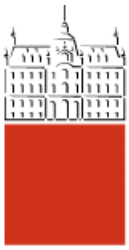
Univerza
v Ljubljani

Fakulteta za
gradbeništvo in
geodezijo

Jamova 2
1000 Ljubljana, Slovenija
telefon (01) 47 68 500
faks (01) 42 50 681
fgg@fgg.uni-lj.si



UNIVERZITETNI ŠTUDIJ
PRVE STOPNJE
GRADBENIŠTVA

Kandidat:

ŠTEFAN MARKIČ

NAPOVED NAJVEČJEGA POSPEŠKA TAL OB POTRESU Z UPORABO PROGRAMA ZA STROJNO UČENJE ENAČB LAGRAMGE

Diplomska naloga št.: 21/B-GR

PREDICTING PEAKGROUND ACCELERATION OF AN EARTHQUAKE BY USING THE MACHINE LEARNING PROGRAM LAGRAMGE

Graduation thesis No.: 21/B-GR

Mentor:
doc. dr. Vlado Stankovski

Somentor:
doc. dr. Iztok Peruš

Član komisije:
prof. dr. Jože Korelc
doc. dr. Alojzij Juvanc
Teja Melink
asist. mag. Robert Rijavec

Predsednik komisije:
izr. prof. dr. Janko Logar

Ljubljana, 25. 04. 2013

ERRATA

Stran z napako	Vrstica z napako	Namesto	Naj bo

IZJAVE

Poskusi in računalniške obdelave so bili izvedeni na strežniku Fakultete za gradbeništvo in geodezijo Univerze v Ljubljani ter na grid infrastrukturi Slovenske iniciative za nacionalni grid.

Podpisani Štefan Markič izjavljam, da sem avtor diplomskega dela z naslovom »Napoved največjega pospeška tal ob potresu z uporabo programa za strojno učenje enačb Lagramge«.

Izjavljam, da je elektronska različica je v vsem enaka tiskani.

Izjavljam, da dovoljujem objavo elektronske različice v repozitoriju UL FGG.

Spodnje Duplje, 15. 4. 2013 Štefan Markič

BIBLIOGRAFSKO – DOKUMENTACIJSKA STRAN IN IZVLEČEK

UDK	004.85:624.042.7(043.2)
Avtor	Štefan Markič
Mentor	doc. dr. Vlado Stankovski
Somentor	doc. dr. Iztok Peruš
Naslov	Napoved največjega pospeška tal ob potresu z uporabo programa za strojno učenje enačb Lagramge
Tip dokumenta	Diplomska naloga – univerzitetni študij
Obseg in oprema	29 str., 12 pregl., 12 sl., 18 en., 5 pril.
Ključne besede	Lagramge, odkrivanje enačb, strojno učenje, največji pospešek tal, potresno inženirstvo, iskanje s snopom

Izvleček

V diplomskem delu preverjamo možnosti za uporabo namenskega programa za strojno učenje enačb Lagramge na specifičnem inženirskem problemu napovedi največjega pospeška tal, ki se zgodi ob potresu. Lagramge pri svojem delovanju uporablja formalizem kontekstno neodvisne gramatike, ki vsebuje pravila za izgradnjo enačb in omeji prostor hipotez - mogočih enačb. V študiji smo razvili tri gramatike, vsako s svojo ravnijo uporabe specifičnega znanja področja potresnega inženirstva, ki ga je sestavljalo 68 enačb iz literature. Za poskuse smo imeli na voljo bazo 3550 zapisov o močnejših potresih, ki smo jo za namene navzkrižne validacije desetkrat naključno razbili na 90-odstotno učno in 10-odstotno testno množico. Algoritem je med izvajanjem hevristično in/ali izčrpno preiskal vse tri prostore hipotez in ovrednotil enačbe s funkcijo srednjega kvadratnega odklona na učni in testni množici. V vsakem poskusu smo določili tri najboljše enačbe glede na kvantitativni kriterij in jih primerjali med seboj ter z modeli študij *Next Generation Attenuation* in modelom evropskih avtorjev Akkarja in Bommerja. Ugotovili smo, da vključitev specifičnega znanja, ki ni niti preohlapna niti preveč specifična, pozitivno vpliva na kvaliteto rezultatov. Poleg tega smo preučili še vpliv vrednosti vhodnih parametrov programa, ki usmerjajo potek iskanja najboljše enačbe. Rezultati kažejo, da bi Lagramge lahko uporabili tako pri podobnih problemih v potresnem inženirstvu kot tudi na drugih inženirskih področjih.

BIBLIOGRAFIC – DOCUMENTALISTIC INFORMATION AND ABSTRACT

UDC	004.85:624.042.7(043.2)
Author	Štefan Markič
Supervisor	Assist. Prof. Vlado Stankovski, Ph.D.
Co-advisor	Assist. Prof. Iztok Peruš, Ph.D.
Title	Predicting Peak Ground Acceleration of an Earthquake by Using the Machine Learning Program Lagramge
Document type	Graduation Thesis – University studies
Notes	29 p., 12 tab., 12. fig., 18 eq., 5 ann.
Key words	Lagramge, equation discovery, machine learning, peak ground acceleration, earthquake engineering, beam search

Abstract

This bachelor's thesis deals with testing the equation discovery system Lagramge when applied to a specific engineering problem of modelling the earthquake's peak ground acceleration. The Lagramge system uses context-free grammar formalism which contains rules for building equations and limits the hypothesis space of possible equations. We developed three different grammars, each incorporating a different level of domain specific knowledge, which included 68 published equations. In the experiments a database of 3550 strong motion earthquake recordings was used and for the purpose of cross validation split 10 times into 90 % learning and 10 % testing sets. The algorithm employed exhaustive and/or heuristic search methods in all three hypothesis spaces and evaluated the equations on the learning and testing datasets using the mean squared error criterion. From each of 4 experiments three best equations were selected on the basis of quantitative criterion and compared with each other and with the equations from the *Next Generation Attenuation* study as well as with the equation developed by the European authors Akkar and Bommer. We found out that inclusion of the domain specific knowledge which is neither too specific nor too general improves the quality of the results. In addition to this, influences of other input parameters guiding the process of equation discovery were examined. The results of this study show that the Lagramge system could also be applied to similar problems in earthquake engineering as well as to other fields of engineering.

ZAHVALE

Najprej bi se rad zahvalil staršema Antoniji in Tadeju za pomoč in podporo v vseh mojih dosedanjih letih življenja in da sta mi omogočila kvalitetno šolanje.

Zahvala gre tudi mojemu dekletu, ki me je tekom celotnega študija vzpodbujalo in mi lektoriralo to delo. Hvala, Katja.

Rad bi se zahvalil še študijskemu kolegu Jaku Dirnbeku, ki je napisal skripto za zaganjanje Lagramgea na grid infrastrukturi.

Prav tako se zahvaljujem izr. prof. dr. Ljupču Todorovskemu za razlago sistema Lagramge in velikodušno pomoč pri skriptiranju.

Zahvala gre akad. prof. dr. Petru Fajfarju, ki je dal idejo za to raziskovalno delo.

Še posebej iskreno se zahvaljujem mentorju doc. dr. Vladu Stankovskemu in somentorju doc. dr. Iztoku Perušu za pomoč in usmerjanje pri raziskovalnem delu.

KAZALO

KAZALO PREGLEDNIC

KAZALO SLIK

OKRAJŠAVE IN SIMBOLI

F	vrsta preloma
GMPE	Ground Motion Prediction Equation
M_w	momentna magnituda
MSE	mean squared error
NGI	nacionalna grid infrastruktura
NGA	Next Generation Attenuation
PGA	peak ground acceleration
R_{jb}	Joyner-Boorova razdalja
SLING	Slovenska iniciativa za nacionalni grid
$V_{s,30}$	povprečna hitrost strižnih valov v gornjih 30 metrih zemeljskega površja

SLOVAR MANJ ZNANIH BESED IN TUJK

Odkrivanje enačb (angl. *Equation Discovery*) je področje strojnega učenja ((angl. *Machine Learning*), ki je specializirano za iskanje enačb.

Prostor hipotez (angl. *Hypothesis Space*) je na področju odkrivanja enačb množica vseh mogočih enačb. V primeru programa Lagramge je ta prostor omejen s kontekstno neodvisno gramatiko.

Izpeljevalno drevo (angl. *Derivation Tree*) je grafična predstavitev izpeljave enačbe v postopku raziskovanja prostora hipotez. Primer lahko vidimo na sliki 2.

Izčrpno iskanje (angl. *Exhaustive Search*) je preiskovanje prostora hipotez na način, da se izpeljejo in preizkusijo vse mogoče formulacije.

Iskanje v snopu (angl. *Beam Search*) je hevristično preiskovanje prostora hipotez na način, da se najprej izpelje in preizkusi le določeno število enačb. Nato se izmed njih in njihovih prvih naslednikov zopet zapomni le določeno število najboljših in nadaljuje.

Srednji kvadratni odklon (angl. *Mean Squared Error*) je statistična mera, ki pove povprečno napako izračunanih vrednosti glede na meritve. Izračuna se ga po enačbi (3).

Navzkrižno preverjanje (angl. *Cross-validation*) je postopek preverjanja, kjer z modelom, razvitim na učni množici (večja podmnožica baze) poskušamo napovedati podatke iz testne množice.

Enačbe napovedi gibanja tal (angl. *Ground Motion Prediction Equations*) so enačbe, ki napovedujejo različne parametre gibanja tal.

Enačba pojemanja (angl. *Attenuation Relationship*) je ime za enačbo napovedi gibanja tal, ki se je uporabljalo v preteklosti.

Parametri gibanja tal (angl. *Ground Motion Parameters*) so največji pospešek tal (Peak Ground Acceleration, PGA), hitrost (Peak Ground Velocity, PGV) in pomik (Peak Ground Displacement, PGD); spektralni pospeški (Spectral Acceleration, S_a), hitrosti (Spectral Velocity, S_v) in pomik (Spectral Displacement, S_d).

Največji pospešek tal (angl. *Peak Ground Acceleration*) je eden izmed parametrov gibanja tal. Odčitamo ga z merilnih naprav, enačbe napovedi gibanja tal pa ga poskušajo predvideti. Prek drugega Newtonovega zakona deluje kot dinamični potresni obtežbi enakovredna statična potresna obtežba na konstrukcijo.

Ta stran je namenoma prazna.

1 UVOD

Enačba je v Slovarju slovenskega knjižnega jezika definirana kot matematični zapis, ki sestoji iz dveh, z enačajem povezanih matematičnih izrazov [1]. Inženirji enačbo dojemamo drugače, saj matematičnim izrazom pripišemo določen pomen in v njih nastopajoče spremenljivke povežemo z realnimi fizikalnimi količinami. Tako s pomočjo enačb z že znanimi spremenljivkami (predpostavljenimi ali izračunanimi) izračunamo še neznane količine in pridemo do novih ugotovitev. Toda kako najdemo tisto enačbo, ki pravilno oz. z zadovoljivo majhno napako opisuje opazovani pojav? Načinov je več, v tej diplomski nalogi pa obravnavamo uporabo namenskega računalniškega programa za strojno odkrivanje enačb na primeru napovedovanja največjega pospeška tal ob potresu.

1.1 Potresno inženirstvo

V Sloveniji je potres vsesplošno znan pojav gibanja zemeljskega površja, ki se zgodi nepričakovano in brez vnaprejšnjega opozorila. Močnejši potresi lahko močno poškodujejo infrastrukturo in povzročijo precej težav ljudem in skupnostim. Pojavljajo se po vsem svetu, vendar ne enako pogosto in intenzivno. Naloga inženirjev je tako ustrezno dimenzioniranje konstrukcije ob zavedanju, da se v času življenjske dobe konstrukcije lahko zgodi močnejši potres in kritično vpliva na varnost konstrukcije. Zaradi narave dogodka jih obravnavamo podobno kot poplave, tj. s povratno dobo, s katero lažje ocenimo tveganje nastanka poškodb in škode. S pomočjo parametrov gibanja tal, ki se zgodijo ob potresu (tj. spektralni pospeški, hitrosti in pomiki; največji pospešek, hitrost in pomik), lahko ocenimo projektno potresno obtežbo na konstrukcijo in tako zagotovimo ustrezno varnost v primeru močnega potresa. Vpliv različnih dejavnikov na parametre gibanja tal je običajno zajet v t. i. enačbi napovedi gibanja tal (GMPE, *angl. ground motion prediction equation*). Pri protipotresnem načrtovanju objektov, ki se projektirajo v vsakdanji praksi, se projektni pospešek tal odčita s karte projektnih pospeškov tal, medtem ko se pri projektiranju zahtevnih objektov (npr. jedrske elektrarne) projektni pospeški določijo s posebnimi študijami, ki vključujejo uporabo različnih enačb gibanja tal.

Med parametri gibanja tal je najpogosteje modeliran največji pospešek tal (PGA, *angl. peak ground acceleration*), ki je enostavna, a učinkovita mera potresnega valovanja. Z njim prek predpisanega postopka nadomestimo dinamično potresno obtežbo s statično obtežbo in s tem poenostavimo projektiranje, zato je njegova določitev izredno pomembna za protipotresno gradnjo. Prav zaradi tega je problem napovedi največjega pospeška tal poskušalo rešiti nemalo znanstvenikov, ki so do danes razvili več kot 280 enačb, ki na različne načine upoštevajo znanje stroke oz. predpostavljajo t. i. enačbo pojemanja (*angl. attenuation relationship*). Povzetek teh študij z obrazloženimi predpostavkami, izbiro podatkov in neodvisnih spremenljivk ter postopke izpeljave najdemo v [2]. Kot primer enačbe za napoved največjega pospeška tal navajamo trenutno najnovejšo enačbo (1) evropskih avtorjev Akkarja in Bommerja, ki sta za svoje raziskave uporabila do tedaj največjo bazo zapisov o močnejših potresih v Evropi [3].

$$\log_{10}(PGA) = a_1 + a_2 \cdot M_{\mathrm{w}} - a_3 \cdot M_{\mathrm{w}}^2 + (a_4 + a_5 \cdot M_{\mathrm{w}}) \cdot \log_{10} \sqrt{R_{\mathrm{jb}}^2 + a_6^2} + \begin{cases} a_A & V_{\mathrm{s},30} < 360\frac{m}{s} \\ a_S & 360\frac{m}{s} \leq V_{\mathrm{s},30} < 800\frac{m}{s} \\ 0 & 800\frac{m}{s} \leq V_{\mathrm{s},30} \end{cases} + \begin{cases} a_N & normalen\ prelom \\ a_R & reverzen\ prelom \\ 0 & zmičen\ prelom \end{cases} \quad (1)$$

V enačbi (1) so koeficienti zamenjani s črkami tako, kot sta avtorja v začetku predpostavila njeno zgradbo. V njej M_{w} označuje magnitudo, R_{jb} razdaljo med navpično projekcijo prelomne ploskve na površje in lokacijo, $V_{s,30}$ povprečno strižno hitrost potresnih valov v zgornjih 30 metrih površja, zadnji člen pa pripadajoče konstante različnih vrst preloma (glej 3.1).

1.2 Odkrivanje enačb

Z razvojem računalnikov se je začelo razmišljati o umetni inteligenci (*angl. artificial inteligence*), pojavile so se nove raziskovalne metode in odprlo se je novo znanstveno področje strojnega učenja. Glavni cilj področja nakazuje že ime, saj si prizadevamo, da bi bili stroji oz. računalniki sposobni posnemati pomembno človeško lastnost – učenje. To pomeni, da želimo usposobiti računalnik, da iz nabora primerov oz. obstoječih podatkov o določenem problemu odkrije pravila, ki te podatke povezujejo. Tako to učenje ni le pomnjenje podatkov in postopkov, temveč tudi posploševanje na nevidene primere, s čimer je zelo podobno človeškemu učenju [4]. Nove metode lahko tako s svojo zmožnostjo modeliranja potrdijo postavljene hipoteze ali pa podani problem samostojno rešijo.

Odkrivanje enačb se je kot podpodročje strojnega učenja uveljavilo v poznih osemdesetih in devetdesetih letih prejšnjega stoletja z razvojem sistemov za odkrivanje enačb (npr. Bacon, EF, E*, Lagrange in Gold-Horn) in ima veliko skupnega s področji induktivnega logičnega programiranja in genetskega programiranja [5]. Prostor možnih hipotez je v teh programih množica vseh enačb, ki jih lahko sestavimo iz podanega nabora operatorjev, funkcij in spremenljivk. Glavna lastnost področja je omejevanje prostora hipotez z uporabniško določenim formalizmom, ki je učnemu sistemu podan kot vhodni podatek in ni vgrajen v samem algoritmu [5]. Naloga sistemov za odkrivanje enačb je poiskati enačbo, ki kar najbolje opisuje dano množico podatkov, kar jih dela zanimive za inženirje, ki še vedno uporabljajo izkustvene in empirične enačbe, temelječe na obsežnih množicah meritev.

Pri postopku izdelave modelov za strojno učenje najprej izberemo in pripravimo primerno množico podatkov. Za namene navzkrižnega vrednotenja (*angl. cross-validation*) jo večkrat neodvisno razdelimo na dva dela. Prvi, večji del, imenujemo učna množica, ki algoritmu služi za učenje. Drugi, manjši del, imenujemo testna množica in ga na koncu uporabimo za ovrednotenje razvitih modelov na podatkih, ki niso bili vključeni v fazo učenja.

Poleg razdelitve podatkov določimo tudi omejitve možnih modelov v prostoru hipotez, pri čemer moramo paziti na problema generalizacije in specifikacije. Slednjega najlažje ponazorimo s teorijo približkov, po kateri vedno obstaja neskončno število polinomskih krivulj (zlepkov), ki so lahko dovolj dober približek poljubnim podanim podatkom. Če tako omejitve prostora hipotez

zastavimo preveč široko, lahko povzročimo neželeno pretirano ujemanje z lokalnimi lastnostmi učnih podatkov in zmanjševanje natančnosti pri podatkih, ki v fazo učenja niso bili vključeni. Nasprotno pa nastane problem generalizacije, če omejitve zastavimo preveč ozko, saj morda prostor hipotez niti ni dovolj velik, da bi vseboval enačbe, ki dajo uporaben približek podanim podatkom [5].

1.3 Definicija problema

Iz opisov v poglavju 1.1 ugotavljamo, da je področje napovedi največjega pospeška tal razmeroma dobro raziskano. Tako nam lahko predstavlja osnovno znanje tega specifičnega področja, ki ga potrebujemo za omejitev prostora vseh možnih enačb, kot je opisano v poglavju 1.2. Namen raziskave je tako razvoj nove enačbe za napoved največjega pospeška tal z uporabo alternativne metode strojnega učenja enačb in s tem utemeljitev novega postopka za študije v potresnem inženirstvu.

1.4 Organizacija diplomskega dela

V prvem poglavju so predstavljene naša motivacija za prikazano raziskavo, definicija interdisciplinarnega problema in delovna hipoteza. V drugem poglavju so opisane specifikacije, posebnosti in način uporabe programa Lagramge. Tretje poglavje je razdeljeno na definiranje gramatik, izbiro baze podatkov o potresih in izbiro ostalih vhodnih parametrov, katerih vpliv na rezultate smo preučevali. V četrtem poglavju so predstavljeni tabele in grafi rezultatov poskusov. V petem poglavju sledi razprava, v kateri komentiramo dobljene rezultate in jih poskušamo ovrednotiti. Prav tako so v tem poglavju zapisane kritike programa in možnosti za nadaljnje delo na obeh področjih. Diplomsko delo se zaključi z viri, literaturo in prilogami, ki vsebujejo tudi tri mednarodno objavljene članke (priloge C [6], D [7] in E [8]).

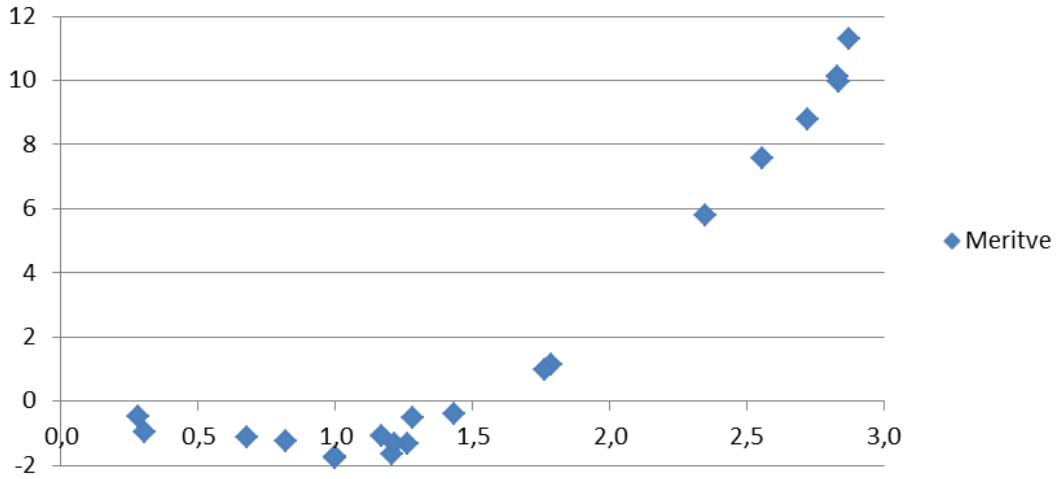
2 LAGRAMGE

Eden izmed namenskih programov za odkrivanje enačb je bil razvit na Univerzi v Ljubljani, njegov avtor Ljupčo Todorovski pa ga je poimenoval Lagramge. Njegov formalizem za opis prostora hipotez pri odkrivanju enačb temelji na kontekstno neodvisni gramatiki. Preko nje uporabnik poda opis iskane oblike enačbe, kar omogoča vgradnjo obstoječega inženirskega znanja v proces učenja [5]. Njena sestava je razložena v poglavju 2.1.

Za nazorno ponazoritev celotnega postopka odkrivanja enačbe z Lagramgeem si zamislimo enostaven primer. Pri opazovanju pojava zabeležimo dvajset meritev dveh spremenljivk (poimenujmo ju x in z), za kateri predvidevamo, da sta v medsebojni odvisnosti. Vrednosti meritev so zapisane v preglednici 1, njihovo grafično predstavitev pa lahko vidimo na sliki 1.

Preglednica 1: Meritve opazovanega pojava

x	z	x	z	x	z	x	z
1,1674	-1,1069	1,4323	-0,4172	1,2834	-0,5016	1,2131	-1,3149
1,2074	-1,6472	2,8287	10,1309	0,8182	-1,2455	2,3492	5,8093
0,9995	-1,7942	0,2828	-0,4780	2,8329	9,9471	0,9995	-1,7241
1,7883	1,1428	1,7604	0,9863	0,6757	-1,1297	2,5553	7,5720
0,3034	-0,9572	2,8722	11,3076	1,2610	-1,3404	2,7201	8,7955



Slika 1: Grafična predstavitev meritev opazovanega pojava

2.1 Kontekstno neodvisna gramatika

Gramatika $G = \{N, T, P, S\}$ je svoje ime dobila zaradi vzporednosti s slovnico pri jeziku, saj določa strukturo enačb, podobno kot slovnica določa strukturo stavkov in povedi. Vsebuje tri množice: množico produkcij (P, *angl. Productions*) in disjunktni množici končnih (T, *angl. Terminal*) in nekončnih (N, *angl. Nonterminal*) simbolov. Simbol $S \in N$ označuje začetni nekončni simbol, iz katerega začnemo tvoriti enačbo. Produkcije, ki predstavljajo slovnična pravila za tvorjenje izrazov, predstavimo v obliki $A \rightarrow \alpha$, pri čemer so simbol na levi strani $A \in N$, v desni strani nastopajoče spremenljivke $\alpha \in (T \cup N)$ in vsi uporabljeni povezovalni operatorji oz. funkcije že definirani. Če so s stališča algoritma te produkcije enakovredne in

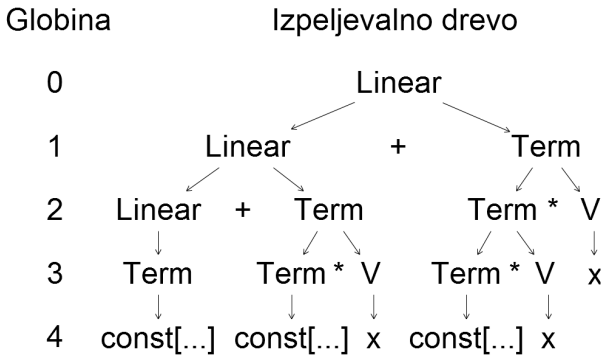
jih pri izpeljavi enačbe izbira in uporablja neodvisno od preostalih členov, ki so že v enačbi, govorimo o kontekstno neodvisni gramatiki [5].

V preglednici 2 vidimo primer enostavne gramatike, iz katere lahko Lagramge sestavi poljubno dolg polinomski izraz poljubne stopnje, zato jo poimenujemo Polinomska gramatika P. Prvi dve produkciji (L1 in L2) omogočata tvorjenje vsote več členov, medtem ko drugi dve produkciji (T1 in T2) omogočata izpeljavo polinomskih in konstantnih členov. V gramatiki so nekončni simboli lahko poimenovani poljubno, le simbol V Lagramgeu vedno predstavlja produkcijo k neodvisnim spremenljivkam, ki poleg simbola $const[\dots]$ (natančneje razložen v poglavju 2.2) pripadajo množici končnih simbolov. Tako v produkciji T2 s simbolom V označimo katero koli izmed neodvisnih spremenljivk, ki jih Lagramge sam prebere iz vhodne datoteke z meritvami. Algoritem tekom izvajanja samodejno doda produkcije $\forall v_i : V \rightarrow variable\_v_i \in P$, zato teh produkcij ne smemo zapisati. Če pa se želimo v gramatiki sklicevati na dotično neodvisno spremenljivko, moramo pred njeno ime postaviti napis '*variable_*'. V produkcijah lahko poleg običajnih matematičnih operatorjev in že definiranih funkcij uporabimo tudi lastne definirane funkcije v programskem jeziku C in tako vključimo specifično znanje iz obravnavanega področja [5].

Preglednica 2: Polinomska gramatika P

Oznaka	Produkcija
L1	Linear $\rightarrow$ Term;
L2	Linear $\rightarrow$ Linear + Term;
T1	Term $\rightarrow$ const[_:-100:0.1:100];
T2	Term $\rightarrow$ Term * V;

Razvoj izraza z uporabo gramatike najlažje ponazorimo s pomočjo izpeljevalnega drevesa, primer katerega vidimo na sliki 2. Začnemo z začetnim simbolom S, ki je v primeru gramatike P enak Linear. Nato s pomočjo produkcij razvejamo drevo, dokler vsi listi drevesa ne pripadajo množici končnih simbolov. Izraz, ki pripada izpeljevalnemu drevesu, dobimo tako, da od leve proti desni zaporedoma preberemo vse končne liste in operacije med njimi. Izraz, katerega izpeljavo ponazarja drevo na sliki 2, lahko vidimo na desni strani enačbe (4).



Slika 2: Ponazoritev razvoja enačbe (4) z izpeljevalnim drevesom z označeno globino (levo)

Vsako izpeljevalno drevo ima svojo višino, ki jo zaradi lažje predstavljive obrnjene rasti (od zgoraj navzdol) imenujemo globina *d* (*angl. depth*) [5]. Z uporabo rekurzivnih produkcij, kot sta

npr. L2 in T2 v gramatiki P, lahko globina drevesa narašča v neskončnost in s tem tudi prostor hipotez. Zato je omejitev globine pri gramatikah, ki vključujejo vsaj eno rekurzivno produkcijo, nujna. S to omejitvijo program poskrbi, da so na določeni globini vsi listi drevesa končni simboli. Primer izpeljevalnega drevesa na sliki 2 ima globino 4, kot je razvidno iz številk, pripisanih na levi strani vsake nove generacije listov. Vrednost parametra *d* omeji globino izpeljevalnih dreves in tako ne dovoli izpeljave zahtevnejših izrazov ter skrajša trajanje algoritma. Če gramatika ne vsebuje nobene rekurzivne produkcije, je parameter nepotreben.

Vsaka kontekstno neodvisna gramatika definira lasten prostor hipotez, v katerem algoritem najde optimalno enačbo, če mu dovolimo izčrpno iskanje vseh možnih enačb in če je optimalna enačba seveda sploh zajeta. Ker pa so mnogi prostori hipotez zelo veliki in bi njihovo izčrpno preiskovanje trajalo preveč časa, lahko algoritem usmerimo v hevristično iskanje v snopu (*angl. beam search*). Program nam omogoča iskanje dveh oblik enačb:

- običajnih enačb oblike $v_d = E$ ali
- diferencialnih enačb oblike $\frac{\partial v_d}{\partial t} = \dot{v}_d = E$,

pri čemer v_d predstavlja odvisno spremenljivko in E izraz, ki ga je možno izpeljati iz podane gramatike s pomočjo izpeljevalnega drevesa [5].

