

4

Estudo de Casos

Com o objetivo de validar a metodologia de projeto desenvolvida neste trabalho, apresentam-se neste capítulo alguns exemplos de modelagem geométrica por meio de *scripts*. Mostra-se, também a análise de estabilidade de um navio FPSO com relação a sua forma, onde o comportamento da embarcação deve ser avaliado de acordo com os critérios de estabilidade intacta e avariada.

4.1

Parametrização Automática da Forma de um Navio

A seguir, mostram-se alguns exemplos de parametrização automática de navios em relação aos parâmetros *LBP*, *Breadth* e *Depth* do casco de um navio tipo FPSO.

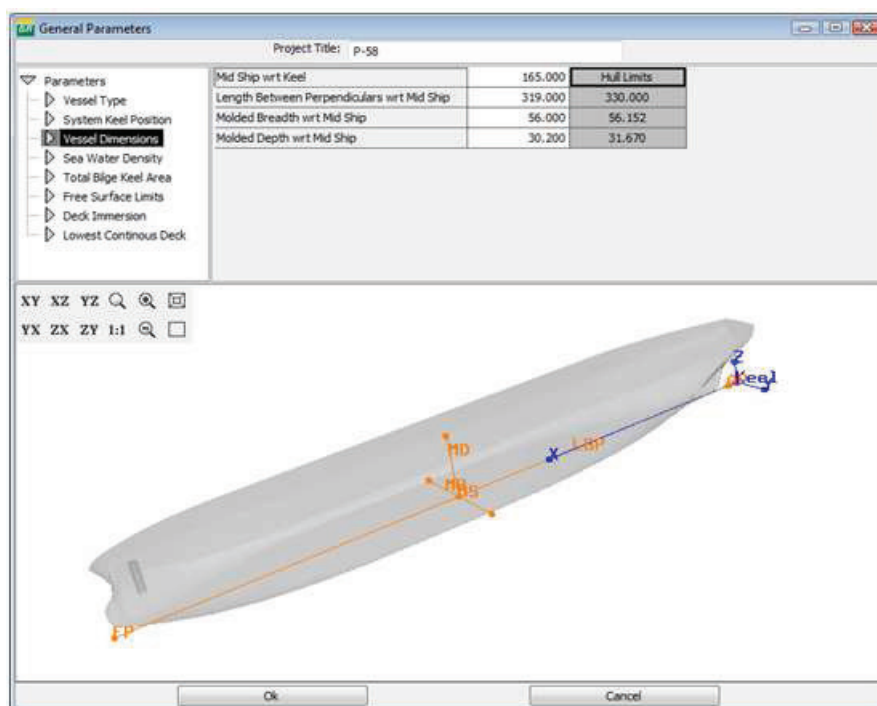


Figura 4.1: Dimensões iniciais do FPSO.

A figura 4.1 mostra o casco de um navio cujos parâmetros estão listados na coluna em cor branca da tabela *Vessel Dimensions*. Estes parâmetros são: *LBP*, *Breadth* e *Depth* cujos valores numéricos são: 319, 56 e 30,2 respectivamente.

O parâmetro *LBP* especifica o comprimento do navio entre as perpendiculares com relação ao eixo X, o parâmetro *Breadth* especifica a boca do navio em relação ao eixo Y e o parâmetro *Depth* especifica a altura do navio em relação ao eixo Z. O valor 165 refere-se ao ponto da seção mestra do navio.

Para modificar de forma automática as proporções do navio, foram implementados três *scripts* em Lua. Estes *scripts* aplicam transformações de escala e translação, respectivamente, para modificar a forma do modelo.

É necessário ressaltar que todas as transformações são feitas de tal forma que o modelo esteja sempre centralizado em relação a sua seção mestra. Isto é, centralizadas em relação à metade do comprimento do navio.

A seguir mostram-se os *scripts* que automatizam a parametrização da geometria do navio FPSO. Estes *scripts* são: *TransformLBP*, *TransformBreadth* e *TransformDepth* respectivamente.

```
algoritmo function TransformLBP(idv1,idv2,NLBP)
1 // get the vertex positions of idv1 and idv2
2 a=mgGetVertex(idv1)
3 b=mgGetVertex(idv2)
4 // compute LBP distance
5 LBP = a:Distance(b)
6 // get factor scale relative to the new value of LBP
7 factor = NLBP/LBP
8 // get the midsection point
9 LBPMx=(a.x+b.x)*0.5
10 LBPMx=(a.y+b.y)*0.5
11 LBPMz=(a.z+b.z)*0.5
12 mgSelect("all")
13 mgTranslateXYZ(LBPMx,LBPMx,LBPMz)
14 mgScaleXYZ(factor,1,1)
15 mgTranslateXYZ(-LBPMx,-LBPMx,-LBPMz)
16 return factor,LBPMx,LBPMx,LBPMz
fim
```

Algoritmo 1: *Script* Lua para parametrizar o navio em relação ao *LBP*

O algoritmo 1 mostra o procedimento necessário para realizar a transformação de um navio em relação ao parâmetro *LBP*. Pode-se perceber que este procedimento realiza três transformações afins. O primeiro passo é calcular o fator de escala que é calculado entre *LBP* atual do modelo e o novo *LBP* informado pelo usuário.

O segundo passo é calcular o valor médio da seção mestra do navio, para poder aplicar o fator de translação. Desta forma, a escala é feita em relação a seção mestra do modelo, conforme salientado anteriormente.

O terceiro, e ultimo passo, é transladar o modelo do navio para a sua posição original.

O procedimento de transformação para mudar o valor do parâmetro *Breadth* é similar ao algoritmo 1, porém a escala é feita em relação ao eixo Y. Este algoritmo é mostrado abaixo.

```

algoritmo function TransformBreadth(idv1,idv2,NBREADTH)
1 // get the vertex positions of idv1 and idv2
2 a=mgGetVertex(idv1)
3 b=mgGetVertex(idv2)
4 // compute Breadth distance
5 BREADTH = a:Distance(b)
6 // get factor scale relative to the new value of NBREADTH
7 factor = NBREADTH/BREADTH
8 // get the midsection point
9 BREADTHMx=(a.x+b.x)*0.5
10 BREADTHMy=(a.y+b.y)*0.5
11 BREADTHMz=(a.z+b.z)*0.5
12 mgSelect("all")
13 mgTranslateXYZ(BREADTHMx,BREADTHMy,BREADTHMz)
14 mgScaleXYZ(factor,1,1)
15 mgTranslateXYZ(-BREADTHMx,-BREADTHMy,-BREADTHMz)
16 return factor,BREADTHMx,BREADTHMy,BREADTHMz
fim

```

Algoritmo 2: Script Lua para parametrizar o navio em relação ao *Breadth*

A transformação do navio em relação ao parâmetro *Depth* é mostrada pelo algoritmo 3 apresentada na próxima pagina.

