \quad C = 1.$$

Así pues, la reta que buscamos es: $r = 3x_1 - 2x_2 = 1$.

Por último, tal y como hemos comentado anteriormente, la distancia que buscamos es la de un punto $P = (P_{1,}\ P_2)$ a una recta $r : Ax_1 + Bx_2 = C$. Para ello, debemos aplicar la fórmula correspondiente:

$$d(P, r) = \frac{|AP_1 + BP_2 - C|}{\sqrt[2]{A^2 + B^2}}.$$

Por lo tanto, la distancia del punto $P = (0, 0)$ a la recta $r : 3x_1 - 2x_2 = 1$ quedaría:

$$d(P, r) = \frac{|3 \times 0 - 2 \times 0 - 1|}{\sqrt[2]{3^2 + (-2)^2}} = \frac{1}{\sqrt[2]{13}}.$$

3.2 Implementación en Matlab de la distancia a la inconsistencia con norma euclídea

En esta sección, comentaremos nuestras principales aportaciones en relación con la implementación en MATLAB del cálculo de la distancia a la inconsistencia con norma euclídea. La función correspondiente se encuentra en el apéndice 4.1.

3.2.1 Caso $\mathbf{0}_{n+1} \notin \mathbf{H}$

Teniendo en cuenta la proposición 3.1.1 apartado i), nos encontramos ante un sistema consistente, por lo que nuestro objetivo es encontrar la distancia a la inconsistencia.

El problema que vamos a resolver es el siguiente:

$$\begin{array}{cc} Min & d\left(0_{n+1},\ z\right) \\ s.a & z \in H \end{array},$$

que teniendo en cuenta la norma euclídea quedaría:

$$\begin{array}{cc} Min & \sqrt[2]{z_1^2 + z_2^2 + ... + z_{n+1}^2} \\ s.a & z \in H \end{array},$$

o bien:

$$\begin{array}{lc} Min & z_1^2 + z_2^2 + ... + z_{n+1}^2 \\ s.a & z = \sum_{i=1}^{m} \lambda_i \begin{pmatrix} a_i \\ b_i \end{pmatrix} + \mu \begin{pmatrix} 0_n \\ -1 \end{pmatrix} \\ con & \lambda_i \geq 0 \; para \; i = 1, ..., m, \\ & \mu \geq 0, \\ & \sum_{i=1}^{m} \lambda_i = 1. \end{array}$$

Tal y como vemos en el modelo anterior, se trata de un modelo de Programación Cuadrática (PC por brevedad), por lo que nos ayudamos de la función "quadprog" de MATLAB para resolverlo. A continuación, comentamos algunos aspectos de sintaxis de dicha función que se ajustan a la ayuda de MATLAB. La función quadprog admite diferentes argumentos de entrada y salida (escríbase “help quadprog” para más detalles). La versión más sencilla es “$x = quadprog(Q, c, A, B)$”, donde $x \in \mathbb{R}^s$ es la solución propuesta por MATLAB del problema:

$$\begin{array}{lc} Min & \frac{1}{2} x' Q x + c' x \\ s.a & Ax \leq B, \end{array}$$

siendo Q una matriz cuadrada simétrica de orden s, $c \in \mathbb{R}^s$, y A,B son las matrices de coeficientes y términos independientes del sistema de desigualdades lineales.

Puede ampliarse el número de argumentos de entrada en los formatos:

- “$x = quadprog(Q, c, A, B, Aeq, Beq)$” que resuelve el problema anterior añadiendo como restricciones el sistema de ecuaciones ‘$Aeq\ x = Beq$ ’.
- “$x = quadprog(Q, c, A, B, Aeq, Beq, LB, UB)$” añade al anterior el vector de cotas inferiores LB y de cotas superiores UB; esto es, $LB \leq x \leq UB$.

Junto a cada una de las modalidades anteriores, se dispone de diferentes opciones respecto de los argumentos de salida: así, por ejemplo si escribimos ‘$[x,\ fval] = quadprog(Q, c, A, b, ...)$ ’ obtendremos, además de la solución propuesta, el valor del objetivo en dicha solución ($fval$).

Ejemplo 3.2.1 *Dado el sistema* $\sigma = \{x_1 \geq 1,\ 2x_1 \geq 3\}$*, sabemos que* $0_2 \notin H$ *y por tanto* $\sigma \in \Theta_c$ *y mediante el modelo anterior calculamos la distancia a la inconsistencia. Dicha distancia coincide con el valor óptimo del problema, calculado mediante MATLAB.*

El problema que debemos resolver es el siguiente:

$$\begin{array}{ll} Min & z_1^2 + z_2^2 \\ s.a & \begin{pmatrix} z_1 \\ z_2 \end{pmatrix} = \lambda_1 \begin{pmatrix} 1 \\ 1 \end{pmatrix} + \lambda_2 \begin{pmatrix} 2 \\ 3 \end{pmatrix} + \mu \begin{pmatrix} 0 \\ -1 \end{pmatrix}, \\ & \lambda_1 + \lambda_2 = 1, \\ & \lambda_1, \lambda_2, \mu \geq 0. \end{array}$$

Si operamos matricialmente y dejamos las ecuaciones igualadas a cero, obtenemos el siguiente problema:

$$\begin{array}{ll} Min & z_1^2 + z_2^2 \\ s.a & z_1 - \lambda_1 - 2\lambda_2 = 0, \\ & z_2 - \lambda_1 - 3\lambda_2 + \mu = 0, \\ & \lambda_1 + \lambda_2 = 1, \\ & \lambda_1, \lambda_2, \mu \geq 0. \end{array}$$

De aquí obtenemos las siguientes matrices, teniendo en cuenta que el orden de las variables que hemos utilizado para la costrucción de las matrices es: $z_1, z_2, \lambda_1, \lambda_2, \mu$.