2.2 Datoteka s podatki

Množico podatkov Lagramgeu podamo v eni ali več tabelah z meritvami odvisne in neodvisnih spremenljivk. V prvi vrstici zapišemo vsa imena spremenljivk, ločena s presledki, v naslednjih vrsticah pa v enakem vrstnem redu njihove vrednosti, ločene s tabulatorjem. Vse vrstice morajo biti zaključene s podpičjem [5]. Tako prepišemo meritve primera v zahtevano obliko:

```
x z;
1.1674    −1.1069;
1.2074    −1.6472;
0.9995    −1.7942;
1.7883     1.1428;
0.3034    −0.9572;
1.2834    −0.5016;
0.8182    −1.2455;
2.8329     9.9471;
0.6757    −1.1297;
1.2610    −1.3404;
1.4323    −0.4172;
2.8287    10.1309;
0.2828    −0.4780;
1.7604     0.9863;
2.8722    11.3076;
1.2131    −1.3149;
2.3492     5.8093;
0.9995    −1.7241;
2.5553     7.5720;
2.7201     8.7955;
```

Ob zagonu programa je obvezna uporaba zastavice *-v*, z vrednostjo katere programu povemo, katera izmed spremenljivk v bazi je izbrana kot odvisna in mora biti na levi strani enačbe. V našem primeru je to spremenljivka z.

Med končnimi simboli vsake gramatike je tudi poseben simbol za označevanje konstant, ki ima sledečo strukturo [5]:

$$const[ime : najmanjša\ vrednost : začetna\ vrednost : največja\ vrednost] \ . \tag{2}$$

Z njim označimo koeficiente v enačbi, ki jih Lagramge umeri na vhodne podatke z dosego najmanjšega srednjega kvadratnega odklona (MSE, *angl. Mean Squared Error*) [5]:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(v_{d,i,m} - v_{d,i,p})^2 \ , \tag{3}$$

kjer je i število vrstic meritev, $v_{d,i,m}$ podana oz. izmerjena vrednost odvisne spremenljivke v vrstici i in $v_{d,i,p}$ preračunana vrednost odvisne spremenljivke s preverjano enačbo in podatki iz vrstice i. Nelinearno optimizacijo konstantnih parametrov program opravi z uporabo algoritma downhill simplex (*angl. Downhill Simplex Algorithm*) in metode Levenberg-Marquardt (Press et al., 1986, cit. po [5]). Te funkcije se pogosto ujamejo v lokalne minimume, zato lahko s parametrom *-m* označimo število ponovnih začetkov optimizacije konstantnih parametrov [5].

2.3 Ostali vhodni parametri

Poleg dveh osrednjih datotek – gramatike in tabele podatkov – pa algoritem usmerjamo v želeno smer tudi z drugimi omejevalnimi parametri. Pri uporabi niso obvezni, a nam omogočajo hevristično preiskovanje prostora vseh možnih hipotez [5].

Algoritem se lahko ustavi na tri različne načine. Privzeto se ustavi, ko preišče celoten prostor enačb oz. ko doseže največjo globino v primeru hevurističnega iskanja. Z zastavico *-c timelimit* lahko omejimo izvajalni čas, katerega trajanje v sekundah določa parameter *-l*. Po preteku omejenega časa Lagramge izpiše najboljše enačbe, ki jih je uspel najti do takrat. Kot tretjo možnost nam program omogoča ustavitev, ko najde enačbo z manjšo vrednostjo funkcije MSE od podane mejne vrednosti [5].

Vrednost parametra *-b* označuje število najboljših enačb, ki si jih Lagramge med izvajanjem sproti zapomni, njegova privzeta vrednost je nastavljena na 25. Ta parameter je najbolj uporaben pri hevurističnem iskanju v snopu (določeno z zastavico *-s beam*), saj le tako lahko nastavimo širino snopa. To iskanje za razliko od izčrpnega ne preišče celotnega prostora možnih formulacij enačb, temveč si zapomni le omejeno število najboljših enačb, potem preizkusi vse njihove naslednike glede na produkcije gramatike in si izmed njih zapomni enako število najboljših [5].

V našem primeru lahko glede na grafično predstavitev pričakujemo kvadratno odvisnost med spremenljivkama x in z. Gramatika P, ki je opisana v preglednici 2, lahko tvori polinome druge stopnje že na globini 4, kot je razvidno s slike 2, zato izberemo *-d* 4. Prostor možnih formulacij

bomo preiskali izčrpno, optimizacijo konstantnih parametrov pa bomo ponovno začeli desetkrat *-m* 10. Preden zaženemo poskuse, v skladu s prakso strojnega učenja razdelimo celotno množico podatkov desetkrat na dva dela (na 90-odstotno učno in 10-odstotno testno množico). Tako smo določili vse potrebno za izvajanje poskusov in lahko zaženemo deset instanc programa, kjer vsaki kot vhodno datoteko s podatki podamo učno množico ene izmed desetih razdelitev.

2.4 Izhodna datoteka

Med izvajanjem Lagramge izdela izhodno besedilno datoteko, v kateri so izpisani vsi nastavljeni parametri poskusa in rezultati. Primer izhodne datoteke, ki predstavlja tudi najboljše rezultate našega primera, sledi temu odstavku. Na začetku je opisana pot do datoteke (ali datotek) s podatki, imena vseh spremenljivk in število vrstic s podatki. Sledi izpis gramatike na bolj zgoščen način, saj so produkcije, ki izhajajo iz istega nekončnega simbola, ločene le z logičnim operatorjem '*ali*' (simbol '|'). Zatem lahko preverimo izbiro odvisne spremenljivke, iskane odvisnosti in največjo dovoljeno globino iskanja. Sledi izpis tabele z zbranimi lastnostmi prostora hipotez; po vrsti si sledijo stolpci: globina, št. možnih enačb, največja dolžina enačbe, najmanjša dolžina enačbe, največje število konstant, uporabljenih v enačbi in največja dolžina izpeljave. Pod tabelo so napisane nastavitve strategije iskanja enačbe in način zaustavitve algoritma. Kot zadnja nastavitev algoritma je izpisano število ponovnih začetkov optimizacije konstant. Če želimo, da nam program izpiše vse enačbe, ki jih tekom izvajanja preizkusi, se lahko odločimo za možnost *verbose*, ki jo nastavimo z zastavico *-V*. Jedro izpisne datoteke tako lahko predstavljajo vse ovrednotene enačbe. Temu sledi število vseh izpeljanih dreves in seznam najboljših enačb (začenši z najboljšo), ki jih je algoritem uspel najti. Na koncu nam algoritem izpiše še skupni čas izvajanja, podan v sekundah [5].

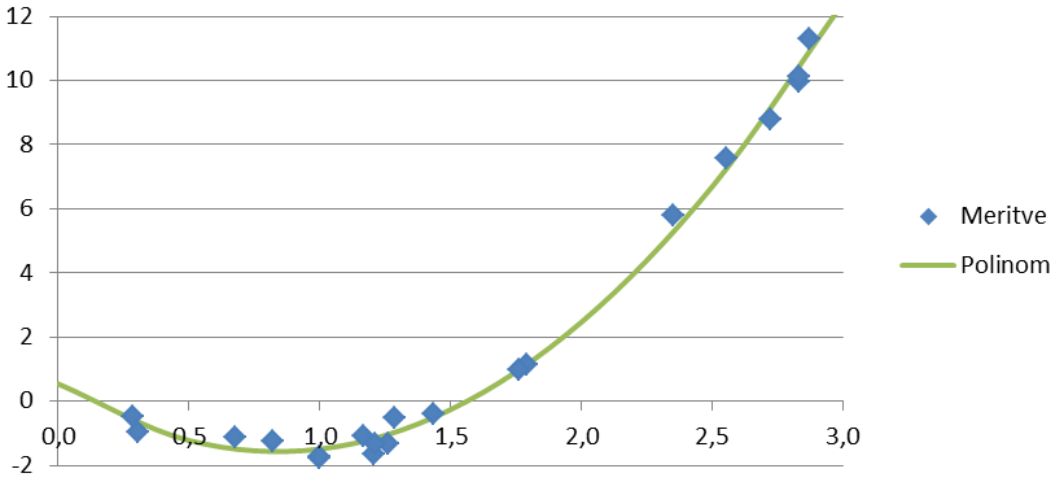
```
Data files: dataPrimer/learning/ld2
Variables: x z
Data length: 18
Grammar definition file: lagramge-release/lib/polinom.gramm
Grammar:
Polynomial -> Term | Polynomial + Term;
Term -> const[_:-1000:0.1:1000] | Term * V;
Equation type: ordinary explicit (z)
Maximal parse tree depth: 4
Atom Polynomial:
depth    #p.trees    max.len.    min.len.    #consts    der.len.
    0           0          -1          -1         -1          -1
    1           0          -1          -1         -1           0
    2           1           1           1          1           2
    3           4           5           3          2           6
    4          15          11           5          3          12
Search strategy: exhaustive
Stopping criterion: none
Search heuristic: sum of squared errors
Restarts of parameter estimation methods: 10
```

```
Verbose: off
15 parse trees evaluated
Best equations:
z = 0.541374 + −5.03003 ∗ x + 2.99834 ∗ x ∗ x {MSE = 0.102965, MDL =
    2.31214}
z = −4.32101 ∗ x + 2.81022 ∗ x ∗ x {MSE = 0.120925, MDL = 1.92843}
z = −2.82075 + 1.58159 ∗ x ∗ x {MSE = 0.894263, MDL = 2.3001}
z = −1.22399 + −1.59676 + 1.58159 ∗ x ∗ x {MSE = 0.894263, MDL = 2.70177}
z = −5.86331 + −303.113 ∗ x + 308.255 ∗ x {MSE = 3.48959, MDL = 5.2971}
z = 298.367 + 5.14205 ∗ x + −304.23 {MSE = 3.48959, MDL = 4.89543}
z = 5.14205 ∗ x + −5.86331 {MSE = 3.48959, MDL = 4.49376}
z = −5.86331 + 5.14205 ∗ x {MSE = 3.48959, MDL = 4.49376}
z = −258.247 + 252.383 + 5.14205 ∗ x {MSE = 3.48959, MDL = 4.89543}
z = 1.09716 ∗ x ∗ x {MSE = 4.80706, MDL = 5.81123}
z = 843.245 ∗ x + −841.078 ∗ x {MSE = 11.6624, MDL = 13.0683}
z = 2.16631 ∗ x {MSE = 11.6624, MDL = 12.2649}
z = 0.925964 + 0.93383 {MSE = 22.0917, MDL = 22.6942}
z = 147.631 + −761.138 + 615.367 {MSE = 22.0917, MDL = 23.0959}
z = 1.85979 {MSE = 22.0917, MDL = 22.2926}
Time elapsed: 0.08 [s]
```

Ko vseh deset instanc zaključi z izvajanjem, zberemo vse izpeljane enačbe in ovrednotimo napako še na pripadajočih testnih množicah, kar nam služi za primerjavo. Izmed vseh 150 izpeljanih enačb (10 instanc, vsaka po 15 enačb na globini 4 z gramatiko P) dobimo najboljšo enačbo, ki je dosegla $MSE = 0,046857$ na testni in $MSE = 0,102965$ na učni množici:

$$z = 0,541374 - 5,03003 \cdot x + 2,99834 \cdot x \cdot x \; . \tag{4}$$

Za primerjavo lahko pogledamo zadnjo izpisano enačbo v izpisni datoteki, ki predpostavlja funkcijsko odvisnost $z = const = 1,85979$ in njeno ovrednoteno napako $MSE = 22.0917$. Velikostni razred napake enačbe (4) na učni množici je za kar dva razreda manjši. Grafično ponazoritev enačbe (4) prikazuje slika 3, na kateri opazimo dobro ujemanje s podatki.



Slika 3: Graf enačbe (4) z vrisanimi meritvami

3 NASTAVITVE POSKUSOV

3.1 Baza podatkov

Za potrebe analize tveganja potrebujemo zadostno zbirko podatkov o prejšnjih dogodkih, iz katerih lahko predvidimo verjetnost obsega prihodnjih dogodkov, zato države gradijo nacionalno merilno infrastrukturo, inštituti pa opazujejo in sistematično popisujejo vsakršno gibanje tal na površini. Določitev reprezentativne baze podatkov, ki nam lahko služi kot osnova za raziskovanje, je težko opravilo in zahteva dobro poznavanje problema. Tako znanstveniki zbirajo za vsak potres in vsako merilno mesto raznovrstne dejavnike, za katere predvidevajo, da bi lahko vplivali na parametre gibanja tal. V glavnem se ti dejavniki delijo na tri kategorije: vpliv vira, poti in lokacije [9].

3.1.1 Vpliv potresnega vira

Glavni dejavnik vira je jakost potresa, ki jo merimo oz. označujemo z magnitudo (*M*) in vedno nastopa v enačbah napovedi gibanja tal. Po svetu je definiranih več različnih magnitudnih lestvic (Richterjeva oz. lokalna magnituda, magnituda določena iz površinskih valov, momentna magnituda itn.), vse pa temeljijo na količini energije, sproščene v epicentru potresa. Izmed teh se je v zadnjih letih uveljavila momentna magnituda. Drugi dejavniki vira, ki pa niso vedno uporabljeni v enačbah napovedi gibanja tal oz. so vanje vključeni na drugačen način, so vrsta preloma (*angl. fault type*), vpliv krovnine (*angl. hanging wall effect*), oblika in velikost prelomne ploskve, splošne tektonske značilnosti prelomnice, usmerjenost in globina epicentra. V literaturi lahko najdemo tudi različne definicije poti, pri čemer nekatere vključujejo podatek o globini [9].

3.1.2 Vpliv poti

Ko potresno valovanje potuje skozi zemljo, izgublja energijo, sproščeno v potresu, zato je pomembno, kako daleč od žarišča potresa opazujemo njegove učinke. Različni avtorji so predlagali različno definirane razdalje (*R*) – pogosteje uporabljene so epicentralna razdalja, hipocentralna razdalja, razdalja do centra prelomnice, razdalja do prelomnice in razdalja do površinske projekcije prelomne ploskve (poimenovana tudi Joyner-Boorova razdalja). Odvisno od definicije razdalje so avtorji kot ločen vpliv poti (oz. vira) upoštevali še globino epicentra (*angl. depth*), ki ima pri kratkih razdaljah velik vpliv, pri večanju razdalje pa ta vpliv močno upade [9].

3.1.3 Vpliv lokacije

Na obnašanje objekta med potresom močno vplivajo lokalne lastnosti temeljnih tal, ki prenašajo potresno obtežbo na temelje in konstrukcijo. Za opis tal se v literaturi pojavljajo različne delitve na geološke razrede in definicije količin, ki jih opisujejo. Med njimi se vse pogosteje uporablja povprečna hitrost strižnih valov v zgornjih 30 metrih površja $V_{\mathrm{s},30}$, ki odpravlja subjektivno oceno projektanta in s tem prispeva k manjši negotovosti in napako [9].

3.1.4 Izbrana baza

Peruš in Fajfar sta za raziskave z metodo CAE (Conditional Average Estimator) zbrala ogromno bazo podatkov 'PF-L', ki je unija podatkovnih baz, uporabljenih v študijah "Next Generation Attenuation models", in študije evropskih avtorjev Akkarja in Bommerja [3, 10, 11]. Sestavljena je iz 3550 zapisov o približno 200 močnejših potresih, ki so se zgodili po vsem svetu. Za namene našega modeliranja smo tako kot neodvisne spremenljivke tudi zaradi enostavnejše primerjave z ostalimi študijami iz baze 'PF-L' izbrali momentno magnitudo (M_{w}), Joyner-Boorovo razdaljo (R_{jb}), povprečno hitrost strižnih valov v zgornjih 30 metrih površja ($V_{\mathrm{s,30}}$) in vrsto preloma (F). Pri slednji smo za lažjo implementacijo v programu zamenjali opise s številčnimi vrednostmi, kot je zapisano v preglednici 3.

Preglednica 3: Številčne vrednosti vrst preloma *F*

Vrednost v bazi	Vrsta preloma
0	normalen prelom
0,5	zmičen prelom
1	reverzen prelom

Kot odvisno spremenljivko smo uporabili geometrično sredino obeh horizontalnih komponent največjega pospeška tal, merjeno kot delež zemeljskega pospeška *g*. Zaradi narave pojava in tudi lažje primerjave z obstoječimi študijami predpostavljamo naravno logaritemsko odvisnost. V preglednici 4 so podane največje, najmanjše in povprečne vrednosti vseh petih spremenljivk, uporabljenih v raziskavi. Pred zagonom programa smo dejanske vrednosti največjega pospeška tal logaritmirali in s tem pripravili v skladu s predpostavljeno funkcijsko odvisnostjo. Na sliki 4 lahko vidimo raztros podatkov v odvisnosti od spremenljivk M_{w}, R_{jb} in F.

Preglednica 4: Karakteristike podatkov po posameznih spremenljivkah

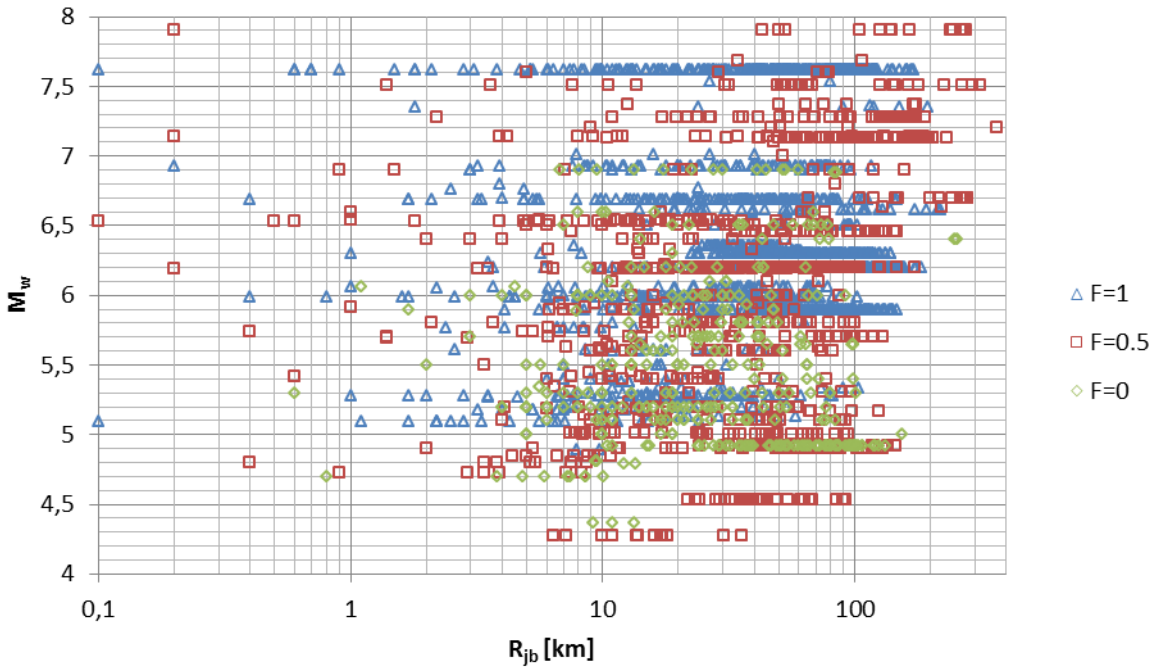
	PGA [g]	M_{w}	R_{jb}[km]	$V_{\mathrm{s,30}}$[m/s]	F
Najmanjša vr.	0,0012	4,27	0	116,4	0
Največja vr.	1,6615	7,90	365,1	2016,1	1
Povprečje	0,0939	6,25	57,1	420,5	0,74

Z vsemi izbranimi spremenljivkami sedaj lahko zapišemo napovedovalni problem:

$$\ln(PGA) = f(M_{\mathrm{w}}, R_{\mathrm{jb}}, V_{\mathrm{s,30}}, F) \tag{5}$$

3.2 Definicije gramatik

Za dosego dobrih rezultatov je izredno pomembna pravilna definicija gramatike [8]. Da bi se tej kar najbolj približali, smo definirali tri gramatike, vsako s svojo stopnjo vgradnje obstoječega znanja potresnega inženirstva. Cilj Splošne gramatike S je vključiti največ splošnosti in ob primerni globini omogočiti tvorjenje najrazličnejših izrazov. Evropska gramatika E je po drugi strani zelo specifična za področje, saj vključuje le izraze, ki so jih razvili drugi avtorji in so objavljeni v [2]. V Združeni gramatiki Z poskušamo združiti ta različna pristopa, s tem ko omogočimo



Slika 4: Porazdelitev meritev v odvisnosti od razdalje R_{jb} [km], magnitude M_{w} in vrste preloma F

Lagramgeu, da sam sestavi najboljšo enačbo iz različnih gradnikov, ki smo jih zasledili pri prepisovanju izrazov za gramatiko E. Po vsaki definiciji gramatike sledi še izpis opisa prostora enačb, ki ga omejuje, kot je opisano v poglavju 2.4.

3.2.1 Splošna gramatika S

Glavni cilj gramatike je, kot nakazuje že ime, čim večja splošnost, njena definicija pa je zapisana v preglednici 5. Produkcije so le skupek enostavnih matematičnih operatorjev in funkcij, ki jih lahko program s pomočjo v produkcije vgrajene rekurzije poljubno kombinira med seboj in s tem doseže najrazličnejše odvisnosti. Produkcija S1 omogoča povečanje števila členov enačbe, medtem ko produkcija S2 dovoljuje tvorjenje polinomskih členov pa tudi drugih kombinacij. Odštevanje v gramatiko ni vključeno, ker lahko z razvojem po produkcijah S2 in S9 pridobimo množenje z negativno konstanto in bi se tako odštevanje po nepotrebnem podvajalo. Deljenje (S3) je vključeno, saj smo raje omejili eksponent potenčne produkcije (S4) na majhne pozitivne vrednosti, se s tem izognili nefizikalnim velikim eksponentom in omogočili modeliranje inverznih odvisnosti. Produkcija S5 omogoča eksponentne odvisnosti, S6 pa logaritemske. Medtem ko smo te odvisnosti (S1-S6) zasledili v modelih iz [2], pa je trigonometrična odvisnost (S7) vključena zavoljo splošnosti gramatike. Produkciji S8 in S9 sta obvezni v vsaki gramatiki, saj vodita (S8 posredno prek predkončnega simbola *V*) h končnim simbolom in tako zagotavljata možnost ustavitve rasti izpeljevalnega drevesa.

Iz karakteristik prostora hipotez gramatike S v preglednici 6 lahko vidimo, zakaj je nujna omejitev največje globine drevesa s parametrom *-d*, saj zaradi rekurzivnosti skoraj vseh produkcij število možnih enačb eksponentno narašča in pri globini sedem že preseže število 10^{50}. Ob

Preglednica 5: Splošna gramatika S

Oznaka	Produkcija
S1	A → A + A;
S2	A →(A) * (A);
S3	A → (A) / (A);
S4	A → pow(A, const[_:0:0.1:4]);
S5	A → exp(A);
S6	A → log(A);
S7	A → cos(A);
S8	A → V;
S9	A → const[_:-1000:0.1:1000];

Preglednica 6: Opis prostora hipotez Splošne gramatike S

depth	#p.trees	max.len.	min.len.	#consts	der.len.
0	0	-1	-1	-1	-1
1	1	1	1	1	1
2	12	7	1	2	3
3	485	19	3	4	7
4	707620	43	6	8	15
5	1.50218e+12	91	9	16	31
6	6.76964e+24	187	12	32	63
7	1.37484e+50	379	15	64	127

tem se je treba zavedati, da s stališča algoritma npr. izraza $M + R$ in $R + M$ nista enaka in med preiskovanjem prostora enačb obravnava oba. Možnosti takih kombinacij je v tej gramatiki veliko, kar po nepotrebnem napihuje prostor hipotez in tako podaljšuje izčrpno iskanje. Prav tako lahko algoritem med kombiniranjem sestavi popolnoma neuporabne izraze kot npr. $\ln(\exp(M))$, ki je enakovreden enostavnejšemu M, kar spet nesmiselno povečuje prostor hipotez.

3.2.2 Evropska gramatika E

Pri razvijanju naslednje gramatike smo poskušali čim bolj izkoristiti možnost Lagramgea, da lahko v gramatiko vključimo specifično znanje izbranega področja. V letu 2011 je izšel zbirni članek vseh doslej znanih enačb, ki napovedujejo največji pospešek tal in spektralne pospeške [2]. V Evropski gramatiki E smo zbrali vse enačbe za največji pospešek tal iz [2], katerih avtorji raziskujejo na evropskih tleh oz. so predlagali enačbe za uporabo v Evropi. Od skupno 289 povzetih študij smo zbrali 57 različnih enačb iz 65 raziskav, zaradi preglednosti pa smo gramatiko priložili diplomskemu delu v prilogi A. Enačbam v prilogi sta pripisana številka in naslov podpoglavja v članku [2]; navedb ob isti enačbi je lahko več, če različni avtorji v svojih člankih predlagajo po zgradbi enako enačbo. Kjer prepisana enačba predvideva uporabo spremenljivk, ki jih v tej študiji nismo izbrali, smo te člene zanemarili oz. zamenjali s konstantnim členom (npr. globino *h* smo vedno zamenjali z $const[\dots]$). Ker se v enačbah pojavljata razdelitvi spremenljivk $V_{\mathrm{s},30}$ in F v razrede, za ta namen definiramo dve funkciji:

Preglednica 7: Opis prostora hipotez Evropske gramatike E

depth	#p.trees	max.len.	min.len.	#consts	der.len.
0	0	-1	-1	-1	-1
1	0	-1	-1	-1	0
2	56	104	9	24	36
3	57	104	40	24	36
4	57	104	-1	24	36

- '*ife*' (*if equal to*), ki preverja enakost, in
- '*ifl*' (*if less than*), ki primerja dve vrednosti po velikosti.

Ob izpolnjevanju preverjenega pogoja vrneta eno vrednost, v ostalih primerih pa drugo vrednost. Zapisani sta na začetku gramatike, njuna uporaba in gnezdenje v produkcijah pa omogočata zapis vseh formulacij, ki jih najdemo v [2]. Za lažjo berljivost enačb smo definirali več pomožnih produkcij, ki so zapisane na koncu gramatike. V preglednici 7 vidimo izpis značilnosti prostora hipotez. Ker gramatika ne vsebuje nobene rekurzivne produkcije, prostor hipotez vsebuje le 57 različnih formulacij, ne glede na večanje globine.

Vendar smo pri prepisovanju na predpisan Lagramgeov način naleteli na nemalo manjših težav, ki so opisane v nadaljevanju. Zaradi preglednosti enačb smo definirali posebni produkciji (*Ko* in *Po*), z uporabo katerih smo Lagramgeu dovolili modeliranje vseh oz. samo pozitivnih vrednosti koeficientov. V obstoječih enačbah so uporabljeni tako desetiški kot tudi naravni logaritmi, vendar smo se odločili za naravne in tako desetiške logaritme v skladu z enačbo (6) pretvorili v naravne, nelogaritmirane enačbe pa logaritmirali [12].

$$\log_a x = \log_a b \cdot \log_b x \tag{6}$$

Ker Lagramge v samih produkcijah samostojnih števil ne dovoljuje, smo zaradi večje preglednosti za vsako vrednost konstante v enačbah, ki so jo avtorji predpostavili pred umerjanjem drugih konstant, definirali njej lastno produkcijo, s čimer smo dobili 17 dodatnih produkcij (v gramatiki imajo skladnjo *K**).

3.2.3 Združena gramatika Z

V tretji gramatiki smo poskusili združiti ideji prejšnjih dveh gramatik in posplošili izraze, ki jih najdemo v gramatiki E. Trudili smo se ohraniti možnost generiranja vseh izrazov, ki nastopajo v gramatiki E, obenem pa omogočiti generiranje novih kombinacij posameznih členov. Ohranili smo funkciji *ife* in *ifl*, ki omogočata uporabo pogojnih stavkov v produkcijah, in prevzeli delitve spremenljivk $V_{\mathrm{s},30}$ in F na razrede, kot to predlagajo nekateri avtorji. Gramatika Z je priložena diplomskemu delu v prilogi B, tu navajamo le značilnosti prostora hipotez, ki kažejo, da prostor hipotez narašča približno s faktorjem 5.

