

3

Definição automática de carregamento ótimo

A formulação inicial mostrada neste capítulo foi feita por Sérgio Álvares Maffra[11] e parte da implementação foi feita por Anderson Pereira, tendo sofrido algumas modificações ou correções para serem utilizadas no Sstab. As funcionalidades de minimizar superfície livre, segregar por tipo de líquido e minimizar a curva de força cortante também foram adicionadas. Mudanças foram feitas para permitir o uso em casos de avaria e alagamento.

Uma embarcação está em equilíbrio quando as forças que agem sobre ela e os momentos gerados por estas estão em equilíbrio. Portanto, empuxo, peso e forças externas, assim como os momentos gerados por estas devem estar em equilíbrio. Ventos e correntes marítimas são exemplos de forças externas.

O carregamento de uma unidade flutuante neste contexto é a carga de líquidos que pode ser armazenada em seus compartimentos. Neste capítulo, explica-se a utilização da programação linear [2] para realizar carregamento automático de uma unidade flutuante e como este tipo de programação resolve um problema não linear. De modo geral, o algoritmo é um procedimento iterativo que, a cada passo, tenta resolver um problema não linear usando a solução dada por uma aproximação linear, até que convirja para uma solução dentro da tolerância. Especificamente, nos casos tratados neste trabalho, a não linearidade está no fato que as possíveis coordenadas do centro de gravidade (LCG e TCG) variam durante o processo de carregamento e descarregamento da plataforma. Torna-se necessário fazer uma aproximação linear ao considerar cada tanque uma carga pontual e assim o equilíbrio da unidade pode ser alcançado, calculando a variação de volume de cada tanque. Após obter o volume de cada tanque, computam-se as verdadeiras coordenadas do centro de gravidade de cada tanque, a partir de sua geometria. Convergência é obtida quando não há mais uma variação significativa entre as coordenadas da carga pontual consideradas e a verdadeiras coordenadas do centro de gravidade de cada tanque. Usa-se a minimização da variação de

líquido em cada tanque como a função objetivo e assegura-se um deslocamento mínimo de líquido.

O algoritmo proposto ainda possui uma heurística alternativa para minimizar a tensão e esforço cortante ao longo da estrutura. Partindo do princípio que a distribuição dos tanques tenha um formato similar ao casco, faz-se uma distribuição homogênea de líquido ao longo dos tanques de carga, de óleo e de lastro.

Ainda existe uma segunda heurística para minimizar a área de superfície líquida dentro dos tanques, que consiste em tentar concentrar a maior quantidade de líquido no menor número de tanques possíveis.

Estes algoritmos foram implementados no programa Sstab e provaram sua eficiência no desenvolvimento de projetos de navios e plataformas de petróleo (Bacia de Campos, Rio de Janeiro, Brasil) e no desenvolvimento de planos de emergência realizados pelas classificadoras DNV, ABS.

3.1.

Programação Linear

O algoritmo proposto utiliza programação linear para computar a configuração de equilíbrio da embarcação. De modo geral, a programação linear tem dois componentes: um conjunto de restrições lineares dadas como inequações e uma função que deve ser maximizada ou minimizada. Um exemplo de programação linear é dado pelas equações 3.1 a 3.6.

$$\text{Minimizar / Maximizar} \quad c^t x \quad (3.1)$$

$$\text{Sujeito a:} \quad a_i^t x \geq b_i \quad i \in M_1 \quad (3.2)$$

$$a_i^t x \leq b_i \quad i \in M_2 \quad (3.3)$$

$$a_i^t x = b_i \quad i \in M_3 \quad (3.4)$$

$$x_j \geq 0 \quad j \in N_1 \quad (3.5)$$

$$x_j \leq 0 \quad j \in N_2 \quad (3.6)$$

Nestas equações, as letras maiúsculas representam vetores e as letras minúsculas representam escalares. X representa o vetor que contém todas as variáveis que devem ser determinadas pelo programa e x representa uma única variável. C^t representa o vetor de pesos que cada uma das variáveis de X pode ter

(equação 3.1). Restrições de igualdade e desigualdade são aceitas. Os três primeiros conjuntos de restrições ($M1$, $M2$ e $M3$) contém o vetor A e o escalar b , que representam as informações de valores associados às restrições.

