

# *Adaptation of the Newton-Raphson and Potra-Pták methods for the solution of nonlinear systems*

## *Adaptação dos métodos de Newton-Raphson e de Potra-Pták para a solução de sistemas de equações não lineares esparsos*

Luiz Antonio Farani de Souza<sup>1</sup>; Emerson Vitor Castelani<sup>2</sup>;  
Wesley Vagner Inês Shirabayashi<sup>3</sup>

### **Abstract**

In this paper we adapt the Newton-Raphson and Potra-Pták algorithms by combining them with the modified Newton-Raphson method by inserting a condition. Problems of systems of sparse nonlinear equations are solved the algorithms implemented in Matlab® environment. In addition, the methods are adapted and applied to space trusses problems with geometric nonlinear behavior. Structures are discretized by the Finite Element Positional Method, and nonlinear responses are obtained in an incremental and iterative process using the Linear Arc-Length path-following technique. For the studied problems, the proposed algorithms had good computational performance reaching the solution with shorter processing time and fewer iterations until convergence to a given tolerance, when compared to the standard algorithms of the Newton-Raphson and Potra-Pták methods.

**Keywords:** Potra-Pták. Space trusses. Nonlinear analysis. Algorithm. Positional Formulation.

### **Resumo**

Neste artigo é feita uma adaptação nos algoritmos dos métodos de Newton-Raphson e de Potra-Pták, combinando-os com o método de Newton-Raphson modificado, por meio da inserção de uma condição. Problemas de sistemas de equações não lineares esparsos são solucionados com os algoritmos implementados em ambiente Matlab®. Em adição, os métodos são adaptados e aplicados em problemas de treliças espaciais com comportamento não linear geométrico. As estruturas são discretizadas por meio do Método Posicional de Elementos Finitos, e as respostas não lineares são obtidas num processo incremental e iterativo, utilizando a técnica de continuação Comprimento de Arco Linear. Para os problemas estudados, os algoritmos propostos tiveram bom desempenho computacional alcançando a solução com menor tempo de processamento e menor número de iterações até a convergência para uma dada tolerância, se comparados aos algoritmos padrões dos métodos de Newton-Raphson e de Potra-Pták.

**Palavras-chave:** Potra-Pták. Treliça espacial. Análise não linear. Algoritmo. Formulação Posicional.

<sup>1</sup> Prof. Dr., Dept. of Civil Engineering, Federal University of Technology, Apucarana, Paraná, Brazil, E-mail: lasouza@utfpr.edu.br

<sup>2</sup> Prof. Dr., Dept. of Mathematics, UEM, Maringá, Paraná, Brazil, E-mail: evcastelani@uem.br

<sup>3</sup> Prof. Dr., Dept. of Mathematics, UEM, Maringá, Paraná, Brazil, E-mail: wvshirabayashi@uem.br

## Introduction

Many problems in Applied Mathematics and Engineering are reduced to finding a nonlinear equations system solution  $\mathbf{F}(\mathbf{x}) = \mathbf{0}$ , and require the development of an iterative method (KIM; CHUN, 2010). The solution of this system is usually difficult to find (AL-TOWAIQ; ABU HOUR, 2017). In the past thirty years or more, with the development of increasingly efficient and fast digital computers, the investigation of nonlinear problems and numerical methods to solve them has increased dramatically (SOUZA; ALVAREZ; LOBAO, 2017).

Newton's iterative method, also known as Newton-Raphson method, with quadratic convergence order, is probably the most well-known and its algorithm is the most widely used (WEERAKOON; FERNANDO, 2000; KIM; CHUN, 2010; MUHAMMAD; MAMAT; WAZIRI, 2013). Various other iterative methods have been developed to find the approximate solution of these systems, such as Quasi Newton methods (KANWAR; SHARMA, 2005; AL-TOWAIQ; ABU HOUR, 2017), Chebyshev-Halley family methods (GUTIERREZ; HERNÁNDEZ, 1997), Jarratt's method (JARRATT, 1969), Potra-Pták's method (POTRA; PTÁK, 1983), Weerakoon and Fernando's method (WEERAKOON; FERNANDO, 2000), Midpoint method (FRONTINI; SORMANI, 2003) among others. It is considered to solve the system of nonlinear equations (KANWAR; KUMAR; BEHL, 2013; FAN; YUAN, 2014):

$$\mathbf{F}(\mathbf{x}) = \mathbf{0}, \quad (1)$$

where  $\mathbf{F} : R^n \rightarrow R^n$  is a continuously differentiable function, where  $\mathbf{F} = (f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_n(\mathbf{x}))^T$  and  $\mathbf{x} = (x_1, x_2, \dots, x_n)^T$ .

The iterative methods for the system solution given in equation (1) are of type  $\mathbf{x}^{(k+1)} = \Phi(\mathbf{x}^{(k)})$  and provide a sequence of vectors  $\{\mathbf{x}^{(k)}\}_{(k=0)}^\infty$ , where the superscript  $k$  is the iteration and  $\mathbf{x}^{(k)}$  is the approximation of the solution in the  $k$ th iteration. From an initial estimate  $\mathbf{x}^{(0)}$ , this sequence may diverge or converge towards a  $i$ th root  $\mathbf{x}^{(*,i)}$  of the system (GRAU-SÁNCHEZ; NOGUERA; GUTIÉRREZ, 2010).

In this paper, we adapt the Newton-Raphson (NR) and Potra-Pták (PP) algorithms by combining them with the modified Newton-Raphson Method (MNR) by inserting a condition. The idea of this modification is: as long as the imposed condition is not satisfied, the sequence of vectors  $\mathbf{x}^{(k)}$  is generated through the iterative process of NR or PP methods; when the approximate solution is close enough

to the root  $\mathbf{x}^{(*,i)}$  (in other words, a condition is satisfied), then subsequent iterations are obtained with the MNR method, until the convergence criteria is satisfied for a given tolerance.

To validate the algorithms of NR and PP hybrid methods developed here, some numerical tests of sparse nonlinear equations system problems were performed with the Matlab® software support (MATLAB, 2015), verifying the computational performance of both. The numerical results obtained in the simulations are compared with algorithms of NR and PP standard methods.

In addition, as an application of the algorithms, problems with spatial trusses and geometric nonlinear behavior are solved. The NR and PP hybrid methods are adapted to the structural problem in an incremental and iterative procedure, associating them with the Linear Arc-Length path-following technique and the Finite Element Positional Method.

Different approaches for nonlinear structural analysis of trusses were proposed. Saffari; Mansouri (2011) suggested a two-point method with fourth-order convergence. Mahdavi *et al.* (2015) proposed an iterative method free from second derivative originated from modified Chebyshev and cubic spline's schemes. Souza *et al.* (2018) proposed new algorithms based on two-step methods with cubic convergence order and combined with the Linear Arc-Length path-following technique. Mohit; Sharifi; Tavakoli (2020) proposed three-step iterative techniques, with fourth-order convergence, for solving nonlinear equations used for geometrically nonlinear analysis of space truss. Dehghani *et al.* (2020) proposed an improved perturbation algorithm to refine the classical methods in numerical computing techniques such as the Newton-Raphson method.

## Iterative methods for nonlinear equation system solution

The Newton-Raphson method is described by the equations (KANWAR; KUMAR; BEHL, 2013):

$$\mathbf{x}^{(k)} = \mathbf{x}^{(k-1)} + \mathbf{dx}^{(k)}, \quad (2)$$

$$\mathbf{dx}^{(k)} = -[\mathbf{F}'(\mathbf{x}^{(k-1)})]^{-1}\mathbf{F}(\mathbf{x}^{(k-1)}), \quad (3)$$

for all  $k = 1, 2, \dots$ , being  $\mathbf{F}'(\mathbf{x}^{(k-1)})$  the Jacobian matrix of  $\mathbf{F}$  in  $\mathbf{x}^{(k-1)}$ . If  $\mathbf{F}'(\mathbf{x}^{(k-1)})$  is not singular in the solution, the Newton method convergence is quadratic (WEERAKOON; FERNANDO, 2000). In its implementation, the explicit calculation of  $[\mathbf{F}'(\mathbf{x}^{(k-1)})]^{-1}$  can be

avoided solving  $\mathbf{F}'(\mathbf{x}^{(k-1)})\mathbf{dx}^{(k)} = -\mathbf{F}(\mathbf{x}^{(k-1)})$  by a direct method, LU factorization, for example, or an iterative method, Conjugate Gradient, for example, and then a new approximation  $\mathbf{x}^{(k)}$  is obtained with the addition of  $\mathbf{dx}^{(k)}$  and  $\mathbf{x}^{(k-1)}$  (SOUZA *et al.*, 2018).

