An Equation-Discovery Approach to Earthquake-Ground-Motion Prediction

Štefan Markič, Vlado Stankovski

Faculty of Civil and Geodetic Engineering, University of Ljubljana, Slovenia, Jamova cesta 2, SI-1000 Ljubljana

Abstract

In active seismic regions an earthquake’s peak ground acceleration (PGA) is required information when designing a building. In this study we employ the state-of-the-art, Lagrange, equation-discovery system to induce an equation that is suitable for modeling the PGA and investigate its applicability. In contrast to traditional modeling techniques the Lagrange system does not presume the structure of the equation and then identify the parameter values; instead, it finds the equation’s structure as well. From the large amount of background knowledge on earthquake engineering we formalize a context-free grammar, which is then used as a guideline for the equation-building procedure. The PF-L data set used for the experiments is taken from the study of Peruš and Fajfar (2010), which is based on the data sets of Chiou et al. (2008) in the project Next Generation Attenuation of Ground Motion and the study of Akkar and Bommer (2010). The best model derived from the grammar is then quantitatively and qualitatively evaluated and compared. The presented results support the proposal to use an equation-discovery tool as an aid to the PGA modeling work and to potentially contribute new knowledge to the field of earthquake engineering.

Keywords: Equation Discovery, Ground Motion Prediction Equations, Peak Ground Acceleration, Lagrange

1. Introduction

An earthquake is a natural phenomenon that manifests itself as a violent, rapid, earth tremor and happens unexpectedly, without prior notice. Strong earthquakes usually cause a lot of difficulties for people and communities; hence, the engineer’s task is to properly design a structure, bearing in mind that a devastating earthquake could occur during its lifetime. In the earthquake engineering domain, the correspondence with physical reality must be taken as the strongest criterion for the acceptability of the developed models along with the estimated prediction accuracy. The ground-motion prediction equations (GMPEs) or attenuation relations, the common name that was used for them (Douglas (2003)), are some of the key elements used by engineers to estimate a possible earthquake load at the site of a structure.

One of the ground-motion parameters is the peak ground acceleration (PGA), the prediction of which is the focus of the present study. More than 250 articles concerning PGA modeling have been published over the past 50 years, which means the area has been well investigated (see Douglas (2011)). Traditionally, the PGA is modeled as a single mathematical formula based on an author’s knowledge about the problem. The parameters included in such a formula are then fitted to the data by using a regression analysis for the prediction accuracy. Consequently, the resulting models are based on various assumptions and data sets and differ significantly in qualitative terms as well as quantitatively. Equation (1) from the study of Akkar and Bommer (2010) is presented here for illustrative purposes and can be described as a typical example of a GMPE.

The variables used in equation (1) are: i) the PGA in [cm/s²]; ii) the moment magnitude \( M_w \); iii) the Joyner-Boore distance \( R_{bb} \) in [km]; iv) the average soil shear-wave velocity in the upper 30 meters of soil underneath the observation spot \( V_{s,30} \) in [m/s]; and v) the faulting mechanism \( F \) (Akkar and Bommer (2010)).

Recently, researchers involved in earthquake engineering have experimented with new approaches when predicting the PGA that do not assume an equation form and have drawn different conclusions. Peruš and Fajfar (2010) used a conditional average estimator (CAE) method, which in contrast to conventional approaches does not make any \( a \ priori \) assumption, and found this method to be a simple but powerful tool, especially in the research environment. Kuehn et al. (2011) used Bayesian networks and concluded that the model they obtained is the maximum \( a \ posteriori \) model; i.e., the most probable model given the data. Gandomi et al. (2011) used multi expression programing (MEP), a machine-learning technique, and found that the generated models predict better than, or comparable with, the previously published regression-based models and, in their opinion, provide relatively simple equations, as opposed to the more complicated models from the Next Generation Attenuation (NGA) project. In summary, the use of non-conventional methods has so far concentrated on improving the prediction results.

With the development of computers a new scientific area was founded, where authors propose machine algorithms that try to imitate learning as an important human property. In equation discovery (ED), a sub-area of machine learning, the algorithms try to find a proper equation formulation that best fits a given data set. All ED systems use some kind of language bias that limits the hypothesis space, which is the space of all the
possible equations constructed from a given set of operators, functions and variables. Such a space is usually infinite, and is therefore restricted by the means of the algorithm. The state-of-the-art, Lagrange, ED system used in this study employs a declarative bias in the form of a context-free grammar (CFG) to limit the hypothesis space, which is given as input information to the system (Todorovski and Dzeroski (1997)). With such a formalism, domain knowledge can be easily provided to the ED system and so guide it toward the expected equation formulations.

Because of the fact that almost all GMPEs take the form of equations, the use of an ED system as an aid in earthquake-engineering design studies may come as a natural choice. Our investigation revealed that ED systems have not been used in earthquake engineering, to the best of our knowledge. Therefore, a specific goal of the present study is to propose a method for using the Lagrange system when modeling the PGA, which is used as a case study because of a particularly large domain knowledge. Bearing in mind the extensive expert requirements when modeling GMPEs, a careful investigation of the ED system is necessary before its usage for modeling the PGA is proposed. Moreover, it is necessary to appropriately incorporate the existing domain knowledge into the ED process, because the experimental set-up itself, if correctly designed, has the potential to yield high-quality results. With the system’s heuristic or exhaustive exploration of defined hypothesis space it is possible to investigate thousands of equation formulations and based on quantitative criteria, such as the mean squared error (MSE) and qualitative criteria like physicality, select the best equation. This procedure is crucial in order that the proposed ED method gains acceptance within the earthquake-engineering community. Fortunately, as we had access to powerful distributed-computing infrastructures, in our experiments all the calculations were pushed to their limits. The goal of this study was also to compare the results obtained with already existing GMPEs.

