

# Projeto Inverso de Bocal Convergente-Divergente

POR ALLAN MOREIRA DE CARVALHO

Universidade Federal do ABC

*24 agosto 2022*

# 1 Introdução

O projeto inverso consiste em, à partir de determinada distribuição do escoamento, determinar-se qual a geometria que o produz. Nesse tipo de projeto os métodos de otimização desempenham papel central. Nesse trabalho, diferentes algoritmos e estratégias de otimização serão analisadas para a solução do problema inverso de um bocal convergente-divergente, inspirado no trabalho [\[1\]](#).

## 2 Objetivo

Os objetivos desse trabalho são:

- Aplicar os métodos de otimização determinísticos disponíveis na biblioteca `scipy` para a solução do problema inverso de projeto aerodinâmico de um bocal convergente-divergente.
- Avaliar a performance dos métodos em dois cenários, utilizando uma parametrização geométrica polinomial com apenas uma variável independente e em outro cenário utilizando uma *spline* cúbica.

## 3 Fundamentação Teórica

### 3.1 Método Numérico para solução do escoamento

Equações de Euler quasi-unidimensionais, descritas em sua forma vetorial como

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{1}{S} \frac{\partial (\mathbf{F} S)}{\partial x} = \mathbf{Q} \quad (1)$$

$$\mathbf{U} = \begin{bmatrix} \rho \\ \rho u \\ e \end{bmatrix}, \mathbf{F} = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ (e + p) u \end{bmatrix}, \mathbf{Q} = \begin{bmatrix} 0 \\ \frac{p}{S} \frac{dS}{dx} \\ 0 \end{bmatrix}$$

o sistema é fechado com a hipótese de gás ideal

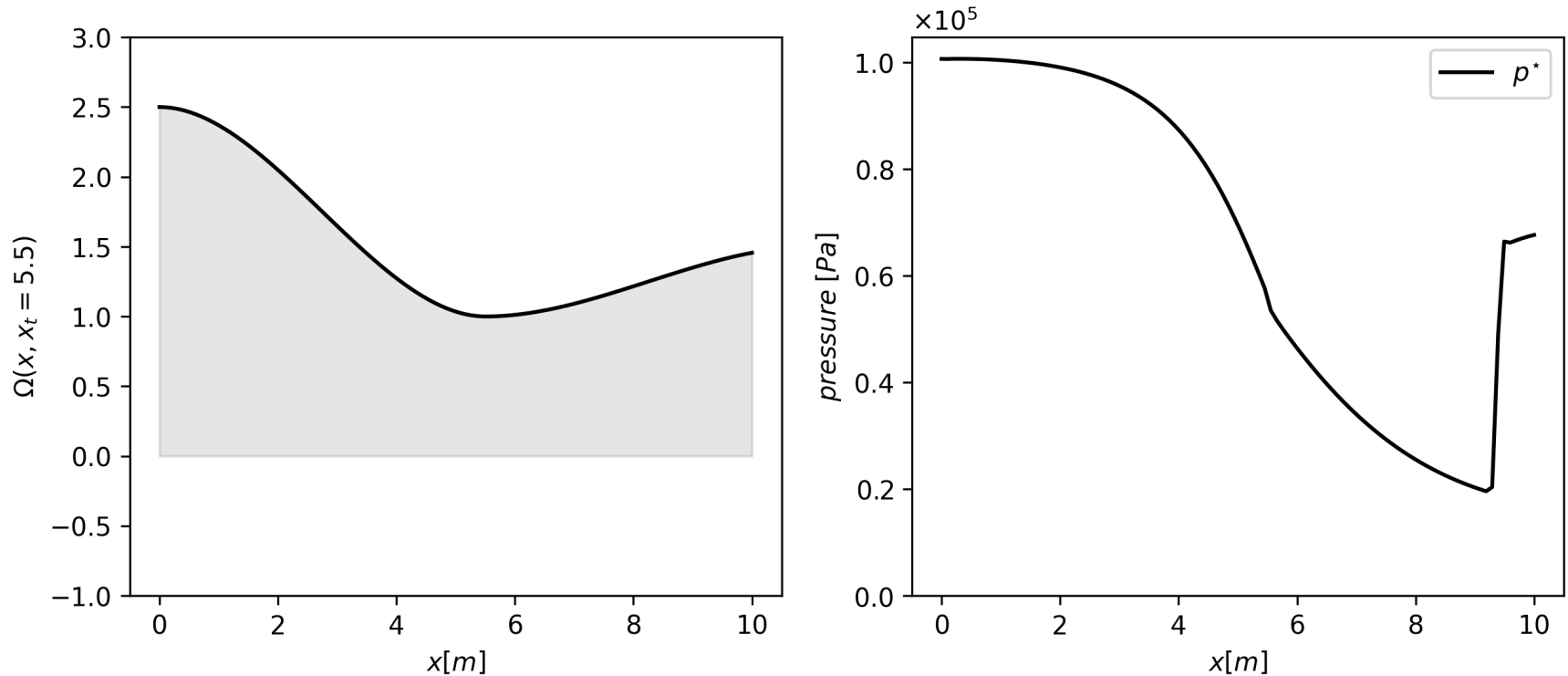
$$e = \rho \left( e_i + \frac{u^2}{2} \right) \quad p = \rho R T = (\gamma - 1) \rho e_i$$

