

LUIZ HENRIQUE MAIORINO BARBARINI

**SÍNTESE DE CASCOS DE EMBARCAÇÕES ATRAVÉS DE MÉTODOS DE
OTIMIZAÇÃO APLICADOS A CURVAS B-SPLINE**

Dissertação apresentada à Escola Politécnica da
Universidade de São Paulo para obtenção do
Título de Mestre em Engenharia

São Paulo
2007

Livros Grátis

<http://www.livrosgratis.com.br>

Milhares de livros grátis para download.

LUIZ HENRIQUE MAIORINO BARBARINI

**SÍNTESE DE CASCOS DE EMBARCAÇÕES ATRAVÉS DE MÉTODOS DE
OTIMIZAÇÃO APLICADOS À CURVAS B-SPLINE**

Dissertação apresentada à Escola Politécnica
da Universidade de São Paulo para obtenção
do Título de Mestre em Engenharia

Área de Concentração:
Engenharia Mecatrônica

Orientador:
Prof. Dr. Emílio Carlos Nelli Silva

São Paulo
2007

Este exemplar foi revisado e alterado em relação à versão original, sob responsabilidade única do autor e com a anuência de seu orientador.

São Paulo, de março de 2007.

Assinatura do autor _____

Assinatura do orientador _____

FICHA CATALOGRÁFICA

Barbarini, Luiz Henrique Maiorino

Síntese de cascos de embarcações através de métodos de otimização aplicados a curvas B-spline / L.H.M. Barbarini. -- ed.rev. -- São Paulo, 2007.

125 p.

Dissertação (Mestrado) - Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia Mecatrônica e de Sistemas Mecânicos.

1.Casco de embarcações (Projeto) 2.Veleiros (Projeto) 3.Superfícies de resposta (Otimização) I.Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia Mecatrônica e de Sistemas Mecânicos II.t.

“A mente que se abre a uma nova idéia
jamais voltará ao seu tamanho original”.

Albert Einstein

À minha família, que construiu o alicerce desta obra desde meus primeiros passos.

AGRADECIMENTOS

Agradeço ao meu orientador pelos conselhos e orientação na construção deste trabalho.

Ao amigo Rodrigo Alvarez, pela cumplicidade e dicas, que muito me ajudaram nos momentos críticos dos estudos.

Também agradeço ao amigo Fábio Salla, por conceber e me apoiar na idéia deste projeto.

RESUMO

Este trabalho apresenta uma ferramenta flexível e eficiente para o campo de projeto e desenho preliminar de cascos de embarcações. O estado da arte de projeto de cascos de embarcações consiste em um processo iterativo onde, primeiramente, definem-se parâmetros geométricos, de estabilidade, e hidrodinâmicos. Em seguida, o casco é modelado através de curvas paramétricas. submetido a uma análise de desempenho, e, após isso, é feita a tomada de decisão para otimizar os parâmetros iniciais. O ciclo se repete até que se atinja o resultado desejado.

O que será apresentado cumpre com a parte de modelagem paramétrica, e é capaz de gerar de forma automática e otimizada curvas que compõem o projeto de uma embarcação de recreio. A ferramenta recebe parâmetros de alto nível, como razões adimensionais, coeficientes de forma e restrições, e desenha tridimensionalmente um casco intrinsecamente suave, de forma que todos os parâmetros especificados sejam atendidos. Uma metodologia de projeto de cascos de veleiros é utilizada em conjunto com curvas B-Splines, submetidas a um algoritmo de otimização, em cuja função objetivo se empregam critérios de suavidade (“fairness criteria”). O problema de otimização de curvas B-Splines é não linear, pela própria formulação da função objetivo e suas restrições. Para tal, utiliza-se um método de Programação Quadrática Seqüencial.

Para a otimização dos parâmetros principais do casco, utiliza-se uma metodologia de baixo custo, baseada na técnica de Superfície de Resposta, que otimiza uma função de mérito, no caso, a resistência residual, variando-se dois parâmetros do casco.

A interface com softwares CAD é desenvolvida com o objetivo de se realizar outras análises, como de curvatura da superfície, além do projeto do restante da embarcação. Neste também pode ser gerado o bloco sólido para integração com softwares de CAM para a usinagem e fabricação do mesmo.

ABSTRACT

This article presents a flexible and efficient tool to the field of ship project and preliminary design by using an approach of parametric modeling of hull forms. The state of the art of design of hulls consists in an iterative process where, first, geometric, stability and hydrodynamic parameters are defined. After that, the hull is modeled through parametric curves, simulated in a performance analysis software, and, after that, it is taken the decision to optimize the initial parameters. The cycle repeats until it reaches the desired result.

This work deals with the part of parametric modeling, and is capable to generate, in an automatic way, optimized curves that compose the design of the hull of a yacht. The tool receives high level parameters, such as non-dimensional ratios, coefficients and restrictions, and draws a three-dimensional hull, intrinsically fair, fulfilling all the specified parameters.

A methodology for the design of sail-boats is presented together with B-Splines curves, submitted to an optimization algorithm, using fairness criteria as the objective function. The optimization problem of B-Splines curves is not linear, because of its mathematical formulation, the objective function and constraints adopted. Therefore, a method of Sequential Quadratic Programming is used.

For the optimization of the main parameters of the hull, a cost-effective methodology, based on the Surface Response Method, is proposed, optimizing a merit criterion, such as, the residuary resistance, changing two parameters of the hull.

The interface with CAD software is implemented in order to allow other analyses, such as surface curvature, and the design of the other parts of the boat. The drawing also can be transformed into a solid model, and integrated with a CAM software to be ready for its manufacturing.

SUMÁRIO

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS

LISTA DE SÍMBOLOS

1	INTRODUÇÃO.....	1
1.1	História do projeto de embarcações a vela	1
1.2	Otimização aplicado ao projeto de veleiros	4
1.2.1	Construção de cascos através de processos com CNC.....	9
1.3	Revisão Bibliográfica	12
1.4	Motivação.....	15
1.5	Objetivos	16
1.6	Organização do texto	17
2	CURVAS PARAMÉTRICAS	19
2.1	Curvas paramétricas	20
2.2	Formulação matricial e continuidade de curvas paramétricas	22
2.3	Curvas de Hermite.....	24
2.4	Curvas de Bézier	27
2.5	Curvas B-Splines.....	32
2.5.1	Funções de base.....	34
2.5.2	Curvas B-Splines não-uniformes e B-Splines cúbicas	36
2.5.3	B-Splines uniformes	36
2.6	Comparação entre as curvas apresentadas	40
3	CONCEITOS DE OTIMIZAÇÃO E METODOLOGIA UTILIZADA.....	43
3.1	Variáveis de Projeto.....	43
3.2	Função Objetivo	44
3.3	Restrições	47
3.4	Análise de Sensibilidade	50
3.5	Algoritmo de otimização utilizado.....	52
3.6	Otimização paramétrica do casco.....	56
3.6.1	Método da Superfície de Resposta (MSR).....	57

3.6.2	Projeto de Experimentos	59
4	PRINCÍPIOS DE PROJETO DE CASCOS DE BARCOS À VELA	62
4.1	Definição das dimensões do casco.....	62
4.2	Modelo hidrodinâmico.....	66
5	MODELAGEM PARAMÉTRICA	70
5.1	Proposta de modelagem.....	70
5.2	Curvas paramétricas do casco	72
5.3	Implementação do software	76
5.4	Rotinas utilizadas em ambiente Matlab.....	77
6	RESULTADOS	79
6.1	Aplicação e desempenho do algoritmo.....	79
6.2	Modelagem dos parâmetros do veleiro YD-40	91
6.3	Comparação com o veleiro YD-40.....	100
6.4	Análise e pós-processamento dos resultados em CAD.....	103
6.5	Otimização paramétrica do casco pelo MSR.....	107
7	DISCUSSÃO DOS RESULTADOS	112
7.1	Análise dos resultados de desempenho do algoritmo.....	112
7.2	Discussão dos resultados do casco gerado	113
7.3	Análise dos resultados da otimização pelo MSR.....	113
8	CONCLUSÕES.....	115
8.1	Desenvolvimentos futuros	116
	LISTA DE REFERÊNCIAS	118
	APÊNDICE.....	123

LISTA DE FIGURAS

Figura 1. Desenho de Garcia de Palácios, ano 1587 (PORTUGAL WEB).....	1
Figura 2. Espiral de projeto clássico	2
Figura 3. Tipos de cascos (trimarã, catamarã e monocasco)	3
Figura 4. Ciclo de projeto convencional	6
Figura 5. Ciclo de projeto com ferramenta de parametrização	6
Figura 6. Processo de síntese de um casco com modelo hidrodinâmico	7
Figura 7. Processo de otimização utilizado neste trabalho	8
Figura 8. Etapas de processo construtivo (veleiro Skipper 30)	10
Figura 9. Exemplos de máquinas CNC para usinagem de embarcações	11
Figura 10. Exemplo de curva paramétrica	20
Figura 11. Exemplo de curva com pontos e derivadas definidas.....	21
Figura 12. Exemplo de curva com laço	22
Figura 13. Exemplo de curva que passa por pontos dados	22
Figura 14. Exemplos de Continuidade de curvas: continuidade G^0 , G^1 e C^1	24
Figura 15. Exemplo de uma curva de Hermite	24
Figura 16. Funções de base de Hermite	27
Figura 17. Exemplo de curva de Bézier	28
Figura 18. Polinômios de Bernstein: funções de base para a curva de Bézier.....	29
Figura 19. Polinômios cúbicos de Bernstein.....	31
Figura 20. Curvas de Bézier contida no fecho convexo dos pontos de controle	32
Figura 21. Classificação das curvas em aberta, “clamped” e fechada	34
Figura 22. Funções de base da B-Spline	38
Figura 23. Exemplo de curva com seus parâmetros de forma	49
Figura 24. Conceito de mínimo local e mínimo global	52
Figura 25. Exemplo de uma superfície de resposta (MACHIAVERNI, 2004)	58
Figura 26. CCD genérico com $k=2$	60
Figura 27. Definições das dimensões principais de um casco	62
Figura 28. Principais linhas do desenho de um casco (FONSECA, 2005).....	65
Figura 29. Gráfico comparativo entre resistência residual e viscosa.....	67
Figura 30. Fluxograma da parametrização das curvas.....	73

Figura 31. Curva de áreas seccionais YD-40.....	74
Figura 32. Resultado utilizando a Função Objetivo E_1	75
Figura 33. Resultado utilizando a Função Objetivo E_2	75
Figura 34. Resultado utilizando a Função Objetivo $E_1 + E_2$	76
Figura 35. Diagrama de blocos da estrutura do processo de otimização.....	78
Figura 36. Curva de uma seção do casco de Wigley e interpolação por uma B-Spline ...	79
Figura 37. Problema de otimização para o teste do algoritmo.....	81
Figura 38. Gráfico do número de iterações versus número de pontos de controle.....	82
Figura 39. Gráfico de avaliações da função versus número de pontos de controle.....	82
Figura 40. Simulação do algoritmo com uma B-Spline com 43 pontos de controle.....	84
Figura 41. Simulação do algoritmo com uma B-Spline com 11 pontos de controle.....	84
Figura 42. Detalhe das curvas com as energias E_1 , E_2 , e E_3	86
Figura 43. Simulação com duas otimizações: energia e, em seguida, curvatura.....	87
Figura 44. Gráfico obtido para os valores da Tabela 6.....	88
Figura 45. Análise de curvatura em CAD: curva original, distorcida e otimizada.....	89
Figura 46. Gráfico com os resultados parciais do método de otimização.....	90
Figura 47. Histórico de convergência para a função objetivo.....	90
Figura 48. Vista isométrica do casco obtido.....	91
Figura 49. Vista superior e lateral do casco obtido.....	92
Figura 50. Problema de otimização da curva das áreas seccionais (Curva 1).....	93
Figura 51. Problema de otimização da curva da baliza maior (Curva 2).....	94
Figura 52. Problema de otimização da curva da linha d'água (Curva 3).....	94
Figura 53. Problema de otimização da curva da quilha (Curva 4).....	95
Figura 54. Problema de otimização da curva da linha de borda (Curva 5).....	96
Figura 55. Problema de otimização do plano de balizas (Curva 6).....	97
Figura 56. Curva com valores iniciais dos ângulos de saída das balizas.....	97
Figura 57. Gráfico de convergência para as Curvas 1 e 2.....	98
Figura 58. Gráfico de convergência para a Curva 2.....	98
Figura 59. À direita curva da linha d'água intermediária; à esquerda, novas balizas.....	99
Figura 60. Exemplo de um processo iterando com novas linhas d'água.....	100
Figura 61. O desenho das linhas do veleiro YD-40 (LARSSON et al., 1994).....	101

Figura 62. Comparação entre YD-40, em cima (LARSSON et al., 1994), e casco gerado, embaixo.....	102
Figura 63. Comparação entre balizas do YD-40, em cima (LARSSON et al., 1994), e balizas geradas, embaixo.	103
Figura 64. Representação em programa CAD do casco obtido.....	104
Figura 65. Desenhos da superfície do casco gerado.....	106
Figura 66. Análise da curvatura em CAD do casco gerado.....	107
Figura 67. Resistência residual em função de F_n para o casco gerado.....	108
Figura 68. Pontos simulados segundo o método CCD.....	109
Figura 69. Superfície de Resposta do problema.....	111
Figura 70. Gráfico de convergência do problema.....	111
Figura 71. Bloco sólido gerado a partir do casco em CAD.....	124
Figura 72. Simulação do fresamento do casco gerado.....	124
Figura 73. Quadro com comandos em código G do molde do casco gerado.....	125

LISTA DE TABELAS

Tabela 1. Formação das funções de base de uma B-Spline cúbica uniforme	39
Tabela 2. Comparativo entre curvas polinomiais cúbicas	40
Tabela 3. Tabela genérica de CCD para duas variáveis.....	60
Tabela 4. Dimensões do veleiro YD-40.....	71
Tabela 5. Conjunto de curvas paramétricas do casco	72
Tabela 6. Saídas do programa para o teste comparativo com as curvas.....	88
Tabela 7. Número de iterações do algoritmo	98
Tabela 8. Comparação dos resultados utilizando ferramenta em CAD.....	105
Tabela 9. Tabela de CCD para C_P e LCB	108

LISTA DE ABREVIATURAS

BFGS	Broyden-Fletcher-Goldfarb-Shanno
CAD	Computer Aided Design
CAM	Computer Aided Manufacturing
CASHD	Computer Aided Ship Hull Design
CCD	Central Composite Design
CFD	Computational Fluid Dynamics
CNC	Comando Numérico Computadorizado
DOE	Design of Experiments
DWL	Design Water Line
MSR	Método da Superfície de Resposta
NURBS	Non-Uniform Rational B-Splines
PQ	Programação Quadrática
PQS	Programação Quadrática Seqüencial

LISTA DE SÍMBOLOS

∇c	Volume de deslocamento
A	Área entre a curva e o eixo x
A_{Bi}	Área da Balizas i
A_{BM}	Área da baliza maior
A_{WP}	Área interior a D_{WL} (“Waterplane Área”)
A_x	Área máxima da seção transversal do casco
B_{WLi}	Boca na linha d’água da baliza i
$B_i^n(t)$	Polinômios de Bernstein
B_{max}	Largura máxima do casco (“Beam maximum”)
B_{MAXBi}	Boca máxima da baliza i
B_{POPA}	Boca na baliza da popa
B_{WL}	Largura na linha d’água (“Beam of waterline”)
C	Curvatura em um ponto
C_B	Coefficiente de bloco
C_c	Coefficiente de carena
C_{MED}	Curvatura média em um trecho
C^n	Continuidade paramétrica
C_p	Coefficiente prismático
C_x	Matriz com os coeficientes
D	Distância entre o ponto mais profundo da quilha até a linha de borda
D_c	Pontal sem considerar a dimensão da quilha (“Depth of the canoe body”)
D_{WL}	Linha d’água de projeto (“Design Waterline”)
E	Medida de carenagem; energia de uma curva
E_c	Integral do quadrado da curvatura ao longo do comprimento da curva
FB	Comprimento da borda livre (“freeboard”)
F_n	Número de Froude
g	Aceleração da gravidade
G	Centro de gravidade
G^n	Continuidade geométrica
G_x	Matriz geométrica

k	Multiplicidade de um nó de uma B-Spline
L	Número de segmentos de uma curva B-Spline
LCB	Centro longitudinal de carena (“Longitudinal Center of buoyance”)
L_{OA}	Comprimento máximo do casco (“Length overall”)
L_{WL}	Comprimento da linha d’água (“Length of waterline”)
m	Número de pontos de controle da curva
\mathbf{M}_B	Matriz de Base de Bézier
\mathbf{M}_{BS}	Matriz de base da B-Spline
\mathbf{M}_H	Matriz de Base de Hermite
n	Grau do polinômio da curva
$N_{i,n}(t)$	Função de Base de uma B-Spline
P_i	Pontos de controle da curva
R_r	Resistência Residual
$s(t)$	Função polinomial de uma curva B-Spline
S_W	Superfície molhada (“Wetted Surface”)
t	Variável paramétrica de uma curva; nó
T	Calado
\mathbf{T}	Vetor das variáveis paramétricas
T_c	Calado sem considerar a profundidade da quilha
T_{CBi}	Calado da baliza i
T_{CBM}	Calado da baliza maior
v	Velocidade do veleiro
x_B, y_B	Posição do ponto inicial da curva
x_{BM}	Coordenada da baliza maior
x_C, y_C	Centróide da área A (momento de primeira ordem)
x_E, y_E	Posição do ponto final da curva
x_F	Coordenada no eixo x do centro de flutuação
x_I, y_I	Coordenadas de um ponto intermediário a ser interpolado
x_{LCB}	Coordenada no eixo x do centro de carena
α_B, α_E	Ângulo da tangente no ponto inicial e final da curva
Δc	Massa deslocada

1 INTRODUÇÃO

1.1 História do projeto de embarcações a vela

Não existem registros para a data exata da invenção do primeiro barco à vela. Sabe-se que gregos e romanos utilizavam-se do vento como propulsão de suas embarcações primitivas. O auge dos barcos a vela, no entanto, ocorreu no século XIX. A partir daí, embarcações a vapor dominaram os meios de transporte, deixando as velas apenas ocupando o espaço nos esportes e lazer.

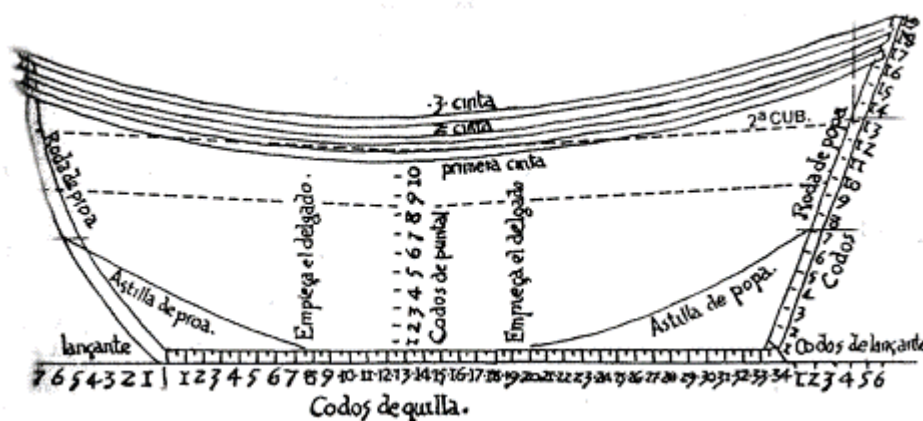


Figura 1. Desenho de Garcia de Palácios, ano 1587 (PORTUGAL WEB).

Veleiros são embarcações propelidas pela ação do vento sobre suas velas. Atualmente, o projeto de um barco à vela evoluiu muito desde quando os desenhos eram feitos no chão, nas Salas de Risco (Figura 1), presentes em todos os estaleiros. Porém, o desenho não deixou de envolver criatividade, experiência, e, principalmente, arte, apesar dos mais modernos recursos de computação gráfica para modelagem tridimensional e ferramentas de simulação e cálculos numéricos.

O processo de projeto clássico de uma embarcação, chamado de espiral de projeto, foi proposto pela primeira vez por J. Harvey Evans em 1954. O método consiste em uma forma racional de projeto, a qual converge para um sistema possível, através da iteração e refinamento das informações. O sucesso de um projeto está relacionado com a interdependência dos seus subsistemas, e a espiral explora este quesito

fortemente. A cada ciclo, todo o sistema é recalculado e, no ciclo final, todos estes subsistemas estão interagindo conjuntamente. Cabe ressaltar que a configuração encontrada está a uma distância desconhecida do projeto ótimo, ou seja, da melhor solução para um desempenho específico.

A Figura 2 ilustra um exemplo de espiral de projeto de um veleiro. Nela, porém, ainda não é levado em conta aspectos financeiros do projeto, ponto de grande relevância em um projeto de Engenharia. Esta informação deve ser levada em consideração ainda na concepção do projeto, quando também se devem considerar informações sobre as embarcações semelhantes que deverão servir como modelo e como referência para alguns coeficientes da geometria.

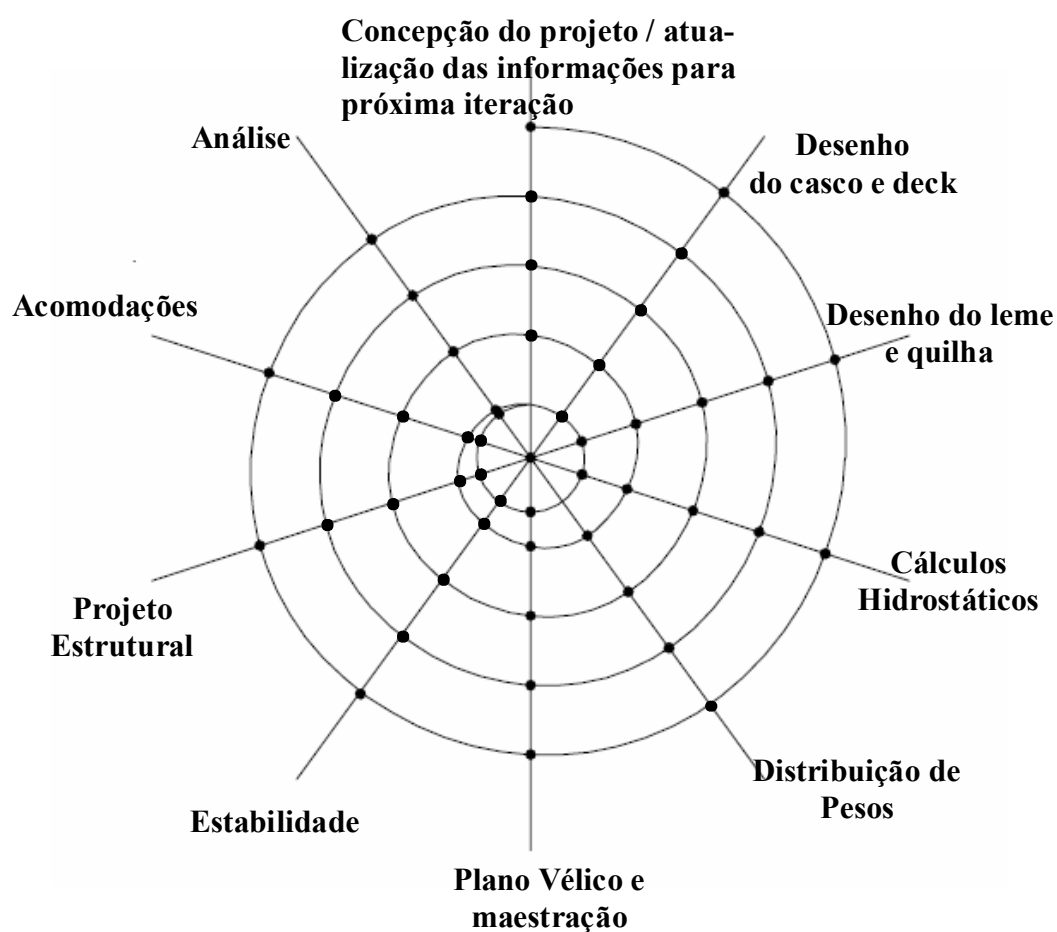


Figura 2. Espiral de projeto clássico

O foco deste trabalho será a proposição de um método de projeto de cascos de veleiros, visando tratar sua concepção como um problema de otimização. Todavia, o desenho do casco não é um fim em si mesmo. Este se faz necessário para que as próximas etapas da espiral de projeto possam ser avaliadas, completando-se o ciclo. Três configurações de cascos classificam os veleiros em monocascos, catamarans e trimarans. Será estudada somente a configuração mais difundida atualmente, chamada de “Monocasco em Sloop”.

Os catamarans, com dois cascos, são conhecidos por sua segurança e estabilidade, o que garante o conforto à tripulação, por não adernar velejando. Trimarans (três cascos) notabilizam-se pela maior velocidade em relação à catamarans e monocascos de mesmo comprimento. Em contrapartida, os monocascos destacam-se pela sua maior eficiência no aproveitamento dos ventos e pelo maior espaço interno para acomodações.



Figura 3. Tipos de cascos (trimarã, catamarã e monocasco)

1.2 Otimização aplicado ao projeto de veleiros

Segundo (VALDENAZZI et al., 2002), otimização da forma de cascos é uma prática utilizada desde quando se começou a projetar embarcações. Historicamente, o primeiro trabalho documentado que aplica sistematicamente formas matemáticas no projeto de embarcações foi escrito pelo um construtor naval sueco, chamado F. H. Chapman, em 1760 (NOWACKI, 1970). O estado da arte em projetos de casco de veleiros lança mão de modernas ferramentas envolvendo algoritmos sofisticados de otimização. Além de cascos de barcos, a literatura que trata métodos de otimização aplicados à Engenharia Naval também utiliza estes algoritmos em outros sistemas como configuração das velas (TYLER et al., 2002) ou otimização da configuração de um veleiro a fim de atender a um requisito de projeto ou desempenho específico para uma regata, como por exemplo, otimizar as dimensões da maestração, parâmetros dos casco, minimizar resultante de forças longitudinais, entre outros (SALLA, 2005). A abordagem através de otimização multidisciplinar também é vista em (GIASSI et al., 2003), segundo o qual, este método ainda é pouco difundido em problemas de projeto de embarcações.

O projeto otimizado do casco de uma embarcação é alvo de grande interesse da indústria naval. A tecnologia e o desenvolvimento de softwares reduzem drasticamente o tempo de lançamento de um novo produto no mercado, além de propiciar um método iterativo para projetos em que se deseja um desempenho específico ou uma característica particular (ABT et al., 2001). Desta maneira, ao se tratar o projeto de cascos como um problema de otimização, busca-se evitar percorrer muitos ciclos da espiral de projeto, alcançando uma configuração desejada de forma mais eficaz.

A Europa é uma das regiões que mais produzem trabalhos nesta área. No artigo (HARRIES et al., 2003), grupos da Alemanha, Suécia, França, Holanda e Itália apresentam um trabalho patrocinado por um fundo de pesquisa, no qual uma competição é promovida para por à prova as ferramentas desenvolvida por cada

equipe. O desafio foi otimizar um mesmo casco de um navio RoRo, utilizando os softwares de cada grupo que aplicavam algoritmos de otimização. O interessante são os resultados obtidos percorrendo diferentes caminhos. Cada grupo implementa sua própria estratégia de otimização nas linhas de código do seu software. Os algoritmos de otimização também variam, sendo que são utilizados os mais diversos tipos de algoritmos, como Método dos Gradientes e algoritmos genéticos multi-objetivos.

Em contrapartida, o que vem se tornando um consenso no campo das pesquisas de modelagem de cascos é a utilização de curvas paramétricas B-Splines (HARRIES et al., 1999; ABT et al., 2001; GIASSI et al., 2003; NARLI et al., 2003). Estas apresentam todos os recursos desejados para a manipulação de curvas, além de uma formulação matemática simples e flexível. Contudo, o problema de otimização envolvendo curvas B-Splines é não-linear, e requer ser resolvido com algoritmos apropriados de programação não linear (exemplo: Programação Quadrática Seqüencial) ou métodos mais sofisticados. Pode-se citar como principal vantagem das curvas B-Spline o controle do tipo local, no qual a variação de um ponto de controle não afeta todos os pontos da curva. Os melhores softwares de CAD ou CASHD (“Computer Aided Ship Hull Design”) encontrados no mercado possuem recursos para desenhos de curvas B-Splines, o que garante a possibilidade de utilização destes, aplicados à modelagem obtida aqui. Esta facilidade poderá ser observada ao longo deste trabalho, onde o desenho final é exportado para um software de CAD, onde são explorados vários recursos de manipulação de superfícies.

O projeto de um barco através da parametrização das curvas que compõem um casco tem como objetivo alterar o ciclo de projeto convencional. Em um projeto, sem o auxílio de um software que gere as curvas paramétricas, é realizado o ciclo exibido na Figura 4, onde o projetista utiliza um software CAD para o desenho do casco, e a partir dele extrai as propriedades do mesmo, avaliando e ajustando as mesmas completando o ciclo.

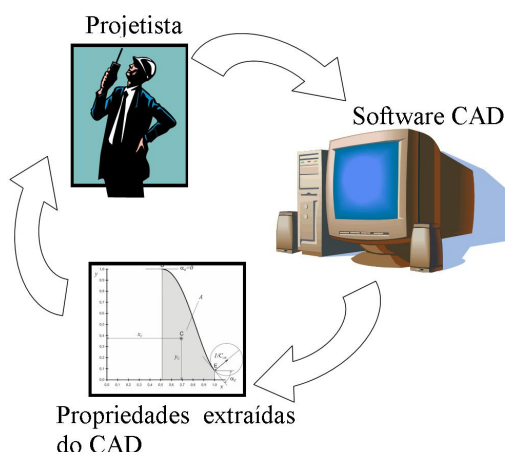


Figura 4. Ciclo de projeto convencional

Na abordagem com a parametrização das curvas do casco, o ciclo se inverte (Figura 5). O projetista especifica numericamente as propriedades e a forma do casco é gerada. Logo, prioriza-se o trabalho de concepção deixando o traçado das linhas como secundário.

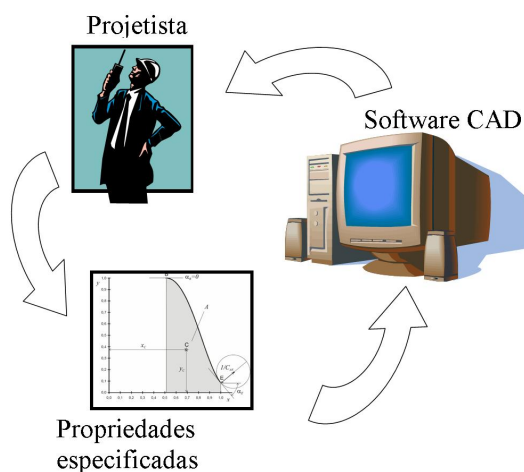


Figura 5. Ciclo de projeto com ferramenta de parametrização

As ferramentas atualmente desenvolvidas (exemplo FRIENDSHIP-Modeler, ABT et al., 2001) seguem, na maioria das vezes, um fluxograma de otimização parecido, onde, inicialmente, parametriza-se o casco, criando um modelo tridimensional de sua forma. Em seguida, simula-se seu desempenho em um software CFD, e, logo, com a análise dos resultados, um algoritmo de otimização tomará a decisão de quais

parâmetros devem ser alterados para que se atinja um critério de mérito desejado. O diagrama de blocos deste processo está ilustrado abaixo, na Figura 6.

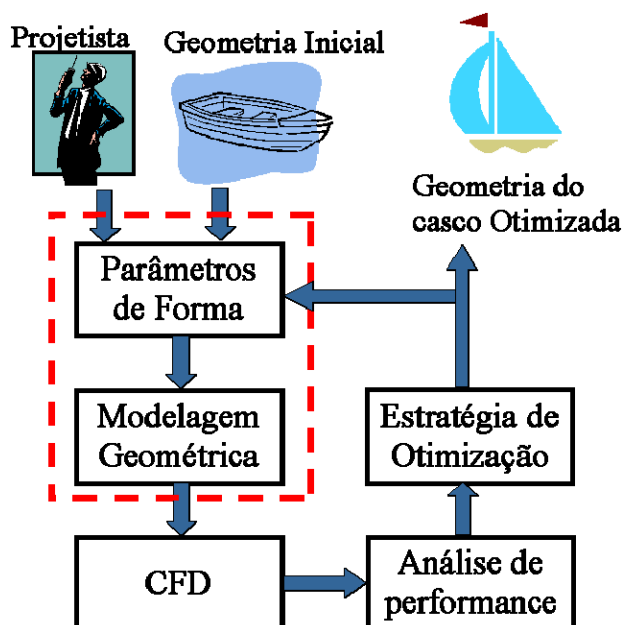


Figura 6. Processo de síntese de um casco com modelo hidrodinâmico

O foco deste trabalho será o desenvolvimento de uma nova metodologia para solucionar os problemas apresentados nos dois primeiros blocos do diagrama (em vermelho), ou seja, a modelagem paramétrica do casco. Adicionalmente, com o intuito de fechar a malha do processo (Figura 7), será apresentada uma solução de baixo custo para a otimização dos parâmetros geométricos do casco, a fim de minimizar a resistência residual do casco. O método utilizado será o de Superfície de Resposta, aplicado a dois parâmetros do casco. Em comparação com o CFD, o cálculo do desempenho do veleiro é mais limitado, no entanto, por ser possível avaliar um alto número de configurações a um custo baixo, acaba provendo resultados finais excelentes, encontrando a configuração ótima do sistema.

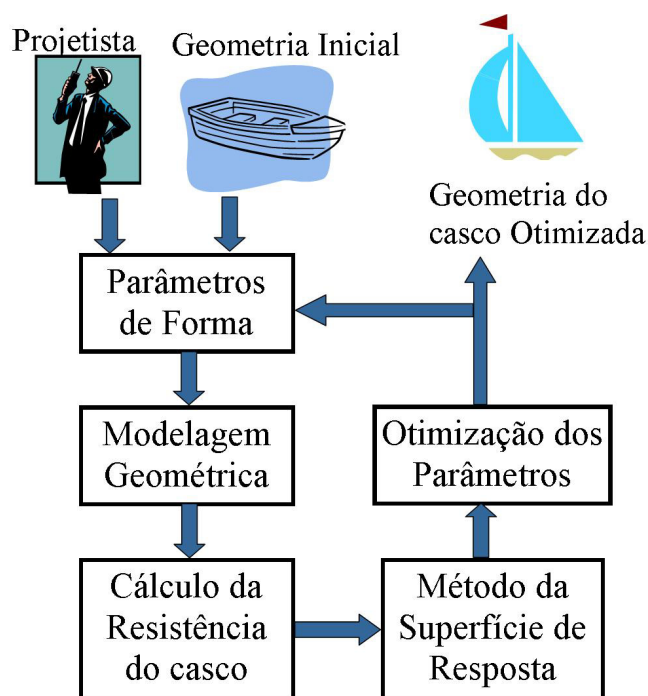


Figura 7. Processo de otimização utilizado neste trabalho

A abordagem paramétrica permite traduzir a linguagem de alto nível de um consumidor para os parâmetros e grandezas físicas de um casco. Modelar um casco significa selecionar os parâmetros a serem levados em conta e designar valores corretos a eles. A suavidade da forma do casco é requerida para melhorar seu desempenho hidrodinâmico, assim como, suas propriedades estéticas. A solução deste complexo problema é alcançada reduzindo-se o problema de suavização de um objeto tridimensional, para uma série de problemas bidimensionais.

A suavidade é geralmente considerada como uma medida subjetiva e dependente do julgamento do projetista. Um dos objetivos principais do processo de suavização é a automatização e, logo, minimizar a intervenção humana subjetiva, que pode conduzir a muitas inconsistências na geometria resultante do casco. Apesar da carenagem do casco ser um objetivo secundário no projeto, esta é altamente relevante em se tratando de embarcações de alto desempenho. Logo, o desenvolvimento de procedimentos automatizados, no qual o processo de suavização é definido de maneira objetiva e alcançado dentro das restrições impostas de projeto, é claramente desejável.

A ferramenta projetada foi desenhada pensando na flexibilidade e adaptabilidade a diferentes modelagens. Ou seja, é possível programar o software para desenhos de diferentes tipos de cascos ou outra superfície semelhante. Também, já é prevista a integração com softwares de CAD, a fim de se exportar o modelo do casco para a finalização do projeto da embarcação completa, ou para um pós-processamento com alguma ferramenta específica dos softwares comerciais. Um passo adiante também será demonstrado (ver Apêndice), integrando o software CAD com as ferramentas de CAM, gerando o código G do casco desenhado, estando pronto para usinagem em CNC.

Assim, todo o processo desde a concepção até a manufatura do casco é abordado de forma automática, produzindo como resultado final um dispositivo rápido e eficiente tanto para a prototipagem de um casco como para a construção de um produto final otimizado para um critério de mérito desejado.

1.2.1 Construção de cascos através de processos com CNC

A fabricação é a última etapa no projeto de um barco. Além do desempenho e estética do casco, a escolha do processo de fabricação é de suma importância e também deve ser levado em conta em projetos de barcos, seja ele um veleiro, uma lancha ou um navio. Atualmente, pouco ainda se ouve falar em processos automáticos computadorizados, prevalecendo processos que envolvem a construção manual de seções ou partes menores, e posterior montagem completa do casco em si. Esta é uma das razões pela qual se aborda o processo de construção por CNC neste trabalho, já que a modelagem paramétrica combinada com a usinagem computadorizada é o estado da arte em projeto de cascos.

Considerando embarcações de recreio, o mais comum processo de fabricação é através de laminação com fibra de vidro. O processo consiste em construir um modelo idêntico ao modelo a ser fabricado. A partir deste, constrói-se um molde que será utilizado para a fabricação em série de várias unidades. Isso porque a construção

do molde possui custo e tempo de fabricação elevados, somente justificando o investimento utilizando-o diversas vezes. A Figura 8 mostra quão trabalhoso é o processo de se construir um novo protótipo. Da esquerda para a direita, de cima para baixo tem-se: construção em madeira das seções do casco, fechamento do modelo, forma do casco com reforços de poliuretano, casco pronto sendo retirado da forma (ESTALEIRO SKIPPER).



Figura 8. Etapas de processo construtivo (veleiro Skipper 30)

Construir moldes rapidamente e a um custo baixo significa maior agilidade de se lançar um produto no mercado e maior facilidade para modificar ou construir um novo produto.