Uma solução possível deve maximizar ou minimizar a função objetivo (3.1) e respeitar todas estas restrições do problema estabelecidas (3.2 a 3.6). Para se obter a solução de nosso problema linear, foi utilizada a biblioteca GLPK¹.

3.2.

Programação linear aplicada ao carregamento de uma embarcação

Das forças descritas acima, apenas o peso resultante do carregamento de líquido nos tanques da embarcação pode ser controlado. Logo, as variáveis utilizadas para realizar o carregamento automático de líquidos devem ser a quantidade de líquido em cada tanque, mais especificamente, a variação de volume dentro de cada tanque. Dadas as variáveis do problema, as restrições de equilíbrio e limites são dadas são descritas pelas equações 3.7 a 3.12.

$$\sum (V_i + \Delta V_i) \rho_i g = B - W \quad i \in T \quad (3.7)$$

$$\sum (V_i + \Delta V_i) \rho_i g X_i = -M_L \quad i \in T \quad (3.8)$$

$$\sum (V_i + \Delta V_i) \rho_i g Y_i = -M_T \quad i \in T \quad (3.9)$$

$$V_i + \Delta V_i < V_i^{\max} \quad i \in T \quad (3.10)$$

$$V_i + \Delta V_i > V_i^{\min} \quad i \in T \quad (3.11)$$

$$\sum (V_c + \Delta V_c) = V_k^{\text{total}} \quad c \in T_k \quad (3.12)$$

V_i Volume inicial do tanque i

ΔV_i Variação do volume do tanque i

$V_i^{\max}$ Capacidade máxima do tanque i

$V_i^{\min}$ Capacidade mínima do tanque i

ρ_i Densidade do fluido dentro do tanque i

g aceleração da gravidade

X_i, Y_i Coordenadas do centro de gravidade do fluido no tanque i

B Força de empuxo

W Peso total da embarcação

M_T, M_L Momento total resultante de forças externas

1-Biblioteca desenvolvida para o projeto GNU, por Andrew Makhorin, do Departamento de Informática Aplicada do Instituto de Aviação de Moscou, Moscou, Rússia.

T	Conjunto de tanques da embarcação
T_k	Conjunto de tanques contendo o mesmo tipo de líquido (k)
V_k^{total}	Volume total de líquido nos tanques pertencentes a T_k

A primeira restrição (3.7) garante o equilíbrio de forças. Para estar em equilíbrio, o peso de líquido nos tanques somado ao peso da plataforma deve ser igual à força de empuxo. A segunda e a terceira restrições (3.8 e 3.9) asseguram o equilíbrio de momentos entre forças externas e internas. Estas três restrições são responsáveis por manter a embarcação em equilíbrio.

As restrições 3.10 e 3.11 são relacionadas às variáveis do problema. A primeira garante que o volume do tanque será no máximo igual a sua capacidade máxima e a segunda garante que o tanque nunca ficará abaixo de seu volume mínimo e impede que tenha volumes negativos.

Não é incomum para uma embarcação conter diferentes tipos de tanque (água potável, óleo, água do mar, combustível, etc). Neste caso, é importante que o algoritmo não misture os volumes de cada tipo. Isto se garante ao considerar a equação 3.12. Estas restrições garantem que os tanques pertencentes ao conjunto T_k possuem um determinado volume de líquido.

Criando um conjunto T_k para cada tipo líquido, garante-se que volume total de cada tipo permanecerá constante. Esta restrição não é necessária para tanques de lastro, já que estes utilizam água do mar e podem se livrar e conseguir mais desta. Já tanques de óleo, o volume total dentro destes deve permanecer constante.

Ao se tentar obter um carregamento automático de uma embarcação, deve-se decidir para quais grupos de tanque a sexta restrição (3.12) deve ou não ser usada. Geralmente, esta restrição deve ser utilizada com tanques de óleo, água potável e combustível.

3.2.1. Forças Externas e Peso

As equações de restrição mostradas anteriormente assumem a existência de forças externas e momentos resultantes destas em seu algoritmo. Na implementação deste algoritmo, estes valores são computados diretamente de um modelo geométrico da embarcação. Este modelo é construído como um conjunto de compartimentos individuais que podem ser classificados como externos (casco

e elementos de convés) ou internos (espaços vazios, tanques, elevadores, corredores, etc.). Cada elemento composto por um conjunto de faces, que por sua vez são compostas por um conjunto de pontos com coordenadas tridimensionais.

