Nonlinear dynamic structural optimization of offshore structures using equivalent static models

Gabriel R. Domingos[1], Eduardo N. Lages[1], Adeildo S. Ramos Jr.[1], Anderson T. Oshiro[2], Mauro C. Oliveira[2]

[1]*Laboratory of Scientific Computing and Visualization, Center of Technology, Federal University of Alagoas*
Av. Lourival Melo Mota, 57072-970, Alagoas, Brazil
gabriel.domingos@lccv.ufal.br, enl@lccv.ufal.br, adramos@lccv.ufal.br
[2]*Center of Research Leopoldo Américo Miguez de Mello, PETROBRAS*
Av. Horácio Macedo, 21941-915, Rio de Janeiro, Brazil
mauro@petrobras.com.br, anderson.oshiro@petrobras.com.br

Abstract. Nonlinear dynamic analyses of offshore structures are well-known for their often expensive computational cost. Optimizing such models is an even more complex task because it requires many nonlinear dynamic analyses in sequence. On the other hand, nonlinear static analyses of offshore structures have significant lower computational cost, and can be used to approximate the nonlinear dynamic nature of the original problem and come up with a valid optimized solution. This work proposes the use of an equivalent static model to perform structural optimization applied to anchor systems of offshore structures. The equivalent static model is built based on the field of displacement of the original model, using a penalization factor to approximate the static displacements field to the dynamic one. To encompass more of the dynamic nature of the problem into the static model, not only the field of displacement is used, but also its convex hull. This strategy allows minor differences between the models to be taken into consideration. This penalization factor, chosen at the start of the optimization, as well as the use of the convex hull of the displacement fields were shown to create a fair correspondence between the models, allowing the problem to be solved in a fraction of the original time, and resulting in valid, optimized results.

Keywords: Structural Optimization, Offshore Structures, Anchor systems, Dynamic Models, Equivalent Static Models

1 Introduction

Offshore exploration of O&G is very common on the Brazilian coast. The design of an offshore production unit is a multidisciplinary and complex task which requires lots of human and financial resources. During its operation period, production units can require a series of adjustments on its systems in order to extend their lifespan. One of these systems, the anchor system, is responsible for maintaining the production unit stable and inside an area on which the risers (components of the production system) can operate properly.

Sometimes due to operational needs, it may be necessary to adjust the anchor system to work on a different arrangement, to mitigate risks of the production system to become partially unoperational. Differently of what was proposed by works of Cruces Girón et al. [1] and Monteiro et al. [2], which present a methodology for designing anchor systems from scratch, this work intends on repositioning a working production unit due to operational needs. However, this work considers Safe Operational Zone for the risers (SAFOP) as a constraint, agreeing with what was proposed in the previously mentioned works.

The definition of the equilibrium position of an offshore unit is made by calibrating the lengths of the anchor lines, and the equilibrium position is only valid if a set of constraints is satisfied. These constraints are based on a series of criteria, such as operation, maintenance, safety, good practices, life-cycle management and so on. Defining a new equilibrium position requires the validation of all these different criteria, which is performed based on the results of a computational simulation of the production unit.

