# Tópicos de Pontos Interiores

Porfirio Suñagua Salgado

IMECC

UNICAMP

Campinas-SP, Maio 2013

# Conteúdo

# Conteúdo

# Conteúdo

# Conteúdo

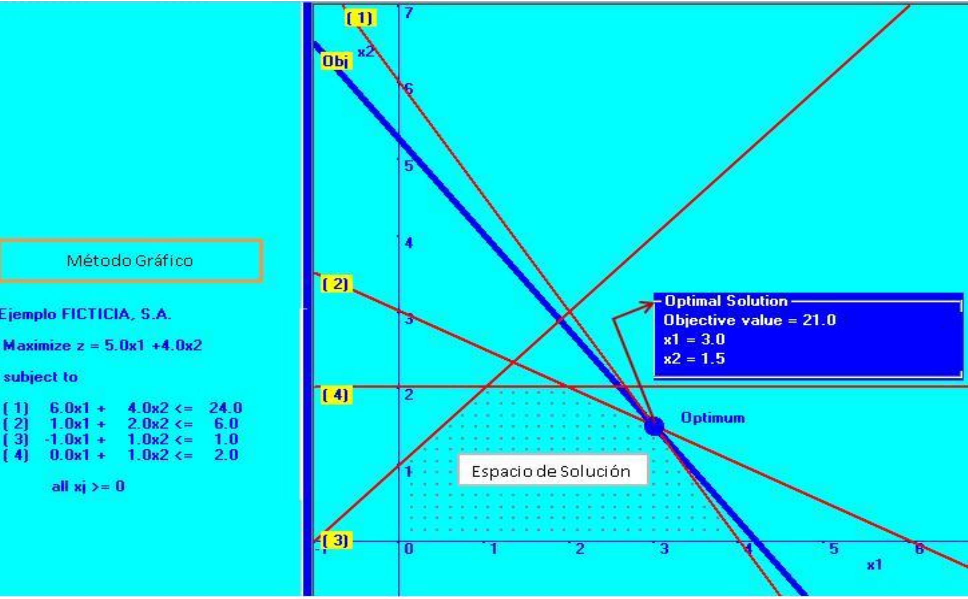
# Conteúdo

# Conteúdo

# Conteúdo

# Conteúdo:

# Solução Factível



# Conteúdo:

# História

- 1967 Primeiro método de pontos interiores, Dikin
- 1985 Primeiro método de Pontos Interiores Polinomial, Karmarkar
- 1986 Método Primal afim escala, redescoberto por varios autores
- 1989 Método Dual afim escala, Adler, Rezende, Veiga, Kamarkar
- 1989 Método Primal-Dual afim escala, Kojima, Mizuno e Yoshise
- 1992 Método Preditor-Corretor, Mehrotra
- 1992 Convergência superlinear de Primal-Dual, Tapia
- 1993 Convergência quadrática de Primal-Dual, Mehrotra
- 1992 Convergência quadrática de Preditor-Corretor, Mehrotra; 1993 Ye e 1994 Potra
- 1996 Método de Múltiplas Correções, Gondzio
- $\downarrow$ Método de Pontos Interiores para PNL
- Evolução do Método Simplex como também de MPI

# História

- 1967 Primeiro método de pontos interiores, Dikin
- 1985 Primeiro método de Pontos Interiores Polinomial, Karmarkar
- 1986 Método Primal afim escala, redescoberto por varios autores
- 1989 Método Dual afim escala, Adler, Rezende, Veiga, Kamarkar
- 1989 Método Primal-Dual afim escala, Kojima, Mizuno e Yoshise
- 1992 Método Preditor-Corretor, Mehrotra
- 1992 Convergência superlinear de Primal-Dual, Tapia
- 1993 Convergência quadrática de Primal-Dual, Mehrotra
- 1992 Convergência quadrática de Preditor-Corretor, Mehrotra; 1993 Ye e 1994 Potra
- 1996 Método de Múltiplas Correções, Gondzio
- ↓ Método de Pontos Interiores para PNL
- Evolução do Método Simplex como também de MPI

# História

- 1967 Primeiro método de pontos interiores, Dikin
- 1985 Primeiro método de Pontos Interiores Polinomial, Karmarkar
- 1986 Método Primal afim escala, redescoberto por varios autores
- 1989 Método Dual afim escala, Adler, Rezende, Veiga, Kamarkar
- 1989 Método Primal-Dual afim escala, Kojima, Mizuno e Yoshise
- 1992 Método Preditor-Corretor, Mehrotra
- 1992 Convergência superlinear de Primal-Dual, Tapia
- 1993 Convergência quadrática de Primal-Dual, Mehrotra
- 1992 Convergência quadrática de Preditor-Corretor, Mehrotra; 1993 Ye e 1994 Potra
- 1996 Método de Múltiplas Correções, Gondzio
- $\downarrow$ Método de Pontos Interiores para PNL
- Evolução do Método Simplex como também de MPI

# História

- 1967 Primeiro método de pontos interiores, Dikin
- 1985 Primeiro método de Pontos Interiores Polinomial, Karmarkar
- 1986 Método Primal afim escala, redescoberto por varios autores
- 1989 Método Dual afim escala, Adler, Rezende, Veiga, Kamarkar
- 1989 Método Primal-Dual afim escala, Kojima, Mizuno e Yoshise
- 1992 Método Preditor-Corretor, Mehrotra
- 1992 Convergência superlinear de Primal-Dual, Tapia
- 1993 Convergência quadrática de Primal-Dual, Mehrotra
- 1992 Convergência quadrática de Preditor-Corretor, Mehrotra; 1993 Ye e 1994 Potra
- 1996 Método de Múltiplas Correções, Gondzio
- ↓ Método de Pontos Interiores para PNL
- Evolução do Método Simplex como também de MPI

# História

- 1967 Primeiro método de pontos interiores, Dikin
- 1985 Primeiro método de Pontos Interiores Polinomial, Karmarkar
- 1986 Método Primal afim escala, redescoberto por varios autores
- 1989 Método Dual afim escala, Adler, Rezende, Veiga, Kamarkar
- 1989 Método Primal-Dual afim escala, Kojima, Mizuno e Yoshise
- 1992 Método Preditor-Corretor, Mehrotra
- 1992 Convergência superlinear de Primal-Dual, Tapia
- 1993 Convergência quadrática de Primal-Dual, Mehrotra
- 1992 Convergência quadrática de Preditor-Corretor, Mehrotra; 1993 Ye e 1994 Potra
- 1996 Método de Múltiplas Correções, Gondzio
- ↓ Método de Pontos Interiores para PNL
- Evolução do Método Simplex como também de MPI

# História

- 1967 Primeiro método de pontos interiores, Dikin
- 1985 Primeiro método de Pontos Interiores Polinomial, Karmarkar
- 1986 Método Primal afim escala, redescoberto por varios autores
- 1989 Método Dual afim escala, Adler, Rezende, Veiga, Kamarkar
- 1989 Método Primal-Dual afim escala, Kojima, Mizuno e Yoshise
- 1992 Método Preditor-Corretor, Mehrotra
- 1992 Convergência superlinear de Primal-Dual, Tapia
- 1993 Convergência quadrática de Primal-Dual, Mehrotra
- 1992 Convergência quadrática de Preditor-Corretor, Mehrotra; 1993 Ye e 1994 Potra
- 1996 Método de Múltiplas Correções, Gondzio
- $\downarrow$ Método de Pontos Interiores para PNL
- Evolução do Método Simplex como também de MPI

# História

- 1967 Primeiro método de pontos interiores, Dikin
- 1985 Primeiro método de Pontos Interiores Polinomial, Karmarkar
- 1986 Método Primal afim escala, redescoberto por varios autores
- 1989 Método Dual afim escala, Adler, Rezende, Veiga, Kamarkar
- 1989 Método Primal-Dual afim escala, Kojima, Mizuno e Yoshise
- 1992 Método Preditor-Corretor, Mehrotra
- 1992 Convergência superlinear de Primal-Dual, Tapia
- 1993 Convergência quadrática de Primal-Dual, Mehrotra
- 1992 Convergência quadrática de Preditor-Corretor, Mehrotra; 1993 Ye e 1994 Potra
- 1996 Método de Múltiplas Correções, Gondzio
- $\downarrow$ Método de Pontos Interiores para PNL
- Evolução do Método Simplex como também de MPI

# História

- 1967 Primeiro método de pontos interiores, Dikin
- 1985 Primeiro método de Pontos Interiores Polinomial, Karmarkar
- 1986 Método Primal afim escala, redescoberto por varios autores
- 1989 Método Dual afim escala, Adler, Rezende, Veiga, Kamarkar
- 1989 Método Primal-Dual afim escala, Kojima, Mizuno e Yoshise
- 1992 Método Preditor-Corretor, Mehrotra
- 1992 Convergência superlinear de Primal-Dual, Tapia
- 1993 Convergência quadrática de Primal-Dual, Mehrotra
- 1992 Convergência quadrática de Preditor-Corretor, Mehrotra; 1993 Ye e 1994 Potra
- 1996 Método de Múltiplas Correções, Gondzio
- $\downarrow$ Método de Pontos Interiores para PNL
- Evolução do Método Simplex como também de MPI

# História

- 1967 Primeiro método de pontos interiores, Dikin
- 1985 Primeiro método de Pontos Interiores Polinomial, Karmarkar
- 1986 Método Primal afim escala, redescoberto por varios autores
- 1989 Método Dual afim escala, Adler, Rezende, Veiga, Kamarkar
- 1989 Método Primal-Dual afim escala, Kojima, Mizuno e Yoshise
- 1992 Método Preditor-Corretor, Mehrotra
- 1992 Convergência superlinear de Primal-Dual, Tapia
- 1993 Convergência quadrática de Primal-Dual, Mehrotra
- 1992 Convergência quadrática de Preditor-Corretor, Mehrotra; 1993 Ye e 1994 Potra
- 1996 Método de Múltiplas Correções, Gondzio
- $\downarrow$ Método de Pontos Interiores para PNL
- Evolução do Método Simplex como também de MPI

# História

- 1967 Primeiro método de pontos interiores, Dikin
- 1985 Primeiro método de Pontos Interiores Polinomial, Karmarkar
- 1986 Método Primal afim escala, redescoberto por varios autores
- 1989 Método Dual afim escala, Adler, Rezende, Veiga, Kamarkar
- 1989 Método Primal-Dual afim escala, Kojima, Mizuno e Yoshise
- 1992 Método Preditor-Corretor, Mehrotra
- 1992 Convergência superlinear de Primal-Dual, Tapia
- 1993 Convergência quadrática de Primal-Dual, Mehrotra
- 1992 Convergência quadrática de Preditor-Corretor, Mehrotra; 1993 Ye e 1994 Potra
- 1996 Método de Múltiplas Correções, Gondzio
- $\downarrow$ Método de Pontos Interiores para PNL
- Evolução do Método Simplex como também de MPI

# História

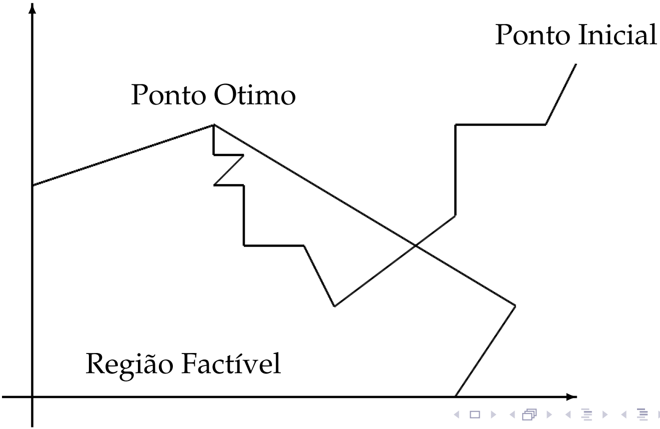
- 1967 Primeiro método de pontos interiores, Dikin
- 1985 Primeiro método de Pontos Interiores Polinomial, Karmarkar
- 1986 Método Primal afim escala, redescoberto por varios autores
- 1989 Método Dual afim escala, Adler, Rezende, Veiga, Kamarkar
- 1989 Método Primal-Dual afim escala, Kojima, Mizuno e Yoshise
- 1992 Método Preditor-Corretor, Mehrotra
- 1992 Convergência superlinear de Primal-Dual, Tapia
- 1993 Convergência quadrática de Primal-Dual, Mehrotra
- 1992 Convergência quadrática de Preditor-Corretor, Mehrotra; 1993 Ye e 1994 Potra
- 1996 Método de Múltiplas Correções, Gondzio
- $\downarrow$ Método de Pontos Interiores para PNL
- Evolução do Método Simplex como também de MPI

# História

- 1967 Primeiro método de pontos interiores, Dikin
- 1985 Primeiro método de Pontos Interiores Polinomial, Karmarkar
- 1986 Método Primal afim escala, redescoberto por varios autores
- 1989 Método Dual afim escala, Adler, Rezende, Veiga, Kamarkar
- 1989 Método Primal-Dual afim escala, Kojima, Mizuno e Yoshise
- 1992 Método Preditor-Corretor, Mehrotra
- 1992 Convergência superlinear de Primal-Dual, Tapia
- 1993 Convergência quadrática de Primal-Dual, Mehrotra
- 1992 Convergência quadrática de Preditor-Corretor, Mehrotra; 1993 Ye e 1994 Potra
- 1996 Método de Múltiplas Correções, Gondzio
- $\downarrow$ Método de Pontos Interiores para PNL
- Evolução do Método Simplex como também de MPI

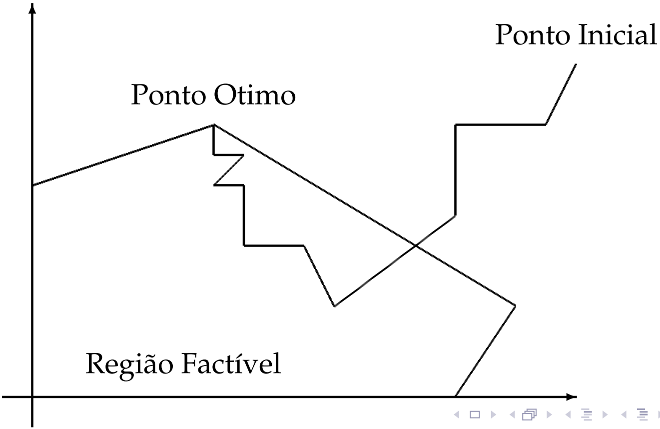
# Comparação: Simplex vs. Pontos Interiores