Preglednica 8: Opis prostora hipotez Združene gramatike Z

depth	#p.trees	max.len.	min.len.	#consts	der.len.
0	0	-1	-1	-1	-1
1	0	-1	-1	-1	0
2	0	-1	-1	-1	2
3	0	-1	-1	-1	25
4	450	84	38	21	39
5	8100	106	51	26	49
6	41850	119	57	29	55
7	210600	132	63	32	61
8	1054350	145	69	35	67

3.3 Nastavitev ostalih vhodnih parametrov

3.3.1 Parameter *-d*

Parameter *-d* omejuje globino izpeljevalnega drevesa in s tem velikost prostora hipotez. Medtem ko je velikost prostora hipotez pri gramatiki E konstantna ob povečevanju višine nad 3, pa gramatiki S in Z vsebujeta rekurzivne produkcije, ki eksponentno širijo prostor hipotez. Zaradi dinamičnega alociranja potrebnega računalniškega spomina lahko posamezna instanca programa hitro zapolni ves spomin, ki ji je na voljo. Tako smo s poskušanjem ugotovili, da je pri gramatiki S največja še možna globina enaka 7, saj se je Lagramge ob zagonu poskusa z višino 8 enostavno sesul. Pri gramatiki Z pa težav s sesutjem programa nismo imeli, vendar smo omejili globino na vrednost 8 zaradi nesmiselnosti enačb, ki bi jih algoritem lahko tvoril pri večjih globinah.

3.3.2 Parameter *-b*

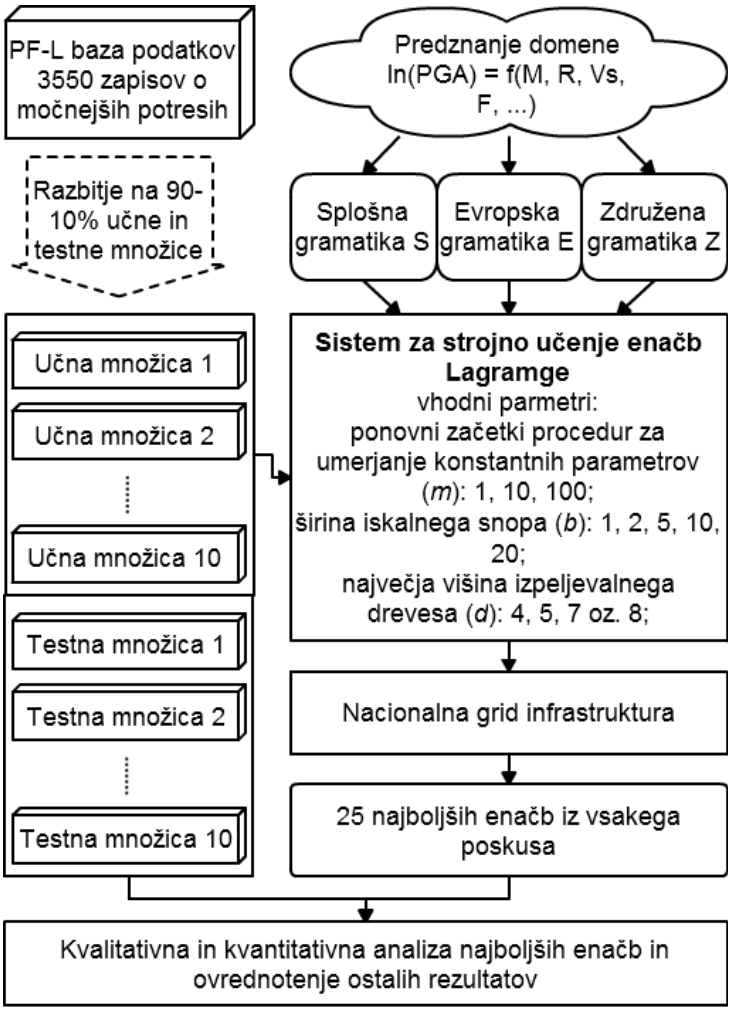
Gramatiki S in Z zaradi ogromne velikosti prostora hipotez omogočata le hevristično preiskovanje v snopu, saj bi izčrpno iskanje trajalo predolgo. Na iskanje v snopu ima parameter *-b* velik vpliv, saj določa širino snopa in s tem število preizkušenih enačb. Za preiskovanje smo določili vrednosti $b \in \{1, 2, 5, 10, 20, 50\}$, saj pri pogostejših korakih ne pričakujemo večjih razlik. Na izčrpno preiskovanje prostora hipotez parameter *-b* ne vpliva, saj le omeji število najboljših enačb, ki si jih Lagramge zapomni in na koncu izpiše, in tako ne zmanjšuje števila preizkušenih enačb. Zato pri poskusih z gramatiko E parametra *-b* nismo uporabili.

3.3.3 Parameter *-m*

Parameter *-m* označuje število ponovnih zagonov metod za umerjanje konstant na vhodne podatke, tj. metod downhill simplex in Levenberg-Marquardt. Te metode se pogosto ujamejo v lokalne minimume, zato z njihovimi ponovnimi zagoni z naključnimi vrednostmi poskušamo najti globalni minimum. Za preiskovanje smo določili vrednosti $m \in \{1, 10, 100\}$, saj pri pogostejših korakih ni pričakovati večjih razlik.

3.4 Infrastruktura za izvajanje

Iz začetnih poskusov smo ugotovili, da posamezne instance programa Lagramge potrebujejo veliko časa in razpoložljivega spomina. Zato smo poskuse izvajali na slovenski nacionalni grid infrastrukturi (NGI), za katero v sklopu Slovenske iniciative za nacionalni grid (SLING) skrbi Arnes. Uporabniki imajo na voljo 4268 procesorjev, ki so razdeljeni v pet računskih gruč. Celotna NGI se obnaša kot supergruča oz. grid, ki uporabnikom omogoča dostop do računalniških kapacitet in podatkovnih shramb, zagotavlja nameščanje programske opreme, varen dostop, označevanje podatkov, rezervacije, obračunavanje in beleženje uporabe itn. Izjemne zmogljivosti omogočajo uporabo novih raziskovalnih metod, ki pa so računsko zelo zahtevne [13]. NGI smo uporabili za paralelno izvajanje več instanc Lagramgea in tako pospešili pridobivanje rezultatov. Celotno strukturo raziskav lahko vidimo na sliki 5.



Slika 5: Prikaz strukture raziskave

Za namen širše uporabe programa Lagramge smo razvili tudi namensko spletno aplikacijo za zaganjanje poskusov na infrastrukturi SLING. Z njo lahko kdor koli zažene poskuse s svojimi podatki, gramatiko in nastavitvami ter pridobi rezultate hitreje, kot če bi poskuse izvajal zaporedoma. S spletnim uporabniškim vmesnikom je uporaba programa poenostavljena [7].

4 REZULTATI

V okviru raziskave smo zagnali več kot sto različnih poskusov, v katerih smo preizkušali nastavitve programa Lagramge in različne sestave gramatik. Tu podajamo rezultate poskusov z gramatiko S in hevrističnim preiskovanjem prostora hipotez; z gramatiko E in izčrpnim preiskovanjem prostora hipotez; z gramatiko Z in hevrističnim preiskovanjem prostora hipotez in z gramatiko Z in izčrpnim preiskovanjem prostora hipotez.

Dobljene najboljše enačbe smo v skladu s prakso navzkrižnega preverjanja uporabili za napoved največjega pospeška tal na podlagi podatkov iz pripadajočih testnih množic. Pri vseh kombinacijah *b* in *m* za gramatiki E in Z smo določili povprečno vrednost in standardni odklon izračunanega MSE najboljših enačb iz vseh desetih testnih množic razdelitve. Dobljene rezultate za gramatiki E in Z prikazujejo preglednice 9-12. Izmed vseh enačb, ki smo jih pridobili s poskusi, spodaj za vsak poskus predstavljamo tri enačbe z najboljšim MSE učne množice in pripadajočim MSE testne množice. Poleg tega prikažemo še razmerja med izmerjenimi in izračunanimi vrednostmi PGA za vsako izmed navedenih enačb, pri čemer 6 meritev, ki presegajo vrednost $1g$, ni vključenih.

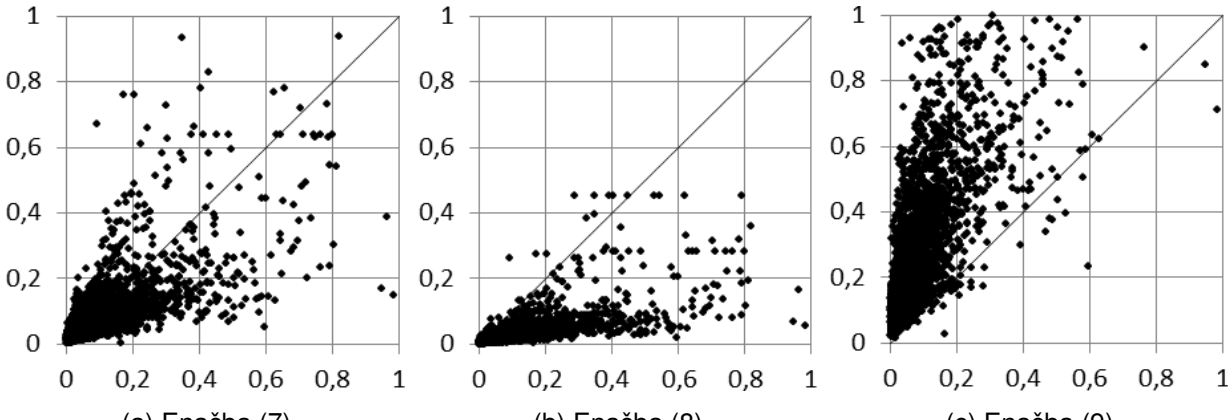
4.1 Splošna gramatika S, hevristično iskanje

Tu podajamo le tri najboljše enačbe glede na MSE testne množice, odkrite z gramatiko S, saj smo imeli nemalo težav pri zaganjanju poskusov zaradi velike porabe dinamičnega spomina in predolgega trajanja posameznih instanc programa (posamezne so trajale tudi več kot dva tedna). Enačba (7) je dosegla $MSE = 0,461798$, enačba (8) $MSE = 0,465409$ in enačba (9) $MSE = 0,465789$ na pripadajoči učni množici. Podobno so dosegle $MSE = 0,456534$, $MSE = 0,465659$ in $MSE = 0,459849$ na pripadajočih testnih množicah.

$$\ln(PGA) = -0,422 * (10,50 + M_{\mathrm{w}} + R_{\mathrm{jb}}^{0,438}) + \exp(-2,13) + M_{\mathrm{w}} \quad (7)$$

$$\ln(PGA) = -0,563 * (8,88 + M_{\mathrm{w}} + R_{\mathrm{jb}}^{0,393}) + M_{\mathrm{w}} + \log(M_{\mathrm{w}}) \quad (8)$$

$$\ln(PGA) = -0,640 * (\log(1000) + M_{\mathrm{w}} + R_{\mathrm{jb}}^{0,370}) + M_{\mathrm{w}} + \log(M_{\mathrm{w}}) \quad (9)$$



(a) Enačba (7) (b) Enačba (8) (c) Enačba (9)

Slika 6: Razmerja med izmerjenim in izračunanim PGA za najboljše tri enačbe gramatike S

4.2 Evropska gramatika E, izčrpno iskanje

Preglednica 9: Povprečje in standardni odklon kritrija MSE pri gramatiki E

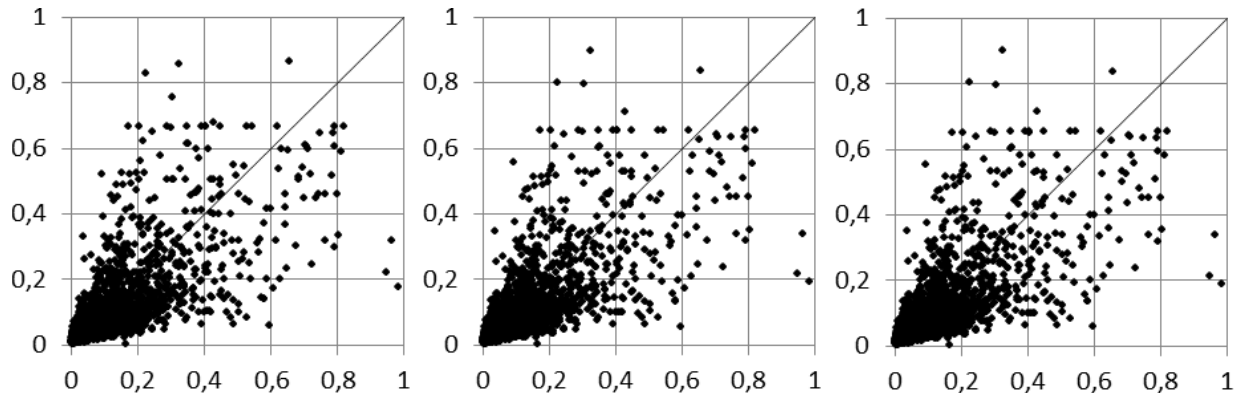
m	Povpr. MSE	Stand. odklon MSE
1	0,418	0,0260
10	0,417	0,0226
100	0,417	0,0258

Sledijo tri najboljše enačbe glede na MSE testne množice, odkrite z gramatiko E. Enačba (10) je dosegla $MSE = 0,379553$, enačba (11) $MSE = 0,380009$ in enačba (12) $MSE = 0,380966$ na pripadajoči učni množici. Podobno so dosegle $MSE = 0,410731$, $MSE = 0,411634$ in $MSE = 0,412068$ na pripadajočih testnih množicah.

$$\ln(PGA) = -7,17 + 2,11 \cdot M_{\mathrm{w}} - 0,120 \cdot M_{\mathrm{w}}^2 - 1,09 \cdot \ln\sqrt{R_{\mathrm{jb}}^2 + 48,3} + \begin{cases} 0 & V_{\mathrm{s,30}} < 360\frac{m}{s} \\ -0,262 & 360\frac{m}{s} \leq V_{\mathrm{s,30}} < 800\frac{m}{s} \\ -0,294 & 800\frac{m}{s} \leq V_{\mathrm{s,30}} \end{cases} + \begin{cases} 0 & normalen\ prelom \\ 0,00881 & reverzen\ prelom \\ -0,0744 & zmičen\ prelom \end{cases} \tag{10}$$

$$\ln(PGA) = 0,914 + 0,688697 \cdot (M_{\mathrm{w}} - 6) - 0,126 \cdot (M_{\mathrm{w}} - 6)^2 - 1,11 \cdot \ln\sqrt{R_{\mathrm{jb}}^2 + 50,9} + \begin{cases} 0,459 & V_{\mathrm{s,30}} < 180\frac{m}{s} \\ 0,296 & 180\frac{m}{s} \leq V_{\mathrm{s,30}} < 360\frac{m}{s} \\ 0,0518 & 360\frac{m}{s} \leq V_{\mathrm{s,30}} < 750\frac{m}{s} \\ 0 & 750\frac{m}{s} \leq V_{\mathrm{s,30}} \end{cases} \tag{11}$$

$$\ln(PGA) = -7,74 + 2,20 \cdot M_{\mathrm{w}} - 0,126 \cdot M_{\mathrm{w}}^2 - 1,10 \cdot \ln\sqrt{R_{\mathrm{jb}}^2 + 50,8} + \begin{cases} 0,293 & V_{\mathrm{s,30}} < 360\frac{m}{s} \\ 0,0430 & 360\frac{m}{s} \leq V_{\mathrm{s,30}} < 800\frac{m}{s} \\ 0 & 800\frac{m}{s} \leq V_{\mathrm{s,30}} \end{cases} \tag{12}$$



(a) Enačba (10) (b) Enačba (11) (c) Enačba (12)

Slika 7: Razmerja med izmerjenim in izračunanim PGA za najboljše tri enačbe gramatike E

4.3 Združena gramatika Z, izčrpno iskanje

Preglednica 10: Povprečje in standardni odklon kriterija MSE pri gramatiki Z in izčrpnem iskanju

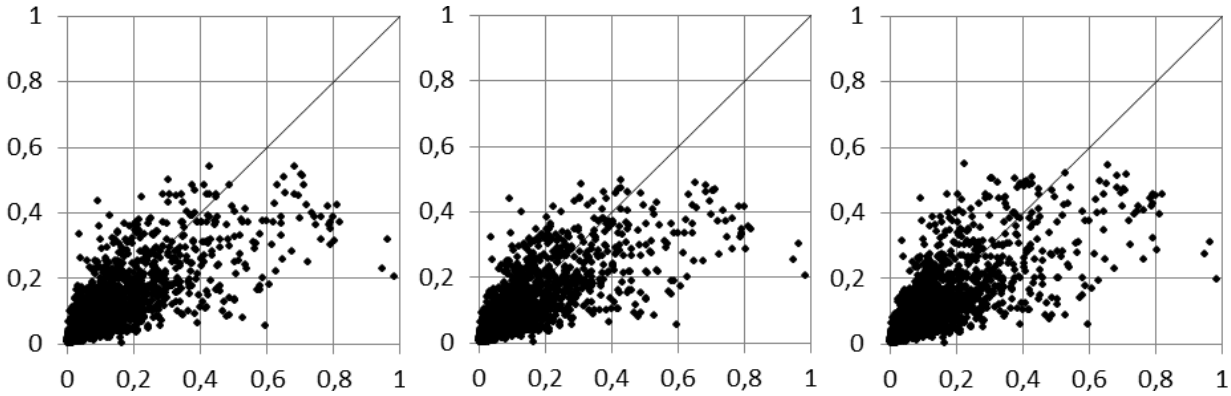
m	Povpr. MSE	Stand. odklon MSE
1	0,436	0,0404
10	0,436	0,0403
100	0,434	0,0398

Sledijo tri najboljše enačbe glede na MSE testne množice, odkrite z gramatiko Z z izčrpnim iskanjem. Enačba (13) je dosegla $MSE = 0,394081$, enačba (14) $MSE = 0,398482$ in enačba (15) $MSE = 0,400691$ na pripadajoči učni množici. Podobno so dosegle $MSE = 0,407487$, $MSE = 0,42625$ in $MSE = 0,38399$ na pripadajočih testnih množicah.

$$\begin{aligned}\ln(PGA) = {} & 1,37 - 0,00675 \cdot \exp(3,36 \cdot M_{\mathrm{w}} - 21,4) \\ & - 0,947 \cdot \exp(-0,174 \cdot M_{\mathrm{w}} + 0,614) \cdot \log(R_{\mathrm{jb}}^2 + 76,6) \\ & + \begin{cases} 0,503 & V_{\mathrm{s},30} < 180\frac{m}{s} \\ 0,296 & 180\frac{m}{s} \leq V_{\mathrm{s},30} < 360\frac{m}{s} \\ 0,0775 & 360\frac{m}{s} \leq V_{\mathrm{s},30} < 800\frac{m}{s} \\ 0 & 800\frac{m}{s} \leq V_{\mathrm{s},30} \end{cases}\end{aligned} \tag{13}$$

$$\begin{aligned}\ln(PGA) = {} & 0,945 - 0,174 \cdot \exp(2,51 \cdot M_{\mathrm{w}} - 18,1) \\ & - 0,845 \cdot \exp(-0,178 \cdot M_{\mathrm{w}} + 0,748) \cdot \log(R_{\mathrm{jb}}^2 + 83,4) \\ & - 0,290 \cdot \log(V_{\mathrm{s},30}/3390)\end{aligned} \tag{14}$$

$$\begin{aligned}\ln(PGA) = {} & 1,23 - 0,118 \cdot (M_{\mathrm{w}} - 6,60)^2 \\ & - 0,190 \cdot \exp(-0,130 \cdot M_{\mathrm{w}} + 1,90) \cdot \log(R_{\mathrm{jb}}^2 + 57,3) \\ & - 0,311 \cdot \log(V_{\mathrm{s},30}/464) + \begin{cases} 0,0951 & normalen\ prelom \\ 0,0720 & reverzen\ prelom \\ 0 & zmičen\ prelom \end{cases}\end{aligned} \tag{15}$$



(a) Enačba (13) (b) Enačba (14) (c) Enačba (15)

Slika 8: Razmerja med izmerjenim in izračunanim PGA za najboljše tri enačbe gramatike Z pri izčrpnem iskanju

4.4 Združena gramatika Z, hevristično iskanje

Preglednica 11: Povprečni MSE pri gramatiki Z in hevrističnem iskanju

m\b	1	2	5	10	20	50
1	0,443	0,443	0,412	0,402	0,404	0,405
10	0,452	0,452	0,452	0,452	0,451	0,451
100	0,444	0,449	0,438	0,404	0,400	0,405

Preglednica 12: Standardni odklon MSE pri gramatiki Z in hevrističnem iskanju

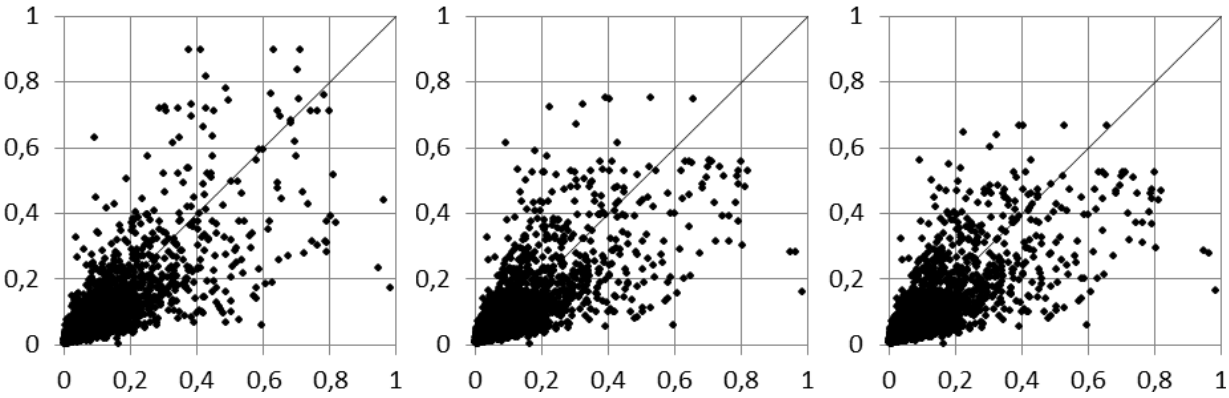
m\b	1	2	5	10	20	50
1	0,0264	0,0264	0,0271	0,0290	0,0240	0,0262
10	0,0255	0,0255	0,0255	0,0255	0,0255	0,0255
100	0,0260	0,0248	0,0262	0,0278	0,0238	0,0254

Sledijo tri najboljše enačbe glede na MSE testne množice, odkrite z gramatiko Z in hevrističnim iskanjem. Enačba (16) je dosegla $MSE = 0,361573$, enačba (17) $MSE = 0,361573$ in enačba (18) $MSE = 0,36203$ na pripadajoči učni množici. Podobno so dosegle $MSE = 0,384603$, $MSE = 0,396234$ in $MSE = 0,394893$ na pripadajočih testnih množicah.

$$\begin{aligned}\ln(PGA) = &- 0,284 + 0,145 \cdot M_{\mathrm{w}} - 59,2 \cdot \exp(-6,20 \cdot M_{\mathrm{w}}) + 0,204 \cdot \exp(0,393 \cdot M_{\mathrm{w}}) \\ &- 0,000130 \cdot \frac{\exp(1,87 \cdot M_{\mathrm{w}})}{R_{\mathrm{jb}} + 81,7} - 1,51 \cdot \ln(R_{\mathrm{jb}} + 10,9) \\ &+ \begin{cases} 0,448 & V_{\mathrm{s,30}} < 180\frac{m}{s} \\ 0,275 & 180\frac{m}{s} \leq V_{\mathrm{s,30}} < 360\frac{m}{s} \\ 0,0340 & 360\frac{m}{s} \leq V_{\mathrm{s,30}} < 800\frac{m}{s} \\ 0 & 800\frac{m}{s} \leq V_{\mathrm{s,30}} \end{cases} + \begin{cases} 0,0897 & normalen\ prelom \\ 0,138 & reverzen\ prelom \\ 0 & zmičen\ prelom \end{cases}\end{aligned} \tag{16}$$

$$\begin{aligned}\ln(PGA) = &- 8,23 - 0,133 \cdot M_{\mathrm{w}}^2 + 2,30 \cdot M_{\mathrm{w}} + 44,5 \cdot \exp(-10,5 \cdot M_{\mathrm{w}}) \\ &- 0,550 \cdot \ln(R_{\mathrm{jb}}^2 + 8,60 \cdot M_{\mathrm{w}}) \\ &+ \begin{cases} 0,354 & V_{\mathrm{s,30}} < 455\frac{m}{s} \\ 0 & V_{\mathrm{s,30}} \geq 455\frac{m}{s} \end{cases} + \begin{cases} 0,110 & normalen\ prelom \\ 0,0760 & reverzen\ prelom \\ 0 & zmičen\ prelom \end{cases}\end{aligned} \tag{17}$$

$$\begin{aligned}\ln(PGA) = &- 8,35 + 2,34 \cdot M_{\mathrm{w}} - 0.135 \cdot M_{\mathrm{w}}^2 - 0,555 \cdot \ln(R_{\mathrm{jb}}^2 + 1,48 \cdot M_{\mathrm{w}}^2) \\ &+ \begin{cases} 0,348 & V_{\mathrm{s,30}} < 458\frac{m}{s} \\ 0 & V_{\mathrm{s,30}} \geq 458\frac{m}{s} \end{cases} + \begin{cases} 0,107 & normalen\ prelom \\ 0,0731 & reverzen\ prelom \\ 0 & zmičen\ prelom \end{cases}\end{aligned} \tag{18}$$



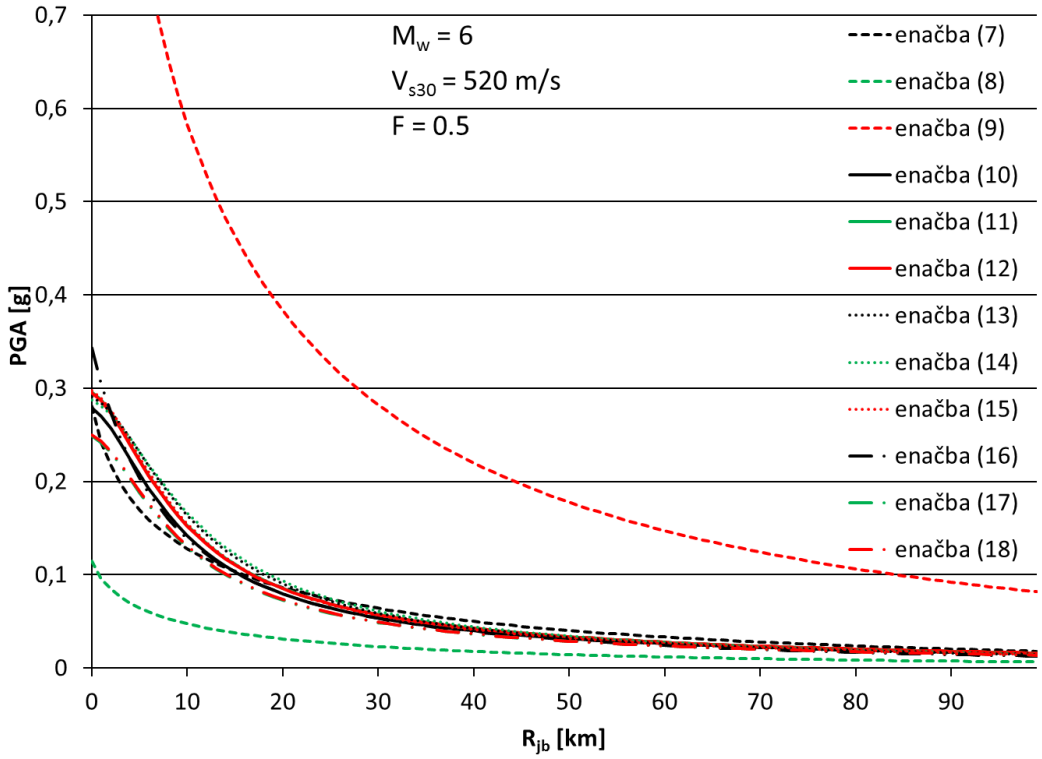
(a) Enačba (16) (b) Enačba (17) (c) Enačba (18)

Slika 9: Razmerja med izmerjenim in izračunanim PGA za najboljše tri enačbe gramatike Z in hevristično iskanje

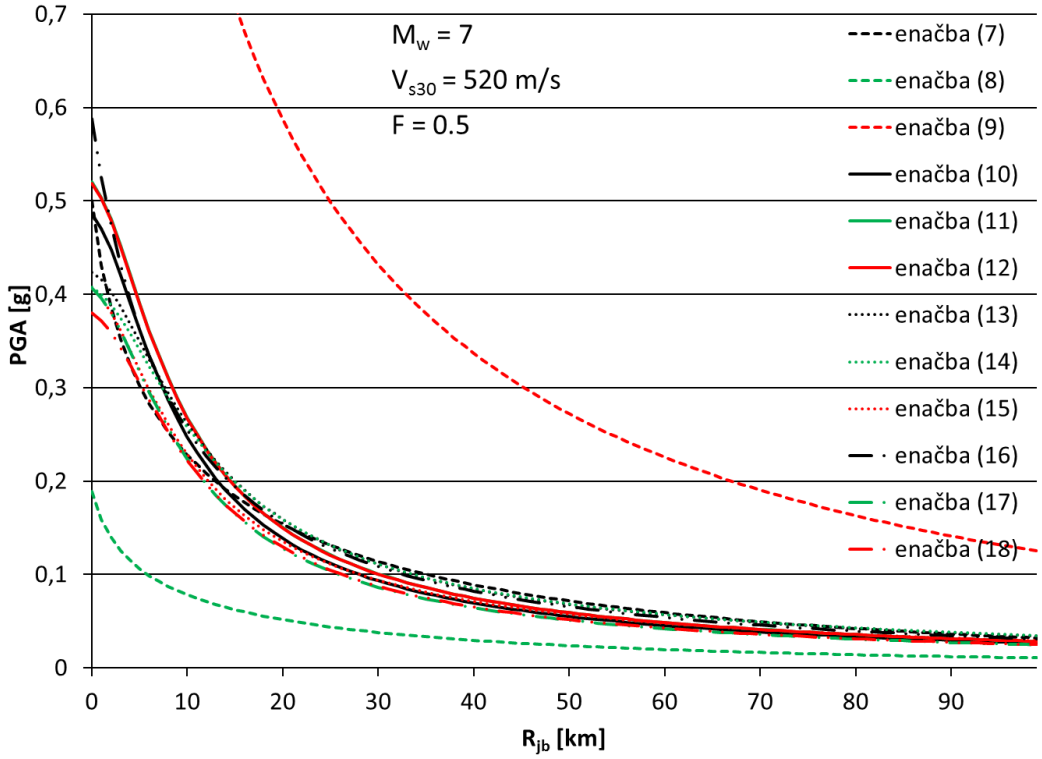
4.5 Skupni rezultati

Za lažjo primerjavo ugotovljenih funkcijskih odvisnosti lahko na sliki 10 vidimo grafe vseh dvanajstih enačb (7)-(18). Narisani sta odvisnosti največjega pospeška tal PGA od razdalje R_{jb} pri zmičnem prelomu $F = 0,5$, hitrosti strižnih valov $V_{\mathrm{s},30} = 520\frac{m}{s}$ in dveh magnitudah (a) $M = 6$ oz. (b) $M = 7$.

