

STARN ZA POPRAVKE, ERRATA

Stran z napako

Vrstica z napako

Namesto

Naj bo

IZJAVE

Podpisani Žiga Šebenik izjavljam, da sem avtor diplomske naloge z naslovom »Testiranje uporabnosti programskega vmesnika SAP2000«.

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

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

Ljubljana, 11. 10. 2013

Žiga Šebenik

BIBLIOGRAFSKO – DOKUMENTACIJSKA STRAN IN IZVLEČEK

UDK:	004:624.042.7 (043.2)
Avtor:	Žiga Šebenik
Mentor:	izr. prof. dr. Matjaž Dolšek
Somentor:	doc. dr. Matevž Dolenc
Naslov:	Testiranje uporabnosti programskega vmesnika SAP2000
Tip dokumenta:	diplomska naloga – univerzitetni študij
Obseg in oprema:	87 str., 6 pregl., 113 sl., 57 en., 3 pril.
Ključne besede:	programski vmesnik, API, SAP2000, parametrične študije, avtomatizacija, nihajni časi, modalna analiza, analiza na osnovi potresnega scenarija, SSPA, vtičnik, baza akceleroگرامov

Izvleček

S pomočjo programskega vmesnika za program SAP2000, ki je program za računanje konstrukcij po metodi končnih elementov, je mogoče vzpostaviti dvosmerno povezavo z drugo programsko opremo. Zato smo v okviru diplomske naloge preučili možnosti uporabe programskega vmesnika za razvoj nove, bolj specifične programske opreme in za razvoj novih možnosti znotraj programa SAP2000.

V prvem delu diplomske naloge najprej predstavimo programski vmesnik, programske jezike in tehnologije, ki smo jih uporabili pri izdelavi programske opreme ter teoretične osnove dinamike konstrukcij. V drugem delu prikazujemo različne možnosti uporabe programskega vmesnika programa SAP2000. Najprej prikažemo program za modalno analizo enostavnih konstrukcij, ki je integriran v program Microsoft Excel, sledi predstavitev vtičnika za dinamično analizo na osnovi potresnega scenarija in prikaz še nekaterih drugih možnosti uporabe programskega vmesnika. V okviru vtičnika za dinamično analizo na osnovi potresnega scenarija smo vzpostavili bazo akceleroگرامov PEER centra iz Kalifornije, ki jo uporabljamo za izbor akceleroگرامov na osnovi potresnega scenarija, ki se nanaša na magnitudo potresa, oddaljenost potresa, tip tal ter nekatere ostale parametre potresov.

Z razvitimi aplikacijami smo pokazali možnost razširitve uporabniškega vmesnika za interakcijo s programom SAP2000, deloma avtomatizirali proces analize na osnovi potresnega scenarija, nadgradili funkcionalnost programa SAP2000 in avtomatizirali parametrične analize. Na te načine lahko izboljšamo procese za projektiranje objektov ter omogočimo bolj detaljne študije, katere so precej zamudne predvsem zaradi priprave podatkov za izvajanje analiz.

BIBLIOGRAPHIC – DOCUMENTALISTIC INFORMATION AND ABSTRACT

UDC: 004:624.042.7 (043.2)
Author: Žiga Šebenik
Supervisor: Associate Prof. Matjaž Dolšek, Ph.D.
Co-supervisor: Assistant Prof. Matevž Dolenc, Ph.D.
Title: Usability testing of application programming interface for SAP2000
Document type: Graduation Thesis – University studies
Scope and tools: 87 p., 6 tab., 113 fig., 57 eq., 3 ann.
Keywords: application programming interface, API, SAP2000, parametric studies, automation, natural frequency, modal analysis, scenario based seismic analysis, SSPA, plugin, strong motion database

Abstract

The SAP2000 Application Programming Interface (API) was developed to enable a two-way link between the SAP2000, which is a general-purpose structural program, and other software. In the context of this thesis, the possibility of using the API to enhance the performance within the SAP2000 program and to develop new, more specific software tools based on SAP2000 is examined.

In the first part of the thesis we present the application programming interface, the implementation of programming languages and technologies used for the production of software, along with theoretical backgrounds of structural dynamics. The second part demonstrates the potential of the API based on applications, which were developed for the purpose of this thesis. The first application is a robust tool for modal analysis of simple structures, which is integrated in Microsoft Excel. The second application is a comprehensive SAP2000 plug-in for dynamic analysis based on seismic scenarios, which can be defined by the magnitude, source-to-site distance, soil type and some other parameters of earthquakes. The plugin is supported by a database, which includes the ground motions from the PEER strong-motion database. The thesis concludes with insight into applications, which can be developed using the SAP2000 API.

It was found that the SAP2000 API makes it possible to develop applications which enhance user interfaces and functionality of the SAP2000 and allow further automation of analysis processes and parametric studies. These applications can significantly facilitate tasks, associated with the design of structures. Furthermore, the optimized functionality and analysis processes enables new, more detailed studies, which are otherwise quite time-consuming in terms of preparing information, which are required for analysis of structures.

ZAHVALA

Zahvaljujem se mentorju izr. prof. dr. Matjažu Dolšku, za ves posvečen čas, strokovno razlago in konstruktivne napotke, s čimer mi je pomagal pri izdelavi diplomske naloge.

Zahvaljujem se somentorju doc. dr. Matevžu Dolencu za vse odgovore na moja vprašanja in nasvete, ki so mi vedno prišli prav.

Hvala tudi asist. dr. Robertu Klincu, predvsem za ves trud in čas, ki ga je vložil v projekt vzpostavitve relacijske baze podatkov.

Hvala vsem prijateljem za njihovo pomoč ter nepozabna študijska leta, ki smo jih preživeli skupaj.

Posebno se zahvaljujem svoji mami, ki mi je bila skozi vsa leta v moralno podporo in vzpodbudo.

Vsem, iskrena hvala.

KAZALO VSEBINE

BIBLIOGRAFSKO – DOKUMENTACIJSKA STRAN IN IZVLEČEK	III
BIBLIOGRAPHIC – DOCUMENTALISTIC INFORMATION AND ABSTRACT	IV
ZAHVALA.....	V
KAZALO VSEBINE.....	VI
KAZALO PREGLEDNIC	VIII
KAZALO SLIK.....	IX

1 UVOD 1**2 TEORETIČNE OSNOVE PROGRAMIRANJA..... 3**

2.1 Dinamične povezovalne knjižnice (DLL)	3
2.2 Programski vmesnik (API)	3
2.3 API za program SAP2000	4
2.4 Uporaba programskega vmesnika za SAP2000.....	5
2.5 Uporabljeni programski jeziki in formati za izmenjavo podatkov	6
2.5.1 Visual Basic for Applications (VBA).....	6
2.5.2 Visual Basic .NET	7
2.5.3 Structured Query Language (SQL).....	7
2.5.4 Hypertext Preprocessor (PHP).....	7
2.5.5 MATLAB	8
2.5.6 Extensible Application Markup Language (XAML).....	8
2.5.7 JavaScript Object Notation (JSON).....	8

3 TEORETIČNE OSNOVE DINAMIKE GRADBENIH KONSTRUKCIJ..... 10

3.1 Sistemi z eno prostostno stopnjo	10
3.1.1 Lastno nihanje	10
3.1.2 Vsiljeno nihanje.....	11
3.1.3 Duhamelov integral	12
3.1.4 Integracija korak za korakom	12
3.1.5 Spektri odziva.....	14
3.2 Sistemi z več prostostnimi stopnjami	14
3.2.1 Lastno nihanje	15
3.2.2 Vsiljeno nihanje.....	15

4 PROGRAM ZA RAČUN NIHAJNIH ČASOV IN MODALNE ANALIZE S SPEKTRI 20

4.1 Motivacija za izdelavo programa	20
4.2 Opis programa	20
4.2.1 Izdelava programa	21
4.2.2 Opis parametrov in algoritmov.....	21
4.3 Navodila za uporabo programa.....	31
4.3.1 Zahteve in namestitve	31
4.3.2 Vnos parametrov za definicijo konstrukcije.....	33
4.3.3 Vnos parametrov za definicijo analize	36
4.3.4 Spreminjanje geometrije konstrukcije	37
4.3.5 Analiza in rezultati	41
4.4 Zaključni komentarji k programu	44

5 VTIČNIK ZA ANALIZO KONSTRUKCIJE NA OSNOVI POTRESNEGA SCENARIJA.... 45

5.1 Motivacija za izdelavo programa	45
---	----

5.2 Opis programa	45
5.2.1 Vzpostavitev baze akceleroگرامov na strežniku fakultete	45
5.2.2 Izdelava programa	51
5.3 Navodila za uporabo programa SSPA verzije 1.0	52
5.3.1 Zahteve in namestitve	52
5.3.2 Odpiranje vtičnika	55
5.3.3 Iskanje v bazi podatkov potresnega delovanja	56
5.3.4 Pregled rezultatov in uvoz izbranih akceleroگرامov v program SAP2000	59
5.3.5 Generiranje obtežnih primerov na podlagi uvoženih akceleroگرامov	64
5.4 Zaključni komentarji k programu	70
6 DRUGI PRIMERI UPORABE PROGRAMSKEGA VMESNIKA SAP2000	72
6.1 Povezava s programom Matlab na primeru generiranja mrež končnih elementov	72
6.1.1 Opis programa DistMesh	72
6.1.2 Prenos mrež končnih elementov v program SAP2000	73
6.2 Povezava s programom Rhinocerus 3D ter vtičnikom Grasshopper 3D	75
7 ZAKLJUČEK	81
VIRI	83

KAZALO PREGLEDNIC

Preglednica 1:	Vrednosti parametrov S , TB , TC in TD , glede na tip tal	25
Preglednica 2:	Uporabljene knjižnice, ki niso vgrajene v ogrodje Microsoft .NET Framework ...	52
Preglednica 3:	Dodatni kriteriji za omejitev iskanja v bazi podatkov potresnega delovanja	58
Preglednica 4:	Tip tal, določen na podlagi vrednosti $Vs30$	58
Preglednica 5:	Opis kategorij posameznih stolpcev z rezultati	60
Preglednica 6:	Ukaza za generiranje mreže končnih elementov enotskega kroga s programom Distmesh v programu Matlab	73

KAZALO SLIK

Slika 1:	Shema komunikacije programskih komponent preko programskega vmesnika.....	4
Slika 2:	Tipičen tok podatkov med razvito programsko opremo in programom SAP2000 s pomočjo programskega vmesnika [8].....	6
Slika 3:	Univerzalna struktura formata JSON, tipa objekt [25]	9
Slika 4:	Univerzalna struktura formata JSON, tipa polje [25]	9
Slika 5:	Podatkovni tipi in strukture, ki so lahko v formatu JSON zastopane v obliki vrednosti, [25].....	9
Slika 6:	Metode za integracijo diferencialne enačbe nihanja, ki so na voljo v programu SAP2000	13
Slika 7:	Pogovorno okno programa SAP2000, iz katerega so razvidne možne metode za kombinacijo pomikov in notranjih sil.....	19
Slika 8:	Shematski prikaz tlorisne razporeditve mase konstrukcije, glede na njen izvor	24
Slika 9:	Grafični vmesnik za prilagoditev osnovne okvirne konstrukcije	26
Slika 10:	Shema dimenzij, za izračun izhodiščnih vrednosti mase in masnega vztrajnostnega momenta etaže	28
Slika 11:	Prikaz odstranjenih celic, ki označujejo nanos porazdeljene obtežbe (odstranjena sta 7. in 8. polje tretje etaže)	29
Slika 12:	Shema dimenzij, za izračun nove mase in masnega vztrajnostnega momenta modificirane etaže.....	30
Slika 13:	Varnostno opozorilo programa Microsoft Excel	32
Slika 14:	Gumb 'Reference' menija 'Orodja' (angl. 'Tools'), s katerim odpremo pogovorno okno za izbiro referenc	32
Slika 15:	Pogovorno okno za izbiro referenc, v katerem obkljukamo dinamično povezovalno knjižnico z imenom 'SAP2000v15'	33
Slika 16:	Osnovni vmesnik za upravljanje programa in podajanje parametrov	33
Slika 17:	Polja za izbiro tipa okvirne konstrukcije, vnos števila njenih razponov in dimenzij.....	34
Slika 18:	Polji za vnos števila etaž in etaže višine	34
Slika 19:	Prikaz izbire materiala iz ponujenega seznama	35
Slika 20:	Prikaz izbire oblike prereza iz ponujenega seznama	35
Slika 21:	Prikaz vnosa dimenzij pravokotnega prereza	35
Slika 22:	Prikaz vnosa dimenzij okroglega prereza	35
Slika 23:	Prikaz izbire standardnih prerezov	35
Slika 24:	Območje za definicijo projektnih spektrov pospeškov	36
Slika 25:	Območje za definicijo obtežnih primerov	37
Slika 26:	Grafični vmesnik za prilagoditev geometrije osnovne okvirne konstrukcije	37
Slika 27:	Izbor celic z orodjem 'Pobriši'	38
Slika 28:	Grafičen prikaz odstranjenih elementov	38
Slika 29:	Izbor celic z orodjem 'Dodaj povezje'	39
Slika 30:	Grafičen prikaz dodanih povezij	39
Slika 31:	Izbor celic z orodjem 'Obrni lokalno os elementa'	39
Slika 32:	Grafičen prikaz elementov z rotirano lokalno osjo.....	39
Slika 33:	Izbor celic z orodjem 'Stene'	40
Slika 34:	Grafičen prikaz dodanih sten	40
Slika 35:	Izbor celice z orodjem 'Info', odebeljen izpis informacij o izbranem elementu in prikaz števila odstranjenih elementov ter polj, števila elementov z obrnjenimi lokalnimi osmi in števila dodanih povezij ter sten celotne konstrukcije	40
Slika 36:	Primer izpisa nihajnih časov in frekvenc za posamezno nihajno obliko	41
Slika 37:	Primer izpisa nihajnih časov, efektivnih mas in njihovih vsot za posamezno nihajno obliko	41
Slika 38:	Primer izpisa horizontalnih pomikov masnih točk za vsako etažo	42
Slika 39:	Primer izpisa prečnih sil in momentov v stebrih za vsako etažo	42
Slika 40:	Osnovni diagram poteka algoritma programa	43
Slika 41:	Oblika spletnih naslovov akceleroگرامov na strežniku Univerze v Berkeleyu	48

Slika 42:	Glava in prvi 2. od 13000-ih vrstic zabeleženih pospeškov prenesene datoteke akcelrograma, zabeleženega na merilnem mestu CHY116 v eni izmed horizontalnih smeri potresa CHI-CHI, Taiwan.....	49
Slika 43:	Grafičen prikaz celotnega akcelrogram, zabeleženega na merilnem mestu CHY116 v eni izmed horizontalnih smeri, potresa CHI-CHI, Taiwan.....	49
Slika 44:	Razporeditev vrednosti nihajnih časov v spektrih pospeškov (linearno merilo).....	50
Slika 45:	Razporeditev vrednosti nihajnih časov v spektrih pospeškov (logaritmično merilo)	50
Slika 46:	Grafičen prikaz spektra pospeškov, izračunanega na podlagi akcelrograma, zabeleženega na merilnem mestu CHY116 v eni izmed horizontalnih smeri, potresa CHI-CHI, Taiwan	51
Slika 47:	Prvo pogovorno okno namestitve ogrodja Microsoft Framework 4.5	53
Slika 48:	Drugo pogovorno okno namestitve ogrodja Microsoft Framework 4.5.....	53
Slika 49:	Prvo pogovorno okno namestitve vtičnika.....	53
Slika 50:	Drugo pogovorno okno namestitve vtičnika.....	53
Slika 51:	Tretje pogovorno okno namestitve vtičnika.....	54
Slika 52:	Četrto pogovorno okno namestitve vtičnika	54
Slika 53:	Peto pogovorno okno namestitve vtičnika	54
Slika 54:	Prikaz odpiranja okna za dodajanje vtičnikov preko menija 'Tools' v programu SAP2000	54
Slika 55:	Okno preko katerega v programu SAP2000 dodamo ali odstranimo nameščene vtičnike.....	55
Slika 56:	Prikaz uspešno nameščenega vtičnika v programu SAP2000.....	55
Slika 57:	Prikaz novo ustvarjenega gumba v meniju 'Tools', s katerim odpremo uspešno nameščen vtičnik SSPA	56
Slika 58:	Pozdravno okno vtičnika.....	56
Slika 59:	Prikaz zavihka z imenom 'Search', razdeljenega na štiri osnovne module	57
Slika 60:	Primer izbire merilnega mesta iz ponujenega seznama	57
Slika 61:	Primer izbire merilnega mesta iz skrčenega seznama v primeru predhodno izbranega potresa	57
Slika 62:	Opozorilo zaradi napačne oblike vnesenega kriterija, kjer je za ločilo dveh števil namesto vejice uporabljena pika	59
Slika 63:	Določitev maksimalnega števila vrnjenih rezultatov	59
Slika 64:	Primer grafa spektrov pospeškov vseh meritev, ki ustrezajo kriterijem iskanja	59
Slika 65:	Primer tabelarnega prikaza vseh meritev, ki ustrezajo kriterijem iskanja.....	60
Slika 66:	V zavihku z imenom 'Map' so na zemljevidu prikazane geografske lokacije potresov in merilnih mest.....	61
Slika 67:	Primer izpisa imena in magnitude potresa, ki se nahaja pod kazalcem miške	61
Slika 68:	Primer izpisa zaporedne identifikacijske številke in imena merilnega mesta, ki se nahaja pod kazalcem miške.....	61
Slika 69:	Prikaz posamezne meritve s klikom na gumb z imenom 'Display'	61
Slika 70:	grafičen prikaz spektrov pospeškov posamezne meritve in meni, preko katerega dostopamo do dodatnih funkcij grafa.....	62
Slika 71:	Logaritmično merilo za os X in Y	62
Slika 72:	Logaritmično merilo za os X in linearno za os Y	62
Slika 73:	Linearno merilo za os X in logaritmično za os Y	62
Slika 74:	Linearno merilo za os X in Y	62
Slika 75:	Izvoz podatkov tabele v datoteko s končnico csv in prikaz treh označenih akcelrogramov v prvem stolpcu tabele, pripravljenih za prenos v program SAP2000	63
Slika 76:	Statusna vrstica programa, v kateri lahko spremljamo stanje prenosa akcelrogramov v program SAP2000	63
Slika 77:	Okno, v katerem so zbrani vsi preneseni akcelrogramu trenutnega modela	64
Slika 78:	Prikaz zavihka z imenom 'Manage Time History Cases', razdeljenega na dva osnovna modula	64
Slika 79:	Okno v katerem je mogoče določiti dodatne parametre obtežbe	65

Slika 80:	Okno za določitev parametrov dušenja	66
Slika 81:	Okno za določitev parametrov časovne integracije	66
Slika 82:	Okno za določitev dodatnih nelinearnih parametrov	66
Slika 83:	Sklop z imenom 'Time Step Data' v katerem nastavimo število in dolžino časovnih korakov	67
Slika 84:	Sklop z imenom 'Select Time History Functions', v katerem izberemo akceleroگرامe, na podlagi katerih se samodejno generirajo obtežni primeri	67
Slika 85:	Polje za vnos poljubne končnice, pripete samodejno generiranim imenom obtežnih primerov	68
Slika 86:	Okno z opozorilom o podvajanju imen obtežnih primerov	68
Slika 87:	Polje za prikaz strukturirane oblike parametrov, ki določajo izbrani obtežni primer	69
Slika 88:	Graf akceleroگرامov izbranega obtežnega primera	69
Slika 89:	Polje 'User Time Steps', v katerem lahko spremenimo število časovnih korakov izbranega obtežnega primera	69
Slika 90:	Drnsnik, s pomočjo katerega lahko spreminjamo število časovnih korakov izbranega obtežnega primera	70
Slika 91:	Grafična ponazoritev trajanja obtežnega primera glede na trajanje akceleroگرامa	70
Slika 92:	Enotski krog razdeljen na enakomerno mrežo trikotnih končnih elementov s stranico $h_0 = 0.2$, s pomočjo programa DistMesh	73
Slika 93:	Krog, razdeljen na enakomerno mrežo končnih elementov	74
Slika 94:	Kvadrat, razdeljen na neenakomerno mrežo končnih elementov	74
Slika 95:	Krog z odprtino, razdeljen na neenakomerno mrežo končnih elementov	74
Slika 96:	Kvadrat s krožno odprtino, razdeljen na neenakomerno mrežo končnih elementov	74
Slika 97:	Elipsa, razdeljena na enakomerno mrežo končnih elementov	75
Slika 98:	Poljuben poligon, razdeljen na enakomerno mrežo končnih elementov	75
Slika 99:	Sfera, razdeljena na neenakomerno mrežo končnih elementov	75
Slika 100:	Torus, razdeljen na enakomerno mrežo končnih elementov	75
Slika 101:	Komponenta 'Point XYZ' z imenom 'Pt' in njeni vhodni ter izhodni parametri	76
Slika 102:	Tri točke v prostoru izrisane v programu Rhinocerus 3D, so rezultat sheme iz Slika 101	76
Slika 103:	Shema primera parametrizacije ploskve v programu Grasshopper 3D	77
Slika 104:	Parametrizirana ploskev izrisana v programu Rhinocerus 3D	77
Slika 105:	Generirana mreža ploskovnih elementov v programu SAP2000	77
Slika 106:	Shema parametrizacije idealizirane premostitvene konstrukcije v programu Grasshopper 3D	78
Slika 107:	Izbrano referenčno stanje premostitvene konstrukcije	79
Slika 108:	Sprememba širine premostitvene konstrukcije, glede na referenčno stanje	79
Slika 109:	Sprememba višine nosilne konstrukcije, glede na referenčno stanje	79
Slika 110:	Sprememba dolžine premostitvene konstrukcije, glede na referenčno stanje	79
Slika 111:	Sprememba števila razponov nosilne konstrukcije, glede na referenčno stanje	79
Slika 112:	Sprememba dimenzij končnih elementov, glede na referenčno stanje	79
Slika 113:	Optimizirana premostitvena konstrukcija	80

»Ta stran je namenoma prazna.«

1 UVOD

Razvoj računalniške tehnologije je v zadnjih desetletjih, poleg družbenih sprememb, močno vplival predvsem na vse gospodarske panoge. Sodobni načini obdelave, shranjevanja ter izmenjave podatkov, avtomatizacija procesov, interoperabilnost, standardizacija tehnologij, splošna dostopnost, vse to so dejavniki, ki omogočajo učinkovitejši ter hitrejši delovni oziroma proizvodni proces. Poleg tega omogočajo izvajanje kompleksnejših raziskav, katerih ugotovitve podajajo nove odgovore na številna vprašanja.

V gradbeništvu je ročno načrtovanje, analizo in dimenzioniranje konstrukcij nadomestila uporaba številnih namenskih programov. Novi pristopi in uporaba programske opreme prinašajo poleg vseh prednosti, ki jih ponuja omenjena tehnologija, tudi nove omejitve:

- Celovit proces načrtovanja gradbenih konstrukcij je možen le z uporabo večjega števila namenskih programov. Posledično je potreben izvoz podatkov iz enega ter njihov uvoz v drug program. Ta postopek je lahko trivialen, nemalokrat pa predstavlja veliko izgubo časa, neekonomičnost ter predvsem možnost napak.
- Programske rešitve zelo redko omogočajo avtomatizacijo določenih delov ali procesov v celoti. Avtomatizacija, poleg prihranka časa in eliminacije človeških napak v procesu izvajanja, služi predvsem razvoju novih metod in izvajanju parametričnih študij.
- Izdelava lastne programske opreme, ki bi se osredotočila na nov pristop analize ali obdelave podatkov, hkrati pa bi bila pri svojem izvajanju odvisna od funkcij in numeričnih metod drugega programa, je praktično nemogoča. Izdelava lastnih programov, s katerimi bi morali predhodno zajeti funkcionalnost, ki je že razvita v drugih programih, je velikokrat časovno ter ekonomsko neupravičena.
- Omejena je izdelava prilagojenih grafičnih vmesnikov, ki omogočajo interakcijo z drugimi programi.
- Omejena je možnost nadgradnje in implementacije novih funkcij v obstoječe programe.

Eden izmed načinov, kako premostiti našete omejitve, se nanaša na uporabo tako imenovanih programskih vmesnikov. Gre za nabor funkcij in metod, ki predstavljajo del programa in omogočajo način za dvosmerno komunikacijo oziroma njegovo povezovanje z drugo programsko opremo, ki omogoča implementacijo teh funkcij. Redki programi namenjeni za analizo konstrukcij na ta način omogočijo dostop do svojih vgrajenih funkcionalnosti in numeričnih metod. Eden izmed takih je program SAP2000 [1].

V diplomski nalogi se bomo posvetili predvsem razvoju aplikacij na osnovi programskega vmesnika SAP2000, ki bodo pretežno koncentrirane na področje potresne analize konstrukcij. Zasnovali bomo prilagodljive grafične vmesnike za interakcijo s programom SAP2000, avtomatizirali procese

načrtovanja, analize ter obdelave podatkov, v program SAP2000 integrirali nov modul za pripravo analize časovnega odziva konstrukcij na osnovi potresnega scenarija ter prikazali primere izmenjave podatkov že obstoječe programske opreme s programom SAP2000, kar omogoča združevanje njihovih funkcionalnosti in inovativne metode analize konstrukcij. Poleg naštetega bomo stremeli k temu, da bo diplomska naloga, skupaj z razvito programsko opremo, nudila pomoč pri spoznavanju metod dela s programskim vmesnikom za SAP2000.

2 TEORETIČNE OSNOVE PROGRAMIRANJA

To poglavje je namenjeno opisu teoretičnih ozadij, ki so potrebna za razumevanje delovanja programskega vmesnika programa SAP2000, ter izdelavo aplikacij, ki temeljijo na njegovi uporabi. Na koncu poglavja so naštetih ter opisani programski jeziki in formati za izmenjavo podatkov, s pomočjo katerih smo izdelali v nadaljevanju opisane programe.

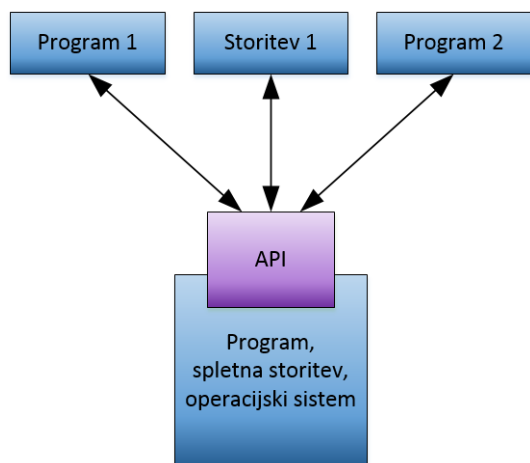
2.1 Dinamične povezovalne knjižnice (DLL)

Dinamične povezovalne knjižnice (ang. dynamic-link library) zagotavljajo način za modularno zasnovo aplikacij v okolju Microsoft Windows, katerih funkcionalnost je tako lažje posodobljena ali ponovno uporabljena. So izvedljive datoteke (angl. executable files), ki omogočajo dostop in uporabo funkcij ter podatkov drugim knjižnicam ali programom. Eno dinamično povezovalno knjižnico lahko uporablja več aplikacij hkrati, kar zmanjša potrebo po razpoložljivem spominu. Povzeto in prevedeno po [2] in [3].

Večina programov v operacijskem sistemu Microsoft Windows ima modularno zasnovu. Slednjega prav tako sestavlja kopica dinamičnih knjižnic, ki jih koristijo druge aplikacije in knjižnice. Tudi program SAP2000 sestavlja množica izvedljivih datotek, v katerih se nahajajo funkcije ter podatki, ki so dinamično, oziroma po potrebi, priklicani iz drugih modulov programa. Pri razvoju aplikacij, ki bodo med izvajanjem vzpostavile povezavo s programom SAP2000 in upravljale s funkcijami programa, bomo izkoristili dinamično povezovalno knjižnico tipa COM z imenom 'SAP2000v15.dll'. COM (angl. component object model) je standardiziran binarni vmesnik za programske komponente, ki omogoča med procesno komunikacijo in dinamično ustvarjanje objektov v različnih programskih jezikih [4]. Dostop do funkcij knjižnice je mogoč preko programskega vmesnika (angl. Application Programming Interface, API).

2.2 Programski vmesnik (API)

Programski vmesnik (angl. Application Programming Interface, API), je vmesnik, ki omogoča uporabo razredov in metod programov, operacijskih sistemov ali spletnih storitev znotraj drugega programa in zagotavlja komunikacijo med posameznimi programskimi komponentami. Po navadi API sestavlja nabor javno dostopnih funkcij. To razvijalcem omogoča izdelavo nove programske opreme, ki temelji na izmenjavi podatkov ter upravljanju z aplikacijo, katere del je API sam.



Slika 1: Shema komunikacije programskih komponent preko programskega vmesnika

Prednosti razvoja in zasnove programske opreme po shemi na sliki 1 so obojestranske. Programi, ki preko API-ja nudijo dostop do svojih funkcij, si zagotovijo širšo in boljšo uporabnost ter večjo funkcionalnost, hkrati pa vzpodbudijo razvoj nove programske opreme. Na drugi strani se tako odpirajo nove priložnosti za izdelavo aplikacij, ki lahko v celoti ali le deloma temeljijo na storitvi osnovnega programa. Na ta način lahko ponujajo širok spekter rešitev in tehnologij, ki tako uporabniku nudijo nove možnosti za uporabo. V zadnjem desetletju je, z razširjenostjo uporabe interneta in razvojem spletnih storitev, opaziti eksponentno rast te tehnologije predvsem na tem področju, vendar uporaba in implementacija programskih vmesnikov ni tuja, predvsem pa ne nova tudi pri klasičnih programih, kjer se je tehnologija pravzaprav razvila. Povzeto in prevedeno po [5] in [6].

Tudi na področju gradbeništva in arhitekture je mogoče najti programsko opremo, ki nudi dostop do svojih funkcij preko programskih vmesnikov. Eden izmed teh je program SAP2000 [1] za analizo in zasnovo konstrukcij podjetja Computers and Structures Inc. iz ZDA. Pri izdelavi aplikacij smo sicer uporabili več programskih vmesnikov, kot na primer Windows API, Bing Maps WPF API, JSON.NET API, DonNetZip Library API in druge, ki so bili potrebni za celovito funkcionalnost izdelane programske opreme.

2.3 API za program SAP2000

Programski vmesnik (API) za SAP2000 je programersko orodje, ki zagotavlja dostop in uporabo tehnologij za načrtovanje ter analizo konstrukcij s tem, da med izvajanjem programa dopušča vzpostavitev direktne povezave med drugimi programi in samim programom SAP2000. Poleg tega omogoča tudi razvoj in izdelavo vtičnikov (angl. plug-in), ki so neposredno dostopni preko obstoječega grafičnega uporabniškega vmesnika in razširijo uporabnost samega programa SAP2000. Programski vmesnik sestavlja knjižnica programske opreme, ki ponuja dostop do elementov in funkcij, s katerimi je mogoče programersko upravljati s programom in s tem zaobiti njegovo klasično upravljanje preko grafičnega vmesnika.