- Ambos métodos são eficientes na prática
- Simplex: Muitas iterações para convergir, mas elas são baratas
- Pontos Interiores: Poucas iterações, mas eles são caras
- O método de Karmarkar só tem interesse na história



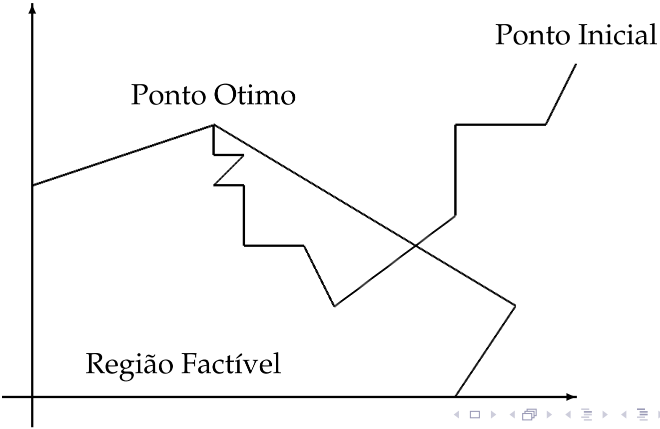
# Comparação: Simplex vs. Pontos Interiores

- Ambos métodos são eficientes na prática
- Simplex: Muitas iterações para convergir, mas elas são baratas
- Pontos Interiores: Poucas iterações, mas eles são caras
- O método de Karmarkar só tem interesse na história



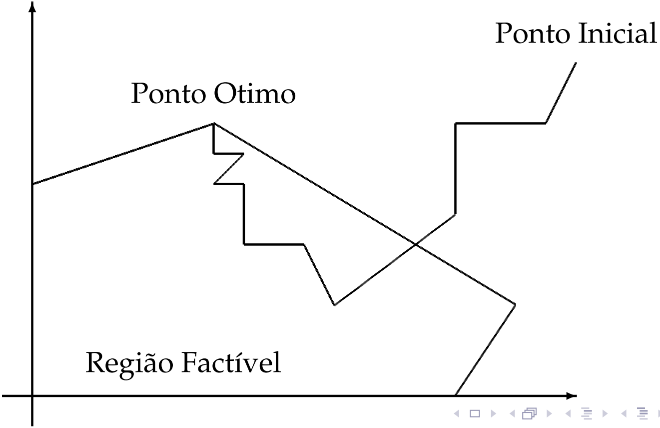
# Comparação: Simplex vs. Pontos Interiores

- Ambos métodos são eficientes na prática
- Simplex: Muitas iterações para convergir, mas elas são baratas
- Pontos Interiores: Poucas iterações, mas eles são caras
- O método de Karmarkar só tem interesse na história



# Comparação: Simplex vs. Pontos Interiores

- Ambos métodos são eficientes na prática
- Simplex: Muitas iterações para convergir, mas elas são baratas
- Pontos Interiores: Poucas iterações, mas eles são caras
- O método de Karmarkar só tem interesse na história



# Problema $PL$ (Primal-Dual)

- Primal - Dual

$$\begin{array}{ll} \text{Min} & c^T x \\ \text{s.a.} & Ax = b \\ & x \geq 0 \end{array} \quad \Leftrightarrow \quad \begin{array}{ll} \text{Max} & b^T y \\ \text{s.a.} & A^T y + z = c \\ & z \geq \mathbf{0} \end{array} \qquad (1)$$

- Para este problema, as condições de otimalidade KKT de primeira ordem são um sistema não linear de equações

$$Ax = b, \quad A^T y + z = c, \quad XZe = 0 \qquad (2)$$

$X = \text{diag}(x)$, $Z = \text{diag}(z)$, $e \in \mathbb{R}^n$ é o vetor de uns, $x \geq 0$ e $z \geq 0$.

- Sistema Aumentado - Sistema Normal:

$$\begin{pmatrix} -D^{-1} & A^T \\ A & 0 \end{pmatrix} \begin{pmatrix} dx \\ dy \end{pmatrix} = \begin{pmatrix} r_1 \\ r_2 \end{pmatrix}, \qquad ADA^T dy = r \qquad (3)$$

$D = Z^{-1}X$ e $dx$, $dy$ direções de Newton

# Problema $PL$ (Primal-Dual)

- Primal - Dual

$$\begin{array}{lll} \text{Min} & c^T x \\ \text{s.a.} & Ax = b \\ & x \geq 0 \end{array} \quad \Leftrightarrow \quad \begin{array}{lll} \text{Max} & b^T y \\ \text{s.a.} & A^T y + z = c \\ & z \geq \mathbf{0} \end{array} \tag{1}$$

- Para este problema, as condições de otimalidade KKT de primeira ordem são um sistema não linear de equações

$$Ax = b, \quad A^T y + z = c, \quad XZe = 0 \tag{2}$$

$X = \text{diag}(x)$, $Z = \text{diag}(z)$, $e \in \mathbb{R}^n$ é o vetor de uns, $x \geq 0$ e $z \geq 0$.

- Sistema Aumentado - Sistema Normal:

$$\begin{pmatrix} -D^{-1} & A^T \\ A & 0 \end{pmatrix} \begin{pmatrix} dx \\ dy \end{pmatrix} = \begin{pmatrix} r_1 \\ r_2 \end{pmatrix}, \qquad ADA^T dy = r \tag{3}$$

$D = Z^{-1}X$ e $dx$, $dy$ direções de Newton

# Problema $PL$ (Primal-Dual)

- Primal - Dual

$$\begin{array}{llcll} \text{Min} & c^T x & & \text{Max} & b^T y \\ \text{s.a.} & Ax = b & \Leftrightarrow & \text{s.a.} & A^T y + z = c \\ & x \geq 0 & & & z \geq \mathbf{0} \end{array} \tag{1}$$

- Para este problema, as condições de otimalidade KKT de primeira ordem são um sistema não linear de equações

$$Ax = b, \quad A^T y + z = c, \quad XZe = 0 \tag{2}$$

$X = \text{diag}(x)$, $Z = \text{diag}(z)$, $e \in \mathbb{R}^n$ é o vetor de uns, $x \geq 0$ e $z \geq 0$.

- Sistema Aumentado - Sistema Normal:

$$\begin{pmatrix} -D^{-1} & A^T \\ A & 0 \end{pmatrix} \begin{pmatrix} dx \\ dy \end{pmatrix} = \begin{pmatrix} r_1 \\ r_2 \end{pmatrix}, \qquad ADA^T dy = r \tag{3}$$

$D = Z^{-1}X$ e $dx, dy$ direções de Newton

# Conteúdo:

# Precondicionador Gondzio et al

- $A = [B, N]$, $D = \mathrm{diag}(D_B, D_N)$, supondo que $D_B^{-1} \approx 0$ e $D_N \approx 0$

$$K = \begin{bmatrix} D_B^{-1} & & B^T \\ & D_N^{-1} & N^T \\ B & N & \end{bmatrix} \approx \begin{bmatrix} & & B^T \\ & D_N^{-1} & N^T \\ B & N & \end{bmatrix}$$

$$ADA^T = BD_BB^T + ND_NN^T \approx BD_BB^T$$

- Precondicionador proposto para o sistema aumentado é

$$P = \begin{bmatrix} & & B^T \\ & D_N^{-1} & N^T \\ B & N & \end{bmatrix}$$

- Para resolver aplica métodos iterativos alternativos MGC

# Precondicionador Gondzio et al

- $A = [B, N]$, $D = \text{diag}(D_B, D_N)$, supondo que $D_B^{-1} \approx 0$ e $D_N \approx 0$

$$K = \begin{bmatrix} D_B^{-1} & & B^T \\ & D_N^{-1} & N^T \\ B & N & \end{bmatrix} \approx \begin{bmatrix} & & B^T \\ & D_N^{-1} & N^T \\ B & N & \end{bmatrix}$$

$$ADA^T = BD_BB^T + ND_NN^T \approx BD_BB^T$$

- Precondicionador proposto para o sistema aumentado é

$$P = \begin{bmatrix} & & B^T \\ & D_N^{-1} & N^T \\ B & N & \end{bmatrix}$$

- Para resolver aplica métodos iterativos alternativos MGC

# Precondicionador Gondzio et al

- $A = [B, N]$, $D = \text{diag}(D_B, D_N)$, supondo que $D_B^{-1} \approx 0$ e $D_N \approx 0$

$$K = \begin{bmatrix} D_B^{-1} & & B^T \\ & D_N^{-1} & N^T \\ B & N & \end{bmatrix} \approx \begin{bmatrix} & & B^T \\ & D_N^{-1} & N^T \\ B & N & \end{bmatrix}$$

$$ADA^T = BD_BB^T + ND_NN^T \approx BD_BB^T$$

- Precondicionador proposto para o sistema aumentado é

$$P = \begin{bmatrix} & & B^T \\ & D_N^{-1} & N^T \\ B & N & \end{bmatrix}$$

- Para resolver aplica métodos iterativos alternativos MGC

# Precondicionador Bergamaschi et al

- Considera problema quadrático $+\frac{1}{2}x^TQx$ na funcção objetivo $E = -(\text{diag}(Q) + D^{-1})$, Cholesky $AE^{-1}A^T = L_0D_0L_0^T$

$$P_2 = \begin{bmatrix} E & A^T \\ A & 0 \end{bmatrix} = \begin{bmatrix} I & 0 \\ AE^{-1} & I \end{bmatrix} \begin{bmatrix} E & 0 \\ 0 & -AE^{-1}A^T \end{bmatrix} \begin{bmatrix} I & E^{-1}A^T \\ 0 & I \end{bmatrix}$$

$$= \begin{bmatrix} I & 0 \\ AE^{-1} & L_0 \end{bmatrix} \begin{bmatrix} E & 0 \\ 0 & -D_0 \end{bmatrix} \begin{bmatrix} I & E^{-1}A^T \\ 0 & L_0^T \end{bmatrix}$$

- Precondicionador é a inversa de $P_2$.

# Precondicionador Bergamaschi et al

- Considera problema quadrático $+\frac{1}{2}x^TQx$ na funcção objetivo $E = -(\text{diag}(Q) + D^{-1})$, Cholesky $AE^{-1}A^T = L_0D_0L_0^T$

$$P_2 = \begin{bmatrix} E & A^T \\ A & 0 \end{bmatrix} = \begin{bmatrix} I & 0 \\ AE^{-1} & I \end{bmatrix} \begin{bmatrix} E & 0 \\ 0 & -AE^{-1}A^T \end{bmatrix} \begin{bmatrix} I & E^{-1}A^T \\ 0 & I \end{bmatrix}$$

$$= \begin{bmatrix} I & 0 \\ AE^{-1} & L_0 \end{bmatrix} \begin{bmatrix} E & 0 \\ 0 & -D_0 \end{bmatrix} \begin{bmatrix} I & E^{-1}A^T \\ 0 & L_0^T \end{bmatrix}$$

- Precondicionador é a inversa de $P_2$.

# Precondicionador Joo-Siong Chai e Kim-Chuan Toh

- $PL$ canalizado: $D = X^{-1}Z + V^{-1}W$

$$\begin{bmatrix} -D & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} g \\ r_p \end{bmatrix}$$

- Seja $D = \text{diag}(D_1, D_2)$, tal que $D_1/\mu = \mathcal{O}(1)$, $\mu D_2 = \mathcal{O}(1)$, $\mu \ll 1$. Partição $A = [A_1, A_2]$, $\Delta x = [\Delta x_1, \Delta x_2]^T$, $g = [g_1, g_2]^T$.
- Sistema Reduzido: $E_1 > 0$ (diagonal), $\Delta x_2 = D_2^{-1}(A_2^T \Delta y - g_2)$.

$$K = \begin{bmatrix} H & B \\ B^T & -\Psi \end{bmatrix} \begin{bmatrix} \Delta y \\ \Delta \tilde{x}_1 \end{bmatrix} = \begin{bmatrix} h \\ F_1^{-1} g_1 \end{bmatrix}$$

$F_1 = E_1 + D_1$, e

$$\Delta \tilde{x}_1 = F_1^{-1} E_1 \Delta x_1 \qquad \Psi = D_1 E_1^{-1}$$

$$H = A\text{diag}(F_1^{-1}, D_2^{-1})A^T \qquad B = A_1 F_1^{-1/2}$$

$$h = r_p + A\text{diag}(F_1^{-1}, D_2^{-1})g$$

# Precondicionador Joo-Siong Chai e Kim-Chuan Toh

- $PL$ canalizado: $D = X^{-1}Z + V^{-1}W$

$$\begin{bmatrix} -D & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} g \\ r_p \end{bmatrix}$$

- Seja $D = \mathrm{diag}(D_1, D_2)$, tal que $D_1/\mu = \mathcal{O}(1)$, $\mu D_2 = \mathcal{O}(1)$, $\mu \ll 1$. Partição $A = [A_1, A_2]$, $\Delta x = [\Delta x_1, \Delta x_2]^T$, $g = [g_1, g_2]^T$.
- Sistema Reduzido: $E_1 > 0$ (diagonal), $\Delta x_2 = D_2^{-1}(A_2^T \Delta y - g_2)$.

$$K = \begin{bmatrix} H & B \\ B^T & -\Psi \end{bmatrix} \begin{bmatrix} \Delta y \\ \Delta \tilde{x}_1 \end{bmatrix} = \begin{bmatrix} h \\ F_1^{-1} g_1 \end{bmatrix}$$

$F_1 = E_1 + D_1$, e

$$\Delta \tilde{x}_1 = F_1^{-1} E_1 \Delta x_1 \qquad \Psi = D_1 E_1^{-1}$$

$$H = A\mathrm{diag}(F_1^{-1}, D_2^{-1}) A^T \qquad B = A_1 F_1^{-1/2}$$

$$h = r_p + A\mathrm{diag}(F_1^{-1}, D_2^{-1}) g$$

# Precondicionador Joo-Siong Chai e Kim-Chuan Toh

- $PL$ canalizado: $D = X^{-1}Z + V^{-1}W$

$$\begin{bmatrix} -D & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} g \\ r_p \end{bmatrix}$$

- Seja $D = \text{diag}(D_1, D_2)$, tal que $D_1/\mu = \mathcal{O}(1)$, $\mu D_2 = \mathcal{O}(1)$, $\mu \ll 1$. Partição $A = [A_1, A_2]$, $\Delta x = [\Delta x_1, \Delta x_2]^T$, $g = [g_1, g_2]^T$.
- Sistema Reduzido: $E_1 > 0$ (diagonal), $\Delta x_2 = D_2^{-1}(A_2^T \Delta y - g_2)$.

$$K = \begin{bmatrix} H & B \\ B^T & -\Psi \end{bmatrix} \begin{bmatrix} \Delta y \\ \Delta \tilde{x}_1 \end{bmatrix} = \begin{bmatrix} h \\ F_1^{-1} g_1 \end{bmatrix}$$

$F_1 = E_1 + D_1$, e

$$\Delta \tilde{x}_1 = F_1^{-1} E_1 \Delta x_1 \qquad \Psi = D_1 E_1^{-1}$$

$$H = A\text{diag}(F_1^{-1}, D_2^{-1})A^T \qquad B = A_1 F_1^{-1/2}$$

$$h = r_p + A\text{diag}(F_1^{-1}, D_2^{-1})g$$

# Cont. Precondicionador de Chai-Toh

- A inversa da matriz de coeficientes do sistema reduzido é

$$K^{-1} = \begin{bmatrix} H^{-1/2}(I-P)H^{-1/2} & H^{-1}BS^{-1} \\ S^{-1}B^TH^{-1} & -S^{-1} \end{bmatrix}$$

$S = B^TH^{-1}B + \Psi$, e $P = H^{-1/2}BS^{-1}B^TH^{-1/2}$ satisfaz $0 \preceq P \preceq I$.