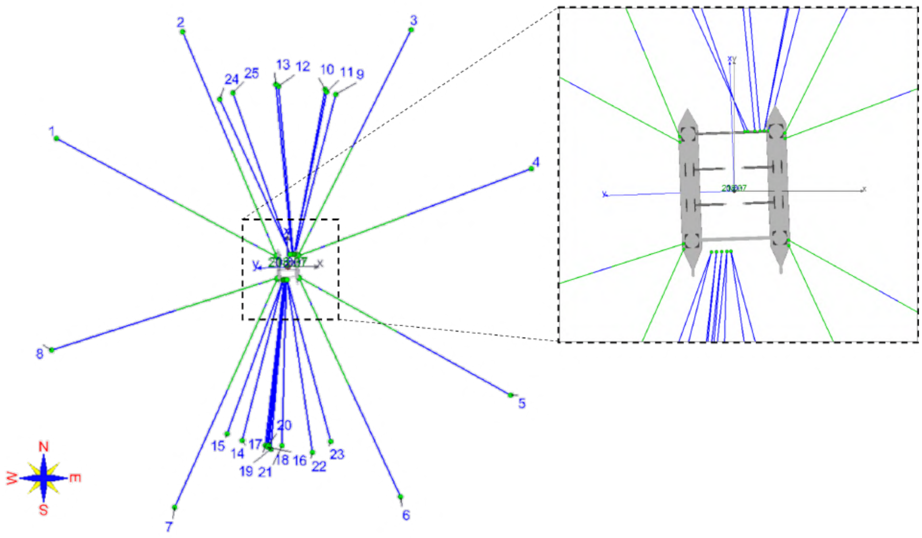


Figure 1. Studied model: semisubmersible with 8 anchor lines (1-8) disposed in 4 clusters, and 17 risers (9-25)

In our case, the simulation is performed using the proprietary software DYNASIM (Nishimoto et al. [3]; Fucatu et al. [4]), which is a software for dynamic analysis of anchored systems. In most cases, these dynamic analyses are expensive in a computational perspective. One sole analysis may take hours — or sometimes days — to run using a personal computer; thus, the task of redefining the equilibrium position and meeting all the necessary criteria easily becomes a fatiguing trial and error process, depending directly on the skills and work experience of the professional responsible for it. Of course, computing techniques, such as parallel and cluster computing, can be used to assess the problem, reducing the time required for a dynamic analysis to run, but the performance of the results will still rely on the professional who is conducting the task.

In the context of anchored systems, a dynamic analysis (DA) is composed of a combination of loads provenient from currents, winds, waves, swells, and the actions of gravity and buoyancy forces, which can be very time-consuming to simulate dynamically. This work proposes a strategy of changing DA for equivalent static analyses (ESA).

This simplification has its cost, and the general behavior of the displacements generated by the ESA fairly differs from the displacements generated by the DA. However, these results can be adjusted by determining a factor k that augments the magnitude of the ESA displacements according to specific metrics, but maintains their direction (see Fig. 3).

To encompass the differences in the direction of the displacement fields, a strategy using the convex hull of the displacement fields is also used. This is further detailed in the text. It is important to note that all results provided by an ESA must be validated afterwards using a DA.

This example provides a great opportunity to explore the problem with a structural optimization approach, as will be discussed in the next section. This makes the search for a solution based on a theoretical formulation, and not on the experience of the professional.

2 Optimization Model

The mathematical formulation of the optimization model is comprised of an objective function — a mathematical function that expresses real criteria that should be minimized — and a set of constraints for the problem — which also express real criteria to mathematical language. Both the objective function and the set of constraints are defined on the space of the design variables, which are the arguments that will determine these functions and define the optimal solution.

A solution for the problem is considered optimal only when it reaches a local or a global minimum of the objective function while also satisfying all the constraints of the problem. The mathematical formulation is presented below, with its variables described in the next subsection.

$$
\begin{aligned}
\min_{\boldsymbol{L}} f(\boldsymbol{L}) = & \sum_{i=1}^{8} (L^i - L_0^i)^2 \\
\text{subject to } & \frac{\mathit{Hull}(\boldsymbol{\Delta})}{\mathit{Hull}(\text{safe limits})} - 1 \leq 0 \\
& 1 - \frac{yaw}{90^\circ} \leq 0 \\
& \frac{yaw}{92^\circ} - 1 \leq 0 \\
& \frac{T^i_{\text{ECs}}}{60\% MBL^i} - 1 \leq 0 \\
& \frac{T^i_{\text{pretensions}}}{40\% MBL^i} - 1 \leq 0 \\
& \frac{\left|\frac{T_b - T_a}{T_a}\right| + \left|\frac{T_a - T_b}{T_b}\right|}{2} - 0.3 \leq 0 \\
& 0.5 \leq \frac{L^i}{L_0^i} \leq 1.5 \\
\text{with } & a = 1, 3, 5, 7 \text{ and } b = 2, 4, 6, 8.
\end{aligned}
\tag{1}
$$

2.1 Design variables

The design variables of the problem are the lengths of the first segments for each anchor line. For the purposes of the optimization procedure, all variables are defined dimensionless, with all their initial values set to 1.0. The bound constraints are then set to 0.5 and 1.5, for each variable. Since there are eight anchor lines, we are using eight design variables.

