Simulation of structural mechanics problems using “physics engines”

João Emanuel Marques de Matos

Instituto Superior Técnico

October 2017

Abstract

As one of the bases for design in Civil Engineering, structural analysis uses computational programs to support most calculations. However, these programs are, mostly, developed for the professionals and its developers are not, therefore, particularly interested on creating separate products that may be more user-friendly for students. This dissertation focuses its work on the development of a didactic computational tool for the study of structural analysis concepts.

In this project, combining physical equations with video games concepts (physic engine), a graphic simulator was developed, implementing a Finite Particle Method (MPF), based in the laws of Newton and kinematics. By only evaluating the behavior of individual, but interconnected, masses (particle system), this method can simulate structures without checking their global equilibrium, as this is done locally in the particles. It is then possible to describe large particle displacements over time and offer real-time simulations of simple structures with physical and geometrically linear or non-linear behaviors.

This program evaluates physical and geometrically linear behaviors, as well as initiate the study of physically non-linear behaviors, in reticulated structures. The users can improve their understanding of concepts learned in various subjects of the Civil Engineering course.

Being a project in progress, there are still possible improvements, as well as necessary ones, because both the accuracy of results and the computational performance are still open to improvement. However, this program already presents results that could be a useful tool in the study/exposition of structural analysis concepts, with which, both, students and professors could benefit, academically.

Key-words: computational didactic tool; real time simulation; structural mechanics; Finite Particle Method; linear analysis; non-linear analysis.

1. Introduction

Structural Mechanics, as being a subject that study stress and strain distributions in structures under mechanic actions, could be called, in a straightforward way, one of the foundations of structural analysis.

In civil engineering, it is of most importance to develop good bases of knowledge in these subjects, because a decent understanding of structures behaviors is fundamental to perform solid and successful projects.

With the actual high-performance computers, civil engineering has benefited from a range of computational tools in structural analysis. However, the focus of these tools is mainly to the professional scope as they prime to achieve accurate and fast results, of complex problems, for a public that has already an understanding of the used concepts.

Most engineering courses are still taught in the traditional manner based on books and notes where the focus is in conveying theoretical concepts, being the use of computational tools encouraged only on more advance stages of the Civil Engineering course.

Without diminishing traditional methods, as they are much needed, with the current technological resources is possible, and beneficial, to conciliate traditional teaching methods with didactic computational tools in which students could complement physic concepts with graphic simulations, visualizing structure movements along time.

With an enormous growth over the last decades, the video game industry has combined physics with graphic and interactive concepts as they try to replicate reality in objects in motion (cars, explosions, characters, bullets, etc). Knowing this, structural mechanics could largely benefit from video games to create computational tools using the concepts of physic and rendering engines.

Aiming the creation of a computational didactic tool for teaching structural mechanics, the work now being presented is based on a previous work, (Lopes 2015), where the focus was on creating a real time simulator applied to analysis of 2D trusses.

This dissertation (Matos 2017) aims to continue that work by extending it to the static and dynamic analysis of frame structures, that is, structures with flexional structural components, and by developing as well a more user-friendly, and interactive, graphic interface.

2. Computational program

Based on what was referred earlier, a real-time simulator was built based on three major modules (engines) that solve the physic equations, update, over time, the information of objects movements and check for changes in both physic simulation output and user input. These engines are: a physics engine, a rendering engine and an event handler.

The physics engine is responsible for solving the physic equations and, in this program, is based on the Finite Particle Method (an iterative process with time increments), as described ahead. As more precise are the equations, and smaller the time step, more accurate are the results (Lopes 2015).

During the simulation, there are changes that need to be acknowledged. Those changes could be user input, at any instant, or value changes, from the physical phenomena in study, over the iterative process. Responsible for this task is the event handler.

To obtain real-time graphic visualization, a Graphical User Interface – GUI is needed. The rendering engine is the one that provides it by gathering information from the physics engine and event handler and displaying it on the program window.

Within a main loop, the program alternates between 'game modes' as requested by the user. In each mode, both the event handler and rendering engine are present, being in the simulation mode also the physic engine, because is the only mode that uses physic calculations.

3. Finite Particle Method (FPM)

The study and understanding of materials and structure behaviors has been, since early days, one of the purposes of engineering. In civil engineering, there are a few methods used to study structural mechanics problems. The most used are various instances of the Finite Element Method – a set of robust technics in the analysis of linear or non-linear, static and dynamic problems. However, there may be, in some cases, some difficulties in the creation of meshed especially in non-linear problems where mesh discontinuities may arise.

Over the years, alternative techniques, have been developed whereby structures are defined by systems of particles and links connecting them in a way that does not lead to mesh discontinuities (because there are no meshes, at least not in the traditional sense of the word when dealing with finite elements. These particle methods have presented good results in non-linear problems, being already used to study fluid dynamics, soil mechanics, heat conduction, etc (Asprone, Auricchio et al. 2014).

More recently, these particle methods have been simplified and used in the analysis of structural elements under large displacements. Its simplicity lays on the facts that only particles are evaluated and movements are described by simple laws of motion.

This dissertation (Matos 2017) uses a Finite Particle Method, based on the VFIFE method (vector form intrinsic finite element method (Ting, Shih et al. 2004)), to evaluate the behavior of structures subject to static and dynamic loads.

3.1. Discretization of structure

The used method considers a structure as a particle system – set of masses (particles) connected by spring elements (rods like elements but also beams).

Each particle is defined by a position vector $\vec{X}^T(t_N)=[x(t_N) \quad y(t_N) \quad \theta(t_N)]$, a mass value, m_β, and a mass moment of inertia, j_β, depending on the number, and properties, of connected rods/links.

$$m_\beta = M_\beta + \sum_{k=1}^{nc} \frac{1}{2} \lambda_k l_k \tag{1}$$

$$j_\beta = J_\beta + \sum_{k=1}^{nc} \frac{m_{k\beta} l_k^2}{12} \tag{2}$$

where λ_k and l_k are, respectively, the mass per unit length and length of rod k and i denotes the radius of gyration of the cross-sections of the rod k (Wu, Tsai et al. 2009).