- Precondicionador proposto baseado em sua inversa é

$$P_c^{-1} = \begin{bmatrix} \hat{H}^{-1} - \hat{H}^{-1}B\hat{S}^{-1}B^T\hat{H}^{-1} & \hat{H}^{-1}B\hat{S}^{-1} \\ \hat{S}^{-1}B^T\hat{H}^{-1} & -\hat{S}^{-1} \end{bmatrix}$$

$\hat{H}$ e $\hat{S}$ são definidas positivas como aproximações de $H$ e $S$ respectivamente tal que $\hat{S} = B^T\hat{H}^{-1}B + \Psi$.

# Cont. Precondicionador de Chai-Toh

- A inversa da matriz de coeficientes do sistema reduzido é

$$K^{-1} = \begin{bmatrix} H^{-1/2}(I-P)H^{-1/2} & H^{-1}BS^{-1} \\ S^{-1}B^TH^{-1} & -S^{-1} \end{bmatrix}$$

$S = B^TH^{-1}B + \Psi$, e $P = H^{-1/2}BS^{-1}B^TH^{-1/2}$ satisfaz $0 \preceq P \preceq I$.

- Precondicionador proposto baseado em sua inversa é

$$P_c^{-1} = \begin{bmatrix} \hat{H}^{-1} - \hat{H}^{-1}B\hat{S}^{-1}B^T\hat{H}^{-1} & \hat{H}^{-1}B\hat{S}^{-1} \\ \hat{S}^{-1}B^T\hat{H}^{-1} & -\hat{S}^{-1} \end{bmatrix}$$

$\hat{H}$ e $\hat{S}$ são definidas positivas como aproximações de $H$ e $S$ respectivamente tal que $\hat{S} = B^T\hat{H}^{-1}B + \Psi$.

# Precondicionador A.R.L. Oliveira e D.C. Sorensen

- Ordena colunas pela norma de $AD$, logo obter base por fatoração $LU$ rectangular
- Precondicionador Separador

$$M^{-1} = \begin{pmatrix} D_B^{1/2} & 0 & D_B^{-1/2}B^{-1} \\ 0 & D_N^{1/2} & 0 \\ D_B^{1/2} & 0 & 0 \end{pmatrix}$$

- Resolver um sistema com matriz de coeficientes $T = I + D_B^{-1/2}B^{-1}ND_NN^TB^{-T}D_B^{-1/2} \approx I$
- Método de Gradientes Conjugados

# Precondicionador A.R.L. Oliveira e D.C. Sorensen

- Ordena colunas pela norma de $AD$, logo obter base por fatoração $LU$ rectangular
- Precondicionador Separador

$$M^{-1} = \begin{pmatrix} D_B^{1/2} & 0 & D_B^{-1/2}B^{-1} \\ 0 & D_N^{1/2} & 0 \\ D_B^{1/2} & 0 & 0 \end{pmatrix}$$

- Resolver um sistema com matriz de coeficientes $T = I + D_B^{-1/2}B^{-1}ND_NN^TB^{-T}D_B^{-1/2} \approx I$
- Método de Gradientes Conjugados

# Precondicionador A.R.L. Oliveira e D.C. Sorensen

- Ordena colunas pela norma de $AD$, logo obter base por fatoração $LU$ rectangular
- Precondicionador Separador

$$M^{-1} = \begin{pmatrix} D_B^{1/2} & 0 & D_B^{-1/2}B^{-1} \\ 0 & D_N^{1/2} & 0 \\ D_B^{1/2} & 0 & 0 \end{pmatrix}$$

- Resolver um sistema com matriz de coeficientes $T = I + D_B^{-1/2}B^{-1}ND_NN^TB^{-T}D_B^{-1/2} \approx I$
- Método de Gradientes Conjugados

# Precondicionador A.R.L. Oliveira e D.C. Sorensen

- Ordena colunas pela norma de $AD$, logo obter base por fatoração $LU$ rectangular
- Precondicionador Separador

$$M^{-1} = \begin{pmatrix} D_B^{1/2} & 0 & D_B^{-1/2}B^{-1} \\ 0 & D_N^{1/2} & 0 \\ D_B^{1/2} & 0 & 0 \end{pmatrix}$$

- Resolver um sistema com matriz de coeficientes $T = I + D_B^{-1/2}B^{-1}ND_NN^TB^{-T}D_B^{-1/2} \approx I$
- Método de Gradientes Conjugados

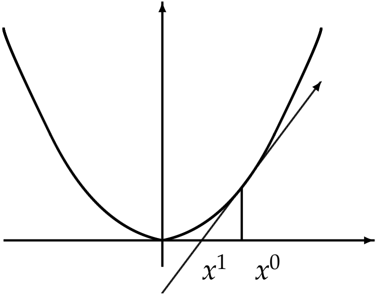
# Conteúdo:

# Método de Newton

- Dado $\mathbf{F} : D_F \subset \mathbb{R}^n \to \mathbb{R}^n$. Equação a resolver $\mathbf{F}(\mathbf{x}) = \mathbf{0}$
- Taylor:

$$\mathbf{F}(\mathbf{x}) \approx \mathbf{F}(\mathbf{x}^0) + \mathbf{J}(\mathbf{x}^0)(\mathbf{x} - \mathbf{x}^0), \qquad \mathbf{J}(\mathbf{x}) = \nabla F(x) \quad \text{(Jacobiana de } \mathbf{F}\text{)}$$

- $x^1 = x^0 + \mathbf{d}, \qquad \mathbf{d} = -\mathbf{J}(\mathbf{x}^0)\mathbf{F}(\mathbf{x}^0)$
- $\mathbf{J}(\mathbf{x}_0)\mathbf{d} = -\mathbf{F}(\mathbf{x}_0)$

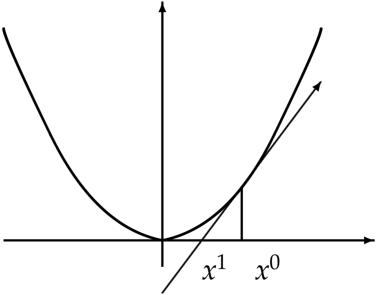


# Método de Newton

- Dado $\mathbf{F} : D_F \subset \mathbb{R}^n \to \mathbb{R}^n$. Equação a resolver $\mathbf{F}(\mathbf{x}) = \mathbf{0}$
- Taylor:

$$\mathbf{F}(\mathbf{x}) \approx \mathbf{F}(\mathbf{x^0}) + \mathbf{J}(\mathbf{x^0})(\mathbf{x} - \mathbf{x^0}), \qquad \mathbf{J}(\mathbf{x}) = \nabla F(x) \quad \text{(Jacobiana de } \mathbf{F}\text{)}$$

- $x^1 = x^0 + \mathbf{d}, \qquad \mathbf{d} = -\mathbf{J}(\mathbf{x^0})\mathbf{F}(\mathbf{x^0})$
- $\mathbf{J}(\mathbf{x_0})\mathbf{d} = -\mathbf{F}(\mathbf{x_0})$

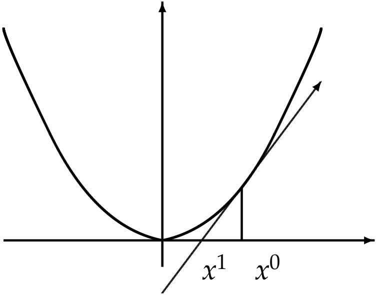


# Método de Newton

- Dado $\mathbf{F} : D_F \subset \mathbb{R}^n \to \mathbb{R}^n$. Equação a resolver $\mathbf{F}(\mathbf{x}) = \mathbf{0}$
- Taylor:

$$\mathbf{F}(\mathbf{x}) \approx \mathbf{F}(\mathbf{x^0}) + \mathbf{J}(\mathbf{x^0})(\mathbf{x} - \mathbf{x^0}), \qquad \mathbf{J}(\mathbf{x}) = \nabla F(x) \quad (\text{Jacobiana de } \mathbf{F})$$

- $x^1 = x^0 + \mathbf{d}, \qquad \mathbf{d} = -\mathbf{J}(\mathbf{x^0})\mathbf{F}(\mathbf{x^0})$
- $\mathbf{J}(\mathbf{x_0})\mathbf{d} = -\mathbf{F}(\mathbf{x_0})$

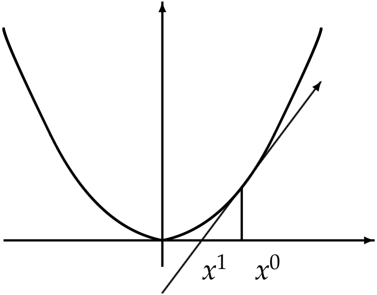


# Método de Newton

- Dado $\mathbf{F} : D_F \subset \mathbb{R}^n \to \mathbb{R}^n$. Equação a resolver $\mathbf{F}(\mathbf{x}) = \mathbf{0}$
- Taylor:

$$\mathbf{F}(\mathbf{x}) \approx \mathbf{F}(\mathbf{x^0}) + \mathbf{J}(\mathbf{x^0})(\mathbf{x} - \mathbf{x^0}), \qquad \mathbf{J}(\mathbf{x}) = \nabla F(x) \quad \text{(Jacobiana de } \mathbf{F}\text{)}$$

- $x^1 = x^0 + \mathbf{d}, \qquad \mathbf{d} = -\mathbf{J}(\mathbf{x^0})\mathbf{F}(\mathbf{x^0})$
- $\mathbf{J}(\mathbf{x_0})\mathbf{d} = -\mathbf{F}(\mathbf{x_0})$



# Sistema Aumentado, Sistema Normal

- Resolvendo Sistema KKT:
  $r_p = b - Ax^0, \quad r_d = c - A^T y^0 - z^0, \quad r_a = -X^0 Z^0 e$

$$\begin{bmatrix} A & 0 & 0 \\ 0 & A^T & I \\ Z & 0 & X \end{bmatrix} \begin{bmatrix} dx \\ dy \\ dz \end{bmatrix} = \begin{bmatrix} r_p \\ r_d \\ r_a \end{bmatrix} \qquad \begin{array}{rcl} Adx & = & r_p \\ A^T dy + dz & = & r_d \\ Zdx + Xdz & = & r_a \end{array} \tag{4}$$

- $Zdx + Xdz = r_a \quad \Rightarrow \quad dz = X^{-1}(r_a - Zdx)$
- $A^T dy + X^{-1} r_a - X^{-1} Z dx = r_d \quad \Rightarrow \quad A^T dy - X^{-1} Z dx = r_d - X^{-1} r_a$
- Sistema Aumentado

$$\begin{array}{rcl} A^T dy - X^{-1} Z dx & = & r_d - X^{-1} r_a \\ Adx & = & r_p \end{array} \tag{5}$$

$$\begin{pmatrix} -D^{-1} & A^T \\ A & 0 \end{pmatrix} \begin{pmatrix} dx \\ dy \end{pmatrix} = \begin{pmatrix} r_d - X^{-1} r_a \\ r_p \end{pmatrix}, \quad D = Z^{-1} X \tag{6}$$

- Sistema Normal $\quad ADA^T dy = r_p + AD(r_d - X^{-1} r_a)$

# Sistema Aumentado, Sistema Normal

- Resolvendo Sistema KKT:
  $r_p = b - Ax^0, \quad r_d = c - A^T y^0 - z^0, \quad r_a = -X^0 Z^0 e$

$$\begin{bmatrix} A & 0 & 0 \\ 0 & A^T & I \\ Z & 0 & X \end{bmatrix} \begin{bmatrix} dx \\ dy \\ dz \end{bmatrix} = \begin{bmatrix} r_p \\ r_d \\ r_a \end{bmatrix} \qquad \begin{aligned} Adx &= r_p \\ A^T dy + dz &= r_d \\ Zdx + Xdz &= r_a \end{aligned} \tag{4}$$

- $Zdx + Xdz = r_a \quad \Rightarrow \quad dz = X^{-1}(r_a - Zdx)$
- $A^T dy + X^{-1} r_a - X^{-1} Z dx = r_d \quad \Rightarrow \quad A^T dy - X^{-1} Z dx = r_d - X^{-1} r_a$
- Sistema Aumentado

$$\begin{aligned} A^T dy - X^{-1} Z dx &= r_d - X^{-1} r_a \\ Adx &= r_p \end{aligned} \tag{5}$$

$$\begin{pmatrix} -D^{-1} & A^T \\ A & 0 \end{pmatrix} \begin{pmatrix} dx \\ dy \end{pmatrix} = \begin{pmatrix} r_d - X^{-1} r_a \\ r_p \end{pmatrix}, \quad D = Z^{-1} X \tag{6}$$

- Sistema Normal $\quad ADA^T dy = r_p + AD(r_d - X^{-1} r_a)$

# Sistema Aumentado, Sistema Normal

- Resolvendo Sistema KKT:
  $r_p = b - Ax^0, \quad r_d = c - A^T y^0 - z^0, \quad r_a = -X^0 Z^0 e$

$$\begin{bmatrix} A & 0 & 0 \\ 0 & A^T & I \\ Z & 0 & X \end{bmatrix} \begin{bmatrix} dx \\ dy \\ dz \end{bmatrix} = \begin{bmatrix} r_p \\ r_d \\ r_a \end{bmatrix} \qquad \begin{array}{rcl} Adx & = & r_p \\ A^T dy + dz & = & r_d \\ Zdx + Xdz & = & r_a \end{array} \tag{4}$$