Todas as transformações aplicadas com relação ao eixo Z devem manter inalterado o sistema de coordenadas locais do navio. Desta forma aplica-se as mesmas transformações do algoritmo 1 com relação ao eixo Z, porém mais uma translação em relação ao eixo Z é feita, considerando o valor do fator de escala.

Isto é, se o fator é maior do que 1, faz-se uma translação positiva no eixo Z, caso o valor seja menor do que 1 é feita uma translação negativa no eixo Z. Nenhuma transformação adicional é necessária se o fator de escala é igual a 1.

Mostrou-se ate este ponto que é possível modificar cada parâmetro que define a forma do navio independentemente. Entretanto estas funções podem ser combinadas indistintamente para obter uma variação mais generalizada com relação às três dimensões principais do navio: *LBP*, *Breadth* e *Depth* como é mostrado no algoritmo 4.

```

algoritmo function TransformDepth(idv1,idv2,NDepth)
  1 // get the vertex positions of idv1 and idv2
  2 a=mgGetVertex(idv1)
  3 b=mgGetVertex(idv2)
  4 // compute Depth distance
  5 DEPTH = a:Distance(b)
  6 // get factor scale relative to the new value of NDEPTH
  7 factor=NDEPTH/DEPTH
  8 // get the midsection point
  9 DEPTHMx=(a.x+b.x)*0.5
 10 DEPTHMy=(a.y+b.y)*0.5
 11 DEPTHMz=(a.z+b.z)*0.5
 12 mgSelect("all")
 13 mgTranslateXYZ(DEPTHMx,DEPTHMy,DEPTHMz)
 14 mgScaleXYZ(1,1,factor)
 15 mgTranslateXYZ(-DEPTHMx,-DEPTHMy,-DEPTHMz)
 16 // keep keel ship at original point relative to Z axis
 17 if (factor>1) then
 18   mgTranslateZ((NDEPTH*0.5)-DEPTHMz)
 19 end
 20 if (factor<1) then
 21   mgTranslateZ(-(DEPTHMz-(NDEPTH*0.5)))
 22 return factor,DEPTHMx,DEPTHMy,DEPTHMz
 23 end
fim

```

Algoritmo 3: *Script* Lua para parametrizar o navio em relação ao *Depth*

A função *TransformShip* do algoritmo 4 faz uso das funções *TransformLBP*, *TransformBreadth*, e *TransformDepth* anteriormente citadas onde os parâmetros de entrada são os novos valores em relação ao comprimento, boca e altura do navio.

```

algoritmo function TransformShip(LBP,Breadth,Depth)
  1 TransformDepth(111,26,Depth)
  2 TransformLBP(76,77,LBP)
  3 TransformBreadth(25,90,Breadth)
  4 end
fim

```

Algoritmo 4: Parametrização do navio em relação aos parâmetros *LBP*, *Breadth* e *Depth*

O resultado da aplicação da função *TransformShip* pode ser visto na figura 4.2, cujos novos valores com relação ao comprimento, boca e altura do novo navio gerado são: 150, 80, e 15 respectivamente. O modelo original pode ser apreciado na figura 4.1.

Salienta-se que se o casco do navio da figura 4.1 possui compartimentos internos, é de fundamental importância que estes compartimentos internos também sejam automaticamente transformados de forma a respeitar a nova geometria do modelo. A transformação destes compartimentos é de fundamental importância para garantir a geração consistente navio transformado.

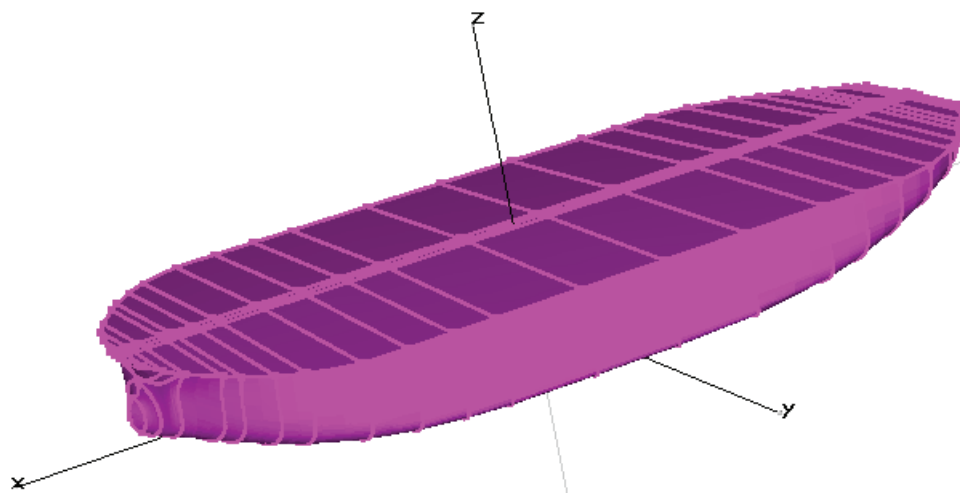


Figura 4.2: Casco externo de um navio modificado pela função *TransformShip*.

4.2

Análise de Estabilidade com Relação à Forma do Modelo

Neste exemplo avalia-se a possibilidade de um navio FPSO ser capaz de absorver um determinado carregamento referente a um módulo a ser posicionado no convés, que eleva o centro de gravidade do sistema global de pesos atuante. Este módulo está sendo considerado proporcional à boca do navio.

O modelo geométrico do navio FPSO é formado por um conjunto de compartimentos independentes, onde cada um deles possui uma fronteira individual. Isto é, tanques internos e externos (lastro, petróleo, óleo diesel, espaços vazios, paiol de amarras, etc).

O casco do FPSO modelado no MG e estudado nesta seção possui as seguintes dimensões: um comprimento entre as perpendiculares em torno de 319 metros, uma boca de 56 metros e uma altura de 30,2 metros.