Najboljša enačba izmed vseh dvanajstih je enačba (16). Na sliki 11 vidimo njeno primerjavo z nekaterimi enačbami iz raziskave [10] in enačbe (1) [3].

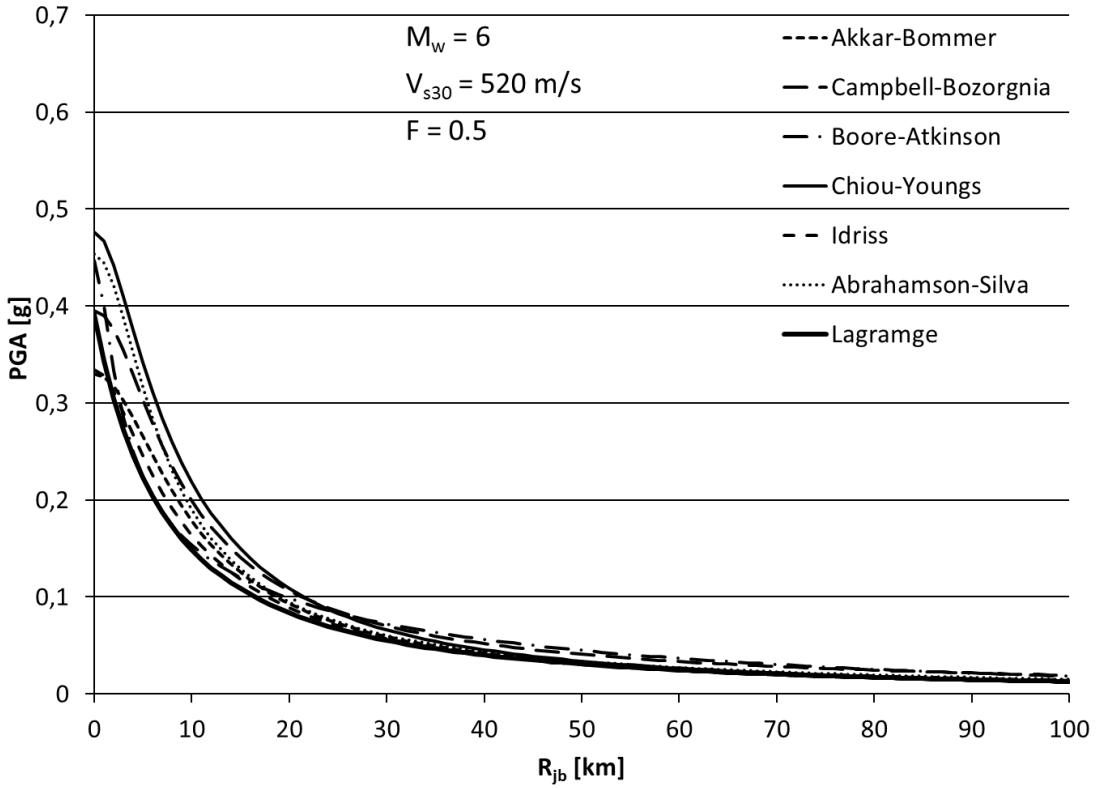


(a) $M = 6$

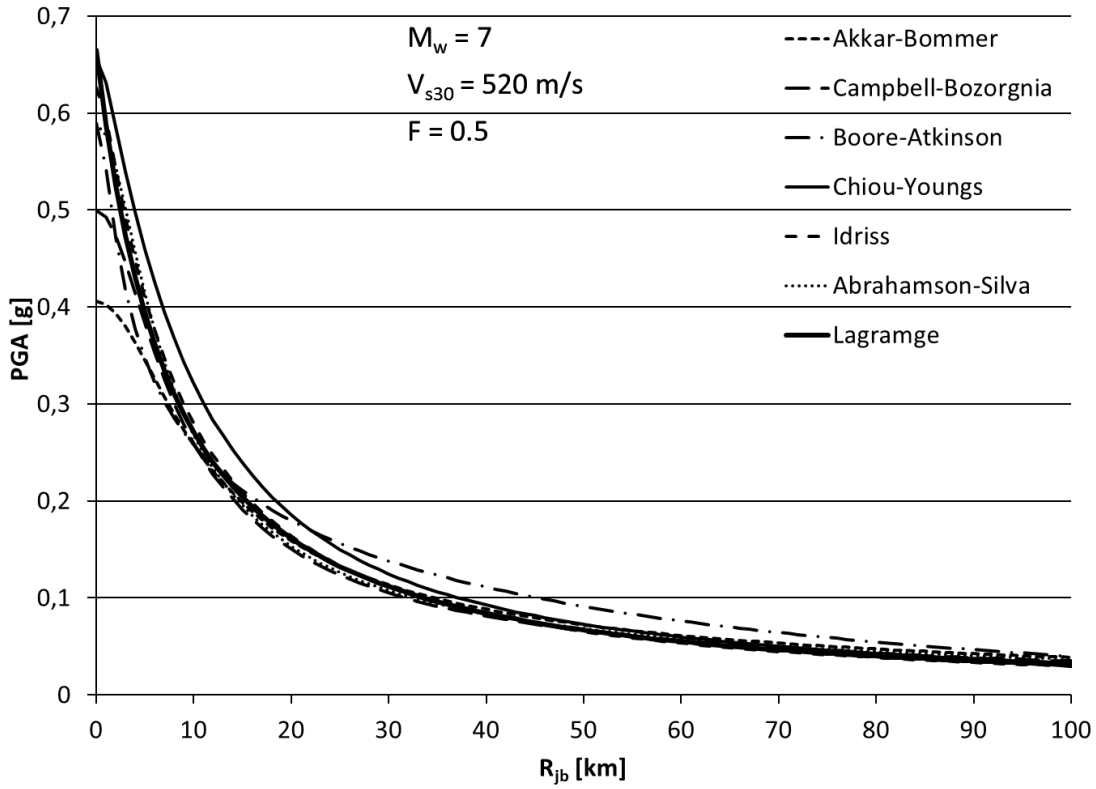


(b) $M = 7$

Slika 10: Odvisnost PGA [g] od R_{jb} [km] enačb (7)-(18) pri $F = 0,5$, $V_{\mathrm{s},30} = 520\frac{m}{s}$ in dveh magnitudah (a) $M = 6$ oz. (b) $M = 7$



(a) $M = 6$



(b) $M = 7$

Slika 11: Odvisnost PGA $[g]$ od R_{jb} $[km]$ enačbe (16), primerjane z izbranimi študijami NGA [10] in enačbo (1) [3] pri $F = 0,5$, $V_{\mathrm{s},30} = 520\frac{m}{s}$ in dveh magnitudah (a) $M = 6$ oz. (b) $M = 7$

5 RAZPRAVA

Namen pričujoče študije je bil preizkus algoritma strojnega učenja za odkrivanje enačb Lagramge na inženirskem problemu napovedovanja največjega pospeška tal. Lagramge s pomočjo uporabniško podane kontekstno neodvisne gramatike omeji prostor možnih enačb in preizkusi le strukture, ki jih lahko razvije z upoštevanjem pravil gramatike. Z vključitvijo obstoječega znanja v gramatiko lahko tako pomembno vplivamo na strukturo želenih enačb in s tem proces odkrivanja enačb usmerimo v pravilnejšo smer ter dobimo zadovoljive rezultate v sprejemljivem času.

V tej raziskavi smo razvili tri kontekstno neodvisne gramatike, pri čemer je vsaka temeljila na drugačnem pristopu. V Splošno gramatiko S (preglednica 5) smo vključili vse operatorje, ki nastopajo v obstoječih enačbah pojemanja, in s pomočjo rekurzivnih klicev dovolili njihovo poljubno kombiniranje. V Evropsko gramatiko E (priloga A) smo prepisali enačbe pojemanja, ki so jih v zadnjih 50 letih predlagali evropski znanstveniki. V Združeni gramatiki Z (priloga B) smo gornja pristopa združili tako, da smo v gramatiko vključili znanje stroke v obliki že utemeljenih členov in funkcij (prepisanih za gramatiko E), ki pa jih je program lahko kombiniral na poljubne načine ter s tem sestavljal in preizkušal nove izraze.

V nadaljevanju predstavljamo kvantitativno in kvalitativno ovrednotenje dobljenih rezultatov. Ugotovljamo, da ima program ob pravilni uporabi kontekstno neodvisnih gramatik velik potencial tako za reševanje obravnavanega in sorodnih problemov kot tudi problemov drugih inženirskih področij, kjer so v uporabi empirične enačbe.

5.1 Kvantitativni kriteriji

V pričujoči raziskavi nas je zanimal vpliv zasnove gramatike na kriterij MSE. Poleg tega smo preverili tudi vpliv izbire dveh vhodnih parametrov v sistemu Lagramge na rezultate – število enačb, ki si jih algoritem zapomni v vsakem koraku pri hevrističnem preiskovanju v snopu *b* in število ponovnih zagonov funkcij optimizacije konstantnih parametrov *m*. V preglednicah 9-12 so zbrana povprečja in standardni odkloni kriterija MSE najboljših enačb posameznih kombinacij *b* in *m* za vseh deset razdelitev baze podatkov na učni in testni del. Če se najprej osredotočimo na vrednosti standardnih odklonov, opazimo da so vse med 5-10 % vrednosti pripadajočega povprečja, kar pomeni, da so povprečja primerljiva med seboj. Kot splošno značilnost nadalje opazimo, da se povprečje MSE z višanjem vrednosti parametra *b* znižuje, vendar se zniževanje zmanjšuje z večjimi vrednostmi *b*. Zmanjšanje napake pri povečanju $b = 1$ na $b = 50$ je kar 10-odstotno, zato v nadaljnih študijah priporočamo uporabo velikih vrednosti parametra *b* (glej [6]). S povečevanjem vrednosti parametra *m* se napaka občutno ne spremeni, večinoma se zmanjša za približno 1-3 % pri višjih vrednostih *m*.

Če primerjamo vrednosti napak med posameznimi preglednicami 9-12, vidimo, da so enačbe, ki so jih predlagali drugi avtorji in smo jih mi zgolj prepisali v gramatiko E, že zelo dobre, saj so dosegle manjšo napako v primerjavi z drugimi poskusi, kar bi lahko bila posledica uporabe podobne baze podatkov pri izdelavi teh modelov. Tako je le hevristično preiskovanje z gramatiko Z pri velikih vrednostih *b* doseglo še manjšo napako (za okoli 1 %), kjer so tudi razlike med učnim

in testnim MSE manjše. Na podlagi primerjave napak najboljših enačb lahko zaključimo, da je poskus z gramatiko S dal najslabše rezultate, boljša sta poskus z izčrpnim preiskovanjem pri gramatiki Z in poskus z gramatiko E, najboljše rezultate pa smo dobili pri poskusu s hevrističnim preiskovanjem pri gramatiki Z.

Na slikah 6-9 so predstavljena ujemanja napovedanih z izmerjenimi vrednostmi največjega pospeška tal. Pri večini se opazi določena najvišja napovedana vrednost za uporabljeno bazo podatkov, npr. $PGA_{\max} \approx 0.91g$ na sliki 9a oz. $PGA_{\max} \approx 0.68g$ na sliki 9c. Ugotavljamo, da so točke na slikah 6a, 7 in 9 približno enakomerno razporejene na obeh straneh črte popolnega ujemanja $y = x$. Medtem pa so točke na slikah 6b in 8 večinoma pod črto, predvsem pri višjih vrednostih, kar nakazuje podcenjevalno lastnost enačb (8) in (13)-(15). Obratno pa enačba (9) precenjuje problem, kar je razvidno s slike 6c.

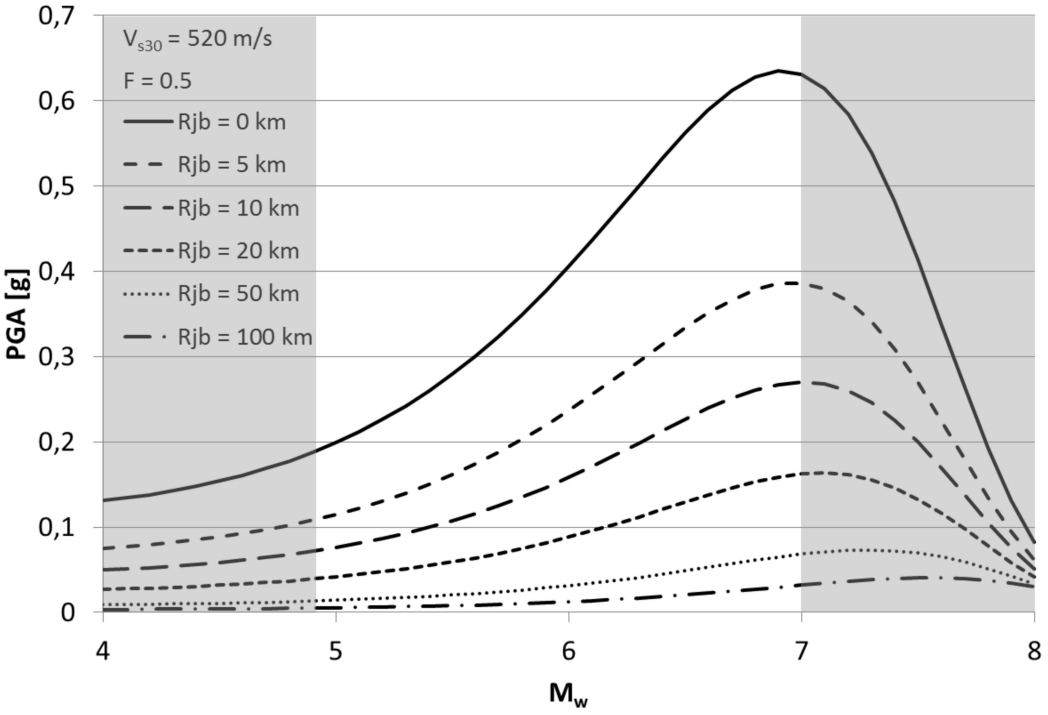
5.2 Kvalitativni kriteriji

Inženirjem ne zadošča zgolj matematična pravilnost enačb, ki jih uporabljajo, temveč tudi pravilen fizikalni pomen količin in medsebojnih zvez, ki nastopajo v enačbi. Zveze in strukturo enačbe je predpostavil Lagramgeev algoritem, ki je sledil navodilom podane gramatike, zato jih je treba pred uporabo v praksi kritično oceniti. Enačbe (7)-(9), ki jih je Lagramge sestavil z gramatiko S, so po kvantitativnem kriteriju precej netočne in strukturno preveč enostavne. Veliko členov ni niti pomnoženih s konstanto, ki bi jih ustrezno umerila na opazovani pojav, zato so te enačbe zanimive zgolj z vidika uporabe gramatik in iskalnih postopkov. S stališča potresnega inženirstva lahko kljub temu pridobimo fizikalno primerno rešitev, saj enačba (7) v primerjavi z drugimi enačbami podobno modelira problem (glej sliki 6a in 10). Enačbe (10)-(12) so razvili drugi avtorji in se nam tako o njihovi fizikalnosti ni treba posebej spraševati. Enačbe (13)-(15) vsebujejo veliko členov, pri katerih je magnituda v eksponentu z osnovo e, iz česar bi lahko sklepali, da ima magnituda bolj nelinearen vpliv na parametre gibanja tal, kot se je predpostavljalo v preteklosti. Vsaka izmed enačb drugače modelira odvisnosti od $V_{\mathrm{s},30}$ in F – dve modelirata strižno hitrost kot zvezno spremenljivko, medtem ko najboljša enačba modelira razdelitev v štiri razrede. Najboljši dve enačbi sta zanemarili vpliv vrste preloma, medtem ko ga enačba (15) vključuje. Prav tako v enačbi (15) nastopa še kvadratni člen magnitude. Enačbe (16)-(18) se, kar se tiče zgradbe, precej bolj razlikujejo od drugih. Prva vsebuje veliko členov z magnitudo, tudi v eksponentu, razdelitev zemljin na štiri razrede in neobičajen člen $\exp(1,87002 \cdot M_{\mathrm{w}}) \cdot (R_{\mathrm{jb}} + 81,6953)^{-1}$. Fizikalni pomen tega člena bi bilo treba še raziskati, saj je ta enačba dosegla najmanjšo napako na svoji učni in testni množici izmed vseh enačb, ki smo jih preizkusili tekom naše raziskave. Drugi dve enačbi sta si po strukturi podobni, saj obe predpostavljata delitev zemljin na dva razreda z računalniško izračunano razmejitvijo med njima (obe okoli 450$\frac{m}{s}$), vendar boljša vsebuje še člen z magnitudo v eksponentu in ima linearno odvisnost magnitude v členu z razdaljo.

Modele med seboj in z drugimi običajno primerjamo le v prvih 50 km, kjer se med seboj tudi najbolj razlikujejo [11], kar je opazno tudi na slikah 10 in 11. Na sliki 10 lahko hitro vidimo, da enačbi (8) in (9) modelirata največji pospešek tal popolnoma drugače od vseh drugih odkritih enačb, saj prva problem podcenjuje, druga pa precenjuje. Zaradi tega predlagamo, da se v prihodnjih študijah poleg MSE uporabi tudi druge kriterije, npr. RMSE (*angl. Root Mean*

Squared Error) ali MAE (*angl. Mean Absolute Error*) [6, 8]. Najboljšo enačbo po kriteriju MSE (16) primerjamo s raziskavami NGA [10] in študijo [3] na sliki 11. Na njej lahko opazimo precej dobro ujemanje z obstoječimi raziskavami in malo večji vpliv spremembe magnitude na spremembo PGA, saj je graf pri $M = 6$ (slika 11a) na spodnji meji grafov drugih avtorjev, medtem ko je graf pri $M = 7$ (slika 11b) na sredini do zgornji meji grafov drugih avtorjev.

Pri uporabi empiričnih enačb v praksi je prav tako kot oblika enačbe pomembno tudi njeno območje veljavnosti, saj se naj bi taka enačba uporabljala zunaj tega območja (v t. i. “sivi coni”) z veliko previdnostjo [11]. Če torej sklepamo po bazi podatkov, so v tej študiji razvite enačbe veljavne, kjer leži večina naših podatkov: $4,9 < M_{\mathrm{w}} < 7,6$ in $0\ km < R_{\mathrm{jb}} < 200\ km$ (glej sliko 4). A vendar, ko pogledamo graf odvisnosti PGA od magnitude M_{w} na sliki 12, opazimo, da se začne graf spuščati, ko magnituda preseže vrednost 7, kar pa je fizikalno nespremenljivo. Drugi avtorji enačb pojemanja se takim pojavom izognejo s posebnimi prijemi ali z začetno izbiro fizikalnejšega nastavka. Zato omejimo uporabo naših modelov znotraj $4,9 < M_{\mathrm{w}} < 7,0$ in $0\ km < R_{\mathrm{jb}} < 200\ km$.



Slika 12: Odvisnost PGA [g] od M_{w} enačbe (16) z označeno “sivo cono” uporabe

5.3 Prihodnje delo

Pri uporabi programa Lagramge smo naleteli na nemalo težav in nevšečnosti, ki bi se jih v prihodnjih raziskavah dalo odpraviti. Algoritem pri izvajanju porabi veliko dinamičnega spomina, saj smo med poskusi zasledili porabo tudi do 200 GB za posamično instanco programa [6]. Z optimizacijo alociranja spomina bi se tako lahko omogočilo predvsem globlje preiskovanje prostorov hipotez, kar v tem trenutku ni možno. Na izračune smo v povprečju morali čakati 2-4 dni, najdaljši poskus pa je tekel kar pol leta. Z dostopom do grid infrastrukture [13] smo razvili spletno aplikacijo [7], ki omogoča vzporedno izvajanje posamičnih instanc, vendar bi se

poleg tega posamičen postopek raziskovanja prostora hipotez lahko razdelil med več procesorjev (paraleliziral), s čimer bi se izvajanje programa še dodatno pospešilo. Ena od možnosti za nadaljnji razvoj bi bila tudi izdelava prijaznega uporabniškega vmesnika, ki bi uporabnika usmerjal med znanjem stroke in dobljenim MSE.

V prihodnje bi bilo dobro dodatno raziskati nelinearni vpliv magnitude, ki se pojavlja v enačbah (13)-(15). Iz zgoraj napisanega sklepamo, da se problema napovedovanja največjega pospeška tal še nekaj časa ne bo dalo rešiti zgolj z avtomatskim modeliranjem. Z zelo različnimi izrazi dobimo precej podobne rešitve in primerljive medsebojne vrednosti napake MSE, pri čemer pa sploh ni nujno, da imajo posamezni izrazi kakršen koli resen fizikalni pomen. Sodeč po grafu na sliki 8c ima enačba 8c lepšo razporeditev na gornji in spodnji strani črte popolnega ujemanja kot enačbi 8a oz. 8b, kar nakazuje hipotezo za nadaljnje raziskave, da enačba z najmanjšo napako ni nujno najbolj fizikalna in najbolj primerna za opis izbranega pojava.

Na področju potresnega inženirstva so se s pridobitvijo solidnih rezultatov v tej raziskavi odprle nove možnosti za nadaljnje raziskave, saj poleg največjega pospeška tal obstajajo tudi drugi parametri gibanja tal, za katere bi lahko na podoben način poiskali novo enačbo, npr. spektralni pospeški, za katere je bilo do leta 2011 objavljenih 188 modelov [2]. Z vedno boljšim poznavanjem narave dogodka so enačbe začele vključevati še dodatne neodvisne spremenljivke, npr. nelinearen vpliv tal, vpliv krovnine ipd., zaradi česar bi se celoten postopek odkrivanja enačb lahko ponovil z razširjeno bazo podatkov in novimi ali izboljšanimi produkcijami v uporabljenih gramatikah.

VIRI IN LITERATURA

[1] Slovar slovenskega knjižnega jezika. 2005. Izdala Slovenska akademija znanosti in umetnosti in Znanstvenoraziskovalni center Slovenske akademije znanosti in umetnosti, Inštitut za slovenski jezik Frana Ramovša. Ljubljana, DZS: 1714 str.

[2] Douglas, J. 2011. Ground-motion prediction equations 1964-2010. Final report, BRGM/RP-59356-FR and PEER/2011/102. Berkeley, University of California, College of Engineering: 444 str.

[3] Akkar, S., Bommer, J.J. 2010. Empirical equations for the prediction of PGA, PGV, and spectral accelerations in Europe, the Mediterranean region, and the Middle east. Seismological Research Letters 81, 2: 195–206.

[4] Rätsch, G. 2004. A Brief Introduction into Machine Learning. V: Proceedings of 21st Chaos Communication Congress 21C3. Creative Commons, Nemčija: 6 str.

[5] Todorovski, L. 1998. Declarative Bias in Equation Discovery. Magistrsko delo. Ljubljana, Univerza v Ljubljani, Fakulteta za računalništvo in informatiko (samozaložba L. Todorovski): 59 str.

[6] Markič, Š., Stankovski, V. 2013. An Equation-Discovery Approach to Earthquake-Ground-Motion Prediction. Engineering Applications of Artificial Intelligence 26, 4: 1339–1347. doi:10.1016/j.engappai.2012.12.005

[7] Markič, Š., Dirnbek, J., Stankovski, V. 2013. A Grid Application for Equation Discovery in the Earthquake Engineering Domain. In: Proceedings of the 3rd International Confernce on Parallel, Distributed, Grid and Cloud Computing for Engineering, Civil-Comp Press: loč. pag.

[8] Markič, Š., Stankovski, V. 2013. Developing Context-Free Grammars for Equation Discovery: An Application in Earthquake Engineering. http://iea-aie2013.few.vu.nl/program.shtml (Pridobljeno 15. 4. 2013.)

[9] Douglas, J. 2003. Earthquake ground motion estimation using strong-motion records: a review of equations for the estimation of peak ground acceleration and response spectral ordinates. Earth-Science Reviews 61, 1-2: 43-104.

[10] Special Issue on the Next Generation Attenuation Project. 2008. Stewart, J. P. (ur.) et al. Earthquake Spectra 24, 1: 341.

[11] Peruš, I., Fajfar, P. 2010. Ground-motion prediction by a non-parametric approach. Earthquake Engineering & Structural Dynamics 39, 12: 1395–1416.

[12] Bronstein, I. N. et al. 2009. Matematični priročnik. Ljubljana, Tehniška založba Slovenije: 967 str.

[13] Slovenska iniciativa za nacionalni grid (SLING). 2012. http://www.sling.si (Pridobljeno 16. 6. 2012)

SEZNAM PRILOG

Ta stran je namenoma prazna.