In eq. (1), the number 8 over the summation makes reference to the eight segment lengths ($L^1 \dots L^8$). Also, in accordance with one of the tension constraints, the indices a and b are used to calculate tensions to each pair of lines that make each cluster (Cluster 1: lines 1 and 2, Cluster 2: lines 3 and 4, and so on).

2.2 Objective function

Since the problem studies a production unit in operation, we established that significant changes on the anchor system may not be desired. For this reason, the objective function was designed to achieve a new equilibrium position that has the arrangement of lines — that is, their lengths — as close as possible from the original arrangement. This is achieved by minimizing the sum of the square differences between the initial lengths of the segments (L_0^i) and their current lengths (L^i).

While minimizing the change in the arrangement, the optimization formulation ensures that every constraint associated with the problem is satisfied in the optimal solution, making it valid.

2.3 Constraints

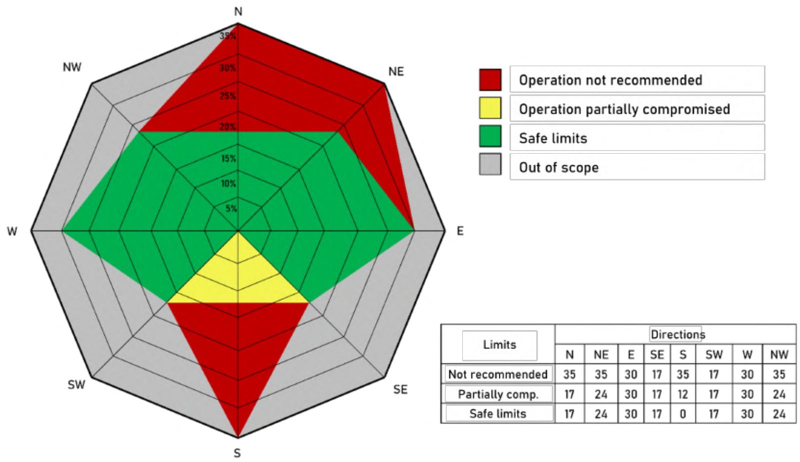
The constraints of the problem arise from various operational criteria, and encompass position, tension and displacement constraints.

The position constraint ensures that the yaw of the production unit in the equilibrium position is greater than 90° and less than 92°, due to operational reasons.

The first tension constraint considers standard regulations. The force of tension in the lines cannot be greater than 60% of their Minimum Breaking Load (MBL) in any environmental case. Also, due to operational reasons, the top tensions in the lines in the equilibrium position cannot be greater than 40% of the line's MBL. The other tension constraint ensures that the tensions in lines of a same cluster cannot differ more than 30% between each other.

The displacement constraint is set as three areas with different levels of operationability, as seen in Fig. 2. For the purposes of this study, we only allowed our optimized results to be inside the safe limits (SAFOP), to ensure maximum operationability at all times. Each sector (direction) allows safe displacements as a fraction of the water depth (WD). That means the numbers in the table presented in Fig. 2 have to be divided by 100 and multiplied by WD to get the amount of horizontal displacement allowed for each level in that direction.

As mentioned before, the adaptation of the DA to the ESA requires a penalization factor k to roughfully match the magnitude of the displacements, and a strategy to encompass the differences in direction is applied.



