Emerald Insight

International Journal of Web Information Systems

Client-server architecture for pre and post-processing of real problems involving two-dimensional generalized coordinates

José Luiz Vilas Boas Fabio Takeshi Matsunaga Neyva Maria Lopes Romeiro Jacques Duílio Brancher

Article information:

To cite this document:
José Luiz Vilas Boas Fabio Takeshi Matsunaga Neyva Maria Lopes Romeiro Jacques Duílio Brancher , (2015),"Client-server architecture for pre and post-processing of real problems involving two-dimensional generalized coordinates", International Journal of Web Information Systems, Vol. 11 Iss 2 pp. 226 - 245
Permanent link to this document:
http://dx.doi.org/10.1108/IJWIS-12-2014-0044

Downloaded on: 01 November 2016, At: 23:10 (PT)
References: this document contains references to 29 other documents.
To copy this document: permissions@emeraldinsight.com
The fulltext of this document has been downloaded 103 times since 2015*

Users who downloaded this article also downloaded:

(2015),"Generic framework for enriching services: a multiagent approach", International Journal of Web Information Systems, Vol. 11 Iss 2 pp. 246-268 http://dx.doi.org/10.1108/IJWIS-10-2014-0037

(2015),"Web spam detection using trust and distrust-based ant colony optimization learning", International Journal of Web Information Systems, Vol. 11 Iss 2 pp. 142-161 http://dx.doi.org/10.1108/IJWIS-12-2014-0047

Access to this document was granted through an Emerald subscription provided by emerald-srm:563821 []

For Authors

If you would like to write for this, or any other Emerald publication, then please use our Emerald for Authors service information about how to choose which publication to write for and submission guidelines are available for all. Please visit www.emeraldinsight.com/authors for more information.

About Emerald www.emeraldinsight.com

Emerald is a global publisher linking research and practice to the benefit of society. The company manages a portfolio of more than 290 journals and over 2,350 books and book series volumes, as well as providing an extensive range of online products and additional customer resources and services.

Emerald is both COUNTER 4 and TRANSFER compliant. The organization is a partner of the Committee on Publication Ethics (COPE) and also works with Portico and the LOCKSS initiative for digital archive preservation.

*Related content and download information correct at time of download.

Downloaded by TASHKENT UNIVERSITY OF INFORMATION TECHNOLOGIES At 23:10 01 November 2016 (PT)

The current issue and full text archive of this journal is available on Emerald Insight at:
www.emeraldinsight.com/1744-0084.htm

Received 2 December 2014
Revised 18 December 2014
Accepted 18 December 2014

Client-server architecture for pre and post-processing of real problems involving two-dimensional generalized coordinates

José Luiz Vilas Boas and Fabio Takeshi Matsunaga
Department of Computer Science, Londrina State University, Londrina, Brazil
Neyva Maria Lopes Romeiro
Department of Mathematics, Londrina State University, Londrina, Brazil, and
Jacques Duílio Brancher
Department of Computer Science, Londrina State University, Londrina, Brazil

Abstract

Purpose – The aim of this paper is to propose a Web environment for pre-processing and post-processing for 2D problems in generalized coordinate systems.

Design/methodology/approach – The system consists of a Web service for client-server communication, a database for user information, simulation requests and results storage, a module of (for) calculation processing (front-end) and a graphical interface for visualization of discretized mesh (back-end).

Findings – The Web system was able to model real problems and situations, where the user can describe the problem or upload a geometry file descriptor, generated from computer graphics software. The Web system, programmed for finite difference solutions, was able to generate a mesh from other complex methods, such as finite elements method, adapting it to the proposed Web system, respecting the finite difference mesh structure.

Research limitations/implications – The proposed Web system is limited to solve partial differential equations by finite difference discretization. We need to study about refinement and parameters adaptations to solve partial differential equations simulated with other methods.

Practical implications – The Web system includes implications for the development of a powerful real problems simulator, which is useful for computational physics researchers and engineers. The Web system uses several technologies, such as Primefaces, JavaScript, JQuery and HTML, to provide an interactive user interface.

International Journal of Web
Information Systems
Vol. 11 No. 2, 2015
pp. 226-245
© Emerald Group Publishing Limited
1744-0084
DOI 10.1108/IJWIS-12-2014-0044

Thanks to CAPES for the Master's scholarship to the researches Fabio Takeshi Matsunaga. This current paper is an extended version of the ICCSA 2014 (International Conference on Computational Science and its Applications) conference paper.

Originality/value – The main contribution of this work is the availability of a generic Web architecture for including other types of coordinate systems and to solve others partial differential equations. Moreover, this paper presents an extended version of the work presented in ICCSA 2014.

Keywords Advanced Web applications, Mobile computing for the Internet, Web databases, Web data integration, Web architecture

Paper type Research paper

1. Introduction

Computational simulations are very important to represent real problems and situations, as they can describe several physical and natural phenomena in a simplified way, through mathematical-computational models, such as partial differential equations (PDEs) – Tonti (2014). These models can be solved by analytic models or by numerical methods, where the last one discretizes a continuous domain in a finite domain. One numerical method that is often used is the finite difference discretization (FDD) method, whose purpose is to divide the geometry represented by Cartesian coordinates in several elemental volumes (Gupta *et al.*, 2013). There are three important steps to be considered on FDD method: pre-processing, processing and post-processing.

The pre-processing consists on the initial data preparation, spatial-temporal domain description (Schwertfirm *et al.*, 2008). Furthermore, this step consists on input parameters definition, such as initial/boundary conditions definition and is considered one of the most onerous steps, as there is a great real data dependency (Munková *et al.*, 2013). In the pre-processing, there is the sub-processing, whose purpose is to divide the geometrical model (2D or 3D) in several analytic elements or nodes, where all the operations are performed. The post-processing step consists on data extraction of partial/final results of simulations, to perform data analysis and graphical visualization or manipulation of the results, allowing to visualize the results in different perspectives, besides allowing the improvement of results precision (Walker and Chapra, 2014).

In most cases of real problems, there is a need to use a discretization and visualization modeling which is adequate to the problem boundary and domain. The physical domain of real problems, the majority being described as (x, y), does not always have a regular and well-defined geometry, which needs to be easily recognized by computer systems and simulators (Maliska, 2004). Considering this, there was an emerging motivation for developing solutions for irregular meshes generation, such as the FDD in generalized coordinates systems (GCSs) (Cirilo and Bortoli, 2006). This method performs a mapping of the nodes of a discretized mesh in an equivalent point in a physical domain, aiming to describe several types of 2D and 3D objects.

In the mentioned context, several solutions involving computational meshes generation, have been proposed (Notsu *et al.*, 2013; Liu and Xing, 2013). However, there are few solutions involving interactive and accessible computational systems involving PDEs resolutions. Most of the solutions use scientific softwares, or interactive tools such as videos (Grande *et al.*, 2010), which require some ideals requirements (memory capacity, a setting ideal hardware and software installation). Issues such as the accessibility of mobile services have also been treated (Dumont and Mourlin, 2007), and these ideas have emerged due to the simulators which have their source codes adapted for each test case and problem.

It has become an emerging motivation in the use of Web technologies for complex applications, as an information retrieval system executed at client-side by end-user programming, aiding users to realize functions by their own demands, without writing any source-code (Han and Tokuda, 2010). The knowledge and learning transfer is practical and allows the reusing of several routines. Another is the use of large data volumes, which requires database and remote server for information storage, which are constantly maintained and easily accessible from anywhere and at any time (Isbasoiu, 2011). The control and version upgrade is automatic and transparent, without any inconvenience to the users. Moreover, the services do not need be reinstalled if any malfunction occurs on the computer.

Some Web-based solutions for several real world problems have been developed (Weng, 2011). However, there are few existing Web-based solutions that have focused on solving PDEs and simulation of various phenomena described by this model, as such most of them are focused on developing a Web template for a specific case (Walker and Chapra, 2014). Recently, a Web service architecture was proposedby Matsunaga *et al.* (2014), which solved PDEs by FDD method in Cartesian and generalized coordinates. However, these systems can be explored and extended to multiple scenarios and by solving real problems in general. In addition, the portability and accessibility of these simulators tools have other important issues to be addressed (Wriedt, 2008), as many of these devices have restricted execution and the processing hardware is limited.

