

INSTITUTO TECNOLÓGICO AUTÓNOMO DE MÉXICO



PRECONDICIONADORES EN EL MÉTODO DE PUNTOS
INTERIORES PARA PROGRAMACIÓN CUADRÁTICA
CONVEXA

TESIS

QUE PARA OBTENER EL TÍTULO DE

LICENCIADO EN MATEMÁTICAS APLICADAS

PRESENTA

JAVIER MONTIEL GONZÁLEZ

ASESOR: DR. ZEFERINO PARADA GARCÍA

MÉXICO, D.F.

2023

“Con fundamento en el artículo 21 y 27 de la Ley Federal del Derecho de Autor y como titular de los derechos moral y patrimonial de la obra titulada “Precondicionadores en el método de puntos interiores para programación cuadrática convexa”, otorgo de manera gratuita y permanente al Instituto Tecnológico Autónomo de México y a la Biblioteca Raúl Baillères Jr. autorización para que fijen la obra en cualquier medio, incluido el electrónico y la divulguen entre sus usuarios, profesores, estudiantes o terceras personas, sin que pueda percibir por la divulgación una contraprestación.”

Javier Montiel González

FECHA

FIRMA

Prefacio

El uso de preconditionadores puede encontrarse en numerosas aplicaciones de optimización [1], por ejemplo:

- Problemas de control óptimo que surgen de la ecuación de Helmholtz. Dicha ecuación tiene aplicaciones en física al resolver problemas de sismología, acústica y radiación electromagnética.
- Problemas de programación estocástica de etapas múltiples. Este tipo de problemas surgen en situaciones prácticas donde se realizan decisiones subsecuentes a lo largo del tiempo. Entre dos decisiones subsecuentes se pueden observar los efectos aleatorios presentes en un sistema; la siguiente decisión toma esto en cuenta, además de la información disponible proveniente del pasado.
- Métodos estocásticos de reducción de varianza como SVRG y Katyusha X. Estos métodos se utilizan para minimizar el riesgo empírico que es una clase importante de problemas de optimización con muchas aplicaciones populares en *machine learning*.

El objetivo de esta tesis es comparar un grupo de preconditionadores para el método de puntos interiores en programación cuadrática convexa. Para dicha comparación, se utilizarán problemas de optimización del repositorio de Netlib con el fin de medir el rendimiento de cada uno.

En el primer capítulo se presenta el algoritmo de puntos interiores para programación cuadrática convexa. Se aborda brevemente el razonamiento detrás del método, se muestra un ejemplo de su funcionamiento y la razón por la cual resulta conveniente el uso de preconditionadores.

En el segundo capítulo, a partir del algoritmo de Lanczos, se desarrolla el método de MINRES, el cual es un método iterativo para resolver sistemas lineales con matrices simétricas indefinidas. Adicionalmente, se analizan las propiedades de convergencia del método.

En el tercer capítulo se expone la propuesta de preconditionador de Tim Rees y Chen Greif con su justificación. Del mismo modo, se analiza otro preconditionador elaborado por Eva K. Lee y Zeferino Parada. Por último, se esboza brevemente el algoritmo del método de puntos interiores preconditionado para programación cuadrática convexa.

En el cuarto capítulo se exponen los resultados numéricos obtenidos sobre un subconjunto de problemas del repositorio Netlib en los siguientes casos: con el algoritmo tradicional sin preconditionador, usando un preconditionador propuesto por Tim Rees y Chen Greif y finalmente con dos preconditionadores elaborados por Eva K. Lee y Zeferino Parada. Por último, se hace una comparación del rendimiento de cada caso.

En el capítulo final se presentan las conclusiones del análisis realizado y potenciales líneas de mejora.

Índice general

| | |
|---|-----------|
| 1. Método de Puntos Interiores para Programación Cuadrática Convexa | 1 |
| 1.1. Introducción | 1 |
| 1.2. El paso de Newton | 3 |
| 1.3. Recorte de paso | 9 |
| 1.4. Algoritmo de puntos interiores para programación cuadrática convexa | 11 |
| 1.5. Prueba del algoritmo | 13 |
| 2. MINRES | 16 |
| 2.1. Proceso de Lanczos | 16 |
| 2.2. Minimización del residual en el subespacio de Krylov de dimensión n | 21 |
| 2.3. Aproximación polinomial de MINRES | 24 |
| 2.4. Convergencia de MINRES | 25 |
| 3. Precondicionadores | 27 |
| 3.1. Normas matriciales | 27 |
| 3.2. El condicional de una matriz | 28 |
| 3.3. Elección de un preconditionador para el método de puntos interiores para programación cuadrática | 31 |
| 3.3.1. Familia de preconditionadores triangulares por bloques | 33 |

| | |
|--|-----------|
| 3.3.2. Familia de preconditionadores simétricos positivos definidos | 37 |
| 3.3.3. Precondicionador A | 40 |
| 3.3.4. Precondicionador B | 47 |
| 3.3.5. Precondicionador C | 48 |
| 3.4. Algoritmo de puntos interiores preconditionado para programación cuadrática convexa | 48 |
| 4. Resultados numéricos | 50 |
| 5. Conclusiones | 63 |

1 Método de Puntos Interiores para Programación Cuadrática Convexa

1.1 Introducción

En este capítulo se describe el problema cuadrático convexo y el método de puntos interiores para el mismo.

Considere el problema:

$$\begin{aligned} \text{Minimizar} \quad & \frac{1}{2}x^T Qx + d^T x \\ \text{Sujeto a} \quad & Ax = b \\ & Fx \geq c, \end{aligned} \tag{1.1.1}$$

donde $Q \in \mathbb{R}^{n \times n}$ es simétrica y positiva definida, $A \in \mathbb{R}^{m \times n}$, $m \leq n$ y $\text{rango}(A) = m$, $F \in \mathbb{R}^{p \times n}$, $d \in \mathbb{R}^n$, $b \in \mathbb{R}^m$ y $c \in \mathbb{R}^p$. Se define

$$F = [f_i]_{i \in I}, \quad c = [c_i]_{i \in I}, \quad I = \{1, 2, \dots, p\}.$$

Por las condiciones necesarias de Karush-Khun-Tucker (KKT), si x^* es una solución de (1.1.1), existen únicos vectores $\lambda^* \in \mathbb{R}^m$ y $\mu^* \in \mathbb{R}^p$ tales que $(x, \lambda, \mu) = (x^*, \lambda^*, \mu^*)$ satisface las siguientes condiciones:

$$\begin{aligned}
 Qx + d + A^T \lambda - F^T \mu &= 0, \\
 Ax - b &= 0, \\
 \mu_i (f_i^T x - c_i) &= 0, \quad i \in I \\
 Fx - c &\geq 0, \\
 \mu &\geq 0.
 \end{aligned}$$

Al introducir la variable de holgura $z = Fx - c$, se pueden reescribir estas condiciones como

$$\begin{aligned}
 Qx + d + A^T \lambda - F^T \mu &= 0, \\
 Ax - b &= 0, \\
 -Fx + z + c &= 0, \\
 \mu_i z_i &= 0, \quad i \in I \\
 (z, \mu) &\geq 0.
 \end{aligned} \tag{1.1.2}$$

Las condiciones de KKT no son solamente necesarias sino también suficientes porque la función objetivo y la región factible son convexas. Por lo tanto, se puede resolver el problema de programación cuadrática (1.1.1) al encontrar la solución del sistema (1.1.2).

Sea

$$H(x, \lambda, \mu, z) = \begin{pmatrix} Qx + A^T \lambda - F^T \mu + d \\ Ax - b \\ -Fx + z + c \\ UZe \end{pmatrix} = 0, \quad (\mu, z) \geq 0, \tag{1.1.3}$$

donde $U = \text{diag}(\mu)$ y $Z = \text{diag}(z)$ son matrices diagonales en $\mathbb{R}^{p \times p}$ y $e = (1, 1, \dots, 1)^T \in \mathbb{R}^p$.

Definición. La trayectoria central asociada al problema (1.1.1) es el conjunto de puntos $(x_\tau, \lambda_\tau, \mu_\tau, z_\tau)$ con $\tau > 0$ tales que

$$H_\tau(x_\tau, \lambda_\tau, \mu_\tau, z_\tau) = \begin{pmatrix} 0 \\ 0 \\ 0 \\ \tau e \end{pmatrix}, (\mu_\tau, z_\tau) > 0. \quad (1.1.4)$$

Se puede demostrar que existe $\tau^* > 0$ tal que para cada $\tau \in (0, \tau^*)$ la solución en (1.1.4) es única [2].

La idea en el método de puntos interiores es usar el método de Newton para aproximar una solución de (1.1.4) tratando que la actualización de μ_τ y z_τ sea positiva. También se disminuye el parámetro τ para tratar de seguir la trayectoria central.

1.2 El paso de Newton

Una iteración del método de Newton a (1.1.4) es

$$H'_\tau \Delta = -H_\tau,$$

cuya ecuación lineal es de la forma

$$\begin{pmatrix} Q & A^T & -F^T & 0 \\ A & 0 & 0 & 0 \\ -F & 0 & 0 & I_p \\ 0 & 0 & Z & U \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta \lambda \\ \Delta \mu \\ \Delta z \end{pmatrix} = - \begin{pmatrix} r_x \\ r_\lambda \\ r_\mu \\ r_z \end{pmatrix}, \quad (1.2.1)$$

donde $r_x = Qx + A^T \lambda - F^T \mu + d$, $r_\lambda = Ax - b$, $r_\mu = -Fx + z + c$ y $r_z = UZe - \tau e$.

El parámetro τ

Para el punto actual (x, λ, μ, z) se considera el promedio

$$\hat{\tau} = \frac{\mu^T z}{p},$$

que es una medida de la condición de complementaridad $\mu_i z_i = 0$.

El valor del parámetro de perturbación se define como

$$\tau = \sigma \hat{\tau}, \text{ donde } \sigma \in [0, 1].$$

Nótese que $\sigma = 0$ determina en (1.1.4) las condiciones necesarias de primer orden para un mínimo local. Para la implementación del método de puntos interiores utilizada en esta tesis, $\sigma = \frac{1}{2}$.

Teorema 1.2.1. Sean $\hat{G} \in \mathbb{R}^{n \times n}$, $A \in \mathbb{R}^{m \times n}$ de rango completo y $u^T \hat{G} u > 0$ para toda $u \in \text{Null}(A) \setminus \{0\}$ entonces la matriz

$$\mathcal{A} = \begin{pmatrix} \hat{G} & A^T \\ A & 0 \end{pmatrix}$$

es no singular.

Demostración. El sistema homogéneo a \mathcal{A} es

$$\mathcal{A}\omega = 0,$$

cuya ecuación lineal es de la forma

$$\begin{pmatrix} \hat{G} & A^T \\ A & 0 \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

i.e.

$$\hat{G}u + A^T v = 0, \quad (1.2.2a)$$

$$Au = 0. \quad (1.2.2b)$$

De la ecuación (1.2.2b) se tiene que u está en el espacio nulo de A . Por otro lado, al multiplicar por u^T la ecuación (1.2.2a) se obtiene

$$\begin{aligned} u^T \hat{G}u + u^T A^T v &= 0 \\ u^T \hat{G}u + (Au)^T v &= 0 \\ u^T \hat{G}u &= 0. \end{aligned}$$

Por hipótesis, $u^T \hat{G}u > 0$ para toda $u \in \text{Null}(A) \setminus \{0\}$, por lo que se concluye que $u = 0$. Al regresar a la ecuación (1.2.2a) y sustituir $u=0$, resulta que

$$A^T v = 0.$$

Como A es de rango completo, sus renglones son linealmente independientes

$$A = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_m \end{bmatrix}.$$

Luego,

$$\begin{aligned} A^T v &= \begin{bmatrix} a_1 & a_2 & \dots & a_m \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_m \end{bmatrix} \\ &= v_1 a_1 + v_2 a_2 + \dots + v_m a_m \\ &= 0. \end{aligned}$$

Como los vectores a_i son linealmente independientes, $v_i = 0$ para toda

$i \in \{1, 2, \dots, m\}$. Por lo tanto, $\omega = 0$. □

Teorema 1.2.2. Sean $Q \in \mathbb{R}^{n \times n}$, $A \in \mathbb{R}^{m \times n}$, $F \in \mathbb{R}^{p \times n}$ y $\mu, z \in \mathbb{R}^p$ tales que Q es simétrica positiva definida en \mathbb{R}^n , $\text{rango}(A)=m$, $\mu_i, z_i > 0$ para $i = 1, \dots, p$ con $U = \text{diag}(\mu)$ y $Z = \text{diag}(z)$, entonces la matriz

$$\mathcal{K} = \begin{pmatrix} Q & A^T & -F^T & 0 \\ A & 0 & 0 & 0 \\ -F & 0 & 0 & I_p \\ 0 & 0 & Z & U \end{pmatrix}$$

es no singular.

Demostración. El sistema lineal homogéneo de \mathcal{K} es

$$\mathcal{K}\omega = 0$$

i.e.

$$Q\omega_x + A^T\omega_\lambda - F^T\omega_\mu = 0, \tag{1.2.3a}$$

$$A\omega_x = 0, \tag{1.2.3b}$$

$$-F\omega_x + \omega_z = 0, \tag{1.2.3c}$$

$$Z\omega_\mu + U\omega_z = 0. \tag{1.2.3d}$$

Al despejar ω_z de la ecuación (1.2.3c), se obtiene

$$\omega_z = F\omega_x.$$

Por otro lado, al sustituir ω_z y despejar ω_μ en la ecuación (1.2.3d), se tiene que

$$\omega_\mu = -Z^{-1}UF\omega_x.$$

Al sustituir ω_μ en la ecuación (1.2.3a), se llega a

$$Q\omega_x + A^T\omega_\lambda + F^T Z^{-1} U F \omega_x = 0.$$

Resulta que el sistema a resolver se reduce a

$$\begin{aligned} (Q + F^T Z^{-1} U F)\omega_x + A^T\omega_\lambda &= 0, \\ A\omega_x &= 0. \end{aligned}$$

Sea $D = Z^{-1}U$, nótese que $Q + F^T D F$ es simétrica positiva definida. Sea $v \in \mathbb{R}^n \setminus \{0\}$, entonces

$$v^T(Q + F^T D F)v = v^T Q v + v^T F^T D F v.$$

Por hipótesis, Q es simétrica positiva definida y como $\mu_i, z_i > 0$, entonces D también es simétrica positiva definida. Por ende, $v^T Q v > 0$ y $v^T F^T D F v \geq 0$, lo que implica que $v^T(Q + F^T D F)v > 0$.

Por el Teorema 1.2.1, la matriz

$$\begin{pmatrix} Q + F^T Z^{-1} U F & A^T \\ A & 0 \end{pmatrix}$$

es no singular, por lo que $\omega_x = 0$ y $\omega_\lambda = 0$. Se sigue que $\omega_\mu = 0$ y $\omega_z = 0$. \square

La solución numérica de la ecuación (1.2.1) no usa factorizaciones sobre la matriz del sistema por dos razones: la dimensión es muy grande y se llenarían entradas que son cero.

Por otro lado, el sistema lineal se puede reducir. Al analizar el sistema de ecuaciones de (1.2.1),

$$Q\Delta x + A^T \Delta \lambda - F^T \Delta \mu = -r_x, \quad (1.2.4a)$$

$$A\Delta x = -r_\lambda, \quad (1.2.4b)$$

$$-F\Delta x + \Delta z = -r_\mu, \quad (1.2.4c)$$

$$Z\Delta\mu + U\Delta z = -r_z, \quad (1.2.4d)$$

y despejar la ecuación (1.2.4c), se obtiene

$$\Delta z = F\Delta x - r_\mu.$$

Al sustituir en la ecuación (1.2.4d), se tiene

$$\begin{aligned} Z\Delta\mu + U(F\Delta x - r_\mu) &= -r_z \\ Z\Delta\mu &= -r_z - U(F\Delta x - r_\mu) \\ \Delta\mu &= -Z^{-1}r_z - Z^{-1}U(F\Delta x - r_\mu) \\ \Delta\mu &= -Z^{-1}U(F\Delta x - r_\mu + U^{-1}r_z). \end{aligned}$$

Sea $\omega = -r_\mu + U^{-1}r_z$, entonces

$$\Delta\mu = -Z^{-1}U(F\Delta x + \omega). \quad (1.2.5)$$

Al sustituir (1.2.5) en (1.2.4a), se sigue que

$$\begin{aligned} Q\Delta x + A^T\Delta\lambda - F^T(-Z^{-1}U(F\Delta x + \omega)) &= -r_x \\ Q\Delta x + A^T\Delta\lambda + F^TZ^{-1}UF\Delta x + F^TZ^{-1}U\omega &= -r_x \\ (Q + F^TZ^{-1}UF)\Delta x + A^T\Delta\lambda &= -r_x - F^TZ^{-1}U\omega. \end{aligned}$$

Sea $\hat{G} = Q + F^T(Z^{-1}U)F$ y $\hat{d} = r_x + F^T(Z^{-1}U)\omega$, se llega al siguiente sistema de ecuaciones:

$$\begin{aligned} \hat{G}\Delta x + A^T\Delta\lambda &= -\hat{d}, \\ A\Delta x &= -r_\lambda. \end{aligned}$$

Finalmente, el sistema lineal (1.2.1) se reduce a:

$$\begin{pmatrix} \hat{G} & A^T \\ A & 0 \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta\lambda \end{pmatrix} = - \begin{pmatrix} \hat{d} \\ r_\lambda \end{pmatrix}, \quad (1.2.6)$$

donde

$$\begin{aligned}\hat{G} &= Q + F^T(Z^{-1}U)F, \\ \hat{d} &= r_x + F^T(Z^{-1}U)\omega, \\ \omega &= -r_\mu + U^{-1}r_z.\end{aligned}\tag{1.2.7}$$

Las incógnitas restantes se resuelven como

$$\begin{aligned}\Delta\mu &= -(Z^{-1}U)(F\Delta x + \omega), \\ \Delta z &= -U^{-1}(Z\Delta\mu + r_z).\end{aligned}$$

Como la matriz \hat{G} es simétrica y positiva definida y $\text{rango}(A) = m$, por el teorema (1.2.1), la matriz en (1.2.6) es simétrica y no singular.