O domínio computacional foi discretizado em 99 elementos finitos, e o critério de convergência das simulações numéricas foi o resíduo  $< 1e - 6$ . O objetivo final é otimizar apenas a geometria e portanto todos os casos considerados consideram um bocal operando com Ar ( $\gamma = 1.4, R = 287 J / kg / K$ ) e nas condições de contorno

$$M_{\text{in}} = 0.01$$

$$p_{0\text{in}} = 104074.0 \text{ Pa}$$

$$p_e = 0.65 \cdot p_{0\text{in}}$$



**Figure 1.** À esquerda a geometria que produz a distribuição de pressão objetivo.

## 3.2 Método de otimização, **trust-constr**

O método [2] pode ser visto como uma combinação de programação linear sequencial quadrática, utilizada para tratar possíveis restrições não lineares e técnicas como ponto interiouro e *trust-region* no tratamento de problemas convexos e não convexos. Dado um problema genérico de otimização, dado por

$$\underset{\mathbf{x}}{\text{minimizar}} \ f(\mathbf{x})$$

, sujeito à

$$h(\mathbf{x}) = 0$$

$$g(\mathbf{x}) \leq 0$$

O problema é reformulado em termos de subproblemas, resolvíveis por programação linear sequencial quadrática, cada um dos subproblemas assume a forma

$$\underset{\mathbf{x}, \mathbf{s}}{\text{minimizar}} \ f(\mathbf{x}, \mathbf{s}) - \mu \sum_{i=1}^m \ln s_i$$

, sujeito à

$$h(\mathbf{x}) = 0$$

$$g(\mathbf{x}) + \mathbf{s} = 0$$

Onde, fazendo  $\mu$  tender à zero, recupera-se o problema original. Para tratar as restrições de desigualdade, foram introduzidas as variáveis de folga  $\mathbf{s}$ . Para determinar a solução do problema define-se o Lagrangeano

$$\mathcal{L}(\mathbf{x}, \mathbf{s}, \boldsymbol{\lambda}_h, \boldsymbol{\lambda}_g) = f(\mathbf{x}) - \mu \sum_{i=1}^m \ln s_i + \boldsymbol{\lambda}_h^T h(\mathbf{x}) + \boldsymbol{\lambda}_g^T (g(\mathbf{x}) + \mathbf{s})$$