- $Zdx + Xdz = r_a \quad \Rightarrow \quad dz = X^{-1}(r_a - Zdx)$
- $A^T dy + X^{-1} r_a - X^{-1} Z dx = r_d \quad \Rightarrow \quad A^T dy - X^{-1} Z dx = r_d - X^{-1} r_a$
- Sistema Aumentado

$$\begin{array}{rcl} A^T dy - X^{-1} Z dx & = & r_d - X^{-1} r_a \\ Adx & = & r_p \end{array} \tag{5}$$

$$\begin{pmatrix} -D^{-1} & A^T \\ A & 0 \end{pmatrix} \begin{pmatrix} dx \\ dy \end{pmatrix} = \begin{pmatrix} r_d - X^{-1} r_a \\ r_p \end{pmatrix}, \quad D = Z^{-1} X \tag{6}$$

- Sistema Normal $\quad ADA^T dy = r_p + AD(r_d - X^{-1} r_a)$

# Sistema Aumentado, Sistema Normal

- Resolvendo Sistema KKT:
  $r_p = b - Ax^0, \quad r_d = c - A^T y^0 - z^0, \quad r_a = -X^0 Z^0 e$

$$\begin{bmatrix} A & 0 & 0 \\ 0 & A^T & I \\ Z & 0 & X \end{bmatrix} \begin{bmatrix} dx \\ dy \\ dz \end{bmatrix} = \begin{bmatrix} r_p \\ r_d \\ r_a \end{bmatrix} \qquad \begin{array}{rcl} Adx & = & r_p \\ A^T dy + dz & = & r_d \\ Zdx + Xdz & = & r_a \end{array} \tag{4}$$

- $Zdx + Xdz = r_a \quad \Rightarrow \quad dz = X^{-1}(r_a - Zdx)$
- $A^T dy + X^{-1} r_a - X^{-1} Z dx = r_d \quad \Rightarrow \quad A^T dy - X^{-1} Z dx = r_d - X^{-1} r_a$
- Sistema Aumentado

$$\begin{array}{rcl} A^T dy - X^{-1} Z dx & = & r_d - X^{-1} r_a \\ Adx & = & r_p \end{array} \tag{5}$$

$$\begin{pmatrix} -D^{-1} & A^T \\ A & 0 \end{pmatrix} \begin{pmatrix} dx \\ dy \end{pmatrix} = \begin{pmatrix} r_d - X^{-1} r_a \\ r_p \end{pmatrix}, \qquad D = Z^{-1} X \tag{6}$$

- Sistema Normal $\quad ADA^T dy = r_p + AD(r_d - X^{-1} r_a)$

# Sistema Aumentado, Sistema Normal

- Resolvendo Sistema KKT:
  $r_p = b - Ax^0, \quad r_d = c - A^Ty^0 - z^0, \quad r_a = -X^0Z^0e$

$$\begin{bmatrix} A & 0 & 0 \\ 0 & A^T & I \\ Z & 0 & X \end{bmatrix} \begin{bmatrix} dx \\ dy \\ dz \end{bmatrix} = \begin{bmatrix} r_p \\ r_d \\ r_a \end{bmatrix} \qquad \begin{array}{rcl} Adx & = & r_p \\ A^Tdy + dz & = & r_d \\ Zdx + Xdz & = & r_a \end{array} \tag{4}$$

- $Zdx + Xdz = r_a \quad \Rightarrow \quad dz = X^{-1}(r_a - Zdx)$
- $A^Tdy + X^{-1}r_a - X^{-1}Zdx = r_d \quad \Rightarrow \quad A^Tdy - X^{-1}Zdx = r_d - X^{-1}r_a$
- Sistema Aumentado

$$\begin{array}{rcl} A^Tdy - X^{-1}Zdx & = & r_d - X^{-1}r_a \\ Adx & = & r_p \end{array} \tag{5}$$

$$\begin{pmatrix} -D^{-1} & A^T \\ A & 0 \end{pmatrix} \begin{pmatrix} dx \\ dy \end{pmatrix} = \begin{pmatrix} r_d - X^{-1}r_a \\ r_p \end{pmatrix}, \qquad D = Z^{-1}X \tag{6}$$

- Sistema Normal $\quad ADA^Tdy = r_p + AD(r_d - X^{-1}r_a)$

# Algoritmo Método Primal–Dual Afim Escala

**Require:** $x^0, y^0, z^0$ tal que $x^0 > 0, z^0 > 0$

$\tau = 0.99995, Maxit = 100$

**while** $k = 0, 1, 2, \cdots, Maxit$ e Condição de Parada **do**

$r_p = b - Ax$

$r_d = c - A^T y - z$

$r_a = XZe$

$D = Z^{-1}X$

Resolver $ADA^T dy = r_p + AD(r_d - X^{-1}r_a)$

$dx = D(A^T dy - r_d + X^{-1}r_a)$

$dz = X^{-1}(r_a - Zdx)$

Calcular o tamanho de passo $\alpha$

$x^{k+1} \leftarrow x^k + \alpha dx, \quad y^{k+1} \leftarrow y^k + \alpha dy, \quad z^{k+1} \leftarrow z^k + \alpha dz$

Avaliar/Testar Condição de Parada

$k \leftarrow k + 1$

**end while**

# Algoritmo Método Primal–Dual Afim Escala

**Require:** $x^0, y^0, z^0$ tal que $x^0 > 0, z^0 > 0$

$\tau = 0.99995, Maxit = 100$

**while** $k = 0, 1, 2, \cdots, Maxit$ e Condição de Parada **do**

$r_p = b - Ax$

$r_d = c - A^T y - z$

$r_a = -XZe$

$D = Z^{-1}X$

Resolver $ADA^T dy = r_p + AD(r_d - X^{-1} r_a)$

$dx = D(A^T dy - r_d + X^{-1} r_a)$

$dz = X^{-1}(r_a - Z dx)$

Calcular o tamanho de passo $\alpha$

$x^{k+1} \leftarrow x^k + \alpha dx, \quad y^{k+1} \leftarrow y^k + \alpha dy, \quad z^{k+1} \leftarrow z^k + \alpha dz$

Avaliar/Testar Condição de Parada

$k \leftarrow k + 1$

**end while**

# Algoritmo Método Primal–Dual Afim Escala

**Require:** $x^0, y^0, z^0$ tal que $x^0 > 0, z^0 > 0$

$\tau = 0.99995$, $Maxit = 100$

**while** $k = 0, 1, 2, \cdots, Maxit$ e Condição de Parada **do**

$r_p = b - Ax$

$r_d = c - A^T y - z$

$r_a = XZe$

$D = Z^{-1}X$

Resolver $ADA^T dy = r_p + AD(r_d - X^{-1}r_a)$

$dx = D(A^T dy - r_d + X^{-1}r_a)$

$dz = X^{-1}(r_a - Zdx)$

Calcular o tamanho de paso $\alpha$

$x^{k+1} \leftarrow x^k + \alpha dx, \quad y^{k+1} \leftarrow y^k + \alpha dy, \quad z^{k+1} \leftarrow z^k + \alpha dz$

Avaliar/Testar Condição de Parada

$k \leftarrow k + 1$

**end while**

# Algoritmo Método Primal–Dual Afim Escala

**Require:** $x^0, y^0, z^0$ tal que $x^0 > 0, z^0 > 0$

$\tau = 0.99995$, $Maxit = 100$

**while** $k = 0, 1, 2, \cdots, Maxit$ e Condição de Parada **do**

$r_p = b - Ax$

$r_d = c - A^T y - z$

$r_a = XZe$

$D = Z^{-1}X$

Resolver $ADA^T dy = r_p + AD(r_d - X^{-1} r_a)$

$dx = D(A^T dy - r_d + X^{-1} r_a)$

$dz = X^{-1}(r_a - Zdx)$

Calcular o tamanho de paso $\alpha$

$x^{k+1} \leftarrow x^k + \alpha dx, \quad y^{k+1} \leftarrow y^k + \alpha dy, \quad z^{k+1} \leftarrow z^k + \alpha dz$

Avaliar/Testar Condição de Parada

$k \leftarrow k + 1$

**end while**

# Algoritmo Método Primal–Dual Afim Escala

**Require:** $x^0, y^0, z^0$ tal que $x^0 > 0, z^0 > 0$

$\tau = 0.99995$, $Maxit = 100$

**while** $k = 0, 1, 2, \cdots, Maxit$ e Condição de Parada **do**

$r_p = b - Ax$

$r_d = c - A^T y - z$

$r_a = XZe$

$D = Z^{-1}X$

Resolver $ADA^T dy = r_p + AD(r_d - X^{-1}r_a)$

$dx = D(A^T dy - r_d + X^{-1}r_a)$

$dz = X^{-1}(r_a - Zdx)$

Calcular o tamanho de paso $\alpha$

$x^{k+1} \leftarrow x^k + \alpha dx, \quad y^{k+1} \leftarrow y^k + \alpha dy, \quad z^{k+1} \leftarrow z^k + \alpha dz$

Avaliar/Testar Condição de Parada

$k \leftarrow k + 1$

**end while**

# Algoritmo Método Primal–Dual Afim Escala

**Require:** $x^0, y^0, z^0$ tal que $x^0 > 0, z^0 > 0$

$\tau = 0.99995, Maxit = 100$

**while** $k = 0, 1, 2, \cdots, Maxit$ e Condição de Parada **do**

$r_p = b - Ax$

$r_d = c - A^T y - z$

$r_a = XZe$

$D = Z^{-1}X$

Resolver $ADA^T dy = r_p + AD(r_d - X^{-1}r_a)$

$dx = D(A^T dy - r_d + X^{-1}r_a)$

$dz = X^{-1}(r_a - Zdx)$

Calcular o tamanho de paso $\alpha$

$x^{k+1} \leftarrow x^k + \alpha dx, \quad y^{k+1} \leftarrow y^k + \alpha dy, \quad z^{k+1} \leftarrow z^k + \alpha dz$

Avaliar/Testar Condição de Parada

$k \leftarrow k + 1$

**end while**

# Algoritmo Método Primal–Dual Afim Escala

**Require:** $x^0, y^0, z^0$ tal que $x^0 > 0, z^0 > 0$

$\tau = 0.99995, Maxit = 100$

**while** $k = 0, 1, 2, \cdots, Maxit$ e Condição de Parada **do**

$r_p = b - Ax$

$r_d = c - A^T y - z$

$r_a = XZe$

$D = Z^{-1}X$

Resolver $ADA^T dy = r_p + AD(r_d - X^{-1}r_a)$

$dx = D(A^T dy - r_d + X^{-1}r_a)$

$dz = X^{-1}(r_a - Zdx)$

Calcular o tamanho de paso $\alpha$

$x^{k+1} \leftarrow x^k + \alpha dx, \quad y^{k+1} \leftarrow y^k + \alpha dy, \quad z^{k+1} \leftarrow z^k + \alpha dz$

Avaliar/Testar Condição de Parada

$k \leftarrow k + 1$

**end while**

# Algoritmo Método Primal–Dual Afim Escala

**Require:** $x^0, y^0, z^0$ tal que $x^0 > 0, z^0 > 0$

$\tau = 0.99995, Maxit = 100$

**while** $k = 0, 1, 2, \cdots, Maxit$ e Condição de Parada **do**

$r_p = b - Ax$

$r_d = c - A^T y - z$

$r_a = XZe$

$D = Z^{-1}X$

Resolver $ADA^T dy = r_p + AD(r_d - X^{-1}r_a)$

$dx = D(A^T dy - r_d + X^{-1}r_a)$

$dz = X^{-1}(r_a - Zdx)$

Calcular o tamanho de paso $\alpha$

$x^{k+1} \leftarrow x^k + \alpha dx, \quad y^{k+1} \leftarrow y^k + \alpha dy, \quad z^{k+1} \leftarrow z^k + \alpha dz$

Avaliar/Testar Condição de Parada

$k \leftarrow k + 1$

**end while**

# Algoritmo Método Primal–Dual Afim Escala

**Require:** $x^0, y^0, z^0$ tal que $x^0 > 0, z^0 > 0$

$\tau = 0.99995, Maxit = 100$

**while** $k = 0, 1, 2, \cdots, Maxit$ e Condição de Parada **do**

$r_p = b - Ax$

$r_d = c - A^T y - z$

$r_a = XZe$

$D = Z^{-1}X$

Resolver $ADA^T dy = r_p + AD(r_d - X^{-1} r_a)$

$dx = D(A^T dy - r_d + X^{-1} r_a)$

$dz = X^{-1}(r_a - Zdx)$

Calcular o tamanho de paso $\alpha$

$x^{k+1} \leftarrow x^k + \alpha dx, \quad y^{k+1} \leftarrow y^k + \alpha dy, \quad z^{k+1} \leftarrow z^k + \alpha dz$

Avaliar/Testar Condição de Parada

$k \leftarrow k + 1$

**end while**

# Algoritmo Método Primal–Dual Afim Escala

**Require:** $x^0, y^0, z^0$ tal que $x^0 > 0, z^0 > 0$

$\tau = 0.99995, Maxit = 100$

**while** $k = 0, 1, 2, \cdots, Maxit$ e Condição de Parada **do**

$r_p = b - Ax$

$r_d = c - A^T y - z$

$r_a = XZe$

$D = Z^{-1}X$

Resolver $ADA^T dy = r_p + AD(r_d - X^{-1}r_a)$

$dx = D(A^T dy - r_d + X^{-1}r_a)$

$dz = X^{-1}(r_a - Zdx)$

Calcular o tamanho de paso $\alpha$

$x^{k+1} \leftarrow x^k + \alpha dx, \quad y^{k+1} \leftarrow y^k + \alpha dy, \quad z^{k+1} \leftarrow z^k + \alpha dz$

Avaliar/Testar Condição de Parada

$k \leftarrow k + 1$

**end while**

# Algoritmo Método Primal–Dual Afim Escala

**Require:** $x^0, y^0, z^0$ tal que $x^0 > 0, z^0 > 0$

$\tau = 0.99995$, $Maxit = 100$

**while** $k = 0, 1, 2, \cdots, Maxit$ e Condição de Parada **do**

$r_p = b - Ax$

$r_d = c - A^T y - z$

$r_a = XZe$

$D = Z^{-1}X$

Resolver $ADA^T dy = r_p + AD(r_d - X^{-1}r_a)$

$dx = D(A^T dy - r_d + X^{-1}r_a)$

$dz = X^{-1}(r_a - Zdx)$

Calcular o tamanho de paso $\alpha$

$x^{k+1} \leftarrow x^k + \alpha dx, \quad y^{k+1} \leftarrow y^k + \alpha dy, \quad z^{k+1} \leftarrow z^k + \alpha dz$