A construção do casco através de máquinas CNC leva ao estado da arte a fabricação barcos, e é uma tendência crescente entre os estaleiros. Obviamente, as dificuldades se encontram em projetar tais máquinas, já que o problema de controlar a usinagem em máquinas de grande dimensão não é trivial. Geralmente, estas fresadoras são construídas sob medida para o tamanho de casco que o estaleiro deseja construir.

Outra possibilidade é usinar blocos menores separadamente, unindo-os para formar o casco completo.

As grandes promessas do processo de usinagem em CNC são: redução de custos, precisão e rapidez na construção das peças. Além disso, alcançam-se resultados excelentes, fiéis ao modelo projetado, simulado e estudado em computador, evitando-se retrabalhos e interferências manuais na construção. Para se ter uma idéia destas máquinas, na Figura 9, à esquerda, está uma das maiores máquinas CNC da América para usinagem de embarcações. Suas dimensões são 25x9x3 metros, suportando cargas de até 59.000kg aproximadamente. Esta máquina possui movimentação em 5 eixos, com uma precisão de até 0.12mm. À direita, vêem-se outros exemplos: um casco sendo usinado, e a forma de um deck acabado.

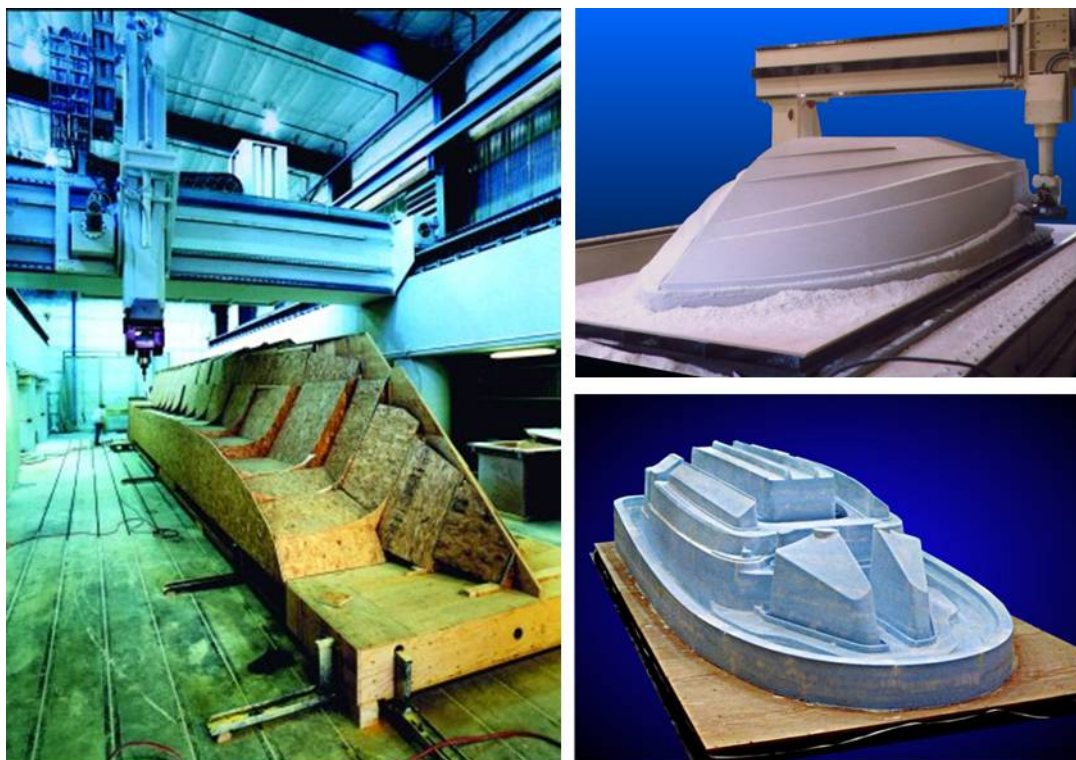


Figura 9. Exemplos de máquinas CNC para usinagem de embarcações

Hoje em dia, existem disponíveis softwares que auxiliam no processo de fabricação e usinagem em CNC, chamados de CAM (“Computer Aided Manufacturing”). No Apêndice deste trabalho, serão demonstrados alguns passos para se obter o produto

final a ser utilizado em uma máquina CNC: o código G da peça. Para tal, utilizaremos o software VisualMill 5.0, que a partir do bloco sólido de um casco, simula as etapas de desbaste e gera automaticamente o código G de uma máquina CNC.

1.3 Revisão Bibliográfica

Atualmente, vários artigos têm sido publicados apresentando trabalhos de desenvolvimento de ferramentas para o projeto otimizado de cascos. A maioria das referências utilizadas neste trabalho foi obtida de autores pertencentes ao projeto “Fantastic” (ver lista de referências, “Functional Design and Optimization of Ship Hull Forms”), patrocinado pelo fundo europeu “European Community Growth Project”. Neste, 14 grupos de pesquisa foram reunidos com o objetivo de aplicar as novas tecnologias de projeto com técnicas de otimização para o campo de desenho de barcos. Este trabalho estudará um dos blocos presentes no processo completo desenvolvido no Projeto “Fantastic” (HARRIES et al., 2003), já que este também emprega simulações em CFD, assunto que não será tratado aqui. Por outro lado, será apresentada uma metodologia simplificada, rápida e eficiente, igualmente indispensável no ciclo de projeto de veleiros.

Em (ANDRADE, 2001), ressalta-se a importância do projeto preliminar de uma embarcação e da parametrização das linhas que compõem o casco. Em seu trabalho, discute-se um novo modelo aplicado a um navio graneleiro, cujos passos e conceitos foram estendidos ao método desenvolvido e apresentado a seguir.

Narli e Sariöz (2003) abordam especificamente o processo de suavização (“fairing”) de curvas B-Splines. Como ponto de partida, a fórmula de Wigley (WIGLEY, 1934) é utilizada para criar uma curva 2D suave. Em seguida, uma distorção a esta curva é aplicada, e, para a curva distorcida, faz-se a otimização da curvatura. O algoritmo utilizado para o problema de otimização é o de busca direta não-linear de Hooke e Jeeves (1961). Como o método originalmente foi desenvolvido para problemas irrestritos, utilizam-se funções de penalização para transformar o problema com

restrições em um sem restrições. Porém, nem sempre o problema de suavização foi tratado como um problema de otimização. (NOWACKI, 1970) apresenta, por exemplo, uma técnica de mínimos quadrados para a suavização de splines.

Em relação aos trabalhos mais recentes encontrados na literatura, percebe-se claramente a complexidade de implementação e desenvolvimento de algoritmos que tratam a otimização de uma superfície, em vez de curvas. Em alguns destes trabalhos, são utilizadas superfícies NURBS, recomendadas para desenhar formas mais complexas como bulbos, por exemplo. A utilização de superfícies para a modelagem do casco é uma decisão que implicará na escolha do algoritmo de otimização. Trabalhos como (HARRIES et al., 2003) aplica algoritmos genéticos para a solução do problema, como o “Simulated Anealing”.

Os primeiros trabalhos que discutiam a importância da representação matemática de cascos, entretanto, aplicavam simplesmente polinômios como formulação das curvas. (REED et. al, 1970) é um exemplo. Neste, Reed e Nowacki destacam, todavia, as deficiências desta formulação (tendência a oscilar quando submetida a um alto número de restrições), mas somente tinham a ambição de obter uma aproximação razoável do barco. Em (NOWACKI, 1970) uma tabela com 32 autores distintos é apresentada, com trabalhos de 1915 a 1970, mostrando as propostas técnicas e funções utilizadas na representação de cascos.

Narli e Sariöz (2003) também apresentam um interessante histórico do progresso das ferramentas e algoritmos de otimização desde a década de 70. Como destaque, pode-se citar (SNAITH et al., 1972), que descreve que os procedimentos de suavização eram baseados em rotinas iterativas onde o projetista observava visualmente o desenho das curvas, e as modificava até que um resultado satisfatório de suavidade fosse alcançado; (SAPIDIS et al., 1990) que propõe um algoritmo automático de suavização para curvas B-Spline. O algoritmo é baseado em remover e recolocar nós na Spline. (HUANZONG et al., 1991) que propôs um método de suavização, minimizando a energia de deformação elástica de curvas que compunham a superfície de um casco; (MORETON et al., 1992) que utilizou técnicas não-lineares

de otimização para minimizar um funcional de suavidade, baseado na variação da curvatura; e (NOWACKI et al., 1994) que desenvolveram métodos para traçar curvas suaves restritas, cujo critério de suavidade é baseado na combinação linear do quadrado da segunda e terceira derivada.

Em relação à otimização do casco através do Método de Superfície de Resposta, (FASSARDI et al., 2006) explora a fundo o tema, fazendo ensaios com diversos parâmetros, ressaltando as vantagens do método frente a simulações custosas de CFD.

Já para o assunto curvas paramétricas, uma vasta literatura com diferentes abordagens pode ser encontrada. Trata-se de um tema muito estudado no campo de modelagem paramétrica, e assunto obrigatório na grade curricular de cursos de computação gráfica (GLEICHER, 2004). Apesar de que o emprego de curvas B-Splines já estar consagrado para representação de cascos, este trabalho apresenta a utilização e formulação completa de B-Splines uniformes, destacando suas vantagens e implicações deste modelo simplificado.

Como exemplo de projeto, e para efeito de demonstração dos resultados, será utilizado o veleiro projetado em (LARSSON et al., 1994), um dos principais livros didáticos de Engenharia para desenho de veleiros. Como referências para os métodos e fundamentos sobre otimização utilizou-se (HAFTKA et al., 1995), (VANDERPLAATS, 1984) e (MYERS et al., 1995). Este último trata especificamente do Método de Superfície de Resposta, abordando também os conceitos de Projeto de Experimentos (do inglês, “Design of Experiment”). Os guias de ajuda do aplicativo Matlab 6.5 também serviram como importante referência, pois, além de servir como manual para as funções e rotinas de otimização do software, também apresentam conceitos e comparações sobre cada método e respectivos algoritmos.

1.4 Motivação

A configuração de um veleiro é determinada por diversos fatores de acordo com o ambiente para o qual se deseja navegar. Velocidades desejadas, condições do mar, clima, desempenho da tripulação e orçamento são fatores que combinados com o projeto de cada componente determinam o sucesso em uma regata.

Se, por um lado, cada vez mais os velejadores exigem características específicas e requisitos complexos para uma embarcação vencedora, por outro, a indústria naval tenta buscar ferramentas de projetos para viabilizar o desenho de cascos que incorporem todos estes requisitos. O esforço e dinheiro para a campanha de uma regata geralmente não pode ser despendido em simulações de grande porte, ou em inúmeros protótipos para se alcançar uma configuração que atenda a todas as restrições ao mesmo tempo.

A pergunta que cabe é: como lidar com a necessidade de se projetar e construir com prazos curtos e recursos limitados novos produtos que possuam características customizadas? Bom seria se fosse possível concentrar o tempo e trabalho na concepção e especificação de um projeto e deixar que o computador faça todo o trabalho operacional, fornecendo como resultado uma solução de projeto que atenda ao que foi solicitado.

A proposta deste trabalho é propor uma ferramenta de baixo custo capaz de automatizar o processo de criação ou modificação (“redesign”) de cascos de veleiros através de uma modelagem paramétrica. O que será abordado é o estudo e implementação de um software em ambiente Matlab, que receberá parâmetros de alto nível, como razões adimensionais, coeficientes de forma e restrições, e gera tridimensionalmente um casco que cumpre com os parâmetros solicitados.

A utilidade da ferramenta se faz presente tanto provendo um desenho preliminar ao projetista, um ponto de partida para seu projeto, quanto para a modificação de uma embarcação existente que servirá de referência para um novo projeto. Assim como

em qualquer projeto que envolve grandes sistemas, o projeto de um veleiro é normalmente dividido em fases, para permitir a manipulação de um grande número de variáveis e especificações. Estas etapas são usualmente conhecidas como: Projeto Preliminar; Projeto de Contrato e Projeto de Construção ou Detalhamento. O projeto preliminar envolve as características principais, as quais têm maior influência nas propriedades que traduzem as principais aspirações dos clientes, e que também estão vinculadas a aspectos econômicos e de segurança (ANDRADE, 2001).

Ainda não existe disponível uma ferramenta semelhante capaz de conciliar ferramentas de Engenharia com as necessidades comerciais de um produto, quer dizer, tudo o que o cliente buscará na sua compra. Porém, o que se procura discutir será a importância de se dispor de um recurso como este para que o fator tempo não seja a restrição desrespeitada do projeto. Cabe ressaltar que a etapa de projeto preliminar é a fase mais emblemática e criativa de todo o processo, quando se realizam as atividades de concepção e avaliação, a fim de gerar, modificar e otimizar alternativas de solução, que se encaixam no espaço de soluções viáveis do projeto, culminando na solução mais eficiente segundo os critérios adotados.

1.5 Objetivos

Esta pesquisa tem como objetivo formular o problema de desenho de um casco como um problema de otimização, e a partir disto, desenvolver uma ferramenta na qual se implementa uma metodologia para síntese de cascos e sua otimização paramétrica.

Buscar-se-á abordar o projeto do casco de uma maneira em que os recursos sejam empregados na concepção do projeto e que a parte iterativa seja feita de forma automática pelo computador. Outro principal objetivo do processo automático de suavização é a minimização da intervenção humana.

Os pontos explorados são:

- 1) Aplicação de técnicas de otimização para síntese de curvas B-Splines, tendo como critério de mérito a suavidade da curva.
- 2) Proposição de uma metodologia para a parametrização de cascos de veleiros a partir das linhas principais de um desenho, geradas pelos métodos de otimização.
- 3) Aplicação da Técnica de Superfície de Resposta para a otimização da resistência residual, tendo como variáveis de projeto os parâmetros geométricos do casco.
- 4) Proposição de um projeto integrado abrangendo desde a entrada de parâmetros de projeto de alto nível de uma embarcação até a geração do código de programa para máquinas CNC.

1.6 Organização do texto

Este trabalho está dividido em oito capítulos. O primeiro é a introdução subdividida em objetivos e revisão bibliográfica, onde são apresentadas as bibliografias utilizadas tanto para a parte de formulação matemática de curvas quanto das recentes publicações no campo de otimização de cascos de veleiros.

O segundo, terceiro e quarto capítulos fazem parte da fundamentação teórica do trabalho. O segundo capítulo apresenta os tipos mais usuais de curvas para representação paramétrica de objetos, e uma discussão sobre vantagens e desvantagens da utilização de cada um, chegando-se na conclusão final do tipo de formulação que será adotada. O terceiro capítulo discute os conceitos de otimização envolvidos no trabalho, bem como, apresenta toda a formulação do problema abordado, percorrendo a função objetivo, restrições, algoritmos utilizados, e por fim, a técnica de otimização por superfície de resposta. O quarto capítulo apresenta a notação que será utilizada no trabalho para as grandezas físicas de um veleiro, seu plano de linhas, e o modelo hidrodinâmico utilizado para a otimização paramétrica do casco.

O quinto capítulo descreve a metodologia de parametrização adotada, explicando a seqüência de otimizações de curvas para o exemplo de veleiro que é desenhado. No capítulo seis são apresentados os resultados obtidos da modelagem, síntese e otimização do casco. A discussão destes resultados é feita no capítulo sete. No capítulo oitavo apresentam-se as conclusões deste trabalho, bem como possíveis propostas de desenvolvimentos futuros. E, por fim, o apêndice demonstra como explorar os recursos de CAD e CAM para usinagem em CNC do casco.

2 CURVAS PARAMÉTRICAS

Intuitivamente, o conceito de curva nos passa a idéia de um objeto, uma linha contínua no espaço. Rigorosamente, segundo (GLEICHER, 2004), uma curva paramétrica é:

Um mapeamento contínuo de um espaço unidimensional para um espaço n -dimensional.

Neste trabalho, iremos utilizar representações de curvas através de equações paramétricas, onde as curvas são definidas através de uma função $f(t)$. Isso porque, para modelagens geométricas, a forma de um objeto deve ser representada unicamente através de números e equações. Somente assim é possível utilizar um modelo computacional para a sua representação, e reproduzi-lo infinitamente ou armazená-lo em um meio permanente, como um disco rígido.

Desde meados do século XX, desenhar curvas suaves para modelar uma topologia específica de interesse era considerado um esforço árduo e tedioso. Em 1959, Paul de Fagnet de Casteljaou, um funcionário da Citroën, deu o primeiro passo para a obtenção de curvas e superfícies suaves combinando e iterando malhas de polígonos para modelagem de chassis. Concomitantemente, agora na Renault, um empregado chamado Pierre Bézier também produziria curvas para o mesmo fim, que atualmente são chamadas de curvas de Bézier. A curva de Bézier somente ficou formalmente conhecida após a publicação do trabalho “Emploi des Machines a Commande Numerique” em 1970 pelo próprio Pierre Bézier.

A escolha do tipo de curva está relacionada ao objeto que se deseja definir, em outras palavras, à sua aplicação. Basicamente, traçar uma curva que passe por um conjunto de pontos caracteriza um problema de interpolação, ao passo que definir uma curva que se aproxime de um determinado conjunto de pontos é um problema de aproximação.

Polinômios foram uma das primeiras escolhas para a solução de ambos os problemas. Todavia, sua principal desvantagem está na sua globalidade, ou seja, ao alterarmos o valor de qualquer um dos pontos, toda a curva é afetada. Isso implica no número reduzido de aplicações que um polinômio pode ter em modelagens de curvas, restrito a pequenos intervalos ou a um número baixo de pontos de controle.

Nos parágrafos a seguir, serão tratadas as principais curvas utilizadas em computação gráfica como Bézier, Hermite, Splines, e B-Splines, com o objetivo de apresentar o desenvolvimento realizado para se chegar à formulação matemática da modelagem de curva escolhida, as curvas B-Splines.

2.1 Curvas paramétricas

Considere a curva da Figura 10. Devido a sua forma, descrever esta curva na forma de uma função $y = f(x)$ pode ser bastante difícil (YAMAGUCHI, 1988). No entanto, pode-se descrever as coordenadas dos pontos da curva como funções $x = f(t)$ e $y = g(t)$. Estas duas equações são apropriadas para descrever curvas e nos leva a definição de curva paramétrica.

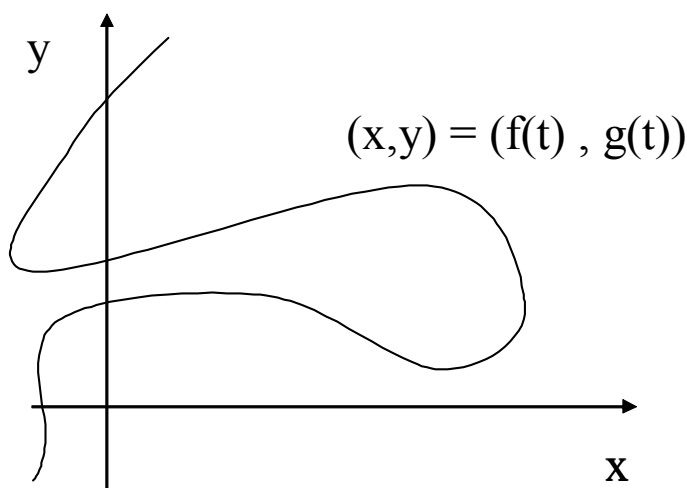


Figura 10. Exemplo de curva paramétrica

Suponha que x e y são funções de uma terceira variável t (parâmetro), dadas pelas equações (paramétricas):

$$x=f(t) \quad y=g(t) \quad (1)$$

Cada valor de t determina um ponto (x,y) no plano. Conforme variamos t , $(x,y)=(f(t),g(t))$ varia, traçando a curva, chamada de curva paramétrica.

Curvas paramétricas são a representação mais comum para curvas em computação gráfica e são muito úteis em processos de modelagem. Curvas paramétricas de segunda ordem apresentam utilização restrita, devido à continuidade (a ser definida a seguir). Curvas paramétricas de terceira ordem ou cúbicas, no entanto, são as curvas mais populares.

Além da formulação paramétrica, existem ainda dois tipos de formulações para as curvas:

- Formulação explícita: função na forma $y = f(x)$
- Formulação implícita: representação do tipo $f(x, y) = 0$

No entanto, somente a formulação paramétrica nos permite resolver, simultaneamente, problemas como:

a) definir uma curva através de seus pontos e de suas derivadas nestes pontos

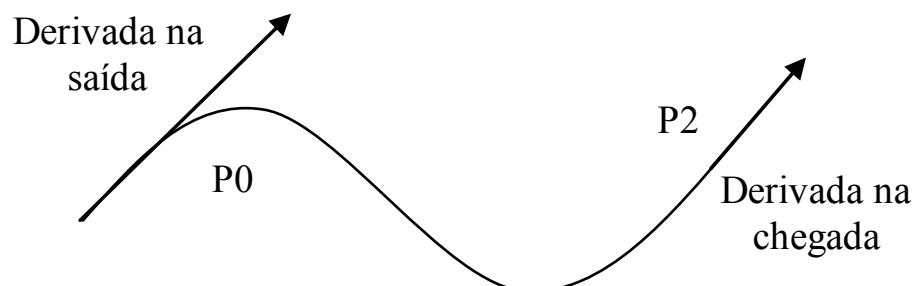


Figura 11. Exemplo de curva com pontos e derivadas definidas

b) criar curvas fechadas ou com laços (múltiplos valores de y dado um valor de x)

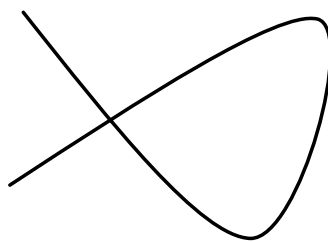


Figura 12. Exemplo de curva com laço

c) obter uma curva suave que passe por um conjunto de pontos

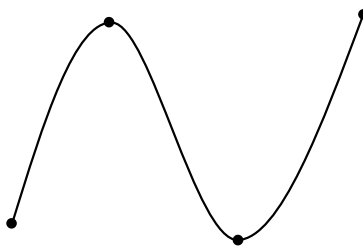


Figura 13. Exemplo de curva que passa por pontos dados

Além disso, a formulação explícita impede descreverem-se curvas com tangentes verticais. Do mesmo modo, a formulação implícita pode fornecer mais soluções do que se deseja, necessitando de mais constantes para definir a solução requerida. Outro ponto que se pode salientar é que, quando é necessário unir duas ou mais curvas, utilizar esta formulação pode se tornar a solução do problema em algo muito complicado.

Estes conceitos serão discutidos mais adiante para demonstrar as formulações dos tipos de curvas que seguem.

2.2 Formulação matricial e continuidade de curvas paramétricas

A forma geral de um segmento de uma curva polinomial (de terceiro grau ou cúbica, por exemplo) pode ser expressa segundo a equação abaixo.

$$\begin{aligned} x(t) &= a_x t^3 + b_x t^2 + c_x t + d_x \\ y(t) &= a_y t^3 + b_y t^2 + c_y t + d_y \end{aligned} \quad (2)$$

A variável paramétrica t é restrita ao intervalo $[0,1]$. Utilizando a forma matricial, podemos escrever:

$$\mathbf{T} = \begin{bmatrix} t^3 \\ t^2 \\ t \\ 1 \end{bmatrix} \quad \mathbf{C} = \begin{bmatrix} a_x & b_x & c_x & d_x \\ a_y & b_y & c_y & d_y \end{bmatrix} \quad (3)$$

Desta forma, um segmento de curva pode ser escrito na forma $\mathbf{q}(t) = (x(t), y(t)) = \mathbf{CT}$.

Com isso exposto, se introduzirá agora o conceito de continuidade de uma curva. Para unir segmentos de curvas paramétricas deve-se assegurar que, nos pontos de união (“joining points”), as derivadas de cada segmento sejam iguais. Desta maneira, é possível garantir a continuidade e a suavidade da curva.

A continuidade de uma curva pode ser calculada da seguinte maneira:

$$\frac{\partial \mathbf{q}(t)}{\partial t} = \left(\frac{\partial x(t)}{\partial t}, \frac{\partial y(t)}{\partial t} \right) = \frac{\partial (\mathbf{CT})}{\partial t} = \mathbf{C} \frac{\partial \mathbf{T}}{\partial t} \quad \text{onde, } \frac{\partial \mathbf{T}}{\partial t} = \begin{bmatrix} 3t^2 \\ 2t \\ 1 \end{bmatrix} \quad (4)$$

É possível definir vários tipos de continuidade:

Continuidade geométrica $G^0 \rightarrow$ as curvas se unem.

Continuidade geométrica $G^1 \rightarrow$ as curvas se unem, com suas tangentes apontando na mesma direção.

Continuidade paramétrica $C^1 \rightarrow$ as curvas se unem, com suas tangentes apontando na mesma direção e com mesma magnitude (em outras palavras, a primeira derivada nos pontos de união são iguais).

Continuidade paramétrica $C^n \rightarrow$ as curvas se unem com a mesma n -ésima derivada nos pontos de união.



Figura 14. Exemplos de Continuidade de curvas: continuidade G^0 , G^1 e C^1

2.3 Curvas de Hermite

A formulação da curva de Hermite surgiu da necessidade de solucionar problemas de definição de uma curva dados seus pontos extremos e as derivadas nestes pontos. Geralmente, são empregadas para interpolar de forma suave pontos, como por exemplo, em animações ou controle de câmera de filmagens. Entender sua formulação matemática (simples, porém poderosa) é o primeiro passo para o completo entendimento da família de curvas Splines e demais polinômios de ajuste de curvas.

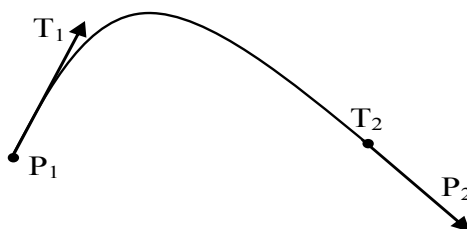


Figura 15. Exemplo de uma curva de Hermite

A forma polinomial de Hermite é definida por dois pontos finais e dois vetores tangentes nestes pontos. Hermite propõe um interpolador local, controlado por 4 fatores a cada intervalo de 2 pontos: P_1 e P_2 , os pontos iniciais e finais, e os vetores tangentes T_1 e T_2 . Estas curvas de ponderação de Hermite são definidas por polinômios de terceira ordem com a forma geral:

$$\begin{aligned} x(t) &= P_x = a_x t^3 + b_x t^2 + c_x t + d_x = \mathbf{C}_x \mathbf{T} = \mathbf{G}_x \mathbf{M}_H \mathbf{T} \\ y(t) &= P_y = a_y t^3 + b_y t^2 + c_y t + d_y = \mathbf{C}_y \mathbf{T} = \mathbf{G}_y \mathbf{M}_H \mathbf{T} \end{aligned} \quad (5)$$

Onde \mathbf{C} é a matriz com os coeficientes, \mathbf{G} é a chamada matriz geométrica e \mathbf{M}_H é a Matriz de Base de Hermite. Para solucionar a matriz geométrica de Hermite, ou seja, determinar a matriz de base é necessário determinar os fatores de a , b , c e d para cada uma das curvas de ponderação P_1 , P_2 , T_1 e T_2 .

Far-se-á a seguir apenas o desenvolvimento para a função $x(t)$, já que para $y(t)$ o raciocínio é análogo. A determinação dos fatores da Equação (5) é feita da seguinte forma:

Curva P_1 : para $t=0$,

$$P_{1x} = a_x 0^3 + b_x 0^2 + c_x 0 + d_x = [0 \ 0 \ 0 \ 1] \cdot \begin{bmatrix} a_x \\ b_x \\ c_x \\ d_x \end{bmatrix} \quad (6)$$

Curva P_2 : para $t=1$,

$$P_{2x} = a_x 1^3 + b_x 1^2 + c_x 1 + d_x = [1 \ 1 \ 1 \ 1] \cdot \begin{bmatrix} a_x \\ b_x \\ c_x \\ d_x \end{bmatrix} \quad (7)$$

As tangentes em P_1 e P_2 podem ser obtidas através da derivada da função em cada um dos pontos:

$$\frac{d}{dt} x(t) = 3a_x t^2 + 2b_x t + c_x \quad (8)$$

Curva T_1 : para $t=0$,

$$T1_x = a_x 3 \cdot 0^2 + b_x 2 \cdot 0 + c_x = [0 \ 0 \ 1 \ 0] \cdot \begin{bmatrix} a_x \\ b_x \\ c_x \\ d_x \end{bmatrix} \quad (9)$$

Curva T_2 : para $t=1$,

$$T_{2x} = a_x 3 \cdot 1^2 + b_x 2 \cdot 1 + c_x = [3 \quad 2 \quad 1 \quad 0] \cdot \begin{bmatrix} a_x \\ b_x \\ c_x \\ d_x \end{bmatrix} \quad (10)$$

Das Equações (6), (7), (9) e (10) obtém-se a seguinte relação:

$$\begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 3 & 2 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} a_x & a_y \\ b_x & b_y \\ c_x & c_y \\ d_x & d_y \end{bmatrix} = \begin{bmatrix} P_{1x} & P_{1y} \\ P_{2x} & P_{2y} \\ T_{1x} & T_{1y} \\ T_{2x} & T_{2y} \end{bmatrix} \quad (11)$$

A Matriz de Base de Hermite é então calculada a partir da relação abaixo:

$$\begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 3 & 2 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} a_x & a_y \\ b_x & b_y \\ c_x & c_y \\ d_x & d_y \end{bmatrix} = \begin{bmatrix} P_{1x} & P_{1y} \\ P_{2x} & P_{2y} \\ T_{1x} & T_{1y} \\ T_{2x} & T_{2y} \end{bmatrix} \quad (12)$$

Para que a equação acima fique verdadeira, é necessário que \mathbf{M}_H seja a matriz inversa da matriz do último fator, anulando o produto com a matriz identidade resultante. Logo, tem-se que a matriz de Hermite é definida por (FOLEY, et al, 1993):

$$\mathbf{M}_H = \begin{bmatrix} 0 & 1 & 0 & 3 \\ 0 & 1 & 0 & 2 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \end{bmatrix}^{-1} = \begin{bmatrix} 2 & -3 & 0 & 1 \\ -2 & 3 & 0 & 0 \\ 1 & -2 & 1 & 0 \\ 1 & -1 & 0 & 0 \end{bmatrix} \quad (13)$$

Finalmente, é possível determinar a função $x(t)$ através da Equação (5). Na sua forma desenvolvida, tem-se:

$$\begin{aligned} x(t) &= P_{1x}(2t^3 - 3t^2 + 1) + P_{2x}(-2t^3 + 3t^2) + T_{1x}(t^3 - 2t^2 + t) + T_{2x}(t^3 - t^2) \\ y(t) &= P_{1y}(2t^3 - 3t^2 + 1) + P_{2y}(-2t^3 + 3t^2) + T_{1y}(t^3 - 2t^2 + t) + T_{2y}(t^3 - t^2) \end{aligned} \quad (14)$$

Na Figura 16, vê-se a geometria de composição da função de base de Hermite, ou seja, as funções polinomiais determinados pela matriz \mathbf{M}_H .

:

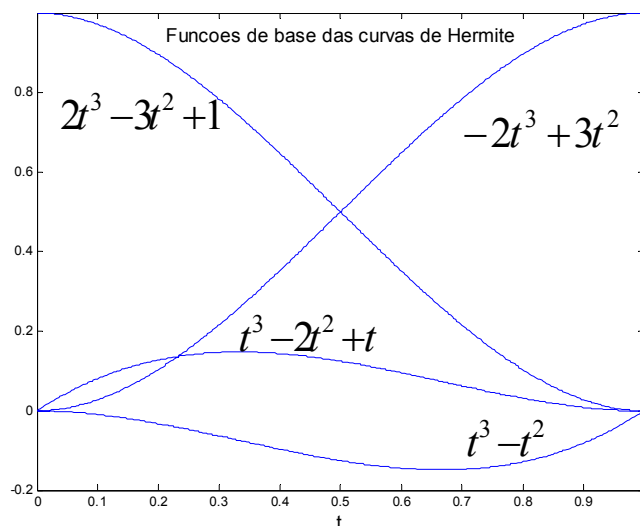


Figura 16. Funções de base de Hermite

Pode-se perceber que para um dado instante t , a curva de Hermite é resultado da soma de todos os fatores de controle (P_1 , P_2 , T_1 e T_2), ponderados pelas curvas de geometria de Hermite.

2.4 Curvas de Bézier

A curva de Bézier foi desenvolvida por Pierre Bézier e ganhou grande importância quando estudada por Paul de Casteljau no fim da década de 1950, com o fim de criar um método eficaz para a modelagem de carros. Baseado nos princípios de Hermite, durante seus trabalhos em projetos de automóveis para a empresa francesa Renault, Bézier acrescentou dois pontos de controle às constantes de Hermite determinando os dois vetores tangentes nos pontos inicial e final.

Naquela época, Bézier trabalhava como engenheiro e concebeu uma nova formulação de curvas para representar carros em computador. De fato, a curva de Bézier é uma forma aproximada de curva polinomial, especialmente concebida para ser controlada de forma conveniente. Na sua forma bidimensional, a curva de Bézier

é hoje base para vários programas gráficos como Adobe Illustrator e Corel Draw. Algumas fontes também, como TrueType e Post Script Type 1 também são armazenadas como curvas de Bézier.

As formas mais importantes das curvas de Bézier são as quadráticas e cúbicas. Curvas de grau maior implicam em um custo mais elevado de cálculo e processamento. O método atualmente mais utilizado para avaliar as curvas de Bézier é o algoritmo de De Casteljaou (FARIN, 1990), que não será tratado neste trabalho.

Dada esta introdução sobre a importância desta formulação até os dias de hoje, seguiremos com sua formulação que é relevante para entendermos os conceitos de curvas B-Splines que veremos mais adiante. Assim como a curva de Hermite, a curva de Bézier também possui pontos de controle, porém quatro (P_1, P_2, P_3 e P_4) ao invés de dois, como na fórmula de Hermite, e dois vetores tangentes (R_1 , e R_4).

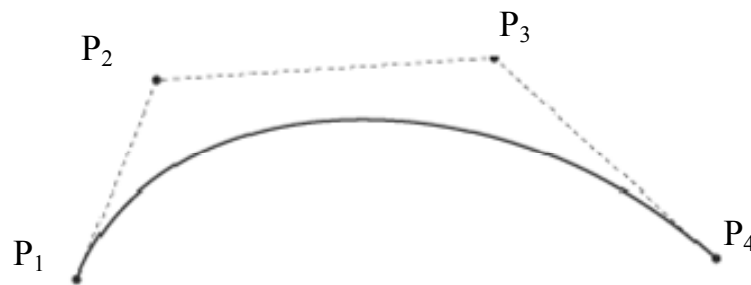


Figura 17. Exemplo de curva de Bézier

Dado que P_1 e P_4 são os “end-points” da curva, e os pontos P_2 e P_3 os pontos intermediários, os vetores tangentes determinados como segue:

$$\begin{aligned} R_1 &= 3(P_2 - P_1) \\ R_4 &= 3(P_4 - P_3) \end{aligned} \tag{15}$$

Assim como demonstrado na seção 2.3 para Hermite, a curva de Bézier também é composta pela ponderação de polinômios, denominados polinômios de Bernstein (Figura 18).

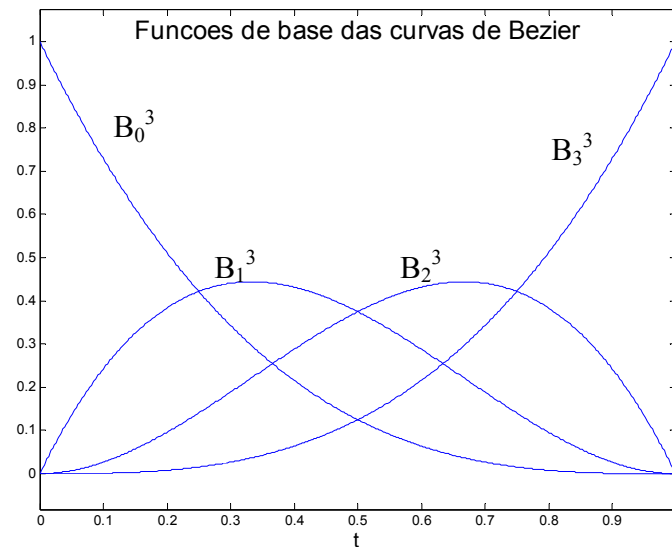


Figura 18. Polinômios de Bernstein: funções de base para a curva de Bézier

Logo para o caso de terceira ordem, o polinômio que descreve a curva de Bézier é:

$$x(t) = P_x = a_x t^3 + b_x t^2 + c_x t + d_x = \mathbf{C}_x \mathbf{T} = \mathbf{G}_x \mathbf{M}_B \mathbf{T} \quad (16)$$

onde,

$$\mathbf{G}_x = [P_{1x} \quad P_{2x} \quad P_{3x} \quad P_{4x}] \quad (17)$$

e

$$\mathbf{M}_B = \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & 6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \quad (18)$$

A Figura 18 demonstra que cada ponto de controle possui influência sobre a curva. Para intervalos maiores que quatro pontos, a função de Bézier expande o conceito de ponderação e assume caráter polinomial, ou seja, controle não-local, onde todos os pontos de controle participam da formação ponderada da curva.

Outra formulação também muito utilizada, e bem conhecida na literatura que trata do assunto, é a curva de Bézier paramétrica $b(t)$ considerada como a somatória dos polinômios de Bernstein multiplicados pelos pontos de controle b_i .

$$b(t) = \sum_{i=0}^n B_i^n(t) \cdot b_i \quad (19)$$

onde n é o grau da curva de Bézier. As funções $B_i^n(t)$ são os polinômios de Bernstein, e são definidos por (FARIN, 1990):

$$B_i^n(t) = \binom{n}{i} t^i (1-t)^{n-i}, \text{ onde } \binom{n}{i} = \begin{cases} \frac{n!}{i!(n-i)!} & \text{se } 0 \leq i \leq n \\ 0 & \text{caso contrário} \end{cases} \quad (20)$$

Para o caso mais comum, onde $n = 3$ tem-se:

$$\begin{aligned} B_0^3(t) &= (1-t)^3 = -t^3 + 3t^2 - 3t + 1 \\ B_1^3(t) &= 3t \cdot (1-t)^2 = 3t^3 - 6t^2 + 3t \\ B_2^3(t) &= 3t^2 \cdot (1-t) = -3t^3 + 3t^2 \\ B_3^3(t) &= t^3 \end{aligned} \quad (21)$$

Repare que os coeficientes de t dos polinômios de Bernstein para o grau 3 são os mesmos coeficientes da matriz \mathbf{M}_B da Equação (18). Cada polinômio $B_i^n(t)$ tem um único valor de máximo, que ocorre em $t=i/n$. Isso causa o que é chamado de controle “pseudo-local”, ou seja, ao movermos um dos pontos de controle, a forma da curva será afetada mais fortemente na região próxima a este ponto, todavia, toda a curva é afetada (PARENTE, 2000). Essa característica muitas vezes não é desejada, o que não ocorre nas curvas B-Splines que veremos a seguir.

