

**Um estudo comparativo da mistura
de metamodelos em otimização global
para problemas de aerodinâmica**

Cristiano Alves Bau

DISSERTAÇÃO APRESENTADA
AO
INSTITUTO DE MATEMÁTICA E ESTATÍSTICA
DA
UNIVERSIDADE DE SÃO PAULO
PARA
OBTENÇÃO DO TÍTULO
DE
MESTRE EM CIÊNCIAS

Programa: Matemática Aplicada
Orientador: Prof. Dr. Luis Carlos dos Santos Castro

São Paulo, janeiro de 2016

**Um estudo comparativo da mistura
de metamodelos em otimização global
para problemas de aerodinâmica**

Esta é a versão original da dissertação elaborada pelo
candidato Cristiano Alves Bau, tal como
submetida à Comissão Julgadora.

Resumo

Bau, C. A. **Um estudo comparativo da mistura de metamodelos em otimização global para problemas de aerodinâmica** 2016.

Dissertação - Instituto de Matemática e Estatística, Universidade de São Paulo, São Paulo, 2016.

Existem muitos problemas na literatura em que sua forma algébrica não é conhecida ou é tão complexa que fica difícil obter uma função na qual se consiga trabalhar. Além disto, existem aqueles problemas computacionalmente caros que podem ter uma forma algébrica definida mas demandam um custo computacional elevado para serem avaliados. Estes tipos de problemas são definidos como caixas pretas e são encontrados em diversas áreas como engenharia, finanças e agricultura. A busca por soluções ótimas de um problema deste tipo, pode demorar horas, dias ou até mesmo semanas. Com isto, metamodelos são bastante utilizados na literatura para aproximar os modelos caixas pretas e trazer um ganho computacional quando as funções caras são avaliadas. Neste trabalho, será testado a mistura destes metamodelos para problemas de aerodinâmica, como por exemplo, encontrar o menor coeficiente de arrasto sofrido por uma asa de avião de acordo com a sua geometria. Este problema pode ser considerado do tipo caixa preta devido ao cálculo do coeficiente de arrasto, pois é preciso encontrar soluções de equações diferenciais parciais que requerem um esforço computacional enorme. Com isto, a mistura de metamodelos substituirá o coeficiente de arrasto em algumas iterações durante o algoritmo de otimização, fazendo com que o processo economize em não avaliar a função computacionalmente cara várias vezes.

Palavras-chave: Aerodinâmica, Metamodelo, Otimização.

Abstract

Bau, C. A. **A comparative study of mixture of surrogate models algorithms for global optimization in aerodynamic problems.** 2016. Dissertação - Instituto de Matemática e Estatística, Universidade de São Paulo, São Paulo, 2016.

There are many problems in the literature with unknown algebraic form or with a so complex formulation that it's hard to find a function to be worked with. Besides that, there exist computationally expensive problems with known algebraic form but that demand too high computational costs to be evaluated. This sort of problems are defined as black-boxes and are found in many areas such as engineering, finance and agriculture. Searching for optimal solutions for this kind of problems can take hours, days or even weeks. Thus, surrogate models are widely used in the literature to approximate black-box models, bringing computational gains when expensive functions are evaluated. In this work a mixture of surrogate models is tested for aerodynamics problems as we consider an example of finding the smallest drag coefficient suffered by an airplane wing according to its geometry. This problem can be considered as one of the black-box type due to the calculation of the drag coefficient, as it is required to find partial differential equations solutions that demand huge computational efforts. Thereby, the mixture of surrogate models substitutes the drag coefficient in some iterations during the optimization algorithm, making the process more efficient since it doesn't evaluate the computationally expensive function many times.

Keywords: Aerodynamic, Surrogate model, Optimization.

Sumário

1	Introdução	2
2	Otimização Através da Mistura de Metamodelos	5
2.1	Definição do Problema Matemático	5
2.2	Algoritmos de Metamodelos	6
2.2.1	Modelos de Regressão Polinomial	7
2.2.2	Regressão Adaptativa Multivariada por Splines	8
2.2.3	Função Base Radial	8
2.2.4	Kriging	9
2.3	Algoritmo de Otimização Global usando Metamodelos Simples	10
2.3.1	Algoritmo	10
2.3.2	Resultados numéricos	12
2.4	Algoritmo de Otimização Global Usando Mistura Metamodelos	21
2.4.1	Atribuições Básicas de Probabilidade e Teoria Matemática da Evidência	22
2.4.2	Algoritmo	25
2.4.3	Resultados numéricos	27
3	Otimização Numérica em Aerodinâmica	33
3.1	Aerodinâmica	33
3.2	A dinâmica dos fluidos	35
3.2.1	Equações de Navier-Stokes	35
3.2.2	Método de Newton para Equações Navier-Stokes Estacionárias . . .	36
3.2.3	Coefficientes de Arrasto e Sustentação	37
3.3	FreeFem++	38
3.4	Parâmetros da forma do aerofólio	40
4	Resultados	42
5	Conclusão	43

Capítulo 1

Introdução

Não é difícil encontrar problemas em que sua estrutura interna seja desconhecida ou que seja tão complexa que fica impossível definir uma forma algébrica para o sistema. Com isto, é um grande desafio encontrar soluções ótimas para estes tipos de problemas, denominados caixas pretas. Eles requerem um custo computacional elevado e o tempo para encontrar a melhor solução aumenta consideravelmente à medida que a complexidade do sistema também aumenta. Apesar da evolução dos computadores, existem funções que podem demorar horas ou até dias para serem avaliadas devido a sua complexidade. Um algoritmo iterativo que busque encontrar o ponto ótimo e que precise calcular estas funções em cada iteração pode ser improdutivo. Desta forma, muito se tem estudado para encontrar boas aproximações para estas funções computacionalmente caras.

Gutman[7] introduziu um método visando encontrar o mínimo global de funções computacionalmente caras e não convexas num conjunto compacto de \mathbb{R}^d . Usou-se a função interpoladora de base radial para definir um modelo de utilidade e com isto encontrar o próximo candidato onde a função objetivo é avaliada. Através de alguns testes numéricos no conjunto de funções de Dixon-Szegö, o teste mostrou-se favorável em comparação com outros algoritmos de otimização global.

Jones, Schonlau e Welch[4], em EGO(Efficient Global Optimization), usaram o modelo estocástico Kriging para encontrar boas aproximações para problemas não lineares e funções multimodais. Com isto, conseguiram encontrar pontos promissores em cada iteração do algoritmo de otimização economizando com avaliações da função computacionalmente cara.

Müller[1], em sua tese de doutorado, combinou modelos de regressão polinomial, funções de base radial, regressão adaptativa multivariada por splines e kriging para substituir a função objetivo, computacionalmente cara, e encontrar pontos promissores na busca do ponto ótimo para o problema de minimizar o escoamento de fósforo em

uma bacia hidrográfica através da reforma de terras em sua volta. A técnica utilizando metamodelos alcançou ótimos resultados para vários tipos de problemas, convergindo para o ótimo global mais rápido que métodos tradicionais como GA (sigla em inglês - Genetic Algorithm), NOMAD[23] (sigla em inglês - Nonsmooth Optimization by Mesh Adaptative Direct Search) e discrete-DDS[22] (sigla em inglês - Discrete Dynamically Dimensioned Search). Veja as figuras 1.1, 1.2, 1.3 que mostram a redução de fósforo na bacia hidrográfica em 20%, 40% e 60% respectivamente.

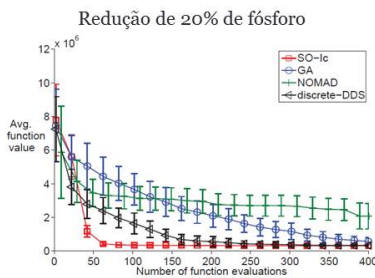


Figura 1.1: 20%

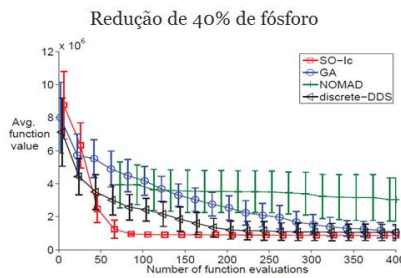


Figura 1.2: 40%

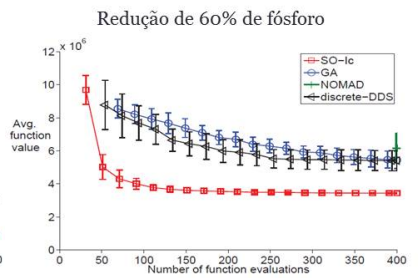


Figura 1.3: 60%

Dos problemas práticos, a indústria aeronáutica busca novas formas de reduzir o consumo de combustível através da redução do arrasto provocado pelo ar nas aeronaves. As asas e o aerofólio são fundamentais para o vôo de um avião e mexer em suas formas para diminuir o arrasto sofrido é um grande desafio pois também são responsáveis pela sustentação da aeronave no ar e um problema de minimização do arrasto precisa estar munido de restrições que mantenham uma certa sustentação. Além do mais, o cálculo do arrasto e da sustentação podem ser caracterizados como modelos de caixa preta, pois dependem das soluções de equações diferenciais parciais, que por sua vez demandam um custo computacional elevado. Assim, o problema de minimizar o arrasto na asa de um avião mantendo uma certa sustentação é tema de estudo para muitos autores que buscam soluções alternativas para que as equações computacionalmente caras sejam avaliadas o mínimo de vezes possível, deixando o método mais eficiente.

Koziel e Leifsson [24] introduziram o método SBO (Surrogate-Based Optimization) que considera malhas com discretização mais grosseiras e convergências mais relaxadas (baixa-fidelidade) para chegar em possíveis candidatos e a partir daí, avalia-se o modelo em malhas mais finas e convergências mais restritas (alta-fidelidade). Isto diminui o número de iterações que usam a alta-fidelidade tornando o algoritmo computacionalmente mais barato. Tesfahunegn, Koziel, Gramanzini, Hosder, Han, e Leifsson [10] fizeram um estudo comparativo de métodos de otimização direta com SBO e concluíram que os modelos têm performances parecidas para os problemas de minimização do arrasto para dois tipos de aerofólio encontrados em aviões, o NACA0012 e o RAE2822. As figuras abaixo, representam as comparações entre o SBO e outros métodos de otimização direta.

As formas ótimas da parte superior do aerofólio NACA0012 e suas respectivas distribuições de pressão são encontradas nas figuras 1.4 e 1.5, já as formas ótimas do RAE2822 e suas respectivas distribuições de pressão são representadas pelas figuras 1.6 e 1.7.

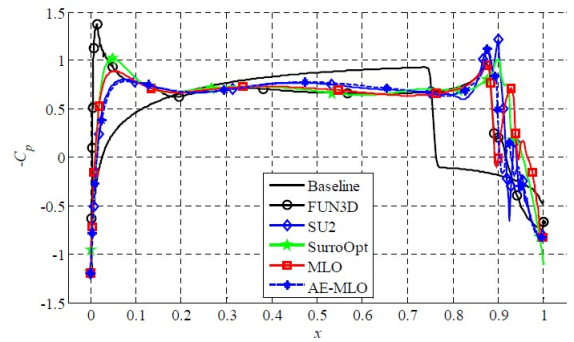
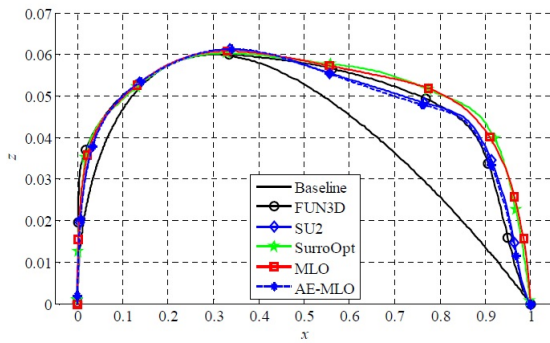


Figura 1.4: Forma baseline e formas otimizadas para o aerofólio NACA0012

Figura 1.5: Distribuição do coeficiente de pressão para as formas otimizadas - NACA0012

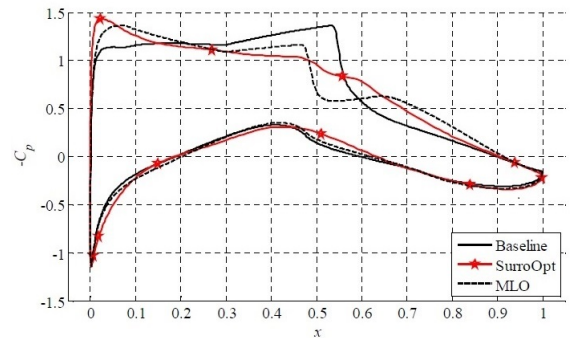
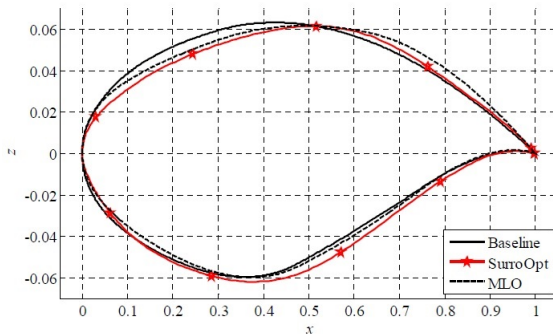


Figura 1.6: Forma baseline e formas otimizadas para o aerofólio RAE2822

Figura 1.7: Distribuição do coeficiente de pressão para as formas otimizadas - RAE2822

Visto isso, a ideia deste trabalho é construir um algoritmo de otimização aplicando a mistura de metamodelos aos problemas de aerodinâmica com intuito de encontrar a solução ótima global sem a necessidade de calcular a função computacionalmente cara várias vezes.