Potra; Pták (1983) developed a two-step method based on the Newton method, consisting of two evaluations of the given vector  $\mathbf{F}$  and requiring only one evaluation of the Jacobian matrix  $\mathbf{F}'$  at each iteration, with cubic convergence and a higher efficiency index than Newton's method. The Potra-Pták method is given by the following equations (SOLEYMANI; SHARMA; TOHIDI, 2012):

$$\mathbf{y}^{(k)} = \mathbf{x}^{(k-1)} + \mathbf{dx}^{(k)}, \quad (4)$$

$$\mathbf{x}^{(k)} = \mathbf{x}^{(k-1)} + \mathbf{dx}^{(k)} + \mathbf{dy}^{(k)}, \quad (5)$$

$$\mathbf{dx}^{(k)} = -[\mathbf{F}'(\mathbf{x}^{(k-1)})]^{-1}\mathbf{F}(\mathbf{x}^{(k-1)}), \quad (6)$$

$$\mathbf{dy}^{(k)} = -[\mathbf{F}'(\mathbf{x}^{(k-1)})]^{-1}\mathbf{F}(\mathbf{y}^{(k-1)}). \quad (7)$$

**Algorithm 1:** The Newton-Raphson hybrid method.

```

1 begin
2   k ← 0
3   error ← 1
4   met ← 0
5   Calculate  $\mathbf{F}'(\mathbf{x}^{(0)})$  and  $\mathbf{F}(\mathbf{x}^{(0)})$ 
6   Decompose the matrix  $\mathbf{F}'(\mathbf{x}^{(0)})$  in the triangular
   matrices  $\mathbf{L}$  and  $\mathbf{U}$ 
7    $\mathbf{y}^{(0)} \leftarrow [\mathbf{L}]^{-1}\mathbf{F}(\mathbf{x}^{(0)})$ 
8    $\mathbf{dx}^{(1)} \leftarrow -[\mathbf{U}]^{-1}\mathbf{y}^{(0)}$ 
9   while  $k < k_{max}$  do
10    k ← k + 1
11     $\mathbf{x}^{(k)} \leftarrow \mathbf{x}^{(k-1)} + \mathbf{dx}^{(k)}$ 
12    Calculate  $\mathbf{F}(\mathbf{x}^{(k)})$ 
13    error ←  $\|\mathbf{F}(\mathbf{x}^{(k)})\|$ 
14    if error < tol then
15      Out  $\mathbf{x}^{(k)}$ 
16      Loop's out
17    end
18    if met = 0 then
19      if  $\|\mathbf{F}(\mathbf{x}^{(k)})\| \leq \min(\eta tol, \|\mathbf{dx}^{(k)}\|)$  then
20        met ← 1
21      end
22    end
23    if met = 0 then
24      Calculate  $\mathbf{F}'(\mathbf{x}^{(k)})$ 
25      Decompose the matrix  $\mathbf{F}'(\mathbf{x}^{(k)})$  in the
      triangular matrices  $\mathbf{L}$  and  $\mathbf{U}$ 
26    end
27     $\mathbf{y}^{(k)} \leftarrow [\mathbf{L}]^{-1}\mathbf{F}(\mathbf{x}^{(k)})$ 
28     $\mathbf{dx}^{(k+1)} \leftarrow -[\mathbf{U}]^{-1}\mathbf{y}^{(k)}$ 
29  end
30 end

```

In the Potra-Pták method the same Jacobian matrix is used to solve the linear equation systems generated at each iteration; thus, these systems can be solved via decomposition (eg LU factorization), since a single factorization at the beginning of the iteration is required. The algorithmic versions of Newton-Raphson and Potra-Pták standard methods are presented in Souza (2015). Algorithms 1 and 2 show the algorithms of NR and PP hybrid methods, respectively, with the LU factorization method.

The input data in Algorithms 1 and 2 are: function  $\mathbf{F}(\mathbf{x}^{(k-1)})$ ; Jacobian matrix  $\mathbf{F}'(\mathbf{x}^{(k-1)})$ ; initial approach  $\mathbf{x}^{(0)}$ ; maximum number of iterations  $k_{max}$ ; tolerance  $tol$ ; and constant  $\eta$ . The algorithm outputs are: approximate solution  $\mathbf{x}^{(k)}$ ; and number of iterations to convergence  $k$ . In the algorithms of the NR and PP hybrid methods, it is observed the insertion of the condition  $\|\mathbf{F}(\mathbf{x}^{(k)})\| \leq \min(\eta tol, \|\mathbf{dx}^{(k)}\|)$ , being  $\eta > 1$  the input constant, see line 19 of Algorithm 1 and line 24 of Algorithm 2.

**Algorithm 2:** The Potra-Pták hybrid method.

```

1 begin
2   k ← 0
3   error ← 1
4   met ← 0
5   Calculate  $\mathbf{F}'(\mathbf{x}^{(0)})$  and  $\mathbf{F}(\mathbf{x}^{(0)})$ 
6   Decompose the matrix  $\mathbf{F}'(\mathbf{x}^{(0)})$  in the triangular matrices
    $\mathbf{L}$  and  $\mathbf{U}$ 
7    $\mathbf{y}^{(0)} \leftarrow [\mathbf{L}]^{-1}\mathbf{F}(\mathbf{x}^{(0)})$ 
8    $\mathbf{dx}^{(1)} \leftarrow -[\mathbf{U}]^{-1}\mathbf{y}^{(0)}$ 
9   while  $k < k_{max}$  do
10    k ← k + 1
11     $\mathbf{x}^{(k)} \leftarrow \mathbf{x}^{(k-1)} + \mathbf{dx}^{(k)}$ 
12    if met = 0 then
13      Calculate  $\mathbf{F}(\mathbf{x}^{(k)})$ 
14       $\mathbf{y}^{(k)} \leftarrow [\mathbf{L}]^{-1}\mathbf{F}(\mathbf{x}^{(k)})$ 
15       $\mathbf{dy}^{(k)} \leftarrow -[\mathbf{U}]^{-1}\mathbf{y}^{(k)}$ 
16       $\mathbf{x}^{(k)} \leftarrow \mathbf{x}^{(k-1)} + \mathbf{dx}^{(k)} + \mathbf{dy}^{(k)}$ 
17    end
18    Calculate  $\mathbf{F}(\mathbf{x}^{(k)})$ 
19    error =  $\|\mathbf{F}(\mathbf{x}^{(k)})\|$ 
20    if error < tol then
21      Output  $\mathbf{x}^{(k)}$ 
22      Loop's output
23    end
24    if met = 0 then
25      if  $\|\mathbf{F}(\mathbf{x}^{(k)})\| \leq \min(\eta tol, \|\mathbf{dx}^{(k)}\|)$  then
26        met ← 1
27      end
28    end
29    if met = 0 then
30      Calculate  $\mathbf{F}'(\mathbf{x}^{(k)})$ 
31      Decompose the matrix  $\mathbf{F}'(\mathbf{x}^{(k)})$  in the triangular
      matrices  $\mathbf{L}$  and  $\mathbf{U}$ 
32    end
33     $\mathbf{y}^{(k)} \leftarrow [\mathbf{L}]^{-1}\mathbf{F}(\mathbf{x}^{(k)})$ 
34     $\mathbf{dx}^{(k+1)} \leftarrow -[\mathbf{U}]^{-1}\mathbf{y}^{(k)}$ 
35  end

```

When approaching the solution of  $\mathbf{F}(\mathbf{x}^{(k)}) = \mathbf{0}$ , or equivalent  $\|\mathbf{F}(\mathbf{x}^{(k)})\| < tol$ , we have that the derivatives of  $\mathbf{F}$  will not undergo major changes. Thus, the cost of new assessments close to the solution becomes unnecessary. We fixed, in our proposal, the linear approximation of  $\mathbf{F}(\mathbf{x}^{(k)})$  close to  $\mathbf{x}^{(k)}$  when  $\|\mathbf{F}(\mathbf{x}^{(k)})\| < \eta tol$ .