También se observa que el sistema lineal (1.2.6) corresponde a las condiciones necesarias de primer orden del problema cuadrático

$$\begin{aligned}\text{Minimizar} \quad & \frac{1}{2}\Delta x^T \hat{G} \Delta x + \hat{d}^T \Delta x \\ \text{sujeto a} \quad & A\Delta x = -r_\lambda,\end{aligned}$$

donde $\Delta\lambda$ es el multiplicador de Lagrange asociado a la restricción de igualdad.

1.3 Recorte de paso

Una vez obtenida la solución del sistema lineal en (1.2.6), se determina $\alpha \in (0, 1]$ tal que $\mu_+ = \mu + \alpha\Delta\mu > 0$ y $z_+ = z + \alpha\Delta z > 0$.

Se definen para cada $i \in I$ los valores

$$\beta_i = \begin{cases} 1 & \text{si } \Delta\mu_i \geq 0 \\ -(\frac{\mu_i}{\Delta\mu_i}) & \text{si } \Delta\mu_i < 0 \end{cases}$$

y

$$\gamma_i = \begin{cases} 1 & \text{si } \Delta z_i \geq 0 \\ -(\frac{z_i}{\Delta z_i}) & \text{si } \Delta z_i < 0 \end{cases}$$

Sea $\hat{\alpha} = \min(\{\beta_i | i = 1, \dots, p\} \cup \{\gamma_i | i = 1, \dots, p\})$. Por último, se tomará

$\alpha = (0.995)^{\min(1, \hat{\alpha})}$ y se actualiza el punto

$$(x, \lambda, \mu, z)_+ = (x, \lambda, \mu, z) + \alpha(\Delta x, \Delta \lambda, \Delta \mu, \Delta z).$$

El valor 0.995 es para evitar la frontera $z_i = 0$ y $\mu_i = 0$.

1.4 Algoritmo de puntos interiores para programación cuadrática convexa

Algoritmo 1 Sistema Lineal Completo

Entradas: $Q \in \mathbb{R}^{n \times n}$ simétrica y positiva definida, $A \in \mathbb{R}^{m \times n}$ ($m \leq n$) con $\text{rango}(A) = m$, $F \in \mathbb{R}^{p \times n}$ ($p \leq n$), $d \in \mathbb{R}^n$, $b \in \mathbb{R}^m$, $c \in \mathbb{R}^p$, tol es el valor mínimo de las condiciones necesarias de KKT y $maxiter$ es el número máximo de iteraciones permitidas.

Salidas: $x^* \in \mathbb{R}^n$, $\lambda^* \in \mathbb{R}^m$ y $\mu^* \in \mathbb{R}^p$.

Pseudo código:

Establecer los valores iniciales $x_0 \in \mathbb{R}^n$, $\lambda_0 \in \mathbb{R}^m$, $\mu_0 \in \mathbb{R}^p$, $z_0 \in \mathbb{R}^p$ con $\mu_0 > 0$, $z_0 > 0$ (ver sección 1.5);

$k \leftarrow 0$;

mientras $\|H(x, \lambda, \mu, z)\| > tol$ & $k < maxiter$ **hacer**

Resolver el sistema lineal (1.2.1)

$$H'(x_k, \lambda_k, \mu_k, z_k)\Delta = -H_{\tau_k}(x_k, \lambda_k, \mu_k, z_k).$$

Determinar $\alpha_k > 0$ tal que

$$\begin{aligned}\mu_k + \alpha_k \Delta \mu &> 0, \\ z_k + \alpha_k \Delta z &> 0\end{aligned}$$

(ver sección 1.3).

Actualizar

$$\begin{aligned}x_{k+1} &\leftarrow x_k + \alpha_k \Delta x; \\ \lambda_{k+1} &\leftarrow \lambda_k + \alpha_k \Delta \lambda; \\ \mu_{k+1} &\leftarrow \mu_k + \alpha_k \Delta \mu; \\ z_{k+1} &\leftarrow z_k + \alpha_k \Delta z; \\ \tau_{k+1} &\leftarrow \frac{1}{2p} \mu_{k+1}^T z_{k+1}; \\ k &\leftarrow k + 1.\end{aligned}$$

fin mientras

Algoritmo 2 Sistema Lineal Reducido

Entradas: $Q \in \mathbb{R}^{n \times n}$ simétrica y positiva definida, $A \in \mathbb{R}^{m \times n}$ ($m \leq n$) con $\text{rango}(A) = m$, $F \in \mathbb{R}^{p \times n}$ ($p \leq n$), $d \in \mathbb{R}^n$, $b \in \mathbb{R}^m$, $c \in \mathbb{R}^p$, tol es el valor mínimo de las condiciones necesarias de KKT y $maxiter$ es el número máximo de iteraciones permitidas.

Salidas: $x^* \in \mathbb{R}^n$, $\lambda^* \in \mathbb{R}^m$ y $\mu^* \in \mathbb{R}^p$.

Pseudo código:

Establecer los valores iniciales $x_0 \in \mathbb{R}^n$, $\lambda_0 \in \mathbb{R}^m$, $\mu_0 \in \mathbb{R}^p$, $z_0 \in \mathbb{R}^p$ con $\mu_0 > 0$, $z_0 > 0$ (ver sección 1.5);

$k \leftarrow 0$;

mientras $\|H(x, \lambda, \mu, z)\| > tol$ & $k < maxiter$ **hacer**

Resolver el sistema lineal (1.2.6)

$$\begin{pmatrix} \hat{G} & A^T \\ A & 0 \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta \lambda \end{pmatrix} = - \begin{pmatrix} \hat{d} \\ r_\lambda \end{pmatrix}.$$

Calcular las incógnitas restantes

$$\begin{aligned} \Delta \mu &\leftarrow -(Z^{-1}U)(F\Delta x + \omega), \\ \Delta z &\leftarrow -U^{-1}(Z\Delta \mu + r_z). \end{aligned}$$

Determinar $\alpha_k > 0$ tal que

$$\begin{aligned} \mu_k + \alpha_k \Delta \mu &> 0, \\ z_k + \alpha_k \Delta z &> 0. \end{aligned}$$

(ver sección 1.3).

Actualizar

$$\begin{aligned} x_{k+1} &\leftarrow x_k + \alpha_k \Delta x; \\ \lambda_{k+1} &\leftarrow \lambda_k + \alpha_k \Delta \lambda; \\ \mu_{k+1} &\leftarrow \mu_k + \alpha_k \Delta \mu; \\ z_{k+1} &\leftarrow z_k + \alpha_k \Delta z; \\ \tau_{k+1} &\leftarrow \frac{1}{2p} \mu_{k+1}^T z_{k+1}; \\ k &\leftarrow k + 1. \end{aligned}$$

fin mientras

1.5 Prueba del algoritmo

Se implementó el método de puntos interiores del Sistema Lineal Reducido para el problema de programación cuadrática en (1.1.1). La implementación se realizó en Matlab y se ejecutó en una computadora con procesador Intel Core i7-1165G7 11va generación, 2.80 GHz, 4 núcleos, 8 procesadores lógicos y 16 GB de RAM..

Se escogió una tolerancia de 1×10^{-6} y un máximo de iteraciones de 100. Los puntos iniciales para el proceso iterativo son los siguientes: x_0 , μ_0 , y z_0 son vectores de unos y λ_0 es un vector de ceros. En el tabla 1.1 se muestran los resultados de correr el método para el problema *KB2* del repositorio Netlib. La columna uno corresponde al número de iteración k . La columna dos indica la norma infinita de las condiciones necesarias de KKT. La columna tres es el valor del parámetro τ . Por último, la columna cuatro muestra el valor del recíproco del condicional de la matriz \hat{G} del sistema lineal (1.2.6) en norma uno. Si el valor es cercano a cero la matriz está mal condicionada, lo cual puede indicar que es cercana a ser singular.

Se puede observar como el recíproco del condicional de la matriz \hat{G} a partir de la segunda iteración comienza a acercarse a cero. A pesar de no mantener una tendencia monótona decreciente, en cada iteración $\text{rcond}(\hat{G})$ toma valores por debajo de la norma de las condiciones de KKT y el valor de τ . El mal condicionamiento de la matriz \hat{G} se debe a las condiciones de complementaridad de (1.1.2). Conforme el algoritmo de puntos interiores se acerca a la solución, μ_i o z_i para toda $i \in I = \{1, \dots, p\}$ se acercan al cero como se muestra en los valores de τ en cada iteración. Valores de $\text{rcond}(\hat{G})$ por debajo del epsilon de la máquina generan una alerta en Matlab sobre posibles errores numéricos al resolver el sistema lineal (1.2.6).

Matrices mal condicionadas pueden generar resultados numéricos erróneos [3]. Muchas técnicas de factorización han sido implementadas para solucionar este problema; sin embargo, para problemas de dimensiones muy grandes la factorización puede tomar mucho tiempo en encontrarse [4]. Un método iterati-

CAPÍTULO 1. MÉTODO DE PUNTOS INTERIORES PARA PROGRAMACIÓN CUADRÁTICA CONVEXA

vo para resolver sistemas de ecuaciones basado en subespacios de Krylov puede ser una buena opción para reducir el tiempo de máquina. No obstante, dado el mal condicionamiento de la matriz es necesario el uso de un preconditionador. En el presente trabajo se analizarán dos familias de preconditionadores para el método de puntos interiores para programación cuadrática convexa.

| Iter | CNPO | tau | rcond(\hat{G}) |
|------|---------------|-------------|--------------------|
| 1 | 489.313118825 | 1.215090291 | 1.000000000 |
| 2 | 77.659268812 | 0.531700292 | 0.005671610 |
| 3 | 36.065280618 | 0.348803027 | 0.005749882 |
| 4 | 21.115247771 | 0.260640934 | 0.001370411 |
| 5 | 10.100045073 | 0.173016722 | 0.000048193 |
| 6 | 6.698112448 | 0.138429588 | 0.000003600 |
| 7 | 1.652597085 | 0.066855643 | 0.000000721 |
| 8 | 0.325180289 | 0.034708325 | 0.000016764 |
| 9 | 0.060506200 | 0.019991067 | 0.000000343 |
| 10 | 0.013183529 | 0.010868284 | 0.000000053 |
| 11 | 0.006576723 | 0.006119429 | 0.000000742 |
| 12 | 0.003529267 | 0.003439191 | 0.000000628 |
| 13 | 0.001955982 | 0.001927606 | 0.000000616 |
| 14 | 0.001092248 | 0.001075820 | 0.000000633 |
| 15 | 0.000610183 | 0.000596484 | 0.000000664 |
| 16 | 0.000339210 | 0.000327413 | 0.000000704 |
| 17 | 0.000187005 | 0.000177174 | 0.000000739 |
| 18 | 0.000101677 | 0.000094226 | 0.000000737 |
| 19 | 0.000054449 | 0.000049280 | 0.000000547 |
| 20 | 0.000028807 | 0.000025447 | 0.000000297 |
| 21 | 0.000015017 | 0.000013067 | 0.000000158 |
| 22 | 0.000007725 | 0.000006704 | 0.000000082 |
| 23 | 0.000003980 | 0.000003434 | 0.000000043 |
| 24 | 0.000002038 | 0.000001753 | 0.000000022 |
| 25 | 0.000001022 | 0.000000889 | 0.000000011 |
| 26 | 0.000000527 | 0.000000449 | 0.000000006 |

Tabla 1.1: Convergencia del método de puntos interiores para el problema KB2

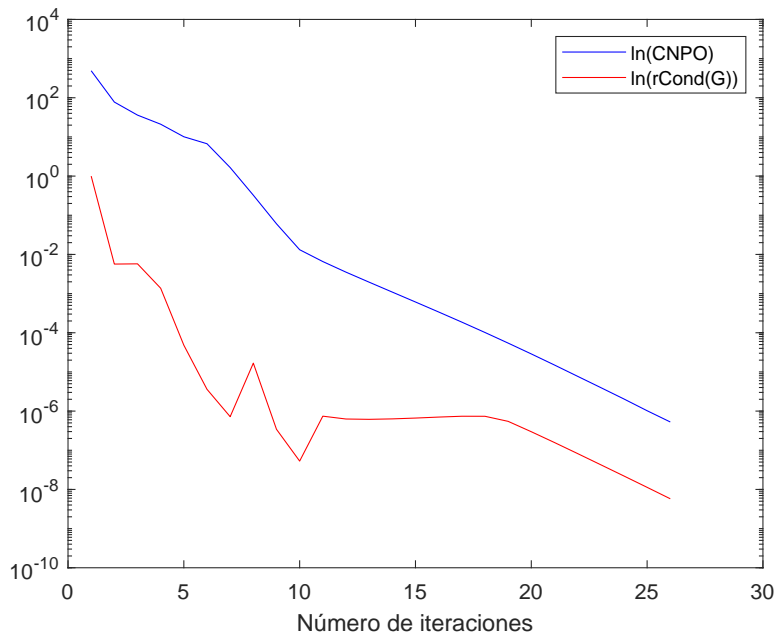


Figura 1.1: Comparativa de CNPO y $\text{rcond}(\hat{G})$ para el problema KB2

2 MINRES

MINRES (*Minimal Residual method*) es un método iterativo para resolver sistemas lineales de la forma $Ax = b$ con A simétrica indefinida. En este método se proyecta un problema m -dimensional a un subespacio de Krylov de menor dimensión. Se tiene que para una matriz $A \in \mathbb{R}^{m \times m}$ de rango completo y un vector $b \in \mathbb{R}^m$, la secuencia de Krylov asociada es el conjunto de vectores $b, Ab, A^2b, A^3b, \dots, A^{m-1}b$. Los subespacios de Krylov son generados por subconjuntos sucesivamente más grandes de estos vectores.

La ventaja de proyectar A en los subespacios de Krylov es que el problema lineal original es reducido a una secuencia de problemas lineales de dimensiones $n = 1, 2, 3, \dots, m$.