The rest of the paper is organized as follows. In Section 2 the Lagrange ED system and its input parameters along with the CFG and the data-set requirements are thoroughly explained. We describe the whole process of the application of the Lagrange ED system to the problem of predicting the PGA in Section 3. Descriptive tables and figures showing the results and the best equation found, together with their explanations, are presented in Section 4. We conclude this study with Section 5, where we discuss and evaluate the presented results and provide some ideas for future research.

2. Lagrange

Equation discovery (ED) is an emerging machine-learning discipline that is closely related to system identification, inductive logic programming and genetic programming. About a dozen ED systems have been described in the literature, among which Bacon of Langley et al. (1987), Lagrange of Todorovski and Dzeroski (1995) and Lagrange of Todorovski and Dzeroski (1997) have received particular attention in the machine-learning community. The Lagrange system seems to be the most suitable for the PGA modeling task at hand, particularly because it uses CFG to specify prior knowledge. For this reason it was selected and used in the present study. The Lagrange ED system has already been applied to several scientific fields of interest. The first experiments with the Lagrange system were made in the area of ecological modeling, e.g., the prediction of phytoplankton growth in the studies of Todorovski et al. (1998) and Kompare et al. (2001). Todorovski and Dzeroski (2001) also applied it to population dynamics, predicting the behavior of prey-predator dependence and found that the integration of specific domain knowledge in the CFG significantly improved the prediction results. Some of the latest applications of the Lagrange system include discovering mathematical models of a mechanically ventilated lung by Ganzert et al. (2010) and the financial forecasting of commodity prices from the London Metal Exchange by Alzaidi and Kazakov (2011).

The problem given to the Lagrange system is denoted with two input files: a data set D and a CFG (Todorovski and Dzeroski (1997)). The input data D = {M, vD, W} consists of one or more tables of measurements or records M of variables W = {v1, v2, . . . , vD}. Among the variables, one must be selected as a dependent variable vD ∈ W. So as to make it easier to understand the grammar building described in the following paragraphs, let us assume that we want to design a CFG that will be able to generate the first three terms of equation (1).

A tuple $CFG = \{N, T, P, S\}$ prescribes the syntax of the right-hand side of an equation. It contains finite disjunctive sets of non-terminals (N) and terminals (T). The Lagrange system uses a special non-terminal symbol V ∈ N, which denotes any of the independent variables from the input data set $W \backslash v_D$; otherwise, any symbol can be used to denote a non-terminal. The

$$ log_{10}(PGA) = \frac{1.04159 + 0.91333 \cdot M_w - 0.08140 \cdot M_w^2}{+(-2.92728 + 0.28120 \cdot M_w)} \cdot \log_{10}\sqrt{R_{1h}^2 + 7.86638^2} \begin{cases} 0.08753 & \text{if } V_{s,30} < 360 \\ 0.01527 & \text{if } 360 \leq V_{s,30} < 800 \\ 0 & \text{if } 800 \leq V_{s,30} \end{cases} \begin{cases} -0.04189 & \text{if } F = \text{normal} \\ 0 & \text{if } F = \text{strike-slip} \\ 0.08015 & \text{if } F = \text{reverse} \end{cases} \tag{1} $$

The Lagrange system release 2.2 used in this study is available as open-source software at URL: http://www-ai.ijs.si/~ljupco/ed/lagrange.html
set \( T \) consists of all the independent variables \( v_j \in W \setminus v_d \) and a special symbol \( \text{const} \), whose syntax in the Lagrange system is as follows:

\[
\text{const}[\text{name} : \text{lowest value} : \text{starting value} : \text{highest value}] \tag{2}
\]

In the case of our example, the set of non-terminals is \( N = \{ \text{Linear}, \text{Term}, V \} \) and the set of terminals is \( T = \{ \text{M}_w, \text{const} \} \).

The productions \( P = \{ P_1, P_2, \ldots, P_n \} \) denote the grammatical rules that relate the non-terminals among themselves (recursion is possible) and to the terminals. The standard form of a production \( P \) is \( A \rightarrow \alpha \), where \( A \in N \), \( \alpha \in N \cup T \) and the operators or functions used are (already or user-) defined in the programming language C. If we want to reference to an explicit variable in a grammar, we must use \( \text{variable}_\text{in} \) in front of its name. However, the productions for \( V \) are added to the grammar automatically during the run-time, as the Lagrange system reads the variables’ names from the input data file, i.e., \( \forall v_j \in W \setminus v_d : V \rightarrow \text{variable}_\text{in} v_j \in P \). We use the annotation with the logical or operator \( A \rightarrow a_1 | a_2 \ldots | a_n \) for productions \( A \rightarrow a_1, A \rightarrow a_2, \ldots, A \rightarrow a_n \). In order to derive the first three terms of equation (1), only addition and multiplication are needed, which are both already predefined in C.

Finally, \( S \in N \) is a special, non-terminal symbol, from which the derivation of the expressions starts. In the case of our example, it is denoted by the symbol \( \text{Linear} \).

The definition of the developed example polynomial grammar is provided in Table 1. Its first four productions provide enough syntax to build the desired polynomial. The first two productions succeeding the non-terminal symbol \( \text{Linear} \) generate any number of terms. The second two productions succeeding the non-terminal symbol \( \text{Term} \) derive these terms into degrees. Note that the last production \( V \rightarrow \text{variable}_\text{in} \text{M} \) is automatically added by the Lagrange system during the run-time and must not be manually included in the grammar, but it is added to Table 1 for completeness.