The particles, by moving, create deformations on the rods and those deformations will generate internal stresses that will be applied back, as equivalent internal forces, on the particles. Along with external forces, and using Newton's second law of motion, the particle equilibrium is verified and the particle acceleration is estimated. Next, using an explicit time integration, new particle velocities and positions are calculated. Those new positions apply new deformations to the rods and the process goes on, iteratively. One feature of this method is that it is not necessary to verify the global equilibrium of the structure, equilibrium is verified locally in each particle.

3.2. Particle motion equations

Being an iterative method, particle movements, over time, must be discrete. For that, the particles movements are described as a set of discrete paths over time segments. In each time segment, $t_a \leq t \leq t_b$, the rods deformations are evaluated based on its geometry at the initial instant of the time segment, t_a, and, with that, all previous movements can be "forgotten" because they have already been considered.

Deformation components

To evaluate the rods deformations, a deformation vector $\vec{u} = \{du \quad dv\}$, representing the relative deformation in both x and y axes, is considered (Figure 1).

To simplify the calculations (by decreasing the number of independent displacements) as well as to reduce possible errors, it is admitted that part of the displacements are rigid body displacements. Because rigid body movements do not create deformations it is possible to eliminate them.

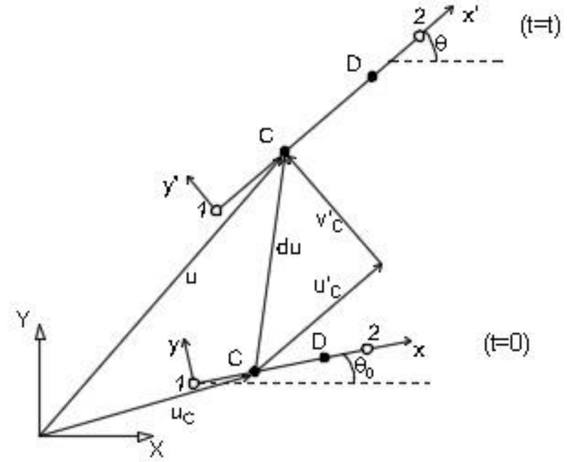

Figure 1:Deformation coordinates (adapted from (Ting, Shih et al. 2004)).

Being θ_0 the rod angle at $t = t_0$, $\theta = \theta_0 + \Delta\theta$ the rod angle at $t = t$ and assuming the displacement of particle 1 as rigid body displacements, the deformation vector of particle 2 can be written as,

$$d\vec{u}_2 = \vec{u}_2 - \vec{u}_1 \qquad (3)$$

If we assume $\Delta\theta$ as a rigid body rotation, (du_2^r), the deformation of the rod is given by the following expression (Figure 2),

$$d\vec{u}_2^d = d\vec{u}_2 - d\vec{u}_2^r \qquad (4)$$

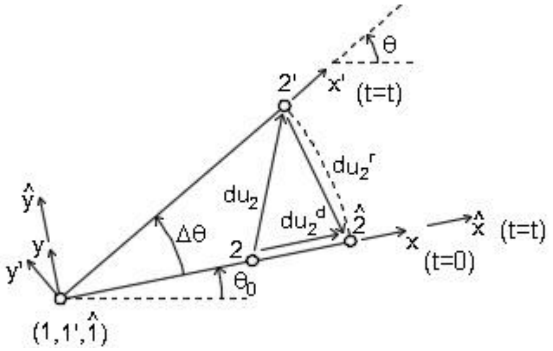

Figure 2: Relative deformation of particles 1 and 2 after removal of rigid body displacements (adapted from (Ting, Shih et al. 2004)).

A simplification was adopted so that the curvature of the rod is not accounted for (Ting, Shih et al. 2004), being possible to assume the rod as always straight. However, this does not mean that the rod has no rotational deformations, as will be presented ahead.

Although the time segment has been referred, in this section, to the initial instant of time $t = t_0$, it is more beneficial to refer it to the initial instant of the actual time step. From now on, it is where it will be referred to (Figure 3).

Applying a time increment, $(t' = t + \Delta t)$, and knowing the initial and final positions of the particles, in this time segment, it is possible to calculate, both, the initial and the final rod angles. Eliminating the rigid body displacements, as described before, it is possible to obtain the incremental deformation of the rod (Figure 3).

$$\underline{R} = \begin{bmatrix} \cos(\Delta\theta) & sen(\Delta\theta) \\ -sen(\Delta\theta) & \cos(\Delta\theta) \end{bmatrix}$$

$$\underline{Q} = \begin{bmatrix} \cos(\theta) & sen(\theta) \\ -sen(\theta) & \cos(\theta) \end{bmatrix}$$

$$\begin{Bmatrix} x'' \\ 0 \end{Bmatrix} = \underline{Q}'' \begin{Bmatrix} x - x_1 \\ y - y_1 \end{Bmatrix}, \qquad \underline{Q}'' = RQ$$

$$d\hat{u}' = \begin{Bmatrix} \hat{u}' \\ \hat{v}' \end{Bmatrix} = \underline{Q} \begin{Bmatrix} u - u_1 \\ v - v_1 \end{Bmatrix}$$

$$d\hat{u}'^r = \begin{Bmatrix} \hat{u}'^r \\ \hat{v}'^r \end{Bmatrix} = (\underline{R}^T - \underline{I}) \begin{Bmatrix} x'' \\ 0 \end{Bmatrix} \qquad (5)$$

$$d\hat{u}'^d = \begin{Bmatrix} \hat{u}'^d \\ \hat{v}'^d \end{Bmatrix} = d\hat{u}' - d\hat{u}'^r = \begin{Bmatrix} \hat{u}' \\ \hat{v}' \end{Bmatrix} - \begin{Bmatrix} \hat{u}'^r \\ \hat{v}'^r \end{Bmatrix} \qquad (6)$$

where $d\hat{u}'$ is the displacement based on particle 1; $d\hat{u}'^r$ is the rigid body displacement; $\underline{R}$, $\underline{Q}''$ and $\underline{Q}$ are rotational matrixes and $\underline{I}$ is the identity matrix.