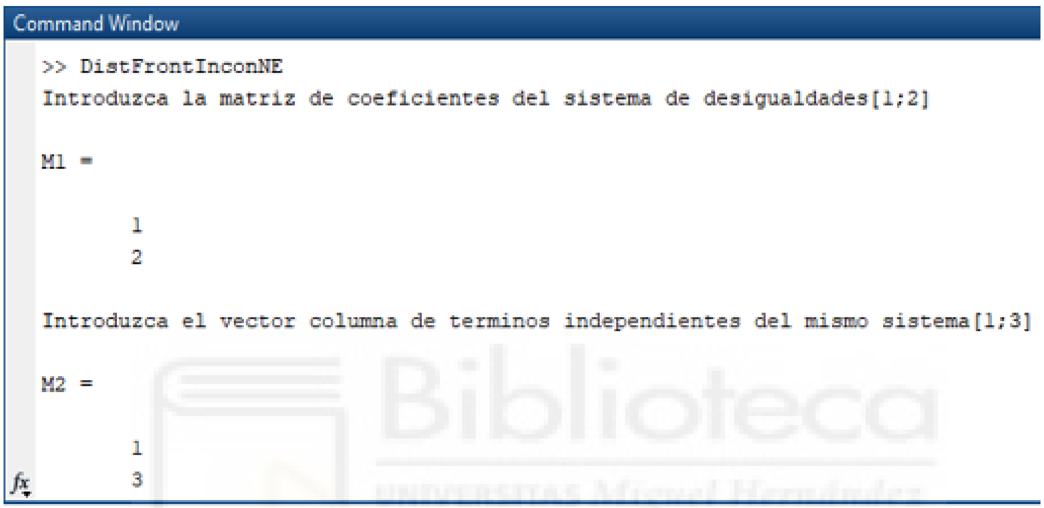
$$C = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}_{\substack{n+m+2 \text{ filas} \\ 1 \text{ columna}}} \qquad Q = \begin{pmatrix} 2 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}_{\substack{n+m+2 \text{ filas} \\ n+m+2 \text{ columnas}}}$$

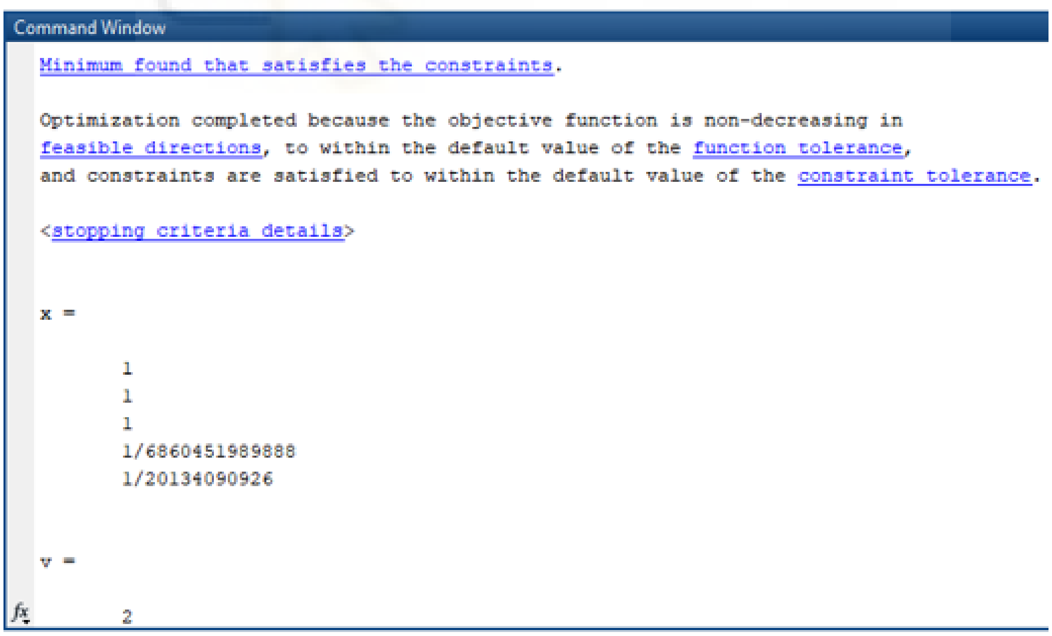
$$Aeq = \begin{pmatrix} 1 & 0 & -1 & -2 & 0 \\ 0 & 1 & -1 & -3 & -1 \\ 0 & 0 & 1 & 1 & 0 \end{pmatrix}_{\substack{n+2 \text{ filas} \\ n+m+2 \text{ columnas}}} \qquad beq = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}_{\substack{n+2 \text{ filas} \\ 1 \text{ columna}}}$$

$$A = [] \qquad b = []$$

$$lb = \begin{pmatrix} -\inf & -\inf & 0 & 0 & 0 \end{pmatrix}_{\substack{1 \text{ fila} \\ n+m+2 \text{ columnas}}} \qquad ub = []$$

Para la obtención de las matrices anteriores, se ha llevado a cabo un programa en MATLAB que basta con que el usuario inserte la matriz de coeficientes y el vector columna de terminos independientes del sistema para que dicho programa sea capaz de construir las matrices automáticamente. Veamos a continuación lo que ocurre cuando ejecutamos el programa y cómo debemos introducir los datos para que posteriormente MATLAB nos proporcione la solución al problema, tal y como vemos en la segunda imagen.





En cuanto a la solución proporcionada por MATLAB, los valores de x en realidad son los valores de las variables que hemos utilizado en el problema, siguiendo el orden que hemos establecido al principio, es decir, $z_1, z_2, \lambda_1, \lambda_2, \mu$. Así pués, en este problema z_1 y z_2

toman el valor 1, por lo que el punto $(1,1)$ del conjunto H es donde se alcanza la mínima norma euclídea. Por último, vemos que el valor óptimo del problema toma el valor 2, pero al tratarse de un problema de Programación Cuadrática, en realidad debemos aplicar la raiz cuadrada de ese valor para obtener el verdadero resultado. Así pues, la distancia que estamos buscando, es decir, la distancia a la inconsistencia es de $\sqrt{2}$ unidades. Esto significa que cualquier modificación de los vectores de coeficientes del sistema cuya norma sea menor que $\sqrt{2}$ conduce a un sistema consistente.

3.3 Implementación en Matlab de la distancia a la consistencia e inconsistencia con norma del supremo

Tal y como hemos hecho en la sección anterior, comentaremos nuestras principales aportaciones en relación con la implementación en MATLAB del cálculo de la distancia a la consistencia/inconsistencia (pero esta vez con norma del supremo). La función correspondiente se encuentra en los apéndices 4.2 y 4.3, respectivamente.

