

UNIVERSIDADE FEDERAL DE PERNAMBUCO  
CENTRO DE TECNOLOGIA E GEOCIÊNCIAS  
DEPARTAMENTO DE ENGENHARIA MECÂNICA

TÚLIO DE MOURA CAVALCANTE

FERRAMENTAS COMPUTACIONAIS PARA GERAÇÃO E SELEÇÃO DE  
GEOMETRIAS DE *MOONPOOL* DE *DRILLSHIP* COM BASE NA  
OPERABILIDADE

RECIFE  
2015

UNIVERSIDADE FEDERAL DE PERNAMBUCO  
CENTRO DE TECNOLOGIA E GEOCIÊNCIAS  
DEPARTAMENTO DE ENGENHARIA MECÂNICA

TÚLIO DE MOURA CAVALCANTE

FERRAMENTAS COMPUTACIONAIS PARA GERAÇÃO E SELEÇÃO DE  
GEOMETRIAS DE *MOONPOOL* DE *DRILLSHIP* COM BASE NA  
OPERABILIDADE

Trabalho apresentado ao Curso de Graduação em  
Engenharia Mecânica, da Universidade Federal de  
Pernambuco, como requisito final para a obtenção do Grau  
de Bacharel em Engenharia Mecânica.

Orientadora: Prof.<sup>a</sup> M.Sc. Paula Suemy Arruda Michima.

RECIFE  
2015

Catálogo na fonte  
Bibliotecária Valdicéa Alves, CRB-4 / 1260

C376f Cavalcante, Túlio de Moura  
Ferramentas computacionais para geração e seleção de geometrias de moonpool de drillship com base na Operabilidade. / Túlio de Moura Cavalcante - Recife: O Autor, 2015.  
69 folhas, Ils.; e Tab.

Orientadora: Profa. M.Sc. Paula Suemy Arruda Michima.

TCC (Graduação) – Universidade Federal de Pernambuco. CTG. Programa de Graduação em Engenharia Mecânica, 2015.  
Inclui Referências Apêndices e Anexos.

1. Engenharia Mecânica. 2. Moonpool. 3. Drillship. 4. Algoritmo evolutivo. 5. Algoritmo gamma. 6. Operabilidade. I. Michima, Paula Suemy Arruda (Orientadora). III. Título.

UFPE

621 CDD (22. ed.)

BCTG/2015-109

# FOLHA DE APROVAÇÃO

TÚLIO DE MOURA CAVALCANTE

FERRAMENTAS COMPUTACIONAIS PARA GERAÇÃO E SELEÇÃO DE  
GEOMETRIAS DE *MOONPOOL* DE *DRILLSHIP* COM BASE NA  
OPERABILIDADE

Este trabalho foi julgado adequado e aprovado para a obtenção do título de bacharel em  
Engenharia Mecânica da Universidade Federal de Pernambuco.

Recife, 03 de Março de 2015

---

Prof.<sup>ª</sup> M.Sc. Paula Suemy Arruda Michima  
Universidade Federal de Pernambuco  
**Orientadora**

**Banca examinadora:**

---

Prof. Ph.D. Marcos Pereira  
Universidade Federal de Pernambuco

---

Prof. M.Sc. Marco Antônio Ladislau  
Petkovic  
Universidade Federal de Pernambuco

A Jesus de Nazaré, que, verdadeiramente, é Deus Todo-Poderoso.

## **AGRADECIMENTOS**

Primeiramente, a Deus, por tudo.

Aos meus pais, pelos exemplos, pela dedicação e pelo apoio em tudo.

Ao meu irmão, pela amizade de sempre.

Aos meus avós, pela amizade e pela sabedoria compartilhada.

A Thayane, meu amor, pela força.

A todos os meus familiares, pelos exemplos e pela amizade.

Ao pessoal do Labcom, de todas as épocas, por todo o aprendizado e amizade.

Aos companheiros de graduação, agora de profissão; sem eles, o curso de Engenharia Mecânica seria interminável.

À minha orientadora, Professora Paula Michima, pelo esforço e atenção.

Aos professores como um todo e à UFPE, por todo o conhecimento adquirido, muito obrigado.

## RESUMO

A manutenção da posição, tanto em termos de localização quanto em termos de movimento em torno do baricentro, de um *drillship* é crucial para a sua operabilidade. Por isso, é preciso investigar estratégias para mitigar os efeitos das excitações provenientes das ondas do mar incidentes e buscar soluções que permitam mantê-lo operando sob a maior variedade de estados de mar possível. Neste trabalho, buscou-se esse objetivo através da variação da geometria e dimensões do *moonpool* com o auxílio de ferramentas computacionais desenvolvidas pelo autor. Os conceitos por trás dos métodos criados mostraram-se bastante interessantes e úteis, além disso, os resultados obtidos indicam a estabilidade e a versatilidade desses algoritmos, além de uma possível convergência, uma vez que se tenha em mãos uma ferramenta que forneça, de forma consistente, os dados de desempenho, em resposta à excitação de ondas, para uma configuração casco-*moonpool*. São descritos os conceitos em que se baseia a elaboração das ferramentas de geração de novos formatos de *moonpool* e de busca da solução ótima, além disso, são apresentados alguns exemplos de testes.

Palavras-chave: *Moonpool*. *Drillship*. Algoritmo evolutivo. Algoritmo *gamma*. Operabilidade.

## **ABSTRACT**

*Maintaining the position in terms of location and motion of a drillship is crucial to its operability, so it is necessary to investigate strategies to mitigate the effects of the excitations from the incident sea waves and search solutions that allow keeping its operation under the largest range of sea states as much as possible. In this work, this goal was pursued by the author through the variation of the moonpool geometry and dimensions by developing computational tools. The concepts behind the developed methods proved to be very interesting and useful, in addition, the results indicate the stability and the versatility of these algorithms, moreover to a possible convergence, since it has at hand a tool that provides, consistently, the performance data in response to excitation waves for a hull-moonpool configuration. The concepts underlying the development of the generation of new moonpool formats and search of the optimal solution tools are described, moreover, some examples of tests are shown.*

*Keywords: Moonpool. Drillship. Evolution algorithm. Gamma algorithm. Operability.*

## LISTA DE FIGURAS

Figura 1.1 – <i>Drillship</i> .....	14
Figura 2.1 - Decomposição da resistência ao avanço. ....	17
Figura 2.2 - <i>Moonpool</i> ótimo de Park <i>et al.</i> (2007). ....	20
Figura 2.3 - <i>Moonpool</i> com câmaras de amortecimento nas laterais.....	21
Figura 2.4 - <i>Moonpools</i> com recuos (ou recessos) traseiro e dianteiro.....	22
Figura 2.5 - <i>Moonpool</i> longitudinalmente convergente.....	22
Figura 3.1 - Conjuntos de valores formando indivíduos.....	24
Figura 4.1 - Solução presa entre os dois picos de ressonância.....	25
Figura 4.2 - Loop do algoritmo genético convencional. ....	26
Figura 4.3 - Loop do algoritmo genético modificado. ....	27
Figura 4.4 - Esquema geral do Algoritmo <i>Gamma</i> .....	29
Figura 4.5 - Parâmetros suficientes para a definição de um indivíduo ( <i>moonpool</i> ). ....	31
Figura 4.6 - <i>Moonpool</i> retangular com quinas arredondadas.....	32
Figura 4.7 - <i>Moonpool</i> com metade “em chanfro” e metade retangular com quina suavizada em elipse.....	32
Figura 4.8 - <i>Moonpool</i> elíptico. ....	32
Figura 4.9 - <i>Moonpool</i> retangular com apenas uma das quinas suavizada em elipse.....	33
Figura 4.10 - Semelhante à Figura 4.7, com a metade “em chanfro” para a proa. ....	33
Figura 4.11 - <i>Moonpool</i> retangular sem os arredondamentos nas quinas.....	33
Figura 4.12 - Referência para a posição de incidência de onda. ....	35
Figura 4.13 - Condições de altura significativa de onda para operabilidade do <i>drillship</i> (Ts=8s). ....	36
Figura 4.14 - Condições de altura significativa de onda para operabilidade do <i>drillship</i> (Ts=10s). ....	37
Figura 4.15 - Condições de altura significativa de onda para operabilidade do <i>drillship</i> (Ts=12s). ....	37
Figura 4.16 - Distribuição de altura e períodos de picos de ondas na bacia de campos. ....	39
Figura 6.1 - Evolução das Iterações no teste 1. Melhores notas para cada conjunto [I1;I2].....	41
Figura 6.2 - Mapa de Operabilidade para a melhor de todas as configurações no teste 1.	

.....	42
Figura 6.3 - <i>Moonpool</i> com maior banda de operabilidade no teste 1.....	43
Figura 6.4 - Mapa do melhor <i>moonpool</i> para [ $l_1=10,00\text{m}$ ; $l_2=6,25\text{m}$ ] no teste 1.....	43
Figura 6.5 - Melhor <i>moonpool</i> para [ $l_1=10,00\text{m}$ ; $l_2=6,25\text{m}$ ] no teste 1.....	43
Figura 6.6 - Mapa do melhor <i>moonpool</i> para [ $l_1=11,50\text{m}$ ; $l_2=6,50\text{m}$ ] no teste 1.....	44
Figura 6.7 - Melhor <i>moonpool</i> para [ $l_1=11,50\text{m}$ ; $l_2=6,50\text{m}$ ] no teste 1.....	44
Figura 6.8 - Mapa do melhor <i>moonpool</i> para [ $l_1=12,25\text{m}$ ; $l_2=5,25\text{m}$ ] no teste 1.....	44
Figura 6.9 - Melhor <i>moonpool</i> para [ $l_1=12,25\text{m}$ ; $l_2=5,25\text{m}$ ] no teste 1.....	45
Figura 6.10 - Evolução das Iterações no teste 2. Melhores notas para cada conjunto [ $l_1;l_2$ ].....	45
Figura 6.11 - Mapa de Operabilidade para a melhor de todas as configurações no teste 2. .....	45
Figura 6.12 - <i>Moonpool</i> com maior banda de operabilidade no teste 2.....	46
Figura A1 - Localização das áreas de medição próximas à de Campos (W3).....	68
Figura A2 - Histogramas direcionais sazonais das características das ondas para W3. Coluna da esquerda: altura significativa; coluna da direita: período significativo.....	69

## **LISTA DE TABELAS**

Tabela 4.1 - Tabela dos parâmetros de operação de perfuração. ....	35
Tabela 4.2 - Dimensões do navio ao qual se referem as Figura 4.12, Figura 4.13 e Figura 4.14.....	38
Tabela 6.1 - Indivíduos tomados como exemplos para efeito de comparações.....	42

## SUMÁRIO

<b>CAPÍTULO 1 - INTRODUÇÃO.....</b>	<b>13</b>
1.1 PROBLEMA.....	13
1.2 PERGUNTA DE PESQUISA.....	15
1.3 JUSTIFICATIVA.....	15
1.4 OBJETIVOS.....	15
1.4.1 Objetivo Geral.....	15
1.4.2 Objetivos Específicos.....	15
<b>CAPÍTULO 2 - FUNDAMENTAÇÃO TEÓRICA.....</b>	<b>16</b>
2.1 MOTIVAÇÃO.....	16
2.2 EMBASAMENTO TEÓRICO.....	16
2.3 TRABALHOS RECENTES.....	19
<b>CAPÍTULO 3 - METODOLOGIA.....</b>	<b>23</b>
<b>CAPÍTULO 4 - DESENVOLVIMENTO DAS FERRAMENTAS.....</b>	<b>24</b>
4.1 ALGORITMO EVOLUTIVO.....	24
4.2 GERAÇÃO DO MOONPOOL.....	30
4.3 AVALIAÇÃO DOS INDIVÍDUOS SEGUNDO OS CRITÉRIOS DE OPERABILIDADE.....	34
<b>CAPÍTULO 5 - TESTES.....</b>	<b>39</b>
<b>CAPÍTULO 6 - RESULTADOS.....</b>	<b>40</b>
<b>CAPÍTULO 7 - CONCLUSÃO.....</b>	<b>46</b>
<b>CAPÍTULO 8 - TRABALHOS FUTUROS.....</b>	<b>47</b>
<i>REFERÊNCIAS.....</i>	<i>48</i>
<i>APÊNDICE A - CÓDIGO FONTE DO ALGORITMO GAMMA.....</i>	<i>50</i>
<i>APÊNDICE B - CÓDIGO FONTE DO PROGRAMA GERADOR DE MOONPOOL....</i>	<i>56</i>
<i>ANEXO A - INFORMAÇÕES ACERCA DAS CARACTERÍSTICAS DE ONDA NA BACIA DE CAMPOS.....</i>	<i>68</i>

## CAPÍTULO 1- INTRODUÇÃO

### 1.1 PROBLEMA

Um navio de perfuração, ou *drillship*, como apresentado na Figura 1.1, é um equipamento móvel utilizado pela indústria do petróleo, inclusive em águas profundas, na produção *offshore*. É uma embarcação com uma sonda de perfuração em seu convés superior, posicionada próxima à meia-nau.

Para dar acesso ao mar à broca de perfuração ou a outras ferramentas necessárias, ou ainda a operários, é preciso que haja um furo passante do convés ao mar. Essa abertura é chamada de *moonpool*.

A presença desse furo torna a estrutura e a hidrodinâmica do *drillship* bastante peculiares, visto que é uma singularidade na superfície que está em contato com a água. Essa circunstância, diante do movimento relativo entre navio e mar, gera, na superfície livre do fluido interno ao *moonpool*, oscilações que podem ser, algumas vezes, até maiores do que as do mar externo à embarcação, como afirmam Hammargren e Törnblom (2012).

Dessa forma, entende-se que tanto um navio, em mar calmo, viajando a velocidade de cruzeiro, quanto o navio parado em correnteza e excitação de ondas do mar externo podem ser situações de geração de oscilações significativas, como relembram Gaillarde e Cotteleer (2005). Essas situações, respectivamente, descrevem um modelo de navio em trânsito e um modelo de um navio em operação.

Quando o navio está em trânsito, a dificuldade gerada pelas oscilações é o aumento do arrasto e, conseqüentemente, da resistência ao avanço da embarcação. No caso do navio parado, para a operação de perfuração, o inconveniente criado pelas grandes oscilações é a movimentação da embarcação justamente num momento em que ela precisaria de um posicionamento preciso, o que pode impossibilitar a atividade.

A solução para esse problema seria, então, alterar o formato, eventualmente o posicionamento e as dimensões do *moonpool*, de forma a minimizar a movimentação da sua água interna ou os efeitos dela no comportamento do navio para a maior banda de frequências de oscilação do mar externo possível, ampliando assim a faixa de operabilidade do *drillship* em atividade de perfuração.

Figura 2.1 – *Drillship*.

Fonte: Página *on-line* da 2B First Consulting.

Avaliando a literatura, percebe-se que há diversos estudos; como os de Fukuda (1977), Aalbers (1984), Gaillarde e Cotteleer (2005), Hammargren e Törnblom (2012), entre outros; a respeito do *moonpool* e seus efeitos durante o deslocamento do navio (as soluções apresentadas, em geral, são apêndices que tendem a reduzir a área do furo em alguma altura), e poucos a respeito dessa questão durante a operação do navio (e que avaliem mudanças de formato, respeitando as limitações de área de abertura mínima para operação). Diante disso, decidiu-se estudar o que tem sido pouco abordado: buscar a melhor solução para a configuração do *moonpool* que maximize a faixa de operabilidade do *drillship* para determinados estados de mar, através apenas das variações de geometria e dimensões na fase de projeto, sem o acréscimo de apêndices posteriores que possam diminuir a secção transversal do furo. Neste texto, são apresentados estudos acerca de *moonpools* com posicionamento fixo, centrados à meia-nau.

## 1.2 PERGUNTA DE PESQUISA

Dados uma faixa de frequência usual das oscilações do mar e um *drillship*, qual a geometria do *moonpool* que resulta na mais ampla banda de operabilidade para o navio?

## 1.3 JUSTIFICATIVA

Ampliar ao máximo a faixa de operabilidade do *drillship* torna a atividade de perfuração menos propensa a suspensões devidas aos efeitos decorrentes do comportamento do casco em estado do mar crítico. Tais suspensões, além de serem economicamente inconvenientes, estão relacionadas a eventos de risco à segurança da tripulação e do equipamento a bordo. No mais, é importante que se busque aumentar a capacidade de operação na maior quantidade possível de situações críticas respeitando as necessidades dimensionais da atividade, sem criar restrições à passagem das pessoas e dos equipamentos porque, dessa forma, o *moonpool* teria sua utilidade limitada.

## 1.4 OBJETIVOS

### 1.4.1 Objetivo Geral

Elaborar um método capaz de indicar as geometrias e dimensões do *moonpool* de um navio sonda de perfuração que maximizam sua faixa de operabilidade para um determinado estado de mar.

### 1.4.2 Objetivos Específicos

I) Apresentar as diferentes estratégias já propostas para a redução das oscilações na água interna ao *moonpool*, das quais foram extraídos conceitos que pautaram a escolha da estratégia de variação de formas e dimensões proposta.

II) Desenvolver um algoritmo que seja capaz de projetar, sobre o casco de

um *drillship*, *moonpools* com geometrias diversas.

III) Desenvolver algoritmo de busca da configuração ótima que resulte no melhor desempenho do *drillship*, segundo critérios de operabilidade.

IV) Sugerir diretrizes para trabalhos futuros.

## **CAPÍTULO 2 - FUNDAMENTAÇÃO TEÓRICA**

### **2.1 MOTIVAÇÃO**

*Drillships* precisam ser capazes de manter uma posição específica fixa em ambientes marítimos, para garantir a qualidade dos seus trabalhos de perfuração. Sabe-se que as forças do vento, das correntes e das ondas, que agem no navio, dificultam a manutenção da posição, mas é importante salientar que as oscilações no fluido interno ao *moonpool* são igualmente problemáticas nessas situações. Além disso, caso o período das ondas externas coincida com o período natural de oscilação da água do *moonpool*, a amplitude de elevação da superfície livre se torna muito grande, devido à ressonância, causando, por vezes, transbordamento no convés, tornando o trabalho de perfuração inviável, tendo que ser interrompido, afirmam Michima e Kawabe (2014).

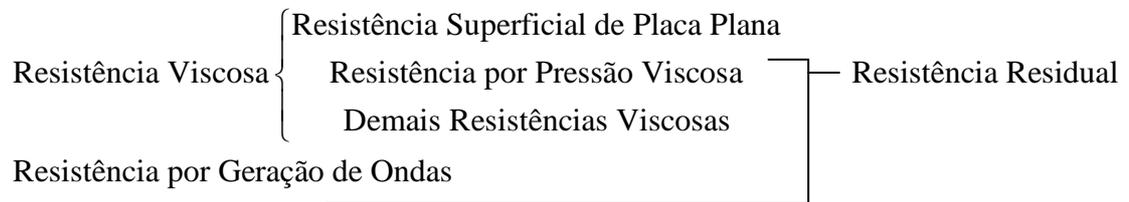
Dessa forma, se torna imprescindível estudar maneiras de garantir a operabilidade do navio, sem restringir o acesso ao mar.