Considering the exposed above, we hypothesized that a Web-based architecture system can be a useful tool for easy problems solution storage for further analyses and for real problems solving by complex methods, with less browser plug-ins, allowing the services accessibility and mobility. The main objective of this work is to develop a client-side based module with pre-processing and post-processing systems for 2D problems in GCS, which access the results calculated in a remote server stored in a remote database. For this purpose, a Web-based interface was proposed, representing a user interactive system, where the user will be able to describe their own geometry and solve several kinds of geometries and enter with pre-defined 2D geometry files provided by real data extraction, facilitating the flexibility to describe the problem domain. Then, this Web-based module can be used as a future model of pre- and post-processing simulations of various types of PDEs, taking advantage of the well-defined structure of the FDD method.

The rest of this paper is organized as follows. Section 2 summarizes some related works. Section 3 describes the architecture and functioning of the Web system. Some experimental results about real scenarios to illustrate the feasibility of proposed system are detailed in Section 4. Finally, conclusions and future research directions are reported in Section 5.

2. Related works

We can notice that the algorithms and methods mentioned above are ideal methods and can be studied further to be used in computing environments, such as the Web systems. A Web and educational service for numerical calculation and physical simulation were developed (Isbasoiu, 2011). This Web service contains a graphic generator for generalized types of studies about physical equations testing and simulation. In this work, there is a necessity of knowing all of the equations representing the physical phenomena and variable notions, even when there is the possibility of phenomena interpretation, providing a creative process, optimal solutions for problems and possible software elements reusing.

Recent researches (Doboš and Steed, 2012) focused on the frameworks development for storing 3D meshes, through a revision control and merge images in the repository (database), which facilitates the operations of reading and writing. This is accomplished through a Web client, for HTML5 and WebGL, responsible for remote visualization of content stored in the repository in a Web environment. In the same research line, Schwartz *et al.* (2013) proposed a WebGL-based framework to represent 3D geometric reflectance information via bidirectional texture functions. All the progressive transmission and interactive rendering of geometries were worked in modern Web browsers.

Weng (2011) proposed a Web site called WebDFEA for post-processing of structures analyzed via finite element method. The work focused on the possibility to visualize these structures in a Web page with the same possibilities of graphical manipulations and controls in computer graphic software, typical of computer aided design (CAD) softwares. The development of WebDFEA was performed using HTML technologies for user interface and database for storing the models.

Web semantic time series processing have been developed to demonstrate how Web technologies can showcases their functionality and server different groups of interest, covering user interface issues (Božić and Winiwarter, 2013). The main contributions about this work were dynamic integration of ontologies, validation and inference based on stored data. The semantic time series processing could be evaluated within research projects and validated under real scenarios to increase its applicability.

Finally, Walker and Chapra (2014) proposed a Web-client model oriented for general purpose simulations. The proposed model was used to develop Web systems with graphical interfaces for real applications, such as simulation of oxygen demand in rivers. Thus, it showed the efficiency of technologies for a specific Web application providing user interactivity and Web visualization. However, the proposed work developed could be extended to offline access and local data storage for further post-processing. Furthermore, the client-side configuration of the proposed model can be onerous for more complex calculations which require high-cost computing, as all the numerical computation are performed on the local machine, i.e. the Web browser.

3. Web system architecture and main modules

The environment for mesh generation and visualization in GCS is composed by the following modules: interface, the main interface of the system for pre- and post-processing and the Web service module, to connect the interface with the remote server and database, which contains all the simulations information. The interface or client side (Figure 1) were developed in Java Server Faces (JSF) framework and Primefaces (Boekel, 2014), widely used to develop Web interface. The Web service was developed in JAX-WS API, which connects with the remote server with Glassfish 3.X server.

In Figure 1, the pre-processing, the user will interact with the interface (Primefaces), performing the spatial/temporal parameters and the problem domain, defining the problem geometry. In this process, the problem graphical representation is performed by Canvas, a JavaScript library with functions to capture the mouse events for drawing. After that, the geometry data are sent to remote server by a JavaScript event and JSF controller layer. On the post-processing, while the calculations are performed, the partial results are exhibited on the interface in runtime, using a JavaScript timer function,

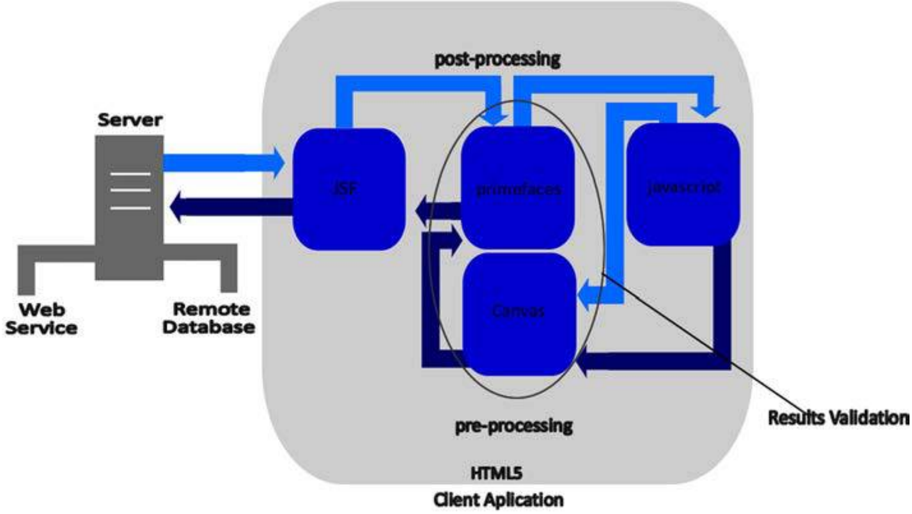


Figure 1.
Main modules of the Web system and the dataflow

Note: The strong blue arrows represent the pre-processing flow and the light blue arrows represent the post-processing data flow

which performs a periodic request to the remote server, using the Pool component, from Primefaces.

When the simulation is finished, the user has access to the matrix results, which can be downloaded in XML format or standard matrix format from scientific simulators, such as Matlab and Scilab. It is also possible to do the reverse operation, doing a file upload with matrices generated by Matlab and Scilab, mount the mesh and perform the simulation. The results and discussion section will be given an example of how to perform such an operation.

3.1 Client-server communication

This module is the main way for making the exchange of information input and output calculations, providing a runtime interaction with the user. To communicate with the Web service, the simple object access protocol, which is responsible for the exchange of messages in eXtensible Markup Language (XML) documents format, was used to provide the client-server interaction (Figure 1). The JAX-WS (Java API for XML Web Services) language was used to implement the Web service. The deployment of services was conducted in GlassFish 3.1 server, as this is a servlet container with an application server, which is useful for integrating Web services in a service-oriented architecture. The GUI was developed using the Primefaces framework, a useful tool for developing dynamic Web pages. The JSF framework was applied to perform session control and connection model-view-controller (MVC) with the interface. The communication and messaging interface has been described with Web Services Description Language (WSDL), which has several indexes methods to be requested by a Web service. The methods range from the search for information in the database, managing user information and requests to the command for executing the application requester (Figure 2).

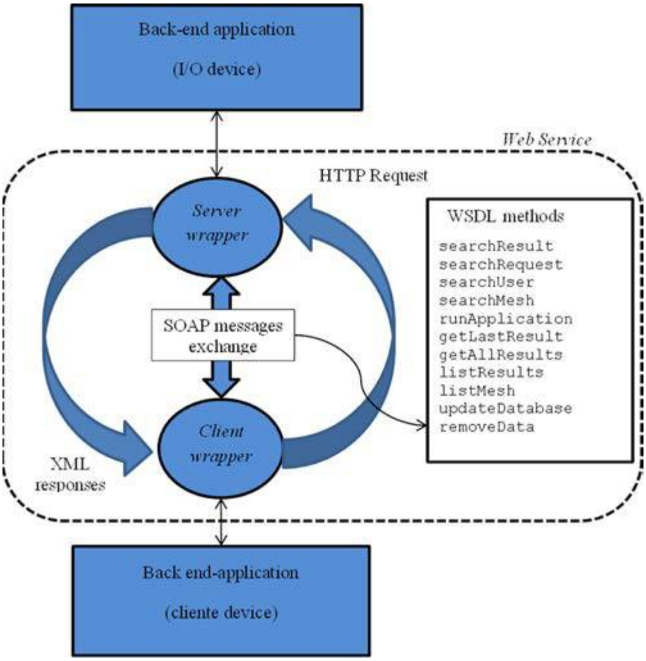


Figure 2.
Web service diagram and some WSDL messages

The developed Web service provides a means for communication with the user and the I/O (input-output) device that consists of a remote server that sends and receives data processing (Figure 2). Once this service is available, the communication with the Web service through a GUI and XML documents builds the system Web, along with the other modules, establishing a transparency between applications.