Limits	Directions							
	N	NE	E	SE	S	SW	W	NW
Not recommended	35	35	30	17	35	17	30	35
Partially comp.	17	24	30	17	12	17	30	24
Safe limits	17	24	30	17	0	17	30	24

Figure 2. Levels of operationability for each direction according to the percentage of water depth

The problem with the adaptation of a DA to an ESA is that the displacement points may satisfy the SAFOP constraint in the ESA, but that may not happen for the DA when validating the optimal results.

The strategy proposed in this work is based simply on checking the displacement constraint not point-to-point, but using the convex hull formed by all the displacement points, and checking if the hull is fully contained inside the safe limits (SAFOP).

It is important to mention that the penalization factor k is not directly applied on the tensions. The factor is applied only on the displacements, and the tensions are then measured in the penalized position.

This adaptation, despite basic and straightforward, has shown to be very effective in producing results with an ESA that are valid when checked using a DA. It solves the problem in a fraction of its original time and produces valid results that can be immediately applied in the field onto the real-case unit.

2.4 Optimization algorithm

For this study, we chose the COBYLA (Constrained Optimization BY Linear Approximations) optimization algorithm by Powell (Powell [5]; Powell [6]). It is a zeroth-order algorithm that supports both nonlinear inequality and equality constraints, based on the construction of linear approximations of the objective function and the constraints via simplexes associated with a trust region procedure. The code used in this work is available in the NLopt library (Johnson [7]), a free/open-source library for nonlinear optimization.

3 Equivalent Static Model

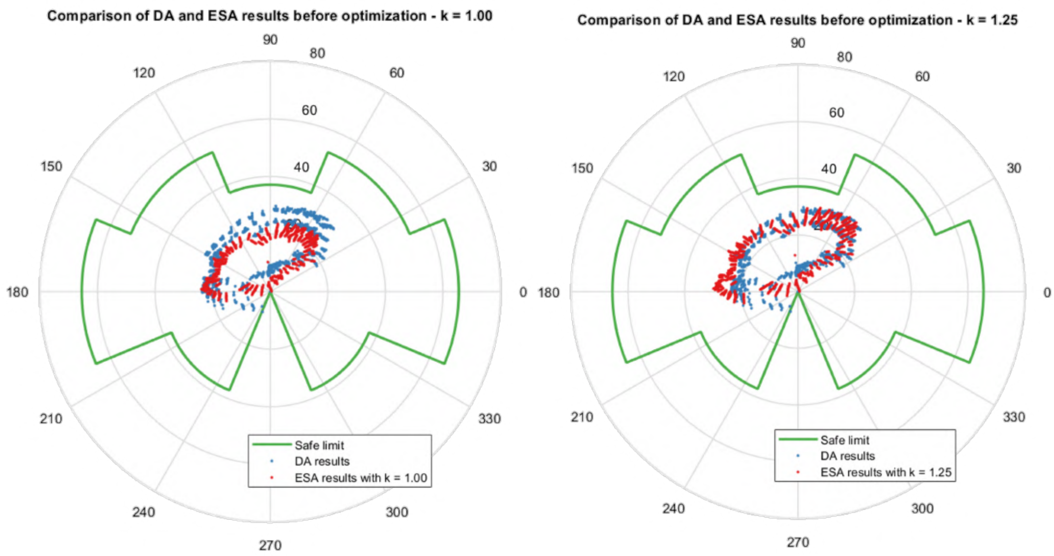
The loads on the equivalent static model are treated as permanent, and the simulation process aims to find the equilibrium position of the model when subjected to the different static loads. This result is achieved by an internal optimization algorithm present in DYNASIM, and is just mentioned here for completeness. It is beyond the scope of this work to explain the intricate details of this formulation.

Another essential part of defining the equivalent static model is the definition of the k factor, which is based on the general behavior of results from both dynamic and equivalent static models. The Fig. 3(a) displays the original displacements obtained by the dynamic model and the original static model, while Fig. 3(b) shows the displacements of the static model penalized by the factor $k = 1.25$.