Sea $u \in \mathbb{R}^k$, recordemos que la norma del supremo $\|u\|_\infty = \max\{|u_i| \text{ para } i = 1, ..., k\}$. Sea $B_\infty = \left\{u \in \mathbb{R}^k \mid \|u\|_\infty \leq 1\right\}$ la bola unidad cerrada en $\|\cdot\|_\infty$.

Gráficamente, en $\mathbb{R}^2$ la bola unidad es:

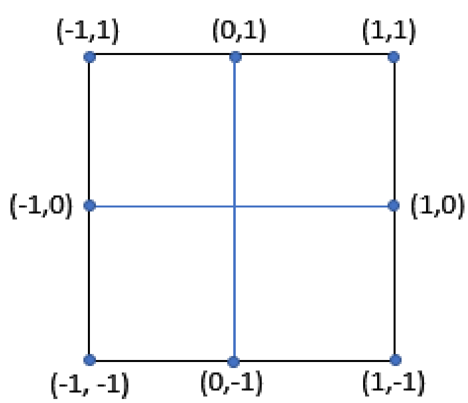


Figura 10: Elaboración propia

3.3.1 Caso $0_{n+1} \notin \mathbf{H}$

Teniendo en cuenta la proposición 3.1.1 apartado i), nos encontramos ante un sistema consistente, por lo que nuestro objetivo es encontrar la distancia a la inconsistencia.

Por el teorema 3.1.1 sabemos que $d(\sigma, bd\,(\Theta_c)) = d(0_{n+1}, bd\,(H)) = d\,(0_n, H)$.

El problema que vamos a resolver es:

$$\begin{array}{ll} Min & \|z\|_{\infty} \\ s.a & z \in H \end{array}.$$

Atendiendo a la definición de norma del supremo quedaría:

$$\begin{array}{ll} Min & \max\{|z_i| \quad \text{para } i = 1, ..., n+1\} \\ s.a & z \in H. \end{array}$$

Ahora, si sustituimos la función objetivo:

$$\begin{array}{ll} Min & d \\ s.a & |z_i| \leq d \quad \text{para } i = 1, ..., n+1\ , \\ & z \in H, \end{array}$$

o lo que es lo mismo:

$$\begin{array}{ll} Min & d \\ s.a & -d \leq z_i \leq d \quad \text{para } i = 1, ..., n+1, \\ & z \in H. \end{array}$$

Para resolverlo, vamos a hacer uso de la función linprog de MATLAB. Recordemos que esta función resuelve problemas de Programación Lineal adaptados al formato:

$$\begin{array}{ll} Min & c^{'}x \\ s.a & Ax \leq b, \\ & Aeq\ x = beq, \\ & LB \leq x \leq UB, \end{array}$$

donde $x \in \mathbb{R}^n$ es el vector de variables de decisión, $c \in \mathbb{R}^n$ es el vector de coeficientes de la función objetivo, '$Ax \leq b$ ' y '$Aeq\ x = beq$ ' representan, respectivamente, el sistema de desigualdades lineales y de ecuaciones lineales en formato matricial, y finalmente LB (del inglés, lower bound) y UB (del inglés, upper bound) son vectores de $\mathbb{R}^n$ cuyas coordenadas coinciden con las cotas inferiores y las cotas superiores, respectivamente, de las variables del problema.

Recordemos a continuación la sintaxis de la función linprog:

- La sintaxis minimal de esta función es: $x = linprog(c, a, b)$.
- Pueden añadirse otros argumentos de entrada con la sintaxis: $x = linprog(c, a, b, Aeq, beq, LB, UB)$ dependiende del tipo de restricciones del problema. Puede escribirse $LB = []$ o $UB = []$ cuando el problema no presente cotas inferiores o superiores.

Junto a cada una de las modalidades anteriores, se dispone de diferentes opciones respecto de los argumentos de salida: así, por ejemplo si escribimos ' $[x,\ fval] = linprog(c, A, b, ...)$ ' obtendremos, además de la solución propuesta, el valor del objetivo en dicha solución ($fval$).

Ejemplo 3.3.1 *Dado $\sigma = \{x_1 + x_2 \geq 0,\ x_1 - x_2 \geq 0\}$, sabemos que $0_3 \notin H$ y por tanto $\sigma \in \Theta_c$ y mediante el modelo anterior calculamos la distancia a la inconsistencia. Dicha distancia coincide con el valor óptimo del problema, calculado mediante MATLAB.*

El modelo que debemos resolver es:

$$
\begin{array}{ll}
Min & d \\
s.a & -d \leq z_1 \leq d, \\
 & -d \leq z_2 \leq d, \\
 & -d \leq z_3 \leq d, \\
 & \begin{pmatrix} z_1 \\ z_2 \\ z_3 \end{pmatrix} = \lambda_1 \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix} + \lambda_2 \begin{pmatrix} 1 \\ -1 \\ 0 \end{pmatrix} + \mu \begin{pmatrix} 0 \\ 0 \\ -1 \end{pmatrix}, \\
 & \lambda_1 + \lambda_2 = 1, \\
 & \lambda_1, \lambda_2, \mu \geq 0.
\end{array}
$$

O lo que es lo mismo:

$$
\begin{array}{ll}
Min & d \\
s.a & z_1 + d \geq 0, \\
 & -z_1 + d \geq 0, \\
 & z_2 + d \geq 0, \\
 & -z_2 + d \geq 0, \\
 & z_3 + d \geq 0, \\
 & -z_3 + d \geq 0, \\
 & z_1 = \lambda_1 + \lambda_2, \\
 & z_2 = \lambda_1 - \lambda_2, \\
 & z_3 = -\mu, \\
 & \lambda_1 + \lambda_2 = 1, \\
 & \lambda_1, \lambda_2, \mu \geq 0.
\end{array}
$$

Transformando las inecuaciones al formato de $\leq$ y las ecuaciones igualadas a cero, quedaría:

$$
\begin{array}{ll}
Min & d \\
s.a & -z_1 - d \leq 0, \\
 & z_1 - d \leq 0, \\
 & -z_2 - d \leq 0, \\
 & z_2 - d \leq 0, \\
 & -z_3 - d \leq 0, \\
 & z_3 - d \leq 0, \\
 & z_1 - \lambda_1 - \lambda_2 = 0, \\
 & z_2 - \lambda_1 + \lambda_2 = 0, \\
 & z_3 + \mu = 0, \\
 & \lambda_1 + \lambda_2 = 1, \\
 & \lambda_1, \lambda_2, \mu \geq 0,
\end{array}
$$

De aquí obtenemos las siguientes matrices, teniendo en cuenta que el orden de las variables que hemos utilizado para la costrucción de las matrices es: $z_1, z_2, z_{3,}\lambda_1, \lambda_2, \mu, d$.