3.2 Requests and results remote storage

The storage method about the user information, calculations and outputs processed by the server is the PostgreSQL database stored in the server. This is accessed by the Web service, which sends information to the front-end and the back-end. We also used the Hibernate framework to allow any manipulation of database information in a more automated way by Java, without the need to generate large and cumbersome SQL strings. To measure the performance of queries and make the most optimized relational mapping object has conducted a number of tests with Apache JMeter software 2:11. This study was of great importance because it allowed us to identify the best way of programming methods, optimizing the entity-relationship modeling of the database, without any information redundancy (Figure 3).

As can be seen in Figure 3, each tuple of the "User" table is associated with multiple requests ("Request") with the respective results ("Result") and mesh information ("Mesh") and a foreign key to the "Request" table, which contains information for each request calculation. This table also has a relationship with the "Mesh", which contains attributes of mesh ID, corresponding ID request, problem domain and the information of the mesh points stored as "varchar", both for axis x (MeshX) as for the axis y (MeshY).

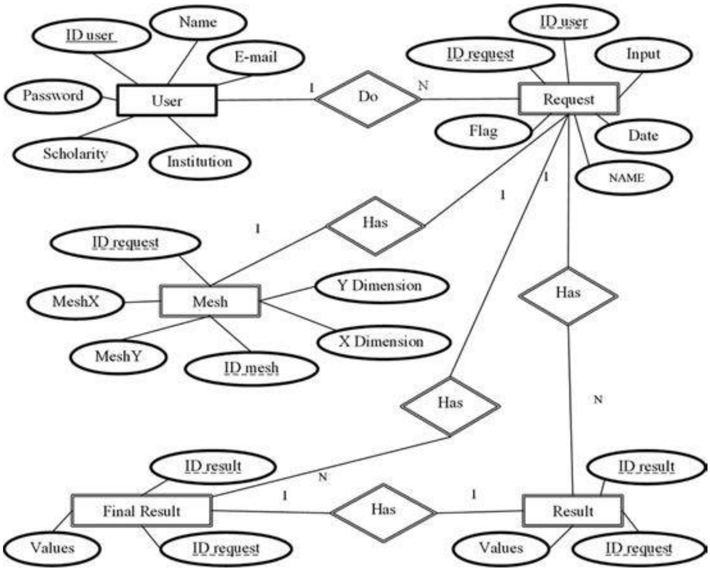


Figure 3.
Entity-relationship diagram of the database composed by five tables

The "Result" table stores results information corresponding to the foreign key in "Request" table for each time instance, containing the information of the results IDs and a trigger that mirrors the results data to the "Final Result", which is a table with all problems solved by all the users. This model also allows the sharing of a set of solved problems and cache storage of previous calculations, where a problem solved need not be calculated again. In addition, the "Request" and "Result" tables have features such "date", "time" and title of the simulation where any user can access solutions requested in a specific time or search by its name.

3.3 Remote server application and Web interface description

The Web application 2D problems involves mesh generation in GCSs (da Silva *et al.*, 2009). The process consists on a discrete definition (x, y) of the geometry boundary, which is interpolated by a numerical method, such as spline interpolation, to perform a continuous definition. Then, a transformation metric – a transformation of (x, y) coordinates to a computational domain $\Phi = (\xi, \eta)$ – is applied to generate the mesh, which is defined by the governing equation:

$$\alpha\Phi_{\xi\xi} + \gamma\Phi_{\eta\eta} - 2\beta\Phi_{\xi\eta} + (P\Phi_{\xi} + Q\Phi_{\eta})/J = 0 \qquad (1)$$

For the mesh generation, the user must initially specify the initial dimensions – N (line numbers in ξ) and M (line numbers in η). In this work, we assumed a structured and uniform mesh, wherein the distance between each node is $\Delta\xi = \Delta\eta = 1$. The method used was the elliptical type, described by the governing equations $\xi = \xi(x, y)$ and $\eta = \eta(x, y)$. The ξ and η lines describe the discretized meshes, whose points are the intersections of these lines determined by the governing equations. To solve these, we must firstly define the boundary conditions γ and the belonging (x, y) points. In this case, the ξ lines start in Γ_1 and end in Γ_3, and the η lines end in Γ_2 and end in Γ_4, as can be seen in Figure 4.

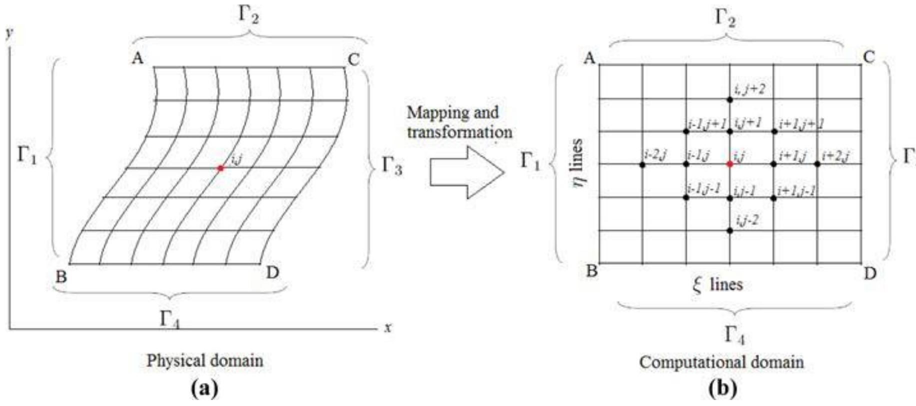
From Figure 4, all points of the physical domain are mapped to an equivalent point in the transformed domain ξ(x, y) and η(x, y). This mapping is performed by metrics denoted by $\alpha = x_{\eta^2} + y_{\eta^2}$, $\gamma = x_{\xi^2} + y_{\xi^2}$ and $\beta = x_\xi x_\eta + y_\xi y_\eta$ and the Jacobian $J = (x_\xi y_\eta - x_\eta y_\xi)^{-1}$, which are responsible for the compensation due to the coordinate system changing. These metrics are applied in the governing equation, which calculates a generic point of the mesh grid φ described by the (x, y) position.

For this work, the PDE applied on the system is the 2D transport equation, a time-dependant equation (Maliska, 2004). Depending on the parameterization, the equation can describe various phenomena, such as energy conduction according to the fluid movement. The σ is the diffusivity parameter, which describes the intensity of transport distribution; ρ is the specific mass and α, β and γ the transformation metrics with the Jacobian J [equation (2)]:

$$\begin{aligned}&(1/J\sigma)(\partial T/\partial t) - J\alpha(\partial^2 T/\partial \xi^2) - J\gamma(\partial^2 T/\partial \eta^2) + \rho u(\partial T/\partial \xi) + \rho v(\partial T/\partial \eta)\\ &= J\alpha(\partial/\partial \xi)(\partial T/\partial \xi) + J\gamma(\partial/\partial \eta)(\partial T/\partial \eta) - 2J\beta(\partial/\partial \xi)(\partial T/\partial \eta)\end{aligned} \tag{2}$$

The transport equation (2D) in generalized coordinates is discretized approximating each partial derivatives of first and second order, using the FDD method (detailed equations can be found in Maliska, 2004). Considering a structured mesh $\Delta\xi = \Delta\eta = 1$, the approximation generates the implicit scheme (Cirilo and Bortoli, 2006). This leads to the linear system Ax = b resolution with eight non-zero diagonals, as they consider the eight neighboring points in the region (i, j) – Figure 1. This system was solved using the iterative Gauss-Seidel method, whose solution is the value of $T_{i,j}$ in each particular grid point, generating a matrix, whose size is the number of mesh grid points generated on mesh discretization. The results of the PDE are a set of matrixes for each time instance. The FDD calculation application on the remote server has been developed in C/C++, one general purpose programming language that runs on multiple platforms.