Capítulo 2

Otimização Através da Mistura de Metamodelos

2.1 Definição do Problema Matemático

Neste trabalho, os problemas de otimização são formulados como problemas de minimização (os problemas de maximização podem ser reformulados com a minimização da forma negativa da função objetivo). Em geral, os problemas de otimização global são representados na seguinte forma:

$$\text{minimize } f(x) \tag{2.1a}$$

$$\text{s.a. } c_j(x) \leq 0, j = 1, 2, \dots, m \tag{2.1b}$$

$$-\infty < x_i^l \leq x_i \leq x_i^u < \infty, i = 1, 2, \dots, k \tag{2.1c}$$

$$x \in \Omega, \tag{2.1d}$$

onde $f(x) : \mathbb{R}^k \rightarrow \mathbb{R}$ representa a função objetivo cujo valor é calculado através de um modelo de simulação computacionalmente caro, $x \in \mathbb{R}^k$ (x poderia pertencer a outro conjunto, porém neste trabalho serão tratados apenas os casos em que x pertence ao conjunto dos números reais), k representa a ordem do problema, $c_j(x)$ são as funções que representam as restrições e x_i^l e x_i^u representam os limites inferiores e superiores, respectivamente. Se não existem restrições $c_j(x)$, o problema é definido como problema de otimização global irrestrito. Caso existam restrições $c_j(x)$, $j = 1, 2, \dots, m$, presume-se que estas funções são calculadas na mesma simulação da função objetivo computacionalmente caro, ou seja, a simulação retorna um valor da função objetivo e um valor da função restrição. Desta forma, é assumido que o conjunto viável não é vazio.

A função objetivo $f(x)$ tem muitas características que podem dificultar a busca pelo ótimo global. Como mencionado anteriormente, a avaliação da função objetivo requer a execução do modelo de simulação computacionalmente caro, o que pode levar alguns

minutos, horas ou até dias. Além disto, por ser do tipo caixa preta, a função objetivo pode não ter forma algébrica definida e com isto, modelos de otimização que se utilizam da derivada e do Hessiano desta função não são adequados. Caso haja função algébrica definida, o cálculo de um problema k -dimensional pode custar, para o cálculo do gradiente e do Hessiano, k -vezes e k^2 -vezes respectivamente o custo de $f(x)$. A função objetivo também pode ser mal comportada e ter muitos mínimos locais e globais, assim métodos que são capazes de pesquisar localmente e globalmente têm sido desenvolvidos para encontrar soluções precisas do problema definido em (2.1a)-(2.1b) dentro de alguns cálculos da função objetivo, sem a necessidade de cálculo de derivadas.

2.2 Algoritmos de Metamodelos

Na busca de boas aproximações para os modelos mencionados na seção (1.1), metamodelos são desenvolvidos para substituir os modelos computacionalmente caros:

$$f(x) = s(x) + \epsilon(x), \quad (2.2)$$

onde $f(x)$ denota as saídas da função computacionalmente caro, $s(x)$ denota a predição do metamodelo no ponto x e $\epsilon(x)$ é a diferença entre os dois.

A ideia é utilizar o metamodelo $s(x)$ no lugar do modelo computacionalmente caro durante o processo de otimização, pois $s(x)$ tem um custo computacional muito menor que $f(x)$. O metamodelo $s(x)$ é usado para prever os valores da função objetivo dentro do domínio e esta informação é então explorada para encontrar candidatos a ponto ótimo da função caro. A superfície de resposta será a base para a construção dos metamodelos, contem toda informação necessária para gerá-los e será atualizada a cada iteração com novos candidatos a mínimo.

Em geral, um algoritmo de metamodelos consiste nos seguintes passos:

Algoritmo 1 *Algoritmo geral de otimização de metamodelos:*

1. *Crie uma amostra inicial e calcule a função objetivo, computacionalmente caro, nestes pontos.*
2. *Use os resultados do passo anterior para encontrar os parâmetros da superfície de resposta.*
3. *Use as informações da superfície de resposta para gerar um metamodelo.*
4. *A partir do metamodelo, encontre o novo candidato a mínimo.*
5. *Dado novo candidato, atualize a superfície resposta.*

6. Volte para o passo 3 até que um determinado critério de parada seja satisfeito.

A técnica de Amostragem por Hipercubo Latino [21] pode ser usada para encontrar uma amostra inicial, como descrito no primeiro passo. No segundo passo, a superfície de resposta é criada com os dados amostrados e seus respectivos valores avaliados na função computacionalmente cara. O terceiro passo, encontra-se o metamodelo que melhor se aproxima da função objetivo, Kriging e Funções de Base Radial podem ser citados como metamodelos interpoladores e Regressão Polinomial e a Regressão Adaptativa Multivariada por Splines como metamodelos não interpoladores. Cada um destes metamodelos serão detalhados nas seções seguintes.

No quarto passo, um novo candidato deverá ser encontrado a partir da busca local ou global do mínimo. A busca local se deve à exploração dos pontos vizinhos a fim de encontrar melhorias do valor da função objetivo. Por outro lado, a busca global se deve à fuga dos ótimos locais e encontrar novas regiões promissoras do domínio onde o ótimo global deve estar. Não é suficiente utilizar um algoritmo que só procura a nível global e não localmente, uma vez que uma região promissora foi encontrada, é de extrema importância fazer uma busca local para encontrar melhorias da função objetivo.

2.2.1 Modelos de Regressão Polinomial

Modelos de regressão polinomial são não interpoladores e largamente usados na engenharia[11]. Um modelo de regressão linear é em geral definido como:

$$s_p = \beta_0 + \sum_{i=1}^k \beta_i x_i \quad (2.3)$$

onde x_i é a i -ésima componente de x . A variável $\hat{\beta}$ é o estimador de mínimos quadrados dos parâmetros $\beta = (\beta_0, \beta_1, \dots, \beta_k)^T$ e é dado por:

$$\hat{\beta} = (X^T X)^{-1} X^T Y \quad (2.4)$$

onde $Y = (y_1, y_2, \dots, y_n)^T$ é o vetor de valores da função objetivo, e

$$X = \begin{bmatrix} 1 & X_1^T \\ 1 & X_2^T \\ \vdots & \vdots \\ 1 & X_n^T \end{bmatrix} \quad (2.5)$$

é a matrix de entradas aumentada com uma coluna a mais para o intercepto β_0 .

O modelo de regressão pode ser estendido para ordens superiores. Um modelo quadratico pode ser definido da seguinte forma:

$$s_p(\mathbf{x}) = \beta_0 + \sum_{i=1}^k \beta_i x_i + \sum_{i=1}^k \beta_{ii} x_i^2 + \sum_{i=1}^{k-1} \sum_{j=i+1}^k \beta_{ij} x_i x_j. \quad (2.6)$$

A forma quadrática do modelo pode capturar curvaturas e interações entre variáveis do modelo verdadeiro melhor que a forma linear.

Existem modelos de ordens superiores que não serão detalhados neste trabalho.

2.2.2 Regressão Adaptativa Multivariada por Splines

Regressão adaptativa multivariada por splines (MARS) [12] são modelos não interpoladores e não paramétricos que são capazes de modelar não linearidades e interações entre variáveis automaticamente. O modelo geral MARS é definido por:

$$s_m(\mathbf{x}) = \sum_{l=1}^L a_l B_l(\mathbf{x}). \quad (2.7)$$

onde a_l são coeficientes constantes e B_l são funções bases que podem ser constantes, rígidas, ou produto de duas ou mais funções rígidas.

Pares de funções rígidas tem a seguinte forma:

$$\max\{0, \mathbf{c} - \mathbf{x}\} \text{ e } \max\{0, \mathbf{x} - \mathbf{c}\}, \quad (2.8)$$

onde \mathbf{c} denota o então chamado nó.

O modelo MARS é construído em dois estágios:

- Passo para frente - pares de funções base são adicionadas ao modelo de forma que a soma do erro residual quadrático seja reduzido o máximo possível.
- Passo para trás - o modelo é podado, a cada passo o termo menos eficaz é excluído de acordo com algum critério de validação cruzada.

2.2.3 Função Base Radial

Os modelos interpoladores chamados de função base radial (RBF) [13,14] usados neste trabalho tem como grande vantagem, a boa modelagem de curvaturas e são definidos como:

$$s_b(\mathbf{x}) = \sum_{l=1}^n \lambda_l \phi(\|\mathbf{x} - \mathbf{x}_l\|_2) + p(x), \quad (2.9)$$

onde $\|\cdot\|_2$ denota a norma Euclidiana, $\lambda_1, \lambda_2, \dots, \lambda_n \in \mathbb{R}$ são parâmetros à serem determinados e $\phi : \mathbb{R}_+ \mapsto \mathbb{R}$ é a função base radial. Geralmente, as funções ϕ do tipo presente na Tabela 1.1.

Tabela 2.1: Modelos RBF geralmente usados, $\rho > 0$.

Nome	$\phi(d)$
Cubica	d^3
Spline "chapa" fina	$d^2 \log d$
Gaussiana	$\exp(-\frac{d^2}{\rho^2})$
Multiquadrática	$\sqrt{d^2 + \rho^2}$
Multiquadrática Inversa	$(d^2 + \rho^2)^{-\frac{1}{2}}$

$p \in \mathcal{P}_{d-1}^n$, onde \mathcal{P}_{d-1}^n é o espaço dos polinômios em k variáveis e grau no máximo $d - 1$. No caso cúbico, por exemplo, $p(x) = b^T x + a$ com $b \in \mathbb{R}^k$ e $a \in \mathbb{R}$. Os parâmetros desconhecidos $\lambda_1, \lambda_2, \dots, \lambda_n, b$ e a são obtidos resolvendo o sistema

$$\begin{bmatrix} \Phi & P \\ P^T & 0 \end{bmatrix} \begin{bmatrix} \lambda \\ \gamma \end{bmatrix} = \begin{bmatrix} y \\ 0 \end{bmatrix}, \quad (2.10)$$

onde

$$P = \begin{bmatrix} X_1^T & 1 \\ X_2^T & 1 \\ \vdots & \vdots \\ X_n^T & 1 \end{bmatrix}, \lambda = \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_n \end{bmatrix}, \gamma = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_k \\ a \end{bmatrix}, \quad (2.11)$$

e $\Phi_{lv} = \phi(\|x_l - x_v\|_2)$, $l, v = 1, \dots, n$, e 0 é uma matriz com todas as entradas iguais a zero e dimensão apropriada. Se $\text{posto}(P) = k + 1$, então a matriz da equação xxxx é não singular e o sistema xxx tem uma única solução. Portanto, é possível encontrar uma única interpolação função base radial da verdadeira função f .

2.2.4 Kriging

Kriging é um metamodelo interpolador e incorpora uma componente estocástica que pode ser explorada por um algoritmo de otimização global [4], mas foi introduzido por Matheron [15]. Este modelo consiste de duas componentes, a primeira é um simples modelo que captura a tendência nos dados e a segunda mede a divergência entre o modelo simples e a função verdadeira. Defina uma matriz de amostra $S = (x_1, \dots, x_n)^T$. Os dados são deslocados e dimensionado de tal modo que a média sobre cada coluna em S e a média dos valores da função objectivo em y são iguais a zero, e que as variações iguais a 1.

Um modelo s_k é construído com um modelo de regressão \mathcal{F} e uma função aleatória z afim de gerar uma saída y a partir do vetor de entradas $\tilde{x} \in \Omega$, onde Ω denota as variáveis de domínio. O modelo pode ser escrito como

$$s_k(\tilde{\mathbf{x}}) = \mathcal{F}(\phi, \tilde{\mathbf{x}}) + z(\tilde{\mathbf{x}}), \quad (2.12)$$

onde o modelo de regressão \mathcal{F} é uma combinação linear de q funções $\varphi_j : \mathbb{R}_+ \mapsto \mathbb{R}$, $j = 1, 2, \dots, q$.

$$\mathcal{F}(\phi, \tilde{\mathbf{x}}) = \psi_1\varphi_1(\tilde{\mathbf{x}}) + \psi_2\varphi_2(\tilde{\mathbf{x}}) + \dots + \psi_q\varphi_q(\tilde{\mathbf{x}}), \quad (2.13)$$

e pode ser escrito na seguinte notação matricial

$$\mathcal{F}(\phi, \tilde{\mathbf{x}}) = \varphi(\tilde{\mathbf{x}})^T \psi, \quad (2.14)$$

onde $\varphi_1(\tilde{\mathbf{x}}) = (\varphi_1(\tilde{\mathbf{x}}), \dots, \varphi_q(\tilde{\mathbf{x}}))^T$ é o vetor de funções, e $\psi = (\psi_1, \dots, \psi_q)^T$ é o vetor parâmetro da regressão. O processo aleatório z é assumido com média zero e covariância

$$\mathbb{V}[z(\mathbf{w}), z(\tilde{\mathbf{x}})] = \sigma^2 \mathcal{R}(\theta, \mathbf{w}, \tilde{\mathbf{x}}) \quad (2.15)$$

entre $z(\mathbf{w})$ e $z(\tilde{\mathbf{x}})$. Aqui σ^2 é a variância e $\mathcal{R}(\theta, \mathbf{w}, \tilde{\mathbf{x}})$ é a correlação dependendo do parâmetro θ que tem que ser otimizado.