During the derivation process we continuously apply productions to all the non-terminals until all the symbols in the expression are terminals. This process can be best depicted with the growth of a derivation tree, as can be seen in Figure 1 for our example. When we include at least one recursive production in the CFG (e.g., Table 1, the first production), the hypothesis space and the length of the derivation process are infinite. Therefore, we bound the complexity of expressions with the maximum-tree-depth parameter \( d \), urging the Lagrange system to ensure that all the symbols at the prescribed depth are terminals. In our example, the appropriate value would be \( d = 4 \), as shown on the left-hand side of Figure 1.

At the end of the derivation process, the generated expression from the CFG can be obtained by reading all the end leaves of the derivation tree from left to right. The expression generated with the derivation tree on Figure 1 is:

\[
\text{const}[] + \text{const}[] \cdot M_w + \text{const}[] \cdot M_w \cdot M_w \tag{3}
\]

Such an expression contains one or more special terminal symbols \( \text{const} \in T \). A non-linear fitting method (downhill simplex or Levenberg-Marquardt algorithm) is used by the Lagrange system to determine the values of these parameters that best fit the given data set. With them the Lagrange system minimizes the value of the MSE function, which is calculated according to the formula:

\[
MSE = \frac{1}{n} \sum_{i=1}^{n} (v_{d,i,\text{measured}} - v_{d,i,\text{predicted}})^2 \tag{4}
\]

where \( n \) is the number of records in the data set and \( v_{d,i,\text{measured}} \) and \( v_{d,i,\text{predicted}} \) are the measured and predicted values of the dependent variable, respectively. For the purpose of this study we also used other variables, which are discussed in detail in subsection 4.1.

The Lagrange system also provides options for the manipulation of the search process. The user can decide whether to search for an ordinary or differential dependence, i.e., \( v_d = E \) or \( \dot{v}_d = E \), respectively, where \( E \) is an expression derived from the CFG. Other than the possibility of an exhaustive search strategy, where all possible equation structures in the limited hypothesis space are tried, it is also possible to use a heuristic beam search strategy. This starts with a number of expressions (e.g., 20) and derives all their first successors, then saves the same number (i.e., 20) of those with the lowest MSE among all of them and repeats. The user can set the number of equations the Lagrange system saves in each step with the value of the input parameter \( b \), also referred to as the beam width. Three stopping criteria are implemented: i) when all of the possible equation structures have been derived and tested; ii) when the Lagrange system finds the first expression with a lower MSE function

<table>
<thead>
<tr>
<th>Depth</th>
<th>Derivation tree</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Linear</td>
</tr>
<tr>
<td>1</td>
<td>Linear + Term</td>
</tr>
<tr>
<td>2</td>
<td>Linear + Term</td>
</tr>
<tr>
<td>3</td>
<td>Term + Term</td>
</tr>
<tr>
<td>4</td>
<td>const[...]</td>
</tr>
</tbody>
</table>

Table 1: Polynomial context-free grammar

<table>
<thead>
<tr>
<th>Production</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear → Linear + Term</td>
<td></td>
</tr>
<tr>
<td>Linear → Term</td>
<td></td>
</tr>
<tr>
<td>Term → Term + V</td>
<td></td>
</tr>
<tr>
<td>Term → const[[-400:0.1:400]</td>
<td></td>
</tr>
<tr>
<td>V → variable_M</td>
<td></td>
</tr>
</tbody>
</table>
than the one prescribed by the user; or iii) when a user-defined CPU time is exceeded (Todorovski and Džeroski (1997)).

Having analyzed the Lagrange algorithm, which is needed to properly design an ED task, the following section focuses on the application of the Lagrange system to the earthquake-engineering problem of forecasting the PGA, i.e., the implementation of domain knowledge.

3. PGA modeling

Natural phenomena and various systems are frequently modeled on the basis of collected data. In such studies, the common goal is to capture the relationships underlying the data. Since the resulting models need to be evaluated and validated, the expert knowledge must be available in an easily understood form. Typically, the simplest and most useful relationships for engineers are mathematical relations. Therefore, when designing the experimental set-up the following aspects were taken into consideration: i) the selection of the data set; ii) the definition of the CFG; and iii) the physicality of the equations. The data set, the grammar that was used to induce the equations, as well as our choice of values for all the input parameters are described in the following subsections.

3.1. Data selection

Before running an experiment to explore a real-world problem a careful choice of data must be made as a prerequisite for obtaining good results. In the engineering domain this means that the gathering, filtering and selection of appropriate data must be made on the basis of a clear vision of the problem itself. The selection of data usually plays an important role when inducing a new GMPE and can depend on a specific purpose.

The significant seismological aspects that influence the ground-motion parameters are considered to be the following: i) the source; ii) the travel path; and iii) the site effects. The source effect can basically be described by the level of stress drop in an earthquake event, the static measure of the released energy in an earthquake - magnitude, the depth of the epicentre and by the mechanism of faulting. A variety of magnitudes are used in the literature, e.g., the local magnitude (also called the Richter magnitude), the surface-wave magnitude, the moment magnitude and many more. The site effect is most commonly characterized by the soil’s shear-wave velocity; however, it can also be considered in a generic way by using site categories (e.g., rock, stiff soil and soft soil in the study of Akkar and Bommer (2010)). The travel-path effect is generally represented by the distance of the observation site from the fault, and there are many defined, e.g., the epicentral distance, the hypocentral distance, the rupture distance, the Joyner-Boore distance and many more (see Douglas (2003) for a detailed description).

In the past, earthquakes have been systematically recorded and the data assembled for research purposes by institutes around the world, e.g., the Pacific Earthquake Engineering Research Center gathered 3551 strong-motion earthquake recordings in a large database for its NGA project (Chiou et al. (2008)). For various reasons not all of the available data is used for experiments. In many cases only earthquakes located within a country or a tectonic region are taken into account, or all the aftershocks are excluded; thus, reducing the initial data set, as is necessary for the particular purpose. For example, by following the data-selection process of Abrahamson and Silva (2008), Kuehn et al. (2011) started with the full NGA data set of 3551 earthquake recordings and reduced it to 3342 by selecting only the representatives of free-field conditions and excluding some records from the Chi-Chi-sequence, duplicate records and those records missing a horizontal component. Sometimes, the researchers use additional non conventional independent variables, e.g., the variable depth-to-top of rupture $Z_{top}$ in the study of Abrahamson and Silva (2008).