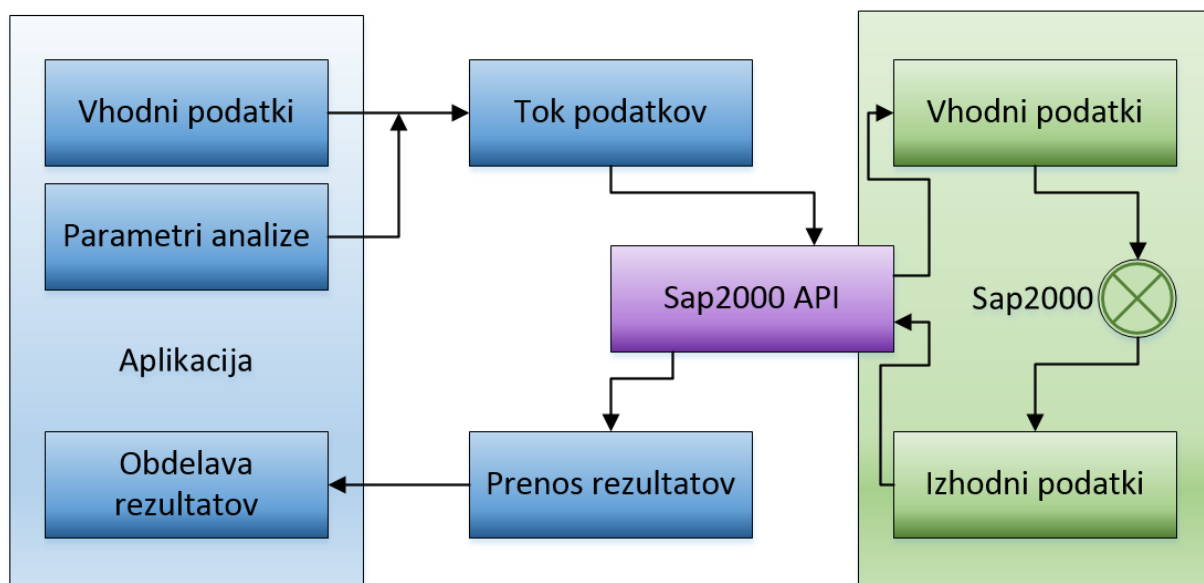
Nekaj glavnih značilnosti, ki jih ponuja programski vmesnik za SAP2000 [7]:

- Neposreden, hiter in učinkovit dostop do vseh naprednih numeričnih metod programa SAP2000.
- Neposreden dvosmerni prenos podatkov brez vmesnih datotek omogoča hitrejši pretok, ki je pomemben predvsem za velike modele.
- Povezave preko programskega vmesnika omogočajo več ciklov za izmenjavo podatkov med aplikacijami, brez ustvarjanja novega modela za vsak cikel.
- Uporaba programskega vmesnika, pri razvijanju nove programske opreme, skoraj gotovo zagotavlja njeno združljivost s prihodnjimi izdajami programa SAP2000, kar je pomembno za uporabnike.
- Povezave preko programskega vmesnika so robustne in pregledne, kar uporabnikom daje visoko stopnjo zaupanja, ki se nanaša na celovitost prenesenih informacij.
- Vsak uporabnik, ki razvija svojo lastno programsko opremo, lahko s pomočjo programskega vmesnika razvije prilagojen vmesnik za delo s programom SAP2000.
- Programski vmesnik je kompatibilen z večino širše uporabljenih programskih jezikov.

2.4 Uporaba programskega vmesnika za SAP2000

Za uporabo programskega vmesnika pri razvijanju novih aplikacij je gotovo potrebno imeti nekaj znanja programiranja, vendar je že z nekaj osnovami mogoče relativno preprosto ustvariti orodja, ki lahko skrajšajo in poenostavijo ponavljajoča se opravila in naloge. Po drugi strani pa je programski vmesnik mogoče uporabiti za razvoj zelo kompleksnih in naprednih aplikacij. V vsakem primeru programski vmesnik za SAP2000 omogoča širok nabor uporabe uveljavljenih programskih jezikov kot so na primer: Visual Basic .NET, Visual Basic for Applications (VBA), Visual C#, Visual C++, Visual Fortran, MATLAB, Python in drugi.

Uporaba aplikacij, ki temeljijo na programskem vmesniku seveda zahteva, da imamo nameščen program SAP2000 verzije 11, pri kateri je bil programski vmesnik prvič vključen ali novejšo verzijo. Vzpostavitev povezave preko programskega vmesnika se je v verziji programa SAP2000 v15.0.0 nekoliko spremenila, zato kompatibilnost s prejšnjimi verzijami programa ni zagotovljena. Sam razvoj aplikacij je odvisen predvsem od izbire programskega jezika in integriranega razvojnega okolja (ang. integrated development environment, IDE). Od te izbire je odvisna uporabljena sintaksa, ki je lastnost programskega jezika in načini, kako se preko programskega vmesnika uspešno vzpostavi povezava s programom SAP2000. Diagram na sliki 2 prikazuje tipično povezavo in tok podatkov med razvito aplikacijo in programom SAP2000 preko programskega vmesnika.



Slika 2: Tipičen tok podatkov med razvito programsko opremo in programom SAP2000 s pomočjo programskega vmesnika [8]

Programski vmesnik verzije programa SAP2000 15 vsebuje 1632 funkcij, s pomočjo katerih je mogoče zadovoljivo upravljati in nadzorovati program SAP2000, kot lahko z njim upravljamo preko grafičnega vmesnika. Funkcije programskega vmesnika so opisane v datoteki 'CSi_OAPI_Documentation.chm' [9], ki se nahaja v namestitvenem direktoriju programa SAP2000. Za vsako je podana kratka razlaga, vhodni parametri, njihovi podatkovni tipi ter primer uporabe, ki je napisan v programskem jeziku Visual Basic for Applications.

Vsaka izmed funkcij, ki sestavljajo programski vmesnik, po končanem izvajanju svojih nalog vrne spremenljivko podatkovnega tipa Long, s katero javi njihovo uspešnost realizacije. V kolikor se zadane operacije v celoti pravilno izvršijo, ima vrnjena spremenljivka vrednost 0, v nasprotnem primeru pa vrednost, ki je od 0 različna. Gre za koristno in hkrati nujno povratno informacijo, ki nam omogoča spremljanje in nadzorovanje samega procesa komunikacije med zunanjo aplikacijo in programom SAP2000.

2.5 Uporabljeni programski jeziki in formati za izmenjavo podatkov

2.5.1 Visual Basic for Applications (VBA)

Programski jezik je razvilo podjetje Microsoft za delo s programi Access, Excel, Word, Outlook in PowerPoint, ki so del paketa Microsoft Office. Namenjen je za avtomatizacijo procesov pri delu z omenjenimi programi ter nudi možnost razvijanja uporabniško določenih funkcij. Omogoča tudi naprednejše funkcije, kot sta med drugimi dostop do programskega vmesnika Windows API in uporaba dinamičnih knjižnic (angl. dynamic link library, DLL) tipa ActiveX/COM [10]. Slednje omogoča, da lahko z uporabo tega jezika preko programskega vmesnika vzpostavimo povezavo s programom SAP2000. Jezik ne spada v skupino objektno orientiranih programskih jezikov, vendar

kljub temu podpira nekaj osnovnih principov takega načina programiranja, kot je na primer delo z razredi in objekti.

Programski jezik smo uporabili pri izdelavi aplikacije za račun nihajnih časov konstrukcij in modalne analize s spektri odziva v programu Microsoft Excel, ki se preko programskega vmesnika poveže s programom za analizo konstrukcij SAP2000.

2.5.2 Visual Basic .NET

Tudi ta programski jezik je razvilo podjetje Microsoft in je v veliki meri nadomestil programski jezik Visual Basic 6 (VB6). Ustvarjen je bil predvsem z namenom, da bi omogočil koncepte objektivno orientiranega programiranja. Zasnovan je tako, da koristi prednosti, ki jih nudi programsko ogrodje .NET (angl. .NET Framework), ki ga sestavljajo programske knjižnice in programsko okolje *Common Language Runtime* (CLR). Značilnost ogrodja .NET je tudi interoperabilnost programskih jezikov. To pomeni, da je programska koda, napisana v enem programskem jeziku, razumljiva ter na voljo za uporabo tudi drugim jezikom, ki so zasnovani na ogrodju .NET, kot so: C#, F#, Windows PowerShell in drugi. Povzeto in prevedeno po [11], [12] in [13].

Programski jezik Visual Basic .NET je primeren predvsem za razvoj aplikacij v okolju Microsoft Windows. Uporabili smo ga pri izdelavi vtičnika za dinamično analizo na osnovi potresnega scenarija, ter na primeru povezave in uporabe programa Rhinoceros 3D [14] in vtičnikom Grasshopper 3D [15] s programom SAP2000.

2.5.3 Structured Query Language (SQL)

Gre za namenski programski jezik, katerega razvoj se je začel v podjetju IBM v začetku sedemdesetih let prejšnjega stoletja. Zasnovan je za delo in upravljanje s podatki shranjenimi v relacijskih podatkovnih bazah, ki so zasnovane v obliki tabel, vrstic in stolpcev. Že več kot 25 let predstavlja standard za tovrstno delo s podatki. Čeprav obstaja kar nekaj različic in nadgradenj programskega jezika, je širše jedro jezika standardizirano, kar zagotavlja veliko kompatibilnost in posledično razširjenost uporabe. Povzeto in prevedeno po [16] in [17].

Programski jezik smo uporabili pri delu z relacijskimi bazami podatkov, tako v aplikaciji za račun nihajnih časov in modalne analize s spektri odziva, kjer smo v programu Microsoft Access ustvarili bazo materialnih karakteristik, kot tudi pri izdelavi programa in namenskega programskega vmesnika [18], ki zagotavlja komunikacijo z ustvarjeno bazo podatkov o potresnem delovanju. Slednja se nahaja na MySQL [19] strežniku fakultete.

2.5.4 Hypertext Preprocessor (PHP)

PHP je skriptni programski jezik, ki se ga lahko uporablja kot splošni jezik, vendar je primarno namenjen razvoju spletnih storitev. V tem primeru, se vsa programska koda nahaja in izvrši na spletnem strežniku. Uporablja se za ustvarjanje dinamičnih spletnih strani, saj ga lahko vključimo v

iste dokumente kot jezik HTML. Čedalje pogosteje pa se njegova uporaba osredotoča na dinamično posredovanje podatkov uporabniku, ki jih ta zahteva iz oddaljenega spletnega strežnika. Povzeto in prevedeno po [20].

V slednji namen smo ga uporabili tudi v našem primeru in napisali aplikacijo ter pripadajoč programski vmesnik [18], ki glede na željo uporabnika, preko spleta vrne zahtevane podatke iz baze potresnega delovanja, ki se nahaja na strežniku fakultete.

2.5.5 MATLAB

Matlab je programski jezik istoimenskega programa za numerično analizo in obdelavo podatkov. Primarno je razvit za delo z matrikami, grafično prezentacijo podatkov in ustvarjanje algoritmov. Omogoča tudi implementacijo konceptov objektivno orientiranega programiranja in dinamičnih knjižnic tipa COM, ki zagotavljajo programsko ogrodje za uporabo binarnih programskih komponent znotraj programa. Povzeto in prevedeno po [21] in [22].

Slednje omogoča, da se preko programskega vmesnika povežemo s programom SAP2000, kar smo prikazali na primeru uporabe programa DistMesh [23] za generiranje mrež končnih elementov v okolju Matlab.

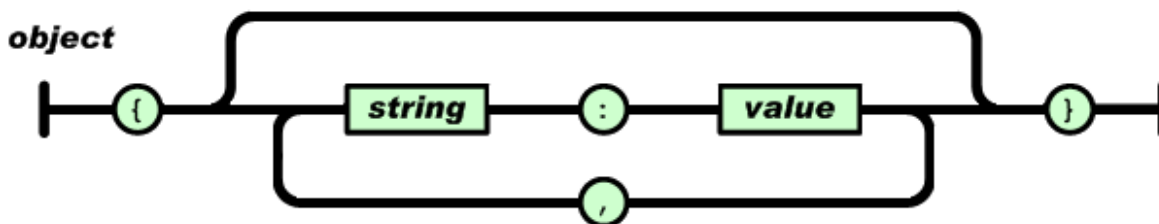
2.5.6 Extensible Application Markup Language (XAML)

Gre za deklarativni označevalni jezik, ki temelji na jeziku XML in se uporablja predvsem za razvoj naprednih uporabniških grafičnih vmesnikov. Implementiran je tako v integrirano razvojno okolje Visual Studio za razvoj aplikacij na ogrodju .NET, kot tudi v oblikovalsko orodje Microsoft Blend za ustvarjanje in načrtovanje grafičnih vmesnikov. Povzeto in prevedeno po [24].

Programski jezik smo, s pomočjo omenjenih programov, uporabili pri načrtovanju, oblikovanju in implementaciji zemljevidov ter prikazu geografskih lokacij znotraj vtičnika za dinamično analizo na osnovi potresnega scenarija.

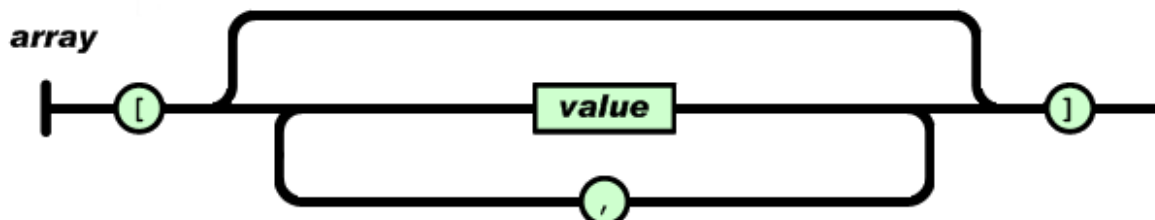
2.5.7 JavaScript Object Notation (JSON)

JSON ni programski jezik ampak standardiziran način oziroma format za izmenjavo podatkov. Izhaja iz programskega jezika JavaScript, vendar je njegova uporaba neodvisna od programskega jezika. Najpogosteje se uporablja za izmenjavo strukturiranih podatkov preko mrežnih povezav in v tem segmentu predstavlja alternativo označevalnemu jeziku XML. Odlikuje ga preprost način branja in pisanja, hkrati pa ga je mogoče enostavno strojno razčleniti ali generirati. Format temelji na dveh univerzalnih strukturah, ki sta podprti v večini programskih jezikov [25]. Prva je zbirka parov v obliki ime-vrednost, prikazana na sliki 3, ki v formatu JSON predstavlja objekt (angl. object).



Slika 3: Univerzalna struktura formata JSON, tipa objekt [25]

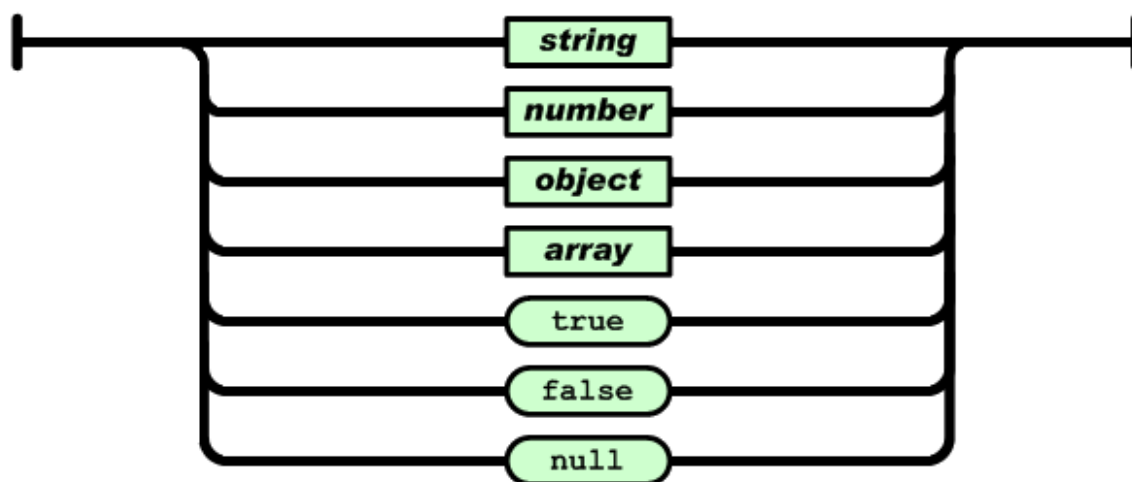
Druga univerzalna struktura je polje (angl. array) urejenih vrednosti, prikazana na sliki 4.



Slika 4: Univerzalna struktura formata JSON, tipa polje [25]

Pri obeh strukturah lahko vrednosti (angl. value), prikazane na sliki 5, predstavljajo različne podatkovne tipe ali pa predstavljajo nov objekt oziroma polje, kar omogoča gnezdenje struktur.

value



Slika 5: Podatkovni tipi in strukture, ki so lahko v formatu JSON zastopane v obliki vrednosti, [25]

Strukturiran format za izmenjavo podatkov JSON smo uporabili pri posredovanju podatkov iz baze potresnega delovanja, ki se nahaja na spletnem strežniku fakultete. Primer posredovanih podatkov v formatu JSON je prikazan v Prilogi A.

3 TEORETIČNE OSNOVE DINAMIKE GRADBENIH KONSTRUKCIJ

Izdelana programska oprema se osredotoča na potresno analizo, zato smo v tem poglavju opisali nekaj osnov dinamike gradbenih konstrukcij, ki so povzete po [26]. Te zajemajo tudi opis nekaterih numeričnih metod, ki jih pri analizi uporablja program SAP2000.

3.1 Sistemi z eno prostostno stopnjo

Sistem z eno prostostno stopnjo ima natanko eno neodvisno koordinato, ki je potrebna in zadostna za opis poljubne deformacijske lege sistema. Primer takega sistema je na primer težnostno nihalo na togi palici.

Iz dinamičnega ravnotežja vseh sil, ki delujejo na maso sistema ter ob upoštevanju teorije prvega reda, se gibanje linearno elastičnega sistema z eno prostostno stopnjo opiše z nehomogeno diferencialno enačbo drugega reda:

$$m\ddot{u} + c\dot{u} + ku = f(t) \quad (1)$$

kjer je

m – masa

c – koeficient viskoznega dušenja

k – togost sistema

\ddot{u} – pospešek

\dot{u} – hitrost

u – pomik

$f(t)$ – zunanja obtežba

Pri potresni obtežbi, kjer pride do dinamičnega odziva konstrukcije zaradi gibanja podpor in ne zaradi dinamične obtežbe, enačbo (1) zapišemo v naslednji obliki

$$m\ddot{u}_r + c\dot{u}_r + ku_r = -m\ddot{u}_p \quad (2)$$

Pri tem označujemo z indeksom r relativni pomik mase, ki je enak razliki med absolutnim pomikom in pomikom podpore, ki ga označujemo z indeksom p. V kolikor pomike izrazimo glede na statično ravnovesno lego, težnostne sile na nihanje nimajo vpliva.

3.1.1 Lastno nihanje

Lastno nihanje je posledica začetnega odmika sistema iz ravnovesne lege ali začetne hitrosti sistema. Zunanje obtežbe ni. V kolikor zanemarimo dušenje ($c = 0$), obravnavamo lastno nedušeno nihanje. Enačbo gibanja tako zapišemo kot:

$$m\ddot{u} + ku = 0 \quad (3)$$

Rešitev enačbe (3) se glasi:

$$u(t) = a \sin(\omega t - \theta) \quad (4)$$

kjer je

a – amplituda nihanja

ω – krožna frekvenca nihanja

θ – fazni zamik

Ker je nihanje nedušeno, sistem ves čas niha z enako amplitudo in frekvenco. Nihajni čas sistema izračunamo z enačbo (5):

$$T = \frac{2\pi}{\omega} \quad (5)$$

Če pri lastnem nihanju upoštevamo dušenje sistema, enačbo (3) napišemo v obliki:

$$m\ddot{u} + c\dot{u} + ku = 0 \quad (6)$$

Rešitev enačbe (6) se glasi:

$$u(t) = a e^{-\xi \omega t} \sin(\omega_D t - \theta) \quad (7)$$

kjer je

a – amplituda nihanja

θ – fazni zamik

ω – krožna frekvenca nedušenega nihanja

ω_D – krožna frekvenca dušenega nihanja

ξ – koeficient kritičnega dušenja, ki je definiran kot razmerje med koeficientom dušenja c in koeficientom dušenja pri kritičnem dušenju c_{cr}

Krožna frekvenca lastnega dušenega nihanja ω_D okoli ravnovesne lege je konstantna, amplituda pa se s časom zmanjšuje. Hitrost padanja amplitude oziroma velikost odmkov iz ravnovesne lege, je pogojena z velikostjo dušenja. Večje kot je dušenje, več energije sistema je potrebne za premagovanje sile dušenja in posledično je na razpolago manj mehanske energije. Če je dušenje sistema tako veliko, da do nihanja okoli ravnovesne lege sploh ne pride, govorimo o kritičnem dušenju. Koeficient kritičnega dušenja je v tem primeru enak $\xi = 1$.

Odstotek kritičnega dušenja gradbenih konstrukcij se giblje od 2% za varjene jeklene in prednapete betonske konstrukcije, vse do 15% za žebeljane lesene konstrukcije.

3.1.2 Vsiljeno nihanje

Vsiljeno nihanje sistema z eno prostostno stopnjo opišemo z enačbo (1). Pri tem predstavlja člen $f(t)$, na desni strani enačaja zunanjo obtežbo oziroma nadomestno obtežbo, če je nihanje sistema posledica gibanja podpor. Slednje je primer potresne obtežbe.

Splošno rešitev nehomogene diferencialne enačbe zapišemo kot vsoto homogene in partikularne rešitve. Homogeni del predstavlja rešitev lastnega nihanja, partikularna rešitev pa je odvisna od funkcije obtežbe $f(t)$. Za enostavne primere obtežbe, kot so na primer konstantna, harmonična ali odsekoma linearna udarna obtežba, lahko izpeljemo analitične rešitve. V nasprotnem primeru se poslužujemo numeričnih metod, ki so tako sestavni del potresne analize.

3.1.3 Duhamelov integral

Celoten odziv linearnih elastičnih sistemov na poljubno obtežbo lahko, z upoštevanjem zakona o superpoziciji, izrazimo kot vsoto odzivov infinitezimalno kratkih sunkov. To storimo z uporabo Duhamelovega integrala. Ob upoštevanju dušenja sistema in predpostavki, da sta začetni pomik in hitrost sistema v začetku delovanja obtežbe enaka nič, se enačba glasi:

$$u(t) = \frac{1}{m \omega_D} \int_0^t f(\tau) e^{-\xi \omega(t-\tau)} \sin \omega_D(t-\tau) d\tau \quad (8)$$

kjer je

t – čas v katerem računamo odziv konstrukcije

τ – čas delovanja infinitezimalnega sunka

Enačba ima analitično rešitev le za enostavne primere, kjer je zunanja obtežba podana v analitični obliki in je integral integrabilen. Pri odzivu sistemov na potresno obtežbo omenjena pogoja nista izpolnjena, zato rešitev integrala določimo s pomočjo numeričnih postopkov. Zadostno natančnost dosežemo z manjšanjem časovnega koraka, ki naj ne bo večji od desetine nihajnega časa sistema.

3.1.4 Integracija korak za korakom

Gre za povsem splošno metodo numeričnega računa dinamičnega odziva sistema, ki omogoča tudi račun v nelinearnem območju. Odziv sistema računamo v kratkih časovnih intervalih Δt , kjer je pogoj dinamičnega ravnotežja izpolnjen zgolj na začetku in koncu posameznega intervala. Vmes predpostavimo konstanten ali linearen potek pospeškov. Do celotnega odziva pridemo po metodi korak za korakom, kjer količine na koncu intervala i predstavljajo začetne pogoje na intervalu $i + 1$. Z manjšanjem dolžine intervalov lahko dosežemo poljubno natančnost, če je metoda stabilna.

Pomik na koncu intervala se ob predpostavki konstantnih pospeškov na območju posameznega intervala za linearne sisteme glasi:

$$\left(\frac{4}{\Delta t^2} m + \frac{2}{\Delta t} c + k \right) u_k = f_k + \left(\frac{4}{\Delta t^2} + \frac{4}{\Delta t} \dot{u}_z + \ddot{u}_z \right) m + \left(\frac{2}{\Delta t} u_z + \dot{u}_z \right) c \quad (9)$$

kjer je

u_z – pomik na začetku intervala

u_k – pomik na koncu intervala

Enačbo (9) lahko zapišemo v obliki:

$$\bar{k} u_k = \bar{f} \quad (10)$$

kjer je

\bar{k} – nadomestna togost

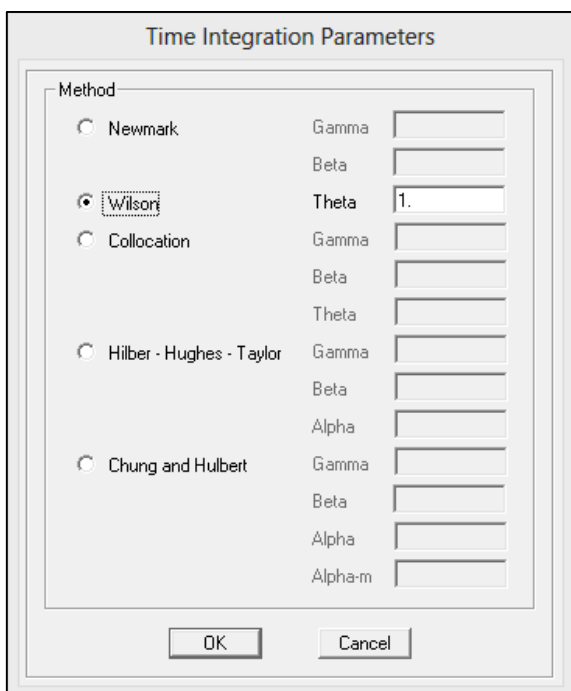
\bar{f} – nadomestna obtežba

Ker smo na vsakem intervalu predpostavili potek pospeškov ter s tem tudi potek hitrosti in pomikov, tako v vsakem koraku dinamični problem prevedemo na statičnega. V primeru nelinearne analize je enačba namesto s pomiki, izražena s spremembami pomikov.

V programu SAP2000 je na voljo več metod za integracijo diferencialne enačbe nihanja in sicer metode: Newmark, Wilson, Collocation, Hiber-Huges-Taylor in Chung and Hubert, kar je razvidno iz slike 6. Pri Wilsonovi θ metodi, predpostavimo linearen potek pospeškov po razširjenem časovnem intervalu Δt_r , katerega dolžino izračunamo po enačbi (11):

$$\Delta t_r = \theta \Delta t \quad (11)$$

Z upoštevanjem začetnih pogojev na začetku intervala, najprej izračunamo pomik, hitrost in pospešek na koncu razširjenega intervala. Nato s pomočjo linearne interpolacije določimo vrednosti na koncu dejanskega intervala, ki hkrati predstavljajo začetne vrednosti naslednjega razširjenega intervala. Stabilnost opisane metode dosežemo z vrednostjo $\theta > 1,37$. V primeru, da je $\theta = 1$, dobimo običajno metodo z linearnim pospeškom.



Slika 6: Metode za integracijo diferencialne enačbe nihanja, ki so na voljo v programu SAP2000

3.1.5 Spektri odziva

Pri projektiranju konstrukcij nas po navadi namesto celotnega časovnega poteka odziva zanima zgolj maksimalni odziv konstrukcije. V ta namen obstajajo diagrami, tako imenovani spektri odziva, iz katerih lahko za sisteme z eno prostostno stopnjo odčitamo maksimalni odziv pri različnih dinamičnih obtežbah.

»Spekter odziva prikazuje maksimalne vrednosti odziva (pomike, hitrosti, pospeške, notranje sile, napetosti ali dinamični faktor) za konstrukcije z eno prostostno stopnjo pri določeni obtežbi« [27].

Spekter odziva izdelamo tako, da izračunamo odziv konstrukcije z eno prostostno stopnjo z lastno frekvenco ω in koeficientom dušenja ξ za podan akcelrogram. Maksimalen odziv predstavlja eno samo točko v spektru pomikov. Če izračun ponavljamo za različne lastne nihajne čase in koeficiente dušenja, lahko izrišemo celoten spekter odziva. Tak spekter velja le za konkreten akcelrogram oziroma nadomestno obtežbo, ki je enaka produktu mase konstrukcije in pospeškov temeljnih tal.

3.2 Sistemi z več prostostnimi stopnjami

Gibanje konstrukcije z več prostostnimi stopnjami opisuje sistem diferencialnih enačb, ki ga je možno izpeljati na različne načine. Iz pogoja o dinamičnem ravnotežju je mogoče izpeljati tak sistem le za enostavnejše konstrukcije, za poljubne konstrukcije pa ga lahko izpeljemo s pomočjo principa o virtualnem delu ali s pomočjo Hemiltonovega principa in Lagrangevih enačb.

Sistem enačb, ki opisuje gibanje sistema z več prostostnimi stopnjami, izpeljan s pomočjo pogoja o dinamičnem ravnotežju:

$$[M]\{\ddot{U}\} + [C]\{\dot{U}\} + [K]\{U\} = \{F(t)\} \quad (12)$$

kjer je

- $[M]$ – masna matrika, ki jo sestavljajo koeficienti m_{ij} , ki predstavljajo vztrajnostno silo v točki i zaradi enotskega pospeška v točki j . Pri tem so vsi pospeški v ostalih točkah enaki nič.
- $[C]$ – matrika dušenja, ki jo sestavljajo koeficienti c_{ij} , ki predstavljajo silo v točki i zaradi enotske hitrosti v točki j . Pri tem so vse hitrosti v ostalih točkah enake nič.
- $[K]$ – togostna matrika, ki jo sestavljajo koeficienti k_{ij} , ki predstavljajo silo v točki i zaradi enotskega pomika v točki j . Pri tem so vsi pomiki v ostalih točkah enaki nič.
- $\{\ddot{U}\}$ – vektor pospeškov
- $\{\dot{U}\}$ – vektor hitrosti
- $\{U\}$ – vektor pomikov
- $\{F(t)\}$ – vektor zunanje obtežbe

3.2.1 Lastno nihanje

Pri določanju lastnih frekvenc in lastnih nihajnih oblik gradbenih konstrukcij ne naredimo velike napake, če zanemarimo vpliv dušenja. Kot smo že omenili, je lastno nihanje posledica začetnega odmika iz ravnovesne lege ali začetne hitrosti sistema. Če zanemarimo vpliv dušenja, lahko enačbo, ki opisuje lastno nihanje sistemov z več prostostnimi stopnjami, zapišemo v obliki:

$$[M]\{\ddot{U}\} + [K]\{U\} = \{0\} \quad (13)$$

Rešitev enačbe (13) se glasi:

$$\{u\} = \{\Phi\} \sin(\omega t - \theta) \quad (14)$$

kjer je

$\{\Phi\}$ – vektor amplitud nihanja posameznih točk

Prvi odvod enačbe (14) po času t :

$$\{\dot{u}\} = \omega \{\Phi\} \cos(\omega t - \theta) \quad (15)$$

Drugi odvod enačbe (14) po času t :

$$\{\ddot{u}\} = -\omega^2 \{\Phi\} \sin(\omega t - \theta) \quad (16)$$

Enačbo (16) in (14) vstavimo v enačbo (13) in dobimo:

$$([K] - \omega^2 [M])\{\Phi\} = \{0\} \quad (17)$$

Enačba (17) predstavlja splošen problem lastnih vrednosti. V kolikor je determinanta sistema homogenih algebraičnih enačb enaka nič, ima tak sistem ne-trivialno rešitev. Z razvojem determinante dobimo karakteristični polinom n -tega reda za sistem z n prostostnimi stopnjami. Lastne vrednosti tako predstavljajo členi ω_n^2 , ki fizikalno odražajo kvadrate lastnih frekvenc, lastne vektorje pa predstavljajo členi $\{\Phi_n\}$, ki v fizikalnem smislu označujejo možne načine nihanja sistema. Sistem lahko niha na toliko različnih načinov, kolikor ima prostostnih stopenj. Vsakemu izmed teh načinov pripada lastna frekvenca. Absolutnih vrednosti amplitud nihanja ni mogoče določiti, možno pa je poiskati njihova medsebojna razmerja.