Os polinômios de Bernstein possuem algumas propriedades que são passadas às curvas de Bézier. A principal delas é que qualquer polinômio é sempre positivo no intervalo de $[0,1]$ e que:

$$B_0^n(0) = 1 \quad (22)$$

$$B_n^n(1) = 1$$

Além disso, pela Figura 19 pode-se ver que a somatória de todos os polinômios resulta na unidade:

$$\sum_{i=0}^n B_i^n(t) = 1 \quad (23)$$

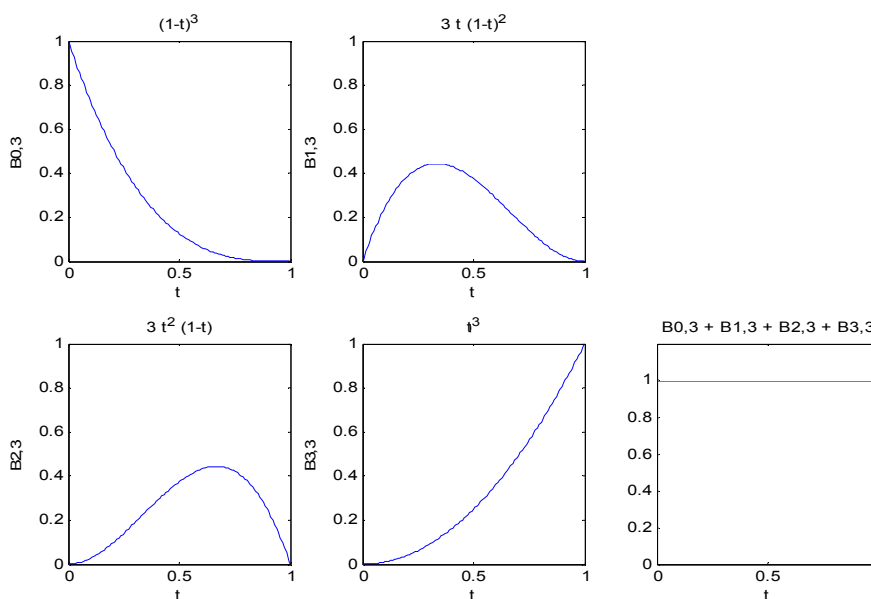


Figura 19. Polinômios cúbicos de Bernstein

Com estas propriedades, podemos chegar a uma propriedade da curva de Bézier, denominada de “convex hull”, que diz que a curva de Bézier está completamente dentro do maior polígono convexo formado pelos pontos de controle P_1 a P_4 . Qualquer transformação de escala, translação ou rotação da curva é possível ser feita apenas realizando as transformações nos pontos de controle. Esta propriedade também é utilizada para a checagem de interferências entre curvas, cujo exemplo de aplicação pode-se citar o cálculo de interferência de braços robóticos. Se as trajetórias de dois braços são descritas por curvas de Bézier, em vez de se realizar o cálculo de possíveis intersecções, basta fazer o teste do polígono convexo, que é muito mais simples (FARIN, 1990).

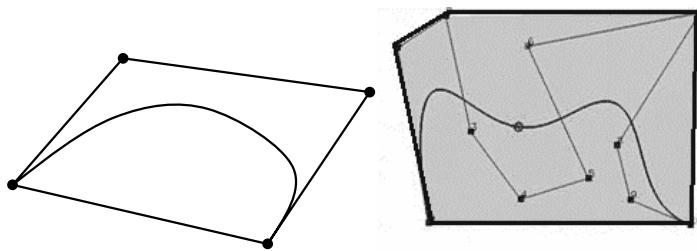


Figura 20. Curvas de Bézier contida no fecho convexo dos pontos de controle

Se necessitarmos unir duas curvas de Bézier com os seguintes pontos de controle:

Curva 1: P_1, P_2, P_3 , e P_4

Curva 2: P_4, P_5, P_6 e P_7

Então, devemos garantir a continuidade paramétrica C^1 e a continuidade geométrica G^1 , já que tendo o ponto P_4 em comum, garantimos já a continuidade C^0 e G^0 .

C^1 é dada por: $P_3 - P_4 = P_4 - P_5$

G^1 é dada por: $P_3 - P_4 = k(P_4 - P_5)$, onde $k > 0$

Se garantirmos que estas relações são válidas, então se tem uma receita para unir os segmentos de curvas.

2.5 Curvas B-Splines

O termo Spline é designado na matemática para definir curvas especiais formadas por segmentos de polinômios. O nome vem de uma prática comum de se utilizar réguas de madeira ou metal, longas e estreitas, juntamente com pesos cuidadosamente escolhidos para determinar curvas de barcos, carros ou aeroplanos. Movendo-se estes pesos ao longo da curva, buscava-se alterar a forma da curva com o intuito de obter uma maior suavidade. Desta propriedade, deriva-se um critério de suavização físico para as splines. O projetista procurava posicionar os pesos de forma que a energia de deformação da spline elástica fosse a mínima possível. Este

conceito de suavidade é a forma primitiva dos conceitos de otimização da energia abordados nos capítulos seguintes.

De fato, as curvas B-Splines têm uma longa história na computação gráfica. Antes da informática, os cálculos numéricos eram feitos na ponta do lápis, e a utilização de polinômios era preferida por ser mais fácil de trabalhar. Com o advento dos computadores, as curvas Splines ganharam importância, primeiramente empregadas na substituição dos polinômios em interpolação de pontos, e em seguida, em outras ferramentas gráficas para se construir curvas suaves e flexíveis.

Assim, atualmente no ramo da computação gráfica, a interpolação com Splines é preferida à com polinômios por proporcionar um mesmo resultado requerendo um menor grau das funções. Até as curvas de Bézier, quando modelam formas complexas, requerem possuir um grau muito elevado, por vezes, proibitivo (acima de grau 10) (FARIN, 1990). Em relação à aproximação de curvas, as Splines se adequam perfeitamente a problemas de formas complexas, e seu modo de manipulação fez com que estas curvas se popularizassem.

As B-Splines são oriundas do modelo de Spline Natural, porém implementando um controle onde os pontos de controle não são interpolados pela curva, ao contrário do que acontece na Spline Natural. Isso implica que para uma curva de Spline Natural, ao modificarmos um dos pontos de controle, toda a curva é afetada. Apesar disso, a curva Spline Natural possui uma suavidade maior que as curvas de Hermite e Bézier.

Já as B-Splines são curvas que apresentam o controle tipo local, ou seja, ao alterarmos um ponto de controle, apenas um pequeno segmento da curva é afetado, e o restante permanece inalterado. A letra B de B-Spline vem de “base”. As B-Splines utilizam funções de base diferentes dos polinômios de Bernstein. A curva com o grau mais comum é a B-Spline cúbica, pois possui a continuidade C^2 , muito útil para modelagem de curvas no espaço.

Assim como nas curvas de Bézier, as funções de base para as B-Spline são utilizadas como ponderação, porém há duas propriedades interessantes que as diferem: seu domínio é subdividido em nós e suas funções de base podem não assumir valor zero ao longo de todo intervalo. Isso resultará na propriedade de controle local, citada anteriormente. Na verdade, curvas de Bézier são um caso especial de curvas B-Spline, e é possível obter uma matriz de transformação que transforma uma curva em outra.

Outra propriedade interessante de curvas B-Spline é que elas podem apresentar-se de três maneiras: aberta, fechada e “clamped” (do inglês, amarrada) (YAMAGUCHI, 1988). Em caso de que a estrutura do vetor de nós não assuma nenhuma função particular, a curva gerada não irá tocar os pontos das extremidades da linha de controle da curva. Esta curva é chamada de curva aberta. Se alterarmos a multiplicidade dos pontos iniciais e finais, impondo que a curva inicie ou termine neles, teremos a chamada “clamped curve”. Se repetirmos alguns nós e pontos de controle, podemos formar uma curva fechada, que começa e termina no mesmo ponto.

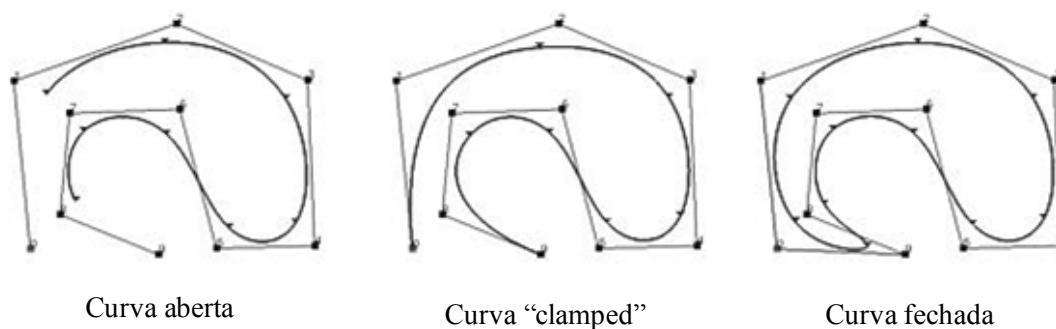


Figura 21. Classificação das curvas em aberta, “clamped” e fechada

2.5.1 Funções de base

Seja t a variável paramétrica que define uma curva B-Spline $s(t)$ de grau n com L segmentos, segundo a expressão (FARIN, 1990):

$$s(t) = \sum_{i=0}^{L+n-1} P_i N_{i,n}(t) \quad (24)$$

Nesta equação, P_i são os pontos de controle e N são as funções de base da curva, que podem ser calculadas recursivamente segundo as expressões abaixo:

$$N_{i,0} = \begin{cases} 1 & \text{se } t_i \leq t \leq t_{i+1} \\ 0 & \text{nos outros casos} \end{cases} \quad (25)$$

$$N_{i,n} = \frac{t - t_{i-1}}{t_{i+n-1} - t_{i-1}} N_{i,n-1}(t) + \frac{t_{i+n} - t}{t_{i+n} - t_i} N_{i+1,n-1}(t)$$

A coordenada t_i é chamada de nó (pontos de junção entre os segmentos da curva). A Equação (25) é usualmente conhecida como Fórmula recursiva de Cox-de Boor. (YAMAGUCHI, 1988) apresenta em detalhes o cálculo recursivo destas funções de base.

Destacam-se algumas propriedades de $N_{i,n}(t)$:

- **Não negativa:** $N_{i,n}(t) \geq 0$, para qualquer i , n e t .
- **Partição da unidade:** a somatória de todas as funções no intervalo $[t_i, t_{i+1}]$ é igual a 1, $\sum N_{i,n}(t) = 1$
- **Controle local:** $N_{i,n}(t) = 0$, se $t \notin [t_i, t_{i+n+1}]$
- **Diferenciabilidade:** todas as derivadas de $N_{i,n}(t)$ existem no interior de um intervalo de nós, onde $N_{i,n}(t)$ é polinomial.
- **Ponto de máximo:** $N_{i,n}(t)$ tem apenas um ponto de máximo, com exceção para $n=0$.
- **Continuidade em um nó com multiplicidade k :** a continuidade na função $N_{i,n}(t)$ em um nó com multiplicidade k é C^{n-k} , Isso quer dizer, para reduzirmos a continuidade de um nó, basta aumentarmos sua multiplicidade.

2.5.2 Curvas B-Splines não-uniformes e B-Splines cúbicas

Se uma curva B-Spline possui seu vetor de nós igualmente espaçados, ou seja, $t_{i+1} - t_i = cte$, esta curva é chamada de uniforme. Caso contrário é chamada de B-Spline não-uniforme. Rigorosamente, o vetor de nós é um conjunto selecionado de pontos finais de subintervalos t_i que definem o intervalo onde uma dada função de base é ativa. Estas curvas são de suma importância principalmente na forma de NURBS (“Non-Uniform Rational B-Splines”), ou seja, na sua forma racional. Neste trabalho serão utilizadas somente curvas B-Splines uniformes, posto que as curvas traçadas não requerem a flexibilidade das curvas não-uniformes.

Conforme falado no início deste capítulo, as curvas de grau 3 têm grande importância, já que possuem continuidade C^2 , o que significa que sua segunda derivada ainda será uma função (uma reta), e esta utilidade será justificada a seguir.

Para avaliar a curva $s(t)$ de grau 3, faz-se necessário o cálculo das funções de base $N_{i,0}$, $N_{i,1}$, $N_{i,2}$ e $N_{i,3}$. A expressão de $s(t)$ então fica:

$$s(t) = P_{i-2}N_{i-2,3} + P_{i-1}N_{i-1,3} + P_iN_{i,3} + P_{i+1}N_{i+1,3} \quad (26)$$

2.5.3 B-Splines uniformes

Assim como as curvas de Bézier são um caso especial de B-Splines, em aplicações com B-Splines de aproximação, podem-se utilizar B-Splines uniformes, que são curvas B-Splines com seus nós espaçados em intervalos iguais. Nestas curvas, o primeiro segmento é definido pelos quatro primeiros pontos de controle e o último segmento de curva é definido pelos 4 últimos pontos de controle. Assim, cada ponto de controle afeta 4 segmentos de curva.

A principal vantagem de uma B-Spline uniforme é sua formulação mais simplificada, no entanto, se paga o preço de uma menor flexibilidade (perdem-se graus de liberdade), a qual, para muitas aplicações, pode não ser um requisito.

Para B-Splines uniformes tem-se que:

$$t_{i+1} - t_i = d \quad (27)$$

Se fizermos uma alteração de índices na função de base N de forma que deixemos os índices apenas em função de t_i , teremos:

$$N_{i,n} = \frac{t - t_i}{t_{i+n} - t_i} N_{i,n-1}(t) + \frac{t_{i+n+1} - t}{t_{i+n+1} - t_{i+1}} N_{i+1,n-1}(t) \quad (28)$$

$$N_{i,n} = \frac{t - t_i}{nd} N_{i,n-1}(t) + \frac{t_i + (n+1)d - t}{nd} N_{i+1,n-1}(t)$$

Desta maneira, os polinômios da função de base da B-Spline se tornam mais simples de se trabalhar. A Tabela 1 mostra o desenvolvimento de todas as funções de base necessárias para o cálculo da B-Spline cúbica uniforme. Na primeira linha, estão os quatro intervalos da função de base $[t_i, t_i+d]$, $[t_i+d, t_i+2d]$, $[t_i+2d, t_i+3d]$, $[t_i+3d, t_i+4d]$. Nas linhas abaixo, estão todos os polinômios que formarão a última função de base cúbica $N_{i,3}$. Nas duas últimas linhas, demonstra-se a mudança de variável de t para u , simplificando a formulação e chegando nas parcelas que compõem a Equação (29). Fora destes intervalos ($t < t_i$, ou $t > t_i + 4d$), o valor da função de base é zero.

Assim, substituindo o valor de $N_{i,3}(u)$ na Equação (26), a equação completa da B-Spline Uniforme fica (NAKAMURA, 1996):

$$s(u) = \frac{(1-u)^3}{6} P_{i-3} + \frac{3u^3 - 6u^2 + 4}{6} P_{i-2} + \frac{-3u^3 + 3u^2 + 3u + 1}{6} P_{i-1} + \frac{u^3}{6} P_i \quad (29)$$

A Figura 22 mostra a curva das funções de base de um B-Spline cúbica. Atente ao fato de que assim como para as curvas de Hermite e Bézier, existe uma ponderação entre cada termo, e sua soma para qualquer $0 \leq u \leq 1$ é igual a unidade.

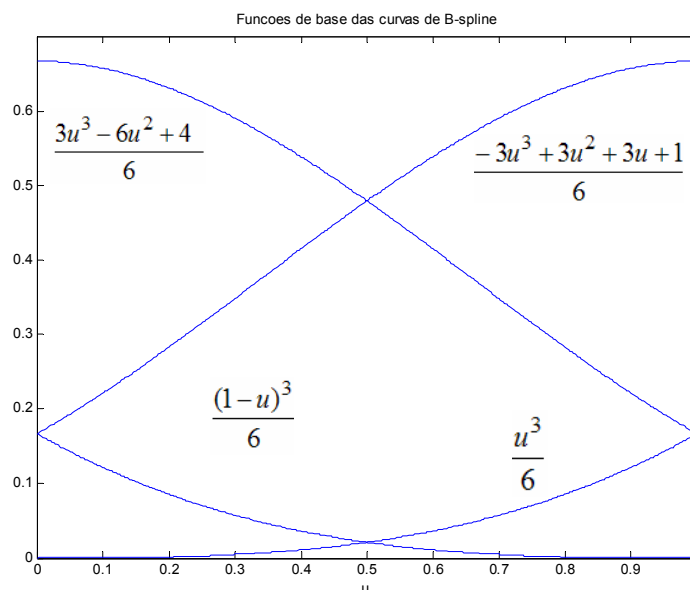


Figura 22. Funções de base da B-Spline

Da mesma maneira como demonstrado anteriormente, é possível expressarmos uma curva B-Spline em seu formato matricial. A matriz de base da B-Spline é dada por:

$$\mathbf{M}_{\text{BS}} = \frac{1}{6} \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 0 & 4 \\ -3 & 3 & 3 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix} \quad (30)$$

A equação matricial que dá a equação polinomial, onde P_i são as coordenadas em x dos pontos de controle, é dada por:

$$x(u) = \frac{1}{6} [P_0 \quad P_1 \quad P_2 \quad P_3] \cdot \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 0 & 4 \\ -3 & 3 & 3 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} u^3 \\ u^2 \\ u \\ 1 \end{bmatrix} \quad (31)$$

Tabela 1. Formação das funções de base de uma B-Spline cúbica uniforme

t	$t_i \leq t \leq t_i+d$	$t_i+d \leq t \leq t_i+2d$	$t_i+2d \leq t \leq t_i+3d$	$t_i+3d \leq t \leq t_i+4d$
$N_{i,0}(t)$	1	0	0	0
$N_{i+1,0}(t)$	0	1	0	0
$N_{i,1}(t)$	$t-t_i$	t_i+2d-t	0	0
$N_{i+1,1}(t)$	0	$t-(t_i+d)$	t_i+3d-t	0
$N_{i,2}(t)$	$\frac{(t-t_i)^2}{2d^2}$	$\frac{(t-t_i)(t_i+2d-t)+(t_i+3d-t)(t-(t_i+d))}{2d^2}$	$\frac{(t_i+3d-t)^2}{2d^2}$	0
$N_{i+1,2}(t)$	0	$\frac{(t-(t_i+d))^2}{2d^2}$	$\frac{(t-(t_i+d))(t_i+3d-t)+(t_i+4d-t)(t-(t_i+2d))}{2d^2}$	$\frac{(t_i+4d-t)^2}{2d^2}$
$N_{i,3}(t)$	$\frac{(t-t_i)^3}{6d^3}$	$\frac{(t-t_i)^2(t_i+2d-t)+(t-t_i)(t_i+3d-t)(t-(t_i+d))}{6d^3}$ + $\frac{(t_i+4d-t)(t-(t_i+d))^2}{6d^3}$	$\frac{(t-t_i)(t_i+2d-t)^2+(t_i+4d-t)(t-(t_i+d))(t_i+3d-t)}{6d^3}$ + $\frac{(t_i+4d-t)^2(t-(t_i+2d))}{6d^2}$	$\frac{(t_i+4d-t)^3}{6d^3}$
u	$u = \frac{(t-t_i)}{d}$	$u = \frac{t-(t_i+d)}{d}$	$u = \frac{t-(t_i+2d)}{d}$	$u = \frac{t-(t_i+3d)}{d}$
$N_{i,3}(u)$	$\frac{u^6}{6}$	$\frac{-3u^3+3u^2+3u+1}{6}$	$\frac{3u^3-6u^2+4}{6}$	$\frac{(1-u)^3}{6}$

2.6 Comparação entre as curvas apresentadas

A decisão de se utilizar umas das três curvas apresentadas neste capítulo, ou buscar alguma variação ou tipo específico de curva, deve ser tomada levando-se em consideração fundamentalmente a aplicação a ser utilizada.

Alguns critérios podem ser utilizados como: grau da continuidade da curva; complexidade da definição matemática e formulação da curva; capacidade da curva de representar o objeto desejado.

Também é possível fazer transformações dos pontos de controle de um tipo de curva para outro utilizando as matrizes de base como matrizes de transformação, sempre que estas possuam mesmo grau. Isso é muito utilizado em softwares de renderização¹ onde curvas B-Splines são transformadas em curvas Bézier que são mais simples de renderizar. Além disso, um usuário pode escolher trabalhar com uma curva, e na verdade, o programa armazena outro tipo de curva, como por exemplo, nas fontes Postscript, onde internamente as curvas são armazenadas como Bézier, mas os desenhistas de fontes utilizam curvas de Hermite. A Tabela 2 nos dá uma visão global das propriedades, vantagens e desvantagens de cada curva:

Tabela 2. Comparativo entre curvas polinomiais cúbicas

<i>Curva</i>	<i>Hermite</i>	<i>Bézier</i>	<i>B-Spline</i>	<i>B-Spline</i>
			<i>Uniforme</i>	<i>Não-uniforme</i>
Propriedade “Convex Hull”	N/A	Sim	Sim	Sim
Pontos de controle interpolam alguns pontos da curva	Sim	Sim	Não	Não
Continuidade Paramétrica C	1	1	2	2
Continuidade Geométrica G	1	1	2	2
Tipo de Controle	global	pseudo-local	local	local

¹ Termo muito utilizado em Computação Gráfica. Do inglês *render*, significa converter (uma imagem gráfica) de um arquivo de dados para uma formato visual, em um dispositivo de vídeo. Coloquialmente, se emprega para falar do processo de se gerar ou criar imagens gráficas em 3D.

Neste trabalho utilizaremos as curvas B-Splines uniformes. A escolha da curva B-Splines se deu pelos seguintes motivos: esta formulação é largamente utilizada na literatura mais recente (ABT et al., 2001; HARRIES et al., 1999; NARLI et al., 2003) de modelagem e síntese de cascos de embarcações. Em (HARRIES et al., 1998) são utilizadas curvas B-Splines uniformes com oito nós. As propriedades das B-Splines que são fundamentais para esta aplicação são: “convex-hull”, controle de forma local, e por serem de fácil manipulação através de transformações de sistemas de coordenadas. Além disso, se a curva atende a propriedade de “convex hull”, garante-se que todos os pesos da curva são positivos, o que implica que curva é calculável e entendível. Qualquer transformação de escala, translação ou rotação da curva é possível ser feita apenas realizando as transformações nos pontos de controle. Vale destacar que os melhores softwares desta área, como AutoCad, Rhinoceros 3D, entre outros, suportam a modelagem por B-Splines.

A utilização da simplificação por B-Splines uniformes também levou em conta a aplicação, principalmente por lidarmos com síntese de curvas por otimização. Tal flexibilidade e generalidade de B-Splines não-uniformes não são de grande importância para o tipo de curvas bidimensionais empregadas nesta modelagem. Além disso, a imposição de certas restrições no método de otimização já anularia o ganho de flexibilidade, caso fosse utilizado B-Splines não-uniformes.

Outro fator foi que pela simplificada formulação matemática, as curvas uniformes apresentariam um problema não custoso de abordagem e definição de funções objetivo e restrições, o que implica em condições de convergência e custos computacionais muito mais favoráveis para o software de otimização.

Além das três curvas apresentadas em detalhe, também foi estudada a utilização de NURBS. Estas são curvas com uma formulação originada da formulação de B-Splines, porém agrega-se uma variável adicional para se obter coordenadas homogêneas. As NURBS permitem o desenho de curvas e superfícies, como esferas e cilindros, as quais não são modeladas por B-Splines, e por isso, são consideradas o que há de mais avançado em computação gráfica. No entanto, para a aplicação deste

trabalho, não se faz necessária a utilização de NURBS, já que as curvas que serão sintetizadas são perfeitamente modeladas na formulação de B-Spline uniforme.

3 CONCEITOS DE OTIMIZAÇÃO E METODOLOGIA UTILIZADA

O método de otimização é empregado neste trabalho com o objetivo de obter parâmetros de uma curva B-Spline que satisfaça determinados critérios pré-definidos. Ao receber do usuário os parâmetros de projeto de uma embarcação, e traduzí-los para obter os parâmetros necessários para o traçado de uma curva, o algoritmo deverá convergir para uma possível curva que cumpra com as exigências impostas. Para tal, define-se uma função objetivo (critério de mérito) que será a função que iremos otimizar e que dependerá das variáveis de projeto, neste caso, os parâmetros da curva.

Para atingirmos a solução otimizada do problema, devem-se impor limitações ao problema, chamadas de restrições. Além de definirmos um domínio viável para a solução, a fim de se evitar resultados discrepantes e fisicamente impossíveis, como uma curva de comprimento zero, por exemplo (HAFTKA et al., 1995).

Em suma, o problema de otimização é:

Encontrar uma curva B-Spline cúbica uniforme, com determinado número de pontos de controle, de forma a satisfazer os parâmetros determinados e ser considerada "suave" em relação ao problema em questão.

A seguir serão melhor definidos os conceitos básicos de um problema de otimização, estruturando a metodologia utilizada para a síntese das curvas B-Spline.

3.1 Variáveis de Projeto

No caso de curvas, as variáveis de projeto são as coordenadas P_i dos pontos de controle da curva. Começa-se, portanto, a notar a importância de se implementar uma curva com um tipo de controle local. Esta facilidade acaba por ajudar muito a convergência do algoritmo, já que ao satisfazer uma restrição em um segmento de uma curva, para tentar satisfazer as demais restrições de outros segmentos, é possível

variar apenas os pontos de controle que alteram tal segmento, não mexendo na forma da curva toda. Por exemplo, uma curva cujas restrições sejam suas tangentes no início e no fim. Quando o algoritmo, variando os pontos de controle iniciais, atinge um valor aceitável para a tangente inicial, e ao variarmos os últimos pontos de controle da curva para buscarmos um valor que determinará a tangente final, a tangente inicial permanecerá inalterada. Isso causa uma independência entre as restrições.

As variáveis de projeto são classificadas como contínuas ou discretas. Neste caso, tem-se que os pontos de controle são variáveis contínuas, já que podem assumir qualquer valor.

No plano xy , os pontos de controle P_i de uma curva plana B-Spline uniforme, possuem coordenadas $P_i(x,y)$. Um problema de otimização se torna mais complexo, quanto maior for o número de variáveis de projeto utilizado. Por conta disso, é comum utilizar-se artifícios de modelagem explorando a simetria a fim de simplificar um problema. Na modelagem em questão, se abrirá mão de um grau de liberdade do ponto de controle, no caso, a variação de sua coordenada y , com o intuito de simplificar a modelagem do problema (ver Figura 46, no capítulo de Resultados, que mostra a variação das variáveis de controle conforme cada iteração). Alguns testes com o algoritmo demonstraram que a perda deste grau de liberdade, ou seja, o ponto de controle podendo variar em apenas uma direção, não implica em prejuízo para o algoritmo de suavização, e o ganho em termos do número de iterações é muito elevado. No capítulo de Resultados, será abordada com mais detalhes esta simplificação feita.

3.2 Função Objetivo

O critério de mérito do problema adotado foi obter uma curva “boa” segundo a avaliação do projetista para cada curva que será desenhada. Considera-se uma curva boa, para o caso de curvas de um casco de uma embarcação, curvas “suaves” (do inglês, “fair”).

(NOWACKI, 1970) faz questão de diferenciar uma curva lisa (“smooth”) de uma curva suave. (“fair”). A lisura da curva é relacionada com a ausência de oscilações, ou seja, dois pontos de inflexão na curva próximos um do outro. Já suavidade é definida subjetivamente por Nowacki como “agradável aos olhos”. De forma mais rigorosa, a suavidade de curva está relacionada à distribuição da curvatura ao longo do comprimento da curva. Uma curva suave deve cumprir com os seguintes requisitos:

- Não possuir ruído
- Não possuir regiões planas
- Apresentar continuidade para a primeira e segunda derivadas
- Livre de variações desnecessárias, como por exemplo, pontos de inflexão limitados e especificados.
- Distribuição da curvatura uniforme
- Preservação de forma

As curvas que satisfazem com estas condições terão uma aparência suave e aprazível para os olhos de um projetista experiente.

Matematicamente, de trabalhos anteriores de curvas de Bézier e de recentes publicações da área (HARRIES et al., 1998), a medida de suavidade uma curva (critério de suavidade ou “fairness criteria”) é quantificada através da energia² da curva, dada pela integral abaixo.

$$E_n = \int_{t_B}^{t_E} \left\{ \left(\frac{d^n x}{dt^n} \right)^2 + \left(\frac{d^n y}{dt^n} \right)^2 \right\} dt \quad (32)$$

² A palavra energia é referida em (YAMAGUCHI, 1988) como energia de dobramento (“bending energy”), onde a spline é relacionada com a teoria de elasticidade.

Onde $n = 1, 2, 3, \dots$. E_n é chamado de medida de carenagem, sendo que E_1 é a medida de elasticidade, E_2 é medida de flexão, e E_3 é a medida de torção.

Assim, pode-se definir uma função multi-objetivo para o problema de otimização, visando minimizar as medidas de carenagem.

$$F = e_1 E_1 + e_2 E_2 + e_3 E_3 \quad (33)$$

Através das constantes e_n , é possível estabelecer pesos para cada E_n , priorizando ou reduzindo sua importância em cada problema. Vale ressaltar que não existe um critério único, ou medida precisa para se determinar se uma curva é ou não suave, pois a decisão se uma curva atende ou não o critério de suavidade é do próprio projetista. Logo, os pesos de cada critério de carenagem na função objetivo são determinados empiricamente, e conforme o tipo de curva que se deseja desenhar.

Utilizando a formulação de uma B-Spline cúbica uniforme, através da Equação (29), o cálculo das medidas de carenagem se torna simples e de fácil tratamento numérico. Para mantermos a notação de utilizarmos como variável paramétrica t , considerar-se-á que t já sofreu a mudança de variável (representada por u na Tabela 1). As funções $x(t)$ e $y(t)$ e suas derivadas são definidas como:

$$\begin{aligned} x(t) &= \frac{(1-t)^3}{6} P_{x_{i-3}} + \frac{3t^3 - 6t^2 + 4}{6} P_{x_{i-2}} + \frac{-3t^3 + 3t^2 + 3t + 1}{6} P_{x_{i-1}} + \frac{t^3}{6} P_{x_i} \\ y(t) &= \frac{(1-t)^3}{6} P_{y_{i-3}} + \frac{3t^3 - 6t^2 + 4}{6} P_{y_{i-2}} + \frac{-3t^3 + 3t^2 + 3t + 1}{6} P_{y_{i-1}} + \frac{t^3}{6} P_{y_i} \end{aligned} \quad (34)$$

$$\begin{aligned} \frac{dx(t)}{dt} &= \frac{-3t^2 + 6t - 3}{6} P_{x_{i-3}} + \frac{9t^2 - 12t}{6} P_{x_{i-2}} + \frac{-9t^2 + 6t + 3}{6} P_{x_{i-1}} + \frac{3t^2}{6} P_{x_i} \\ \frac{dy(t)}{dt} &= \frac{-3t^2 + 6t - 3}{6} P_{y_{i-3}} + \frac{9t^2 - 12t}{6} P_{y_{i-2}} + \frac{-9t^2 + 6t + 3}{6} P_{y_{i-1}} + \frac{3t^2}{6} P_{y_i} \end{aligned} \quad (35)$$

$$\begin{aligned}\frac{dx^2(t)}{dt^2} &= \frac{-6t+6}{6}P_{x_{i-3}} + \frac{18t-12}{6}P_{x_{i-2}} + \frac{-18t+6}{6}P_{x_{i-1}} + \frac{6t}{6}P_{x_i} \\ \frac{dy^2(t)}{dt^2} &= \frac{-6t+6}{6}P_{y_{i-3}} + \frac{18t-12}{6}P_{y_{i-2}} + \frac{-18t+6}{6}P_{y_{i-1}} + \frac{6t}{6}P_{y_i}\end{aligned}\quad (36)$$

$$\begin{aligned}\frac{dx^3(t)}{dt^3} &= P_{x_{i-3}} + 3P_{x_{i-2}} - 3P_{x_{i-1}} + P_{x_i} \\ \frac{dy^3(t)}{dt^3} &= P_{y_{i-3}} + 3P_{y_{i-2}} - 3P_{y_{i-1}} + P_{y_i}\end{aligned}\quad (37)$$

Substituindo os valores das derivadas na Equação (33), obtém-se uma integral que pode ser resolvida, inclusive por métodos numéricos.

3.3 Restrições

Para entendermos melhor o problema de otimização deste trabalho, e suas restrições, vamos abordar agora os parâmetros de projeto de uma curva genérica plana.

Pode-se impor a uma curva os seguintes parâmetros de forma:

1. Posição do ponto inicial da curva: x_B, y_B

$$\begin{aligned}x_B &= x|_{t=0} \\ y_B &= y|_{t=0}\end{aligned}\quad (38)$$

2. Posição do ponto final da curva: x_E, y_E

$$\begin{aligned}x_E &= x|_{t=1} \\ y_E &= y|_{t=1}\end{aligned}\quad (39)$$

3. Ângulo da tangente no ponto inicial e final da curva: α_B, α_E

$$\begin{aligned}\text{para uma B-Spline não fechada, } \alpha_B &= \overrightarrow{P_0P_1} \\ \text{para uma B-Spline não fechada, } \alpha_E &= \overrightarrow{P_{m-2}P_{m-1}}\end{aligned}\quad (40)$$

Onde P_i , são os pontos de controle, e m é o número de pontos de controle da curva.

4. Curvatura em um ponto: C

$$C = \frac{x'y'' - x''y'}{(x'^2 + y'^2)^{3/2}} \quad \text{onde } x' = \frac{dx(t)}{dt}, \text{ etc.} \quad (41)$$

5. Área entre a curva e o eixo x :

$$A = \frac{1}{2} \left[\int_{t_B}^{t_E} (yx' - xy') dt + y_E x_E - y_B x_B \right] \quad (42)$$

6. Centróide da área A : x_C e y_C (momento de primeira ordem)

$$x_C A = M_y = \frac{1}{3} \left[\int_{t_B}^{t_E} (yx' - xy') x dt + y_E x_E^2 - y_B x_B^2 \right] \quad (43)$$

$$y_C A = M_x = \frac{1}{3} \left[\int_{t_B}^{t_E} (yx' - xy') y dt + \frac{1}{2} y_E^2 x_E - \frac{1}{2} y_B^2 x_B \right]$$

7. Interpolação em um ponto intermediário I : x_I, y_I

$$x_I = x \Big|_{t=t_I} \quad (44)$$

$$y_I = y \Big|_{t=t_I}$$

A Figura 23 mostra uma curva com os parâmetros citados a cima. Observe que nem sempre são necessários todos estes parâmetros para desenhar uma curva. Há casos que estes parâmetros são uma incógnita do problema, e deverão ser calculados após a obtenção da curva. Por exemplo, considere uma curva que passa por um ponto e possua tangentes iniciais e finais determinadas. Logo, teremos três restrições no problema. Após a síntese desta curva pelo método de otimização, se encontrará a curva mais suave que atenda estes critérios. Se a área desta curva for um dado importante para o projeto, podemos calculá-la após a curva pronta. Os pontos iniciais e finais também podem ser impostos utilizando-se da multiplicidade dos nós, não precisando ser tratados como uma restrição, exercendo o papel de condição de contorno do problema.

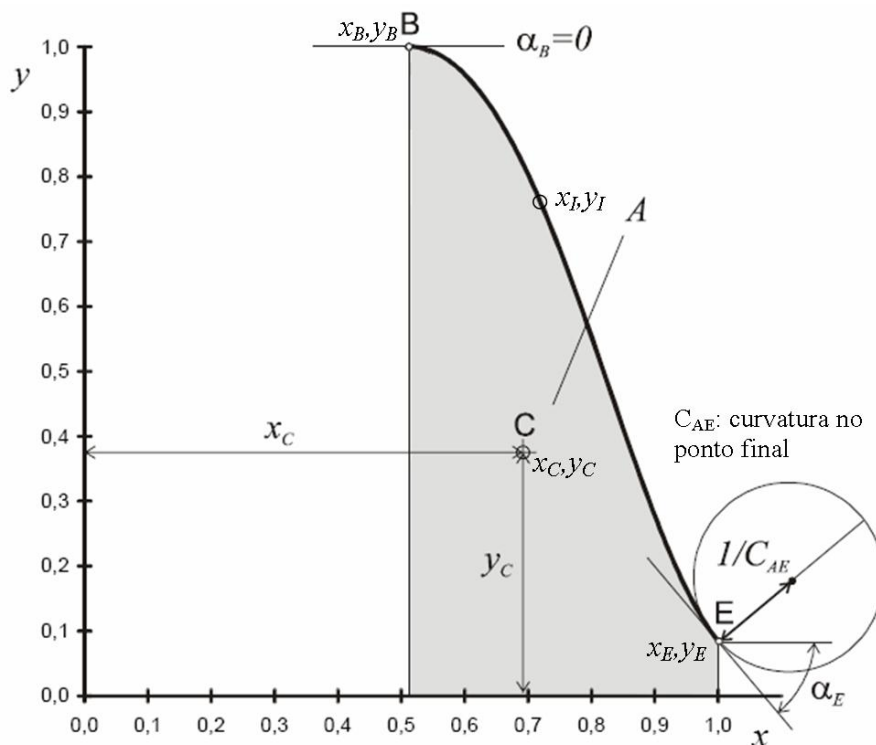


Figura 23. Exemplo de curva com seus parâmetros de forma

Portanto, não é trivial especificarmos as restrições que devem ser impostas para cada curva. A escolha de quais parâmetros impor a cada curva varia caso a caso. Imagine, por exemplo, a curva de áreas seccionais de um veleiro. Certamente, sua área e centróide são parâmetros importantes no desenho da curva. A curvatura em um ponto específico não é uma condição necessária, já que a função objetivo já buscará uma curva suave.

Utilizar restrições de igualdade encarece o custo computacional do algoritmo. Desta forma, deve-se transformar esta restrição em uma restrição de desigualdade. Suponhamos que queremos obter uma curva com área igual a A_0 . Pode-se reformular esta restrição de duas formas.