2.1 Proceso de Lanczos

El proceso de Lanczos es un algoritmo iterativo que transforma una matriz A simétrica y de rango completo a su forma Hessenberg. De manera analítica, una reducción completa de A a una forma Hessenberg a través de una transformación ortogonal similar Q , $Q^T Q = I = Q Q^T$, se escribe como $A = Q T Q^T$ o $AQ = QT$, con T una matriz Hessenberg:

$$T = \begin{pmatrix} t_{11} & t_{12} & \cdots & t_{1m} \\ t_{21} & t_{22} & & \\ & \ddots & \ddots & \vdots \\ & & t_{m,m-1} & t_{mm} \end{pmatrix}.$$

Como A es simétrica, se tiene que:

$$\begin{aligned} A &= A^T \\ \Rightarrow QTQ^T &= QT^TQ^T \\ \Rightarrow T &= T^T. \end{aligned}$$

Por otro lado, como T es una matriz de Hessenberg por definición $t_{ij} = 0$ para $i > j + 1$. Como $T = T^T$, entonces $t_{ij} = t_{ji}$ y $t_{ji} = 0$ para $j > i + 1$. Por lo tanto, T es una matriz tridiagonal de la forma:

$$\begin{pmatrix} \alpha_1 & \beta_1 & & & \\ \beta_1 & \alpha_2 & \beta_2 & & \\ & \beta_2 & \alpha_3 & \ddots & \\ & & \ddots & \ddots & \beta_{m-1} \\ & & & \beta_{m-1} & \alpha_m \end{pmatrix}.$$

Al utilizar métodos iterativos se considera que m es grande, por lo que calcular la factorización completa es demasiado costoso computacionalmente [4]. En su lugar, se utilizan las primeras n columnas de $AQ = QT$. Sea $Q_n \in \mathbb{R}^{m \times n}$ que contiene las primeras n columnas de Q :

$$Q_n = [q_1 | q_2 | \cdots | q_n].$$

Sea T_n la sección izquierda superior de T de dimensión $(n+1) \times n$, la cual también es una matriz Hessenberg:

$$T_n = \begin{pmatrix} \alpha_1 & \beta_1 & & & \\ \beta_1 & \alpha_2 & \beta_2 & & \\ & \beta_2 & \alpha_3 & \ddots & \\ & & \ddots & \ddots & \beta_{n-1} \\ & & & \beta_{n-1} & \alpha_n \\ & & & & \beta_n \end{pmatrix}.$$

Entonces se tiene que:

$$AQ_n = Q_{n+1}T_n, \quad (2.1.1)$$

esto es:

$$A \begin{bmatrix} q_1 | \dots | q_n \end{bmatrix} = \begin{bmatrix} q_1 | \dots | q_{n+1} \end{bmatrix} \begin{pmatrix} \alpha_1 & \beta_1 & & & \\ \beta_1 & \alpha_2 & \beta_2 & & \\ & \beta_2 & \alpha_3 & \ddots & \\ & & \ddots & \ddots & \beta_{n-1} \\ & & & \beta_{n-1} & \alpha_n \\ & & & & \beta_n \end{pmatrix}.$$

La n -ésima columna de esta ecuación puede escribirse de la siguiente manera:

$$Aq_n = \beta_{n-1}q_{n-1} + \alpha_n q_n + \beta_n q_{n+1}. \quad (2.1.2)$$

En otras palabras, el n -ésimo paso satisface una relación de recurrencia de tres términos. A continuación, se presenta un pseudo código del proceso de Lanczos:

Algoritmo 3 Proceso de Lanczos

```

 $\beta_0, q_0 = 0, b = \text{arbitrario}, q_1 = \frac{b}{\|b\|}$ 
for  $n=1,2,3,\dots$  hacer
     $v = Aq_n$ 
     $\alpha_n = q_n^T v$ 
     $v = v - \beta_{n-1}q_{n-1} - \alpha_n q_n$ 
     $\beta_n = \|v\|$ 
     $q_{n+1} = \frac{v}{\beta_n}$ 
fin for

```

La iteración de Lanczos es simplemente la iteración del algoritmo de Gram-Schmidt que implementa la relación (2.1.2). El poder del proceso de Lanczos recae en las múltiples aplicaciones que tiene y en los algoritmos que se derivan del mismo. Para el presente trabajo, la interpretación más importante se expone en el siguiente teorema:

Teorema 2.1.1. *Los vectores q_j generados por el proceso de Lanczos forman una base del subespacio de Krylov de dimensión n generado por A y b . En otras palabras,*

$$\mathcal{K}_n(A, b) = \text{span}\{b, Ab, \dots, A^{n-1}b\} = \text{span}\{q_1, q_2, \dots, q_n\} \subset \mathbb{R}^n.$$

Demostración. Sea $x \in \text{span}\{q_1, \dots, q_n\}$, se quiere demostrar que $x \in \mathcal{K}_n(A, b)$. Por el Algoritmo 3, $q_1 = \frac{b}{\|b\|}$, por lo que $q_1 \in \mathcal{K}_1(A, b)$. Ahora, de la ecuación (2.1.2) se puede despejar q_{n+1} y se obtiene que:

$$q_{n+1} = \frac{1}{\beta_n}(Aq_n + \beta_{n-1}q_{n-1} - \alpha_n q_n).$$

Sea $n=1$

$$\begin{aligned}
 q_2 &= \frac{1}{\beta_1}(Aq_1 - \beta_0 q_0 - \alpha_1 q_1) \\
 &= \frac{1}{\beta_1}\left(A\frac{b}{\|b\|} - \alpha_1 \frac{b}{\|b\|}\right) \\
 &= \frac{1}{\beta_1\|b\|}Ab - \frac{\alpha_1}{\beta_1\|b\|}b.
 \end{aligned}$$

Se sigue que $q_2 \in \mathcal{K}_2(A, b)$. Supongamos que $q_n \in \mathcal{K}_n(A, b)$ para $n \in \mathbb{N}$. Queremos demostrar que $q_{n+1} \in \mathcal{K}_{n+1}(A, b)$.

Por la ecuación (2.1.2) se tiene que:

$$q_{n+1} = \frac{1}{\beta_n}(Aq_n - \beta_{n-1}q_{n-1} - \alpha_n q_n)$$

Por hipótesis, $q_n \in \mathcal{K}_n(A, b)$ y $q_{n-1} \in \mathcal{K}_{n-1}(A, b)$, luego

$$\begin{aligned} q_{n+1} &= \frac{1}{\beta_n}(A(\sigma_1 b + \dots + \sigma_n A^{n-1} b) - \beta_{n-1}(\gamma_1 b + \dots + \gamma_{n-1} A^{n-2} b) \\ &\quad - \alpha_n(\sigma_1 b + \dots + \sigma_n A^{n-1} b)) \\ &= \frac{1}{\beta_n}(\sigma_1 A b + \dots + \sigma_n A^n b - \beta_{n-1}\gamma_1 b - \dots - \beta_{n-1}\gamma_{n-1} A^{n-2} b \\ &\quad - \alpha_n \sigma_1 b - \dots - \alpha_n \sigma_n A^{n-1} b) \\ &= \frac{1}{\beta_n}((- \beta_{n-1}\gamma_1 - \alpha_n \sigma_1)b + (\sigma_1 - \beta_{n-1}\gamma_2 - \alpha_n \sigma_2)Ab + \dots + \\ &\quad (\sigma_{n-2} - \beta_{n-1}\gamma_{n-1} - \alpha_n \sigma_{n-1})A^{n-2}b + (\sigma_{n-1} - \alpha_n \sigma_n)A^{n-1}b \\ &\quad + \sigma_n A^n b), \end{aligned}$$

con $\sigma_i, \gamma_j \in \mathbb{R}$ para $i \in \{1, \dots, n\}$ y $j \in \{1, \dots, n-1\}$. Por lo tanto, $q_{n+1} \in \mathcal{K}_{n+1}(A, b)$ para toda $n \in \mathbb{N}$. Se sigue que:

$$\text{span}\{q_1, \dots, q_n\} \subset \mathcal{K}_n(A, b).$$

Ahora, sea $x \in \mathcal{K}_n(A, b)$, se quiere demostrar que $x \in \text{span}\{q_1, \dots, q_n\}$. Por el Algoritmo 3, sabemos que $q_1 = \frac{b}{\|b\|}$. Despejando b se tiene que $b = \|b\|q_1$ que pertenece al $\text{span}\{q_1\}$. Sabemos que:

$$\begin{aligned} q_2 &= \frac{1}{\beta_1}(Aq_1 - \beta_0 q_0 - \alpha_1 q_1) \\ &= \frac{1}{\beta_1}(A \frac{b}{\|b\|} - \alpha_1 q_1) \\ &= \frac{1}{\beta_1 \|b\|} Ab - \frac{\alpha_1}{\beta_1} q_1 \\ \implies Ab &= \beta_1 \|b\| q_2 + \alpha_1 \|b\| q_1 \in \text{span}\{q_1, q_2\} \end{aligned}$$

Supongamos que $A^{n-1}b \in \text{span}\{q_1, \dots, q_n\}$ para $n \in \mathbb{N}$. Queremos demos-

trar que $A^n b \in \text{span}\{q_1, \dots, q_{n+1}\}$.

$$\begin{aligned}
 A^n b &= A(A^{n-1}b) \\
 &= A(\sigma_1 q_1 + \dots + \sigma_n q_n) \\
 &= \sigma_1 A q_1 + \dots + \sigma_n A q_n \\
 &= \sigma_1 (\beta_0 q_0 + \alpha_1 q_1 + \beta_1 q_2) + \dots + \sigma_n (\beta_{n-1} q_{n-1} + \alpha_n q_n + \beta_n q_{n+1}).
 \end{aligned}$$

Por lo tanto, $A^n b \in \text{span}\{q_1, \dots, q_{n+1}\}$ para toda $n \in \mathbb{N}$. Se sigue que

$$\mathcal{K}_n(A, b) \subset \text{span}\{q_1, \dots, q_{n+1}\}$$

.

□

Por el teorema (2.1.1), el proceso de Lanczos puede verse como una construcción sistemática de bases ortonormales para sucesivos subespacios de Krylov. Para expresar esta observación de forma matricial, definimos la *matriz de Krylov*, K_n , como la matriz de dimensiones $m \times n$

$$K_n = \begin{bmatrix} b | Ab | \dots | A^{n-1}b \end{bmatrix}. \quad (2.1.3)$$

Como las columnas de K_n son linealmente independientes, tiene una factorización QR:

$$K_n = Q_n R_n, \quad (2.1.4)$$

donde Q_n es la misma matriz que en la ecuación (2.1.1), ya que tanto en la ecuación (2.1.4) y (2.1.1) se genera Q_n con el algoritmo de Gram-Schmidt.

2.2 Minimización del residual en el subespacio de Krylov de dimensión n

Sea $x_* = A^{-1}b$, la solución exacta del sistema lineal. La idea de MINRES es la siguiente: en el paso n , se debe aproximar x_* con un vector $x_n \in \mathcal{K}_n(A, b)$

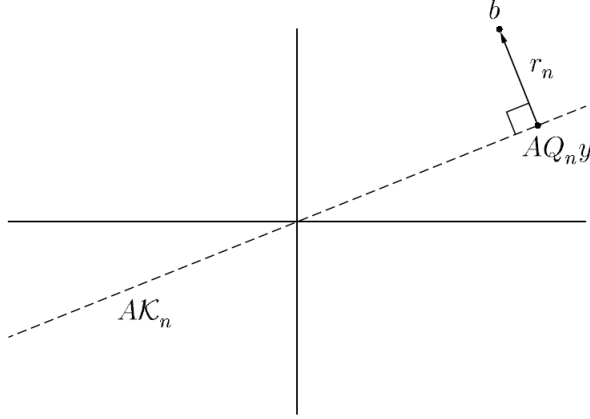


Figura 2.1: Problema de mínimos cuadrados del método MINRES [4]

que minimice la norma del residual $r_n = b - Ax_n$. En otras palabras, se debe encontrar x_n a través de un problema de mínimos cuadrados, el cual se ilustra en la figura 2.1.

Sea K_n la matriz definida en (2.1.3), se sigue que:

$$AK_n = [Ab | A^2b | \dots | A^n b]. \quad (2.2.1)$$

El espacio generado por las columnas de esta matriz es $AK_n(A, b)$. Entonces el problema se reduce a encontrar un vector $c \in \mathbb{R}^n$ tal que:

$$\arg\min_{c \in \mathbb{R}^n} \|AK_n c - b\|_2^2. \quad (2.2.2)$$

Para resolver este problema, el procedimiento numéricamente estable utiliza el proceso de Lanczos [4]. Por medio de este proceso, se construye una secuencia de matrices de Krylov Q_n cuyas columnas q_1, q_2, \dots generan los subespacios sucesivos de Krylov $\mathcal{K}_n(A, b)$. Así, se define $x_n = Q_n y$ en lugar de $x_n = K_n c$. En sustitución al problema de minimización (2.2.2), el nuevo problema de mínimos cuadrados consiste en encontrar un vector $y \in \mathbb{R}^n$ tal

que:

$$\operatorname{argmín}_{y \in \mathbb{R}^n} \|AQ_n y - b\|_2^2. \quad (2.2.3)$$

El problema (2.2.3) es de dimensión $m \times n$. Sin embargo, por la estructura de los subespacios de Krylov, esencialmente es de dimensión $(n + 1) \times n$. A continuación, se dará la demostración de la afirmación anterior: al aplicar (2.1.1), se transforma el problema de minimización (2.2.3) en

$$\operatorname{argmín}_{y \in \mathbb{R}^n} \|Q_{n+1} T_n y - b\|_2^2.$$

Ahora, ambos vectores dentro de la norma están en el espacio columna de Q_{n+1} . Entonces, al multiplicar por la izquierda con Q_{n+1}^T la norma no cambia. En consecuencia, el problema es equivalente a:

$$\operatorname{argmín}_{y \in \mathbb{R}^n} \|T_n y - Q_{n+1}^T b\|_2^2.$$

Finalmente, por construcción de las matrices de Krylov $\{Q_n\}$, $Q_{n+1}^T b$ es igual a $\|b\|e_1$, donde $e_1 = (1, 0, 0, \dots)^T$. La forma final de problema de mínimos cuadrados de MINRES es la siguiente:

$$\operatorname{argmín}_{y \in \mathbb{R}^n} \|T_n y - \|b\|e_1\|_2^2.$$

En el paso n de MINRES se resuelve este problema para y y luego se calcula $x_n = Q_n y$. A continuación, se presenta un pseudo código del algoritmo de MINRES:

Algoritmo 4 MINRES

```

 $q1 = \frac{b}{\|b\|_2}$ 
mientras  $\|r_n\|_2 < tol$  hacer
    ⟨Paso  $n$  del proceso de Arnoldi, Algoritmo 3⟩
    Encontrar  $y$  que minimize  $\|T_n y - \|b\|e_1\|_2 = \|r_n\|_2$ 
     $x_n = Q_n y$ .
fin mientras

```

2.3 Aproximación polinomial de MINRES

Sea $x \in \mathcal{K}_n(A, b)$,

$$x = \sigma_0 b + \sigma_1 A b + \cdots + \sigma_{n-1} A^{n-1} b. \quad (2.3.1)$$

Esta expresión tiene una descripción compacta: es un polinomio en A multiplicado por b . Esto es, si q es un polinomio $q(z) = \sigma_0 + \sigma_1 z + \cdots + \sigma_{n-1} z^{n-1}$, entonces se tiene que:

$$x = q(A)b. \quad (2.3.2)$$

Como cualquier vector de la forma (2.3.1) puede representarse de la forma (2.3.2), los métodos basados en subespacios de Krylov pueden analizarse en términos de matrices de polinomios.

Sea $P_n = \{\text{polinomios } p \text{ de grado } \leq n \text{ con } p(0) = 1\}$. El iterando x_n del método de MINRES puede escribirse como

$$x_n = q_n(A)b,$$

donde q_n es un polinomio de grado $n - 1$ cuyos coeficientes son las entradas del vector c del problema de minimización (2.2.2). Por lo tanto, tenemos que el residual correspondiente

$$r_n = b - Ax_n = (I - Aq_n(A))b.$$

Sea $p_n(z) = 1 - zq_n(z)$,

$$r_n = p_n(A)b,$$

para algún polinomio $p_n \in P_n$. El algoritmo de MINRES encuentra los coeficientes de p_n que minimizan la norma del residual. Por ende, MINRES resuelve el siguiente problema de aproximación sucesivamente para $n = 1, 2, 3, \dots$

MINRES Problema de Aproximación.

$$\min_{p_n \in P_n} \|p_n(A)b\|_2.$$

2.4 Convergencia de MINRES

MINRES converge de manera monótona, es decir,

$$\|r_{n+1}\|_2 \leq \|r_n\|_2.$$

La razón es que $\|r_n\|$ es el valor más pequeño para el subespacio $\mathcal{K}_n(A, b)$. Por ende, al expandir $\mathcal{K}_n(A, b)$ a $\mathcal{K}_{n+1}(A, b)$ se tienen dos casos: la norma se reduce o no cambia.

En ausencia de errores de redondeo, el método de MINRES converge en a lo más m pasos:

$$\|r_m\|_2 = 0.$$

Para datos genéricos A y b , esto pasa porque $\mathcal{K}_m(A, b) = \mathbb{R}^m$, y en casos aislados, si x_* llega a estar en $\mathcal{K}_n(A, b)$ para alguna $n < m$, la convergencia ocurrirá antes. El resultado anterior es poco práctico, ya que se busca que MINRES converja a la precisión deseada en $n \ll m$ pasos.