During the calculation processing, the PostgreSQL libraries were integrated for recording the results in the database. This application is called by a method of the Web service, after the request from the front-end. After the process, the Web system needs a



Physical domain
(a)

Computational domain
(b)

Figure 4.
Computation of a mesh and the ξ and η lines

way for the user to access and view the simulation results. This is performed at runtime of the problem being solved. Considering this, an interactive tool, developed in JavaScript, was implemented for visualization of discretized meshes in generalized coordinates, as well as a means of user interaction such as query results and access to manipulation of the geometry.

All of the calculations are performed in the remote server and the results are integrated on the Web interface. The Web mesh generator and the simulation results module, which activity diagram is described on Figure 5, consists of user interface developed in Javascript and Primefaces, along with functions of Canvas library for the design of the geometries. The resulted interface is described in Figure 6.

The developed GUI receives user input and displays the output of the corresponding geometry, following the seven steps, according to the activity diagram (Figure 5) and Web interface (Figure 6):

(1) *Dimension of the problem*: The user define the maximum size of x and y dimension.

(2) *Geometry points*: The user defines the segmental equation $[(x_0,y_0),(x_1,y_1)]$ with the Γ_n associated. The system performs automatic connection of the segments, where the first point of Γ_n is equals to the last of the Γ_{n-1} (for $1 \leq n \leq 3$). If $n = 4$, the last point of Γ_4 is the first of the Γ_1.

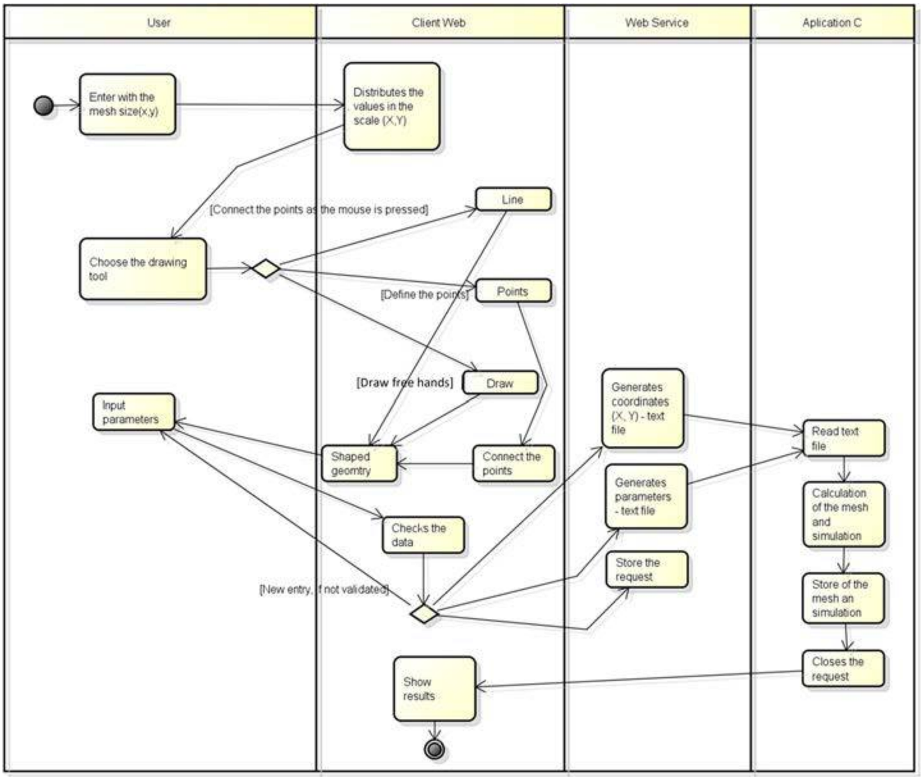


Figure 5.
Activity diagram about the Web mesh generator system

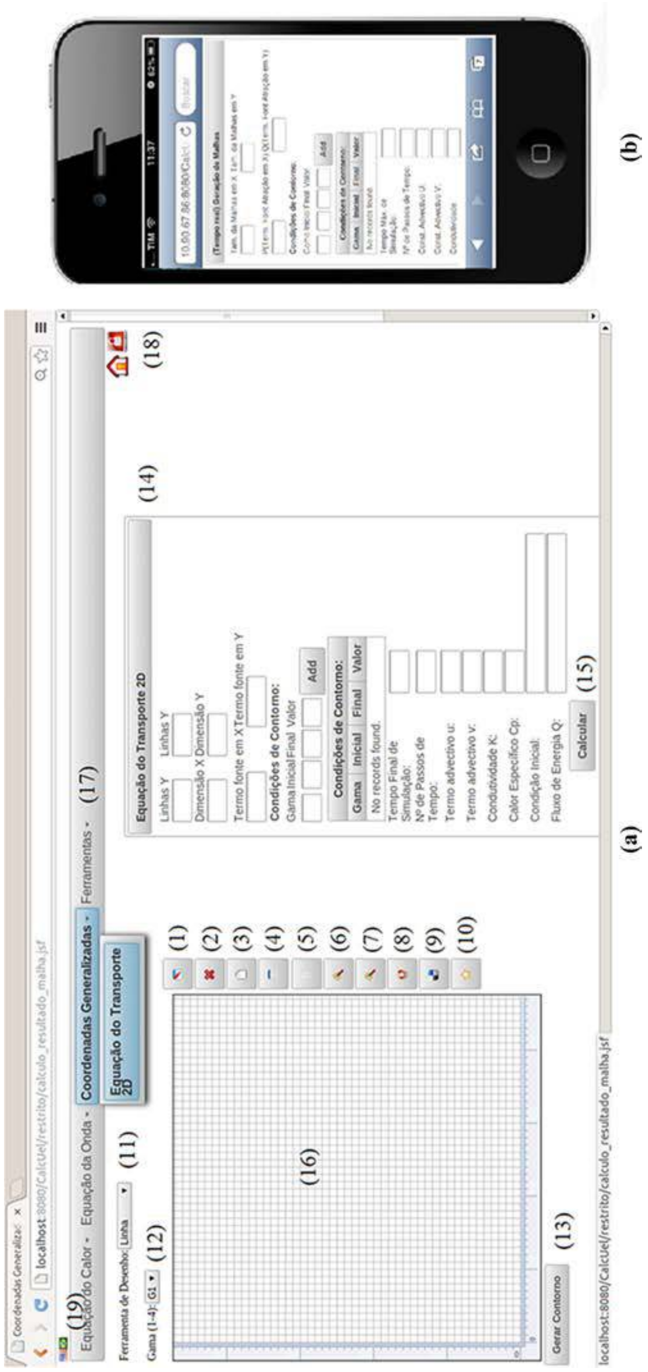


Figure 6.
Mesh generator interface developed in a Web browser (a) and a mobile device (b)

(3) *Mesh grid size*: Number of ξ and η lines.

(4) *Mesh information storage*: The calculations processing module stores this information in the table in the database.

(5) *PDE input parameters, based 2D transport equation (Maliska, 2004)*: Maximum simulation time (T_{max}) and the number of time instances (N_t) of $\Delta t = (T_{max})/(N_{t-1})$ length, initial condition $T(\xi,\eta,0)$ in Ω, Dirichlet boundary condition $T(\xi, \eta, t)$ in $\partial\Omega_0 \times [0, T_{max}]$, where $\partial\Omega_0$ is the boundary geometry and Neumann boundary condition $\partial T/\partial n$ in $\partial(\Omega - \Omega_0) \times [0,T_{max}]$ to simulate the flow over the boundary.

(6) The calculation processing module performs the simulation for each instance of time. The Poll process, a PrimeFaces artifice, makes constant queries to the database to verify the existence of new results.