Avaliar/Testar Condição de Parada

$k \leftarrow k + 1$

**end while**

# Algoritmo Método Primal–Dual Afim Escala

**Require:** $x^0, y^0, z^0$ tal que $x^0 > 0, z^0 > 0$

$\tau = 0.99995$, $Maxit = 100$

**while** $k = 0, 1, 2, \cdots, Maxit$ e Condição de Parada **do**

$r_p = b - Ax$

$r_d = c - A^T y - z$

$r_a = XZe$

$D = Z^{-1}X$

Resolver $ADA^T dy = r_p + AD(r_d - X^{-1}r_a)$

$dx = D(A^T dy - r_d + X^{-1}r_a)$

$dz = X^{-1}(r_a - Zdx)$

Calcular o tamanho de paso $\alpha$

$x^{k+1} \leftarrow x^k + \alpha dx, \quad y^{k+1} \leftarrow y^k + \alpha dy, \quad z^{k+1} \leftarrow z^k + \alpha dz$

Avaliar/Testar Condição de Parada

$k \leftarrow k + 1$

**end while**

# Algoritmo Método Primal–Dual Afim Escala

**Require:** $x^0, y^0, z^0$ tal que $x^0 > 0, z^0 > 0$

$\tau = 0.99995, Maxit = 100$

**while** $k = 0, 1, 2, \cdots, Maxit$ e Condição de Parada **do**

$r_p = b - Ax$

$r_d = c - A^T y - z$

$r_a = XZe$

$D = Z^{-1}X$

Resolver $ADA^T dy = r_p + AD(r_d - X^{-1}r_a)$

$dx = D(A^T dy - r_d + X^{-1}r_a)$

$dz = X^{-1}(r_a - Zdx)$

Calcular o tamanho de paso $\alpha$

$x^{k+1} \leftarrow x^k + \alpha dx, \quad y^{k+1} \leftarrow y^k + \alpha dy, \quad z^{k+1} \leftarrow z^k + \alpha dz$

Avaliar/Testar Condição de Parada

$k \leftarrow k + 1$

**end while**

# Algoritmo Método Primal–Dual Afim Escala

**Require:** $x^0, y^0, z^0$ tal que $x^0 > 0, z^0 > 0$
$\tau = 0.99995, Maxit = 100$
**while** $k = 0, 1, 2, \cdots, Maxit$ e Condição de Parada **do**
$r_p = b - Ax$
$r_d = c - A^T y - z$
$r_a = XZe$
$D = Z^{-1}X$
Resolver $ADA^T dy = r_p + AD(r_d - X^{-1} r_a)$
$dx = D(A^T dy - r_d + X^{-1} r_a)$
$dz = X^{-1}(r_a - Zdx)$
Calcular o tamanho de paso $\alpha$
$x^{k+1} \leftarrow x^k + \alpha dx, \quad y^{k+1} \leftarrow y^k + \alpha dy, \quad z^{k+1} \leftarrow z^k + \alpha dz$
Avaliar/Testar Condição de Parada
$k \leftarrow k + 1$
**end while**

# Algoritmo Método Primal–Dual Afim Escala

**Require:** $x^0, y^0, z^0$ tal que $x^0 > 0, z^0 > 0$

$\tau = 0.99995, Maxit = 100$

**while** $k = 0, 1, 2, \cdots, Maxit$ e Condição de Parada **do**

$r_p = b - Ax$

$r_d = c - A^T y - z$

$r_a = XZe$

$D = Z^{-1}X$

Resolver $ADA^T dy = r_p + AD(r_d - X^{-1} r_a)$

$dx = D(A^T dy - r_d + X^{-1} r_a)$

$dz = X^{-1}(r_a - Z dx)$

Calcular o tamanho de paso $\alpha$

$x^{k+1} \leftarrow x^k + \alpha dx, \quad y^{k+1} \leftarrow y^k + \alpha dy, \quad z^{k+1} \leftarrow z^k + \alpha dz$

Avaliar/Testar Condição de Parada

$k \leftarrow k + 1$

**end while**

# Conteúdo:

# PC

- Direção Afim Escala: É a direção de Newton como *direção preditora*.
- Direção de Correção: Que compensa a aproximação linear do Método de Newton
- Direção de Centragem: Uma vez que existe uma direção de correção vamos perturbar o problema com um parâmetro $\mu$ escolhido de forma mais inteligente que no método de seguidor de caminho.

- $\alpha = 1, \Rightarrow r_p^{k+1} = b - A(x + dx) = b - Ax - Adx = r_p^k - Adx = 0$
- $(X + D_X)(Z + D_Z)e = (XZ + ZD_X + XD_Z + D_XD_Z)e = -r_a + r_a + D_XD_Ze = D_XD_Ze$

# PC

- Direção Afim Escala: É a direção de Newton como *direção preditora*.
- Direção de Correção: Que compensa a aproximação linear do Método de Newton
- Direção de Centragem: Uma vez que existe uma direção de correção vamos perturbar o problema com um parâmetro $\mu$ escolhido de forma mais inteligente que no método de seguidor de caminho.

- $\alpha = 1, \Rightarrow r_p^{k+1} = b - A(x + dx) = b - Ax - Adx = r_p^k - Adx = 0$
- $(X + D_X)(Z + D_Z)e = (XZ + ZD_X + XD_Z + D_X D_Z)e = -r_a + r_a + D_X D_Z e = D_X D_Z e$

# PC

- Direção Afim Escala: É a direção de Newton como *direção preditora*.
- Direção de Correção: Que compensa a aproximação linear do Método de Newton
- Direção de Centragem: Uma vez que existe uma direção de correção vamos perturbar o problema com um parâmetro $\mu$ escolhido de forma mais inteligente que no método de seguidor de caminho.

- $\alpha = 1, \Rightarrow r_p^{k+1} = b - A(x + dx) = b - Ax - Adx = r_p^k - Adx = 0$
- $(X + D_X)(Z + D_Z)e = (XZ + ZD_X + XD_Z + D_XD_Z)e = -r_a + r_a + D_XD_Ze = D_XD_Ze$

# PC

- Direção Afim Escala: É a direção de Newton como *direção preditora*.
- Direção de Correção: Que compensa a aproximação linear do Método de Newton
- Direção de Centragem: Uma vez que existe uma direção de correção vamos perturbar o problema com um parâmetro $\mu$ escolhido de forma mais inteligente que no método de seguidor de caminho.

- $\alpha = 1, \Rightarrow r_p^{k+1} = b - A(x + dx) = b - Ax - Adx = r_p^k - Adx = 0$
- $(X + D_X)(Z + D_Z)e = (XZ + ZD_X + XD_Z + D_X D_Z)e = -r_a + r_a + D_X D_Z e = D_X D_Z e$

# PC

- Direção Afim Escala: É a direção de Newton como *direção preditora*.
- Direção de Correção: Que compensa a aproximação linear do Método de Newton
- Direção de Centragem: Uma vez que existe uma direção de correção vamos perturbar o problema com um parâmetro $\mu$ escolhido de forma mais inteligente que no método de seguidor de caminho.

- $\alpha = 1, \Rightarrow r_p^{k+1} = b - A(x + dx) = b - Ax - Adx = r_p^k - Adx = 0$
- $(X + D_X)(Z + D_Z)e = (XZ + ZD_X + XD_Z + D_X D_Z)e = -r_a + r_a + D_X D_Z e = D_X D_Z e$

# Cont PC

- Preditor

$$\begin{pmatrix} A & 0 & 0 \\ 0 & A^T & I \\ Z & 0 & X \end{pmatrix} \begin{pmatrix} \widetilde{dx} \\ \widetilde{dy} \\ \widetilde{dz} \end{pmatrix} = \begin{pmatrix} r_p \\ r_d \\ r_a \end{pmatrix} \tag{7}$$

- Corretor

$$\begin{pmatrix} A & 0 & 0 \\ 0 & A^T & I \\ Z & 0 & X \end{pmatrix} \begin{pmatrix} dx \\ dy \\ dz \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ -\widetilde{D_X}\widetilde{D_Z}e \end{pmatrix} \tag{8}$$

- Parâmetro de Perturbação

$$\mu = \sigma\frac{\gamma}{n}, \qquad \sigma = \left(\frac{\tilde{\gamma}}{\gamma}\right)^p, \qquad \tilde{\gamma} = (x + \tilde{\alpha}_p\widetilde{dx})^T(z + \tilde{\alpha}_d\widetilde{dz}), \qquad p = 2 \text{ ou } 3.$$

# Cont PC

- Preditor

$$\begin{pmatrix} A & 0 & 0 \\ 0 & A^T & I \\ Z & 0 & X \end{pmatrix} \begin{pmatrix} \widetilde{dx} \\ \widetilde{dy} \\ \widetilde{dz} \end{pmatrix} = \begin{pmatrix} r_p \\ r_d \\ r_a \end{pmatrix} \tag{7}$$

- Corretor

$$\begin{pmatrix} A & 0 & 0 \\ 0 & A^T & I \\ Z & 0 & X \end{pmatrix} \begin{pmatrix} dx \\ dy \\ dz \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ -\widetilde{D_X}\widetilde{D_Z}e \end{pmatrix} \tag{8}$$

- Parâmetro de Perturbação

$$\mu = \sigma \frac{\gamma}{n}, \qquad \sigma = \left(\frac{\tilde{\gamma}}{\gamma}\right)^p, \qquad \tilde{\gamma} = (x + \tilde{\alpha}_p \widetilde{dx})^T (z + \tilde{\alpha}_d \widetilde{dz}), \qquad p = 2 \text{ ou } 3.$$

# Cont PC

- Preditor

$$\begin{pmatrix} A & 0 & 0 \\ 0 & A^T & I \\ Z & 0 & X \end{pmatrix} \begin{pmatrix} \widetilde{dx} \\ \widetilde{dy} \\ \widetilde{dz} \end{pmatrix} = \begin{pmatrix} r_p \\ r_d \\ r_a \end{pmatrix} \tag{7}$$

- Corretor

$$\begin{pmatrix} A & 0 & 0 \\ 0 & A^T & I \\ Z & 0 & X \end{pmatrix} \begin{pmatrix} dx \\ dy \\ dz \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ -\widetilde{D_X}\widetilde{D_Z}e \end{pmatrix} \tag{8}$$

- Parâmetro de Perturbação

$$\mu = \sigma \frac{\gamma}{n}, \qquad \sigma = \left(\frac{\tilde{\gamma}}{\gamma}\right)^p, \quad \tilde{\gamma} = (x + \tilde{\alpha}_p \widetilde{dx})^T (z + \tilde{\alpha}_d \widetilde{dz}), \quad p = 2 \text{ ou } 3. \tag{9}$$

# Cont PC

- Utilizamos a perturbação na direção de correção ao mesmo tempo que fazemos a correção não linear

$$\begin{pmatrix} A & 0 & 0 \\ 0 & A^T & I \\ Z & 0 & X \end{pmatrix} \begin{pmatrix} \widehat{dx} \\ \widehat{dy} \\ \widehat{dz} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \mu e - \widetilde{D_X}\widetilde{D_Z} e \end{pmatrix} \tag{10}$$

- A direção final é a soma das direções $\tilde{d}$ e $\hat{d}$.

$$\begin{pmatrix} A & 0 & 0 \\ 0 & A^T & I \\ Z & 0 & X \end{pmatrix} \begin{pmatrix} dx \\ dy \\ dz \end{pmatrix} = \begin{pmatrix} r_p \\ r_d \\ r_s \end{pmatrix}, \quad \begin{array}{l} r_a = -XZe \\ r_c = \mu e - XZe = \mu e + r_a \\ r_s = \mu e - XZe - \widetilde{D_X}\widetilde{D_Z} e \end{array} \tag{11}$$

$$r_s = r_c - \widetilde{D_X}\widetilde{D_Z} e$$

# Cont PC

- Utilizamos a perturbação na direção de correção ao mesmo tempo que fazemos a correção não linear

$$\begin{pmatrix} A & 0 & 0 \\ 0 & A^T & I \\ Z & 0 & X \end{pmatrix} \begin{pmatrix} \widehat{dx} \\ \widehat{dy} \\ \widehat{dz} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \mu e - \widetilde{D_X}\widetilde{D_Z}e \end{pmatrix} \tag{10}$$

- A direção final é a soma das direções $\tilde{d}$ e $\hat{d}$.

$$\begin{pmatrix} A & 0 & 0 \\ 0 & A^T & I \\ Z & 0 & X \end{pmatrix} \begin{pmatrix} dx \\ dy \\ dz \end{pmatrix} = \begin{pmatrix} r_p \\ r_d \\ r_s \end{pmatrix}, \quad \begin{aligned} r_a &= -XZe \\ r_c &= \mu e - XZe = \mu e + r_a \\ r_s &= \mu e - XZe - \widetilde{D_X}\widetilde{D_Z}e \end{aligned} \tag{11}$$

$$r_s = r_c - \widetilde{D_X}\widetilde{D_Z}e$$

# Algoritmo Preditor-Corretor

**Require:** $x^0, y^0, z^0$ tal que $x^0 > 0, z^0 > 0$

$\tau = 0.99995$

$Maxit = 100$

**while** $k = 0, 1, 2, \cdots, Maxit$ e Condição de Parada **do**

[illegible]

**end while**

# Algoritmo Preditor-Corretor

**Require:** $x^0, y^0, z^0$ tal que $x^0 > 0, z^0 > 0$

$\tau = 0.99995$

$Maxit = 100$

**while** $k = 0, 1, 2, \cdots, Maxit$ e Condição de Parada **do**

**end while**

# Algoritmo Preditor-Corretor

**Require:** $x^0, y^0, z^0$ tal que $x^0 > 0, z^0 > 0$

$\tau = 0.99995$

$Maxit = 100$

**while** $k = 0, 1, 2, \cdots, Maxit$ e Condição de Parada **do**

Calcule a direção afim escala: $\widetilde{dx}$, $\widetilde{dy}$ e $\widetilde{dz}$