Sabemos que $\|r_n\|_2 = \|p_n(A)b\|_2 \leq \|p_n(A)\|_2 \|b\|_2$ es mínimo. Con excepción de problemas donde el lado derecho de la ecuación b tiene una estructura particular relacionada con la matriz A , el factor crítico que determina el tamaño de residual es $\|p_n(A)\|_2$. Esto es, lo que determina la convergencia de MINRES es generalmente la desigualdad:

$$\frac{\|r_n\|_2}{\|b\|_2} \leq \inf_{p_n \in P_n} \|p_n(A)\|_2. \quad (2.4.1)$$

Sean p un polinomio y S un subconjunto de los números reales, definimos

el escalar $\|p\|_S$ como:

$$\|p\|_S = \sup_{z \in S} |p(z)|. \quad (2.4.2)$$

Como A es simétrica, por el teorema espectral es diagonalizable, $A = V\Lambda V^{-1}$ para una matriz V no singular y Λ una matriz diagonal. Entonces se tiene que:

$$\|p(A)\|_2 \leq \|V\|_2 \|p(\Lambda)\|_2 \|V^{-1}\|_2 = \text{cond}(V) \|p\|_{\Lambda(A)},$$

$$\text{con } \text{cond}(V) = \|V\|_2 \|V^{-1}\|_2.$$

Al combinar este resultado con (2.4.1) nos lleva al siguiente teorema:

Teorema 2.4.1. *En el paso n del método de MINRES, el residual r_n satisface que:*

$$\frac{\|r_n\|_2}{\|b\|_2} \leq \inf_{p_n \in P_n} \|p_n(A)\|_2 \leq \text{cond}(V) \inf_{p_n \in P_n} \|p_n\|_{\Lambda(A)},$$

donde $\Lambda(A)$ es el conjunto de valores propios de A , V es una matriz no singular de vectores propios y $\|p_n\|_{\Lambda(A)}$ está definida por (2.4.2).

Este teorema puede resumirse de la siguiente manera. Si A no está muy lejos de ser normal en el sentido que $\text{cond}(V)$ no sea muy grande, y si existe un polinomio en P_n cuyo tamaño en el espectro de $\Lambda(A)$ decrezca rápidamente con n , entonces MINRES converge rápidamente.

3 Precondicionadores

Un preconditionador \mathcal{P} es una transformación que condiciona un problema dado, por ejemplo, $Ax = b$ en una forma que sea más adecuada para resolver por métodos numéricos. Normalmente, se utiliza para reducir el condicional de una matriz.

3.1 Normas matriciales

Definición. Decimos que $\|\cdot\|$ es una norma en $\mathbb{R}^{m \times n}$ si y solo si

1. $\|A\| \geq 0 \forall A \in \mathbb{R}^{m \times n}$
2. $\|\alpha A\| = |\alpha| \|A\| \forall A \in \mathbb{R}^{m \times n}, \forall \alpha \in \mathbb{R}$
3. $\|A + B\| \leq \|A\| + \|B\| \forall A, B \in \mathbb{R}^{m \times n}$

Definición. Decimos que la norma matricial $\|\cdot\|$ en $\mathbb{R}^{m \times n}$ es consistente si y solo si

$$\|AB\| \leq \|A\| \|B\|, \forall A \in \mathbb{R}^{m \times p} \forall B \in \mathbb{R}^{p \times n}.$$

Normas inducidas

Sea $\|\cdot\|_{\bigcirc}$ una norma vectorial en \mathbb{R}^n y $\|\cdot\|_{\square}$ una norma vectorial en \mathbb{R}^m . Definimos la norma inducida correspondiente en el espacio $\mathbb{R}^{m \times n}$ de la siguiente manera:

$$\|A\|_{\square \bigcirc} = \max\{\|Ax\|_{\square} \mid \|x\|_{\bigcirc} = 1\}.$$

Si $\square = \bigcirc$, la matriz identidad I_n cumple que

$$\|I_n\|_{\bigcirc} = \max\{\|I_n x\|_{\bigcirc} \mid \|x\|_{\bigcirc} = 1\} = 1.$$

Para el presente trabajo se utilizarán normas matriciales que cumplan la condición que $\|I_n\| = 1$.

3.2 El condicional de una matriz

Sea A una matriz cuadrada invertible, el condicional de A se define como

$$\text{cond}(A) = \|A\| \|A^{-1}\|. \quad (3.2.1)$$

Por convención, $\text{cond}(A) = \infty$, si A es singular.

El condicional de una matriz mide la sensibilidad de la solución de un sistema de ecuaciones lineales ante perturbaciones en los datos de entrada (en la matriz del sistema A o en el lado derecho b) [4]. Si pequeños cambios en los datos de entrada generan grandes cambios en la solución, se tiene un sistema mal condicionado. Por ende, sistemas mal condicionados son más sensibles a errores de redondeo.

El valor del condicional de una matriz depende de la norma utilizada y puede, para algunas matrices, variar considerablemente entre normas. Para el presente trabajo se considerarán normas matriciales consistentes. Para el conjunto de normas consistentes, se cumple que el condicional de una matriz en el espacio $\mathbb{R}^{n \times n}$ es al menos 1. Lo anterior se debe a que la norma de la matriz identidad, $\|I_n\|$, es igual a 1 y las normas consistentes cumplen que

$$1 = \|I_n\| = \|AA^{-1}\| \leq \|A\| \|A^{-1}\| = \text{cond}(A).$$

Lema 3.2.1. *Sea $A \in \mathbb{R}^{n \times n}$ una matriz no singular y $\|\cdot\|$ una norma consis-*

tente, entonces

$$\frac{1}{\text{cond}(A)} \leq \inf \left\{ \frac{\|A - B\|}{\|A\|} \mid B \in \mathcal{S} \right\}$$

con $\mathcal{S} = \{B \in \mathbb{R}^{n \times n} \mid B \text{ es singular}\}$.

Demostración. Sea $B \in \mathcal{S}$, entonces existe un vector $x \in \mathbb{R}^n$, $x \neq 0$, tal que $Bx = 0$. Resulta que

$$\begin{aligned} \|Ax\| &= \|Ax - 0\| \\ &= \|Ax - Bx\| \\ &= \|(A - B)x\| \\ &\leq \|A - B\| \|x\|, \end{aligned}$$

para cualquier matriz B .

Por otro lado,

$$\begin{aligned} \|x\| &= \|A^{-1}Ax\| \\ &\leq \|A^{-1}\| \|Ax\|, \end{aligned}$$

lo que implica que

$$\begin{aligned} \frac{\|x\|}{\|A^{-1}\|} &\leq \|Ax\| \\ &\leq \|A - B\| \|x\|. \end{aligned}$$

Como $x \neq 0$, se obtiene que

$$\frac{1}{\|A^{-1}\|} \leq \|A - B\|,$$

por lo tanto,

$$\frac{1}{\text{cond}(A)} = \frac{1}{\|A^{-1}\| \|A\|} \leq \frac{\|A - B\|}{\|A\|}.$$

Como el conjunto $\left\{ \frac{\|A - B\|}{\|A\|} \mid B \in \mathcal{S} \right\}$ está acotado inferiormente, entonces tiene ínfimo. Por lo anterior,

$$\frac{1}{\text{cond}(A)} \leq \inf \left\{ \frac{\|A - B\|}{\|A\|} \mid B \in \mathcal{S} \right\}.$$

□

Lema 3.2.2. *El conjunto $\mathcal{S} = \{B \in \mathbb{R}^{n \times n} | B \text{ es singular}\}$ es cerrado para cualquier norma matricial.*

Demostración. De hecho no necesitamos explícitamente la norma matricial. Sea $\{B_k\}$ una secuencia de matrices singulares tales que

$$\lim_{k \rightarrow \infty} B_k = B^*.$$

Como el determinante de una matriz cuadrada es una función continua de las entradas de la matriz, tenemos que

$$\begin{aligned} \lim_{k \rightarrow \infty} \det(B_k) &= \det(\lim_{k \rightarrow \infty} B_k) \\ \Rightarrow 0 &= \det(B^*). \end{aligned}$$

Por lo tanto, B^* es singular y \mathcal{S} es un conjunto cerrado. □

De los lemas 3.2.1 y 3.2.2 se sigue lo siguiente:

Teorema 3.2.3. *Sea $A \in \mathbb{R}^{n \times n}$ una matriz no singular y $\|\cdot\|$ una norma consistente, entonces*

$$\frac{1}{\text{cond}(A)} \leq \min \left\{ \frac{\|A - B\|}{\|A\|} \mid B \in \mathcal{S} \right\}$$

con $\mathcal{S} = \{B \in \mathbb{R}^{n \times n} | B \text{ es singular}\}$.

Más aún, se puede demostrar que

$$\frac{1}{\text{cond}(A)} = \min \left\{ \frac{\|A - B\|}{\|A\|} \mid B \in \mathcal{S} \right\}.$$

La demostración de la igualdad se puede encontrar en [5]. Dicha demostración da pie al siguiente teorema:

Teorema 3.2.4. *Sea $A \in \mathbb{R}^{n \times n}$ una matriz invertible y $\|\cdot\|$ una norma matricial consistente, el condicional de la matriz A indica la distancia relativa de A a la matriz singular más cercana en $\mathbb{R}^{n \times n}$. Específicamente,*

$$\frac{1}{\text{cond}(A)} = \min \left\{ \frac{\|A - B\|}{\|A\|} \mid B \in \mathcal{S} \right\}.$$

con $\mathcal{S} = \{B \in \mathbb{R}^{n \times n} \mid B \text{ es singular}\}$.

Una matriz mal condicionada se puede aproximar muy bien por una matriz singular. Sin embargo, el mal condicionamiento no necesariamente se relaciona con singularidad. Una matriz singular es mal condicionada pero una matriz mal condicionada no necesariamente es singular, por ejemplo, las matrices de Hilbert.

Desde un punto de vista numérico, el costo computacional de calcular (3.2.1) es muy alto. En la práctica se estima de una manera que no es costosa computacionalmente.

En el presente trabajo, se utilizará la función $rcond()$ de Matlab que estima el recíproco del condicional de una matriz en norma-1 por medio del estimador de la librería LAPACK [6]. Si una matriz A está lejos de ser singular, $rcond(A)$ es cercano a uno. Por el contrario, si $rcond(A)$ es cercano a cero, la matriz está cerca del conjunto de matrices singulares.

3.3 Elección de un preconditionador para el método de puntos interiores para programación cuadrática

No existe una estrategia de preconditionamiento ideal para todo tipo de matrices [7]. Generalmente lo que se busca es un balance entre varios objetivos -efectividad de \mathcal{P} , un costo bajo de procesamiento y almacenamiento \mathcal{P} , solucionar el sistema lineal $\mathcal{P}y = r$ de una manera computacionalmente barata- que varían de problema a problema.

En la literatura se pueden encontrar buenas estrategias de preconditiona-

miento que han sido propuestas para tipos específicos de matrices. Usualmente, un preconditionador se define de tal manera que el sistema, $\mathcal{P}y = r$, sea una versión simplificada del problema original $Ax = b$. Como en muchas otras áreas de optimización y análisis numérico, el conocimiento de la estructura y origen del problema es clave para encontrar técnicas efectivas para resolver el problema. Con esto en mente, se analizará la matriz

$$\mathcal{A} = \begin{pmatrix} \hat{G} & A^T \\ A & 0 \end{pmatrix} \quad (3.3.1)$$

que representa la matriz de la ecuación (1.2.6), la cual nos puede generar resultados numéricos erróneos en el método de puntos interiores.

Para este capítulo se supondrá que $\hat{G} \in \mathbb{R}^{n \times n}$ es simétrica y positiva semi-definida con nulidad p y que $A \in \mathbb{R}^{m \times n}$ es de rango completo. Nótese que si \mathcal{A} es no singular, $\text{Null}(A) \cap \text{Null}(\hat{G}) = \{0\}$.

Lema 3.3.1. *Si \mathcal{A} es no singular, entonces $\text{Null}(A) \cap \text{Null}(\hat{G}) = \{0\}$.*

Demostración. Supongamos que $\text{Null}(A) \cap \text{Null}(\hat{G}) \neq \{0\}$.

Sea $x \in \text{Null}(A) \cap \text{Null}(\hat{G})$ con $x \neq 0$. El sistema lineal homogéneo asociado a \mathcal{A} es de la forma

$$\begin{pmatrix} \hat{G} & A^T \\ A & 0 \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}.$$

Se elijen $u = x$ y $v = 0$, $(x, 0)^T \neq (0, 0)$. Resulta que

$$\begin{aligned} \begin{pmatrix} \hat{G} & A^T \\ A & 0 \end{pmatrix} \begin{pmatrix} x \\ 0 \end{pmatrix} &= \begin{pmatrix} \hat{G}x \\ Ax \end{pmatrix} \\ &= \begin{pmatrix} 0 \\ 0 \end{pmatrix}. \end{aligned}$$

Lo que es una contradicción porque \mathcal{A} es no singular. Por lo tanto,

$$\text{Null}(A) \cap \text{Null}(\hat{G}) = \{0\}.$$

□

El resultado anterior se utilizará en el siguiente análisis.

3.3.1. Familia de preconditionadores triangulares por bloques

Sea el preconditionador

$$\mathcal{M} = \begin{pmatrix} \hat{G} + A^T W^{-1} A & k A^T \\ 0 & W \end{pmatrix},$$

donde k es un escalar y $W \in \mathbb{R}^{m \times m}$ una matriz de pesos simétrica positiva definida. Para encontrar los valores propios de la matriz preconditionada $\mathcal{M}^{-1} \mathcal{A}$, se resuelve el problema

$$\mathcal{M}^{-1} \mathcal{A} \hat{x} = \lambda \hat{x},$$

con $\hat{x} = (x, y)^T$ y $\lambda \neq 0$. Por facilidad, se analizará el problema generalizado de valores propios

$$\mathcal{A} \hat{x} = \lambda \mathcal{M} \hat{x}, \tag{3.3.2}$$

en lugar del problema original. El problema generalizado de valores propios está dado por:

$$\begin{pmatrix} \hat{G} & A^T \\ A & 0 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \lambda \begin{pmatrix} \hat{G} + A^T W^{-1} A & k A^T \\ 0 & W \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

i.e.

$$\hat{G}x + A^T y = \lambda \hat{G}x + \lambda A^T W^{-1} Ax + \lambda k A^T y, \quad (3.3.3a)$$

$$Ax = \lambda W y. \quad (3.3.3b)$$

De (3.3.3b) se obtiene

$$y = \frac{1}{\lambda} W^{-1} Ax.$$

Al sustituir en (3.3.3a), se tiene que

$$\hat{G}x + A^T \left(\frac{1}{\lambda} W^{-1} Ax \right) = \lambda \hat{G}x + \lambda A^T W^{-1} Ax + \lambda k A^T \left(\frac{1}{\lambda} W^{-1} Ax \right).$$

Al multiplicar por λ , se sigue que

$$\lambda \hat{G}x + A^T W^{-1} Ax = \lambda^2 \hat{G}x + \lambda^2 A^T W^{-1} Ax + k \lambda A^T W^{-1} Ax, .$$

Al agrupar términos, se obtiene la siguiente ecuación:

$$\lambda(\lambda - 1)\hat{G}x + (\lambda^2 + k\lambda - 1)A^T W^{-1} Ax = 0. \quad (3.3.4)$$

Sea $\lambda = 1$, al sustituir en (3.3.4) obtenemos:

$$\begin{aligned} (1^2 + k - 1)A^T W^{-1} Ax &= 0 \\ \iff k A^T W^{-1} Ax &= 0. \end{aligned}$$

Como el espacio nulo de $A^T W^{-1} A$ es de dimensión $n - m$, sin importar la elección de k , $\lambda = 1$ es un valor propio de (3.3.2) con multiplicidad algebraica $n - m$.

Por otro lado, por hipótesis hay p vectores linealmente independientes que pertenecen al espacio nulo de \hat{G} . Al analizar (3.3.4), para cada uno de estos vectores se pueden encontrar dos valores propios λ que satisfacen $\lambda^2 + k\lambda - 1 =$

0. Resulta que

$$\lambda_{\pm} = \frac{-k \pm \sqrt{k^2 + 4}}{2},$$

cada uno con multiplicidad algebraica p .