2.3 Algoritmo de Otimização Global usando Meta-modelos Simples

Como já mencionado anteriormente, uma função do tipo caixa preta exige um custo computacional muito alto e um algoritmo de otimização levaria muito tempo para encontrar a solução ótima, pois seria preciso muitas avaliações da função objetivo. Com o intuito de diminuir o custo computacional do algoritmo, será introduzido nesta seção, um método que aproxima a função computacionalmente cara por um metamodelo, computacionalmente mais barato. Assim, o processo de busca de novos candidatos a mínimo usará o metamodelo que é mais barato que a função objetivo.

2.3.1 Algoritmo

O algoritmo começa com uma busca global separando o domínio em n^k regiões, onde n representa o número de divisões feitas em cada variável do domínio e k representa o número de dimensão do problema. São feitas buscas superficiais em cada região e posteriormente uma procura local em regiões cuja busca global apontou candidatos mais promissores. Em Müller [1], o algoritmo começa com uma busca global em regiões não exploradas e em seguida são feitas buscas locais usando o mínimo do metamodelo ou fazendo buscas mais finas em regiões densamente exploradas, espera-se que o mínimo esteja nestas regiões.

Em sua busca global, o algoritmo começa gerando amostras através da técnica de Amostragem por Hipercubo Latino, que maximiza a mínima distância entre os pontos amostrados, e em geral, $k + 1$ amostras são requeridas para construção do metamodelo. Em seguida, um novo candidato a mínimo é encontrado através do metamodelo. Então avalia-se a função, computacionalmente cara, neste ponto e a superfície de resposta é atualizada com o novo candidato.

Na busca local, o algoritmo explora regiões promissoras encontradas na busca global, a partir da superfície de resposta que contem candidatos promissores a mínimo, se calcula o metamodelo. Em seguida um novo candidato é encontrado e seu respectivo valor da função computacionalmente cara é encontrado. Atualiza-se a superfície de resposta e repete o processo até que uma determinada condição seja satisfeita.

Um algoritmo que usa apenas um metamodelo simples para substituir a função computacionalmente cara consiste dos seguintes passos:

Algoritmo 2 *Algoritmo de otimização usando metamodelos simples:*

1. *Separe o domínio em n^k regiões.*
2. *Em cada região, faça os seguintes passos*
 - 2.1 *Crie uma amostra inicial e avalie a função, computacionalmente cara, nestes pontos.*
 - 2.2 *construa a superfície de resposta a partir da amostra e dos respectivos pontos avaliados na função cara.*
 - 2.3 *Construa o metamodelo a partir da superfície de resposta.*
 - 2.4 *Encontre o novo candidato através do metamodelo.*
 - 2.5 *Atualize a superfície de resposta com o novo candidato encontrado no passo anterior.*
 - 2.6 *Volte ao passo 2.3 até que uma condição de parada seja satisfeita.*
3. *Determine regiões com os melhores candidatos, onde o algoritmo fará uma busca local.*
4. *Em cada região determinada no passo anterior, faça os seguintes passos*
 - 4.1 *Assuma como superfície de resposta, a última adquirida no processo de busca global.*
 - 4.2 *Construa o metamodelo através das superfície de resposta obtida.*
 - 4.3 *Encontre o novo candidato através do metamodelo.*
 - 4.4 *Atualize a superfície de resposta com o novo candidato.*

4.5 Volte ao passo 4.2 até que uma condição de parada seja satisfeita.

Os passos descritos acima representam a forma mais simples de um algoritmo de otimização usando metamodelos, em cada iteração existe apenas um tipo de modelo aproximando a função computacionalmente cara. Como cada modelo tem características próprias e podem ter comportamentos diferentes em cada região avaliada, uma proposta de mistura de metamodelos será implementada nas próximas seções. Antes disto, testou-se o algoritmo acima para algumas funções, cujas soluções são conhecidas.

2.3.2 Resultados numéricos

Esta seção resume os resultados aplicados para cinco problemas de otimização global de Dixon e Szegö [16] encontrados na tabela 2.2, embora os problemas tenham descrição analítica e suas soluções ótimas sejam conhecidas, eles serão tratados como modelos de caixa-preta para testar a veracidade do algoritmo 2.

Tabela 2.2: Problemas para testes numéricos
ID - identificação do problema, k - dimensão do problema,
LM - mínimo local, GM - mínimo global

ID	Nome	k	Qtd mínimo Local/Global	Mínimo global
B	Branin	2	O LM/ 3 GM	0,40
G	Goldstein - Price	2	muitos LM/ 1 GM	3,00
H3	Hartmann	3	4 LM/ 1 GM	-3,86
H6	Hartmann	6	4 LM/ 1 GM	-3,32
S10	Shekel 10	4	10 LM/ 1 GM	-10,54

Neste trabalho serão feitos testes para diferentes tamanhos de amostra s , substituindo o valor $k + 1$ presente no algoritmo 2. Serão feitos testes para diferentes tamanhos de amostra para verificar o quão rápido o algoritmo converge a medida que se aumenta o tamanho da amostra. Em Müller [1], todos os testes foram realizados para amostras de tamanho $k + 2$.

Em cada problema, todos os metamodelos foram testados separadamente para três tamanhos diferentes de amostra $s = 10, 30e50$, realizando 10 ensaios em cada teste. O algoritmo começa com uma busca global, dividindo o domínio em 2^k regiões de mesmo tamanho e realizando 15 iterações em cada região. A região que contém o ponto de menor valor para função objetivo é tomada como a região mais promissora para encontrar o mínimo global e é então explorada na busca local, através da realização de 50 iterações nesta região. Ao fim das 50 iterações, o mínimo global será o menor valor encontrado para função objetivo nesta região.

Para cada problema, serão mostrados boxplots para mostrar os resultados dos 10 ensaios de cada metamodelo e seus respectivos tamanhos de amostra. Além disto, será adicionado uma tabela mostrando o passo médio de convergência e uma outra tabela com a média dos erro entre o mínimo global encontrado com o metamodelo e o verdadeiro mínimo do problema.

Teste do problema B: Função Branin

Brannin é um problema bi-dimensional com $x_1 \in [-5, 10]$, $x_2 \in [0, 15]$ e função definida por:

$$f(x_1, x_2) = \left(x_2 - \frac{5.1x_2}{4\pi^2} + \frac{5x_1}{\pi} - 6\right) + 10\left(1 - \frac{1}{8\pi}\cos(x_1)\right) + 10 \quad (2.16)$$

O problema contém três mínimos globais com valor ótimo $f^* = 0.3979$, localizados em $(x_1^*, x_2^*) = (-\pi, 12.275)$, $(x_1^*, x_2^*) = (\pi, 2.275)$ e $(x_1^*, x_2^*) = (3\pi, 2.475)$ e não existe ótimos locais.

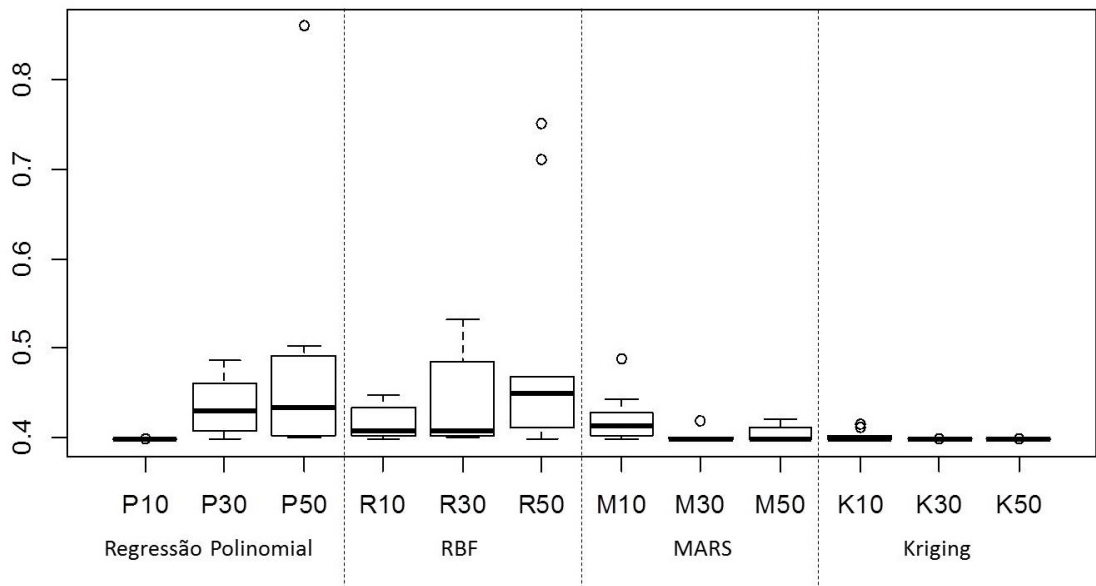


Figura 2.1: Comparação entre metamodelos - Função Branin

Os resultados obtidos são apresentados no boxplot da figura 2.1. Pode-se notar que para este problema, os metamodelos interpoladores obtiveram os melhores resultados, sendo o Kriging, o metamodelo que encontrou os melhores resultados. A performance dos algoritmos que usaram a regressão polinomial e o RBF não obtiveram resultados muito

bons quando comparados com os outros metamodelos. Além disso, o algoritmo usando a regressão polinomial não convergiu para o mínimo global, com erro inferior a 0.1, em uma oportunidade e o RBF deixou de convergir em duas.

A tabela 2.3 apresenta a média dos erros obtidos e o que se pode ver é que o modelo Kriging obteve os menores erros para os testes realizados com as amostras de tamanhos $s = 30$ e 50 . Já a regressão polinomial apresentou erros baixos para os testes realizados com amostra de tamanho $s = 10$.

Tabela 2.3: Média dos Erros - Função Brannin

	Regressão Polinomial	RBF	MARS	Kriging
10	0.0006	0.0193	0.0232	0.0041
30	0.0368	0.0330	0.0029	0.0001
50	0.0323	0.0350	0.0058	0.0001

Os passos convergência apresentados na tabela 2.4 foram obtidos na etapa da busca local, ou seja, foi computado a partir do momento em que já se tem uma região promissora para o mínimo global. O que se nota é que todos os metamodelos convergiram no primeiro passo quando foram usados amostras de tamanho $s = 50$ e para as menores amostras, Regressão Polinomial, RBF e Kringing convergiram para o mínimo nos primeiros passos em quase todos os ensaios.

Tabela 2.4: Convergência - Função Brannin

	Regressão Polinomial	RBF	MARS	Kriging
10	1	2	4	1
30	1	1	2	1
50	1	1	1	1

Teste do problema G: Função Goldstein - Price

Goldstein-Price é um problema bi-dimensional com $x_1, x_2 \in [-2, 2]$ e função definida por:

$$f(x_1, x_2) = \frac{[1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)] \times [30 + (2x_1 - 3x_2)^2(18 - 31x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)]}{(2.17)}$$

Goldstein-Price tem um único mínimo global com valor ótimo $f^* = 3.0$, localizado em $(x_1^*, x_2^*) = (0, -1)$ e muitos mínimos locais espalhados pelo domínio.

Os resultados obtidos são apresentados no boxplot da figura 2.2. Pode-se notar que para este problema, Kriging obteve os melhores resultados para todos os tamanhos de amostra, RBF e MARS tiveram bons resultados quando foram usadas amostras de tamanhos $s = 30$ e 50 , já a regressão polinomial convergiu para o mínimo global em poucos ensaios independente do tamanho da amostra. Com exceção do Kriging, todos os outros metamodelos tiveram dificuldades em convergir para o mínimo nas amostras de tamanho $s = 10$.

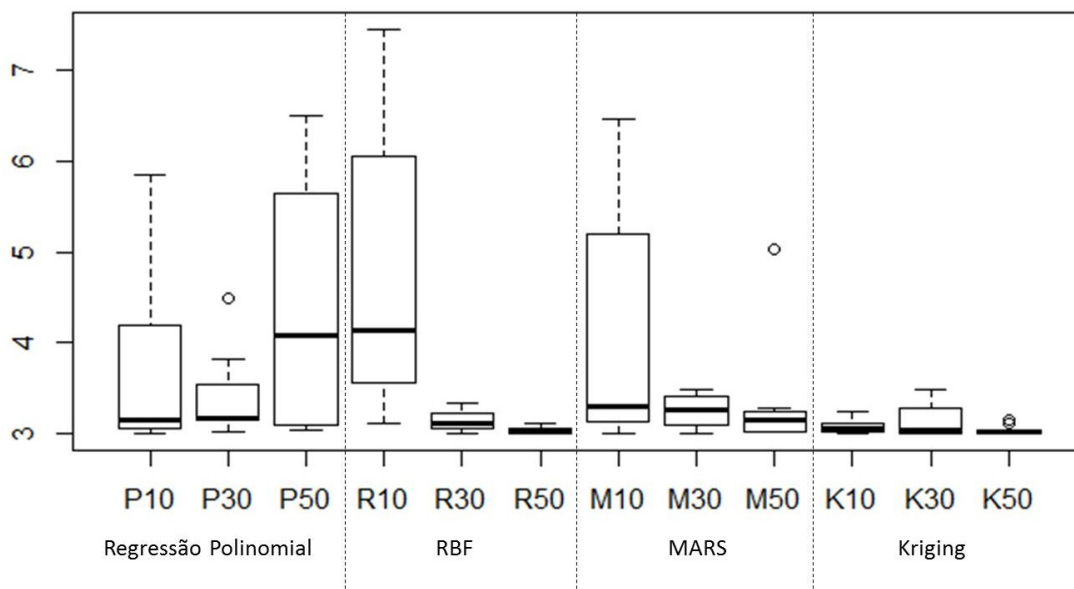


Figura 2.2: Comparação entre metamodelos - Função Goldstein Price

A tabela 2.5 apresenta a média dos erros obtidos e o que se pode ver é que o modelo Kriging obteve os menores erros para os testes realizados com amostra de tamanho $s = 10$. Para os testes realizados com amostras de tamanhos $s = 30$ e 50 o RBF se saiu ligeiramente melhor, seguido pelo Kriging. O metamodelo de Regressão Polinomial obteve pior performance que os outros metamodelos.