### **2.2 EMBASAMENTO TEÓRICO**

No estudo da resistência do mar ao navio, é possível decompô-la em dois termos: a resistência viscosa e a resistência por geração de ondas. A resistência viscosa, por sua vez, pode ser decomposta em resistência por atrito superficial de placa plana (considerando-se um escoamento sobre uma placa plana com a mesma área que a parte submersa do casco do navio), resistência por pressão viscosa (em oposição ao desvio do fluxo da água causado pelo movimento do navio), além das demais formas menos atuantes de resistência viscosa. De todos esses termos, apenas a resistência por atrito superficial de placa plana pode ser calculada, sendo todas as outras reunidas no termo

de resistência residual, como mostra a Figura 2.1.

Figura 2.1 - Decomposição da resistência ao avanço.



Fonte: Elaborada pelo autor.

Essa decomposição é bastante útil para os testes com modelos, que estimam a resistência da água a uma embarcação real a partir da resistência a um modelo geometricamente semelhante. Essa estimativa é feita com base na decomposição descrita acima e na Lei da Comparação de William Froude, que diz que a razão entre as resistências residuais ( $R_R$ ) de dois navios geometricamente semelhantes é o cubo da razão entre os seus comprimentos ( $L$ ), se a razão entre as velocidades ( $V$ ) deles for a raiz quadrada da razão entre os seus comprimentos (Lewis, 1988). Matematicamente:

$$\frac{V_1}{V_2} = \sqrt{\frac{L_1}{L_2}} \Rightarrow \frac{R_{R1}}{R_{R2}} = \left(\frac{L_1}{L_2}\right)^3 \quad (2.1)$$

Assim, pode-se relacionar a velocidade do modelo à velocidade do navio para estimar a resistência residual sofrida pelo último a partir do que foi medido no primeiro.

Considerando que o coeficiente de resistência residual ( $C_R$ ) é dado pela expressão (2.2):

$$C_R = \frac{R_R}{\left(\frac{1}{2}\rho \cdot S \cdot V^2\right)} \quad (2.2)$$

Em que  $V$  é a velocidade relativa entre a embarcação e a água do mar,  $S$  é a área característica para a parcela da resistência que se quer calcular, enquanto  $\rho$  é a densidade da água; é possível perceber que se navios diferentes, geometricamente semelhantes, mantiverem a mesma razão  $V^2/gL$ , em que  $g$  é a aceleração da gravidade, eles terão o mesmo coeficiente de resistência residual  $C_R$ .

A razão  $V^2/gL$  é o quadrado do chamado número de Froude, que é definido como mostra a expressão (2.3):

$$F_n = \frac{V}{\sqrt{gL}} \quad (2.3)$$

Assim, a estimativa da resistência do mar um navio real pode ser feita da seguinte forma:

Seja  $C_T$  o coeficiente de resistência total,  $C_F$  o coeficiente de resistência por atrito de placa plana, tem-se que:

$$C_T = C_R + C_F \quad (2.4)$$

Onde  $C_F$  pode ser calculado segundo a expressão determinada pela International Towing Tank Conference (ITTC), de 1957:

$$C_F = \frac{0,0075}{(\log_{10} Re - 2)^2} \quad (2.5)$$

Em que  $Re$  é o número de Reynolds para a velocidade relativa do escoamento do mar externo sobre o casco. Daí, faz-se um teste com um modelo em escala reduzida, com a velocidade relativa entre água e barco ajustada de forma que o número de Froude do modelo seja igual ao do navio. Dessa forma, os coeficientes de resistência residual da embarcação real e do modelo serão iguais.

Desse teste, extrai-se  $C_T$  e calcula-se  $C_F$ , para o modelo. Assim, pode-se determinar o  $C_R$  do modelo, que será o mesmo do navio.

O  $C_F$  do navio pode ser calculado, assim teremos o  $C_T$  do navio.

No caso de um *drillship*, a presença do *moonpool*, apesar de não afetar as componentes de resistência por geração de ondas, uma vez que a forma do casco até uma profundidade pouco menor que o calado não é alterada, como explicam Hammargren e Törnblom (2012), provoca o deslocamento das linhas de corrente ao longo da parte inferior do navio, o que cria uma parcela adicional de resistência por pressão viscosa, uma vez que haverá mais perturbações no fluxo.

Os incrementos e decrementos de resistência supracitados podem ser considerados pequenos diante da resistência total à embarcação, porém há perturbações mais significativas que ocorrem ao fluido interno ao *moonpool*. Essas perturbações, que originarão mais termos adicionais de resistência, podem, para fins de estudo, ser decompostas da seguinte forma: o movimento vertical da coluna de fluido (ou movimento de pistão) e o movimento longitudinal entre as paredes verticais (ou movimento de *sloshing*), afirmam Hammargren e Törnblom (2012).

## 2.3 TRABALHOS RECENTES

Alguns estudos a respeito da influência do *moonpool* durante o deslocamento do navio apresentam conclusões relevantes e extensíveis ao caso abordado nesse texto. Essas conclusões serão apresentadas nos parágrafos seguintes.

Para um *moonpool* quadrado, Sadiq e Xiong-liang (2008), citados por Hammargren e Törnblom (2012), afirmam que, a baixos números de Froude, o modo pistão é iniciado quando a frequência do fluido, a frequência natural induzida pelo campo de fluxo irrotacional dentro do *moonpool* e a frequência da oscilação da camada limite assumem o mesmo valor. Quando  $F_n$  cresce, as frequências de separação da camada limite e das oscilações da superfície livre deixam de ser iguais e o fenômeno de pistão ocorre devido à energia dissipada pelos vórtices criados na aresta frontal do *moonpool*.

Quanto à pressão nas paredes do *moonpool*, ela está relacionada à velocidade do navio. Conforme a velocidade cresce, essa pressão é diminuída, como afirma Molin (2001), citado por Hammargren e Törnblom (2012).

Qualquer movimento oscilatório só pode ser bem caracterizado pelo conhecimento de duas variáveis: amplitude e frequência. No caso estudado, é fundamental que se descubra como é possível reduzir a amplitude da oscilação da água interna ao *moonpool* e como mantê-la afastada da frequência de ressonância. Sabe-se que a frequência de ressonância depende principalmente da geometria do *moonpool*, enquanto a velocidade do navio determina a amplitude das oscilações. Os vórtices gerados na aresta frontal do *moonpool* têm a mesma frequência da coluna de água, afirmam Hammargren e Törnblom (2012).

Quando as oscilações da coluna de água se tornam muito grandes, elas podem dificultar a formação de novos vórtices na aresta frontal do *moonpool*, travando o sistema naquela frequência, isso é chamado de efeito de histerese, o que pode levar à sustentação da frequência de ressonância por muito tempo, alertam Veer e Tholen (2008).

Segundo os experimentos feitos pela *Delft University* (Veer e Tholen, 2008) e pela SHI (Hye-Jong *et al.*, 2008), o efeito de *sloshing* é esperado quando a razão entre o calado ( $D$ ) e o comprimento do *moonpool* ( $L$ ) for conforme a seguinte expressão:

$$\begin{cases} D/L < 0,35 \sim 0,39 \text{ (Delft University)} \\ D/L < 0,66 \text{ (SHI)} \end{cases} \quad (2.6)$$

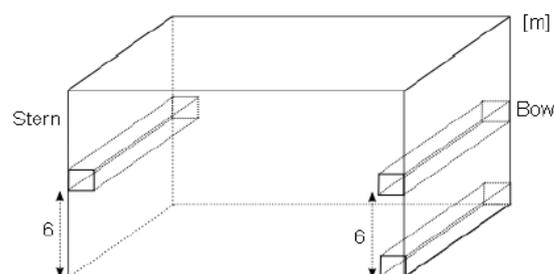
Devemos atentar para o fato de que o efeito de pistão é mais acentuado em *moonpool* de comprimento curto e *sloshing* é mais acentuado em *moonpool* de comprimento longo, como enfatizam Hammargren e Törnblom (2012).

Mais recentemente, tem-se desenvolvido também métodos numéricos capazes de prever o comportamento do fluido no *moonpool*, com o objetivo de estudar esses fenômenos.

No trabalho de Gaillarde e Cotteleer (2005), há um esforço para se obter uma reprodução relativamente boa das características do fluxo e, baseados nos resultados obtidos, há a projeção de que utilizar, em algoritmos futuros, formulações baseadas em métodos de Volume de Fluido renderá melhoras significativas na simulação de problemas desse tipo.

No trabalho de Park *et al.* (2007), citado por Hammargren e Törnblom (2012), o ANSYS FLUENT foi usado para simular a variação da resistência ao avanço para diferentes formatos de *moonpool* com blocos de amortecimento. Eles fizeram uma simulação tridimensional para cinco geometrias simplificadas de *moonpool*. O *moonpool* ótimo, de acordo com os resultados de Park *et al.* (2007), é o que pode ser visto na Figura 2.2.

Figura 2.2 - *Moonpool* ótimo de Park *et al.* (2007).



Fonte: Hammargren e Törnblom (2012).

Para diminuir a resistência ao avanço do navio, causada pela existência do *moonpool*, pode-se usar diversos artifícios para evitar a formação dos vórtices ou amortecer as oscilações da coluna de água.

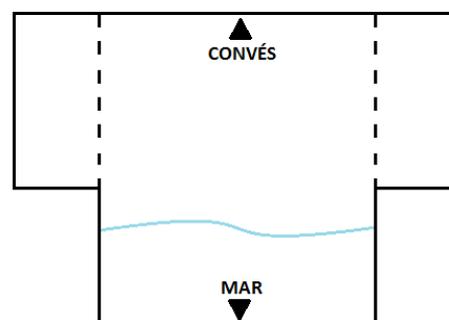
Para *moonpools* curtos, podem ser usadas cunhas na aresta frontal, para, de maneira a diminuir o arrasto, desviar o fluxo e deslocar os vórtices, empurrando-os para baixo, afastando-os do casco do navio. Pode-se também usar “quebra-vórtices” na parte posterior do poço. Não há muita certeza quanto à efetividade desses dispositivos para *moonpools* longos, de maneira que, enquanto Veer e Tholen (2008) afirmam que eles são efetivos, Hye-Jong *et al.* (2008) afirmam que não.

É possível que se usem *flaps* para diminuir as oscilações no *moonpool*, mas isso seria desvantajoso ao propósito do presente trabalho, pois implicaria na restrição do diâmetro por onde o equipamento de perfuração pode operar. Quanto mais próximo de zero estiver o ângulo dos *flaps*, mais efetivos eles são, mas mais fechado está o *moonpool*.

Visando o encurtamento do *moonpool* ou o aumento do amortecimento das oscilações, respectivamente, podem ser aplicadas divisórias transversais ou chapas horizontais presas nas laterais do *moonpool*, cuja melhor posição é logo abaixo da superfície livre, segundo Aalbers (1984) e Fukuda (1977).

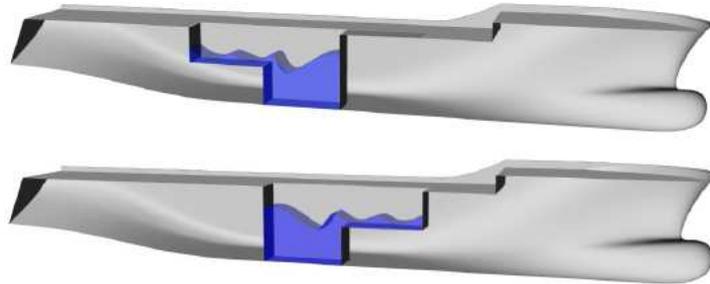
Também podem ser utilizadas câmaras de amortecimento nas laterais do *moonpool* (Figura 2.3) ou mesmo podem ser projetados recuos a partir de certa altura do poço (Figura 2.4). Além disso, pode-se ainda aplicar uma geometria longitudinalmente convergente para o *moonpool* (Figura 2.5), a fim de reduzir as oscilações, como sugerem Hammargren e Törnblom (2012).

Figura 2.3 - *Moonpool* com câmaras de amortecimento nas laterais.



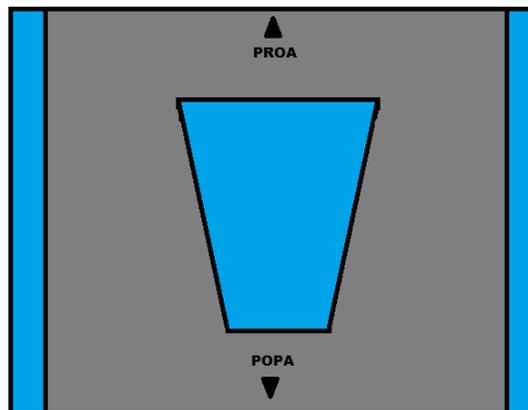
Fonte: Elaborada pelo autor.

Figura 2.4 - *Moonpools* com recuos (ou recessos) traseiro e dianteiro.



Fonte: Hammargren e Törnblom (2012).

Figura 2.5 - *Moonpool* longitudinalmente convergente.



Fonte: Elaborada pelo autor.

Pode-se ainda combinar velocidade e calado, a fim de reduzir as oscilações, afirmam Gaillarde e Cotteleer (2005). Porém, como o escopo deste trabalho é o de operação, essa estratégia não é uma em solução pertinente.

No trabalho de Torres (2007), há um estudo sobre a influência do *moonpool* no movimento vertical de uma plataforma em monocoluna. Sua conclusão é de que há dois períodos de ressonância: um próximo ao período de ressonância da plataforma sem *moonpool* e o outro próximo ao da oscilação vertical da água interna ao *moonpool* quando o movimento da plataforma está restringido. O autor também concluiu que o período natural da água interna ao *moonpool* com restrição do tipo duto é maior que o período natural do movimento vertical da monocoluna sem *moonpool*.

Torres (2007) verificou ainda que quanto maior a restrição ao *moonpool*, ou seja, quanto menor a área aberta dele, menor a oscilação da água interna. Da mesma

forma, o período natural dessa água sofre um acréscimo em seu valor. Na mesma dissertação, também afirmou que a situação em que o movimento da água interna influencia significativamente o movimento da plataforma é aquela em que o período natural do *moonpool* com plataforma fixa é menor ou igual ao período natural da monocoluna sem *moonpool*.

### **CAPÍTULO 3 - METODOLOGIA**

Sabendo que há pouca literatura a respeito do problema específico de que trata esse trabalho, iniciou-se uma pesquisa por trabalhos que abordassem problemas correlatos.

Partindo de dois artigos relativamente recentes e bastante completos, o de Hammargren e Törnblom (2012), que trata do problema do *moonpool* do navio em trânsito, e o de Torres (2007), que trata do problema da abertura na plataforma monocoluna, foi possível descobrir e ter acesso aos outros textos que servem como bibliografia a esse trabalho e que estão citados nas referências, inclusive a trabalhos clássicos, como o de Aalbers (1984) e o de Fukuda (1977), entre outros, que serviram de base teórica para esta pesquisa.

Dessa bibliografia, foram absorvidas as conclusões que foram apresentadas no capítulo 2.3, as quais foram tomadas como primeiras pistas de quais tipos de modificações significariam, dentro da realidade abordada nesse texto, uma melhora na operabilidade e serviram como uma das bases para a elaboração da forma de avaliação dos *moonpools* que foi utilizada nesse trabalho, que, apesar de não fazer parte do escopo desse estudo, e por isso não será detalhada, precisava ser construída de forma coerente com a realidade e os conhecimentos já adquiridos.

Foi desenvolvido, então, um método para gerar *moonpools* com diferentes geometrias e foi elaborado um algoritmo capaz de projetar sobre o casco de uma embarcação essas aberturas.

Foi desenvolvido também, em conjunto com a orientadora desse trabalho, Professora Paula Michima, um algoritmo evolutivo para executar a busca do melhor *moonpool*, segundo a avaliação de operabilidade adotada.

O algoritmo evolutivo, que foi elaborado na linguagem Matlab e baseia-se em

parte no algoritmo genético consiste, de forma resumida, no seguinte:

- 1) Gerar os primeiros indivíduos.
- 2) Avaliá-los e classificá-los segundo critérios convenientes.
- 3) A partir dessa informação, selecionar os melhores.
- 4) Com os indivíduos selecionados, criar a próxima geração (inclui, em geral, a possibilidade de mutações).

Voltar ao passo 2.

Naturalmente, o desenvolvimento de cada uma dessas etapas para o caso focado nesse texto implica em desdobramentos que serão detalhados mais adiante.

Para o algoritmo evolutivo, cada *moonpool* será um indivíduo definido por um conjunto de parâmetros, que lhe renderão uma “nota”, um “grau de operabilidade”, a partir do que será possível selecionar o melhor. Considerando, por exemplo, que os parâmetros que definem a geometria e dimensões do furo fossem  $m_1$ ,  $m_2$ ,  $m_3$ ,  $m_4$ ,  $l_1$  e  $l_2$ , os indivíduos seriam como os que podem ser vistos na Figura 3.1.

Figura 3.1 - Conjuntos de valores formando indivíduos.

Parâmetros	Indivíduo 1	Indivíduo 2	Indivíduo 3
$m_1$	0,5	0,8	0,6
$m_2$	1	1,3	0,7
$m_3$	0,9	0,9	0,6
$m_4$	0,5	0,5	0,5
$l_1$	1,2	0,7	0,2
$l_2$	0	0,1	1,4

Fonte: Elaborada pelo autor.