3.2.2 Vsiljeno nihanje

Vsiljeno nihanje sistema z več prostostnimi stopnjami opišemo z enačbo (12). Za reševanje tega sistema enačb se uporabljata predvsem dve metodi in sicer modalna analiza, ki je primerna zgolj za elastično analizo, saj temelji na principu superpozicije ter numerična metoda direktne integracije, ki je primerna tudi za nelinearno analizo.

Modalna analiza

Sistem enačb (12), predstavlja sistem odvisnih enačb. V kolikor izvedemo transformacijo sistema na način, da so vse matrike diagonalne, postanejo enačbe medsebojno neodvisne in jih je mogoče reševati s pomočjo postopkov in metod za sisteme z eno prostostno stopnjo.

Do omenjene transformacije pridemo tako, da uvedemo nov vektor pomikov $\{Y\}$:

$$\{U\} = [\Phi]\{Y\} \quad (18)$$

kjer je

$\{Y\}$ – vektor posplošenih pomikov

$[\Phi]$ – matrika lastnih nihajnih oblik nedušenega sistema

Dejanski pomiki so tako izraženi kot linearna kombinacija lastnih nihajnih oblik nedušenega sistema. Enačba (18) predstavlja transformacijo med n-dimenzionalnima koordinatnima sistemoma, ki jo lahko zapišemo tudi v obliki:

$$\{U\} = \sum_{i=1}^n \{\Phi_i\} Y_i \quad (19)$$

kjer je

Y_i – amplituda posameznih lastnih nihajnih oblik

$\{\Phi_i\}$ – vektor posamezne lastne nihajne oblike

Ker ima na pomike konstrukcije v splošnem pomemben vpliv le nekaj nihajnih oblik, nam metoda omogoča hiter in dovolj natančen izračun že z upoštevanjem $m \ll n$ najnižjih nihajnih oblik. Pri tem je m število nihajnih oblik, ki pomembno vplivajo na pomike konstrukcije, n pa število vseh nihajnih oblik.

Če torej v sistem enačb (12) vstavimo enačbo (18) in njene odvode, dobimo:

$$[\Phi]^T [M] \Phi \{\ddot{Y}\} + [\Phi]^T [C] \Phi \{\dot{Y}\} + [\Phi]^T [K] \Phi \{Y\} = [\Phi]^T \{F(t)\} \quad (20)$$

Oziroma:

$$[M]\{\ddot{Y}\} + [C]\{\dot{Y}\} + [K]\{Y\} = \{F(t)\} \quad (21)$$

kjer je:

$$[M] = [\Phi]^T [M] [\Phi]$$

$$[C] = [\Phi]^T [C] [\Phi]$$

$$[K] = [\Phi]^T [K] [\Phi]$$

$$\{F\} = [\Phi]^T \{F\}$$

Enačba (21) predstavlja sistem neodvisnih diferencialnih enačb, kjer vsaka enačba ustreza eni nihajni obliki. Vsako od njih lahko zapišemo v obliki:

$$M_i \ddot{Y}_i + C_i \dot{Y}_i + K_i Y_i = F_i(t) \quad (22)$$

Enačba (22) predstavlja enačbo nihanja sistema z eno prostostno stopnjo, ki smo ga opisali v poglavju 3.1. Rešimo jo lahko s katero koli od opisanih metod za vse nihajne oblike, ki pomembno vplivajo na odziv sistema.

V primeru potresne obtežbe, kjer obtežbo predstavlja gibanje podpor, ter ob upoštevanju vzbujanja konstrukcije v eni smeri, zapišemo sistem enačb (12) v obliki:

$$[M]\{\ddot{U}_r\} + [C]\{\dot{U}_r\} + [K]\{U_r\} = -\ddot{u}_p[M]\{s\} \quad (23)$$

Če za reševanje sistema uporabimo modalno analizo, dobimo po transformaciji v glavne koordinate za vsako nihajno obliko enačbo (24):

$$\ddot{Y}_i + 2\xi_i\omega_i\dot{Y}_i + \omega_i^2 Y_i = \frac{-\ddot{u}_p(t)\{\Phi_i\}^T[M]\{s\}}{M_i} \quad (24)$$

Pomik v glavnih koordinatah se z reševanjem enačbe (24) s pomočjo Duhamelovega integrala glasi:

$$Y_i(t) = \frac{\{\Phi_i\}^T[M]\{s\}}{\omega_i M_i} \int_0^t \ddot{u}_t(\tau) e^{-\xi_i \omega_i(t-\tau)} \sin \omega_i(t-\tau) d\tau = \frac{L_i}{M_i} \frac{D_i(t)}{\omega_i} = \Gamma_i \frac{D_i(t)}{\omega_i} \quad (25)$$

kjer je

$\{s\}$ – vektor vplivnih koeficientov, ki predstavljajo premike toge konstrukcije pri premikih temeljnih tal za enoto v obravnavani smeri.

D_i – integral razviden iz enačbe (25)

Γ_i – faktor participacije, definiran z enačbo (26), ki služi za oceno vpliva posamezne nihajne oblike na celotne pomike

$$\Gamma_i = \frac{L_i}{M_i} = \frac{\{\Phi_i\}^T[M]\{s\}}{\{\Phi_i\}^T[M]\{\Phi_i\}} \quad (26)$$

Pomike v osnovnem koordinatnem sistemu nato izračunamo z linearno kombinacijo:

$$\{U(t)\} = [\Phi]\{Y(t)\} = \sum_{i=1}^m \{\Phi_i\} Y_i(t) = \sum_{i=1}^m \{\Phi_i\} \Gamma_i \frac{D_i(t)}{\omega_i} \quad (27)$$

Notranje sile pa dobimo z enačbo:

$$\{F_E(t)\} = [K]\{U(t)\} = [K][\Phi]\{Y(t)\} = [K] \sum_{i=1}^m \{\Phi_i\} Y_i(t) \quad (28)$$

Pri konstrukcijah z več prostostnimi stopnjami se maksimalne notranje sile v splošnem ne pojavijo pri maksimalnih pomikih. V praksi se tako uporabljajo približne metode, s katerimi ocenimo maksimalne vrednosti pomikov in notranjih sil. Za vsako od nihajnih oblik, ki pomembno prispeva k odzivu sistema, rešimo enačbo (24) in poiščemo maksimum funkcije $Y(t)$. Če rešujemo enačbo s pomočjo danih spektrov odziva, lahko približen maksimalen odziv določimo na sledeč način.

Maksimalno vrednost pomika v glavnih koordinatah določimo z enačbo (28):

$$Y_{i \max} = F_i S_d(\omega_i, \xi_i) \quad (28)$$

kjer je

$S_d(\omega_i, \xi_i)$ – vrednost odziva v spektru pomikov S_d , odvisna od lastne frekvence in faktorja kritičnega dušenja

Maksimalne pomike v osnovnem koordinatnem sistemu dobimo z enačbo (29), notranje sile pa z enačbo (30):

$$\{U_i\}_{\max} = \{\Phi_i\} F_i S_d(\omega_i, \xi_i) \quad (29)$$

$$\{F_{Ei}\}_{\max} = [M] \{\Phi_i\} F_i S_a(\omega_i, \xi_i) \quad (30)$$

kjer je

$S_a(\omega_i, \xi_i)$ – vrednost odziva v spektru pospeškov S_a , odvisna od lastne frekvence in faktorja kritičnega dušenja

S transformacijo v osnovni koordinatni sistem dobimo maksimalni pomik in pripadajoče notranje sile za posamezno nihajno obliko. Rezultanto vrednosti dobimo z eno izmed metod za kombiniranje vrednosti nihajnih oblik.

Metoda geometrijske vsote oziroma metoda SRSS (angl. Square Root of the Sum of the Squares) je v praksi najbolj pogosta in temelji na predpostavkah, da so nihajni časi kratki v primerjavi s trajanjem obtežbe, da je spekter frekvenc, ki nastopajo v obtežbi široko zastopan in da so v posameznih nihajnih oblikah nihanja statično neodvisna. Pomike in notranje statične količine po tej metodi kombiniramo z enačbami:

$$\{U\}_{\max} = \sqrt{\sum_{i=1}^m \{U_i\}_{\max}^2} \quad (31)$$

$$\{F_E\}_{\max} = \sqrt{\sum_{i=1}^m \{F_{Ei}\}_{\max}^2} \quad (32)$$

Metoda popolne kvadrature kombinacije oziroma metoda CQC (angl. Complete Quadratic Combination) vključuje vse medsebojno možne produkte komponent vektorja, ki ustrezajo posameznim nihajnim oblikam. Primerna je za kombinacijo vrednosti konstrukcij, pri katerih imajo lastne frekvence ali zgolj nekatere od njih podobne vrednosti. S tem, nihanja v posameznih nihajnih oblikah ne izpolnjujejo pogoja o statični neodvisnosti. Posamezne komponente vektorjev pomikov in notranjih statičnih količin po tej metodi kombiniramo z enačbama (33) in (34):

$$u_{k \max} = \sqrt{\sum_{i=1}^m \sum_{j=1}^m \rho_{ij} u_{ik} u_{jk}} \quad (33)$$

$$f_{Ek \max} = \sqrt{\sum_{i=1}^m \sum_{j=1}^m \rho_{ij} f_{Eik} f_{Ejk}} \quad (34)$$

kjer je

$u_{k \max}$ – komponenta vektorja $\{U_{\max}\}$

u_{ik} – komponenta vektorja $\{U_i\}$

$f_{Ek \max}$ – komponenta vektorja $\{F_{E \max}\}$

f_{Eik} – komponenta vektorja $\{F_{Ei}\}$

ρ_{ij} – faktorji, ki definirajo sklopljenost posameznih nihajnih oblik ($0 \leq \rho_{ij} \leq 1$)

Obe metodi za kombinacijo pomikov in notranjih sil sta na voljo tudi v programu SAP2000, kar je razvidno iz slike 7.

Load Case Data - Response Spectrum

Load Case Name: Set Def Name

Notes:

Load Case Type: Design...

Modal Combination

☒ CQC ☐ SRSS ☐ Absolute ☐ GMC ☐ NRC 10 Percent ☐ Double Sum

GMC f1: GMC f2: Periodic + Rigid Type:

Directional Combination

☒ SRSS ☐ CQC3 ☐ Absolute

Scale Factor:

Modal Load Case

Use Modes from this Modal Load Case:

Loads Applied

Load Type	Load Name	Function	Scale Factor
Accel	U1	UNIFRS	1

☐ Show Advanced Load Parameters

Other Parameters

Modal Damping:

Slika 7: Pogovorno okno programa SAP2000, iz katerega so razvidne možne metode za kombinacijo pomikov in notranjih sil

4 PROGRAM ZA RAČUN NIHAJNIH ČASOV IN MODALNE ANALIZE S SPEKTRI

4.1 Motivacija za izdelavo programa

Glede na dejstva, da pred izdelavo diplomske naloge nismo imeli še nobenih resnejših izkušenj z uporabo programskega vmesnika za SAP2000, da na to temo ne obstaja nobena ustrezna literatura in prav tako ni odprto dostopnih programov, z izjemo nekaj predstavitvenih primerov [28], [29] in [30], je bil glavni cilj diplomske naloge raziskati uporabnost programskega vmesnika za SAP2000 in ugotoviti njegove morebitne omejitve ter prednosti. Skozi ta proces raziskovanja, ki ga je nemalokrat spremljala vrsta poizkusov, variant in različnih pristopov, smo začeli dobivati predstavo, kaj nam sam programski vmesnik pravzaprav omogoča. Začele so se pojavljati prve ideje, za katere pa še vedno nismo bili docela prepričani, da so izvedljive. Zadalimo si cilj, da z uporabo programskega vmesnika ustvarimo lasten vmesnik za interakcijo s programom SAP2000, preko katerega bo mogoče s čim manjšim številom vhodnih podatkov v programu SAP2000 generirati novo konstrukcijo, izvesti analizo ter dobljene rezultate izvoziti, ter jih do določene mere naknadno obdelati. Tako je nastal program za določitev nihajnih časov konstrukcij, ki smo mu kasneje dodali še možnost modalne analize s spektri odziva.

Posredni a hkrati nezanimljiv namen je bil izdelati odprto dostopen program, katerega programska koda je javno dostopna in lahko služil kot učni primer za vsakogar, ki se bo želel naučiti ali poglobiti znanje uporabe programskega vmesnika za SAP2000. Trudili smo se, da je le ta ustrezno opremljena s komentarji, ki lahko zelo olajšajo njeno razumevanje ter omogočajo lažje sledenje logičnim procesom. Program se tako lahko brez kakršnih koli zadržkov spremeni, nadgradi ali pa se le del programske kode uporabi v drugih rešitvah. Tudi to predstavlja enega izmed načinov, da bi spodbudili širšo uporabo programskega vmesnika in iskanje rešitev, ki lahko razširijo osnovne funkcionalnosti programa SAP2000.

4.2 Opis programa

Program uporabniku omogoča, da z relativno majhnim številom vhodnih podatkov določi vse parametre, ki so potrebni za generiranje ravninskih ali prostorskih okvirnih konstrukcij, ter parametre za izvedbo modalne analize s spektri pospeškov. Ti podatki se nato samodejno, preko programskega vmesnika, prenesejo v SAP2000, kjer se izvede sama analiza podane konstrukcije. Po končani analizi se po istem principu prenesejo rezultati nazaj v naš program, kjer jih do določene mere obdelamo skladno s standardi Evrokod.

Naknadno smo zasnovali še vmesnik, ki omogoča modifikacije osnovne okvirne konstrukcije kot so: odstranitev posameznih delov oziroma elementov konstrukcije, dodajanje povezij, rotacijo lokalnih osi posameznih prereзов ter dodajanje sten. Na ta način lahko hitro in enostavno pripravimo veliko variacij osnovne okvirne konstrukcije.

4.2.1 Izdelava programa

Pri izdelavi tega programa smo se odločili, da ga bomo napisali v programskem jeziku Visual Basic for Application (v nadeljevanju VBA), ki je pravzaprav del programskega paketa Microsoft Office. K tej odločitvi je prispevalo dejstvo, da gre za programsko opremo, ki je danes med najbolj razširjenimi in predstavlja osnovo za vsakršno inženirsko delo. Kot smo že omenili, služi program tudi v učne namene, zato z našo izbiro razširimo potencialno skupino uporabnikov, hkrati pa velja VBA za enega izmed lažjih in uporabniku prijaznejših programskih jezikov. Z nekaj osnovnega znanja programiranja bo tako naša rešitev zagotovo skoraj v celoti razumljiva vsakomur, ki ga programiranje vsaj malo zanima.

Programska koda je torej napisana znotraj programa Microsoft Excel, ki služi tudi kot vmesnik za podajanje in prikaz rezultatov izvršene analize. Pri izdelavi smo uporabili še program Microsoft Access, v katerem smo ustvarili manjšo bazo materialov oziroma njihovih karakteristik.

4.2.2 Opis parametrov in algoritmov

Geometrija

Geometrija ravninskega okvira je definirana s štirimi parametri in sicer: s številom razponov v smeri X, z dimenzijo razpona v smeri X, s številom etaž in z višino etaže. Za definicijo prostorske okvirne konstrukcije moramo k parametrom, ki definirajo ravninskega, dodati še število razponov v smeri Y in dimenzijo razpona v smeri Y.

Smer X in Y sta medsebojno pravokotni. Skupaj z osjo Z, ki je usmerjena v nasprotni smeri delovanja gravitacijske sile, tvorijo globalni kartezični koordinatni sistem. Njegovo izhodišče je vedno na koti terena ter presečišču simetral prostorskega okvira oziroma presečišču simetrane in ravninskega okvira.

Položaj izhodišča globalnega koordinatnega sistema glede na konstrukcijo določimo z enačbami:

$$X = \frac{n_x x}{2} \quad (35)$$

$$Y = \frac{n_y y}{2} \quad (36)$$

$$Z = 0 \quad (37)$$

n_x – število razponov v smeri X

x – dimenzija enega razpona v smeri X

n_y – število razponov v smeri Y

y – dimenzija enega razpona v smer Y

Okvir tvorijo linijski končni elementi, ki vzporedno s smerjo Z predstavljajo stebre, v ravnini X-Y pa povezovalne grede konstrukcije. Vse povezave med posameznimi elementi so modelirane kot toge. Izjema je zgolj vpetost konstrukcije v temelje, katero je mogoče modelirati kot togo ali idealno vrtljivo podprto s sproščenimi vsemi tremi zasuki R_x , R_y in R_z .

Material

Da bi zmanjšali število materialnih parametrov, ki jih mora podati uporabnik, smo ustvarili manjšo bazo materialov oziroma njihovih karakteristik. Tako lahko zgolj izberemo ime želenega materiala, njegove lastnosti pa se bodo samodejno prenesle iz baze podatkov, ter nato preko programskega vmesnika v program SAP2000. Trenutno je v bazi definiranih 18 materialov, od tega 14 betonov s trdnostnimi razredi od C12/15 vse do C 90/105 in 4 konstrukcijska jekla razredov S235, S275, S355 in S450.

Prednost tega je tudi, da lahko uporabnik po svojih željah in potrebah v priloženi bazi spreminja obstoječe lastnosti materialov ali dodaja nove, le ti pa bodo nato avtomatsko na izbiro znotraj našega programa. Baza materialov se nahaja v priloženi datoteki z imenom 'SAP_Material.accdb'.

Za našo analizo, kjer smo se omejili na račun nihajnih časov konstrukcije in na račun modalne analize s spektri pospeškov, potrebujemo za vsak material zgolj dva parametra. Elastični modul je v elastičnem območju, kjer velja Hookov zakon, definiran kot razmerje med napetostjo in spremembo dolžine, Poissonovo število pa podaja razmerje med relativnimi prečnimi in vzdolžnimi deformacijami. Strižnega modula nam ni potrebno podajati posebej, saj se v programu SAP2000 izračuna samodejno na podlagi enačbe (38):

$$G = \frac{E}{2(1 + \mu)} \quad (38)$$

kjer je

E – elastični modul [kN/m²]

μ – Poissonovo število [–]

G – strižni modul [kN/m²]

Programski vmesnik SAP2000 je pri definiranju materialov in podajanju njihovih lastnosti zasnovan tako, da je potrebno vsak material opredeliti v enega izmed naslednjih razredov:

- 1 – jeklo
- 2 – beton
- 3 – brez
- 4 – aluminij
- 5 – hladno valjano jeklo
- 6 – armatura

7 – tendon (material oziroma objekt, s pomočjo katerega simuliramo efekte prednapenjanja in naknadnega napenjanja [31])

Materiali v naši bazi so razvrščeni v prvi oziroma drugi razred, na podlagi katerega program SAP2000 samodejno definira prostorninsko težo. Za jeklo privzame vrednost $76,97 \text{ kN/m}^3$, za beton pa $23,56 \text{ kN/m}^3$. Vrednosti bi lahko naknadno z uporabo ustrezne funkcije programskega vmesnika spremenili, vendar bomo v tem primeru uporabili pred-nastavljene vrednosti.

Prečni prerezi

Do sedaj smo omenili, da je konstrukcija modelirana z linijskimi elementi, ki predstavljajo stebre in grede okvira. Kasneje bomo pokazali, kako lahko konstrukciji v posameznih poljih dodamo diagonalna povezja ali stene. Za vse našteje tipe elementov je potrebno definirati prečne prereze, ki so neposredno odvisni od dveh parametrov in sicer geometrije ter materiala. Tako kot smo parametre, ki definirajo posamezen material združili v njegovo ime, tudi tu lahko storimo enako s to razliko, da je baza geometrijskih karakteristik tipičnih prerezov že pripravljena v programu SAP2000.

Za linijske elemente smo tako v našem programu omogočili izbiro prerezov iz baze programa SAP2000, v kateri so zbrani tipični prerezi, ki se uporabljajo v Evropi. Razdelili smo jih v podskupine in sicer prerezi tipa H (120 prerezov), tipa I (75 prerezov), tipa C (33 prerezov), tipa T (450 prerezov), tipa Kotnik (154 prerezov), tipa Dvojni kotnik (2354 prerezov), tipa Dvojni C (609 prerezov), tipa Cev (23 prerezov) in tipa Škatlast (567 prerezov). Poleg omenjenih, lahko definiramo tudi poljubne pravokotne in okrogle prereze.

Stene smo modelirali s ploskovnimi končnimi elementi tipa Shell. Za določitev njihovih geometrijskih karakteristik prečnih prerezov potrebujemo zgolj podatek o debelini.

Pri določanju prečnih prerezov omenjenih štirih tipov elementov, poleg geometrijskih lastnosti, izberemo še enega izmed materialov. V programu smo omogočili tudi rotacijo lokalnih osi elementov, kar bomo pokazali v nadaljevanju.

Masa konstrukcije

Program SAP2000 samodejno izračuna maso, katere vir so lastne teže stebrov, gred, diagonal in sten. To maso nato porazdeli v vozlišča končnih elementov [32].

Dodatno maso konstrukcije, ki je posledica stalne in spremenljive obtežbe, v katerih mora biti zajeta tudi lastna teža medetažnih plošč in ostalih nekonstrukcijskih elementov, poda uporabnik v obliki enakomerno porazdeljene ploskovne obtežbe. S pomočjo enačbe (39) in enačbe (40) izračunamo maso m ter masni vztrajnostni moment J vsake etaže.

$$m = \frac{p A}{g} \quad (39)$$

$$J = m \frac{I_x + I_y}{A} \quad (40)$$

kjer je

m – masa etaže [t]

J – masni vztrajnostni moment etaže okoli osi Z [t m²]

p – enakomerno porazdeljena ploskovna obtežba [kN/m²]

A – površina etaže [m²]

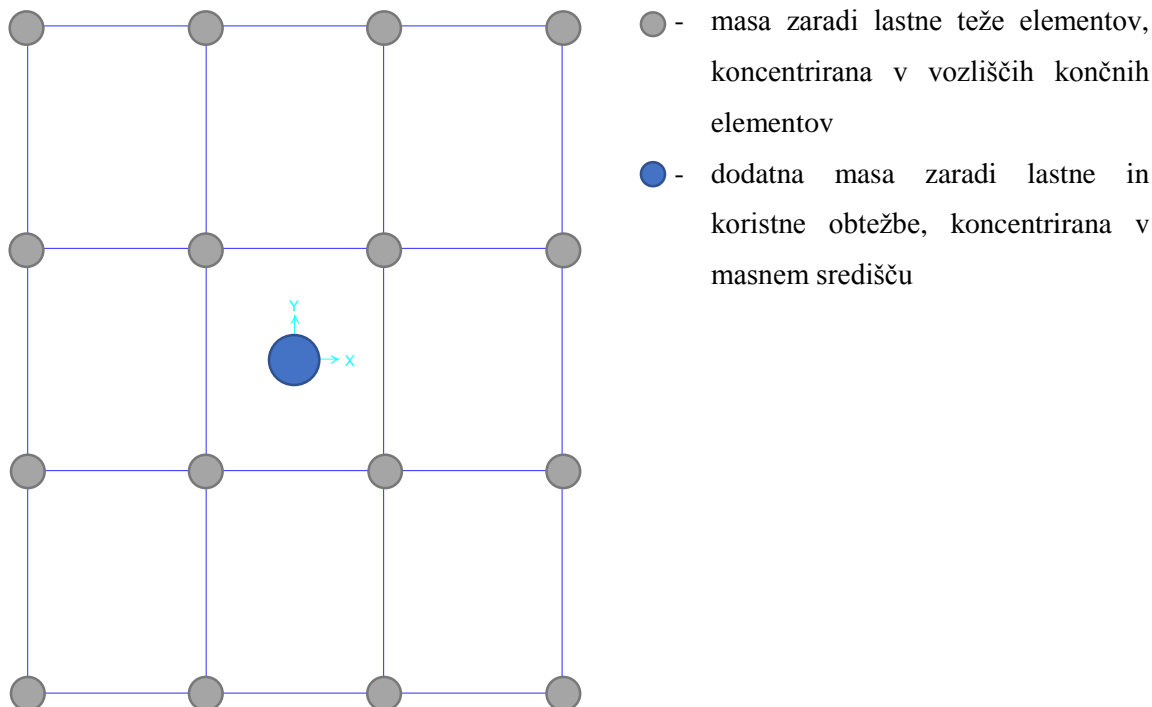
g – gravitacijski pospešek [m/s²]

I_x – vztrajnostni moment etaže okoli osi X [m⁴]

I_y – vztrajnostni moment etaže okoli osi Y [m⁴]

Dodatno maso podamo kot koncentrirano maso v masnem središču etaže, ki se zaradi predpostavke o enakomerno porazdeljeni obtežbi ujema s težiščem etaže. Ta se nahaja v izhodišču globalnega koordinatnega sistema. Za prostorsko konstrukcijo v vsaki etaži v smereh X in Y, na podlagi vhodnih parametrov izračunamo maso m ter masni vztrajnostni moment J okoli osi Z, za ravninsko konstrukcijo pa maso m .

Razporeditev mase po tlorisu etaže konstrukcije, glede na njen izvor je prikazana na sliki 8.



Slika 8: Shematski prikaz tlorisne razporeditve mase konstrukcije, glede na njen izvor

Zaradi upoštevanja predpostavke o togih ploščah, vse točke etaže povežemo v togo diafragmo in s tem poenotimo njihove horizontalne pomike v smeri X in smeri Y. Na ta način zmanjšamo število prostostnih stopenj konstrukcije.

Projektni spekter pospeškov

Konstrukcije so sposobne prenašati potresne vplive tudi v nelinearnem območju, zato lahko pri projektiranju upoštevamo sile, ki so manjše kot sile, ki ustrezajo elastičnemu odzivu. V izogib nelinearni analizi, lahko na račun sposobnosti konstrukcije, da z duktilnim obnašanjem sipa energijo in na račun dodatne nosilnosti, opravimo elastično analizo z zmanjšanim spektrom odziva. Take spektre imenujemo projektni spektri. Vrednost pospeškov in s tem velikost sil, zmanjšamo s faktorjem obnašanja q . Povzeto po [33].

Projektni spekter za vodoravni komponenti potresnega vpliva je v skladu s [34] določen z enačbami (41) – (44).

$$S_d(T) = a_g S \left[\frac{2}{3} + \frac{T}{T_B} \left(\frac{2.5}{q} - \frac{2}{3} \right) \right] \quad \text{za} \quad 0 \leq T \leq T_B \quad (41)$$

$$S_d(T) = a_g S \frac{2.5}{q} \quad \text{za} \quad T_B \leq T \leq T_C \quad (42)$$

$$S_d(T) \begin{cases} = a_g S \frac{2.5}{q} \left[\frac{T_C}{T} \right] \\ \geq \beta a_g \end{cases} \quad \text{za} \quad T_C \leq T \leq T_D \quad (43)$$

$$S_d(T) \begin{cases} = a_g S \frac{2.5}{q} \left[\frac{T_C T_D}{T^2} \right] \\ \geq \beta a_g \end{cases} \quad \text{za} \quad T_D \leq T \quad (44)$$

Za definicijo projektnega spektra, mora tako uporabnik v programu podati vrednost faktorja obnašanja q , projektni pospešek a_g za tla tipa A in izbrati tip tal na podlagi katerega so v skladu s [34] in [35] definirani še parametri S , T_B , T_C in T_D . Vrednosti so prikazane v preglednici 1.

Preglednica 1: Vrednosti parametrov S , T_B , T_C in T_D , glede na tip tal

Tip tal	S	$T_B(s)$	$T_C(s)$	$T_D(s)$
A	1.0	0.1	0.4	2.0
B	1.2	0.15	0.5	2.0
C	1.15	0.20	0.6	2.0
D	1.35	0.20	0.8	2.0
E	1.7	0.1	0.4	2.0

Privzeta vrednost faktorja beta, $\beta = 0,2$, je določena skladno s [34] ter [35] in je v programu ni mogoče prilagajati. Določa spodnjo mejo pri vodoravnem projektnem spektru.

Uporabnik lahko s pomočjo programa poda neomejeno število projektnih spektrov, ki jih nato uporabi pri definiranju obtežnih primerov.

Spreminjanje konstrukcije

Do sedaj smo omenili, da lahko s pomočjo programa ustvarimo in pripravimo analizo le za pravilne ravninske in prostorske okvirne konstrukcije. To storimo na podlagi parametrov, ki definirajo

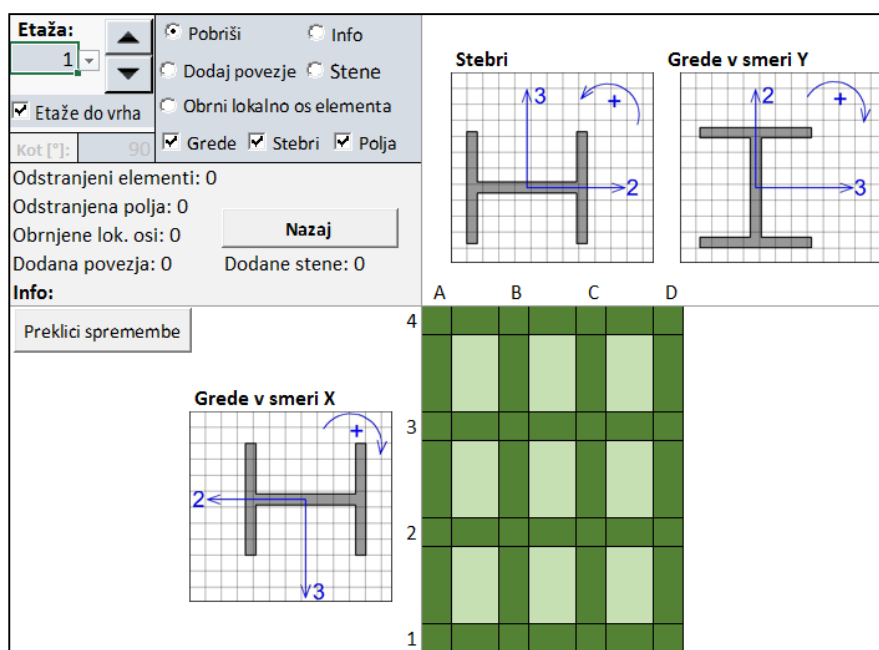
geometrijo konstrukcije, materiale, prečne prereze ter projektne spektre pospeškov. Z namenom, da bi omogočili nekoliko bolj svobodno oziroma poljubno kreiranje geometrije, smo v programu Microsoft Excel ustvarili poenostavljen grafični vmesnik, s pomočjo katerega je uporabniku omogočeno, da osnovno konstrukcijo na preprost in hiter način prilagodi.