Os passos convergência apresentados na tabela 2.6 foram obtidos na etapa da busca local, ou seja, foi computado a partir do momento em que já se tem uma região promissora para o mínimo global. O metamodelo Kriging convergiu mais rápido que os outros metamodelos para todo tamanho de amostra, MARS e RBF tiveram performances parecidas

Tabela 2.5: Média dos Erros - Goldstein Price

	Regressão Polinomial	RBF	MARS	Kriging
10	0.6418	1.7487	0.9499	0.0816
30	0.2576	0.1299	0.2611	0.1436
50	2.0267	0.0325	0.3219	0.0379

com leve vantagem para MARS e a Regressão Polinomial teve a pior convergência entre todos os metamodelos.

Tabela 2.6: Convergência - Goldstein Price

	Regressão Polinomial	RBF	MARS	Kriging
10	34	21	18	6
30	20	15	11	2
50	15	8	8	2

Teste dos problemas H3: Funções Hartmann 3

A classe das funções Hartmann são definidas por:

$$f(\mathbf{x}) = - \sum_{\iota=1}^m c_{\iota} \exp \left\{ - \sum_{i=1}^k a_{\iota i} (x_i - p_{\iota i})^2 \right\} \quad (2.18)$$

onde $\mathbf{x} = (x_1, x_2, x_3, \dots, x_k)$ e $x_i \in [0, 1], \forall i$. O valor de m é definido como $m = 4$ e os parâmetros para o problema tri-dimensional são dados na tabela 2.7.

Tabela 2.7: Hartmann 3

ι	$a_{\iota i}$			c_{ι}	$p_{\iota i}$		
1	3.0	10	30	1.0	0.3689	0.1170	0.2673
2	0.1	10	35	1.2	0.4699	0.4387	0.7470
3	3.0	10	30	3.0	0.1091	0.8732	0.5547
4	0.1	10	35	3.2	0.0382	0.5743	0.8828

O problema tri-dimensional de Hartmann tem um valor de ótimo global $f^* = -3.8628$, localizado em $(x_1^*, x_2^*, x_3^*) = (0.1146, 0.5556, 0.8525)$ e quatro mínimos locais. Os valores da função são $-c_{\iota}, \iota = 1, \dots, 4$.

Os resultados obtidos são apresentados no boxplot da Figura 2.3. Pode-se notar que para este problema, Kriging obteve os melhores resultados para todos os tamanhos de amostra, a Regressão Polinomial obteve bons resultados para as amostras de tamanhos $s = 10$ e 30 . O metamodelo RBF de amostra tamanho $s = 50$ obteve o pior resultado entre todos.

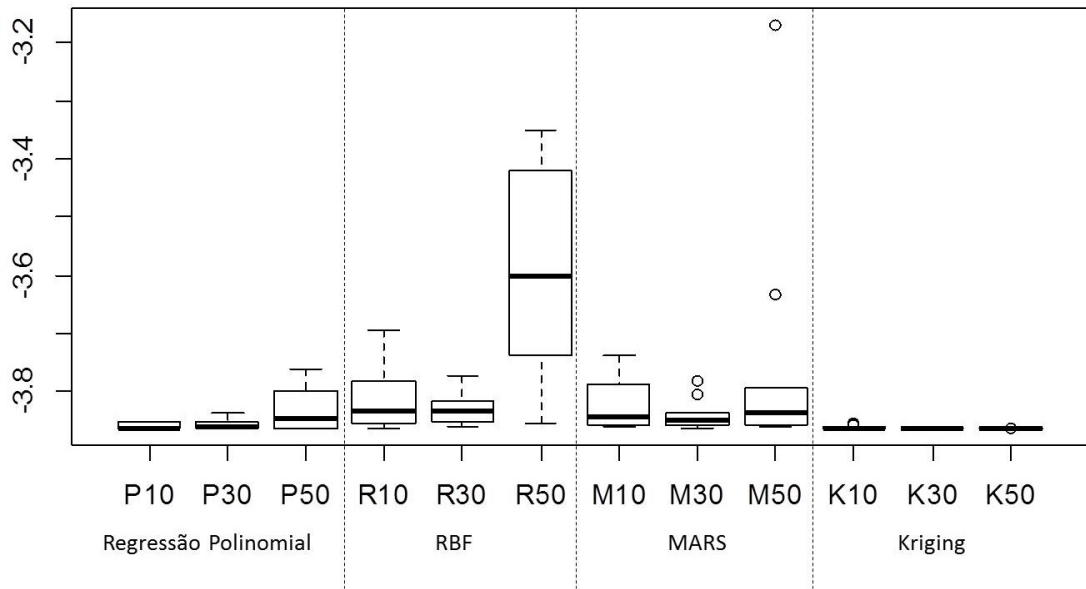


Figura 2.3: Comparação entre metamodelos - Função Hartmann 3

A tabela 2.8 apresenta a média dos erros obtidos e o que se pode ver é que o modelo Kriging obteve os menores erros para os testes realizados com todos os tamanhos de amostra. Apenas uma amostra do metamodelo MARS não convergiu com erro menor que 0.5, porém a pior média dos erros foi do metamodelo RBF usando amostra de tamanho $s = 50$.

Tabela 2.8: Média dos Erros - Hartmann 3

	Regressão Polinomial	RBF	MARS	Kriging
10	0.0033	0.0486	0.0426	0.0019
30	0.0078	0.0326	0.0241	0.0002
50	0.0315	0.2688	0.1121	0.0000

Os passos convergência apresentados na tabela 2.9 foram obtidos na etapa da busca local, ou seja, foi computado a partir do momento em que já se tem uma região promissora para o mínimo global. O metamodelo Kriging convergiu mais rápido que os outros metamodelos para as amostras de tamanho $s = 10$ e a Regressão Polinomial convergiu mais rápido para os algoritmos de tamanho $s = 30$ e 50 .

Tabela 2.9: Convergência - Hartmann 3

	Regressão Polinomial	RBF	MARS	Kriging
10	29	30	28	20
30	2	12	8	7
50	2	6	6	4

Teste dos problemas H6: Funções Hartmann 6

A classe das funções Hartmann foram definidos na equação 2.18, onde $\mathbf{x} = (x_1, x_2, x_3, \dots, x_k)$ e $x_i \in [0, 1], \forall i$. O valor de m é definido como $m = 4$ e os parâmetros para o problema de dimensão seis são dados na tabela 2.10.

Tabela 2.10: Hartmann 6

ι	$a_{\iota i}$						c_{ι}	$p_{\iota i}$					
1	10.0	3.0	17.0	3.5	1.7	8.0	1.0	0.1312	0.1696	0.5569	0.0124	0.8283	0.5886
2	0.05	10.0	17.0	0.1	8.0	14.0	1.2	0.2329	0.4135	0.8307	0.3736	0.1004	0.9991
3	3.0	3.5	1.7	10.0	17.0	8.0	3.0	0.2348	0.1451	0.3522	0.2883	0.3047	0.665
4	17.0	8.0	0.05	10.0	0.1	14.0	3.2	0.4047	0.8828	0.8732	0.5743	0.1091	0.0381

O problema de dimensão seis alcança o mínimo global em $(x_1^*, x_2^*, x_3^*, x_4^*, x_5^*, x_6^*) = (0.2017, 0.1500, 0.4769, 0.2753, 0.3117, 0.6573)$ com valor $f^* = -3.3224$ e possui quatro mínimos locais. Os valores da função $-c_{\iota}, \iota = 1, \dots, 4$.

Os resultados obtidos são apresentados no boxplot da Figura 2.4. Todos os metamodelos tiveram melhores resultados para os ensaios feitos com as amostras de tamanho $s = 30$ e 50 , já para os ensaios com amostra de tamanho $s = 10$ tiveram resultados inferiores e para o metamodelo de Regressão Polinomial não se obteve convergência menor que 0.3.

A tabela 2.11 apresenta a média dos erros obtidos e o que se pode ver é que todos os algoritmos tiveram baixos erros para os testes com amostras de tamanhos $s = 30$ e 50 ,

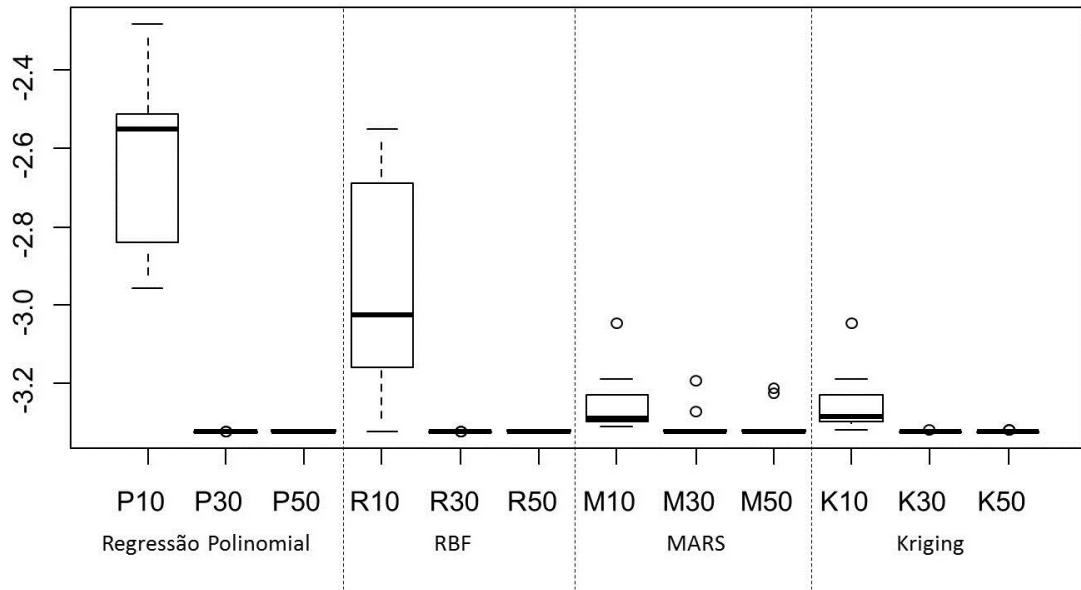


Figura 2.4: Comparação entre metamodelos

porém os ensaios realizados com $s = 10$ tiveram resultados piores, sendo a Regressão polinomial com a pior performance.

Tabela 2.11: Média dos Erros - Função Hartmann 6

	Regressão Polinomial	RBF	MARS	Kriging
10	0.7087	0.3705	0.0695	0.0713
30	0.0000	0.0000	0.0185	0.0009
50	0.0002	0.0020	0.0224	0.0013

Os passos convergência apresentados na tabela 2.12 foram obtidos na etapa da busca local, ou seja, foi computado a partir do momento em que já se tem uma região promissora para o mínimo global. Todos precisaram de mais passos para convergir nos ensaios realizados com amostra de tamanho $s = 10$, sendo que a Regressão Polinomial não convergiu nenhuma vez com erro menor que 0.3. Já para as amostras de tamanhos $s = 30$ e 50 , o metamodelo de Regressão Polinomial obteve o melhor desempenho.

Tabela 2.12: Convergência - Função Hartmann 6

	Regressão Polinomial	RBF	MARS	Kriging
10	0	37	32	31
30	2	14	9	9
50	2	7	7	6

Teste do problema S10: Função Shekel

A definição geral do problema de Shekel é:

$$f(\mathbf{x}) = - \sum_{i=1}^m \frac{1}{c_i + \sum_{j=1}^k (x_j - a_{ij})^2} \quad (2.19)$$

e as restrições das variáveis são dadas por $\mathbf{x} \in [0, 10]$. Os parâmetros são:

$$A = \begin{bmatrix} 4 & 1 & 8 & 6 & 3 & 2 & 5 & 8 & 6 & 7 \\ 4 & 1 & 8 & 6 & 7 & 9 & 5 & 1 & 2 & 3.6 \\ 4 & 1 & 8 & 6 & 3 & 2 & 3 & 8 & 6 & 7 \\ 4 & 1 & 8 & 6 & 7 & 9 & 3 & 1 & 2 & 3.6 \end{bmatrix} \quad c = \frac{1}{10}(1, 2, 2, 4, 4, 6, 3, 7, 5, 5)^T.$$

Os resultados obtidos são apresentados no boxplot da Figura 2.5. Os metamodelos Kriging e MARS tiveram os melhores resultados, apesar de que o Kriging não convergiu, com erro menor que 0.5 para dois ensaios com amostra de tamanho $s = 10$ e MARS deixou de convergir em um ensaio. A Regressão Polinomial obteve os piores resultados e não convergiu para o mínimo em todos os ensaios que utilizaram amostra do tamanho $s = 10$.

A tabela 2.13 apresenta a média dos erros obtidos e o que se pode ver é que o MARS obteve menores erros, seguidos pelo Kriging. A Regressão Polinomial apresentou os piores erros de todos os metamodelos.