## CAPÍTULO 4 - DESENVOLVIMENTO DAS FERRAMENTAS

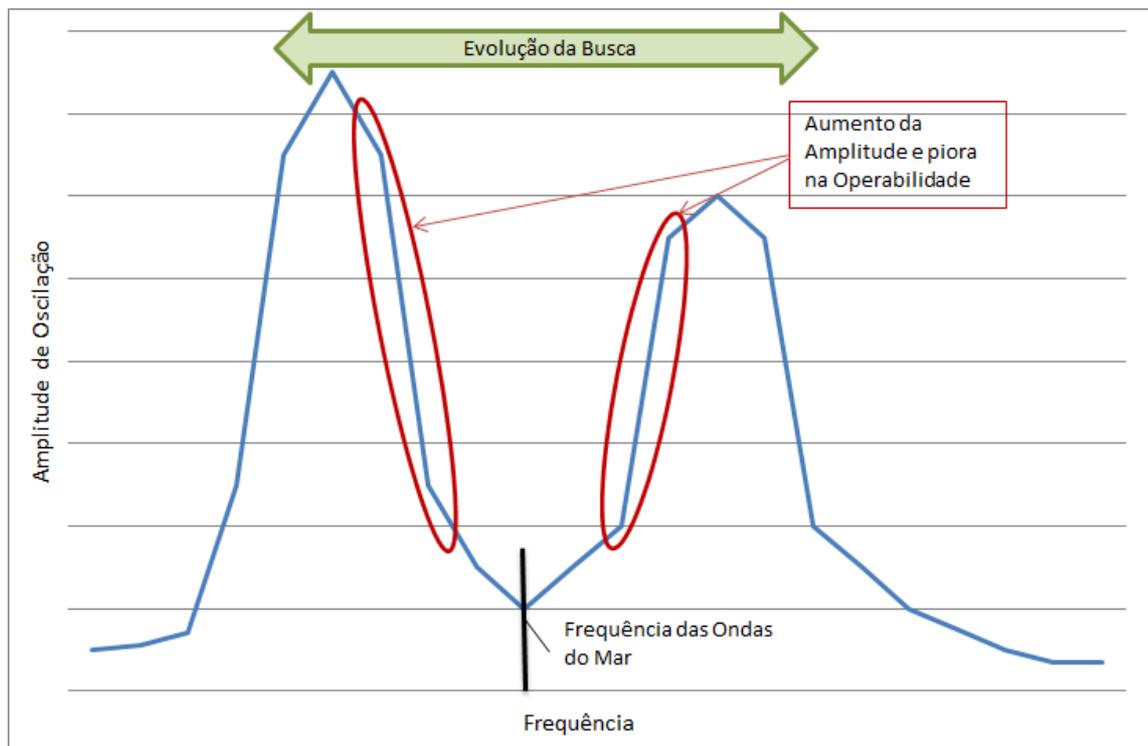
### 4.1 ALGORITMO EVOLUTIVO

Para construir ferramentas eficientes, é preciso conhecer claramente as características do problema a ser enfrentado. No caso em questão, é preciso evitar o crescimento da amplitude das oscilações num sistema em que a velocidade relativa mar-navio é baixa. Ou seja, lembrando Hammargren e Törnblom (2012), é necessário que o sistema mantenha-se afastado das frequências de ressonância. Considerar-se-á, nesse

estudo, que o sistema, como o da Figura 4.1, possui duas frequências de ressonância, como no caso da monocoluna (Torres, 2007), pelo fato desse sistema também não envolver velocidades de deslocamento. Ou seja, há duas, formas possíveis de se trabalhar:

- I. Fazendo com que as duas frequências de ressonância sejam muito menores ou muito maiores que os valores contidos na faixa de frequências típicas das ondas do mar onde o navio opera.
- II. Fazendo com que o intervalo definido entre as duas frequências de ressonância contenha a faixa de frequências típicas das ondas do mar onde o navio opera, com os dois picos de ressonância suficientemente afastados dessa faixa.

Figura 4.1 - Solução presa entre os dois picos de ressonância.



Fonte: Elaborada pelo autor.

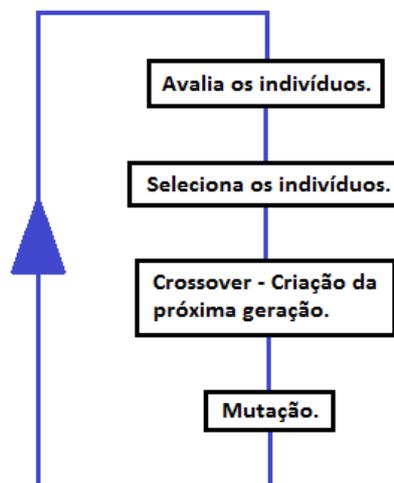
A existência de dois picos de ressonância também cria um inconveniente numérico para algoritmos de busca: se, em algum momento, durante as iterações, a solução momentânea estiver entre os dois picos de ressonância, conforme a evolução da busca for se aproximando de um dos picos, haverá uma tendência de direcionamento

dela para o sentido oposto, passando a caminhar rumo ao outro pico de ressonância, o que poderia deixar a busca da solução “presa” nessa região, restrita à situação (II) (Vide Figura 4.1). Tal condição pode conduzir a soluções instáveis, nas quais pequenos incrementos ou decrementos na frequência das ondas poderiam conduzir a grandes amplitudes de oscilação do fluido interno ao *moonpool*.

É sabido que pequenas mudanças no conjunto de parâmetros do *moonpool* podem mudar completamente a resposta do navio às oscilações do mar. É interessante criar, com base nisso, um mecanismo através do qual o algoritmo de busca possa “saltar” para as outras “regiões possíveis” de soluções para o problema, que sejam mais estáveis.

A fim de conseguir esse efeito, foi elaborado um algoritmo evolutivo diferente do algoritmo genético convencional. Em geral, esse tipo de algoritmo induz mutações em alguns indivíduos recém-gerados após cada crossover, como pode ser visto na Figura 4.2.

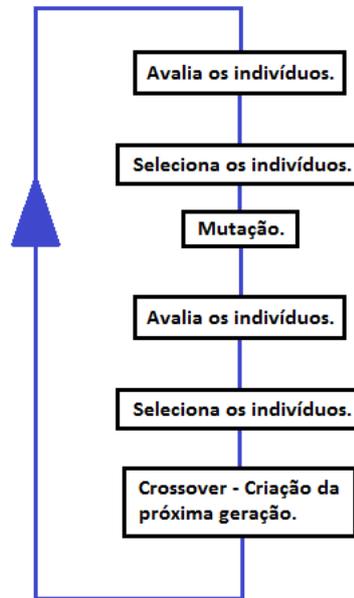
Figura 4.2 - Loop do algoritmo genético convencional.



Fonte: Elaborada pelo autor.

Já no algoritmo proposto nesse trabalho, a mutação é induzida, após uma seleção, em todos os indivíduos. Em seguida, entre originais e mutantes, são selecionados novamente os melhores, daí sim, ocorre o *crossover*. Essa modificação nos permite saltar para outras regiões do universo de indivíduos e pode ser vista Figura 4.3.

Figura 4.3 - Loop do algoritmo genético modificado.



Fonte: Elaborada pelo autor.

O algoritmo genético convencional recebe esse nome por se basear na teoria da evolução de Darwin, em que a mutação ocorre em poucos indivíduos, no momento do surgimento de uma nova geração. O algoritmo proposto nesse trabalho pode também ser classificado como algoritmo genético, por fazer uso do *crossover* como ferramenta para buscar melhores soluções, porém não admite a mutação nessa etapa, em vez disso, cria mutantes a partir de todos os indivíduos selecionados em um determinado grupo, para que sejam avaliados em conjunto com os originais. Tal artifício foi inspirado nas estratégias evolutivas multimembros citadas por Schwefel (1981), das quais se aproxima, mas também nas histórias de ficção da *Marvel Comics*, em que, diversas vezes, mutantes surgem a partir da exposição de indivíduos a radiação gama. Daí o nome com o qual será feita referência ao algoritmo proposto nesse trabalho: Algoritmo *Gamma*.

Outra adaptação feita para que se chegasse ao Algoritmo *Gamma* foi a segregação dos “parâmetros argumentos”, parâmetros independentes que são argumentos de funções que definem outros parâmetros. Por exemplo, se um indivíduo é definido por seis parâmetros, sendo dois “parâmetros argumentos”, três como funções desses dois e mais um parâmetro independente “não-argumento”, aquelas duas

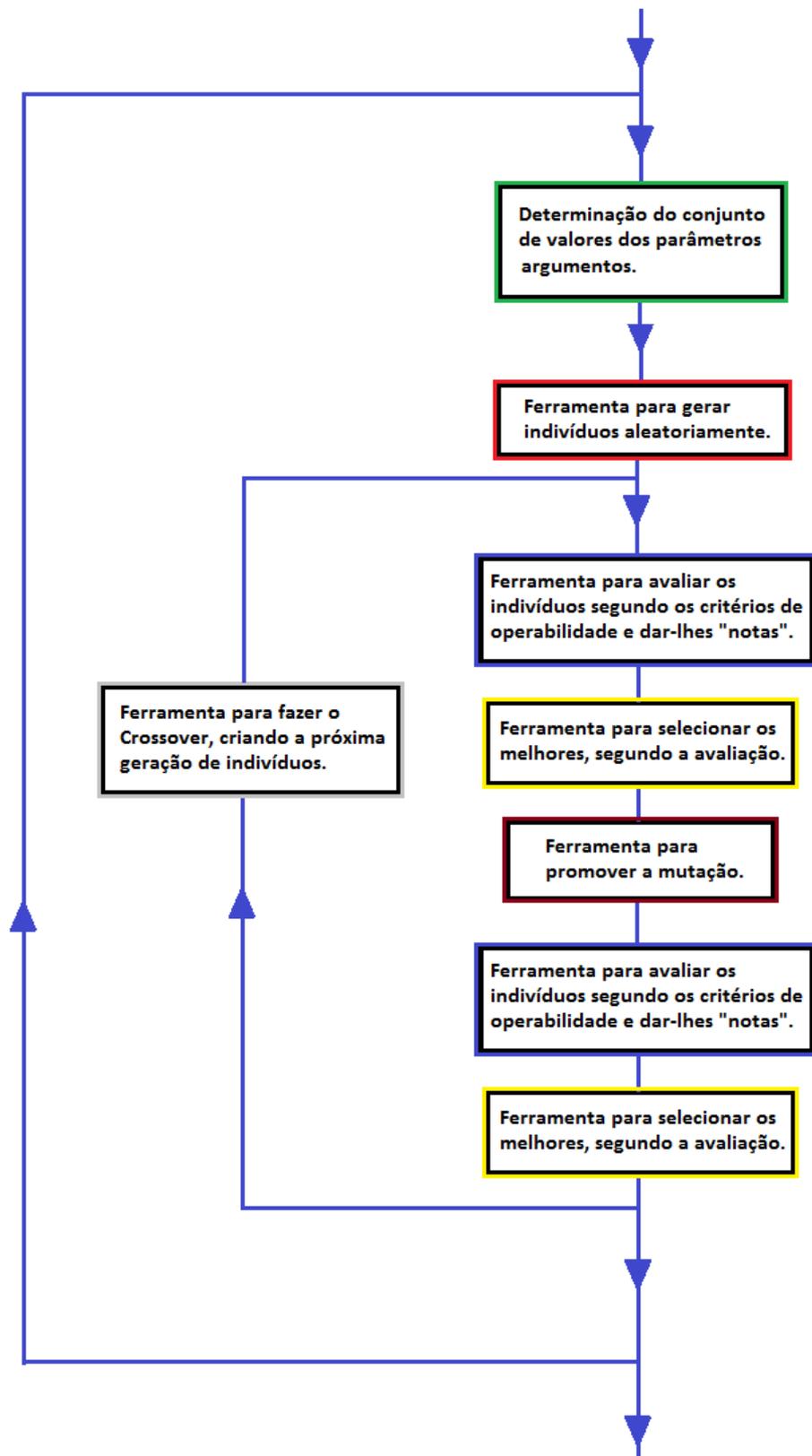
primeiras variáveis não entrarão no *loop* evolutivo, mas serão contemplados via varredura, num laço maior, que contém o *loop* evolutivo. Com isso, garante-se um refinamento maior nas combinações de variáveis, além de uma cobertura de todo o universo de valores possíveis para os parâmetros argumentos, implicando, conseqüentemente, numa melhor cobertura do conjunto dos valores possíveis para as variáveis que são funções deles.

Diante disso, foi elaborado o esquema geral do Algoritmo *Gamma*, que pode ser visto na Figura 4.4, apontando as ferramentas que precisariam ser desenvolvidas para cumpri-los.

Segundo a Figura 4.4, há cinco ferramentas computacionais que precisariam ser desenvolvidas para a construção do programa:

- a. Ferramenta para gerar indivíduos aleatoriamente.
- b. Ferramenta para avaliar os indivíduos segundo os critérios de operabilidade e dar-lhes “notas”.
- c. Ferramenta para selecionar os melhores, segundo a avaliação.
- d. Ferramenta para promover a mutação.
- e. Ferramenta para fazer o *crossover*, criando a próxima geração de indivíduos.

Dessas, apenas a “Ferramenta para selecionar os melhores, segundo a avaliação” não terá seu mecanismo detalhado, pela sua simplicidade, já que se trata de uma busca simples por varredura. Todas as outras terão seus métodos apresentados nas seções 4.2 e 4.3.

Figura 4.4 - Esquema geral do Algoritmo *Gamma*.

Fonte: Elaborada pelo autor.

## 4.2 GERAÇÃO DO *MOONPOOL*

No esquema geral, apresentado na Figura 4.4, vê-se três ferramentas que envolvem a geração de novos indivíduos: “Ferramenta para gerar indivíduos aleatoriamente”, “Ferramenta para promover mutação” e “Ferramenta para fazer o Crossover, criando a próxima geração de indivíduos”. Para criar um indivíduo, é preciso saber o que o define. Já foi dito anteriormente que cada indivíduo corresponde a um *moonpool* e é definido por um conjunto de variáveis que interferem no grau de operabilidade do *drillship*, resta clarificar o que significa cada uma dessas variáveis.

Primeiramente, devido ao foco desse trabalho, que é avaliar o impacto da variação do formato do *moonpool* nas suas características hidrodinâmicas, todas as variáveis pertinentes são referentes à forma geométrica da base de um furo prismático, ou seja, furos com recessos não fazem parte do escopo. Dito isso, definiremos o *moonpool* com seis parâmetros,  $m_1$ ,  $m_2$ ,  $m_3$ ,  $m_4$ ,  $l_1$  e  $l_2$ , da forma que pode ser vista na Figura 4.5, que mostra apenas metade dele, já que se trata de um ente longitudinalmente simétrico.

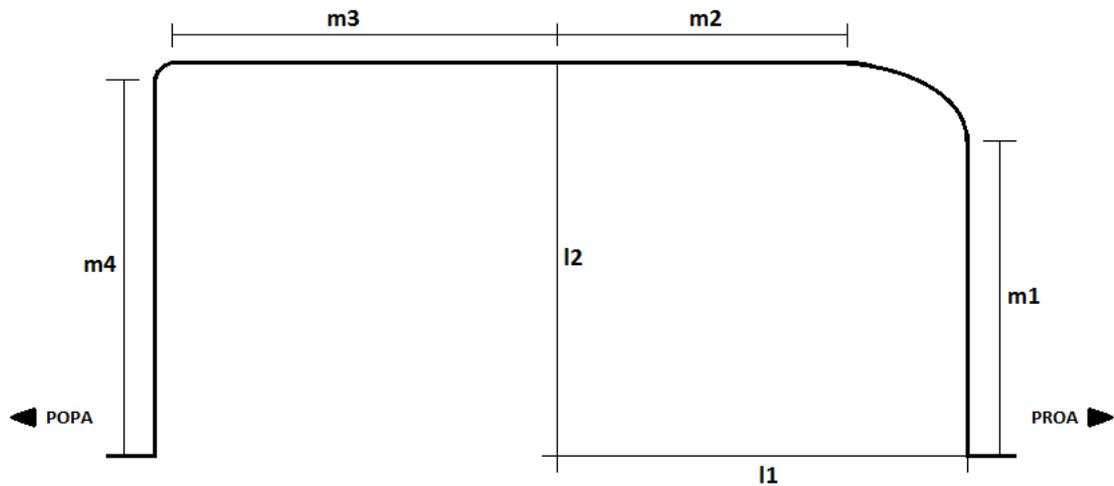
Dessa forma podemos conceituar que todo *moonpool* avaliado nesse trabalho é, geometricamente, um prisma cuja base é um polígono com dimensão transversal máxima de  $2 \cdot l_2$  e dimensão longitudinal máxima de  $2 \cdot l_1$ .

Pode-se entender tal polígono como um retângulo com as quinas suavizadas por polígonos concordantes, de maneira que  $m_1$  e  $m_2$  determinam a posição do centro de um desses polígonos (na verdade, de dois, por simetria), e  $m_3$  e  $m_4$  determinam a posição do centro do outro. Conforme os valores de  $m_1$  e  $m_2$  crescem, aproximando-se, respectivamente, de  $l_2$  e  $l_1$ , a metade do *moonpool* que está à proa se aproxima do formato retangular, por outro lado, conforme eles diminuem, essa parte do *moonpool* se aproxima do formato do polígono que suaviza a quina. O mesmo entendimento pode ser estendido, por analogia, ao efeito de  $m_3$  e  $m_4$  no semirretângulo que está à popa.

Os polígonos que suavizam as quinas do *moonpool* podem ser entendidos como elipses aproximadas por um conjunto de segmentos de reta. As quantidades de segmentos, para cada quina, são determinadas pelos parâmetros, que não aparecem na figura,  $nf$ , para as quinas que estão para a proa, e  $nt$ , para as quinas que estão para a popa. Quanto maiores os valores desses parâmetros, mais próximas de elipses estarão as quinas, por outro lado se tais parâmetros assumirem o valor 1, as quinas terão chanfros

retos.

Figura 4.5 - Parâmetros suficientes para a definição de um indivíduo (*moonpool*).



Fonte: Elaborada pelo autor.

Para consolidar a compreensão de como o Algoritmo *Gamma* manipula essa parametrização, vide Figura 4.4, vale classificar os parâmetros que definem o *moonpool* da seguinte forma:

- $l_1$  e  $l_2$  são parâmetros argumentos, portanto ficarão fora do *loop* evolutivo.
- $m_1$ ,  $m_2$ ,  $m_3$  e  $m_4$  são parâmetros dependentes.
- $nf$  e  $nt$  são parâmetros independentes não-argumentos.

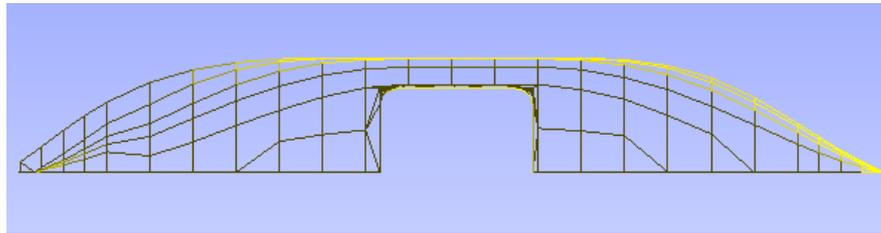
Assim compreende-se que haverá uma varredura formando combinações de  $l_1$  e  $l_2$  e, para cada combinação, haverá uma busca evolutiva dos melhores valores de  $m_1$ ,  $m_2$ ,  $m_3$ ,  $m_4$ ,  $nf$  e  $nt$ , para aquele par  $[l_1, l_2]$ . Ao final da varredura pode-se seccionar o melhor dentre todos os escolhidos em cada combinação.

Variando esses parâmetros, podemos conseguir diversos formatos de furo. Para que se compreenda melhor o poder desse tipo de parametrização, são mostradas, em figuras seguintes, algumas geometrias de *moonpools* que podem ser obtidos a partir dela.

Quando se tem, por exemplo,  $m_1 = m_4 = 0,8 \cdot l_2$  e  $m_2 = m_3 = 0,8 \cdot l_1$ , com  $nf = nt = 10$ , obtém-se um *moonpool* como o que é mostrado na

Figura 4.6, com as quinás arredondadas.