Throughout the literature there are a lot of combinations of variables used to determine GMPEs. As in Peruš and Fajfar (2010) and Akkar and Bommer (2010), the independent variables we use in this study are: i) the moment magnitude $M_w$, which as stated by Chiou and Youngs (2008) is the simplest measure for correlating the amount of energy released in an earthquake; ii) the source-to-site Joyner-Boore distance $R_{jb}$ in [km]; iii) the style-of-faulting $F$; and iv) the average soil shear-wave velocity in the upper 30 meters of soil underneath the observation spot $V_{s,30}$ in [m/s], as they enable an easier comparison with other studies and are the most modern choice. The PGA parameter is the dependent variable, measured in [g]-units, and is defined as the geometrical average of both horizontal components. Previous studies have shown that strong-motion amplitudes are log-normally distributed (e.g., see Douglas and Smit (2001)), which we incorporated into our study by stating the ED problem as follows:

$$\ln(\text{PGA}) = f(M_w, R_{jb}, V_{s,30}, F) \quad (5)$$

Peruš and Fajfar (2010) assembled two new databases for their experiments with the CAE method from which the larger PF-L database is also used in this study. It is a union of databases used in the following studies, which all root from Europe: Abrahamson and Silva (2008), Boore and Atkinson (2008), Campbell and Bozorgnia (2008), Chiou and Youngs (2008), Idriss (2008) and Akkar and Bommer (2010). The whole data set consists of 3550 recordings and includes aftershocks. All of the variables are continuous, except for the style-of-faulting variable $F$. It is defined as a non-dimensional parameter that has a value of $i) F = 0$ for normal faults; ii) $F = 0.5$ for strike-slip faults; and iii) $F = 1$ for reverse faults (Peruš and Fajfar (2010)). The descriptive statistics of the PF-L data set are presented in Table 2. For illustrative purposes Figure 2 presents the moment magnitude $M_w$ vs. the Joyner-Boore distance $R_{jb}$ and the faulting mechanism $F$ data distribution. From Table 2 and Figure 2 can be seen that the data set is very unbalanced, especially for high magnitudes and short distances.

Based on the problem statement in equation (5), the original PF-L data set was preprocessed by converting the actual values of the PGA into their logarithmic values. This arrangement resulted in much faster calculations and better performance of the parameter-fitting algorithms implemented in the Lagrange
system. The whole data set was randomly split 10 times into the learning and testing sets in a 90 % to 10 % proportion, with the purpose of a 10-fold cross validation. That is, the testing set will not be seen by the algorithm during the ED process, but will be used as “future recordings” that we are trying to forecast with the new formula and the models selected for comparison.

3.2. Developing Grammar for PGA

In this section we analyze various aspects of the works related to the problem of modeling the PGA. The design of a CFG that would incorporate the existing domain knowledge was one of the most difficult tasks undertaken. In the course of this study it was necessary to systematically examine all the existing equation structures for the PGA that may form the basis for the specification and use of existing domain knowledge in the ED process and specify the grammar productions. Here, we provide a summary of the equation structures that were considered as information and prior-knowledge sources for the specification of a new CFG.

A worldwide summary of all the found GMPEs that take the form of an equation, published until 2010 with a detailed explanation of the derivation of each equation, can be found in Douglas (2011). We observed that each of the studies made slightly different assumptions and/or used modern modeling approaches, therefore the existing PGA models vary significantly in terms of their complexity and the use of various rules. Some equation structures have over 30 elements and are difficult to explain to non-experts, e.g., the model derived in the study of Campbell and Bozorgnia (2008). However, a careful examination reveals that each equation element (partial function) is based on some physical assumption and the authors’ observations and knowledge about the problem. On the other hand, some authors started by specifying a simple functional form and added complexity to the equation gradually, by observing the statistical significance of each modification and its influence on the prediction error, e.g., in the example of Cauzzi and Pacioli (2008), where the authors experimented by including i) an anelastic decay term; ii) a quadratic magnitude dependence; and iii) a magnitude-dependent decay term, to find out that none of these additional equation elements contributes significantly to the prediction accuracy of the initial PGA equation.

Along with this ED process, the strongest criterion for the selection of the best possible equation must be its correspondence with the natural phenomenon. As a baseline, in reality the parameter PGA represents the maximum ground acceleration that happens during an earthquake event and can never take negative values. For example, if a polynomial function is assumed as a model for the PGA, it is difficult to effectively ensure, by means of the CFG, that the induced equations will take only positive values. However, as we defined the problem statement in equation (5), the calculated PGA values will never take negative values because the antilogarithm of the right-hand side is always positive.

For the purpose of this study, we defined one grammar, conveniently named Katja for reference (see Table 3). It was designed in order to take into account the prior knowledge at a high level of detail. The actual productions of this grammar were defined by systematically studying the formulae designed by earthquake engineers over the past 50 years collected by Douglas (2011).