**end while**

# Algoritmo Preditor-Corretor

**Require:** $x^0, y^0, z^0$ tal que $x^0 > 0, z^0 > 0$

$\tau = 0.99995$

$Maxit = 100$

**while** $k = 0, 1, 2, \cdots, Maxit$ e Condição de Parada **do**

Calcule a direção afim escala: $\widetilde{dx}$, $\widetilde{dy}$ e $\widetilde{dz}$

Calcule o tamanho de passo $\widetilde{\alpha_p}$ e $\widetilde{\alpha_d}$

Calcule $\tilde{\gamma}$, $\sigma$ e $\mu$

Calcule a direção final usando (11)

Calcule o tamanho de passo $\alpha_p$ e $\alpha_d$

$x^{k+1} \leftarrow x^k + \alpha_p dx$

$y^{k+1} \leftarrow y^k + \alpha_d dy$

$z^{k+1} \leftarrow z^k + \alpha_d dz$

Avaliar/Testar Condição de Parada

$k \leftarrow k + 1$

**end while**

# Algoritmo Preditor-Corretor

**Require:** $x^0, y^0, z^0$ tal que $x^0 > 0, z^0 > 0$

$\tau = 0.99995$

$Maxit = 100$

**while** $k = 0, 1, 2, \cdots, Maxit$ e Condição de Parada **do**

Calcule a direção afim escala: $\widetilde{dx}$, $\widetilde{dy}$ e $\widetilde{dz}$

Calcule o tamanho de passo $\widetilde{\alpha_p}$ e $\widetilde{\alpha_d}$

Calcule $\tilde{\gamma}$, $\sigma$ e $\mu$

Calcule a direção final usando (11)

Calcule o tamanho de passo $\alpha_p$ e $\alpha_d$

$x^{k+1} \leftarrow x^k + \alpha_p dx$

$y^{k+1} \leftarrow y^k + \alpha_d dy$

$z^{k+1} \leftarrow z^k + \alpha_d dz$

Avaliar/Testar Condição de Parada

$k \leftarrow k + 1$

**end while**

# Algoritmo Preditor-Corretor

**Require:** $x^0, y^0, z^0$ tal que $x^0 > 0, z^0 > 0$

$\tau = 0.99995$

$Maxit = 100$

**while** $k = 0, 1, 2, \cdots, Maxit$ e Condição de Parada **do**

Calcule a direção afim escala: $\widetilde{dx}$, $\widetilde{dy}$ e $\widetilde{dz}$

Calcule o tamanho de passo $\widetilde{\alpha_p}$ e $\widetilde{\alpha_d}$

Calcule $\tilde{\gamma}$, $\sigma$ e $\mu$

Calcule a direção final usando (11)

Calcule o tamanho de passo $\alpha_p$ e $\alpha_d$

$x^{k+1} \leftarrow x^k + \alpha_p dx$

$y^{k+1} \leftarrow y^k + \alpha_d dy$

$z^{k+1} \leftarrow z^k + \alpha_d dz$

Avaliar/Testar Condição de Parada

$k \leftarrow k + 1$

**end while**

# Algoritmo Preditor-Corretor

**Require:** $x^0, y^0, z^0$ tal que $x^0 > 0, z^0 > 0$

$\tau = 0.99995$

$Maxit = 100$

**while** $k = 0, 1, 2, \cdots, Maxit$ e Condição de Parada **do**

Calcule a direção afim escala: $\widetilde{dx}$, $\widetilde{dy}$ e $\widetilde{dz}$

Calcule o tamanho de passo $\widetilde{\alpha_p}$ e $\widetilde{\alpha_d}$

Calcule $\tilde{\gamma}$, $\sigma$ e $\mu$

Calcule a direção final usando (11)

Calcule o tamanho de passo $\alpha_p$ e $\alpha_d$

$x^{k+1} \leftarrow x^k + \alpha_p dx$

$y^{k+1} \leftarrow y^k + \alpha_d dy$

$z^{k+1} \leftarrow z^k + \alpha_d dz$

Avaliar/Testar Condição de Parada

$k \leftarrow k + 1$

**end while**

# Algoritmo Preditor-Corretor

**Require:** $x^0, y^0, z^0$ tal que $x^0 > 0, z^0 > 0$

$\tau = 0.99995$

$Maxit = 100$

**while** $k = 0, 1, 2, \cdots, Maxit$ e Condição de Parada **do**

Calcule a direção afim escala: $\widetilde{dx}$, $\widetilde{dy}$ e $\widetilde{dz}$

Calcule o tamanho de passo $\widetilde{\alpha_p}$ e $\widetilde{\alpha_d}$

Calcule $\tilde{\gamma}$, $\sigma$ e $\mu$

Calcule a direção final usando (11)

Calcule o tamanho de passo $\alpha_p$ e $\alpha_d$

$x^{k+1} \leftarrow x^k + \alpha_p dx$

$y^{k+1} \leftarrow y^k + \alpha_d dy$

$z^{k+1} \leftarrow z^k + \alpha_d dz$

Avaliar/Testar Condição de Parada

$k \leftarrow k + 1$

**end while**

# Algoritmo Preditor-Corretor

**Require:** $x^0, y^0, z^0$ tal que $x^0 > 0, z^0 > 0$

$\tau = 0.99995$

$Maxit = 100$

**while** $k = 0, 1, 2, \cdots, Maxit$ e Condição de Parada **do**

Calcule a direção afim escala: $\widetilde{dx}$, $\widetilde{dy}$ e $\widetilde{dz}$

Calcule o tamanho de passo $\widetilde{\alpha_p}$ e $\widetilde{\alpha_d}$

Calcule $\tilde{\gamma}$, $\sigma$ e $\mu$

Calcule a direção final usando (11)

Calcule o tamanho de passo $\alpha_p$ e $\alpha_d$

$x^{k+1} \leftarrow x^k + \alpha_p dx$

$y^{k+1} \leftarrow y^k + \alpha_d dy$

$z^{k+1} \leftarrow z^k + \alpha_d dz$

Avaliar/Testar Condição de Parada

$k \leftarrow k + 1$

**end while**

# Algoritmo Preditor-Corretor

**Require:** $x^0, y^0, z^0$ tal que $x^0 > 0, z^0 > 0$

$\tau = 0.99995$

$Maxit = 100$

**while** $k = 0, 1, 2, \cdots, Maxit$ e Condição de Parada **do**

Calcule a direção afim escala: $\widetilde{dx}$, $\widetilde{dy}$ e $\widetilde{dz}$

Calcule o tamanho de passo $\widetilde{\alpha_p}$ e $\widetilde{\alpha_d}$

Calcule $\tilde{\gamma}$, $\sigma$ e $\mu$

Calcule a direção final usando (11)

Calcule o tamanho de passo $\alpha_p$ e $\alpha_d$

$x^{k+1} \leftarrow x^k + \alpha_p dx$

$y^{k+1} \leftarrow y^k + \alpha_d dy$

$z^{k+1} \leftarrow z^k + \alpha_d dz$

Avaliar/Testar Condição de Parada

$k \leftarrow k + 1$

**end while**

# Algoritmo Preditor-Corretor

**Require:** $x^0, y^0, z^0$ tal que $x^0 > 0, z^0 > 0$

$\tau = 0.99995$

$Maxit = 100$

**while** $k = 0, 1, 2, \cdots, Maxit$ e Condição de Parada **do**

Calcule a direção afim escala: $\widetilde{dx}$, $\widetilde{dy}$ e $\widetilde{dz}$

Calcule o tamanho de passo $\widetilde{\alpha_p}$ e $\widetilde{\alpha_d}$

Calcule $\tilde{\gamma}$, $\sigma$ e $\mu$

Calcule a direção final usando (11)

Calcule o tamanho de passo $\alpha_p$ e $\alpha_d$

$x^{k+1} \leftarrow x^k + \alpha_p dx$

$y^{k+1} \leftarrow y^k + \alpha_d dy$

$z^{k+1} \leftarrow z^k + \alpha_d dz$

Avaliar/Testar Condição de Parada

$k \leftarrow k + 1$

**end while**

# Algoritmo Preditor-Corretor

**Require:** $x^0, y^0, z^0$ tal que $x^0 > 0, z^0 > 0$

$\tau = 0.99995$

$Maxit = 100$

**while** $k = 0, 1, 2, \cdots, Maxit$ e Condição de Parada **do**

Calcule a direção afim escala: $\widetilde{dx}$, $\widetilde{dy}$ e $\widetilde{dz}$

Calcule o tamanho de passo $\widetilde{\alpha_p}$ e $\widetilde{\alpha_d}$

Calcule $\tilde{\gamma}$, $\sigma$ e $\mu$

Calcule a direção final usando (11)

Calcule o tamanho de passo $\alpha_p$ e $\alpha_d$

$x^{k+1} \leftarrow x^k + \alpha_p dx$

$y^{k+1} \leftarrow y^k + \alpha_d dy$

$z^{k+1} \leftarrow z^k + \alpha_d dz$

Avaliar/Testar Condição de Parada

$k \leftarrow k + 1$

**end while**

# Algoritmo Preditor-Corretor

**Require:** $x^0, y^0, z^0$ tal que $x^0 > 0, z^0 > 0$

$\tau = 0.99995$

$Maxit = 100$

**while** $k = 0, 1, 2, \cdots, Maxit$ e Condição de Parada **do**

Calcule a direção afim escala: $\widetilde{dx}$, $\widetilde{dy}$ e $\widetilde{dz}$

Calcule o tamanho de passo $\widetilde{\alpha_p}$ e $\widetilde{\alpha_d}$

Calcule $\tilde{\gamma}$, $\sigma$ e $\mu$

Calcule a direção final usando (11)

Calcule o tamanho de passo $\alpha_p$ e $\alpha_d$

$x^{k+1} \leftarrow x^k + \alpha_p dx$

$y^{k+1} \leftarrow y^k + \alpha_d dy$

$z^{k+1} \leftarrow z^k + \alpha_d dz$

Avaliar/Testar Condição de Parada

$k \leftarrow k + 1$

**end while**

# Algoritmo Preditor-Corretor

**Require:** $x^0, y^0, z^0$ tal que $x^0 > 0, z^0 > 0$

$\tau = 0.99995$

$Maxit = 100$

**while** $k = 0, 1, 2, \cdots, Maxit$ e Condição de Parada **do**

Calcule a direção afim escala: $\widetilde{dx}$, $\widetilde{dy}$ e $\widetilde{dz}$

Calcule o tamanho de passo $\widetilde{\alpha_p}$ e $\widetilde{\alpha_d}$

Calcule $\tilde{\gamma}$, $\sigma$ e $\mu$

Calcule a direção final usando (11)

Calcule o tamanho de passo $\alpha_p$ e $\alpha_d$

$x^{k+1} \leftarrow x^k + \alpha_p dx$

$y^{k+1} \leftarrow y^k + \alpha_d dy$

$z^{k+1} \leftarrow z^k + \alpha_d dz$

Avaliar/Testar Condição de Parada

$k \leftarrow k + 1$

**end while**

# Algoritmo Preditor-Corretor

**Require:** $x^0, y^0, z^0$ tal que $x^0 > 0, z^0 > 0$

$\tau = 0.99995$

$Maxit = 100$

**while** $k = 0, 1, 2, \cdots, Maxit$ e Condição de Parada **do**

Calcule a direção afim escala: $\widetilde{dx}$, $\widetilde{dy}$ e $\widetilde{dz}$

Calcule o tamanho de passo $\widetilde{\alpha_p}$ e $\widetilde{\alpha_d}$

Calcule $\tilde{\gamma}$, $\sigma$ e $\mu$

Calcule a direção final usando (11)

Calcule o tamanho de passo $\alpha_p$ e $\alpha_d$

$x^{k+1} \leftarrow x^k + \alpha_p dx$

$y^{k+1} \leftarrow y^k + \alpha_d dy$

$z^{k+1} \leftarrow z^k + \alpha_d dz$

Avaliar/Testar Condição de Parada

$k \leftarrow k + 1$

**end while**

# Obs

- É necessário resolver dois sistemas lineares por iteração com a mesma matriz de coeficientes.
- O esforço por iteração é maior
- Na pratica o número de iterações do método Preditor–Corretor é pouco mais que a metade em relação ao método Seguidor de Caminho.
- É possível provar convergência quadrática para esse método, se consideramos que os dois sistemas lineares constituem uma iteração.
- Este método poderia ser chamado *Método de Newton Perturbado* ($\mu$) *Modificado* ($(x, z) > 0$) de ordem 1 (1 correção linear).
- Este método é a base de todas implementações atuais de pontos interiores.

# Obs

- É necessário resolver dois sistemas lineares por iteração com a mesma matriz de coeficientes.
- O esforço por iteração é maior
- Na pratica o número de iterações do método Preditor–Corretor é pouco mais que a metade em relação ao método Seguidor de Caminho.
- É possível provar convergência quadrática para esse método, se consideramos que os dois sistemas lineares constituem uma iteração.
- Este método poderia ser chamado *Método de Newton Perturbado* ($\mu$) *Modificado* ($(x, z) > 0$) de ordem 1 (1 correção linear).
- Este método é a base de todas implementações atuais de pontos interiores.

# Obs

- É necessário resolver dois sistemas lineares por iteração com a mesma matriz de coeficientes.
- O esforço por iteração é maior
- Na pratica o número de iterações do método Preditor–Corretor é pouco mais que a metade em relação ao método Seguidor de Caminho.
- É possível provar convergência quadrática para esse método, se consideramos que os dois sistemas lineares constituem uma iteração.
- Este método poderia ser chamado *Método de Newton Perturbado* ($\mu$) *Modificado* ($(x, z) > 0$) de ordem 1 (1 correção linear).
- Este método é a base de todas implementações atuais de pontos interiores.

# Obs

- É necessário resolver dois sistemas lineares por iteração com a mesma matriz de coeficientes.
- O esforço por iteração é maior
- Na pratica o número de iterações do método Preditor–Corretor é pouco mais que a metade em relação ao método Seguidor de Caminho.
- É possível provar convergência quadrática para esse método, se consideramos que os dois sistemas lineares constituem uma iteração.
- Este método poderia ser chamado *Método de Newton Perturbado* $(\mu)$ *Modificado* $((x, z) > 0)$ de ordem 1 (1 correção linear).
- Este método é a base de todas implementações atuais de pontos interiores.

# Obs

- É necessário resolver dois sistemas lineares por iteração com a mesma matriz de coeficientes.
- O esforço por iteração é maior
- Na pratica o número de iterações do método Preditor–Corretor é pouco mais que a metade em relação ao método Seguidor de Caminho.
- É possível provar convergência quadrática para esse método, se consideramos que os dois sistemas lineares constituem uma iteração.
- Este método poderia ser chamado *Método de Newton Perturbado* $(\mu)$ *Modificado* $((x, z) > 0)$ de ordem 1 (1 correção linear).
- Este método é a base de todas implementações atuais de pontos interiores.