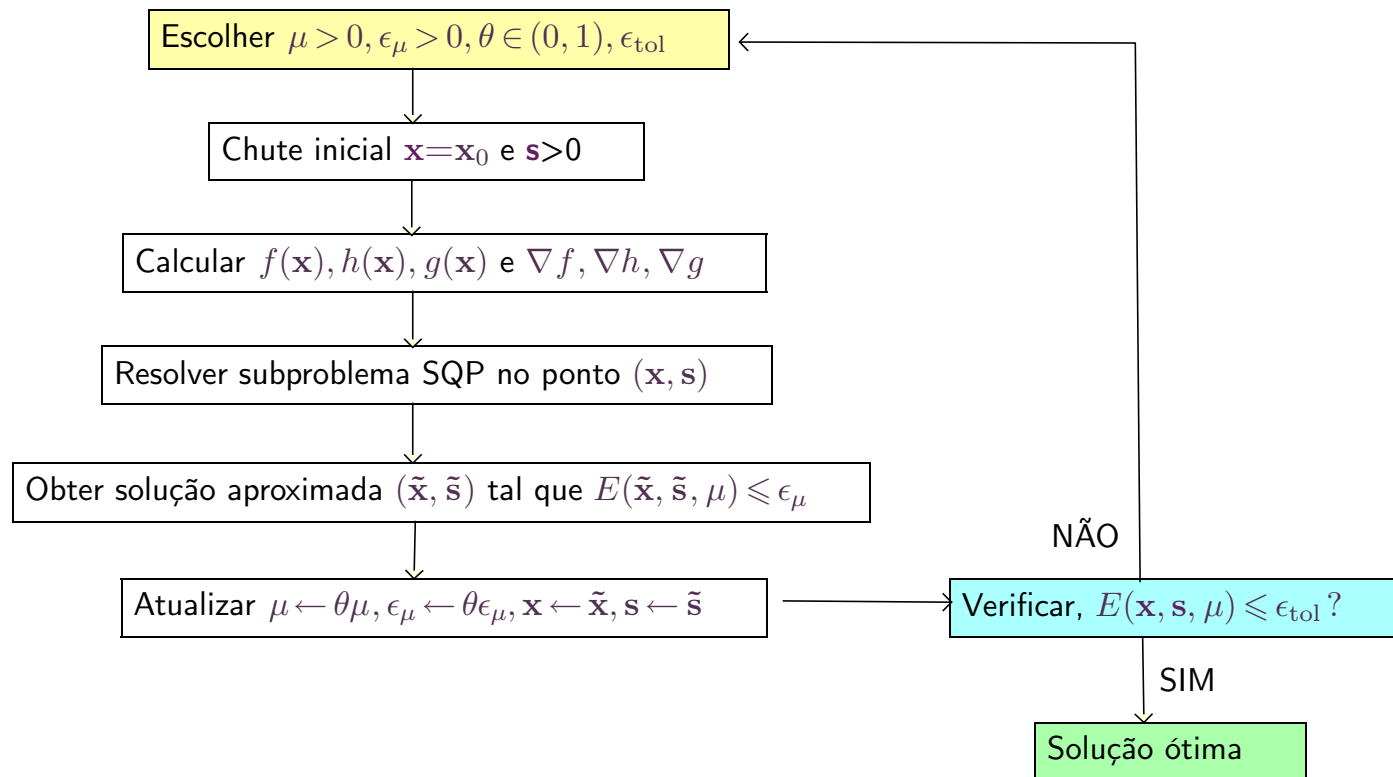
O subproblema é então solucionado de maneira aproximada, com uma precisão arbitrário  $\epsilon_\mu$ , que é decrementada a cada iteração, de modo que  $E(\tilde{\mathbf{x}}, \tilde{\mathbf{s}}; \mu) \leq \epsilon_\mu$ , onde

$$E(x, s; \mu) = \max (\|\nabla f(\mathbf{x}) + A_h(\mathbf{x})\boldsymbol{\lambda}_h + A_g(\mathbf{x})\boldsymbol{\lambda}_g\|_\infty, \|S\boldsymbol{\lambda}_g - \mu e\|_\infty, \|h(\mathbf{x})\|_\infty, \|g(\mathbf{x}) + \mathbf{s}\|_\infty)$$

, onde  $A_h$  e  $A_g$  são as matrizes de gradientes das restrições, dadas por

$$A_h(\mathbf{x}) = [\nabla h^1(\mathbf{x}), \dots, \nabla h^t(\mathbf{x})] \quad A_g(\mathbf{x}) = [\nabla g^1(\mathbf{x}), \dots, \nabla g^m(\mathbf{x})]$$

Para manter a dimensionalidade correta,  $e = [1, \dots, 1]^T$ , e  $S = \text{diag}(s^1, \dots, s^m)$ .



**Figure 2.** Fluxograma para algoritmo de ponto interior trust-constr.



## 4 Formulação do Problema

Nesse trabalho, pretende-se obter a geometria  $\Omega$  de um bocal convergente-divergente que forneça uma determinada distribuição pressão alvo  $p^*$ . A métrica de erro utilizada é a média da norma  $L_2$ , e a função objetivo  $J$ , pode então ser escrita como