The Katja CFG can be used to generate almost all the existing simple formulae, even those that have split a variable in classes and require the use of if-clauses. Its use in the Lagrange system first leads from the root symbol Eq to a number of non-terminal functions (see Table 3). These functions are named FM, FR, FV and FF after the dependence they model, i.e., $f(M_e)$, $f(R_b)$, $f(V_{s,30})$ and $f(F)$, respectively. This trend of explicitly dividing the effects among variables and summarizing them is seen in the latest studies, e.g., in the study of Campbell and Bozorgnia (2008). In their study, however, the $f_{dis} = f(M,R_{UP})$, which we have also incorporated in the Katja grammar with the production $FR \rightarrow Ko \cdot FM1 \cdot FR1$, i.e. $FR$ can also be $f(M_e,R_b)$. Each of these functions can then be succeeded with their own special sub-functions, which have been gathered during the process of a literature review. Note that these are not all of the possibilities seen in the literature, but just the most often modeled dependencies of the variables used in this study. In the Katja CFG we incorporated the possibility for the functions $f(V_{s,30})$ and $f(F)$ to be zero with the productions $FV1 \rightarrow K0$ and $FF \rightarrow K0$. The parameter $const$ used in the productions for $FV1$ is limited to values between 0 and 4000 with a starting value of 800, which is the division value between the rock and the soil classes.

It is very beneficial that the Lagrange system makes it possible to build productions with all the operators or functions defined in the C programming language (Todorovski and Džeroski (1997)). In order to employ if-structures in the grammar, the functions $if$ and $ife$ were defined that allow a comparison of two values for their smallness or equality, respectively. For their definition see the top of Table 3. We also defined 11 auxiliary productions to improve the readability of the whole grammar, even though including such productions in the grammar is not obligatory. Four productions ($Ma, Ra, Vs, Fa$) lead from the variables’ symbols to the variables’ addresses known to the Lagrange algorithm. The remaining 7 productions lead to constant parameters, and all except for the parameter $Ko$ are set to

<table>
<thead>
<tr>
<th>statistic</th>
<th>$PGA\ [m/s^2]$</th>
<th>$M_e$</th>
<th>$R_b\ [km]$</th>
<th>$V_{s,30}\ [m/s]$</th>
<th>$F$</th>
</tr>
</thead>
<tbody>
<tr>
<td>minimum</td>
<td>0.0012</td>
<td>4.27</td>
<td>0</td>
<td>116.4</td>
<td>0</td>
</tr>
<tr>
<td>average</td>
<td>0.0939</td>
<td>6.25</td>
<td>57.1</td>
<td>420.5</td>
<td>0.74</td>
</tr>
<tr>
<td>maximum</td>
<td>1.6615</td>
<td>7.90</td>
<td>365.1</td>
<td>2016.1</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 2: Average, minimum and maximum values of all the variables used in this study
a single value. Note that numbers cannot be used in the grammar explicitly. The parameter(s) $\text{const}$ succeeding Ko, which are fitted to data after the derivation, are limited to values between -100 and 100, as those are greater than the largest values seen in the literature.

The use of the Katja CFG makes it possible to limit the space of possible equations to only those that are the most plausible according to the studied domain knowledge. Thus, the grammar is an important instrument in the research and experimentation process that directs the ED system towards more appropriate equation models.

### 3.3. Input Parameters

After defining the Katja CFG to be used for the experiments, it was also necessary to properly set the various parameters that control the exploration of the hypothesis space of possible equations. The following is a brief overview of these parameters.

**Tree Depth $d$.** The parameter maximum tree depth, $d$, limits the depth of the production tree. The algorithm implemented in the Lagramge system evaluates the equations that are composed of only terminals at the prescribed depth, as described in Section 2. With increasing $d$, the hypothesis space for the Katja grammar increases by approximately a factor of 75, as can be seen in Table 4 in the first and second columns. We were able to run an exhaustive search with $d = 5$, which makes it possible to generate almost all the existing simple equation formulations taken from the study of Douglas (2011) using the Katja grammar. The other columns in Table 4 are the maximum and minimum length of an expression, the maximum number of terminal symbols $\text{const}$ in an expression and the total number of productions applied when deriving the maximum-length expression. The value $-1$ means that the property cannot be calculated.

**Beam Width $b$.** The Lagramge system can operate in two search modes, i.e., heuristic and exhaustive; the latter is the default option. Ideally, if time permits, the Lagramge system can gradually generate and evaluate all the possible equations and find the best one according to the chosen criterion. As is clear from Table 4, an exhaustive search beyond $d = 5$ is not a good option, since at $d = 6$ the Lagramge system would produce $\approx 7.4 \cdot 10^6$ expressions with the Katja CFG, i.e., if the derivation and fitting of one equation takes only one second to calculate, the whole procedure takes three months to complete. For this reason it was necessary to use the heuristic search algorithm by setting the parameter beam width $b$ when exploring deeper in the hypothesis space. Its value determines the number of best equations that the Lagramge system will retain at each step of the search process. This makes it possible to observe the influence of the value of the parameter $b$ on the prediction accuracy. Increasing $b$ is recommended as long as the prediction accuracy increases.

Many times when experimenting with various grammars, the Lagrange system exhausted the available memory due to a too big value of $b$. The memory consumption depends largely on the choice of the CFG, and for the Katja grammar presented in this article, it was practically impossible to experiment with $b = 100$ or more because the Lagramge system exhausted all the available virtual memory, although we also aimed at values of 200, 500 and more. This is why the values chosen for experiments are $b \in \{1, 2, 5, 10, 20, 50\}$.

**Parameter Fitting Restarts $m$.** Each time an equation is generated it contains many terminals $\text{const}$ (see Table 3, production Ko) and is tested against the input learning data set, which involves the use of parameter-fitting methods. These methods are likely to catch in local minimums; therefore, we can determine the number of restarts with the parameter $m$, and it was also necessary to observe its influence on the results. For its values we chose $m \in \{1, 10, 100\}$, as we did not expect major differences with smaller steps or with greater values.