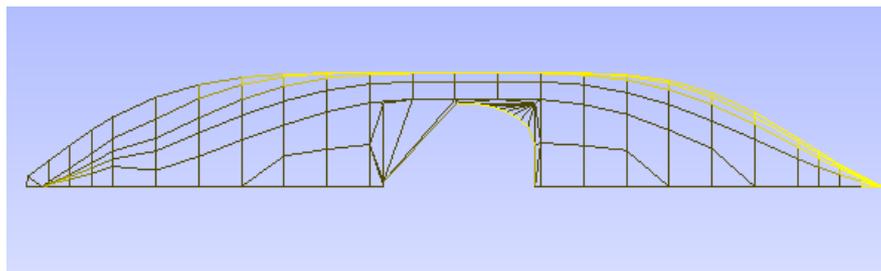
Figura 4.6 - *Moonpool* retangular com quinas arredondadas.



Fonte: Elaborada pelo autor.

Quando se tem, por exemplo,  $m_1 = 0,5 \cdot l_2$ ,  $m_2 = 0,2 \cdot l_1$ ,  $m_3 = 0$  e  $m_4 = 0,1 \cdot l_2$ , com  $nf = 8$  e  $nt = 1$ , obtém-se o que é mostrado na Figura 4.7.

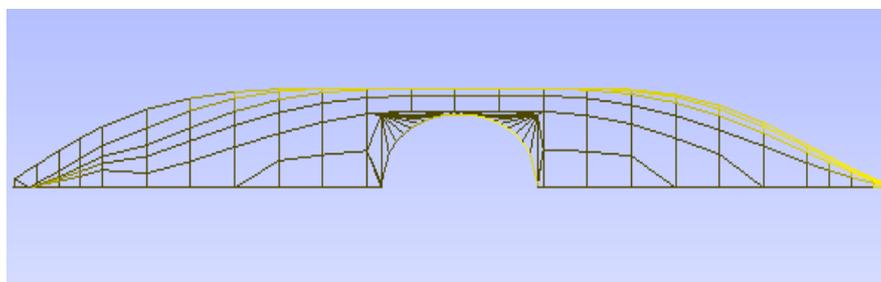
Figura 4.7 - *Moonpool* com metade “em chanfro” e metade retangular com quina suavizada em elipse.



Fonte: Elaborada pelo autor.

Já com  $m_1 = m_2 = m_3 = m_4 = 0$  e com  $nf = nt = 8$ , obtém-se o que é mostrado na Figura 4.8.

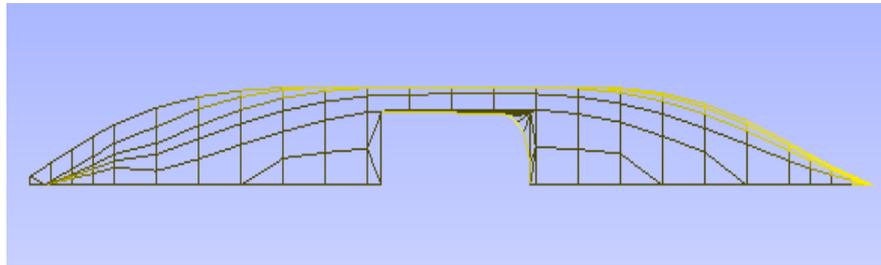
Figura 4.8 - *Moonpool* elíptico.



Fonte: Elaborada pelo autor.

De outra forma, determinando-se  $m_4 = l_2$  e  $m_3 = l_1$ , com  $0 < m_1 < l_2$ ,  $0 < m_2 < l_1$  e  $nf > 1$ , obtém-se o que é mostrado na Figura 4.9.

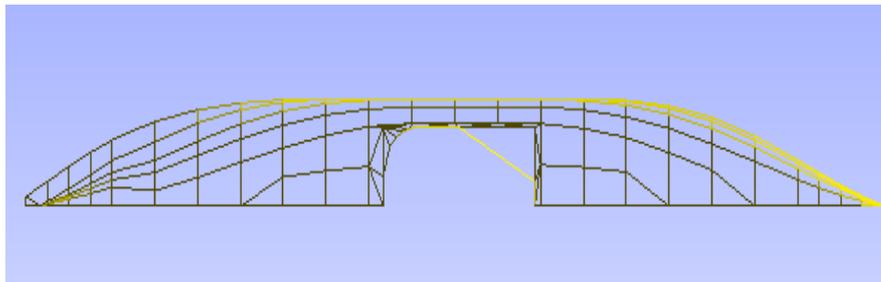
Figura 4.9 - *Moonpool* retangular com apenas uma das quinas suavizada em elipse.



Fonte: Elaborada pelo autor.

Pode-se, da mesma forma, chanfrar as quinas à proa, como mostra a Figura 4.10.

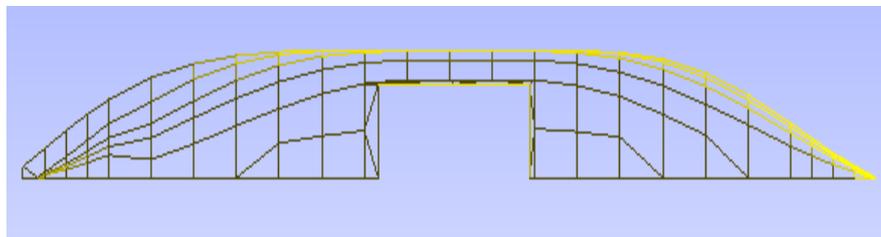
Figura 4.10 - Semelhante à Figura 4.7, com a metade “em chanfro” para a proa.



Fonte: Elaborada pelo autor.

Usando analogia semelhante, fazendo  $m1 = m4 = l2$  e  $m2 = m3 = l1$ , obtém-se o que é mostrado na Figura 4.11.

Figura 4.11 - *Moonpool* retangular sem os arredondamentos nas quinas.



Fonte: Elaborada pelo autor.

Para conseguir as imagens acima, bem como atender um dos objetivos específicos desse trabalho, foi feito um algoritmo para gerar o *moonpool* sobre um casco de navio dado, a partir de um conjunto de pontos e elementos, da seguinte forma:

- 1) Eliminam-se os pontos que tenham as seguintes características, em termos de coordenadas,  $0 < y < l/2$  e  $-l/2 < x < l/2$ .
- 2) Eliminam-se os elementos que contenham esses pontos.
- 3) Acrescentam-se os pontos que forem necessários à geração do polígono da base do prisma (o *moonpool* será sempre uma abertura prismática), a parte que ficará aberta para o mar.
- 4) Constroem-se elementos triangulares nas quinas, acomodando todo tipo possível de disposição de pontos nessa região.
- 5) Copiam-se os pontos da base para o topo, alterando-se apenas as cotas, para gerar a parte que ficará aberta para o convés.
- 6) Constroem-se elementos retangulares associando os pontos do polígono ao mar com os do polígono ao convés.
- 7) Escrevem-se os dados de pontos e elementos num formato em que possa ser lido pelo Gmsh<sup>1</sup>, para visualização.

Os códigos fontes dos algoritmos que executam essas ações estão no Apêndice B.

### 4.3 AVALIAÇÃO DOS INDIVÍDUOS SEGUNDO OS CRITÉRIOS DE OPERABILIDADE

De acordo com os critérios adotados por Michima e Kawabe (2014), vide Tabela 4.1 parâmetros utilizados, segundo os quais a operabilidade do navio é avaliada, devem ser o transbordo da água do *moonpool* (*moonpool overflow*), a limitação do compensador de arfagem (*heave*), a limitação do DPS (*dynamic positioning system*) para compensar a deriva e o momento fletor à meia-nau (no caso em estudo, o máximo módulo de esforço se dá na metade do casco).

Os valores específicos (*specific value*), na tabela, indicam os valores máximos que determinam o limite entre operável e inoperável. Assim sendo, a altura de borda livre (*free board height*) determina a altura máxima que a linha d'água do *moonpool*

---

<sup>1</sup> Gmsh: Software livre para geração de malhas de elementos finitos desenvolvido por Christophe Geuzaine e Jean-François Remacle.

pode atingir e o *stroke* do compensador de arfagem determina a amplitude máxima de oscilação vertical que o navio pode ter e que o equipamento ainda consegue anular ou atenuar. A potência do DPS (*total DPS power*) determina potência instalada máxima, para vencer as forças de deriva (*drift*) mantendo o casco na mesma posição, e o critério da IACS (*International Association of Classification Societies*) indica o módulo máximo aceitável do momento fletor à meia nau (*midship bending*), que é definido pelo valor máximo em 25 anos com probabilidade de exceder de  $10^{-8}$ , baseada em análise de longo prazo. Segundo Michima e Kawabe (2014), esse valor, convertido para o prazo de 1 mês (considerando a distribuição de probabilidade como logarítmica) é aproximadamente  $3 \cdot 10^{-5}$ .

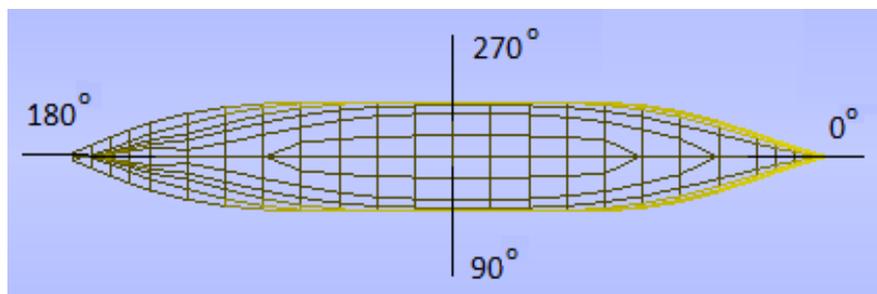
Tabela 4.1 - Tabela dos parâmetros de operação de perfuração.

Item	specific value	
moon pool overflow	free board height	4 m
heave compensator	stroke	3.5 m
Drift	Total DPS power	34260 PS
Midship bending mt.	IACS S11	1month max

Fonte: Michima e Kawabe (2014).

O ângulo de incidência das ondas estudado neste modelo assume os valores de  $0^\circ$  a  $360^\circ$ , com intervalos de  $45^\circ$ , e o sentido é estabelecido com o valor  $0^\circ$  à proa e crescente a boreste, da forma como mostra a Figura 4.12.

Figura 4.12 - Referência para a posição de incidência de onda.



Fonte: Elaborada pelo autor.

A estratégia para a seleção da configuração casco-*moonpool* mais adequado, é gerar para cada indivíduo um mapa de operabilidade que indica o quão segura, em

termos de ausência de interrupção, seria uma operação de perfuração com aquele *moonpool* em um estado de mar com um valor típico de período significativo de onda ( $T_s$ ), que é o período médio do último terço da distribuição.

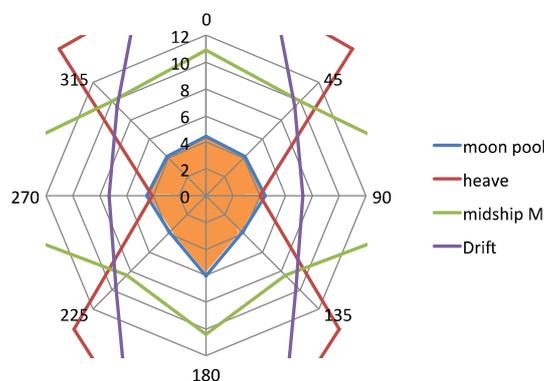
Esse mapa de operabilidade, gerado originalmente por uma ferramenta que simula o comportamento no mar do *drillship* é composto por gráficos sobrepostos, cada um indicando o desempenho do *moonpool* segundo um parâmetro da Tabela 4.1. Essa sobreposição resulta numa região de intersecção cujo valor da área é o que é chamado neste texto como "grau de operabilidade". Esse grau é a “nota de desempenho” do *moonpool*.

Para este trabalho, em caso de dois *moonpools* receberem a mesma nota, o desempate é feito da seguinte forma:

- 1- Será considerado mais adequado o *moonpool* que apresentar o maior valor, no mapa de operabilidade, para  $90^\circ$ .
- 2- Em caso de empate no quesito (1), será considerado mais adequado o *moonpool* que apresentar o maior valor, no mapa de operabilidade, para  $180^\circ$ .

A Figura 4.13, por exemplo, mostra um mapa de operabilidade. O desempenho de uma mesma embarcação varia com o período significativo das ondas excitantes, como se pode observar nos gráficos seguintes, extraídos de Michima e Kawabe (2014).

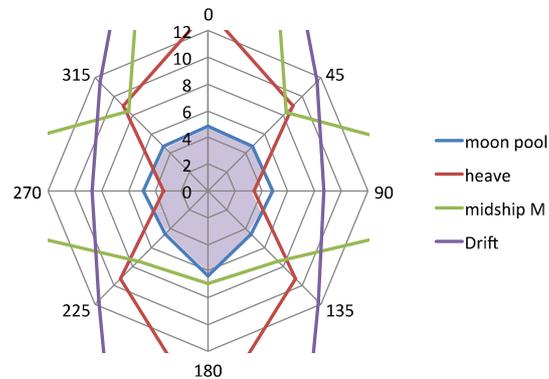
Figura 4.13 - Condições de altura significativa de onda para operabilidade do *drillship* ( $T_s=8s$ ).



Fonte: Michima e Kawabe (2004).

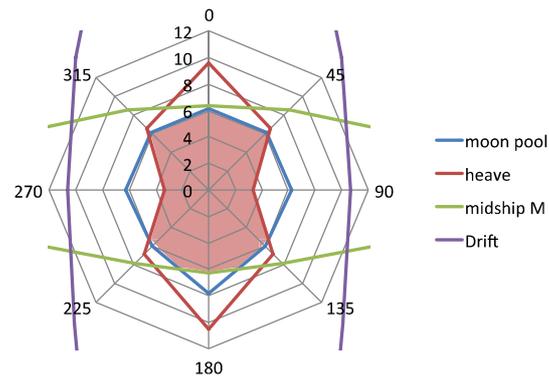
A área colorida, em intersecção, indica a região em que o navio poderia operar atendendo aos critérios supracitados.

Figura 4.14 - Condições de altura significativa de onda para operabilidade do *drillship* ( $T_s=10s$ ).



Fonte: Michima e Kawabe (2004).

Figura 4.15 - Condições de altura significativa de onda para operabilidade do *drillship* ( $T_s=12s$ ).



Fonte: Michima e Kawabe (2014).

Nos exemplos acima, o navio é um Série 60 com as dimensões principais indicadas na Tabela 4.2. Para essa configuração, as frequências naturais de ressonância são  $\omega = 0,76 \text{ rad/s}$ , equivalente a  $T = 8,26s$ , para o modo pistão, e  $\omega = 1,27 \text{ rad/s}$ , equivalente a  $T = 4,94s$ , para o modo *sloshing*. Isso justifica a redução da área de operabilidade a frequências mais altas ( $T = 8s$ , por exemplo) em comparação com o a previsão de desempenho a  $T = 12s$ , equivalente a  $\omega = 0,52 \text{ rad/s}$ . A proximidade do período significativo de onda com o período natural do navio faz com que a resposta à excitação de ondas aumente e, conseqüentemente, a região de operabilidade do navio submetido a essas condições diminua.

Uma busca completa da configuração ótima de *moonpool* que vai operar na região da Bacia de Campos, por exemplo, deve atender no mínimo à faixa de período

significativo entre  $T = 6s$  e  $T = 10s$ , como se pode extrair das extrapolações de Pianca *et al.* (2010) (vide Anexo A).

Dada a complexidade de uma análise mais abrangente, em termos faixa de períodos, no presente trabalho utiliza-se, para fins de validação, o período significativo de onda de  $T_s = 10s$ . Este valor foi escolhido por já haver dados publicados acerca dele fora da região de ressonância (vide Figura 4.14 e Anexo A) e por estar contido no intervalo [9s, 13s], sugerido pela como sendo aquele que concentra os períodos de pico com as maiores alturas significativas de onda, nas estatísticas de 1 ano a 100 anos, para região da Bacia de Campos, onde há a atividade mais intensa de prospecção de petróleo na costa brasileira.

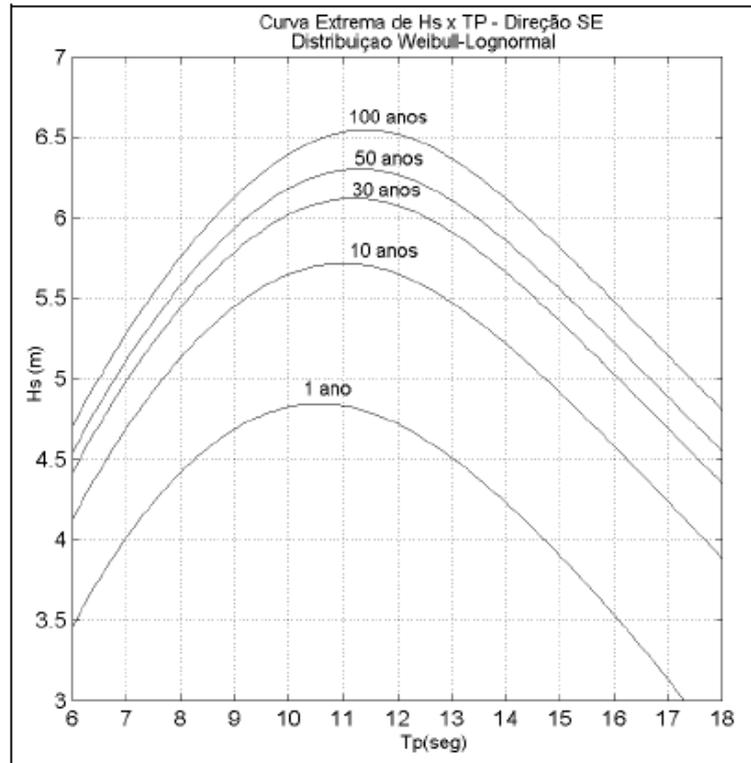
Tabela 4.2 - Dimensões do navio ao qual se referem as Figura 4.13, Figura 4.14 e Figura 4.15.

L	203.4 ( m )
B	29.0 ( m )
d	11.6 ( m )
moon pool length	20.3 ( m )
moon pool breadth	11.9 ( m )
$OG_z$	0.800 ( m )
$k_{xx}/B$	0.469
$k_{yy}/L$	0.243

Fonte: Michima e Kawabe (2014).

Para este trabalho, em vez de sobrepor os gráficos dos parâmetros anteriormente descritos (como na Figura 4.15), buscou-se obter diretamente o polígono que delimitaria a região cuja área indicaria o grau de operabilidade para os *moonpools*. O desenvolvimento do método de geração desse gráfico não será apresentado, por não ser pertinente ao escopo desse estudo.

Figura 4.16 - Distribuição de altura e períodos de picos de ondas na bacia de campos.