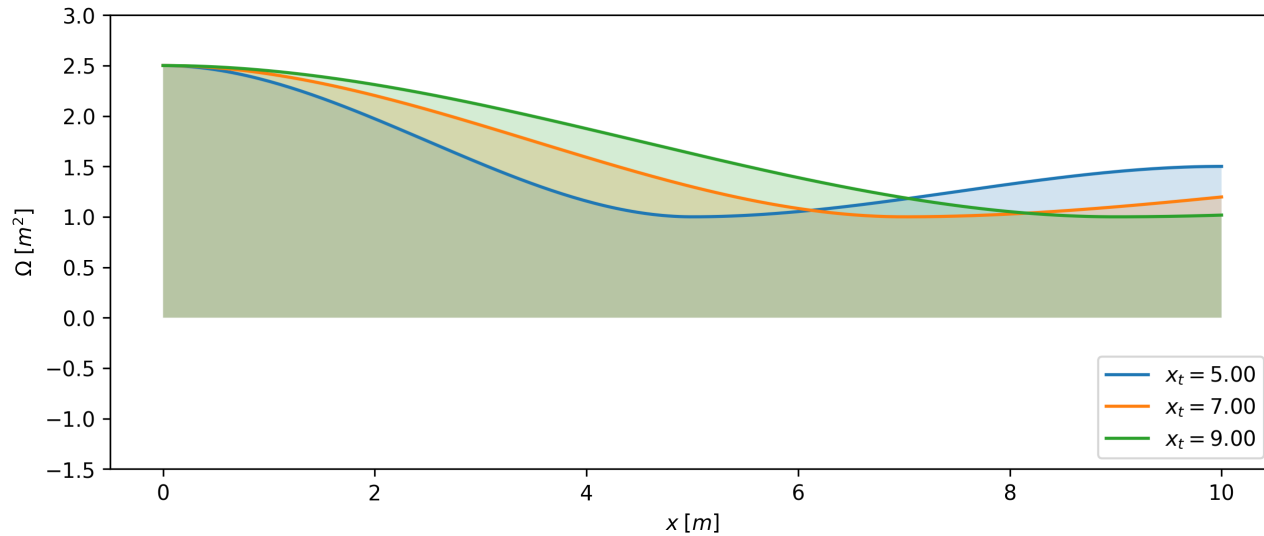
$$J(\Omega) = \frac{1}{N} \sqrt{\sum_{i=1}^N (p^* - p)^2}$$

Foram propostas duas formas de solucionar o problema inverso, utilizando um polinômio com apenas um parâmetro de controle, e uma *spline* cúbica com 8 pontos. Em ambos os casos foram aplicadas restrições nos algoritmos que assim permitem, são eles: Nelder-Mead, L-BFGS-B, TNC, SLSQP, Powell, e trust-constr.

## 4.1 Parametrização com polinômio

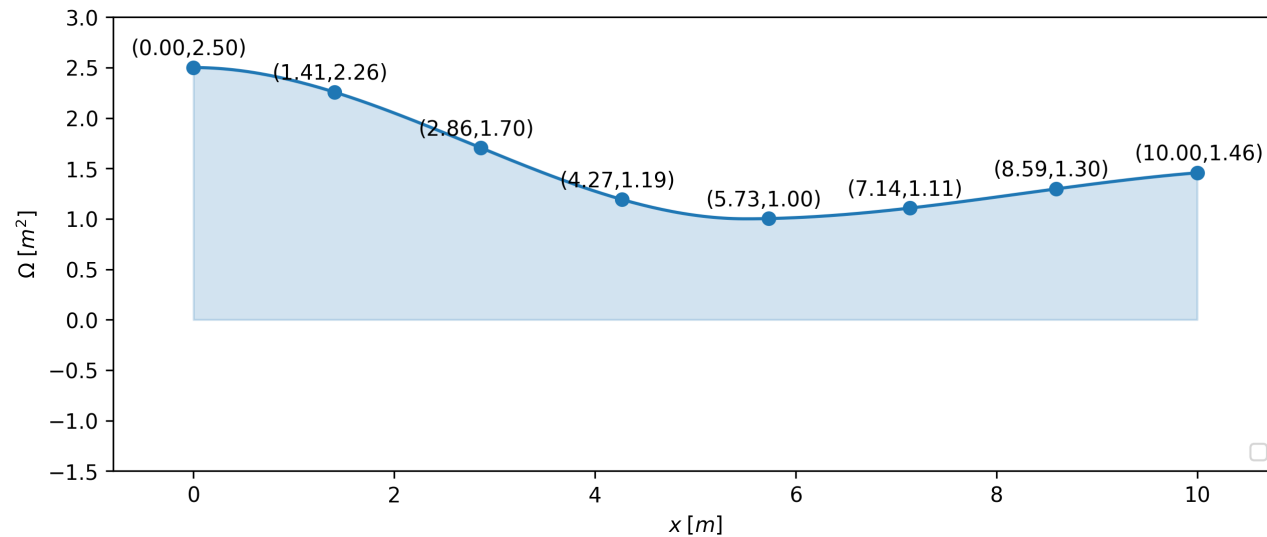
O polinômio proposto em [3], determina uma distribuição de áreas de seção transversal

$$\Omega(x_t) = \begin{cases} 2.5 + 3\left(\frac{x}{x_t} - 1.5\right)\left(\frac{x}{x_t}\right)^2 & , x < x_t \\ 3.5 - \left(\frac{x}{x_t}\right)\left(6 - 4.5\left(\frac{x}{x_t}\right) + \left(\frac{x}{x_t}\right)^2\right) & , x \geq x_t \end{cases} \quad (2)$$