Na voljo je več orodij s katerimi lahko:

- pobrišemo posamezne dele osnovne konstrukcije kot so stebri, grede ali polja v ravnini etaže, ki predstavljajo nanos ploskovne obtežbe,
- dodajamo togo vpeta diagonalna povezja med sosednjimi stebri konstrukcije,
- za poljuben kot obračamo lokalne osi prečnih prerezov gred in stebrov in
- med posameznimi osmi stebre in grede nadomestimo s stenami.

Konstrukcijo prikažemo v ravninah X-Y oziroma izrišemo tloris posamezne etaže, kot je razvidno iz slike 9. Vsaka izmed celic predstavlja en element konstrukcije. Da bi dobili pravilno geometrijsko razmerje med posameznimi razponi, prilagodimo velikost celic na podlagi vhodnih parametrov. Dodatno označimo osi konstrukcije po enakem vzorcu, kot bodo kasneje označene v programu SAP2000. Osi X označimo s številkami (1, 2, 3, ...), osi Y pa s črkami po vzorcu (A, B, C, ... , Z, AA, AB, AC, ...).

Temno zeleno obarvane kvadratne celice, ki se nahajajo na presečišču osi v smereh X in Y, predstavljajo stebre, ki ležijo pod nivojem obravnavane etaže. Z enako barvo so obarvane tudi celice, ki povezujejo posamezne stebre in predstavljajo grede na nivoju etaže. S svetlo zeleno barvo so obarvane celice, ki označujejo posamezna polja med osmi in predstavljajo nanos enakomerno porazdeljene ploskovne obtežbe.



Slika 9: Grafični vmesnik za prilagoditev osnovne okvirne konstrukcije

Z izbiro posameznega orodja lahko uporabnik s klikom na izbrano celico ali z označitvijo skupine celic, to orodje uporabi na konstrukciji. Uporabniku se modifikacije konstrukcije odražajo s spremembo barve ali obrob celic. V ozadju programa pa se sprememba barve celice odraža na lastnostih posameznih objektov (stebri, grede, polja), ki se nato preko programskega vmesnika prenesejo in upoštevajo v modelu programa SAP2000.

Uporabo funkcij grafičnega vmesnika bomo podrobneje opisali v navodilih za uporabo programa. Na tem mestu pa opišimo še algoritem za izračun težišča, mase in masnega vztrajnostnega momenta v primeru, da uporabnik del konstrukcije odstrani.

Kot smo že omenili, program SAP2000 maso zaradi lastne teže elementov izračuna samodejno, na podlagi volumna in specifične teže uporabljenih materialov, ter jo razporedi v vozlišča končnih elementov. Maso, ki je posledica obtežbe, izračunamo s pomočjo našega programa in jo koncentriramo v masno središče.

V primeru, da uporabnik odstrani katero izmed polj, ki predstavlja nanos enakomerno porazdeljene ploskovne obtežbe, se samodejno, s pomočjo spodnjega algoritma izračuna nova masa etaže, novo masno središče, ki se ujema s težiščem etaže in nov masni vztrajnostni moment.

V programu SAP2000 vedno najprej izrišemo osnovno okvirno konstrukcijo. Dimenzije so označene na sliki 10, v enačbah pa bomo z indeksom i označevali izhodiščne vrednosti. Izhodišče za vsako etažo tako predstavlja težišče T^i , ki se ujema z izhodiščem globalnega koordinatnega sistema, masa m^i izračunana z enačbo (45):

$$m^i = \frac{p A^i}{g} \quad (45)$$

kjer je

$$A^i = L_x L_y$$

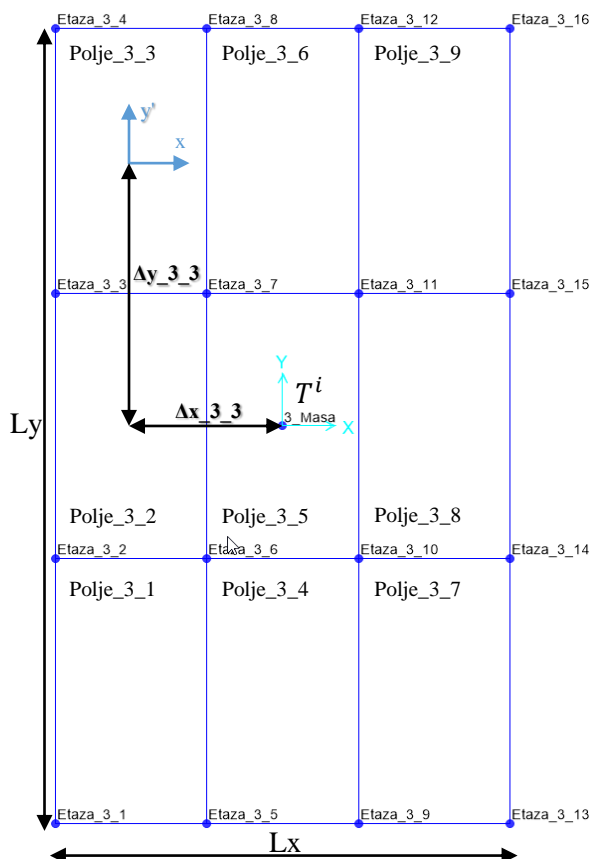
ter masni vztrajnostni moment J^i izračunan z enačbo (46):

$$J^i = m^i \frac{I_x^i + I_y^i}{A^i} \quad (46)$$

kjer je

$$I_x^i = \frac{L_x L_y^3}{12}$$
$$I_y^i = \frac{L_y L_x^3}{12}$$

Glede na pozicijo posameznega polja etaže, za vsakega izračunamo ter shranimo še razdaljo med lokalnim in globalnim izhodiščem koordinatnega sistema Δx_j in Δy_j .



Slika 10: Shema dimenzij, za izračun izhodiščnih vrednosti mase in masnega vztrajnostnega momenta etaže

Nato za vsako etažo preverimo posamezno celico, ki označuje nanos porazdeljene obtežbe. Primer etaže, z odstranjenimi polji, ki označujejo nanos porazdeljene obtežbe, je prikazan na sliki 11, z indeksom p pa bomo v enačbah označevali vrednosti, ki ne nanašajo na posamezno polje. V kolikor je celica označena kot izbrisana, od celotne ploščine etaže A^i najprej odštejemo ploščino posameznega polja A_j^p . Hkrati izračunamo še moment izbrisane polja S_{xj}^p okoli osi X in S_{yj}^p okoli osi Y, glede na izhodišče globalnega koordinatnega sistema. Postopek ponovimo za vsako odstranjeno polje. Novo ploščino ΔA in momenta ΔS_x ter ΔS_y modificirane etaže izračunamo z enačbami (47)–(49):

$$\Delta A = A^i - \sum_j^n A_j^p \quad (47)$$

$$\Delta S_x = \sum_j^n A_j^p (\Delta x_j) \quad (48)$$

$$\Delta S_y = \sum_j^n A_j^p (\Delta y_j) \quad (49)$$

pri čemer je n število izbranih polj etaže, Δx_j in Δy_j pa razdalje od težišča posameznih polj do izhodišča globalnega koordinatnega sistema.

	A	B	C	D
4				
	Polje_3_3		Polje_3_6	Polje_3_9
3				
	Polje_3_2		Polje_3_5	Polje_3_8
2				
	Polje_3_1		Polje_3_4	Polje_3_7
1				

Slika 11: Prikaz odstranjenih celic, ki označujejo nanos porazdeljene obtežbe (odstranjena sta 7. in 8. polje tretje etaže)

Razdalji do novega težišča ΔX in ΔY etaže dobimo s pomočjo enačb (50) in (51):

$$\Delta X = \frac{\Delta S_x}{\Delta A} \quad (50)$$

$$\Delta Y = \frac{\Delta S_y}{\Delta A} \quad (51)$$

Po izračunu novega težišča etaže T , za vsako izbrisano polje j izračunamo, glede na novo težišče, njegov vztrajnostni moment I_{xj}^p okoli osi X z enačbo (52) in I_{yj}^p okoli osi Y z enačbo (53). Novo težišče in razdalje so označene na sliki 12.

$$I_{xj}^p = \frac{l_x l_y^3}{12} + l_x l_y (\Delta y_j + \Delta Y)^2 \quad (52)$$

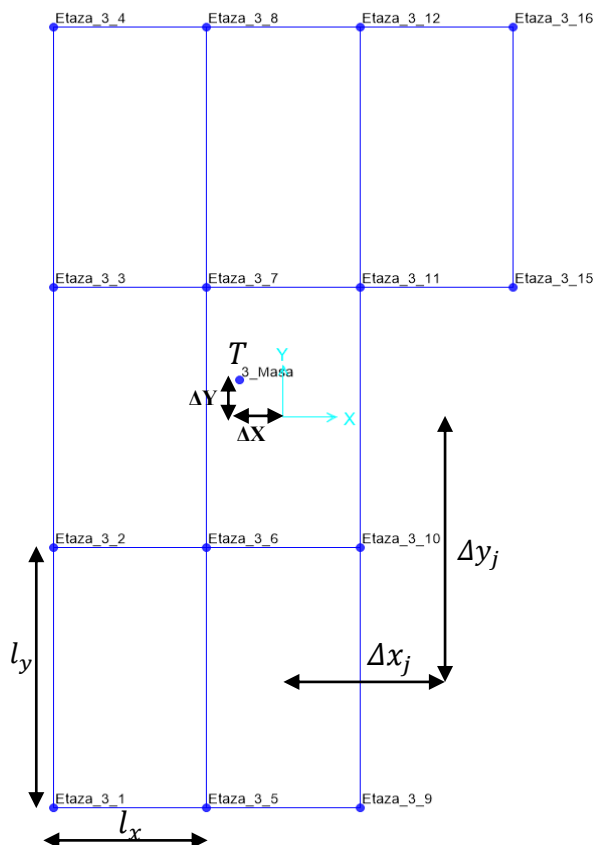
$$I_{yj}^p = \frac{l_y l_x^3}{12} + l_x l_y (\Delta x_j + \Delta X)^2 \quad (53)$$

Nova vztrajnostna momenta etaže ΔI_x in ΔI_y dobimo z enačbo (54) in enačbo (55):

$$\Delta I_x = I_x^i + A^i \Delta Y^2 - \sum_j^n I_{xj}^p \quad (54)$$

$$\Delta I_y = I_y^i + A^i \Delta X^2 - \sum_j^n I_{yj}^p \quad (55)$$

pri čemer je n število izbranih polj etaže.



Slika 12: Shema dimenzij, za izračun nove mase in masnega vztrajnostnega momenta modificirane etaže

Celotno maso Δm in masni vztrajnostni moment ΔJ okoli osi Z za spremenjeno etažo, dobimo z enačbama (56) in (57).

$$\Delta m = \frac{p \Delta A}{g} \quad (56)$$

$$\Delta J = \Delta m \frac{\Delta I_x + \Delta I_y}{\Delta A} \quad (57)$$

Opisani algoritem se izvrši za vsako etažo posebej.

Analiza in izpis rezultatov

Za račun nihajnih časov konstrukcije, uporabniku ni potrebno podati nobenih dodatnih parametrov. Nastavi lahko le minimalno oziroma maksimalno želeno število izračunanih nihajnih oblik, ki je odvisno od števila prostostnih stopenj konstrukcije. S predpostavko o togih ploščah, smo točke na nivoju etaže povezali v togo diafragmo in s tem poenotili njihove horizontalne premike ter zmanjšali število prostostnih stopenj konstrukcije.

Za modalno analizo s spektri odzivov, mora uporabnik ustvariti obtežni primer, pri čemer poda predhodno definirane spektre pospeškov v smeri globalnih osi X in Y. Po želji lahko smer delovanja spektrov, glede na posamezno globalno os, rotira za poljuben kot. Število ustvarjenih obtežnih primerov za posamezen model ni omejeno.

Po končanem vnosu vseh potrebnih parametrov, se v programu SAP2000 ustvari model konstrukcije in izvrši modalna analiza s spektri pospeškov, katere rezultati se nato avtomatsko prenesejo in izpišejo v programu Microsoft Excel. Za kombinacijo pomikov in notranjih sil posameznih nihajnih oblik, smo uporabili metodo geometrijske vsote oziroma metodo SRSS.

V prvi tabeli se izpišejo nihajne oblike, pripadajoči nihajni čas T [s], frekvence f [Hz], krožne frekvence ω [rad/s] in lastne vrednosti, ki predstavljajo rešitev posplošenega problema lastnih vrednosti in predstavljajo kvadrate krožnih frekvenc ω^2 [rad²/s²], enačba (17).

V drugi tabeli za vsako nihajno obliko ponovno izpišemo nihajni čas T [s], efektivne modalne mase v smereh X, Y in okoli smeri Z ter njihove vsote. Pri tem v skladu s [34] obarvamo vse nihajne oblike, katerih efektivne modalne mase so večje od 5% celotne mase konstrukcije in označimo nihajno obliko, pri kateri je vsota efektivnih modalnih mas večja ali enaka 90% celotne mase konstrukcije.

V tretji tabeli izpišemo horizontalne pomike etaž v metrih, ki so posledica modalne analize s spektri odziva. V kolikor smo definirali več obtežnih primerov, dobimo v sklopu te tabele rezultate za vsakega izmed njih.

Na zadnje izpišemo še prečne sile [kN] in momente [kNm] v stebrih. V tem sklopu prav tako dobimo rezultate za vsak definiran obtežni primer posebej. Za vsako etažo prikažemo prečne sile V_2 in V_3 , ter momente M_2 in M_3 . Za boljšo vizualno predstavo o razporeditvi notranjih sil po konstrukciji, rezultate obarvamo. Pri tem z rdečo barvo označimo večje ter s postopnim prehodom do zelene manjše vrednosti notranjih sil. Maksimalne vrednosti prečnih sil in momentov posameznih obtežnih primerov dodatno označimo z odebeljenim izpisom.

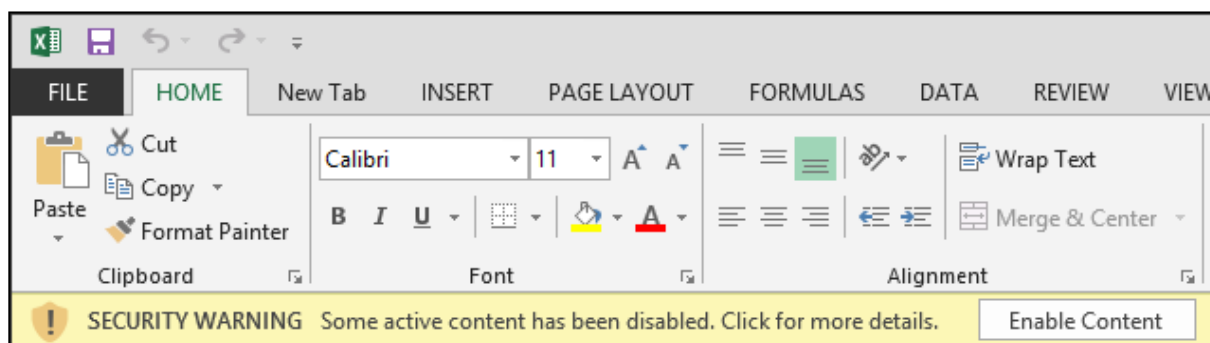
4.3 Navodila za uporabo programa

4.3.1 Zahteve in namestitve

Za delovanje programa je potrebno imeti na računalniku nameščen Microsoft Excel 2007 ali novejšo verzijo ter SAP2000 15.0.0 ali novejšo različico iste verzije.

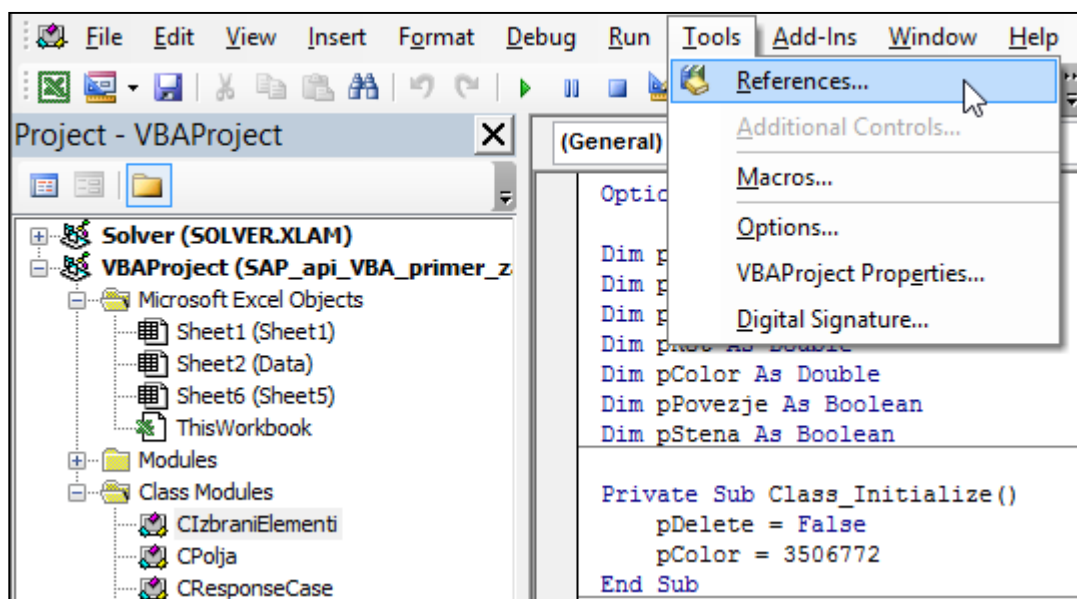
Sam program sestavljata dve datoteki in sicer Excelov delovni zvezek z imenom 'Program', preko katerega upravljamo s programom SAP2000 in datoteka 'SAP_Material.accdb' v kateri je shranjena baza materialnih karakteristik. Datoteki lahko shranimo na poljubno mesto na disku, vendar se morata obe nahajati v isti mapi, kjer se mora prav tako nahajati priložena mapa z imenom 'Slike'. V njej so shranjene slike prečnih prerezov elementov in služijo za vizualno pomoč uporabniku pri rotaciji lokalnih osi elementov. Imena datoteke v kateri se nahaja baza podatkov ne smemo spreminjati, saj v nasprotnem primeru program ne bo deloval.

Zaradi varnostnih razlogov je uporaba programske kode ob odprtju programa Excel onemogočena. Če nas pričaka varnostno opozorilo prikazano na sliki 13, ki nas na to opozori, je potrebno klikniti na gumb 'Omogoči vsebino' (angl. Enable Content).



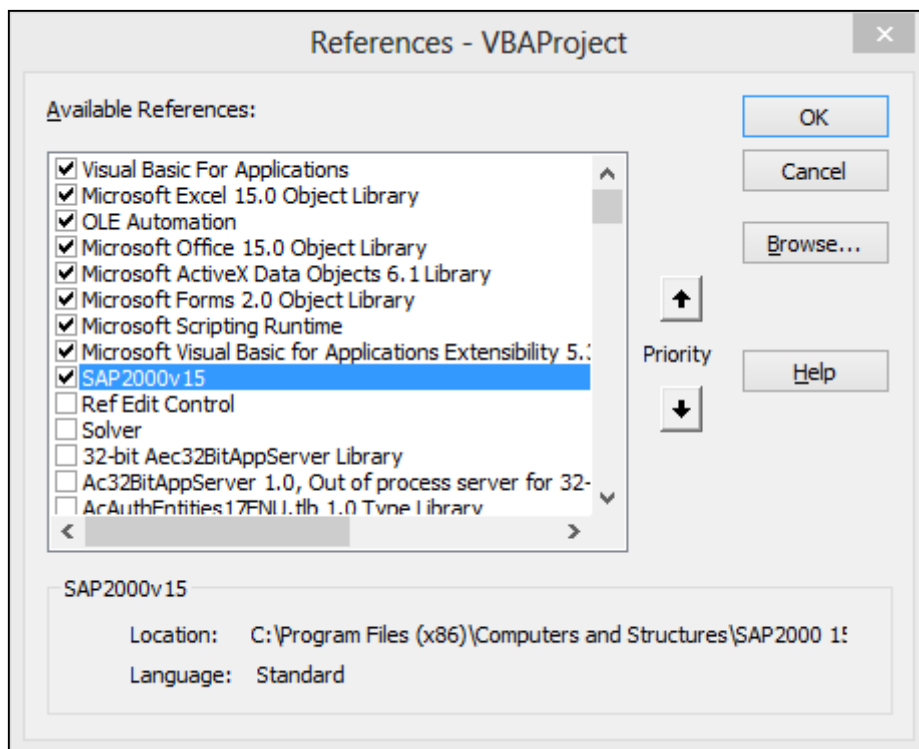
Slika 13: Varnostno opozorilo programa Microsoft Excel

Program za svoje delovanje potrebuje dostop do dinamične knjižnice z imenom 'SAP2000v15.dll', ki smo jo med samim programiranjem že dodali na seznam referenc. Če je program SAP2000 verzije 15 nameščen na računalniku, se omenjena knjižnica poišče in doda samodejno. V kolikor se želimo o tem prepričati, v orodnem traku programa Excel kliknemo na zavihek 'Razvijalec' (angl. Developer) in nato gumb 'Visual Basic'. Odpre se novo okno, preko katerega dostopamo do programske kode. Omenjeno okno lahko odpremo tudi s pritiskom kombinacije tipk na tipkovnici (alt+F11). V orodni vrstici kliknemo na 'Orodja' (ang. Tools) in nato 'Reference' (angl. References), kar prikazuje slika 14.



Slika 14: Gumb 'Reference' menija 'Orodja' (angl. 'Tools'), s katerim odpremo pogovorno okno za izbiro referenc

Odpre se okno prikazano na sliki 15, v katerem se mora med drugimi nahajati in biti obkljukana tudi knjižnica 'SAP2000v15'. V kolikor te knjižnice ni, je potrebno klikniti na gumb 'Dodaj' (angl. Browse) in jo poiskati na disku. Nahaja se v inštalacijskem direktoriju programa SAP2000.



Slika 15: Pogovorno okno za izbiro referenc, v katerem obkljukamo dinamično povezovalno knjižnico z imenom 'SAP2000v15'

4.3.2 Vnos parametrov za definicijo konstrukcije

Ob zagonu programa se nam odpre na sliki 16 prikazan vmesnik za podajanje parametrov, s katerimi definiramo geometrijo konstrukcije, prečne prereze, materiale in parametre za analizo.

Ime Modela		Odpri rezultate		Material		Prerez	
Številno razponov		Smer X	Smer Y	Stebri		C 25 / 30	
Dimenzija razpona		4.00 m	5.00 m			Pravokoten	
Številno etaž		1				t2 0.30 m (vzp. osi X)	
Višina Etaže		3.20 m				t3 0.30 m (vzp. osi Y)	
Obtežba		20.00 kN/m2				Material	
Številno nihajnih oblik		1				Grede	
Tip tal		A				C 20 / 25	
Faktor obnašanja q		2				Pravokoten	
Dušenje		0.05				t2 0.20 m (vzp. osi Z)	
						t3 0.20 m	
						Material	
						Diagonale	
						S 235	
						Kotnik	
						L60X10	
						t2 0.05 m	
						t3 0.05 m	
						Material	
						Stene	
						C 25 / 30	
						Debelina	
						0.15 m	
						Pripravi model	
						Dodaj RS Case	
						Odstrani vse	
						Izberi smer	
						Ime spektra	
						Scale Faktor	
						Kot [°]	
						U1	
						U2	

Slika 16: Osnovni vmesnik za upravljanje programa in podajanje parametrov

V celico C3 vpišemo ime modela, ki bo ob zagonu analize služil za kreiranje istoimenske mape v direktoriju programa in poimenovanje datoteke modela v programu SAP2000 ter Excelove datoteke z

rezultati. Če model z istim imenom že obstaja, nas program na to opozori. V kolikor opozorila ne upoštevamo, bomo model in rezultate že obstoječega modela nadomestili z novimi.

V naslednjem koraku s pritiskom na gumb '3D Okvir' oziroma '2D Okvir' določimo, ali bomo analizirali prostorsko oziroma ravninsko konstrukcijo. Če izberemo slednjo, v polju C6 vpišemo ali s pritiskom na puščice določimo število razponov v smeri X ter v polju C7 njihov medsebojni razmak v metrih. Podajanje parametrov za smer Y je v tem primeru onemogočeno. Ob izbiri prostorskega okvira, podamo še parametre za smer Y. Vnos parametrov je prikazan na sliki 17.

	Smer X	Smer Y	
Število razponov	6	7	<input type="radio"/> 3D Okvir
Dimenzija razpona	4.00 m	5.00 m	<input checked="" type="radio"/> 2D Okvir

Slika 17: Polja za izbiro tipa okvirne konstrukcije, vnos števila njenih razponov in dimenzij

Na enak princip določimo število etaž in etažno višino v metrih, kar prikazuje slika 18.

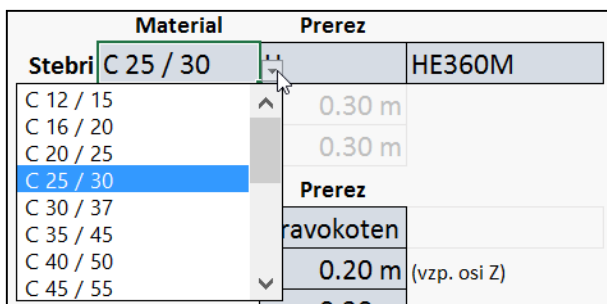
Število etaž	4
Višina etaže	3.20 m

Slika 18: Polji za vnos števila etaž in etaže višine

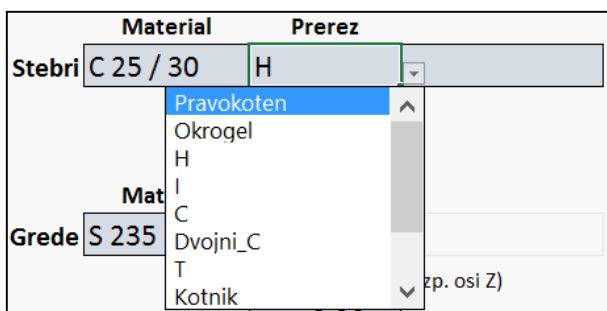
S klikom na enega izmed gumbov 'Togo podprto' oziroma 'Vrtljivo podprto', določimo način podpiranja vseh točk na koti terena pri $Z=0$.

V celici C13 podamo projektno obtežbo konstrukcije, na podlagi katere program določi maso. Obtežba mora poleg koristne, zajemati tudi lastno težo medetažne plošče in ostalih nekonstrukcijskih elementov nad njo. Pri prostorskih konstrukcijah podajamo enakomerno porazdeljeno obtežbo v kN/m^2 , pri ravninskih pa linijsko obtežbo v kN/m .

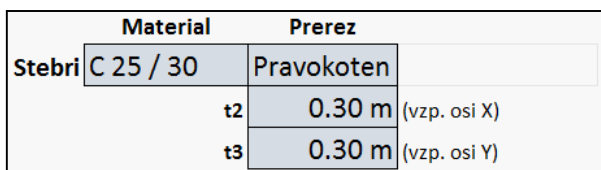
Po določitvi naštetih parametrov določimo material, obliko ter dimenzije prečnih prereзов stebrov in gred, ter po potrebi tudi diagonalnih povezij in sten. Princip določitve je enak za vse, zato bomo pokazali le primer definicije prečnega prereza stebrov. Pri določitvi prereza sten je, poleg materiala, potrebno podati zgolj debelino stene v metrih.



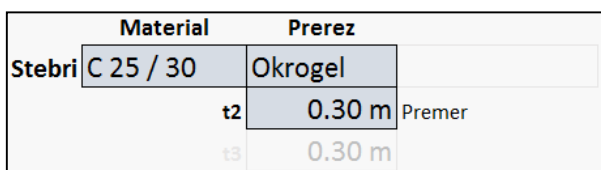
Slika 19: Prikaz izbire materiala iz ponujenega seznama



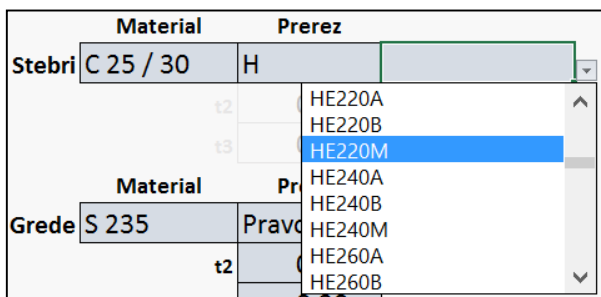
Slika 20: Prikaz izbire oblike prereza iz ponujenega seznama



Slika 21: Prikaz vnosa dimenzij pravokotnega prereza



Slika 22: Prikaz vnosa dimenzij okroglega prereza



Slika 23: Prikaz izbire standardnih prerezov

Na sliki 19 je prikazano, kako v celici H4 iz ponujenega seznama izberemo želen material. Na voljo so vsi materiali definirani v bazi materialnih karakteristik, ki se nahaja v priloženi datoteki 'SAP_Material.accdb'. Če v bazo dodamo novo definicijo materiala, bo ta ob naslednjem zagonu programa avtomatsko na voljo v prikazanem seznamu.

Slika 20 prikazuje, kako v sosednji celici I4 iz seznama izberemo obliko prečnega prereza. Na voljo so pravokotni in okrogli prerezi, prerezi tipa H, I, C, Dvojni C, T, Kotnik, Dvojni kotnik ter cevasti in škatlasti prerezi.

V primeru izbire pravokotnega prereza, se aktivirata celici I5 in I6, v kateri je potrebno vpisati podatke o dimenziji prereza v metrih. Celice za vnos podatkov so prikazane na sliki 21. Lokalna os prereza t2 je vzporedna globalni osi X, lokalna os prereza t3 pa globalni osi Y.

V primeru izbire okroglega prereza, se aktivira celica I5, v katero je potrebno podati premer prečnega prereza v metrih. Celice za vnos podatkov so prikazane na sliki 22.

V primeru izbire katere koli druge oblike prečnega prereza, se aktivira polje J4 prikazano na sliki 23, v katerem je potrebno iz ponujenega seznama izbrati enega izmed standardnih prerezov tega tipa.

4.3.3 Vnos parametrov za definicijo analize

Število nihajnih oblik je pogojeno s številom prostostnih stopenj konstrukcije. Da bi skrajšali čas računa in velikost oziroma obseg rezultatov, lahko v celicah C16 in D16 omejimo obseg tako, da nastavimo minimalno ali maksimalno želeno število izračunanih nihajnih oblik. V kolikor je podana minimalna vrednost večja od dejanskega števila nihajnih oblik modela konstrukcije, se bodo izračunali ter izpisali rezultati vseh.