In other words, if  $\|\mathbf{F}(\mathbf{x}^{(k)})\| < \eta tol$ , it means that we are close to meeting the condition  $\|\mathbf{F}(\mathbf{x}^{(k)})\| < tol$ , and we stop evaluating the derivatives of  $\mathbf{F}$  and fix the last Jacobian matrix to solve the remaining linear systems until the condition  $\|\mathbf{F}(\mathbf{x}^{(k)})\| < tol$ . Since it is not possible to accurately determine the meaning of near or far from the solution, the parameter  $\eta$  is a heuristic parameter.

Thereby, when the condition  $\|\mathbf{F}(\mathbf{x}^{(k)})\| \leq \min\|\eta tol, \|\mathbf{d}\mathbf{x}^{(k)}\|\|$  is satisfied, the iterations become the modified Newton-Raphson method, instead of the Newton-Raphson or Potra-Pták method and, in this case, a single system of linear equations is solved and only the vector  $\mathbf{F}$  is updated, since the Jacobian matrix  $\mathbf{F}'$  is kept constant in the iterative cycle.

### Numerical examples

The algorithms developed in this paper were implemented in the Matlab® software, version 8.6 R2015b (MATLAB, 2015), and the computer simulations were realized in the Core i7 - 3537U computer with 8GB RAM Memory. In this section, numerical tests of nonlinear equation systems are presented to validate the proposed algorithms. Then, the algorithms are adapted to the structural problem and are applied to solve three spatial truss problems with geometric nonlinear behavior: circular dome; trussed beam; and circular truss.

#### Systems of sparse nonlinear equations

If the stopping criteria is satisfied  $\|\mathbf{F}(\mathbf{x}^{(k)})\| < tol$ , so the approximate solution  $\mathbf{x}(k)$  is taken as the root of the system given in equation (1). The systems of sparse nonlinear equations with  $\eta \geq 1000$  variables studied are described in Table 1. The methods are compared based on the following parameters: processing time in seconds ( $t$ ); total number of iterations to convergence to the system solution ( $k_{total}$ ) and the Euclidean norm of the vector  $\|\mathbf{F}\|$ .

For the calculation of the processing time, it was used the stopwatch timer functions tic and toc. The tic function records the current time, and the toc function uses the recorded value to calculate the elapsed time. The program was run three times for each simulation and, then, the processing time was calculated using the average of the values.

**Table 1** – Systems of sparse nonlinear equations.

Nonlinear equations system	
<b>Problem 1*</b>	$f_i = \cos(x_i) - 1 = 0, i = 1, 2, \dots, n$ $\mathbf{x}^* = [0 \ 0 \ 0 \ \dots \ 0]^T$ $\mathbf{x}^{(0)} = [0.5 \ 0.5 \ 0.5 \ \dots \ 0.5]^T$ $tol = 1.0 \times 10^{-15}$
<b>Problem 2*</b>	$f_i = \exp(x_i) - 1 = 0, i = 1, 2, \dots, n$ $\mathbf{x}^* = [0 \ 0 \ 0] \dots [0]^T$ $\mathbf{x}^{(0)} = [0.5 \ 0.5 \ 0.5 \ \dots \ 0.5]^T$ $tol = 1.0 \times 10^{-15}$
<b>Problem 3</b>	$f_i = \exp(x_i) + x_i x_{i+1}^2 - x_{i+1}^3 - 1 = 0,$ $i = 1, 2, \dots, n - 1$ $f_n = \exp(x_1) + x_n^2 - 1 = 0$ $\mathbf{x}^* = [0 \ 0 \ \dots \ 0]^T$ $\mathbf{x}^{(0)} = [5 \ 5 \ \dots \ 5]^T$ $tol = 1.0 \times 10^{-15}$
<b>Problem 4*</b>	$f_i = \exp(x_{i+1}) + x_i^2 - 1 = 0,$ $i = 1, 2, \dots, n - 1$ $f_n = \exp(x_{i+1}) + x_1^2 - 1 = 0$ $\mathbf{x}^* = [0 \ 0 \ \dots \ 0]^T$ $\mathbf{x}^{(0)} = [2 \ 2 \ \dots \ 2]^T$ $tol = 1.0 \times 10^{-15}$

\*In Darvishi; Shin (2011)

Source: The authors.

The methods considered for the analysis are: Newton-Raphson method, Potra-Pták method, Newton-Raphson hybrid method, Algorithm 1, and Potra-Pták hybrid method, Algorithm 2. Numerical results are presented in the Table 2, where  $n$  is the number of equations and  $nz$  is the number of non-null elements of the Jacobian matrix  $\mathbf{F}'$ .

It is observed that Algorithm 2 reached the solution with a shorter processing time compared to the other methods. In the simulations, a constant  $\eta$  was adopted as same value as  $n$ . Although the total number of iterations until the convergence to the solution to be equal to or greater in simulations compared to the standard Potra-Pták method, the hybrid Potra-Pták method was more efficient in processing time.

#### Application in spatial trusses with nonlinearity geometric problems

The nonlinear equations system that governs the static equilibrium of a structure with geometric nonlinear behavior (REZAIIEE-PAJAND *et al.*, 2014; MAXIMIANO *et al.*, 2014; SOUZA *et al.*, 2017; SOUZA *et al.*, 2018) is given by:

$$\mathbf{g} = \lambda \mathbf{F}_r - \mathbf{F}_{int}(\mathbf{d}) = \mathbf{0}, \quad (8)$$

**Table 2** – Numerical results of the systems of sparse nonlinear equations.

		Solution method	$\mathbf{k}_{\text{total}}$	$\ \mathbf{F}\ $	$t(\text{s})$
<b>Problem 1</b>					
$n = 10000$		Algorithm 2 (hybrid PP)	101	0	9.51
		Algorithm 1 (hybrid NR)	97	0	13.91
$nz = 10000$		Newton-Raphson	26	0	16.63
		Potra-Pták	19	0	12.03
<b>Problem 2</b>					
$n = 15000$		Algorithm 2 (hybrid PP)	4	0	4.07
		Algorithm 1 (hybrid NR)	6	0	7.03
$nz = 15000$		Newton-Raphson	6	0	8.41
		Potra-Pták	4	0	5.42
<b>Problem 3</b>					
$n = 10000$		Algorithm 2 (hybrid PP)	350	$8.88 \times 10^{-16}$	10.66
		Algorithm 1 (hybrid NR)	314	$8.88 \times 10^{-16}$	14.07
$nz = 20000$		Newton-Raphson	28	$4.44 \times 10^{-16}$	17.27
		Potra-Pták	20	$2.22 \times 10^{-16}$	12.12
<b>Problem 4</b>					
$n = 10000$		Algorithm 2 (hybrid PP)	7	0	3.78
		Algorithm 1 (hybrid NR)	9	0	6.05
$nz = 20000$		Newton-Raphson	9	0	6.95
		Potra-Pták	6	0	4.52

Source: The authors.