Priloga A Evropska gramatika E

Tukaj prilagamo Evropsko gramatiko E, kakršna je bila uporabljena v poskusih. Več o njeni izpeljavi je zapisano v poglavju 3.2.2.

```
%{
#include <math.h>
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);
}
%}
E -> Ko + Ko * Ma + Ko * log(Ra);
//2.12 Ambraseys 1975a, 1975b, 1978a; 2.16 Ambraseys 1978b; 2.122 Bommer et
    al 1996
E -> Po + Po * Ma - Po * log(Ra + K25);
//2.18 Faccioli 1978
E -> Ko + Ko * Ma + Ko * log(Ra + K25);
//2.22 Faccioli 1979
E -> Ko + Ko * Ma + Ko * log(Ra + Ko);
//2.23 Faccioli & Agalbato 1979; 2.59 Petrovski & Marcellini 1988; 2.84
   Stamatovska & Petrovski 1991; 2.205 Skarlatoudis et al. 2004
E -> Ko + Ko * Ma + Ko * log(Ra + Ko * exp(Ko * Ma));
//2.34 PML 1982; 2.46 PML 1985
E -> Po + Po * Ma - Po * log(Ra);
//2.35 Schenk 1982; 2.40 Schenk 1984
E -> Ko + Ko * Ma + Ko * log(Ra + Ko * exp(Ko * Ma)) + ife(Fa, K1, Ko, K0);
//2.46 PML 1985
E -> Ko + Ko * Ma + Ko * log(sqrt(pow(Ra, K2) + Ko)) + ifl(Vs, K800, Ko, K0
   );
//2.50 Sabetta & Pugliese 1987; 2.260 Massa et al. 2008; 2.276 Rupakhety &
   Sigbjörnsson 2009
E -> Ko + Ko * Ma + Ko * log(sqrt(pow(Ra, K2) + Ko * exp(Ko * Ma))) + ifl(
   Vs, K800, Ko, K0);
//2.50 Sabetta & Pugliese 1987
E -> Ko + Ko * Ma - log(sqrt(pow(Ra, K2) + Ko)) / log(K10) + Ko * sqrt(pow(
   Ra, K2) + Ko);
//2.67 Ambraseys 1990; 2.74 Ambraseys & Bommer 1991, 1992; 2.88 Sigbjö
   rnsson & Baldvinsson 1992; 2.118 Sarma & Free 1995; 2.165 Smit et al.
   2000; 2.191 Sigbjörnsson & Ambraseys 2003
E -> Po + Po * Ma - Po * log(Ra) - Po * Ra;
//2.72 Sigbjörnsson 1990
E -> Ko + Ko * Ma - K083 / log(K10) * log(sqrt(pow(Ra, K2) + Ko)) + Ko *
   sqrt(pow(Ra, K2) + Ko);
//2.74 Ambraseys & Bommer 1991, 1992
E -> Ko + Ko * Ma - K05 * log(Ra) - Po * Ra;
//2.76 García–Fernández & Canas 1991, 1995
```

```
E -> Ko + Ko * Ma  + Ko * log(sqrt(pow(Ra, K2) + Ko))+ Ko * sqrt(pow(Ra, K2
    ) + Ko);
//2.86 Ambraseys 1992; 2.102 Ambraseys & Srbulov 1994; 2.113 Ambraseys
    1995; 2.128 Sarma & Srbulov 1996; 2.191 Sigbjörnsson & Ambraseys 2003
E -> Ko + Ko * Ma + Ko * log(Ra + Ko) + ifl(Vs, K800, Ko, K0);
//2.92 Theodulidis & Papazachos 1992
E -> Ko + Ko * Ma - log(Ra) + Ko * Ra;
//2.108 Musson et al. 1994
E -> Ko + Ko * Ma + Ko * Ra + log(ifl(Ra, K100, (K1 / Ra), (pow(K100 / Ra,
    K083) / (K100))));
//2.108 Musson et al. 1994
E -> Ko + Ko * Ma + Ko * log(sqrt(pow(Ra, K2) + Ko * exp(Ko * Ma))) + Ko *
    sqrt(pow(Ra, K2) + Ko * exp(Ko * Ma));
//2.113 Ambraseys 1995
E -> Ko + Ko * Ma + Ko * log(sqrt(pow(Ra, K2) + Ko)) + Ko * sqrt(pow(Ra, K2
    ) + Ko) + Ko * log(Vs);
//2.113 Ambraseys 1995
E -> Ko + Ko * Ma + Ko * pow(Ma, K2) + Ko * log(sqrt(pow(Ra, K2) + Ko)) +
    Ko * sqrt(pow(Ra, K2) + Ko) + ifl(Vs, K800, Ko, K0);
//2.118 Sarma & Free 1995; 2.124 Free 1996, Free et al. 1998
E -> Ko + Ko * Ma + Ko * log(sqrt(pow(Ra, K2) + Ko)) + Ko * sqrt(pow(Ra, K2
    ) + Ko) + ifl(Vs, K800, Ko, K0);
//2.118 Sarma & Free 1995
E -> Ko + Ko * Ma + Ko * log(sqrt(pow(Ra, K2) + Ko)) + ifl(Vs, K180, Ko,
    ifl(Vs, K360, Ko, ifl(Vs, K800, Ko, K0)));
//2.119 Ambraseys et al. 1996, Simpson 1996; 2.120 Ambraseys & Simpson
    1996, Simpson 1996; 2.179 Schwarz et al. 2002
E -> ifl(Vs, K800, S1, S1);
//2.146 Sarma & Srbulov 1998
E -> Ko + Ko * Ma - log(Ra) / log(K10) + Ko * Ra;
//2.148 Smit 1998
E -> Po + Po * log(Ma) - Po * log(Ra);
//2.152 Ólafsson & Sigbjörnsson 1999
E -> Ko + Ko * Ma + Ko * Ra + ifl(Vs, K180, Ko, ifl(Vs, K360, Ko, ifl(Vs,
    K800, Ko, Ko)));
//2.157 Ambraseys & Douglas 2000, 2003, Douglas 2001b
E -> Ko + Ko * (Ma - K6) + Ko * pow(Ma - K6, K2) + Ko * log(sqrt(pow(Ra, K2
    ) + Ko)) + Ko * log(ifl(Vs, K360, K200, ifl(Vs, K800, K400, K700)) / Ko)
    ;
//2.175 Gülkan & Kalkan 2002; 2.198 Kalkan & Gülkan 2004b, 2005
E -> Ko + Ko * Ma + Ko * log(sqrt(pow(Ra, K2) + Ko)) + ifl(Vs, K360, Ko,
    ifl(Vs, K800, Ko, K0));
//2.181 Tromans & Bommer 2002
E -> Ko + Ko * Ma + Ko * log(sqrt(pow(Ra, K2) + Ko)) + ifl(Vs, K180, Ko,
    ifl(Vs, K360, Ko, ifl(Vs, K750, Ko, K0))) + ife(Fa, K1, Ko, ife(Fa, K0,
    Ko, K0));
//2.187 Bommer et al 2003
E -> Ko + Ko * Ma - Po * log(Ra);
//2.189 Halldórsson & Sveinsson 2003
```

```
E -> Ko + Ko * Ma - Po * log(Ra) - Po * Ra;
//2.189 Halldórsson & Sveinsson 2003
E -> Ko + Ko * Ma + Ko * log(sqrt(pow(Ra, K2) + Ko)) + ifl(Vs, K360, K0,
    ifl(Vs, K800, Ko, Ko)) + ife(Fa, K0, K0, ife(Fa, K1, Ko, Ko));
//2.192 Skarlatoudis et al. 2003
E -> Ko + Ko * Ma + Ko * pow(Ma, K2) + Ko * log(sqrt(pow(Ra, K2) + Ko)) +
    ifl(Vs, K360, K0, ifl(Vs, K800, Ko, Ko)) + ife(Fa, K0, K0, ife(Fa, K1,
    Ko, Ko));
//2.192 Skarlatoudis et al. 2003
E -> Ko + Ko * Ma + Ko * log(Ra + Ko) + ifl(Vs, K360, K0, ifl(Vs, K800, Ko,
     Ko)) + ife(Fa, K0, K0, ife(Fa, K1, Ko, Ko));
//2.192 Skarlatoudis et al. 2003
E -> Ko + (Ko + Ko * Ma) * Ma + (Ko + Ko * Ma) * log(sqrt(pow(Ra, K2) + Ko)
    );
//2.195 Bragato 2004; 2.263 Slejko et al. 2008
E -> Ko + Ko * Ma + Ko * log(sqrt(pow(Ra, K2) + Ko));
//2.195 Bragato 2004; 2.205 Skarlatoudis et al. 2004; 2.275 Pétursson &
    Vogfjörd 2009
E -> Ko + Ko * (Ma - K6) + Ko * pow(Ma - K6, K2) + Ko * pow(Ma, K3) + Ko *
    log(sqrt(pow(Ra, K2) + Ko)) + Ko * log(ifl(Vs, K360, K200, ifl(Vs, K800,
     K400, K700)) / Ko);
//2.197 Gülkan & Kalkan 2004a
E -> Ko + Ko * (Ma - K6) + Ko * pow(Ma - K6, K2) + Ko * log(sqrt(pow(Ra, K2
    ) + Ko)) + ifl(Vs, K180, Ko, ifl(Vs, K360, Ko, ifl(Vs, K750, Ko, K0)));
//2.202 Özbey et al. 2004
E -> Ko + Ko * Ma + (Ko + Ko * Ma) * log(sqrt(pow(Ra, K2) + Ko)) + ifl(Vs,
    K180, Ko, ifl(Vs, K360, Ko, ifl(Vs, K800, Ko, K0))) + ife(Fa, K1, Ko,
    ife(Fa, K0, Ko, ife(Fa, K05, Ko, K0)));
//2.207 Ambraseys et al. 2005a; 2.208 Ambraseys et al. 2005b
E -> Ko + Ko * Ma + Ko * pow(Ma, K2) + (Ko + Ko * Ma + Ko * pow(Ma, K2)) *
    log(sqrt(pow(Ra, K2) + Ko)) + ifl(Vs, K180, Ko, ifl(Vs, K360, Ko, ifl(Vs
    , K800, Ko, K0))) + ife(Fa, K1, Ko, ife(Fa, K0, Ko, ife(Fa, K05, Ko, K0)
    ));
//2.207 Ambraseys et al. 2005a; 2.208 Ambraseys et al. 2005b
E -> Ko + Ko * Ma + Ko * Ra;
//2.209 Bragato 2005
E -> Ko + Ko * Ma + Ko * Ra + Ko * Ma * Ra + Ko * pow(Ma, K2) + Ko * pow(Ra
    , K2);
//2.209 Bragato 2005
E -> Ko + (Ko + Ko * Ma) * Ma + (Ko + Ko * pow(Ma, K3)) * log(sqrt(pow(Ra,
    K2) + Ko));
//2.210 Bragato & Slejko 2005
E -> Ko + Ko * Ma + Ko * pow(Ma, K2) + (Ko + Ko * Ma) * log(sqrt(pow(Ra, K2
    ) + Ko)) + ifl(Vs, K360, Ko, ifl(Vs, K800, Ko, K0)) + ife(Fa, K1, Ko,
    ife(Fa, K0, Ko, K0));
//2.235 Akkar & Bommer 2007b; 2.239 Bommer et al. 2007; 2.277 Akkar &
    Bommer 2010
E -> Ko + Ko * Ma - Po * log(sqrt(pow(Ra, K2) + Ko)) + ifl(Vs, K360, Ko,
    ifl(Vs, K800, Ko, K0)) + ife(Fa, K0, K0, Ko);
```

```
//2.242 Danciu & Tselentis 2007a, 2007b
E -> Ko + Ko * Ma + Ko * log(Ra) + ifl(Vs, K180, Ko, ifl(Vs, K360, Ko, ifl(
    Vs, K800, Ko, K0))) + ife(Fa, K1, Ko, ife(Fa, K0, Ko, ife(Fa, K05, Ko,
    K0)));
//2.254 Cauzzi & Faccioli 2008, Cauzzi 2008, Cauzzi et al. 2008
E -> Ko + Ko * Ma + Ko * log(Ra) + Ko * log(Vs / Ko) + ife(Fa, K1, Ko, ife(
    Fa, K0, Ko, ife(Fa, K05, Ko, K0)));
//2.254 Cauzzi & Faccioli 2008, Cauzzi 2008, Cauzzi et al. 2008
E -> Ko + Ko * Ma + Ko * pow(Ma, K2) + Ko * log(Ra) + ifl(Vs, K180, Ko, ifl
    (Vs, K360, Ko, ifl(Vs, K800, Ko, K0))) + ife(Fa, K1, Ko, ife(Fa, K0, Ko,
     ife(Fa, K05, Ko, K0)));
//2.254 Cauzzi & Faccioli 2008, Cauzzi 2008, Cauzzi et al. 2008
E -> Ko + Ko * Ma + Ko * pow(Ma, K2) - log(Ra + Ko * exp(Ko * Ma)) / log(
    K10) + Ko * Ra + ifl(Vs, K180, Ko, ifl(Vs, K360, Ko, ifl(Vs, K800, Ko,
    K0)));
//2.256 Cotton et al. 2008
E -> Ko + Ko * Ma + (Ko + Ko * Ma) * log(sqrt(pow(Ra, K2) + Ko)) + ifl(Vs,
    K360, Ko, ifl(Vs, K800, Ko, K0));
//2.260 Massa et al. 2008
E -> Ko + Ko * Ma + Ko * pow(Ma, K2) + Ko * log(sqrt(pow(Ra, K2) + Ko)) +
    ifl(Vs, K360, Ko, ifl(Vs, K800, Ko, K0));
//2.260 Massa et al. 2008
E -> Ko + Ko * (Ma - K6) + Ko * pow(Ma - K6, K2) + Ko * log(Ra) + ifl(Vs,
    K180, Ko, ifl(Vs, K360, Ko, ifl(Vs, K800, Ko, K0)));
//2.266 Akyol & Koaragöz 2009
E -> Ko + Ko * Ma + Ko * pow(Ma, K2) + Ko * log(Ra + Ko * exp(Ko * Ma + Ko
    * pow(Ma, K2)));
//2.275 Pétursson & Vogfjörd 2009
E -> Ko + Ko * Ma + Ko * pow(Ma, K2) + Ko * log(Ra);
//2.275 Pétursson & Vogfjörd 2009
E -> Ko + Ko * Ma - log(Ra + Ko * exp(Ko * Ma)) / log(K10) - Ko * Ra;
//2.275 Pétursson & Vogfjörd 2009
E -> Ko + Ko * Ma + Ko * pow(Ma, K2) + Ko * log(Ra + Ko * exp(Ko * Ma)) +
    ifl(Vs, K180, Ko, ifl(Vs, K360, Ko, ifl(Vs, K800, Ko, K0))) + ife(Fa, K1
    , Ko, ife(Fa, K0, Ko, ife(Fa, K05, Ko, K0)));
//2.283 Faccioli et al. 2010
E -> Ko + Ko * Ma + Ko * pow(Ma, K2) + Ko * log(Ra + Ko * exp(Ko * Ma)) +
    Ko * log(Vs / Ko) + ife(Fa, K1, Ko, ife(Fa, K0, Ko, ife(Fa, K05, Ko, K0)
    ));
//2.283 Faccioli et al. 2010
S1 -> Ko + Ko * Ma + Ko * log(Ra) + Ko * Ra;
Ma -> variable_M;
Ra -> variable_R;
Vs -> variable_Vs;
Fa -> variable_F;
Ko -> const[_:-4000:0.1:4000];
Po -> const[_:0:0.1:4000];
K0 -> const[_:0:0:0];
K05 -> const[_:0.5:0.5:0.5];
```

```
K083 -> const[_:0.83:0.83:0.83];
K1 -> const[_:1:1:1];
K2 -> const[_:2:2:2];
K3 -> const[_:3:3:3];
K6 -> const[_:6:6:6];
K10 -> const[_:10:10:10];
K25 -> const[_:25:25:25];
K100 -> const[_:100:100:100];
K180 -> const[_:180:180:180];
K200 -> const[_:200:200:200];
K360 -> const[_:360:360:360];
K400 -> const[_:400:400:400];
K700 -> const[_:700:700:700];
K750 -> const[_:750:750:750];
K800 -> const[_:800:800:800];
```

Literatura

[1] Douglas, J. 2011. Ground-motion prediction equations 1964-2010. Final report, BRGM/RP-59356-FR and PEER/2011/102. Berkeley, University of California, College of Engineering: 444 str.

Ta stran je namenoma prazna.

Priloga B Združena gramatika Z

Tukaj prilagamo Združeno gramatiko Z, kakršna je bila uporabljena v poskusih. Več o njeni izpeljavi je zapisano v poglavju 3.2.3.

```
%{
#include <math.h>
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 + FVs + FF;
FM -> (FM + Ko * FM1);
FM -> Ko * FM1;
FM1 -> Ma;
FM1 -> pow(Ma, K2);
FM1 -> (Ma + Ko);
FM1 -> pow(Ma + Ko, K2);
FM1 -> exp(Ko * Ma);
FR -> FM * FR1;
FR -> (FR + Ko * FR1);
FR -> Ko * FR1;
FR1 -> log(Ra);
FR1 -> log(Ra + Ko);
FR1 -> log(Ra + FM);
FR1 -> log(pow(Ra, K2) + Ko);
FR1 -> log(pow(Ra, K2) + FM);
FR1 -> Ra;
FR1 -> pow(Ra, K2);
FVs -> FM1 * FVs1;
FVs -> FVs1;
FVs1 -> ifl(Vs, K800, Ko, K0);
FVs1 -> ifl(Vs, K360, Ko, ifl(Vs, K800, Ko, K0));
FVs1 -> ifl(Vs, K180, Ko, ifl(Vs, K360, Ko, ifl(Vs, K800, Ko, K0)));
FVs1 -> Ko * log(Vs / const[_:0:800:3000]);
FVs1 -> ifl(Vs, const[_:0:800:3000], Ko, K0);
FF -> ife(Fa, K1, Ko, ife(Fa, K0, Ko, K0));
FF -> 0;
Ma -> variable_M;
Ra -> variable_R;
Vs -> variable_Vs;
Fa -> variable_F;
Ko -> const[_:-100:0.1:100];
K0 -> const[_:0:0:0];
K05 -> const[_:0.5:0.5:0.5];
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];
```

Priloga C Markič, Š., Stankovski, V. An Equation-Discovery Approach to Earthquake-Ground-Motion Prediction

Diplomskemu delu prilagamo znanstveni članek z naslovom *An Equation-Discovery Approach to Earthquake-Ground-Motion Prediction*, objavljenem v letošnjem letu v reviji *Engineering Applications of Artificial Intelligence* 26, 4 na straneh 1339–1347.
doi:10.1016/j.engappai.2012.12.005

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, Lagramge, 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 Lagramge 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 [21], which is based on the data sets of [10] in the project Next Generation Attenuation of Ground Motion and the study of [2]. 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, Lagramge

C.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 ([12]), 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 [13]). 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 (C.1) from the study of [2] is presented here for illustrative purposes and can be described as a typical example of a GMPE.

$$\begin{aligned}\log_{10}(PGA) = \; & 1.04159 + 0.91333 \cdot M_{\mathrm{w}} - 0.08140 \cdot M_{\mathrm{w}}^2 \\ & + (-2.92728 + 0.28120 \cdot M_{\mathrm{w}}) \cdot \log_{10} \sqrt{R_{\mathrm{jb}}^2 + 7.86638^2} \\ & + \begin{cases} 0.08753 & \text{if} \quad V_{\mathrm{s},30} < 360\frac{m}{s} \\ 0.01527 & \text{if} \quad 360\frac{m}{s} \leq V_{\mathrm{s},30} < 800\frac{m}{s} \\ 0 & \text{if} \quad 800\frac{m}{s} \leq V_{\mathrm{s},30} \end{cases} \\ & + \begin{cases} -0.04189 & \text{if} \quad \mathsf{F = normal} \\ 0 & \text{if} \quad \mathsf{F = strike\text{-}slip} \\ 0.08015 & \text{if} \quad \mathsf{F = reverse} \end{cases} \end{aligned} \tag{C.1}$$

The variables used in equation (C.1) are:

- the PGA in $[cm/s^2]$;
- the moment magnitude M_{w};
- the Joyner-Boore distance R_{jb} in $[km]$;
- the average soil shear-wave velocity in the upper 30 meters of soil underneath the observation spot $V_{\mathrm{s},30}$ in $[m/s]$; and
- the faulting mechanism F ([2]).

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. [21] 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. [19] 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. [15] 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, Lagramge, 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 ([25]). 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 Lagramge 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 C.2 the Lagramge 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 Lagramge ED system to the problem of predicting the PGA in Section C.3. Descriptive tables and figures showing the results and the best equation found, together with their explanations, are presented in Section C.4. We conclude this study with Section C.5, where we discuss and evaluate the presented results and provide some ideas for future research.

C.2 Lagramge

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 [20], Lagrange of [24] and Lagramge of [25][1] have received particular attention in the machine-learning community. The Lagramge 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 Lagramge ED system has already been applied to several scientific fields of interest. The first experiments with the Lagramge system were made in the area of ecological modeling, e.g., the prediction of phytoplankton growth in the studies of [27] and [18]. [26] 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 Lagramge system include discovering mathematical models of a mechanically ventilated lung by [16] and the financial forecasting of commodity prices from the London Metal Exchange by [5].

[1] The Lagramge 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

The problem given to the Lagramge system is denoted with two input files: a data set D and a CFG ([25]). The input data $D = \{M, v_d, W\}$ consists of one or more tables of measurements or records M of variables $W = \{v_1, v_2, \ldots, v_n\}$. Among the variables, one must be selected as a dependent variable $v_d \in W$. 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 (C.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 Lagramge system uses a special non-terminal symbol $V \in N$, 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 set T consists of all the independent variables $v_i \in W \backslash v_d$ and a special symbol $const$, whose syntax in the Lagramge system is as follows:

$$const[name : lowest\ value : starting\ value : highest\ value] \tag{C.2}$$

In the case of our example, the set of non-terminals is $N = \{Linear, Term, V\}$ and the set of terminals is $T = \{M_{\mathrm{w}}, const[\ldots]\}$.

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 *variable_* in front of its name. However, the productions for V are added to the grammar automatically during the run-time, as the Lagramge system reads the variables' names from the input data file, i.e., $\forall v_i \in W \backslash v_d : V \rightarrow variable\_v_i \in P$. We use the annotation with the logical or operator $A \rightarrow \alpha_1 | \alpha_2 | \ldots | \alpha_n$ for productions $A \rightarrow \alpha_1, A \rightarrow \alpha_2, \ldots, A \rightarrow \alpha_n$. In order to derive the first three terms of equation (C.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 $Linear$.

The definition of the developed example polynomial grammar is provided in Table C.1. Its first four productions provide enough syntax to build the desired polynomial. The first two productions succeeding the non-terminal symbol $Linear$ generate any number of terms. The second two productions succeeding the non-terminal symbol $Term$ derive these terms into degrees. Note that the last production V $\rightarrow$ variable_M is automatically added by the Lagramge system during the run-time and must not be manually included in the grammar, but it is added to Table C.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 C.1 for our example. When we include at least one recursive production in the CFG (e.g., Table C.1, the first production), the hypothesis space

Table C.1: Polynomial context-free grammar

Linear → Linear + Term
Linear → Term
Term → Term · V
Term → const[_:-400:0.1:400]
V → variable_M

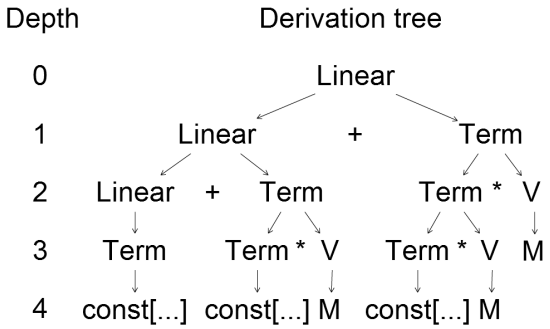


Figure C.1: Derivation tree for the first three terms of equation (C.1) with marked depth

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 Lagramge 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 C.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 C.1 is:

$$const[\dots] + const[\dots] \cdot M_{\mathrm{w}} + const[\dots] \cdot M_{\mathrm{w}} \cdot M_{\mathrm{w}} \tag{C.3}$$

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

$$MSE = \frac{1}{n} \sum_{i=1}^{n} (v_{d,i,measured} - v_{d,i,predicted})^2 \tag{C.4}$$

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

The Lagramge 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 Lagramge 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:

- when all of the possible equation structures have been derived and tested;
- when the Lagramge system finds the first expression with a lower MSE function than the one prescribed by the user; or
- when a user-defined CPU time is exceeded ([25]).

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

C.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:

- the selection of the data set;
- the definition of the CFG; and
- 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.

C.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 source, the travel path and 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 [2]). 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 [12] 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 ([10]). 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 [1], [19] 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 [1].

Throughout the literature there are a lot of combinations of variables used to determine GMPEs. As in [21] and [2], the independent variables we use in this study are:

- the moment magnitude M_{w}, which as stated by [11] is the simplest measure for correlating the amount of energy released in an earthquake;
- the source-to-site Joyner-Boore distance R_{jb} in $[km]$;
- the style-of-faulting F; and
- the average soil shear-wave velocity in the upper 30 meters of soil underneath the observation spot $V_{\mathrm{s},30}$ 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 [14]), which we incorporated into our study by stating the ED problem as follows:

$$\ln(PGA) = f(M_{\mathrm{w}}, R_{\mathrm{jb}}, V_{\mathrm{s},30}, F) \tag{C.5}$$

[21] 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 the common NGA database, except for the last, which roots from Europe: [1], [7], [8], [11], [17] and [2]. 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

Table C.2: Average, minimum and maximum values of all the variables used in this study

statistic	$PGA\ [m/s^2]$	M_{w}	$R_{\mathrm{jb}}\ [km]$	$V_{\mathrm{s,30}}\ [m/s]$	F
minimum	0.0012	4.27	0	116.4	0
average	0.0939	6.25	57.1	420.5	0.74
maximum	1.6615	7.90	365.1	2016.1	1

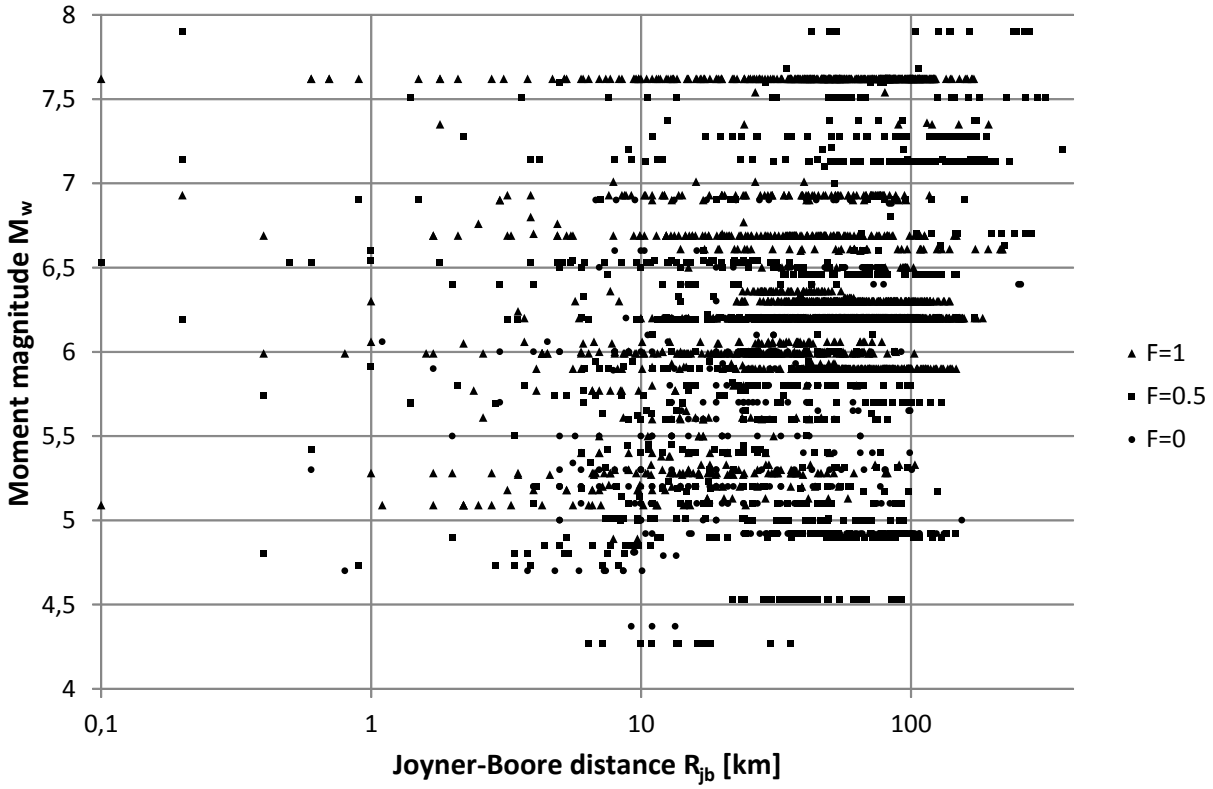


Figure C.2: Data distribution with respect to M_{w}, R_{jb} and F

- $F = 0$ for normal faults;
- $F = 0.5$ for strike-slip faults; and
- $F = 1$ for reverse faults ([21]).

The descriptive statistics of the PF-L data set are presented in Table C.2. For illustrative purposes Figure C.2 presents the moment magnitude M_{w} vs. the Joyner-Boore distance R_{jb} and the faulting mechanism F data distribution. From Table C.2 and Figure C.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 (C.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 Lagramge 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.