O empuxo é determinado ao computar as partes submersas de cada compartimento externo. Usando o algoritmo de interseção de malha implementado na biblioteca do MG [4], os compartimentos externos são divididos em duas partes, uma acima e outra abaixo do plano de superfície do oceano. O volume de cada parte submersa e seu centro de gravidade é utilizado para calcular a força e o centro do empuxo.

O mesmo procedimento é utilizado para calcular o centro de gravidade de cada tanque. Usando um procedimento similar a busca binária [5], computa-se a posição da superfície livre deste de cada tanque a partir de seu volume e da posição atuais. Após dividir o tanque em dois volumes (superfícies secas e molhadas), de modo similar ao que foi feito com o casco, o centro de gravidade pode ser determinado, utilizando a superfícies molhadas para determinar seu centro geométrico.

3.2.2. Função Objetivo

O último componente a ser discutido é a função objetivo do problema proposto. Como dito anteriormente, a função objetivo é usada como forma de se obter a melhor solução entre aquelas possíveis. Para este caso, escolhe-se como função objetivo a que minimiza a soma da variação total de líquidos em todos os tanques, conforme mostrado na equação 3.13.

Uma segunda possibilidade seria selecionar o meio mais rápido de ajuste dos volumes, ou seja, o meio mais rápido de transferir todo o líquido necessário. Para tanto, seria necessário saber a capacidade de bombeamento para cada tanque. A função objetivo seria a 3.14.

$$\text{Minimizar} \quad \sum |\Delta V_i| \quad i \in T \quad (3.13)$$

$$\text{Minimizar} \quad \sum |P_i \Delta V_i| \quad i \in T \quad (3.14)$$

P_i Capacidade de bombeamento para cada tanque

3.3. Programação linear seqüencial

Observa-se que as coordenadas X e Y do centro de gravidade dos tanques não são constantes, variando não só com a posição e orientação do tanque. Na verdade, estas coordenadas são funções também do volume contido no tanque. O algoritmo aqui desenvolvido considera estas duas coordenadas (x_i, y_i) constantes, caso contrário não haveria uma solução linear. Para considerar essa variação do centro de gravidade, o algoritmo deve ser implementado como uma programação linear seqüencial.

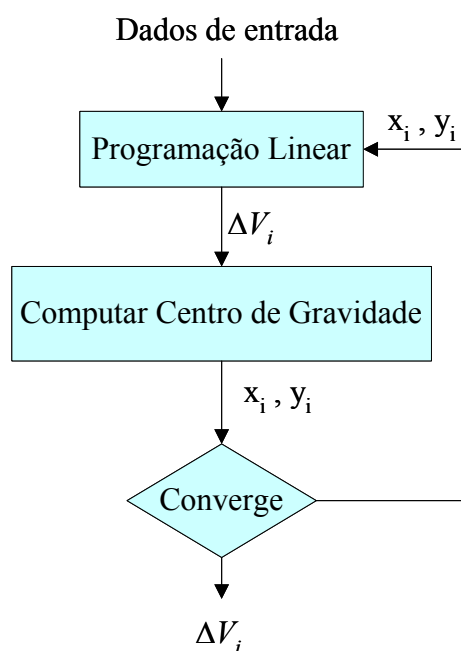


Figura 3.1- Diagrama de fluxo do algoritmo de carregamento automático.

A seqüência deste algoritmo é ilustrada na Figura 3.1. A programação Linear é utilizada para estimar a variação de volume (ΔV_i). Esta variação é então utilizada para calcular o centro de gravidade de cada tanque (x_i, y_i). Finalmente, o novo centro de gravidade é comparado com o anterior. Caso haja um tanque em que a distância entre o antigo e o novo centro de gravidade seja maior que a tolerância, o algoritmo recomeça o processo. Caso contrário, o algoritmo pára e a variação de volume de cada tanque é dada como resultado. O algoritmo também é interrompido caso não haja convergência após um número máximo de iterações.