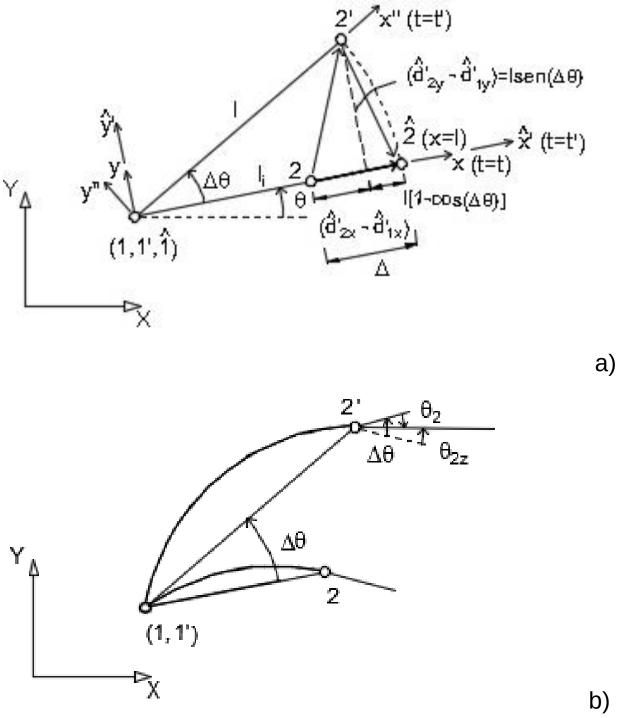

Figure 3: Detailed representation of relative deformations of particles 1 and 2 (adapted from (Ting, Shih et al. 2004)). a) Axial deformation. b) Rotational deformation.

As presented in Figure 3, the relative displacements are given by,

$$\hat{d}'^d_{1x} = \hat{d}'^d_{1y} = 0$$

$$\hat{d}'^d_{2y} = \hat{d}'_{2y} - \hat{d}'^r_{2y} = \hat{d}'_{2y} - [\hat{d}'_{1y} + l sen(\Delta\theta)] = 0 \qquad (7)$$

$$\hat{d}'^d_{2x} = \hat{d}'_{2x} - \hat{d}'^r_{2x} = \hat{d}'_{2x} - (\hat{d}'_{1x} - l[1 - cos(\Delta\theta)])$$

In equations (7), it is possible to see that only three displacements are independent: $\hat{d}'_{1x} = \hat{d}'^d_{1x}$, $\hat{d}'_{1y} = \hat{d}'^d_{1y}$ and $\Delta\theta$. Although those displacements were already reduced from six to three independent displacements, what we want is independent deformation. To obtain that we start to admit our displacement vector as,

$$(\overrightarrow{\hat{d}'}_e)^T = \{\hat{d}'_{1x} \quad \hat{d}'_{1y} \quad \theta_{1z} \quad \hat{d}'_{2x} \quad \hat{d}'_{2x} \quad \theta_{2z}\}$$

As we have been doing along this section, we separate the rigid body displacements, $\overrightarrow{\hat{d}'}_e^{\,r}$, from the relative ones, $\overrightarrow{\hat{d}'}_e^{\,d}$.

$$\overrightarrow{\hat{d}'}_e = \overrightarrow{\hat{d}'}_e^{\,r} + \overrightarrow{\hat{d}'}_e^{\,d} \tag{8}$$

$$(\overrightarrow{\hat{d}'}_e^{\,r})^T = \{\hat{d}'_{1x} \quad \hat{d}'_{1y} \quad \Delta\theta \quad \hat{d}'^{r}_{2x} \quad \hat{d}'^{r}_{2y} \quad \Delta\theta\} \tag{9}$$

$$(\overrightarrow{\hat{d}'}_e^{\,d})^T = \{0 \quad 0 \quad \theta_1 \quad \Delta \quad 0 \quad \theta_2\} \tag{10}$$

Using equations (7) – (10), we obtain the expressions of the relative displacement, corresponding to the independent deformations, and respective vector $\overrightarrow{\hat{d}_e^*}$.

$$\Delta = \hat{d}'_{2x} - (\hat{d}'_{1x} - l[1 - cos(\Delta\theta)])$$

$$\theta_1 = \theta_{1z} - \Delta\theta \; ; \qquad \theta_2 = \theta_{2z} - \Delta\theta$$

$$\overrightarrow{\hat{d}_e^*} = \begin{Bmatrix} \Delta \\ \theta_1 \\ \theta_2 \end{Bmatrix} \tag{11}$$

Internal stresses

Associated with the rod deformations, presented in equation (11), are rod internal stresses. To obtain those, shape functions are used, to evaluate stress and strain, and, by a virtual work formulation, we calculate three independent internal stresses, $\vec{f}_e^*$, associated to each of the independent deformations.

$$\vec{f}_e^* = \begin{bmatrix} EA/l & 0 & 0 \\ 0 & 4EI/l & 2EI/l \\ 0 & 2EI/l & 4EI/l \end{bmatrix} \begin{Bmatrix} \Delta \\ \theta_1 \\ \theta_2 \end{Bmatrix} \tag{12}$$

Admitting one internal stress to each one of both end particle displacements, there are still three internal stresses left. By force equilibrium on the rod, the remaining internal stresses are calculated in the rod local axis.

$$\sum F_x = 0 \qquad f_{1x} = -f_{2x}$$

$$\sum M_1 = 0 \qquad f_{2y} = -(m_{1z} + m_{2z})/l \tag{13}$$

$$\sum F_y = 0 \qquad f_{1y} = -f_{2y}$$

$$\vec{f}_e^{int} = \begin{Bmatrix} f_{1x} \\ f_{1y} \\ m_{1z} \\ f_{2x} \\ f_{2y} \\ m_{2z} \end{Bmatrix} = \begin{Bmatrix} -EA\Delta/l \\ (6EI/l^2)(\theta_1 + \theta_2) \\ (2EI/l)(2\theta_1 + \theta_2) \\ EA\Delta/l \\ -(6EI/l^2)(\theta_1 + \theta_2) \\ (2EI/l)(\theta_1 + 2\theta_2) \end{Bmatrix} \tag{14}$$

Being the internal stresses calculated in the rod local axis, it would be necessary a coordinate change to apply equivalent forces on the particles, which are in the global axis. However, in this program, we used polar coordinates and because of that a coordinate change it is not necessary.