In order to determine k, one must run a DA and a ESA using the initial model, and get its fields of displacements, such as in Fig. 3(a). Afther that, the value of k can be determined so that, when multiplied by it, both dynamic and equivalent static displacements will have matching peaks of magnitude.

From Fig. 3(b), it can be observed that the correspondence between the models after the penalization is not perfect. Nevertheless, this procedure, in conjunction with the convex hull strategy mentioned in Section 2.3 still yields excellent results, as will be further demonstrated.

The final adjustment in the equivalent static model involves selecting critical load cases. Since we are working with environmental loads, we have to investigate numerous different combinations of actions to ensure evaluating the worst-case scenario for structural safety. However, only a part of these combinations will produce critical displacements that activate the constraints of the optimization problem. Because of that, a strategy is implemented to select only those cases that represent the worst-case scenarios to participate of the optimization problem.



(a) (b)

Figure 3. General behavior of the displacements generated by a DA and an ESA for the model with (a) $k = 1.00$ and (b) $k = 1.25$, before optimization

Considering n as the number of design varibles ($n = 8$ for this example), the algorithm COBYLA will, by default, take steps in the first $n + 1$ iterations only to gather enough information and build the simplex structure necessary to perform the optimization. During these first $n + 1$ iterations, no steps towards a point of minimum are taken. These steps are defined by internal criteria of the algorithm, stressing each one of the design variables separately, while keeping the others fixed. In these steps, we evaluate our model with all possible environmental cases.

At the $n + 2$th iteration, the first step towards optimization is taken. However, on this step, we choose the critical load cases which will be applied to our model in order reduce the number of cases and accelerate the optimization process. The critical load cases are determined by analyzing each one of the first $n + 1$ steps and determining the c load cases that stress the model the most for each iteration (the parameter c is controlled by the user). After obtaining c critical load cases for each of the first $n+1$ of the optimization, a histogram of frequencies is built, counting the number of times each critical load case repeats itself, from 1 to $n + 1$ times.

Finally, the d most frequent load cases are chosen as definite, and only these d cases are evaluated for the rest of the optimization procedure (the parameter d is controlled by the user). In our tests, from the original universe of $1{,}716$ environmental cases, we used only 100 as definite. The average time to perform a static optimization reduced from around 5 hours to a little more than 1 hour, using a Windows 11 64-bit operating system, an Intel i7-12700H CPU and 16 GB of installed RAM. The optimization code was written using Julia programming language (Bezanson et al. [8]) and NLopt library (Johnson [7]).

4 Results

Since the objective of this work is to solve a real-case scenario, the initial results shown below and in Fig. 4 belong to the as laid model, which originated the need to solve this problem.

The average differences in tension for the lines of the clusters in this model were $[0.60\%$, 81.9%, 17.7%, $31.9\%]$. The vector of top tensions on the lines in the equilibrium position is: $[970.23$, 976.15, 975.79, 462.03, 515.50, 432.47, 413.34, $302.13]$ kN, with the values of all lines respecting the maximum of 40% MBL. The

yaw in the equilibrium position in the as laid model was equal to $91.853°$. The results presented here have been generated by a dynamic simulation of the as laid arrangement, without performing any optimization.

By studying these results and comparing with the operational requirements expressed in the set of constraints mentioned in the Section 2.3, we observe that the displacement constraint is not satisfied, with one point of the field of displacements being out of the safe limits (see Fig. 4). Also, one of the tension constraints is not satisfied, with Clusters 2 and 4 with average difference in tension greater than 30%.