Este casco é modelado completamente utilizando um plano de simetria com apenas 14 superfícies definidas como superfícies de coons bilineares [Coons64] e por uma única superfície do tipo *skinned* [Lira02] que modela grande parte do costado do navio. A figura 4.3 mostra o plano de balizas e as superfícies geradas do casco do FPSO estudado neste exemplo.

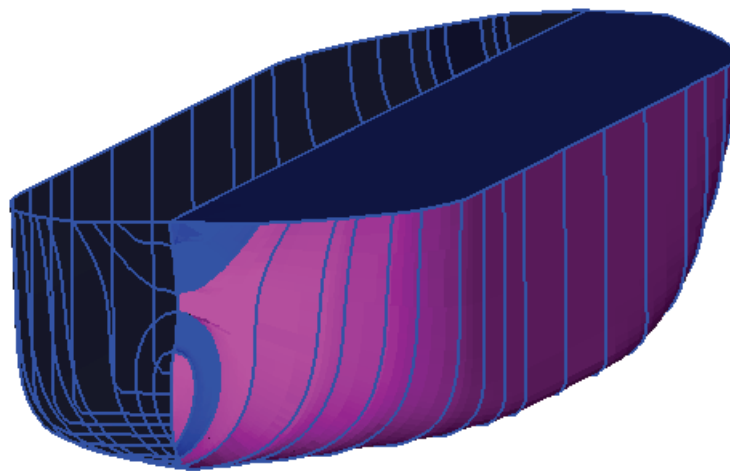


Figura 4.3: Casco de um navio modelado por superfícies *coons* e *skinned*.

O comportamento de uma estrutura flutuante é avaliado de acordo com os critérios de estabilidade intacta, apenas sob a ação do vento, e de acordo com os critérios de estabilidade avariada, após sofrer uma avaria resultando em perda do empuxo da embarcação. O conceito de estabilidade refere-se a uma posição adequada do metacentro (estabilidade inicial) e a uma razão maior

que zero entre o momento de restauração e o momento de tombamento. Estes critérios estão regulamentados por organizações internacionais tais como a IMO (*International Maritime Organization*) [IMO78], NMD (*Norwegian Maritime Directorate*), entre outros.

Na fase de projeto do FPSO, as condições de carregamento são organizadas em classes que usualmente separam os casos das condições intactas e avariadas, onde estas condições são bem definidas pela IMO.

Um destes critérios indica que o modelo a ser testado deve passar por situações de avarias. Logo, partindo-se de uma condição intacta, deve-se também simular algumas situações de avaria para um tanque ou grupo de tanques. Os requisitos mínimos de estabilidade após a avaria segundo a MARPOL 73/78 [IMO78] são:

- Os valores positivos do braço de endireitamento da curva de estabilidade estática devem se estender por uma faixa de no mínimo 20 graus além da posição de equilíbrio.
- A área sob a curva dos braços de endireitamento até 20 graus após o ponto de equilíbrio ou até a imersão de um tanque não deve ser menor do que 0,0175 rad.m, em associação a um braço de endireitamento residual máximo de no mínimo 0,10 metros.
- No estágio final, o ângulo de inclinação não deverá exceder 25 graus.

A representação gráfica do critério de estabilidade em avaria pode ser vista na figura 4.4.

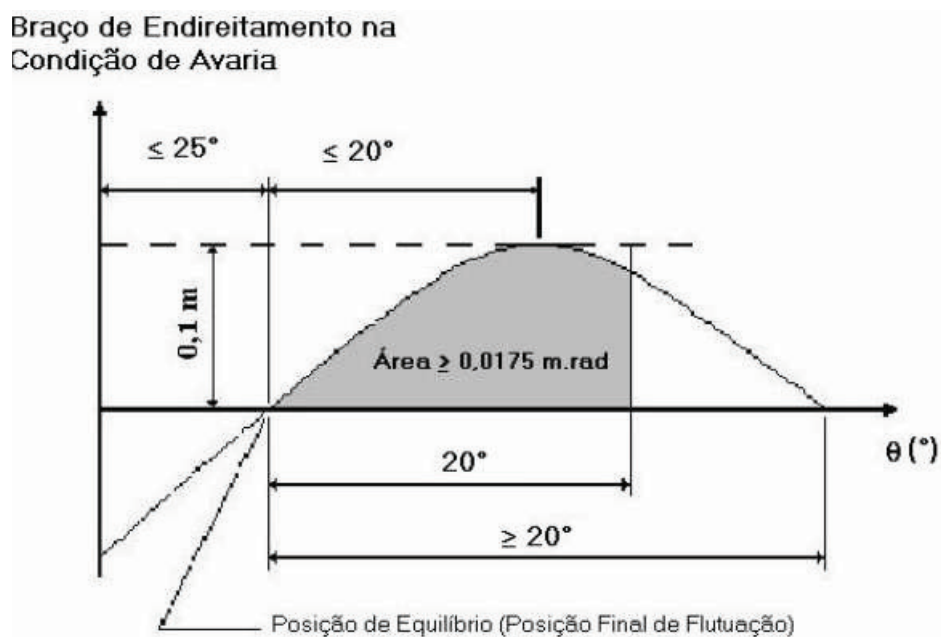


Figura 4.4: Critério de estabilidade em avaria.

4.2.1

Condição Inicial de Avaria

A seguir mostra-se a figura do caso inicial do navio FPSO, onde a avaria escolhida é chamada de DC (*Damaged Condition*) e representa o caso mais crítico dentre as avarias de projetos definidas pelas sociedades classificadoras (IMO, NMD, etc).

Esta avaria consiste no abaloamento de uma antepara, o que permite a entrada de água em dois compartimentos de bombordo, e do alagamento simultâneo do paiol de amarras. Os tanques avariados estão mostrados em vermelho na figura 4.5 e realçados em torno de uma circunferência de forma a localizar o problema.