V območje celic C18:C21 na sliki 24, vstavimo parametre, ki definirajo projektni spekter pospeškov. V celici 'Tip tal' iz seznama izberemo eno izmed ustreznih oznak od A do E, v preostalih pa ročno vnesemo še podatke o projektnem pospešku, faktorju obnašanja in dušenju.

Spekter se definira s pritiskom na gumb 'Dodaj RS Funkcijo'. Oznaka RS je kratica za v programu SAP2000 uporabljen angleški izraz Response Spectrum oziroma spekter odziva. Definiramo lahko poljubno število spektrov. Z njihovim dodajanjem se osvežujejo celice, ki prikazujejo število definiranih spektrov ter seznam, iz katerega je mogoče razbrati ime oziroma zaporedno oznako spektra ter parametre, ki ga definirajo.

S pritiskom na gumb 'Odstrani vse', lahko izbrišemo vse do sedaj ustvarjene spektre pospeškov. V kolikor smo te spektre že uporabili pri definiciji obtežnih primerov, se izbrišejo tudi ti.

Tip tal	A	Dodaj RS Funkcijo
ag	0.225	
Faktor obnašanja q	2	
Dušenje	0.05	
		Odstrani vse
Število RS Funkcij: 2		
RsFunkcija0		A 0.225 3 0.05
RsFunkcija1		A 0.225 2 0.05

Slika 24: Območje za definicijo projektnih spektrov pospeškov

Obtežne primere definiramo v območju celic G20:J21, prikazanem na sliki 25. Za vsako izbrano smer vzbujanja, kjer je U1 vzporedna z globalno osjo X, U2 pa z globalno osjo Y, iz seznama izberemo enega izmed predhodno definiranih spektrov pospeškov. V sosednjih celicah lahko nastavimo še velikostni faktor in kot zasuka delovanja spektra glede na izbrano smer.

Obtežni primer definiramo s pritiskom na gumb 'Dodaj RS Case', njihovo število pa zopet ni omejeno. Podobno kot pri definiranju projektnih spektrov odzivov, se tudi tu, z dodajanjem obtežnih primerov osvežujejo celice, ki prikazujejo njihovo število ter seznam, iz katerega je razvidno ime oziroma zaporedna oznaka obtežnega primera ter parametri, ki ga definirajo.

S pritiskom na gumb 'Odstrani vse', lahko pobrišemo vse do sedaj ustvarjene obtežne primere.

Dodaj RS Case		Odstrani vse	
Izberi smer	Ime spektra	Scale Faktor	Kot [°]
U1	RsFunkcija0	9.81	0
U2	RsFunkcija1	9.81	0

Število RS Case-ov: 1

RsCase0 U1 RsFunkcija0 9.81 0 ; U2 RsFunkcija1 9.81 0 ;

Slika 25: Območje za definicijo obtežnih primerov

4.3.4 Spreminjanje geometrije konstrukcije

Če želimo spremeniti osnovno zasnovo ravninskega oziroma prostorskega okvira, kliknemo na gumb 'Spremeni geometrijo'. Odpre se nov list prikazan na sliki 26, na katerem se izriše tloris etaže konstrukcije, iz katerega je razvidna pozicija stebrov, gred in nanos obtežbe. Glede na izbiro oblike prečnih prereзов, se ob konstrukciji prikažejo ustrezne slike, ki označujejo osnovno pozicijo lokalnih koordinatnih osi posameznih prereзов stebrov in gred glede na globalni koordinatni sistem. Na slikah je označena tudi pozitivna smer zasuka elementov.

Etaža:

1

☒ Etaže do vrha

Kot [°]: 90

Odstranjeni elementi: 0

Odstranjena polja: 0

Obrnjene lok. osi: 0

Dodana povezja: 0

Info:

Prekllici spremembe

☐ Pobriši ☐ Info

☐ Dodaj povezje ☐ Stene

☐ Obrni lokalno os elementa

☒ Grede ☒ Stebri ☒ Polja

Nazaj

Dodane stene: 0

Stebri

Grede v smeri Y

Grede v smeri X

	A	B	C	D	E	F	G
4							
3							
2							
1							

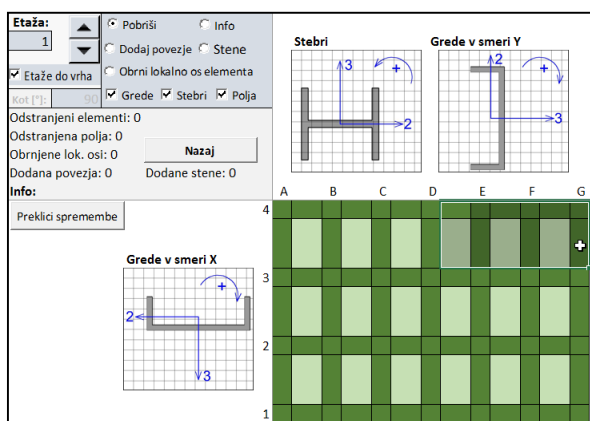
Slika 26: Grafični vmesnik za prilagoditev geometrije osnovne okvirne konstrukcije

Spremembe konstrukcije lahko izvedemo neodvisno za vsako etažo posebej ali za več etaž hkrati. V kolikor želimo izbrati slednjo možnost, obkljukamo opcijo 'Etaže do vrha' prikazano na sliki 26. Spremembe, ki jih bomo podali za trenutno izbrano etažo, se bodo tako odražale na vseh, ki se nahajajo nad njo.

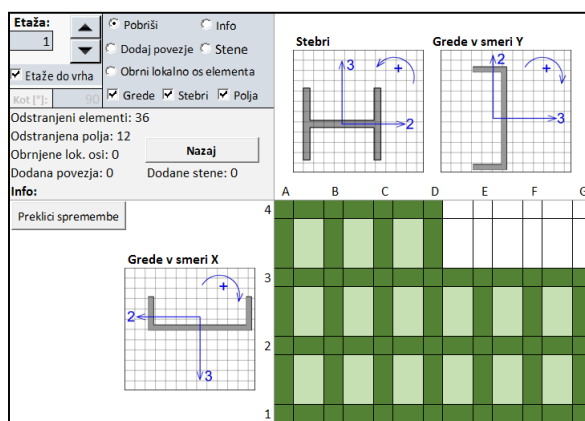
Trenutno izrisana etaža se izpiše v zgornjem levem kotu. Če kliknemo na omenjeno celico, lahko iz ponujenega seznama izberemo prikaz katere koli etaže. S pritiskom na puščice lahko izrišemo etažo, ki se nahaja pod ali nad trenutno.

Desno od opcije izbora etaže se nahajajo orodja, s katerimi je mogoče spreminjati osnovno geometrijo. Izbrano orodje uporabimo tako, da kliknemo na zeleno celico, ki predstavlja enega izmed elementov ali pa označimo več celic hkrati, ter orodje uporabimo na vseh označenih elementih. Pod orodji se nahaja še izbira območja, na katerem ima izbrano orodje svoj vpliv. Če želimo orodje uporabiti na vseh elementih, potem obkljukamo vse opcije prikazane na sliki 26 in sicer 'Grede', 'Stebri' ter 'Polja'. Če katera izmed teh opcij ni izbrana, orodje na elementu ne bo imelo vpliva, kljub temu da kliknemo na celico ali pa je celica zajeta v izbranem območju.

Z orodjem 'Pobriši', odstranimo označene elemente. Primer označitve elementov z orodjem 'Pobriši' je prikazan na sliki 27. V prikazanem primeru odstranimo vse označene stebre, grede in polja z obtežbo. Celice izbranih elementov se obarvajo z belo barvo, kar prikazuje slika 28. Odstranjeni stebri predstavljajo stebre pod izrisano etažo. Če orodje uporabimo na izbranih elementih, se ti ponovno izrišejo in obarvajo z zeleno barvo.

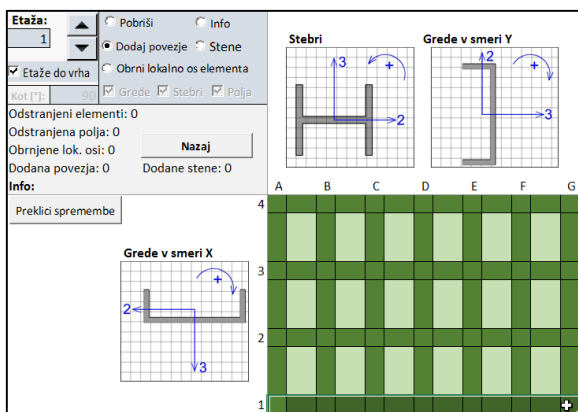


Slika 27: Izbor celic z orodjem 'Pobriši'

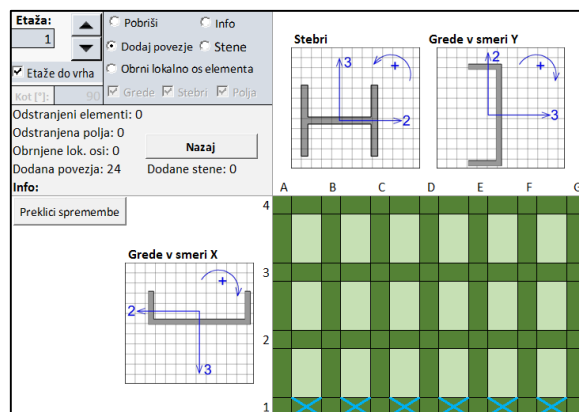


Slika 28: Grafičen prikaz odstranjenih elementov

Z orodjem 'Dodaj povezja', dodamo diagonalno povezje v polju med dvema stebroma, ki se nahajata med izbrano etažo in etažo pod njo. Ker povezje dodajamo med dva stebra, so veljavna območja izbranega orodja le celice, ki predstavljajo grede konstrukcije. Orodje ne bo imelo vpliva na celice, ki označujejo stebre, stene ali polja obtežbe. Primer označitve elementov z orodjem 'Dodaj povezja' je prikazan na sliki 29. Dodano povezje v celici vizualno prikažemo z dvema diagonalnima modrima črtama, kar je razvidno iz slike 30. V kolikor z izbranim orodjem kliknemo v celico, ki že vsebuje povezje, se le to odstrani.

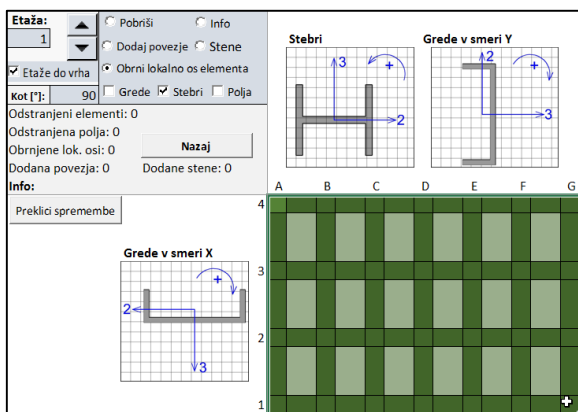


Slika 29: Izbor celic z orodjem 'Dodaj povezje'

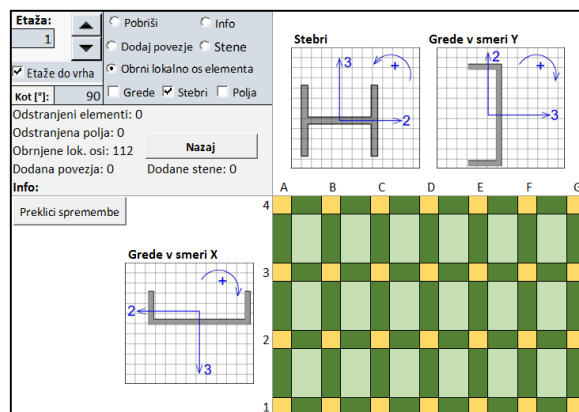


Slika 30: Grafičen prikaz dodanih povezij

Z orodjem 'Obrni lokalno os elementa', lahko rotiramo prečne prereze. Pri tem si pomagamo s prikazanimi slikami ob konstrukciji, ki nakazujejo osnovne položaje prečnih prerezov in pozitivno smer rotacije tako za stebre kot grede. Z izbiro orodja se aktivira celica, v kateri podamo kot rotacije v stopinjah, ki je lahko pozitiven ali negativen. Na sliki 31 je prikazano, kako hkrati rotiramo lokalne osi vseh stebrov konstrukcije za 90°. Pri izbiri veljavnosti območja orodja, obkljukamo zgolj opcijo 'Stebri'. Tako lahko označimo celotno konstrukcijo, pri tem pa bomo za izbrani kot rotirali le lokalne osi stebrov, ne pa tudi gred. Celice rotiranih prečnih prerezov vizualno ponazorimo z rumeno barvo, kot prikazuje slika 32. V kolikor z izbranim orodjem označimo celico, katere prerez smo že rotirali, se ta povrne v osnovi položaj, celica pa se obarva zeleno.

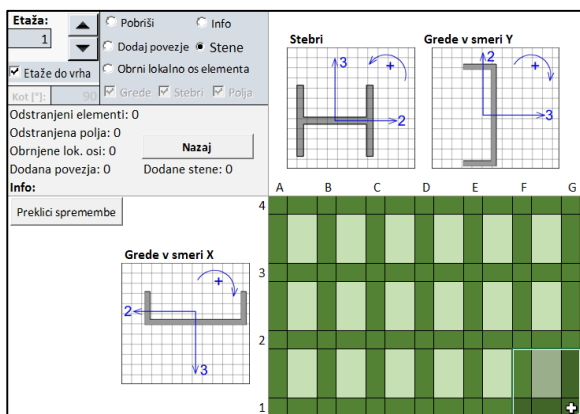


Slika 31: Izbor celic z orodjem 'Obrni lokalno os elementa'

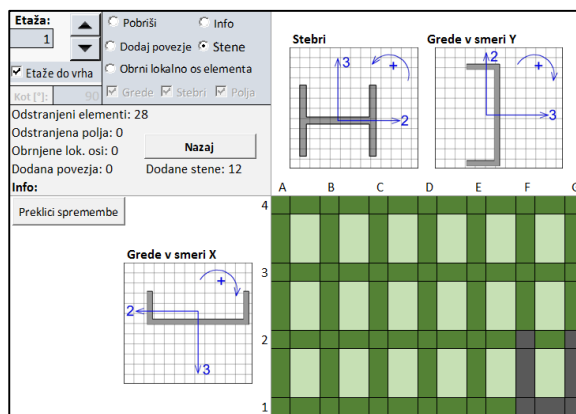


Slika 32: Grafičen prikaz elementov z rotirano lokalno osjo

Z orodjem 'Stene', lahko med posameznimi osmi konstrukcije, stebre in grede nadomestimo s stenami. Stene dodajamo med trenutno izrisano etažo in etažo pod njo. V kolikor dodamo stene v prvi etaži, bodo te podprte na enak način kot stebri, se pravi togo ali vrtljivo, odvisno od izbire. Enako kot pri orodju 'Dodaj povezja', so območja veljavnosti orodja 'Stene' le celice, ki predstavljajo grede konstrukcije. Primer označitve dela konstrukcije z orodjem 'Stene' je prikazan na sliki 33. Celice, kjer smo dodali stene, vizualno ponazorimo s sivo barvo, kar prikazuje slika 34. Če želimo stene odstraniti, lahko to storimo z orodjem 'Pobriši'.

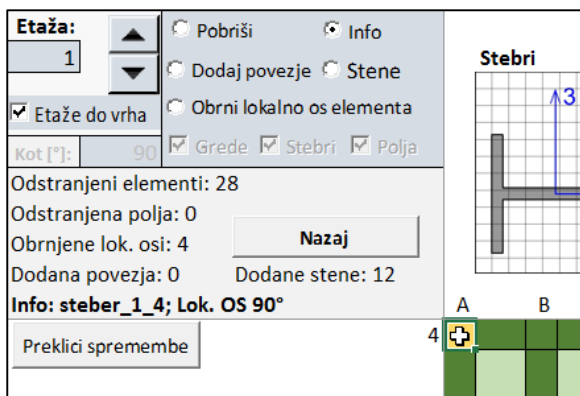


Slika 33: Izbor celic z orodjem 'Stene'



Slika 34: Grafičen prikaz dodanih sten

Z orodjem 'Info' je mogoče prikazati vse informacije posamezno označene celice. Če s tem orodjem označimo celico, se nam na zaslonu izpiše kateri element predstavlja, ali je izbrisan, ali je na tem mestu dodano povezje oziroma stena ter ali je prečni prerez rotiran in če, za koliko stopinj. Primer je prikazan na sliki 35. To orodje, poleg obarvanih celic, omogoča dodaten nadzor nad izrisano konstrukcijo.



Slika 35: Izbor celice z orodjem 'Info', odebelen izpis informacij o izbranem elementu in prikaz števila odstranjenih elementov ter polj, števila elementov z obrnjenimi lokalnimi osi in števila dodanih povezij ter sten celotne konstrukcije

Z uporabo orodij se hkrati samodejno osvežujejo celice, ki za celotno konstrukcijo prikazujejo število odstranjenih elementov oziroma polj, število dodanih povezij in sten, ter število elementov, katerim smo rotirali lokalni koordinatni sistem. Slednje je prikazano na sliki 35. Namen prikaza je dodatna kontrola in sledljivost spreminjanja konstrukcije.

Ko zaključimo z vsemi spremembami osnovne konstrukcije, se s klikom na gumb 'Nazaj' vrnemo na osnovni delovni list za podajanje parametrov. V kolikor s spremembami nismo zadovoljni oziroma bi želeli konstrukcijo povrniti v prvotno stanje, to lahko storimo s klikom na gumb 'Preklici spremembe'.

4.3.5 Analiza in rezultati

Po vnosu vseh parametrov lahko zaženemo analizo s klikom na gumb 'Pripravi model', prikazanem na sliki 16. Program SAP2000 mora biti pri tem predhodno zaprt. V kolikor vsi potrebni parametri za analizo niso podani, nas program na to opozori.

Od tu dalje je ves proces avtomatiziran. Najprej se odpre program SAP2000, nato pa se preko programskega vmesnika prenesejo vsi ukazi in potrebni parametri, ki smo jih določili. Trajanje celotnega procesa je odvisno od velikosti modela. Med tem procesom se v delovni mapi, kjer se nahaja naš program ustvari nova mapa z imenom modela, ki smo ga določili na začetku. Vanjo se shrani trenutno analiziran model konstrukcije ter novo ustvarjena Excelova datoteka z rezultati. Po končani analizi se program SAP2000 samodejno zapre, hkrati pa se v programu aktivirata dva nova gumba in sicer 'Odpri rezultate' in 'Odpri model', ki sta prav tako prikazana na sliki 16. S klikom na prvega se odpre Excelov dokument v katerem so zbrani vsi rezultati, s klikom na drugega pa lahko odpremo ustvarjen model v programu SAP2000. Primer rezultatov analize togo podprte betonske okvirne konstrukcije s tremi razponi v smeri X dimenzij 4,00 m, štirimi razponi v smeri Y dimenzij 5,00 m in štirimi etažami višine 3,20 m je prikazan na slikah 36-39.

Primer izpisa nihajnih časov in frekvenc za posamezno nihajno obliko v generirani datoteki z rezultati je prikazan na sliki 36.

Nihajni casi in frekvence:							
	Load Case	Step Type	Step Num	Nihajni cas [s]	Frekvenca [Hz]	Krozna Frekvenca [rad/s]	Lastna Vrednost [rad2/s2]
	Modal	Mode	1	0.5324	1.8784	11.8021	139.2894
	Modal	Mode	2	0.5119	1.9536	12.2750	150.6750
	Modal	Mode	3	0.4405	2.2701	14.2634	203.4444
	Modal	Mode	4	0.1629	6.1376	38.5637	1487.1566
	Modal	Mode	5	0.1581	6.3234	39.7311	1578.5601
	Modal	Mode	6	0.1368	7.3091	45.9244	2109.0471
	Modal	Mode	7	0.0873	11.4596	72.0030	5184.4370
	Modal	Mode	8	0.0859	11.6424	73.1513	5351.1100
	Modal	Mode	9	0.0749	13.3579	83.9304	7044.3080
	Modal	Mode	10	0.0599	16.6990	104.9227	11008.7750

Slika 36: Primer izpisa nihajnih časov in frekvenc za posamezno nihajno obliko

Primer izpisa efektivnih mas in njihovih vsot za posamezno nihajno obliko v generirani datoteki z rezultati je prikazan na sliki 37.

Nihajni casi in efektivne mase:										
	Load Case	Step Type	Step Num	Nihajni cas [s]	Ux	Uy	Rz	SumUx	SumUy	SumRz
	Modal	Mode	1	0.532379	0.000000	0.828687	0.000000	0.000000	0.828687	0.000000
	Modal	Mode	2	0.511869	0.832199	0.000000	0.000000	0.832199	0.828687	0.000000
	Modal	Mode	3	0.440511	0.000000	0.000000	0.834072	0.832199	0.828687	0.834072
	Modal	Mode	4	0.162930	0.000000	0.113215	0.000000	0.832199	0.941902	0.834072
	Modal	Mode	5	0.158143	0.111923	0.000000	0.000000	0.944122	0.941902	0.834072
	Modal	Mode	6	0.136816	0.000000	0.000000	0.110526	0.944122	0.941902	0.944598
	Modal	Mode	7	0.087263	0.000000	0.044204	0.000000	0.944122	0.986106	0.944598
	Modal	Mode	8	0.085893	0.042680	0.000000	0.000000	0.986802	0.986106	0.944598
	Modal	Mode	9	0.074862	0.000000	0.000000	0.042315	0.986802	0.986106	0.986913
	Modal	Mode	10	0.059884	0.000000	0.013893	0.000000	0.986802	1.000000	0.986913

Slika 37: Primer izpisa nihajnih časov, efektivnih mas in njihovih vsot za posamezno nihajno obliko

Primer izpisa horizontalnih pomikov masnih točk za vsako etažo v generirani datoteki z rezultati je prikazan na sliki 38.

Pomiki masnih tock:		
	U1 - RsCase0 [m]	U2 - RsCase0 [m]
1. etaža	0.004485	0.004566
2. etaža	0.011033	0.011397
3. etaža	0.016358	0.017004
4. etaža	0.019534	0.020379

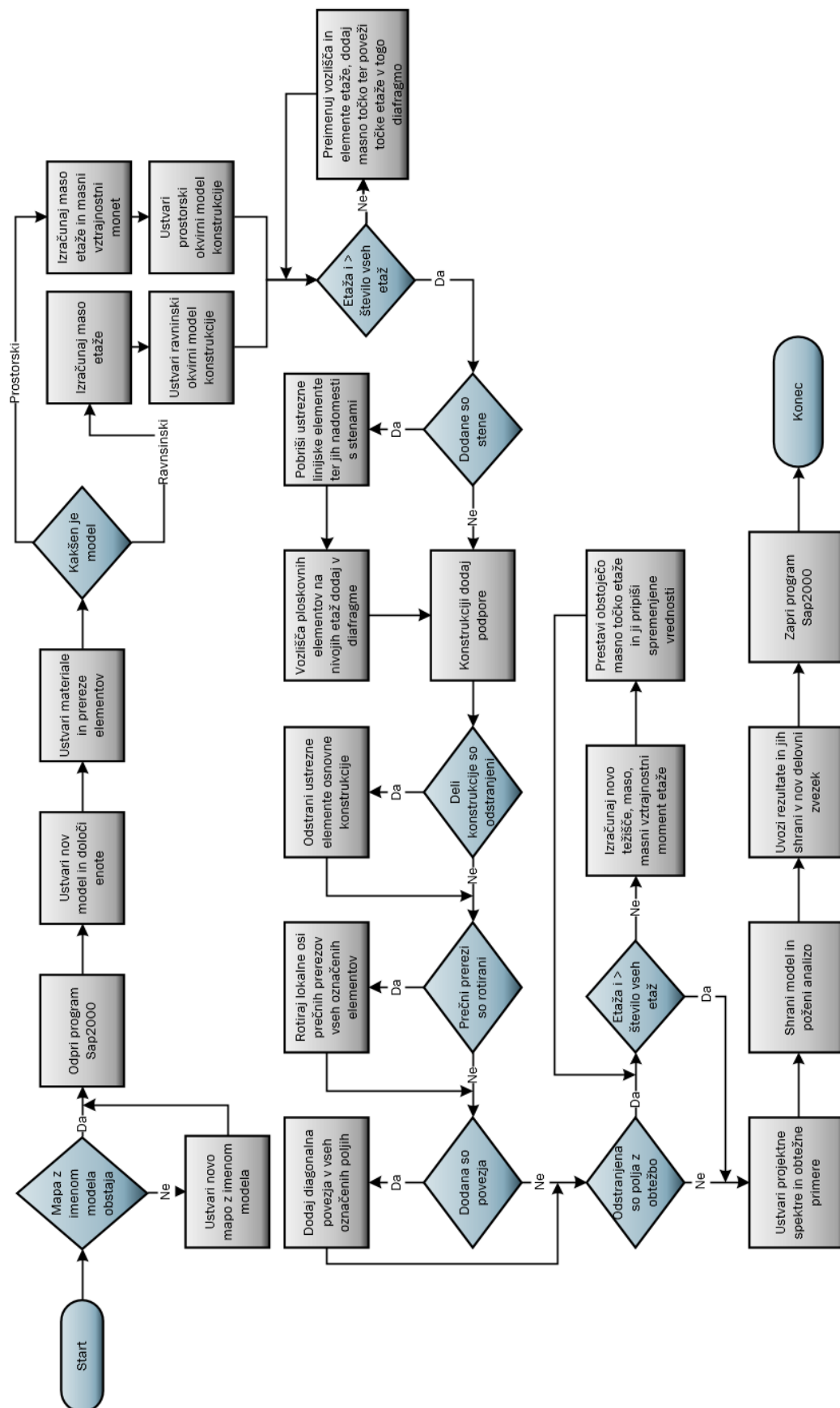
Slika 38: Primer izpisa horizontalnih pomikov masnih točk za vsako etažo

Primer izpisa prečnih sil in momentov v stebrih za vsako etažo v generirani datoteki z rezultati je prikazan na sliki 39.

Notranje sile v stebrih:				1. etaža -V2- RScase0 [kN]				1. etaža -V3- RScase0 [kN]				1. etaža -M2- RScase0 [kNm]				1. etaža -M3- RScase0 [kNm]			
	45.994	59.701	59.701	45.994		43.403	43.403	43.403	43.403		107.968	107.968	107.968	107.968		109.574	123.888	123.888	109.574
	45.994	59.701	59.701	45.994		55.945	55.945	55.945	55.945		121.072	121.072	121.072	121.072		109.574	123.888	123.888	109.574
	45.994	59.701	59.701	45.994		55.212	55.212	55.212	55.212		120.308	120.308	120.308	120.308		109.574	123.888	123.888	109.574
	45.994	59.701	59.701	45.994		55.945	55.945	55.945	55.945		121.072	121.072	121.072	121.072		109.574	123.888	123.888	109.574
	45.994	59.701	59.701	45.994		43.403	43.403	43.403	43.403		107.968	107.968	107.968	107.968		109.574	123.888	123.888	109.574
2. etaža -V2- RScase0 [kN]				2. etaža -V3- RScase0 [kN]				2. etaža -M2- RScase0 [kNm]				2. etaža -M3- RScase0 [kNm]							
34.818	60.129	60.129	34.818	31.941	31.941	31.941	31.941	52.305	52.305	52.305	52.305	57.215	96.401	96.401	57.215				
34.817	60.129	60.129	34.817	55.175	55.175	55.175	55.175	88.477	88.477	88.477	88.477	57.215	96.401	96.401	57.215				
34.817	60.129	60.129	34.817	54.006	54.006	54.006	54.006	86.557	86.557	86.557	86.557	57.215	96.401	96.401	57.215				
34.817	60.129	60.129	34.817	55.175	55.175	55.175	55.175	88.477	88.477	88.477	88.477	57.215	96.401	96.401	57.215				
34.818	60.129	60.129	34.818	31.941	31.941	31.941	31.941	52.305	52.305	52.305	52.305	57.215	96.401	96.401	57.215				
3. etaža -V2- RScase0 [kN]				3. etaža -V3- RScase0 [kN]				3. etaža -M2- RScase0 [kNm]				3. etaža -M3- RScase0 [kNm]							
26.708	45.908	45.908	26.708	24.578	24.577	24.577	24.578	50.39	50.39	50.39	50.39	53.029	81.489	81.489	53.029				
26.708	45.907	45.907	26.708	42.14	42.14	42.14	42.14	76.337	76.336	76.336	76.337	53.028	81.489	81.489	53.028				
26.708	45.907	45.907	26.708	41.573	41.573	41.573	41.573	75.609	75.609	75.609	75.609	53.028	81.489	81.489	53.028				
26.708	45.907	45.907	26.708	42.14	42.14	42.14	42.14	76.337	76.336	76.336	76.337	53.028	81.489	81.489	53.028				
26.708	45.908	45.908	26.708	24.578	24.577	24.577	24.578	50.39	50.39	50.39	50.39	53.029	81.489	81.489	53.029				
4. etaža -V2- RScase0 [kN]				4. etaža -V3- RScase0 [kN]				4. etaža -M2- RScase0 [kNm]				4. etaža -M3- RScase0 [kNm]							
12.89	27.419	27.419	12.89	11.478	11.478	11.478	11.478	29.425	29.425	29.425	29.425	31.141	54.317	54.317	31.141				
12.889	27.419	27.419	12.889	25.155	25.155	25.155	25.155	51.65	51.65	51.65	51.65	31.14	54.317	54.317	31.14				
12.889	27.419	27.419	12.889	24.617	24.617	24.617	24.617	50.637	50.637	50.637	50.637	31.14	54.317	54.317	31.14				
12.889	27.419	27.419	12.889	25.155	25.155	25.155	25.155	51.65	51.65	51.65	51.65	31.14	54.317	54.317	31.14				
12.89	27.419	27.419	12.89	11.478	11.478	11.478	11.478	29.425	29.425	29.425	29.425	31.141	54.317	54.317	31.141				

Slika 39: Primer izpisa prečnih sil in momentov v stebrih za vsako etažo

Na sliki 40 je prikazan osnovni diagram poteka algoritma programa.



Slika 40: Osnovni diagram poteka algoritma programa

4.4 Zaključni komentarji k programu

Z izdelavo programa smo izpolnili zastavljene cilje in pokazali, da je mogoče z uporabo programskega vmesnika prilagoditi vmesnik za delo s programom SAP2000 ter ustvariti namenska orodja, ki poenostavijo in avtomatizirajo procese načrtovanja, analize in obdelave rezultatov. Pokazali smo, kako vzpostaviti dvosmerno komunikacijo med programom Microsoft Excel in SAP2000 ter izdelali program, ki omogoča enostavno generiranje konstrukcij, določitev njihovih nihajnih časov in modalno analizo s spektri pospeškov ter izpis rezultatov. Program je mogoče prilagoditi lastnim potrebam ter ga v prihodnosti dopolniti s funkcijami kot so dodatna orodja za modifikacijo osnovnih okvirnih konstrukcij, možnost določitve različnih prečnih prerezov posameznim tipom elementov in dodatna obdelava rezultatov. V kolikor bi imeli na voljo ustrezne podatke o nosilnih konstrukcijah, lahko tako orodje služi za hitro oceno potresne ogroženosti širšega urbanega območja.