Fonte: Ribeiro *et al.* (2004).

## CAPÍTULO 5 - TESTES

Utilizando-se o mesmo casco de Michima e Kawabe (2014), para efeitos de comparação, e assumindo-se os dados de mar recolhidos da literatura, foram feitos testes combinando variações dos parâmetros  $l1$  e  $l2$ , dentro de um intervalo convenientemente determinado, e, para cada par  $[l1, l2]$ , buscou-se, através do Algoritmo *Gamma*, o *moonpool* com o melhor grau de operabilidade.

O intervalo factível para  $l1$  e  $l2$  foi determinado com base nas dimensões do *moonpool* do *drillship* Sertão, construído pela *Samsung Heavy Industries Co Ltda*, que possui comprimento de 25,6m e largura de 12,48m (página *on-line* da Shahin Engenharia). Dessa forma, seria desejável que o retângulo de maior área possível nos nossos testes tivesse aproximadamente essas medidas. Por outro lado, determinou-se

que não seria aceitável uma abertura com área menor que 30% da área do *moonpool* do *drillship* Sertão. Esse valor foi escolhido por se aproximar da área do círculo de diâmetro 12,48m. Assim, determinou-se que os valores possíveis para  $l_1$  e  $l_2$  seriam:

$$l_1=[10; 10,5; 11; 11,5; 12; 12,25; 12,5; 13].$$

$$l_2=[5; 5,25; 5,5; 5,75; 6; 6,25; 6,5; 6,75; 7].$$

Dessa forma,  $l_1=13$  resultaria num comprimento de 26m e  $l_2=7$  resultaria numa largura de 14m, por outro lado,  $l_1=10$  e  $l_2=5$ , com  $m_1=m_2=m_3=m_4=0$  e  $n_f=nt=1$ , resultariam num losango com área de 100m<sup>2</sup>, atendendo a todas as condições requeridas.

Vale ressaltar ainda que, havendo 8 valores possíveis para  $l_1$  e 9 para  $l_2$ , a busca realizada, variando os valores dessas duas variáveis, resultaria em 72 “campeões”, um para cada combinação desses parâmetros.

Assim, ao fim da escolha dos 72, faz-se conveniente a comparação deles entre si, para que se selecione o que tem a maior de todas as bandas de operabilidade.

Foram realizados dois testes idênticos, diferenciados apenas pela geração aleatória do primeiro conjunto de indivíduos, para avaliar, justamente, se isso impõe interferências apreciáveis no resultado final.

Dessa forma foram feitos os testes, cujos resultados são apresentados a seguir.

## CAPÍTULO 6 - RESULTADOS

A evolução dos resultados ao longo das iterações da varredura pode ser vista na Figura 6.1.

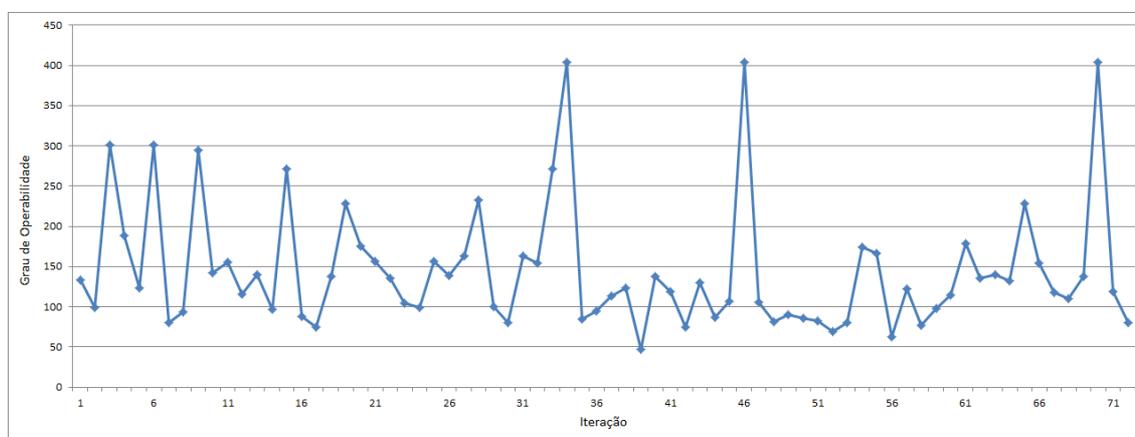
Pode-se observar que os três melhores desempenhos alcançaram valores bastante semelhantes, os das iterações de número 34, 46 e 70, tendo sido o da iteração 46 considerado o de maior grau de operabilidade dentre todos os “campeões” do teste 1.

A fim de relacionar os parâmetros geométricos com os resultados em termos de operabilidade, construiu-se a Tabela 6.1, que mostra os seguintes indivíduos:

- O de melhor desempenho absoluto no teste 1 (T1), obtido na iteração 46 da varredura;
- O de melhor desempenho obtido na iteração 47 (I47-T1), escolhido para

- análise pela semelhança entre os pares  $[l1, l2]$  das iterações 46 e 47;
- O melhor da iteração 34 (I34-T1), escolhido por ter um desempenho semelhante ao da iteração 46);
  - O melhor da iteração 6 (I6-T1), por ser um dos picos locais do teste 1, com grau de operabilidade intermediário,;
  - O de melhor desempenho no teste 2 (T2), escolhido para fins de comparação entre os dois testes.

Figura 6.1 - Evolução das Iterações no teste 1. Melhores notas para cada conjunto  $[l1;l2]$ .



Fonte: Elaborada pelo autor.

Pode-se observar que os dois indivíduos considerados os melhores de cada teste têm um comprimento  $l1$  maior que o valor médio do intervalo possível, o que indica que *moonpools* mais longos longitudinalmente são melhores que os mais curtos, como sugerem Hammargren e Törnblom (2012), quando afirmam que o efeito pistão está mais presente em *moonpools* de comprimento menor. Por outro lado, há uma indicação de que existe uma largura  $l2$  ideal específico para cada comprimento  $l1$ , visto que um pequeno incremento em  $l2$ , da iteração 46 para a 47 do teste 1, implica numa alteração brusca no máximo valor possível de grau de operabilidade na iteração. A mesma característica pode ser observada na Figura 6.10, nas vizinhanças do pico de grau de operabilidade.

Pode se observar também, nas Figura 6.2, Figura 6.4, Figura 6.6, Figura 6.8 e Figura 6.11 que o desempenho a  $180^\circ$  é melhor do que nas outras direções, o que dá especial segurança para a região da popa, inclusive para os hélices. Essa tendência é observada em todos os “campeões” de cada iteração.

Tanto na Tabela 6.1 quanto nas Figura 6.3, Figura 6.5, Figura 6.7, Figura 6.9 e Figura 6.12 pode-se observar a tendência de que o grau de operabilidade cresce conforme os parâmetros  $m_1$ ,  $m_2$ ,  $m_3$ ,  $m_4$ ,  $l_1$ ,  $l_2$  diminuem, ou seja, para os indivíduos gerados no teste, conforme a geometria da abertura se aproxima de um losango com a menor área possível, o desempenho dela melhora. Isso vai ao encontro do que afirma Torres (2007), que indica que quanto menor a área do *moonpool*, menores são as oscilações da água interna.

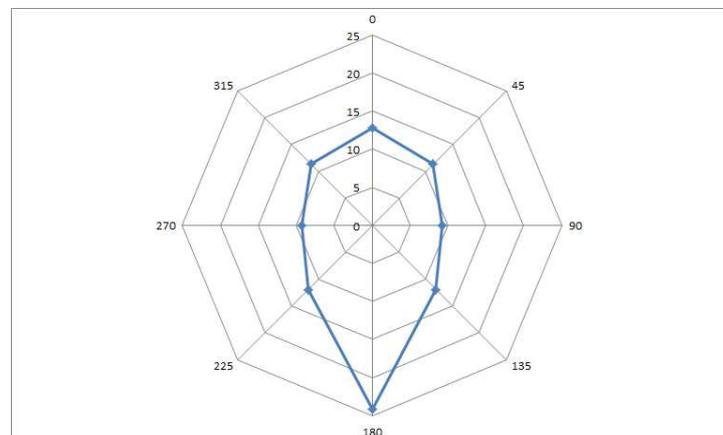
Em comparação com o *moonpool* mapeado na Figura 4.14, pode-se observar que as melhores geometrias dos testes 1 e 2 (Figura 6.2 e Figura 6.11, respectivamente) apresentam grau de operabilidade maior que o daquele.

Tabela 6.1 - Indivíduos tomados como exemplos para efeito de comparações.

	T1	l6 - T1	l47 - T1	l34 - T1	T2
m1	0,56	0	0	0	0
m2	1,36	3,33	4,08	2,56	0
m3	0	0	1,36	1,28	0
m4	1,11	2,08	1,17	0,72	0
nf	1,00	1,00	10,00	1,00	1,00
nt	1,00	3,00	8,00	1,00	1,00
l1	12,25	10,00	12,25	11,50	12,00
l2	5,00	6,25	5,25	6,50	6,00
NOTA	404,12	301,34	105,87	404,12	771,06

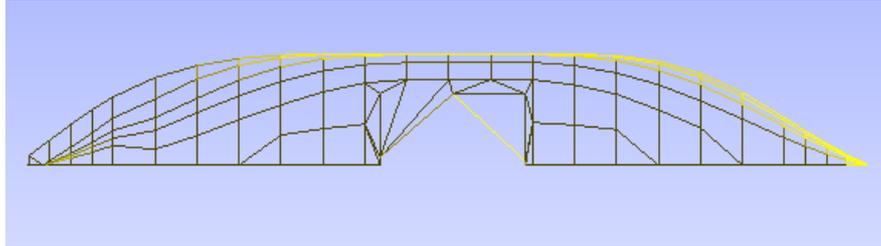
Fonte: Elaborada pelo autor.

Figura 6.2 - Mapa de Operabilidade para a melhor de todas as configurações no teste 1.



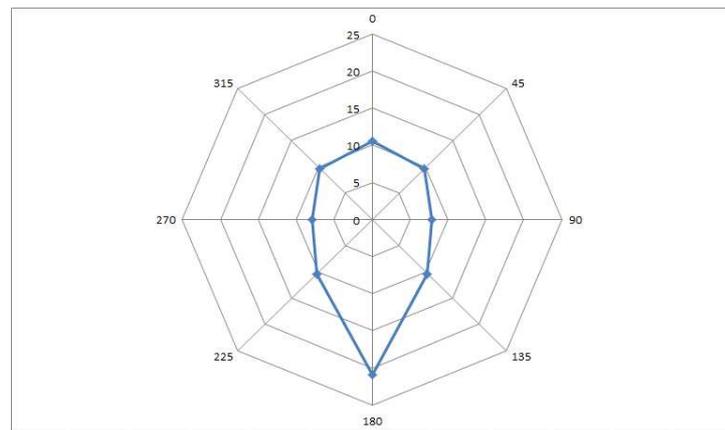
Fonte: Elaborada pelo autor.

Figura 6.3 - *Moonpool* com maior banda de operabilidade no teste 1.



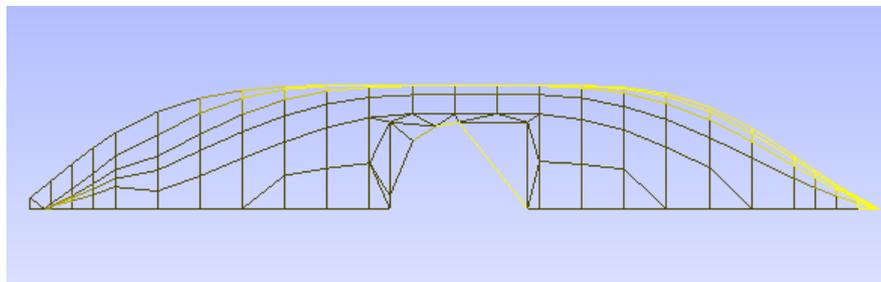
Fonte: Elaborada pelo autor.

Figura 6.4 - Mapa do melhor *moonpool* para  $[l_1=10,00\text{m}; l_2=6,25\text{m}]$  no teste 1.



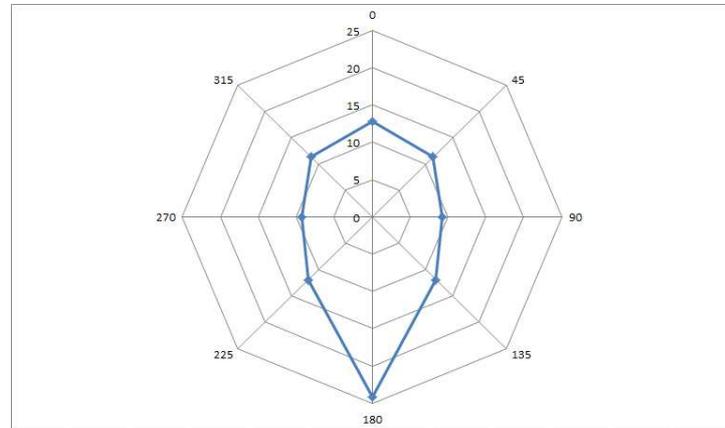
Fonte: Elaborada pelo autor.

Figura 6.5 - Melhor *moonpool* para  $[l_1=10,00\text{m}; l_2=6,25\text{m}]$  no teste 1.



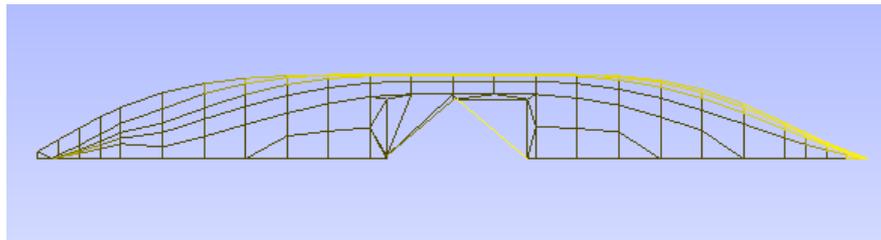
Fonte: Elaborada pelo autor.

Figura 6.6 - Mapa do melhor *moonpool* para [ $l_1=11,50\text{m}$ ;  $l_2=6,50\text{m}$ ] no teste 1.



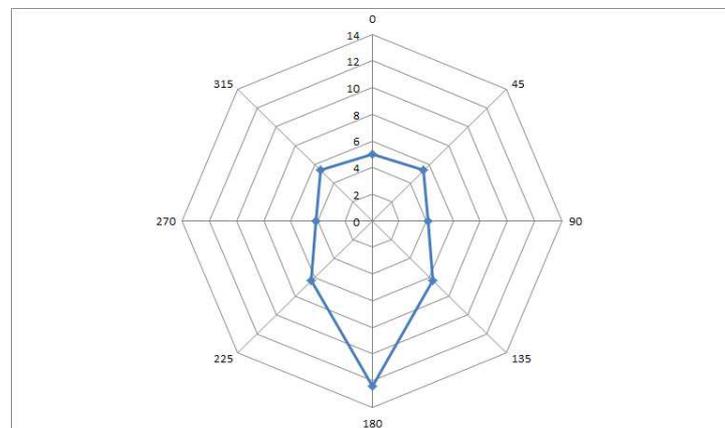
Fonte: Elaborada pelo autor.

Figura 6.7 - Melhor *moonpool* para [ $l_1=11,50\text{m}$ ;  $l_2=6,50\text{m}$ ] no teste 1.



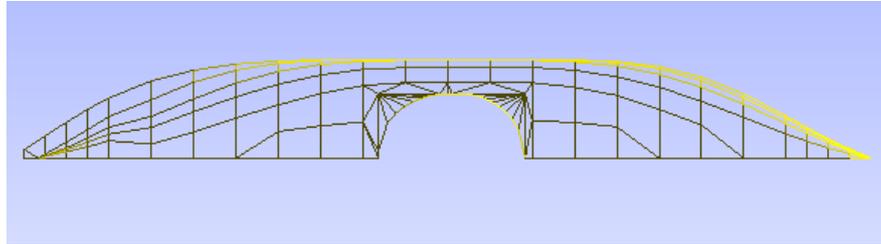
Fonte: Elaborada pelo autor.

Figura 6.8 - Mapa do melhor *moonpool* para [ $l_1=12,25\text{m}$ ;  $l_2=5,25\text{m}$ ] no teste 1.



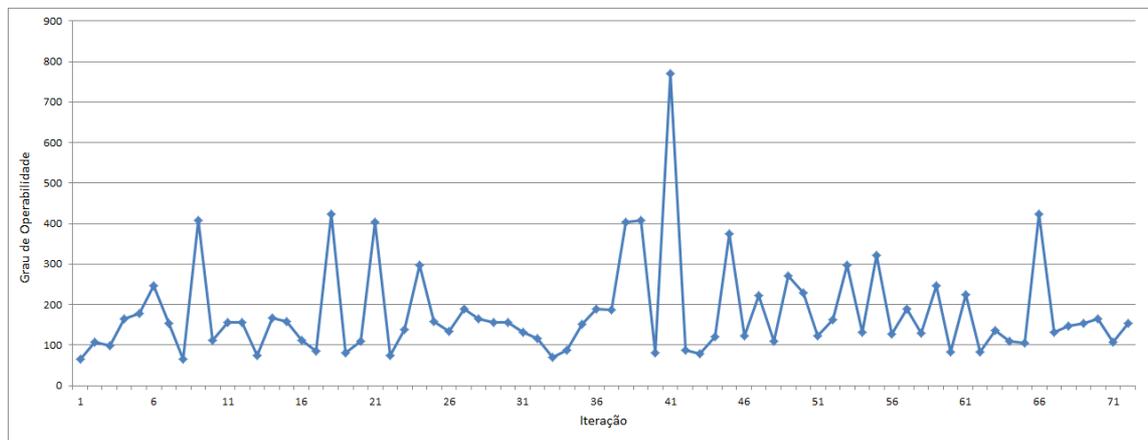
Fonte: Elaborada pelo autor.

Figura 6.9 - Melhor *moonpool* para  $[l_1=12,25m; l_2=5,25m]$  no teste 1.



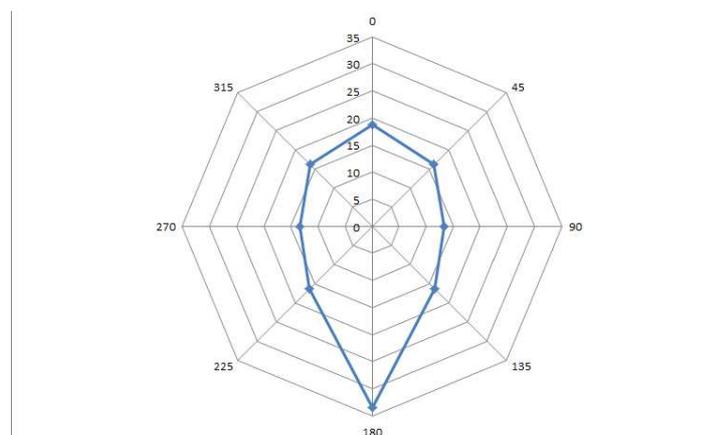
Fonte: Elaborada pelo autor.