where  $\mathbf{g}$  is the vector of unbalanced forces,  $\mathbf{F}_{int}$  is the vector of internal forces, evaluated as a function of the coordinate vector at the nodal points of structure  $\mathbf{d}$ , and  $\lambda$  is the load parameter responsible for scaling the reference vector  $\mathbf{F}_r$ . The equation (8) is the system with  $(n + 1)$  variables, being  $n$  nodal coordinates ( $\mathbf{d}$ ) and one load parameter ( $\lambda$ ), but only  $n$  equations. So, a constraint equation is added to the system, given by (LEON *et al.*, 2011):

$$c(\mathbf{d}, \lambda) = 0. \quad (9)$$

The structural problem, equations (8) and (9), is solved incrementally and iteratively: for each load step, there is a nonlinear system of equations to be solved, in other words, *LS* systems are solved. Applying the Newton-Raphson method to the system, the following structural problem appears (BATHE, 2006; SOUZA, 2015):

$$\mathbf{K}^{(k-1)} \delta \mathbf{d}^{(k)} = \mathbf{g}^{(k)} = \lambda^{(k)} \mathbf{F}_r - \mathbf{F}_{int}(\mathbf{d}^{(k-1)}), \quad (10)$$

$\mathbf{K}^{(k-1)} = \frac{\delta \mathbf{g}^{(k-1)}}{\delta \mathbf{d}}$  is the representative stiffness matrix of the structural system. The total load parameter ( $\lambda$ ) and the nodal coordinate vector ( $\mathbf{d}$ ) in a load step  $t + \Delta t$  and iteration  $k$  are updated by, respectively

$$\lambda^{(k)} = \lambda^{(k-1)} + \delta \lambda^{(k)}, \quad (11)$$

$$\mathbf{d}^{(k)} = \mathbf{d}^{(k-1)} + \delta \mathbf{d}^{(k)}. \quad (12)$$

Combining the equations (10) and (11), the expression appears in  $\delta \mathbf{d}^{(k)}$  (CRISFIELD, 1991):

$$\delta \mathbf{d}^{(k)} = \delta \mathbf{d}_g^{(k)} + \delta \lambda^{(k)} \delta \mathbf{d}_r^{(k)}, \quad (13)$$

where  $\delta \lambda^{(k)}$  is the sub-increment of the load parameter that must be evaluated over the iterative cycle, and  $\delta \mathbf{d}_g^{(k)}$  and  $\delta \mathbf{d}_r^{(k)}$  are obtained, respectively, by:

$$\delta \mathbf{d}_g^{(k)} = [\mathbf{K}^{(k-1)}]^{-1} \mathbf{g}^{(k-1)}, \quad (14)$$

$$\delta \mathbf{d}_r^{(k)} = [\mathbf{K}^{(k-1)}]^{-1} \mathbf{F}_r. \quad (15)$$

The load parameter increments ( $\Delta \lambda$ ) and the nodal coordinates vector ( $\Delta \mathbf{d}$ ) in the load step  $t + \Delta t$  and iteration  $k$  are evaluated by, respectively:

$$\Delta \lambda^{(k)} = \Delta \lambda^{(k-1)} + \delta \lambda^{(k)}, \quad (16)$$

$$\Delta \mathbf{d}^{(k)} = \Delta \mathbf{d}^{(k-1)} + \delta \mathbf{d}^{(k)}. \quad (17)$$

The sub-increment of load  $\delta \lambda^{(k)}$  is evaluated at each iteration in the incremental process by the path-following technique proposed by (RIKS, 1972, 1979). The equation for determining the correction of  $\delta \lambda^{(k)}$  is:

$$\delta \lambda^{(k)} = - \frac{\Delta \mathbf{d}^{(0)T} \delta \mathbf{d}_g^{(k)}}{\Delta \mathbf{d}^{(0)T} \delta \mathbf{d}_r^{(k)}}. \quad (18)$$

The algorithms adapted to the structural problem, associated with the Linear Arc-Length technique, referring to the hybrid Newton-Raphson and Potra-Pták methods are described in Algorithms 3 and 4, respectively. The algorithms of the standard Newton-Raphson and Pótra-Pták methods are presented in Souza *et al.* (2018).

**Algorithm 3:** Hybrid Newton-Raphson method adapted to the structural problem.

```

1 begin
2    $\Delta d \leftarrow 0, \lambda \leftarrow 0$ 
3   for  $LS \leftarrow 1, \dots, LS_{max}$  do
4      $met \leftarrow 0$ 
5     Calculate the stiffness matrix  $\mathbf{K}$ 
6     Decompose the matrix  $\mathbf{K}$  in the triangular
       matrices  $\mathbf{L}$  and  $\mathbf{U}$ 
7      $\mathbf{y} \leftarrow [\mathbf{L}]^{-1} \mathbf{F}_r(\Delta P)$ 
8      $\delta \mathbf{d}_r \leftarrow [\mathbf{U}]^{-1} \mathbf{y}$ 
9      $\Delta \lambda \leftarrow \Delta l / \|\delta \mathbf{d}_r\|$ 
10    if  $\Delta \mathbf{d}^T \delta \mathbf{d}_r < 0$  then
11       $\Delta \lambda \leftarrow -\Delta \lambda$ 
12    end
13     $\Delta \mathbf{d}^{(0)} \leftarrow \Delta \lambda \delta \mathbf{d}_r$ 
14     $\Delta \mathbf{d} \leftarrow \Delta \mathbf{d}^{(0)}$ 
15     $\mathbf{g} \leftarrow (\lambda + \Delta \lambda) \mathbf{F}_r(\Delta P) - \mathbf{F}_{int}(\mathbf{d} + \Delta \mathbf{d})$ 
16    for  $k \leftarrow 1, \dots, k_{max}$  do
17       $\mathbf{y} \leftarrow [\mathbf{L}]^{-1} \mathbf{g}$ 
18       $\delta \mathbf{d}_g \leftarrow [\mathbf{U}]^{-1} \mathbf{y}$ 
19       $\delta \lambda \leftarrow -(\Delta \mathbf{d}^{(0)T} \delta \mathbf{d}_g) / (\Delta \mathbf{d}^{(0)T} \delta \mathbf{d}_r)$ 
20       $\delta \mathbf{d} \leftarrow \delta \mathbf{d}_g + \delta \lambda \delta \mathbf{d}_r$ 
21       $\Delta \mathbf{d} \leftarrow \Delta \mathbf{d} + \delta \mathbf{d}$ 
22       $\Delta \lambda \leftarrow \Delta \lambda + \delta \lambda$ 
23       $\mathbf{g} \leftarrow (\lambda + \Delta \lambda) \mathbf{F}_r(\Delta P) - \mathbf{F}_{int}(\mathbf{d} + \Delta \mathbf{d})$ 
24      if  $\|\mathbf{g}\| \leq tol \|\mathbf{F}_r(\Delta P)\|$  then
25        Finish the execution of for
26      end
27      if  $met = 0$  then
28        if  $error \leq \min(\eta tol, \|\delta \mathbf{d} / \Delta \mathbf{d}\|)$ 
29          then
30             $met \leftarrow 1$ 
31          end
32        end
33        if  $met = 0$  then
34          Calculate  $\mathbf{K}(\mathbf{d} + \Delta \mathbf{d})$ 
35          Decompose the matrix  $\mathbf{K}$  in the
            triangular matrices  $\mathbf{L}$  and  $\mathbf{U}$ 
36           $\mathbf{y} \leftarrow [\mathbf{L}]^{-1} \mathbf{F}_r$ 
37           $\delta \mathbf{d}_r \leftarrow [\mathbf{U}]^{-1} \mathbf{y}$ 
38        end
39         $\mathbf{d} \leftarrow \mathbf{d} + \Delta \mathbf{d}$ 
40         $\lambda \leftarrow \lambda + \Delta \lambda$ 
41         $\Delta l \leftarrow^0 \Delta l (Nd/k)^{0.5}$ 
42      end
43    end
44  end
45 end