Particle movement

Knowing the internal stresses and applying them, as equivalent forces, on the particles, we have all the information to verify the local equilibrium on the particle. Using Newton's second law of motion, the particle accelerations are calculated as follows in matrix form,

$$\begin{bmatrix} m & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & j \end{bmatrix} \begin{Bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{\theta} \end{Bmatrix} = \begin{Bmatrix} P_x \\ P_y \\ Q_z \end{Bmatrix} + \sum_{i=1}^{nc} \begin{Bmatrix} p_{ix} \\ p_{iy} \\ q_{iz} \end{Bmatrix} - \sum_{i=1}^{nc} \begin{Bmatrix} f_{ix} \\ f_{iy} \\ m_{iz} \end{Bmatrix} \tag{15}$$

where m and j are, respectively, the particle mass and (polar) mass moment of inertia; $\vec{\ddot{X}}$ is the acceleration vector; $\vec{P}$ and $\vec{p}$ are, respectively, the exterior forces applied on particles and rods; $\vec{Q}$ and $\vec{q}$ are, respectively, the exterior moments applied on particles and rods; $\vec{f}$ and $\vec{m}$ are, respectively, the internal forces and moment and nc are the number of rods connected to the particle.

Although equation (15) is presented in cartesian coordinates, the use of polar coordinates compiles the x and y components into one, simplifying the calculations.

3.3. Explicit time integration

Along the last subchapter, it was presented the methodology to obtain rod deformations, rod internal stresses and particle acceleration, all in the same time segment. However, being an iterative method, it is necessary to reflect the time increment between iterations.

There are some ways to do it, like a simple central difference, a fourth order Runge-Kutta approximation or a second order Taylor series approximation. In this work, the last one was adopted.

Being $\vec{X}_N$ the position vector of a particle possible displacements $(x, y \; e \; \theta)$, the acceleration can be approximated by,

$$\vec{\ddot{X}}_N = \frac{2}{\Delta t^2}(\vec{X}_{N+1} - \vec{X}_N - \vec{\dot{X}}_N \Delta t) \tag{16}$$

Applying equation (16) in equation (15), the particle position and velocity for the next iteration are obtained as follows,

$$\vec{X}_{N+1} = \vec{X}_N + \vec{\dot{X}}_N \Delta t + \frac{\Delta t^2 (F^{ext} - F^{int})}{2\underline{M}} \tag{17}$$

$$\vec{\dot{X}}_{N+1} = \vec{\dot{X}}_N + \vec{\ddot{X}}_N \Delta t = \vec{\dot{X}}_N + \frac{(F^{ext} - F^{int})}{\underline{M}} \Delta t \tag{18}$$

3.4. Dynamic relaxation

A discrete method used to evaluate structures under large displacements has been presented, however, nothing has been said about the convergence of the solutions. One of the main goals of this dissertation (Matos 2017) is to obtain a static solution, corresponding to the elastic solution of the problem. To achieve that, the concept of dynamic relaxation was used.

Dynamic relaxation takes advantage of the damping of dynamic oscillations as the balance between internal and external forces is obtained. The damping can be of two types: viscous or kinetic. Although both types have their benefits, to achieve a static solution in this program (Liew, Mele et al. 2016) kinetic damping was adopted.

The equation of a damped oscillator is presented below,

$$\underline{M}\ddot{\vec{X}}_N + \underline{C}\dot{\vec{X}}_N + \underline{K}\vec{X}_N = P, \qquad em\ t = t_N \tag{19}$$

Using a simple central difference and, accounting that the kinetic damping does not need damping coefficients, as it uses only kinetic energy values, the velocity can be expressed as,

$$\dot{\vec{X}}_{N+{}^1/_2} = \dot{\vec{X}}_{N-{}^1/_2} + \frac{\Delta t}{\underline{M}}(P - \underline{K}\vec{X}) \tag{20}$$

The kinetic energy is given by,

$$\vec{U}_N = \frac{1}{2}\underline{M}\dot{\vec{X}}^2 \tag{21}$$

In each iteration, the kinetic energy is calculated and, by comparing it to the previous iteration value, if a decrease is noted it is assumed that, in the previous time segment, an energy peak has occurred and the velocity is set to zero, simulating the energy dissipation.

So, towards a decrease of kinetic energy, ($\vec{U}_{N\text{-}1}<\vec{U}_N>\vec{U}_{N+1}$), as we are in the instant $t = t_{N+1}$, it means that a kinetic energy peak occurred between $t{=}t_{N\text{-}1}$ and $t = t_N$. To simplify, we assume its occurrence in $t{=}t_{N\text{-}{}^1/_2}$.

To avoid divergences, the particle position is restarted to the peak instant (Alamatian 2012), its expression being given by (Rezaiee-Pajanda and Rezaeea 2014),

$$\vec{X}_{N-{}^1/_2} = \vec{X}_{N+1} - \frac{3}{2}\Delta t\dot{\vec{X}}_{N+{}^1/_2} + \frac{\Delta t^2}{2}\frac{P - \underline{K}\vec{X}}{\underline{M}} \tag{22}$$

The velocity is given by the following expression,

$$\dot{\vec{X}}_{N+{}^1/_2} = \frac{1}{2}\frac{\Delta t}{\underline{M}}(P - \underline{K}\vec{X}) \tag{23}$$

3.5. Computational procedure

A summarized version of the computational procedure, of the implemented FPM, is presented as follows:

1. Initial information at t=0, entered by the user: initial displacements, external forces or support conditions.
2. Calculate residual forces, by particles equilibrium.
3. Calculate particles accelerations, using Newton's second law of motion.
4. Calculate new particle positions and velocities, by explicit time integration.
5. Calculate kinetic energy.
6. Calculate rods deformation.
7. Calculate nodal internal forces.
8. Update external information.
9. If the simulation has not been stopped, return to step 2.

4. Numerical examples

In the previous section, a finite particle method was presented. To extend the created program functionalities and applications, the presented method was applied to three types of problems: static solutions, incremental elasto-plastic analysis and oscillatory dynamic behaviors.

To validate the results, as well as to check its accuracy, the method was applied to different types of structures. For the sake of brevity only one case of each type of structure is presented here.