# Obs

- É necessário resolver dois sistemas lineares por iteração com a mesma matriz de coeficientes.
- O esforço por iteração é maior
- Na pratica o número de iterações do método Preditor–Corretor é pouco mais que a metade em relação ao método Seguidor de Caminho.
- É possível provar convergência quadrática para esse método, se consideramos que os dois sistemas lineares constituem uma iteração.
- Este método poderia ser chamado *Método de Newton Perturbado* ($\mu$) *Modificado* ($(x, z) > 0$) de ordem 1 (1 correção linear).
- Este método é a base de todas implementações atuais de pontos interiores.

# Conteúdo:

Métodos diretos e iterativos

# Propriedades dos Sistemas

- Qual sistema resolver?

| Equação Normal | Sistema Aumentado |
|---|---|
| $ADA^T dy = r$ | $\begin{pmatrix} -D^{-1} & A^T \\ A & 0 \end{pmatrix} \begin{pmatrix} dx \\ dy \end{pmatrix} = \begin{pmatrix} r_1 \\ r_2 \end{pmatrix}$ (12) |

| Equações Normais | Sistema Aumentado |
|---|---|
| Definida Positiva | Indefinida |
| Simétrica | Simétrica |
| posto $m$ | posto $m+n$ |
| Perda da estrutura esparsa | Utiliza matrizes originais |
| Somente $D$ altera na matriz | Somente $D$ altera na matriz |
| Muito mais mal-condicionado | mal-condicionado |
| Cholesky (Precondicionado) | Bunch–Parlet |
| Gradientes Conjugados (Prec.) | Varios Métodos iterativos |

# Propriedades dos Sistemas

- Qual sistema resolver?

| Equação Normal | Sistema Aumentado |
|---|---|
| $ADA^T dy = r$ | $\begin{pmatrix} -D^{-1} & A^T \\ A & 0 \end{pmatrix} \begin{pmatrix} dx \\ dy \end{pmatrix} = \begin{pmatrix} r_1 \\ r_2 \end{pmatrix}$ |

(12)

| Equações Normais | Sistema Aumentado |
|---|---|
| Definida Positiva | Indefinida |
| Simétrica | Simétrica |
| posto $m$ | posto $m+n$ |
| • Perda da estrutura esparsa | Utiliza matrizes originais |
| Somente $D$ altera na matriz | Somente $D$ altera na matriz |
| Muito mais mal-condicionado | mal-condicionado |
| Cholesky (Precondicionado) | Bunch–Parlet |
| Gradientes Conjugados (Prec.) | Varios Métodos iterativos |

# Decomposição de Cholesky para matrizes esparsas

- $\mathbb{B} = ADA^T, \quad B_{ij} = \sum_{k=1}^{n} D_{kk} A_{ik} A^T_{kj} = \sum_{k=1}^{n} D_{kk} A_{ik} A_{jk}$
- Enchimento

$$\begin{pmatrix} \ast & \ast & \ast & \ast & \ast \\ \ast & \ast & \cdot & \cdot & \cdot \\ \ast & \cdot & \ast & \cdot & \cdot \\ \ast & \cdot & \cdot & \ast & \cdot \\ \ast & \cdot & \cdot & \cdot & \ast \end{pmatrix} \Rightarrow \quad L = \begin{pmatrix} \ast & \ast & \ast & \ast & \ast \\ \cdot & \ast & \ast & \ast & \ast \\ \cdot & \cdot & \ast & \ast & \ast \\ \cdot & \cdot & \cdot & \ast & \ast \\ \cdot & \cdot & \cdot & \cdot & \ast \end{pmatrix}$$

- Reordenado

$$\begin{pmatrix} \ast & \cdot & \cdot & \cdot & \ast \\ \cdot & \ast & \cdot & \cdot & \ast \\ \cdot & \cdot & \ast & \cdot & \ast \\ \cdot & \cdot & \cdot & \ast & \ast \\ \ast & \ast & \ast & \ast & \ast \end{pmatrix} \Rightarrow \quad L = \begin{pmatrix} \ast & \cdot & \cdot & \cdot & \ast \\ \cdot & \ast & \cdot & \cdot & \ast \\ \cdot & \cdot & \ast & \cdot & \ast \\ \cdot & \cdot & \cdot & \ast & \ast \\ \cdot & \cdot & \cdot & \cdot & \ast \end{pmatrix}$$

# Decomposição de Cholesky para matrizes esparsas

- $\mathbb{B} = ADA^T, \quad B_{ij} = \sum_{k=1}^{n} D_{kk} A_{ik} A^T_{kj} = \sum_{k=1}^{n} D_{kk} A_{ik} A_{jk}$
- Enchimento

$$\begin{pmatrix} * & * & * & * & * \\ * & * & . & . & . \\ * & . & * & . & . \\ * & . & . & * & . \\ * & . & . & . & * \end{pmatrix} \quad \Rightarrow \quad L = \begin{pmatrix} * & * & * & * & * \\ . & * & * & * & * \\ . & . & * & * & * \\ . & . & . & * & * \\ . & . & . & . & * \end{pmatrix}$$

- Reordenado

$$\begin{pmatrix} * & . & . & . & * \\ . & * & . & . & * \\ . & . & * & . & * \\ . & . & . & * & * \\ * & * & * & * & * \end{pmatrix} \quad \Rightarrow \quad L = \begin{pmatrix} * & . & . & . & * \\ . & * & . & . & * \\ . & . & * & . & * \\ . & . & . & * & * \\ . & . & . & . & * \end{pmatrix}$$

# Decomposição de Cholesky para matrizes esparsas

- $\mathbb{B} = ADA^T, \quad B_{ij} = \sum_{k=1}^{n} D_{kk}A_{ik}A_{kj}^T = \sum_{k=1}^{n} D_{kk}A_{ik}A_{jk}$
- Enchimento

$$\begin{pmatrix} * & * & * & * & * \\ * & * & . & . & . \\ * & . & * & . & . \\ * & . & . & * & . \\ * & . & . & . & * \end{pmatrix} \Rightarrow \quad L = \begin{pmatrix} * & * & * & * & * \\ . & * & * & * & * \\ . & . & * & * & * \\ . & . & . & * & * \\ . & . & . & . & * \end{pmatrix}$$

- Reordenado

$$\begin{pmatrix} * & . & . & . & * \\ . & * & . & . & * \\ . & . & * & . & * \\ . & . & . & * & * \\ * & * & * & * & * \end{pmatrix} \Rightarrow \quad L = \begin{pmatrix} * & . & . & . & * \\ . & * & . & . & * \\ . & . & * & . & * \\ . & . & . & * & * \\ . & . & . & . & * \end{pmatrix}$$

# Conteúdo:

# *PL* Penalizado

- Problema *PL* canalizado misto

$$\begin{array}{lll} (P) & \text{Min} & c^T x \\ & \text{s.a.} & \qquad Ax = b \\ & & Ex + v = u \\ & & (x, v) \geq 0 \end{array} \tag{13}$$

- Sub-Problema, $\rho$ parâmetro de penalização

$$\begin{array}{lll} (P_\rho) & \text{Min} & c^T x + \frac{\rho}{2}\left(\|b - Ax\|^2 + \|u - Ex - v\|^2\right) \\ & \text{s.a.} & (x, v) \geq 0 \end{array} \tag{14}$$

Função de penalização $\mathcal{P}(x, v) = \frac{\rho}{2}\left(\|b - Ax\|^2 + \|u - Ex - v\|^2\right)$

# *PL* Penalizado

- Problema *PL* canalizado misto

$$
\begin{array}{lll}
(P) & \text{Min} & c^T x \\
 & \text{s.a.} & \quad\; Ax = b \\
 & & Ex + v = u \\
 & & (x, v) \geq 0
\end{array} \tag{13}
$$

- Sub-Problema, $\rho$ parâmetro de penalização

$$
\begin{array}{lll}
(P_\rho) & \text{Min} & c^T x + \frac{\rho}{2}\left(\|b - Ax\|^2 + \|u - Ex - v\|^2\right) \\
 & \text{s.a.} & (x, v) \geq 0
\end{array} \tag{14}
$$

Função de penalização $\mathcal{P}(x, v) = \frac{\rho}{2}\left(\|b - Ax\|^2 + \|u - Ex - v\|^2\right)$

# KKT

- Sistema KKT perturbado

$$
\begin{aligned}
Ax + \delta y &= b \\
Ex + v - \delta w &= u \\
A^T y + z - E^T w &= c \\
XZe &= \mu e \\
VWe &= \mu e
\end{aligned}
\tag{15}
$$

- Sistema Newton

$$
\begin{aligned}
A dx + \delta dy &= r_p, & r_p &= b - Ax - \delta y \\
E dx + dv - \delta dw &= r_u, & r_u &= u - Ex - v + \delta w \\
A^T dy + dz - E^T dw &= r_d, & r_d &= c - A^T y - z + E^T w \\
Z dx + X dz &= r_s, & r_s &= -XZe + \mu e \\
W dv + V dw &= r_r, & r_r &= -VWe + \mu e
\end{aligned}
\tag{16}
$$

# KKT

- Sistema KKT perturbado

$$
\begin{aligned}
Ax + \delta y &= b \\
Ex + v - \delta w &= u \\
A^T y + z - E^T w &= c \\
XZe &= \mu e \\
VWe &= \mu e
\end{aligned}
\tag{15}
$$

- Sistema Newton

$$
\begin{aligned}
A dx + \delta dy = r_p, &\quad r_p = b - Ax - \delta y \\
E dx + dv - \delta dw = r_u, &\quad r_u = u - Ex - v + \delta w \\
A^T dy + dz - E^T dw = r_d, &\quad r_d = c - A^T y - z + E^T w \\
Z dx + X dz = r_s, &\quad r_s = -XZe + \mu e \\
W dv + V dw = r_r, &\quad r_r = -VWe + \mu e
\end{aligned}
\tag{16}
$$

# Sistema Aumentado e Normal

- $D = (X^{-1}Z + E^T S^{-1} W E)^{-1}$ e $r_1 = r_d - X^{-1} r_s + E^T S^{-1} r_r - E^T S^{-1} W r_u$

$$\begin{pmatrix} -D^{-1} & A^T \\ A & \delta I \end{pmatrix} \begin{pmatrix} dx \\ dy \end{pmatrix} = \begin{pmatrix} r_1 \\ r_p \end{pmatrix} \tag{17}$$

- Sistema Normal

$$(ADA^T + \delta I)dy = r_p + AD(r_d - X^{-1} r_s + E^T S^{-1} r_r - E^T S^{-1} W r_u) = r_p + ADr_1 \tag{18}$$

- Equivalência a Sub-Problema

$$\begin{array}{lll} (P_\delta) & \text{Min} & c^T x + \frac{\delta}{2} y^T y + \frac{\delta}{2} w^T w \\ & \text{s.a.} & Ax + \delta y = b \\ & & Ex + v - \delta w = u \\ & & (x, v) \geq 0, \ y \text{ livre}, \ w \geq 0 \end{array}$$

# Sistema Aumentado e Normal

- $D = (X^{-1}Z + E^T S^{-1} W E)^{-1}$ e $r_1 = r_d - X^{-1} r_s + E^T S^{-1} r_r - E^T S^{-1} W r_u$

$$\begin{pmatrix} -D^{-1} & A^T \\ A & \delta I \end{pmatrix} \begin{pmatrix} dx \\ dy \end{pmatrix} = \begin{pmatrix} r_1 \\ r_p \end{pmatrix} \tag{17}$$

- Sistema Normal

$$(ADA^T + \delta I) dy = r_p + AD(r_d - X^{-1} r_s + E^T S^{-1} r_r - E^T S^{-1} W r_u) = r_p + ADr_1 \tag{18}$$

- Equivalência a Sub-Problema

$$\begin{array}{lll} (P_\delta) & \text{Min} & c^T x + \frac{\delta}{2} y^T y + \frac{\delta}{2} w^T w \\ & \text{s.a.} & Ax + \delta y = b \\ & & Ex + v - \delta w = u \\ & & (x, v) \geq 0, \ y \text{ livre}, \ w \geq 0 \end{array}$$

# Sistema Aumentado e Normal

- $D = (X^{-1}Z + E^T S^{-1} W E)^{-1}$ e $r_1 = r_d - X^{-1} r_s + E^T S^{-1} r_r - E^T S^{-1} W r_u$

$$\begin{pmatrix} -D^{-1} & A^T \\ A & \delta I \end{pmatrix} \begin{pmatrix} dx \\ dy \end{pmatrix} = \begin{pmatrix} r_1 \\ r_p \end{pmatrix} \tag{17}$$

- Sistema Normal

$$(ADA^T + \delta I) dy = r_p + AD(r_d - X^{-1} r_s + E^T S^{-1} r_r - E^T S^{-1} W r_u) = r_p + ADr_1 \tag{18}$$

- Equivalência a Sub-Problema

$$\begin{array}{lll} (P_\delta) & \text{Min} & c^T x + \frac{\delta}{2} y^T y + \frac{\delta}{2} w^T w \\ & \text{s.a.} & Ax + \delta y = b \\ & & Ex + v - \delta w = u \\ & & (x, v) \geq 0,\ y \text{ livre},\ w \geq 0 \end{array}$$

# Conteúdo:

# Nova abordagem

- Aurélio: Base por fatoração $LU$ rectangular de $A$
- Porfirio: Base por fatoração $LU$ rectangular de $A^T$
- Melhor condicionamento da Base

| Nome do Problema | LUret Número de Condição | LUstd Número de Condição |
|---|---|---|
| afiro | 118.9738 | 33.3923 |
| fit1p | 5.0716e+03 | 4.0253e+03 |
| fit2p | 2.2670e+04 | 5.9354e+03 |
| grow22 | 1.9056e+17* | 653.0329 |
| israel | 1.7269e+07* | 2.9008e+04 |
| kb2 | 2.8085e+03 | 2.8085e+03 |
| Maros-r7 | 4.1548e+06* | 187.3688 |

# Nova abordagem

- Aurélio: Base por fatoração $LU$ rectangular de $A$
- Porfirio: Base por fatoração $LU$ rectangular de $A^T$
- Melhor condicionamento da Base

| Nome do Problema | LUret Número de Condição | LUstd Número de Condição |
|---|---:|---:|
| afiro | 118.9738 | 33.3923 |
| fit1p | 5.0716e+03 | 4.0253e+03 |
| fit2p | 2.2670e+04 | 5.9354e+03 |
| grow22 | 1.9056e+17* | 653.0329 |
| israel | 1.7269e+07* | 2.9008e+04 |
| kb2 | 2.8085e+03 | 2.8085e+03 |
| Maros-r7 | 4.1548e+06* | 187.3688 |

# Nova abordagem

- Aurélio: Base por fatoração $LU$ rectangular de $A$
- Porfirio: Base por fatoração $LU$ rectangular de $A^T$
- Melhor condicionamento da Base

| Nome do Problema | LUret Número de Condição | LUstd Número de Condição |
|---|---:|---:|
| afiro | 118.9738 | 33.3923 |
| fit1p | 5.0716e+03 | 4.0253e+03 |
| fit2p | 2.2670e+04 | 5.9354e+03 |
| grow22 | 1.9056e+17* | 653.0329 |
| israel | 1.7269e+07* | 2.9008e+04 |
| kb2 | 2.8085e+03 | 2.8085e+03 |
| Maros-r7 | 4.1548e+06* | 187.3688 |

# Implementação UMFPACK e AMD

| | LUstd | LUret |
|---|---|---|
| Problemas abordados | 179 | 179 |
| Problemas resolvidos | 141 | 115 |
| Só por cada método | 32 | 6 |
| Não resolvidos | 38 | 64 |
| Prob. c/outro erro | 1 | 16 |
| Resolvidos Luret e Lustd | 109 | 109 |
| Tempo total dos 109 problemas | 4563.98 | 4497.81 |

- Enquanto a tempos não existe uma diferença significativa a favor de um dos métodos
- LUstd é mais robusto que LUret, embora que LUstd resolvió quase todos os problemas de fort#, (não são problemas grandes)
- 16 problemas com LUret falharam por outras razões como Wrong Branch, falha de segmentação no Linux, etc.
- LUstd tem problemas de fatoração para problemas maiores como kra30a, ste36a. precisa melhorar algoritmo LUstd.