Za najtežavnejše pri izdelavi se je izkazalo pomanjkanje odprto kodne programske opreme, ki pri svojem delovanju uporablja programski vmesnik za SAP2000. S pomočjo te, bi lahko na obstoječih primerih hitreje in enostavneje osvojili znanje njegove uporabe. S tem programom zapolnjujemo tudi ta prazen prostor in nudimo odličen primer celovite implementacije programskega vmesnika. S komentarji opremljena programska koda bo olajšala sledenje logičnim procesom in omogočala hitro osvojitvev principov ter načinov njegove uporabe.

5 VTIČNIK ZA ANALIZO KONSTRUKCIJE NA OSNOVI POTRESNEGA SCENARIJA

5.1 Motivacija za izdelavo programa

Znanje, ki smo ga pridobili z izdelavo programa opisanega v četrtem poglavju, nam omogoča razvoj programske opreme, ki bo služila raziskovalnemu delu in potresni analizi zahtevnejših konstrukcij. Preučili smo dodatne funkcije programskega vmesnika, ki nudijo integracijo razvitih aplikacij v obstoječ grafični vmesnik programa SAP2000. Integracija zagotavlja boljšo uporabniško izkušnjo, hkrati pa eliminira potrebo po morebitni tretji programski opremi. Zastavljeni cilji so razširitev funkcionalnosti programa SAP2000 pri analizi časovnega odziva konstrukcij, integracija izdelanega programa v obstoječi grafični vmesnik, njegova preprosta uporaba in namestitvev ter vzpostavitev relacijske baze podatkov potresnega delovanja na spletnem strežniku fakultete. Izdelava slednje predstavlja predpogoj za delovanje programa, hkrati pa nudi možnost njene uporabe pri ostalih raziskavah in razvoju novih programskih rešitev.

5.2 Opis programa

Program je zasnovan kot vtičnik (angl. plug-in) za program SAP2000, kar pomeni, da je neposredno integriran v njegov obstoječi grafični vmesnik. To pogojuje uporabo angleškega jezika pri njegovi zasnovi. Ime programa v angleškem jeziku je 'Scenario Based Seismic Performance Assessment Plug-in' (v nadaljevanju SSPA). Na podlagi izbire seizmičnega scenarija omogoča pripravo analize, s pomočjo katere je mogoče oceniti potresno zmogljivost konstrukcije izpostavljene potresnemu delovanju, ki ustreza kriterijem podanega scenarija. Tega je mogoče definirati z uporabo dinamičnega uporabniškega vmesnika, preko katerega dostopamo do ustvarjene relacijske baze podatkov o potresnem delovanju, ki se nahaja na spletnem strežniku fakultete. Trenutna verzija programa omogoča hiter in zanesljiv prenos preko 7000 različnih funkcij akceleroگرامov, ki so potrebne za pripravo analize časovnega odziva konstrukcije. Poleg vmesnika za pripravo scenarijev, smo zasnovali še grafični vmesnik, preko katerega lahko na podlagi uvoženih akceleroگرامov in ostalih podanih parametrov, samodejno generiramo poljubno število obtežnih primerov za omenjeno analizo.

Implementacija baze podatkov o potresnem delovanju, skupaj z avtomatiziranim načinom generiranja obtežnih primerov, nudi nov pristop k analizi časovnega odziva konstrukcij in razširi funkcionalnost programa SAP2000.

5.2.1 Vzpostavitev baze akceleroگرامov na strežniku fakultete

Pred izdelavo programa je bilo potrebno poiskati ustrezno bazo podatkov, ki bi jo lahko uporabili v naš namen. Na spletu je na voljo kar nekaj takih baz, kjer so zbrani podatki o gibanju in pospeških tal zaradi potresnega delovanja. Našteli bomo le nekaj primerov.

The European Strong Motion Database (ESD) [36]

Na spletu se nahaja interaktivna relacijska baza, ki vsebuje več kot 3000 zapisov gibanja tal ter pripadajoče podatke o potresih in merilnih mestih s širšega evropskega območja. Od tega je približno 2000-im časovnim odzivom dodan tudi pripadajoči elastični spekter pospeškov. Podatke so prispevali tako posamezniki, kot izobraževalne in raziskovalne ustanove, med drugimi tudi Inštitut za konstrukcije, potresno inženirstvo in računalništvo (IKPIR), ki je del Fakultete za gradbeništvo in geodezijo Univerze v Ljubljani. Baza je bila nazadnje posodobljena v letu 2008.

Peer Strong Motion Database (PEER) [37]

Baza podatkov je nastala pod okriljem pacifiškega potresno inženirskega raziskovalnega centra s sedežem na Univerzi v Kaliforniji, Berkeley, ZDA. (angl. Pacific Earthquake Engineering Research Center, PEER).

Na spletu je bila prvič objavljena že leta 1999 v mnogo manjšem obsegu kot danes. Od takrat je doživela več posodobitev. Eno večjih kot del projekta *Next Generation of Ground-Motion Attenuation Models for the western United States (NGA West)*, ki je bil zaključen v letu 2008. V njej je zbranih več kot 3500 zapisov gibanja tal, čez 10500 akceleroogramov ter skoraj toliko spektrov pospeškov za več kot 170 potresov, opazovanih in zabeleženih iz približno 1600 merilnih mest. Trenutno potekata dva nova projekta, *NGA East* in *NGA West 2*. Predviden zaključek slednjega je ob koncu leta 2013, v okviru katerega bo razširjena tudi baza podatkov. V letu 2010 je bil prenovljen interaktivni spletni vmesnik za iskanje in prenos podatkov iz baze.

Center for Engineering Strong Motion Data (CESMD) [38]

Center je bil ustanovljen s strani ameriškega geološkega inštituta (angl. US Geological Survey, USGS) in kalifornijskega geološkega inštituta (angl. California Geological Survey, CGS) z namenom združitve baz podatkov o gibanju tal, ki so nastale v okviru sledečih organizacij in projektov: *CGS California Strong Motion Instrumentation Program*, *the USGS National Strong Motion Project* in *the Advanced National Seismic System (ANSS)*.

Italian Accelerometric Archive (ITACA) [39]

Baza, ki se nahaja na strani italijanskega nacionalnega inštituta za geofiziko in vulkanologijo (ita. Istituto Nazionale di Geofisica e Vulcanologia) vsebuje podatke o gibanju tal, pridobljene preko italijanske nacionalne seizmografske mreže. Trenutna verzija zajema podatke o potresih med letom 1972 in 2011. Predvidena posodobitev je načrtovana v mesecu decembru 2013, kjer bodo dodani podatki do leta 2013.

Swiss Seismological Service [40]

Baza je nastala pod okriljem švicarske nacionalne agencije za opazovanje potresov (nem. Schweizerischer Erdbebendienst, ETH Zürich, SED). Na voljo so podatki zbrani iz približno stotih merilnih mest. Večina teh se nahaja v Švici in so del nacionalne opazovalne mreže, nekaj pa se jih nahaja v bližnji regiji.

GeoNet New Zealand [41]

Cilj projekta GeoNet je vzpostaviti moderni opazovalni sistem geološke ogroženosti Nove Zelandije, ki zajema zaznavo in obdelavo podatkov o potresih, zemeljskih plazovih, tsunamijih in vulkanskem delovanju. V okviru projekta je tako nastala tudi nacionalna baza podatkov o gibanju tal zaradi potresnega delovanja, ki je med drugimi prosto dostopnimi podatki razpoložljiva na spletu.

NIED strong-motion seismograph networks Japan [42]

Bazo tvorijo podatki pridobljeni iz dveh nacionalnih mrež merilnih postaj. Mreže K-NET (angl. Kyoshin network), ki jo sestavlja več kot 1000 merilnih postaj in so enakomerno razporejene po celotni Japonski na medsebojnih razdaljah 20 km, ter mreže KiK-net (angl. Kiban Kyoshin network), ki jo na več kot 700 lokacijah sestavljajo pari seizmografov. Pri tem se en seizmograf nahaja na zemeljskem površju, drugi pa v geološki vrtini. Za obe mreži skrbi nacionalni raziskovalni inštitut geoloških znanosti in preprečevanje naravnih nezgod (angl. National Research Institute for Earth Science and Disaster Prevention, NIED).

Podatki iz omenjenih baz so prosto dostopni in brezplačni. V splošnem pogoji uporabe zahtevajo le, da jasno navedemo kje smo jih pridobili. Za uporabo nekaterih naštetih storitev je potrebna brezplačna registracija. Vsi našteti spletni vmesniki za prenos podatkov temeljijo na principu podajanja kriterijev, na podlagi katerih napravimo poizvedbo v bazi. Izpišejo se nam vsi zapisi, ki ustrezajo podanim zahtevam. V naslednjem koraku izberemo rezultate in jih prenesemo na disk, kjer jih nato uredimo v skladu z našimi potrebami.

Nobena od naštetih storitev nima javno dostopnega programskega vmesnika (angl. API), ki bi na preprost način omogočal interakcijo obstoječe baze z drugo programsko opremo ali spletno storitvijo. Tako je njihova uporaba v te namene praktično nemogoča, saj nimamo nobenega nadzora in možnosti prilagajanja vhodnih poizvedbenih kriterijev in načina ter oblike vrnjenih rezultatov iz baze podatkov. Preostala nam je torej le možnost, da izberemo tako bazo podatkov, ki ustreza našim potrebam in jo poskusimo v celoti prenesti na strežnik Fakultete za gradbeništvo in geodezijo. V tem primeru tako dobimo dostop do vseh podatkov, sami pa lahko napišemo programski vmesnik, ki bo omogočal uporabo te baze v poljubno razviti programski opremi.

Ne glede na izbrano bazo je ročen prenos podatkov preko obstoječih spletnih vmesnikov zaradi velikega obsega praktično nemogoč. Prav tako na ta način ni mogoče prenesti vseh podatkov, ki so potrebni za vzpostavitev ustrezne relacijske baze na strežniku fakultete.

Glede na dva glavna kriterija in sicer, da baza podatkov vsebuje predvsem zapise potresov močnejših magnitud in da je te podatke mogoče prenesti v taki obliki ter obsegu, ki omogoča vzpostavitev kopije baze podatkov s primerljivo funkcionalnostjo, smo za prenos izbrali bazo pacifiškega potresno inženirskega raziskovalnega centra (v nadaljevanju, baza podatkov PEER).

Prenos baze podatkov PEER je potekal v treh fazah. V prvi smo prenesli vse metapodatke, ki so združeni v Excelovi datoteki, ki je objavljena na spletnem naslovu [43]. Tu je za vseh 3551 posameznih seizmoloških meritev zbranih 129 lastnosti, ki med drugimi zajemajo podatke o zabeleženem potresu (ime, magnituda, geografska lokacija in globina žarišča, tip preloma idr.), podatke o merilnem mestu, tipu tal, razdaljah in druge podatke. Vse lastnosti so našteje in opisane v dokumentaciji [44].

Druga faza je bila časovno ter količinsko najbolj obsežna in je zajemala prenos akceleroگرامov oziroma zapisov digitalnih meritev pospeškov tal zaradi močnega potresnega delovanja. Vsako, od skupno 3551. meritev, sestavljajo trije akceleroگرامi, od katerih dva odražata časovni potek pospeškov tal merjenih v dveh medsebojno pravokotnih smereh horizontalne ravnine, tretji pa odraža pospeške tal merjene v vertikalni ravnini. Vsak akceleroگرام je na strežniku Univerze v Berkeleyu shranjen v tekstovni datoteki, katerih imena so navedena v tabeli [43], ki smo jo prenesli v prvi fazi.

Spletni naslovi akceleroگرامov so sestavljeni iz naslova spletnega strežnika, ki je enak za vse akceleroگرامe in imena datoteke, ki je tvorjeno iz imena potresa ter oznake merilnega mesta. Primer oblike spletnega naslova je prikazan na sliki 41.

http://peer.berkeley.edu/nga_files/ath/CHICHI/CHY116-N.AT2
└──────────────────────────────────┘
└──────────────────────────────────┘
naslov strežnika
ime datoteke

Slika 41: Oblika spletnih naslovov akceleroگرامov na strežniku Univerze v Berkeleyu

Poznavanje metodologije shranjevanja datotek nam je omogočilo, da smo lahko proces prenašanja posameznih akceleroگرامov avtomatizirali. V programu Matlab smo napisali krajši program, s pomočjo katerega so bili na lokalni računalnik preneseni vsi razpoložljivi akceleroگرامi. V Prilogi B je objavljen samodejno generiran izpis programa, iz katerega je razvidno, katerih datotek ni bilo mogoče prenesti. Vzrok za to je lahko, da ti akceleroگرامi dejansko ne obstajajo ali pa je njihovo ime drugačno od navedenega v tabeli [43]. Število neuspelo prenesenih datotek je 80 od skupno 10653.

Vsaka prenesena datoteka ima glavo, prikazano na sliki 42, v kateri je navedenih nekaj osnovnih podatkov in sicer: ime potresa ter merilnega mesta, datum registracije potresa, število zabeleženih

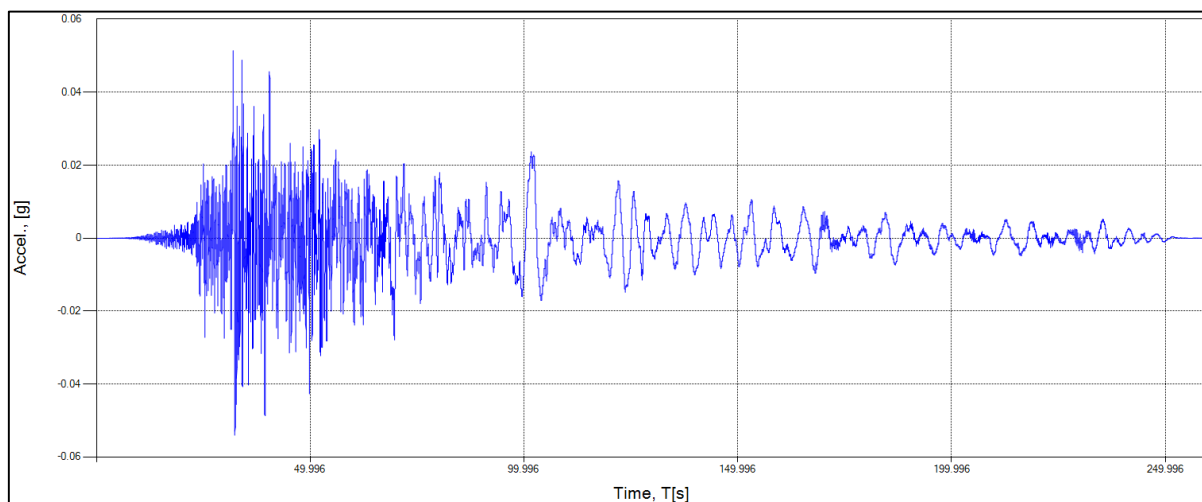
vrednosti pospeškov tal v enotah gravitacijskega pospeška in njihov medsebojni časovni korak, ki je odvisen od seizmografa. Slednje dva podatka smo s pomočjo napisanega programa v okolju Matlab izluščili iz vsake datoteke in ju dodali v Excelovo datoteko [43], ki smo jo prenesli v prvi fazi. Potrebovali ju bomo pri pripravi analize časovnega odziva v programu SAP2000. Glave smo nato izbrisali in s tem prihranili tudi nekaj prostora, podatke pa iz petih prenesli v en stolpec. Slednje smo prav tako storili s pomočjo namensko napisanega programa v okolju Matlab. Tako obdelane datoteke so bile pripravljene za prenos na strežnik.

Dolžine akceleroگرامov so zelo različne in se gibljejo od nekaj sekund do nekaj minut. Povprečno trajanje meritve je dolgo približno 90 sekund in zajema okoli 20.000 zabeleženih vrednosti pospeškov tal. Skupna velikost vseh obdelanih datotek akceleroگرامov je približno 1,6 GB.

```
1 PEER NGA STRONG MOTION DATABASE RECORD
2 CHI-CHI 09/20/99, CHY116, N
3 ACCELERATION TIME HISTORY IN UNITS OF G
4 65000 0.0040 NPTS, DT
5 -0.770940E-04 -0.770163E-04 -0.769378E-04 -0.768652E-04 -0.767906E-04
6 -0.767132E-04 -0.766312E-04 -0.765490E-04 -0.764657E-04 -0.763848E-04
7 -0.763031E-04 -0.762235E-04 -0.761395E-04 -0.760584E-04 -0.759764E-04
```

Slika 42: Glava in prvi 2. od 13000-ih vrstic zabeleženih pospeškov prenesene datoteke akceleroگرامa, zabeleženega na merilnem mestu CHY116 v eni izmed horizontalnih smeri potresa CHI-CHI, Taiwan

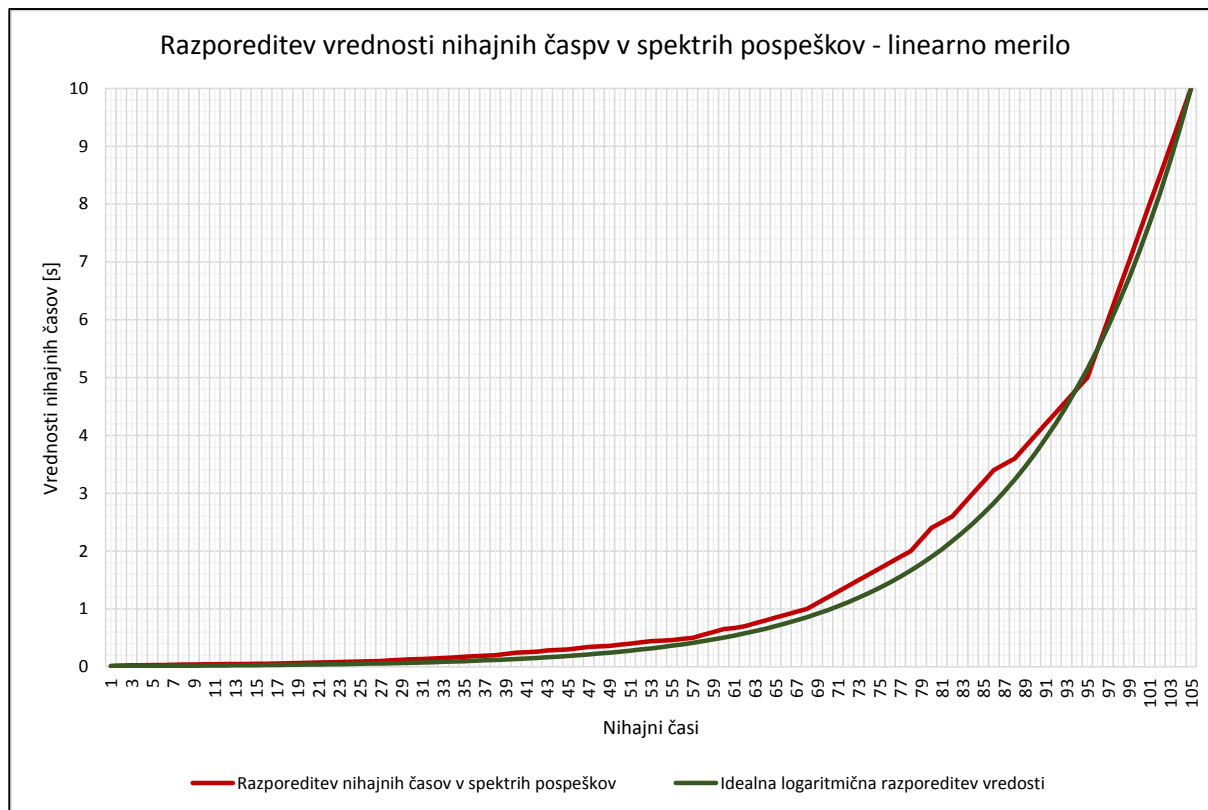
Primer grafičnega prikaza akceleroگرامa je prikazan na sliki 43.



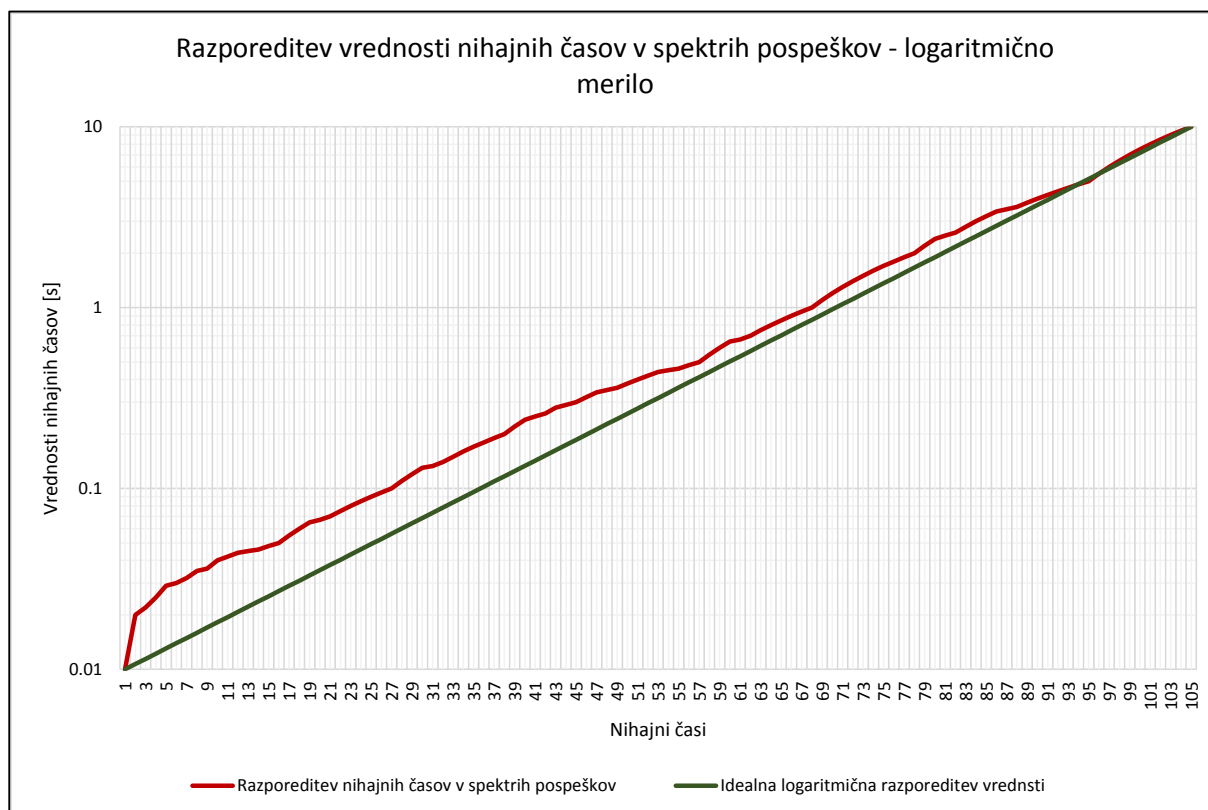
Slika 43: Grafičen prikaz celotnega akceleroگرامa, zabeleženega na merilnem mestu CHY116 v eni izmed horizontalnih smeri, potresa CHI-CHI, Taiwan

Vsakemu akceleroگرامu pripada še spekter pospeškov, ki smo jih pridobili v zadnji fazi. V sklopu raziskovalne skupine profesorja Jacka W. Bakerja [45], ki deluje na Univerzi Stanford v Združenih Državah Amerike in se ukvarja z razvojem in uporabo probabilističnih ter statističnih metod za oceno tveganja poškodb objektov zaradi potresnega delovanja, so bile razvite aplikacije, v okviru katerih so v programu Matlab že shranjeni spektri pospeškov baze podatkov PEER. Omenjena programska oprema, vključno z bazo pripadajočih podatkov, se nahaja na spletnem naslovu [46]. Spektre pospeškov smo izvozili in jih shranili v tekstovne datoteke. Vrednosti pospeškov so izračunane za 105

niha jnih časov, katerih vrednosti so približno logaritm ično razporejene na intervalu od 0,01 s do 10 s, kar je prikazano na sliki 44 in sliki 45.

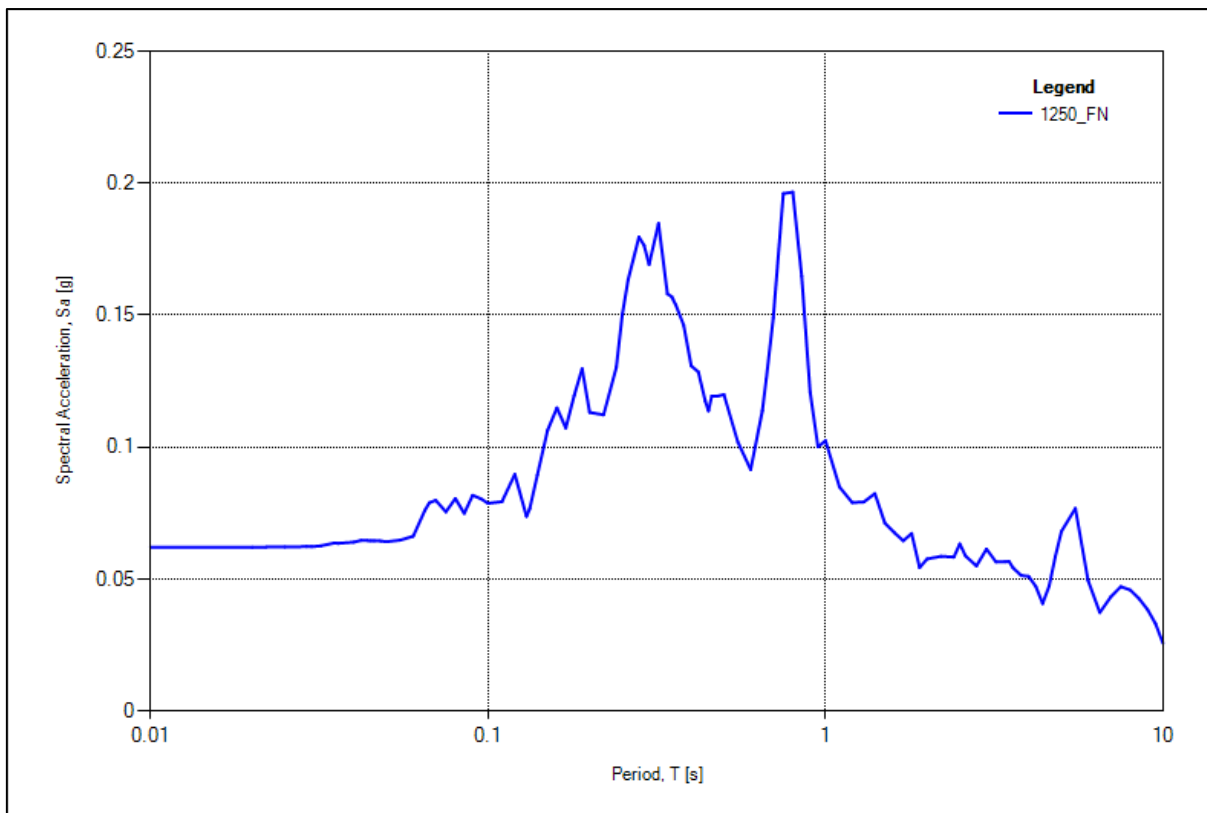


Slika 44: Razporeditev vrednosti niha jnih časov v spektrih pospeškov (linearno merilo)



Slika 45: Razporeditev vrednosti niha jnih časov v spektrih pospeškov (logaritm ično merilo)

Primer grafičnega prikaza spektra pospeškov izračunanega na podlagi akceleroograma iz slike 43 je prikazan na sliki 46.



Slika 46: Grafičen prikaz spektra pospeškov, izračunanega na podlagi akceleroograma, zabeleženega na merilnem mestu CHY116 v eni izmed horizontalnih smeri, potresa CHI-CHI, Taiwan

Vse zbrane podatke smo nato prenesli na spletni strežnik fakultete in ustvarili relacijsko podatkovno bazo v za to posebej namenjenem sistemu za upravljanje z relacijskimi bazami MySQL [19]. Poleg tega je bilo potrebno napisati še namenski program skupaj s programskim vmesnikom [18], ki prav tako teče na strežniku fakultete. Napisan je v programskih jezikih PHP in SQL ter skrbi za komunikacijo med bazo podatkov in programsko opremo, ki do te baze dostopa. Uporabniki torej do baze nimajo neposrednega dostopa, pač pa do podatkov lahko dostopajo le na način, ki ga narekuje programski vmesnik omenjenega programa. Le tega je mogoče kadar koli dopolniti in prilagoditi drugim potrebam, kar pomeni, da vzpostavljena kopija baze podatkov PEER nudi prost dostop in prilagodljiv način ter vrsto vrnjenih podatkov. To omogoča njeno integracijo v najrazličnejšo programsko opremo. Bazo je mogoče v prihodnosti tudi razširiti, pri tem pa je potrebno paziti na strukturo podatkov, ki se mora ujemati s strukturo obstoječe relacijske podatkovne baze.

5.2.2 Izdelava programa

Program smo napisali v razvojnem okolju Microsoft Visual Studio [47] ter pri tem, poleg že vgrajenih knjižnic v ogrodje Microsoft Framework .NET, za celotno funkcionalnost programa uporabili še knjižnice našteje in opisane v preglednici 2. Program je napisan v programskem jeziku Visual Basic .NET in deloma v programskem jeziku XAML. Za implementacijo zemljevidov smo uporabili

Microsoftovo storitev Bing Maps [48] ter program Microsoft Expression Blend [49], s katerim smo zasnovali grafični vmesnik za njihovo upravljanje.

Preglednica 2: Uporabljene knjižnice, ki niso vgrajene v ogrodje Microsoft .NET Framework

Dinamična knjižnica	Opis
SAP2000v15.dll	Dinamična knjižnica je del programa SAP2000 v kateri so zbrane funkcije in podatki za upravljanje s programom, do katerih dostopamo preko programskega vmesnika.
Newtonsoft.Json.dll	Dinamična knjižnica je del ogrodja JSON.NET [50], ki je namenjeno uporabi v ogrodju Microsoft Framework .NET. Služi za obdelavo, generiranje in pretvarjanje strukturiranih podatkov v formatu JSON.
Ionic.Zip.dll	V dinamično knjižnico DotNetZip [51] so vgrajene metode za delo s stisnjenimi datotekami formata ZIP.
Microsoft.Maps.MapControl.WPF.dll	Dinamična knjižnica je del razvojnega programskega paketa (angl. Software Development Kit, SDK) z imenom 'Bing Maps Windows Presentation Foundation (WPF) Control, Version 1.0' [52], ki združuje orodja za upravljanje in implementacijo zemljevidov storitve 'Microsoft Bing Maps' v klasične programe.
RichTextBoxLinks.dll	Dinamična knjižnica predstavlja .NET komponento, ki omogoča povezovanje teksta in spletnih naslovov [53].
MRG.Controls.UI.dll	Dinamična knjižnica predstavlja .NET komponento za grafično animacijo, ki služi kot povratna informacija uporabniku o stanju izvajanja nalog programa [54].