Tabela 2.13: Média dos Erros - Função Shekel

	Regressão Polinomial	RBF	MARS	Kriging
10	4.5769	3.0399	0.3258	1.5119
30	0.3058	0.2541	0.0426	0.0426
50	1.7759	0.2174	0.0023	0.0393

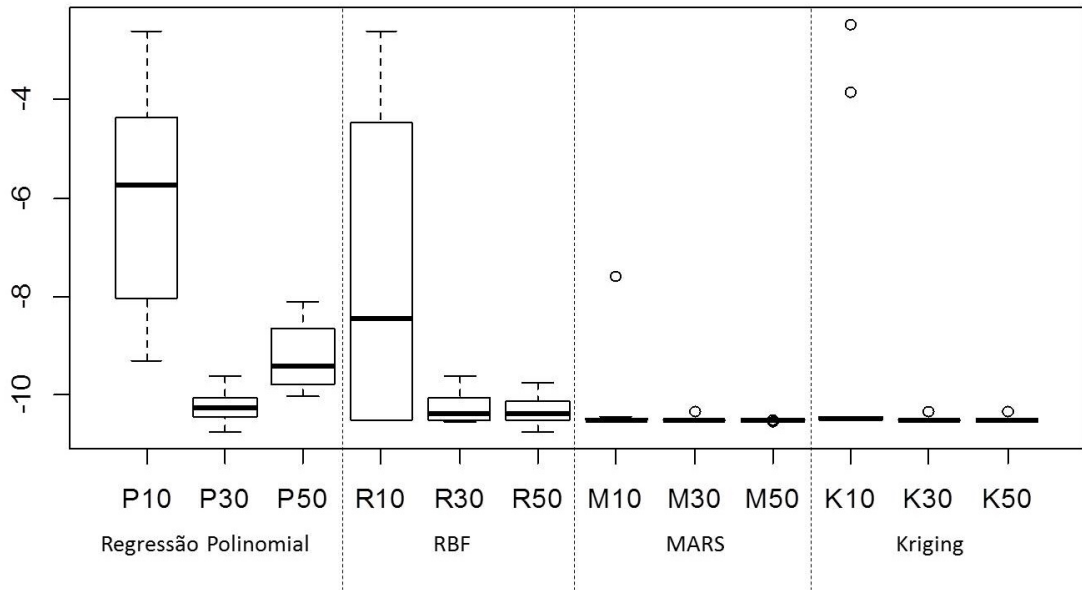


Figura 2.5: Comparação entre metamodelos

Os passos convergência apresentados na tabela 2.12 foram obtidos na etapa da busca local, ou seja, foi computado a partir do momento em que já se tem uma região promissora para o mínimo global. MARS e Kriging tiveram convergências similares para todos os tamanhos de amostra, já a Regressão Polinomial e RBF não convergiram em nenhum dos ensaios para amostras de tamanho $s = 10$.

Tabela 2.14: Convergência - Função Shekel

	Regressão Polinomial	RBF	MARS	Kriging
10	0	0	31	35
30	31	37	15	15
50	26	29	9	10

2.4 Algoritmo de Otimização Global Usando Mistura Metamodelos

Como visto em Goel, Haftka, Shyy e Queipo [17] e em Viana & Haftka [18], os metamodelos têm características próprias e não há um padrão que defina o melhor para todo

tipo de problema. Enquanto MARS são bons com não linearidades, a regressão polinomial de grau um seria o modelo mais indicado para problemas lineares. Para caixas pretas, em que nada se conhece a respeito da álgebra que envolve a função objetivo, seria interessante um modelo que se ajuste da melhor forma possível ao problema. Desta forma, a mistura de metamodelos SO-M (sigla em inglês Surrogate Optimization - Mixture) é um único modelo que tem como objetivo extrair a melhor performance dos metamodelos simples. O SO-M é definidos como:

$$s_{mix}(x) = \sum_{r \in \mathcal{M}} w_r s_r(x), \text{ onde } \sum_{r \in \mathcal{M}} w_r = 1, \quad (2.20)$$

onde $w_r \geq 0$ são os pesos dados em cada r -ésimo modelo simples, \mathcal{M} é o conjunto de todos os metamodelos e $s_r(x)$ é a predição do r -ésimo modelo no ponto x .

Goel, Haftka, Shyy e Queipo [17] sugeriram diferentes abordagens para determinar os pesos dos metamodelo. No entanto, apenas uma das abordagens permite separar a influência de boas e más características do modelo. Além disso, nesta abordagem, parâmetros devem ser ajustados, que é, em geral, uma tarefa difícil.

Viana & Haftka [18] também consideraram mistura de metamodelos e sugeriram a otimização de uma função auxiliar a fim de obter uma matriz de aproximação. Esta matriz é usada para determinar os pesos do modelo, mas pode levar a pesos negativos ou maiores que um e assim podem tornar os resultados imprecisos.

Müller [1] usou a Teoria Matemática da Evidência Dempster-Shafer(DST) [19, 20] para superar os problemas mencionados nos trabalhos acima. Esta teoria fornece meios de combinar informações de diferentes fontes a fim de construir um grau de crença que definirá o melhor modelo. Esta função de credibilidade é construída a partir das chamadas Atribuições Básicas de Probabilidade - BPA (sigla em inglês Basic Probability - Assignments) que contém informações sobre certas hipóteses (elementos focais) que serão combinadas. Três funções são geralmente associadas com BPAs, nomeadas como crença, plausibilidade e funções de probabilidade pignística (BetP).

2.4.1 Atribuições Básicas de Probabilidade e Teoria Matemática da Evidência

Em termos de metamodelos, as BPAs podem ser derivadas, por exemplo, de estatísticas características tais como coeficientes de correlação e outras medidas de erro. Assim, é possível que um metamodelo tenha características conflitantes, isto é, um metamodelo bom tem alto coeficiente de correlação e baixos erros e um metamodelo ruim tem baixo

coeficiente de correlação e altos erros. Estes conflitos devem ser levados em conta no cálculo da crença para escolha de um determinado metamodelo.

Coeficiente de Correlação - CC (sigla em inglês Correlation Coefficient), Raíz do Erro Quadrático Médio - RMSE (sigla em inglês Root Mean Squared Error), Erro Médio Absoluto - MAE (sigla em inglês Mean Absolute Error) e Desvio Médio Absoluto - MAD (sigla em inglês Median Absolute Deviation) foram escolhidos por Müller [1] e também foram usados neste trabalho como modelos característicos para o SO-M.

Seja e_i , o erro do período de previsão i , dado por

$$e_i = y_i - f_i \quad (2.21)$$

onde y_i é o valor real nesse período e f_i o valor previsto.

As estatísticas características são definidos como:

$$CC = \frac{\sum_{i=1}^n (f_i - \bar{f})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (f_i - \bar{f})^2} \sqrt{\sum_{i=1}^n (y_i - \bar{y})^2}} \quad (2.22)$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^n e_i^2}{n}} \quad (2.23)$$

$$MAD = \frac{1}{n} \sum_{i=1}^n |y_i - \bar{y}| \quad (2.24)$$

$$MAE = \frac{1}{n} \sum_{i=1}^n |f_i - y_i| \quad (2.25)$$

O Coeficiente de Correlação reflete o quão bem os metamodelos são capazes de capturar o comportamento da verdadeira função objetivo. Quanto mais próximo de um, indica que o metamodelo é capaz de prever os valores da função objetivo. Já as medidas de erro devem ser próximas de zero para indicar o quanto os metamodelos são precisos.

As BPAs são construídas a partir das estatísticas características, todas elas devem ser dimensionadas de modo que a soma sobre todos os metamodelos sejam iguais a um e as condições de não negatividade devem ser satisfeitas. Ou seja, cada estatística característica calculada nos metamodelos deve ser não negativa e a soma de todas elas deve ser um. O DST é aplicado para determinar as probabilidades pignistic para cada metamodelo e com

base nestes valores, pode ser decidido que um de todos os metamodelos é considerado o melhor. Ou no caso de modelos de mistura, define-se os pesos de cada metamodelo simples. CC, RMSE, MAE, e MAD também devem ser calculados para cada mistura a fim de determinar o melhor modelo de mistura. Os valores das estatísticas características são normalizadas para obter as BPAs para cada mistura de:

$$m_r^{CC} = \frac{CC_r}{\sum_{j \in \mathcal{M}} CC_j}, \quad m_r^{RMSE} = \frac{\frac{1}{RMSE_r}}{\sum_{j \in \mathcal{M}} \frac{1}{RMSE_j}},$$

$$m_r^{MAE} = \frac{\frac{1}{MAE_r}}{\sum_{j \in \mathcal{M}} \frac{1}{MAE_j}}, \quad m_r^{MAD} = \frac{\frac{1}{MAD_r}}{\sum_{j \in \mathcal{M}} \frac{1}{MAD_j}},$$

onde \mathcal{M} é o conjunto de todas combinações possíveis dos metamodelos e r é o índice do r -ésimo modelo na combinação. Quatro conjuntos de evidências são obtidas, isto é, os modelos contidos em cada mistura considerados a partir dos elementos focais, e os conjuntos são as evidências BPAs para cada modelo que irá por sua vez ser utilizados na teoria da decisão.

Segue abaixo, um exemplo que ilustra o cálculo dos pesos para mistura de metamodelos. Para facilitar a ilustração, P representa o modelo de regressão linear, R representa o modelo RBF, K representa o modelo Kriging e M representa o modelo MARS. Depois do Leave-One-Out e Cross-Validation, as BPAs são dados como seguinte:

$$m_P^{CC} = 0.29, \quad m_R^{CC} = 0.29, \quad m_K^{CC} = 0.42, \quad m_M^{CC} = 0,$$

$$m_P^{RMSE} = 0.11, \quad m_R^{RMSE} = 0.24, \quad m_K^{RMSE} = 0.25, \quad m_M^{RMSE} = 0.40,$$

$$m_P^{MAE} = 0.10, \quad m_R^{MAE} = 0.24, \quad m_K^{MAE} = 0.25, \quad m_M^{MAE} = 0.41,$$

$$m_P^{MAD} = 0.17, \quad m_R^{MAD} = 0.27, \quad m_K^{MAD} = 0.26, \quad m_M^{MAD} = 0.30,$$

Depois de aplicar o DST, as probabilidades pignísticas são dadas por

$$BetP(P) = 0.05, \quad BetP(R) = 0.40, \quad BetP(K) = 0.55, \quad BetP(M) = 0,$$

e com isto, se um metamodelo simples for usado no próximo passo, o Kriging é o escolhido por ter o maior valor da probabilidade pignística. Caso um modelo de mistura for requerido, as probabilidades pignísticas são usadas para calcular os pesos do modelo.

$$w_p = \frac{BetP(P)}{BetP(P)+BetP(R)} = \frac{1}{9}, \quad w_r = \frac{BetP(R)}{BetP(P)+BetP(R)} = \frac{8}{9},$$

Assumindo que a mistura seja composta apenas dos modelos P e R, os pesos são dados por:

onde w_p é o peso do modelo de regressão polinomial e w_r é o peso para o RBF, respectivamente. Assim, $w_p + w_r = 1$ e os pesos são computados no modelo de mistura de acordo com a equação 2.20:

$$s_{mix} = w_p s_p(\mathbf{x}) + w_r s_r(\mathbf{x}) = \frac{1}{9} s_p(\mathbf{x}) + \frac{8}{9} s_r(\mathbf{x}).$$

A partir deste resultado, conclui-se que o metamodelo s_{mix} seria composto por 11.1% do modelo s_p , representado pela regressão polinomial e 88.9% do modelo s_r , representado pelo modelo RBF.

2.4.2 Algoritmo

Assim como na seção 2.3.2, o algoritmo começa com uma busca global onde o domínio é separado em n^k regiões, onde n representa o número de divisões feitas em cada variável do domínio e k representa o número de dimensão do problema. São feitas buscas superficiais em cada região e posteriormente uma procura local em regiões cuja busca global apontou candidatos mais promissores.

Em cada região da busca global, o algoritmo começa gerando uma amostra através da técnica de Amostragem por Hipercubo Latino, que maximiza a mínima distância entre os pontos amostrados, e em geral, uma amostra de $k+2$ pontos é requerida para construção do metamodelo. Aqui é exigido que a amostra contenha um ponto a mais do que a amostra usada no algoritmo da seção 2.3.2, pois além dos $k + 1$ pontos usados para construir o metamodelo, um ponto adicional é usado para testar a qualidade do modelo. Depois da amostra ter sido gerada, a superfície de resposta é construída a partir dos pontos amostrados e avaliados com a função computacionalmente cara. Em seguida, as técnicas de Leave-One-Out e Cross-Validation são usadas para dar informações para o cálculo das estatísticas pignísticas de cada metamodelo. Veja o passo a passo mostrado a seguir.

Leave-One-Out e Cross-Validation

1. *Separe um ponto da superfície de resposta e calcule o metamodelo escolhido com o pontos restantes.*
2. *Avalie o ponto que foi separado usando o metamodelo calculado no passo anterior.*
3. *Guarde a avaliação do ponto avaliado no passo anterior.*

4. *Repita os passos anteriores para todos os pontos da amostra.*

Use as avaliações feitas com a função computacionalmente cara e os pontos avaliados em leave-one-out e cross-validation para calcular as estatísticas características CC, RMSE, MAD e MAE. Isto dará informações para encontrar o melhor metamodelo simples ou a melhor mistura de metamodelos através da Teoria Matemática da Evidência DST. Em seguida, um novo candidato a mínimo é encontrado através do metamodelo escolhido, assim a função computacionalmente cara é avaliada e os parâmetros da superfície de resposta serão atualizados.