Restrição de igualdade: $A(x) = A_0$

Duas restrições de desigualdade: $A(x) \leq A_0$ e $A(x) \geq A_0$ (45)

Restrição de desigualdade com relaxamento: $(A(x) - A_0) - c \leq 0$

A técnica de utilizar o critério de relaxamento dá ao algoritmo uma flexibilidade maior para buscar a solução ótima. A constante c atua como uma margem de erro admissível para o problema. Podemos não necessitar que a área da curva seja exatamente A_0 , mas $A_0 \pm c$. Ainda, pode-se desejar que a variação de $A(x)$ em relação a A_0 seja de, por exemplo, 1%. Logo, a restrição do problema fica ainda melhor condicionada numericamente.

$$\frac{(A(x) - A_0)}{A_0} - c \leq 0 \quad \text{com } c = 1\% \quad (46)$$

Desta forma, a restrição fica normalizada, aumentando a chance de sucesso do método. A constante c pode ser um critério de projeto, ou ser determinada empiricamente, dependendo do compromisso entre custo computacional versus precisão do resultado (HAFTKA et al., 1995).

3.4 Análise de Sensibilidade

Pode-se lançar mão da análise de sensibilidade de um problema de otimização para identificar quais variáveis de projetos são mais sensíveis para a função objetivo e suas restrições. Esta análise permite a percepção de oportunidades na modelagem, como eliminar uma certa variável de projeto que, nas proximidades da região ótima, não influi na função objetivo. Todavia, o motivo principal de se realizar a análise de sensibilidades é que estas são utilizadas no algoritmo de otimização, logo, estudá-las significa conhecer melhor o problema.

A sensibilidade de uma função em relação à uma variável de projeto é obtida derivando-se esta função em relação a esta variável. No problema exposto aqui, as variáveis de projeto são as coordenadas y dos pontos de controle da curva. Nos

capítulos anteriores, foram apresentadas as expressões tanto das curvas utilizadas como da função objetivo do problema. Logo, é possível obter uma expressão analítica da sensibilidade tanto da função objetivo quanto de suas restrições.

Da Equação (33), tem-se a formulação da função multi-objetivo do problema. Como as formulações das parcelas E_i são semelhantes, diferindo apenas do grau da derivada das funções $x(t)$ e $y(t)$, será apresentado um exemplo onde $e_2=1$ e $e_1=e_3=0$, ou seja, apenas a energia de flexão E_2 ativa.

Logo, função objetivo fica:

$$F = 0E_1 + E_2 + 0E_3 \quad (47)$$

Da Equação (32), tem-se que:

$$F = \int_{t_B}^{t_E} \left\{ \left(\frac{d^2x}{dt^2} \right)^2 + \left(\frac{d^2y}{dt^2} \right)^2 \right\} dt = \int_{t_B}^{t_E} \left(\frac{d^2x}{dt^2} \right)^2 dt + \int_{t_B}^{t_E} \left(\frac{d^2y}{dt^2} \right)^2 dt \quad (48)$$

Como apenas as coordenadas y dos pontos de controle são variáveis, a primeira parcela da Equação (48) é nula. Portanto, a sensibilidade da função objetivo em relação às variáveis de projeto fica:

$$\frac{dF}{dP_{y_i}} = \frac{d}{dP_{y_i}} \int_{t_B}^{t_E} \left(\frac{d^2y}{dt^2} \right)^2 dt \quad (49)$$

A Equação (35) mostra a primeira derivada da função $y(t)$. Derivando-a novamente em relação a t e substituindo na equação acima, tem-se:

$$\frac{dF}{dP_{y_i}} = \frac{d}{dP_{y_i}} \int_{t_B}^{t_E} \left((-t+1)P_{y_{i-3}} + (3t-2)P_{y_{i-2}} + (-3t+1)P_{y_{i-1}} + tP_{y_i} \right)^2 dt \quad (50)$$

É possível continuar o desenvolvimento desta expressão, fazendo a integração em relação a t , em seguida derivando a função resultante para cada P_{yi} . O resultado é uma função muito elaborada, onde aparecerão termos em t^3 , t^2 , e t , multiplicados por combinações dos pontos de controle P_{yi} . Fazendo a mesma análise para as restrições do problema, é possível também obter expressões analíticas da sensibilidade semelhantes para cada uma das restrições.

Estas expressões analíticas das sensibilidades poderiam ser utilizadas para os cálculos do gradiente da função objetivo no algoritmo de otimização. Optou-se, no entanto, por utilizar o cálculo das sensibilidades pelo método de diferenças finitas. Isso porque, ao se utilizar o Matlab, o custo computacional para o cálculo dos gradientes em relação ao problema como um todo é pequeno, e não implica em problemas de convergência, já que se trata de expressões polinomiais.

3.5 Algoritmo de otimização utilizado

Definido o problema de otimização, o próximo conceito é discutir a região de localização de sua solução. A parte do domínio em que as restrições são respeitadas é denominada de domínio viável. Portanto, a modelagem e escolha das restrições influenciarão na localização da solução ótima. Para a compreensão de como podemos construir da melhor forma o conjunto de restrições de um problema de síntese de curvas, abordaremos os conceitos de mínimo local e mínimo global, e problemas convexos.

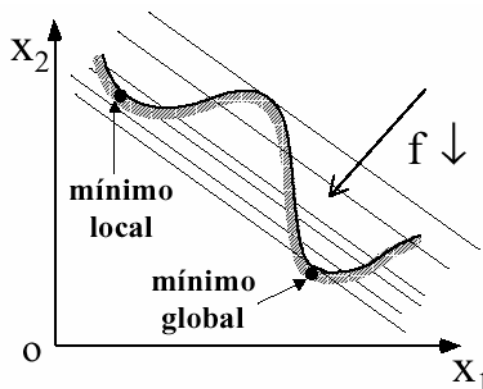


Figura 24. Conceito de mínimo local e mínimo global

O mínimo local aparece em uma região da função $f(x_1, x_2)$ considerada uma “cavidade secundária”, e seu valor é maior que o mínimo global. Esta é uma situação comum nos problemas de otimização, e está relacionada também com a adoção de valores iniciais discrepantes, ou longe da solução ótima. A certeza de que o ponto ótimo encontrado é o mínimo global somente é possível em problemas convexos. (Um problema é dito convexo se a função objetivo do problema e seu domínio viável são convexos).

Na prática, deve-se analisar se o “chute inicial” dos pontos de controle é, em primeiro lugar, condizente com a formulação matemática proposta, e se estes valores levam o algoritmo à solução ótima. Logo, adotar diferentes valores para as variáveis de projeto é uma prática saudável, pois, previne-se resultados inesperados do algoritmo.

Para se definir o algoritmo a ser utilizado também é necessário conhecer se o problema em questão é linear ou não-linear. Um problema de otimização é dito linear se sua função objetivo e suas restrições são lineares, do contrário, o problema é não-linear. Se o problema de otimização é linear, pode ser resolvido diretamente com métodos de programação linear.

O problema de otimização de curvas B-Splines é não linear, pela própria formulação da função objetivo e suas restrições. Portanto, devem-se buscar métodos numéricos não-lineares com restrições para a resolução do problema.

Conforme já citado, utiliza-se neste trabalho o software comercial *MATLAB* para a implementação do problema, que possui rotinas com algoritmos de minimização de funções com restrições. A rotina utilizada é a *fmincon*, que implementa algoritmos para minimização não-linear de problemas com restrições (MATLAB, 2002).

Esta rotina possui dois modos de utilização, o primeiro, chamado de Otimização de Larga-Escala, utiliza o método de Newton ("interior-reflective Newton method")

(VANDERPLAATS, 1984). O segundo, chamado de Otimização de Média-Escala, utiliza-se do método de Programação Quadrática Seqüencial (PQS, ou “SQP”, do inglês). Este método é um dos mais empregados em problemas de programação não-linear e é o que é utilizado em todas as otimizações feitas neste trabalho. Segundo (SCHITTKOWSKI, 1985), o método PQS supera em eficiência, acurácia e porcentagem de soluções bem sucedidas, todos os outros métodos testados e avaliados.

A idéia deste método é aproximar, a cada iteração, o problema não-linear por uma sucessão de subproblemas de programação quadrática (PQ) com restrições lineares. A aproximação da matriz Hessiana da função Lagrangeana é feita utilizando o método de Quase-Newton com aproximação BFGS (“Broyden-Fletcher-Goldfarb-Shanno”) (VANDERPLAATS, 1984). Esta é utilizada para formar o subproblema, cuja solução proverá uma direção de busca para o método de busca linear. O problema de programação quadrático é definido por:

$$\begin{aligned} \min_x F(x) = \frac{1}{2} x^T A x + b x \quad \text{sujeito a:} \\ Bx \leq c \\ x \geq 0 \end{aligned} \tag{51}$$

onde A é a matriz Hessiana.

Basicamente, a implementação do PQS consiste em 3 estágios a cada iteração: atualizar a matriz Hessiana da função Lagrangeana; resolver o subproblema de Programação Quadrática; realizar busca linear e avaliação do valor da função mérito. O *fmincon* também elabora a otimização com funções multi-objetivo. (MATLAB, 2002)

O critério de parada utilizado para todas as otimizações deste trabalho (todas utilizam o *fmincon*) é o critério padrão do *MATLAB*. Isto é, critério de parada padrão do *fmincon* é que a máxima magnitude da derivada direcional na direção de busca seja menor que $2 \cdot 10^{-4}$, e que as restrições não oscilem mais que 10^{-6} .

Como limitações da rotina *fmincon* pode-se citar: a rotina somente aceita variáveis reais, a função objetivo e restrições do problema devem ser contínuas, e por fim, o método pode retornar uma solução de mínimo-local.

Utilizando a formulação do MATLAB para a rotina *fmincon*, o problema de otimização de curvas B-Spline, em sua forma genérica fica:

$$\min_{P_y} \sum_1^3 E_i \quad \text{sujeito a:} \quad (52)$$

$$c(x) \leq 0$$

$$ceq(x) = 0$$

$$A \cdot x \leq b$$

$$Aeq \cdot x = beq$$

$$lb \leq x \leq ub$$

[Pout] = fmincon(@fobjetivo,Pin,A,b,Aeq,beq,lb,ub,@restrnlinear,options,ctes)

Onde,

Pout: Pontos de controle da curva otimizada.

fobjetivo: $F = e_1 E_1 + e_2 E_2 + e_3 E_3$

Pin: valores iniciais dos pontos de controle da curva inicial.

restrnlinear: conjunto de restrições não-lineares (Equações (38) a (44))

options: conjunto de opções para configuração da rotina

ctes: outros parâmetros da otimização

Esta é a formulação básica de um problema de otimização utilizando o *fmincon*. Na Equação (52), as restrições estão apresentadas genericamente, e representam as Equações (38) a (44), que podem ou não ser consideradas de acordo com cada curva. Veremos a seguir que nem todas as restrições são úteis na síntese de uma curva

específica, e a determinação de qual delas utilizar é a arte do problema. Lembrando que quanto maior o número de restrições, mais complexo fica o problema. A formulação de cada problema de otimização implementado para cada curva será mostrada à diante.

3.6 Otimização paramétrica do casco

Dada uma forma geométrica de casco, pode-se desejar alterar as suas dimensões, de forma a obter um casco modificado de melhor desempenho, ou seja, otimizado para uma condição específica de regata. O método de otimização paramétrica será utilizado para se obter a melhor configuração do casco que minimize um critério de mérito escolhido. Devido ao fato de que, na maioria das vezes, o comportamento hidrodinâmico do veleiro é complexo, e não sendo possível descrever analiticamente a função objetivo e seus gradientes, faz-se necessário utilizar na otimização a técnica de Superfície de Resposta.

A metodologia descrita a seguir explora de forma sistemática os parâmetros relevantes de um projeto, dentro de seu domínio viável, encontrando a melhor configuração para os requisitos dados.

(NOWACKI, 1970) cita que o problema de minimizar a resistência de um barco é “um dos problemas mais fascinantes de hidrodinâmica de embarcações”. No exemplo em questão, será utilizada como função objetivo a resistência residual, que será obtida através da série sistemática de veleiros da Delft University of Technology (LARSSON et al., 1994). Na formulação utilizada, as variáveis de projeto escolhidas estão descritas analiticamente na formulação da função objetivo, e o problema poderia ser resolvido analiticamente. No entanto, o que se busca é avaliar a eficácia do método, sendo possível utilizar quaisquer combinação de variáveis para qualquer função objetivo que se deseja otimizar. Caso a mesma resistência residual fosse obtida através de simulações em CFD, o Método da Superfície de Resposta, a ser apresentado a seguir, resolveria da mesma forma o problema, e não seria possível utilizar um método analítico.

3.6.1 Método da Superfície de Resposta (MSR)

O Método de Superfície de Resposta (MSR) é um conjunto de técnicas estatísticas e matemáticas úteis para o desenvolvimento, aperfeiçoamento e otimização de processos (MYERS et al., 1995). A metodologia de Superfície de Resposta combina técnicas de ajuste de curvas e superfícies por regressão com otimização utilizando as superfícies geradas.

O método consiste em avaliar a função objetivo com uma combinação adequada das variáveis de projeto, obtendo um número suficiente de valores da função objetivo, que serão utilizados para aproximar uma superfície (Figura 25). Esta superfície descreverá o comportamento da função objetivo dentro de um domínio especificado das variáveis de projeto. Em seguida, através de um método de otimização encontra-se o valor ótimo da função objetivo, obtendo-se a melhor combinação para as variáveis de projeto.

Neste trabalho, será utilizado um modelo de segunda ordem para descrever o comportamento da função objetivo. (MYERS et. Al, 1995) cita algumas razões pelas quais este modelo é largamente utilizado:

1. O modelo de segunda ordem é bastante flexível, e pode descrever uma grande variedade de formas de funções.
2. A estimação dos parâmetros (β 's) é simples no modelo de segunda ordem, e pode ser utilizado o método dos mínimos quadrados.
3. Existe uma considerável experiência prática indicando que o modelo de segunda ordem funciona bem na solução de problemas reais de superfície de resposta.

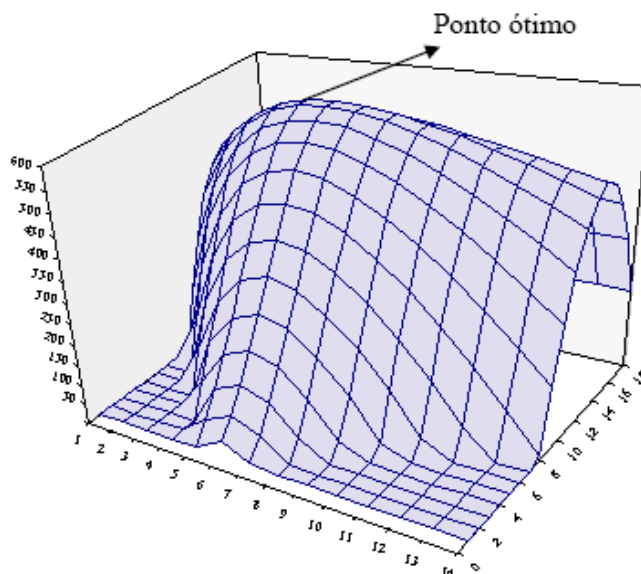


Figura 25. Exemplo de uma superfície de resposta (MACHIAVERNI, 2004)

Logo, a superfície de resposta $y(x_1, x_2)$ pode ser descrita da seguinte forma, em termos das variáveis de projeto (MACHIAVERNI et. al, 2005):

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1^2 + \beta_4 x_2^2 + \beta_5 x_1 x_2 \quad (53)$$

Fazendo a mudança de variável para as variáveis x_3 , x_4 e x_5 , da seguinte maneira:

$$\begin{aligned} x_3 &= x_1^2 \\ x_4 &= x_2^2 \\ x_5 &= x_1 x_2 \end{aligned} \quad (54)$$

Pode-se reescrever a Equação (53) na forma de uma regressão linear:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_4 x_4 + \beta_5 x_5 \quad (55)$$

Os coeficientes β_i 's são determinados através do método de mínimos quadrados. Utilizando duas variáveis de projeto e n experimentos, os coeficientes são calculados como segue:

$$\boldsymbol{\beta} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{Y} \quad (56)$$

onde:

$$\mathbf{Y} = \begin{bmatrix} y_1 \\ y_2 \\ \dots \\ y_n \end{bmatrix}, \mathbf{X} = \begin{bmatrix} 1 & x_{11} & x_{21} & \dots & x_{51} \\ 1 & x_{12} & x_{22} & \dots & x_{52} \\ \dots & \dots & \dots & \dots & \dots \\ 1 & x_{1n} & x_{2n} & \dots & x_{5n} \end{bmatrix}, \boldsymbol{\beta} = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \beta_3 \\ \beta_4 \\ \beta_5 \end{bmatrix} \quad (57)$$

Obtida a superfície de segunda ordem que aproxima a função objetivo, é possível obter a configuração ótima através de uma série de procedimentos matemáticos. Com o objetivo de utilizar ferramentas de otimização neste trabalho, utilizar-se-á novamente a rotina *fmincon*, do software MATLAB, já descrita anteriormente.

3.6.2 Projeto de Experimentos

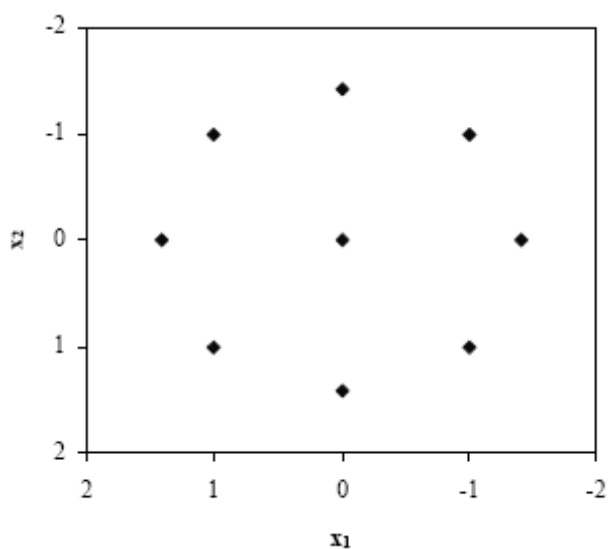
Descrito o procedimento de obtenção pelo MSR, o próximo passo é determinar quais combinações das variáveis de projeto devem ser utilizadas para avaliar a função objetivo, a fim de prover os pontos iniciais para o desenho da superfície de resposta. Para resolver problema será apresentado um método chamado de Projeto de Experimentos ou DOE (do inglês, “Design of Experiments”) (MYERS et al., 1995).

Trata-se de uma técnica que determina os valores das variáveis de projeto mais apropriada para aproximar uma superfície ao comportamento da função objetivo, no domínio especificado. O método mais popular, dentre os encontrados na literatura, é o CCD (“Central Composite Design”) (MYERS et al., 1995), que determina pontos com a finalidade de melhor estimar os parâmetros dos termos lineares e de interação entre as variáveis. A Tabela 3 mostra os valores das variáveis genéricas utilizadas no CCD para duas variáveis. Este valores formam uma circunferência de raio igual a 1, conforme visto na Figura 26.

Tabela 3. Tabela genérica de CCD para duas variáveis

#	x_1	x_2
1	0	0
2	1	1
3	-1	1
4	-1	-1
5	1	-1
6	α	0
7	0	α
8	$-\alpha$	0
9	0	$-\alpha$

O parâmetro α é a distância axial que varia de 1 a $k^{0.5}$, onde k é o número de variáveis de projeto. Neste trabalho, será utilizado $\alpha = k^{0.5}$, fazendo com que os pontos pertençam a uma circunferência comum. Isso proporciona ao projeto a propriedade de rotabilidade, que assegura que os valores a uma mesma distância de um centro de projeto possuem a mesma variância.

Figura 26. CCD genérico com $k=2$

Os pontos $(1,1)$, $(-1,1)$, $(-1,-1)$ e $(1,-1)$ ajudarão a estimar os termos lineares e as interações entre dois fatores. Os pontos $(0, \sqrt{2})$, $(\sqrt{2}, 0)$, $(0, -\sqrt{2})$, $(-\sqrt{2}, 0)$ e $(0,0)$ são utilizados para a estimação dos termos quadráticos.

4 PRINCÍPIOS DE PROJETO DE CASCOS DE BARCOS À VELA

4.1 Definição das dimensões do casco

Define-se casco como o corpo do navio sem mastreação, aparelhos acessórios ou qualquer outro arranjo. Sua principal característica de forma é ter um plano de simetria (plano diametral) que se imagina passar pelo eixo da quilha (FONSECA, 2005). O casco de um veleiro possui uma forma tridimensional complexa e envolve mais que puramente equações ou Engenharia, é um produto da arte e experiência de seu projetista. Com o advento de softwares de CAD, o desenho de um casco se tornou algo mais rápido e com a maior vantagem de permitir sua representação em diversas vistas e em 3D.

Um casco é definido por algumas medidas geométricas básicas, e outras mais específicas que não serão abordadas aqui. Utilizaremos a nomenclatura convencional em inglês, utilizada em (LARSSON et al., 1994). A Figura 27 mostra uma veleiro com as principais dimensões desenhadas. A linha d'água de projeto ("Design Waterline") é indicada pela sigla *DWL*.

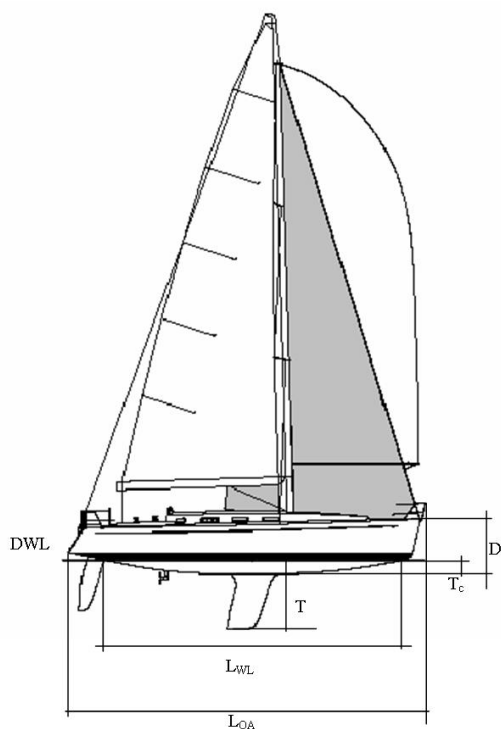


Figura 27. Definições das dimensões principais de um casco

As principais dimensões, coeficientes e centros geométricos mais comumente utilizados são:

a. Dimensões

L_{OA} (“Length overall”): comprimento máximo do casco

L_{WL} (“Length of waterline”): comprimento da linha d’água

B_{max} (“Beam maximum”): largura máxima do casco

B_{WL} (“Beam of waterline”): largura na linha d’água.

T (“Draft”): calado ou profundidade a que se encontra o ponto mais baixo da quilha

T_c (“Draft of the canoe body”): calado sem considerar a profundidade da quilha

D (“Depth”): Pontal

D_c (“Depth of the canoe body”): Pontal sem considerar a dimensão da quilha

∇_c : Volume de deslocamento, volume deslocado pela parte imersa do casco

Δ_c : Massa deslocada

A_{BM} : Área de maior seção transversal do casco (seção mestra)

FB (“Freeboard”): comprimento da borda livre, distância entre a linha de borda e a linha d’água.

A_{WP} (“Waterplane Area”): é a área interior à linha d’água de projeto, e é importante, pois determina o peso por mm de imersão, ou seja, o peso adicional para afundar o casco em certa distância.

b. Coeficientes

C_p : Coeficiente prismático, relação entre o volume deslocado e o volume de um sólido com o comprimento do casco e a área igual à seção transversal da parte imersa.

C_B : Coeficiente de bloco, relação entre o volume deslocado e o volume do paralelepípedo envolvente.

c. Centros Geométricos

LCB (“Longitudinal Center of buoyance”): centro longitudinal de carena

LCF (“Longitudinal Center of flotation”): centro longitudinal de flutuação

G: Centro de gravidade

C_c: Centro de carena: é o centro de gravidade da água deslocada (ponto de aplicação da força de empuxo).

Usualmente, o casco é representado por um conjunto de cortes que formam duas famílias de curvas, chamado de plano de linhas. O plano de linhas é a representação mais antiga e comum do casco. As linhas horizontais são chamadas de linhas d’água e as linhas verticais de balizas. As linhas das balizas são desenhadas geralmente seguindo a regra de numerá-las da proa para a popa de 0 até 10, espaçadas de forma equidistante, de forma que a baliza 0 coincida com o início da D_{WL} (linha d’água de projeto) e a baliza 10 cruze o final da D_{WL} . Balizas -1 e 11 podem ser desenhadas para ilustrar cortes adicionais.

As linhas d’água, especialmente, a linha d’água de projeto deve possuir características geométricas adequadas, pois está relacionada às propriedades da embarcação como resistência ao avanço, facilidade de construção, arranjo geral, entre outros. Além disso, a suavidade ou carenagem das linhas ou superfície do casco influi diretamente sobre o escoamento em torno do casco e geração de vórtices, principal fator responsável pela resistência de forma. Para um escoamento suave, sem separação da camada limite e grande geração de vórtices, não deve haver descontinuidades ou mudanças bruscas na linha do casco. E isso deve ser interpretado e traduzido matematicamente pela ferramenta de síntese da curva.

Na vista frontal do casco, aproveita-se da simetria do casco para representar metade frontal à direita e a metade de trás do casco à esquerda. Outros cortes podem ser também feitos a fim de ajudar a representação e percepção da superfície, como por exemplo, cortes longitudinais (chamados de linhas de alto) e cortes diagonais (Figura 28). No entanto, bastam dois cortes para ser possível representar a superfície e, por isso, somente as linhas das balizas e linhas d’água serão utilizadas neste trabalho.

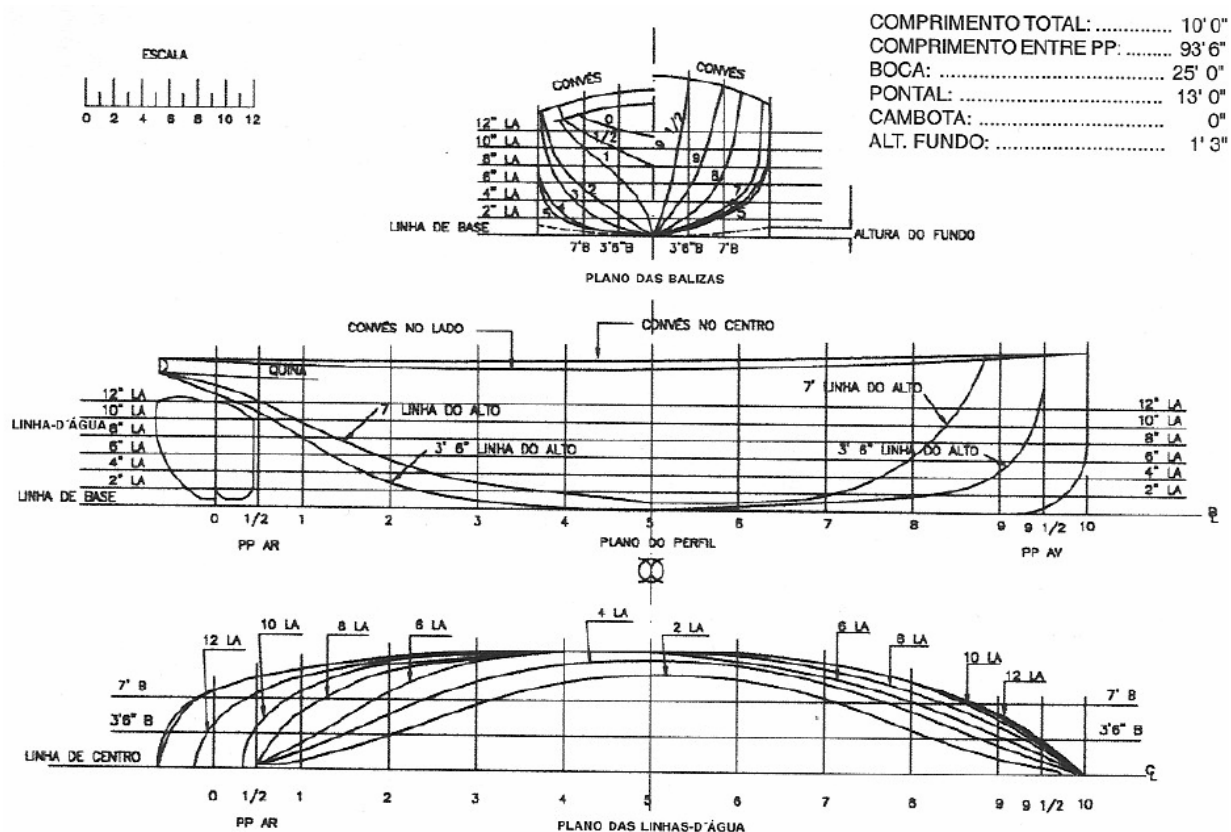


Figura 28. Principais linhas do desenho de um casco (FONSECA, 2005)

Outra definição importante é a definição de carena. Carena é um termo empregado para designar o invólucro do casco nas obras vivas (parte do casco abaixo do plano de flutuação) (FONSECA, 2005). A superfície da carena somada à superfície do costado, representa a área total da superfície do casco. O volume da carena é o volume entre a superfície molhada e um dado plano de flutuação. Diz-se, também, que um casco é carenado quando sua superfície é arredondada.

O projeto de casco de veleiros envolve cálculos sobre estabilidade, modelos hidrostáticos e modelos hidrodinâmicos. Em (LARSSON et al., 1994), estes três tópicos são abordados e são apresentados modelos simplificados, porém largamente utilizados. (OOSSANEM, 1993) por sua vez apresenta modelos mais completos, considerando também a influência hidrodinâmica da quilha e leme sobre o casco, e vice-versa.

Neste trabalho, não serão abordados em detalhe conceitos de projeto de veleiros, uma vez que a síntese das curvas que compõem um casco receberá como entradas os parâmetros calculados e especificados pelo projetista, e grandezas resultantes dos modelos citados acima. Ao gerar o desenho do casco, o método utilizado aqui, ajuda o projetista a confirmar ou alterar premissas e parâmetros, além de permitir simulações mais complexas, sendo de grande utilidade no ciclo de projeto da embarcação.

4.2 Modelo hidrodinâmico

O casco, leme e quilha são as partes do sistema hidrodinâmico de um veleiro, e geram forças de arrasto devido ao movimento de avanço. Devido à complexidade de seu formato e do fato de cortar a interface em dois meios, a resistência ao avanço sofrida pelo casco não possui um modelo analítico completo, todavia, existem disponíveis diferentes modelos empíricos para a estimação das forças geradas (SALLA, 2005).

As principais componentes da resistência do casco são a resistência viscosa, cuja maior componente é a resistência friccional, e a resistência residual, cuja maior componente é a resistência de onda. Em velocidades baixas, a resistência viscosa predomina devido às forças friccionalis entre o casco e a água. Porém, à medida que a velocidade aumenta, a parcela predominante passa a ser a resistência de onda, que ocorre devido às ondas geradas pelo casco, dissipando energia. A explicação do modelo completo detalhando todas as componentes da resistência pode ser encontrada em (LARSSON et al., 1994).

O número de Froude é definido como uma velocidade adimensional, onde a velocidade é dividida pela raiz quadrada da aceleração da gravidade vezes o comprimento da linha d'água.

$$F_n = \frac{v}{\sqrt{g \cdot L_{WL}}} \quad (58)$$

onde v é a velocidade em metros por segundo e g é a aceleração da gravidade. A Figura 29 mostra uma comparação entre a resistência residual (R_r) e viscosa (R_v) para o casco de 40 pés estudado neste trabalho, e que será apresentado a seguir. A resistência friccional foi calculada segundo a fórmula que a relaciona com o coeficiente de atrito viscoso, o quadrado da velocidade do barco e a área molhada (LARSSON et al, 1994).

Em baixas velocidades, a resistência viscosa (R_v), proporcional à área molhada do casco, corresponde a grande parte da resistência total. Nestas condições, o adernamento contribui para a sua redução, pois diminui também a área molhada. Já para altas velocidades, há claramente a predominância da resistência residual (R_r), que cresce aproximadamente com a quarta potência da velocidade do barco. Por esta razão, se estudará neste trabalho somente a otimização da resistência residual, para uma certa velocidade definida.

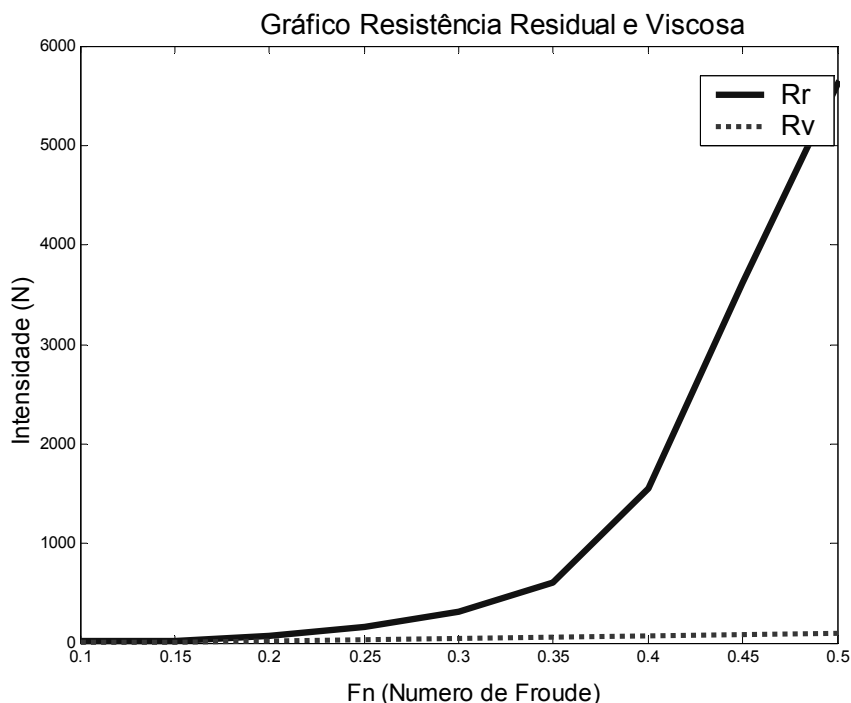


Figura 29. Gráfico comparativo entre resistência residual e viscosa

A resistência residual agrupa o conjunto de resistência dependente do formato do casco. O modelo empírico utilizado neste trabalho para calcular esta componente foi proposto pela Delft University of Technology, na Holanda, e compreende uma série sistemática obtida através de ensaios com diversos veleiros semelhantes. Para este modelo, a força é proporcional ao peso da embarcação e aos seguintes fatores, contidos nos limites abaixo:

$$\begin{aligned}
 2,76 &\leq \frac{L_{WL}}{B_{WL}} \leq 5,00 \\
 2,46 &\leq \frac{B_{WL}}{T_C} \leq 19,32 \\
 4,34 &\leq \frac{L_{WL}}{(V_C)^{1/3}} \leq 8,50 \\
 0,00 &\leq LCB \leq -6,00 \\
 0,52 &\leq C_p \leq 0,60
 \end{aligned} \tag{59}$$

onde LCB é dado como um percentual de L_{WL} , sendo positivo da metade em diante. Seu cálculo pode ser dado da seguinte maneira:

$$LCB = -\left(\frac{xLCB}{L_{WL}} - 0,5\right) \cdot 100 \tag{60}$$

Logo, a formulação da resistência residual (R_r) em Newtons é calculada como segue. Para um determinado número de Froude, encontram-se coeficientes a_i tabelados. Existem duas tabelas e duas formulações distintas para valores diferentes de F_n . A primeira compreende os valores de F_n entre 0,125 e 0,450:

$$\begin{aligned}
 R_r = &(A_C g)(a_0 + a_1 C_p + a_2 LCB + a_3 (B_{WL}/T_c) + \\
 &+ a_4 L_{WL}/\nabla_C^{1/3} + a_5 C_p^2 + a_6 C_p L_{WL}/\nabla_C^{1/3} + a_7 LCB^2 + \\
 &+ a_8 (L_{WL}/\nabla_C^{1/3})^2 + a_9 (L_{WL}/\nabla_C^{1/3})^3
 \end{aligned} \tag{61}$$

E para valores de F_n compreendidos entre 0,475 e 0,75:

$$R_r = (A_C g) (c_0 + c_1 L_{WL}/B_{WL} + c_2 A_{WP}/\nabla C^{2/3} + c_3 L_{WL} + c_4 (L_{WL}/B_{WL})^2 + c_5 (L_{WL}/B_{WL}) (A_{WP}/\nabla C^{2/3})^3) \quad (62)$$

(LARSSON et al., 1994) sugere que variações de 10 a 20% podem ocorrer na resistência residual, variando-se os parâmetros de forma do casco, como $L_{WL}/\nabla C^{1/3}$, C_P e LCB . Por este motivo, no estudo da otimização da resistência residual, estes dois últimos parâmetros serão utilizados para exemplificar a metodologia de otimização pelo Método de Superfície de Resposta com duas variáveis.

A escolha destes parâmetros é bastante pertinente. Além de ambos estarem diretamente envolvidos no modelo de resistência residual, o coeficiente prismático está fortemente relacionado com a geometria do casco, pois é o índice que melhor expressa o quanto um barco é fino nas extremidades. Um C_P alto significa que o barco possui o seu volume bem distribuído ao longo do seu comprimento. Caso contrário, o barco possui o seu volume concentrado no centro. O LCB é outro parâmetro utilizado para avaliar a distribuição do volume ao longo do comprimento do casco, além da estabilidade. (FASSARDI et. al, 2006) também utiliza estes dois parâmetros na busca de um casco de veleiro otimizado para um mesmo volume de deslocamento através da otimização paramétrica com a técnica de Superfície de Resposta.

5 MODELAGEM PARAMÉTRICA

5.1 Proposta de modelagem

A modelagem paramétrica de um casco consiste em transformar a forma do casco em equações matemáticas, permitindo, além de sua armazenagem de forma eficiente através de parâmetros, modificar a forma geométrica apenas alterando estes valores de parâmetros.

Na literatura da área, não somente balizas e linhas d'águas são parametrizadas no design de um casco. (HARRIES et al., 1998) utiliza doze curvas planas, divididas em um conjunto de curvas básicas primárias e secundárias, para a modelagem de um casco. Em seguida, criam-se seções do casco, para posteriormente parametrizá-lo com uma superfície. Já em (HARRIES et al., 2001), a modelagem é feita com uma abordagem onde a modelagem do casco é feita somente a partir de parâmetros de curvas e superfícies B-Splines sujeitas a um algoritmo de otimização, cujo critério de suavidade é a medida de flexão (E_2).