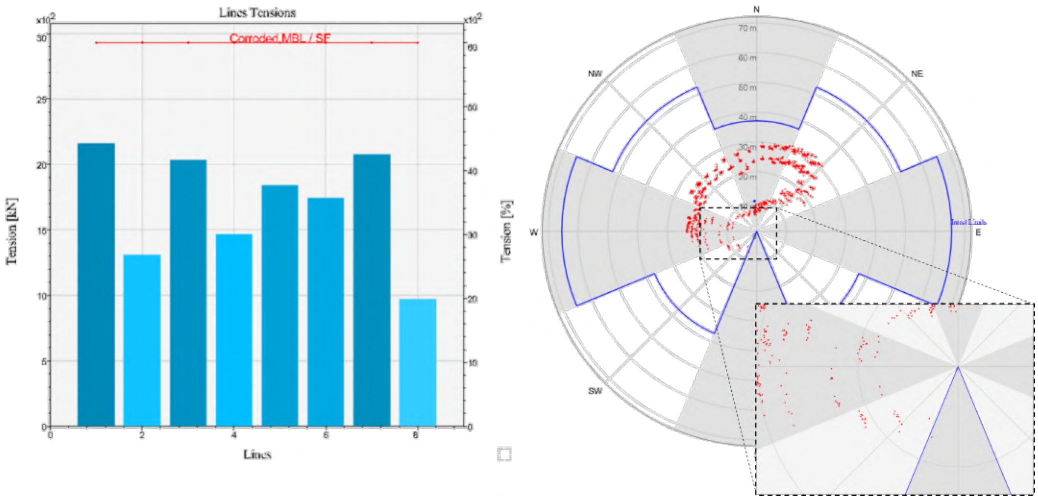


Figure 4. Maximum line top tensions due to the influence of environmental cases along with their limit of 60% MBL (red line), and the field of displacements for the as laid model

Given this input, we applied the methodology described in this work, building the equivalent static analysis and applying the optimization procedure, using the as laid arrangement as the initial condition for our optimization. The number of definite load cases chosen was $d = 100$, the penalization parameter was set to $k = 1.25$, and the displacement constraint used the convex hull procedure described in Section 2.3. The results of the optimization are shown below and in Fig. 5.

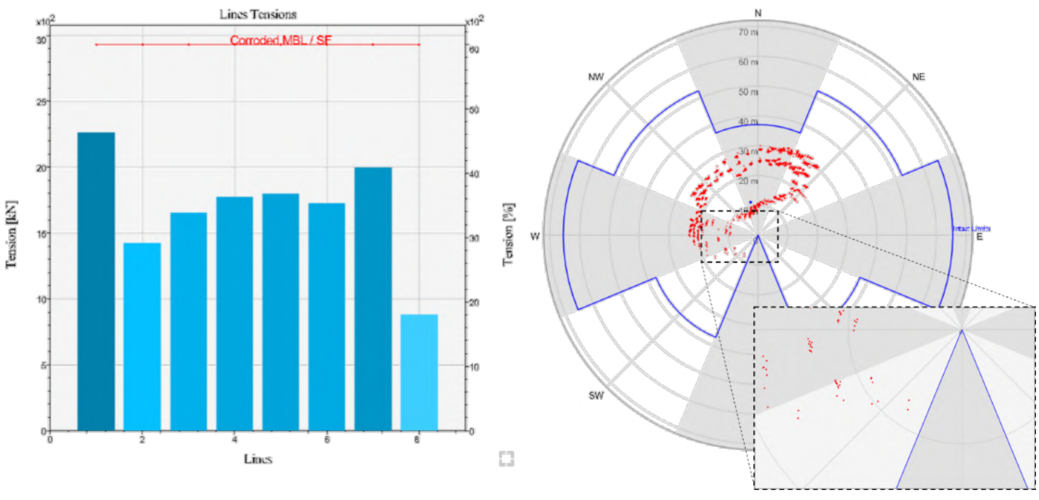


Figure 5. Maximum line top tensions due to the influence of environmental cases along with their limit of 60% MBL (red line), and the field of displacements for the optimized model

The average differences in tension for the lines of the clusters in the optimized model were [3.60%, 30.0%, 14.6%, 30.0%]. The vector of top tensions on the lines in the equilibrium position is: [1,245.22, 1,201.46, 761.63, 566.68, 491.50, 424.99, 407.00, 302.87] kN, with the values of all lines still respecting the maximum of 40% MBL. The yaw in the equilibrium position in the optimized model was equal to $90.886°$. The results presented here have been generated by a dynamic simulation of the optimal solution, used for validating its results.