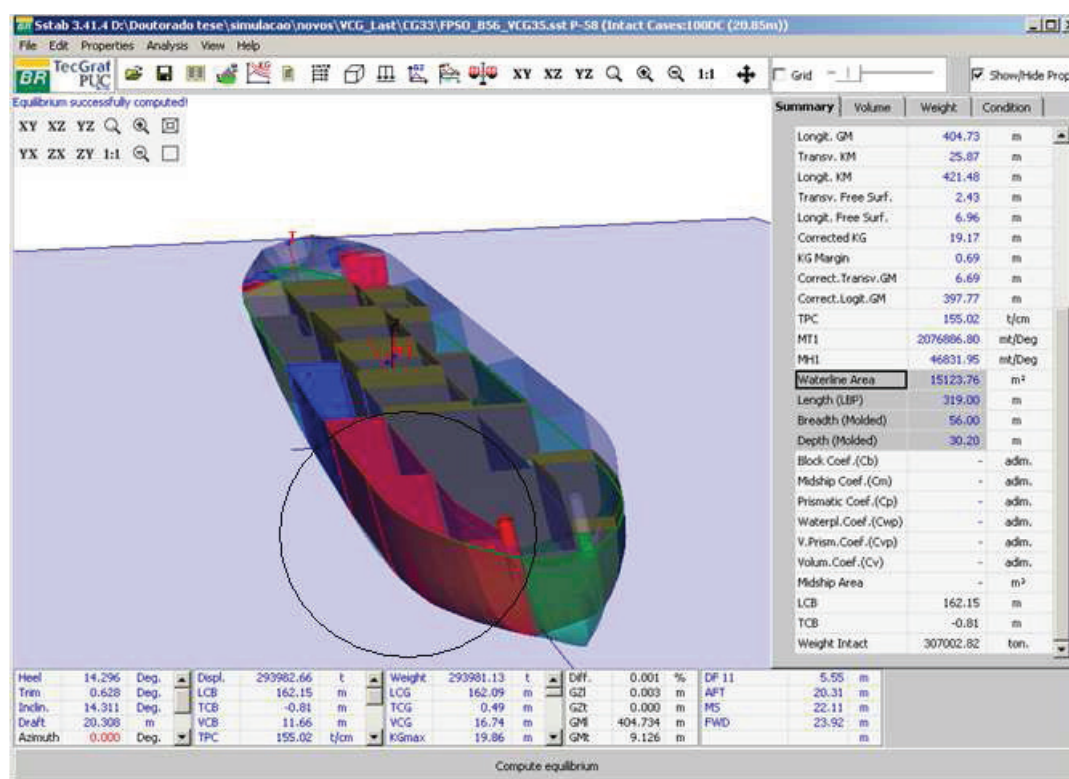


Figura 4.5: Condição avariada do navio FPSO cuja boca é 56 metros.

Após o cálculo do equilíbrio da plataforma, verifica-se um ângulo de 14,29 graus de banda para bombordo e um trim elevado de proa, com borda livre de bombordo de apenas 2,43 metros. Como resultado adicional obtêm-se arquivos contendo as informações de equilíbrio mostradas na tabela *Summary* da figura 4.5.

Os resultados do diagrama de estabilidade do FPSO, cuja boca original é 56 metros, estão disponíveis também em arquivo. Apresenta-se uma visualização gráfica destes resultados com a plotagem do diagrama de

estabilidade (figura 4.6) e a sua correspondente tabela numérica, onde aparece um resumo das restrições impostas pela MARPOL 73/78 (figura 4.7).

Uma vez analisadas as condições de avaria de projeto, a qual o FPSO está sujeito, verificou-se que a condição avariada DC não atende apenas o critério de faixa de estabilidade mínimo de 20 graus, definido pela MARPOL 73/78. Isto pode ser visto na tabela *Marpol Damaged* da figura 4.7, onde o critério de faixa de estabilidade está assinalado em vermelho e destacado por uma elipse. Todos os outros critérios estão respeitando as restrições da MARPOL 73/78.

Considerou-se para este exemplo o VCG (*Vertical Center Gravity*) do peso leve da estrutura, após o embarque do módulo, em torno de 33 metros.

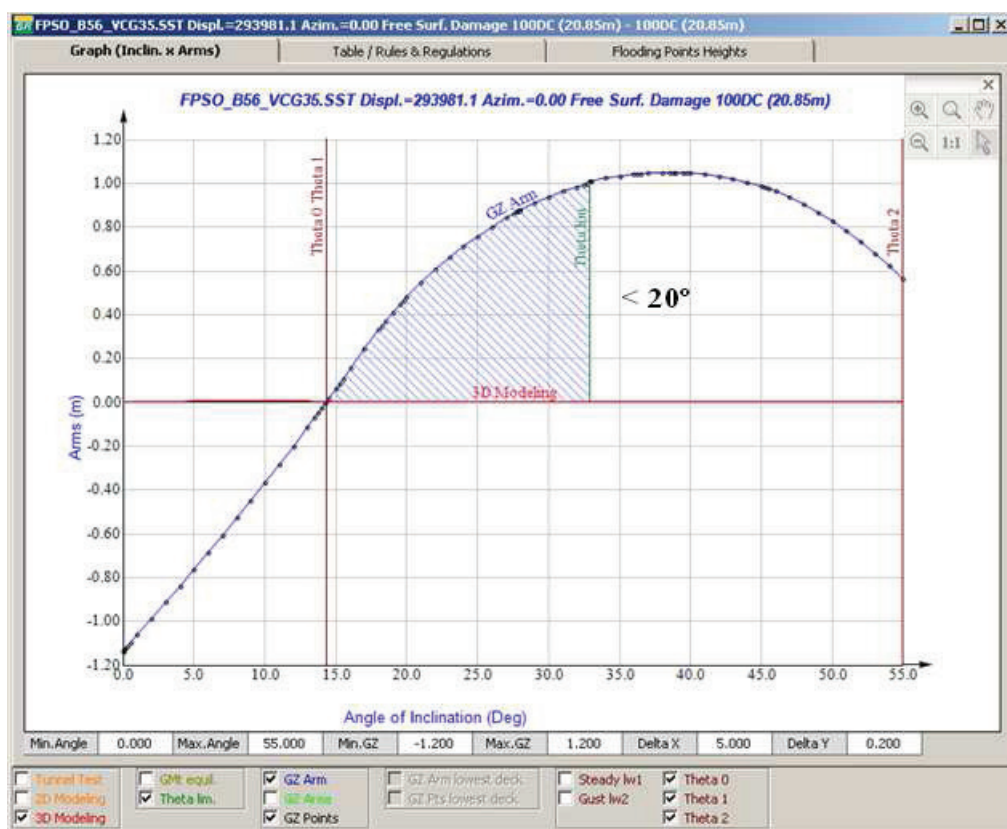


Figura 4.6: Diagrama da estabilidade da FPSO cuja boca é 56 metros.