Neste trabalho será utilizado um conjunto de curvas próprio e uma estratégia traçada para a modelagem do casco, combinando a metodologia de (HARRIES et al., 2001) com os conceitos de projeto de (LARSSON et al., 1994). O objetivo é obter os parâmetros necessários a fim de desenhar as curvas de todas as balizas sujeitas à minimização das medidas de carenagem. O casco gerado será intrinsecamente suave, posto que as curvas que o compõe também o são, já que foram submetidas aos critérios de suavidade.

O sistema que será introduzido, não utiliza a modelagem do casco por B-Splines de superfície, pelos motivos explicados anteriormente. A principal vantagem disto é reduzir a complexidade do problema de suavização de uma superfície tridimensional, descrevendo-a com um modelo bidimensional (NARLI et al., 2003). Ademais, o sistema está pronto para a integração com qualquer software CAD comercial, onde é possível utilizar as ferramentas sofisticadas destes para desenho de superfícies (por

exemplo, a ferramenta conhecida como “loft”, que gera uma superfície B-Spline através da transição de perfis). Neste momento, a escolha da B-Spline também é justificada, pois estes softwares já possuem esta formulação de curva implementada em sua palheta de desenho.

Em softwares comerciais mais especializados, existem pacotes projetados para o desenho naval, onde se consegue avaliar a qualidade da superfície de forma gráfica, determinar parâmetros hidrostáticos ou hidrodinâmicos através de simulações com CFD. Desta maneira, o projeto do casco como um todo se torna bastante flexível e poderoso, uma vez que interage com os demais blocos de ferramentas de projeto de uma embarcação, aproveitando o que já existe de consagrado em termos de aplicativos de CAD.

Para efeito de demonstração e validação da ferramenta, será utilizado como exemplo o veleiro projetado didaticamente em (LARSSON et al., 1994), chamado de YD-40, cujos dados de projetos, curvas características, dimensões estão disponíveis. No final, é possível realizar uma comparação entre a forma e propriedades geométricas do casco do veleiro YD-40 e o casco desenhado pela ferramenta. Espera-se que o casco gerado seja visualmente semelhante ao casco do veleiro YD-40, e que possua as mesmas propriedades físicas, já que suas dimensões principais serão as mesmas.

As dimensões do YD-40 utilizadas nesta modelagem são:

Tabela 4. Dimensões do veleiro YD-40

L_{OA} [m]	L_{WL} [m]	B_{WL} [m]	B_{MAX} [m]	T_C [m]	C_P	∇c [m ³]	x_{LCB} [m]	A_{WP} [m ²]
12,05	10,02	3,17	3,71	0,57	0,56	7,632	5,36	22,61

A borda livre FB adotada foi de 1,3 m, aproximadamente a medida utilizada em veleiros semelhantes de 40 pés. Para o veleiro YD-40, os parâmetros principais do casco, expostos na Tabela 4, foram previamente calculados pelo projetista. Ou seja, a definição destes parâmetros passa obrigatoriamente por um processo anterior, no qual se definem um perfil de missão para a embarcação, suas características

principais, e em seguida, alguns cálculos devem ser feitos para se chegar à definição destas grandezas.

5.2 Curvas paramétricas do casco

A seguir listaremos as curvas utilizadas na modelagem paramétrica do casco. Ao lado de cada curva estão os parâmetros físicos do casco necessário para a parametrização da curva. A coluna “Dados Obtidos” mostra que informações esta curva provê para a próxima curva ou para o desenho do casco.

Tabela 5. Conjunto de curvas paramétricas do casco

#	Curva	Parâmetros necessários	Dados Obtidos
1	Curva de Áreas Seccionais	$L_{WL}; \nabla c; LCB; C_P;$	A_{BM}, A_{Bi}, x_{BM}
2	Curva da Baliza Maior	$A_{BM}; B_{MAX}; B_{WL}; FB$	T_{CBM}
3	Curva da Linha D'água	$L_{WL}; x_{BM}; B_{WL}; A_{WP}$	B_{WLi}
4	Curva da Linha da Quilha	$L_{WL}; x_{BM}; T_{CBM}; LOA$	T_{CBi}
5	Curva da Linha de Borda	$LOA; B_{MAX}; B_{POPA}; x_{BM}$	B_{MAXBi}
6	Plano de Balizas	$B_{MAXBi}; B_{WL}; A_{Bi}; T_{CBi}; FB$	Casco

Onde:

A_{BM} – Área da baliza maior (seção mestra) = $(V/L_{WL} * C_P)$

A_{Bi} – Área da baliza i

x_{BM} – coordenada da baliza maior

T_{CBM} – Calado da baliza maior

B_{WLi} – Boca na linha d'água da baliza i

T_{CBi} – Calado da baliza i

B_{POPA} – Boca na baliza da popa

B_{MAXBi} – Boca máxima da baliza i

Assim, cada curva é desenhada submetida ao algoritmo de otimização a partir de dados de entradas de projeto do casco previamente estabelecidos, seja como um ponto de partida, seja como valor de uma iteração anterior. Ao se desenhar determinada curva, esta fornece uma informação para o desenho da próxima curva, um parâmetro geométrico ou uma grandeza física. Os parâmetros obtidos de cada curva são armazenados, e ao final, compõe o desenho do casco completo, totalmente parametrizado e modelado por B-Splines.

No entanto nem todas as curvas, dependerão apenas da curva anterior. O fluxo funciona como segue (Figura 30): a Curva 1 recebe os parâmetros de entrada. A Curva 2 recebe da Curva 1 o parâmetro A_{BM} . A Curva 3 utiliza o parâmetro x_{BM} da Curva 1, mas não utiliza nenhum parâmetro da Curva 2. A Curva 4 recebe o parâmetro x_{BM} da Curva 1 e T_{CBM} da Curva 2, mas independe da Curva 3. A Curva 5 recebe x_{BM} da Curva 1 e independe das demais anteriores. Finalmente as Curvas 6 recebe A_{Bi} da Curva 1, B_{WLi} da Curva 3, T_{CBi} da Curva 4 e B_{MAXBi} da Curva 5.

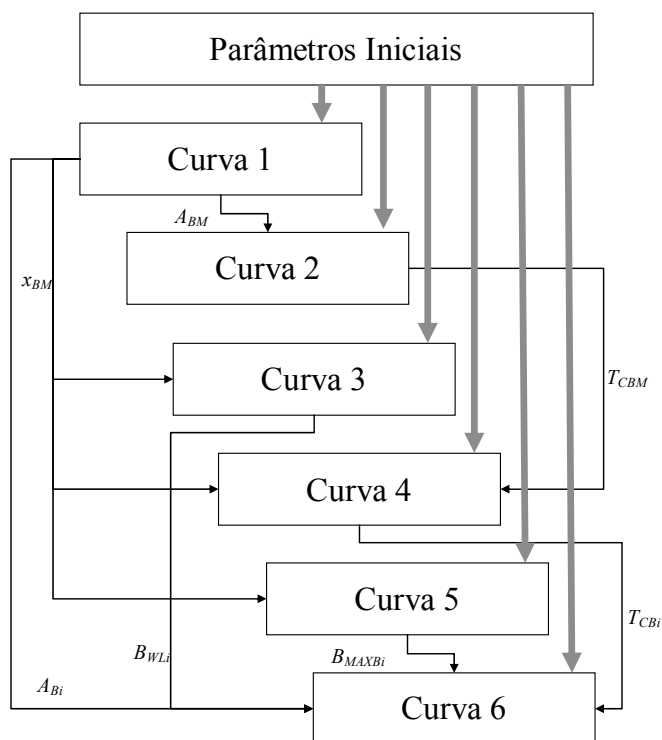


Figura 30. Fluxograma da parametrização das curvas

Como exemplo, vamos citar o desenho da curva de áreas seccionais do casco, cujo problema de otimização detalhado será apresentado a seguir, no capítulo de Resultados. A curva de áreas seccionais nos dá o volume deslocado pelo casco, que multiplicado pela densidade da água, nos dá o peso do total do veleiro. O volume de deslocamento deve ser um valor especificado pelo projetista, na pré-concepção do veleiro.

Na Figura 31, vê-se a curva inicial, com todas as variáveis de projeto (pontos de controle da curva) iguais a 1 (pontos de partida). A curva gerada pelo algoritmo atende a todos os parâmetros de projeto e submetida ao critério de suavidade. As linhas numeradas verticais em preto são os cortes das balizas. O valor da coordenada y no gráfico nos dá a área de cada baliza. O ponto máximo da curva nos dá o valor da área da baliza maior (seção mestra). Como hipótese neste trabalho, será adotada que a boca máxima das balizas se dá na seção de maior área (seção mestra).

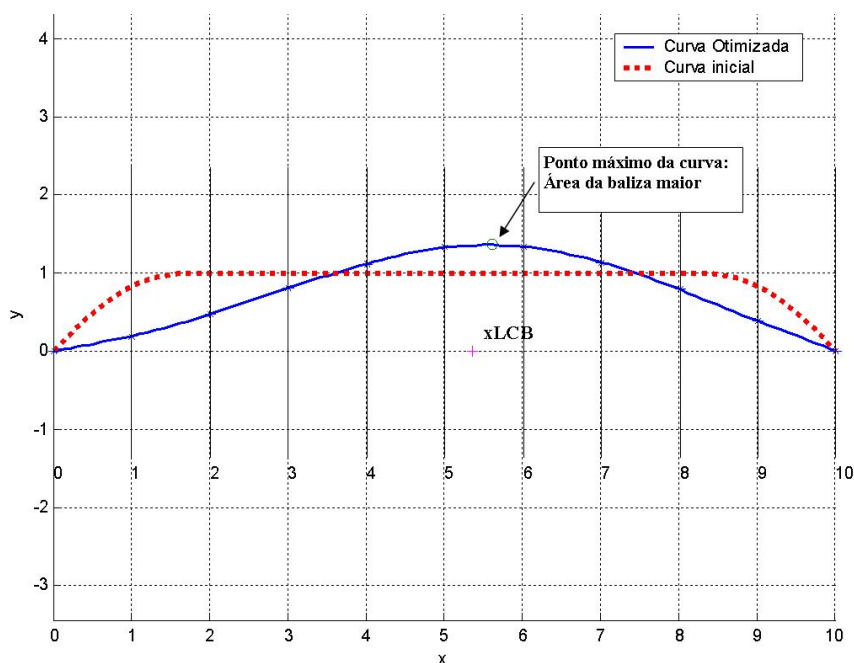


Figura 31. Curva de áreas seccionais YD-40

O resultado final do desenho da curva também está ligado à escolha do critério de carenagem ou a combinação ponderada deles, citado na seção 3.2. As Figuras 32, 33

e 34 mostram os diferentes resultados de uma otimização da curva de áreas seccionais, sujeitas às mesmas condições de contorno, porém com diferentes funções objetivo.

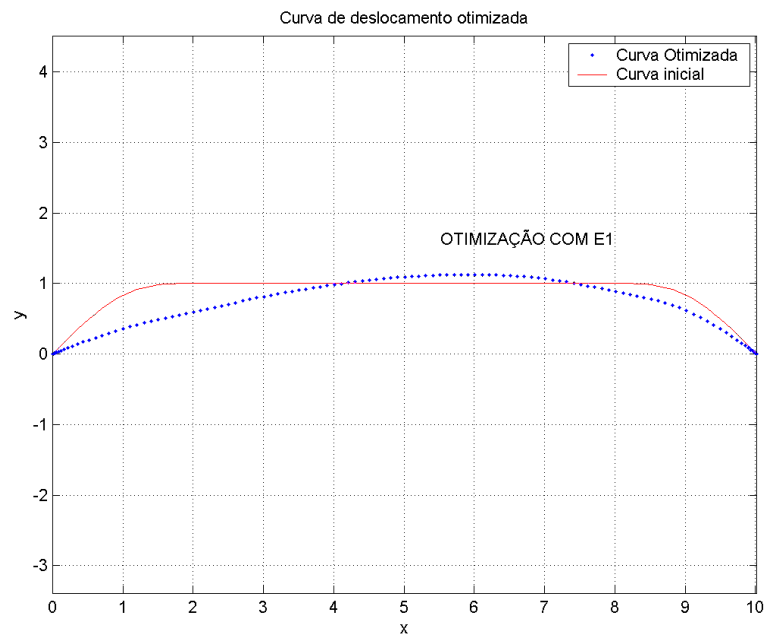


Figura 32. Resultado utilizando a Função Objetivo E_1

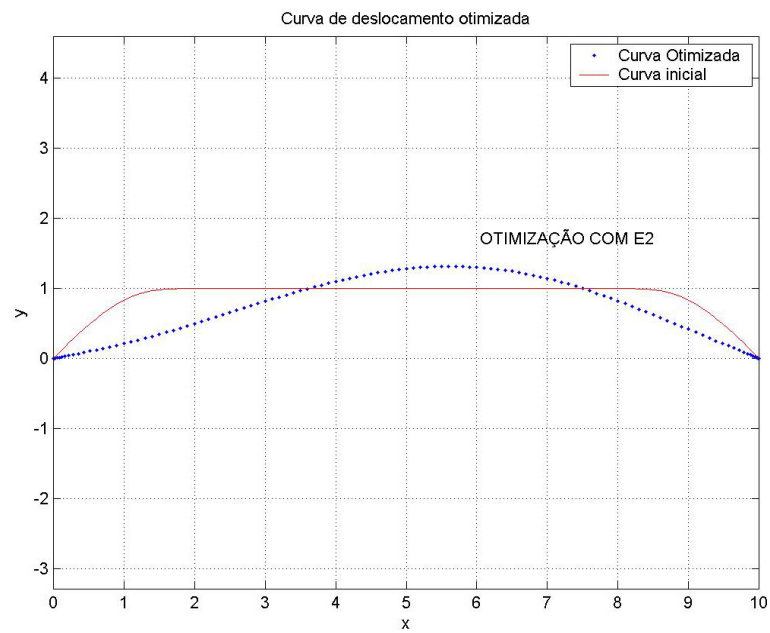


Figura 33. Resultado utilizando a Função Objetivo E_2

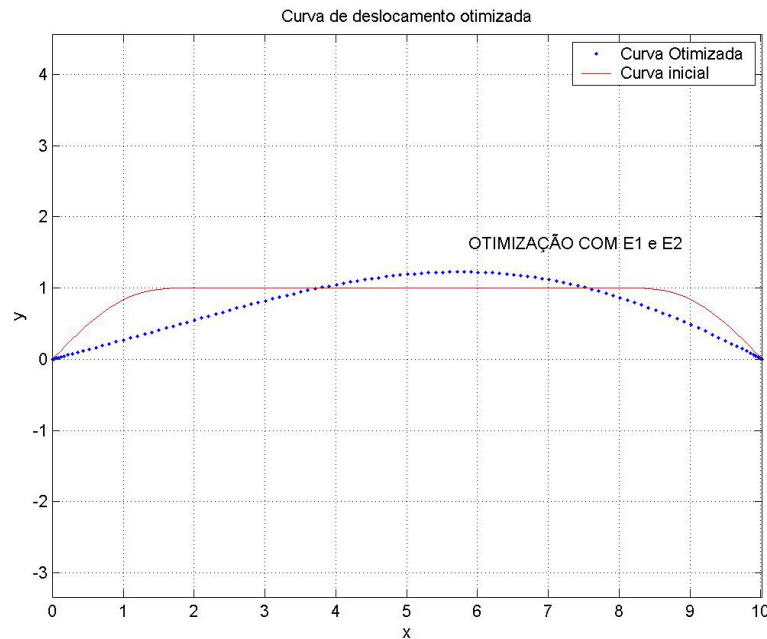


Figura 34. Resultado utilizando a Função Objetivo $E_1 + E_2$.

Na literatura, não existe justificativa física ou matemática ou determinação para a escolha de um ou de outro critério. Em (HARRIES et al., 98) e (HARRIES et al., 99) é sempre utilizada a função E_2 (medida de flexão). Outra observação é que por se tratar de medidas diferentes, a ordem de grandeza das funções E_1 , E_2 e E_3 não são necessariamente às mesmas. Logo, a escolha de uma função multi-objetivo, combinando estas duas funções deve levar em conta este fato, ou recorrer a algum tipo de normalização.

5.3 Implementação do software

O software desenvolvido neste trabalho, em ambiente Matlab, envolve elaboração de modelos matemáticos e métodos numéricos. Um software para fins científicos deve necessariamente possuir as seguintes características:

1. ser matematicamente correto
2. prover eficiência (velocidade e utilização de memória)
3. ser de fácil manutenção e ampliação

Destaca-se o terceiro item como de maior importância, devido à complexidade de parâmetros e configurações do programa. Softwares atualmente existentes no mercado, como por exemplo, FRIENDSHIP-Modeler, desenvolvido pela FRIENDSHIP Systems, nos mostram que os usuários destes softwares devem ser fortemente especializados no assunto, pois mesmo para os casos mais simples, exige-se que o operador programe alguns parâmetros de entrada para o sistema.

Logo, devido à grande variedade de modelos existentes, a ferramenta desenvolvida também exigirá do usuário algumas linhas de programação para diferentes aplicações.

Por outro lado, a grande vantagem é a adaptabilidade com diferentes interfaces, seja esta, por exemplo, um software de CFD ou CAD. Na versão atual, o programa é capaz de exportar as curvas geradas, já posicionadas corretamente no espaço tridimensional, para um programa CAD através de um arquivo no formato DXF.

Além disso, a integração com os demais blocos de um processo de síntese de casco por otimização com modelos hidrodinâmicos tipo CFD, já é prevista e possível de se implementar.

5.4 Rotinas utilizadas em ambiente Matlab

Um dos critérios que levou à escolha do software Matlab para implementação desta ferramenta foi seu poderoso conjunto de rotinas e comandos numéricos. O cálculo das equações das funções de carenagem, por exemplo, são feitos através de rotinas do próprio Matlab para integração numérica e cálculo de áreas de curvas.

Com isso, se ganha muito em termos de tempo e desempenho do método de otimização, já que não é necessário utilizar ou programar novos algoritmos de integração numérica (como o método de Simpson, largamente utilizado na literatura da área), ao contrário de se fosse escolhido uma linguagem de programação em C++, por exemplo. O desenho das curvas, no plano ou em 3D também é bem simples e

eficiente em ambiente Matlab, não havendo necessidade de se construir a interface gráfica com o usuário.

As rotinas para traçado e manipulação das curvas B-Spline, no entanto, exigiram programação própria. As rotinas existentes no Matlab, não atendiam a necessidade de implementação das curvas. Funções para o traçado de curvas B-Spline cúbicas uniformes foram escritas, bem como, funções que calculam suas derivadas obtidas algebricamente, para melhor precisão de cálculo. De forma simplificada, o processo de otimização das curvas é ilustrado no fluxograma da Figura 35.

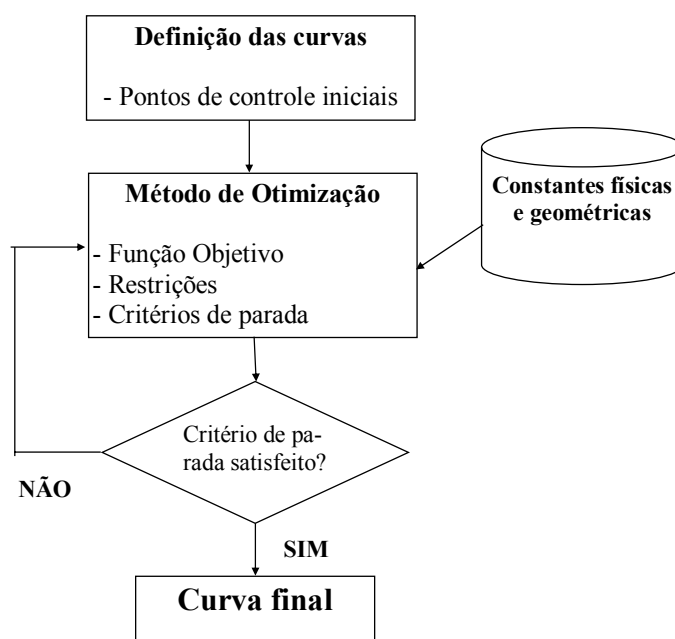


Figura 35. Diagrama de blocos da estrutura do processo de otimização

Em (HARRIES et al., 1998), as curvas B-Splines são implementadas com um vetor de nós uniforme, com oito nós (valor considerado razoável). Para a interpolação da superfície, oito curvas B-Splines são desenhadas. Neste trabalho, utilizam-se também B-Splines uniformes, porém com dez nós. Isso apenas por motivos de arredondamento e para se ganhar mais graus de liberdade para as curvas, já que este aumento não oferece um custo computacional expressivo. Seguindo a forma mais comum, utilizada em (LARSSON et al., 1994), o casco é modelado através de 11 curvas (balizas 0 a 10).

6 RESULTADOS

6.1 Aplicação e desempenho do algoritmo

Conforme abordado na seção 3.5, o algoritmo utilizado para a resolução do problema de otimização é implementado através da rotina *fmincon*, do Matlab. Para obter-se uma medida da eficiência do algoritmo e da modelagem do problema proposto, far-se-á a mesma comparação apresentada em (NARLI et al., 2003), onde o autor interpola uma curva B-spline a partir de uma curva plana de um casco, obtido da fórmula de Wigley (WIGLEY, 1934). Este casco possuiu uma superfície intrinsecamente suave, e, portanto, linhas d'água e balizas também suave. A Figura 36 abaixo ilustra, à esquerda, a construção de uma curva com o equacionamento de Wigley e com as proporções aproximadas de uma baliza do veleiro YD-40. À direita, redesenhamos a mesma com a formulação de um B-Spline cúbica uniforme.

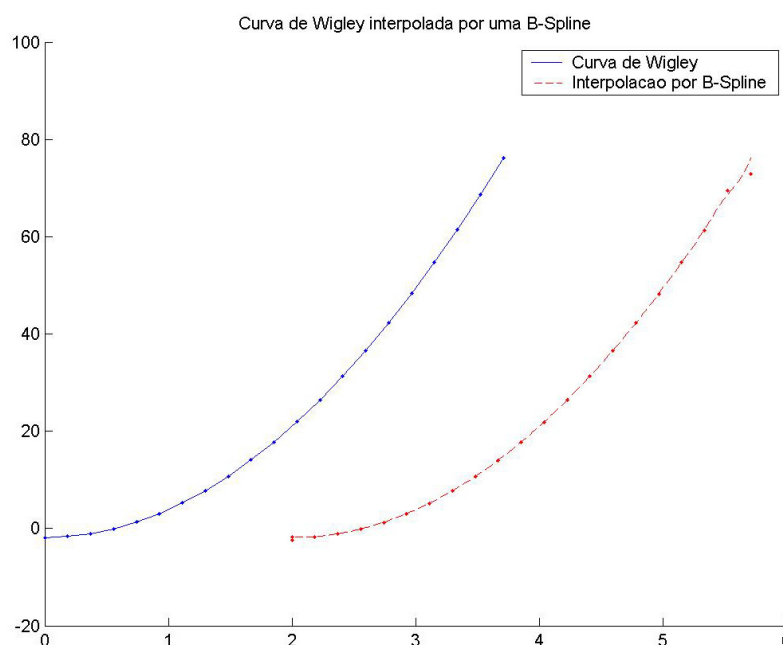


Figura 36. Curva de uma seção do casco de Wigley e interpolação por uma B-Spline

A comparação feita por Narli é a seguinte: após interpolar a curva original por uma B-Spline, aplica-se uma distorção aleatória distribuída ao longo da curva. A partir da

curva distorcida, aplica-se o algoritmo de otimização, e da curva obtida compara-se as curvaturas, constatando-se a melhoria da suavidade da curva. A curva obtida deve ser semelhante geometricamente à curva original, o que significa que o desejável é ter o maior ganho em termos de suavização para um pequeno desvio dos pontos de controle. A função objetivo utilizada por Narli, que chamaremos de E_c é a área sob a curva do quadrado da curvatura, expressa pela integral abaixo:

$$E_c(s) = \int C(s)^2 ds \quad (63)$$

onde s é o comprimento do arco da curva. A função $C(s)$ é a função da Equação (41). Note que, neste caso, o autor utiliza a curvatura como função objetivo e não como restrição. Abaixo discutiremos melhor esta diferença baseando-se nos gráficos obtidos. Em (FARIN, 1990), E_c é chamada de energia de deformação (do inglês, “strain energy”).

Para os testes no algoritmo elaborado, buscar-se-á comparar a diferença dos resultados quanto ao número de pontos de controle, utilização da curvatura na função objetivo e desempenho do algoritmo quanto à convergência e número de iterações.

As premissas e restrições adotadas para os testes foram:

- a. Segundo já explicado, os pontos de controle somente possuirão 1 grau de liberdade, podendo apenas variar ao longo do eixo y .
- b. Conforme feito em (NARLI et al., 2003), a curva com a formulação de Wigley é distorcida aleatoriamente. No problema em questão, utilizou-se a função $rand()$ do Matlab, que gera um número aleatório de 0 a 1, que é multiplicado por um fator de 0,2 e em seguida multiplicado pela coordenada y do ponto de controle da curva.
- c. A restrição do problema de otimização é que os pontos de controle somente poderão variar em 10% de seu valor. Isso para que a nova curva seja semelhante em sua forma à curva original distorcida.
- d. Os pontos de controle inicial e final devem permanecer os mesmos (condição de contorno).

O problema de otimização implementado para os testes do algoritmo está apresentado na Figura 37. Para algumas simulações, a função objetivo é alterada, conforme explicado adiante. Para cada simulação, a curva é novamente distorcida aleatoriamente. Isso foi escolhido para deixar os testes o mais genérico possível.

Otimização da curva distorcida

minimizar $E_1 + E_2$

P_{yi}

tal que:

$$|P_{yi} - P_{yi}^*| \leq 10\%$$

Valores iniciais: pontos de controle da curva distorcida

Figura 37. Problema de otimização para o teste do algoritmo

Como resultado, obteve-se que 100% das simulações convergiram para uma solução. Em relação à variação ao número de pontos de controle, pode-se observar pelos gráficos das Figuras 38 e 39, que o crescimento do esforço computacional aumenta exponencialmente com o número de pontos de controle da curva. Nota-se que para um mesmo número de pontos de controle, pode ocorrer de haver mais de um resultado. Isso porque, simulou-se mais de uma vez o mesmo número de pontos de controle, porém a curva inicial distorcida variava aleatoriamente.

Pelas simulações feitas, constatou-se que o número de pontos de controle influi pouco na qualidade do resultado. Ou seja, não há uma relação entre o número de pontos de controle e a maior suavidade da curva, tendo como referência a curva com a formulação de Wigley. Obviamente que a curva distorcida com maior número de pontos de controle apresenta uma maior energia, já que há mais oscilações na curva, e por isso, a taxa de redução do valor das energias é muito maior.

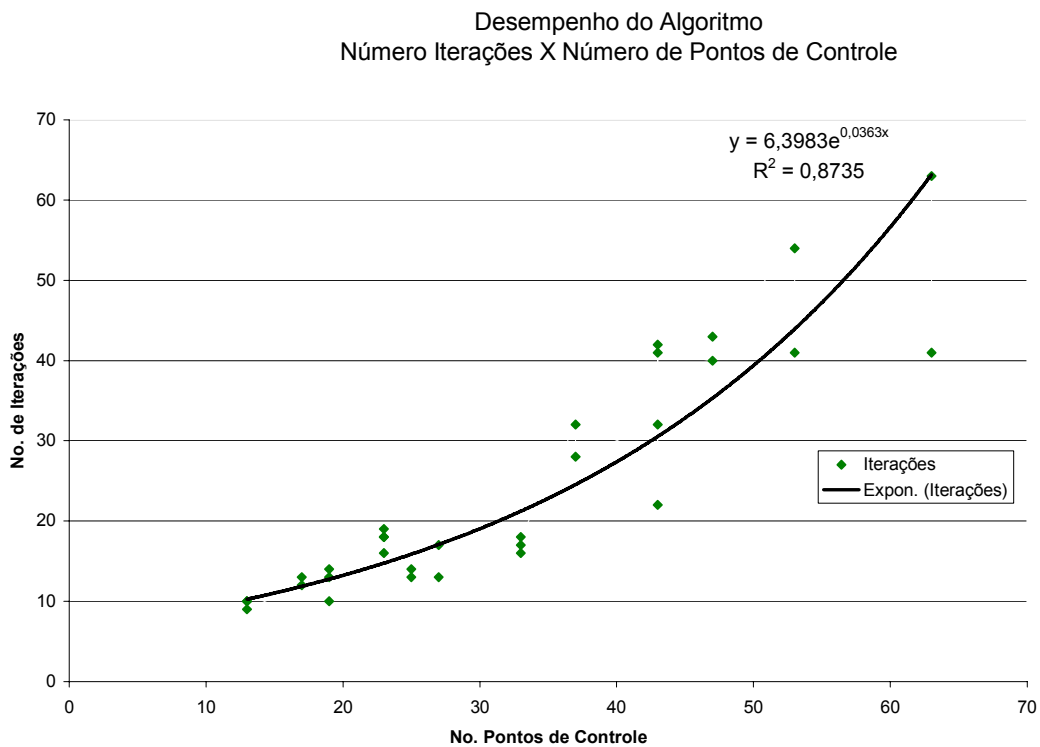


Figura 38. Gráfico do número de iterações versus número de pontos de controle

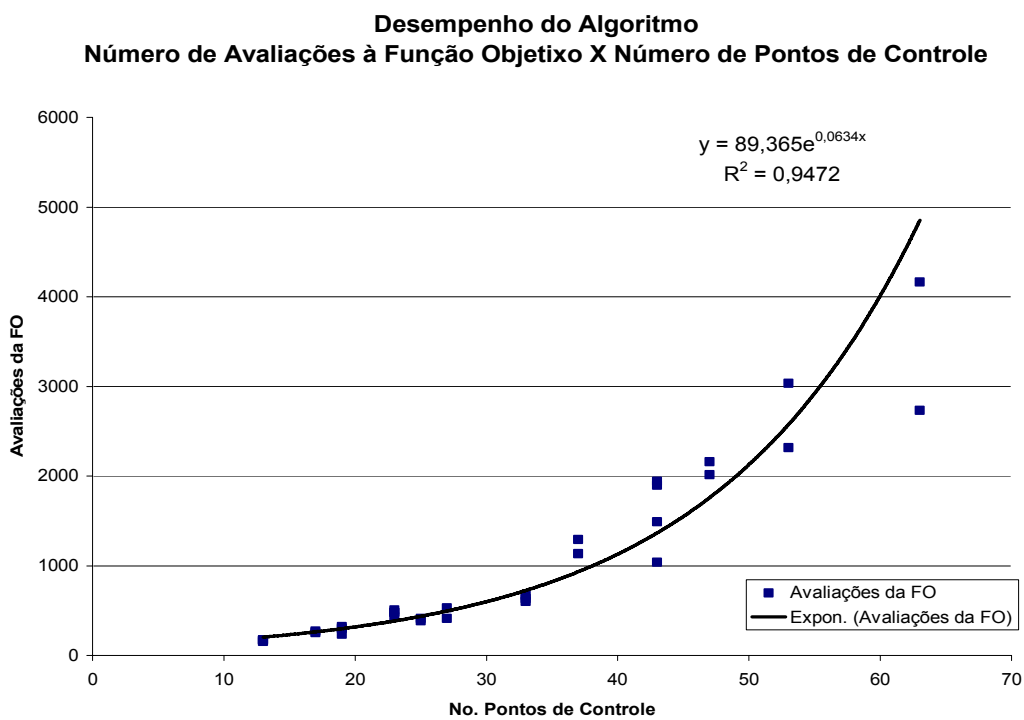


Figura 39. Gráfico de avaliações da função versus número de pontos de controle

Mas esta é uma característica pouco relevante para o caso prático, onde não haverá uma curva distorcida propositadamente. A conclusão que se chega é que poucos pontos de controle, no caso da síntese de curvas por otimização, implica em um custo computacional baixo e benefício satisfatório, em termos de suavidade da curva. Em aplicações de interpolação de curvas ou pontos, no entanto, um maior número de pontos de controle pode ser mais eficiente. Comprova-se também a tese de que a variação dos pontos de controle somente em uma direção produz resultados aceitáveis, simplificando-se matematicamente o modelo, a implementação do software e os custos computacionais.

A Figura 40 e a Figura 41 mostram os resultados obtidos com simulações com curvas com 43 e 11 pontos de controle. As curvas desenhadas na linha -100 do eixo y estão deslocadas para melhor visualização e representam as curvaturas. No gráfico, estão desenhadas também as respectivas energias E_1 e E_2 de cada curva. Conforme citado anteriormente, existe uma diferença de ordem de grandeza entre energias E_n , isso pode ser visto nos dois gráficos. O importante é notar que as curvas construídas com 43 e 11 pontos de controle apresentaram resultados semelhantes e igualmente satisfatórios.

Um aspecto que foi percebido durante as simulações foi que quanto maior o número de pontos de controle, mais a energia E_2 consegue ser minimizada, ao passo que com número de pontos de controle reduzidos, a energia E_1 é melhor otimizada. Isso pode ser explicado porque E_1 está relacionado com a forma da curva, e E_2 está relacionado com a curvatura. Logo, para vários pontos de controle (Figura 40), com pequenas variações se minimiza as irregularidades da curva, mantendo-se a forma geral. Em contrapartida, com poucos pontos de controle (Figura 41), a variação destes modifica a curva mais amplamente, alterando mais sua forma. Portanto, constata-se que existe uma relação entre os pesos de cada energia na função objetivo e o número de pontos de controle da curva.

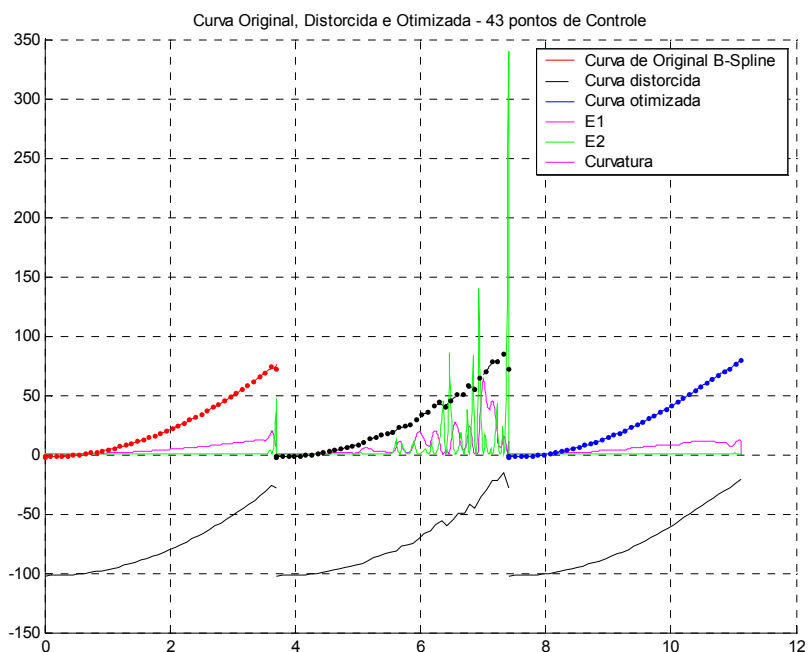


Figura 40. Simulação do algoritmo com uma B-Spline com 43 pontos de controle

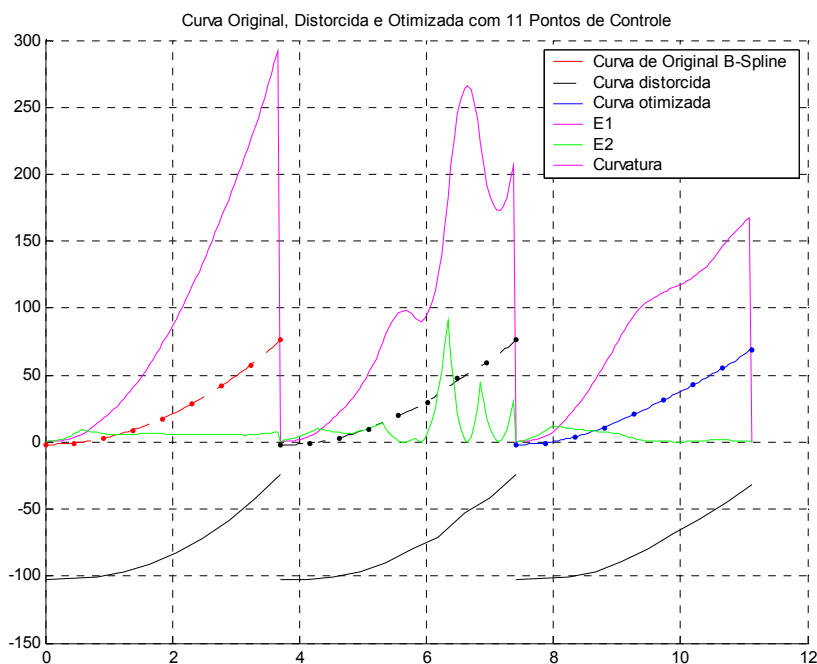


Figura 41. Simulação do algoritmo com uma B-Spline com 11 pontos de controle

Passando a analisar a função objetivo, compararam-se os resultados obtidos utilizando a combinação das energias com e sem a inclusão da curvatura na fórmula. Também se analisou resultados utilizando somente a curvatura como função objetivo. Após diversas simulações, os resultados foram que quando utilizada a

combinação das energias E_1 , E_2 e E_3 , a inclusão ou não de E_c na função objetivo gerava diferenças desprezíveis. Em seguida, utilizou-se a combinação ponderada E_1 com E_c , e E_2 com E_c . Os resultados obtidos foram que a energia E_2 e a E_c têm efeitos semelhantes para o método, ou seja, que uma está relacionada à outra. Isto faz sentido, pois já mencionamos que a energia E_2 (medida de flexão) é mesmo uma medida da curvatura da curva. No entanto, cabe ressaltar que a curvatura e a medida de flexão são conceitos diferentes. A curvatura independe da formulação de curva utilizada, pois é uma propriedade da curva em si.

Lembrando novamente o que foi citado na seção 5.2, onde se faz referência a (HARRIES et al., 98) e (HARRIES et al., 99). Estes utilizam somente a energia E_2 como função objetivo, ao passo que (NARLI et al., 2003) utiliza somente a integral da curvatura, E_c . Neste trabalho, no entanto, utilizar-se-á a ponderação das energias E_1 , E_2 e E_3 , com os fatores de ponderação determinados empiricamente para cada caso, já que constatou-se que desta forma é possível dispor de mais recursos para a síntese de curvas, ou seja, caso queiramos suavizar a forma de uma curva, podemos também atribuir um peso não nulo à E_1 .