```

**Algorithm 4:** Hybrid Potra-Pták method adapted to the structural problem.

```

1 begin
2    $\Delta \mathbf{d} \leftarrow 0, \lambda \leftarrow 0$ 
3   for  $LS \leftarrow 1, \dots, LS_{max}$  do
4      $met \leftarrow 0$ 
5     Calculate the stiffness matrix  $\mathbf{K}$ 
6     Decompose the matrix  $\mathbf{K}$  in the triangular
       matrices  $\mathbf{L}$  and  $\mathbf{U}$ 
7      $\mathbf{y} \leftarrow [\mathbf{L}]^{-1} \mathbf{F}_r(\Delta P)$ 
8      $\delta \mathbf{d}_r \leftarrow [\mathbf{U}]^{-1} \mathbf{y}$ 
9      $\Delta \lambda \leftarrow \Delta l / \|\delta \mathbf{d}_r\|$ 
10    if  $\Delta \mathbf{d}^T \delta \mathbf{d}_r < 0$  then
11       $\Delta \lambda \leftarrow -\Delta \lambda$ 
12    end
13     $\Delta \mathbf{d}^{(0)} \leftarrow \Delta \lambda \delta \mathbf{d}_r$ 
14     $\Delta \mathbf{d} \leftarrow \Delta \mathbf{d}^{(0)}$ 
15     $\mathbf{g} \leftarrow (\lambda + \Delta \lambda) \mathbf{F}_r(\Delta P) - \mathbf{F}_{int}(\mathbf{d} + \Delta \mathbf{d})$ 
16    for  $k \leftarrow 1, \dots, k_{max}$  do
17       $\mathbf{y} \leftarrow [\mathbf{L}]^{-1} \mathbf{g}$ 
18       $\delta \mathbf{d}_g \leftarrow [\mathbf{U}]^{-1} \mathbf{y}$ 
19       $\delta \lambda_1 \leftarrow -(\Delta \mathbf{d}^{(0)T} \delta \mathbf{d}_g) / (\Delta \mathbf{d}^{(0)T} \delta \mathbf{d}_r)$ 
20       $\delta \mathbf{d}_1 \leftarrow \delta \mathbf{d}_g + \delta \lambda_1 \delta \mathbf{d}_r$ 
21       $\Delta \mathbf{d} \leftarrow \Delta \mathbf{d} + \delta \mathbf{d}_1$ 
22       $\Delta \lambda \leftarrow \Delta \lambda + \delta \lambda_1$ 
23      if  $met = 0$  then
24         $\mathbf{g} \leftarrow (\lambda + \Delta \lambda) \mathbf{F}_r(\Delta P) - \mathbf{F}_{int}(\mathbf{d} + \Delta \mathbf{d})$ 
25         $\mathbf{y} \leftarrow [\mathbf{L}]^{-1} \mathbf{g}$ 
26         $\delta \mathbf{y}_g \leftarrow [\mathbf{U}]^{-1} \mathbf{y}$ 
27         $\delta \lambda_2 \leftarrow -(\Delta \mathbf{d}^{(0)T} \delta \mathbf{y}_g) / (\Delta \mathbf{d}^{(0)T} \delta \mathbf{d}_r)$ 
28         $\delta \mathbf{d}_2 \leftarrow \delta \mathbf{y}_g + \delta \lambda_2 \delta \mathbf{d}_r$ 
29         $\Delta \mathbf{d} \leftarrow \Delta \mathbf{d} + \delta \mathbf{d}_2$ 
30         $\Delta \lambda \leftarrow \Delta \lambda + \delta \lambda_2$ 
31      end
32       $\mathbf{g} \leftarrow (\lambda + \Delta \lambda) \mathbf{F}_r(\Delta P) - \mathbf{F}_{int}(\mathbf{d} + \Delta \mathbf{d})$ 
33      if  $\|\mathbf{g}\| \leq tol \|\mathbf{F}_r(\Delta P)\|$  then
34        Finish the execution of for
35      end
36      if  $met = 0$  then
37        if  $error \leq \min(\eta tol, \|\delta \mathbf{d}_1 / \Delta \mathbf{d}\|)$ 
38          then
39             $met \leftarrow 1$ 
40          end
41        end
42        if  $met = 0$  then
43          Calculate  $\mathbf{K}(\mathbf{d} + \Delta \mathbf{d})$ 
44          Decompose the matrix  $\mathbf{K}$  in the
            triangular matrices  $\mathbf{L}$  and  $\mathbf{U}$ 
45           $\mathbf{y} \leftarrow [\mathbf{L}]^{-1} \mathbf{F}_r$ 
46           $\delta \mathbf{d}_r \leftarrow [\mathbf{U}]^{-1} \mathbf{y}$ 
47        end
48         $\mathbf{d} \leftarrow \mathbf{d} + \Delta \mathbf{d}$ 
49         $\lambda \leftarrow \lambda + \Delta \lambda$ 
50         $\Delta l \leftarrow^0 \Delta l (Nd/k)^{0.5}$ 
51      end
52    end
53  end
54 end

```

The input data in Algorithms 3 and 4 are: initial Arc-Length ( $^0 \Delta l$ ); maximum number of iterations in each load step ( $k_{max}$ ); number of desired iterations in each

load step ( $Nd$ ); tolerance ( $tol$ ); load incremental ( $\Delta P$ ); maximum number of load steps ( $LS_{max}$ ); and constant  $\eta$ .

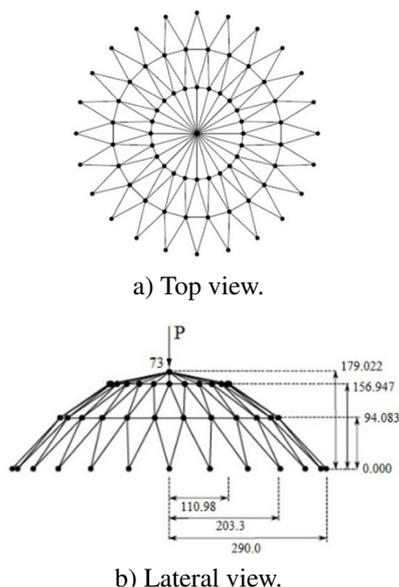
The algorithm outputs are: nodal coordinate vector ( $\mathbf{d}$ ); total load parameter ( $\lambda$ ); total number of load steps ( $LS$ ); total number of accumulated iterations until the convergence to the solution ( $k_{total}$ ); and nodal displacement vector ( $\mathbf{u}$ ).

The spatial trusses are discretized through the Finite Element Positional Method. Originally developed by Coda (2003), this formulation is an alternative approach to nonlinear problems, and considers nodal positions as nonlinear system variables instead of the displacements (CODA; GRECO, 2004). The implemented algorithms are computationally evaluated according to the following parameters: processing time in seconds ( $t$ ); total number of load steps ( $LS$ ) and accumulated iterations ( $k_{total}$ ); and average number of iterations per load step ( $k_{av}$ ).

### Circular dome with 168 bars and 73 nodes

Figures 1a) and 1b) shows a circular dome whose bars have the dimensionless axial stiffness  $EA = 1.0 \times 10^4$ . At the base of the truss there are pinned type supports and at the top of the truss a vertical load of intensity  $P$  is applied. It has 73 nodes, 168 bars and 219 nodal coordinates. The parameters considered in the simulations are: initial Arc-Length  $^0\Delta l = 6.0$ ; maximum number of iterations in each cycle  $k_{max} = 100$ ; number of desired iterations  $Nd = 3$ ; tolerance  $tol = 1.0 \times 10^{-10}$ ; load incremental  $\Delta P = 100$ ; and  $\eta = 1000$ . The numerical results  $LS$ ,  $k_{total}$ ,  $k_{av}$  and  $t$  are shown in Table 3 for simulations with the Newton-Raphson, Hybrid Newton-Raphson, Potra-Pták and Hybrid Potra-Pták solution methods.