3.4. Melhorias no Resultado

O equilíbrio de um sistema flutuante é um requisito essencial para um algoritmo de carregamento automático. Em alguns casos, no entanto, é necessário levar outros fatores em conta. Dois requisitos extras foram adicionados ao algoritmo: minimização da tensão e esforço cortante ao longo da estrutura e minimização da superfície líquida dentro dos tanques. A necessidade de minimizar a superfície livre de uma plataforma se deve ao aumento de estabilidade, quando se diminui a superfície livre.

3.4.1. Minimizar tensão e esforço cortante

Testes realizados com o algoritmo de carregamento automático aplicado a modelos de navios mostraram que alguns resultados fornecidos por tal algoritmo resultavam em curvas de momento e esforço cortante acima dos limites admissíveis para grandes navios. Uma curva irregular também foi observada para o esforço cortante. Este fato é decorrente de uma distribuição irregular de peso pelo navio, causada por uma distribuição irregular de líquido pelo navio.

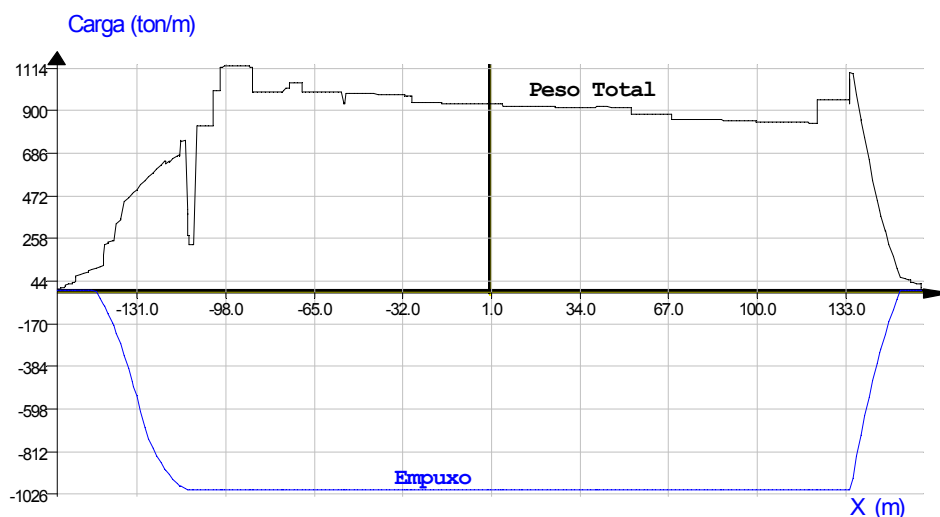


Figura 3.2- Curva de Empuxo para um navio.

Com o objetivo de reduzir este efeito, duas novas restrições foram incorporadas ao algoritmo. Estas duas novas restrições são responsáveis pelo controle do volume dentro dos tanques e tentam gerar uma distribuição homogênea de líquidos. Isto, na maioria dos casos, resultará numa redução da

tensão. A Figura 3.2 mostra duas curvas, uma de empuxo e outra de peso total para um navio. Em teoria, se o navio é carregado e a curva deste carregamento equivale à curva de empuxo, as curvas de cisalhamento (diagrama de esforço cortante) e momento (diagrama de momento fletor) se anulariam. Logo, o ideal é que a curva de peso total da embarcação fosse o mais próximo possível do inverso da curva de empuxo, como é demonstrado na Figura 3.2.