Para a medida de torção, E_3 , na maioria das simulações feitas, é atribuído um peso nulo. Na literatura, pouco se encontra trabalhos que citam ou utilizam a medida de torção para a otimização de curvas planas, já que a torção é mais aplicável a curvas tridimensionais. Após várias tentativas, as simulações realizadas com a ferramenta implementada mostraram que, em alguns casos, a energia E_3 apresenta resultados muito semelhantes a E_2 , também mais influentes na curvatura e suavidade da curva, que em sua forma, como ocorre com E_1 . A Figura 42 mostra um gráfico com o detalhe das curvas das três energias. A partir desta, é possível notar que as funções de energia não são simplesmente a derivada uma da outra, por exemplo, E_2 ser igual a derivada de E_1 . Mas, conforme a Equação (32), os termos estão elevados ao quadrado.

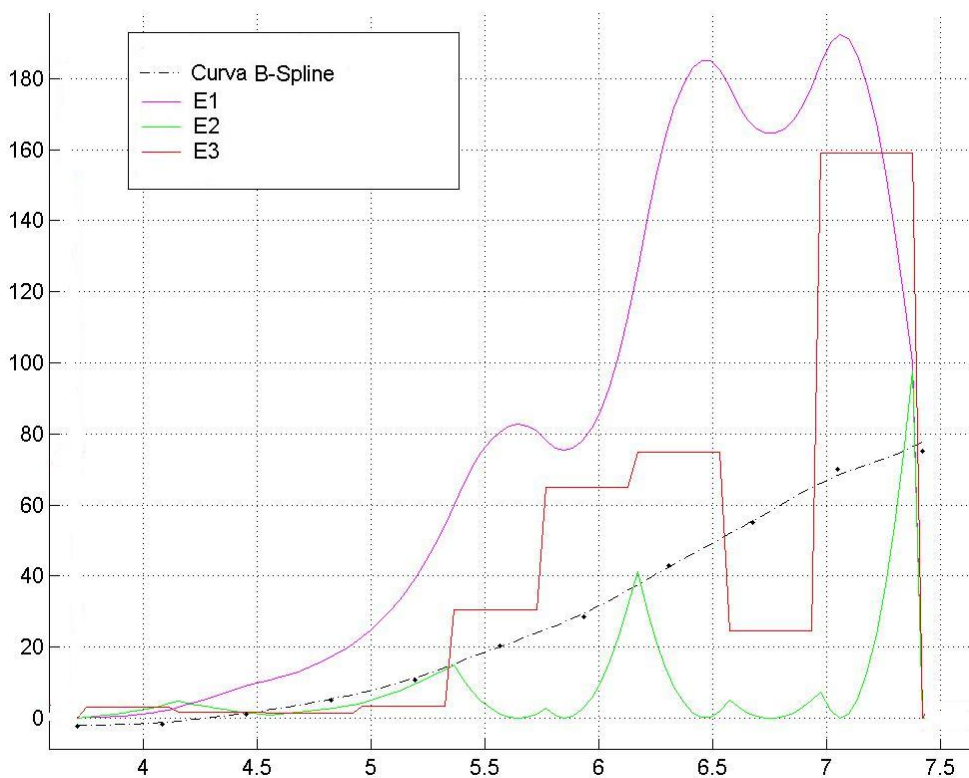


Figura 42. Detalhe das curvas com as energias E_1 , E_2 , e E_3

Para averiguar por outro caminho a conclusão de que E_2 e E_c produzem o mesmo resultado, foi feita uma simulação onde primeiro se otimizava a curva segundo a função objetivo com a combinação das energias (terceiro gráfico da esquerda para a direita da Figura 43). Em seguida, da curva resultante se aplicava novamente o algoritmo, agora com uma nova função objetivo igual à Equação (63) (gráfico à direita da Figura 43). O resultado foi que não se nota diferença entre as curvas obtidas, e os pontos de controle de uma otimização para a outra variavam minimamente.

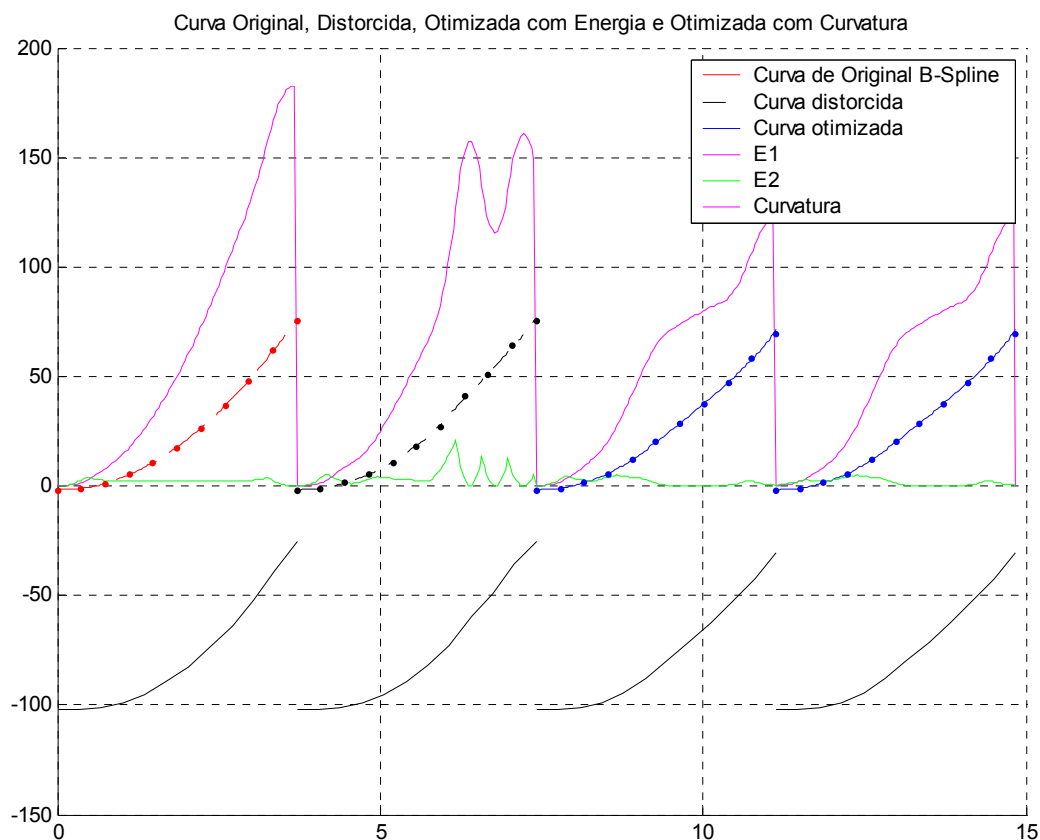


Figura 43. Simulação com duas otimizações: energia e, em seguida, curvatura.

Para demonstrar uma medida quantitativa da eficiência do algoritmo, a Tabela 6 apresenta os resultados numéricos dos valores calculados pelo método, neste teste comparativo com a curva com a formulação de Wigley, conforme o problema de otimização apresentado na Figura 37. Porém, a função objetivo utilizada é a Equação (33), com e_1 , e_2 e e_3 iguais a 1. A Figura 44 mostra os gráficos das curvas obtidas para esta simulação. Nota-se que em relação à curva distorcida, o ponto de partida do algoritmo, a melhoria da suavidade mostrada no gráfico é comprovada pelos resultados numéricos. Em relação à curva com a formulação de Wigley, curva já bastante suave, também se obtém uma pequena melhora.

Tabela 6. Saídas do programa para o teste comparativo com as curvas

	E_1	E_2	E_3	E_c
Curva com a formulação de Wigley	243,45	8,2173	3,5778	0,1398
Curva Distorcida	291,469	27,4533	141,4394	0,0869
Curva Otimizada	240,859	7,8289	1,6732	0,0102
Melhoria em relação à curva distorcida	17%	71%	99%	88%
Melhoria em relação à curva de original	1%	5%	53%	93%

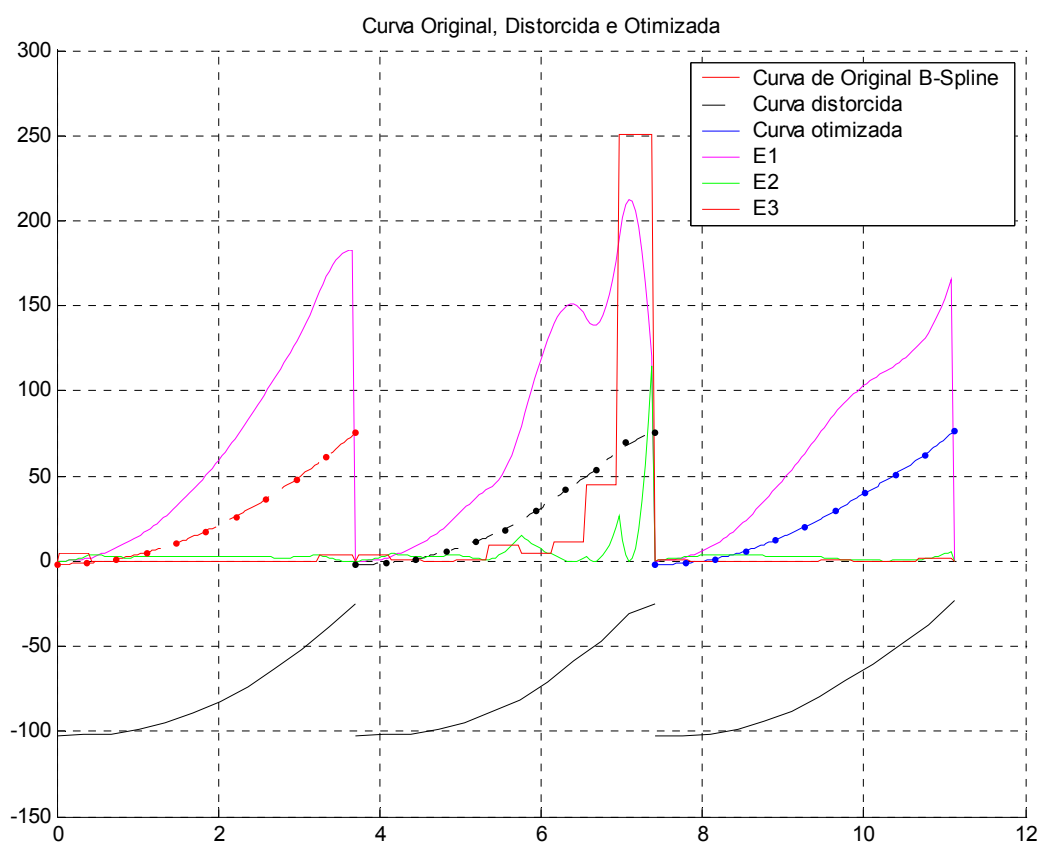


Figura 44. Gráfico obtido para os valores da Tabela 6

A Figura 45 mostra uma análise feita em um software de CAD, mostrando visualmente a curvatura ao longo das curvas (respeitando a ordem da esquerda para direita: curva original, distorcida e otimizada). Percebe-se uma diferença na curva otimizada, que possui as menores amplitudes do gráfico. Os gráficos da curvatura podem ser apresentados ao projetista, ajudando-o a identificar regiões não-suaves (FARIN, 1990). Este procedimento pode ser visto como uma versão

computadorizada do método tradicional de suavização manual, feito com sucesso com tiras flexíveis de madeira e pesos em projetos do século XVII, antes de estarem disponíveis os computadores (NARLI et al., 2003).

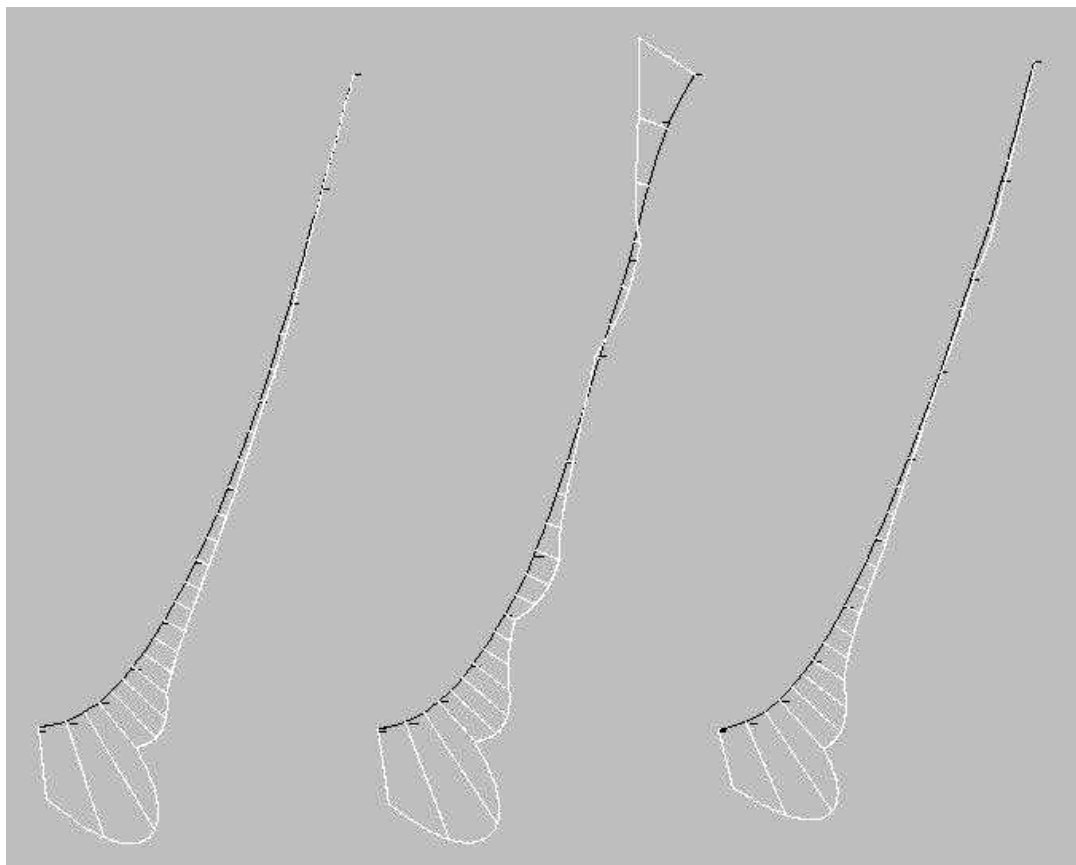


Figura 45. Análise de curvatura em CAD: curva original, distorcida e otimizada

Por último, abaixo na Figura 46, pode-se ver uma simulação cuja solução ótima foi encontrada após 13 iterações. A linha tracejada mostra a curva original distorcida. A linha cheia mostra a curva final otimizada. Os pontos na figura mostram os resultados parciais do método de otimização. Neste caso, o problema de otimização é o mesmo mostrado na Figura 37, porém, com uma curva mais distorcida, permitiu-se que os pontos de controle pudessem variar em até 50% de seu valor original (restrição: $|P_{yi} - P_{yi}^*| \leq 50\%$) para permitir chegar-se em uma curva final suave. O gráfico com o histórico de convergência da função objetivo ($E_1 + E_2$) está apresentado na Figura 47, onde o eixo y corresponde aos valores da função objetivo normalizada para o intervalo entre 0 (valor ótimo) e 1 (valor inicial).

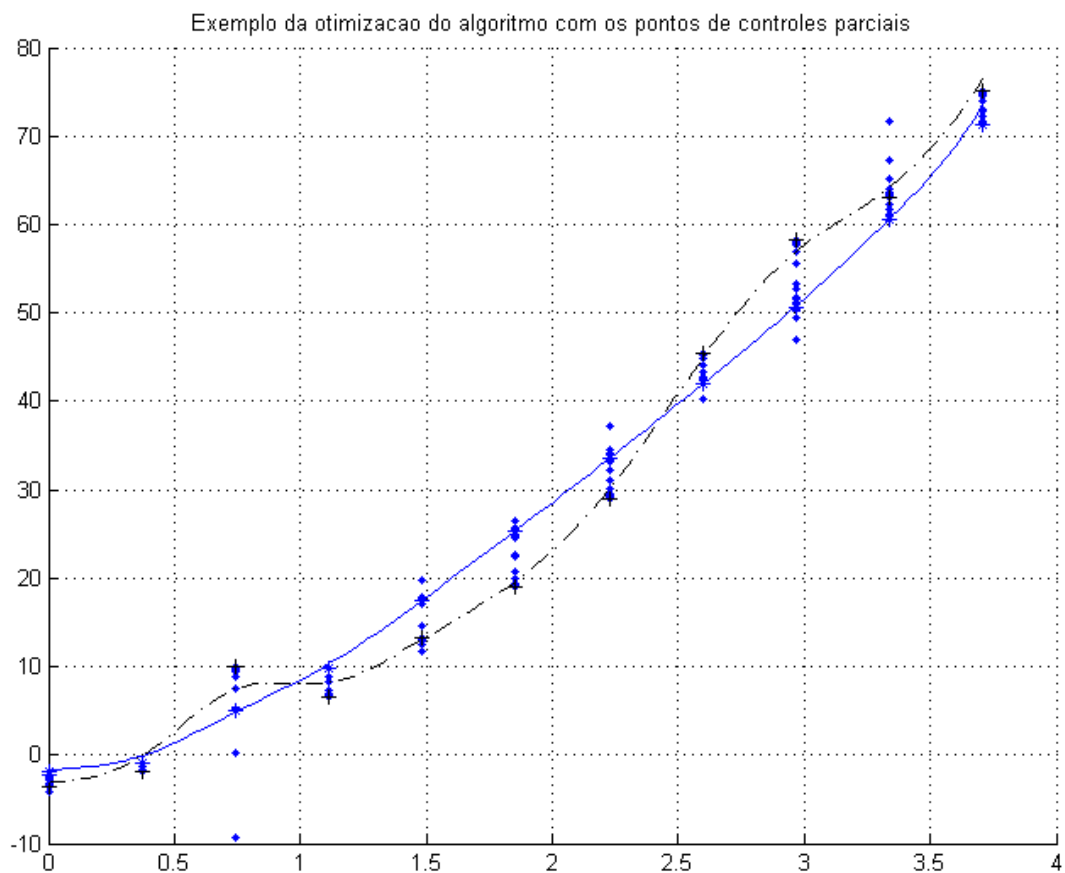


Figura 46. Gráfico com os resultados parciais do método de otimização

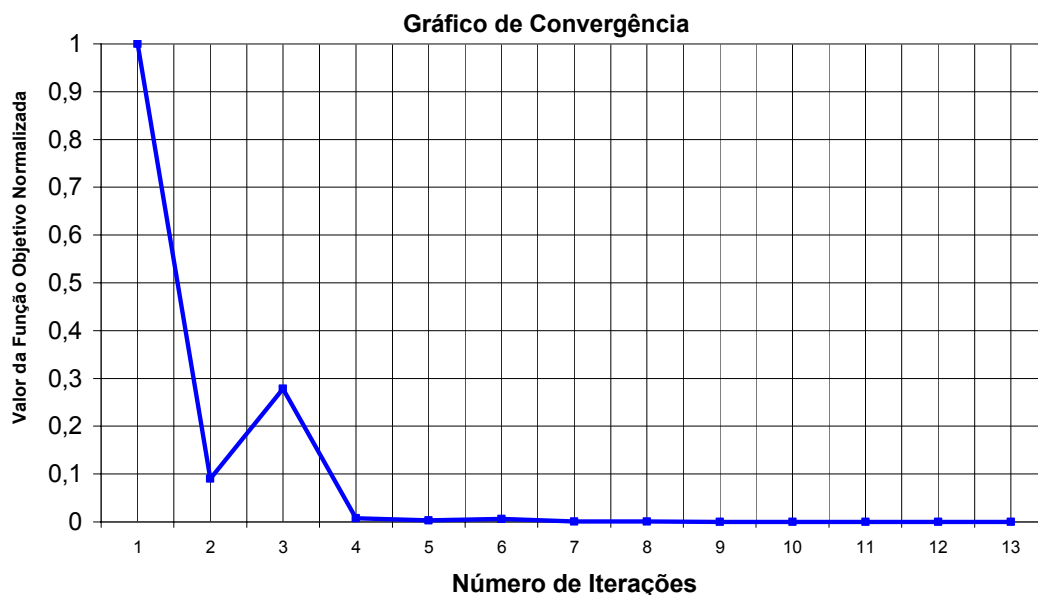


Figura 47. Histórico de convergência para a função objetivo

6.2 Modelagem dos parâmetros do veleiro YD-40

Conforme já adiantado na seção 5.1, como demonstração do método, foi feita a modelagem do veleiro projetado em (LARSSON et al., 1994). O produto final da ferramenta desenvolvida neste trabalho segue o fluxograma apresentado na Figura 7, e pode ser vista nas Figuras 48 e 49. Nestas figuras constam todas as curvas sintetizadas no software implementado, com exceção da curva de áreas seccionais. A curva da seção mestra pode ser vista entre as balizas 5 e 6, desenhada com uma linha tracejada.

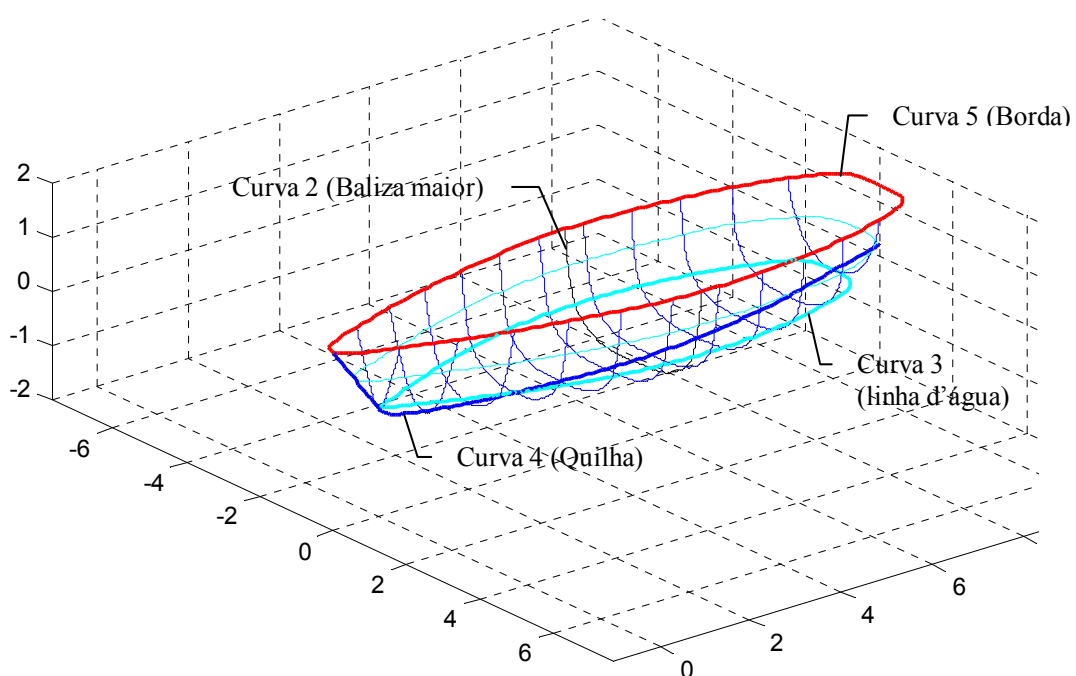


Figura 48. Vista isométrica do casco obtido

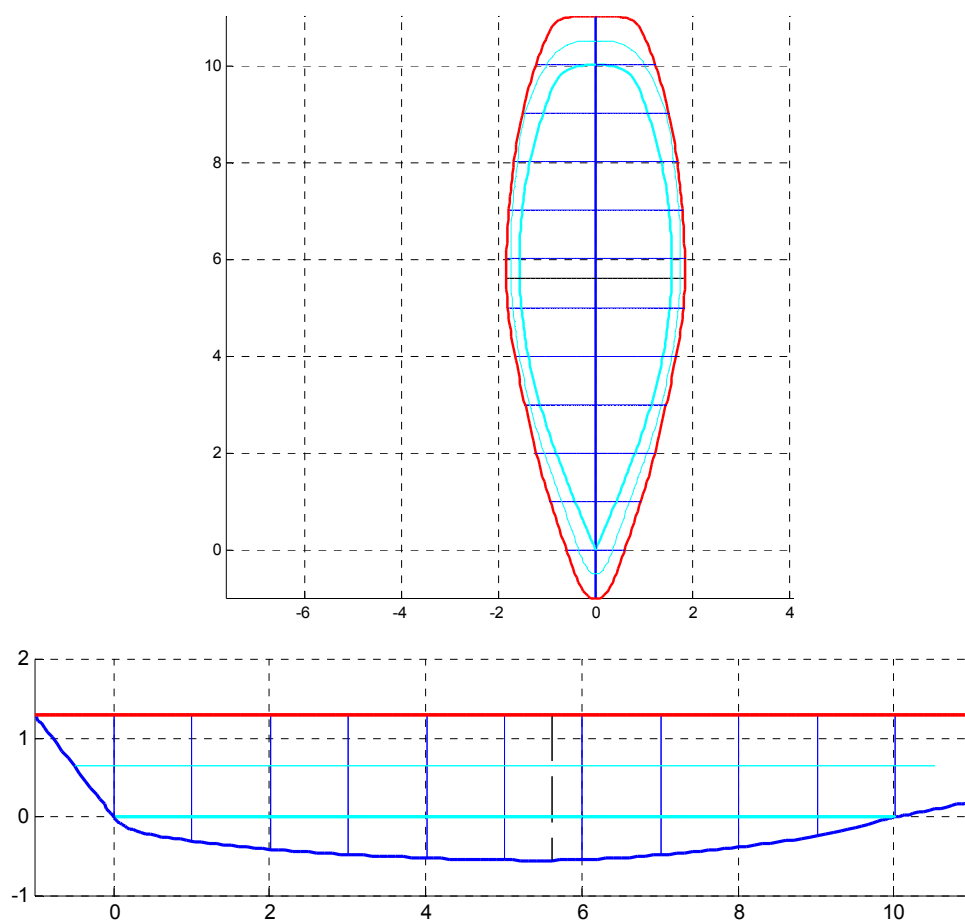


Figura 49. Vista superior e lateral do casco obtido

Os quadros a seguir apresentam individualmente as curvas obtidas em cada processo de otimização. Os gráficos das curvas de 1 a 5 (conforme a Tabela 5) mostram também a curva da qual se iniciou o processo de otimização, ou seja, o chute inicial. Na maioria das vezes, a fim de se utilizar uma curva inicial sem qualquer semelhança com a curva desejada (porém, uma curva ainda dentro do domínio viável do problema), utilizaram-se como ponto de partida os pontos de controle iguais a 1.

Para a maioria das curvas, utilizou-se 10 pontos de controle variáveis. Isso quer dizer que a curva possui sempre no total 2 pontos de controle a mais com multiplicidade 3 nas extremidades. Para algumas curvas, aumentou-se ou diminuiu o número de pontos de controle para adaptar-se ao número de seções da curva (por exemplo, a Curva 1 (área seccionais das balizas) com 9 seções e 9 pontos de controle variáveis, e Curva 5, com 12 seções e 12 pontos de controle variáveis).

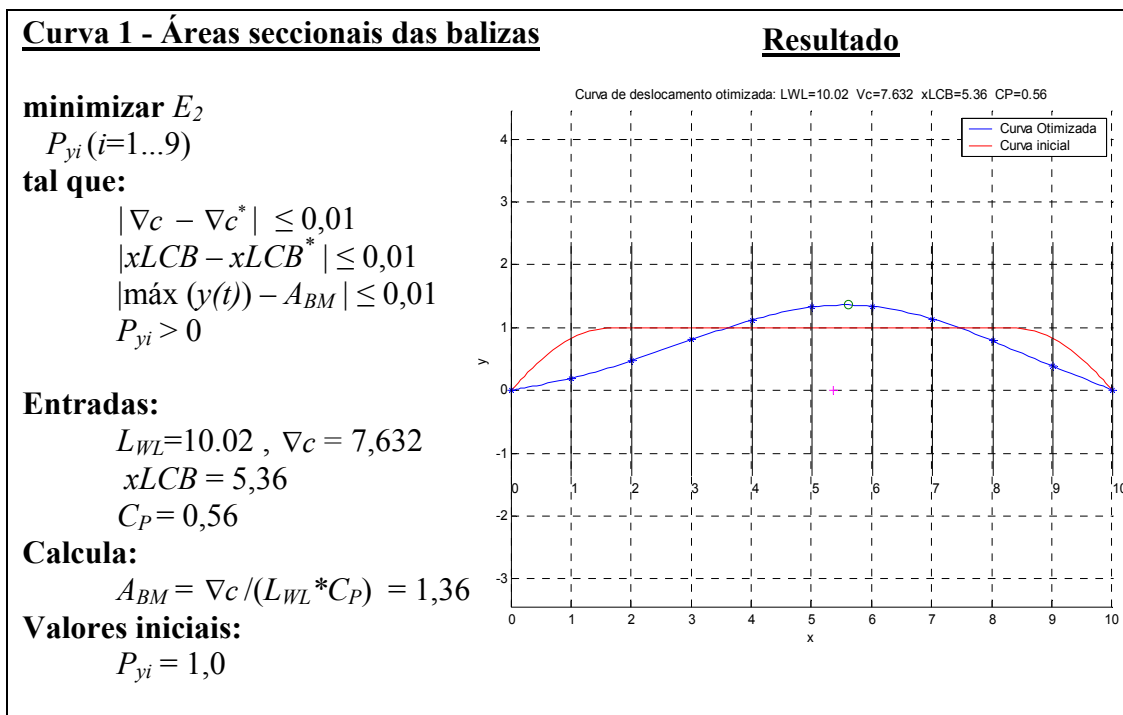


Figura 50. Problema de otimização da curva das áreas seccionais (Curva 1)

Para a Curva 2 (Figura 51), a otimização é feita para uma meia baliza invertida e utilizou-se uma curva inicial com pontos mais próximos dos pontos desejados. Isso porque, a restrição da B_{WL} é de difícil convergência. A função objetivo foi obtida empiricamente, impondo-se um peso maior à restrição E_2 , que está relacionada à uma curvatura suave.

Na Curva 3 (Figura 52), pode ocorrer que as bocas de balizas vizinhas sejam muito semelhantes. Neste caso deve-se adicionar uma restrição impondo uma curvatura mínima ou uma curvatura média mínima em determinado trecho da curva. Na simulação realizada, esta restrição não foi ativada.

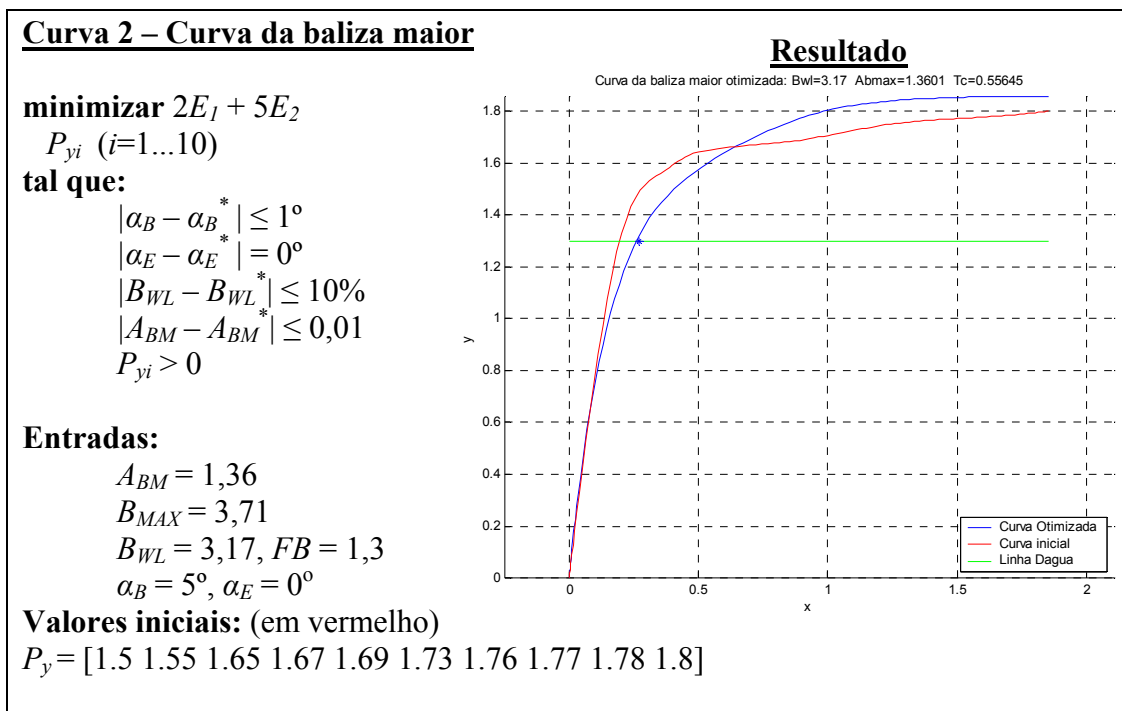


Figura 51. Problema de otimização da curva da baliza maior (Curva 2)

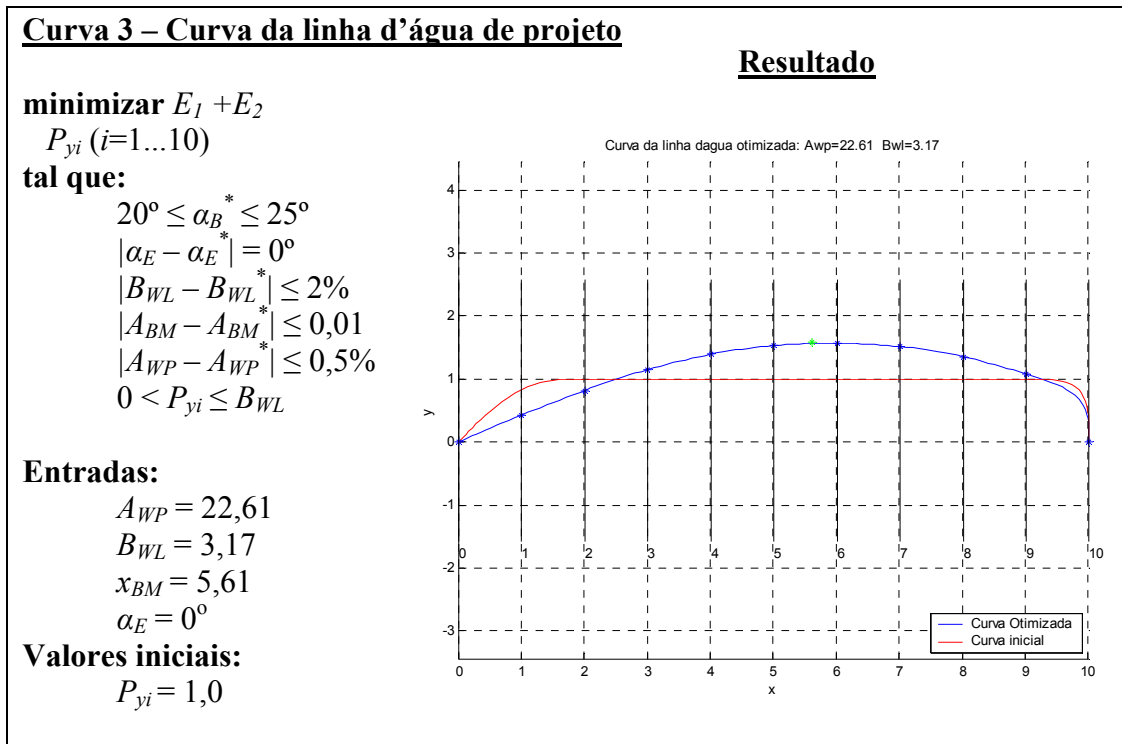


Figura 52. Problema de otimização da curva da linha d'água (Curva 3)

Para a Curva 4 (Figura 53), as linhas da proa e popa, são retas com a mesma inclinação dos ângulos de entrada e saída da linha da quilha. Estas não fazem parte do processo de otimização. Para contornar o problema de a linha da quilha ser muito aplainada, foi imposta uma curvatura média mínima (C_{MED}) para todo o trecho da curva. Com isso, é gerada uma linha mais arredondada, e sem regiões planas. A plotagem da curva foi invertida (pontos da curva com coordenadas y negativas) apenas para a melhor visualização. O processo de otimização utiliza as coordenadas positivas.

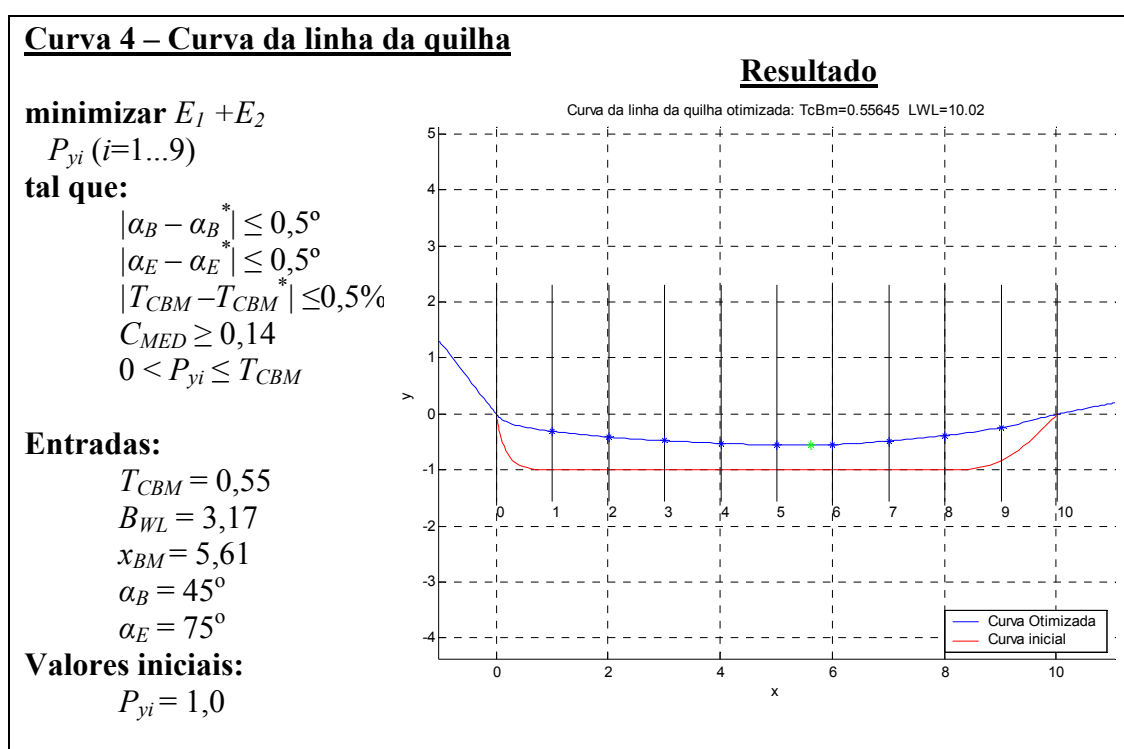


Figura 53. Problema de otimização da curva da quilha (Curva 4)

Para curva da linha de borda, Curva 5 (Figura 54), foram assumidos alguns valores para grandezas que mais estão relacionadas com a estética do casco do que com seu desempenho, já que ficam acima da linha d'água. Por exemplo, a distância entre o início da L_{OA} e o início da L_{WL} , chamada de d_{PROA} , foi adotada como sendo a metade da diferença entre os dois comprimentos. Além disso, foi necessário escolher um valor para a boca da parte traseira do casco (B_{POPA}). Este valor foi obtido empiricamente.