4.1. Static solution

Figure 4 shows a frame structure under a constant horizontal force. Both internal stresses and displacements obtained from the program were compared with the exact solution.

Using kinetic damping (section 3.4) to obtain static solution, it was possible to get satisfactory results in both internal stresses and displacement, taking into account the purpose of the program.

Being a didactic tool where the user uses the graphic visualization as a convergence criterion, and possibly does not know in advance the solution (in case of more inexperienced users), errors could be a little higher than what is usually required for computation tools (around 2 or 3%). This happens because, with this method, as the simulation gets close of the exact solution, the values increments become smaller and take longer to achieve, for example, a small centesimal increment. With this, and being the graphic values rounded to the first decimal place, the user realizes the convergence, graphically, a little "early".

However, there is not any criteria that forces the user to stop the simulation, it is the user that decides when to stop it using the 'pause' button. Because of that "freedom", it is possible to get as close as the user wants of the exact solution (never achieving it off course as this is an approximate method), it just takes more simulation time.

That said, the user can choose to use the program to obtain results with very small errors (with longer simulations), or to get acceptable approximations, just to understand the structure behavior, for example.

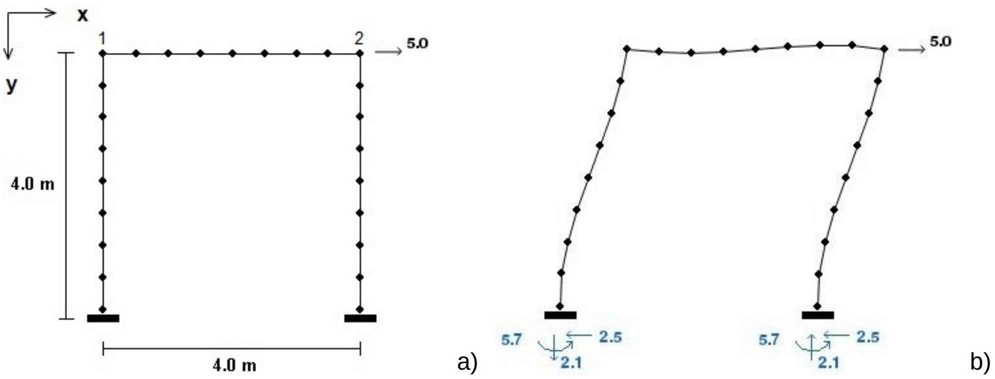

Figure 4: Frame structure. a) Undeformed configuration. b) Deformed configuration.

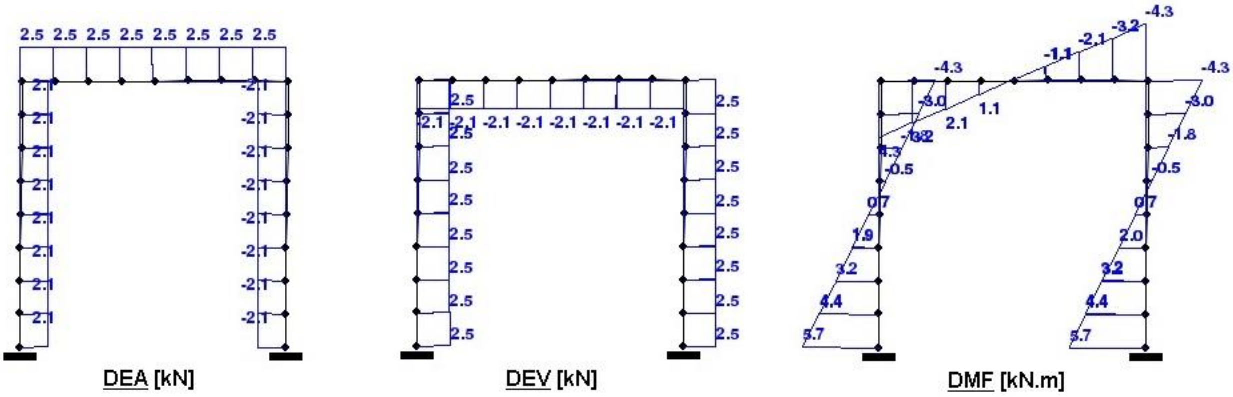

Figure 5: Frame structure - Diagrams.

4.2. Incremental elasto-plastic analysis

Although very simply, it made sense to start to include in this work some non-linearity, as the applied formulation has already shown good results in the analyses of non-linear behaviors (Asprone, Auricchio et al. 2014).

Admitting that the used materials have an elastic perfectly plastic behavior, it was assumed that the element total plastification occurred with the more stressed sections plastification – axial stress interaction was not considered.

Simulations were carried out on a beam, Figure 6, with a single vertical load at mid-span. The beam is an IPE100 steel beam with a plastic bending moment of 10,84 kN.m.

As the loading was increased (by the user) along the simulation, plastic hinges were formed as the more stressed sections reached its resistant capacity. In this structure, two plastic hinges were needed for the structure to become a mechanism and collapse.

By the observation of Figure 8 and Figure 9, it is visible the formation of the two needed plastic hinges, in the most stressed sections. The collapse mechanism is achieved for a load of 21,8 kN (0,57% error) and, following that, the structure continues to deform without load increment, as expected.

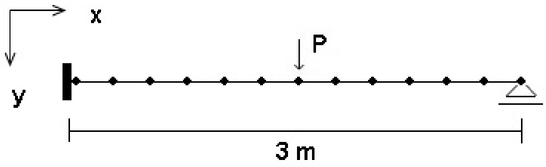

Figure 6: Beam - Undeformed configuration.

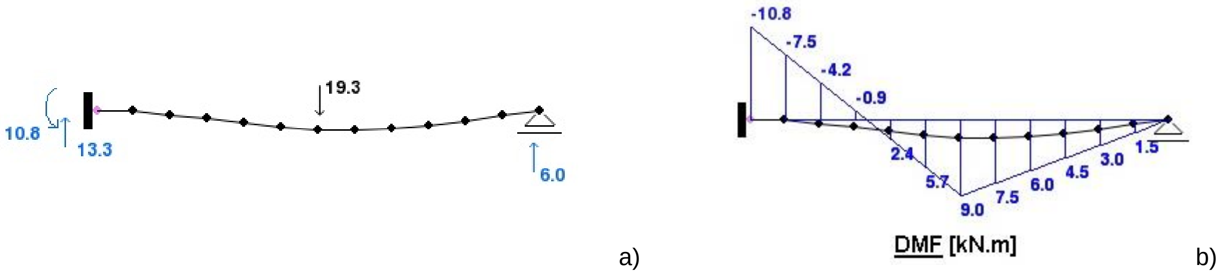

Figure 7: 1st Plastic hinge. a) Deformed configuration. b) Bending moments diagram.