Um algoritmo que usa a mistura de metamodelos para substituir a função computacionalmente cara, consiste dos seguintes passos:

Algoritmo 3 *Algoritmo de otimização usando mistura de metamodelos:*

1. *Separe o domínio em n^k regiões.*
2. *Em cada região, faça os seguintes passos*
 - 2.1 *Crie uma amostra e avalie cada ponto na função computacionalmente cara.*
 - 2.2 *Constua a superfície de resposta com a amostra gerada e os pontos avaliados na função computacionalmente cara.*
 - 2.3 *A partir da superfície de resposta obtida, construa os metamodelos simples.*
 - 2.4 *Use o Leave-One-Out e Cross-Validation para definir os pesos para um metamodelo de mistura.*
 - 2.5 *Encontre o novo candidato através do mínimo encontrado pelo metamodelo escolhido no passo anterior.*
 - 2.6 *Atualize a superfície de resposta com o novo candidato.*
 - 2.7 *Volte ao passo 2.3 até que uma condição de parada seja satisfeita.*
3. *Determine regiões com os melhores candidatos, onde o algoritmo fará uma busca local.*
4. *Em cada região determinada no passo anterior, faça os seguintes passos*
 - 4.1 *Assuma como superfície de resposta, a última adquirida no processo de busca global.*
 - 4.2 *Constua a superfície de resposta com a amostra gerada e os pontos avaliados na função computacionalmente cara.*
 - 4.3 *A partir da superfície de resposta obtida, construa os metamodelos simples.*

4.4 Use o Leave-One-Out e Cross-Validation para definir os pesos para um meta-modelo de mistura.

4.5 Encontre o novo candidato através do mínimo do melhor metamodelo.

4.6 Atualize a superfície de resposta com o novo candidato.

4.7 Volte ao passo 4.2 até que uma condição de parada seja satisfeita.

Os passos descritos acima representam a forma de um algoritmo que usa mistura de metamodelos para encontrar o mínimo global de um problema do tipo caixa preta, em cada iteração são calculados metamodelos que misturam metamodelos simples. Este tipo de algoritmo resolve o problema encontrado na seção 2.3.2, onde cada metamodelo simples respondia bem a um determinado tipo de problema. Na próxima seção o leitor encontrará os resultados numéricos para os testes realizados na seção 2.3.3 usando a mistura de metamodelos.

2.4.3 Resultados numéricos

Nesta seção, serão apresentados testes realizados com o algoritmo de otimização usando mistura de metamodelos, descrito pelo algoritmo3. Assim como na seção 2.3.3, cada problema foi testado com três tamanhos de amostra $s = 10, 30$ e 50 . Em cada problema, com seu respectivo tamanho de amostra, foram realizados 10 ensaios, totalizando 30 ensaios para cada problema. O algoritmo começa com uma busca global, dividindo o domínio em 2^k regiões de mesmo tamanho, onde k é a dimensão do problema. Em cada região são realizados 15 iterações e a região que contém o ponto de menor valor para função objetivo é tomada como a região mais promissora para encontrar o mínimo global. Então ela é explorada na busca local através da realização de 50 iterações. Ao fim das 50 iterações, o mínimo global será o menor valor encontrado para função objetivo nesta região.

Para cada problema, serão mostrados boxplots para mostrar os resultados dos 10 ensaios para cada tamanho de amostra s . Além disto, será adicionado uma tabela mostrando o passo médio de convergência e a média dos erro entre o mínimo global encontrado com o metamodelo e o verdadeiro mínimo do problema.

Teste do problema B: Função Branin

Os resultados obtidos são apresentados no boxplot da Figura 2.6. A mistura de metamodelos apresentou bons resultados, convergindo para o mínimo global em todos os ensaios e tendo os melhores resultados para os ensaios realizados com tamanhos de amostra $s = 30$ e 50 . Na média, o metamodelo Kriging foi quem sobressaiu, tendo sempre uma participação muito maior que os outros modelos.

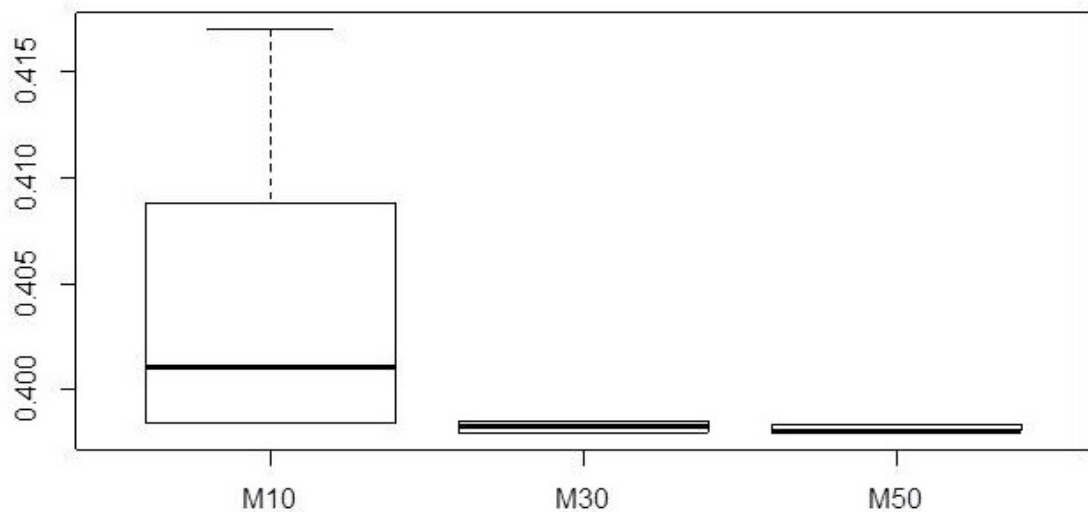


Figura 2.6: Resultados da mistura de metamodelos usando diferentes tamanhos de amostra - Função Brannin

A tabela 2.15 apresenta a média dos erros obtidos e o passo médio de convergência da busca na etapa de busca local, ou seja, foi computado a partir do momento em que já se tem uma região promissora para o mínimo global. O que se observou foi que o metamodelo apresentou erros baixos, sempre convergindo para o mínimo global nos primeiros passos.

Tabela 2.15: Média dos Erros e Convergência - Função Brannin

	Média dos Erros	Convergência
10	0.0053	1
30	0.0003	1
50	0.0002	1

Teste do problema G: Função Goldstein Price

Os resultados obtidos são apresentados no boxplot da Figura 2.7. A mistura de metamodelos apresentou bons resultados, deixando de convergir em apenas um dos ensaios. Na média, o metamodelo Kriging foi quem sobressaiu, tendo sempre uma participação nos pesos da mistura muito maior que os outros modelos.

A tabela 2.16 apresenta a média dos erros obtidos e o passo médio de convergência da busca na etapa de busca local, ou seja, foi computado a partir do momento em que já se tem uma região promissora para o mínimo global. A não convergência de um dos

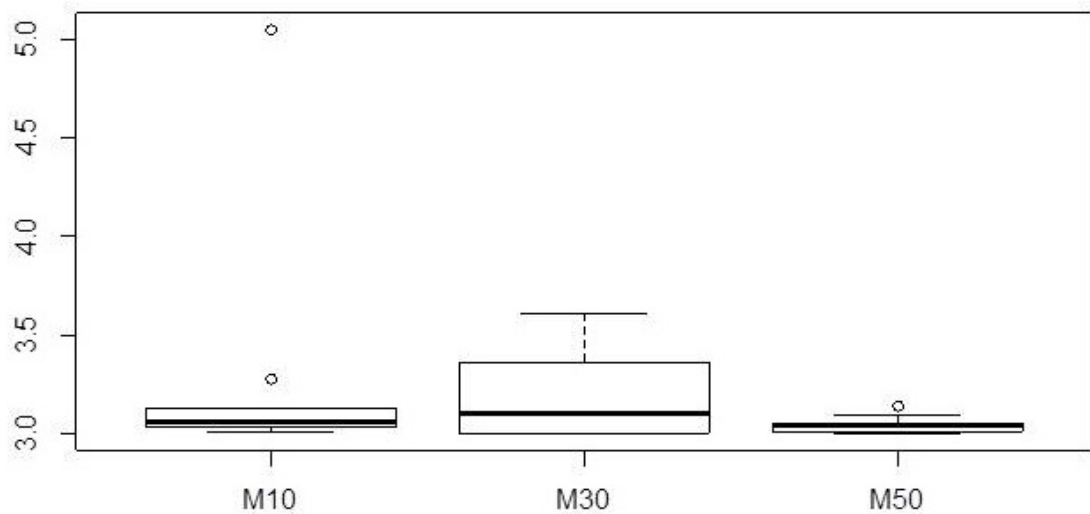


Figura 2.7: Resultados da mistura de metamodelos usando diferentes tamanhos de amostra - Função Goldstein Price

ensaios para amostra de tamanho $s = 10$ fez com que o erro ficasse maior. Para os outros tamanhos de amostra, o algoritmo se comportou bem, com baixos erros e a convergência para o mínimo ficou menor à medida que se aumentou o tamanho da amostra.

Tabela 2.16: Média dos Erros e Convergência - Função Goldstein Price

	Média dos Erros	Convergência
10	0.2814	26
30	0.1808	19
50	0.0490	10

Teste do problema H3: Função Hartmann de três dimensões

Os resultados obtidos são apresentados no boxplot da Figura 2.8. A mistura de metamodelos apresentou bons resultados para os ensaios utilizando amostras do tamanho $s = 30$ e 50 , deixando de convergir em apenas um dos ensaios para amostra de tamanho $s = 50$. Os ensaios com amostra de tamanho $s = 10$ não apresentaram um bom resultado, não convergindo para o mínimo global na maioria dos testes. Na média, o metamodelo Kriging foi quem sobressaiu, seguido da Regressão Polinomial e MARS, o RBF teve muito baixa participação no peso da mistura.

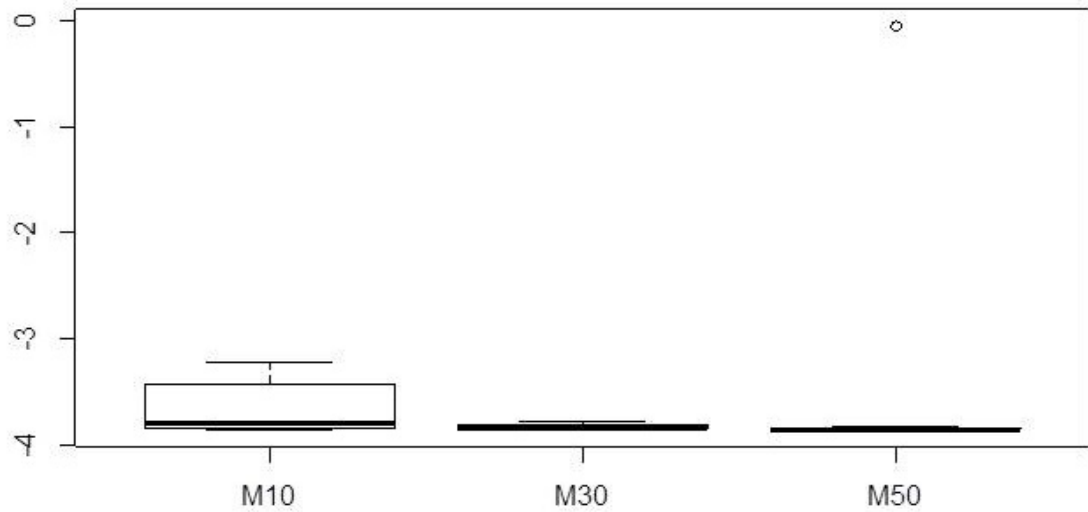


Figura 2.8: Resultados da mistura de metamodelos usando diferentes tamanhos de amostra - Função Hartmann 3

A tabela 2.17 apresenta a média dos erros obtidos e o passo médio de convergência da busca na etapa de busca local, ou seja, foi computado a partir do momento em que já se tem uma região promissora para o mínimo global. A não convergência de um dos ensaios para amostra de tamanho $s = 50$ fez com que o erro ficasse maior. Os ensaios realizados com amostras de tamanho $s = 10$ tiveram erros grandes com relação aos outros ensaios e um passo de convergência maior.

Tabela 2.17: Média dos Erros e Convergência - Função Hartmann 3

	Média dos Erros	Convergência
10	0.9439	26
30	0.0257	7
50	0.3968	4

Teste do problema H6: Função Hartmann de seis dimensões

Os resultados obtidos são apresentados no boxplot da Figura 2.9. A mistura de metamodelos apresentou bons resultados para os ensaios utilizando amostras do tamanho $s = 30$ e 50 . Assim como para o problema tri-dimensional, o metamodelo Kriging foi quem sobressaiu, seguido da Regressão Polinomial e MARS, o RBF teve muito baixa participação no peso da mistura.

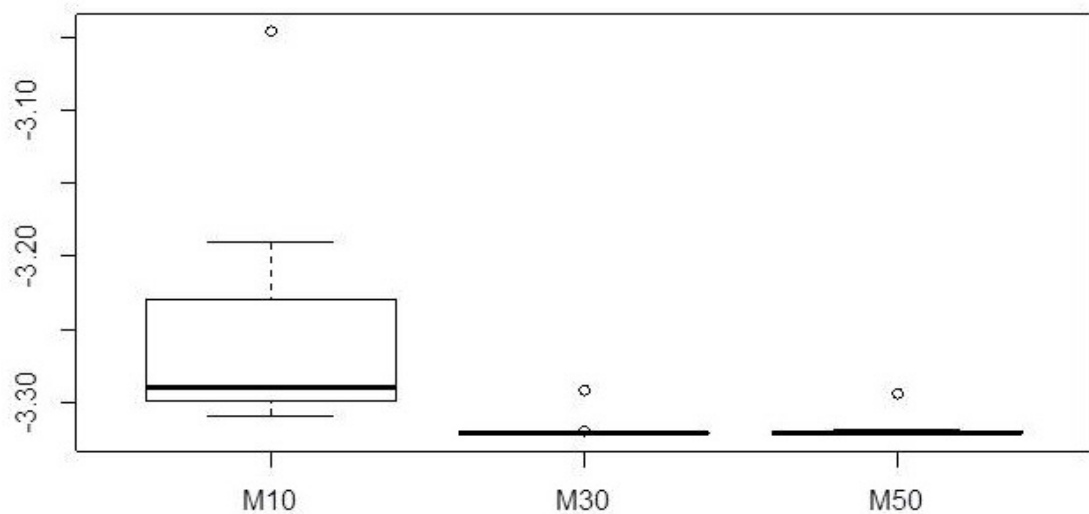


Figura 2.9: Resultados da mistura de metamodelos usando diferentes tamanhos de amostra - Função Hartmann 6

A tabela 2.18 apresenta a média dos erros obtidos e o passo médio de convergência da busca na etapa de busca local, ou seja, foi computado a partir do momento em que já se tem uma região promissora para o mínimo global. O algoritmo apresentou baixos erros e o passo de convergência diminuiu a medida que se aumentou a amostra.