(7) The GUI module performs the discretized mesh design and distribution of values over that. This procedure consists of the following steps:
    - Transforms the string of the x and y grid points extracted from the "Mesh" table into a set of vectors and sent to the front-end.
    - Draws first the ξ lines and then the η lines using the Path resource (from Canvas), generating the full grid (whole mesh).
    - Performs the distribution of values for each node in the grid (intersection of ξ and η lines), by defining the color of the dots according to the scale calculated on the basis of the higher value of the mesh, generating a legend of chain colors (Matsunaga *et al.*, 2014). This procedure is repeated until all instances of time are calculated and updated.
    - The color chain layout will be used for filling the geometry to assign a value for each grid point. These are considered for positive and negative values. Regardless of the results obtained from PDE simulated, the same scheme is used for problems of different distribution of temperature, energy or concentration, which are calculated and distributed over the tone. The trend is that the warmer colors represent higher values and cooler ones represent the lower values.

The Web mesh generator interface of Figure 6, both accessible by a Web browser such as a mobile device, has some interactivity schemes with the user, as geometry drawing and grid visualization. In addition, the user can view the numerical values for each grid point, which are also filled with a color set by the colors chain scheme. Other types of post-processing steps are possible to be applied to the system using resources from JQuery, a cross-browser JavaScript library. Some of other post-processing steps are zoom tool on a specific region and computer graphics analysis, such as rotation, translation and scale. These resources are an important aid in the analysis, as they allow viewing of a higher level of image detail, analyzing in the different perspectives and provide interactivity to the user. Hence, the user does not need any other software or application installed on the own machine, which are often paid or high-cost.

The Web system can read geometry descriptor files (GDF), to optimize the mesh generation processing. One of GDF is the object files (OBJ), which define the geometry for objects in Wavefront's Advanced Visualizer, specified as lines, polygons and curves and can also be used to transfer the geometry data to other applications, such as several scientific and computer graphics software, such as, Matlab and CAD systems. These files are edited by the user in ASCII format or generated from computer graphics application, then can be uploaded on the Web system which will generate the object mesh model according to the vertex coordinates. The .OBJ file is composed by various types of datasets, such as geometric vertices [(x, y, z) coordinates], normal vectors, face definition by set of vertices, Bezier curves paths and even materials texture data. In our case, the z axis is ignored because the systems works with 2D objects, and takes just the geometric vertices information.

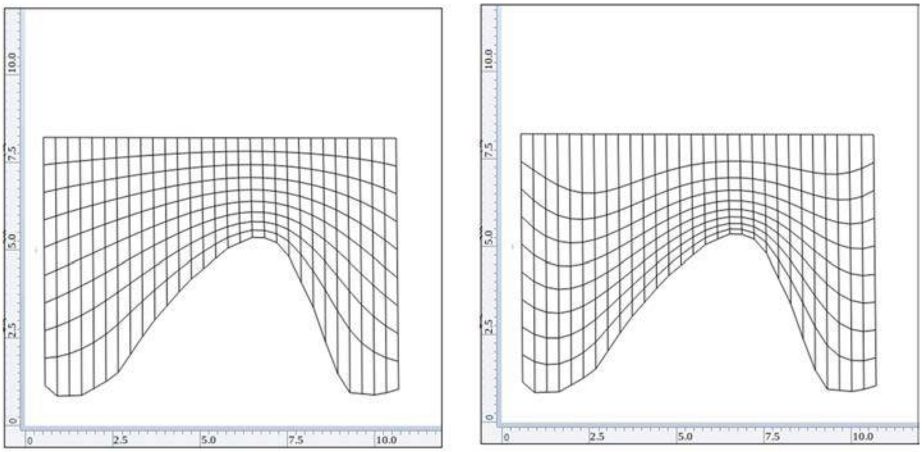
Another GDF readable from the Web system are GEO files, which are geometry files generated by the GMSH Open Source Mesh Generator (Geuzaine and Remacle, 2009), a meshing framework providing structured and unstructured mesh algorithms for 2D and 3D geometries. The GEO files contain extra information, as face description and normal vectors, useful for complex meshes description and computer graphics application. All of these formats are converted in a structure which describes all the mesh grid points (x, y), described as below:

1	x_1	y_1	0
2	x_2	y_2	0
3	x_3	y_3	0
4	x_4	y_4	0
5	x_5	y_5	0
6	x_6	y_6	0
7	x_7	y_7	0
8	x_8	y_8	0
.	.	.	.
.	.	.	.
.	.	.	.
n	x_n	y_n	0

All of the points are disposed considering that the mesh grid points are read in x axis direction. The vertices information is converted on a standard mesh grid format, described on database model section. Considering that the points form a structured mesh, you can receive files generated several discretization methods as input, making the necessary adjustments to the system.

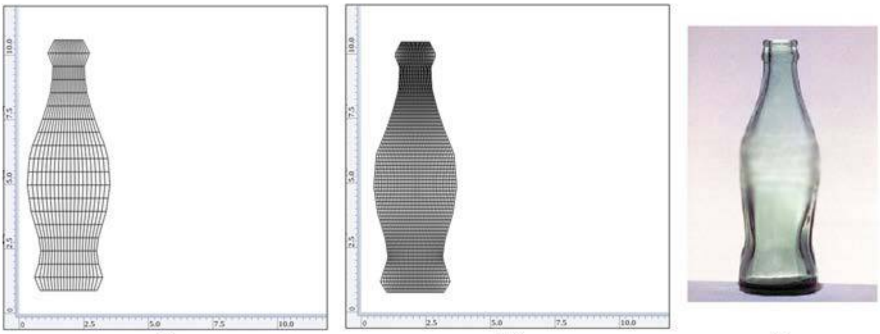
4. Results and discussions

For the Web system for PDEs simulation testing, some geometry meshes were defined and modeled. Initially, the user specified the coordinates of the physical domain that describes the geometry. In this case, it can make manual measurements for obtaining points or submitting a data file that already contains all the predefined points. The Figures 7 and 8 illustrate an example of the generated mesh displayed on the Web application developed in JavaScript interface. The geometry boundary points were obtained by millimetric measurements of the object silhouette. The user does not need to provide all points, as the spline interpolation method applied on the discrete definition calculates the continuous modeling of the whole geometry.



(a) (b)

Figure 7. Example of mesh generated from a manually described geometry in two types of deformation, considering P and Q parameters

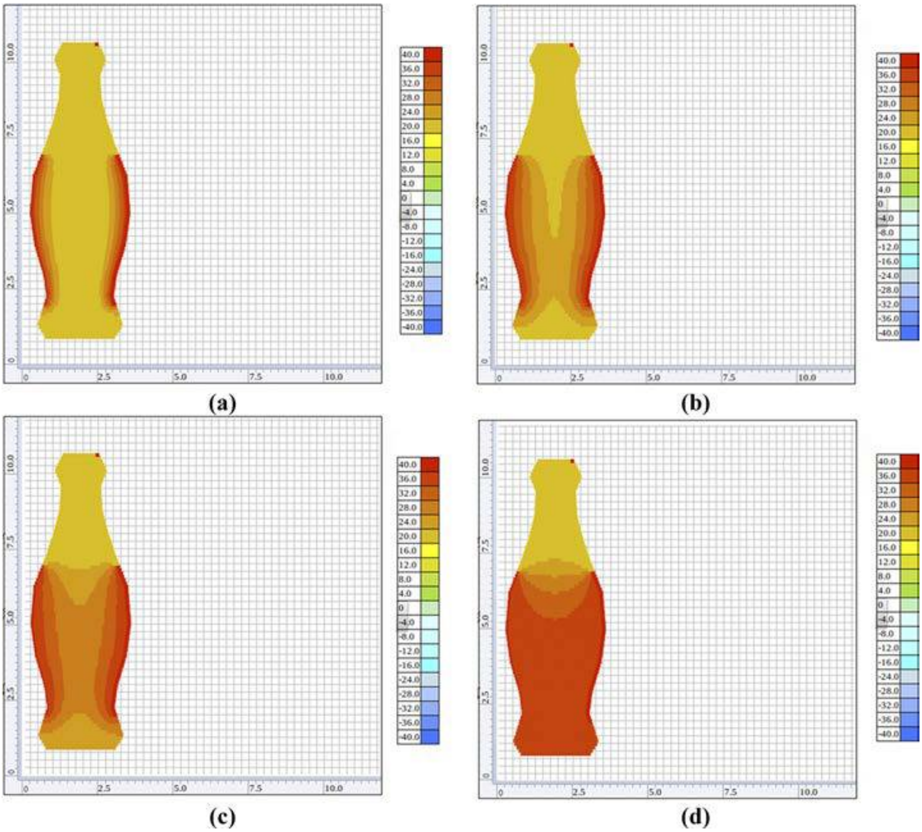


(a) (b) (c)

Figure 8. Example of a generated mesh of a bottle figures in different level of details – (a) and (b) – compared to the physical domain (c)

The bottle geometry, which physical domain is shown in Figure 8(c), was modeled considering different level of detail – 20 × 20 (6A) and 40 × 70 (5B). Accordingly, several objects may be shaped by some points of the 2D boundary. It may be noted in Figures 7 and 8 which were generated structured meshes (the layout of the lines of intersection points form a checkered mesh), regardless of the geometry. As long as the geometry is preferably convex, the mesh generator discretizes with the least possible gap, wherein lines of sights were performed in one direction for $Q = 1$ (Figure 7(b)) and without any attraction (Figure 7(a)). This discretization so structured facilitates the application of any EDP over the discretized mesh.