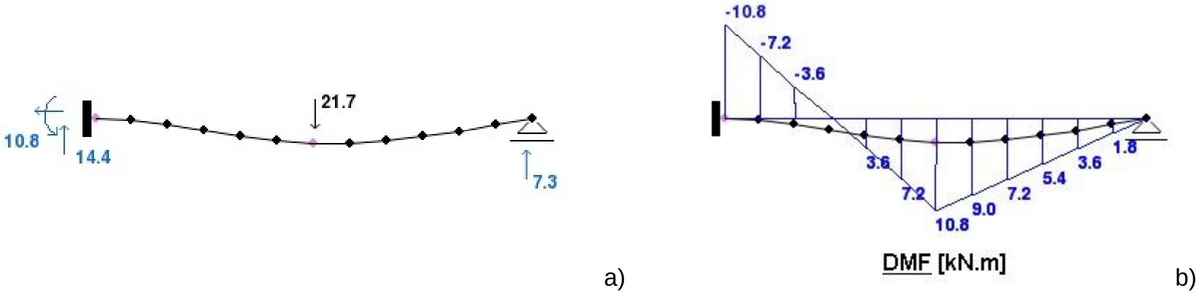

Figure 8: 2nd Plastic hinge. a) Deformed configuration. b) Bending moments diagram.

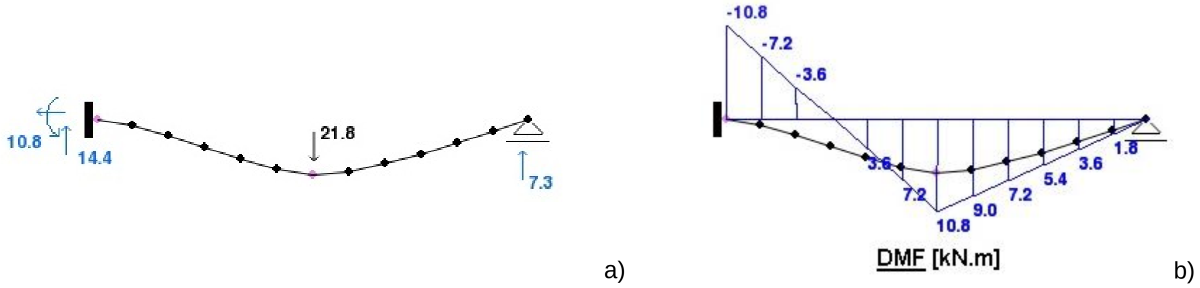

Figure 9: Collapse mechanism. a) Deformed configuration. b) Bending moments diagram.

In the next figure, is presented the mid-span trajectory where is easily identified the structure loss of stiffness.

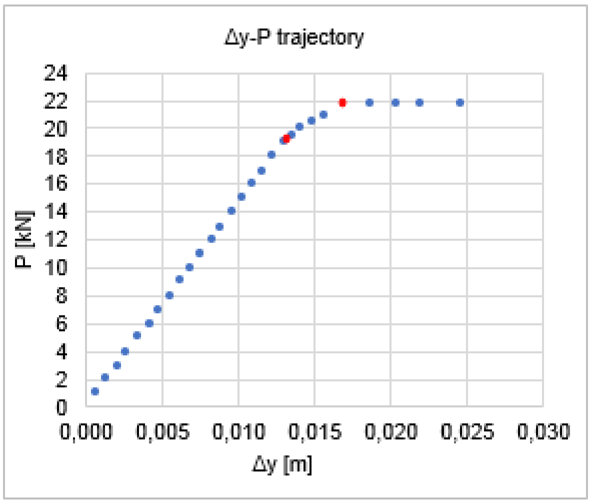

Figure 10: Mid-span particle trajectory.

4.3. Dynamic analysis

Evaluating structure behaviors under variable actions (over time) allows, for example to understand how a structure will react under seismic loads and, with that, designing the structure accordingly.

In this work, (Matos 2017), it was evaluated a portal frame structure, with only one degree of freedom (Figure 11) in both a free damped oscillatory motion and a forced oscillatory motion under a harmonic load. The natural angular frequency of the structure was compared to exact values to validate the results.

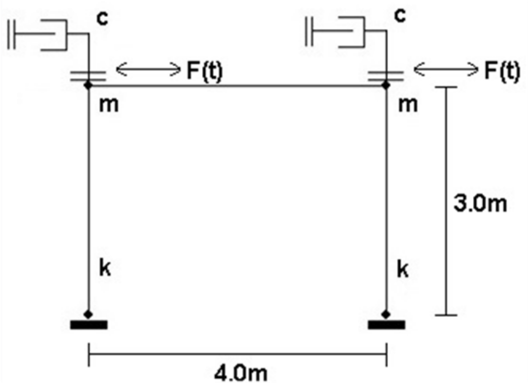

Figure 11: Evaluated portal frame structure - dynamic analysis.

Using the damped oscillators equation of motion (equation (24)) applied to the degree of freedom, it was obtained the solution accordingly to the evaluated motion.

$$m\ddot{x} + c\dot{x} + kx = F(t) \tag{24}$$

where m is the mass of the degree of freedom, c is the damping coefficient and k is the structure stiffness.