Tabela 2.18: Média dos Erros e Convergência - Função Hartmann 6

	Média dos Erros	Convergência
10	0.0695	24
30	0.0039	9
50	0.0041	2

Teste do problema S10: Função Shekel

Os resultados obtidos são apresentados no boxplot da Figura 2.10. A mistura de metamodelos apresentou bons resultados para quase todos os ensaios, deixando de convergir para apenas um ensaio realizado com amostra do tamanho $s = 10$. Os metamodelos MARS e Kriging tiveram maiores participações na mistura em todos os ensaios.

A tabela 2.19 apresenta a média dos erros obtidos e o passo médio de convergência da busca na etapa de busca local, ou seja, foi computado a partir do momento em que já se

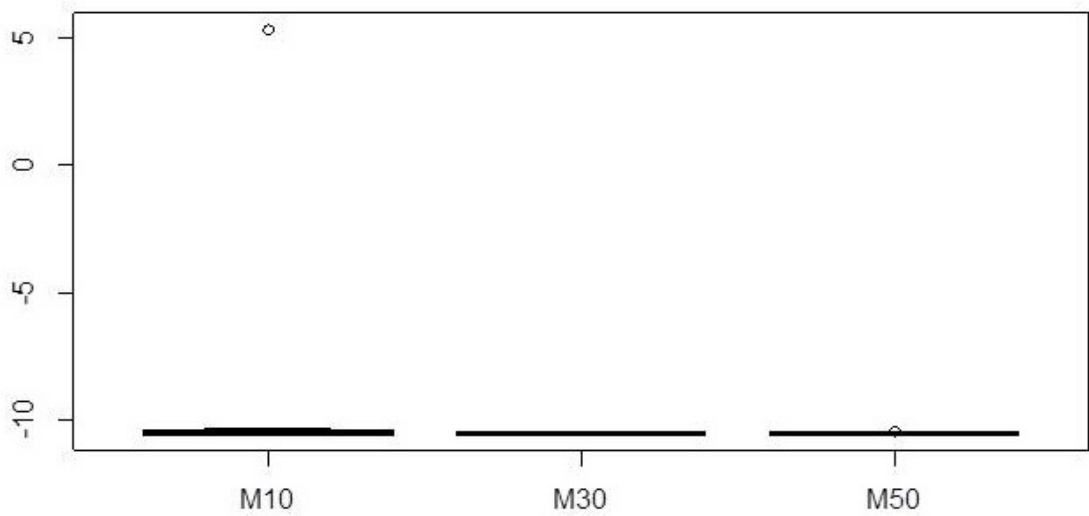


Figura 2.10: Resultados da mistura de metamodelos usando diferentes tamanhos de amostra - Função Shekel

tem uma região promissora para o mínimo global. A não convergência de um ensaio fez com que a média dos erros para os ensaios realizados com amostra $s = 10$ ficasse mais alto. Como já visto para os outros problemas, o passo médio de convergência diminui à medida que o tamanho da amostra aumenta.

Tabela 2.19: Média dos Erros e Convergência - Função Shekel

	Média dos Erros	Convergência
10	1.6442	23
30	0.0226	12
50	0.0217	9

Capítulo 3

Otimização Numérica em Aerodinâmica

3.1 Aerodinâmica

A aerodinâmica é um ramo importante da mecânica dos fluidos e está relacionada com o estudo do movimento dos corpos dentro de fluidos como o próprio ar. Há relatos de que George Cayley[2], considerado o pai da aerodinâmica, já estudava modelos de planadores no século XIX, mas foi no início do século seguinte, com o surgimento dos primeiros automóveis e aviões, que o estudo da aerodinâmica ganhou força. A indústria automotiva e aeronáutica precisava melhorar cada vez mais a eficiência de suas máquinas, os automóveis e aviões precisavam se locomover com o menor atrito possível com o ar, assim seriam mais rápidos e gastariam menos combustível.

A aeronáutica só foi capaz de fazer um avião sair do chão com a ajuda do Princípio de Bernoulli[3] e da segunda lei de Newton. O Princípio de Bernoulli relaciona a velocidade do fluxo do ar e a pressão correspondente, desta forma tem-se que para maiores velocidades de fluxo, correspondem menores valores de pressão, assim como para aumentos de pressão, correspondem diminuições na velocidade de fluxo. A segunda lei de Newton (princípio fundamental da dinâmica) é responsável por explicar o movimento do avião através das quatro forças listadas abaixo.

- Peso (W) - força que atua na direção do centro da terra e sua magnitude corresponde a todas as partes do avião, desde o combustível até as pessoas que estão dentro. Durante um vôo, o peso muda constantemente à medida que o avião consome combustível, assim a distribuição do peso pode mudar, gerando instabilidade no vôo e por isso o piloto deve constantemente ajustar os controles para manter o avião equilibrado.
- Sustentação (L) - força perpendicular à direção do movimento do vôo, é responsável

por compensar o peso e fazer com que o avião permaneça no ar. Todas as partes do avião contribuem para a sustentação do avião, mas as asas são as responsáveis por gerar a maior parte desta força.

- Arrasto (D) - força de resistência do ar que age no sentido contrário e paralelo ao movimento. Sua magnitude depende de vários fatores, tais como a forma como foi constituído o avião, a viscosidade do ar e a velocidade com que se movimenta.
- Tração (T) - força que empurra o avião de forma a compensar o arrasto e é dirigida ao longo do eixo longitudinal. Sua magnitude depende exclusivamente da potência exercida pelo motor.

As três figuras abaixo mostram os movimentos de um avião e como estão agindo as quatro forças citadas acima. A Figura 3.1 mostra a decolagem onde o ângulo de ataque $\theta > 0$ e as forças de tração e sustentação estão sobressaindo sobre as outras. A Figura 3.2 mostra um vôo de cruzeiro onde a aeronave encontra-se em equilíbrio com velocidade constante, nesta situação o ângulo de ataque $\theta = 0$, $T = D$ e $W = L$. Por último, a Figura 3.3 mostra um pouso onde o ângulo de ataque $\theta < 0$ e as forças peso e arrasto sobressaem sobre as outras.

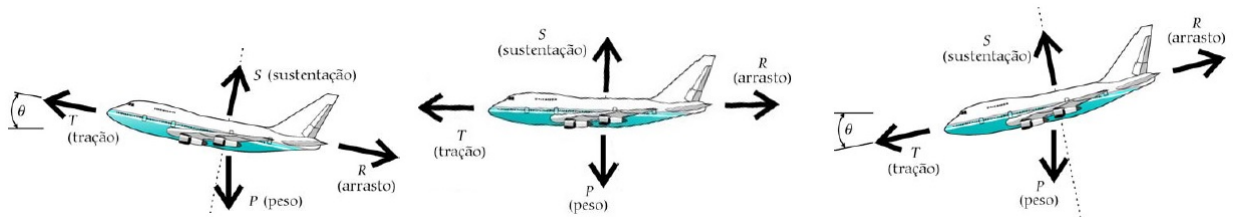


Figura 3.1: Decolagem

Figura 3.2: Cruzeiro

Figura 3.3: Pouso

Depois de entender como agem as principais forças presentes no voo, problemas de otimização podem ser apresentados na busca de encontrar soluções que melhorem a eficiência das aeronaves. As formas como as asas, aerofólios e outras partes são construídas podem diminuir o arrasto do ar e fazer com que o avião gaste menos combustível e obtenha maior velocidade em seu voo. A dificuldade destes problemas é que podem ser tão complexos que fica inviável encontrar formas analíticas para aplicar modelos de otimização convencionais, desta forma, metamodelos podem ser a solução para estes problemas. Nas próximas seções, serão detalhados como construir um problema de otimização que minimize o arrasto presente numa asa de avião durante o seu voo.

3.2 A dinâmica dos fluidos

Os fluidos são substâncias cuja estrutura molecular não oferece resistência à tensões externas, isto é, uma pequena força já é capaz de deformar uma partícula de fluido. Os fluidos contemplam principalmente líquidos e gases, e distinguem-se pela facilidade com que eles são deformados. Apesar de termos fluidos com constituições moleculares bastante diferentes, como, por exemplo, distâncias entre moléculas muito maiores nos gases que nos líquidos, é de interesse compreendermos apenas a sua dinâmica macroscópica. Com isso temos, por exemplo, que os líquidos e gases obedecem as mesmas leis de movimento. Isso ainda nos permite fazer a hipótese de que os fluidos possuem estrutura contínua, apesar de sabermos que microscopicamente possuem uma estrutura molecular complexa. Do ponto de vista prático isso nos permite dizer que por menor que seja uma parcela, ou partícula, de fluido, esta parcela ainda contém infinitas moléculas. Os estudos da dinâmica dos fluidos envolve compreender a evolução temporal e espacial de propriedades de um fluido como o campo de velocidades, pressão, densidade e temperatura.

Neste trabalho será dado maior enfoque ao estudo da pressão que pode ser obtida através das equações de Navier-Stokes [25] (equações diferenciais parciais que descrevem o escoamento dos fluidos). A solução geral destas equações é muito poderosa, pois permite uma descrição completa de todas as possíveis situações do fluxo e assim encontrar os coeficientes de arrasto C_D e sustentação C_L que serão utilizados no algoritmo de otimização, o qual busca a melhor eficiência de um aerofólio de acordo com a sua forma. Entretanto, apenas casos muito simples das equações de Navier Stokes são possíveis de encontrar soluções analiticamente, então os problemas podem ser resolvidos numericamente através do método dos elementos finitos. Porém este tipo de solução consome um tempo computacional muito grande e num algoritmo de otimização que avalie estas equações várias vezes, pode levar muito tempo. Para evitar isto, o algoritmo de otimização que use a mistura de metamodelos, presente na seção 2.4.2, pode ser utilizado.

3.2.1 Equações de Navier-Stokes

As equações que regem a dinâmica dos fluidos são construídas a partir de leis de conservação de massa, momento e energia. Aplicando estas leis a um elemento de fluido em escoamento e considerando as forças que atuam sobre o elemento, resulta as equações de Navier-Stokes. O problema consiste em encontrar os campos velocidade $u = (u_i)_{i=1}^d$ e pressão p do fluxo satisfazendo o domínio $\Omega \subset \mathbb{R}^d (d = 2, 3)$ dados por:

$$u_t + (u \cdot \nabla) u - \nu \Delta u + \frac{1}{\rho} \nabla p = f, \text{ em } [0, T] \times \Omega, \quad (3.1a)$$

$$\nabla \cdot u = 0 \text{ em } [0, T] \times \Omega, \quad (3.1b)$$

$$u = 0 \text{ em } [0, T] \times \partial\Omega, \quad (3.1c)$$

$$u(0, \cdot) = u_0 \text{ em } \Omega, \quad (3.1d)$$

$$\int_{\Omega} p dx dy = 0 \text{ em } [0, T], \quad (3.1e)$$

onde ν é a viscosidade do fluido, ρ é a massa específica, $\nabla = (\partial_i)_{i=1}^d$, $\Delta = \nabla \cdot \nabla$, f é a força atuante, $\Omega \subset \mathbb{R}^2$ é o domínio, $\partial\Omega$ é a fronteira, u_0 é a velocidade inicial e $[0, T]$ é o intervalo de tempo.

Observe que a normalização da pressão 3.1e tem o objetivo de assegurar a unicidade da pressão, pois se p é solução, $p + c$ também poderia ser, para qualquer $c \in \mathbb{R}$.

Note que eliminando o primeiro termo do lado esquerdo da equação 3.1b obtem-se as equações de Navier-Stokes em regime estacionário. Um exemplo do método de Newton para encontrar soluções para estas equações pode ser visto na próxima seção.

3.2.2 Método de Newton para Equações Navier-Stokes Estacionárias

O problema consiste em encontrar os campos velocidade e pressão do fluxo satisfazendo o domínio $\Omega \subset \mathbb{R}^d (d = 2, 3)$ das equações de Newton-Stokes Estacionária dados por:

$$(u \cdot \nabla) u - \nu \Delta u + \frac{1}{\rho} \nabla p = 0, \quad (3.2a)$$

$$\nabla \cdot u = 0, \quad (3.2b)$$

A forma fraca é encontrar u e p tal que para $\forall v$ e $\forall q$

$$\int_{\Omega} ((u \cdot \nabla) u) v + \nu \nabla u : \nabla v - p \nabla \cdot v - q \nabla \cdot u = 0 \quad (3.3)$$

O algoritmo de Newton para resolver o problema não linear é:

Encontrar $u \in V$ tal que $F(u) = 0$ onde $F : V \rightarrow V$.