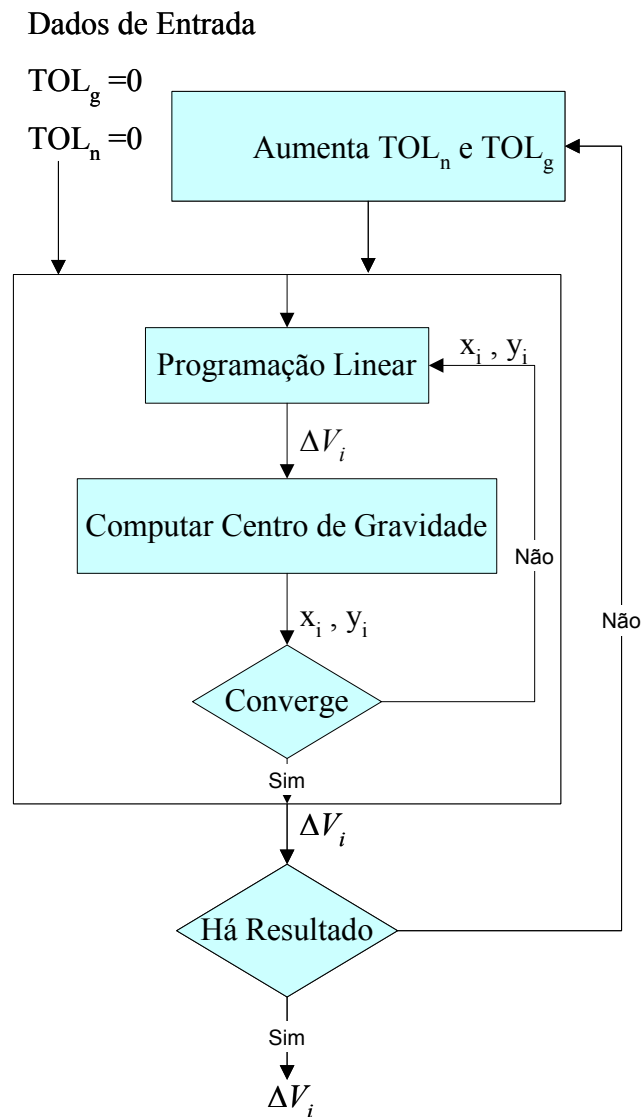


Figura 3.3- Determinação de TOL_g e TOL_n .

As novas restrições podem ser vistas nas equações 3.15 e 3.16. Dois novos parâmetros foram adicionados: tolerância global usada por todos os tanques (TOL_g) e uma outra tolerância usada apenas entre tanques vizinhos (TOL_n). Como as equações 3.15 e 3.16 mostram, a quantidade de líquido num tanque será limitado pelo volume em seus vizinhos.

$$\left| \frac{v_i + \Delta v_i}{V_i^{\max}} - \frac{v_k + \Delta v_k}{V_k^{\max}} \right| < TOL_g \quad \begin{matrix} i \in T, k \in T \\ i \neq k \end{matrix} \quad (3.15)$$

$$\left| \frac{v_i + \Delta v_i}{V_i^{\max}} - \frac{v_j + \Delta v_j}{V_j^{\max}} \right| < TOL_n \quad i \in T, j \in N_i \quad (3.16)$$

TOL_g Tolerância global de variação de volume

TOL_n Tolerância entre vizinhos para variação de volume

N_i Conjunto de tanques k vizinhos ao tanque i

O valor ideal para TOL_g e TOL_n é zero, o que corresponde à distribuição ideal de líquido. Mas este valor não é aceitável para a maioria dos casos. A determinação de TOL_g e TOL_n é computado utilizando o procedimento iterativo mostrado na Figura 3.3. O algoritmo parte de tolerâncias TOL_g e TOL_n baixas e realizam os procedimentos de programação linear descritos anteriormente, mas adicionam-se as restrições 3.15 e 3.16 à lista de restrições. Ao final, se não há resultados, aumentam-se as restrições e reinicia-se o processo.

3.4.2. Minimizar superfície livre

O efeito de superfície livre ocorre somente quando o tanque está parcialmente cheio. Tanques vazios e tanques completamente cheios não apresentam superfície livre. Em se tratando de plataformas ou auto-elevatórias, há um interesse em se obter uma resposta em que a área total de superfície livre dos tanques seja mínima. Isto porque o efeito de superfície livre faz com que o centro de gravidade da plataforma varie quando a embarcação oscila por algum motivo.

O objetivo então é encontrar uma solução para o conjunto de tanques, em que o maior número possível de tanques estejam vazios ou completamente cheios. Deve-se obter uma função objetivo que maximize o número de tanques vazios e conseqüentemente também acumule o maior volume possível de líquido no menor número possível de tanques, obtendo um maior número de tanques completamente cheios. A função 3.17 então substituirá a função 3.13.

$$\text{Minimizar} \quad \sum V_i + \Delta V_i \quad i \in T \quad (3.17)$$

3.5. Resultados

Foram selecionados três exemplos para mostrar as possibilidades do algoritmo de carregamento automático.