Los $2(m - p)$ valores propios restantes se pueden calcular con la siguiente ecuación:

$$-\frac{\lambda^2 - \lambda}{\lambda^2 + k\lambda - 1} \hat{G}x = A^T W^{-1} Ax. \quad (3.3.5)$$

En consecuencia, para $x \in \mathbb{R}^n - (Null(\hat{G}) \cup Null(A))$ y dado que \hat{G} es semi-definida positiva y W es positiva definida, tenemos

$$-\frac{\lambda^2 - \lambda}{\lambda^2 + k\lambda - 1} > 0. \quad (3.3.6)$$

Luego, se puede escribir $\frac{\lambda^2 - \lambda}{\lambda^2 + k\lambda - 1} = -\mu^2$ para alguna $\mu \in \mathbb{R}$, $\mu > 0$ y se puede reescribir la ecuación anterior de la siguiente manera:

$$(1 + \mu^2)\lambda^2 + (k\mu^2 - 1)\lambda - \mu^2 = 0.$$

Por ende,

$$\lambda_{\pm} = \frac{-(k\mu^2 - 1) \pm \sqrt{(k\mu^2 - 1)^2 + 4(1 + \mu^2)\mu^2}}{2(1 + \mu^2)}.$$

Esta expresión da una fórmula explícita en términos de los valores propios generalizados de la ecuación (3.3.5) y puede ser usada para identificar los intervalos en los que los valores propios se encuentran. Al examinar el caso $k = -1$, que corresponde a asignar al bloque (1,2) del preconditionador el valor $-A^T$. Por ende, se tiene que $\lambda = 1$ con multiplicidad $n - m$, y $\lambda_{\pm} = \frac{1 \pm \sqrt{5}}{2}$,

cada uno con multiplicidad p . Para los valores propios restantes se tiene

$$\begin{aligned}
 \lambda_{\pm} &= \frac{-(-\mu^2-1) \pm \sqrt{(-\mu^2-1)^2 + 4(1+\mu^2)\mu^2}}{2(1+\mu^2)} \\
 &= \frac{\frac{1}{(1+\mu^2)} \left[(\mu^2+1) \pm \sqrt{(\mu^2+1)^2 + 4(1+\mu^2)\mu^2} \right]}{\frac{1}{(1+\mu^2)} 2(1+\mu^2)} \\
 &= \frac{1 \pm \frac{1}{(1+\mu^2)} \sqrt{(\mu^2+1)^2 \left(1 + 4 \frac{\mu^2}{1+\mu^2}\right)}}{2} \\
 &= \frac{1 \pm \sqrt{1 + \frac{4\mu^2}{1+\mu^2}}}{2}.
 \end{aligned}$$

Como λ_+ es una función estrictamente creciente de μ en el intervalo $(0, \infty)$ (y λ_- es estrictamente decreciente), los intervalos que contienen los valores propios restantes pueden encontrarse al calcular $\lim_{\mu \rightarrow 0, \infty} \lambda_{\pm}(\mu)$. Al resolver el límite anterior, se obtiene que los valores propios faltantes caen dentro del intervalo $\left(\frac{1-\sqrt{5}}{2}, 0\right) \cup \left(1, \frac{1+\sqrt{5}}{2}\right)$. Hay que resaltar que como \hat{G} es singular, muchos de los valores propios generalizados serán grandes, en cuyo caso los correspondientes valores propios λ_{\pm} están lejos del cero. Por ejemplo, muchos de los valores propios negativos con $k = -1$ tenderán a $\frac{1-\sqrt{5}}{2}$. Como los $2p$ valores propios $\frac{k \pm \sqrt{k^2+4}}{2}$ no están acotados mientras k crece a infinito, se concluye que k debe tener un tamaño moderado.

Por otro lado, para examinar numéricamente los valores propios de la matriz $\mathcal{M}^{-1}\mathcal{A}$, se tomará $k = 0$ y $W = I$. En la figura 3.1, se muestran los valores propios ordenados de la matriz preconditionada del problema *AFIRO* del repositorio Netlib como un problema de programación cuadrática con restricciones (detalle en la sección 4). Se puede observar que $\lambda = 1$ tiene una multiplicidad alta y los valores propios restantes se encuentran en el intervalo $(-1, 0)$.

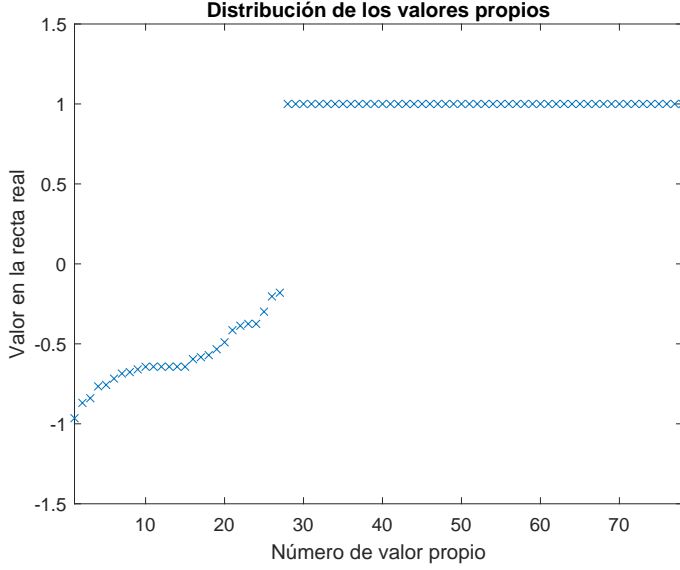


Figura 3.1: Valores propios de $\mathcal{M}^{-1}\mathcal{A}$ del problema cuadrático para *AFIRO* en la primera iteración del algoritmo 2

3.3.2. Familia de preconditionadores simétricos positivos definidos

Sea el preconditionador

$$\mathcal{N} = \begin{pmatrix} \hat{G} + (1 + \alpha^2)A^T H^{-1} A & \alpha A^T \\ \alpha A & H \end{pmatrix} \quad (3.3.7)$$

donde $H \in \mathbb{R}^{m \times m}$ es simétrica positiva definida y $\alpha \in \mathbb{R}$. Básicamente el parámetro α indicará si la matriz de restricciones lineales es tomada en cuenta en (3.3.7). Elecciones comunes serán $\alpha \in \{-1, 0, 1\}$. La elección de $\alpha = 0$ da como resultado una matriz diagonal por bloques. Mientras que $\alpha = 1$ considera la matriz de restricciones lineales originales. El objetivo principal de H dentro

de (3.3.7) es asegurarse que \mathcal{N} no sea singular pero su papel va más allá. De hecho, la matriz \mathcal{N} es positiva definida.

Lema 3.3.2. Sean $\hat{G} \in \mathbb{R}^{n \times n}$ semidefinida positiva y $A \in \mathbb{R}^{m \times n}$ de rango completo tal que $u^T \hat{G} u > 0$ para toda $u \in \{x \in \mathbb{R}^n \setminus \{0\} | Ax = 0\}$, la matriz $\hat{G} + A^T H^{-1} A$ es simétrica positiva definida en \mathbb{R}^n .

Demostración. Se supondrá que existe $x \in \mathbb{R}^n$ tal que $(\hat{G} + A^T H^{-1} A)x = 0$, luego $x^T (\hat{G} + A^T H^{-1} A)x = 0$.

Sea $x \in \text{Null}(A) \setminus \{0\}$, se tiene que $x^T \hat{G} x > 0$, lo que contradice nuestra hipótesis inicial.

Sea $x \notin \text{Null}(A)$, $x^T \hat{G} x + x^T A^T H^{-1} A x > 0$, ya que \hat{G} es semidefinida positiva, $Ax \neq 0$ y H es positiva definida, lo que nos lleva a otra contradicción. \square

Teorema 3.3.3. Sean $\hat{G} \in \mathbb{R}^{n \times n}$ semidefinida positiva y $A \in \mathbb{R}^{m \times n}$ de rango completo tal que $u^T \hat{G} u > 0$ para toda $u \in \{x \in \mathbb{R}^n \setminus \{0\} | Ax = 0\}$, la matriz \mathcal{N} es simétrica positiva definida en \mathbb{R}^{n+m} .

Demostración. Del lema 3.3.2 y el hecho que H es simétrica positiva definida, existe una factorización de Cholesky para las matrices

$$\hat{G} + A^T H^{-1} A = BB^T \text{ y } H = CC^T.$$

Nótese que $H^{-1} = C^{-T} C^{-1}$. Sea $L = \begin{pmatrix} B & \alpha A^T C^{-T} \\ 0 & C \end{pmatrix}$, se tiene que

$$\begin{aligned} LL^T &= \begin{pmatrix} BB^T + \alpha^2 A^T C^{-T} C^{-1} A & \alpha A^T C^{-T} C^T \\ \alpha C C^{-1} A & C C^T \end{pmatrix} \\ &= \begin{pmatrix} \hat{G} + A^T H^{-1} A + \alpha^2 A^T H^{-1} A & \alpha A^T \\ \alpha A & H \end{pmatrix} \\ &= \begin{pmatrix} \hat{G} + (1 + \alpha^2) A^T H^{-1} A & \alpha A^T \\ \alpha A & H \end{pmatrix} \\ &= \mathcal{N}. \end{aligned}$$

Por lo que \mathcal{N} tiene una factorización de Cholesky y por lo tanto, es simétrica positiva definida. \square

Análogamente a lo que se realizó en la sección 3.3.1, para encontrar los valores propios de la matriz preconditionada $\mathcal{N}^{-1}\mathcal{A}$ se analizará el problema generalizado de valores propios $\mathcal{A}\hat{x} = \lambda\mathcal{N}\hat{x}$, i.e.

$$\begin{pmatrix} \hat{G} & A^T \\ A & 0 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \lambda \begin{pmatrix} \hat{G} + (1 + \alpha^2)A^T H^{-1}A & \alpha A^T \\ \alpha A & H \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}. \quad (3.3.8)$$

De la segunda ecuación en (3.3.8) se obtiene que

$$Ax = \alpha\lambda Ax + \lambda Hy,$$

y al despejar y

$$y = \left(\frac{1 - \alpha\lambda}{\lambda} \right) H^{-1}Ax.$$

Por otro lado, al analizar la primera ecuación en (3.3.8)

$$\hat{G}x + A^T y = \lambda(\hat{G} + (1 + \alpha^2)A^T H^{-1}A)x + \lambda\alpha A^T y$$

$$(\lambda - 1)\hat{G}x + \lambda(1 + \alpha^2)A^T H^{-1}Ax + (\alpha\lambda - 1)A^T y = 0.$$

Al sustituir y y multiplicar por λ la ecuación anterior,

$$\lambda(\lambda - 1)\hat{G}x + \lambda^2(1 + \alpha^2)A^T H^{-1}Ax - (\alpha\lambda - 1)^2 A^T H^{-1}Ax = 0$$

$$\lambda(\lambda - 1)\hat{G}x + \lambda^2 A^T H^{-1}Ax + \lambda^2 \alpha^2 A^T H^{-1}Ax - (\alpha^2 \lambda^2 - 2\alpha\lambda + 1)A^T H^{-1}Ax = 0$$

$$\lambda(\lambda - 1)\hat{G}x + (\lambda^2 + 2\alpha\lambda - 1)A^T H^{-1}Ax = 0. \quad (3.3.9)$$

Nótese que la ecuación anterior es igual a (3.3.4), con $k = 2\alpha$, por lo que los valores propios de $\mathcal{M}^{-1}\mathcal{K}$ son los siguientes:

1. $\lambda_1 = 1$ de multiplicidad $(n - m)$
2. $\lambda_2 = -\alpha + \sqrt{\alpha^2 + 1}$ de multiplicidad p
3. $\lambda_3 = -\alpha - \sqrt{\alpha^2 + 1}$ de multiplicidad p
4. Los $2(m - p)$ valores propios restantes son reales, diferentes de cero y se encuentran al resolver

$$(\beta^2 + 1)\lambda^2 + (2\alpha\beta^2 - 1)\lambda - \beta^2 = 0$$

con $\beta \in \mathbb{R} \setminus \{0\}$.

Resulta que

$$\lambda_4 = \frac{-(2\alpha\beta^2 - 1) \pm \sqrt{(2\alpha\beta^2 - 1)^2 + 4(\beta^2 + 1)\beta^2}}{2(\beta^2 + 1)}$$

Esta expresión da una fórmula explícita y puede ser usada para identificar los intervalos en los que los valores propios se encuentran.

3.3.3. Precondicionador A

La elección de $k = 0$ para la familia de preconditionadores triangulares por bloques, genera una matriz diagonal por bloques simétrica positiva definida de la forma

$$\mathcal{M} = \begin{pmatrix} \hat{G} + A^T W^{-1} A & 0 \\ 0 & W \end{pmatrix}, \quad (3.3.10)$$

la cual se propone como preconditionador para el método de puntos interiores en [7]. Nótese que al escoger $\alpha = 0$ para la familia de preconditionadores simétricos positivos definidos, se obtiene el mismo preconditionador.

Sea $QR = A^T$ la factorización QR de A^T , donde $Q \in \mathbb{R}^{n \times m}$ y $R \in \mathbb{R}^{m \times m}$. Sea $Z \in \mathbb{R}^{n \times (n-m)}$ una base ortogonal del espacio nulo de A ($AZ=0$). Como $Z \cup Q$ forman una base ortogonal de \mathbb{R}^n , cualquier vector $x \in \mathbb{R}^n$ puede escribirse como $x = Zx_z + Qx_q$.

Se define la matriz de $(n + m) \times (n + m)$

$$P = \begin{pmatrix} Z & Q & 0 \\ 0 & 0 & I \end{pmatrix}. \quad (3.3.11)$$

Sea $v = (x_z, x_q, y)^T$ donde $Pv = (x, y)^T$, el problema generalizado de valores propios puede ser escrito como $P^T \mathcal{A}Pv = \lambda P^T \mathcal{M}Pv$. El lado izquierdo de la ecuación es igual a:

$$\begin{aligned} P^T \mathcal{A}Pv &= \begin{pmatrix} Z^T & 0 \\ Q^T & 0 \\ 0 & I \end{pmatrix} \begin{pmatrix} \hat{G} & A^T \\ A & 0 \end{pmatrix} \begin{pmatrix} Z & Q & 0 \\ 0 & 0 & I \end{pmatrix} \begin{pmatrix} x_z \\ x_q \\ y \end{pmatrix} \\ &= \begin{pmatrix} Z^T \hat{G} & Z^T A^T \\ Q^T \hat{G} & Q^T A^T \\ A & 0 \end{pmatrix} \begin{pmatrix} Z & Q & 0 \\ 0 & 0 & I \end{pmatrix} \begin{pmatrix} x_z \\ x_q \\ y \end{pmatrix} \\ &= \begin{pmatrix} Z^T \hat{G} & (AZ)^T \\ Q^T \hat{G} & Q^T A^T \\ A & 0 \end{pmatrix} \begin{pmatrix} Z & Q & 0 \\ 0 & 0 & I \end{pmatrix} \begin{pmatrix} x_z \\ x_q \\ y \end{pmatrix} \\ &= \begin{pmatrix} Z^T \hat{G} & 0 \\ Q^T \hat{G} & Q^T A^T \\ A & 0 \end{pmatrix} \begin{pmatrix} Z & Q & 0 \\ 0 & 0 & I \end{pmatrix} \begin{pmatrix} x_z \\ x_q \\ y \end{pmatrix} \\ &= \begin{pmatrix} Z^T \hat{G}Z & Z^T \hat{G}Q & 0 \\ Q^T \hat{G}Z & Q^T \hat{G}Q & Q^T A^T \\ AZ & AQ & 0 \end{pmatrix} \begin{pmatrix} x_z \\ x_q \\ y \end{pmatrix}. \end{aligned}$$

Nótese que $R = Q^T A^T$ y $R^T = AQ$, luego

$$P^T \mathcal{A}Pv = \begin{pmatrix} Z^T \hat{G}Z & Z^T \hat{G}Q & 0 \\ Q^T \hat{G}Z & Q^T \hat{G}Q & R \\ AZ & R^T & 0 \end{pmatrix} \begin{pmatrix} x_z \\ x_q \\ y \end{pmatrix}.$$

Por otro lado, el lado derecho es igual a

$$\begin{aligned}
 \lambda P^T M P v &= \lambda \begin{pmatrix} Z^T & 0 \\ Q^T & 0 \\ O & I \end{pmatrix} \begin{pmatrix} \hat{G} + A^T W^{-1} A & 0 \\ 0 & W \end{pmatrix} \begin{pmatrix} Z & Q & 0 \\ 0 & 0 & I \end{pmatrix} \begin{pmatrix} x_z \\ x_q \\ y \end{pmatrix} \\
 &= \lambda \begin{pmatrix} Z^T \hat{G} + Z^T A^T W^{-1} A & 0 \\ Q^T \hat{G} + Q^T A^T W^{-1} A & 0 \\ 0 & W \end{pmatrix} \begin{pmatrix} Z & Q & 0 \\ 0 & 0 & I \end{pmatrix} \begin{pmatrix} x_z \\ x_q \\ y \end{pmatrix} \\
 &= \lambda \begin{pmatrix} Z^T \hat{G} + (AZ)^T W^{-1} A & 0 \\ Q^T \hat{G} + RW^{-1} A & 0 \\ 0 & W \end{pmatrix} \begin{pmatrix} Z & Q & 0 \\ 0 & 0 & I \end{pmatrix} \begin{pmatrix} x_z \\ x_q \\ y \end{pmatrix} \\
 &= \lambda \begin{pmatrix} Z^T \hat{G} & 0 \\ Q^T \hat{G} + RW^{-1} A & 0 \\ 0 & W \end{pmatrix} \begin{pmatrix} Z & Q & 0 \\ 0 & 0 & I \end{pmatrix} \begin{pmatrix} x_z \\ x_q \\ y \end{pmatrix} \\
 &= \lambda \begin{pmatrix} Z^T \hat{G} Z & Z^T \hat{G} Q & 0 \\ Q^T \hat{G} Z + RW^{-1} A Z & Q^T \hat{G} Q + RW^{-1} A Q & 0 \\ 0 & 0 & W \end{pmatrix} \begin{pmatrix} x_z \\ x_q \\ y \end{pmatrix} \\
 &= \lambda \begin{pmatrix} Z^T \hat{G} Z & Z^T \hat{G} Q & 0 \\ Q^T \hat{G} Z & Q^T \hat{G} Q + RW^{-1} R^T & 0 \\ 0 & 0 & W \end{pmatrix} \begin{pmatrix} x_z \\ x_q \\ y \end{pmatrix}.
 \end{aligned}$$