**Figure 3.** Distribuição de área polinomial, determinada pela equação (2).

## 4.2 Parametriação com *Spline* cúbica



**Figure 4.** Parametrização utilizando uma *spline* cúbica com 8 pontos de controle.

O problema de otimização nesse caso é dado por

$$\underset{x_t}{\text{minimizar}} J(\Omega(\mathbf{x}))$$

, sujeito à

$$0 < x_i \leq 2.5 \quad i = \{1, 2, \dots, 8\}$$

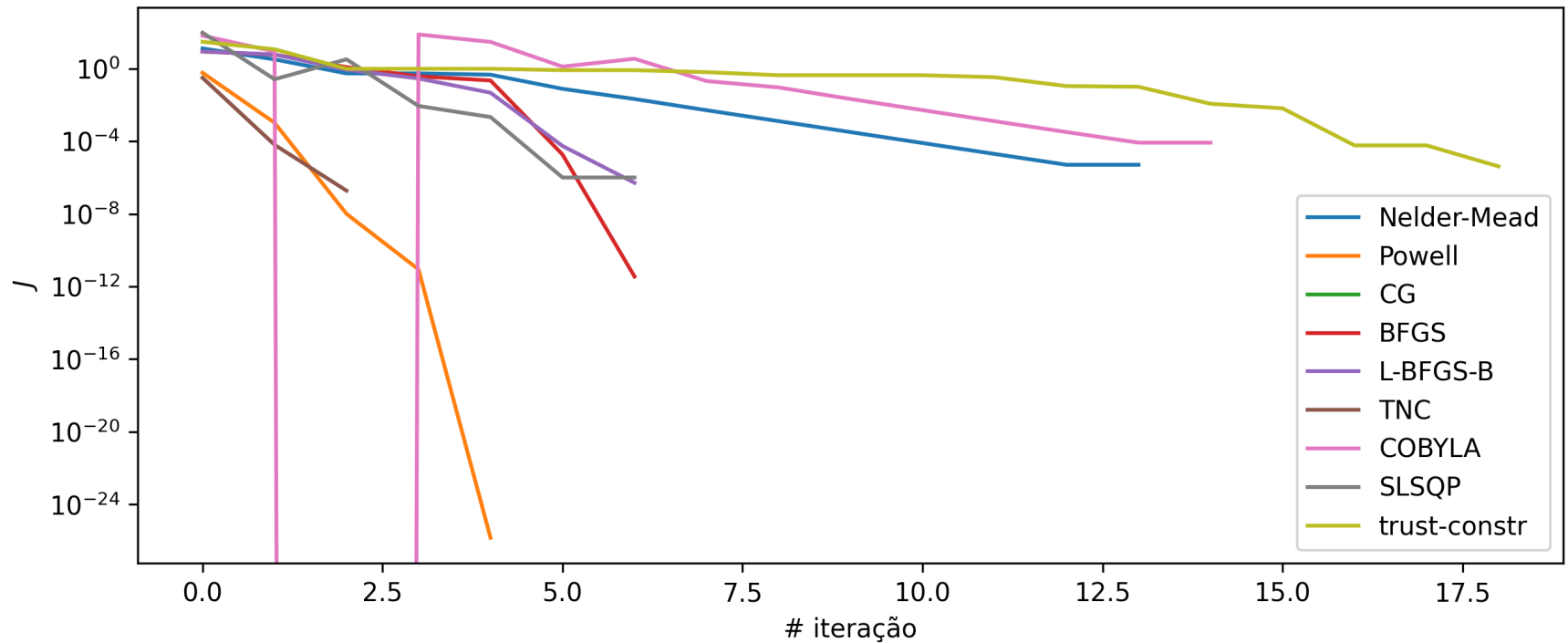
## 5 Implementação Numérica

Os métodos de otimização utilizados são parte da biblioteca `scipy`. Os seguintes algoritmos determinísticos foram analisados:

- Nelder-Mead - Algoritmo simplex, [4, 5, 6].
- Powell - Método de Powell modificado, [7, 8].
- CG - Variant of Fletcher-Reeves, [9].
- BFGS - Método quasi-Newton de Broyden, Fletcher, Goldfarb, and Shanno, [9].
- L-BFGS-B - Modificação do método BFGS para problemas restritos e pouca memória [10].
- TNC - Newton truncado, [9, 11].
- COBYLA - Método de busca direta, otimização com restrição por aproximação linear, [12, 13, 14].
- SLSQP - Programação linear sequencial, mínimos quadrados, [15].
- `trust-constr` - Algoritmo de região confiável, o `scipy` alterna as implementações, no caso de restrições de igualdade, utiliza *Byrd-Omojokun Trust-Region SQP* [16, 9], para restrições de desigualdade muda para *trust-region interior point* [2].

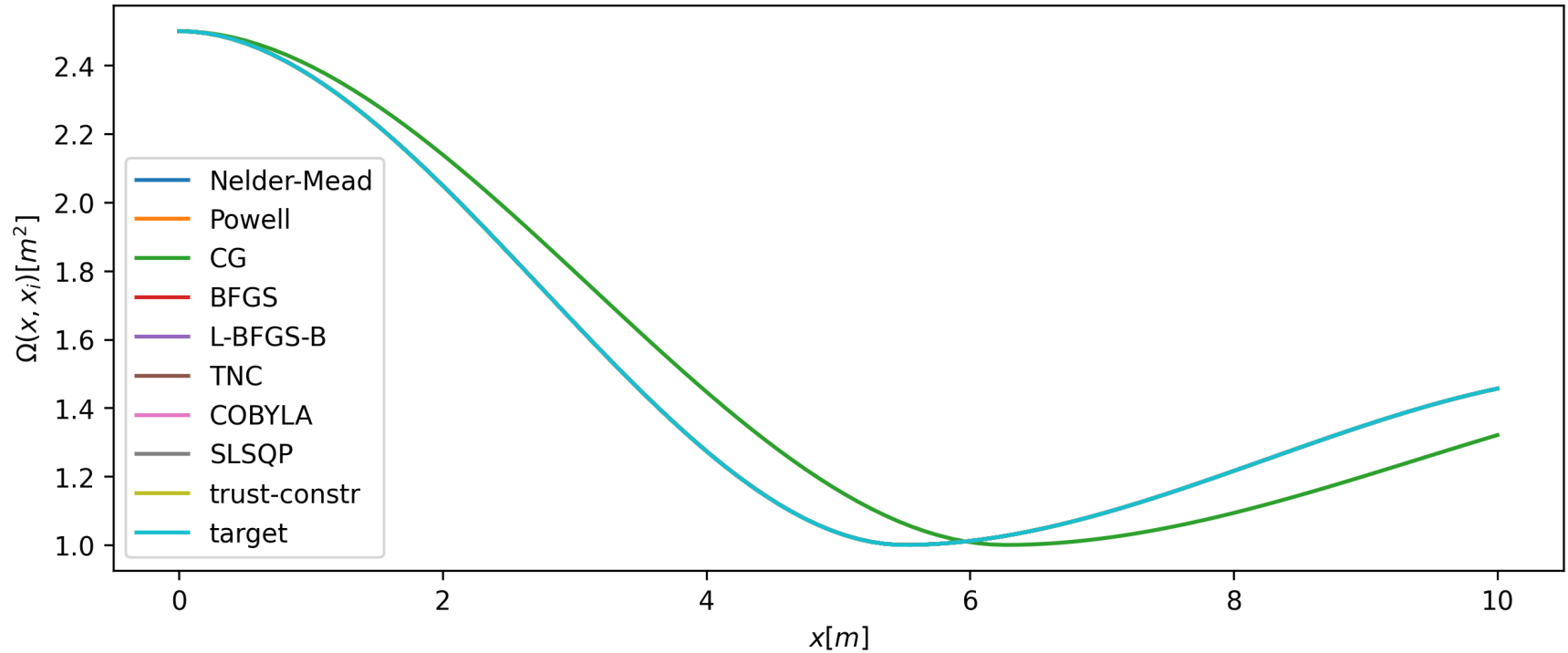
Para o desenvolvimento dessa atividade foi desenvolvido um script em Python, as dependências para utilização do script são: `matplotlib`, `numpy`, `scipy`, `pandas` e o módulo `solvers`, desenvolvido para executar o solver para o escoamento quasi-unidimensional. O script é rodado em linha de comando, uma descrição de cada parâmetro pode ser acessada com o comando: `python optXti.py -h`. Os códigos necessários para execução estão disponíveis no repositório

## 6 Resultados



**Figure 5.** Convergência dos métodos analisados utilizando uma parametrização polinomial com apenas uma variável de projeto.