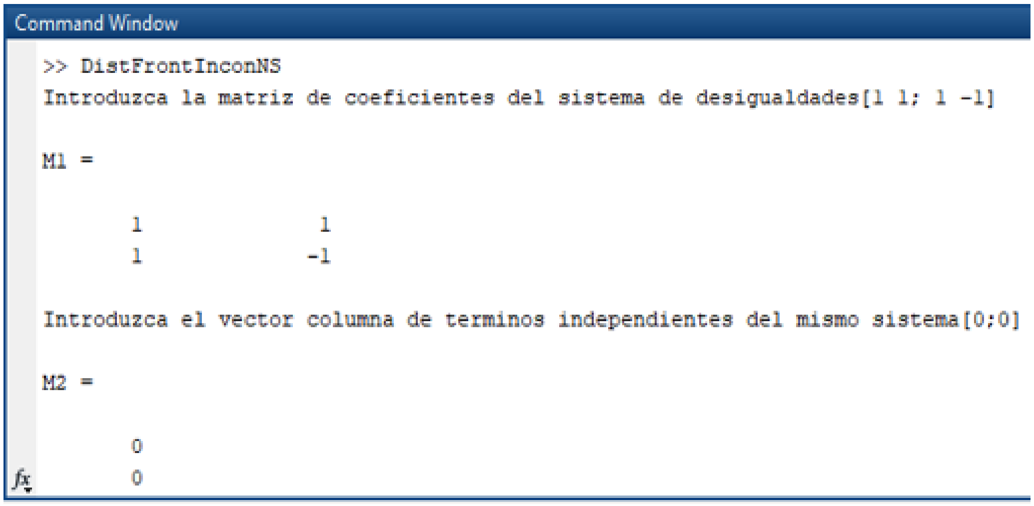
$$C = \underset{\substack{n+m+3 \text{ filas} \\ 1 \text{ columna}}}{\begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}} \qquad A = \underset{\substack{2*(n+1) \text{ filas} \\ n+m+3 \text{ columnas}}}{\begin{pmatrix} -1 & 0 & 0 & 0 & 0 & 0 & -1 \\ 1 & 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & -1 & 0 & 0 & 0 & 0 & -1 \\ 0 & 1 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & -1 & 0 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 & 0 & 0 & -1 \end{pmatrix}}$$

$$b = \underset{\substack{2*(n+1) \text{ filas} \\ 1 \text{ columna}}}{\begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}} \qquad Aeq = \underset{\substack{n+2 \text{ filas} \\ n+m+3 \text{ columnas}}}{\begin{pmatrix} 1 & 0 & 0 & -1 & -1 & 0 & 0 \\ 0 & 1 & 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 \end{pmatrix}}$$

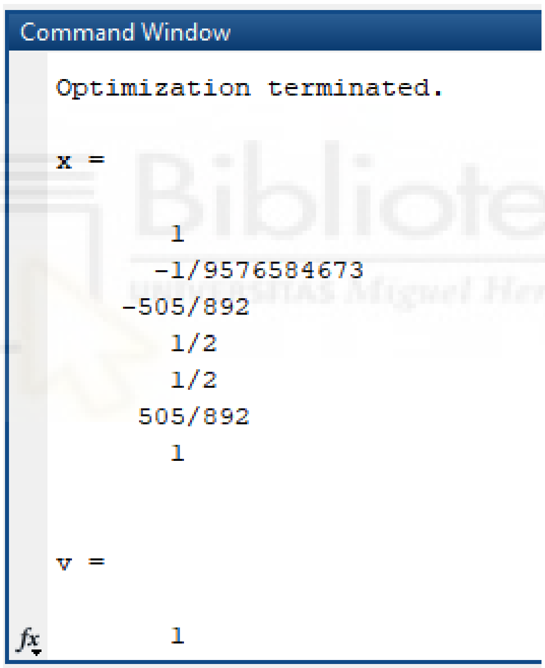
$$beq = \underset{\substack{n+2 \text{ filas} \\ 1 \text{ columna}}}{\begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}} \qquad lb = \underset{\substack{1 \text{ fila} \\ n+m+3 \text{ columnas}}}{\begin{pmatrix} -\inf & -\inf & -\inf & 0 & 0 & 0 & -\inf \end{pmatrix}}$$

$$ub = ()$$

Tal y como hemos comentado en la sección anterior, para la obtención de las matrices, se ha llevado a cabo un programa en MATLAB que basta con que el usuario inserte la matriz de coeficientes y el vector columna de terminos independientes del sistema para que sea capaz de construir dichas matrices de forma sistemática. A continuación, en la primera imagen podemos ver lo que ocurre cuando ejecutamos dicho programa y cómo debemos introducir los datos para que posteriormente MATLAB nos proporcione la solución al problema, tal y como vemos en la segunda imagen.



```
Command Window
>> DistFrontInconNS
Introduzca la matriz de coeficientes del sistema de desigualdades[1 1; 1 -1]

M1 =

     1     1
     1    -1

Introduzca el vector columna de terminos independientes del mismo sistema[0;0]

M2 =

     0
     0
```



```
Command Window
Optimization terminated.

x =

       1
    -1/9576584673
  -505/892
     1/2
     1/2
   505/892
       1


v =

       1
```


En cuanto a la solución proporcionada por MATLAB, los valores de x en realidad son los valores de las variables que hemos utilizado en el problema, siguiendo el orden que hemos establecido al principio, es decir, $z_1, z_2, z_3, \lambda_1, \lambda_2, \mu, d$.