**Figure 1** – Circular dome.



**Source:** Adapted from Papadrakakis (1981).

**Table 3** – Numerical results for the circular dome, number of unknowns: 219 nodal coordinates and force parameter.

Solution method	$LS$	$k_{total}$	$k_{av}$	$t(s)$
Newton-Raphson	16	56	3.50	0.75
Hybrid Newton-Raphson (Algorithm 3)	16	56	3.50	0.69
Potra-Pták	14	36	2.57	0.61
Hybrid Potra-Pták (Algorithm 4)	14	37	2.64	0.54

**Source:** The authors.

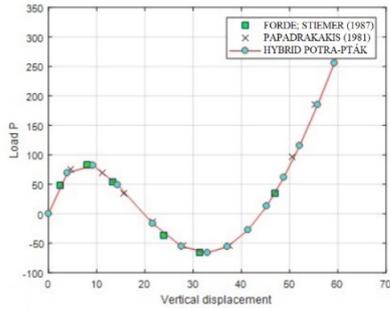
The graph relating the vertical displacement and the load  $P$  at the top of the truss, equilibrium path, is shown in Figure 2a). This curve represents the nonlinear response of the structure. Each point on the path, marker, represents a static equilibrium configuration of the structure. This figure shows the presence of two load limit points in the path, characterized by horizontal tangents, parallel to the axis of displacements. The first limit point represents the maximum load  $P$  that the structure can support before snap-through. At intermediate positions, the load  $P$  decreases and reverses signal twice while the displacement continues to increase. When the snap-through effect ends at the second point, the structure begins to support additional load values. Good agreement is observed between the results obtained with the hybrid Potra-Pták method (FORDE; STIEMER, 1987; PAPADRAKAKIS, 1981).

Figure 2b) shows the distribution map of the non-null elements of the stiffness matrix  $\mathbf{K}$ , obtained with the Matlab® spy function. It is observed that this matrix is sparse, whose non-null elements appear on the main diagonal and in parallel diagonals, with sparse degree 94.43 %, number of null elements/total number of elements.

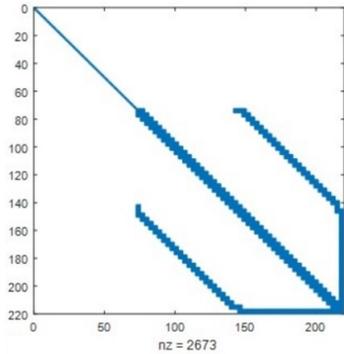
Large matrices require large storage space, and even though there are computers with the largest memory capacity today, it is usually not enough to store the square matrix. To make matrix storage and operations less expensive, you can use techniques based on storing non-zero values, such as CSC, Compressed Sparse Row, and CSR, Compressed Sparse Row (ANTUNES FILHO; XAVIER, 2013). In the program developed in Matlab® environment the sparse function was used, which stores the non-null elements of the original matrix, disregarding the elements equal to zero.

Figure 3 shows the undeformed (before loading) and deformed dome configurations, corresponding to  $LS = 14$ , pointing to the bars that are tractioned (black color) and compressed (red color), obtained with the method Hybrid Potra-Pták.

**Figure 2** – Circular dome.



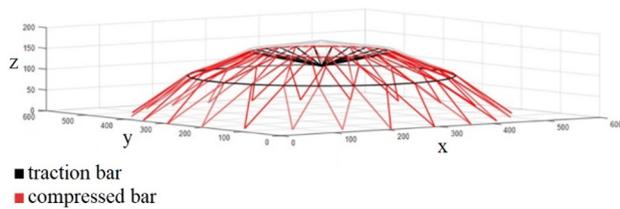
a) Equilibrium path.



b) Distribution map of the non-null elements of the stiffness matrix **K**, where  $nz$  is the total number of non-null elements.

Source: The authors.

**Figure 3** – Undeformed and deformed circular dome configurations, with the indication of the tractioned (black color) and compressed (red color) bars.

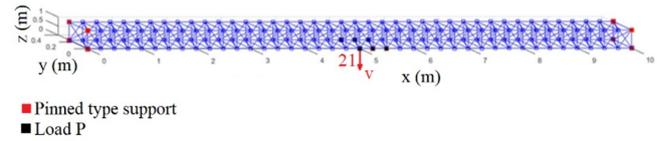


Source: The authors.

**Space trussed beam with 566 bars and 164 nodes**

Let the space trussed beam shown in Figure 4 be 10.0 m long, 1.0 m high and 0.25 m deep. For all bars the cross-sectional area  $0.01 \text{ m}^2$  and modulus of elasticity  $E = 205.0 \text{ GPa}$  were considered. The end nodes are restricted in  $x$ ,  $y$  and  $z$  directions. The truss is requested by loads of intensity  $P$  at the indicated nodes, as shown in Figure 4. The structure has 164 nodes, 566 bars and 492 nodal coordinates. The parameters considered in the simulations are:  ${}^0\Delta l = 0.5$ ;  $k_{max} = 100$ ;  $Nd = 3$ ;  $tol = 1.0 \times 10^{-6}$ ;  $\Delta P = 100 \text{ N}$ ; and  $\eta = 10$ . The numerical results  $LS$ ,  $k_{total}$ ,  $k_{av}$  and  $t$  are shown in Table 4.

**Figure 4** – Structural scheme of the space trussed beam.



Source: The authors.

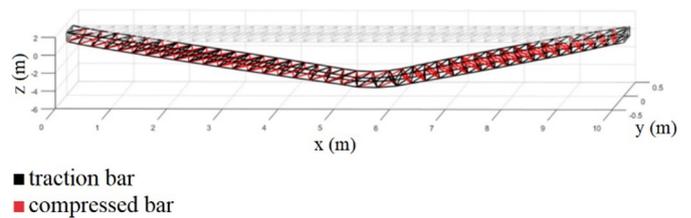
**Table 4** – Numerical results for the trussed beam, number of unknowns: 492 nodal coordinates and one force parameter.

Solution method	LS	$k_{total}$	$k_{av}$	$t(s)$
Newton-Raphson	81	280	3.45	17.73
Hybrid Newton-Raphson (Algorithm 3)	81	280	3.45	17.49
Potra-Pták	73	218	2.98	15.74
Hybrid Potra-Pták (Algorithm 4)	72	195	2.70	13.76

Source: The authors.

Figure 5 shows the undeformed and deformed trussed beam configurations obtained in the last equilibrium configuration,  $LS = 72$ , with the hybrid Potra-Pták method. The graph relating the vertical displacement at node 21 and the force  $P$  is shown in Figure 6a). Figure 6b) shows the distribution map of the non-null elements of the stiffness matrix **K**, with a sparsity degree of 95.41 %.

**Figure 5** – Structural scheme of the space trussed beam.

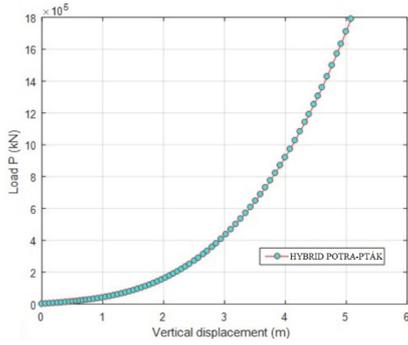


Source: The authors.

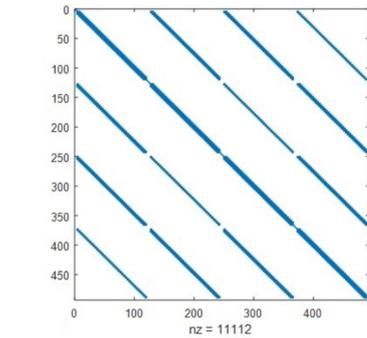
**Space circular truss with 210 bars and 90 nodes**

Figure 7 shows a circular spatial truss with 90 nodes, 210 elements and 270 nodal coordinates. At the top of the structure, height  $z = 4.4641 \text{ m}$ , vertical loads are applied to nodes of intensity  $0.1 P$ , negative  $z$  direction; and at dimension  $z = 3.4641 \text{ m}$ , vertical loads  $P$ , negative  $z$  direction, and horizontal loads  $0.1 P$ , negative  $y$  direction, are applied to the indicated nodes. The base nodes, dimension  $z = 0 \text{ m}$ , are constrained in the  $x$ ,  $y$  and  $z$  directions, pinned type support. The modulus of elasticity  $E$  and the cross-sectional area  $A$  of the bars are  $210 \times 10^6 \text{ kPa}$  and  $0.1 \text{ m}^2$ , respectively. In the simulations the parameters are considered:  ${}^0\Delta l = 7.0 \times 10^{-4}$ ;  $K_{max} = 150$ ;  $Nd = 4$ ;  $tol = 1.0 \times 10^{-7}$ ;  $\Delta P = 1.0 \text{ kN}$ ; and  $\eta = 10$ .