C.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 [13]. 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 [8]. 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 [9], where the authors experimented by including an anelastic decay term, a quadratic magnitude dependence and 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 (C.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 C.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 [13].

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 Lagramge system first leads from the root symbol Eq to a number of non-terminal functions (see Table C.3). These functions are named FM, FR, FV and FF after the dependence they model, i.e., $f(M_{\mathrm{w}})$, $f(R_{\mathrm{jb}})$, $f(V_{\mathrm{s},30})$ 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 [8].

In their study, however, the $f_{dis} = f(M, R_{RUP})$, 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 $f(M_{\mathrm{w}}, R_{\mathrm{jb}})$. 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 $f(V_{\mathrm{s},30})$ 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 Lagramge system makes it possible to build productions with all the operators or functions defined in the C programming language ([25]). In order to employ if-structures in the grammar, the functions *ifl* 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 C.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 Lagramge algorithm. The remaining 7 productions lead to constant parameters, and all except for the parameter Ko are set to a single value. Note that numbers cannot be used in the grammar explicitly. The parameter(s) $const$ 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.

C.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 C.2. With increasing d, the hypothesis space for the Katja grammar increases by approximately a factor of 75, as can be seen in Table C.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 [13] using the Katja grammar. The other columns in Table C.4 are the maximum and minimum length of an expression, the maximum number of terminal symbols $const$ 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.

Table C.3: The Katja context-free grammar

```
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]
```

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 C.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 Lagramge 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 $b \in \{1, 2, 5, 10, 20, 50\}$.

Table C.4: A descriptive overview of the exploration space for a tree depth of up to 8 for the Katja grammar

d	$no.equ.$	$max.len.$	$min.len.$	$max.no.const.$	$deriv.len.$
0	0	-1	-1	-1	-1
1	0	-1	-1	-1	0
2	0	-1	-1	-1	2
3	0	-1	-1	-1	25
4	960	84	18	21	39
5	100800	120	26	29	55
6	$\approx 7.4 \cdot 10^6$	156	43	37	71
7	$\approx 5.2 \cdot 10^8$	192	60	45	87
8	$\approx 3.7 \cdot 10^{10}$	228	77	53	103

Parameter Fitting Restarts *m*. Each time an equation is generated it contains many terminals $const$ (see Table C.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.

C.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 mOSAIC Cloud project. This resulted in systems that are used for the development and management of distributed applications as in [22] and [23]. 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 Lagramge 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 C.3. While experimenting, we pushed the required memory for running the Lagramge 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 Lagramge system.

C.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.

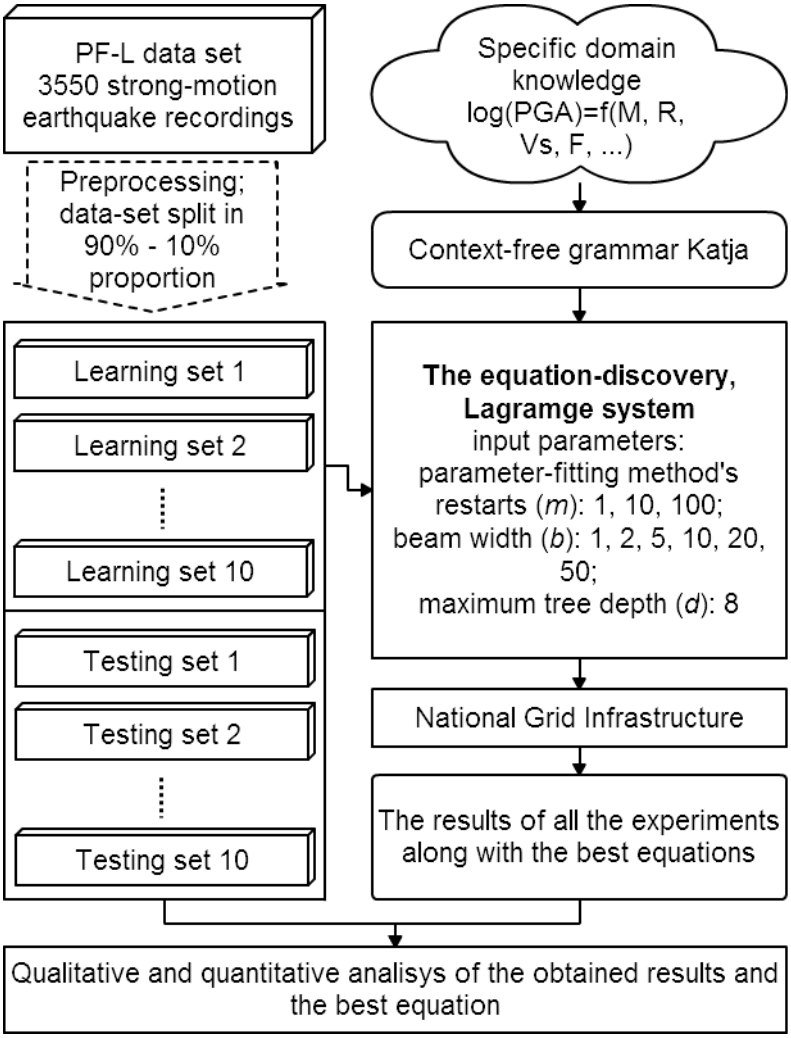


Figure C.3: A diagram of the experimental set-up

C.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 (C.6). It was found when running the Lagramge system with the heuristic beam search, $d = 8$, $b = 100$, $m = 50$ for the first learning 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.

$$\begin{aligned}\ln(PGA) = \; & 4,57353 - 1,69293 \cdot M_{\mathrm{w}} + 0,2417 \cdot M_{\mathrm{w}}^2 \\ & -6,67613 \cdot e^{-7,60198 \cdot M_{\mathrm{w}}} - 0,00918368 \cdot \frac{e^{1,3707 \cdot M_{\mathrm{w}}}}{R_{\mathrm{jb}}+100} \\ & -1,67822 \cdot ln(R_{\mathrm{jb}} + 12,7587) - 0,291666 \cdot ln\frac{V_{\mathrm{s},30}}{4000} \\ & + \begin{cases} 0.1254 & \text{if F = 0} \quad \text{(normal)} \\ 0 & \text{if F = 0.5} \quad \text{(strike-slip)} \\ 0.1188 & \text{if F = 1} \quad \text{(reverse)} \end{cases}\end{aligned} \tag{C.6}$$

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 C.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 C.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.

Table C.5: Average MSE on the testing data set for each 10-fold cross-validation split for the Katja grammar

b	$m = 1$	$m = 10$	$m = 100$
1	0.597	0.596	0.595
2	0.595	0.594	0.593
5	0.593	0.593	0.590
10	0.548	0.535	0.522
20	0.516	0.518	0.513
50	0.510	0.510	0.509

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

b	$m = 1$	$m = 10$	$m = 100$
1	0.041	0.041	0.040
2	0.042	0.042	0.040
5	0.042	0.042	0.040
10	0.041	0.025	0.032
20	0.030	0.031	0.029
50	0.029	0.030	0.029

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 [3]:

- correlation coefficient R

$$R = \frac{\sum_{i=1}^{n} (m_i - \overline{m}) * (p_i - \overline{p})}{\sqrt{\sum_{i=1}^{n} (m_i - \overline{m})^2 \sum_{i=1}^{n} (p_i - \overline{p})^2}} \tag{C.7}$$

- root mean-squared error RMSE

Table C.7: Calculated averages and standard deviations of the R, RMSE and MAE criteria on the testing data sets: *i*) of equation (C.6) developed by the Lagramge system; *ii*) of equation (C.1) developed by [2]; and *iii*) of the CAE method proposed by [21]

Criteria		Lagramge	[2]	[21]
R	$\overline{R}$	0.827	0.802	0.841
	σ_R	0.014	0.015	0.013
RMSE	$\overline{RMSE}$	0.629	0.678	0.609
	σ_{RMSE}	0.019	0.019	0.020
MAE	$\overline{MAE}$	0.491	0.528	0.474
	σ_{MAE}	0.017	0.014	0.018

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

- mean absolute error MAE

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|m_i - p_i| \tag{C.9}$$

where m_i and p_i are the measured and predicted values of the PGA of the i^{th} record with an average of $\overline{m}$ and $\overline{p}$, 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 (C.8), the RMSE is the square root of the MSE, defined with equation (C.4). The average and standard deviation on all 10 testing data sets of these criteria for three models are shown in Table C.7:

- in the first column for equation (C.6), discovered by Lagramge;
- in the second column for equation (C.1) proposed by [2]; and
- in the third column for the CAE method of [21].

According to the calculated values, this method performs better than the method of [2], but worse than the CAE method of [21]. Note, however, that the CAE method does not provide a formula that could be used in the engineers' daily work.

Figure C.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.

C.4.2 Qualitative Analysis

When we observe the very structure of the found equation (C.6), we can see that two constants reached their maximum value, one in the denominator with R_{jb} at value 100 and the other in

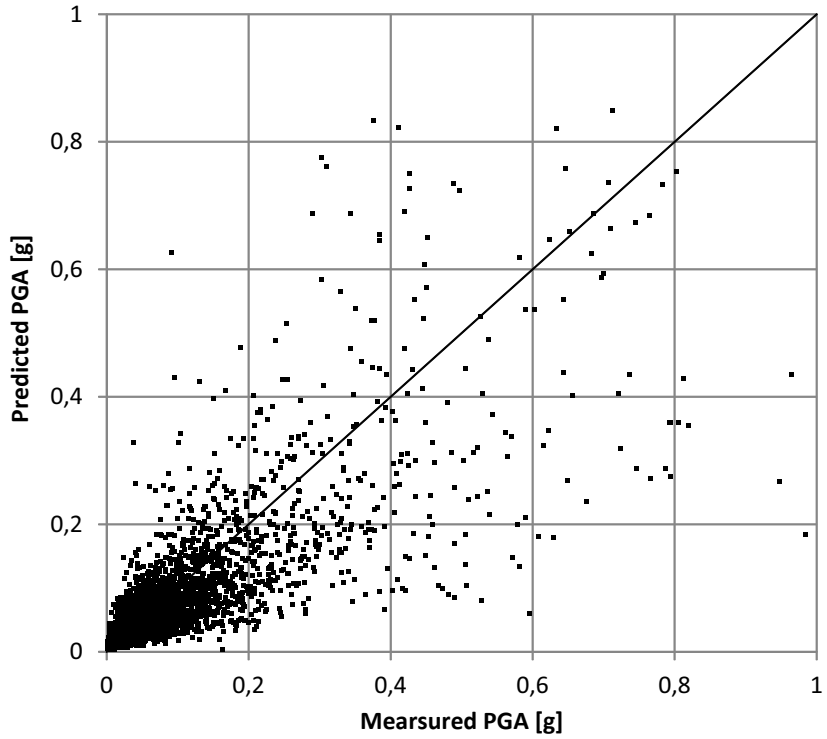


Figure C.4: Predicted vs. measured PGA of equation (C.6) for the whole data set

the $V_{\mathrm{s},30}$ term at value 4000. In the equation there are three M_{w}, one R_{jb}, one M_{w}-R_{jb}, one $V_{\mathrm{s},30}$ and one F dependencies. The Lagramge system found that the $V_{\mathrm{s},30}$ and F terms are not negligible, even though the algorithm could choose such productions. The $V_{\mathrm{s},30}$ term was found as a continuous dependency, rather than divided into classes.

Equation (C.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 [1], [7], [8], [11] and [17] and with the formula designed by [2]. Equation (C.6) is labeled Lagramge for convenience and its graph is aligned with the graphs corresponding to the formulae of these authors in Figures C.5 and C.6, drawn for magnitudes of 6 and 7, respectively. It is clear that equation (C.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 (C.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 [6]). The majority of data lies between the values of $4.9 \leq M_{\mathrm{w}} \leq 7.6$ and $0\ km \leq R_{\mathrm{jb}} \leq 200\ km$ (see Figure C.2), therefore, according to the data, the newly induced equations could be applied within these borders. However, it is clear from Figure C.7, that this equation models physically only up to $M_{\mathrm{w}} = 7$, as the

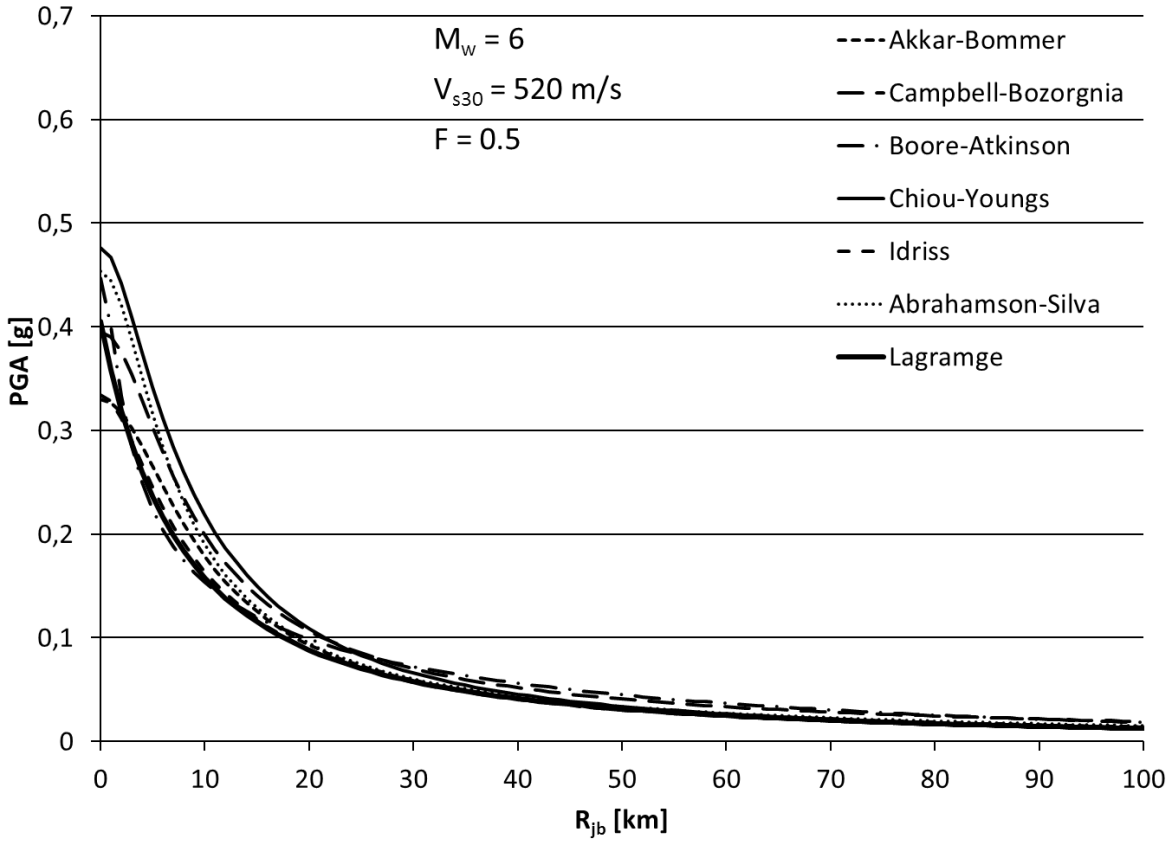


Figure C.5: Graph of PGA versus R_{jb} for equation (C.6) found by the Lagramge system aligned with the graphs of models from other authors

graph of the predicted PGA for $M_{\mathrm{w}} = 8$ 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 [19].

C.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., MEP in the study of [4]. Compared to MEP and other machine-learning techniques, our results indicate that the Lagramge 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,

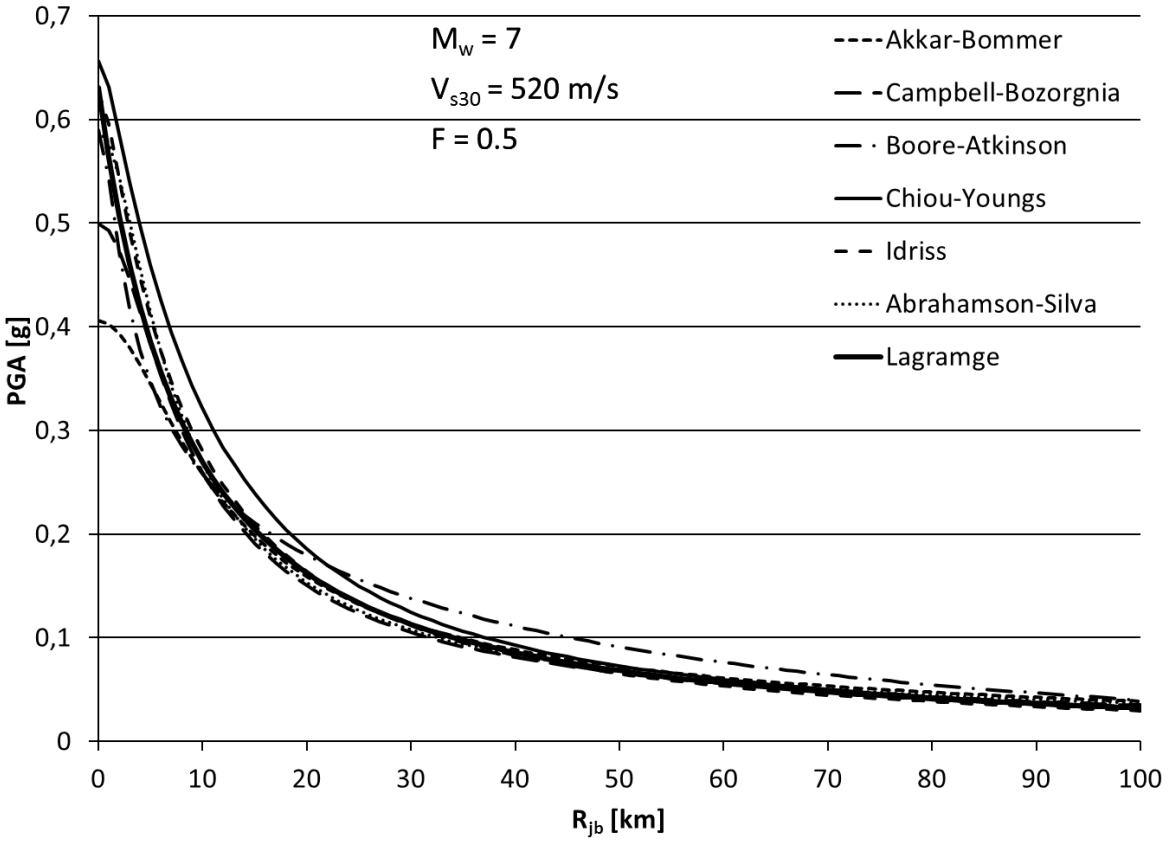


Figure C.6: Graph of PGA versus R_{jb} for equation (C.6) found by the Lagramge system aligned with the graphs of models from other authors

equally importantly, the calculated quantitative criteria for the new equation are better than those obtained for the equation of [2]. Since the CAE method of [21] 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 [23]) could be used to speed up the calculations and explore a large hypothesis space of possible equations.

The application range of equation (C.6) is according to the qualitative analysis $4.0 \leq M_{\mathrm{w}} \leq 7.0$ and $0\ km \leq R_{\mathrm{jb}} \leq 200\ km$. 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.

C.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

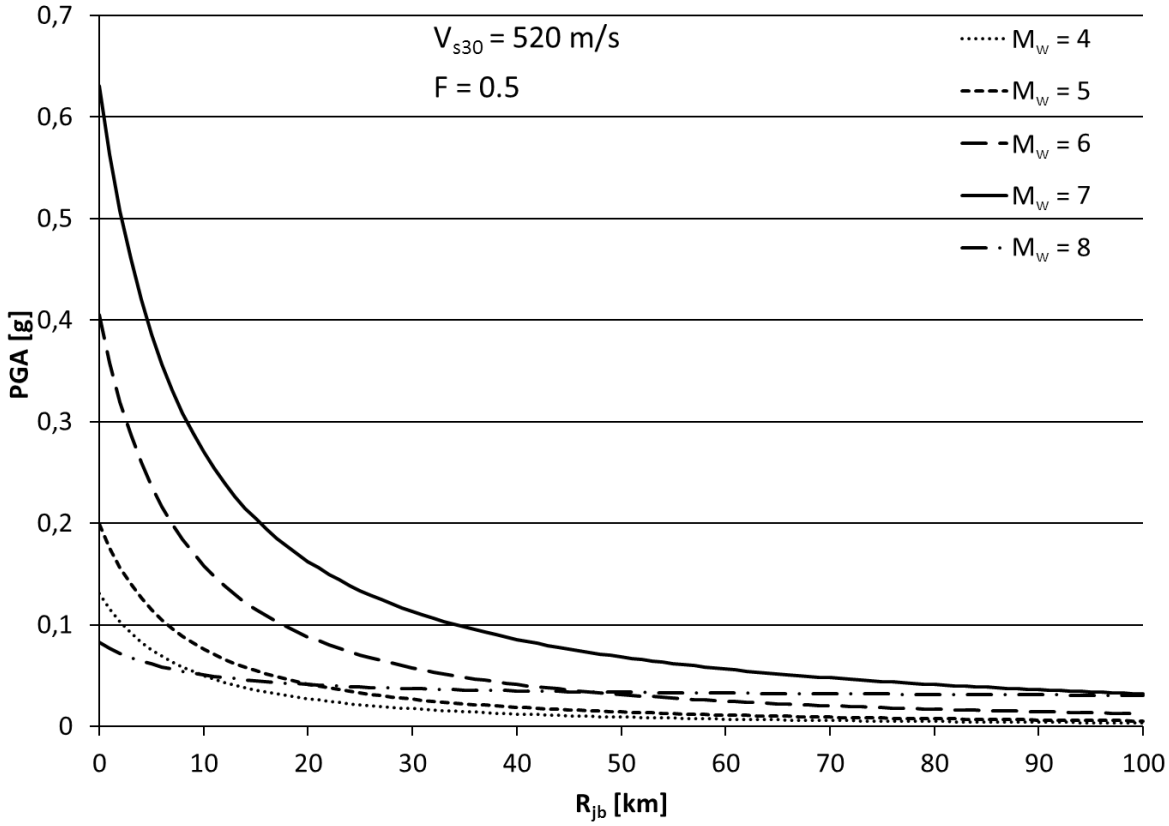


Figure C.7: Graph of PGA versus R_{jb} for equation (C.6) found by the Lagramge system for $M_{\mathrm{w}} \in \{4, 5, 6, 7, 8\}$

the developed equations reported in [13]; 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 [11]). 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 paralelization of the Lagramge 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 Lagramge 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 Lagramge 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.

References

[1] Abrahamson, N., Silva, W. 2008. Summary of the Abrahamson and Silva NGA ground-motion relations. Earthquake Spectra 24, 1: 67–97.

[2] Akkar, S.,Bommer, J.J.2010. Empirical equations for the prediction of PGA, PGV, and spectral accelerations in Europe, the Mediterranean region, and the Middle east. Seismological Research Letters 81: 195–206.

[3] Alavi, A.H., Gandomi, A.H., 2011. Prediction of principal ground-motion parameters using a hybrid method coupling artificial neural networks and simulated annealing. Compututers & Structures 89: 2176–2194.

[4] Alavi, A.H., Gandomi, A.H., Modaresnezhad, M., Mousavi, M., 2011. New ground-motion prediction equations using multi expression programing. Journal of Earthuake Engineering 15: 511–536.

[5] Alzaidi, A.,Kazakov, D.,2011. Equation discovery for financial forecasting in the context of islamic banking, in: In Proc. of the Eleventh IASTED International Conference on Artificial Intelligence and Applications AIA 2011, Morgan Kaufmann: 97–103.

[6] Bommer, J.J., Douglas, J.,Scherbaum, F. Cotton, F., Bungum, H., Fäh, D., 2010. On the selection of ground-motion prediction equations for seismic hazard analysis. Seismological Research Letters 81: 783–793.

[7] Boore, D.M., Atkinson, G.M. 2008. Ground-motion prediction equations for the average horizontal component of PGA, PGV, and 5%-damped PSA at spectral periods between 0.01 s and 10.0 s. Earthquake Spectra 24, 1: 99–138.

[8] 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 to 10 s. Earthquake Spectra 24, 1: 139–171.

[9] Cauzzi, C., Faccioli, E., 2008. Broadband (0.05 to 20 s) prediction of displacement response spectra based on worldwide digital records. Journal of Seismology 12: 453–475.

[10] Chiou, B., Darragh, R., Gregor, N., Silva, W. 2008. NGA project strong-motion database. Earthquake Spectra 24, 1: 23–44.

[11] Chiou, B.S.J., Youngs, R.R. 2008. An NGA model for the average horizontal component of peak ground motion and response spectra. Earthquake Spectra 24, 1: 173–215.

[12] Douglas, J., 2003. Earthquake ground motion estimation using strong-motion records: a review of equations for the estimation of peak ground acceleration and response spectral ordinates. Earth-Science Reviews 61: 43–104.

[13] Douglas, J., 2011. Ground-motion prediction equations 1964-2010. Final report, BRGM/RP-59356-FR and PEER/2011/102, publisher Pacific Earthquake Engineering Research Center: 444 p. 9 illustration.

[14] Douglas, J., Smit, P.M. 2001. How accurate can strong ground motion attenuation relations be? Bulletin of the Seismological Society of America 91: 1917–1923.

[15] Gandomi, A.H., Alavi, A.H., Mousavi, M., Tabatabaei, S.M. 2011. A hybrid computational approach to derive new ground-motion prediction equations. Engineering Applications of Artifficial Intelligence 24: 717–732.

[16] Ganzert, S., Möller, K., Kramer, S., Kersting, K., Guttmann, J. 2010. Identifying mathematical models of the mechanically ventilated lung using equation discovery, in: Dössel, O., Schlegel, W.C. (Eds.), World Congress on Medical Physics and Biomedical Engineering, September 7 - 12, 2009, Munich, Germany. Springer Berlin Heidelberg. 25/4 of *IFMBE Proceedings*: 1524–1527.

[17] Idriss, I.M. 2008. An NGA empirical model for estimating the horizontal spectral values generated by shallow crustal earthquakes. Earthquake Spectra 24, 1: 217–242.

[18] Kompare, B., Todorovski, L., Džeroski, S., 2001. Modelling and prediction of phytoplankton growth with equation discovery: case study-Lake Glumsø, Denmark. Verhandlungen des Internationalen Verein Limnologie 27: 3626–3631.

[19] Kuehn, N.M., Riggelsen, C., Scherbaum, F. 2011. Modeling the joint probability of earthquake, site, and ground-motion parameters using bayesian networks. Bulletin of the Seismological Society of America 101: 235–249.

[20] Langley, P. Simon, H., Bradshaw, G., 1987. Computational Models of Learning. Springer, Berlin. p. 217.

[21] Peruš, I., Fajfar, P., 2010. Ground-motion prediction by a non-parametric approach. Earthquake Engineering & Structural Dynamics 39: 1395–1416.

[22] Stankovski, V., Swain, M., Kravtsov, V., Niessen, T. Wegener, D., Kindermann, J., Dubitzky, W., 2008 a. Grid-enabling data mining applications with DataMiningGrid: An architectural perspective. Future Generation Computer Systems 24: 259–279.

[23] Stankovski, V., Swain, M., Kravtsov, V., Niessen, T., Wegener, D., Rohm, M., Trnkoczy, J., May, M., Franke, J., Schuster, A., Dubitzky, W., 2008 b. Digging deep into the data mine with DataMiningGrid. Internet Comput., IEEE12: 69–76.

[24] Todorovski, L., Džeroski, S., 1995. Discovering dynamics: From inductive logic programming to machine discovery. Journal of Intelligent Information Systems: 89–108.

[25] Todorovski, L., Džeroski, S., 1997. Declarative bias in equation discovery, in: In Proceedings of the Fourteenth International Conference on Machine Learning, Morgan Kaufmann. p.: 376–384.

[26] Todorovski, L., Džeroski, S., 2001. Using domain knowledge on population dynamics modeling for equation discovery, in:De Raedt, L., Flach, P. (Eds.), Machine Learning: ECML 2001. Springer Berlin Heidelberg. 2167 of *Lecture Notes in Computer Science*, p.: 478–490.

[27] Todorovski, L., Džeroski, S., Kompare, B., 1998. Modelling and prediction of phytoplankton growth with equation discovery. Ecological Modeling 113: 71–81.

Priloga D Markič, Š., Dirnbek, J., Stankovski, V. A Grid Application for Equation Discovery in the Earthquake Engineering Domain

Diplomskemu delu prilagamo objavo na konferenci *The Third International Conference on Parallel, Distributed, Grid and Cloud Computing for Engineering (PARENG 2013)*, ki je bila organizirana med 25. in 27. marcom 2013 v Pécsu na Madžarskem, z naslovom *A Grid Application for Equation Discovery in the Earthquake Engineering Domain*.