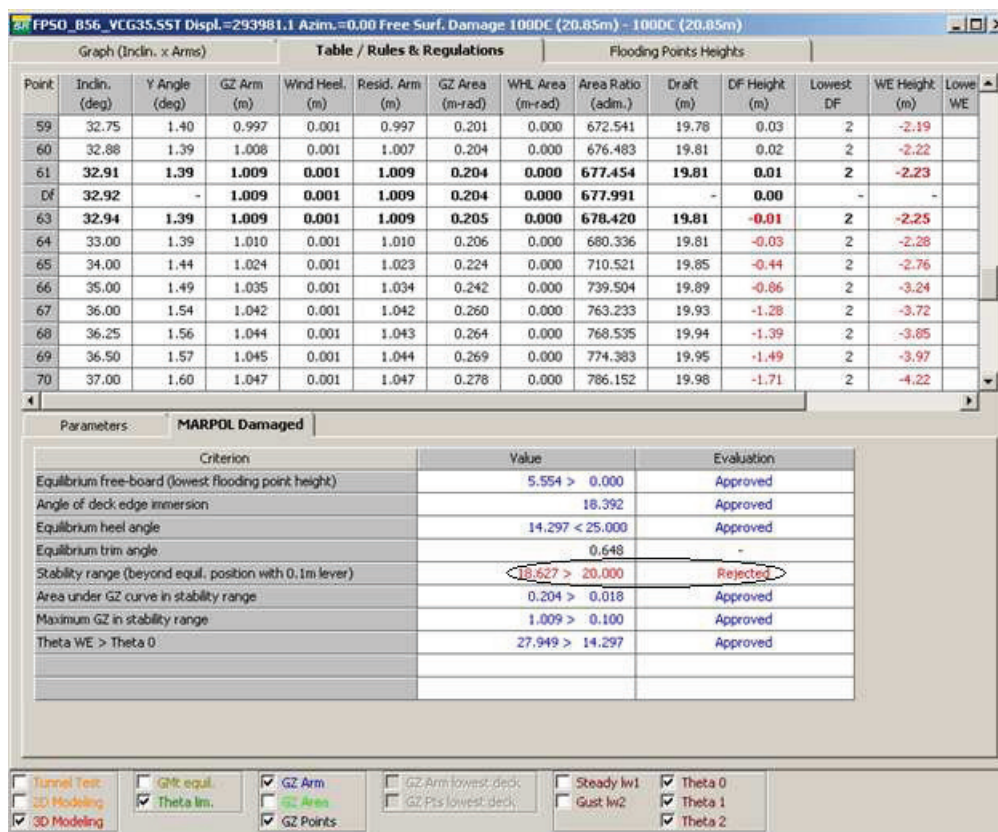


Figura 4.7: Tabela do diagrama de estabilidade do navio FPSO cuja boca é 56 metros.

4.2.2

Variação Automática da Forma do FPSO

Neste exemplo, entende-se que para que o sistema atenda a todos os critérios de avaria impostas pela MARPOL 73/78, a geometria da estrutura deve ser alterada de forma automática, variando-se para isto o peso leve da estrutura e o calado de equilíbrio.

Um parâmetro que pode ser considerado para a definição da nova forma do casco é a boca do FPSO. Cabe ressaltar, no entanto, que o fato de aumentar a boca da FPSO não garante que a nova forma da estrutura resulte em um ganho significativo da faixa de estabilidade, pois este está diretamente relacionado com o GM (distância entre o centro de massa e o metacentro) inicial na sua condição intacta em equilíbrio.

Salienta-se que a modelagem abordada nesse trabalho permite total independência entre as transformações possíveis de serem aplicadas no sistema flutuante. Existe, entretanto, uma correlação geométrica entre os diversos tanques internos (lastro, petróleo, óleo diesel, espaços vazios, paiol de amarras, etc) e a forma do casco do FPSO, que não pode ser alterada de forma

independente. Tal correlação é criteriosamente atendida pela programação utilizada, que faz as mesmas transformações com as classes de compartimentos que estão relacionadas.

A seguir apresenta-se o algoritmo 5, implementado no MG, que faz as devidas alterações tanto no casco do FPSO quanto nos seus respectivos tanques internos.

```

algoritmo TransformFPSO_Geometry()
1  currentdir="SMOD_B56_without_sst/"
2  // defines breath value to 58
3  B=58
4  path = "D:/mg/data/tese/"..currentdir
5  // defines auxiliar function to perform transformation
6  function TransformBREADTHAux(factor,Bmx,Bmy,Bmz)
7      mgSelect("all")
8      mgTranslateXYZ(Bmx,Bmy,Bmz)
9      mgScaleXYZ(1,factor,1)
10     mgTranslateXYZ(-Bmx,-Bmy,-Bmz)
11  end
12 // defines current directory work for hull volumes
13 directory="D:/mg/data/tese/"..currentdir.."Hull_Volumes/"
14 mgLoad(directory.."HULL1.mg");
15 factor,Bmx,Bmy,Bmz=TransformBreadth(25,90,B)
16 mgSave(directory.."HULL1.mg")
17 mgExportSstab()
18 // for all directories performs transformation
19 directory="D:/mg/data/tese/"..currentdir.."Ballast_Tanks/"
20 dofile(path.."script_ballast_tank.lua")
21 directory="D:/mg/data/tese/"..currentdir.."Cargo_Tanks/"
22 dofile(path.."script_cargo_tank.lua")
23 ...
24 directory="D:/mg/data/tese/"..currentdir.."Void_Spaces/"
25 dofile(path.."script_void_spaces.lua")
26 end
fim

```

Algoritmo 5: *Script* genérico para transformar um navio FPSO.

Este algoritmo define uma função auxiliar chamada de *TransformBREADTHAux* que recebe como parâmetros um fator de escala e um ponto de referência, obtidos pela função *TransformBreadth*, previamente definida no MG.

A função *TransformFPSO_Geometry* percorre, de forma automática, todos os diretórios onde encontram-se os tanques internos da embarcação, e aplica a função *TransformBREADTHAux* para mudar a sua forma.

A leitura e armazenamento dos modelos geométricos são feitos utilizando os comandos `mgLoad` e `mgSave`, respectivamente. A exportação dos dados para o programa de análise de estabilidade estática (`Sstab`), é feita utilizando o comando `mgExportSstab()`.