# Implementação UMFPACK e AMD

| | LUstd | LUret |
|---|---|---|
| Problemas abordados | 179 | 179 |
| Problemas resolvidos | 141 | 115 |
| Só por cada método | 32 | 6 |
| Não resolvidos | 38 | 64 |
| Prob. c/outro erro | 1 | 16 |
| Resolvidos Luret e Lustd | 109 | 109 |
| Tempo total dos 109 problemas | 4563.98 | 4497.81 |

- Enquanto a tempos não existe uma diferença significativa a favor de um dos métodos
- LUstd é mais robusto que LUret, embora que LUstd resolvió quase todos os problemas de fort#, (não são problemas grandes)
- 16 problemas com LUret falharam por outras razões como Wrong Branch, falha de segmentação no Linux, etc.
- LUstd tem problemas de fatoração para problemas maiores como kra30a, ste36a. precisa melhorar algoritmo LUstd.

# Implementação UMFPACK e AMD

| | LUstd | LUret |
|---|---|---|
| Problemas abordados | 179 | 179 |
| Problemas resolvidos | 141 | 115 |
| Só por cada método | 32 | 6 |
| Não resolvidos | 38 | 64 |
| Prob. c/outro erro | 1 | 16 |
| Resolvidos Luret e Lustd | 109 | 109 |
| Tempo total dos 109 problemas | 4563.98 | 4497.81 |

- Enquanto a tempos não existe uma diferença significativa a favor de um dos métodos
- LUstd é mais robusto que LUret, embora que LUstd resolvió quase todos os problemas de fort#, (não são problemas grandes)
- 16 problemas com LUret falharam por outras razões como Wrong Branch, falha de segmentação no Linux, etc.
- LUstd tem problemas de fatoração para problemas maiores como kra30a, ste36a. precisa melhorar algoritmo LUstd.

# Implementação UMFPACK e AMD

| | LUstd | LUret |
|---|---|---|
| Problemas abordados | 179 | 179 |
| Problemas resolvidos | 141 | 115 |
| Só por cada método | 32 | 6 |
| Não resolvidos | 38 | 64 |
| Prob. c/outro erro | 1 | 16 |
| Resolvidos Luret e Lustd | 109 | 109 |
| Tempo total dos 109 problemas | 4563.98 | 4497.81 |

- Enquanto a tempos não existe uma diferença significativa a favor de um dos métodos
- LUstd é mais robusto que LUret, embora que LUstd resolvió quase todos os problemas de fort#, (não são problemas grandes)
- 16 problemas com LUret falharam por outras razões como Wrong Branch, falha de segmentação no Linux, etc.
- LUstd tem problemas de fatoração para problemas maiores como kra30a, ste36a. precisa melhorar algoritmo LUstd.

# Problema

Acabou memoria de meu computador para problemas maiores a 130 mil variáveis!!!!

# Bibliografia


- Hall J. Al-Jeiroudi G., Gondzio J., *Precontioning indefite system in interior point methods for large scale linear optimization*, Optimization Methods and Software **23** (2008), 345–363.
- Luca Bergamaschi, Jacek Gondzio, and Giovanni Zilli, *Preconditioning indefinite systems in interior point methods for optimization*, Computational Optimization and Applications **28** (2004), no. 2, 149–171.
- Zilli G. Bergamaschi L., Gonszio J., *Precontioning indefite system in interior point methods for optimization*, Computational Optimization and Application **28** (2004), 149–171, Kluwer Academic Publishers – Netherlands.
- Toh K. Chai J., *Preconditioning and iterative solution of symetric indefinite linear system arising from interior point methods for linear programming*, Computational Optimization and Applications **36** (2007), 221–247.

# Bibliografia


- Hall J. Al-Jeiroudi G., Gondzio J., *Precontioning indefite system in interior point methods for large scale linear optimization*, Optimization Methods and Software **23** (2008), 345–363.

- Luca Bergamaschi, Jacek Gondzio, and Giovanni Zilli, *Preconditioning indefinite systems in interior point methods for optimization*, Computational Optimization and Applications **28** (2004), no. 2, 149–171.

- Zilli G. Bergamaschi L., Gonszio J., *Precontioning indefite system in interior point methods for optimization*, Computational Optimization and Application **28** (2004), 149–171, Kluwer Academic Publishers – Netherlands.

- Toh K. Chai J., *Preconditioning and iterative solution of symetric indefinite linear system arising from interior point methods for linear programming*, Computational Optimization and Applications **36** (2007), 221–247.

# Bibliografia


- Hall J. Al-Jeiroudi G., Gondzio J., *Precontioning indefite system in interior point methods for large scale linear optimization*, Optimization Methods and Software **23** (2008), 345–363.
- Luca Bergamaschi, Jacek Gondzio, and Giovanni Zilli, *Preconditioning indefinite systems in interior point methods for optimization*, Computational Optimization and Applications **28** (2004), no. 2, 149–171.
- Zilli G. Bergamaschi L., Gonszio J., *Precontioning indefite system in interior point methods for optimization*, Computational Optimization and Application **28** (2004), 149–171, Kluwer Academic Publishers – Netherlands.
- Toh K. Chai J., *Preconditioning and iterative solution of symetric indefinite linear system arising from interior point methods for linear programming*, Computational Optimization and Applications 36 (2007), 221–247.

# Bibliografia


- Hall J. Al-Jeiroudi G., Gondzio J., *Precontioning indefite system in interior point methods for large scale linear optimization*, Optimization Methods and Software **23** (2008), 345–363.
- Luca Bergamaschi, Jacek Gondzio, and Giovanni Zilli, *Preconditioning indefinite systems in interior point methods for optimization*, Computational Optimization and Applications **28** (2004), no. 2, 149–171.
- Zilli G. Bergamaschi L., Gonszio J., *Precontioning indefite system in interior point methods for optimization*, Computational Optimization and Application **28** (2004), 149–171, Kluwer Academic Publishers – Netherlands.
- Toh K. Chai J., *Preconditioning and iterative solution of symetric indefinite linear system arising from interior point methods for linear programming*, Computational Optimization and Applications **36** (2007), 221–247.

# Bibliografia

Timothy A Davis and Iain S Duff, *An unsymmetric-pattern multifrontal method for sparse lu factorization*, SIAM Journal on Matrix Analysis and Applications **18** (1997), no. 1, 140–158.

Luenberger David G., *Linear and nonlinear programming*, 2nd ed., Springer, Standford University USA, 2005.

Gondzio J., *Interior point method 25 year later*, European Journal of Operational Research (2011), 12–34, Elsevier doi 10.1016/j.ejor.2011.09.017 http://maths.ed.ac.uk/~gondzio.

Oliveira Aurelio R. L., *Mt503: Programação linear*, Aulas de Pósgraduação em Matemática Aplicada no IMECC–UNICAMP, Ago-Dez 2011.

# Bibliografia


- Timothy A Davis and Iain S Duff, *An unsymmetric-pattern multifrontal method for sparse lu factorization*, SIAM Journal on Matrix Analysis and Applications **18** (1997), no. 1, 140–158.
- Luenberger David G., *Linear and nonlinear programming*, 2nd ed., Springer, Standford University USA, 2005.
- Gondzio J., *Interior point method 25 year later*, European Journal of Operational Research (2011), 12–34, Elsevier doi 10.1016/j.ejor.2011.09.017 http://maths.ed.ac.uk/~gondzio.
- Oliveira Aurelio R. L., *Mt503: Programação linear*, Aulas de Pósgraduação em Matemática Aplicada no IMECC–UNICAMP, Ago-Dez 2011.

# Bibliografia


- Timothy A Davis and Iain S Duff, *An unsymmetric-pattern multifrontal method for sparse lu factorization*, SIAM Journal on Matrix Analysis and Applications **18** (1997), no. 1, 140–158.
- Luenberger David G., *Linear and nonlinear programming*, 2nd ed., Springer, Standford University USA, 2005.
- Gondzio J., *Interior point method 25 year later*, European Journal of Operational Research (2011), 12–34, Elsevier doi 10.1016/j.ejor.2011.09.017 http://maths.ed.ac.uk/~gondzio.
- Oliveira Aurelio R. L., *Mt503: Programação linear*, Aulas de Pósgraduação em Matemática Aplicada no IMECC–UNICAMP, Ago-Dez 2011.

# Bibliografia


- Timothy A Davis and Iain S Duff, *An unsymmetric-pattern multifrontal method for sparse lu factorization*, SIAM Journal on Matrix Analysis and Applications **18** (1997), no. 1, 140–158.
- Luenberger David G., *Linear and nonlinear programming*, 2nd ed., Springer, Standford University USA, 2005.
- Gondzio J., *Interior point method 25 year later*, European Journal of Operational Research (2011), 12–34, Elsevier doi 10.1016/j.ejor.2011.09.017 http://maths.ed.ac.uk/~gondzio.
- Oliveira Aurelio R. L., *Mt503: Programação linear*, Aulas de Pósgraduação em Matemática Aplicada no IMECC–UNICAMP, Ago-Dez 2011.

# Bibliografia


S. Mehrotra, *On the implementation of a primal-dual interior point method*, SIAM Journal on Optimization **2** (1992), 575–601.

Sorensen D. C. Oliveira A. R. L., *A new class of preconditioners for large-scale linear systems from interior point methods for linear programming*, Linear Algebra and its Applications **394** (2005), 1–24, England http://www.ime.unicamp.br/~aurelio/artigos/cor.pdf.

Mehrotra S., *Asymptotic convergence in a generalized-predictor-corrector method*, Manuscript, Dept. of Industrial Engineering and Management Sciences, 1992, Northwestern University, Evanston, IL 60208, USA.

Bocanegra Silvana, *Algoritmos de newton-krylov precondicionados para métodos de pontos interiores*, Impresso, Universidade Federal de Minas Gerais, Belo Horizonte, Dezembro 2005.

# Bibliografia

S. Mehrotra, *On the implementation of a primal-dual interior point method*, SIAM Journal on Optimization **2** (1992), 575–601.

Sorensen D. C. Oliveira A. R. L., *A new class of preconditioners for large-scale linear systems from interior point methods for linear programming*, Linear Algebra and its Applications **394** (2005), 1–24, England http://www.ime.unicamp.br/∼aurelio/artigos/cor.pdf.

Mehrotra S., *Asymptotic convergence in a generalized-predictor-corrector method*, Manuscript, Dept. of Industrial Engineering and Management Sciences, 1992, Northwestern University, Evanston, IL 60208, USA.

Bocanegra Silvana, *Algoritmos de newton-krylov precondicionados para métodos de pontos interiores*, Impresso, Universidade Federal de Minas Gerais, Belo Horizonte, Dezembro 2005.

# Bibliografia


S. Mehrotra, *On the implementation of a primal-dual interior point method*, SIAM Journal on Optimization **2** (1992), 575–601.

Sorensen D. C. Oliveira A. R. L., *A new class of preconditioners for large-scale linear systems from interior point methods for linear programming*, Linear Algebra and its Applications **394** (2005), 1–24, England http://www.ime.unicamp.br/∼aurelio/artigos/cor.pdf.

Mehrotra S., *Asymptotic convergence in a generalized-predictor-corrector method*, Manuscript, Dept. of Industrial Engineering and Management Sciences, 1992, Northwestern University, Evanston, IL 60208, USA.

Bocanegra Silvana, *Algoritmos de newton-krylov precondicionados para métodos de pontos interiores*, Impresso, Universidade Federal de Minas Gerais, Belo Horizonte, Dezembro 2005.

# Bibliografia

S. Mehrotra, *On the implementation of a primal-dual interior point method*, SIAM Journal on Optimization **2** (1992), 575–601.

Sorensen D. C. Oliveira A. R. L., *A new class of preconditioners for large-scale linear systems from interior point methods for linear programming*, Linear Algebra and its Applications **394** (2005), 1–24, England http://www.ime.unicamp.br/∼aurelio/artigos/cor.pdf.

Mehrotra S., *Asymptotic convergence in a generalized-predictor-corrector method*, Manuscript, Dept. of Industrial Engineering and Management Sciences, 1992, Northwestern University, Evanston, IL 60208, USA.

Bocanegra Silvana, *Algoritmos de newton-krylov precondicionados para métodos de pontos interiores*, Impresso, Universidade Federal de Minas Gerais, Belo Horizonte, Dezembro 2005.

# Agradecimento

OBRIGADO

Pela sua Atenção

Porfirio Suñagua Salgado

# Agradecimento

OBRIGADO

Pela sua Atenção

Porfirio Suñagua Salgado