Figure 6 – Trussed beam.



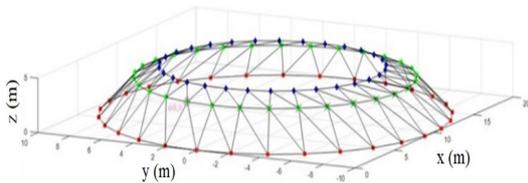
a) Equilibrium path



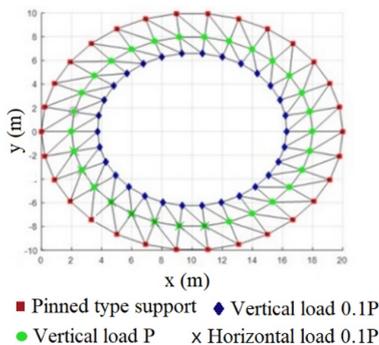
b) Distribution map of non-null elements of the stiffness matrix  $\mathbf{K}$ , where  $nz$  is the total number of non-null elements.

Source: The authors.

Figure 7 – Structural scheme of the circular spatial truss.



a) 3D view.



b) Plan view.

Source: The authors.

Table 5 presents the numerical results  $LS$ ,  $k_{total}$ ,  $k_{av}$  and  $t$ , for simulations with the implemented solution methods.

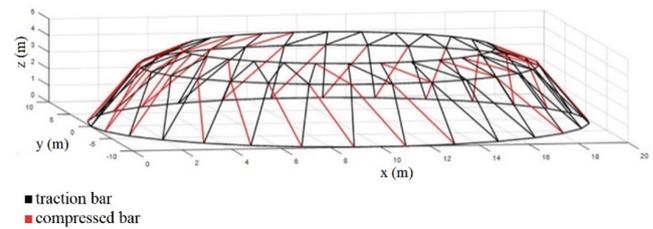
Table 5 – Numerical results for the circular truss, number of unknowns: 270 nodal coordinates and one force parameter.

Solution method	$LS$	$k_{total}$	$k_{av}$	$t(s)$
Newton-Raphson	150	603	4.02	6.22
Hybrid Newton-Raphson (Algorithm 3)	150	603	4.02	5.77
Potra-Pták	127	375	2.95	4.97
Hybrid Potra-Pták (Algorithm 4)	127	373	2.93	4.76

Source: The authors.

Figure 8 shows the deformed spatial circular truss referring to the last equilibrium configuration,  $LS = 127$ , with the indication of the tractioned and compressed bars.

Figure 8 – Deformed configuration of the circular truss, indicating the traction bars (black color) and compressed bars (red color).



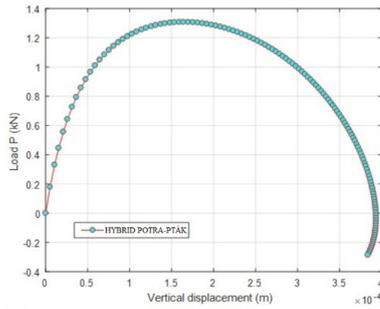
Source: The authors.

The equilibrium path, vertical displacement curve at node 60 versus load  $P$ , is shown in Figure 9a); the distribution map of the non-zero elements of the stiffness matrix  $\mathbf{K}$  is illustrated in Figure 9b).

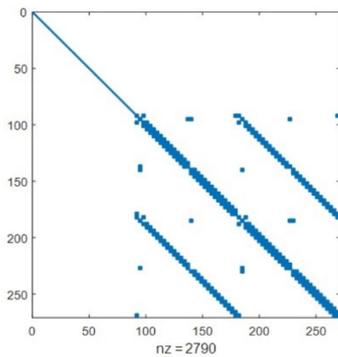
Figure 9a) shows the presence of two limit points in the path. The first point is a load limit point whose tangent is parallel to the axis of displacements; the second is a limit point of displacement, and is characterized by a tangent parallel to the axis of loads. An equilibrium configuration can be stable or unstable. Stable configurations occur with increasing load and displacement; unstable configurations may occur with decreasing load and increasing displacement, or decreasing load and displacement.

The constant curves  $\eta$  versus time  $t$  and  $\eta$  versus total number of iterations per load step are shown in Figures 10a) and 10b), respectively, obtained with the hybrid Potra-Pták method, Algorithm 4. It is observed that by varying the value of  $\eta$  the processing time also changes; the best result achieved is for  $\eta = 40$ , whose processing time  $t$  is 4.40 s.

Figure 9 – Circular truss.



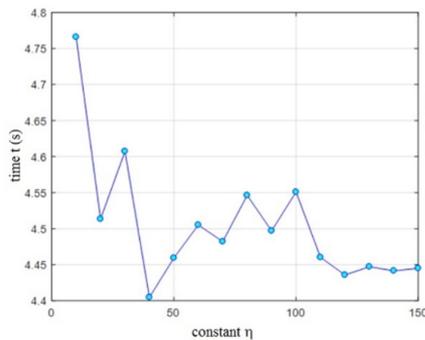
a) Equilibrium path.



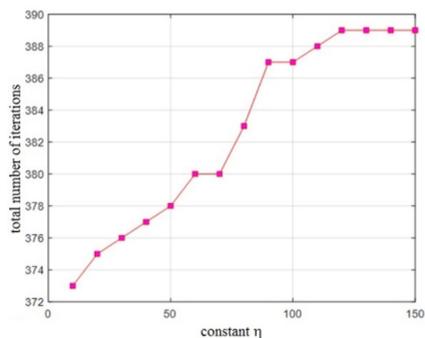
b) Distribution map of non-null elements of the stiffness matrix  $K$ .

Source: The authors.

Figure 10 – Simulations of the circular spatial truss with Algorithm 4.



a) Curve  $\eta$  versus time  $t$ .



b) Curve  $\eta$  versus total number of iterations per load step.

Source: The authors.

## Conclusion

A modification of the Newton-Raphson and Potra-Pták algorithms was proposed by inserting a condition, associating them with the modified Newton-Raphson method. The presented numerical examples demonstrated computational efficiency and implemented algorithms precision, obtaining the solution of the studied problems with a shorter processing time, according to Tables 2-5.

## Acknowledgments

The authors wish to thank the Graduate Program in Mathematics of the State University of Maringá and the Federal Technological University of Paraná for their support in the development of this research.

## References

AL-TOWAIQ, M. H.; ABU HOUR, Y. S. Two improved classes of Broyden's methods for solving nonlinear systems of equations. *Journal of Mathematics and Computer Science - JMCS*, Stockholm, v. 17, n. 1, p. 22-31, 2017. DOI: 10.22436/jmcs.017.01.02

ANTUNES FILHO, A.; XAVIER, A. C. C. Solução de sistemas lineares esparsos utilizando CUDA: uma comparação de desempenho em sistemas Windows e Linux. *Revista de Ciências Exatas e Tecnologia*, Valinhos, v. 8, n. 8, p. 181-195, 2013. DOI: <https://doi.org/10.17921/1890-1793.2013v8n8p181-195>

BATHE, K. J. *Finite element procedures*. New Jersey: Prentice-Hall, Inc., 2006.

CODA, H. B. *Análise não linear geométrica de sólidos e estruturas: uma formulação posicional baseada no MEF*. 2003. Tese (Doutorado) - Escola de Engenharia de São Carlos, Universidade de São Paulo, São Paulo, 2003.

CODA, H. B.; GRECO, M. A simple FEM formulation for large deflection 2D frame analysis based on position description. *Computer methods in applied mechanics and engineering*, Amsterdam, v. 193, n. 33-35, p. 3541-3557, 2004. DOI: <https://doi.org/10.1016/j.cma.2004.01.005>

CRISFIELD, M. A. *Non-linear finite element analysis of solids and structures*. New York: John Wiley & Sons, Inc., 1991. v. 1.