Así pués, en este problema z_1, z_2 y z_3 toman los valores 1, 0 y $-\frac{505}{892}$, respectivamente, por lo que el vector $(1, 0, -\frac{505}{892})$ del conjunto H es donde se alcanza la mínima norma del supremo, mientras que la variable d toma el valor 1 (eso quiere decir que el valor de dicha norma es de 1 unidad).

Por último, vemos que el valor óptimo del problema también toma el valor 1, por lo que la distancia que estamos buscando, es decir, la distancia a la inconsistencia, es de una unidad. En otras palabras, cualquier modificación de los vectores de coeficientes del sistema cuya norma sea menor que 1 conduce a un sistema consistente.

3.3.2 Caso $0_{n+1} \in$ H

Teniendo en cuenta la proposición 3.1.1 apartado iii), nos encontramos ante un sistema que puede ser consistente de la frontera o inconsitente.

Por el teorema 3.1.1 sabemos que $d(\sigma, bd\,(\Theta_c)) = d(0_{n+1}, bd\,(H))$.

Para calcular la distancia de 0_{n+1} a $bd\,(H)$, hemos de calcular el radio de la mayor bola en $\|\cdot\|_\infty$ que se encuentra contenida en H.

Así pués, el problema que vamos a resolver es el siguiente:

$$\begin{array}{ll} Max & r \\ s.a & r * B_\infty \subset H. \end{array}$$

Nota:

$$r * B_\infty = \left\{u \in \mathbb{R}^k \mid \|u\|_\infty \leq r\right\}.$$

Teniendo en cuenta que H es convexo, el hecho de que la bola $r * B_\infty \subset H$ es equivalente a que estén sus puntos extremos.

Ejemplo 3.3.2 *En* $\mathbb{R}^3$ *los puntos extremos son:* $(1,1,1)\,,\ (1,-1,1)\,,\ (-1,1,1)\,,\ (-1,-1,1)\,,\ (1,-1,-1)\,,\ (-1,1,-1)\,,\ (-1,-1,-1)\,,\ (1,1,-1)\,.$

Siendo $\sigma = \{x_1 + x_2 \geq 0,\ x_1 - x_2 \geq 0,\ -x_1 \geq 1\}$*, sabemos que* $0_3 \in H$ *y por tanto* $\sigma \in \Theta$*.El sistema puede ser consistente de la frontera, en cuyo caso la distancia* $d(0, bd\,(H))$ *sería cero, o bien del interior del conjunto de sistemas inconsistentes, en cuyo caso esta distancia sería positiva. Dicha distancia coincide con el valor óptimo del problema, calculado mediante MATLAB.*

Gráficamente, en $\mathbb{R}^3$ la bola unidad es:

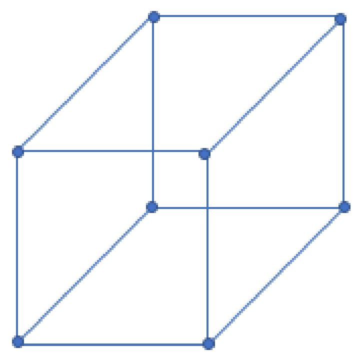


Figura 11: Elaboración propia

A la hora de realizar el modelo, solamente debemos tener en cuenta los puntos extremos cuya tercera coordenada sea positiva ya que gráficamente observamos que si los puntos extremos situados en la parte superior forman parte del conjunto H, podemos asegurar que todos los demás también lo harán.

Por lo tanto, el modelo que debemos resolver es el siguiente:

$$\begin{aligned}
&Max \quad r\\
&s.a \quad \begin{pmatrix} r \\ r \\ r \end{pmatrix} = \lambda_{11}\begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix} + \lambda_{12}\begin{pmatrix} 1 \\ -1 \\ 0 \end{pmatrix} + \lambda_{13}\begin{pmatrix} -1 \\ 0 \\ 1 \end{pmatrix} + \mu_1\begin{pmatrix} 0 \\ 0 \\ -1 \end{pmatrix},\\
&\begin{pmatrix} r \\ -r \\ r \end{pmatrix} = \lambda_{21}\begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix} + \lambda_{22}\begin{pmatrix} 1 \\ -1 \\ 0 \end{pmatrix} + \lambda_{23}\begin{pmatrix} -1 \\ 0 \\ 1 \end{pmatrix} + \mu_2\begin{pmatrix} 0 \\ 0 \\ -1 \end{pmatrix},\\
&\begin{pmatrix} -r \\ r \\ r \end{pmatrix} = \lambda_{31}\begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix} + \lambda_{32}\begin{pmatrix} 1 \\ -1 \\ 0 \end{pmatrix} + \lambda_{33}\begin{pmatrix} -1 \\ 0 \\ 1 \end{pmatrix} + \mu_3\begin{pmatrix} 0 \\ 0 \\ -1 \end{pmatrix},\\
&\begin{pmatrix} -r \\ -r \\ r \end{pmatrix} = \lambda_{41}\begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix} + \lambda_{42}\begin{pmatrix} 1 \\ -1 \\ 0 \end{pmatrix} + \lambda_{43}\begin{pmatrix} -1 \\ 0 \\ 1 \end{pmatrix} + \mu_4\begin{pmatrix} 0 \\ 0 \\ -1 \end{pmatrix},\\
&\lambda_{11} + \lambda_{12} + \lambda_{13} = 1,\\
&\lambda_{21} + \lambda_{22} + \lambda_{23} = 1,\\
&\lambda_{31} + \lambda_{32} + \lambda_{33} = 1,\\
&\lambda_{41} + \lambda_{42} + \lambda_{43} = 1,\\
&r, \lambda_{11}, \lambda_{12}, \lambda_{13}, \lambda_{21}, \lambda_{22}, \lambda_{23}, \lambda_{31}, \lambda_{32}, \lambda_{33}, \lambda_{41}, \lambda_{42}, \lambda_{43}, \mu_1, \mu_2, \mu_3, \mu_4 \geq 0.
\end{aligned}$$