The total time of optimization for this example was 87 minutes, using equivalent static analyses. It is valid to highlight that the optimization took 129 iterations to converge to a result, with the first $n+1 = 9$ iterations evaluated with $1{,}716$ environmental cases and the next 120 iterations with only 100 environmental cases, as discussed in Section 3. The time taken to run the single DA used for validating the results was 145 minutes.

5 Conclusions

It is possible to see from the results shown that the optimized result is valid, because it satisfies all constraints of the problem in the DA. This means that the optimization procedure was able to calibrate a model which was given as input and obtain a solution that satisfied all the imposed criteria, including those which were not satisfied in the input model.

Extrapolating the numbers seen in Section 4, if the whole optimization was performed using DA, the 129 iterations would take $18{,}705$ minutes to run. This number may not be exact, as we are extrapolating from measured data, but it clearly demonstrates the efficiency of the strategy proposed in this work. It is important to highlight that using this strategy, only two DAs are needed: the first to determine the value of k, as discussed in Section 3, and the second to validate the optimal results.

In the end, the problem can be solved and validated in a fraction of the time it would take to run an optimization with DA, and without relying directly on the experience of the responsible professional. The whole operation runs automatically, searching for an optimal solution according to a mathematical formulation, and does not require the professional to undergo any fatiguing trial and error procedures, and even providing a quick-response tool for the team on the field.

The results presented in this work also apply to different objective functions, constraints and metrics for the determination of the k factor and the d definite environmental cases, and can be explored to solve more complex tasks, even designing an anchor system from scratch, taking into consideration SAFOP constraints.

Acknowledgements. The authors acknowledge PETROBRAS and CNPq for the financial support.

Authorship statement. The authors hereby confirm that they are the sole liable persons responsible for the authorship of this work, and that all material that has been herein included as part of the present paper is either the property (and authorship) of the authors, or has the permission of the owners to be included here.

References

[1] A. R. Cruces Girón, F. N. Corrêa, A. O. Vázquez Hernández, and B. P. Jacob. An integrated methodology for the design of mooring systems and risers. *Marine Structures*, vol. 39, pp. 395–423, 2014.

[2] B. F. Monteiro, J. S. Baioco, C. H. Albrecht, B. S. L. P. Lima, and B. P. Jacob. Optimization of mooring systems in the context of an integrated design methodology. *Marine Structures*, vol. 75, pp. 102874, 2021.

[3] K. Nishimoto, C. H. Fucatu, and I. Q. Masetti. Dynasim – a time domain simulator of anchored FPSO. *Journal of Offshore Mechanics and Artic Engineering-Transactions of the ASME*, vol. 124, n. 4, pp. 203–211, 2002.

[4] C. H. Fucatu, E. A. Tannuri, E. N. Lages, E. S. S. Silveira, F. M. G. Ferreira, K. Nishimoto, and M. C. Oliveira. Dynasim. Registration number: BR512014001550-5, 2016.

[5] M. J. D. Powell. *A Direct Search Optimization Method that Models the Objective and Constraint Functions by Linear Interpolation*, pp. 51–67. Springer Netherlands, Dordrecht, 1994.

[6] M. J. D. Powell. Direct search algorithms for optimization calculations. *Acta Numerica*, vol. 7, pp. 287—-336, 1998.

[7] S. G. Johnson. The NLopt nonlinear-optimization package. `https://github.com/stevengj/nlopt`, 2007.

[8] J. Bezanson, A. Edelman, S. Karpinski, and V. B. Shah. Julia: A fresh approach to numerical computing. *SIAM Review*, vol. 59, n. 1, pp. 65–98, 2017.