Abstract. Equation discovery (ED) is an emerging area of machine learning, where the main objective is to find a proper equation structure that best describes the dependencies between variables in the given data set. Recent studies have shown that ED systems can be useful in the engineering disciplines in general and earthquake engineering in particular. Since the implemented algorithms are usually computationally intensive in nature it is beneficial to use a grid infrastructure. The goal of the present study was to develop a Web application and associated scripts that would make it possible to execute the experiments with the Lagramge ED system on the grid, in our case the Slovene national grid initiative (SLING) infrastructure. With the use of such application, the Lagramge ED methods are available in a simplified form to the whole scientific community. A use case from the domain of earthquake engineering is presented, where equation models for the earthquake's peak ground acceleration are needed in order to design seismically safe structures. The presented Web application can be used also in other engineering disciplines, where equations are commonly used as models.

Keywords. Web application, equation discovery, scientific gateway, grid, Lagramge, peak ground acceleration

D.1 Introduction

In many engineering disciplines, including civil engineering, there are needs to be able to develop, deploy and use distributed computing applications. Application areas include: earthquake engineering, structural analysis, hydrological modeling, traffic management and many more.

Modern grid and cloud computing environments, such as infrastructure as a service (IaaS) and platform as a service (PaaS) providers facilitate application development as well as access to distributed computing infrastructures, however, the enabling of existing engineering applications for distributed computing is still a very complex task. This means that civil engineers still need the assistance of distributed computing experts in order to re-design, develop and deploy their applications and that the maintenance and gradual improvements of such applications are still inefficient. Due to such problems, the uptake of distributed computing infrastructures in the engineering areas is still very low.

Currently, with the main purpose to give a boost to such an uptake, we have witnessed new frameworks facilitating the seamless development of Web applications that may serve as front ends to the distributed computing infrastructures. Such environments are also called scientific gateways (with related portlet technologies), e.g. generic purpose gateway technology in the

EU FP7 project SCI-BUS. A scientific gateway can be established as a front end to the distributed computing infrastructure, and usually the technology makes it possible to connect to more than one type of infrastructure (a so-called distributed computing bridge serves as middleware) making it possible to utilise existing infrastructures.

The Academic and research network of Slovenia (ARNES) provides and maintains the national grid infrastructure under the Slovene national grid initiative (SLING). The researchers can use a total of 4268 processors and associated storage facilities to run their high-resource demanding jobs. The cluster supports NorduGrid ARC and gLite middleware solutions for running the jobs, which may take a lot of time to carefully prepare various scripts and thoroughly test them.

In the present study, we investigate the problem of building a Web application that would make it possible to run a very recent application for equation discovery in the area of earthquake engineering [1]. The application itself takes as input a data set and a context-free grammar (CFG) and provides as output an equation model of the investigated phenomenon. The goals of this study are therefore to develop a special purpose Web application (scientific gateway) for the whole scientific community to use, which will serve as a front-end for various applications in the earthquake engineering and beyond, where equation-discovery (machine-learning) systems are needed. Moreover, since the employed equation-discovery algorithm is generic and has already been used in other domains, the use of the Web application itself is not limited to the earthquake engineering domain.

Key enabling technologies, which will be integrated in this study are basic Web application development languages such as PHP and the NorduGrid ARC software which runs on the SLING. The equation-discovery application has been developed previously and is implemented in the C and FORTRAN programming languages [2].

The paper is organised as follows. In Section D.2 we explain the state of the art in the area of equation discovery. We describe the Web application architecture together with the equation-discovery system in Section D.3. An example of usage is provided from our previous study in Section D.4. We conclude this study by analysing the benefits of the presented Web application in Section D.5.

D.2 State of the art review

Equation discovery (ED) is an emerging area of machine learning, where the main objective is to find a proper equation structure that best describes the dependencies between variables in the given data set. ED relies on system identification methods, which assume a particular equation structure that can be constructed from a given set of operators, functions and variables. The equation structure usually contains various constant parameters, the values of which are determined by means of numerical methods. The ED systems are built to investigate potentially great number of equation structures by using shear computational power and based on the approach it is possible to choose the one that best fits the data set in both quantitative and qualitative terms.

Naturally, not all possible equation structures represent an appropriate model for a given phenomenon. Therefore, it is necessary to use a formalism to define a hypothesis space of all the plausible equation structures. This hypothesis space can be infinite for many complex real-world phenomena, hence it is necessary to implement heuristics in the underlying ED algorithm, so that more suitable equation structures are being tested. About a dozen ED systems have been described in the literature, among which Bacon [3], Lagrange [4] and Lagramge [2] have received particular attention in the machine-learning community. The Lagramge system uses a context-free grammar (CFG) to generate hypothetical equation structures and which is presented as input information to the system [2]. The CFG is a formalism, which makes it possible to provide domain knowledge to the ED system and so guide it toward more plausible equation formulations.

The Lagramge ED system has already been applied to several scientific fields of interest. The first experiments with the Lagramge system were made in the area of ecological modeling, e.g., the prediction of phytoplankton growth [5, 6] or predicting the behavior of prey-predator dependence [7]. Some of the latest applications of the Lagramge system include discovering mathematical models of a mechanically ventilated lung in [8] and the financial forecasting of commodity prices from the London metal exchange [9]. The Lagramge system seems to be suitable for addressing engineering problems, particularly because it uses CFG to specify prior knowledge [1]. However, to enable its broader usage, a new Web application is developed to make this software available to more researchers around the world.

D.3 Application architecture

In this section the developed application architecture which incorporates the key requirements for the needed Web application is described. The whole process of ED consists of three separate phases:

1. the preparation of the multi job, which involves:
    - the input of the data set,
    - the input of the designed CFG,
    - the selection of other settings and
    - the submission;
2. the overview of the running jobs; and
3. the acquisition of the results.

The Web application was developed in the PHP, HTML and Bash programming languages. The whole architecture of the developed Web application is presented in Figure D.1. We incorporated these findings into three tasks, which are described in the following subsections.

D.3.1 Multi job preparation

The procedure of the multi job preparation is based on the required information for the Lagramge ED system, provided in [2], and the readme file enclosed with Lagramge source files. The

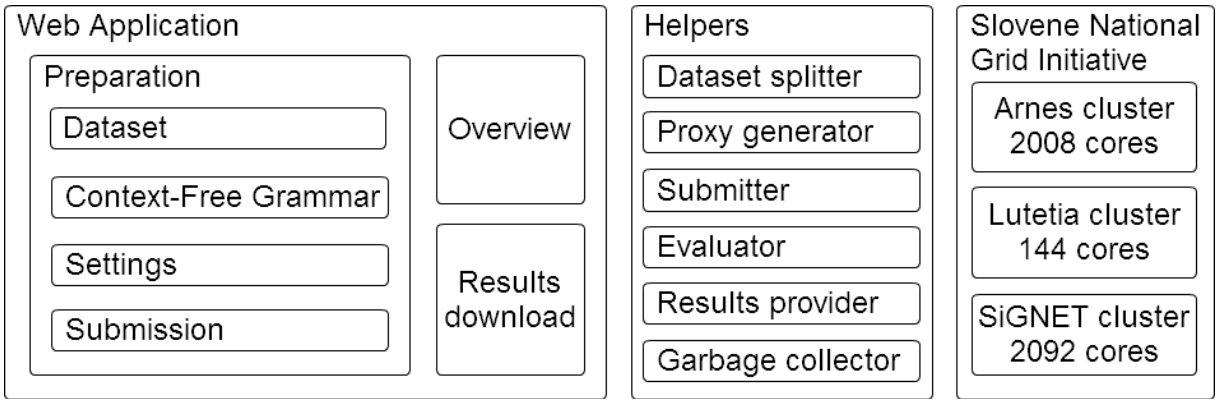


Figure D.1: The architecture of the developed Web application

problem given to the Lagramge system is denoted with two input files (a data set and a CFG), and other settings to guide the ED process (the number of best equation saved, the type of the searched equation, the number of parameter fitting methods' restarts, the search strategy, the height of the derivation tree, the evaluation criteria function and the stopping criteria).

For the purpose of the Web application, these have been separated into a step-by-step process for easier access and usage as described earlier. An explanation of both files and all the settings, together with their implementation in the developed Web application is presented in following subsubsections.

Data set. The first step in the multi job preparation is the input of a representative data set. The input data set $D = \{M, v_d, W\}$ consists of one or more tables of measurements or records M of variables $W = \{v_1, v_2, \ldots, v_n\}$. It has to be specially formatted, always starting with a line of space-separated variables' names followed by lines of tab-separated values with the decimal dot notation. Each line must end with a semicolon [2].

Among the variables, one must be selected as a dependent variable $v_d \in W$ for which the equation is induced. One of the independent variables may describe the time values of the measurements $v_t \in W \backslash v_d$, which is needed when searching differential equations. Values of both parameters can be set in the third step.

Context-free grammar. The second step is the input of the designed CFG. The tuple $CFG = \{N, T, P, S\}$ prescribes the syntax of the right-hand side of an equation. It contains finite disjunctive sets of nonterminals (N) and terminals (T). The Lagramge system uses a special non-terminal symbol $V \in N$, which denotes any of the independent variables from the input data set $W \backslash \{v_d, v_t\}$; otherwise, any symbol can be used to denote a non-terminal. The set T consists of all the independent variables $v_i \in W \backslash \{v_d, v_t\}$ and a special symbol $const$, whose syntax in the Lagramge system is as follows:

$$const[name : lowest\ value : starting\ value : highest\ value] \tag{D.1}$$

The most important part of the CFG are the productions $P = \{P_1, P_2, \ldots, P_n\}$, which 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 in the grammar to an explicit variable, we must use *variable_* in front of its name. However, the productions for V are added to the grammar automatically during the run-time, as the Lagramge system reads the variables' names from the input data file, i.e., $\forall v_i \in W \backslash \{v_d, v_t\} : V \rightarrow variable\_v_i \in P$. Finally, $S \in N$ is a special, nonterminal symbol, from which the derivation of the expressions starts.

The CFG has to be specially formatted. At the beggining we enclose everything that will be literally interpret by the grammar compiler in a pair of %{ and %}, i.e., we include the $\#include < math.h >$ sentence for the operators and simple functions definitions and provide any user-defined functions written in the programming language C. Note that functions' names should use only small characters. Following must be a list of all the defined productions, each ending with a semicolon [2]. The users can make changes in the grammar on the fly as the Web application tries to compile the given grammar and marks the place of possible error in the CFG.

Settings selection. The third step is the set up of various parameters that guide the search process. The obligatory dependent variable input parameter must be selected among the variables provided in the data set and has no default value. The user can choose whether he would like to cross validate the results. The Web application provides the option to randomly split the data set in 90 % – 10 % proportion into learning and testing sets 10 times. The number of the best equations given as the ED task's results can be set in the beam width input parameter, which has a default value of 25 (positive integer). All other parameters and their respectful options are listed here.

• *Expression derivation.* During the derivation process the Lagramge system continuously applies productions to all the nonterminals until all the symbols in the expression are terminals. When we include at least one recursive production in the CFG, the hypothesis space and the length of the derivation process are infinite. Therefore, we bound the complexity of expressions with the maximum-tree-height parameter, which has a default value of 5 (positive integer), urging the Lagramge system to ensure that all the symbols at the prescribed height are terminals.

• *Equation type.* 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. When modeling differential equations, one of the independent variables may also describe the time values of the measurements $v_t \in W \backslash v_d$ or the time interval can be set between successive lines of measurements assuming equal time spacing (positive real number).

• *Parameter fitting.* A derived expression contains one or more special terminal symbols $const \in T$. With a non-linear fitting method the Lagramge system minimizes the value of the MSE function or the MDL function, which introduces penalty for equation complexity. They are calculated according to the formulae:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(v_{d,i,measured} - v_{d,i,predicted})^2 \tag{D.2}$$

$$MDL = MSE + \frac{l}{10 \cdot l_{max}} \cdot \sigma_{vd} \tag{D.3}$$

where n is the number of records in the data set, $v_{d,i,measured}$ and $v_{d,i,predicted}$ are the measured and predicted values of the dependent variable, respectively, l is the length of the expression, l_{max} is the length of the largest expression generated with the given grammar up to the given height and σ_{vd} is the standard deviation of the $v_{d,measured}$. These methods are likely to catch in local minimums; therefore, we can determine the number of restarts with the parameter-fitting-method restarts input parameter, which has a default value of 0 (nonnegative integer).

• *Search strategy.* The Lagramge system provides two search strategies. The default option is the exhaustive search, where the algorithm derives all the possible equation structures defined with the given CFG and prescribed height and tries them against the given data set. This procedure, though, may take a lot of time, especially for a large hypothesis space. This is why a heuristic *beam search* strategy is implemented, which starts with a number of expressions (the number is equal to the beam width) and derives all their first successors, then saves the same number of those with the lowest value of the chosen criterion among all of them and repeats.

• *Stopping criteria.* Three stopping criteria are implemented into the Lagramge system:

  – when all of the possible equation structures have been derived and tested, which is the default option;
  – when the Lagramge system finds the first expression with a lower criterion value than the one prescribed by the user (positive real number); or
  – when a user-defined CPU time is exceeded (positive real number, given in minutes).

Multi job submission. At the end of the preparation process a final overview over the data set, the CFG and all input parameters is provided for the final checking as shown in Figure D.2. The user is encouraged to fix any mistakes made in previous steps and then submit.

When submitted, the Web application runs the code given in Algorithm D.1. First, it provides a reference number for the multi job and second, a configuration file for the user's archive. The data set is then randomly split to a 90 % learning set and 10 % testing set if the cross-validation option is selected. A multi job is formed and is then automatically submitted to the grid with the CFG, a learning data set and the set of options. The algorithm then waits till the execution on the grid finishes and collects the results afterwards. These are then evaluated and compressed for download.

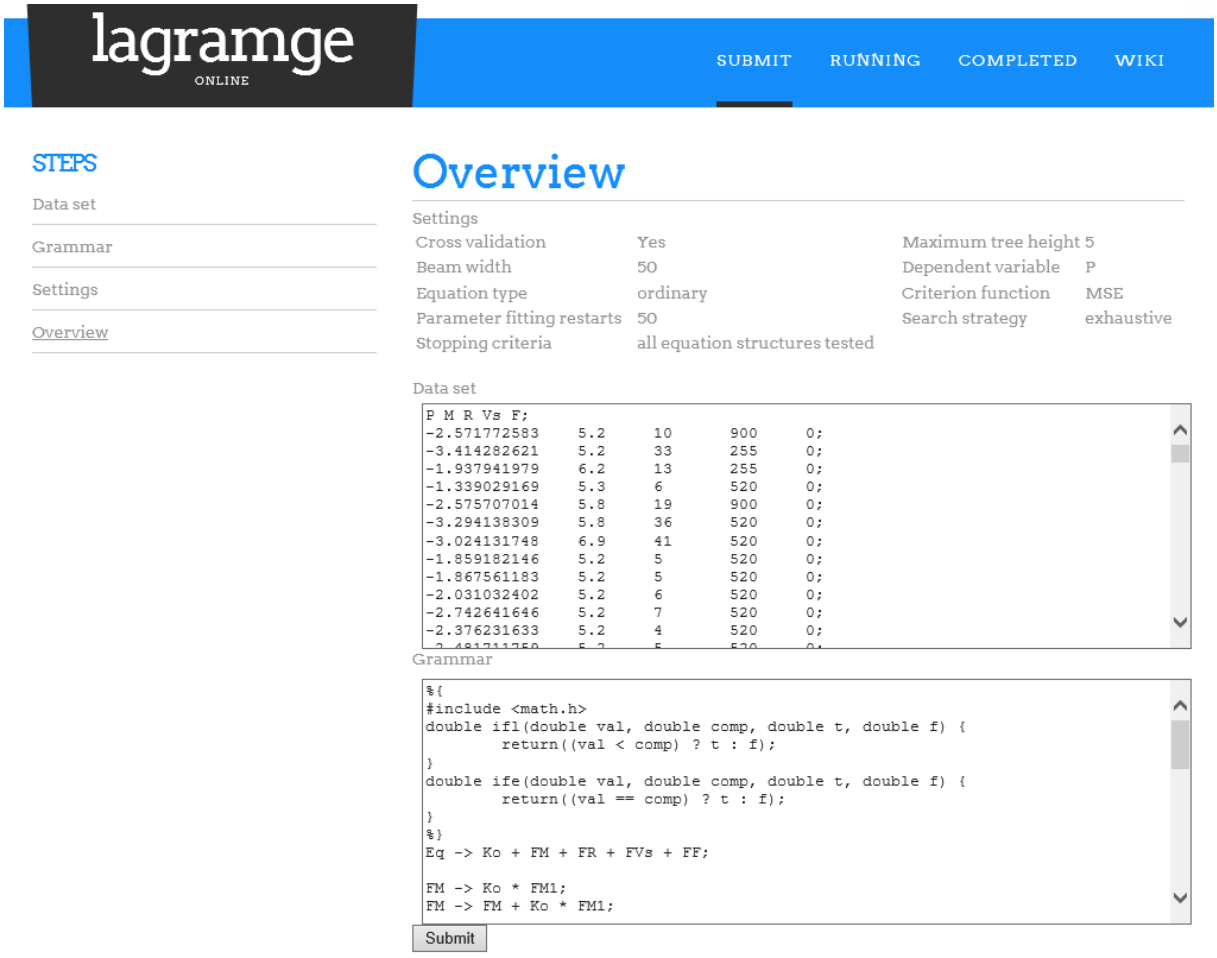


Figure D.2: The overview before the submission

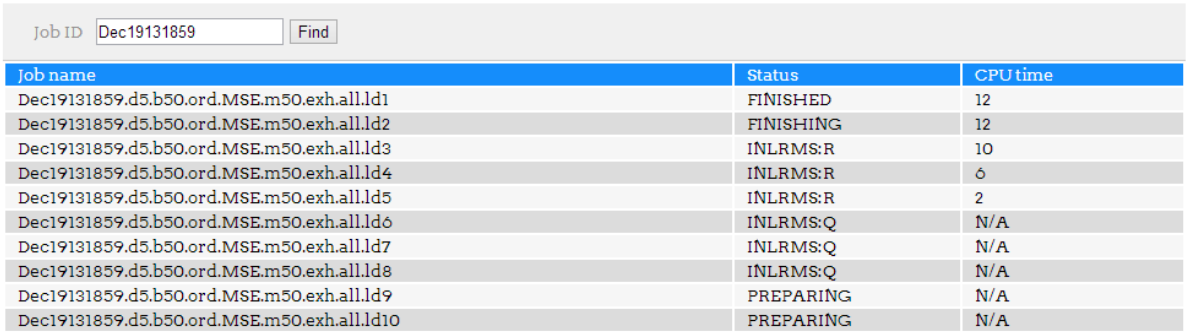


Figure D.3: A list of running jobs

D.3.2 Active jobs overview

An overview over the running jobs is necessary in order to control and monitor the progress of the ED process. This is provided via the SLING monitoring environment, where the job names together with the job status and elapsed time can be obtained. When the users provide their reference number, the jobs belonging to their experiment are printed on the page as shown in Figure D.3.

Algorithm 1 The main algorithm

Require: data set D, CFG, options
Ensure: equations
generate reference number (RN)
RN, D, CFG, options $\rightarrow$ config.file
make directories
if cross validation **then**
    for $\$i = 1..10$ **do**
        randomly select $\frac{1}{10}D \rightarrow$ test$i data
        D\test$i data $\rightarrow$ learn$i data
    end for
else
    $D \rightarrow$ learn data
end if
generate proxy
submit the multi job with CFG, learn data and options
while not finished **do**
end while
collect the results
if cross validation **then**
    for $\$i = 1..10$ **do**
        evaluate equations with test$i data
    end for
end if
equations $\rightarrow$ output file
compress files $\rightarrow$ results.zip

D.3.3 Obtaining the results

Specific scripts had to be implemented to query for the completed tasks and provide them to the end user that submitted the experiment. When the multi job is finished, the Web application collects the results from the Grid. The data collection process may take few minutes due to the large quantities of results that need to be transferred. These are then evaluated in a cross-validation process and gathered in a specially formatted file designed as tab-separated values with the columns:

- the data split number;
- the elapsed time;
- the number of tried equations;
- the value of criterion chosen on the learning data set;
- the value of the MSE criterion on the testing data set; and
- the derived equation,

as shown in Figure D.4. At the end, the user can download the compressed results.

D.4 Using the Web application

The developed Web application was used to induce a ground-motion prediction equation (GMPE) suitable for modeling the peak ground acceleration (PGA) that happens in an earthquake event.

Test	Time	NoEqu.	LearnMSE	TestMSE	Equ.
1	9.72	11	0.421539	0.436742	P = -1.59607 + 0.603807 * M + -1.30512 * ln(R + 10.1414
1	9.72	11	0.422969	0.43752	P = -1.57498 + 0.616434 * M + -1.31884 * ln(R + 10.4865
2	7.4	11	0.483534	0.444497	P = -1.76222 + 0.603479 * M + -1.27002 * ln(R + 9.36284
2	7.4	11	0.483518	0.445358	P = -1.71081 + 0.614113 * M + -1.2862 * ln(R + 9.73413)
3	14.72	17	0.439311	0.443783	P = -1.61474 + 0.593672 * M + -1.28727 * ln(R + 9.87803
3	14.72	17	0.439311	0.443783	P = -1.61475 + (0.150342 * M + 0.44333 * M) + -1.28727
3	14.72	17	0.438975	0.444804	P = -1.58144 + 0.607943 * M + -1.30461 * ln(R + 10.3207
4	8.17	11	0.469981	0.442355	P = -1.62763 + 0.600441 * M + -1.29249 * ln(R + 10.1689
4	8.17	11	0.471073	0.443309	P = -1.60269 + 0.6149 * M + -1.30905 * ln(R + 10.588) +
5	9.24	11	0.456505	0.43677	P = -1.6936 + 0.602249 * M + -1.27972 * ln(R + 9.82337)
5	9.24	11	0.453958	0.437697	P = -1.67615 + 0.6162 * M + -1.29413 * ln(R + 10.1987)
6	14.39	17	0.456637	0.441535	P = -1.75829 + 0.598626 * M + -1.26545 * ln(R + 9.36271
6	14.39	17	0.456636	0.441535	P = -1.75828 + (0.891628 * M + -0.293002 * M) + -1.265
6	14.39	17	0.457275	0.442646	P = -1.73389 + 0.613988 * M + -1.28235 * ln(R + 9.77602
7	8.67	11	0.462123	0.443094	P = -1.82432 + 0.604532 * M + -1.2575 * ln(R + 9.325) +
7	8.67	11	0.463151	0.444046	P = -1.7904 + 0.617393 * M + -1.27316 * ln(R + 9.69782)
8	9.18	11	0.424271	0.438592	P = -1.56018 + 0.598452 * M + -1.30484 * ln(R + 10.27)
8	9.18	11	0.426472	0.439558	P = -1.54732 + 0.613623 * M + -1.32025 * ln(R + 10.673)
9	7.95	11	0.401571	0.437732	P = -1.64857 + 0.596275 * M + -1.28264 * ln(R + 9.93623
9	7.95	11	0.403731	0.438565	P = -1.62146 + 0.609233 * M + -1.29762 * ln(R + 10.3116
10	8.48	11	0.417520	0.442521	P = -1.65517 + 0.588875 * M + -1.27269 * ln(R + 9.74108
10	8.48	11	0.416513	0.443624	P = -1.61984 + 0.603641 * M + -1.29028 * ln(R + 10.1584

Figure D.4: The results file (cropped on the right-hand side)

The GMPE provides the correlation between different seismically important variables (e.g., PGA, magnitude, source-to-site distance and many more) and help the engineer to estimate a possible earthquake load at the structure site. The problem was solved following the steps described previously.

D.4.1 Entering data set

Peruš & Fajfar [10] assembled two new databases for their experiments from which the larger PF-L database is also used in the present study. The whole data set consists of 3550 recordings and includes aftershocks. The independent variables are:

- the moment magnitude M_{w};
- the style-of-faulting F;
- the source-to-site Joyner-Boore distance R_{jb} in [km]; and
- the average soil shear-wave velocity in the upper 30 meters of soil underneath the observation spot $V_{\mathrm{s},30}$ in [m/s].

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

- $F = 0$ for normal faults;
- $F = 0.5$ for strike-slip faults; and
- $F = 1$ for reverse faults.

The PGA parameter is the dependent variable, measured in [g]-units, and is defined as the geometrical average of both horizontal components [10].

Previous studies have shown that strong-motion amplitudes are log-normally distributed [11], which we incorporated by stating the ED problem as follows:

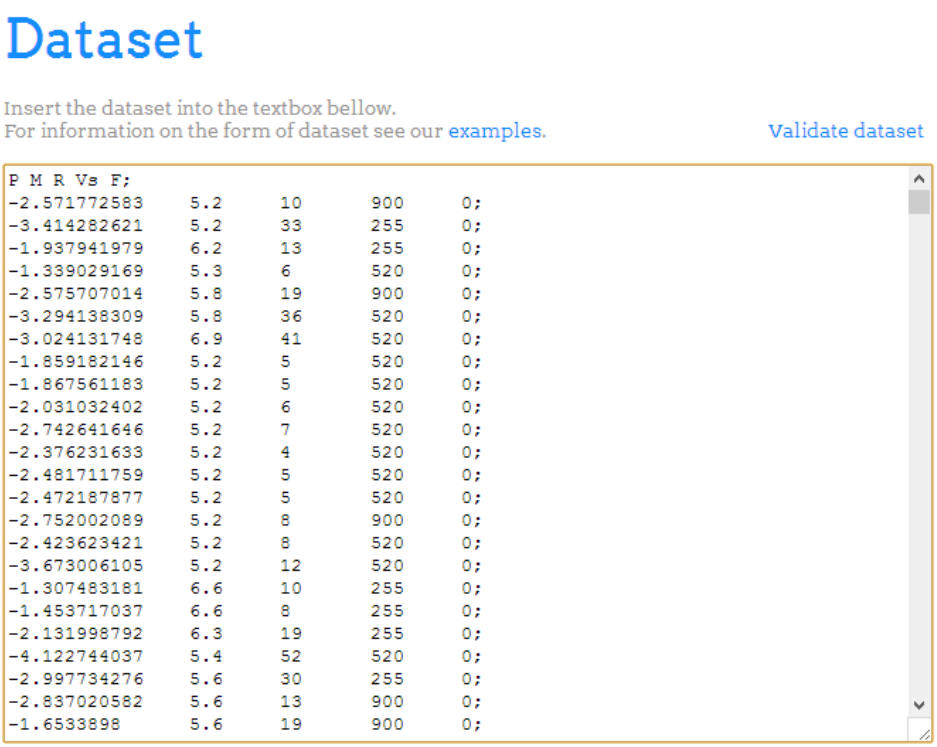


Figure D.5: The submitted data set

$$\ln(PGA) = f(M_{\mathrm{w}}, R_{\mathrm{jb}}, V_{\mathrm{s},30}, F) \tag{D.4}$$

Based on the problem statement in equation (D.4), the original PF-L data set was preprocessed by converting the actual values of the PGA into their logarithmic values. The beginning of the values given to Lagramge system is shown in Figure D.5.

D.4.2 Specifying the context-free grammar

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 [12]. 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. The actual productions of the grammar were defined by systematically studying the formulae designed by earthquake engineers over the past 50 years and selecting the most often modeled dependencies of the variables used in this study.