Si operamos matricialmente y transformamos las ecuaciones correspondientes a las μs en inecuaciones de tal forma que desaparezcan las μs, obtenemos el siguiente modelo:

$$
\begin{aligned}
Min \quad & -r \\
s.a & \\
& r - \lambda_{11} - \lambda_{12} + \lambda_{13} = 0, \\
& r - \lambda_{11} + \lambda_{12} = 0, \\
& r - \lambda_{21} - \lambda_{22} + \lambda_{23} = 0, \\
& -r - \lambda_{21} + \lambda_{22} = 0, \\
& -r - \lambda_{31} - \lambda_{32} + \lambda_{33} = 0, \\
& r - \lambda_{31} + \lambda_{32} = 0, \\
& r - \lambda_{41} - \lambda_{42} + \lambda_{43} = 0, \\
& -r - \lambda_{41} + \lambda_{42} = 0, \\
& \lambda_{11} + \lambda_{12} + \lambda_{13} = 1, \\
& \lambda_{21} + \lambda_{22} + \lambda_{23} = 1, \\
& \lambda_{31} + \lambda_{32} + \lambda_{33} = 1, \\
& \lambda_{41} + \lambda_{42} + \lambda_{43} = 1, \\
& r - \lambda_{13} \leq 0, \\
& r - \lambda_{23} \leq 0, \\
& r - \lambda_{33} \leq 0, \\
& r - \lambda_{43} \leq 0, \\
& r, \lambda_{11}, \lambda_{12}, \lambda_{13}, \lambda_{21}, \lambda_{22}, \lambda_{23}, \lambda_{31}, \lambda_{32}, \lambda_{33}, \lambda_{41}, \lambda_{42}, \lambda_{43} \geq 0.
\end{aligned}
$$

Teniendo en cuenta que el orden de las variables que hemos utilizado para la construcción de las matrices es: $r, \lambda_{11}, \lambda_{12}, \lambda_{13}, \lambda_{21}, \lambda_{22}, \lambda_{23}, \lambda_{31}, \lambda_{32}, \lambda_{33}, \lambda_{41}, \lambda_{42}, \lambda_{43}$, se obtiene:

$$C = \begin{pmatrix} -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}'_{\substack{\text{1 fila} \\ 1+m*(2^n) \text{ columnas}}}$$

$$Aeq = \begin{pmatrix} 1 & -1 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & -1 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & 1 \\ -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{pmatrix}_{\substack{n*(2^n)+(2^n) \text{ filas} \\ 1+m*(2^n) \text{ columnas}}}$$

$$beq = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}'_{\substack{\text{1 fila} \\ n*(2\hat{}n)+(2\hat{}n) \text{ columnas}}}$$

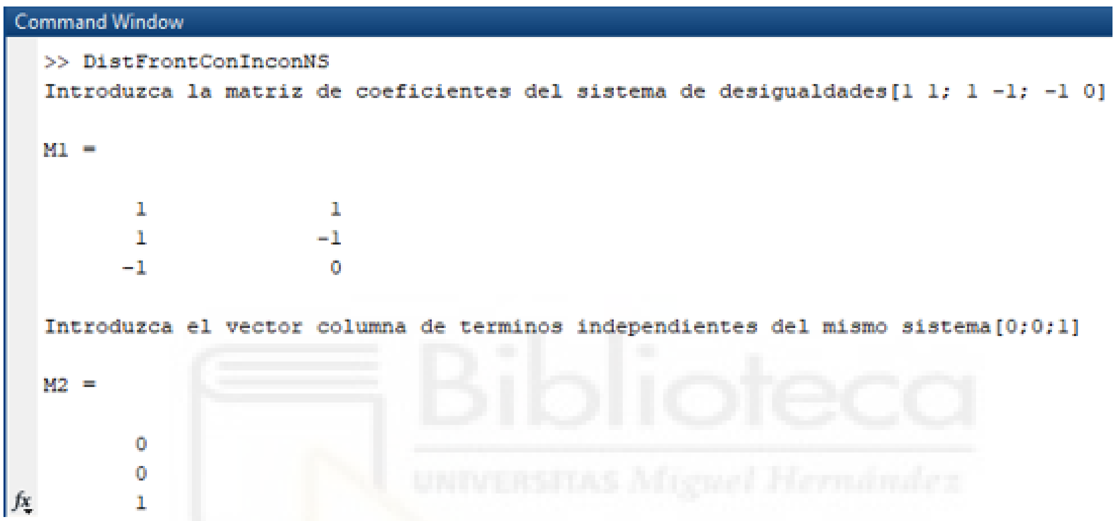
$$A = \begin{pmatrix} 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \end{pmatrix}_{\substack{(2^n) \text{ filas} \\ 1+m*(2^n) \text{ columnas}}}$$

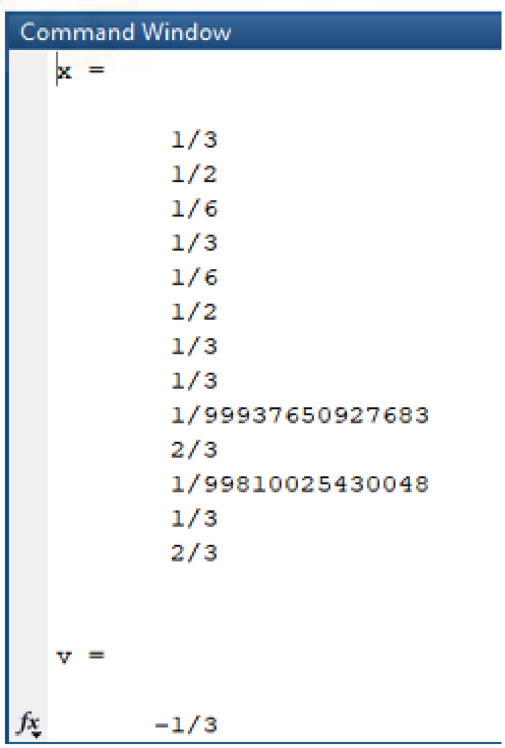
$$b = \begin{pmatrix} 0 & 0 & 0 & 0 \end{pmatrix}'_{\substack{\text{1 fila} \\ (2\hat{}n) \text{ columnas}}}$$

$$lb = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}_{\substack{\text{1 fila} \\ 1+m*(2^n) \text{ columnas}}}$$

$$ub = ()$$

Una vez más, para la obtención de las matrices, se ha llevado a cabo un programa en MATLAB que basta con insertar la matriz de coeficientes y el vector columna de terminos independientes del sistema para que sea capaz de construir dichas matrices de forma sistemática. A continuación, en la primera imagen vemos lo que ocurre cuando ejecutamos dicho programa y cómo debemos introducir los datos para que posteriormente MATLAB nos proporcione la solución al problema, tal y como vemos en la segunda imagen.