En consecuencia, se tiene el siguiente sistema:

$$\begin{pmatrix} Z^T \hat{G} Z & Z^T \hat{G} Q & 0 \\ Q^T \hat{G} Z & Q^T \hat{G} Q + RW^{-1} R^T & 0 \\ 0 & R^T & 0 \end{pmatrix} \begin{pmatrix} x_z \\ x_q \\ y \end{pmatrix} = \lambda \begin{pmatrix} Z^T \hat{G} Z & Z^T \hat{G} Q & 0 \\ Q^T \hat{G} Z & Q^T \hat{G} Q + RW^{-1} R^T & 0 \\ 0 & 0 & W \end{pmatrix} \begin{pmatrix} x_z \\ x_q \\ y \end{pmatrix}. \quad (3.3.12)$$

Para $\lambda = 1$, el sistema se reduce a

$$\begin{pmatrix} 0 & 0 & 0 \\ 0 & -RW^{-1} R^T & R \\ 0 & R^T & -W \end{pmatrix} \begin{pmatrix} x_z \\ x_q \\ y \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}. \quad (3.3.13)$$

Sea e_i la i -ésima columna de la matriz identidad de dimension $n-m$, $I_{(n-m) \times (n-m)}$. Nótese que hay $n-m$ vectores propios que pueden ser escritos como

$$(x_z, x_q, y) = (e_i, 0, 0). \quad (3.3.14)$$

De la ecuación (3.3.13) se obtiene el siguiente sistema lineal:

$$-RW^{-1}R^T x_q + Ry = 0, \quad (3.3.15a)$$

$$R^T x_q - Wy = 0. \quad (3.3.15b)$$

De (3.3.15b)

$$\begin{aligned} R^T x_q &= Wy \\ y &= W^{-1}R^T x_q. \end{aligned}$$

Al sustituir en (3.3.15a),

$$-RW^{-1}R^T x_q + RW^{-1}R^T x_q = 0.$$

Al tomar $x_q = e_j$ la j -ésima columna de la matriz identidad $I_{m \times m}$, se obtienen m vectores propios linealmente independientes de la forma:

$$(x_z, x_q, y) = (0, e_j, W^{-1}R^T e_j). \quad (3.3.16)$$

Ahora, con $\lambda = -1$ la ecuación (3.3.12) se reduce a

$$\begin{pmatrix} 2Z^T \hat{G}Z & 2Z^T \hat{G}Q & 0 \\ 2Q^T \hat{G}Z & 2Q^T \hat{G}Q + RW^{-1}R^T & R \\ 0 & R^T & W \end{pmatrix} \begin{pmatrix} x_z \\ x_q \\ y \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}.$$

Sea $x^* = Zx_z^* + Qx_q^*$ un vector en el espacio nulo de \hat{G} , $\hat{G}(Zx_z^* + Qx_q^*) = 0$, se sigue que

$$\begin{pmatrix} 2Z^T \hat{G}Z & 2Z^T \hat{G}Q & 0 \\ 2Q^T \hat{G}Z & 2Q^T \hat{G}Q + RW^{-1}R^T & R \\ 0 & R^T & W \end{pmatrix} \begin{pmatrix} x_z^* \\ x_q^* \\ y \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

$$\begin{pmatrix} 2Z^T \hat{G}Zx_z^* + 2Z^T \hat{G}Qx_q^* \\ 2Q^T \hat{G}Zx_z^* + 2Q^T \hat{G}Qx_q^* + RW^{-1}R^T x_q^* + Ry \\ R^T x_q^* + Wy \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

$$\begin{pmatrix} 2Z^T \hat{G}(Zx_z^* + Qx_q^*) \\ 2Q^T \hat{G}(Zx_z^* + Qx_q^*) + RW^{-1}R^T x_q^* + Ry \\ R^T x_q^* + Wy \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

$$\begin{pmatrix} 0 \\ RW^{-1}R^T x_q^* + Ry \\ R^T x_q^* + Wy \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

i.e.

$$RW^{-1}R^T x_q^* + Ry = 0, \quad (3.3.17a)$$

$$R^T x_q^* + Wy = 0. \quad (3.3.17b)$$

De (3.3.17b)

$$\begin{aligned} Wy &= -R^T x_q^* \\ y &= -W^{-1}R^T x_q^*. \end{aligned}$$

Como la dimensión del espacio nulo de \hat{G} es p , hay p vectores propios de la forma

$$(x_z, x_q, y) = (x_z^*, x_q^*, -W^{-1}R^T x_q^*). \quad (3.3.18)$$

Se supondrá que $|\lambda| \neq 1$. Podemos revisar la ecuación (3.3.5) para $k = 0$, ya que este preconditionador es un caso particular de la familia de preconditionadores

de la sección 3.3.1

$$\begin{aligned} A^T W^{-1} A x &= -\frac{\lambda^2 - \lambda}{\lambda^2 - 1} \hat{G} x \\ \iff A^T W^{-1} A x &= -\frac{\lambda(\lambda - 1)}{(\lambda + 1)(\lambda - 1)} \hat{G} x \\ \iff A^T W^{-1} A x &= -\frac{\lambda}{\lambda + 1} \hat{G} x. \end{aligned}$$

Sea $\mu = -\frac{\lambda}{\lambda + 1}$, por (3.3.6) $\mu > 0$, por lo que $-\frac{\lambda}{\lambda + 1} > 0$.

Se analizarán los dos posibles casos:

Si $\lambda < 0$ y $\mu > 0$, entonces $\lambda + 1 > 0$, lo que implica que $\lambda > -1$.

Si $\lambda > 0$ y $\mu > 0$, entonces $\lambda + 1 < 0$, lo que implica que $\lambda < -1$ que contradice que $\lambda > 0$.

Por lo tanto, $\lambda \in (-1, 0)$. Para encontrar una expresión para los vectores propios correspondientes, se reducirá la ecuación (3.3.12) a un problema de valores propios en x_q . Se tiene el sistema de ecuaciones

$$Z^T \hat{G} Z x_z + Z^T \hat{G} Q x_q = \lambda Z^T \hat{G} Z x_z + \lambda Z^T \hat{G} Q x_q, \quad (3.3.19a)$$

$$Q^T \hat{G} Z x_z + Q^T \hat{G} Q x_q + R y = \lambda Q^T \hat{G} Z x_z + \lambda R W^{-1} R x_q, \quad (3.3.19b)$$

$$R^T x_q = \lambda W y. \quad (3.3.19c)$$

De (3.3.19c)

$$y = \frac{1}{\lambda} W^{-1} R^T x_q.$$

Al despejar (3.3.19a)

$$\begin{aligned} Z^T \hat{G} Z x_z + Z^T \hat{G} Q x_q &= \lambda Z^T \hat{G} Z x_z + \lambda Z^T \hat{G} Q x_q \\ Z^T \hat{G} Z x_z - \lambda Z^T \hat{G} Z x_z &= \lambda Z^T \hat{G} Q x_q - Z^T \hat{G} Q x_q \\ (1 - \lambda) Z^T \hat{G} Z x_z &= -(1 - \lambda) (Z^T \hat{G} Q x_q) \\ Z^T \hat{G} Z x_z &= -Z^T \hat{G} Q x_q. \end{aligned}$$

Nótese que $Z^T \hat{G} Z$ es positiva definida. Sea $x \in \mathbb{R}^{(n-m)}$, como $Z = [z_1 | z_2 | \dots | z_{n-m}]$

se tiene que

$$\begin{aligned}
 x^T Z^T \hat{G} Z x &= (Zx)^T \hat{G} Z x \\
 &= (Z(x_1, \dots, x_{n-m}))^T \hat{G} Z(x_1, \dots, x_{n-m}) \\
 &= (x_1 z_1 + \dots + x_{n-m} z_{n-m})^T \hat{G} (x_1 z_1 + \dots + x_{n-m} z_{n-m}) \\
 &= v^T \hat{G} v,
 \end{aligned}$$

con $v \in \text{gen}\{z_1, \dots, z_n\}$. Por hipótesis, \hat{G} es positiva definida en el espacio nulo de A , por lo tanto

$$x^T Z^T \hat{G} Z x > 0.$$

Resulta que

$$x_z = -(Z^T \hat{G} Z)^{-1} (Z^T \hat{G} Q x_q),$$

al sustituir en (3.3.19b)

$$\begin{aligned}
 Q^T \hat{G} Z (- (Z^T \hat{G} Z)^{-1} (Z^T \hat{G} Q x_q)) + Q^T \hat{G} Q x_q + R(\frac{1}{\lambda} W^{-1} R^T x_q) &= \\
 \lambda Q^T \hat{G} Z (- (Z^T \hat{G} Z)^{-1} (Z^T \hat{G} Q x_q)) + \lambda R W^{-1} R x_q
 \end{aligned}$$

$$\begin{aligned}
 \frac{1}{\lambda} R W^{-1} R^T x_q - \lambda R W^{-1} R^T x_q &= \\
 (1 - \lambda)(Q^T \hat{G} Z (Z^T \hat{G} Z)^{-1} (Z^T \hat{G} Q x_q)) + (\lambda - 1) Q^T \hat{G} Q x_q.
 \end{aligned}$$

Sea $C = Z^T \hat{G} Q$,

$$\begin{aligned}
 (\frac{1}{\lambda} - \lambda) R W^{-1} R^T x_q &= (1 - \lambda)(C^T (Z^T \hat{G} Z)^{-1} C - Q^T \hat{G} Q) x_q \\
 (\frac{1-\lambda^2}{\lambda}) R W^{-1} R^T x_q &= (1 - \lambda)(C^T (Z^T \hat{G} Z)^{-1} C - Q^T \hat{G} Q) x_q \\
 (\frac{(1-\lambda)(1+\lambda)}{\lambda}) R W^{-1} R^T x_q &= (1 - \lambda)(C^T (Z^T \hat{G} Z)^{-1} C - Q^T \hat{G} Q) x_q \\
 (\frac{1+\lambda}{\lambda}) R W^{-1} R^T x_q &= (C^T (Z^T \hat{G} Z)^{-1} C - Q^T \hat{G} Q) x_q \\
 (1 + \lambda) R W^{-1} R^T x_q &= \lambda(C^T (Z^T \hat{G} Z)^{-1} C - Q^T \hat{G} Q) x_q
 \end{aligned}$$

$$R W^{-1} R^T x_q = \lambda(C^T (Z^T \hat{G} Z)^{-1} C - Q^T \hat{G} Q - R W^{-1} R^T) x_q. \quad (3.3.20)$$

Se ha probado el siguiente teorema:

Teorema 3.3.4. *La matriz preconditionada $\mathcal{M}^{-1}\mathcal{A}$ tiene valores propios $\lambda = 1$, de multiplicidad n y vectores propios (3.3.14) y (3.3.16), y $\lambda = -1$, de multiplicidad p y vectores propios dados por (3.3.18). Los valores propios restantes caen en el intervalo $(-1, 0)$ y satisfacen el problema generalizado de valores propios (3.3.20).*

El teorema 3.3.4 ilustra la fuerte agrupación espectral cuando el bloque (1,1) de la matriz \mathcal{A} definida en (3.3.1) es singular. Una dificultad asociada con el método de puntos interiores es el incremento del mal condicionamiento del bloque (1,1) al acercarse a la solución [7].

Hay dos aspectos críticos a considerar en la aplicación del preconditionador. El primero es la elección de la matriz W . Luego, dada una matriz de pesos, elegir un método eficiente de factorización o iterativo para resolver sistemas lineales con el preconditionador.

La elección más simple de una matriz de pesos es una matriz diagonal, lo que hace que invertir W se trivial. Sea $W = \gamma I$, la matriz identidad escalada, γ puede ser elegida tal que el término $\frac{1}{\gamma}A^T A$ sea comparable en norma con \hat{G} . Nótese que como \hat{G} cambia en cada iteración, γ también debe ser actualizado. La elección de $\gamma = \frac{\|A\|^2}{\|\hat{G}\|}$ probó ser efectiva en los experimentos numéricos de [7], ya que redujó el número de iteraciones del método iterativo MINRES al resolver

$$\mathcal{M}^{-1}\mathcal{A}\tilde{x} = \mathcal{M}^{-1}\tilde{b},$$

con $\tilde{x} = (\Delta x, \Delta \lambda)^T$ y $\tilde{b} = -(\hat{d}, r_\lambda)^T$, con \hat{d} y r_λ definidos en (1.2.7).

3.3.4. Precondicionador B

La elección de $\alpha = 1$ para la familia de preconditionadores simétricos positivos definidos de la sección 3.3.2, genera un preconditionador de la forma

$$\mathcal{N} = \begin{pmatrix} \hat{G} + 2A^T H^{-1} A & A^T \\ A & H \end{pmatrix}. \quad (3.3.21)$$

Al sustituir $\alpha = 1$ en las fórmulas de los valores propios, se sigue que la matriz preconditionada $\mathcal{N}^{-1}\mathcal{A}$ tiene valores propios $\lambda_1 = 1$ de multiplicidad $(n - m)$, $\lambda_2 = -1 + \sqrt{2}$ y $\lambda_3 = -1 - \sqrt{2}$ cada uno de multiplicidad p . Los $2(m - p)$ valores propios restantes se encuentran al resolver el $\lim_{\beta \rightarrow 0, \infty} \lambda_4(\beta)$ y caen en el intervalo $(-1 - \sqrt{2}, 0) \cup (-1 + \sqrt{2}, 1)$.

De la misma forma que con el preconditionador A, la elección de la matriz de pesos será la matriz identidad escalada, $H = \gamma I$ con $\gamma \in \mathbb{R}$, $\gamma = \frac{\|A\|^2}{\|G\|}$ y se utilizará el método de MINRES para resolver

$$\mathcal{N}^{-1}\mathcal{A}\tilde{x} = \mathcal{N}^{-1}\tilde{b}.$$

3.3.5. Precondicionador C

La elección de $\alpha = -1$ para la familia de preconditionadores simétricos positivos definidos de la sección 3.3.2, genera un preconditionador de la forma

$$\mathcal{N} = \begin{pmatrix} \hat{G} + 2A^T H^{-1} A & -A^T \\ -A & H \end{pmatrix}. \quad (3.3.22)$$

Al sustituir $\alpha = -1$ en las fórmulas de los valores propios, se sigue que la matriz preconditionada $\mathcal{N}^{-1}\mathcal{A}$ tiene valores propios $\lambda_1 = 1$ de multiplicidad $(n - m)$, $\lambda_2 = 1 + \sqrt{2}$ y $\lambda_3 = 1 - \sqrt{2}$ cada uno de multiplicidad p . Los $2(m - p)$ valores propios restantes se encuentran al resolver el $\lim_{\beta \rightarrow 0, \infty} \lambda_4(\beta)$ y caen en el intervalo $(1 - \sqrt{2}, 0) \cup (1, 1 + \sqrt{2})$.