After the computational mesh generated, the 2D equation energy can be applied. Figure 9 illustrates an example of a simulation of temperature distribution (energy) over the bottle geometry (Figure 8). It is a situation in which a bottle in an ambient temperature (to the initial condition given by $T(\xi, \eta, 0) = 20$ degrees) is picked up with hands at both ends [Figure 9(a)], being subjected to a condition Dirichlet boundary of 40 degrees in specific regions of $\Gamma_1 = [2.5, 7]$ and $\Gamma_3 = [2.5, 7]$, in addition to parameters



(a) (b) (c) (d)

Figure 9. Outputs simulation of 2D energy equation in a generalized geometry in multiple instances of time, considering the heat distribution in a bottle geometry over the time

$T_{max} = 10{,}000$, $N_t = 100$, $\sigma = 1$, $u = v = 0$ (no fluid over the surface) and $q = 0$ (no fluid over the boundary). As applied the methodology to specify colors in points, the color warmer tones (red) represent higher temperatures, and cooler tones represent lower temperatures.

The simulation shown in Figure 9 illustrates the heat conduction on the irregular surface over time. Therefore, the system appears to be feasible to solve real problems and situations in a Web environment, especially in the case in which real situations are not usually described with a rectangular geometry, but with a generalized geometry. For this case, situations as the movement of fluids, external heat sources, exchange of energy between the middle and the driver and advective parameters were disregarded.

In the case of Figures 7 and 8, the user described manually the own geometry boundary, where the server application was responsible to calculate the mesh grid points. However, the definition of a refined mesh can be onerous for the system. Considering that, the user can upload a GEO or a OBJ file, which is an already generated mesh file, where the server application will adapt this file to run over the Web application. Some examples are described in Figure 10, which describes Igapo Lake surface, whose geometry was generated by finite elements method.

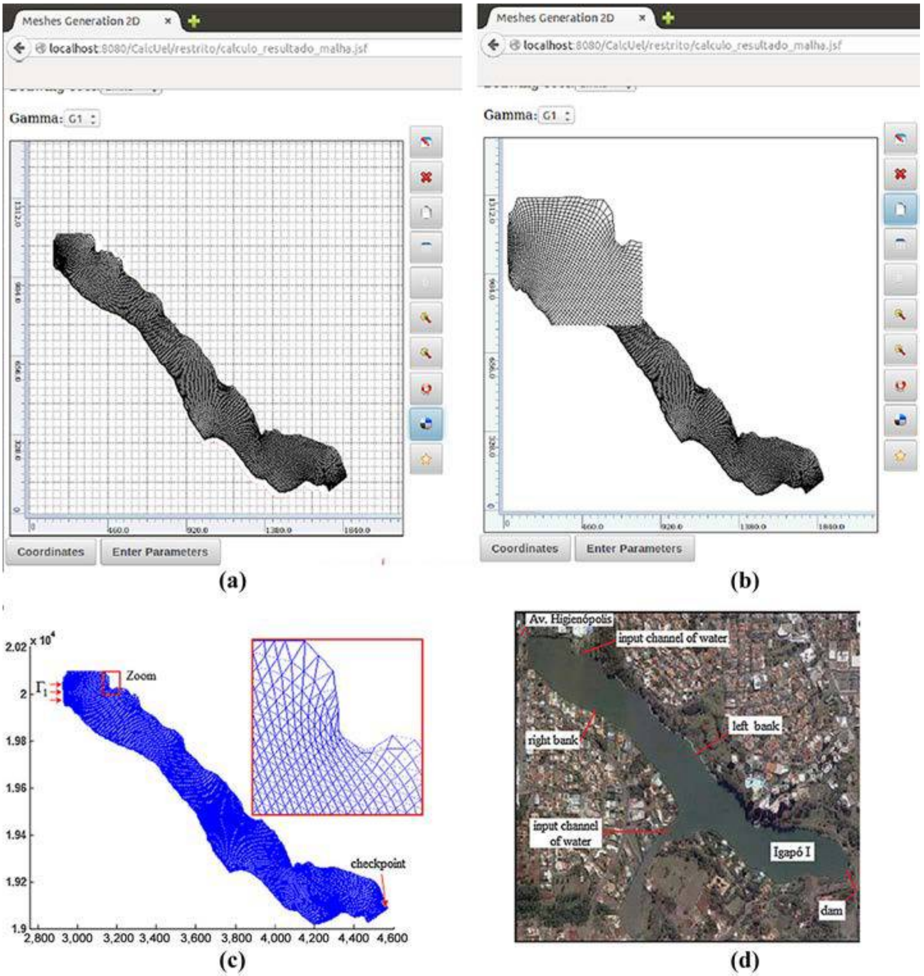


Figure 10.
Igapo Lake mesh generated from an uploaded

Notes: GEO file (a), considering the zoom post-processing operation using JQuery image zoom resource (b), the original problem, solved by finite elements method, obtained from Romeiro et al. (c) and the original physical domain (d)

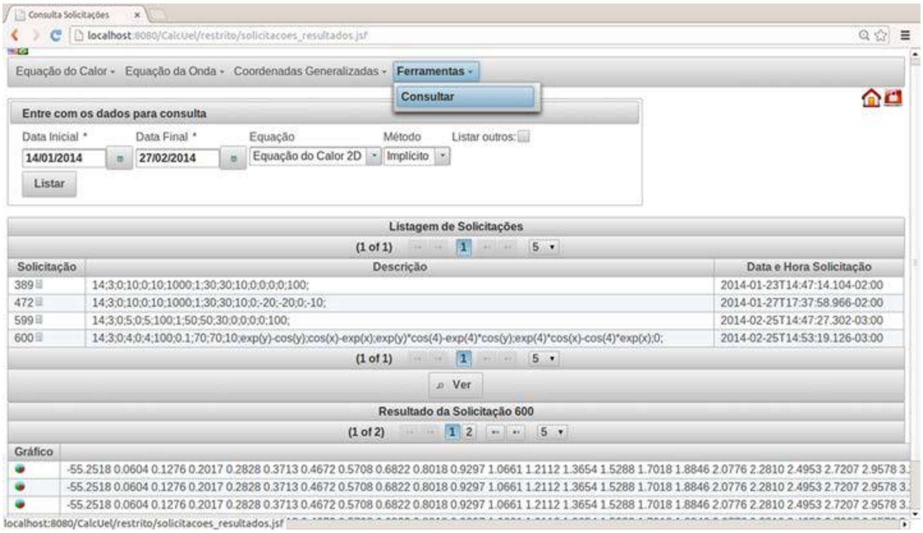
In the case of Figure 10, the realistic Igapo Lake geometry was generated and visualized on the Web interface. First, the user uploaded a GDF file, which contains all vertices and face commands to describe the triangular mesh elements, generated from finite elements method (Romeiro *et al.*, 2011). However, during the GDF file parsing, the face commands for triangular meshes (Figure 10(c)) are disregarded, to only generate the finite difference structured mesh, described as (ξ, η) lines. This shows that the Web system is capable of generating meshes from other models, in this case the finite elements method, but after performing the necessary adaptations, as the current system is programmed for the FDD method. From the generated mesh, it is possible to carry out PDE simulations, such as the application of the transport equation for pollutants dispersion simulation, as applied by

Romeiro *et al.* However, because the real problem was solved by EDP through a finite element method, we need to readjust the parameters, such as mesh refinement and velocity field calculated via Navier-Stokes equation, to be simulated without instability.