### 3.4. Running Experiments

Bearing in mind the computational complexity, we speeded up the calculation by relying on our Slovenian National Grid Infrastructure and the experience gained in the course of the European projects DataMiningGrid, InteliGrid and the ongoing mOAIC Cloud project. This resulted in systems that are used for the development and management of distributed applications as in Stankovski et al. (2008a) and Stankovski et al. (2008b). As a result, it was possible to make calculations that would normally take many years on a single computer in just a few days.

In the course of this study, various grammars, operating modes and parameters for the Lagrange system were investigated. In grid-computing terminology, a series of experiments that represent independent computational tasks is also referred to as a multi-job. In most cases the multi-jobs contained 180 experiments: $m \in \{1, 10, 100\}$, $b \in \{1, 2, 5, 10, 20, 50\}$, and each calculation was performed for one learning data set out of the 10 random splits (90 % for learning and 10 % for testing data). The diagram of the experimental set-up can be seen in Figure 3. While experimenting, we pushed the required memory for running the Lagrange system to its limits, also on the Slovenian National Grid Infrastructure - sometimes approximately 200 GB of virtual memory were required to run a single instance of the Lagrange system.

### 4. Results

In the following subsections the obtained results of the final series of experiments corresponding to the Katja grammar are presented, both in terms of the different quantitative and qualitative criteria, i.e., the best equation that was obtained in the experiments.

#### 4.1. Quantitative analysis

The Lagramge system evaluated each derived equation with a parameter-fitting method by minimizing the MSE on the learning data set. The equations were then sorted in ascending order according to the calculated MSE. The best equation found which reached the lowest MSE on its learning data set is (6). It was found when running the Lagramge system with the heuristic beam search, $d = 8$, $b = 100$, $m = 50$ for the first learning
double ifl(double val, double comp, double t, double f) {
    return((val < comp) ? t : f);
}

double ife(double val, double comp, double t, double f) {
    return((val == comp) ? t : f);
}

Eq -> Ko + FM + FR + FV + FF
FM -> (FM + Ko · FM1) | Ko · FM1
FM1 -> Ma | pow(Ma, K2) | pow(Ma + Ko, K2) | pow(Ma + Ko, const[._1:1.5:5]) | exp(Ko · Ma)
FR -> Ko · FM1 · FR1 | FR + Ko · FR1 | Ko · FR1
FR1 -> ln(Ra + Ko) | ln(Ra + Ko · FM1) | ln(pow(Ra, K2) + Ko) |
     | ln(pow(Ra, K2) + Ko · FM1) | pow(Ra + Ko, -K1) | pow(Ra + Ko, -K2)
FV -> FM1 · FV1 | FV1
FV1 -> ifl(Vs, K180, Ko, ifl(Vs, K360, Ko, ifl(Vs, K800, Ko, K0))) | K0 |
     | Ko · ln(Vs / const[._0:800:4000]) | ifl(Vs, const[._0:800:4000] , Ko, K0)
FF -> ife(Fa, K1, Ko, ife(Fa, K0, Ko, K0)) | K0
Ma -> variable_Ma
Ra -> variable_Ra
Vs -> variable_Vs
Fa -> variable_Fa
Ko -> const[._-100:0.1:100]
K0 -> const[._0:0:0]
K1 -> const[._1:1:1]
K2 -> const[._2:2:2]
K180 -> const[._180:180:180]
K360 -> const[._360:360:360]
K800 -> const[._800:800:800]

<table>
<thead>
<tr>
<th>d</th>
<th>no.equ.</th>
<th>max.len.</th>
<th>min.len.</th>
<th>max.no.const</th>
<th>deriv.len.</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>25</td>
</tr>
<tr>
<td>4</td>
<td>960</td>
<td>84</td>
<td>18</td>
<td>21</td>
<td>39</td>
</tr>
<tr>
<td>5</td>
<td>100800</td>
<td>120</td>
<td>26</td>
<td>29</td>
<td>55</td>
</tr>
<tr>
<td>6</td>
<td>~7.4·10^6</td>
<td>156</td>
<td>43</td>
<td>37</td>
<td>71</td>
</tr>
<tr>
<td>7</td>
<td>~5.2·10^8</td>
<td>192</td>
<td>60</td>
<td>45</td>
<td>87</td>
</tr>
<tr>
<td>8</td>
<td>~3.7·10^10</td>
<td>228</td>
<td>77</td>
<td>53</td>
<td>103</td>
</tr>
</tbody>
</table>
data set with the Katja CFG. For this equation the calculated MSE on the first learning data set is 0.3828, while the calculated MSE for the corresponding testing data set is 0.3823.

For the purpose of observing the $b$ and $m$ parameters’ influence on the results, the equation with the minimum MSE on the learning data set was selected in each trial of the multi-job. Then, the MSE was also calculated on the testing (not previously used) data set. In Table 5 we present the average MSE on the testing data set for each 10-fold cross validation and for all combinations of the $b$ and $m$ values. Table 6 contains the corresponding standard deviation for the MSE on the testing data sets. It is clear that the standard deviation is relatively low, especially for large $b$ values, which means that the obtained results for the MSE can be compared among each other. The calculated MSE decreases by 15% at $b = 20$ when compared to that at $b = 5$, which supports the use of big $b$ values in future experiments. It is a general observation that the MSE does not decrease significantly with higher values of $m$.