1. Escolha $u_0 \in \mathbb{R}^n$
2. tome $i = 0$
3. Resolva $DF(u_i)w_i = F(u_i)$
4. $u_{i+1} = u_i - w_i$
5. $i = i + 1$
6. volte ao passo 3 até que $\|w_i\| \leq \epsilon$

Onde $DF(u)$ é o diferencial de F no ponto u , esta é uma aplicação linear tal que:

$$F(u + \delta) = F(u) + DF(u)\delta + o(\delta)$$

Para Navier Stokes, F e DF são:

$$F(u, p) = \int_{\Omega} ((u \cdot \nabla)u) \cdot v + \nu \nabla u : \nabla v - p \nabla \cdot v - q \nabla \cdot u$$

$$DF(u, p)(\delta u, \delta p) = \int_{\Omega} ((\delta u \cdot \nabla)u) \cdot v + ((u \cdot \nabla)\delta u) \cdot v + \nu \nabla \delta u : \nabla v - \delta p \nabla \cdot v - q \nabla \cdot \delta u$$

3.2.3 Coeficientes de Arrasto e Sustentação

A solução geral das equações de Navier-Stokes é muito poderosa, pois permite uma descrição completa de todas as possíveis situações do fluxo que darão a distribuição de pressão C_p e neste trabalho será usado o software FreeFem++ que encontra solução de equações diferenciais parciais numericamente através da técnica de elementos finitos.

A performance de um aerofólio pode ser caracterizado pelos coeficientes de arrasto (C_D) e sustentação (C_L):

- C_D - corresponde à força na direção do fluxo, causando a resistência da aeronave frente ao ar.
- C_L - corresponde à força na direção ortogonal ao fluxo, responsável pela sustentação da aeronave no ar.

Os coeficientes de arrasto e sustentação presentes num aerofólio com ângulo de ataque α podem ser obtidos a partir das seguintes equações:

$$C_D = C_x \cos \alpha + C_z \sin \alpha \quad (3.4)$$

$$C_L = -C_x \sin \alpha + C_z \cos \alpha \quad (3.5)$$

onde C_x e C_z são coeficientes de força horizontal e normal respectivamente, e são calculados pela integração da distribuição de pressão em torno da superfície do aerofólio, como as seguintes equações:

$$C_x = \oint C_p \sin \theta ds \quad (3.6)$$

$$C_z = - \oint C_p \cos \theta ds \quad (3.7)$$

onde ds é a área da superfície do aerofólio e θ é o ângulo que a área faz com o eixo horizontal, a figura 3.4 ilustra melhor o ângulo θ .

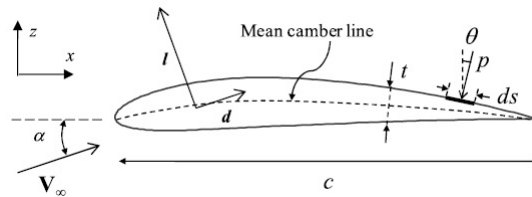


Figura 3.4: Análise do Aerofólio

3.3 FreeFem++

FreeFem++ é um software livre que resolve equações diferenciais parciais numericamente e como o próprio nome diz, este software é baseado no método dos elementos finitos para encontrar as soluções. Além do mais, o FreeFem++ é altamente adaptativo, pois pode trabalhar com muitos fenômenos que envolvem sistemas acoplados que exigem diferentes aproximações de elementos finitos e possivelmente diferentes malhas.

O software tem como principais características a descrição do problema por sua formulação variacional, trabalha com multi-variáveis e multi-equações, fácil descrição analíticas das fronteiras por pedaços, gerador automático de malhas, alto nível de interação com o usuário, grande variedade de elementos finitos, ferramentas para definir descontinuidades, boa plataforma gráfica, etc.

Segue abaixo um resumo de como trabalhar com o FreeFem++ para encontrar soluções através do Método dos Elementos Finitos.

- *As restrições Γ são descritas analiticamente por uma equação paramétrica de x e y .*
- *A malha é gerada dentro do domínio com um certo número de pontos, como pode ser visto na Figura 3.5 que representa a malha usando o aerofólio NACA0012 como restrição.*

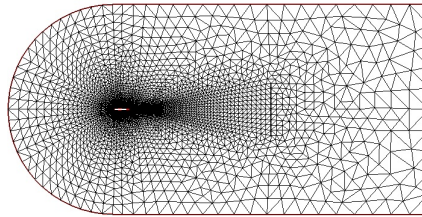


Figura 3.5: Malha - NACA0012

- *Um espaço de elementos finitos é definido.*
- *O lado direito da equação é definido.*
- *O problema é então resolvido por algum método de elementos finitos e a solução do campo pode ser vista na Figura 3.6 que representa a distribuição do fluxo em volta do aerólio NACA0012.*

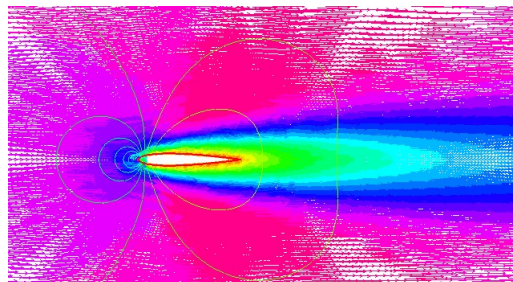


Figura 3.6: Distribuição do Fluxo - NACA0012

3.4 Parâmetros da forma do aerofólio

A forma do aerofólio pode ser escolhida de acordo com a famosa fórmula de 4 dígitos, NACA. Ela é interessante porque expressa a forma do aerofólio como uma função analítica de três parâmetros, que controlam o arqueamento máximo, a localização do ponto de arqueamento máximo e a espessura máxima do aerofólio, como pode ser visto na Figura 3.7.

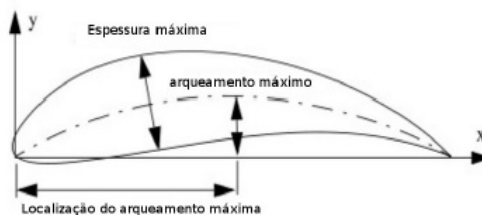


Figura 3.7: Parâmetros do aerofólio para a fórmula NACA

Uma grande variedade de aerofólios podem ser criadas variando estes três parâmetros. Um sistema numérico é usado para definir NACA de 4 dígitos, onde o primeiro número indica o valor máximo da ordenada da linha média em porcentagem com relação ao comprimento da corda. O segundo número indica a distância a partir da aresta, está localizado o ponto de arqueamento máximo em porcentagem com relação ao comprimento da corda. Os dois últimos números indicam a espessura máxima em porcentagem com relação ao comprimento da corda. Assim, o NACA0012 tem 0 por cento de arqueamento localizado a 0 por cento da corda a partir da aresta e 12 por cento de espessura. A seção de uma asa é obtida combinando a linha de arqueamento máximo e a distribuição de espessura como indica a 3.7.

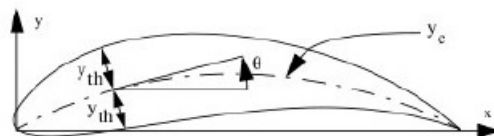


Figura 3.8: Parâmetros do aerofólio para a fórmula NACA

$$\begin{cases} x_u = x - y_{th} \cos \theta \\ y_u = y_c - y_{th} \sin \theta \end{cases} \quad (3.8)$$

e

$$\begin{cases} x_L = x - y_{th} \cos \theta \\ y_L = y_c - y_{th} \sin \theta \end{cases} \quad (3.9)$$

Onde (x_u, y_u) e (x_L, y_L) são as coordenadas das partes superior e inferior respectivamente.

A distribuição de espessura e o arqueamento são dados por:

$$y_{th} = 5tc \left[0.17735 \sqrt{\frac{x}{c}} - 0.075597 \frac{x}{c} - 0.212836 \left(\frac{x}{c}\right)^2 + 0.17363 \left(\frac{x}{c}\right)^3 - 0.06254 \left(\frac{x}{c}\right)^4 \right] \quad (3.10)$$

$$y_c = \frac{m}{p^2} \left[2p \frac{x}{c} - \left(\frac{x}{c}\right)^2 \right] \text{ para } \frac{x}{c} \leq p \quad (3.11)$$

$$y_c = \frac{m}{1-p^2} \left[1 - 2p + 2p \frac{x}{c} - \left(\frac{x}{c}\right)^2 \right] \text{ para } p \leq \frac{x}{c} \quad (3.12)$$

Nessas expressões c é o comprimento da corda, m é o arqueamento máximo, p é a localização deste máximo e t é a espessura máxima.

Capítulo 4

Resultados

Os resultados deste capítulo estão sendo obtidos e serão incluídos na versão final do trabalho.

Capítulo 5

Conclusão

Os resultados deste capítulo estão sendo obtidos e serão incluídos na versão final do trabalho.

Referências Bibliográficas

- [1] Müller, J., “Surrogate Model Algorithms for Computationally Expensive Black-Box Global Optimization Problems”
- [2] Croicu, A. M., Houssaini, M.Y., Jameson, A. and Klopfer, G., “Robust Airfoil Optimization Using Maximum Expect Value and Expect Maximum Value approaches”, AIAA Journal, Vol.50, N°9, September 2012, pp. 1905-1919
- [3] Sharif, B, Wang,G.G. and ElMekkawy, T. Y., “Mode Pursuing Sampling Method for Discrete Variable Optimization on Expensive Black-Box Functions”, Journal of Mechanical Design, FEBRUARY 2008, Vol. 130 / 021402-1
- [4] Jones, D.R., Schonlau, M. and Welch, W.J., “Efficient Global Optimization of Expensive Black-Box Functions”, Journal of Global Optimization 13 : 455-492, 1998
- [5] Jones, D.R., “A taxonomy of global optimization methods based on response surfaces”, Journal of Global Optimization, 21:345-383, 2001
- [6] Holmström, K., Quttined N.H. and Edvall, M. M., “An Adaptive Radial Basis Algorithm (ARBF) for Expensive Black-Box Mixed-Integer Constrained Global Optimization”
- [7] Gutmann, H.-M., “A radial basis function method for global optimization”, Journal of Global Optimization, 19:201-227, 2001
- [8] Regis, R.G., “Particle swarm with radial basis function surrogates for expensive black-box optimization”, Journal of Computational Science 5 (2014) 12-23
- [9] Lions, P.L., “Mathematical Topics in Fluid Mechanics”, volume 1: Incompressible Models, Oxford University Press, Oxford, 1996
- [10] Tesfahunegn, Y.A.,Koziel,S., Gramanzini, J.-R., Hosder, S., Han, Z.-H. and Leifsson, L. “Application of Direct and Surrogate-Based Optimization to Two-Dimensional Benchmark Aerodynamic Problems: A Comparative Study”, AIAA SciTech, 5-9 January 2015, Kissimmee, Florida, 53rd AIAA Aerospace Sciences Meeting

- [11] Myers,R.H. and Montgomery,D.C. “Response Surface Methodology,Process and Product Optimization using Designed Experiments”, Wiley - Interscience Publication, 1995
- [12] Friedman,J.H., “Multivariate adaptive regression splines”, The Annals of Statistics, 19:1-67, 1991
- [13] Duchon,J.,“Constructive Theory of Functions of Several Variables.” Springer-Verlag, Berlin, 1977
- [14] Powell,M.J.D., “The Theory of Radial Basis Function Approximation”,Advances in Numerical Analysis, vol. 2: wavelets, subdivision algorithms and radial basis functions”, Oxford University Press, Oxford, pp. 105-210, 1992
- [15] Matheron,G. “Principles of geostatistics”, Economic Geology, 58:1246-1266, 1963.
- [16] Dixon,L.C.W. and Szegö, G., “The global optimization problem: an introduction In: Towards Global Optimization”, volume 2. North-Holland, Amsterdam, 1978
- [17] Goel,T., Haftka,R.T., Shyy,W. e Queipo,N.V., “Ensemble of surrogates”,Structural Multidisciplinary Optimization, 33:199-216, 2007
- [18] Viana,F.A.C. & Haftka,R.T., “Using multiple surrogates for minimization of the RMS error in metamodeling”, In Proceedings of the ASME 2008, International Design Engineering Technical Conferences & Computers and Information in Engineering Conference DETC2008-49240, 2008
- [19] Dempster,A.P., “A generalization of Bayesian inference”, Journal of the Royal Statistical Society, Series B 30:205-247, 1968
- [20] Shafer,G., “A Mathematical Theory of Evidence”, Princeton University Press, 1976
- [21] Myers,R.H. e Montgomery, D.C., “Response Surface Methodology”,Process and Product Optimization using Designed Experiments, Wiley-Interscience Publication, 1995
- [22] Tolson,B.A., Asadzadeh,A., Maier,H.R. e Zecchin,A., ”Hybrid discrete dynamically dimensioned search (HD-DDS) algorithm for water distribution system design optimization.”, Water Resources Research, pages W12416, DOI:10.1029/2008WR007673, 20
- [23] Abramson,M.A., Audet,C., Couture,G., Dennis Jr,J.E. e Le Digabel,S., ”The NOMAD project”, Software disponível em <http://www.gerad.ca/nomad>, 2011
- [24] Koziel, S. e Leifsson, L., ”Surrogate-Based Aerodynamic Shape Optimization by Variable-Resolution Models”, AIAA JOURNAL Vol. 51, No. 1, January 2013

- [25] BATCHELOR, G. K. "An introduction to fluid dynamics", Cambridge University Press, 1999
- [26] FERZIGER, J. H.; PERIC, M. "Computational methods for fluid dynamics", Springer-Verlag, 1999