Dividing all terms for the mass value, a more general form of equation (24) is obtained,

$$\ddot{x} + 2\zeta p\dot{x} + p^2 x = \frac{F(t)}{m} \tag{25}$$

where ζ is the damping coefficient and p is the natural angular frequency, both given by the following expressions,

$$p = \sqrt{\frac{k}{m}} \tag{26}$$

$$\zeta = \frac{c}{2mp} \tag{27}$$

Free oscillatory motion

The exact solution of the free oscillatory motion ($F(t) = 0$) of the structure was found using the following solution of the equation (25),

$$x(t) = e^{-\zeta p t}(\bar{x}\cos(p_d t + \theta) \quad (28)$$

where p_d is the damped structure frequency and $\bar{x}$ and θ represent the initial conditions. The respective expressions are presented next.

$$p_d = p\sqrt{1-\zeta^2} \quad (29)$$

$$\bar{x} = \frac{\sqrt{(\dot{x}_0 + \zeta p x_0)^2 + p_d^2 x_0^2}}{p_d} \quad (30)$$

$$\theta = arctg\left(\frac{\dot{x}_0 + \zeta p x_0}{p_d x_0}\right) \quad (31)$$

In this simulation, it was used the following values for the mass, damping coefficient and initial displacement.

Table 1: Structure characteristics.

m [kg]	x₀ [m]	c [N.s/m]	F(t)
50*	0.1	500*	0

*50% split for each end of the beam.

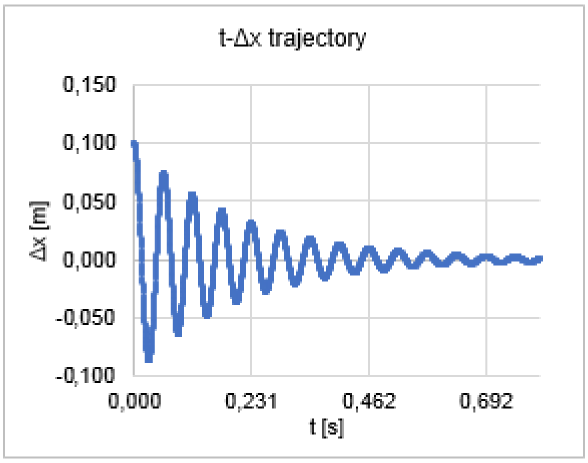

Figure 12: Exact solution of the free oscillatory motion.

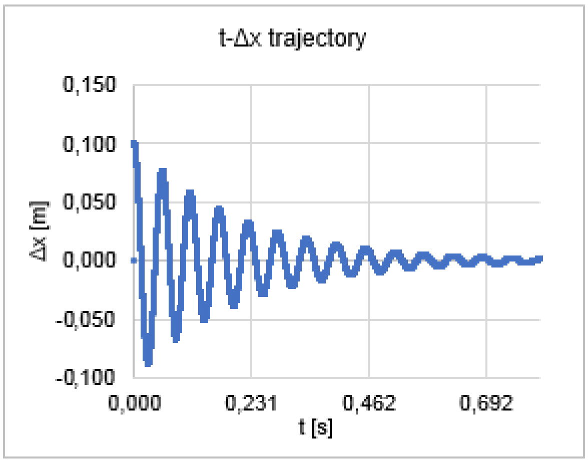

Figure 13: Solution obtained from the program.

Activating the option 'Export to Excel' in the environment properties of this program, it is possible to get elements information of the current simulation. Selecting the element and pressing 'e' on the keyboard, the program creates an Excel file (.xlsx) with the element information. In the presented simulation, the exported data (Figure 13) concerned the right end particle of the horizontal beam element in Figure 11.

To compare the obtained results, they are presented next in the form of a table.

Table 2: Free oscillatory motion results.

	T_d [s]	p_d [rad/s]	ζ
Exact results	0.0577	108.810 *	0.0459 **
Program results	0.0568	110.620	0.0452
Error [%]	1.56	1.66	1.53

*Obtained using expression (26). **Obtained using expression (27).

The damping coefficient value resulting of the program simulation was obtained using the logarithmic decrement method.

$$2\pi\zeta = ln\left(\frac{x_i^{max}}{x_{i+j}^{max}}\right), \quad with \ \frac{2\pi}{T_d} = p_d \quad (32)$$

where x_i^{max} and x_{i+j}^{max} are two consecutive oscillation peaks displacements.

Forced oscillatory motion

In this regime, the structure is under a harmonic loading (equation (33)) applied on the degree of freedom mass.

$$F(t) = \bar{F}\cos(wt) \quad (33)$$

The structure response is the overlap of two solutions. One was already presented on the free oscillatory motion and, is independent of the external loading, is always the same (for each structure). That solution is known as the general solution. The second solution, presented in this section, is the particular solution and its equation can be written as follows,

$$x_p(t) = \bar{x}_p\cos(wt - \emptyset) \quad (34)$$

where $\bar{x}_p$ is the maximum amplitude, β_1 is the dynamic amplification coefficient and $\emptyset$ is the difference between the action and the structure response. Its expressions are presented as,

$$\bar{x}_p = \beta_1\frac{\bar{F}}{k} \quad (35)$$

$$\beta_1 = \frac{1}{\sqrt{(1-\bar{w}^2)^2 + (2\zeta\bar{w})^2}} \quad (36)$$

$$\emptyset = arctg\left(\frac{2\zeta\bar{w}}{1-\bar{w}^2}\right) \quad (37)$$

In this simulation, it was used the following values for the mass, damping coefficient and initial displacement.

Table 3: Structure characteristics.

m [kg]	x₀ [m]	c [N.s/m]	F(t)
50*	0.1	500*	10.cos(wt)*

*50% split for each end of the beam.

One way of knowing the fundamental frequency of a structure is by applying various harmonic loads, each one with a different angular frequency, w. With the maximum dynamic displacements of each structure response, calculate the dynamic amplification values, using the following expression, and obtain the β_1 curve.

$$u_e = \beta_1 u_d \tag{38}$$

where u_e and u_d are, respectively, the static and dynamic displacements.

The static displacement was also obtained using the program, but applying a constant external load with the same magnitude of the maximum amplitude of equation (33).

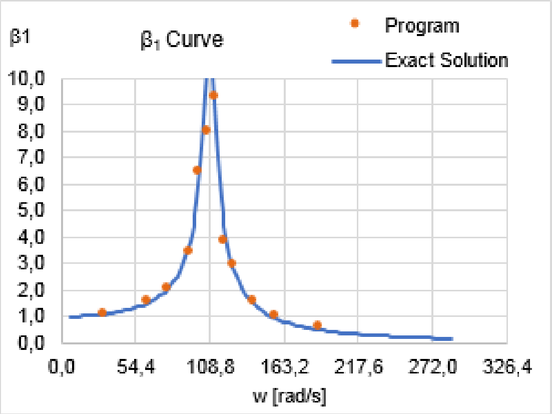

Figure 14: Comparison between the exact β_1 values and the ones obtained with the program.