Sam program je zasnovan kot vtičnik in zato preveden v dinamično povezovalno knjižnico. Aplikacije torej ni mogoče samostojno zagnati in uporabljati neodvisno od ostale programske opreme, ampak jo lahko uporabljamo le znotraj programa SAP2000 verzije 15.0.0 ali novejše.

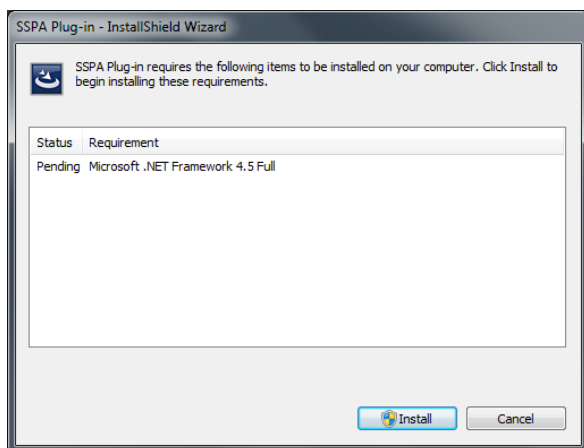
Da bi uporabniku olajšali postopek ročne namestitve, kjer je za delovanje programa v operacijskem sistemu Microsoft Windows potrebna registracija dinamičnih knjižnic preko konzolnega okna, smo za ta proces zasnovali namenski grafični vmesnik. Ta uporabnika vodi skozi cel proces namestitve vtičnika ter samodejno prekopira, shrani in registrira potrebne datoteke za delovanje programa.

5.3 Navodila za uporabo programa SSPA verzije 1.0

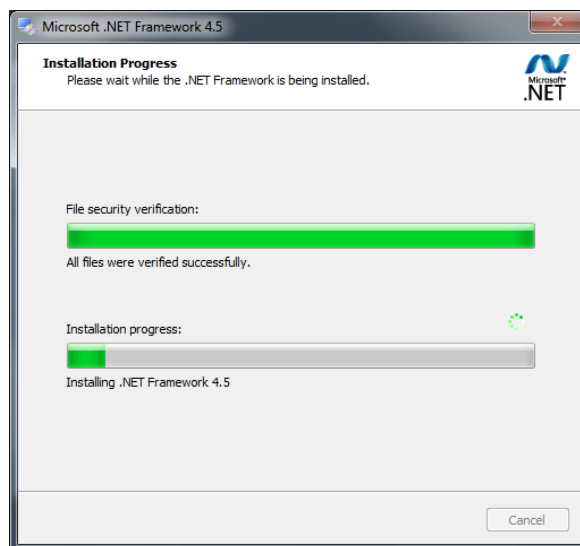
5.3.1 Zahteve in namestitve

Za delovanje vtičnika je potrebno imeti nameščen program SAP2000 verzije 15.0.0 ali novejše ter programsko ogrodje Microsoft Framework verzije 4.5 [55] ali novejše. V kolikor uporabljamo operacijski sistem Microsoft Windows 8, je slednji že nameščen, v nasprotnem primeru pa se namesti med postopkom inštalacije vtičnika. Ker vtičnik temelji na pridobitvi podatkov iz baze potresnega delovanja, ki se nahaja na oddaljenem strežniku Fakultete za gradbeništvo in geodezijo Univerze v Ljubljani, je za njegovo uporabo potrebna tudi internetna povezava.

Pred namestitvijo vtičnika, ki jo pričnemo z dvojnim klikom na datoteko 'setup.exe', ki se nahaja v mapi z imenom 'SSPA_Plugin', je potrebno zapreti program SAP2000. V primeru, da na računalniku ni nameščenega ogrodja Microsoft .Net Framework 4.5, se ta namesti pred začetkom namestitve vtičnika SSPA, kar prikazujeta slika 47 in slika 48. Namestitev vtičnika se samodejno nadaljuje po ponovnem zagonu računalnika.

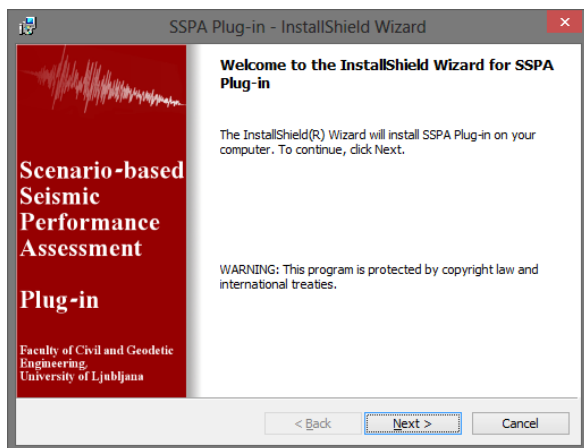


Slika 47: Prvo pogovorno okno namestitve ogrodja Microsoft Framework 4.5

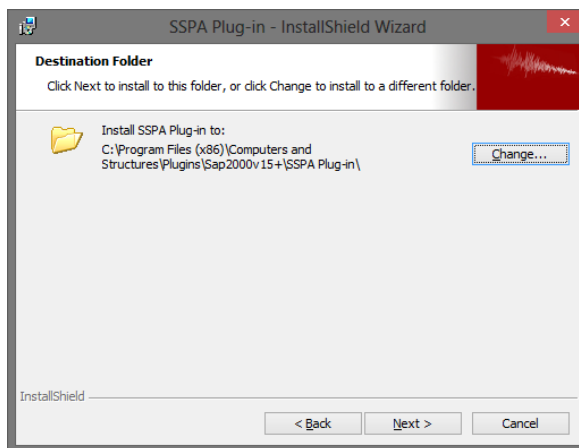


Slika 48: Drugo pogovorno okno namestitve ogrodja Microsoft Framework 4.5

Namestitev vtičnika SSPA se prične s pozdravnim pogovornim oknom prikazanim na sliki 49, v katerem za nadaljevanje pritisnemo na gumb 'Next'. V drugem pogovornem oknu, prikazanem na sliki 50, lahko spremenimo mesto na disku, kamor se bodo shranile datoteke programa. Mesto na disku kamor namestimo program je poljubno. Za nadaljevanje kliknemo na gumb 'Next'.

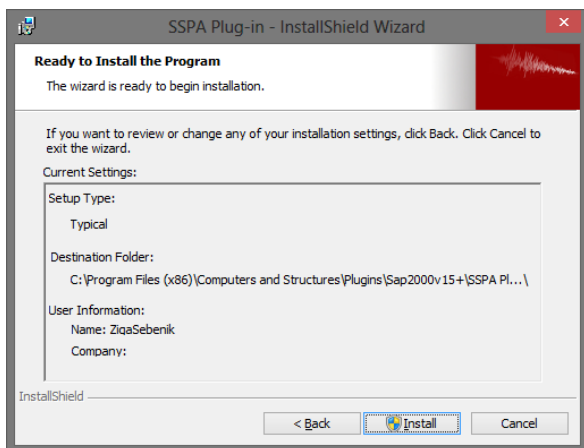


Slika 49: Prvo pogovorno okno namestitve vtičnika

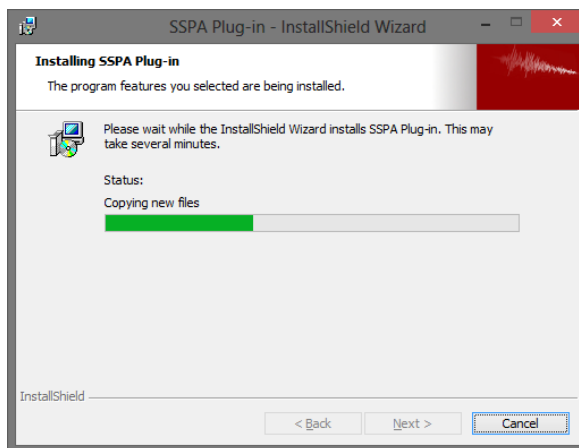


Slika 50: Drugo pogovorno okno namestitve vtičnika

V tretjem pogovornem oknu na sliki 51, še enkrat pregledamo konfiguracijo ter pričnemo z namestitvijo vtičnika s pritiskom na gumb 'Install'. Odpre se četrto pogovorno okno, v katerem lahko spremljamo potek namestitve, kar prikazuje slika 52.

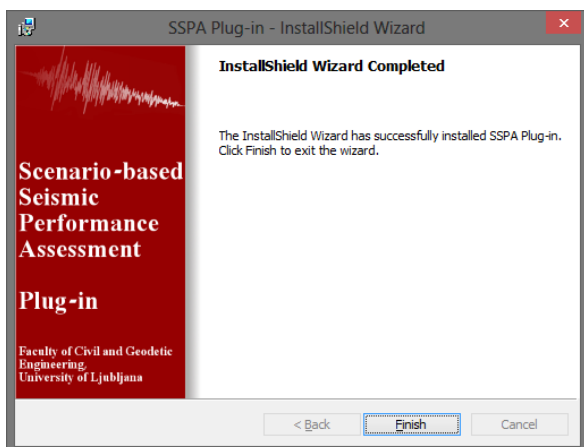


Slika 51: Tretje pogovorno okno namestitve vtičnika



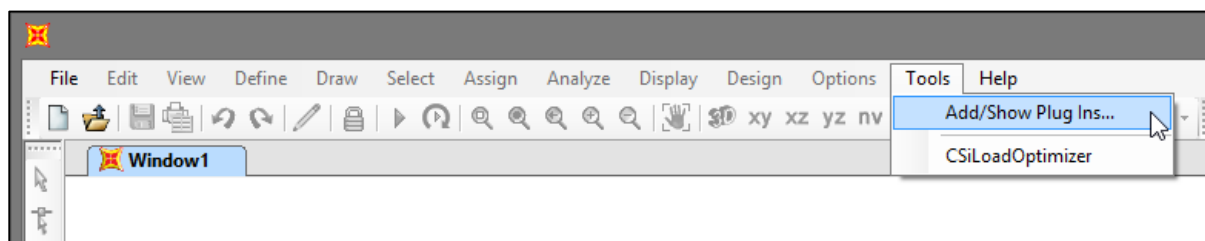
Slika 52: Četrto pogovorno okno namestitve vtičnika

Po uspešno nameščenem programu, se odpre peto pogovorno okno, prikazano na sliki 53, v katerem s pritiskom na gumb 'Finish' zaključimo proces namestitve.



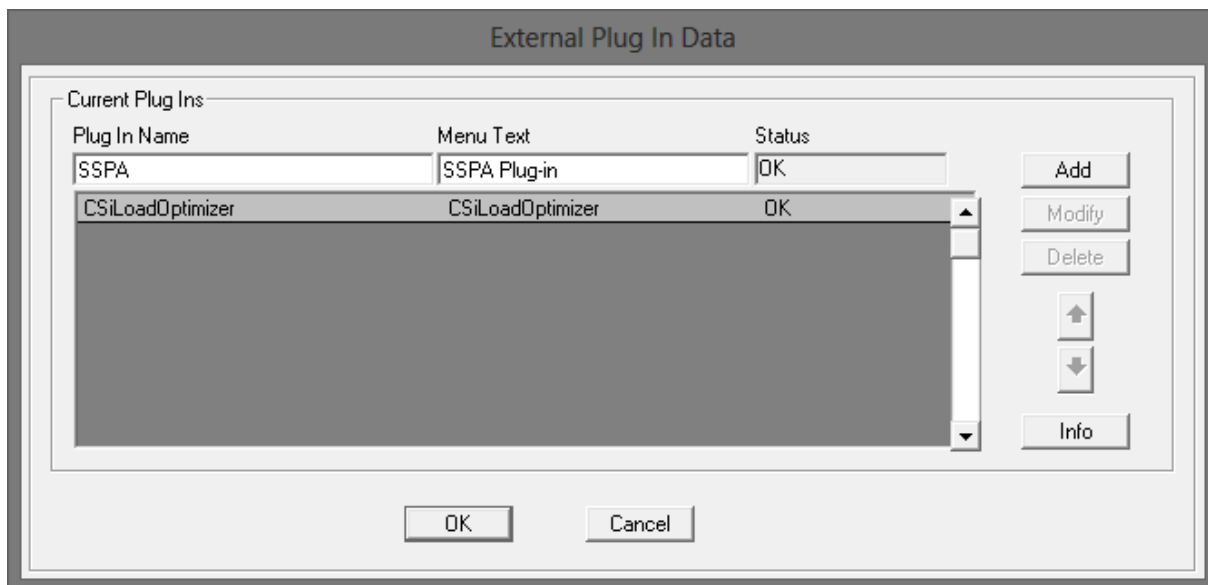
Slika 53: Peto pogovorno okno namestitve vtičnika

Vtičnik je tako uspešno nameščen na računalniku, da lahko do njega dostopamo preko programa SAP2000, pa je potrebno storiti naslednje. Odpremo program SAP2000 in v orodni vrstici, prikazani na sliki 54, kliknemo na meni 'Tools' ter nato na gumb 'Add/Show Plug Ins...!.



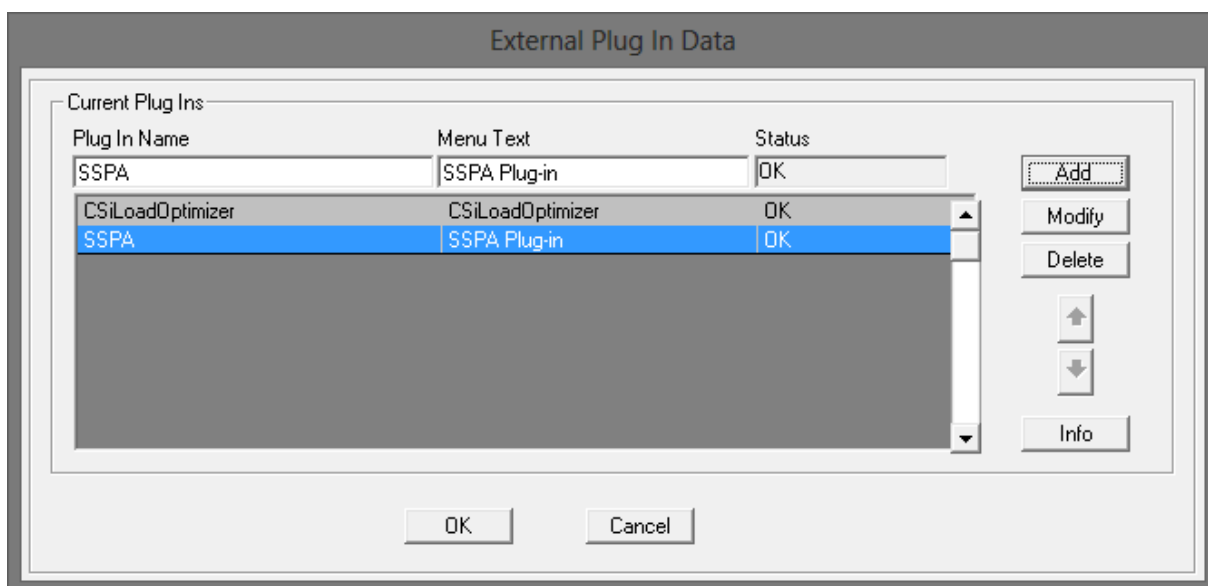
Slika 54: Prikaz odpiranja okna za dodajanje vtičnikov preko menija 'Tools' v programu SAP2000

Odpre se pogovorno okno na sliki 55, kjer lahko dodamo ali odstranimo nameščene vtičnike, do katerih dostopamo preko menija v programu SAP2000. V polje 'Plug In Name' je potrebno vpisati ime dinamične knjižnice in sicer SSPA, v polje 'Menu Text' pa lahko vpišemo poljuben tekst, ki predstavlja ime novega gumba preko katerega bomo dostopali do vtičnika. Po izpolnitvi obeh polj, kliknemo na gumb 'Add'.



Slika 55: Okno preko katerega v programu SAP2000 dodamo ali odstranimo nameščene vtičnike

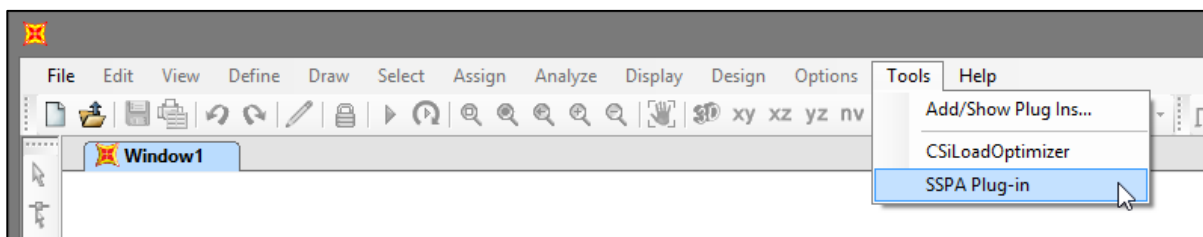
Na sliki 56 je prikazan uspešno nameščen vtičnik, kar označuje tekst 'OK' v polju 'Status'. Pogovorno okno lahko sedaj zapremo.



Slika 56: Prikaz uspešno nameščenega vtičnika v programu SAP2000

5.3.2 Odpiranje vtičnika

Pred uporabo vtičnika je priporočljivo shraniti trenutni model, da ne bi prišlo do morebitne izgube podatkov. Na sliki 57 je prikazano, kako odpremo vtičnik SSPA preko menija 'Tools' in klikom na gumb z imenom, ki smo ga definirali pri njegovi namestitvi v program SAP2000.



Slika 57: Prikaz novo ustvarjenega gumba v meniju 'Tools', s katerim odpremo uspešno nameščen vtičnik SSPA

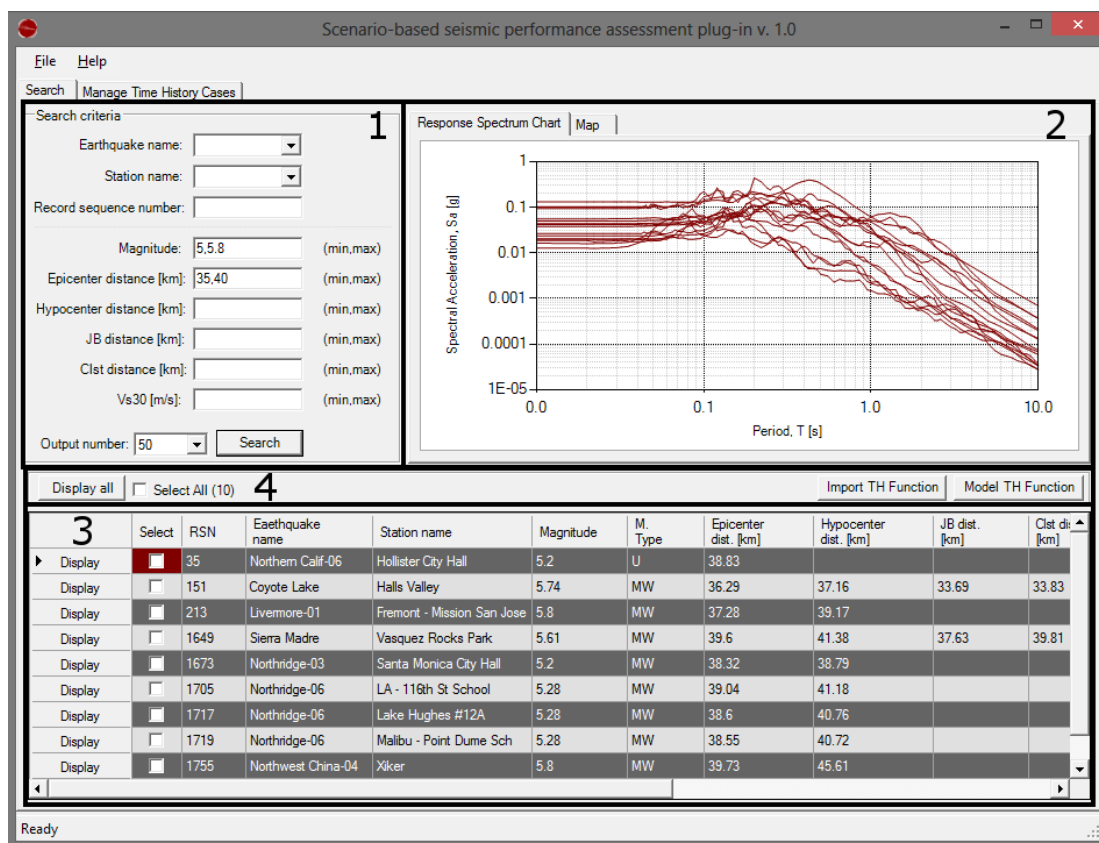
Nato se pojavi pozdravno okno vtičnika prikazano na sliki 58. Ta na zaslonu ostane nekaj sekund, medtem pa program preveri povezavo z oddaljenim strežnikom, kjer se nahaja baza podatkov potresnega delovanja in naloži grafični vmesnik za upravljanje s programom.



Slika 58: Pozdravno okno vtičnika

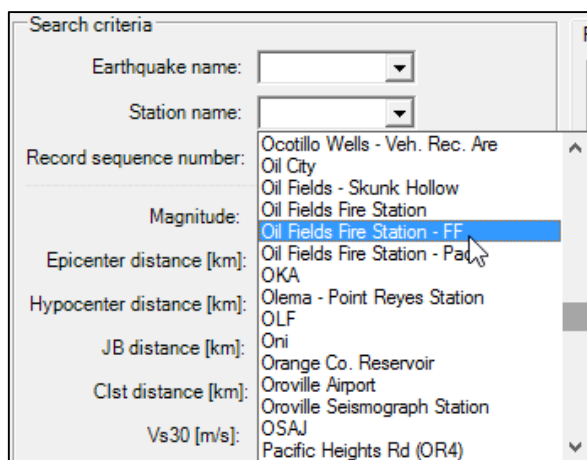
5.3.3 Iskanje v bazi podatkov potresnega delovanja

Zavihek z imenom 'Search' na sliki 59 sestavljajo štirje moduli. Prvi modul služi za podajanje kriterijev, na podlagi katerih poiščemo ustrezne podatke o gibanju tal, drugi in tretji modul služita za grafični in tabelarni prikaz podatkov, v četrtem modulu pa upravljamo s trenutnim prikazom podatkov in uvozom akceleroگرامov v program SAP2000.

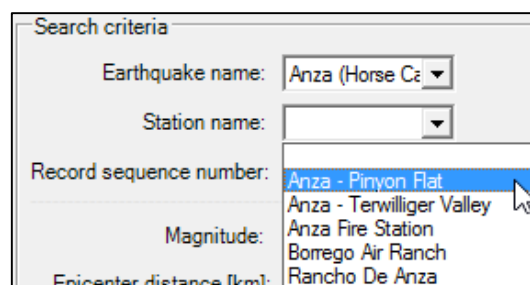


Slika 59: Prikaz zavihka z imenom 'Search', razdeljenega na štiri osnovne module

V modulu 1 je možno iskanje v bazi potresnega delovanja na podlagi devetih kriterijev. Prvi trije omogočajo iskanje glede na ime potresa (angl. Earthquake name), ime merilnega mesta (angl. Station name) in zaporedno številko meritve v bazi (angl. Record sequence number). Iskanje po omenjenih kriterijih je uporabno predvsem takrat, ko želimo poiskati točno določeno meritev. Ime potresa ali merilnega mesta izberemo iz ponujenega seznama, na katerem so zbrana vsa imena iz baze podatkov – slika 60. V kolikor izberemo ime potresa, se seznam z imeni merilnih mest osveži in je tako mogoče iz njega izbrati le merilna mesta, ki so zaznala izbran potres – slika 61.



Slika 60: Primer izbire merilnega mesta iz ponujenega seznama



Slika 61: Primer izbire merilnega mesta iz skrčenega seznama v primeru predhodno izbranega potresa

V polje z zaporedno številko meritve lahko vpišemo eno samo ali več celih števil večjih od nič. Vnos števil je mogoč na več načinov: (1) vpišemo lahko eno samo število, npr. '105', (2) vpišemo lahko več števil ločenih z vejico ali presledkom, npr. '105,106,1345', (3) vpišemo lahko območje števil ločeno z dvopičjem, npr. '345:365', (4) vpišemo lahko kombinacijo načina (2) in (3) npr. '786,890,1200:1220'. Morebitne podvojene vrednosti bodo avtomatsko odstranjene. Trenutno število meritev v bazi je 3551.

V naslednjih šestih poljih prvega modula lahko za omejitev iskanja podamo dodatne kriterije, ki so opisani v preglednici 3.

Preglednica 3: Dodatni kriteriji za omejitev iskanja v bazi podatkov potresnega delovanja

Ime kriterija v programu	Opis
Magnitude	Magnituda potresa je mera za količino sproščene energije med potresom.
Epicentre distance [km]	Razdalja med merilnim mestom in epicentrom potresa, ki predstavlja točko na zemeljskem površju nad žariščem potresa.
Hypocentre distance [km]	Razdalja med merilnim mestom in žariščem potresa.
JB distance [km]	Joyner-Boore razdalja je najkrajša horizontalna razdalja med merilnim mestom in vertikalno projekcijo preloma.
Clst distance [km]	Najkrajša razdalja med merilnim mestom in prelomljenim območjem.
Vs30 [m/s]	Povprečna vrednost hitrosti strižnega valovanja v zgornjih 30-ih metrih tal, na podlagi katere so opredeljena tla na lokaciji.

Opredelitev tal v razrede, skladno s [34], na podlagi vrednosti Vs30, je prikazana v preglednici 4.

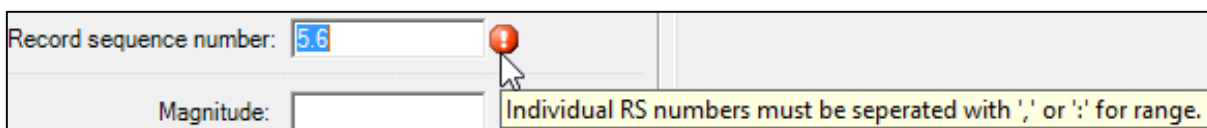
Preglednica 4: Tip tal, določen na podlagi vrednosti Vs30

Tip tal	Opis stratigrafskega profila	Vs30 [m/s]
A	Skala ali druga skali podobna geološka formacija, na kateri je največ 5 m slabšega površinskega materiala	> 800
B	Zelo gost pesek, prod ali zelo toga glina, debeline vsaj nekaj 10 m, pri kateri mehanske značilnosti z globino postopoma naraščajo	360-800
C	Globoki sedimenti gostega ali srednje gostega peska, prod ali toge gline globine nekaj deset do več sto metrov	180-360
D	Sedimenti rahlih do srednje gostih nevezljivih zemljin (z nekaj mehкими vezljivimi plastmi ali brez njih) ali pretežno mehkih do trdnih vezljivih zemljin	< 180
E	Profil tal, kjer je površinska aluvialna plast z debelino med okrog 5 in 20 metri in vrednostmi Vs, ki ustrezajo tipoma C ali D, leži na bolj togem materialu z Vs > 800 m/s	
S1	Sedimenti, ki vsebujejo najmanj 10 m debele plasti mehke gline/melja z visokim indeksom plastičnosti (PI>40) in visoko vsebnostjo vode	< 100 (indikativno)
S2	Tla, podvržena likvefakciji, občutljive gline ali drugi profili tal, ki niso vključeni v tipe A-E ali S1	

V vseh šest dodatnih polj podajamo kriterije za iskanje v obliki območja med minimalno in maksimalno vrednostjo, ki sta ločeni z vejico ali presledkom. Kot decimalno ločilo uporabljamo piko. Za omejitev območja magnitude lahko na primer vnesemo '6.8,7.2'. V kolikor bi želeli omejiti kriterij

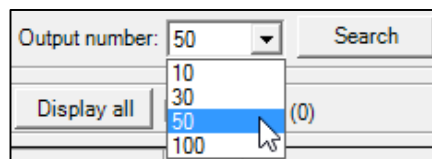
na točno določeno vrednost, podamo enako vrednost za maksimum in minimum območja. To lahko dosežemo tudi, če v polje podamo eno samo vrednost kriterija.

Za iskanje v bazi podatkov o potresnem delovanju mora biti izpolnjeno najmanj eno izmed devetih polj, ostala pa so lahko prazna. V kolikor so kriteriji vneseni v več polj, se med posameznimi predpostavi konjunkcija oziroma logična operacija 'IN'. Tako dobimo iz baze samo podatke, ki ustrezajo vsem vnesenim kriterijem. V kolikor v katero koli polje vnesemo kriterij v napačni obliki, nas program na to opozori in prikaže pravilno obliko. Primer opozorila zaradi nepravilne oblike vnosa podatkov je prikazan na sliki 62.



Slika 62: Opozorilo zaradi napačne oblike vnesenega kriterija, kjer je za ločilo dveh števil namesto vejice uporabljena pika

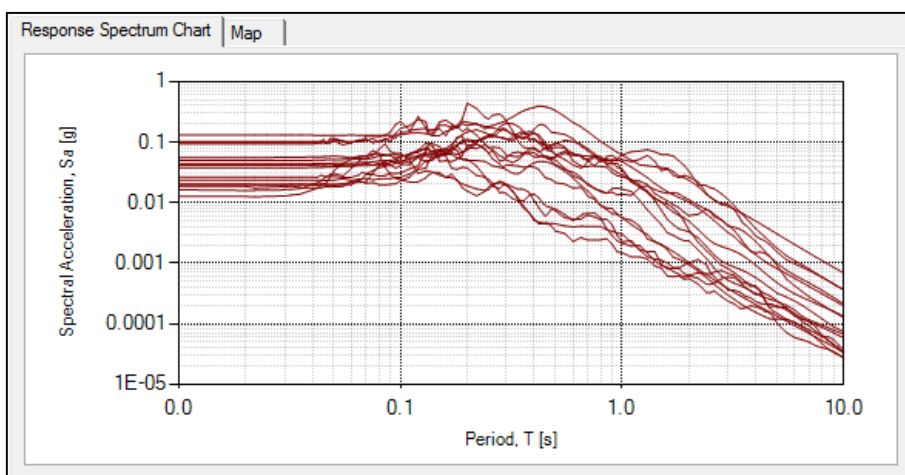
Iskanje pričnemo s klikom na gumb 'Search'. Pred-nastavljeno maksimalno število vrnjenih rezultatov je omejeno na 50, lahko pa ga spremenimo tako, da pred iskanjem izberemo eno izmed vrednosti iz seznama 'Output number' (10, 30, 50, 100), prikazanega na sliki 63.



Slika 63: Določitev maksimalnega števila vrnjenih rezultatov

5.3.4 Pregled rezultatov in uvoz izbranih akceleroگرامov v program SAP2000

Rezultati iskanja, ki ustrezajo iskanim kriterijem so prikazani na grafu (slika 64), tabeli (slika 65) in zemljevidu (slika 66). Na grafu se izrišejo spektri pospeškov vseh meritev, kar je prikazano na sliki 64.