Salienta-se que na exportação dos dados geométricos para o `Sstab` realiza-se uma verificação da consistência topológica das malhas geradas e detecta-se se existe simetria no modelo. Caso haja simetria somente é armazenada a parte simétrica e o restante do modelo é salvo mediante transformações de espelhamento. Isto é feito com o intuito de manter a malha sempre simétrica para evitar o surgimento de trim ou banda ao se fazer as análises estáticas do modelo.

Devido à escolha da linguagem Lua para estender o MG, por meio de uma interface de linha de comandos, foi incorporado também no ambiente `Sstab` a linguagem Lua, de tal forma a poder utilizar *scripts* criando assim um conjunto de comandos específicos para análise estática.

Estes comandos têm o seguinte padrão de sintaxe: iniciam-se com as letras `sst` seguidos da ação que executarão (Ex: `ssSetMarpolCondition`, `sstSetIntactCondition`, `sstGetLightWeight`,...,etc.). A combinação destes comandos de análise estática com os comandos da linguagem Lua permite ao usuário criar *scripts* para por exemplo realizar por exemplo uma estimativa de pesos de forma automatizada.

4.2.3

Estimativa de Pesos do Navio FPSO

Uma vez obtidos os modelos geométricos necessários para as devidas simulações no `Sstab`, decidiu-se manter constante a relação de volume deslocado por volume total.

Adota-se a hipótese, bastante realista, que o peso leve aumenta de forma proporcional à variação de volume deslocado. Para tanto, consideramos que a proporção de volume submerso do casco neste exemplo se mantém constante (63% no caso abordado), tomando como referência a condição de equilíbrio intacto.

Esta proporção de volume deslocado é controlada por uma tolerância passada como parâmetro para função *FindLightWeight*, assim como o fator de volume deslocado a ser considerado. Desta forma, uma procura do peso leve é feita, adicionando-se ou retirando-se peso até se alcançar a tolerância imposta, que define o peso leve correto da embarcação, onde o volume deslocado abaixo do plano de água deve ser proporcional a 63% do volume total. A seguir apresenta-se o algoritmo 6 na próxima página que descreve a função

FindLightWeight implementado no Sstab.

```

algoritmo function FindLightWeight(factor,toler)
  1 // set model into an intact condition
  2 sstSetIntactCondition()
  3   while true do
  4     // get current weight
  5     weight=sstGetLightWeight()
  6     // compute equilibrium
  7     sstComputeEquil()
  8     // get total volume and volume displaced
  9     totalVol=sstGetHullVol()
 10     floodedVol=sstGetHullVolDisplaced()
 11     percentVal=floodedVol/totalVol
 12     // check user tolerance
 13     if (percentVal-factor <= toler) then
 14       break
 15     end
 16     if (percentVal > factor) then
 17       weight=weight+(weight*0.01)
 18     end
 19     if (percentVal < factor) then
 20       weight=weight-(weight*0.01)
 21     end
 22   end
 23 end
fim

```

Algoritmo 6: Automatização da busca do peso leve.

```

algoritmo function ComputeGammaAngles(DC)
  1 // enable damage condition DC
  2 sstSetDamageCondition(DC)
  3 // use MARPOL criterias
  4 sstSetMarpolCriteria()
  5 sstComputeEquil()
  6 // compute stability diagram
  7 sstComputeStabilityDiagram()
  8 val=sstGetGammaRange()
  9 // check range value
 10 if (val > 20) then return 1
 11 else return 0 end
 12 end
fim

```

Algoritmo 7: Calculo de faixa de estabilidade por *scripts*.

Por outro lado, a função *ComputeGammaAngles*, mostrada no algoritmo 7 da pagina anterior, verifica se uma determinada condição de avaria, passada como parâmetro, respeita as restrições impostas pela MARPOL 73/78. Isto é, calcula-se o diagrama de estabilidade para encontrar a faixa mínima de ângulos possíveis além da sua posição de equilíbrio.

A seguir mostra-se a função *CheckModels*, implementada no algoritmo 8. Esta função considera uma variação da boca do navio FPSO variando entre 50 e 70 metros. Além disso, considera o VCG para todos os modelos lidos em torno de 33 metros, tanto na condição intacta quanto na condição avariada.

Para cada boca considerada é carregado um arquivo, que tem armazenadas as condições intacta e avariada e é calculado o volume deslocado na sua condição intacta em torno de 63% (função *FindLightWeight*). A seguir é calculado o diagrama de estabilidade do modelo (função *CheckGammaAngles*) e extrai-se o valor da variação da faixa de ângulos de estabilidade para a condição avariada. Pode-se perceber que este script alterna entre a condição intacta e avariada.

```

algoritmo function CheckModels(VCG,BMin,BMax)
1  currentdir="D:/mg/data/tese/FPSOwith_sst/"
2  // VCG value is set equal to 33
3  // BMin and BMax refer a range of Breadth values
4  for Boca=BMin,BMax,1 do
5    if (sstLoad(currentdir.."FPSO_"..Boca.."mg")) then
6      SetVCG(VCG)
7      FindLightWeight()
8      ret=CheckGammaAngles()
9      if (ret) then print("gamma range achieved")
10     else print("gamma range not achieved") end
11   else
12     print("file not found")
13     continue
14   end
15 end
16 end

fim

```

Algoritmo 8: *Script* genérico do cálculo de estabilidade de um conjunto de FPSOs.

4.2.4

Resultados Obtidos

Foram avaliados diferentes modelos gerados de forma automática por meio de *scripts*. Foi verificado que as alterações de geometria pouco influenciaram o coeficiente de bloco, que ficou próximo de 0,70 metros. Desta forma, pode-se dizer que a característica de flutuação projetada para o FPSO mantém-se intacta.

As propriedades de estabilidade estática, obtidas com o cálculo do equilíbrio dos modelos conforme as variações da boca e dos pesos, são apresentadas na tabela 4.1.

A condição inicial do FPSO é indicada por uma linha horizontal cuja borda está ressaltada na tabela 4.1. Nos testes considerados percebe-se que três casos convergem na faixa de estabilidade mínima para a condição avariada DC. Estes casos estão indicados em negrito na tabela 4.1.

A variável B faz referência ao valor corrente da boca do navio. O Volume Parcial mostrado na tabela é o volume total que está abaixo do plano da água, com relação ao peso corrente para cada valor de B. A coluna Desloc. indica o peso deslocado do navio.