Figura 6.10 - Evolução das Iterações no teste 2. Melhores notas para cada conjunto  $[l_1;l_2]$ .



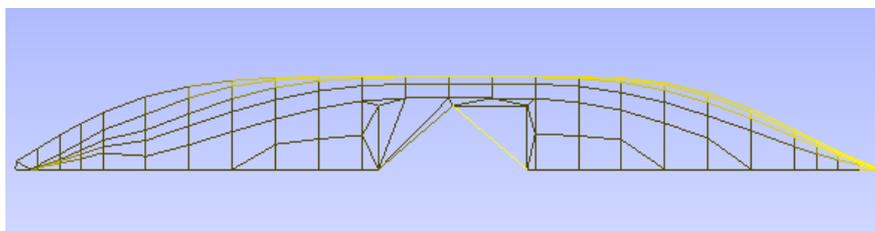
Fonte: Elaborada pelo autor.

Figura 6.11 - Mapa de Operabilidade para a melhor de todas as configurações no teste 2.



Fonte: Elaborada pelo autor.

Figura 6.12 - *Moonpool* com maior banda de operabilidade no teste 2.



Fonte: Elaborada pelo autor.

Deve-se ressaltar também o fato de que os dois testes feitos resultaram em geometrias bastante semelhantes, mesmo partindo de diferentes conjuntos iniciais de indivíduos, o que indica que o método é estável.

Por outro lado, vale ressaltar que as configurações sugeridas não refletem necessariamente as aproximações ótimas reais, dado que não foi aplicada uma formulação numérica que reproduza de forma consistente o do comportamento real do navio em resposta às ondas, na etapa de construção do mapa de operabilidade.

Uma vez que se use tal recurso, cujo desenvolvimento não era pertinente a esse trabalho, para construir gráficos fiéis à realidade, a configuração ótima real pode ser obtida.

## CAPÍTULO 7 - CONCLUSÃO

O presente trabalho alcançou seus objetivos: após o estudo das estratégias propostas por outros autores para solucionar o problema do *moonpool* de *drillship* em outros contextos, desenvolveu-se um algoritmo de busca capaz de indicar a geometria que maximizasse a banda de operabilidade para um determinado estado de mar e um algoritmo capaz de gerar um *moonpool* com diferentes geometrias sobre o casco de uma embarcação numa estrutura de dados determinada.

O método de busca mostrou-se estável, dando resultados concordantes em testes diferentes, indo ao encontro dos conhecimentos adquiridos na literatura. Além disso, é um método teoricamente bastante interessante e que pode ser aprimorado futuramente.

O método de geração de *moonpool* mostrou-se bastante versátil, sendo capaz de gerar aberturas com diferentes formas geométricas (desde elipses até retângulos, incluindo losangos, retângulos com quinas arredondadas, geometrias com parte das quinas arredondada e parte chanfrada, etc.). O conceito do método é extremamente útil e pode ser aproveitado em próximos trabalhos.

## **CAPÍTULO 8 - TRABALHOS FUTUROS**

A ideia por trás do trabalho como um todo é muito fértil e certamente renderá outras obras, para as quais fica a sugestão de que se invista num método consistente de simulação das características hidrodinâmicas do *drillship* em operação, para que se possa construir os gráficos de operabilidade com mais precisão e conseguir um método convergente de seleção de *moonpools*.

## REFERÊNCIAS

2B FIRST CONSULTING. **Drillship**. A imagem está disponível em: <<http://www.2b1stconsulting.com/drillship/>>. Acesso em 16 de junho de 2014.

SHAHIN ENGENHARIA. **Navio-sonda Sertão**. As informações acerca do navio estão disponíveis em: <<http://www.schahin.com.br/pt/areas-de-negocio/petroleo-e-gas/sondas-de-producao/navio-sertao>>. Acesso em 22 de fevereiro de 2015.

AALBERS, A. B. **The water motions in a moonpool**. *Ocean Engineering*, vol. 11, n. 6, 557–579, 1984.

FUKUDA, K. **Behaviour of water in vertical well with bottom openings of ship, and its effect on ship motions**. *Journal of the Society of Naval Architects of Japan*, vol. 141, 107-122, 1977.

GAILLARDE, G.; COTTELEER, A. **Water motion in moonpools empirical and theoretical approach**. Maritime Research Institute Netherlands e HMC Heerema, 2005.

HAMMARGREN, E.; TÖRNBLÖM, J. **Effect of the Moonpool on the Total Resistance of a Drillship**. *Master of Science Thesis*. Goteborgo: Chalmers University of Technology, 2012.

HYE-JONG, S.; MAN-WHAN, K.; SOON-HO, C.; SEUNG-MYUN, H. **Drag reduction of recess type moonpool under vessel's forward speed**. *International Conference on Offshore Mechanics and Arctic Engineering*. Estoril, 2008.

LEWIS, E. V. (Editor). *Principles of Naval Architecture*. Jersey City: The Society of Naval Architects and Marine Engineers, 1988.

MICHIMA, P. S.; KAWABE, H. **Preliminary Chart of Drill Ship Operability Considering Moon-Pool Sloshing Effects**. Recife: Mechanical Engineering Department, Federal University of Pernambuco, 2014.

MOLIN, B. **On the piston and sloshing modes in moonpools**. *Journal of Fluid Mechanics*, pp. vol. 430, pp 27-50, 2001.

PARK, J. J.; KIM, M. S.; LEE, H. S.; AHN, Y. K.; KIM, Y. B.; HA, M. K. **Development of Design Technologies for Optimum Moonpool Shapes of Drill Ship**. *International Conference on Offshore Mechanics and Arctic Engineering*. San Diego, 2007.

PIANCA, C.; MAZZINI, P. L. F.; SIEGLE, E. **Brazilian offshore wave climate based on NWW3 reanalysis.** *Brazilian Journal of Oceanography*. São Paulo, 2010.

SADIQ, S.; XIONG-LIANG, Y. **Multi-Dimensional Numerical Free Surface VOF Modeling with Moon Pool Experiments.** *International Conference on Offshore Mechanics and Arctic Engineering*. Estoril, 2008.

SCHWEFEL, H.-P. *Numerical Optimization of Computer Models*. Chichester: Wiley, 1981.

RIBEIRO, E. O.; LIMA, J. A. M.; NUNES, L. M. P.; GUERRA, L. A. A. **Distribuição Conjunta de Alturas Significativas e Períodos de Pico de Ondas na Bacia de Campos.** *III Congresso Brasileiro de P&D em Petróleo e Gás*. Rio de Janeiro, 2004.

TORRES, F. G. **Estudo do Moonpool como Sistema de Minimização de Movimento em uma Plataforma do Tipo Monocoluna.** *Dissertação de Mestrado*. Escola Politécnica da Universidade de São Paulo. São Paulo, 2007.

VEER, R. V.; THOLEN, H. J. **Added Resistance of Moonpools in Calm Water.** *International Conference on Offshore Mechanics and Arctic Engineering*. Estoril, 2008.

## APÊNDICE A - CÓDIGO FONTE DO ALGORITMO GAMMA.

### 1- Main do Algoritmo Gamma.

```

clear
clc

% Basic Fundamental Settings -----%
ni=10 ; % Total number of individuals to each evaluation.
l1min=0.17; % Minimum length (or bigger radius).
l2min=0.09; % Minimum width (or smaller radius).
l1max=0.27; % Maximum length (or bigger radius).
l2max=0.16; % Maximum width (or smaller radius).
%-----%

% Based on the typical moonpool length and width, we set the limits for L1
% and L2 -----%
L1=[10 10.5 11 11.5 12 12.25 12.5 13] ;
L2=[5 5.25 5.5 5.75 6 6.25 6.5 6.75 7];
y=1 ; % Counter;
%-----%

for w1=1:size(L1,2),
    for w2=1:size(L2,2),

        l1=L1(w1); % Varying the l1;
        l2=L2(w2); % Varying the l2;

        % Parameters generation according to the choiced l1 and l2 -----%
        [ m1, m2, m3, m4, nf, nt, p ] = Geracao_de_Parametros ( l1, l2 );
        % Parameters which will be combined to construct individuals.
        %-----%

        t=1; % Counter to know if it is in the first loop.
        q=1; % Breaking variable.

        while q==1

            if t==1
                [ I ] = Geracao_de_Individuos( p, ni ); % Initial generation just
                % for the first loop.
            else
                [ I ] = Crossover( I4, I, ni ) ; % New generation for all
                % the other loops.
            end

            [ nota, desemp1, desemp2 ] = ...
                Build_Moonpools( I, l1, l2 ) ; % Build and evaluate moonpools
                % in 'I'.

            no=round(0.5*ni) ; % How much percent of the
                % individuals will
                % be selected as the best.

            [ I2, nota2, des21, des22 ] = ...
                Selecao( nota, desemp1, ...
                    desemp2, I, no ) ; % Selection of the 'no' best
                % individuals.

            [ I22 ] = Mutacao( I2, p, no ) ; % Criation of mutants from
                % each individual.

            [ nota, desemp1, desemp2 ] = ...
                Build_Moonpools( I22, l1, l2 ) ; % Build and evaluate moonpools
                % in 'I'.

            %- Select the two best to make the crossover -----%
            [ I4, nota4, des41, des42 ] = ...
                Selecao( nota, desemp1, ...
                    desemp2, I22, 2 ) ; % Select between originals and
                % mutants, the two best.

            %- Breaking criterion -----%
            F(t)=0.5*(nota4(1,1)+nota4(1,2)) ; % Calculate the breaking
                % parameter, average between
                % the bests.

            if t>2

```

```

        [ q ] = Avaliacao( F, q, t ) ; % With the breaking parameter,
                                     % evaluate if it is the moment
                                     % to break.
    end
    t=t+1 ; % Increase 't'.

end

% Keeping important parameters -----%
m1=I4(1,1); %
m2=I4(2,1); %
m3=I4(3,1); % ASSIGN PARAMETERS.
m4=I4(4,1); %
nf=I4(5,1); %
nt=I4(6,1); %
[ A, D1, D2, GR ] = Matriz_Desempenho( l1, l2, m1, m2, m3, m4, nt, nf );

KI(:,y)=I4(:,1) ;
Kn(:,y)=nota4(:,1) ;
KR(y,:)=GR ;
Kl(y,:)=[l1 l2] ;
Kd1(:,y)=des41(:,1);
Kd2(:,y)=des42(:,1);
y=y+1 ;
%-----%

end
end

%- Select the best of the bests -----%
[ I10, nota10, des10, des20 ] = ...
Selecao( Kn, Kd1, ...
Kd2, KI, 1 );

% And keep also its important parameters -----%
for i=1:size(Kn,2),
    if (nota10==Kn(i))&&(des10==Kd1(i))&&(des20==Kd2(i))
        l10=Kl(i,1);
        l20=Kl(i,2);
        GR10=KR(i,:);
    end
end
end
%-----%

```

## 2- Funções.

```

function [ q ] = Avaliacao( F, q, t )
%Function which evaluate if it is the moment to stop looping, changing 'q' from 1 to 0.

a1=F(t)-F(t-1) ; % Compare the current value of 'F' with the last one.

a2=F(t-1)-F(t-2); % Compare the last value of 'F' with the value before it.

if (a1<0.1)&&(a2<0.1)
    q=0; % If in the tree last loops, 'F' did not increase so much, it is better to stop.
end
end

-----

function [ nota, desemp1, desemp2 ] = Build_Moonpools( I, l1, l2 )
% Build and evaluate moonpools -----%

t=1; % Counter.
nota=zeros(1,size(I,2));

for i=1:size(I,2),
    m1=I(1,i); %
    m2=I(2,i); %
    m3=I(3,i); % ASSIGN PARAMETERS.
    m4=I(4,i); %
    nf=I(5,i); %
    nt=I(6,i); %

    % Calculate the Operability of the ship-moonpool -----%
    [ A, D1, D2, GR ] = Matriz_Desempenho( l1, l2, m1, m2, m3, m4, nt, nf );
    %-----%

    % Keep the grades in 'nota' -----%
    nota(t)=A;
    desemp1(t)=D1;

```

```

    desemp2(t)=D2;
    t=t+1;
    %-----%
end

end



---


function [ I ] = Crossover( I4, I, ni )
% Function which generate new individuals as sons of two input individuals.
% The 'father' is the first column of 'I4' and the 'mother' is the second.

for i=1:ni,
    for j=1:size(I4,1),
        a=randi(2,1,1); % If 'a'=1, the new individual receives
                        % information from its 'father'.

        I(j,i)=I4(j,a); % If 'a'=2, the new individual receives
                        % information from its 'mother'.
    end
end

end



---


function [ I ] = Geracao_de_Individuos( p, n )
% This is the random generation of individuals just for the first loop.

for i=1:n,
    for j=1:6,          % There are 6 parameters.
        e=randi(size(p,1),1,1); % Choose a random position in 'p'.
        I(j,i)=p(e,j);      % Assign that position value in 'I'.
    end
end

end



---


function [ m1, m2, m3, m4, nf, nt, p ] = Geracao_de_Parametros (l1, l2)
% Function which fix the available values for each parameter.

dy=l2/9; % This way, we can have ten possible values for m1 and m4.
dx=l1/9; % This way, we can have ten possible values for m2 and m3.

t=1;
for i=0:dy:l2,
    m1(t)=i; % Construction of m1.
    m4(t)=i; % Construction of m4.
    t=t+1;
end

t=1;
for i=0:dx:l1,
    m2(t)=i; % Construction of m2.
    m3(t)=i; % Construction of m3.
    t=t+1;
end

nf=[1 2 3 4 5 6 7 8 9 10]; % number of sides of the quarter of polygon
% which will represent the first rounding
% corner. A 'Ten sides' is enough refined.

nt=[1 2 3 4 5 6 7 8 9 10]; % number of sides of the quarter of polygon
% which will represent the second rounding
% corner. A 'Ten sides' is enough refined.

p(:,1)=m1; %
p(:,2)=m2; %
p(:,3)=m3; % ASSIGNING VALUES.
p(:,4)=m4; %
p(:,5)=nf; %
p(:,6)=nt; %

end



---


function [ A, D1, D2, GR ] = Matriz_Desempenho( l1, l2, m1, m2, m3, m4, nt, nf )
%
pi=3.1415;
a1=l1-m2;
a2=l1-m3;
b1=l2-m1;
b2=l2-m4;

```

```

% Calculate the areas of the corners -----%
if nf<5
    x1=0:a1/nf:a1;
    for i=1:size(x1,2),
        y1(i)=b1*sqrt(1-(x1(i)/a1)^2);
    end
    SA1=0;
    for i=1:size(x1,2)-1;
        v1=[x1(i) y1(i) 0];
        v2=[x1(i+1) y1(i+1) 0];
        c=cross(v1,v2);
        sa=norm(c)/2;
        SA1=SA1+sa;
    end
else
    SA1=0.25*(pi*a1*b1);
end

if nt<5
    x2=0:a2/nf:a2;
    for i=1:size(x2,2),
        y2(i)=b2*sqrt(1-(x2(i)/a2)^2);
    end
    SA2=0;
    for i=1:size(x2,2)-1;
        v1=[x2(i) y2(i) 0];
        v2=[x2(i+1) y2(i+1) 0];
        c=cross(v1,v2);
        sa=norm(c)/2;
        SA2=SA2+sa;
    end
else
    SA2=0.25*(pi*a2*b2);
end
% -----%

% Polynomial Coefficient 'a' -----%
IR=(2*0.27*(0.16^3))/3;
I1=(11*m1^3)/3;
I5=(11*m4^3)/3;
I3=((m2*b1^3)/3)+m2*b1*m1^2;
I4=((m3*b2^3)/3)+m3*b2*m4^2;

if nf==1
    I2=((a1*b1^3)/12)+((a1*b1)/2)*m1;
else
    I2=((pi*a1*b1^3)/16)+((pi*a1*b1)/4)*(m1^2);
end

if nt==1
    I6=((a2*b2^3)/12)+((a2*b2)/2)*m4;
else
    I6=((pi*a2*b2^3)/16)+((pi*a2*b2)/4)*(m4^2);
end
SI=I1+I2+I3+I4+I5+I6;
a=IR/SI;
%-----%

% Area calculation for the moonpool (A) and for a rectangle with the same
% width and length (AR) -----%
A=SA1+SA2+a1*m1+a2*m4+m2*12+m3*12;
AR=2*(11*12);
ar=A/AR; % Aspect Ratio.
%-----%

% Polynomial Coefficient 'b' -----%
E=11*12;
b=1-(E/A);
%-----%

% Polynomial Coefficient 'c' -----%
A1=SA1+a1*m1+m2*12;
A2=SA2+a2*m4+m3*12;
if A1>A2
    c=A2/A1;
else
    c=A1/A2;
end
%-----%

% Allocating the matrix for Pontual Trends -----%
TendPont=zeros(7,11);
TendPont(:,1)=[4; 6; 8; 10; 12; 14; 16];

% Allocating the matrix for Polinomial Coefficient -----%

```

```

CoefPolin=zeros(3,11);
CoefPolin(:,1)=[1; 2; 3];

% Setting original parameters -----
CoefPolinOrigin=zeros(3,11);
CoefPolinOrigin(:,3)=[0.1909; -3.2925; 18.794];
CoefPolinOrigin(:,5)=[0.1462; -2.5484; 16.075];
CoefPolinOrigin(:,7)=[0.2084; -2.2128; 13.388];
CoefPolinOrigin(:,9)=[0.1459; -2.3934; 16.337];
CoefPolinOrigin(:,11)=[0.234; -5.02; 33.43];
%-----

% Filling of Polinomial Coefficient Matrix -----
for i=1:5,
    CoefPolin(1,2*i)=a;
    CoefPolin(2,2*i)=b;
    CoefPolin(3,2*i)=c;
end

for i=3:2:11,
    for j=1:3,
        CoefPolin(j,i)=CoefPolin(j,i-1)*CoefPolinOrigin(j,i);
    end
end
%-----

% Filling of Pontual Trends Matrix -----
for i=3:2:11,
    for j=1:7,
        TendPont(j,i)=TendPont(j,1)^2*CoefPolin(1,i)+ ...
            TendPont(j,1)*CoefPolin(2,i)+CoefPolin(3,i);
    end
end

for i=1:5,
    for j=1:7,
        if TendPont(j,2*i+1)>ar
            TendPont(j,2*i)=TendPont(j,2*i+1);
        else
            TendPont(j,2*i)=ar-round(TendPont(j,2*i+1))+TendPont(j,2*i+1);
        end
    end
end
%-----

% Build the Radar Graph Matrix -----
R=zeros(7,9);
R(:,1)=TendPont(:,1);
R(:,2)=TendPont(:,2);
R(:,3)=TendPont(:,4);
R(:,4)=TendPont(:,6);
R(:,5)=TendPont(:,8);
R(:,6)=TendPont(:,10);
R(:,7)=TendPont(:,8);
R(:,8)=TendPont(:,6);
R(:,9)=TendPont(:,4);
%-----

% Calculate Radar Graph Area Only for Ts=10s -----
alpha=pi/4;
SA=0;
for i=2:8,
    if i==8
        v1=[0 R(4,i) 0];
        v2=[R(4,1)*sin(alpha) R(4,1)*sin(alpha) 0];
    else
        v1=[0 R(4,i) 0];
        v2=[R(4,i+1)*sin(alpha) R(4,i+1)*sin(alpha) 0];
    end
    c=cross(v1,v2);
    sa=norm(c)/2;
    SA=SA+sa;
end
A=SA;
%-----

% Tie break parameters -----
D1=TendPont(4,6);
D2=TendPont(4,10);
%-----

% Radar Graph Vector (Ts=10s) -----
GR=R(4,:);
%-----

end