De maneira geral, exceto o algoritmo CG, todos os demais recuperaram muito precisamente a geometria alvo, como mostra a Figura 6.



**Figure 6.** Resultados da otimização para o modelo paramétrico com uma única variável.

## 6.1 Utilizando parametrização com *spline* cúbica

Método	# avaliações	# iterações	$J$
SLSQP	233	22	0.02003277456604602
CG	7715	389	0.011622164354443488
COBYLA	422	422	0.24756052150364757
BFGS	1194	69	0.008818300170018417
L-BFGS-B	279	24	0.1267727940841186
Powell	2635	26	0.009481766589186796
trust-constr	1620	170	0.011477187238382045
TNC	909	15	0.4344095665774871
Nelder-Mead	580	380	0.14700252215240825

**Table 1.** Performance dos método para uma parametrização com *spline* cúbica.



### 6.1.1 Restrições mais rígidas

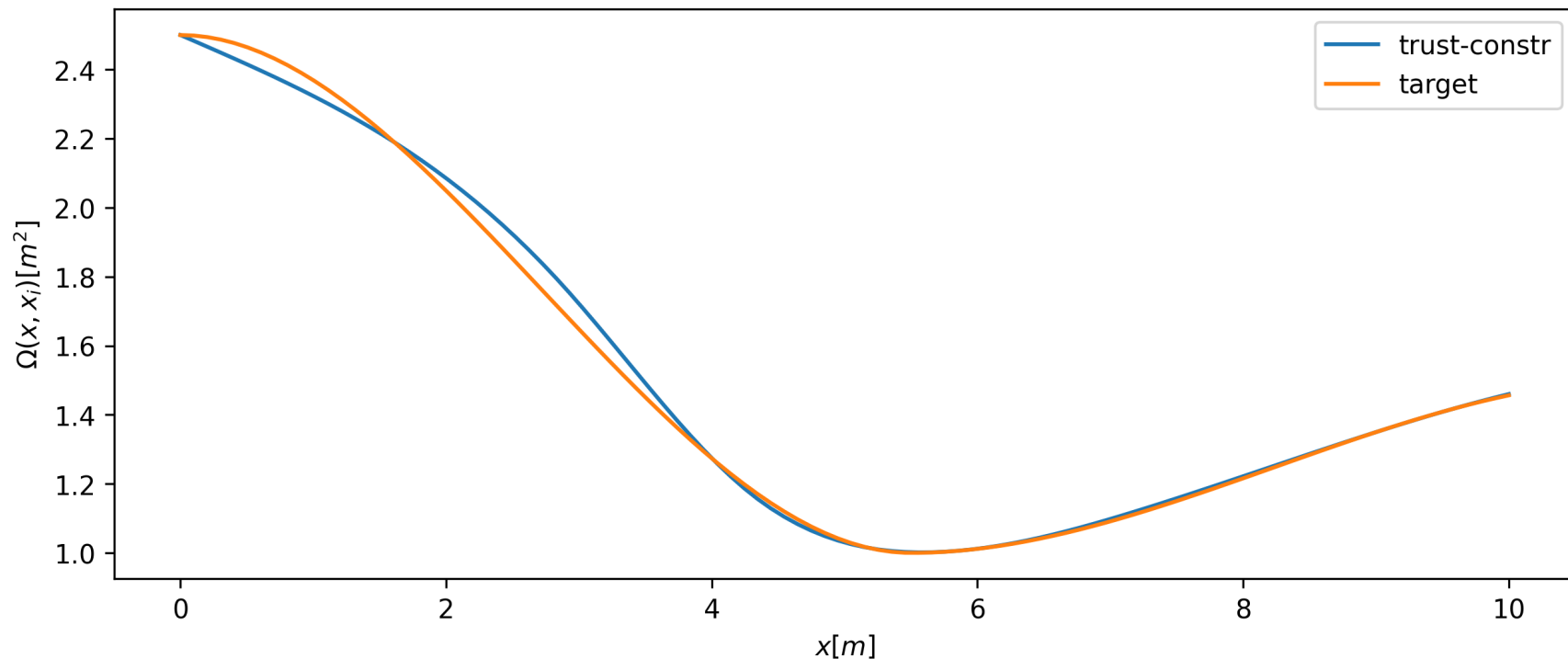
A fim de tornar as dificuldades relacionadas ao problema inverso mal posto, foram aplicadas condições de restrições mais rígidas, fixando-se as áreas na entrada e na saída do bocal para os valores alvo, nesse caso o problema de otimização torna-se

$$\underset{x_t}{\text{minimizar}} J(\Omega(x_t))$$

, sujeito à

$$\begin{aligned} x_1 &= 2.5 \\ 0 < x_i &\leq 2.5 \quad i = \{2, \dots, 7\} \\ x_8 &= 1.5 \end{aligned}$$