En cuanto a la solución proporcionada por MATLAB, los valores de x en realidad son los valores de las variables que hemos utilizado en el problema, siguiendo el orden que hemos establecido al principio: $r, \lambda_{11}, \lambda_{12}, \lambda_{13}, \lambda_{21}, \lambda_{22}, \lambda_{23}, \lambda_{31}, \lambda_{32}, \lambda_{33}, \lambda_{41}, \lambda_{42}, \lambda_{43}$.

Así pués, en este problema r mide la distancia a la frontera de la consistencia/inconsitencia, cuyo valor es de $\frac{1}{3}$.

Por último, vemos que el valor óptimo es negativo pero debemos recordar que para la implementación en Matlab hicimos un cambio en la función objetivo, de maximizar la transformamos a minimizar, por lo que realmente el valor óptimo es de $\frac{1}{3}$. En cuanto a su interpretación, al ser positivo, es seguro que 0_{n+1} está en el interior de H, y por tanto el sistema es inconsistente. Así pues, el valor óptimo nos indica la distancia a la consistencia; esto es, el tamaño de la menor perturbación del sistema que lo transforma en un sistema consistente.

3.3.3 Ejemplos

En los ejemplos vistos hasta ahora conocíamos de antemano en qué caso nos encontrábamos ya que queriamos ilustrar lo visto en cada apartado. A continuación, vamos a exponer unos ejemplos que nos van a permitir saber a posteriori en que caso nos encontramos.

Ejemplo 3.3.3 *Siendo* $\sigma = \{-x_1 - x_2 \geq -2, \;\; x_1 - x_2 \geq -1, \;\; x_1 \geq 0, \;\; x_2 \geq 0\}$, *nuestro objetivo es calcular la distancia a la consistencia o a la inconsistencia, según corresponda.*

Como a priori no sabemos si se trata de un sistema consistente o inconsistente, probamos primero con el programa que hemos desarrollado y hemos llamado "DistInconNS" y vemos que el valor óptimo que nos devuelve MATLAB es de $\frac{2}{7}$ unidades, por lo que al ser estrictamente mayor que 0, podemos asegurar que se trata de un sistema consistente y lo que hemos calculado ha sido la distancia a la inconsistencia.

Ejemplo 3.3.4 *Siendo* $\sigma = \{x_1 + x_2 \geq -1, \quad -x_1 - x_2 \geq -1, \quad x_1 \geq 0, \quad x_2 \geq 0\}$, *se desea calcular la distancia a la consistencia o a la inconsistencia, según corresponda.*

Tal y como hemos comentado antes, no sabemos si se trata de un sistema consistente o inconsistente por lo que probamos primero con el programa que hemos desarrollado y hemos llamado "DistInconNS" y vemos que el valor óptimo que nos devuelve MATLAB es de 0 unidades, lo cual quiere decir que puede que esté en el interior del conjunto H o bien en la frontera, por lo que ahora ejecutamos el segundo programa llamado "DistConsNS". Al resolverlo vemos que el valor óptimo del problema es $\frac{1}{3}$, por lo que al tratarse de un valor positivo podemos asegurar que 0_3 está en el interior de H y por tanto se trata de un sistema inconsistente y cuya distancia a la consistencia es $\frac{1}{3}$.

Ejemplo 3.3.5 *Siendo* $\sigma = \{x_1 + x_2 \geq 2, \quad x_1 - x_2 \geq -3, \quad -x_1 + x_2 \geq 5, \quad x_1 \geq 0\}$, *se pretende calcular la distancia a la consistencia o a la inconsistencia, según corresponda.*

Una vez más, no sabemos si se trata de un sistema consistente o inconsistente así que probamos primero con el programa "DistInconNS" y vemos que el valor óptimo que nos devuelve MATLAB es de 0 unidades, lo cual quiere decir que puede que esté dentro del conjunto H o bien en la frontera, por lo que ahora ejecutamos el segundo programa "DistConsNS". El valor óptimo vuelve a ser 0, por lo que podemos asegurar que se trata de un sistema que se encuentra en la frontera de los consistentes, la cual coincide con la frontera de los inconsistentes. En este caso, en el que los valores óptimos de ambos programas es cero, para analizar la consistencia/inconsistencia han de utilizarse otras herramientas. En cualquier caso, el problema de hallar la distancia a la frontera de la consistencia/inconsistencia se ha resuelto.

Capítulo 4

Apéndice: Códigos en MATLAB

4.1 Distancia a la frontera de la inconsistencia con norma euclídea

```
%El presente programa resuelve el problema que minimiza la distancia
%de un sistema a la frontera de la consistencia, a través del sistema
%de desigualdades lineales A*x>=b.
clear
M1=input('Introduzca la matriz de coeficientes del sistema de desigualdades')
M2=input('Introduzca el vector columna de terminos independientes del mismo sistema')
M3=[M1 M2]'
[m,n]=size(M1)
c=zeros(n+m+2,1)
Q=zeros(n+m+2)
%Añadimos doses a la diagonal de la submatriz de tamaño m
for i=1:n+1
Q(i,i)=2
end
%Matrices A y B vacías ya que no tenemos inecuaciones
A=[]
b=[]
%Inicializamos la matriz Aeq
Aeq=zeros(n+2,n+m+2)
%Rellenamos la primera submatriz de Aeq con unos en la diagonal
```

```
for i=1:n+1
Aeq(i,i)=1
end
%Rellenamos la última fila de la matriz Aeq con unos donde corresponda
for j=n+2:n+m+1
Aeq(n+2,j)=ones
end
%Rellenamos la última columna de la matriz Aeq con el vecton (0n -1)'
Aeq(n+1,n+m+2)=-1
%Rellenamos con los datos iniciales
for i=1:n+1
for j=n+2:n+m+1
Aeq(i,j)=-(M3(i,j-(n+1)))
end
end
%Inicializamos la matriz beq
beq=zeros(n+2,1)
beq(n+2,1)=1
%Inicializamos las cotas superiores e inferiores
ub=[]
lb=zeros(1,n+m+2)
for j=1:n+1
lb(1,j)=-inf
end
%Resolvemos el problema
[x,v]=quadprog(Q,c,A,b,Aeq,beq,lb,ub)
```