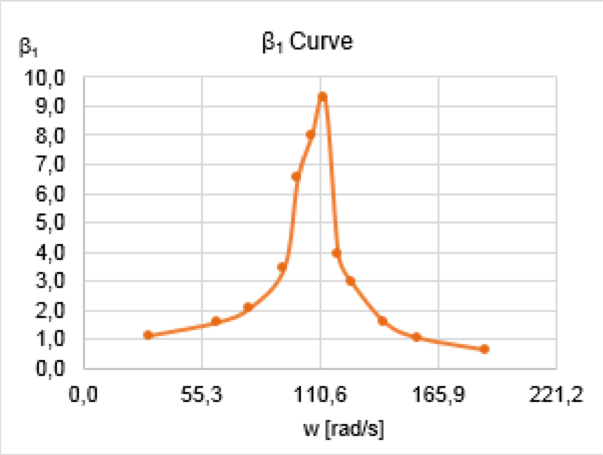

Figure 15: β_1 values obtained with the program.

In Figure 15, it is visible that close to the exact fundamental frequency of the structure, the program shows more error in the results. However, using only the values obtained from the program (Figure 15), it is possible to identify the action frequency to which the structure shows more response and with that obtain the fundamental frequency of the structure with good approximation (110,6 rad/s compared to the exact value of 108,81 rad/s - 1,65% error).

5. Computational performance

Performance is always a requirement for a computational program, especially for an interactive one, as the user does not only want satisfactory results, but wants them fast.

This program has two main ways of improving simulation times: by the used hardware and by software optimization.

To show the influence of the hardware, the same structure was simulated in two different computers (Table 4) and the total simulation time (the time used by the program modules) was compared (Table 5).

Table 4: Used computers specifications.

Hardware component	Computer 1	Computer 2
CPU	Intel Core 2 Duo T6500 2.1GHz	Intel Core i5-7200 2.5GHz
RAM	3 GB	8 GB
GPU	NVIDIA GeForce G103M	NVIDIA GeForce 930MX
Graphic memory	0,5 GB	2 GB

The simulated structure was a simple supported beam (L=2m) with a constant vertical force applied at mid-span.

Table 5: Simulation times.

	Computer 1	Computer 2
Simulation time [s]	22.3 (100%)	3.9 (100%)
Physic engine [s]	21.2 (95.2%)	3.5 (91.2%)
Graphic engine [s]	0.4 (1.7%)*	0.1 (1.8%)*
Event handler [s]	1.5 (6.5%)*	0.5 (12.0%)*

*These modules are not used only in the simulation mode, being this the total used time.

From the listed values, it is possible to verify that the use of a more recent computer (the other is a quite old model) speeds up the simulation by a factor of 6.

However, the biggest improvement that may be seen on the simulation times is not due to the use of a faster machine but rather on improving the actual code itself. A fully skilled programmer would know how to take advantage of the iterative and vectorized features of the formulation. In fact, using vectorized programming, an iterative method, as the one used in this work, would take much less time to carry out all the physical calculations and significantly reduce the overall simulation time (Liew, Mele et al. 2016).

Besides hardware, it is possible to “control” the simulation time with the value of the time-step, however there are some restrictions to its value – rods discretization and the magnitude of the loading needs to be taken into account.

The coarser the discretization, the larger are the rod segments and, because of that, each particle will have more mass. More mass means smaller accelerations, by Newton’s law, and it is possible to use bigger time-steps.

However, if the applied load is substantial, considering the used rod section, it might create high stresses and, with a bigger time-step, the simulation could diverge.

This program therefore requires some “sensivity” in the choice of the time-step. By default, the program uses a value that covers most of the possible cases. The following values may be used as a reference.

Table 6: Time-step reference values.

Discretization (nº of segments per rod)	Static analysis	Dynamic analysis
4	9E-5s	1E-7s
8	6E-5s	

6. Conclusion

This work is an advance in the development of a computational tool focusing on the didactic component in the field of structural mechanics. Alongside with concepts used in video-games, an interactive program was developed with the purpose of helping students in the understanding of structural mechanics concepts.

A Finite Particle Method used for the analysis of frame structures was presented. Based on the 2nd Newton’s law of motion and simple kinematics, particles equilibrium is constantly enforced and, with that, the global equilibrium of the structure does not need to be verified. Being all particles independent, this method is advantageous in the simulation of problems with large displacements, usually associated with non-linear behaviors.

The results from the numerical examples, demonstrated the capability of the method in simulating frame structures for static solutions, collapse mechanisms (using an elasto-plastic analysis) and dynamic behaviors.

Although both the dynamic analysis and elasto-plastic analysis presented are introductory, promising results have been shown and could lead to future developments.

7. References

Alamatian, J. (2012). "A new formulation for fictitious mass of the Dynamic Relaxation method with kinetic damping."

Asprone, D., et al. (2014). "Modified Finite Particle Method: Applications to Elasticity and Plasticity Problems ".

Liew, A., et al. (2016). "Vectorised graphics processing unit accelerated dynamic relaxation for bar and beam elements."

Lopes, P. (2015). Software development for assistance in the learning of structural analysis. Departamento de Engenharia Civil, Instituto Superior Técnico. **Mestrado**.

Matos, J. d. (2017). Simulação de problemas da mecânica estrutural utilizando "physics engines". Departamento de Engenharia Civil, Instituto Superior Técnico. **Mestrado**.

Rezaiee-Pajanda, M. and H. Rezaeea (2014). "Fictitious Time Step for the Kinetic Dynamic Relaxation Method."

Ting, E. C., et al. (2004). "Fundamentals of a Vector Form Intrinsic Finite Element: Part I. Basic Procedure and A Plane Frame Element."

Wu, T.-Y., et al. (2009). "Dynamic elastic-plastic and large deflection analyses of frame structures using motion analysis of structures."