```

---

```

function [ I22 ] = Mutacao( I2, p, no )
% Function which creates a mutant from each individual.

I22=zeros(size(I2,1),2*size(I2,2)); % This 'I22' will be the matrix
% with the originals and the mutants.

I21=I2 ; % This 'I21' will be the matrix
% with the originals to be modified.

for i=1:size(I2,2),
    e1=randi(size(I2,1),1,3);
    [ H ] = Geracao_de_Individuos( p, no ); % Generate random individuals
% in 'H'.

    I21(e1(1),i)=H(e1(1),i) ; % Modify the individuals from
% 'I21' with parts of the 'H'
% ones.

    I21(e1(2),i)=H(e1(2),i) ; % Same as above.
    I21(e1(3),i)=H(e1(3),i) ; % Same as above.
end

for i=1:size(I2,2),
    I22(:,i)=I2(:,i); % Put the originals in 'I22'.
end

b=size(I2,2);

for i=1:size(I21,2),
    I22(:,b+i)=I21(:,i); % Put the mutants in 'I22'.
end

end

function [ K4, f2, dt1, dt2 ] = Selecao( f, d1, d2, K, a )
% Function which select the individuals with best grades.

f1=f ; % Array of grades.
t=1 ;

% Lookig for the best grades -----%
for j=1:a, % 'a' is the amount of best individuals we want.
    max=0 ;
    dref1=0;
    dref2=0;
    for i=1:size(f1,2),
        if (f1(i)>max)||((f1(i)==max)&&(d1(i)>dref1))||((f1(i)==max)&& ...
            (d1(i)==dref1)&&(d2(i)>dref2))
            max=f1(i) ;
            dref1=d1(i);
            dref2=d2(i);
            c=i ;
        end
    end
    K4(:,t)=K(:,c);
    f2(:,t)=f(:,c);
    dt1(:,t)=d1(c);
    dt2(:,t)=d2(c);
    t=t+1 ;
    f1(c)=0 ;
end
%-----%
end

```

## APÊNDICE B - CÓDIGO FONTE DO PROGRAMA GERADOR DE *MOONPOOL*.

### 1- *Main* do Programa Gerador de *Moonpool*.

```

clear
clc

l1=0.262; %
l2=0.15 ; %
m1=0 ; %
m2=0 ; % ASSIGNING PARAMETERS
m3=0 ; %
m4=0 ; %
nf=1 ; %
nt=1 ; %

% Read the ship file -----%
[ quad,tri,bound,pontos,v,ov,verifica_t,verifica_q,verifica_b, ...
  num_elem, num_vertices, num_elem_t, num_elem_q,...
  num_elem_b, e ] = Pre_Leitor;
%-----%

% 'v' is the points data matrix. 'z' is the lowest z coordinate and o is
% the nearest x coordinate from zero -----%
[ z, o ] = Z_0( v );
%-----%

% Eliminate the points, lines and elements which where in moonpool area --%
[ quad, v, num_elem_q, num_vertices, num_elem, tri ] = ...
  Elim_Linhas_Pontos( v, l1, l2, quad, num_elem_t, num_elem_q, ...
  num_vertices, num_elem, tri );
%-----%

% Build the moonpool -----%
[ p ] = Geom_Pontos( m1, m2, m3, m4, l1, l2, nf, nt, z );
[ v, num_vertices, ov ] = Acrescenta_Pontos( v, p, num_vertices, ov );
%-----%

% Assemble moonpool elements -----%
[ tri, quad, num_elem_t, num_elem, num_elem_q, e ] = ...
  Geom_Elementos( tri, p, quad, v, nf, nt, num_elem_t,...
  num_elem, num_elem_q, e );
%-----%

%Gera um arquivo chamado 'malha1', para ser visualizado no gmsb.
Pre_Escrevedor(quad,tri,bound,pontos,v,ov,verifica_t,verifica_q,verifica_b, ...
  num_elem, num_vertices, num_elem_t, num_elem_q, num_elem_b)

```

### 2- Funções.

```

function [ v, num_vertices, ov ] = Acrescenta_Pontos( v, p, num_vertices, ov )
% Function which add the moonpool points to ship points.

K=size(v,1) ; % 'v' is the matrix os the ship points, without that
for i=1:size(p,1), % which where in the moonpool region.
    v(K+i,:)=p(i,:); % Add moonpool points (p) to 'v'.
end
%

num_vertices=size(v,1); % Adjust the new number of vertices.

for i=1:size(v,1), %
    ov(i)=i ;% Adjust the order number.
end
%

end

function [ z, o ] = Z_0( v )
% Function which find the 'x' value which is nearest from '0' and the
% lowest value for 'z'. -----%

```

```

s=v(:,3); % 'z' values.
z=min(s); % Lowest 'z' values.

% It is known that, in our type of ship file, the 'zero point' is put on
% the midship position, so we need to find the nearest coordinate from the
% 'zero point' to keep the moonpool centralized. -----%

for i=1:size(v,1),
    if (v(i,1)>0)&&(v(i+1,1)<0)
        o=v(i+1,1); % Nearest x coordinate from 'zero point'.
    end
end
end

-----%
function [ quad, v, num_elem_q, num_vertices, num_elem, tri ] = ...
    Elim_Linhas_Pontos( v, a, b, quad, num_elem_t, ...
        num_elem_q, num_vertices, num_elem, tri )
% Function which eliminate, from the ship file, the points and lines which
% were placed on the region where will be constructed the moonpool. -----%

% Verify which are the points which need to be eliminated and zero them. -%
t=1;
for i=1:size(v,1),
    if (v(i,1)<a)&&(v(i,1)>a)&&(v(i,2)<b)
        v(i,:)=0 0 0; % Zero the points which are in the moonpool region.
        vk(t)=i      ; % Keep the zeroed position in 'vk'.
        t=t+1      ;
    end
end
end
-----%

% Adjust the element data, eliminating that ones which are in the moonpool
% region. -----%
for i=1:size(quad,1),
    for j=1:size(vk,2),
        if (quad(i,6)==vk(j))|| (quad(i,7)==vk(j))|| (quad(i,8)==vk(j)) ...
            || (quad(i,9)==vk(j))
            quad(i,:)=zeros(1,9); % Zero elements.
        end
    end
end

% Note that only quadrangles are eliminated because in moonpool available
% region there are only quadrangles.

t=1;
for i=1:size(quad,1),
    if quad(i,1)~=0
        quadl(t,:)=quad(i,:); % Keep the non-zero elements in 'quadl'.
        t=t+1      ;
    end
end

t=1;
for i=1:size(v,1),
    if v(i,1)~=0
        v1(t,:)=v(i,:); % Keep the non-zero points in 'v1'.
        t=t+1      ;
    end
end

quad=zeros(size(quadl,1),size(quadl,2)); % Build new zero 'quad'.
v=zeros(size(v1,1),size(v1,2))      ; % Build new zero 'v'.

%- Adjust the 'order numbers' of the points on the elements. -----%
for i=1:size(quadl,1),
    for j=6:9,
        x=0;
        for k=1:size(vk,2), % Go through 'vk' and
            if quadl(i,j)>=vk(k) % if for each 'vk' position which is lower
                x=x+1; % than quadl(i,j), increases 'x' by one.
            end
        end
        quadl(i,j)=quadl(i,j)-x; % Adjust 'quadl(i,j)'.
    end
end

%-- We need also to adjust the points order numbers in triangles. -----%
for i=1:size(tri,1),
    for j=6:8,
        x=0;
        for k=1:size(vk,2), % Go through 'vk' and
            if tri(i,j)>=vk(k) % if for each 'vk' position which is lower
                x=x+1; % than quadl(i,j), increases 'x' by one.
            end
        end
    end
end

```

```

        end
    end
    tri(i,j)=tri(i,j)-x; % Adjust 'tri(i,j)'.
end
end

quad=quad1;
v=v1;

num_vertices=size(v,1) ; % Adjust the amount of points.
num_elem_q=size(quad,1); % Adjust the amount of quadrangles.

for i=1:size(quad,1),
    quad(i,1)=num_elem_t+i; % Adjust the order numbers of 'quad'.
end

num_elem=num_elem_q+num_elem_t; % Adjust the amount of quadrangles.
% End of adjustment of element data, eliminating that ones which are in
% moonpool region. -----%

end

function [ tri, quad, num_elem_t, num_elem, num_elem_q, e ] = ...
    Geom_Elements( tri, p, quad, v, nF, nt, num_elem_t,...
        num_elem, num_elem_q, e )
% Function which assemble moonpool elements -----%

p1(1)=63; %
p1(2)=62; %
p1(3)=61; %
p1(4)=68; % Handmade assemble of points which were already part of the hull
p1(5)=73; % and will be part of moonpool. This is a limitation of this
p1(6)=78; % program, we did not find a clever way to find, for a generic
p1(7)=83; % ship, the points of the hull which will also be part of the
p1(8)=84; % moonpool.
p1(9)=85; %

for i=1:size(v,1),
    for j=1:size(p,1),
        if v(i,:)=p(j,:)
            p1(j+9)=i; % Assemble of the moonpool points which will also
        end % be part of the hull, but were not part of it yet.
    end
end

U=size(p1,2);
H=size(e,1);
T=H;

%-----%
e(H+1,:)=[H+1 p1(2) p1(1) p1(10) p1(10)] ; %
e(H+2,:)=[H+2 p1(10) p1(11) p1(2) p1(2)] ; %
e(H+3,:)=[H+3 p1(2) p1(11) p1(U-1) p1(U-1)] ; %
e(H+4,:)=[H+4 p1(3) p1(2) p1(U-1) p1(U-1)] ; %
e(H+5,:)=[H+5 p1(3) p1(U-1) p1(4) p1(4)] ; %
e(H+6,:)=[H+6 p1(4) p1(U-1) p1(11+nf) p1(11+nf)] ; % Assigning
e(H+7,:)=[H+7 p1(4) p1(11+nf) p1(5) p1(5)] ; % of
e(H+8,:)=[H+8 p1(5) p1(11+nf) p1(11+nf+1) p1(11+nf+1)] ; % moonpool
e(H+9,:)=[H+9 p1(5) p1(11+nf+1) p1(11+nf+2) p1(11+nf+2)] ; % elements.
e(H+10,:)=[H+10 p1(5) p1(11+nf+2) p1(6) p1(6)] ; %
e(H+11,:)=[H+11 p1(6) p1(11+nf+2) p1(U) p1(U)] ; %
e(H+12,:)=[H+12 p1(6) p1(U) p1(7) p1(7)] ; %
e(H+13,:)=[H+13 p1(7) p1(U) p1(8) p1(8)] ; %
e(H+14,:)=[H+14 p1(U) p1(11+nf+nt+1) p1(8) p1(8)] ; %
e(H+15,:)=[H+15 p1(11+nf+nt+1) p1(11+nf+nt+2) p1(8) p1(8)] ; %
e(H+16,:)=[H+16 p1(11+nf+nt+2) p1(9) p1(8) p1(8)] ; %
H=size(e,1) ; %
for i=1:nF, %
    e(H+i,:)=[H+i p1(U-1) p1(11+i-1) p1(11+i) p1(11+i)] ; %
end %
for i=1:nt, %
    e(H+nf+i,:)=[H+nf+i p1(11+nf+i+1) p1(11+nf+i+2) p1(U) p1(U)]; %
end %
%-----%

%- Assemble the elements also in gmsh format to see the moonpool by gmsh -%
R=size(tri,1);

for i=1:size(e,1)-T,
    tri(R+i,1)=R+i ; %
    tri(R+i,2)=2 ; %
    tri(R+i,3)=2 ; %
    tri(R+i,4)=0 ; % Assign the triangles in gmsh format.
end

```

```

    tri(R+i,5)=6 ; %
    tri(R+i,6)=e(T+i,2); %
    tri(R+i,7)=e(T+i,3); %
    tri(R+i,8)=e(T+i,4); %
end

R1=size(tri,1); % Correct the amount of triangles.
num_elem_t=R1 ; %

for i=1:size(quad,1),
    quad(i,1)=R1+i; % Correct the quadrangle orders.
end

M=0.5*(size(p,1)-2); % Adjustment number to combine points.

for i=1:M-1, % Cobining
    eq(i,:)=[p1(10+M+i-1) p1(10+i-1) p1(10+i) p1(10+M+i)]; % points to make
end % elements.

F=size(quad,1);

for i=1:size(eq,1),
    quad(F+i,1)=F+i ; %
    quad(F+i,2)=3 ; %
    quad(F+i,3)=2 ; %
    quad(F+i,4)=0 ; %
    quad(F+i,5)=6 ; % Assign the quadrangles in gmsh format.
    quad(F+i,6)=eq(i,1); %
    quad(F+i,7)=eq(i,2); %
    quad(F+i,8)=eq(i,3); %
    quad(F+i,9)=eq(i,4); %
end

F1=size(quad,1) ; %
num_elem_q=F1 ; % Adjust the related variables.
num_elem=num_elem-R+R1-F+F1; %
%-----%

end

```

---

```

function [ p ] = Geom_Pontos( m1, m2, m3, m4, l1, l2, nf, nt, z )
% Function which makes the moonpool points data matrix.

if m2==0
    m2=0.0001 ; % To avoid singularity like some ellipse radius being
elseif m2==l1 % zero.
    m2=l1-0.0001; %
end
if m3==0
    m3=0.0001 ; % Same as above.
elseif m3==l1 %
    m3=l1-0.0001; %
end
if m1==0
    m1=0.0001 ; % Same as above.
elseif m1==l2 %
    m1=l2-0.0001; %
end
if m4==0
    m4=0.0001 ; % Same as above.
elseif m4==l2 %
    m4=l2-0.0001; %
end

dx=(l1-m2)/nf ; %
x1(1)=l1 ; % Determine the 'x' coordinates of the first quarter
for i=1:nf, % of ellipse.
    x1(i+1)=l1-dx*i; %
end

dx=(l1-m3)/nt ; %
x2(1)=-l1 ; % Determine the 'x' coordinates of the second quarter
for i=1:nt, % of ellipse.
    x2(i+1)=-l1+dx*i; %
end

a=l1-m2; % First radius of the first ellipse.
b=l2-m1; % Second radius of the first ellipse.
Y1=m1 ; % 'y' coordinate of first ellipse center.
X1=m2 ; % 'x' coordinate of first ellipse center.

for i=1:size(x1,2), % Determine the 'y' coordinates of
    y1(i)=b*sqrt(1-((x1(i)-X1)/a)^2)+Y1; % the first quarter of ellipse.
end

```

```

a=l1-m3; % First radius of the second ellipse.
b=l2-m4; % Second radius of the second ellipse.
Y2=m4 ; % 'y' coordinate of second ellipse center.
X2=-m3 ; % 'x' coordinate of second ellipse center.

t=1 ; %
for j=size(x2,2):-1:1, %
    f(t)=x2(j) ; % Reorder x2.
    t=t+1 ; %
end %
x2=f ; %

for i=1:size(x2,2), % Determine the 'y' coordinates of
    y2(i)=b*sqrt(1-((x2(i)-X2)/a)^2)+Y2; % the second quarter of ellipse.
end

x(1)=l1; % Set the first point of the moonpool to be (l1,0).
y(1)=0 ; %

for i=1:size(y1,2), %
    y(i+1)=y1(i); % Then assemble the first ellipse coordinates to
    x(i+1)=x1(i); % moonpool.
end

x(i+1)=0 ; % Assemble the coordinates of the middle moonpool point.
y(i+1)=l2 ; %
R=size(y,2); %

for i=1:size(y2,2), %
    y(i+R)=y2(i); % Then assemble the second ellipse coordinates to
    x(i+R)=x2(i); % moonpool.
end

R=size(y,2); % Set the last point of the moonpool to be (-l1,0)
x(R+1)=-l1 ; %
y(R+1)=0 ; %

z1=z*ones(1,size(y,2)); % The lowest 'z' coordinate.
z2=zeros(1,size(y,2)); % The highest 'z' coordinate, that is zero, because
% we analyse the submerged part of the ship.

for i=1:size(z1,2),
    p(i,:)=[x(i) y(i) z1(i)]; % First half of moonpool points.
end

for i=1:size(z2,2),
    p(i+size(z1,2),:)= [x(i) y(i) z2(i)]; % Second half of moonpool points.
end

R=size(p,1) ; %
p(R+1,:)=[l1 l2 z] ; % Build the moonpool points data matrix.
p(R+2,:)=[-l1 l2 z]; %

end

function [ quad,tri,bound,pontos,v,ov,verifica_t,verifica_q,verifica_b, ...
    num_elem, num_vertices, num_elem_t, num_elem_q, ...
    num_elem_b, e ] = Pre_Leitor
% Function which reads the data from the ship geometry file and make data
% matrixes. -----

% Catch the amount of points and elements -----
[num_vertices num_elem]=textread('teste.txt',' %u %u', 1);
v=zeros(num_vertices,2) ;
ov=zeros(num_vertices,1) ;
%-----

% Catch the points coordinates from the ship geometry file -----
for i=1:num_vertices,
    [a b c d]=textread('teste.txt','%f %f %f %f',1,...
        'headerlines',i);
    v(i,1)=b ;
    v(i,2)=c ;
    v(i,3)=d ;
    ov(i)=i ;
end
%-----

% Catch the elements from the ship geometry file -----
for i=1:num_elem,
    [e(i,1) e(i,2) e(i,3) e(i,4) e(i,5)]=textread('teste.txt',...
        '%u %u %u %u %u',1,'headerlines',i+num_vertices);
end
%-----