For example, the productions FM for the magnitude dependencies $f(M_{\mathrm{w}})$ are shown in Figure D.6 below the starting production Eq. We included a recursive call to be able to summarize many subfunctions, which are succeeding the FM1 nonterminal symbol. The productions for R_{jb}, $V_{\mathrm{s},30}$ and F are defined similarly. As the ratio between fault types F is not known and the variable $V_{\mathrm{s},30}$ is often divided into classes, two conditional functions *ifl* and *ife* were defined that allow a comparison of two values for their smallness or equality, respectively. These definitions can be seen at the beginning of the CFG. The whole grammar is not reported in this

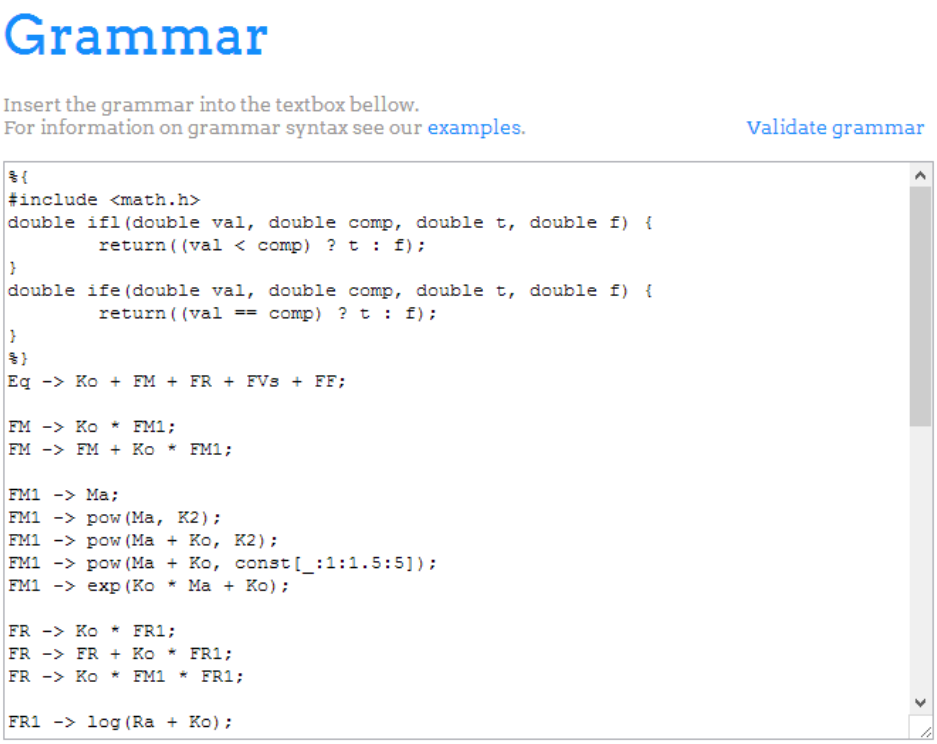


Figure D.6: The submitted CFG

study due to lack of space, however, the reader in encoureaged to find it in [1]. The use of this 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.

D.4.3 Settings

The settings chosen for two experiments (first with exhaustive search, second with beam search) were the following as shown in Figure D.7.

- The option for the cross validation was selected.
- According to ED problem definition in equation (D.4) the dependency we searched for was ordinary.
- The dependent variable was the PGA.
- We chose the MSE criterion, as the length of the GMPE is not that much important.
- The parameter fitting method's restart was set to 50.
- The stopping criteria was the default (all equations tested).
- In the case of exhaustive search, the maximum tree height was 5; in the case of beam search, the maximum tree height was 8.
- The number of equations saved was 50.

D.4.4 Results

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 equations found which reached the lowest

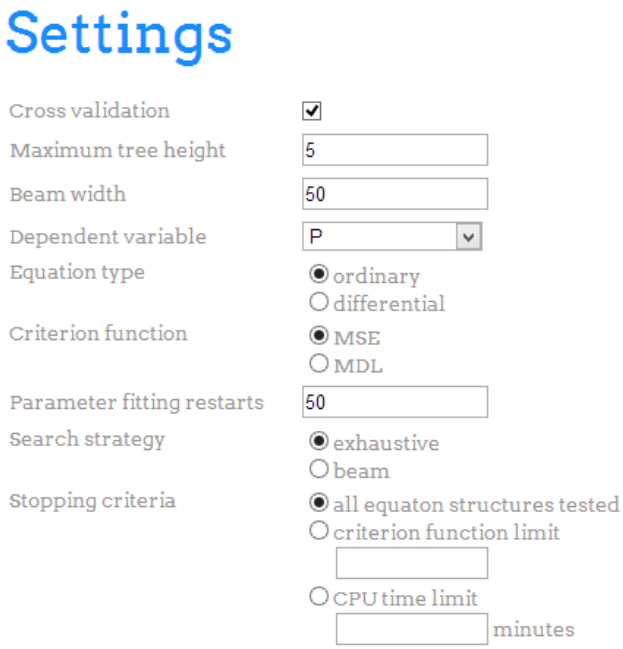


Figure D.7: The selected settings for the experiment with the exhaustive search option

MSE on corresponding learning data set are (D.5) and (D.6). Equation (D.5) reached $MSE = 0.3894$ on its corresponding testing set, while equation (D.6) reached $MSE = 0.3823$.

$$\begin{aligned}\ln(PGA) = \; & 1,23491 - 0,117808 \cdot (M_{\mathrm{w}} - 6,60126)^2 \\ & -0,189924 \cdot e^{-0,129448 \cdot M_{\mathrm{w}} + 1,89843} \cdot \ln(R_{\mathrm{jb}}^2 + 57,2879) \\ & -0,310872 \cdot \ln \frac{V_{\mathrm{s},30}}{464,231} \\ & + \begin{cases} 0,0951288 & \text{if F = 0} \quad \text{(normal)} \\ 0 & \text{if F = 0.5} \quad \text{(strike-slip)} \\ 0,0720131 & \text{if F = 1} \quad \text{(reverse)} \end{cases}\end{aligned} \tag{D.5}$$

$$\begin{aligned}\ln(PGA) = \; & 4,57353 - 1,69293 \cdot M_{\mathrm{w}} + 0,2417 \cdot M_{\mathrm{w}}^2 \\ & -6,67613 \cdot e^{-7,60198 \cdot M_{\mathrm{w}}} - 0,00918368 \cdot \frac{e^{1,3707 \cdot M_{\mathrm{w}}}}{R_{\mathrm{jb}} + 100} \\ & -1,67822 \cdot \ln(R_{\mathrm{jb}} + 12,7587) - 0,291666 \cdot \ln \frac{V_{\mathrm{s},30}}{4000} \\ & + \begin{cases} 0.1254 & \text{if F = 0} \quad \text{(normal)} \\ 0 & \text{if F = 0.5} \quad \text{(strike-slip)} \\ 0.1188 & \text{if F = 1} \quad \text{(reverse)} \end{cases}\end{aligned} \tag{D.6}$$

D.5 Conclusions

The development of the presented Web application is intended for a broader usage of the state-of-the-art ED system Lagramge. With a simplified user interface, the researchers can concentrate more on their research purpose and need not worry about details related to the underlying infrastructure. In this way the research process is shorter and can be managed more efficiently.

A use case from the domain of earthquake engineering is presented, which roots from our

previous study [1]. However, the presented Web application can be used not only in the field of earthquake engineering, but also in other engineering disciplines, where equations are commonly used as models.

Limitations of this Web application root from the limitations of the Lagramge system and the SLING infrastructure. Known problem is that the time limit of a single job is two weeks, making a long exhaustive search impossible, which may be overcome by using the heuristic search possibility, rather than the exhaustive one.

With the use of Grid technology, many users can submit and run their jobs simultaneously in a resilient and fault-tolerant way. The ten fold-cross validation process is fastened by parallel execution of all ten learning processes, therefore providing the results at least ten times faster than the consecutive execution.

Acknowledgements

The authors are grateful to Iztok Peruš and Peter Fajfar for fruitful discussions and for providing the PF-L data set which was used for the experiments in this study. Special thanks go to Ljupčo Todorovski for guidance when using his Lagramge equation-discovery system. This research is partially funded by the European grant FP7-ICT-2009-5-256910 mOSAIC-cloud.eu.

References

[1] Markič, Š., Stankovski, V. An Equation-Discovery Approach to Earthquake-Ground-Motion Prediction. Engineering Applications of Artificial Intelligence, Engineering Apllication of Artificial Inteligence, 26, 4: 1339–1347.
doi:10.1016/j.engappai.2012.12.005

[2] L., Todorovski ,Džeroski, S. 1997 Declarative bias in equation discovery. in: Proceedings of the 14th International Conference on Machine Learning, Morgan Kaufmann: 376–384.

[3] . Langley, P, Simon, H., Bradshaw, G. 1987. Computational Models of Learning, Springer, Berlin.

[4] Todorovski, L., Džeroski, S., 1995. Discovering dynamics: From inductive logic programming to machine discovery. Journal of Intelligent Information Systems 4: 89–108.

[5] Todorovski L., Džeroski S., Kompare, B. 1998. Modelling and prediction of phytoplankton growth with equation discovery. Ecological Modeling 113, 1–3: 71–81.

[6] Kompare, B., Todorovski, L., Džeroski, S. 2001. Modelling and prediction of phytoplankton growth with equation discovery: case study–Lake Glumsø, Denmark. Verhandlungen des Internationalen Verein Limnologie 27: 3626–3631.

[7] Todorovski, L., Džeroski, S. 2001. Using domain knowledge on population dynamics modeling for equation discovery. in Machine Learning: ECML 2001, De Raedt, L., Flach, P., editors, Volume 2167 of Lecture Notes in Computer Science, Springer Berlin / Heidelberg: 478–490.

[8] Ganzert, S., Möller, K., Kramer, S., Kersting, K., Guttmann, J. 2010. Identifying mathematical models of the mechanically ventilated lung using equation discovery. in: World Congress on Medical Physics and Biomedical Engineering, Dössel, O., Schlegel, W.C., editors, IFMBE Proceedings 25/4, Springer Berlin Heidelberg: 1524–1527.

[9] Alzaidi, A., Kazakov, D. 2011. Equation discovery for financial forecasting in the context of islamic banking. in: Proceedings of the 11th IASTED International Conference on Artificial Intelligence and Applications (AIA 2011), Morgan Kaufmann: 97–103.

[10] Peruš, I., Fajfar, P. 2010. Ground-motion prediction by a non-parametric approach. Earthquake Engineering & Structural Dynamics 39: 1395–1416.

[11] Douglas, J., Smit, P.M. 2001. How accurate can strong ground motion attenuation relations be? Bulletin of the Seismological Society of America 91, 6: 1917–1923.

[12] Douglas, J. Ground-motion prediction equations 1964–2010. Final report, BRGM/RP-59356-FR and PEER/2011/102, Pacific Earthquake Engineering Research Center: 444 p., 9 illustrations.

Priloga E Markič, Š., Stankovski, V. Developing Context-Free Grammars for Equation Discovery: An Application in Earthquake Engineering

Diplomskemu delu prilagamo objavo (v tisku) na konferenci *The 26th International Conference on Industrial, Engineering & Other Applications of Applied Intelligent Systems (IEA/AIE 2013)*, ki je organizirana med 17. in 21. junijem 2013 v Amsterdamu na Nizozemskem, z naslovom *Developing Context-Free Grammars for Equation Discovery: An Application in Earthquake Engineering*.

Abstract. In the machine-learning area of equation discovery (ED) context-free grammars (CFG) can be used to generate equation structures that best describe the dependencies in a given data set. Our goal is to investigate the possible strategies of incorporating domain knowledge into a CFG, and evaluate the effect on the obtained results in the ED process. As a case study, the Lagramge ED system is used to discover equations that predict the peak ground acceleration (PGA) in an earthquake event. Existing equations for PGA represent rich domain knowledge and are used to form three different CFGs. The obtained results demonstrate that the inclusion of domain knowledge in the CFG which is neither too general, neither too specific, may lead to new, high-precision equation models for PGA.

Keywords. equation discovery, Lagramge, context-free grammar, domain knowledge, earthquake engineering, peak ground acceleration

E.1 Introduction

Equation discovery (ED) is a sub area of machine learning aiming at automatic induction of mathematical models expressed as equations. The goal is to find an equation structure from a given set of operators, functions and variables that represents an appropriate model for the provided data set. ED systems like Lagramge[2] use the context-free grammar (CFG) formalism to restrict the hypothesis space of possible equation structures [4, 6, 8]. Usually, this is achieved by incorporating domain knowledge in the productions of the CFG. The construction of a CFG, however, requires considerable know-how, as also noted by [4], and may range from more general to more explicit specification of existing equation structures. The goals of the present study are therefore:

- to investigate the possible ways of forming the CFGs;
- to compare the various ways of inclusion of domain knowledge; and
- to observe the effects on the obtained results, by following a motivating example.

E.1.1 Case study.

In civil engineering an important task is to properly design a structure, bearing in mind that a devastating earthquake could occur during its lifetime. The ground-motion prediction equations

[2] The Lagramge release 2.2 used in this study is available as open-source software at URL: http://www-ai.ijs.si/∼ljupco/ed/lagrange.html (accessed 6^{th} February 2012)

$$\begin{aligned}\log_{10}(PGA) = \; & 1.04159 + 0.91333 \times M_{\mathrm{w}} - 0.08140 \times M_{\mathrm{w}}^2 \\ & +(-2.92728 + 0.28120 \times M_{\mathrm{w}}) \times \log_{10}\sqrt{R_{\mathrm{jb}}^2 + 7.86638^2} \\ & + \begin{cases} 0.08753 & \text{if} \quad V_{\mathrm{s},30} < 360 \\ 0.01527 & \text{if} \quad 360 \leq V_{\mathrm{s},30} < 800 \\ 0 & \text{if} \quad 800 \leq V_{\mathrm{s},30} \end{cases} + \begin{cases} -0.04189 & \text{if} \quad \mathsf{F = 0} \\ 0 & \text{if} \quad \mathsf{F = 0.5} \\ 0.08015 & \text{if} \quad \mathsf{F = 1} \end{cases}\end{aligned} \tag{E.1}$$

$$\ln(PGA) = f(M_{\mathrm{w}}, R_{\mathrm{jb}}, V_{\mathrm{s},30}, F) \tag{E.2}$$

help the structural engineer to estimate the possible earthquake load by providing the correlation between seismically important variables, (e.g., peak ground acceleration (PGA)) and significant seismological aspects (e.g., magnitude and distance) [2].

An example of a modern equation from [1] is presented in Eq. (E.1), and the ED problem is formulated as Eq. (E.2).

E.1.2 Data set.

The PF-L data set used in this study consists of 3550 earthquake recordings and is taken from the study of [7]. The data set is very sparse at high magnitudes and short distances. The independent variables used in this study are similarly to [6, 1, 7]:

- the moment magnitude M_{w};
- the source-to-site Joyner-Boore distance R_{jb} (km);
- the average soil shear-wave velocity in the upper 30 meters of soil $V_{\mathrm{s},30}$ (m/s); and
- the style-of-faulting F with values of
  - $F = 0$ for normal;
  - $F = 0.5$ for strike-slip; and
  - $F = 1$ for reverse faults.

The dependent variable is PGA (g-units), defined as the geometrical average of both horizontal components.

E.2 The Lagramge System

The Lagramge ED system [8] takes as input two input files: a data set and a CFG. The data set consists of a table of measurements of dependent $ln(PGA)$ and independent variables $M_{\mathrm{w}}, R_{\mathrm{jb}}, V_{\mathrm{s},30}, F$. The $CFG = \{N, T, P, S\}$ prescribes the syntax of an equation. First, it contains finite disjunctive sets of non-terminals N and terminals T. The terminals are all the independent variables and a special symbol *const*, which is explained under the parameter fitting paragraph. The most important part of the CFG are the productions $P = \{P_1, P_2, \ldots, P_n\}$, which denote the grammatical rules that relate the non-terminals among themselves and to the terminals. The standard form of a production P is $A \to \alpha$, where $A \in N$, $\alpha \in N \cup T$. The operators or functions used can be already or user-defined in the programming language C. The Lagramge system uses the annotation with the logical *or* operator $A \to \alpha_1 | \alpha_2 | \ldots | \alpha_n$ for

productions $A \rightarrow \alpha_1, A \rightarrow \alpha_2, \ldots, A \rightarrow \alpha_n$. Finally, $S \in N$ is a special non-terminal symbol, from which the derivation of the expressions starts.

E.2.1 Parameter fitting.

During the derivation process, Lagramge continuously applies productions to all the non-terminals until all the symbols in the expression (formula structure) are terminals. Such an expression contains one or more special terminal symbols *const*, the syntax of which is $[name : lowest\ value : starting\ value : highest\ value]$. A non-linear fitting method, either Downhill Simplex or Levenberg-Marquardt, is used to determine the values of these symbols. The fitting procedure can be repeated (by setting up the Lagramge parameter m). In the ED process the Lagramge system minimizes the value of the Mean-Squared Error (MSE) function given in Eq. (E.3), in which n is the number of records and PGA and $\widehat{PGA}$ are the measured and predicted values of the PGA, respectively.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(PGA - \widehat{PGA})^2 \tag{E.3}$$

E.2.2 Search strategies.

The Lagramge system provides two search strategies:

- an exhaustive search strategy, where all possible equation structures in the hypothesis space are fitted; and
- a heuristic, also called beam search strategy, according to which one can set the number of equations saved in each production step with the value of the input parameter beam width b.

E.3 Approaches to the CFG Definition

Three unique CFGs were defined, each with a different level of incorporation of domain knowledge that also takes the form of equations systematised by [3]. The defined CFGs are presented in the following paragraphs along with a description of the rationale of the approaches. As the ratio between fault types F is not known and the variable $V_{\mathrm{s},30}$ is often divided into classes, two conditional functions *ifl* and *ife* were defined that compare two values for their smallness or equality, respectively. To improve the readability of the grammars twelve auxiliary productions were defined:

- Ma, Ra, Vs and Fa productions lead to the variables' addresses known to the Lagramge algorithm, e.g. Ma $\rightarrow$ variable_M;
- K0, K1, K2, K180, K360, K750, K800 productions lead to presumed constant values, e.g. K0 $\rightarrow$ const[_:0:0:0]; and
- the production Ko $\rightarrow$ const[_:-100:0.1:100] denotes the symbol *const* fitted to the data which is limited to values between -100 and 100 based on the literature review [3].

The definitions and productions were included in all CFGs and are presented in Table E.1A.

Table E.1: *ife* and *ifl* functions, auxiliary productions and designed CFGs

A) *ifl* and *ife* functions and auxiliary productions	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); }
		Ma $\rightarrow$ variable_M
		. . . similar . . .
		K0 $\rightarrow$ const[_:0:0:0]
		. . . similar . . .

B) General CFG	A $\rightarrow$ A + A | (A) $\times$ (A) | (A) / (A) | pow(A, const[_:0:0.1:5]) | exp(A) | log(A) | Ma | Ra | Vs | Fa | Ko	
C) Specialized CFG – Eq. (E.1) as illustrative example	E $\rightarrow$ Ko + Ko $\times$ Ma + Ko $\times$ pow(Ma, K2) + (Ko + Ko $\times$ Ma) $\times$ log(sqrt(pow(Ra, K2) + Ko)) + ifl(Vs, K360, Ko, ifl(Vs, K750, Ko, K0)) + ife(Fa, K0, Ko, ife(Fa, K1, Ko, K0))	
D) Intermediate CFG	Eq $\rightarrow$ Ko + FM + FR + FV + FF FM $\rightarrow$ (FM + Ko $\times$ FM1) | Ko $\times$ FM1 FM1 $\rightarrow$ Ma | pow(Ma, K2) | pow(Ma + Ko, const[_:1:1.5:5]) | exp(Ko $\times$ Ma) FR $\rightarrow$ Ko $\times$ FM1 $\times$ FR1 | FR + Ko $\times$ FR1 | Ko $\times$ FR1 FR1 $\rightarrow$ ln(Ra + Ko) | ln(Ra + Ko $\times$ FM1) | ln(pow(Ra, K2) + Ko) | ln(pow(Ra, K2) + Ko $\times$ FM1) | pow(Ra + Ko, -K1) | pow(Ra + Ko, -K2) FV $\rightarrow$ FM1 $\times$ FV1 | FV1 FV1 $\rightarrow$ K0 | ifl(Vs, K180, Ko, ifl(Vs, K360, Ko, ifl(Vs, K800, Ko, K0))) | Ko $\times$ ln(Vs/const[_:0:800:4000]) | ifl(Vs, const[_:0:800:4000] , Ko, K0) FF $\rightarrow$ ife(Fa, K1, Ko, ife(Fa, K0, Ko, K0)) | K0	

E.3.1 General CFG.

With the CFG provided in Table E.1B very diverse equation structures can be built and tested, hence, it is named General. It contains all the different functions and operators used in already existing equations, which can be combined together in all possible ways with the possibility of recursion. The exponent in the power term is limited between values of 0 and 5, as negative powers are not needed because the production A $\rightarrow$ (A)/(A) can generate them and powers greater than 5 were not seen in [3].

E.3.2 Specialized CFG.

For the second CFG all the existing equation structures developed for the PGA modeling published by European authors from these subsections of Section 2 of the study [3] were transcribed in grammar productions: 12, 16, 18, 22, 23, 34, 35, 40, 46, 50, 59, 67, 72, 74, 76, 84, 86, 88, 92, 102, 108, 113, 118–120, 124, 128, 146, 152, 157, 165, 175, 179, 181, 187, 189, 191, 192, 195, 197, 198, 202, 205–211, 235, 239, 242, 254, 256, 260, 263, 266, 275, 276, 277, 282, 283, 288 and 289. An explicit use of the depth variable h was substituted with a *const* parameter, as the PF-L database does not include such information. The resulting Specialized CFG (see Table E.1C) includes all-together 62 different equation structures from 64 published articles and is not included here due to lack of space.

Table E.2: Calculated averages and standard deviations of the MSE criterion on the testing data sets for Eq. (E.1), (E.4), (E.5), (E.6) and (E.7)

Equation	(E.1)	(E.4)	(E.5)	(E.6)	(E.7)
$\overline{MSE}$	0.4595	0.4569	0.4138	0.4135	0.3957
σ_{MSE}	0.0257	0.0296	0.0244	0.0245	0.0236

E.3.3 Intermediate CFG.

The design of the third CFG was one of the most difficult tasks undertaken. The actual productions were defined by abstracting the formulae copied for the Specialized CFG and provide the combinatorial freedom of the General CFG. The use of this CFG (see Table E.1D) first leads from the root symbol Eq to non-terminal functions FM, FR, FV and FF, named after the dependence they model, e.g., FM for $f(M_{\mathrm{w}})$. Each of these functions can then be succeeded with their own special sub-functions gathered during the literature review [3]. The Intermediate CFG limits the space of possible equations to only those that are the most plausible according to the studied domain knowledge.

E.4 Results

Based on some initial trial and error experiments using the recently developed Web application [5] it was decided to explore the hypothesis space:

- of the General CFG with beam search $d = 7, b = 50$;
- of the Specialized CFG with exhaustive search; and
- of the Intermediate CFG with exhaustive search $d = 4$ and beam search $d = 10, b = 50$.

The parameter m for fitting restarts was set to 50. The PF-L data set was preprocessed by converting the PGA into their logarithmic values and randomly split 10 times in a 90 % to 10 % proportion, with the purpose of a 10-fold cross validation. The best equations found from all four experiments are:

- Eq. (E.4) for the General CFG;
- Eq. (E.5) for the Specialized CFG;
- Eq. (E.6) for the Intermediate CFG with exhaustive search; and
- Eq. (E.7) for the Intermediate CFG with beam search.

The averages and standard deviations of the MSE criterion on all testing data sets for the Eqs. (E.1), (E.4), (E.5), (E.6) and (E.7) are shown in Table E.2.

E.5 Conclusions

In this study, the Lagramge ED system was used to induce equations that predict the earthquake's PGA. In the past decades many authors have addressed this problem and developed numerous equations, which represent rich and specific domain knowledge. The existing domain knowledge was formalised in the CFGs to find new, potentially more accurate models by following three different approaches. The first approach (General CFG) took into account only the

$$\ln(PGA) = -0.563429 \times (8.8822 + 2 \times M_{\mathrm{w}} + R_{\mathrm{jb}}^{0.392507} + \ln(M_{\mathrm{w}})) \tag{E.4}$$

$$\begin{aligned}\ln(PGA) = & -4.78208 + 1.90683 \times M_{\mathrm{w}} - 0.152475 \times M_{\mathrm{w}}^2 \\ & + (-2.20764 + 0.169162 \times M_{\mathrm{w}}) \times \ln\sqrt{R_{\mathrm{jb}}^2 + 61.1355} \\ & + \begin{cases} 0.490597 & \text{if} \quad V_{\mathrm{s},30} < 180 \\ 0.297185 & \text{if} \quad 180 \leq V_{\mathrm{s},30} < 360 \\ 0.0577997 & \text{if} \quad 360 \leq V_{\mathrm{s},30} < 750 \\ 0 & \text{if} \quad 750 \leq V_{\mathrm{s},30} \end{cases} + \begin{cases} 0.07482 & \text{if} \quad \mathsf{F = 0} \\ 0 & \text{if} \quad \mathsf{F = 0.5} \\ 0.09791 & \text{if} \quad \mathsf{F = 1} \end{cases}\end{aligned} \tag{E.5}$$

$$\begin{aligned}\ln(PGA) = & \, 1.23491 - 0.117808 \times (M_{\mathrm{w}} - 6.60126)^2 \\ & - 0.18992 \times \exp(-0.12945 \times M_{\mathrm{w}} + 1.8984) \times \ln(R_{\mathrm{jb}}^2 + 57.2879) \\ & - 0.310872 \times \ln\frac{V_{\mathrm{s},30}}{464.231} + \begin{cases} 0.0951288 & \text{if} \quad \mathsf{F = 0} \\ 0 & \text{if} \quad \mathsf{F = 0.5} \\ 0.0720131 & \text{if} \quad \mathsf{F = 1} \end{cases}\end{aligned} \tag{E.6}$$

$$\begin{aligned}\ln(PGA) = & \, 4.5735 - 1.6929 \times M_{\mathrm{w}} + 0.2417 \times M_{\mathrm{w}}^2 - 6.6761 \times \exp(-7.6020 \times M_{\mathrm{w}}) \\ & - 0.0091837 \times \frac{\exp(1.3707 \times M_{\mathrm{w}})}{R_{\mathrm{jb}} + 100} - 1.6782 \times \ln(R_{\mathrm{jb}} + 12.7587) \\ & - 0.291666 \times \ln\frac{V_{\mathrm{s},30}}{4000} + \begin{cases} 0.1254 & \text{if} \quad \mathsf{F = 0} \\ 0 & \text{if} \quad \mathsf{F = 0.5} \\ 0.1188 & \text{if} \quad \mathsf{F = 1} \end{cases}\end{aligned} \tag{E.7}$$

basic functions and operators which are present in the existing equations. The second approach (Specialized CFG) took into account the exact formulae taken from 64 articles of European authors [3]. The third approach (Intermediate CFG) was a combination of the previous two and took into account only the most often modeled variables' dependencies in existing equations, but allowing them to be combined freely.

Our investigation shows that the use of domain knowledge contributes to the discovery of more precise equation models for PGA. The inclusion of strict equation structures in Specialized CFG provided a 10% reduction of the MSE criterion when compared to General CFG. However, the combination of both approaches in Intermediate CFG performed even better providing a 15% reduction. We conclude that careful definition of grammar productions defines an infinite, but quality hypothesis space and may lead to obtaining the best results.

Acknowledgments

The authors are grateful to Iztok Peruš and Peter Fajfar for providing the PF-L data set and fruitful discussions. Special thanks go to Ljupčo Todorovski for guidance when using the Lagramge ED system.

References

[1] Akkar, S., Bommer, J.J. 2010. Empirical equations for the prediction of PGA, PGV, and spectral accelerations in Europe, the Mediterranean region, and the Middle east. Seismological Research Letters 81, 2: 195–206

[2] Douglas, J. 2003. Earthquake ground motion estimation using strong-motion records: a review of equations for the estimation of peak ground acceleration and response spectral ordinates. Earth-Science Reviews 61, 1-2: 43–104.

[3] Douglas, J. 2011. Ground-motion prediction equations 1964–2010. Final report, BRGM/RP-59356-FR and PEER/2011/102, Pacific Earthquake Engineering Research Center: 444 p., 9 illustrations.

[4] Kompare, B., Todorovski, L., Džeroski, S. 2001. Modelling and prediction of phytoplankton growth with equation discovery: case study–Lake Glumsø, Denmark. Verhandlungen des Internationalen Verein Limnologie 27: 3626–3631.

[5] Markič, Š., Dirnbek, J., Stankovski, V. 2013. A Grid Application for Equation Discovery in the Earthquake Engineering Domain. In: Proceedings of the 3rd International Confernce on Parallel, Distributed, Grid and Cloud Computing for Engineering, Civil-Comp Press.

[6] Markič, Š, Stankovski, V.2013. An equation-discovery approach to earthquake-ground-motion prediction. Engineering Applications of Artificial Inteligence 26, 4: 1339–1347. doi:10.1016/j.engappai.2012.12.005

[7] Peruš, I., Fajfar, P. 2010. Ground-motion prediction by a non-parametric approach. Earthquake Engineering & Structural Dynamics 39, 6: 1395–1416.

[8] Todorovski, L., Džeroski, S. 1997. Declarative bias in equation discovery. in: Proceedings of the 14th International Conference on Machine Learning, 376–384.