La elección de la matriz de pesos es la misma que para los preconditionadores A y B. Se utilizará el método de MINRES para resolver

$$\mathcal{N}^{-1}\mathcal{A}\tilde{x} = \mathcal{N}^{-1}\tilde{b}.$$

3.4 Algoritmo de puntos interiores preconditionado para programación cuadrática convexa

Algoritmo 5 Sistema Lineal Reducido Precondicionado

Entradas: $Q \in \mathbb{R}^{n \times n}$ simétrica y positiva definida, $A \in \mathbb{R}^{m \times n}$ ($m \leq n$) con $\text{rango}(A) = m$, $F \in \mathbb{R}^{p \times n}$ ($p \leq n$), $d \in \mathbb{R}^n$, $b \in \mathbb{R}^m$ y $c \in \mathbb{R}^p$;

Salidas: $x^* \in \mathbb{R}^n$, $\lambda^* \in \mathbb{R}^m$, $\mu^* \in \mathbb{R}^p$;

Pseudo código:

Establecer los valores iniciales $x_0 \in \mathbb{R}^n$, $\lambda_0 \in \mathbb{R}^m$, $\mu_0 \in \mathbb{R}^p$, $z_0 \in \mathbb{R}^p$ con $\mu_0 > 0$, $z_0 > 0$;

$k \leftarrow 0$;

tol es una tolerancia determinada

$maxiter$ es el número máximo de iteraciones permitidas

mientras $\|H(x, \lambda, \mu, z)\|_2 > tol$ & $k < maxiter$ **hacer**

 Resolver el sistema lineal (1.2.6) con el método de MINRES usando el preconditionador \mathcal{P}^{-1}

 Resolver las incógnitas restantes

$$\begin{aligned}\Delta\mu &\leftarrow -(Z^{-1}U)(F\Delta x + \omega), \\ \Delta z &\leftarrow -U^{-1}(Z\Delta\mu + r_z); \end{aligned}$$

Determinar $\alpha_k > 0$ tal que

$$\begin{aligned}\mu_k + \alpha_k \Delta\mu &> 0; \\ z_k + \alpha_k \Delta z &> 0; \end{aligned}$$

Actualizar

$$\begin{aligned}x_{k+1} &\leftarrow x_k + \alpha_k \Delta x; \\ \lambda_{k+1} &\leftarrow \lambda_k + \alpha_k \Delta \lambda; \\ \mu_{k+1} &\leftarrow \mu_k + \alpha_k \Delta \mu; \\ z_{k+1} &\leftarrow z_k + \alpha_k \Delta z; \\ \gamma_{k+1} &\leftarrow \frac{1}{2p} \mu_{k+1}^T z_{k+1}; \\ k &\leftarrow k + 1; \end{aligned}$$

fin mientras

4 Resultados numéricos

La implementación del método de puntos interiores para programación cuadrática convexa se realizó en Matlab, en una computadora con procesador Intel Core i7-1165G7 11va generación, 2.80 GHz, 4 núcleos, 8 procesadores lógicos y 16 GB de RAM. Las pruebas numéricas se realizaron en un subconjunto de problemas lineales del repositorio Netlib [8] en formato .mat. Los problemas lineales son de la forma

$$\begin{array}{ll} \text{Minimizar} & c^T x \\ \text{s.a.} & Ax = 0 \\ & x \geq 0. \end{array}$$

Con el fin de tener un problema de programación cuadrática como en (1.1.1), se modificó la función objetivo agregando el término $x^T x$. Así, se tienen problemas con el siguiente formato:

$$\begin{array}{ll} \text{Minimizar} & x^T x + c^T x \\ \text{s.a.} & Ax = 0 \\ & x \geq 0. \end{array}$$

Los puntos iniciales para el proceso iterativo son los siguientes: x_0 , μ_0 , y z_0 son vectores de unos y λ_0 es un vector de ceros. Tomamos una tolerancia de 1×10^{-6} y un máximo de iteraciones de 100.

Para la estimación del tiempo de ejecución de cada uno de los algoritmos

| Problema | Filas | Columnas | Entradas | NoCero |
|----------|-------|----------|----------|--------|
| AFIRO | 27 | 51 | 1377 | 7.4 % |
| BLEND | 74 | 114 | 8436 | 6.2 % |
| FIT1D | 24 | 1049 | 25176 | 53.3 % |
| FIT1P | 627 | 1677 | 1051479 | 0.9 % |
| GROW15 | 300 | 645 | 193500 | 2.9 % |
| GROW22 | 440 | 946 | 416240 | 2.0 % |
| GROW7 | 140 | 301 | 42140 | 6.2 % |
| KB2 | 43 | 68 | 2924 | 10.7 % |
| NESM | 662 | 3105 | 2055510 | 0.7 % |
| RECIPELP | 91 | 204 | 18564 | 3.7 % |
| SCAGR7 | 129 | 185 | 23865 | 1.9 % |
| SCSD1 | 77 | 760 | 58520 | 4.1 % |
| SCSD6 | 147 | 1350 | 198450 | 2.2 % |
| SCSD8 | 397 | 2750 | 1091750 | 0.8 % |
| SCTAP1 | 300 | 660 | 198000 | 0.9 % |
| SCTAP2 | 1090 | 2500 | 2725000 | 0.3 % |
| SCTAP3 | 1480 | 3340 | 4943200 | 0.2 % |
| SHARE2B | 96 | 162 | 15552 | 5.0 % |
| STAIR | 356 | 620 | 220720 | 1.8 % |
| STOCFOR1 | 117 | 165 | 19305 | 2.6 % |

Tabla 4.1: Dimensión de los problemas Netlib

que se evaluarán en esta sección, se utilizará la función *timeit(f)* de Matlab que mide el tiempo (en segundos) requerido para ejecutar la función *f*. *timeit* ejecuta la función *f* múltiples veces y regresa la mediana de las mediciones, por lo que se tiene una medición robusta del tiempo de cómputo.

Para las pruebas numéricas nos enfocamos en los problemas cuadráticos que tuvieran solución para el algoritmo 1 de la sección 1.4. En la tabla 4.1 se reportan las dimensiones, el número de entradas y el porcentaje de entradas diferentes de cero de la matriz de restricciones de igualdad para cada problema. En la tabla 4.2 se muestra el número de iteraciones y el tiempo estimado que le tomó al algoritmo del Sistema Lineal Completo converger a la tolerancia

| Problema | Iteraciones | Tiempo (s) |
|----------|-------------|------------|
| AFIRO | 29 | 0.018 |
| BLEND | 29 | 0.138 |
| FIT1D | 39 | 21.645 |
| FIT1P | 70 | 209.631 |
| GROW15 | 24 | 6.111 |
| GROW22 | 24 | 16.124 |
| GROW7 | 23 | 0.86 |
| KB2 | 26 | 0.044 |
| NESM | 51 | 838.738 |
| RECIPELP | 26 | 0.522 |
| SCAGR7 | 54 | 0.934 |
| SCSD1 | 22 | 7.153 |
| SCSD6 | 22 | 32.294 |
| SCSD8 | 23 | 253.677 |
| SCTAP1 | 51 | 14.779 |
| SCTAP2 | 45 | 486.14 |
| SCTAP3 | 42 | 934.124 |
| SHARE2B | 45 | 0.501 |
| STAIR | 67 | 15.382 |
| STOCFOR1 | 45 | 0.543 |

Tabla 4.2: Resultados obtenidos por el algoritmo 1 descrito en la sección 1.4

deseada. Por otro lado, en la tabla 4.3 se muestra el rendimiento del algoritmo del Sistema Lineal Reducido para el mismo conjunto de problemas. La columna cuatro *MalCond* indica si Matlab lanzó una alerta por el mal condicionamiento de la matriz (1.2.6) durante la ejecución. Al igual que con el Sistema Lineal Completo, el algoritmo convergió a la tolerancia deseada.

Al comparar el rendimiento de estos algoritmos, se observa que ambos realizan el mismo número de iteraciones. Sin embargo, el tiempo estimado del algoritmo del Sistema Lineal Reducido es menor para todos los casos como se ve en la figura 4.1.

| Problema | Iteraciones | Tiempo (s) | MalCond |
|----------|-------------|------------|---------|
| AFIRO | 29 | 0.011 | No |
| BLEND | 29 | 0.059 | Sí |
| FIT1D | 39 | 13.262 | No |
| FIT1P | 70 | 89.824 | Sí |
| GROW15 | 24 | 1.561 | No |
| GROW22 | 24 | 4.448 | No |
| GROW7 | 23 | 0.186 | No |
| KB2 | 26 | 0.013 | No |
| NESM | 51 | 124.006 | No |
| RECIPELP | 26 | 0.086 | No |
| SCAGR7 | 54 | 0.243 | Sí |
| SCSD1 | 22 | 1.205 | No |
| SCSD6 | 22 | 5.514 | No |
| SCSD8 | 23 | 42.895 | No |
| SCTAP1 | 51 | 2.828 | Sí |
| SCTAP2 | 45 | 81.596 | Sí |
| SCTAP3 | 42 | 156.113 | Sí |
| SHARE2B | 45 | 0.134 | Sí |
| STAIR | 67 | 3.548 | Sí |
| STOCFOR1 | 45 | 0.128 | No |

Tabla 4.3: Resultados obtenidos por el algoritmo 2 descrito en la sección 1.4

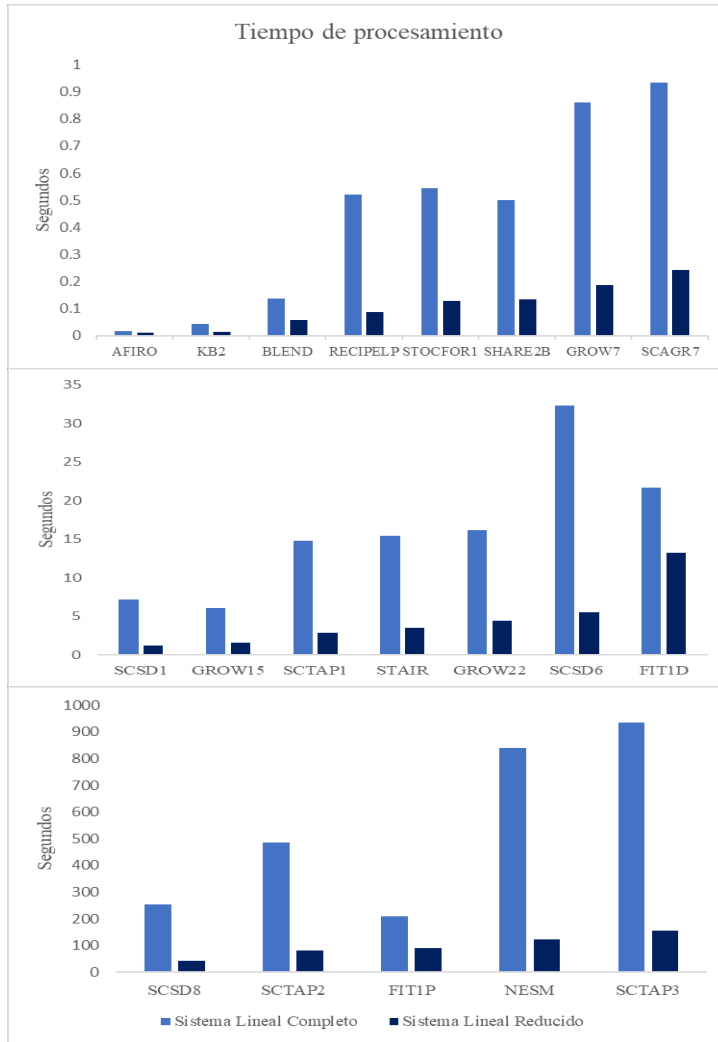


Figura 4.1: Comparación de los tiempos de los algoritmos 1 y 2 de la sección 1.4

Para la implementación del algoritmo del Sistema Lineal Reducido Precondicionado se utilizó la función *minres* de Matlab. Los parámetros de entrada de la función son los siguientes:

- A una matriz de $n \times n$
- b es un vector de dimensión n
- tol es la tolerancia del método
- $maxit$ es el máximo número de iteraciones del método
- M_1 una matriz de $n \times n$
- M_2 una matriz de $n \times n$

El método resuelve el sistema $H^{-1}AH^{-T}y = H^{-1}b$ donde $y = H^Tx$ y $H = M^{1/2} = (M_1M_2)^{1/2}$, con M un preconditionador simétrico positivo definido tal que $M = M_1M_2$ es la factorización de Cholesky. Cabe mencionar que el algoritmo no calcula explícitamente H . Para nuestra implementación se tomó una tolerancia de 1×10^{-5} y un máximo número de iteraciones de 100.

En la implementación fue necesario incluir otro criterio de paro: si el recíproco del condicional de la matriz (1.2.6) es menor al epsilon de la máquina, el algoritmo se detiene. El recíproco del condicional debe ser mayor al epsilon de la máquina porque las matrices M dejan de ser positivas definidas para la precisión de la máquina, lo que causa un error en la función *minres*. En la figura 4.2 podemos observar como el método de puntos interiores preconditionado no logra llegar a la tolerancia deseada para el problema BLEND. Los picos que se observan en la gráfica ocurren cuando el preconditionador A deja de ser positivo definido para la precisión de la máquina.

En las tablas 4.4-4.6 se muestran distintas métricas del trabajo requerido al aplicar los preconditionadores del capítulo 3 a los problemas seleccionados. En estas tablas, $\|CNPO\|$ es la norma infinito de las condiciones necesarias de KKT al finalizar la ejecución del método e *iter* denota el total de iteraciones del

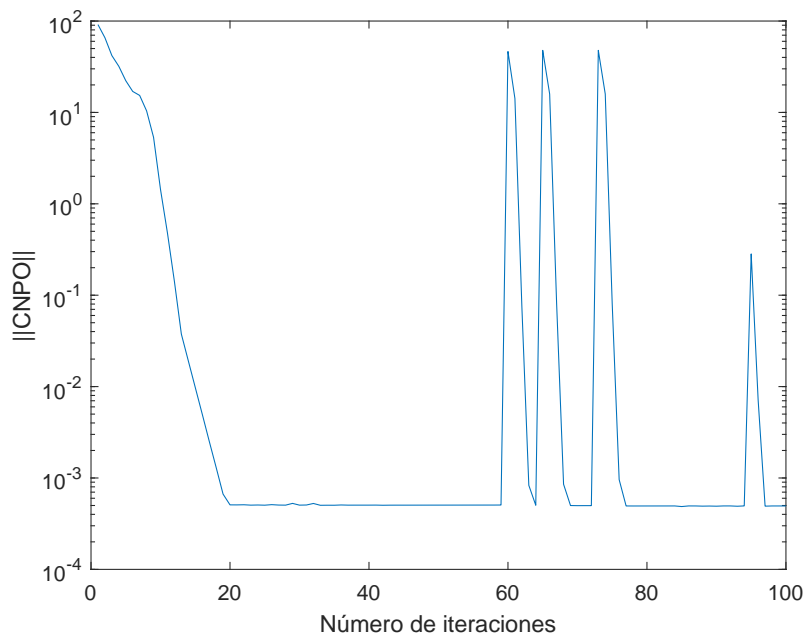


Figura 4.2: Convergencia del Algoritmo 5 con el preconditionador A para el problema BLEND

método de puntos interiores. Por otro lado, Mr es el promedio de iteraciones del método de *minres* por cada paso del método de puntos interiores y $TotMr$ es la suma del total de las iteraciones de *minres* realizadas en todos los pasos. Por último, *tiempo* es el tiempo estimado de ejecución y *rcond* indica si el método se detuvo por el mal condicionamiento de la matriz (1.2.6).

| Problema | $\ CNPO\ $ | iter | Mr | TotMr | tiempo | rcond |
|----------|------------|------|-------|-------|---------|-------|
| AFIRO | 0.00000004 | 37 | 3.54 | 131 | 0.136 | No |
| BLEND | 0.00050449 | 23 | 38.96 | 896 | 0.289 | Sí |
| FIT1D | 0.00000010 | 48 | 20.02 | 961 | 24.936 | No |
| FIT1P | 0.00262942 | 65 | 9.58 | 623 | 97.042 | Sí |
| GROW15 | 0.00000057 | 24 | 7.50 | 180 | 4.265 | No |
| GROW22 | 0.00000059 | 24 | 7.71 | 185 | 9.737 | No |
| GROW7 | 0.00000085 | 23 | 6.13 | 141 | 0.465 | No |
| KB2 | 0.00000332 | 30 | 43.13 | 1294 | 0.195 | Sí |
| NESM | 0.00000059 | 52 | 5.73 | 298 | 207.711 | No |
| RECIPELP | 0.00000047 | 36 | 66.31 | 2387 | 1.569 | No |
| SCAGR7 | 0.18937015 | 40 | 9.47 | 379 | 0.506 | Sí |
| SCSD1 | 0.00005377 | 26 | 12.50 | 325 | 3.583 | Sí |
| SCSD6 | 0.00004307 | 26 | 16.69 | 434 | 16.852 | Sí |
| SCSD8 | 0.00003905 | 27 | 14.67 | 396 | 109.694 | Sí |
| SCTAP1 | 0.00440818 | 42 | 28.10 | 1180 | 12.798 | Sí |
| SCTAP2 | 0.00611384 | 39 | 33.92 | 1323 | 385.564 | Sí |
| SCTAP3 | 0.00375246 | 37 | 33.11 | 1225 | 892.475 | Sí |
| SHARE2B | 0.00128478 | 61 | 50.41 | 3075 | 1.904 | Sí |
| STAIR | 0.00391336 | 57 | 6.30 | 359 | 7.674 | Sí |
| STOCFOR1 | 0.00005312 | 50 | 29.02 | 1451 | 1.307 | Sí |