- DARVISHI, M. T.; SHIN, B. C. High-order Newton-Krylov methods to solve systems of nonlinear equations. *Journal of the Korean Society for Industrial and Applied Mathematics*, Seoul, v. 15, n. 1, p. 19-30, 2011.
- DEGHANI, H.; MANSOURI, I.; FARZAMPOUR, A.; HU, J. W. Improved homotopy perturbation method for geometrically nonlinear analysis of space trusses. *Applied Sciences*, Basel, v. 10, n. 8, p. 2987, 2020. DOI: 10.3390/app10082987
- FAN, J.; YUAN, Y. A regularized Newton method for monotone nonlinear equations and its application. *Optimization Methods and Software*, New York, v. 29, n. 1, p. 102-119, 2014. DOI: <https://doi.org/10.1080/10556788.2012.746344>
- FORDE, B. W. R.; STIEMER, S. F. Improved arc length orthogonality methods for nonlinear finite element analysis. *Computers & Structures*, Elmsford, v. 27, n. 5, p. 625-630, 1987. DOI: [https://doi.org/10.1016/0045-7949\(87\)90078-2](https://doi.org/10.1016/0045-7949(87)90078-2)
- FRONTINI, M.; SORMANI, E. Some variant of Newton's method with third-order convergence. *Applied Mathematics and Computation*, New York, v. 140, n. 2-3, p. 419-426, 2003.
- GRAU-SÁNCHEZ, M.; NOGUERA, M.; GUTIÉRREZ, J. M. On some computational orders of convergence. *Applied Mathematics Letters*, Elmsford, v. 23, n. 4, p. 472-478, 2010. DOI: <https://doi.org/10.1016/j.aml.2012.04.012>
- GUTIERREZ, J. M.; HERNÁNDEZ, M. A. A family of Chebyshev-Halley type methods in Banach spaces. *Bulletin of the Australian Mathematical Society*, Cambridge, v. 55, n. 1, p. 113-130, 1997.
- JARRATT, P. Some efficient fourth order multi-point methods for solving equations. *BIT Numerical Mathematics*, Dordrecht, v. 9, n. 2, p. 119-124, 1969.
- KANWAR, V.; KUMAR, S.; BEHL, R. Several new families of Jarratt's method for solving systems of nonlinear equations. *Applications and Applied Mathematics*, Texas, v. 8, n. 2, p. 701-716, 2013.
- KANWAR, V.; SHARMA, J. R. A new family of Secant-like method with super-linear convergence. *Applied Mathematics and Computation*, New York, v. 171, n. 1, p. 104-107, 2005.
- KIM, Y. L.; CHUN, C. New twelfth-order modifications of Jarratt's method for solving nonlinear equations. *Studies in Nonlinear sciences*, Pakistan, v. 1, n. 1, p. 14-18, 2010.
- LEON, S. E.; PAULINO, G. H.; PEREIRA, A.; MENEZES, I. F.; LAGES, E. N. A unified library of nonlinear solution schemes. *Applied Mechanics Reviews*, New York, v. 64, n. 4, 2011.
- MAHDAVI, S. H.; RAZAK, H. A.; SHOJAEI, S.; MAHDAVI, M. S. A comparative study on application of Chebyshev and spline methods for geometrically nonlinear analysis of truss structures. *International Journal of Mechanical Sciences*, Oxford, v. 101, p. 241-251, 2015.
- MATLAB. *Version 8.6.0 (R2015b)*. Natick, Massachusetts: The Math Works Inc., 2015.
- MAXIMIANO, D. P.; SILVA, A. R. D.; SILVEIRA, R. A. M. Iterative strategies associated with the normal flow technique on the nonlinear analysis of structural arches. *Rem: Revista Escola de Minas*, Ouro Preto, v. 67, n. 2, p. 143-150, 2014.
- MOHIT, M.; SHARIFI, Y.; TAVAKOLI, A. Geometrically nonlinear analysis of space trusses using new iterative techniques. *Asian Journal of Civil Engineering*, [s. l.], v. 21, p. 785-795, 2020.
- MUHAMMAD, K.; MAMAT, M.; WAZIRI, M. Y. A Broyden's-like Method for solving systems of Nonlinear Equations. *World Applied Sciences Journal*, Pakistan v. 21, p. 168-173, 2013.
- PAPADRAKAKIS, M. Post-buckling analysis of spatial structures by vector iteration methods. *Computers & Structures*, Elmsford, v. 14, n. 5-6, p. 393-402, 1981.
- POTRA, F. A.; PTÁK, V. Nondiscrete induction and an inversion-free modification of Newton's method. *Časopis pro Pěstování Matematiky*, Praha, v. 108, n. 4, p. 333-341, 1983.
- REZAIIEE-PAJAND, M.; SALEHI-AHMADABAD, M.; GHALISHOOYAN, M. Structural geometrical nonlinear analysis by displacement increment. *Asian Journal of Civil Engineering*, [s. l.], v. 15, n. 5, p. 633-653, 2014.
- RIKS, E. The application of Newton's method to the problem of elastic stability. *Journal of Applied Mechanics*, New York, v. 39, n. 4, p. 1060-1065, 1972.
- RIKS, E. An incremental approach to the solution of snapping and buckling problems. *International journal of solids and structures*, [s.l.], v. 15, n. 7, p. 529-551, 1979.

- SAFFARI, H.; MANSOURI, I. Non-linear analysis of structures using two-point method. *International Journal of Non-Linear Mechanics*, Elmsford, v. 46, n. 6, p. 834-840, 2011.
- SOLEYMANI, F.; SHARMA, R. L. X.; TOHIDI, E. An optimized derivative-free form of the Potra–Pták method. *Mathematical and Computer Modelling*, Oxford, v. 56, n. 5-6, p. 97-104, 2012.
- SOUZA, E. A. *Métodos iterativos para problemas não lineares*. 2015. Dissertação (Mestrado) - Universidade Federal Fluminense, Volta Redonda, 2015.
- SOUZA, L. A. F. Static nonlinear analysis of piles cap based on the Continuum Damage Mechanics. *Semina: Ciências Exatas e Tecnológicas*, Londrina, v. 36, n. 2, p. 85-94, 2015. DOI: 10.5433/1679-0375.2015v36n2p85
- SOUZA, E. A.; ALVAREZ, G. B.; LOBAO, D. C. Comparação Numérica entre Métodos Iterativos para Problemas Não Lineares. *Proceeding Series of the Brazilian Society of Computational and Applied Mathematics*, São Carlos, v. 5, n. 1, 2017. DOI: <https://doi.org/10.5540/03.2017.005.01.0316>
- SOUZA, L. A. F.; CASTELANI, E. V.; SHIRABAYASHI, W. V. I.; MACHADO, R. D. Métodos iterativos de terceira e quarta ordem associados à técnica de comprimento de arco linear. *Ciência & Engenharia*, Uberlândia, v. 26, n. 1, p. 39-49, 2017.
- SOUZA, L. A. F.; CASTELANI, E. V.; SHIRABAYASHI, W. V. I.; ALIANO FILHO, A.; MACHADO, R. D. Trusses Nonlinear Problems Solution with Numerical Methods of Cubic Convergence Order. *TEMA*, São Carlos, v. 19, n. 1, p. 161-179, 2018. DOI: <https://doi.org/10.5540/tema.2018.019.01.0161>
- SOUZA, L. A. F.; MARTINS, R. S. V.; XAVIER, J. C.; PORTO, J. H. Application and comparison of numerical methods in the solution of systems of linear equations in space trusses problems. *Semina: Ciências Exatas e Tecnológicas*, Londrina, v. 39, n. 1, p. 49-60, 2018. DOI: 10.5433/1679-0375.2018v39n1p49
- WEERAKOON, S.; FERNANDO, T. G. I. A variant of Newton's method with accelerated third-order convergence. *Applied Mathematics Letters*, Elmsford, v. 13, n. 8, p. 87-93, 2000.

Received: Apr 9, 2021  
Accepted: May 7, 2021  
Published: June 3, 2021