Existing studies have used various data sets and variables, making it difficult to systematically compare the obtained mathematical models. Therefore, we provide additional criteria, following the study of Alavi and Gandomi (2011):

- correlation coefficient $R$

$$ R = \frac{\sum_{i=1}^{n} (m_i - \bar{m}) \times (p_i - \bar{p})}{\sqrt{\sum_{i=1}^{n} (m_i - \bar{m})^2 \times \sum_{i=1}^{n} (p_i - \bar{p})^2}} \quad (7) $$

- root mean-squared error RMSE

$$ RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (m_i - p_i)^2} = \sqrt{MSE} \quad (8) $$

- mean absolute error MAE

$$ MAE = \frac{1}{n} \sum_{i=1}^{n} |m_i - p_i| \quad (9) $$

where $m_i$ and $p_i$ are the measured and predicted values of the PGA of the $i^{th}$ record with an average of $\bar{m}$ and $\bar{p}$, respectively, and $n$ is the number of records. Note that a higher $R$ value and lower RMSE and MAE values indicate a more precise model. As marked in equation (8), the RMSE is the square root of the MSE, defined with equation (4). The average and standard deviation on all 10 testing data sets of these criteria for three models are shown in Table 7: i) in the first column for equation (6), discovered by Lagramge; ii) in the second column for equation (1) proposed by Akkar and Bommer (2010); and iii) in the third column for the CAE method of Peruš and Fajfar (2010). According to the calculated values, this method performs better than the method of Akkar and Bommer (2010), but worse than the CAE method of Peruš and Fajfar (2010). Note, however, that the CAE method does not provide a formula that could be used in the engineers’ daily work.

Figure 4 shows predicted vs. measured values of the PGA for the whole database. It is clear that the points are scattered on both sides of the ideal-fit line $y = x$ with a majority slightly below it. Also, the maximum calculated prediction of $PGA \approx 0.85g$ can be seen. Six measured values for $PGA$ exceed the value of $1g$ and are not depicted in the graph.

### 4.2. Qualitative Analysis

When we observe the very structure of the found equation (6), we can see that two constants reached their maximum value, one in the denominator with $R_{by}$ at value 100 and the other in the $V_{s,30}$ term at value 4000. In the equation there are three $M_w$, one $R_{by}$, one $M_w$-$R_{by}$, one $V_{s,30}$ and one $F$ dependencies. The Lagramge system found that the $V_{s,30}$ and $F$ terms are not negligible, even though the algorithm could choose such productions. The $V_{s,30}$ term was found as a continuous dependency, rather than divided into classes.

Equation (6), which was found during our experiments with the Lagramge system is compared with formulae designed in the course of the NGA project, published by Abrahamson and Silva (2008), Boore and Atkinson (2008), Campbell and Bozorgnia (2008), Chiou and Youngs (2008) and Idriss (2008) and with the formula designed by Akkar and Bommer (2010). Equation (6) is labeled Lagramge for convenience and its graph is aligned with the graphs corresponding to the formulae of these authors in Figures 5 and 6, drawn for magnitudes of 6 and 7, respectively. It is clear that equation (6) found by the Lagramge system models the PGA completely in the range of these models, although somewhat at the lower border.

The quantitative results are relevant, bearing in mind that the graphs of equation (6) found by the Lagramge system are well aligned with the graphs of other existing equations. In such a case the importance of the prediction accuracy on previously unknown testing data sets should not be underestimated.

For engineers it is important to know the application range of any newly proposed equation. A common observation is that any extrapolation of the developed models outside the boundaries of the used data-set range is to be avoided (e.g., see Bommer et al. (2010)). The majority of data lies between the values of $4.9 \leq M_w \leq 7.6$ and $0 \leq R_{by} \leq 200 \text{ km}$ (see Figure 2), therefore, according to the data, the newly induced equations could be applied within these borders. However, it is clear from Figure 7, that this equation models physically only up to $M_w = 7$, as the graph of the predicted PGA for $M_w = 8$ is physically not acceptable, because the PGA for higher magnitudes cannot be lower than that of lower magnitudes. This feature of the data set, however, was also found with the use of Bayesian networks in the study of Kuehn et al. (2011).

### 5. Conclusions

This study presents a new methodology for using ED methods in earthquake engineering and an application of the Lagramge ED system for modeling the PGA. In the field of earthquake engineering in a large number of studies published so far, conventional research methods are still in use. Machine-learning methods have been applied in just a few cases, e.g.,
Table 5: Average MSE on the testing data set for each 10-fold cross-validation split for the Katja grammar

<table>
<thead>
<tr>
<th>b</th>
<th>m = 1</th>
<th>m = 10</th>
<th>m = 100</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.597</td>
<td>0.596</td>
<td>0.595</td>
</tr>
<tr>
<td>2</td>
<td>0.595</td>
<td>0.594</td>
<td>0.593</td>
</tr>
<tr>
<td>5</td>
<td>0.593</td>
<td>0.593</td>
<td>0.590</td>
</tr>
<tr>
<td>10</td>
<td>0.548</td>
<td>0.535</td>
<td>0.522</td>
</tr>
<tr>
<td>20</td>
<td>0.516</td>
<td>0.518</td>
<td>0.513</td>
</tr>
<tr>
<td>50</td>
<td>0.510</td>
<td>0.510</td>
<td>0.509</td>
</tr>
</tbody>
</table>

Table 6: Standard deviation of the MSE $\sigma_{MSE}$ on the testing data sets for each 10-fold cross-validation split for the Katja grammar

<table>
<thead>
<tr>
<th>b</th>
<th>m = 1</th>
<th>m = 10</th>
<th>m = 100</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.041</td>
<td>0.041</td>
<td>0.040</td>
</tr>
<tr>
<td>2</td>
<td>0.042</td>
<td>0.042</td>
<td>0.040</td>
</tr>
<tr>
<td>5</td>
<td>0.042</td>
<td>0.042</td>
<td>0.040</td>
</tr>
<tr>
<td>10</td>
<td>0.041</td>
<td>0.025</td>
<td>0.032</td>
</tr>
<tr>
<td>20</td>
<td>0.030</td>
<td>0.031</td>
<td>0.029</td>
</tr>
<tr>
<td>50</td>
<td>0.029</td>
<td>0.030</td>
<td>0.029</td>
</tr>
</tbody>
</table>