O primeiro exemplo mostra uma plataforma com capacidade para 54.000 toneladas de água em seus tanques de lastro e operando numa cota de 34,00 metros. O total de peso leve neste exemplo é de 26.200 toneladas e o Centro de gravidade deste peso está em -2,5m longitudinais e 6,8m bombordo (coordenadas locais).

Após selecionar todos os 44 tanques de lastro para balancear o modelo na cota de operação desejada, a Figura 3.4 mostra o carregamento dos tanques definido pelo algoritmo. O líquido dentro dos tanques é mostrado em azul fosco, enquanto os volumes vazios são mostrados transparentes. A Figura 3.4(a) mostra o resultado para o algoritmo com minimização de superfície líquida e a Figura 3.4(b) para o algoritmo sem a minimização. Nota-se que a distribuição de líquido para o primeiro é concentrada no menor número de tanques possíveis. Já o segundo há uma área grande de superfície livre.

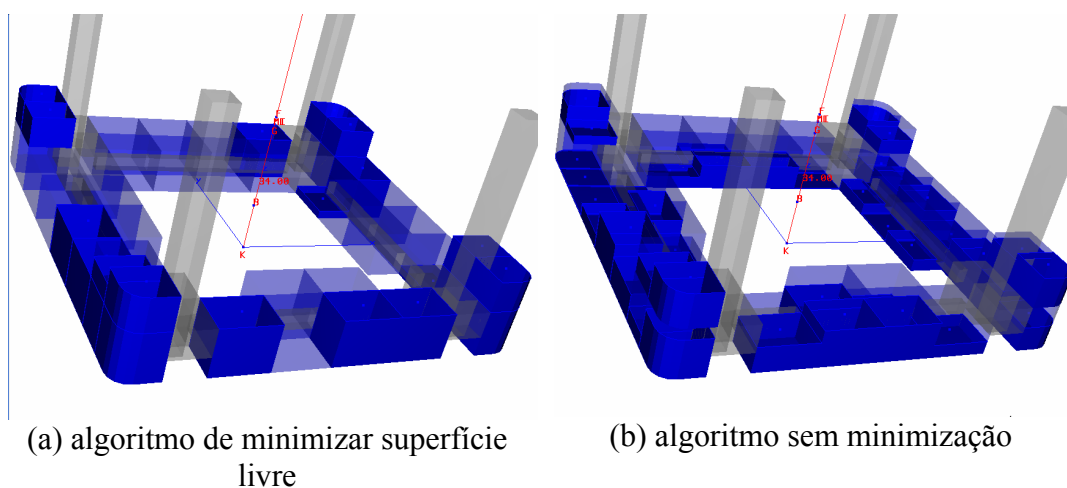


Figura 3.4- Carregamento de tanques após utilização do algoritmo de carregamento

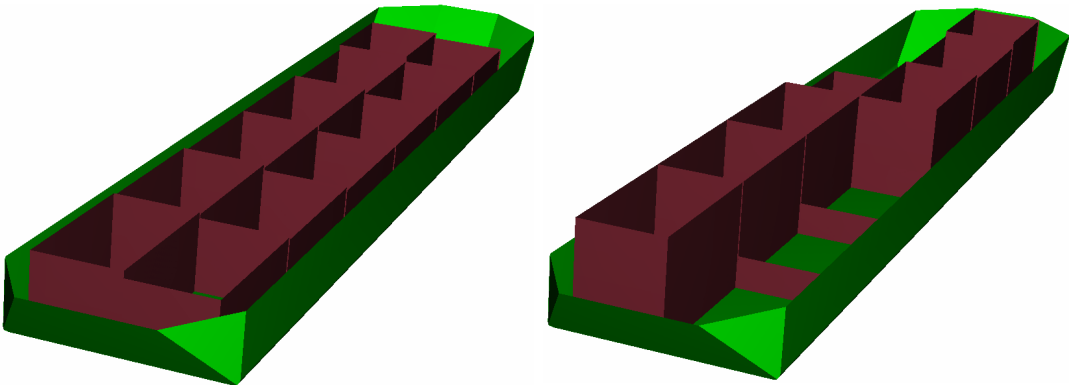
A Tabela 3.1 mostra esses resultados em números, comparando ambos os algoritmos. Observe como a superfície livre para o algoritmo com minimização de superfície livre ficou pequena e concentrada em apenas 3 tanques. Isto faz com que o KG varie muito pouco quando a plataforma oscila. Já para o algoritmo sem

minimização, essa superfície ficou imensa e espalhada ao longo de 42 tanques. Isto provoca uma grande variação de KG, quando a plataforma oscila.