Os dados das colunas 1 até a 7, da tabela 4.1, foram obtidos considerando o modelo na sua condição intacta. As colunas 8 até a 10 são os resultados obtidos considerando a avaria corrente DC.

Considerando os modelos que foram aprovados pelas restrições da MARPOL 73/78 na condição avariada, caso deseje-se que a FPSO suporte o carregamento corrente, a geometria adequada a ser adotada seria o modelo cuja boca é de 58,5 metros, pois este tem o menor custo.

A seguir mostram-se as figuras 4.8, 4.9 e 4.10 que satisfazem a faixa de estabilidade imposta pela MARPOL 73/78 para o caso onde a FPSO possui uma boca em torno de 58,5 metros.

Após o cálculo do equilíbrio do navio FPSO, verifica-se um ângulo de 12,13 graus de banda para bombordo e um baixo trim de proa, com borda livre de bombordo de apenas 2,41 metros. Da mesma forma os resultados e as informações de equilíbrio são mostradas na tabela *Summary* da figura 4.8.

As figuras 4.9 e 4.10 mostram, respectivamente, o diagrama e a tabela de estabilidade para o navio FPSO cuja boca é de 58,5 metros. Nesta configuração todos os critérios impostos pela MARPOL 73/78 passam a ser atendidos satisfatoriamente.

B	Calado	Desloc.	Vol. Total	Vol. Parcial	Vol. %	Peso	KG	GM	Faixa
50	20,45	272501,4	420870,8	265836	0,63	25000	15,75	5,20	20,26
52	20,48	284098,8	437712,1	277152	0,63	35000	16,01	5,79	19,04
53	20,51	290179,0	446125,0	283084	0,63	41000	16,17	6,07	18,71
54	20,48	295194,3	454540,0	287977	0,63	46000	16,26	6,42	18,80
56	20,52	306998,8	471375,4	299499	0,63	57806	16,56	7,02	18,62
57	20,44	311197,3	479798,9	303591	0,63	62000	16,62	7,44	19,09
58	20,37	315391,4	488215,5	307684	0,63	66194	16,69	7,86	19,54
58,2	20,37	316715,3	489893,6	308976	0,63	67518	16,72	7,92	19,49
58,5	20,35	318701,2	492418,8	310913	0,63	69503	16,78	8,00	22,46
59	20,38	321197,6	496631,4	313349	0,63	72000	16,84	8,19	19,53
60	20,39	327003,8	505044,1	319014	0,63	77806	16,98	8,54	19,52
62	20,42	338616,3	521879,8	330343	0,63	89418	17,30	9,24	19,48
70	20,39	382199,5	589219,1	372867	0,63	133000	18,30	12,6	19,81
72	20,41	393831,4	606054,0	384215	0,63	144632	18,60	13,49	34,10

Tabela 4.1: Resultados da variação da boca da FPSO.

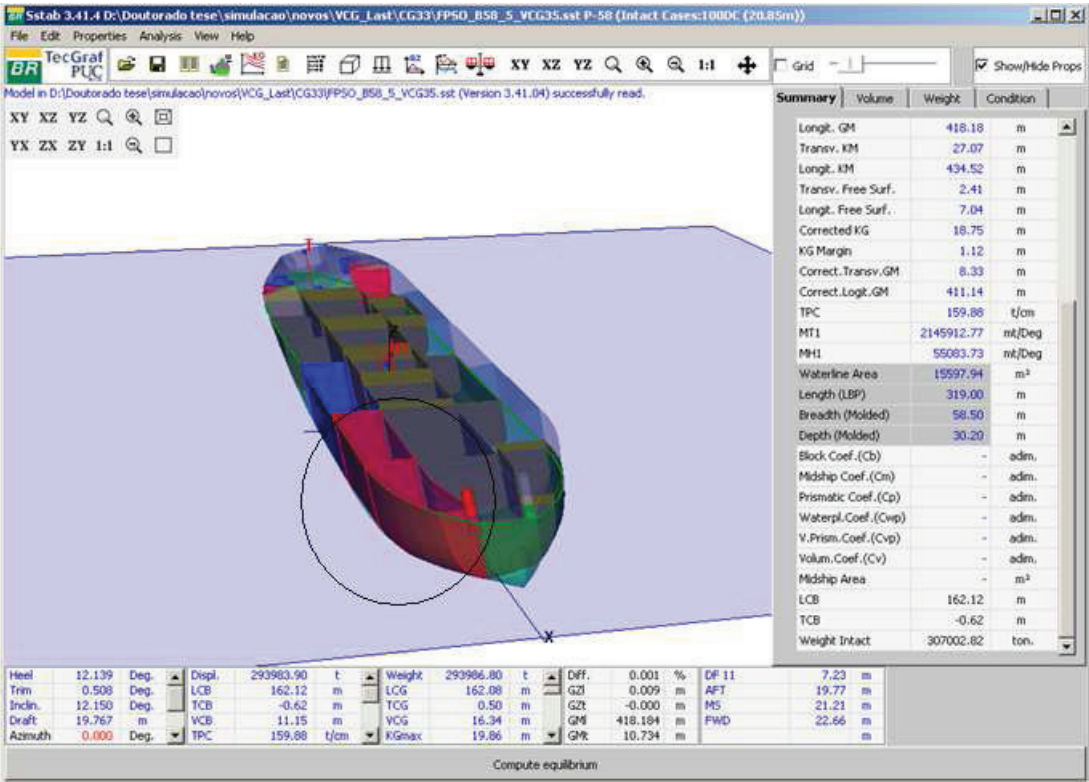


Figura 4.8: Condição avariada do navio FPSO com boca de 58.5 metros.