```

```

% Write the element data in gmsh format, this way we can visualize the ship
% computational model by gmsh -----
pontos=[] ;
bound=[] ;
num_elem_b=0;
num_elem_t=1;
num_elem_q=1;

for i=1:num_elem,
    if e(i,4)==e(i,5)
        tri(num_elem_t,1)=num_elem_t+num_elem_b;
        tri(num_elem_t,2)=2 ;
        tri(num_elem_t,3)=2 ;
        tri(num_elem_t,4)=0 ;
        tri(num_elem_t,5)=6 ;
        tri(num_elem_t,6)=e(i,2) ;
        tri(num_elem_t,7)=e(i,3) ;
        tri(num_elem_t,8)=e(i,4) ;
        num_elem_t=num_elem_t+1 ;
    else
        quad(num_elem_q,1)=num_elem_q+num_elem_b+num_elem_t;
        quad(num_elem_q,2)=3 ;
        quad(num_elem_q,3)=2 ;
        quad(num_elem_q,4)=0 ;
        quad(num_elem_q,5)=6 ;
        quad(num_elem_q,6)=e(i,2) ;
        quad(num_elem_q,7)=e(i,3) ;
        quad(num_elem_q,8)=e(i,4) ;
        quad(num_elem_q,9)=e(i,5) ;
        num_elem_q=num_elem_q+1 ;
    end
end

num_elem_t=num_elem_t-1;
num_elem_q=num_elem_q-1;

for i=1:size(quad,1),
    quad(i,1)=num_elem_t+i;
end

if num_elem_t==0
    verifica_t=0;
else
    verifica_t=1;
end
if num_elem_q==0
    verifica_q=0;
else
    verifica_q=1;
end
if num_elem_b==0
    verifica_b=0;
else
    verifica_b=1;
end
%-----

end

function Pre_Escrevedor(quad,tri,bound,pontos,v,ov,verifica_t,verifica_q,verifica_b, ...
    num_elem, num_vertices, num_elem_t, num_elem_q, num_elem_b)
%
fid = fopen('malhal.msh','wt')
fprintf(fid,'%MeshFormat\n')
fprintf(fid,'2.2 0 8\n')
fprintf(fid,'%EndMeshFormat\n')
fprintf(fid,'%Nodes\n')
fprintf(fid,'%u\n',num_vertices)
for i=1:num_vertices,
    fprintf(fid,'%u %f %f %f\n',ov(i),v(i,1),v(i,2),v(i,3))
end
fprintf(fid,'%EndNodes\n')
fprintf(fid,'%Elements\n')
fprintf(fid,'%u\n',num_elem)
for i=1:size(pontos,1),
    fprintf(fid,'%u %u %u %u %u %u\n',pontos(i,1),pontos(i,2),pontos(i,3) ...
        ,pontos(i,4),pontos(i,5),pontos(i,6))
end
if verifica_b==1
    for i=1:num_elem_b
        fprintf(fid,'%u %u %u %u %u %u %u\n',bound(i,1),bound(i,2),bound(i,3) ...
            ,bound(i,4),bound(i,5),bound(i,6),bound(i,7))
    end
end

```

```

end
if verifica_t==1
    for i=1:num_elem_t
        fprintf(fid,'%u %u %u %u %u %u %u %u\n',tri(i,1),tri(i,2),tri(i,3) ...
            ,tri(i,4),tri(i,5),tri(i,6),tri(i,7),tri(i,8))
    end
end
if verifica_q==1
    for i=1:num_elem_q
        fprintf(fid,'%u %u %u %u %u %u %u %u %u\n',quad(i,1),quad(i,2),quad(i,3) ...
            ,quad(i,4),quad(i,5),quad(i,6),quad(i,7),quad(i,8),quad(i,9))
    end
end
fprintf(fid,'$EndElements\n')
fclose(fid)
end

```

### 3- Arquivo de entrada de dados.

#### a. Teste.txt

```

154 126
1      1.509      0      0
2      1.50203    0      -0.06532
3      1.49469    0      -0.13065
4      1.49132    0      -0.16113
5      1.47136    0      -0.17419
6      1.4328     0.02046  0
7      1.4328     0.01818  -0.06532
8      1.4328     0.01564  -0.13065
9      1.4328     0.00988  -0.16113
10     1.4328     0      -0.17419
11     1.3566     0.04223  0
12     1.3566     0.03929  -0.06532
13     1.3566     0.03433  -0.13065
14     1.3566     0.02218  -0.16113
15     1.3566     0      -0.17419
16     1.2804     0.06749  0
17     1.2804     0.06259  -0.06532
18     1.2804     0.05475  -0.13065
19     1.2804     0.03689  -0.16113
20     1.2804     0      -0.17419
21     1.2042     0.09361  0
22     1.2042     0.08763  -0.06532
23     1.2042     0.07604  -0.13065
24     1.2042     0.05362  -0.16113
25     1.2042     0      -0.17419
26     1.0518     0.1426   0
27     1.0518     0.13345  -0.06532
28     1.0518     0.1195   -0.13065
29     1.0518     0.09072  -0.16113
30     1.0518     0      -0.17419
31     0.8994     0.18004  0
32     0.8994     0.17199  -0.06532
33     0.8994     0.1573   -0.13065
34     0.8994     0.126    -0.16113
35     0.8994     0.07335  -0.17419
36     0.8994     0      -0.17419
37     0.747      0.20355  0
38     0.747      0.19854  -0.06532
39     0.747      0.18598  -0.13065
40     0.747      0.15785  -0.16113

```

41	0.747	0.10843	-0.17419
42	0.747	0	-0.17419
43	0.5946	0.21444	0
44	0.5946	0.21237	-0.06532
45	0.5946	0.20467	-0.13065
46	0.5946	0.18063	-0.16113
47	0.5946	0.13931	-0.17419
48	0.5946	0.06966	-0.17419
49	0.5946	0	-0.17419
50	0.4422	0.2177	0
51	0.4422	0.21683	-0.06532
52	0.4422	0.21379	-0.13065
53	0.4422	0.19554	-0.16113
54	0.4422	0.15761	-0.17419
55	0.4422	0.07881	-0.17419
56	0.4422	0	-0.17419
57	0.2898	0.2177	0
58	0.2898	0.2177	-0.06532
59	0.2898	0.21727	-0.13065
60	0.2898	0.20159	-0.16113
61	0.2898	0.16751	-0.17419
62	0.2898	0.08376	-0.17419
63	0.2898	0	-0.17419
64	0.1374	0.2177	0
65	0.1374	0.2177	-0.06532
66	0.1374	0.21727	-0.13065
67	0.1374	0.20159	-0.16113
68	0.1374	0.16785	-0.17419
69	0.1374	0.08392	-0.17419
70	0.1374	0	-0.17419
71	-0.015	0.2177	0
72	-0.015	0.2177	-0.06532
73	-0.015	0.21727	-0.13065
74	-0.015	0.20159	-0.16113
75	-0.015	0.16785	-0.17419
76	-0.015	0.08392	-0.17419
77	-0.015	0	-0.17419
78	-0.1674	0.2177	0
79	-0.1674	0.2177	-0.06532
80	-0.1674	0.21727	-0.13065
81	-0.1674	0.20099	-0.16113
82	-0.1674	0.16785	-0.17419
83	-0.1674	0.08392	-0.17419
84	-0.1674	0	-0.17419
85	-0.3198	0.2177	0
86	-0.3198	0.2177	-0.06532
87	-0.3198	0.21488	-0.13065
88	-0.3198	0.19716	-0.16113
89	-0.3198	0.1613	-0.17419
90	-0.3198	0.08065	-0.17419
91	-0.3198	0	-0.17419
92	-0.4722	0.2177	0
93	-0.4722	0.21694	-0.06532
94	-0.4722	0.20814	-0.13065
95	-0.4722	0.18486	-0.16113
96	-0.4722	0.14351	-0.17419

97	-0.4722	0.07176	-0.17419
98	-0.4722	0	-0.17419
99	-0.6246	0.21574	0
100	-0.6246	0.21085	-0.06532
101	-0.6246	0.19272	-0.13065
102	-0.6246	0.1643	-0.16113
103	-0.6246	0.11833	-0.17419
104	-0.6246	0.05917	-0.17419
105	-0.6246	0	-0.17419
106	-0.777	0.20921	0
107	-0.777	0.19539	-0.06532
108	-0.777	0.16686	-0.13065
109	-0.777	0.13608	-0.16113
110	-0.777	0.0893	-0.17419
111	-0.777	0	-0.17419
112	-0.9294	0.19528	0
113	-0.9294	0.16883	-0.06532
114	-0.9294	0.13145	-0.13065
115	-0.9294	0.10281	-0.16113
116	-0.9294	0.05774	-0.17419
117	-0.9294	0	-0.17419
118	-1.0818	0.17155	0
119	-1.0818	0.12964	-0.06532
120	-1.0818	0.09277	-0.13065
121	-1.0818	0.06814	-0.16113
122	-1.0818	0.03122	-0.17419
123	-1.0818	0	-0.17419
124	-1.2342	0.13367	0
125	-1.2342	0.09252	-0.04355
126	-1.2342	0.06836	-0.0871
127	-1.2342	0.05323	-0.13065
128	-1.2342	0.03871	-0.16113
129	-1.2342	0	-0.17419
130	-1.3104	0.10863	0
131	-1.3104	0.06357	-0.04355
132	-1.3104	0.04507	-0.0871
133	-1.3104	0.03585	-0.13065
134	-1.3104	0.0254	-0.16113
135	-1.3104	0	-0.17419
136	-1.3866	0.08012	0
137	-1.3866	0.0357	-0.04355
138	-1.3866	0.02329	-0.0871
139	-1.3866	0.01934	-0.13065
140	-1.3866	0.01411	-0.16113
141	-1.3866	0	-0.17419
142	-1.4628	0.04964	0
143	-1.4628	0.00936	-0.04355
144	-1.4628	0.00522	-0.0871
145	-1.4628	0.00478	-0.13065
146	-1.4628	0.00443	-0.16113
147	-1.4628	0	-0.17419
148	-1.539	0.01938	0
149	-1.4881	0	-0.04355
150	-1.48413	0	-0.0871
151	-1.48322	0	-0.13065
152	-1.48261	0	-0.16113

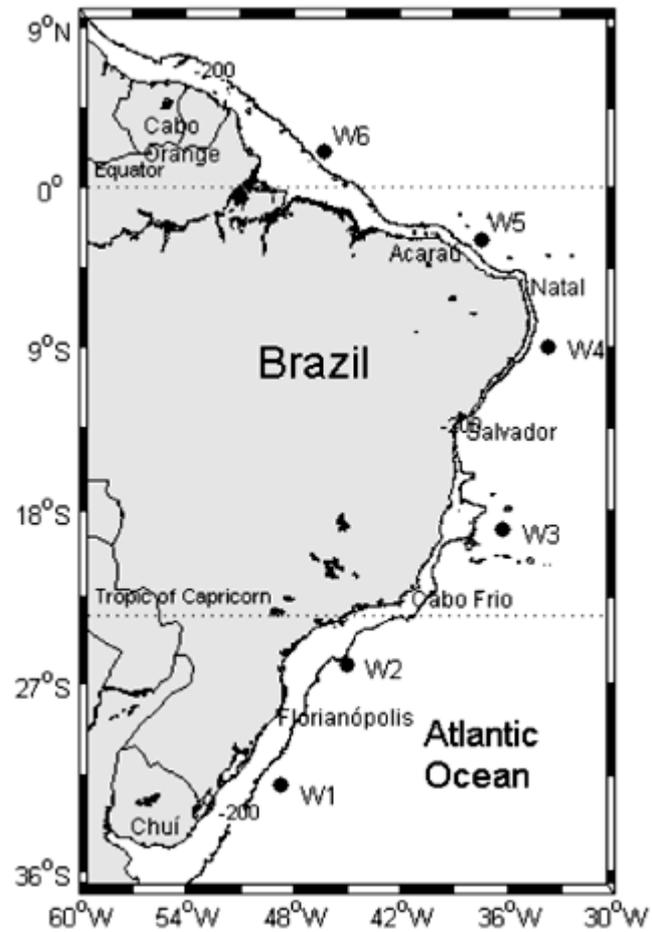
153	-1.48185	0	-0.17419	
154	-1.5415	0	0	
1	1	2	7	6
2	2	3	8	7
3	3	4	9	8
4	4	5	10	9
5	6	7	12	11
6	7	8	13	12
7	8	9	14	13
8	9	10	15	14
9	11	12	17	16
10	12	13	18	17
11	13	14	19	18
12	14	15	20	19
13	16	17	22	21
14	17	18	23	22
15	18	19	24	23
16	19	20	25	24
17	21	22	27	26
18	22	23	28	27
19	23	24	29	28
20	24	25	30	29
21	26	27	32	31
22	27	28	33	32
23	28	29	34	33
24	29	30	35	34
25	30	36	35	35
26	31	32	38	37
27	32	33	39	38
28	33	34	40	39
29	34	35	41	40
30	35	36	42	41
31	37	38	44	43
32	38	39	45	44
33	39	40	46	45
34	40	41	47	46
35	41	42	48	47
36	42	49	48	48
37	43	44	51	50
38	44	45	52	51
39	45	46	53	52
40	46	47	54	53
41	47	48	55	54
42	48	49	56	55
43	50	51	58	57
44	51	52	59	58
45	52	53	60	59
46	53	54	61	60
47	54	55	62	61
48	55	56	63	62
49	57	58	65	64
50	58	59	66	65
51	59	60	67	66
52	60	61	68	67
53	61	62	69	68
54	62	63	70	69

55	64	65	72	71
56	65	66	73	72
57	66	67	74	73
58	67	68	75	74
59	68	69	76	75
60	69	70	77	76
61	71	72	79	78
62	72	73	80	79
63	73	74	81	80
64	74	75	82	81
65	75	76	83	82
66	76	77	84	83
67	78	79	86	85
68	79	80	87	86
69	80	81	88	87
70	81	82	89	88
71	82	83	90	89
72	83	84	91	90
73	85	86	93	92
74	86	87	94	93
75	87	88	95	94
76	88	89	96	95
77	89	90	97	96
78	90	91	98	97
79	92	93	100	99
80	93	94	101	100
81	94	95	102	101
82	95	96	103	102
83	96	97	104	103
84	97	98	105	104
85	99	100	107	106
86	100	101	108	107
87	101	102	109	108
88	102	103	110	109
89	103	104	111	110
90	104	105	111	111
91	106	107	113	112
92	107	108	114	113
93	108	109	115	114
94	109	110	116	115
95	110	111	117	116
96	112	113	119	118
97	113	114	120	119
98	114	115	121	120
99	115	116	122	121
100	116	117	123	122
101	118	119	125	124
102	119	120	126	125
103	120	121	127	126
104	121	122	128	127
105	122	123	129	128
106	124	125	131	130
107	125	126	132	131
108	126	127	133	132
109	127	128	134	133
110	128	129	135	134

111	130	131	137	136
112	131	132	138	137
113	132	133	139	138
114	133	134	140	139
115	134	135	141	140
116	136	137	143	142
117	137	138	144	143
118	138	139	145	144
119	139	140	146	145
120	140	141	147	146
121	142	143	149	148
122	143	144	150	149
123	144	145	151	150
124	145	146	152	151
125	146	147	153	152
126	148	149	154	154

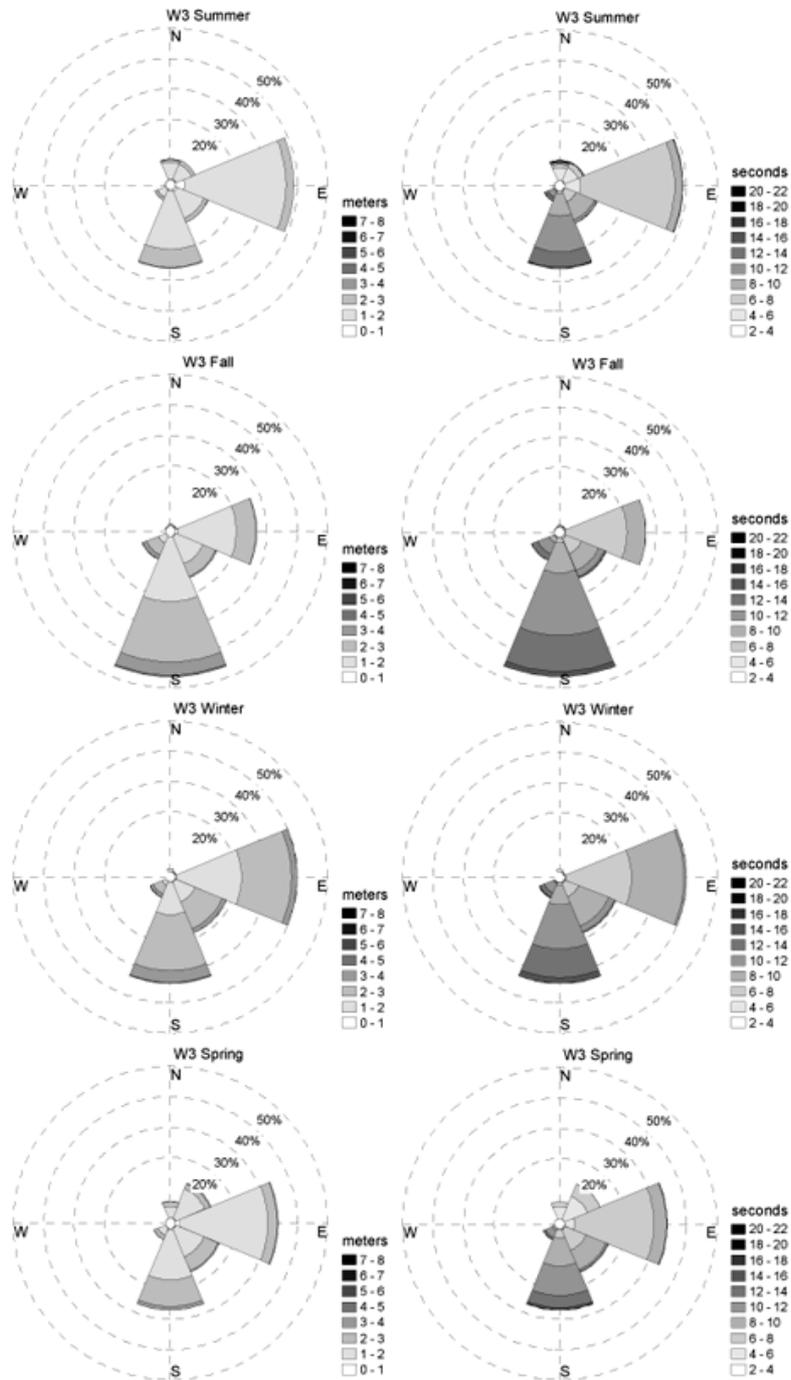
## ANEXO A - INFORMAÇÕES ACERCA DAS CARACTERÍSTICAS DE ONDA NA BACIA DE CAMPOS.

Figura A1 - Localização das áreas de medição próximas à de Campos (W3).



Fonte: Extraído de Pianca *et al.*(2010).

Figura A2 - Histogramas direcionais sazonais das características das ondas para W3. Coluna da esquerda: altura significativa; coluna da direita: período significativo.



Fonte: Extraído de Pianca *et al.*(2010).