4.2 Distancia a la frontera de la inconsistencia con norma del supremo

```
clear
M1=input('Introduzca la matriz de coeficientes del sistema de desigualdades')
M2=input('Introduzca el vector columna de terminos independientes del mismo sistema')
M3=[M1 M2]'
```

```
[m,n]=size(M1)
%Inicializamos la matriz c
c=zeros(n+m+3,1)
c(n+m+3,1)=1
%Inicializamos la matriz A
A=zeros(2*(n+1),n+m+3)
%Rellenamos las n+1 primeras columnas de la matriz
for i=1:2*(n+1)
for j=1:n+1
if mod(i,2)~=0 %El modulo nos develve el resto de la division.
A(i,(floor(i/2)+ mod(i,2)))=(-1) %Floor nos redondea al entero inferior
else
A(i,(floor(i/2)+ mod(i,2)))= 1
end
end
end
%Rellenamos la última columna de A
for i=1:2*(n+1)
A(i,n+m+3)=-1
end
%Inicializamos la matriz b
b=zeros(2*(n+1),1)
%Inicializamos la matriz Aeq
Aeq=zeros(n+2,n+m+3)
%Rellenamos la primera submatriz de Aeq con unos en la diagonal
for i=1:n+1
Aeq(i,i)=1
end
%Rellenamos la última fila de la matriz Aeq con unos donde corresponda
for j=n+2:n+m+1
Aeq(n+2,j)=1
end
%Rellenamos la penúltima columna de la matriz Aeq con el vecton (0n -1)'
Aeq(n+1,n+m+2)=1
%Rellenamos con los datos iniciales
for i=1:n+1
```

```
for j=n+2:n+m+1
Aeq(i,j)=-(M3(i,j-(n+1)))
end
end
%Inicializamos la matriz beq
beq=zeros(n+2,1)
beq(n+2,1)=1
%Inicializamos las cotas superiores e inferiores
ub=[]
lb=zeros(1,n+m+3)
for j=1:n+1
lb(1,j)=-inf
end
lb(1,n+m+3)=-inf
%Resolvemos el problema
[x,v]=linprog(c,A,b,Aeq,beq,lb,ub)
```

4.3 Distancia a la frontera de la consistencia e inconsistencia con norma del supremo

```
clear
M1=input('Introduzca la matriz de coeficientes del sistema de desigualdades')
M2=input('Introduzca el vector columna de terminos independientes del mismo sistema')
M3=[M1 M2]'
M4=M2'
M5=M1'
[m,n]=size(M1)
%Inicializamos la matriz c
c=zeros(1+m*(2^n),1)
c(1,1)=(-1)
%Inicializamos la matriz A
A=zeros(2^n,1+m*(2^n))
%Rellenamos la primera columna den la matriz A
```

```
for i=1:2^n
A(i,1)=1
end
%Rellenamos el resto de la matriz A
cont=1
for i=1:2^n
for j=1:m*(2^n)+1
if j>=2+m*(i-1)&& j<=1+m*i
A(i,j)=-M4(1,cont)
cont=cont+1
end
end
cont=1
end
%Inicializamos la matriz b
b=zeros(2^n,1)
%Inicializamos la matriz Aeq
Aeq=zeros(n*(2^n)+(2^n),1+m*(2^n))
%Obtenemos todas las posibles combinaciones de n elementos
M=dec2bin(0:2^n-1)-'0'
M=1-2*M
v=[]
for i=1:2^n
v=[v;M(i,:)']
end
%Rellenamos la primera columna de la matriz Aeq
for i=1:n*(2^n)
Aeq(i,1)=v(i,1)
end
%Rellenamos con los datos iniciales la matriz Aeq
cont2=1 %Representa las columnas
cont3=1 %Representa las filas
aux=0
for i=1:n*(2^n)
for j=2:1+m*(2^n)
aux=floor((i-1)/n)%Floor nos redondea al entero inferior
```

```
if j>=2+(aux*m) && j<=(aux*m)+m+1
Aeq(i,j)=-M5(cont3,cont2)
cont2=cont2+1
end
end
cont3=cont3+1
if cont3==n+1
cont3=1
end
cont2=1
end
%Rellenamos las últimas filas de la matriz Aeq
cont4=1
for i=n*(2^n)+1:n*(2^n)+(2^n)
for j=2:1+m*(2^n)
if j>=2+m*(cont4-1) && j<=2+m*(cont4-1)+(m-1)
Aeq(i,j)=1
end
end
cont4=cont4+1
end
%Inicializamos la matriz beq
beq=zeros(n*(2^n)+(2^n),1)
%Rellenamos las 2^n ultimas filas de la matriz beq
for i=n*(2^n)+1:n*(2^n)+(2^n)
beq(i,1)=1
end
%Inicializamos las cotas superiores e inferiores
ub=[]
lb=zeros(1,1+m*(2^n))
%Resolvemos el problema
[x,v]=linprog(c,A,b,Aeq,beq,lb,ub)
```

Bibliografía

[1] Bazaraa, M.S., Jarvis, J.J., Sherali, H.D. (1998): *Programación Lineal y Flujo en Redes*, Limusa.

[2] Bertsimas, D. , Tsitsiklis, J.N. (2008): *Introduction to Linear Optimization, Dynamic Ideas and Athena Scientific*, Belmont, Massachusetts.

[3] Goberna, M.A. , Jornet, V. , Puente, R. (2004): *Optimización Lineal. Teoría, métodos y modelos.* McGraw-Hill / Interamericama de España, Madrid.

[4] Goberna, M.A. , López, M.A. (1998): *Linear Semi-Infinite Optimization*, John Wiley and Sons, Chichester (UK).

[5] Hettich, R. , Kortanek, K.O. (1993): *Semi-infinite programming: Theory, methods and applications*, SIAM Review 35, 380-429.

[6] Luenberger, D. G. (1989): *Programación Lineal y No Lineal*, Addison-Wesley Iberoamericana, México.

[7] Sierksma, G. , Zwols, Y. (2015): *Linear and Integer Optimization.* CRC Press, Taylor&Francis Group, Boca Raton.