## 6.2 Resultado com restrições mais apertadas



**Figure 7.** Resultado do projeto inverso utilizando o método `trust-constr` e restrições de igualdade para as áreas na entrada e na saída do bocal. O processo tomou assombrosos 8722 avaliações da função objetivo.

## 7 Conclusão

Diversos algoritmos de otimização foram analisados na solução do problema proposto, no caso do projeto inverso com apenas uma variável, exceto o algoritmo **CG**, todos os demais apresentaram um excelente desempenho, recuperando de maneira muito precisa a geometria objetivo. No caso da otimização multivariável o problema mostrou-se bastante desafiador, uma vez que os algoritmos convergem para pontos de mínimo que não são representam à geometria objetivo, tal comportamento deve-se ao problema inverso em questão ser mal posto por natureza. Para aliviar esse problema restrições mais severas foram impostas, e desse modo o algortimo **trust-constr**, que fora brevemente descrito, pode recuperar a geometria alvo, ainda que, de maneira menos eficiente que a parametrização polinomial, como esperado, o custo da otimização multivariável envolvendo solução de CFD cresce excessivamente comparado ao caso uni-variável.

# Bibliography

- [1] Markus Rumpfkeil and David Zingg. A general framework for the optimal control of unsteady flows with applications. In *45th AIAA Aerospace Sciences Meeting and Exhibit*. American Institute of Aeronautics and Astronautics, jan 2007.
- [2] Richard H. Byrd, Mary E. Hribar, and Jorge Nocedal. An interior point algorithm for large-scale nonlinear programming. *SIAM Journal on Optimization*, 9(4):877–900, jan 1999.
- [3] Geum-Su Yeom and Jung-II Choi. Efficient exact solution procedure for quasi-one-dimensional nozzle flows with stiffened-gas equation of state. *International Journal of Heat and Mass Transfer*, 137:523–533, jul 2019.
- [4] Fuchang Gao and Lixing Han. Implementing the nelder-mead simplex algorithm with adaptive parameters. *Computational Optimization and Applications*, 51(1):259–277, may 2010.
- [5] J. A. Nelder and R. Mead. A simplex method for function minimization. *The Computer Journal*, 7(4):308–313, jan 1965.
- [6] Margaret H. Wright. Direct search methods: once scorned, now respectable. 1996.
- [7] M. J. D. Powell. An efficient method for finding the minimum of a function of several variables without calculating derivatives. *The Computer Journal*, 7(2):155–162, feb 1964.
- [8] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipes 3rd Edition: The Art of Scientific Computing*. Cambridge University Press, 3 edition, 2007.
- [9] J. Nocedal and S. Wright. *Numerical Optimization*. Springer Series in Operations Research and Financial Engineering. Springer New York, 2006.
- [10] Ciyu Zhu, Richard H. Byrd, Peihuang Lu, and Jorge Nocedal. Algorithm 778: I-BFGS-B. *ACM Transactions on Mathematical Software*, 23(4):550–560, dec 1997.
- [11] Stephen G. Nash. Newton-type minimization via the lanczos method. *SIAM Journal on Numerical Analysis*, 21(4):770–788, aug 1984.
- [12] M. J. D. Powell. Direct search algorithms for optimization calculations. *Acta Numerica*, 7:287–336, jan 1998.

- [13] M. J. D. Powell. A direct search optimization method that models the objective and constraint functions by linear interpolation. 1994.
- [14] M. J. D. Powell. A view of algorithms for optimization without derivatives 1. 2007.
- [15] D. Kraft. *A Software Package for Sequential Quadratic Programming*. Deutsche Forschungs- und Versuchsanstalt für Luft- und Raumfahrt Köln: Forschungsbericht. Wiss. Berichtswesen d. DFVLR, 1988.
- [16] Marucha Lalee, Jorge Nocedal, and Todd Plantenga. On the implementation of an algorithm for large-scale equality constrained optimization. *SIAM Journal on Optimization*, 8(3):682–706, aug 1998.