	Algoritmo (a) Min. Sup. livre	Algoritmo (b) Sem Min.	Diferença (a) – (b)
Tanques vazios	19	0	19
Tanques completamente cheios	20	2	18
Tanques parcialmente cheios	3	42	-39
Superfície livre (m ²)	388	5434	-5046

Tabela 3.1-Carregamento de tanques para ambos os algoritmos

O segundo exemplo consiste em usar o algoritmo para minimizar a tensão ao longo do eixo longitudinal. O objetivo é manter o peso total igual, mas com uma melhor distribuição da carga para diminuir as forças de cisalhamento e o momento ao longo do eixo longitudinal. A Figura 3.5 (a) mostra o carregamento final dos tanques para o segundo exemplo. A malha em verde mostra o casco externo, enquanto o marrom mostra preenchimento dos tanques de óleo.



(a) algoritmo de cisalhamento mínimo

(b) algoritmo sem cisalhamento mínimo

Figura 3.5 – Navio carregado

A Figura 3.5 (b) mostra o mesmo caso, mas não foi utilizado o algoritmo de cisalhamento mínimo. Nota-se que em 3.5(a) há uma melhor distribuição de óleo do que em 3.5(b). Ambas as Figuras estão na mesma posição de equilíbrio.

As Figuras 3.6 e 3.7 mostram os gráficos de força de cisalhamento e momento respectivamente, comparando o resultado para os dois algoritmos. Note que o algoritmo de cisalhamento mínimo dá resultados sempre mais próximos de zero que um algoritmo sem a minimização.