The use of a structured discretization allows the easy reading of the mesh descriptor file, as all the mesh grid points can be stored as a 2D matrix format. Moreover, the structured discretization allows that all results (mesh points and all instances results) are stored in the same way in the database, independent from the equation applied. The Web system provides a query system for problems' solution performed by own user or other users (Figure 11). From the list obtained in this search, you can expand each of them by consulting their results in all instances of time.

The searching of results in the database helps to solve problems (Figure 11). This retrieval system, in addition to request cached for the return of problems that have been solved, can be extended to other purposes. An example is the use of the concept of relative errors (Reimer and Cheviakov, 2013) for the system to list a set of similar problems, in addition to that having exactly the same inputs, the results of which have differences in values that are smaller than the tolerance error. Thus, the system searches for solutions in which a given variable has no significant influence on the final result, with a possible future implementation of PDEs models of sensitivity analysis (Weibe and Huisinga, 2011).

A fundamental property of the proposed work is that the Web system can run on any device, even mobile (Figure 12), which has limited hardware, where the user needs only a Web browser and the Internet connection, due to the simple configuration of the Web system. Thus, the user does not need to have knowledge about mathematical formulas and write programs to cover their necessities, due to the integration of several Web applications (client and server side), considering the same approach adopted by Han and Tokuda (2010). The user enters only with the basic input parameters and performs interactions with the mesh generator, and the remote server is responsible for complex calculations. In this case, the performance of the executions and the use of computational resources depend



Entre com os dados para consulta

Data Inicial *: 14/01/2014
Data Final *: 27/02/2014
Equação: Equação do Calor 2D
Método: Implícito
Listar outros: ☐

Listagem de Solicitações

(1 of 1)

Solicitação	Descrição	Data e Hora Solicitação
389	14;3;0;10;0;10;1000;1;30;30;10;0;0;0;0;100;	2014-01-23T14:47:14.104-02:00
472	14;3;0;10;0;10;1000;1;30;30;10;0;-20;-20;0;-10;	2014-01-27T17:37:58.966-02:00
599	14;3;0;5;0;5;100;1;50;50;30;0;0;0;0;100;	2014-02-25T14:47:27.302-03:00
600	14;3;0;4;0;4;100;0.1;70;70;10;exp(y)-cos(y);cos(x)-exp(x);exp(y)*cos(4)-exp(4)*cos(y);exp(4)*cos(x)-cos(4)*exp(x);0;	2014-02-25T14:53:19.126-03:00

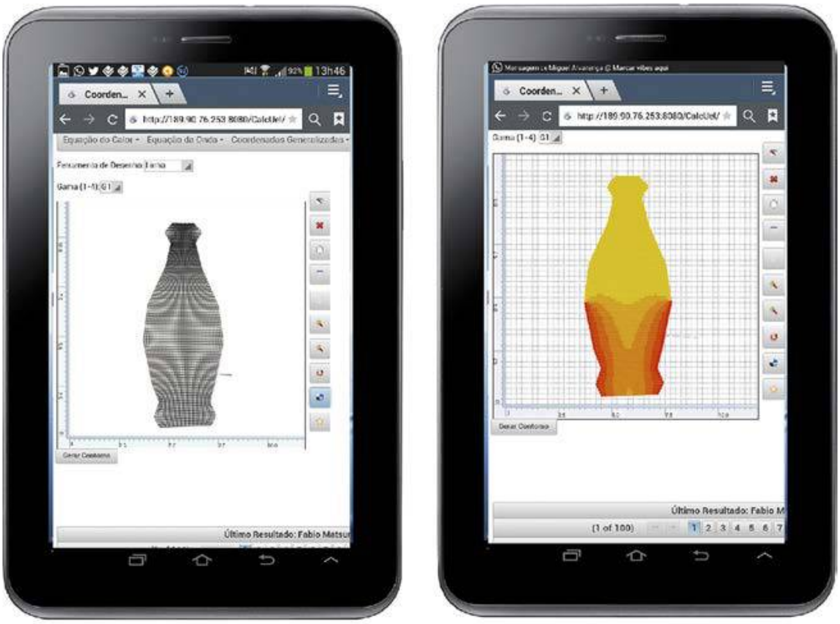
(1 of 1)

Resultado da Solicitação 600

(1 of 2)

```
-55.2518 0.0604 0.1276 0.2017 0.2828 0.3713 0.4672 0.5708 0.6822 0.8018 0.9297 1.0661 1.2112 1.3654 1.5288 1.7018 1.8846 2.0776 2.2810 2.4953 2.7207 2.9578
-55.2518 0.0604 0.1276 0.2017 0.2828 0.3713 0.4672 0.5708 0.6822 0.8018 0.9297 1.0661 1.2112 1.3654 1.5288 1.7018 1.8846 2.0776 2.2810 2.4953 2.7207 2.9578
-55.2518 0.0604 0.1276 0.2017 0.2828 0.3713 0.4672 0.5708 0.6822 0.8018 0.9297 1.0661 1.2112 1.3654 1.5288 1.7018 1.8846 2.0776 2.2810 2.4953 2.7207 2.9578
```

Figure 11. Searching of solved problems system, with a problem consultation and their parameters used



(a) (b)

Figure 12.
Mesh generation visualization – pre-processing/sub-processing stage (a) – and resolution of an PDE over the mesh – processing stage (b) – in the Web system accessed from a mobile device

exclusively on the remote server requirements, and the user does not need to have a high performance device to simulate numerical computations (Barboza *et al.*, 2010).

As we could see on related literature, the PDEs are widely used in the context of modeling and simulation. Regardless of the final application, even for digital games (Barboza *et al.*, 2010) and educational services (Singh *et al.*, 2012), using Web technologies allows users to perform only basic tasks, such as inserting the input parameters. However, the related works handled very complex methods, such as finite elements method. This method requires a very complex database to store huge amounts of data due to the complexity of the resulted structure based on a triangular mesh (Weng, 2011). On this proposed work, we started an adaptation of finite elements-based problems taking advantage of well-structured mesh generated from FDD method, aiming to posteriorly simulate PDEs solved by other methods.

As in research involving Web environments for pre- and post-processing, the proposed work involves Web technologies applications that allow the joining of numerical methods concepts with Web applications and their respective interactive tools available. This generates a generic Web architecture that also allows accessibility and mobility services (Figure 12). Furthermore, using the JavaScript and JQuery on the mesh generator/user interface module, other kinds of post-processing steps were possible to be applied on the system, such as specific region analysis to apply a multiblock discretization model, that is, a refinement in parts of geometry, as applied by Cirilo and Bortoli (2006).

With the research system, the solved problems can be found and listed anywhere and at any time, creating a collaborative system so that researchers, teachers, students and even people from other institutions can have access to the PDEs database resolved. The Web system can be used as virtual laboratory physical experiments. This theme has also been widely discussed in various studies (Singh *et al.*, 2012; Perus *et al.*, 2013) and contributes significantly to the knowledge transferring in practice and modern teaching.

5. Conclusions and future works

The work proposed in this goal was achieved as expected. It was possible to integrate the module pre- and post-processing meshes with applications that perform calculations processing of PDEs by the method of finite differences. Thus, the simulation of real problems and the automation of all processes in a Web and heterogeneous environment enable all actors involved and modules to work together at runtime. The recording of the data processed, was another crave and has reached its importance, where the user will have a record of everything that was calculated to further consult them for future comparisons and post-processing operations. It will also allow sharing solutions to problems and leave the system available to the scientific community.

The main contribution of our work is the availability of the interface module to be adapted in other kinds of Web service architecture. The Web interface only performs a connection to a remote database by Web service to consult the generated meshes results. Only few adaptations on the module are necessary if the developer desires to simulate other kinds of PDE solved by FDD, as the generated mesh is stored in a standardized format, due to the well-structured property of the mesh point arrangements, taking advantage of the well-defined structure that the finite difference method provides as results. Considering that, the proposed system is an online library with processed geometries and solved 2D problems, and can be used as a collaborative Web system.