Tabla 4.4: Rendimiento del preconditionador A

| Problema | $\ CNPO\ $ | iter | Mr | TotMr | time | rcond |
|----------|------------|------|-------|-------|---------|-------|
| AFIRO | 0.00000004 | 39 | 6.08 | 237 | 0.038 | No |
| BLEND | 0.00033033 | 24 | 55.38 | 1329 | 0.338 | Sí |
| FIT1D | 0.00000008 | 49 | 35.90 | 1759 | 17.485 | No |
| FIT1P | 0.24522032 | 66 | 17.88 | 1180 | 94.965 | Sí |
| GROW15 | 0.00000057 | 24 | 19.96 | 479 | 5.016 | No |
| GROW22 | 0.00000059 | 24 | 20.50 | 492 | 12.124 | No |
| GROW7 | 0.00000085 | 23 | 15.13 | 348 | 0.979 | No |
| KB2 | 0.00001321 | 33 | 74.64 | 2463 | 0.378 | Sí |
| NESM | 0.00000080 | 50 | 9.40 | 470 | 223.924 | No |
| RECIPELP | 0.00000066 | 39 | 86.46 | 3372 | 3.484 | No |
| SCAGR7 | 0.03791244 | 42 | 15.24 | 640 | 0.623 | Sí |
| SCSD1 | 0.00007717 | 26 | 36.50 | 949 | 7.575 | Sí |
| SCSD6 | 0.00006806 | 26 | 43.04 | 1119 | 29.433 | Sí |
| SCSD8 | 0.00007458 | 27 | 30.89 | 834 | 95.965 | Sí |
| SCTAP1 | 0.00733910 | 43 | 73.63 | 3166 | 27.695 | Sí |
| SCTAP2 | 0.00517681 | 40 | 75.70 | 3028 | 377.465 | Sí |
| SCTAP3 | 0.00337930 | 38 | 72.55 | 2757 | 601.605 | Sí |
| SHARE2B | 0.00139199 | 84 | 71.60 | 6014 | 3.862 | Sí |
| STAIR | 0.00060421 | 57 | 11.14 | 635 | 9.049 | Sí |
| STOCFOR1 | 0.00165567 | 53 | 47.58 | 2522 | 2.537 | Sí |

Tabla 4.5: Rendimiento del preconditionador B

| Problema | $\ CNPO\ $ | iter | Mr | TotMr | time | rcond |
|----------|------------|------|-------|-------|---------|-------|
| AFIRO | 0.00000003 | 39 | 5.95 | 232 | 0.043 | No |
| BLEND | 0.00054329 | 24 | 56.62 | 1359 | 0.353 | Sí |
| FIT1D | 0.00000839 | 49 | 34.00 | 1666 | 20.379 | Sí |
| FIT1P | 0.07878637 | 66 | 16.94 | 1118 | 95.578 | Sí |
| GROW15 | 0.00000057 | 24 | 14.92 | 358 | 4.283 | No |
| GROW22 | 0.00000059 | 24 | 15.42 | 370 | 9.720 | No |
| GROW7 | 0.00000043 | 24 | 11.46 | 275 | 0.944 | No |
| KB2 | 0.00004459 | 30 | 77.77 | 2333 | 0.359 | Sí |
| NESM | 0.00000093 | 53 | 8.66 | 459 | 234.538 | No |
| RECIPELP | 0.00000056 | 43 | 89.00 | 3827 | 4.227 | No |
| SCAGR7 | 0.18502127 | 41 | 15.41 | 632 | 0.890 | Sí |
| SCSD1 | 0.00008010 | 26 | 25.69 | 668 | 5.631 | Sí |
| SCSD6 | 0.00006095 | 26 | 34.12 | 887 | 23.850 | Sí |
| SCSD8 | 0.00013803 | 27 | 25.81 | 697 | 109.918 | Sí |
| SCTAP1 | 0.00113474 | 43 | 52.02 | 2237 | 19.236 | Sí |
| SCTAP2 | 0.00487233 | 39 | 56.64 | 2209 | 221.240 | Sí |
| SCTAP3 | 0.00259558 | 37 | 57.11 | 2113 | 383.931 | Sí |
| SHARE2B | 0.00248268 | 74 | 73.15 | 5413 | 3.274 | Sí |
| STAIR | 0.00188256 | 57 | 10.56 | 602 | 6.848 | Sí |
| STOCFOR1 | 0.00033476 | 50 | 41.02 | 2051 | 1.516 | Sí |

Tabla 4.6: Rendimiento del preconditionador C

| Problema | PrecondA | PrecondB | PrecondC |
|----------|-------------------|-------------------|-------------------|
| AFIRO | 0.00000004 | 0.00000004 | 0.00000003 |
| BLEND | 0.00050449 | 0.00033033 | 0.00054329 |
| FIT1D | 0.00000010 | 0.00000008 | 0.00000839 |
| FIT1P | 0.00262942 | 0.24522032 | 0.07878637 |
| GROW15 | 0.00000057 | 0.00000057 | 0.00000057 |
| GROW22 | 0.00000059 | 0.00000059 | 0.00000059 |
| GROW7 | 0.00000085 | 0.00000085 | 0.00000043 |
| KB2 | 0.00000332 | 0.00001321 | 0.00004459 |
| NESM | 0.00000059 | 0.00000080 | 0.00000093 |
| RECIPELP | 0.00000047 | 0.00000066 | 0.00000056 |
| SCAGR7 | 0.18937015 | 0.03791244 | 0.18502127 |
| SCSD1 | 0.00005377 | 0.00007717 | 0.00008010 |
| SCSD6 | 0.00004307 | 0.00006806 | 0.00006095 |
| SCSD8 | 0.00003905 | 0.00007458 | 0.00013803 |
| SCTAP1 | 0.00440818 | 0.00733910 | 0.00113474 |
| SCTAP2 | 0.00611384 | 0.00517681 | 0.00487233 |
| SCTAP3 | 0.00375246 | 0.00337930 | 0.00259558 |
| SHARE2B | 0.00128478 | 0.00139199 | 0.00248268 |
| STAIR | 0.00391336 | 0.00060421 | 0.00188256 |
| STOCFOR1 | 0.00005312 | 0.00165567 | 0.00033476 |

Tabla 4.7: Comparación de la precisión de los preconditionadores

En la tabla 4.7 se tiene la comparación de la norma infinito de las condiciones necesarias de KKT para cada uno de los preconditionadores propuestos. Nótese que el preconditionador A aproxima mejor la solución en doce de los veinte problemas analizados, el preconditionador B aproxima mejor la solución en siete problemas y el uso del preconditionador C en seis. Para los problemas GROW15 y GROW22, usar cualquiera de los preconditionadores lleva al mismo resultado.

| Problema | PrecondA | PrecondB | PrecondC |
|----------|-----------|-----------|-----------|
| AFIRO | 37 | 39 | 39 |
| BLEND | 23 | 24 | 24 |
| FIT1D | 48 | 49 | 49 |
| FIT1P | 65 | 66 | 66 |
| GROW15 | 24 | 24 | 24 |
| GROW22 | 24 | 24 | 24 |
| GROW7 | 23 | 23 | 24 |
| KB2 | 30 | 33 | 30 |
| NESM | 52 | 50 | 53 |
| RECIPELP | 36 | 39 | 43 |
| SCAGR7 | 40 | 42 | 41 |
| SCSD1 | 26 | 26 | 26 |
| SCSD6 | 26 | 26 | 26 |
| SCSD8 | 27 | 27 | 27 |
| SCTAP1 | 42 | 43 | 43 |
| SCTAP2 | 39 | 40 | 39 |
| SCTAP3 | 37 | 38 | 37 |
| SHARE2B | 61 | 84 | 74 |
| STAIR | 57 | 57 | 57 |
| STOCFOR1 | 50 | 53 | 50 |

Tabla 4.8: Comparación del número de iteraciones

Al comparar el total de iteraciones del método de puntos interiores, se puede observar que el preconditionador A es la mejor opción en 19 de los problemas (ver el tabla 4.8), siendo el mejor preconditionador en este rubro. De igual forma, al analizar el tiempo de ejecución estimado, se tiene que el preconditionador con mejor rendimiento es el A para doce de los problemas analizados (ver figura 4.3). Esto es consistente con el promedio de iteraciones que realiza el método de *minres* por cada paso del método de puntos interiores. Los códigos de cada uno de los algoritmos analizados se pueden encontrar en <https://github.com/javierMoG/PrecondicionadoresMetodoPuntosInteriores.git>

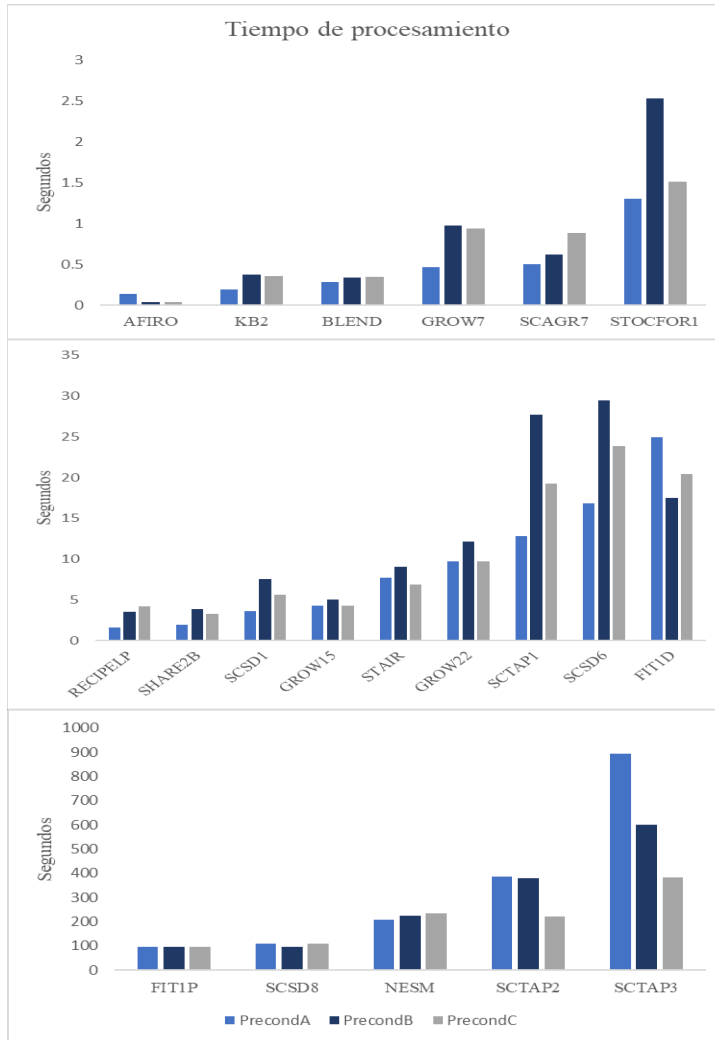


Figura 4.3: Comparación de los tiempos de ejecución

5 Conclusiones

En el capítulo anterior, se pudo observar la diferencia de rendimiento de utilizar el Sistema Lineal Completo al Sistema Lineal Reducido. El tiempo de procesamiento es significativamente menor al usar el segundo. No obstante, se tiene el problema del mal condicionamiento de la matriz (1.2.6), lo que puede generar resultados erróneos. A pesar de la existencia de varias técnicas de factorización que buscan reducir el mal condicionamiento del sistema lineal a resolver, para matrices de dimensiones muy grandes y con muchos ceros (*sparse matrix*) la factorización puede tomar mucho tiempo en calcularse. El uso de métodos de subespacios de Krylov para resolver sistemas lineales con preconditionadores ha mostrado tener buenos resultados para este tipo de matrices [7].

Al elegir un preconditionador \mathcal{P} se toman en cuenta factores como su efectividad, el costo de procesamiento y almacenamiento y que resuelva el sistema lineal $\mathcal{P}y = r$ de manera eficiente. Las familias de preconditionadores vistas en el capítulo 3, tienen la atractiva propiedad de mejorar el agrupamiento de los valores propios del sistema (1.2.6) con el incremento del mal condicionamiento del bloque (1,1). Como se observó en la sección 1.5, el mal condicionamiento de la matriz \hat{G} aumenta conforme el método de puntos interiores se acerca a la solución.

En esta tesis se comparó el rendimiento de tres preconditionadores: A, B y C. El preconditionador A obtuvo el valor más pequeño en norma infinito de las condiciones necesarias de KKT en doce de los veinte problemas analizados.

De igual manera, en siete de estos problemas el preconditionador A llegó a la tolerancia deseada, mismo número que el preconditionador B. Con base en lo anterior, el preconditionador A fue el más preciso. En lo que respecta al total de iteraciones realizadas del método de puntos interiores, el preconditionador A tuvo el menor número de iteraciones en diecinueve de los veinte problemas. Para el tiempo de procesamiento, el preconditionador A fue el mejor en doce de los problemas analizados. Si observamos el promedio de iteraciones del método de *minres* por cada paso del método de puntos interiores, el preconditionador A obtuvo el menor número en todos los problemas. Por lo tanto, es el preconditionador más eficiente. En términos de costo de procesamiento y almacenamiento, el preconditionador A es la mejor opción al ser una matriz diagonal por bloques. En conclusión, el preconditionador A tuvo el mejor rendimiento de todos los preconditionadores evaluados.

Como posibles mejoras al trabajo realizado, se evaluaría el uso de otro método de subespacios de Krylov para resolver sistemas lineales. El método de *minres* preconditionado de Matlab mostró un rendimiento menor al método *mldivide* [9]. Por otro lado, el método predictor-corrector que se utilizó en [7] y [10] es un algoritmo más estable que el método de puntos interiores implementado en este trabajo, por lo que se podría mejorar la convergencia del Sistema Lineal Reducido Precondicionado y evitar los picos observados en la figura 4.2.

Bibliografía

- [1] Optimization Online, “Preconditioners applications,” Disponible en <https://optimization-online.org/?s=Preconditioners+applications>(2022/09/28).
- [2] S. J. Wright, *Primal-dual interior-point methods*. Philadelphia, PA: Society for Industrial and Applied Mathematics, 1997 , pp. 36-40.
- [3] T. Sauer, *Numerical Analysis*, 2nd ed. Estados Unidos de América: Pearson, 2013.
- [4] L. N. Trefethen and D. Bau, *Numerical linear algebra*. Siam, 2022, vol. 181, , pp.93-95, 243-248, 250-254, 266-271, 276-278.
- [5] W. Kahan, “Numerical linear algebra,” *Canadian Mathematical Bulletin*, vol. 9, no. 5, p. 757–801, 1966.
- [6] “LAPACK—Linear Algebra PACKage,” Disponible en <http://www.netlib.org/lapack/> (2022/04/09).
- [7] T. Rees and C. Greif, “A preconditioner for linear systems arising from interior point optimization methods,” *SIAM Journal on Scientific Computing*, vol. 29, no. 5, pp. 1992–2007, 2007.
- [8] W. Wang, “NETLIB Linear Programming Problem Collection,” Disponible en http://homepage.ntu.edu.tw/~wwang/cola.lab/test_problems/netlib_lp/(2022/06/11).

- [9] MathWorks, “mldivide,” Disponible en [https://la.mathworks.com/help/matlab/ref/mldivide.html\(2022/09/28\)](https://la.mathworks.com/help/matlab/ref/mldivide.html(2022/09/28)).
- [10] E. K. Lee and Z. Parada, “A new family of symmetric positive definite pre-conditioners for quadratic interior point methods,” 2011, research work, School of Industrial & Systems Engineering, Georgia Institute of Technology, Atlanta, USA.
- [11] S. J. Wright, J. Nocedal *et al.*, *Numerical optimization*. New York: Springer Science, 1999.
- [12] Z. Parada, “Notas de clase de optimización numérica,” 2020, Documento de Trabajo, Departamento de Matemáticas, ITAM.
- [13] C. Greif and D. Schötzau, “Preconditioners for saddle point linear systems with highly singular (1,1) blocks,” *ETNA, Special Volume on Saddle Point Problems*, vol. 22, pp. 114–121, 2006.
- [14] S. H. Friedberg, A. J. Insel, and L. E. Spence, *Linear Algebra*, 4th ed. Pearson, Prentice Hall, 2003.
- [15] MathWorks, “rcond,” Disponible en [https://la.mathworks.com/help/matlab/ref/rcond.html\(2022/04/09\)](https://la.mathworks.com/help/matlab/ref/rcond.html(2022/04/09)).
- [16] —, “rcond warning differs from computed value of rcond,” Disponible en [https://la.mathworks.com/matlabcentral/answers/588805-rcond-warning-differs-from-computed-value-of-rcond\(2022/04/09\)](https://la.mathworks.com/matlabcentral/answers/588805-rcond-warning-differs-from-computed-value-of-rcond(2022/04/09)).
- [17] D. G. Luenberger, *Introduction to Linear and Nonlinear Programming*, 2nd ed. Stanford, CA: Springer, 2003.
- [18] MathWorks, “minres,” Disponible en [https://www.mathworks.com/help/matlab/ref/minres.html\(2022/05/16\)](https://www.mathworks.com/help/matlab/ref/minres.html(2022/05/16)).
- [19] —, “timeit,” Disponible en [https://www.mathworks.com/help/matlab/ref/timeit.html\(2022/08/03\)](https://www.mathworks.com/help/matlab/ref/timeit.html(2022/08/03)).

- [20] —, “Measure the performance of your code,” Disponible en https://www.mathworks.com/help/matlab/matlab_prog/measure-performance-of-your-program.html(2022/08/03).