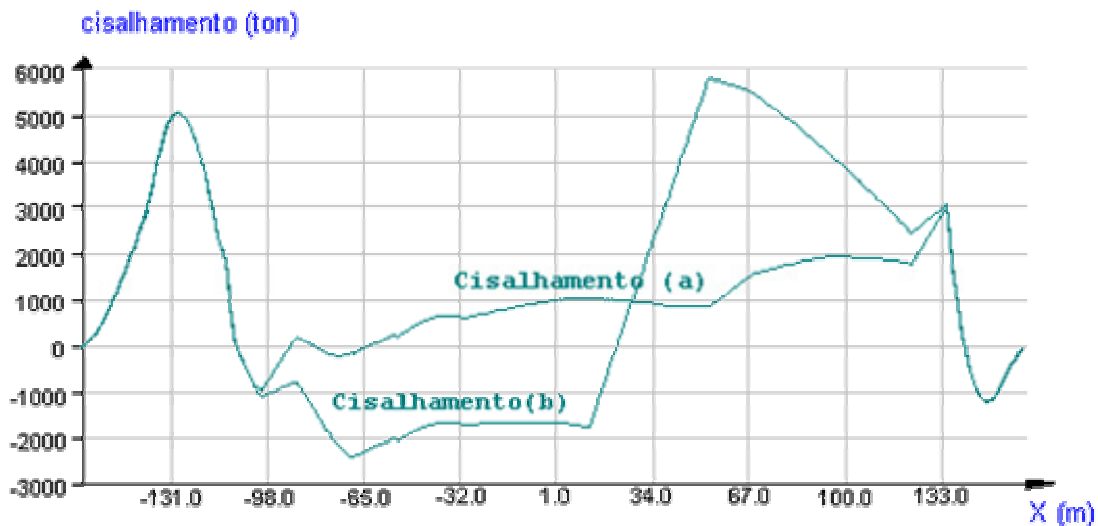


Figura 3.6-Forças de Cisalhamento referentes à Figura 3.5

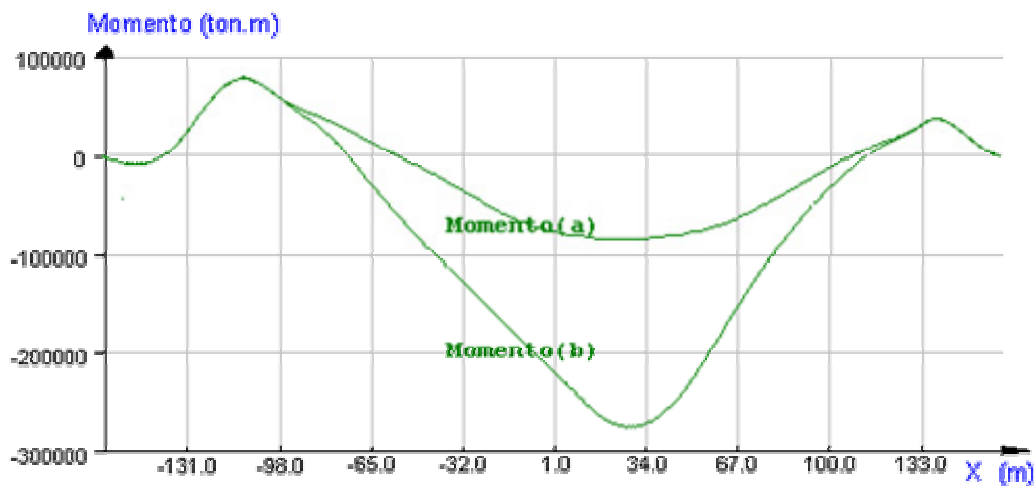


Figura 3.7 – Momento ao longo do eixo longitudinal referente à Figura 3.5

O algoritmo também funciona para plataformas danificadas ou inundadas. O último exemplo mostra o uso do algoritmo numa situação de emergência, quando necessário restaurar a plataforma para a sua posição correta após um tanque ter sido danificado. Considere que a plataforma em operação sofre uma ruptura em um tanque e a Figura 3.8(a) mostra o modelo em equilíbrio após o acidente. O

tanque danificado é mostrado em vermelho. O objetivo é retornar a plataforma a uma posição de calado paralelo. Para isto, é necessária uma redistribuição dos líquidos que se encontram nos outros tanques.

Como é visto na Figura 3.8(b), aplicando o algoritmo, foi possível restaurar a plataforma para a sua posição inicial, mesmo tendo um tanque danificado. É possível até mesmo aplicar os algoritmos de minimização de superfície livre ou minimização de tensão ao longo do eixo da embarcação e se obterá o melhor resultado dentro dos limites de uma plataforma danificada. A única diferença é que o líquido dentro de tanques avariados não pode ser alterado pelo algoritmo e entra como força externa nas restrições de equilíbrio, assim como o líquido de tanques inundados.

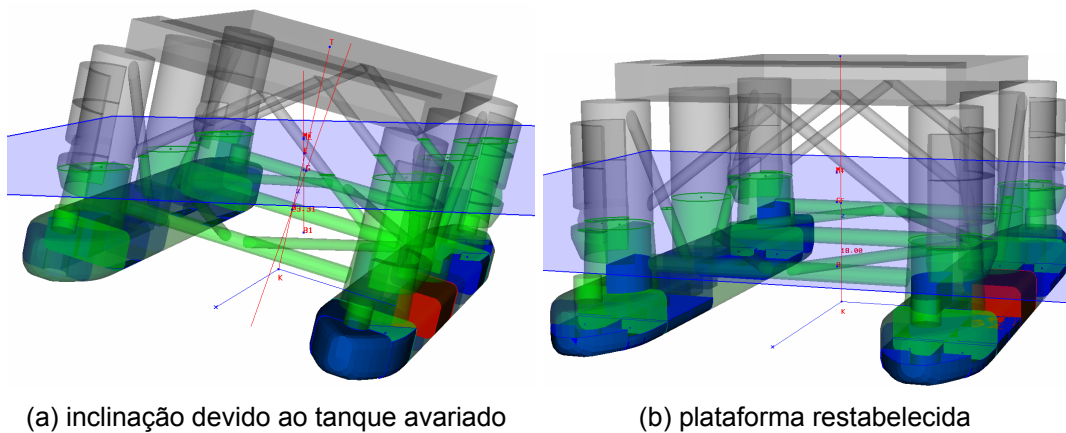


Figura 3.8- Plataforma inclinada devido a um tanque avariado.