To solve a problem, the users only need to provide input parameters of the problem, and the calculation processes are executed on a remote server. With this application, the deployment of the Web service allows the access from anywhere and anytime, through a browser or a mobile device that has the Internet access. Thus, the service becomes a collaborative system of several problems and solutions sharing, being a useful tool for knowledge transferring through a modern technological learning and education.

As parameters tests conducted to date, we can prove how efficient the system is, providing its users the possibility to test their results in a simulated real environment. To better improvement, the system will read predefined file formats, processed by researchers or computer graphics softwares in another room of your choice, and thus represent a domain with known geometries, regards the structure of the pattern file that will be offered by the software. We also intend to develop other post-processing tools, such as option to process a part of an area already processed for the purpose of obtaining greater detail in a region and implement the option of free drawing, this feature will allow drawing unconventional ways, such as drawing a free-hand figure.

As a future perspective, we intend to transform the client environment in a Web framework, allowing communication with any other application as the user's choice and solving other kinds of problems described by several kinds of PDE, such as Navier-Stokes equation. Another aim is to implement the resolution of three-dimensional problems, which is possible to modify the Web interface layer to allow the simulation and visualization of problems, especially in the GUI and processing layer calculations. The other way is to allow other kinds of mesh descriptor files or by processing results generation for later displaying in simulator (client) to allow the processing of other CAD software outputs.

References

Barboza, D.C., Junior, H.L., Clua, E.W.G. and Vinod, E.F.R. (2010), "A simple architecture for digital games on demand using low performance resources under a cloud computing paradigm", *Proceedings of 2010 Brazilian Symposium on Games and Digital Entertainment, IEEE Computer Society, Washington, DC*, pp. 33-39.

Boekel, R.V. (2014), "Por dentro do PrimeFaces 2.2 – artigo revista java magazine 93", available at: www.devmedia.com.br/por-dentro-do-primefaces-2-2-artigo-revista-java-magazine-93/21651 $\backslash$\#ixzz2mpXXhQAL

Božić, B. and Werner, W. (2013), "A showcase of semantic time series processing", *International Journal of Web Information Systems*, Vol. 9 No. 2, pp. 117-141.

Cirilo, E.R. and Bortoli, A.L.D. (2006), "Cubic splines for trachea and bronchial tubes grid generation", *Semina: Ciências Exatas e Tecnologicas*, Vol. 27, pp. 147-155.

da Silva, W.P., Precker, J.W., e Silva, D.D., e Silva, C.D. and de Lima, A.G.B. (2009), "Numerical simulation of diffusive processes in solids of revolution via the finite volume method and generalized coordinates", *International Journal of Heat and Mass Transfer*, Vol. 52 No. 22, pp. 4976-4985.

Doboš, J. and Steed, A. (2012), "3D revision control framework", *Proceedings of the 17th International Conference on 3D Web Technology (Web3D '12), Los Angeles, CA*, pp. 121-129.

Dumont, C. and Mourlin, F. (2007), "A mobile computing architecture for numerical simulation", *Proceedings of International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies 2007 (UBICOMM'07)*, available at: http://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=4402791, pp. 68-74.

Geuzaine, C. and Remacle, J.-F. (2009), "GMSH: a 3-D finite element mesh generator with built-in pre- and post-processing facilities", *International Journal of Numerical Methods in Engineering*, Vol. 79 No. 11, pp. 1309-1331.

Grande, A., Gonzalez, O., Pereda, J.A. and Vegas, A. (2010), "Educational computer simulations for visualizing and understanding the interaction of electromagnetic waves with metamaterials", *Education Engineering (EDUCON), 2010 IEEE*, available at: http://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=5487606, 14-16 April, pp. 543-547.

Gupta, P.K., Singh, J., Rai, K. and Rai, S. (2013), "Solution of the heat transfer problem in tissues during hyperthermia by finite difference-decomposition method", *Applied Mathematics and Computation*, Vol. 219 No. 12, pp. 6882-6892.

Han, H. and Tokuda, T. (2010), "Towards flexible and lightweight integration of web applications by end-user programming", *International Journal of Web Information Systems*, Vol. 6 No. 4, pp. 359-373.

Isbasoiu, E.C. (2011), "Collections of web services for applications in physics", *Proceedings of 2011 Second International Conference on Intelligent Systems, Modelling and Simulation, Kuala Lumpur, 25-27 January*, pp. 291-325.

Liu, Y. and Xing, H.L. (2013), "A boundary focused quadrilateral mesh generation algorithm for multi-material structures", *Journal of Computational Physics*, Vol. 232 No. 1, pp. 516-528.

Maliska, C.R. (2004), *Transferência de Calor e Mecânica dos Fluidos Computacional*, Editora Livros Técnicos e Científicos Editora, Rio de Janeiro.

Matsunaga, F.T., Vilas Boas, J.L., Romeiro, N.M.L. and Brancher, J.D. (2014), "A web system for solving real problems involving partial differential equations in generalized coordinates", *Computational Science and Its Applications (ICCSA 2014), Lecture Notes in Computer Science*, Vol. 8584, pp. 665-680.

Munková, D., Munk, M. and Vozár, M. (2013), "Data pre-processing evaluation for text mining: transaction/sequence model", *Procedia Computer Science*, Vol. 18, pp. 1198-1207.

Notsu, H., Ueyama, D. and Yamaguchi, M. (2013), "A self-organized mesh generator using pattern formation in a reaction-diffusion system", *Applied Mathematical Letters*, Vol. 26 No. 2, pp. 201-206.

Perus, I., Klinc, R., Dolenc, M. and Dolsek, M. (2013), "A web-based methodology for the prediction of approximate IDA curves", *Earthquake Engineering & Structural Dynamics*, Vol. 42 No. 1, pp. 43-60.

Reimer, A.S. and Cheviakov, A.F. (2013), "A matlab-based finite-difference solver for the Poisson problem with mixed Dirichlet-Neumann boundary conditions", *Computer Physics Communications*, Vol. 184 No. 3, pp. 783-798.

Romeiro, N.M., Castro, R.G., Cirilo, E.R. and Natti, P.L. (2011), "Local calibration of coliforms parameters of water quality problem at Igapó I Lake, Londrina, Parana, Brazil", *Ecological Modelling*, Vol. 222 No. 11, pp. 1888-1896.

Schwartz, C., Ruiters, R., Weinmann, M. and Klein, R. (2013), "WebGL-based streaming and presentation of objects with bidirectional texture functions", *Journal on Computing and Cultural Heritage*, Vol. 6 No. 3, pp. 1-21.

Schwertfirm, F., Mathew, J. and Manhart, M. (2008), "Improving spatial resolution characteristics of finite difference and finite volume schemes by approximate deconvolution pre-processing", *Computers & Fluids*, Vol. 37 No. 9, pp. 1092-1102.

Singh, V., Dubey, R., Panigrahi, P.K. and Muralidhar, K. (2012), "An educational website on interferometry", *Proceedings of 2012 IEEE International Conference on Technology Enhanced Education (ICTEE), Kerala, 3-5 January*, pp. 1-10.

Teh, S. and Koh, H. (2013), "Mobile apps for water quality simulation", *Proceedings of International Conference on Teaching, Assessment and Learning for Engineering (TALE), Bali, 26-29 August*, pp. 182-187.

Tonti, E. (2014), "Why starting from differential equations for computational physics?", *Journal of Computational Physics*, Vol. 257 No. 2, pp. 1260-1290.

Walker, J.D. and Chapra, S.C. (2014), "A client-side web application for interactive environmental simulation modelling", *Environmental Modelling & Software*, Vol. 55, pp. 49-60.

Weibe, A.Y. and Huisinga, W. (2011), "Error-controlled global sensitivity analysis of ordinary differential equations", *Journal of Computational Physics*, Vol. 230 No. 17, pp. 6824-6842.

Weng, W.C. (2011), "Web-based post-processing visualization system for finite element analysis", *Advances in Engineering Software*, Vol. 42 No. 6, pp. 398-407.

Wriedt, T. (2008), "Mie theory 1908, on the mobile phone 2008", *Journal of Quantitative Spectroscopy and Radiative Transfer*, Vol. 109 No. 8, pp. 1543-1548.

Corresponding author
Jacques Duílio Brancher can be contacted at: jacques@uel.br