Table 7: Calculated averages and standard deviations of the R, RMSE and MAE criteria on the testing data sets: i) of equation (6) developed by the Lagrange system; ii) of equation (1) developed by Akkar and Bommer (2010); and iii) of the CAE method proposed by Peruš and Fajfar (2010)

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>R</td>
<td>$R_{\sigma}$</td>
<td>0.827 0.014</td>
<td>0.802 0.015</td>
</tr>
<tr>
<td>RMSE</td>
<td>$RMSE_{\sigma}$</td>
<td>0.629 0.019</td>
<td>0.678 0.019</td>
</tr>
<tr>
<td>MAE</td>
<td>$MAE_{\sigma}$</td>
<td>0.491 0.017</td>
<td>0.528 0.014</td>
</tr>
</tbody>
</table>
MEP in the study of Alavi et al. (2011). Compared to MEP and other machine-learning techniques, our results indicate that the Lagrange system as an emerging algorithm deserves attention from the engineering communities for several reasons.

First, the use of a CFG, where we can include prior domain knowledge and guide the algorithm towards expected results, is very convenient, bearing in mind that mathematical formulae are frequently used in the engineers’ daily work.

Second, in our study, an extensive literature review revealed various equation structures for the PGA, which were modeled by productions of the designed grammar. The best equation found is in the range of existing NGA studies with respect to the qualitative criterion; however, equally importantly, the calculated quantitative criteria for the new equation are better than those obtained for the equation of Akkar and Bommer (2010). Since the CAE method of Peruš and Fajfar (2010) is obviously performing better, further improvements of the obtained formulae could be made.

Third, using shear computing power it is possible to formulate and fit a much greater range of equations, unlike conventional methods. For such reasons, grid-computing and the recently developed cloud-computing infrastructures and associated approaches (as in Stankovski et al. (2008b)) could be used to speed up the calculations and explore a large hypothesis space of possible equations.

The application range of equation (6) is according to the qualitative analysis $4.0 \leq M_S \leq 7.0$ and $0 \text{ km} \leq R_J \leq 200 \text{ km}$. The extrapolation outside the magnitude boundaries should be avoided and outside the distance boundaries made with great caution.

The results presented in this study suggest that ED systems should be regarded as a useful aid in engineering design, particularly because they are capable of exploring a much wider space of possible equation formulations, defined by a CFG.

### 5.1. Further Work

In future studies, possible gains in prediction accuracy could be obtained both by improving the CFG or the data set. The Katja grammar, presented in this study, cannot express all of the developed equations reported in Douglas (2011); therefore, its definition could be improved. It could also be specialized based on the designated use of the PGA equation, or include some other functions defined by various authors (e.g., min and max in the study of Chio and Youngs (2008)). A refined selection of the data set and/or the inclusion of more variables and/or the inclusion of more records could provide better prediction results and perhaps reveal not yet discovered knowledge about the problem of predicting the PGA.

The method itself could also be improved to be able to account for the inter- and intra-event variability, which is common investigation approach in earthquake engineering studies. Faster calculations could be obtained by the parallelization of the Lagrange algorithm.

This method may initiate a whole range of ED studies in the domain of earthquake engineering. The ground motion parameter PGA, modeled in this study, is not the commonly used intensity measure for structural design anymore; it is nowadays being replaced by the elastic spectral acceleration, the prediction of which could be the focus of future studies, because the Lagrange system shows good performance and results.

### Acknowledgements

The authors are grateful to Iztok Peruš and Peter Fajfar for fruitful discussions, for providing the PF-L data set, which was used for the experiments in this study and for the PGA graphs of various authors. Special thanks go to Ljupčo Todorovski for guidance when using his Lagrange ED system. The authors are also grateful to ARNES for making available the National Grid Infrastructure for our purposes. This research is partially funded by the European grant FP7-ICT-2009-5-256910 mOSAIC-cloud.eu.


Campbell, K.W., Bozorgnia, Y., 2008. NGA ground motion model for the geometric mean horizontal component of PGA, PGV, PGD and 5% damped linear elastic response spectra for periods ranging from 0.01 s to 10 s. Earthq. Spectra 24, 139–171.

Cauzzi, C., Faccioli, E., 2008. Broadband (0.05 to 20 s) prediction of displacement response spectra based on worldwide digital records. J. of Seismol. 12, 453–475.


Figure 2: Data distribution with respect to $M_w$, $R_{jb}$, and $F$

\[ \ln(PGA) = 4.57353 - 1.69293 \cdot M_w + 0.2417 \cdot M_w^2 \\ -6.67613 \cdot e^{-7.60198 \cdot M_w} - 0.00918368 \cdot e^{2.3707 \cdot M_w} \\ -1.67822 \cdot \ln(R_{jb} + 12.7587) - 0.291666 \cdot \ln\left(\frac{V_{s30}}{1000}\right) \]

\begin{align*}
  &\begin{cases}
    0.1254 & \text{if } F = 0 \text{ (normal)} \\
    0 & \text{if } F = 0.5 \text{ (strike-slip)} \\
    0.1188 & \text{if } F = 1 \text{ (reverse)}
  \end{cases}
\end{align*}

Figure 3: A diagram of the experimental set-up
Figure 4: Predicted vs. measured PGA of equation (6) for the whole data set

Figure 5: Graph of PGA versus $R_{th}$ for equation (6) found by the Lagrange system aligned with the graphs of models from other authors
Figure 6: Graph of $PGA$ versus $R_{jb}$ for equation (6) found by the Lagrange system aligned with the graphs of models from other authors.

Figure 7: Graph of $PGA$ versus $R_{jb}$ for equation (6) found by the Lagrange system for $M_w \in \{4, 5, 6, 7, 8\}$.