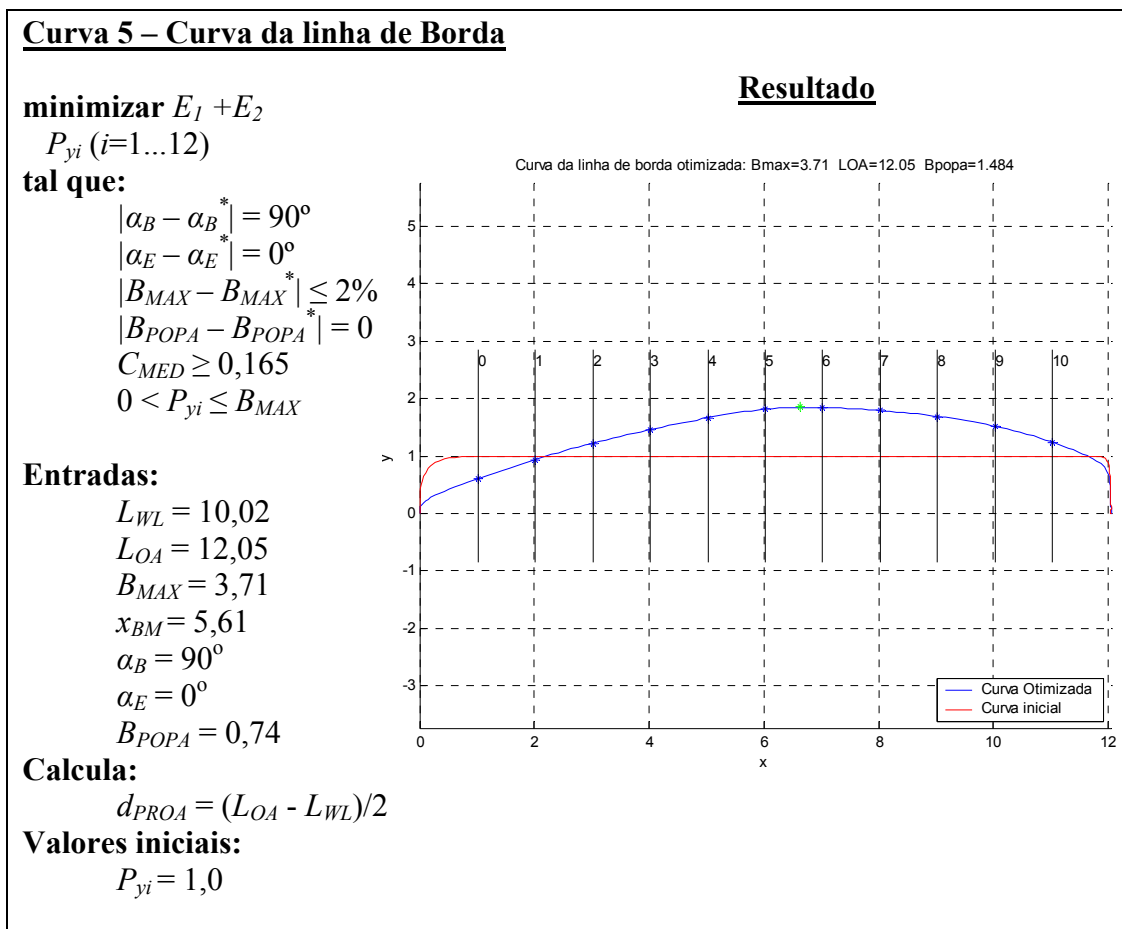


Figura 54. Problema de otimização da curva da linha de borda (Curva 5)

Por fim, as curvas das balizas (Figura 55) são geradas com o mesmo processo de otimização, através de uma função de *loop*, que recebe vetores com as grandezas necessárias das balizas i (de 0 a 10). Os valores iniciais dos ângulos finais obedecem à curva mostrada na Figura 56, onde, à medida que se avança no comprimento do casco, diminui o ângulo final da curva. A curva inicial fornecida para o algoritmo é uma reta que liga as extremidades da curva. Como restrição, limita-se o valor máximo da curvatura de cada baliza, cujo valor também é obtido empiricamente (C_{emp}). As balizas 0 e 10 utilizam um número de pontos de controle variáveis menor, devido a sua forma mais simples. Para este caso, utilizaram-se 6 pontos de controle.

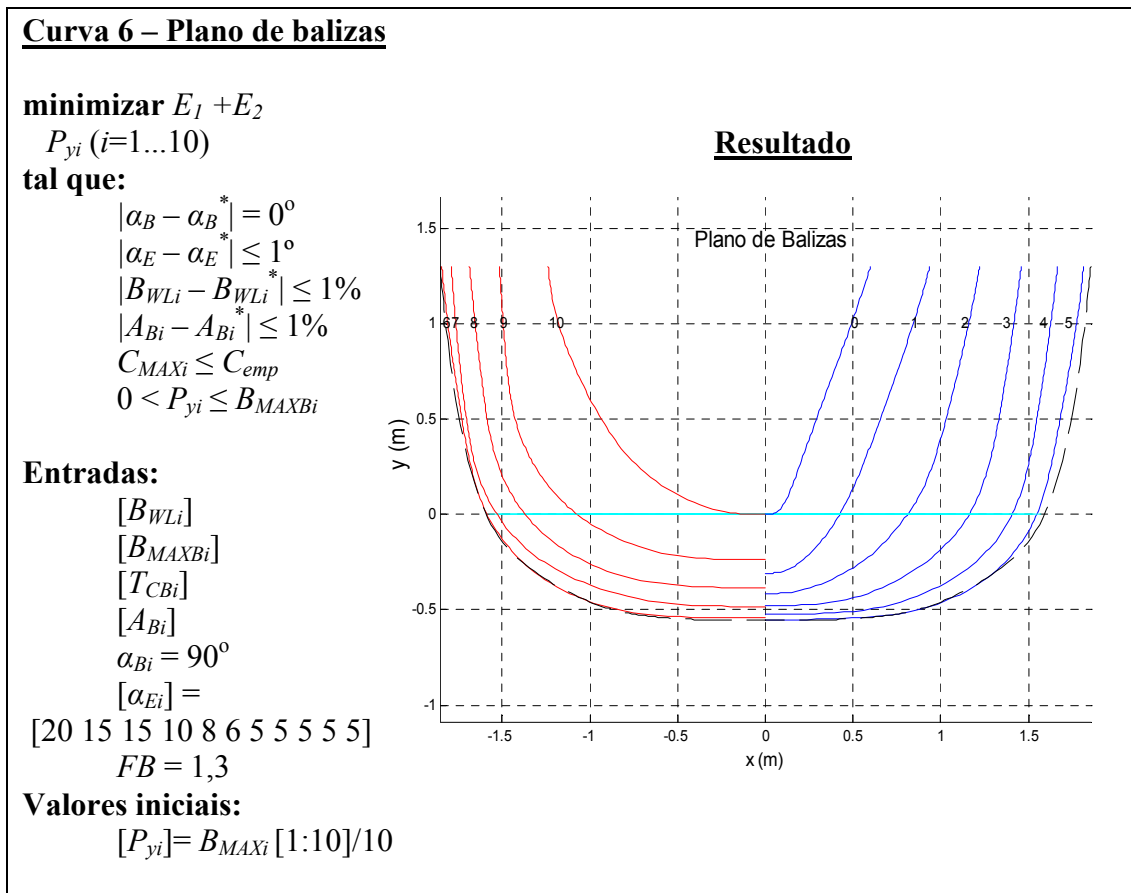


Figura 55. Problema de otimização do plano de balizas (Curva 6)

Ângulos iniciais de saída das balizas

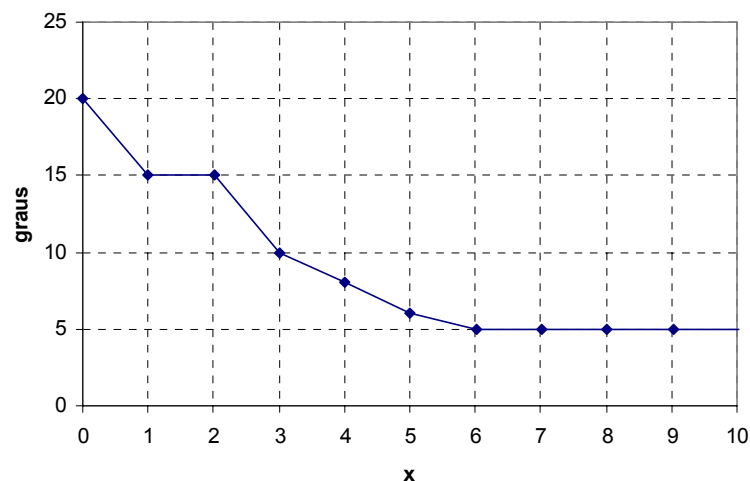


Figura 56. Curva com valores iniciais dos ângulos de saída das balizas

Na Figura 57 e na Figura 58, podem ser vistos os gráficos com os históricos de convergência da função objetivo (normalizada para valores entre 0 e 1) para as

Curvas 1 a 2. Todavia, o número iterações realizadas pelo algoritmo para todas as curvas pode ser visto na Tabela 7.

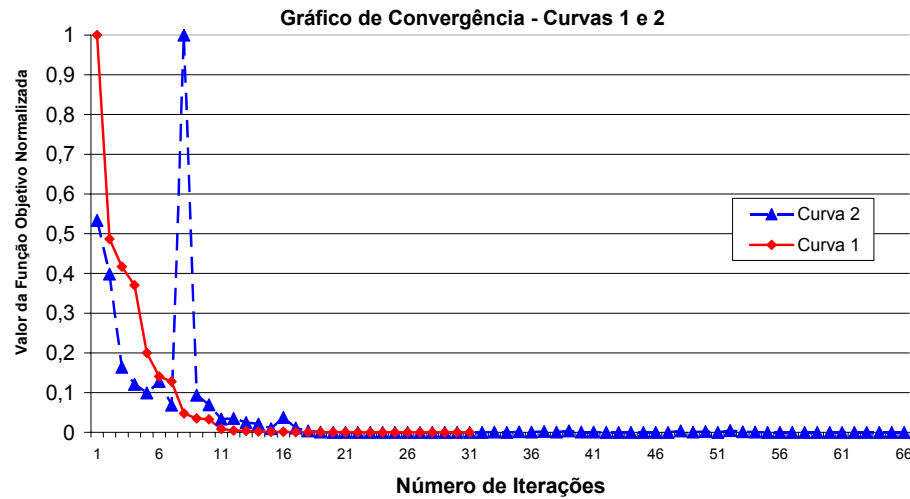


Figura 57. Gráfico de convergência para as Curvas 1 e 2

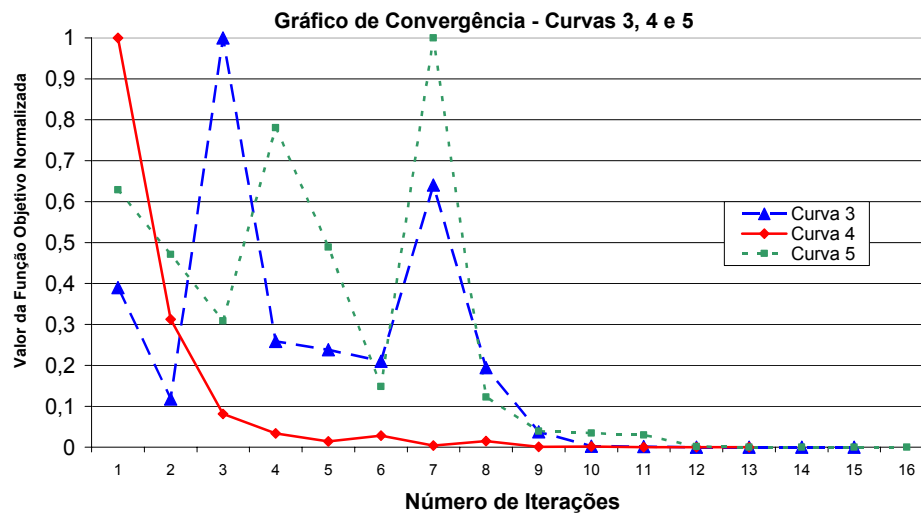


Figura 58. Gráfico de convergência para a Curva 2

Tabela 7. Número de iterações do algoritmo

Curva	Número de Iterações	Curva	Número de Iterações	Curva	Número de Iterações
Curva 1	31	Baliza 0	5	Baliza 5	26
Curva 2	66	Baliza 1	12	Baliza 6	36
Curva 3	15	Baliza 2	21	Baliza 7	32
Curva 4	13	Baliza 3	14	Baliza 8	28
Curva 5	16	Baliza 4	22	Baliza 9	40
				Baliza 10	9

Também com o objetivo de demonstrar possibilidades de potencializar a modelagem inicial, programou-se o sistema para, após gerar as curvas das balizas, traçar uma linha d'água intermediária, na metade da altura da distância de borda livre. Em seguida, com esta curva intermediária, faz-se uma nova iteração, modificando as balizas encontradas para ajustar-se melhor à linha d'água intermediária (Figura 59).

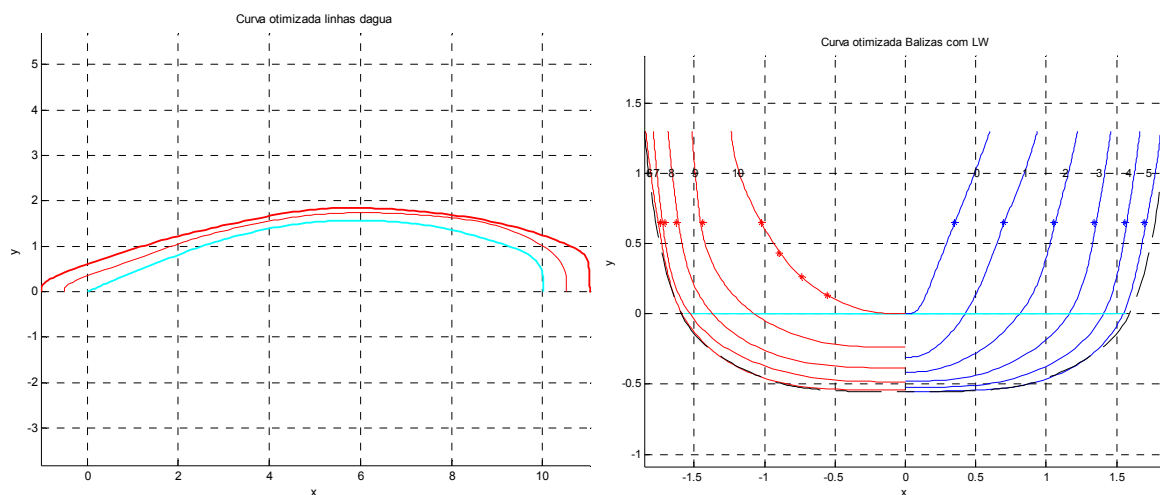


Figura 59. À direita curva da linha d'água intermediária; à esquerda, novas balizas

Obtém-se, então, um novo conjunto de balizas, muito semelhantes ao conjunto primeiramente obtido, porém, com um ponto de restrição a mais, o ponto de intersecção da curva da linha d'água intermediária. Este processo pode ser feito indefinidamente, com outras linhas d'água em outras alturas, ou qualquer outro parâmetro desejado (Figura 60).

Como visto na Figura 48, compondo tridimensionalmente as curvas obtidas forma-se uma casco, com todas suas linhas de desenho principais. Este casco pode ser considerado intrinsecamente liso, já que é formado por curvas submetidas a um critério de suavidade. Ainda assim, o próprio gráfico gerado pelo Matlab já permite que a figura seja rotacionada e visualizada em todas as vistas, ajudando ao projetista na avaliação em relação à aceitar ou modificar os parâmetros de entrada.

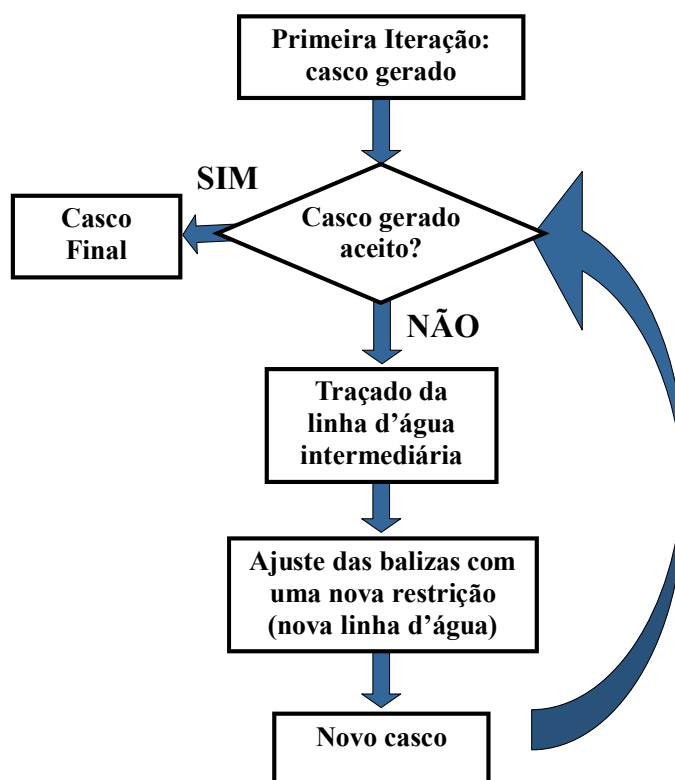


Figura 60. Exemplo de um processo iterando com novas linhas d'água

6.3 Comparação com o veleiro YD-40

O veleiro YD-40, utilizado por Larsson em seu livro “Principles of Yacht Design”, é projetado passo a passo ao longo dos capítulos com o objetivo de ajudar o leitor a entender os aspectos práticos da aplicação da teoria e fórmulas apresentadas. Trata-se de um veleiro de 40 pés, concebido para navegação oceânica, com acomodação para quatro pessoas, podendo ser navegado por apenas dois tripulantes. A Figura 61 mostra um desenho extraído de (LARSSON et al., 1994) com as vistas lateral, frontal e superior deste veleiro.

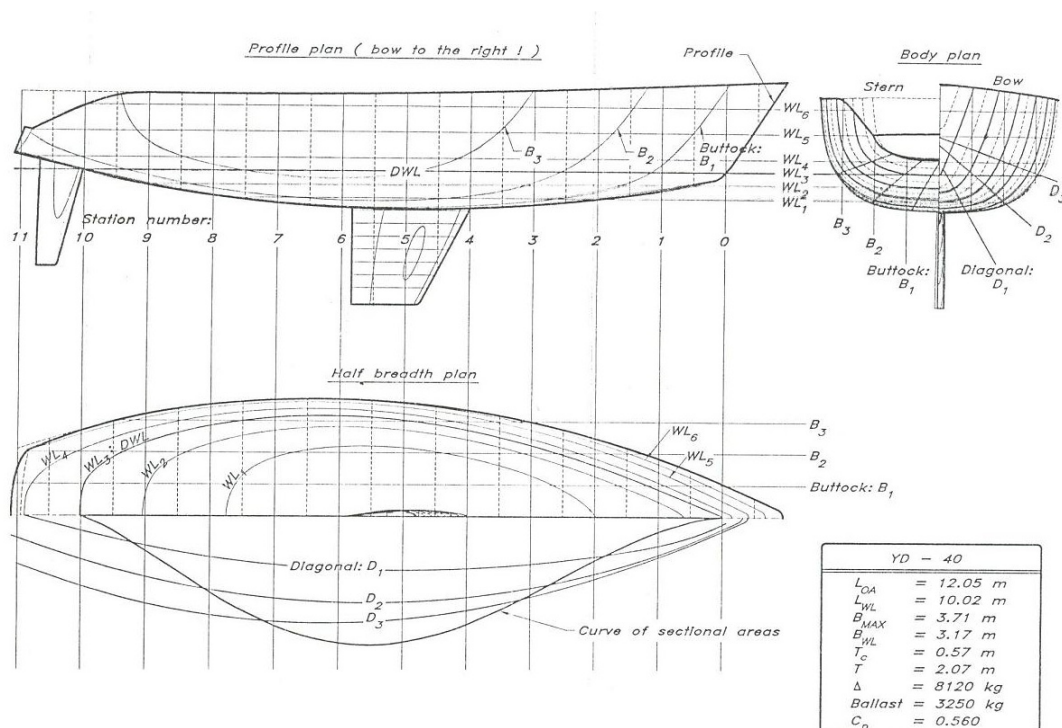


Figura 61. O desenho das linhas do veleiro YD-40 (LARSSON et al., 1994)

Também para efeito didático, e para validar o desempenho da ferramenta implementada neste trabalho, utilizaram-se as medidas do casco do veleiro YD-40 como entrada para o desenho de um casco. Abaixo, nas Figuras 62 e 63, apresenta-se uma comparação entre o desenho retirado de (LARSSON et al., 1994), onde aparece o casco do veleiro YD-40 em perspectiva já com a quilha e o leme, e o casco gerado a partir das dimensões do casco do YD-40. Visualmente, é possível afirmar que o casco gerado é muito semelhante ao caso do YD-40. A Figura 63 é uma aproximação das balizas do YD-40 e foi obtida traçando-se curvas, sobrepondo as balizas da Figura 61. Pela comparação entre as curvas das balizas, comprova-se a semelhança entre ambos os cascos, com exceção das balizas da popa, já que a linha de borda é tratada como uma curva plana somente, e não é rebaixada na parte posterior do casco.

Devido ao fato de que o casco gerado apresenta as mesmas medidas do casco do YD-40, como deslocamento, comprimento da linha d'água, boca máxima, entre outras já citadas anteriormente, o cálculo teórico da resistência residual, velocidade do casco,

resistência induzida também chegarão a um mesmo resultado. (OOSSANEN, 1993; LARSSON, 1994).

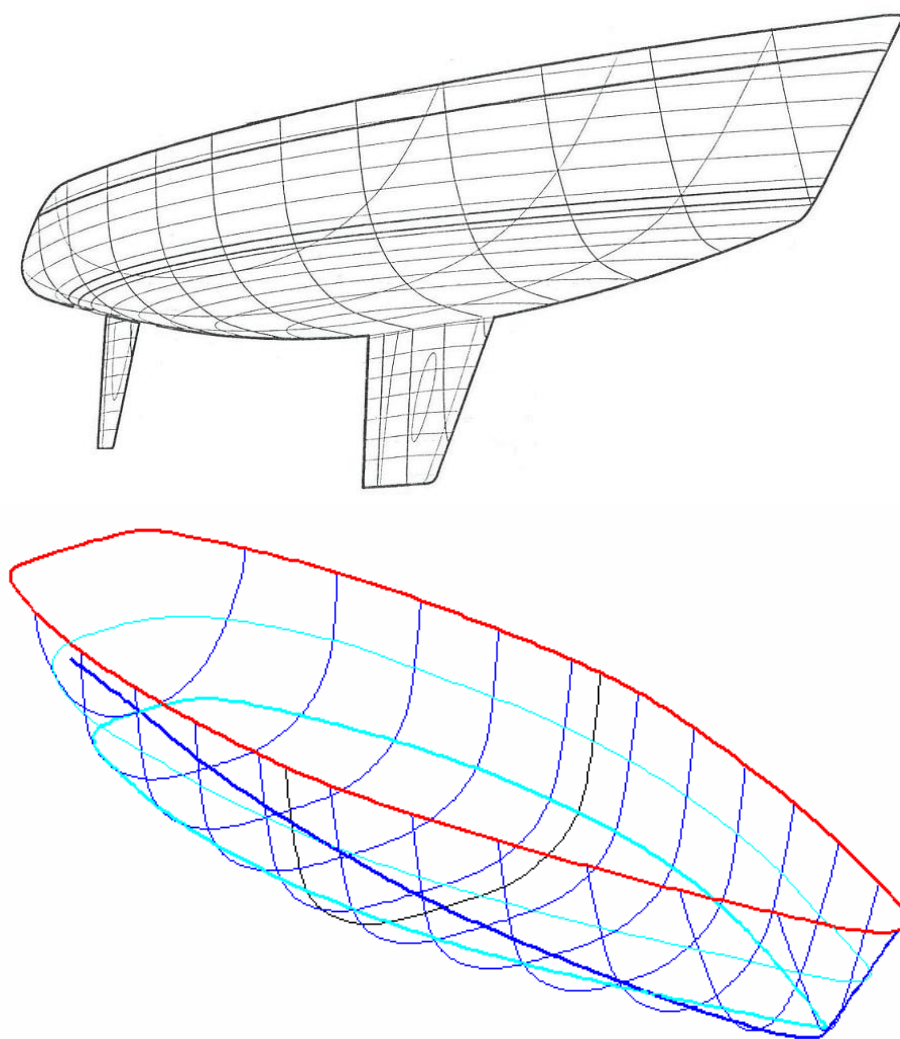


Figura 62. Comparação entre YD-40, em cima (LARSSON et al., 1994), e casco gerado, embaixo.

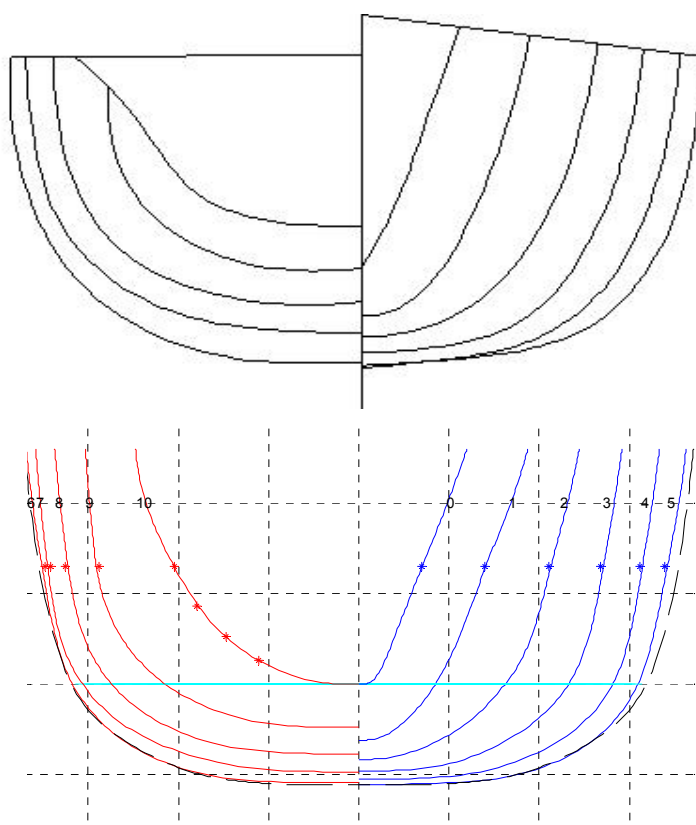


Figura 63. Comparação entre balizas do YD-40, em cima (LARSSON et al., 1994), e balizas geradas, embaixo.

6.4 Análise e pós-processamento dos resultados em CAD

Atualmente, existem inúmeros softwares CAD no mercado para projeto e desenho técnico. Estas ferramentas ficaram consagradas, pois agilizam e facilitam o trabalho do desenhista. Quando adentramos no campo da Engenharia Naval, os desenhos em CAD possibilitam a integração com pacotes de análise numérica, geométrica, ou mesmo hidrodinâmica. Existem também módulos que simulam as condições de fabricação da peça desenhada (programas CAM), expandindo o horizonte do desenhista, dando-lhe uma visão do futuro do seu desenho: a construção.

Embasados nestes argumentos, faz sentido que a ferramenta projetada também permita que o desenho gerado usufrua dos recursos avançados disponíveis para desenho e análise em CAD. Dessa forma, elaboraram-se rotinas que exportam as linhas geradas em Matlab para extensão DXF, utilizadas nos softwares de CAD. A

Figura 64 mostra o modelo exportado para o CAD, pronto para ser manipulado pelos recursos apropriados de desenho.

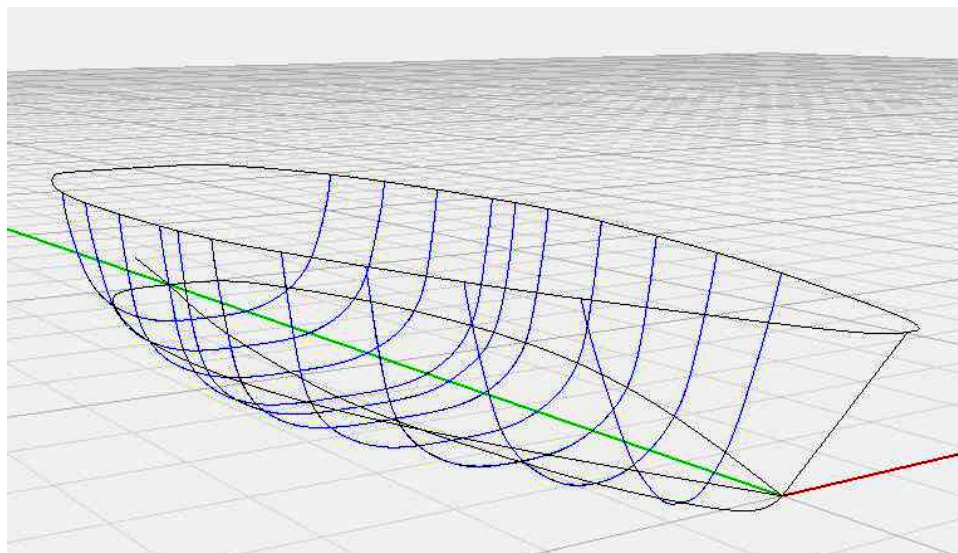


Figura 64. Representação em programa CAD do casco obtido

Para as análises feitas a seguir, será utilizado o programa Rhinoceros 3.0, um programa CAD muito utilizado na modelagem de superfícies e já com algumas ferramentas úteis no desenho naval. A vantagem do Rhinoceros 3D em relação aos outros softwares analisados, é que no momento da criação de um “loft”, o mesmo se dá em NURBS (“Nonuniform Rational B-Splines”). De forma geral, trata-se de um método de geração de superfícies através da transição de perfis. Ocorre que, ao realizar o “loft”, o Rhinoceros 3D permite que seja definido o número de pontos de controle que delimita a superfície em questão, vindo a possibilitar que o arquivo possa ser manipulado quanto a sua definição.

Com a ferramenta “loft”, as curvas das balizas geradas por otimização necessariamente interpolarão a superfície. No entanto, a superfície não irá acompanhar perfeitamente as curvas da linha d’água e de borda. Este é um problema relativamente pequeno, já que é possível contorná-lo adicionando meias-balizas na proa e na popa do casco, a fim de diminuir o erro de interpolação da superfície.

A Tabela 8 apresenta algumas medidas calculadas pelo software CAD do casco gerado, comparando-as com as medidas do casco original do YD-40. As variações são encontradas nas medidas em que se permite que o algoritmo encontre uma solução dentro de uma tolerância especificada. Os resultados foram considerados satisfatórios, pois, na prática, caso desejássemos construir um casco com as mesmas características do veleiro YD-40, considerando as tolerâncias de fabricação, estas diferenças podem ser desprezadas.

Tabela 8. Comparação dos resultados utilizando ferramenta em CAD

	<i>YD-40</i>	<i>Casco Gerado</i>	<i>desvio</i>
Volume de deslocamento [m³]	7,632	7,74518	1,461%
Superfície molhada [m²]	25,220	25,2445	0,097%
Boca máxima [m]	3,17	3,17338	0,107%
Comprimento da linha d'água [m]	10,02	10,02	0,000%
Área interior a D_{WL} (A_{WP}) [m²]	22,61	22,6063	0,016%
Centro de flutuação (x_F) [m]	5,674	0; 5,5391	2,435%
Centro de carena (x_{LCB}) [m]	5,36	0; 5,36396	0,074%

Lembrando o exposto na seção 5.1, escolheu-se, neste trabalho, seguir com uma modelagem simplificada e orientada à otimização de apenas as curvas planas que compõem o casco. No entanto, análises quanto à forma geométrica da superfície são fundamentais para que o projetista decida entre aceitar ou descartar o casco gerado. A Figura 65 mostra dois desenhos da interpolação das balizas geradas por uma superfície. O desenho superior mostra claramente a malha de linhas que formam a superfície. Abaixo, pode-se ver já a superfície renderizada, passando a idéia do produto final do casco.

A principal justificativa para realizar a manipulação das superfícies somente em ambiente CAD é a possibilidade de explorar ao máximo a gama de recursos já disponíveis e desenvolvidos para projetos de barcos. Certamente, o recurso mais importante é a análise da curvatura do casco. Por décadas, o comportamento da

curvatura foi objeto de estudo daqueles que desenvolviam ferramentas de otimização de cascos.

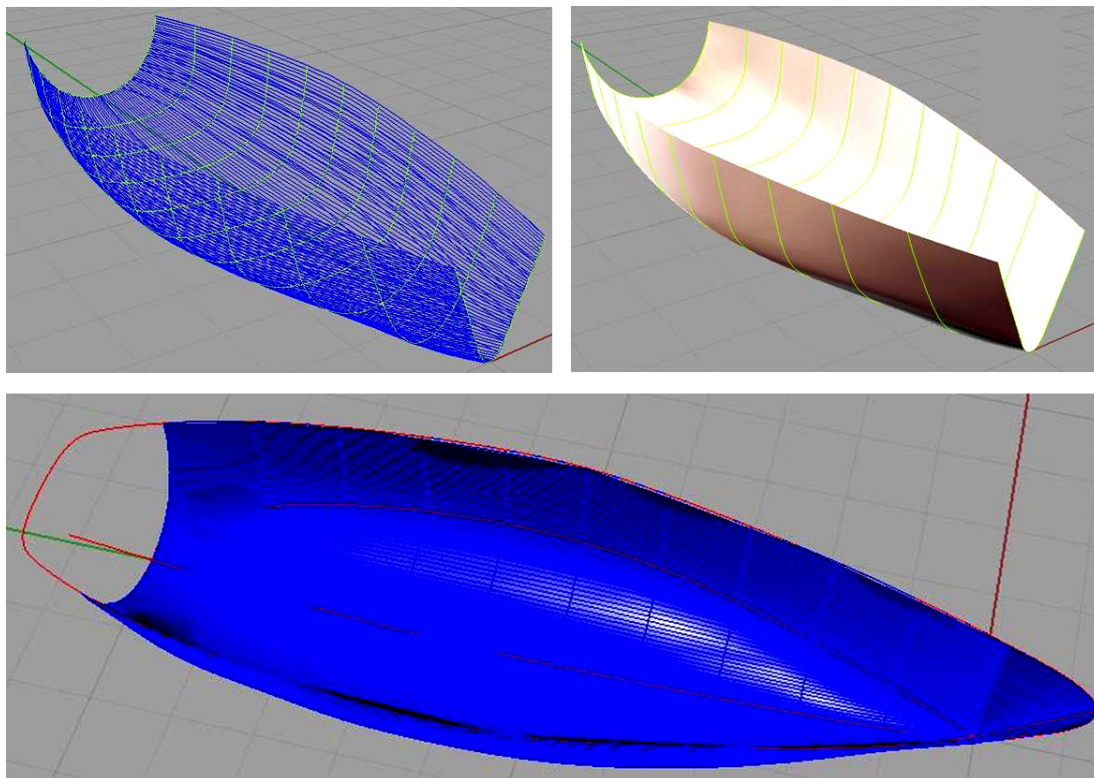


Figura 65. Desenhos da superfície do casco gerado

A Figura 66 apresenta esta análise de curvatura realizada em ambiente CAD. A escala de cores representa do azul escuro para o vermelho a menor para a maior curvatura. Note que na proa é onde aparecem pontos em vermelho, indicando os maiores valores de curvatura. Para esta análise, a resolução da malha gerada, o número de polígonos da superfície, entre outros, influirão na qualidade do resultado. Tal complexidade, foi uma das razões de se preferir desenvolver a integração com o software CAD a implementar as próprias rotinas para realizar esta análise.

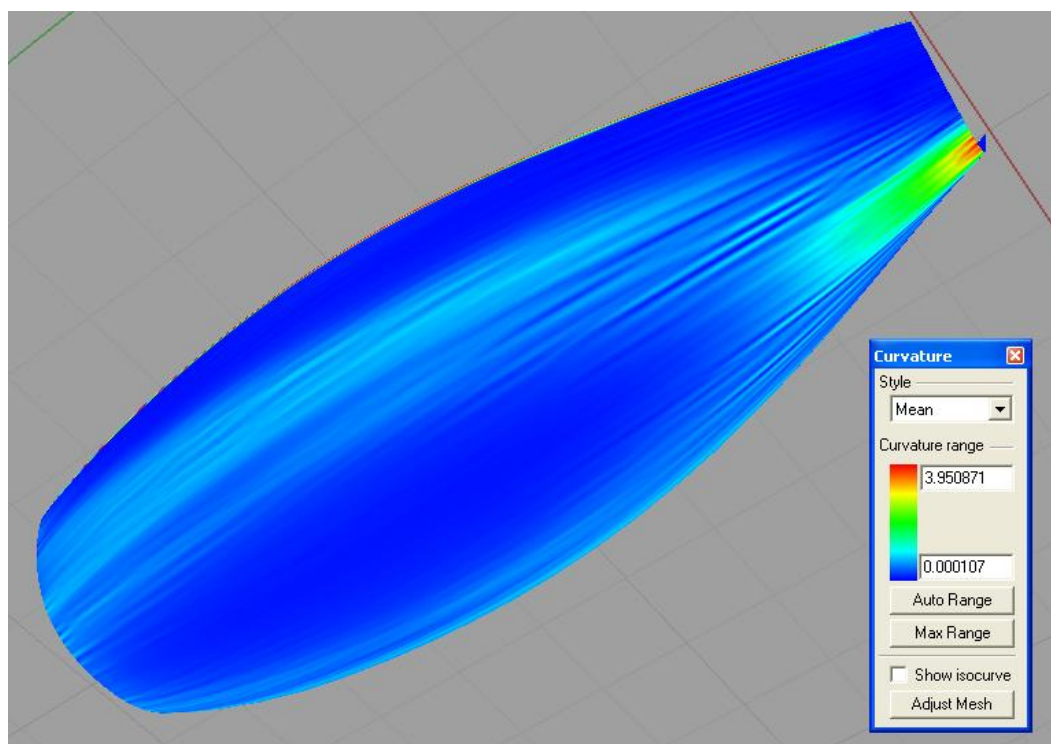


Figura 66. Análise da curvatura em CAD do casco gerado

6.5 Otimização paramétrica do casco pelo MSR

Conforme descrito na seção 3.6, o primeiro passo para obter a superfície é definir quais combinações das variáveis de projeto serão utilizadas como amostras para interpolar a superfície. Como exposto, utilizar-se-á no experimento em questão as variáveis C_p e LCB . Conforme descrito em (LARSSON et al., 1994) os limites superiores e inferiores das estas variáveis são:

$$\begin{aligned} 0,00 &\leq LCB \leq -6,00 \\ 0,52 &\leq C_p \leq 0,60 \end{aligned} \quad (64)$$

A função objetivo será a resistência residual da embarcação, calculada segundo as Equações (61) e (62). A Figura 67 mostra a resistência residual em função de F_n para o casco gerado neste trabalho. A otimização do casco será feita para uma velocidade arbitrada igual a 8 nós. Logo, pela Equação (58), tem-se que o número de Froude correspondente é 0,4. Neste ponto o valor da resistência residual é de

aproximadamente 1.550 N, ao passo que o valor da resistência viscosa é de 66 N, e portanto, sua influência será desprezada na função objetivo.