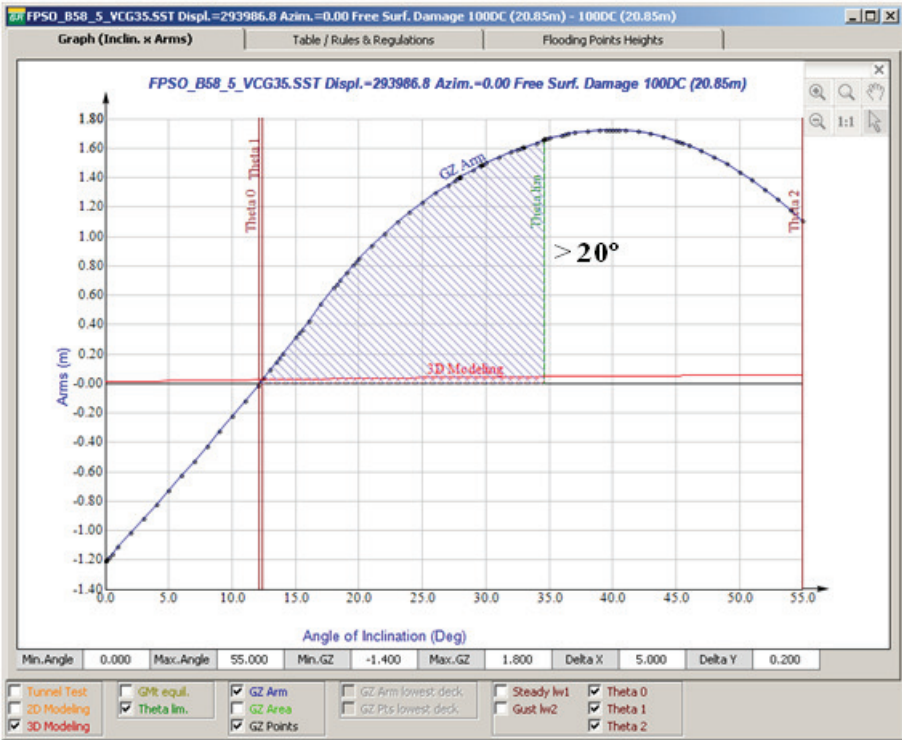


Figura 4.9: Diagrama da estabilidade da FPSO com boca de 58.5 metros.

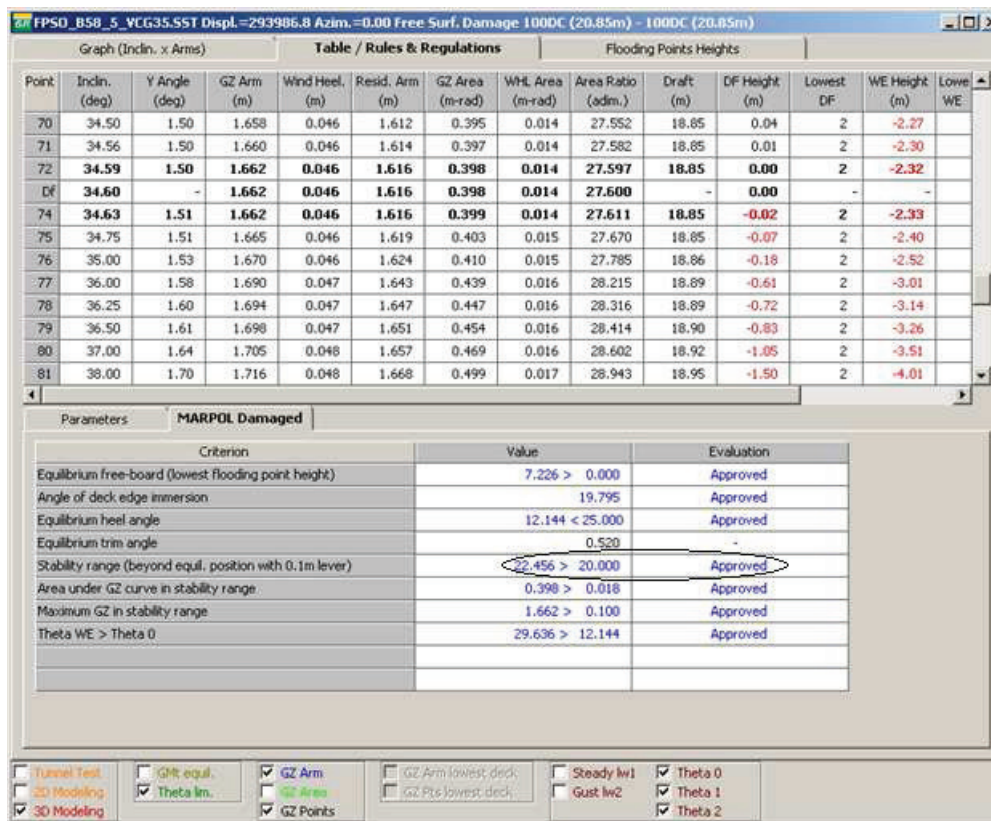


Figura 4.10: Tabela do diagrama de estabilidade do navio FPSO com boca de 58.5 metros.

O critério de faixa de estabilidade neste caso está correto e possui ainda uma folga 2,4 graus, a qual é ressaltado por uma elipse (figura 4.10).

Os *scripts* implementados para procurar a geometria ótima do FPSO, que suporte a avaria em questão, podem considerar ou não todas as restrições impostas pela MARPOL 73/78 (faixa de ângulos mínima, área do braço de endireitamento, etc). No presente exemplo, somente foi considerada a restrição de faixa mínima de ângulos, pois foi a única que não foi satisfeita com a avaria imposta quando a boca do navio é 56 metros.

A figura 4.11 mostra a variação da faixa de estabilidade alcançada pelas condições de avaria com relação à variação da boca da embarcação.

Pode-se verificar que a curva gerada não segue uma tendência linear. Isto é, aumentar ou diminuir a boca do modelo não significa que a faixa mínima de ângulos terá uma variação proporcional à boca. Um aumento na boca do navio pode provocar tanto um ganho quanto uma perda na faixa de ângulos de equilíbrio, após atingida a estabilidade na condição avariada. Trata-se de uma procura pelo mínimo local que represente o menor custo e atenda a todos os requisitos exigidos pelas sociedades classificadoras (MARPOL 73/78).

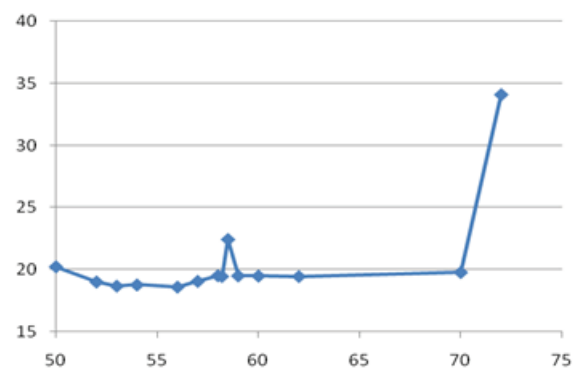


Figura 4.11: Gráfico da variação da faixa de estabilidade com relação à boca da FPSO.