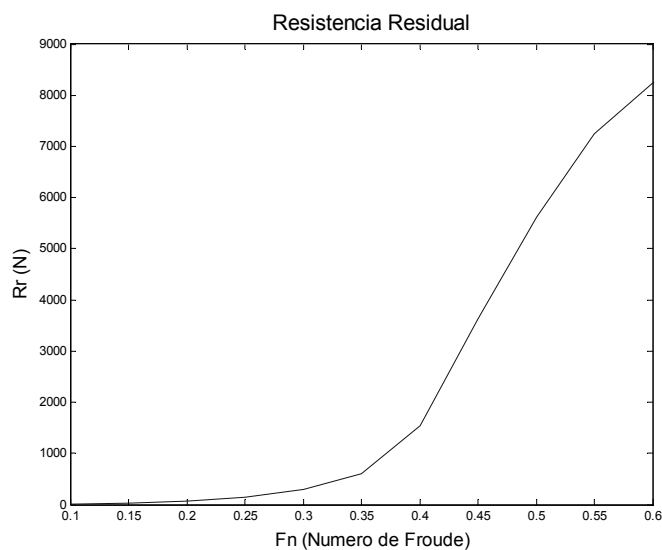


Figura 67. Resistência residual em função de F_n para o casco gerado

A Tabela 9 mostra, segundo o método de CCD, os nove pontos que serão utilizados para a interpolação da superfície de resposta da resistência residual. A última coluna é o valor da resistência residual calculado a partir da Equação (61). A Figura 68 mostra um gráfico com a disposição dos pontos calculados.

Tabela 9. Tabela de CCD para C_p e LCB

	x_1	x_2	C_p	LCB	R_r
1	0	0	0,56	-3,00	15.561
2	1	1	0,59	-0,88	16.526
3	-1	1	0,53	-0,88	18.999
4	-1	-1	0,53	-5,12	18.086
5	1	-1	0,59	-5,12	15.613
6	$\sqrt{2}$	0	0,60	-3,00	15.294
7	0	$\sqrt{2}$	0,56	0,00	18.146
8	$-\sqrt{2}$	0	0,52	-3,00	18.591
9	0	$-\sqrt{2}$	0,56	-6,00	16.854

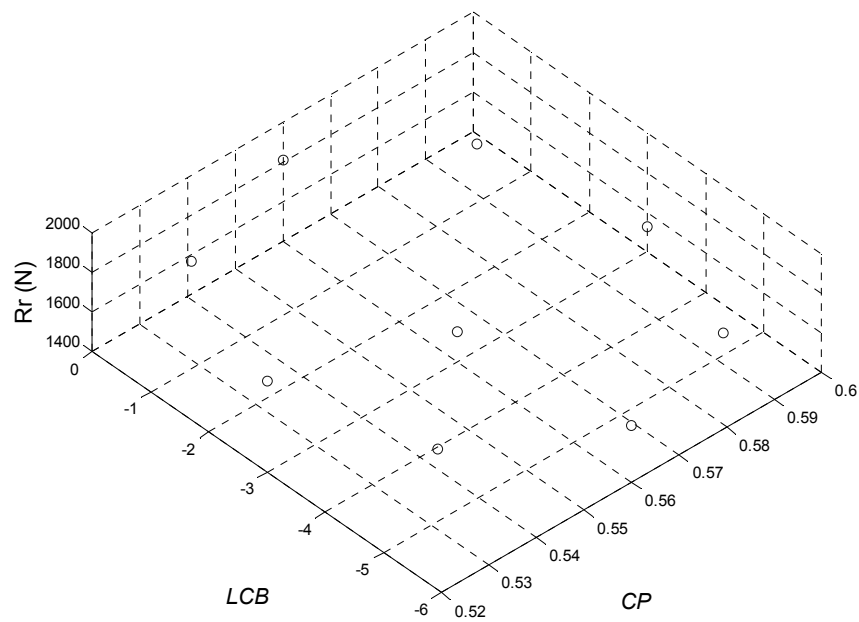


Figura 68. Pontos simulados segundo o método CCD

Com os valores das matrizes \mathbf{X} e \mathbf{Y} , obtém-se $\boldsymbol{\beta}$ a partir da Equação (56).

$$\mathbf{X} = \begin{bmatrix} 1 & 0,56 & -3,00 & 0,31 & 9,00 & -1,68 \\ 1 & 0,59 & -0,88 & 0,34 & 0,77 & -0,51 \\ 1 & 0,53 & -0,88 & 0,28 & 0,77 & -0,46 \\ 1 & 0,53 & -5,12 & 0,28 & 26,21 & -2,71 \\ 1 & 0,59 & -5,12 & 0,34 & 26,21 & -3,02 \\ 1 & 0,60 & -3,00 & 0,36 & 9,00 & -1,80 \\ 1 & 0,56 & 0 & 0,31 & 0 & 0 \\ 1 & 0,52 & -3,00 & 0,27 & 9,00 & -1,56 \\ 1 & 0,56 & -6,00 & 0,31 & 36,00 & -3,36 \end{bmatrix} \quad \mathbf{Y} = \begin{bmatrix} 15561 \\ 16526 \\ 18999 \\ 18086 \\ 15613 \\ 15294 \\ 18146 \\ 18591 \\ 16854 \end{bmatrix} \quad (65)$$

$$\boldsymbol{\beta} = \begin{bmatrix} 31192,31 \\ -100799,05 \\ 150,79 \\ 86319,29 \\ -0,17 \cdot 10^{-7} \end{bmatrix} \quad (66)$$

Assim, tem-se que a função da superfície $R_r(C_p, LCB)$ fica:

$$R_r(C_p, LCB) = 31192,31 - 100799,05C_p + 150,79LCB + 86319,29C_p^2 + 21,54LCB^2 - 0,17 \cdot 10^{-7} C_p LCB \quad (67)$$

A partir daí, monta-se o problema de otimização, que achará o ponto de mínimo da resistência residual:

$$\begin{aligned} \min R_r(C_p, LCB) \quad & \text{sujeito a:} \\ 0,00 \leq LCB \leq -6,00 & \\ 0,52 \leq C_p \leq 0,60 & \end{aligned} \quad (68)$$

Fazendo a análise de sensibilidade para o problema, temos as funções de sensibilidade para cada uma das variáveis de projeto:

$$\frac{dR_r}{dC_p} = -100799,05 + 172638,58C_p - 0,17 \cdot 10^{-7} LCB \quad (69)$$

$$\frac{dR_r}{dLCB} = 150,79 + 43,08LCB - 0,17 \cdot 10^{-7} C_p \quad (70)$$

Devido à robustez do algoritmo e por se tratarem de expressões polinomiais, ambas foram calculadas por diferenças finitas, não apresentando problemas de convergência.

Como ponto de partida para o método, utilizou-se os parâmetros do casco gerado pela ferramenta ($C_p = 0,56$ e $LCB = -3,493$). Na Figura 69, é apresentada a superfície de resposta do problema, e o ponto ótimo encontrado: $C_p = 0,58$ e $LCB = -3,499$. O valor da função objetivo no ponto inicial é de 1.550,72. No ponto ótimo o valor da função objetivo é 1.501,53 N, ou seja, uma variação de 3,17%.

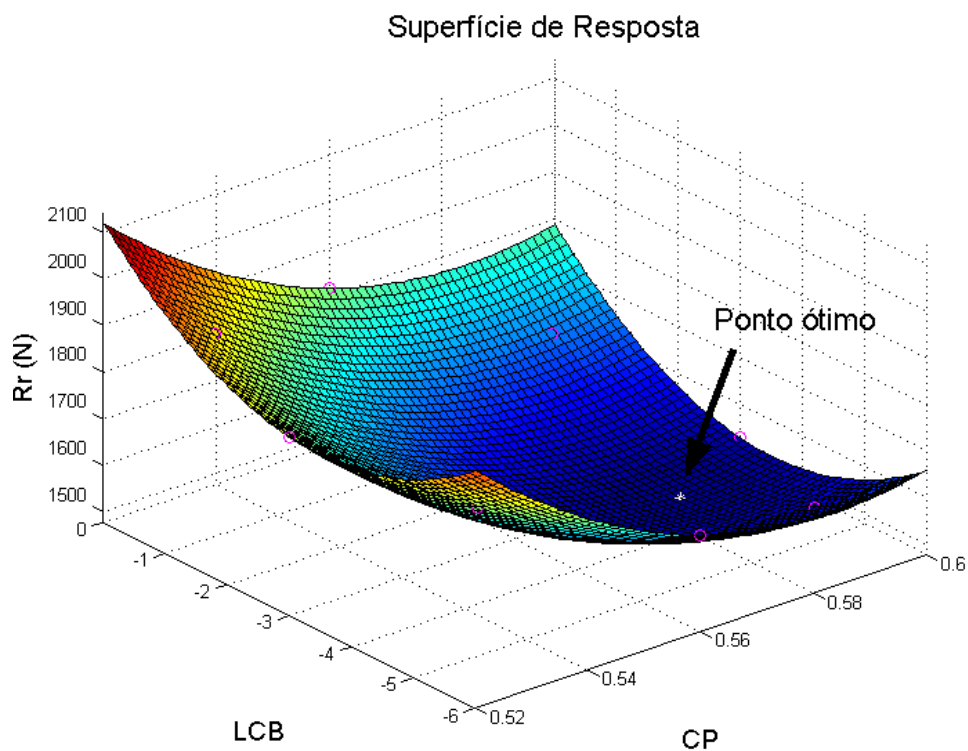


Figura 69. Superfície de Resposta do problema

A Figura 70 mostra o gráfico de convergência da função objetivo do problema de otimização. O algoritmo atinge o critério de parada (ver seção 3.5) em 6 iterações. Em y temos o valor da função objetivo normalizada, que para $y = 1$ corresponde a $R_r = 1.525,61$ N (valor inicial) e para $y = 0$ corresponde a $R_r = 1.501,53$ N (valor ótimo).

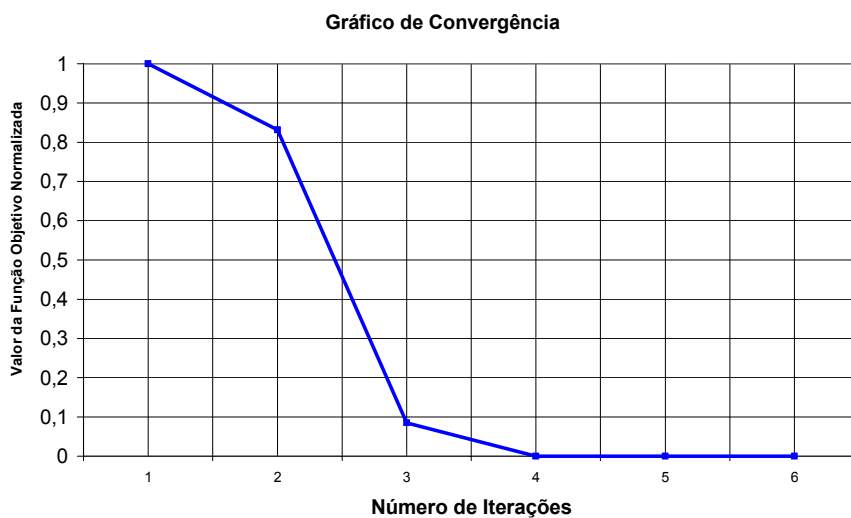


Figura 70. Gráfico de convergência do problema

7 DISCUSSÃO DOS RESULTADOS

7.1 Análise dos resultados de desempenho do algoritmo

A seção 6.1 apresenta resultados obtidos com a suavização de uma curva distorcida, comparando-a com a curva original com a formulação de Wigley. Ao longo da explanação do raciocínio e método utilizado, chega-se a conclusões importantes, que confirmaram também as premissas utilizadas na implementação. Em suma, tem-se que:

- a. Utilizar apenas um grau de liberdade para os pontos de controle diminui a complexidade do modelo matemático de otimização, e, ainda assim, alcança-se resultados satisfatórios em termos da minimização de energia da curva.
- b. O número de pontos de controle tem uma relação exponencial com o custo computacional do algoritmo. Todavia, a qualidade do resultado da suavização não é tão influenciada por este fator. Também, constatou-se que a ponderação das diferentes energias na função objetivo é afetada pelo número de pontos de controle utilizado.
- c. A ponderação das energias na função objetivo propicia mais recursos para a síntese das curvas. A energia de torção não produz resultados interessantes para curvas planas. A integral da curvatura e a energia de flexão produzem o mesmo efeito para o algoritmo, e não há necessidade de adicionar E_c na fórmula da função objetivo.

Considerando isto, podem-se dar como satisfatórios os resultados da modelagem proposta para a suavização da curva B-Spline. A modelagem implementada atende e é orientada ao problema em questão. Em relação aos trabalhos semelhantes encontrados na literatura, obtêm-se resultados considerados satisfatórios, apresentando uma formulação diferente e simplificada.

7.2 Discussão dos resultados do casco gerado

O casco obtido na ferramenta, desenvolvida em Matlab, é visualmente bastante semelhante ao casco do qual se obteve os parâmetros de entrada. Além disso, ainda é possível criar novas estratégias para variações da seqüência de geração das curvas ou mesmo fazer outras iterações, como foi apresentado, criando-se uma linha d'água intermediária, por exemplo.

De todas as maneiras, o casco gerado no Matlab não é o desenho final de um casco a ser fabricado. Como visto, as balizas -1 e 11 não foram sintetizadas pela ferramenta. Isso porque nestas o aspecto estético e construtivo é mais importante que sua suavidade em si, além de que estas formam a parte do casco sobre a linha da água. O formato do deck e a união com a borda do casco também não é escopo do método. Logo, faz-se necessário a finalização destas e demais partes para completar o projeto, lembrando-se sempre que o projeto de um barco obedecerá a metodologia do ciclo de projeto (ver Figura 2), já explanada anteriormente.

7.3 Análise dos resultados da otimização pelo MSR

Em termos percentuais, a otimização da resistência residual é pequena. Isso porque quase não houve variação no *LCB*, que já se encontrava próximo do ponto ótimo. Isso nos leva a crer que o casco projetado por (LARSSON et al., 1994) provavelmente já teria suas dimensões escolhidas de forma a ter uma resistência baixa ou ótima. Além disso, caso o método não houvesse convergido para o ponto ótimo, outra iteração de otimização pelo MSR poderia ter sido feita, interpolando-se outra superfície apenas nas proximidades do ponto ótimo encontrado. Para o problema estudado o ponto ótimo foi encontrado facilmente pelo algoritmo e isso pode ser comprovado calculando-se o ótimo da função analítica da resistência residual (Equações (61) e (62)).

O método apresentado é capaz de analisar várias outras combinações de parâmetros ou mesmo utilizar outras velocidades do barco. O que se buscou foi demonstrar que o

MSR é uma alternativa interessante e de baixo custo para substituir análises pesadas de otimização, e que, por permitir várias combinações de projeto de maneira simples, acaba por trazer bons resultados.

Neste trabalho escolheram-se os parâmetros C_p e LCB , mas, na prática, o projetista deve escolher as variáveis de acordo com as restrições de projeto do armador. Por exemplo, caso o comprimento da linha d'água não seja fixo, ou não haja uma exigência de calado, estes também podem atuar como variáveis de projeto.

Pensando no processo de otimização completo, como exposto na Figura 6, o MSR pode ser uma alternativa recomendável de algoritmo de otimização no bloco chamado Estratégia de Otimização, que envolve a inteligência de tomada de decisão sobre quais parâmetros alterar no casco. Novamente, como uma alternativa simples frente a algoritmos genéticos, como utilizado em (HARRIES et al., 2003).

8 CONCLUSÕES

Neste trabalho, foi implementada uma ferramenta para a síntese de cascos de veleiros a partir de parâmetros de alto nível, e em seguida, aborda-se a modificação de alguns desses parâmetros iniciais, a fim de se minimizar a resistência residual do casco. Aplicando o algoritmo desenvolvido a um exemplo, aponta-se que o modelo elaborado fornece resultados compatíveis com a experiência e prática de projetos de embarcações. Além disso, considerando o desenho preliminar de um casco, a ferramenta desempenha um papel de auxílio à tomada de decisões e análise. Logo, a modelagem desenvolvida está correta e adequada como ferramenta de síntese de casco de embarcações de recreio. A possibilidade de utilização de outras ferramentas, complementando o processo de síntese do casco, faz com que o método sirva como uma poderosa ferramenta dentro do ciclo de projeto.

Conforme citado em (HOCHKIRCH et al., 2002), este tipo de ferramenta não substitui o especialista, mas provê um instrumento personalizado de modelagem. Deve ser enfatizado que a experiência e habilidade do projetista são fundamentais na escolha de quais parâmetros escolher e seus valores, além de validar a forma final obtida. No entanto, esta abordagem permite que o foco seja sempre as atividades de projeto e não de modelagem. Além disso, deve-se sempre levar em consideração que, apesar da ferramenta cumprir com todas as restrições do problema, pode ser gerado um produto que não preencha as necessidades de vendas ou do comprador potencial.

Tampouco, deve-se pensar que existe um casco ótimo para todas as condições, mas sim, uma forma geométrica melhor para determinada condição. O processo eficiente de otimização permite com que o projetista investigue as variações, sem custo excessivo de tempo e materiais, estando sob seu controle todas as variáveis de projeto, restrições e critério de mérito.

A ferramenta cumpre com seu objetivo de proporcionar um desenho preliminar de casco que atende a todos os parâmetros requeridos dentro das tolerâncias permitidas. Ferramentas mais complexas podem ser agregadas ao modelo, porém não se deve

pensar que os softwares sofisticados serão sempre os melhores para o projeto. Utilizar a ferramenta correta para cada aplicação, considerando custo de horas de trabalho e dinheiro, ou mesmo combinar o que há de melhor já disponível, é sempre o caminho mais sensato a ser seguido.

8.1 Desenvolvimentos futuros

O trabalho apresentado aborda a síntese de cascos de veleiros, focando na modelagem paramétrica e síntese das curvas B-Spline do casco. Como desenvolvimento futuro, pode-se estudar com maior enfoque a técnica de superfície de resposta introduzida aqui. Trata-se de um campo amplo, que propicia ao projetista simular inúmeras configurações de casco para diversos critérios de mérito diferentes, segundo seu próprio desejo. (FASSARDI, 2006) apresenta um trabalho bem recente e completo abordando somente este tema. Uma possibilidade de estudo extremamente útil seria implementar uma otimização multi-objetivo, onde a busca-se otimizar a simultaneamente resistência residual e a estabilidade do casco. Neste caso, estas são propriedades conflitantes, e de fundamental importância no projeto do veleiro.

Outro ponto que este trabalho abre possibilidades de estudo, já partindo pelo método mais comum de otimização de cascos, é o campo da simulação por CFD. O desenho em CAD, ou mesmo uma malha de pontos, pode ser introduzido a um software de CFD para a avaliação das forças de resistência sofridas pelo casco, utilizando modelos mais complexos que séries sistemáticas, como apresentado.

A seqüência da simulação em CFD é desenvolver um algoritmo inteligente, capaz de decidir quais variáveis se deseja otimizar ou modificar, e, a partir daí, submeter o modelo a um processo de otimização dos parâmetros escolhidos.

Outra abordagem seria desenvolver uma ferramenta capaz de traduzir um perfil de missão específico para grandezas de entrada da ferramenta implementada neste

trabalho. Por exemplo, projetar um veleiro capaz de suportar determinada carga ou navegar a uma velocidade específica para determinada condição de regata.

Todos estes novos pontos visam sofisticar o software implementado e apresentado neste trabalho, sempre com o objetivo de automatizar e reduzir custo e tempo no projeto de cascos.

LISTA DE REFERÊNCIAS

ABT, C.; BADE, S.D.; BIRK, L.; HARRIES, S., **Parametric Hull Form Design – A Step Towards One Week Ship Design**, 8th International Symposium on Practical Design of Ships and Other Floating Structures · PRADS 2001, Shanghai, September 2001.

ANDRADE, B. L. R. de, **Modelo de Síntese e Otimização por Múltiplos Critérios para o Projeto Preliminar de Embarcações**, Tese de Doutorado, Escola Politécnica da Universidade de São Paulo, São Paulo, 2001.

CHONG, E. K. P, ZAK, H. S., **An Introduction to Optimization**, Willey-Interscience, Second Edition, 2001, pp. 237.

ESTALEIRO SKIPPER, Rio Grande do Sul, Brasil, Processo de fabricação do veleiro Skipper 30, disponível em <<http://www.skipper.com.br>>. Acesso em: 29 de julho de 2006.

FARIN, G., **Curves and Surfaces for Computer Aided Geometric Design**, Second Edition, Academic Press, Arizona, EUA, 1990.

FASSARDI, C., HOCHKIRCH, K., **Sailboat Design by Response Surface Optimization**, 2nd High Performance Yacht Design Conference, Auckland, fevereiro de 2006.

FOLEY, J. D., van Dam, A., FEINER, S. K., HUGHES, J. F. and PHILIPS, R. L. **Introduction to Computer Graphics**, Addison-Wesley, Reading, Massachusetts, EUA, 1993.

FONSECA, M. M., **Arte Naval**, vol. 1, 7ª Ed., Serviço de Documentação da Marinha, Rio de Janeiro, 2005.

FUNCTIONAL DESIGN AND OPTIMISATION OF SHIP HULL FORMS, EC GROWTH Project GRD1-1999-10666/G3RD-CT 2000-00096, Disponível em: <<http://www.ec-nantes.fr/sirehna/fantastic/Publications/Publications.htm>>

Acesso em: 30 de abril de 2006.

GLEICHER, M., **A curve Tutorial for Introductory Computer Graphics**, Department of Computer Science, University of Wisconsin, MAYERS, R. H., MONTGOMERY, D. C., **Response Surface Methodology: Process and Product Optimization Using Designed Experiments**, Wiley Series in Probability and Statistics, EUA, 1995.

GIASSI, A., MAISONNEUVE, J., BENNIS, J., **Multidisciplinary Design Optimisation and Robust Design Approaches Applied to the Concurrent Design**, COMPIT 2003, Hamburgo, 2003.

HAFTKA, R. T., GÜRDAL, Z., **Elements of Structural Optimization, Solid Mechanics and its Applications**, Kluwer Academic Publishers, Dordrecht, The Netherlands, 1995.

HARRIES, S., ABT, C., **Parametric design optimization of sailing yachts**, The Fourteenth Chesapeake, Sailing Yacht Symposium, Annapolis, Maryland, USA, 1998.

HARRIES, S., NOWACKI, H., **Form parameter Approach to the Design of Fair Hull Shapes**, 10th International Conference on Computer Applications in Shipbuilding – ICCAS '99, Massachusetts Institute of Technology, Cambridge, MA, USA, 1999.

HARRIES, S., ABT, C., HOCHKIRCH, K., **Hydrodynamic Modeling of Sailing Yachts**, The 15th CHESAPEAKE SAILING YACHT SYMPOSIUM (CSYS), Annapolis, 2001.

HARRIES, S., JANSON, L., MAISONNEUVE, J., MARZI, J., RAVEN, H., VALDENAZZI, F., **The FANTASTIC roro: CFD optimisation of the forebody and experimental verification**, NAV 2003, Palermo, 2003.

HOCHKIRCH, K.; RÖDER, K.; ABT, C.; HARRIES, S., **Advanced Parametric Yacht Design**, High Performance Yacht Design Conference, Auckland, Nova Zelândia, 2002.

HUANZONG, R., GANG, C. and WEIRONG, Z., **Nonuniform B-Spline Mesh Fairing Method**, Proceedings of Computer Applications in the Automation of Shipyard Operation and Ship Design VI, Elsevier Science Publishers, 261-272, Amsterdam, 1991.

LARSSON, L., ELIASSON, R., **Principles of Yacht Design**, International Marine, EUA, 1994.

MACHIAVERNI, R. M., **Desenvolvimento de uma metodologia de otimização de embalagens metálicas**, trabalho de formatura apresentado à Escola Politécnica da USP, São Paulo, 2004.

MACHIAVERNI, R. M., STUMP, F. V., SILVA, E. C. N., **Design Optimization of a Metallic Package**, 6th World Congress of Structural and Multidisciplinary Optimization, Rio de Janeiro, 2005.

MATLAB Documentation - **Optimization Toolbox: fmincon**, MATLAB, Release 13, version 6.5. The MathWorks Inc. 2002.

MORETON, H.P., SEQUIN, C.H., **Functional Optimization for Fair Surface Design**, Computer Graphics 26, 167-176, University of California, Berkeley, EUA, 1992.

MYERS, R. H., MONTGOMERY, D. C. **Response Surface Methodology**, New York: John Wiley & Sons, Inc., 1995

NAKAMURA, S., **Numerical Analysis and Graphic Visualization with Matlab**. ed. Prentice Hall PTR, 477p., New Jersey, 1996.

NARLI, E., SARIÖZ, K., **The Automated Fairing of Ship Hull Lines Using Formal Optimization Methods**, Istanbul Technical University, faculty of Naval Architecture and Ocean Engineering, Istanbul, Turkey, 2003.

NOWACKI, H., **Computer-aided ship design: Lecture notes**, Dept. of Naval Architecture and Marine Engineering, University of Michigan, 1970.

NOWACKI, H., LIU, X., **Fairing Composite Polynomial Curves with Constraints**, Computer Aided Geometric Design, Vol. 11, 1-15, 1994.

OOSSANEN, P. van, **Predicting the Speed of Sailing Yachts**, SNAME Transactions, Vol.101, 1993.

PARENTE, E. Jr., **Análise de Sensibilidade e Otimização de Estruturas Geometricamente Não-Lineares**, Tese de doutorado, Departamento de Engenharia Civil da PUC-Rio, Rio de Janeiro, 2000.

PORTUGAL WEB, Portugal, Conteúdo sobre a história de Portugal. Disponível em <<http://www.portugalweb.net/historia/naval/caravela.asp>>. Acesso em: 12 de setembro de 2006.

REED, A. M., NOWACKI, H., **Automatic creation of fair ship lines**, Dept. of Naval Architecture and Marine Engineering, College of Engineering, University of Michigan, 1970.

SALLA, F. H. M., **Otimização da configuração de embarcações à vela**, trabalho de formatura apresentado à Escola Politécnica da USP, São Paulo, 2005.

SAPIDIS, N., FARIN, G., **Automatic Fairing Algorithm for B-Spline Curves**, Computer Aided Design, Vol. 22, 121-129, Butterworth-Heinemann, EUA, 1990.

SCHITTKOWSKI, K., NLQPL: A FORTRAN-Subroutine Solving Constrained Nonlinear Programming Problems, Annals of Operations Research, Vol. 5, pp 485-500, 1985.

SNAITH, G.R., PARKER, M.N., **Ship Design with Computer Aids**, Transaction of North East Coast Institution of Engineers and Shipbuilders, p. 151-172, 1972.

TYLER, D., MARGOT, G., GIANLUCA, I., **Optimization of Yard Sectional Shape and Configuration for a modern Clipper Ship**, The International HISWA Symposium on Yacht Design and Yacht Construction, 2002.

VALDENAZZI, F., HARRIES, F., VIVIANI, U., ABT, K., **Seakeeping Optimisation of Fast Vessels by Means of Parametric Modeling**, HSMV 02, Baia, Nápoles, Italia, 2002.

VANDERPLAATS, G. N., Numerical Optimization Techniques for Engineering Design: with Applications, Mcgraw-Hill, New York, 1984.

WIGLEY, W.C.S., A Comparison of Experiment and Calculated Wave Proles and Wave Resistance for a Form Having Parabolic Waterlines, Proceedings of Royal Society, Series A, 1934.

YAMAGUCHI, F., Curves and Surfaces in Computer Aided Geometric Design, Springer-Verlag, Berlin, 1988.

APÊNDICE

Nos parágrafos a seguir, será apresentada uma seqüência de passos para demonstrar uma maneira de abordarmos o problema exposto neste trabalho do começo ao fim, ou seja, até a etapa de fabricação do casco. O exemplo em questão utiliza um software CAM de mercado chamado VisualMill 5.0, integrado com o software CAD Rhinoceros3D.

A integração com as ferramentas de CAD e CAM neste trabalho tem como objetivo apenas demonstrar e propor uma seqüência de passos para se obter o produto final de um casco, desde sua concepção, passando por sua síntese de forma otimizada. Em um processo tradicional de projeto de cascos, sistemas CAD cumprem apenas a função de mecanização das atividades de desenho, com todas as dificuldades e limitações inerentes. No processo proposto, o CAD recebe as curvas previamente definidas matematicamente, e permite então seguir aos próximos passos do projeto.

Demonstra-se o ganho, em termos de esforço e tempo, que se pode alcançar através de um processo que exige pouca interferência humana, e a qualidade dos resultados obtidos. Além disso, não somente ao casco é aplicável a modelagem em CAD e CAM. O deck, leme, e bolina também podem ser usinados, e onde os ganhos do emprego do CNC são enormes. No deck, por exemplo, é possível, na fase de desenho, projetar os detalhes complexos, como apoios e degraus, e usiná-los já na mesma peça. No leme e bolina, consegue-se uma alta precisão nas peças projetadas, garantindo o desempenho planejado.

A Figura 71 mostra o primeiro passo para a usinagem em CNC: a definição da matriz na qual se extrairá o casco. Definir o tamanho do bloco, ou os tamanhos dos blocos menores, dependerá da extensão do curso da ferramenta que a máquina CNC possui. Neste exemplo, simularemos o fresamento de um molde com bloco único, apenas por ser mais didático e de fácil ilustração.

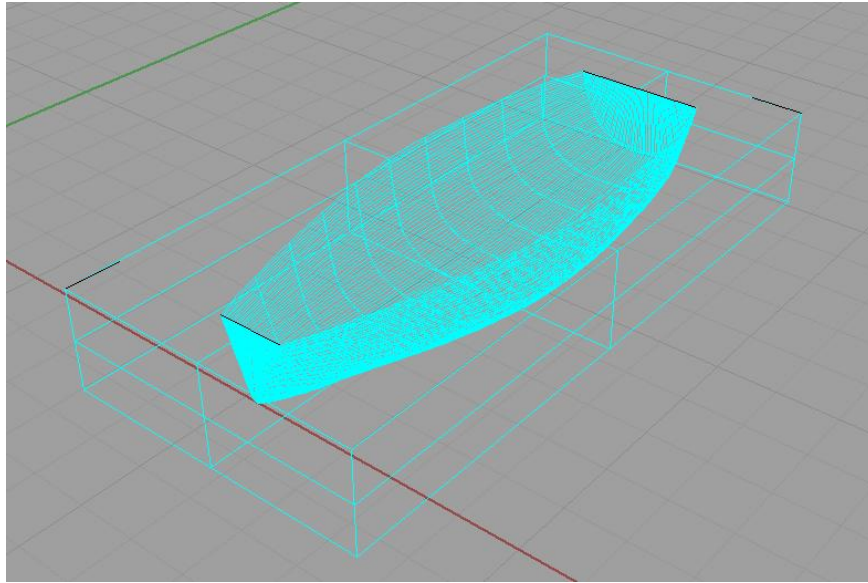


Figura 71. Bloco sólido gerado a partir do casco em CAD

O software CAM simula o fresamento do bloco (Figura 72) de acordo com os parâmetros de entrada como ferramenta, rotas de usinagem, acabamento, tolerâncias, etc. Além de simulações de melhor caminho de corte, tempo de usinagem e mesmo a qualidade da peça, o software VisualMill escreve o código CNC para o comando de máquina escolhida.

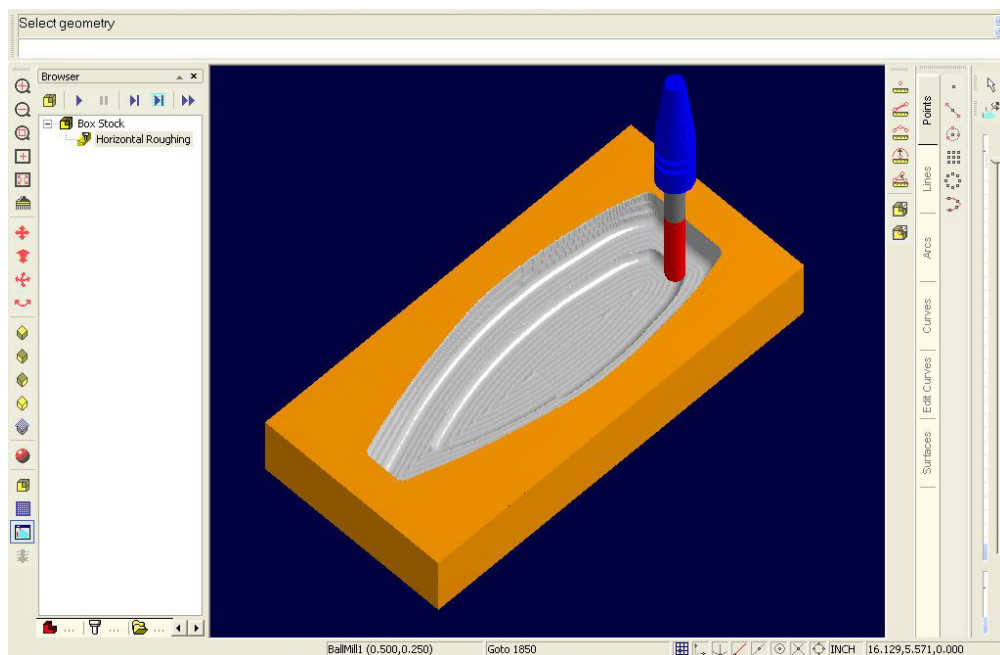


Figura 72. Simulação do fresamento do casco gerado

A Figura 73 apresenta um quadro com a listagem de comandos em código G do molde do casco gerado. Não será escopo deste trabalho abordar os detalhes e funções de cada comando. O objetivo é demonstrar uma solução de projeto, enxergando todas as etapas do começo ao fim, da concepção à fabricação.

O código G abaixo realiza a usinagem tridimensionalmente do casco, em uma máquina com 4 eixos. Todavia, pode-se também desejar cortar as seções do casco, conforme visto na Figura 8, para a construção em madeira ou metal de um modelo, ou mesmo das chapas que formarão o casco. Isso é largamente utilizado, hoje em dia, através de corte a laser ou plasma.

```
G90
G21
G0X-0.0748Y6.084
Z1.55
G1X-0.0754Y6.055Z1.5398F2.7
X-0.0739Y5.7031Z1.4114
X-0.0567Y5.2952Z1.2628
X0.Y4.7862Z1.0763
X0.0445Y5.1417F3.7
X0.0585Y5.3162
X0.0753Y5.7707
X0.0705Y6.4372
X0.0395Y7.2448
X-0.0428
X-0.0552Y7.0147
X-0.0754Y6.0559
...
```

Figura 73. Quadro com comandos em código G do molde do casco gerado

Livros Grátis

(<http://www.livrosgratis.com.br>)

Milhares de Livros para Download:

[Baixar livros de Administração](#)

[Baixar livros de Agronomia](#)

[Baixar livros de Arquitetura](#)

[Baixar livros de Artes](#)

[Baixar livros de Astronomia](#)

[Baixar livros de Biologia Geral](#)

[Baixar livros de Ciência da Computação](#)

[Baixar livros de Ciência da Informação](#)

[Baixar livros de Ciência Política](#)

[Baixar livros de Ciências da Saúde](#)

[Baixar livros de Comunicação](#)

[Baixar livros do Conselho Nacional de Educação - CNE](#)

[Baixar livros de Defesa civil](#)

[Baixar livros de Direito](#)

[Baixar livros de Direitos humanos](#)

[Baixar livros de Economia](#)

[Baixar livros de Economia Doméstica](#)

[Baixar livros de Educação](#)

[Baixar livros de Educação - Trânsito](#)

[Baixar livros de Educação Física](#)

[Baixar livros de Engenharia Aeroespacial](#)

[Baixar livros de Farmácia](#)

[Baixar livros de Filosofia](#)

[Baixar livros de Física](#)

[Baixar livros de Geociências](#)

[Baixar livros de Geografia](#)

[Baixar livros de História](#)

[Baixar livros de Línguas](#)

[Baixar livros de Literatura](#)
[Baixar livros de Literatura de Cordel](#)
[Baixar livros de Literatura Infantil](#)
[Baixar livros de Matemática](#)
[Baixar livros de Medicina](#)
[Baixar livros de Medicina Veterinária](#)
[Baixar livros de Meio Ambiente](#)
[Baixar livros de Meteorologia](#)
[Baixar Monografias e TCC](#)
[Baixar livros Multidisciplinar](#)
[Baixar livros de Música](#)
[Baixar livros de Psicologia](#)
[Baixar livros de Química](#)
[Baixar livros de Saúde Coletiva](#)
[Baixar livros de Serviço Social](#)
[Baixar livros de Sociologia](#)
[Baixar livros de Teologia](#)
[Baixar livros de Trabalho](#)
[Baixar livros de Turismo](#)