Slika 64: Primer grafa spektrov pospeškov vseh meritev, ki ustrezajo kriterijem iskanja

Drugi podatki o posameznih meritvah so zapisani v tabeli, kot prikazuje slika 65.

	Select	RSN	Earthquake name	Station name	Magnitude	M. Type	Epicenter dist. [km]	Hypocenter dist. [km]	JB dist. [km]	Clst di- [km]
Display	<input checked="" type="checkbox"/>	35	Northern Calif-06	Hollister City Hall	5.2	U	38.83			
Display	<input type="checkbox"/>	151	Coyote Lake	Halls Valley	5.74	MW	36.29	37.16	33.69	33.83
Display	<input type="checkbox"/>	213	Livermore-01	Fremont - Mission San Jose	5.8	MW	37.28	39.17		
Display	<input type="checkbox"/>	1649	Sierra Madre	Vasquez Rocks Park	5.61	MW	39.6	41.38	37.63	39.81
Display	<input type="checkbox"/>	1673	Northridge-03	Santa Monica City Hall	5.2	MW	38.32	38.79		
Display	<input type="checkbox"/>	1705	Northridge-06	LA - 118th St School	5.28	MW	39.04	41.18		
Display	<input type="checkbox"/>	1717	Northridge-06	Lake Hughes #12A	5.28	MW	38.6	40.76		
Display	<input type="checkbox"/>	1719	Northridge-06	Malibu - Point Dume Sch	5.28	MW	38.55	40.72		
Display	<input type="checkbox"/>	1755	Northwest China-04	Xiker	5.8	MW	39.73	45.61		

Slika 65: Primer tabelarnega prikaza vseh meritev, ki ustrezajo kriterijem iskanja

V prvem stolpcu tabele na sliki 65 z imenom 'Select' lahko izberemo meritve, katerih akceleroگرامe bomo prenesli v program SAP2000. Ostale kategorije stolpcev s podatki so opisane v preglednici 5.

Preglednica 5: Opis kategorij posameznih stolpcev z rezultati

Kategorija stolpca v programu	Opis
RSN	Zaporedna številka meritve v podatkovni bazi.
Earthquake name	Ime potresa.
Station name	Ime merilne postaje, k je zabeležila potres.
Magnitude	Magnituda potresa.
Magnitude Type	Tip magnitude, kjer naslednje oznake predstavljajo: MW = momentna magnituda (angl. Moment magnitude), ML = lokalna magnituda (angl. Local magnitude), MS = površinska magnituda (angl. Surface wave magnitude), U = neznan tip magnitude (angl. Unknown type).
Epicenter distance [km]	Razdalja od merilnega mesta do epicentra potresa.
Hypocenter distance [km]	Razdalja od merilnega mesta do žarišča potresa.
Joyner-Boore distance [km]	Joyner-Boore razdalja od merilnega mesta do vertikalne projekcije preloma.
Closest distance [km]	Najkrajša razdalja med merilnim mestom in prelomljenim območjem.
Vs30 [m/s]	Povprečna vrednost hitrosti strižnega valovanja v zgornjih 30-ih m tal.
Year	Leto potresa.
Hypocenter Latitude [°]	Geografska širina žarišča potresa.
Hypocenter Longitude [°]	Geografska dolžina žarišča potresa.
Station Latitude [°]	Geografska širina merilnega mesta.
Station Longitude [°]	Geografska dolžina merilnega mesta.

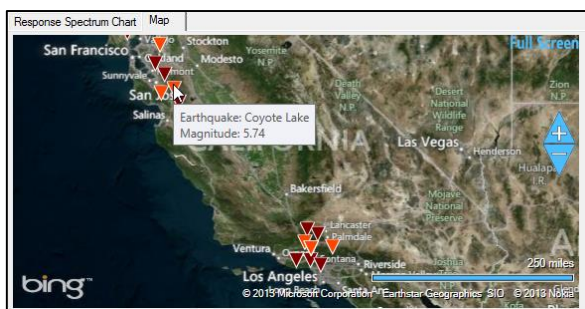
Rezultati v tabeli na sliki 65 so prvotno razvrščeni glede na zaporedno identifikacijsko številko meritve, kar pa lahko spremenimo. Če kliknemo na ime stolpca, se vrstice prerazporedijo tako, da bodo podatki v izbranem stolpcu prikazani od najmanjšega proti največjemu, oziroma po abecednem vrstnem redu. S ponovnim klikom na ime istega stolpca, se vrstice razvrstijo v obratnem vrstnem redu. Če katerega izmed podatkov ni v bazi, je celica v izbrani kategoriji prazna.

S klikom na zavihek 'Map' v drugem modulu zavihka 'Search', si lahko na zemljevidu ogledamo vse lokacije najdenih potresov, ki so označene z oranžnimi trikotniki in vse lokacije najdenih merilnih mest, ki so označene z rdečimi trikotniki – slika 66-68. S kazalcem miške se lahko premaknemo nad trikotnik, ki označuje potres, kar nam odpre okno v katerem se izpišeta ime potresa in njegova

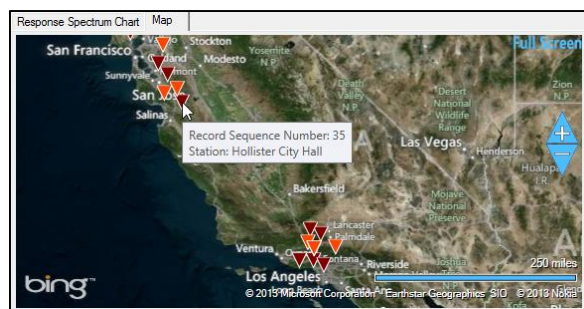
magnituda. Slednje prikazuje slika 67. Če se s kazalcem miške premaknemo nad trikotnik, ki označuje merilno mesto potresa, pa se nam v oknu izpiše identifikacijska številka meritve in ime merilnega mesta, ker je prikazano na sliki 68.



Slika 66: V zavihku z imenom 'Map' so na zemljevidu prikazane geografske lokacije potresov in merilnih mest



Slika 67: Primer izpisa imena in magnitude potresa, ki se nahaja pod kazalcem miške



Slika 68: Primer izpisa zaporedne identifikacijske številke in imena merilnega mesta, ki se nahaja pod kazalcem miške

S klikom na trikotnik, ki označuje potres, se na zemljevidu prikažejo zgolj merilna mesta, ki so zaznala omenjen potres in ustrezajo iskanim kriterijem, s klikom na trikotnik, ki označuje merilno mesto, pa se na zemljevidu prikaže samo potres, ki ga je le to zaznalo.

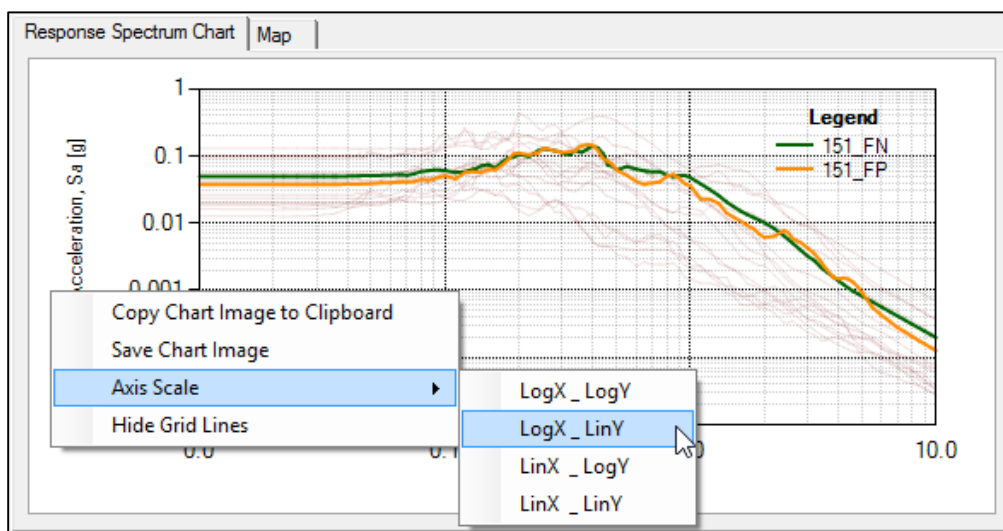
Prikaz ene meritve je mogoč tudi s pritiskom na gumb 'Display' na začetku vsake vrstice v tabeli z rezultati, kar je prikazano na sliki 69. Izbrana vrstica se obarva rdeče.

	Select	RSN	Earthquake name	Station name	Magnitude	M. Type	Epicenter dist. [km]	Hypocenter dist. [km]	JB dist. [km]	Clst dist. [km]
Display	<input type="checkbox"/>	35	Northern Calif-06	Hollister City Hall	5.2	U	38.83			
Display	<input type="checkbox"/>	151	Coyote Lake	Halls Valley	5.74	MW	36.29	37.16	33.69	33.83
Display	<input checked="" type="checkbox"/>	213	Livermore-01	Fremont - Mission San Jose	5.8	MW	37.28	39.17		
Display	<input type="checkbox"/>	1649	Sierra Madre	Vasquez Rocks Park	5.61	MW	39.6	41.38	37.63	39.81
Display	<input type="checkbox"/>	1673	Northridge-03	Santa Monica City Hall	5.2	MW	38.32	38.79		
Display	<input type="checkbox"/>	1705	Northridge-06	LA - 116th St School	5.28	MW	39.04	41.18		
Display	<input type="checkbox"/>	1717	Northridge-06	Lake Hughes #12A	5.28	MW	38.6	40.76		
Display	<input type="checkbox"/>	1719	Northridge-06	Malibu - Point Dume Sch	5.28	MW	38.55	40.72		
Display	<input type="checkbox"/>	1755	Northwest China-04	Xiker	5.8	MW	39.73	45.61		

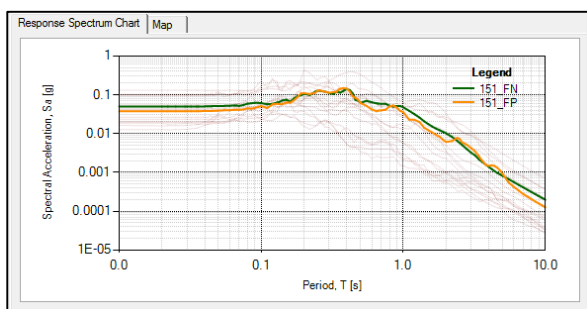
Slika 69: Prikaz posamezne meritve s klikom na gumb z imenom 'Display'

Poleg prikaza geografske lokacije izbrane meritve na zemljevidu, se na grafu spektrov pospeškov sedaj spektra, izračunana na podlagi dveh horizontalnih akceleroگرامov izbrane meritve, obarvata z rumeno in zeleno barvo – slika 70. Tako sta jasno ločena od ostalih. Če z desnim gumbom miške kliknemo na graf, se odpre meni prikazan na sliki 70, v katerem so na voljo naslednje funkcije: (1) Sliko grafa lahko kopiramo v odložišče in jo prilepimo v drugem programu, (2) sliko grafa lahko shranimo na disk, (3) možno je, neodvisno glede na os X, na kateri so prikazani nihajni časi in os Y, ki prikazuje vrednosti pospeškov, izbrati logaritmčno ali linearno merilo – slika 71-74, (4) možno je skriti mrežo pomožnih črt na grafu.

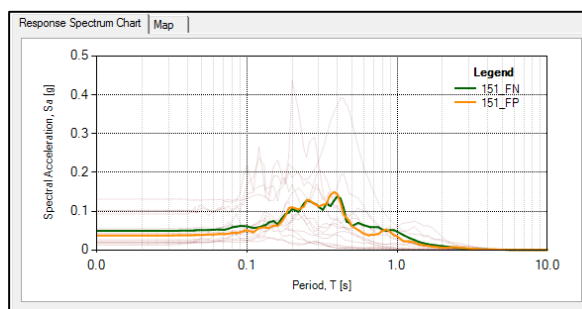
Če želimo ponovno prikazati vse meritve, kliknemo na gumb 'Display all' v modulu 4.



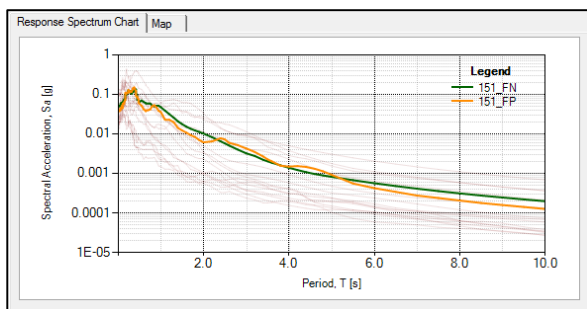
Slika 70: grafičen prikaz spektrov pospeškov posamezne meritve in meni, preko katerega dostopamo do dodatnih funkcij grafa



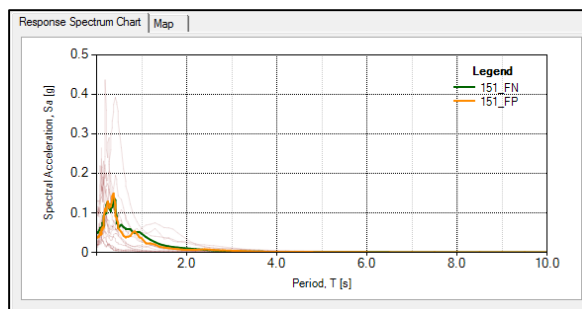
Slika 71: Logaritmčno merilo za os X in Y



Slika 72: Logaritmčno merilo za os X in linearno za os Y



Slika 73: Linearno merilo za os X in logaritmčno za os Y

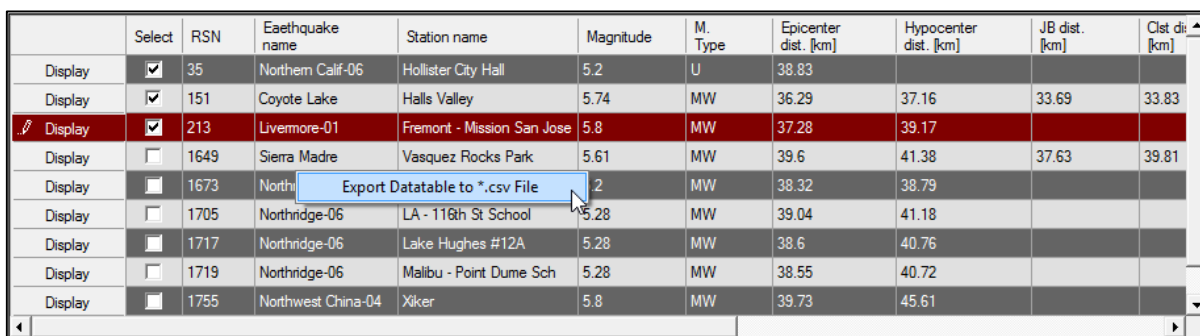


Slika 74: Linearno merilo za os X in Y

Merila osi je mogoče spreminjati tudi z vrtenjem kolesčka na miški, kadar se s kazalcem miške nahajamo nad grafom spektrov pospeškov.

Podatke iz tabele rezultatov je mogoče izvoziti v datoteko s končnico csv (angl. Comma separated values), kar storimo z desnim klikom miške na tabelo in klikom na 'Export Datatable to *.csv File', kar prikazuje slika 75. V datoteki so posamezne vrednosti ločene z vejico. Mogoče jo je odpreti v katerem koli pregledovalniku besedil ali v programom Microsoft Excel, ki jo odpre v obliki razpredelnice.

Meritve, katerih akceleroگرامe želimo prenesti v program SAP2000, izberemo tako, da jih obkljukamo v prvem stolpcu tabele rezultatov z imenom 'Select' – slika 75. Če želimo na enkrat obkljukati vse meritve v tabeli, to storimo tako, da v modulu 4 obkljukamo izbirno polje z imenom 'Select All'.



	Select	RSN	Earthquake name	Station name	Magnitude	M. Type	Epicenter dist. [km]	Hypocenter dist. [km]	JB dist. [km]	Clist di. [km]
Display	<input checked="" type="checkbox"/>	35	Northern Calif-06	Hollister City Hall	5.2	U	38.83			
Display	<input checked="" type="checkbox"/>	151	Coyote Lake	Halls Valley	5.74	MW	36.29	37.16	33.69	33.83
Display	<input checked="" type="checkbox"/>	213	Livermore-01	Fremont - Mission San Jose	5.8	MW	37.28	39.17		
Display	<input type="checkbox"/>	1649	Sierra Madre	Vasquez Rocks Park	5.61	MW	39.6	41.38	37.63	39.81
Display	<input type="checkbox"/>	1673	Northridge-06	LA - 116th St School	5.2	MW	38.32	38.79		
Display	<input type="checkbox"/>	1705	Northridge-06	Lake Hughes #12A	5.28	MW	39.04	41.18		
Display	<input type="checkbox"/>	1717	Northridge-06	Lake Hughes #12A	5.28	MW	38.6	40.76		
Display	<input type="checkbox"/>	1719	Northridge-06	Malibu - Point Dume Sch	5.28	MW	38.55	40.72		
Display	<input type="checkbox"/>	1755	Northwest China-04	Xiker	5.8	MW	39.73	45.61		

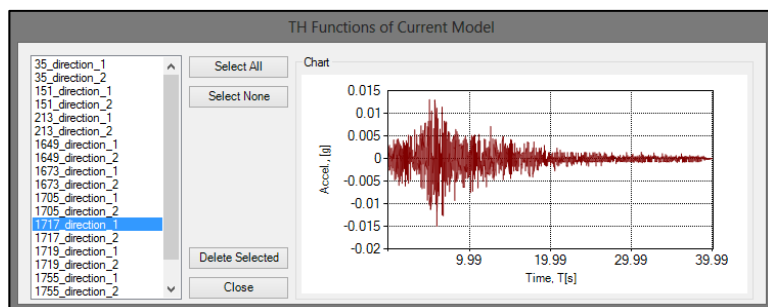
Slika 75: Izvoz podatkov tabele v datoteko s končnico csv in prikaz treh označenih akceleroگرامov v prvem stolpcu tabele, pripravljenih za prenos v program SAP2000

Akceleroگرامe izbranih meritev prenesemo s klikom na gumb 'Import TH Function', stanje prenosa pa lahko spremljamo v statusni vrstici programa, prikazani na sliki 76.



Slika 76: Statusna vrstica programa, v kateri lahko spremljamo stanje prenosa akceleroگرامov v program SAP2000

Po končanem prenosu, lahko s klikom na gumb 'Model TH Function' v modulu 4 odpremo novo okno programa na sliki 77, kjer so na seznamu našteje vse funkcije časovnega poteka trenutnega modela oziroma vsi akceleroگرامi, ki smo jih prenesli s pomočjo vtičnika SSPA. S klikom na posameznega, se izriše njegov graf, ki ga lahko kopiramo v odložišče ali shranimo na disk. V omenjenem oknu lahko tudi izbrisemo uvožene funkcije akceleroگرامov, ki jih ne potrebujemo. Označimo jih na seznamu in kliknemo na gumb 'Delete Selected'. Okno lahko zapremo s klikom na gumb 'Close'.



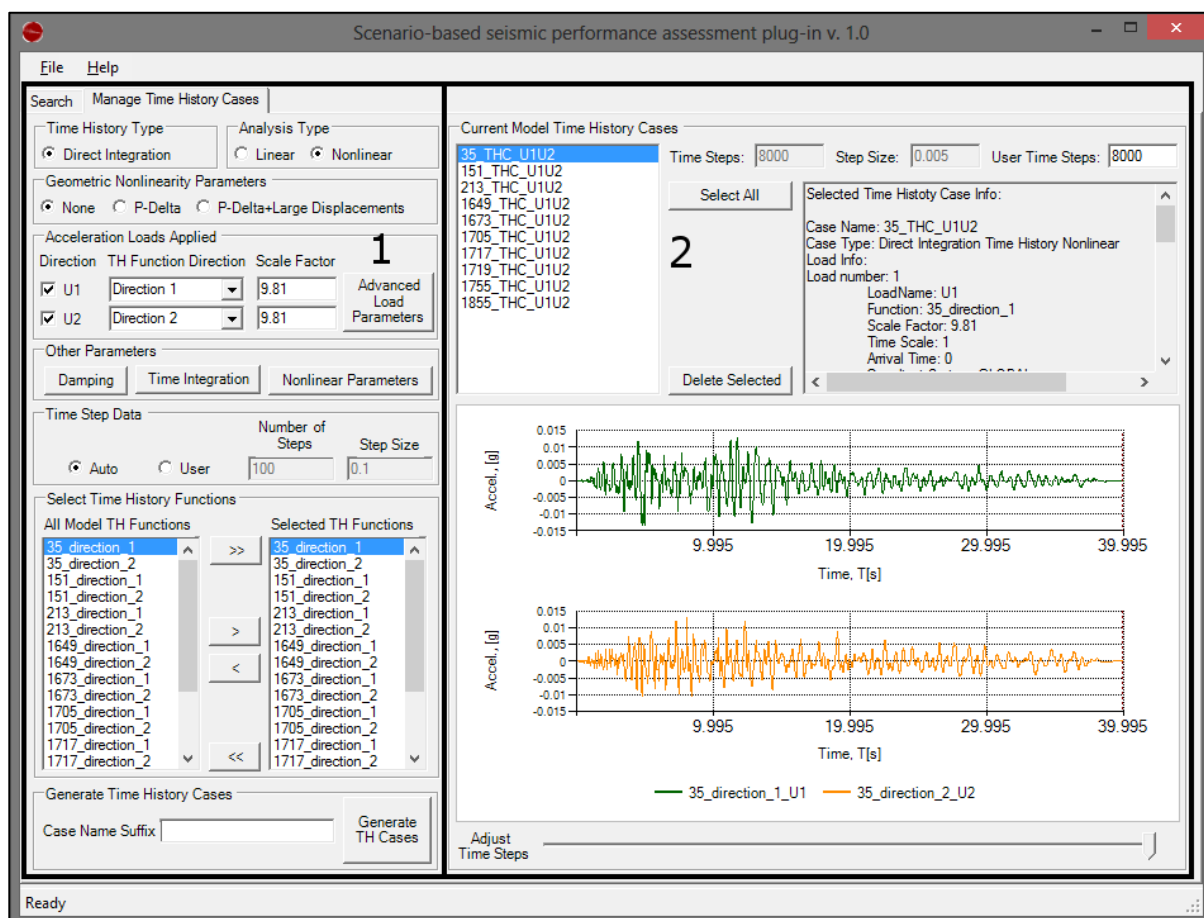
Slika 77: Okno, v katerem so zbrani vsi preneseni akceleroگرامi trenutnega modela

Če želimo odstraniti akceleroگرام, s pomočjo katerega je definiran obtežni primer, moramo najprej odstraniti slednjega, sicer odstranitev akceleroagrama ni mogoča.

Imena akceleroagramov so definirana v obliki zaporedne identifikacijske oznake in ene izmed dveh horizontalnih smeri, v kateri je bil akceleroagram zabeležen.

5.3.5 Generiranje obtežnih primerov na podlagi uvoženih akceleroagramov

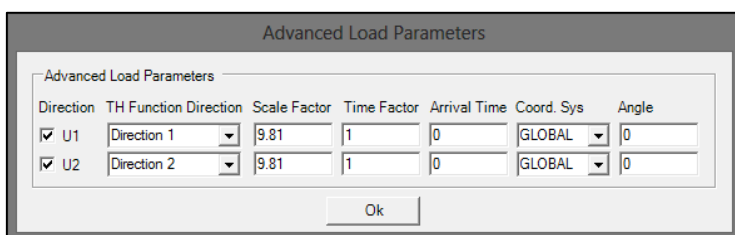
V zavihku z imenom 'Manage Time History Cases' na sliki 78, lahko avtomatsko generiramo poljubno število obtežnih primerov za analizo časovnega odziva konstrukcije. Zavihek je razdeljen na dva modula, kjer je prvi namenjen generiranju drugi pa pregledu obtežnih primerov.



Slika 78: Prikaz zavihka z imenom 'Manage Time History Cases', razdeljenega na dva osnovna modula

Pred generiranjem obtežnih primerov je potrebno v prvem modulu podati naslednje parametre. V območju 'Time History Type' lahko v tej verziji programa zaenkrat izberemo zgolj metodo direktne integracije. V območju 'Analysis Type' izberemo med linearno in nelinearno analizo. V kolikor izberemo slednjo, se aktivira območje 'Geometric Nonlinearity Parameters', kjer določimo geometrijske nelinearne parametre.

V območju 'Acceleration Loads Applied' določimo obtežbo, ki je definirana s časovnim potekom pospeška tal in na konstrukcijo lahko deluje v eni ali dveh medsebojno pravokotnih smereh. Smer delovanja določimo z obkljukanjem izbirnih polj 'U1', ki predstavlja koordinatno os x in 'U2', ki predstavlja koordinatno os y. V kolikor ne nastavimo drugače, gre za osi globalnega koordinatnega sistema. Iz sosednjega seznama 'TH Function Direction' izberemo eno izmed dveh medsebojno pravokotnih smeri, v katerih so bili zabeleženi pospeški tal. Akceleroگرامi so definirani z deležem gravitacijskega pospeška, zato je potrebno v sosednje polje 'Scale Factor', glede na izbrane enote, vpisati pravilen faktor. Pred-nastavljena vrednost 9,81 ustreza enotam kN, m, s. Dodatne parametre obtežbe kot so: časovni faktor, čas pričetka delovanja obtežbe, koordinatni sistem in kot rotacije smeri delovanja obtežbe glede na koordinatni sistem, je mogoče nastaviti v novem oknu na sliki 79, ki ga odpremo s klikom na gumb 'Advanced Load Parameters'.



Slika 79: Okno v katerem je mogoče določiti dodatne parametre obtežbe

V sklopu drugih parametrov 'Other Parameters', s klikom na gumba 'Damping' in 'Time Integration' odpremo novi okni, v katerih določimo parametre dušenja (slika 80) in časovne integracije (slika 81).

Slika 80: Okno za določitev parametrov dušenja

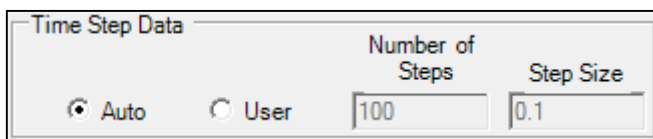
Slika 81: Okno za določitev parametrov časovne integracije

Če smo predhodno izbrali nelinearno analizo, se v okviru istega sklopa aktivira še dodaten gumb z imenom 'Nonlinear Parameters', s katerim odpremo okno prikazano na sliki 82, kjer lahko nastavimo dodatne nelinearne parametre.

Slika 82: Okno za določitev dodatnih nelinearnih parametrov

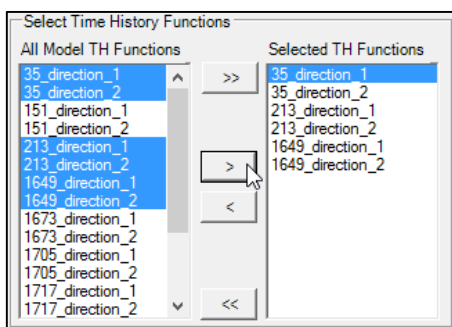
V naslednjem sklopu z imenom 'Time Step Data' na sliki 83, nastavimo število in dolžino časovnih korakov. Na voljo sta dve možnosti. V kolikor izberemo način 'Auto', bosta število in dolžina časovnih korakov samodejno nastavljena za vsak obtežni primer na podlagi števila in dolžine časovnih korakov pripadajočega akcelorograma. Trajanje časovnega odziva za vsak obtežni primer bo tako enako, kot je

trajanje pripadajočega akceleroograma. Drugi način imenovan 'User' pa dopušča uporabniku, da nastavi poljubno število in dolžino časovnih korakov, ki bo enako za vse istočasno generirane obtežne primere.



Slika 83: Sklop z imenom 'Time Step Data' v katerem nastavimo število in dolžino časovnih korakov

V sklopu imenovanem 'Select Time History Functions' na sliki 84 izberemo akceleroграme oziroma funkcije časovnega odziva, na podlagi katerih bodo samodejno generirani obtežni primeri s parametri, ki smo jih definirali v prejšnjih korakih.



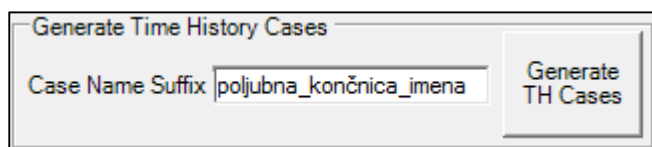
Slika 84: Sklop z imenom 'Select Time History Functions', v katerem izberemo akceleroграme, na podlagi katerih se samodejno generirajo obtežni primeri

Na seznamu z imenom 'All Model TH Functions' so zbrani vsi akceleroграmi, ki smo jih za trenutni model s pomočjo vtičnika uvozili v program SAP2000, izbrani akceleroграmi na podlagi kateri se generirajo obtežni primeri pa se prenesejo na seznamu z imenom 'Selected TH Functions'. Akceleroграme izberemo tako, da v seznamu z imenom 'All Model TH Functions' označimo želene in jih s klikom na gumb '>' prenesemo na seznam 'Selected TH Functions'. Če želimo na enkrat izbrati vse akceleroграme, to storimo s pritiskom na gumb '>>'. Podobno akceleroграme tudi odstranimo iz seznama izbranih, kjer s pritiskom na gumb '<' odstranimo le označene, s pritiskom na gumb '<<' pa hkrati odstranimo vse akceleroграme.

Obtežne primere časovnega odziva generiramo s klikom na gumb 'Generate TH Cases' – slika 85. Samodejno generiranje temelji na sledečem principu. V kolikor smo definirali obtežbo zgolj v eni smeri in izbrali smer zabeleženega akceleroograma 'Direction 1', bo ustvarjeno toliko obtežnih primerov, kolikor akceleroogramov, ki so zabeleženi v tej smeri je na seznamu izbranih. V kateri smeri je zabeležen posamezen akceleroogram je razvidno iz njegovega imena. Analogno velja za smer 'Direction 2'.

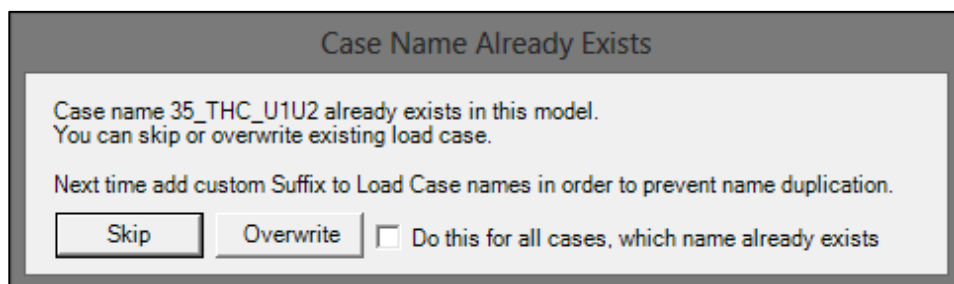
V kolikor definiramo obtežbo v dveh medsebojno pravokotnih smereh in v eni smeri izberemo akceleroogram zabeležen v smeri 'Direction 1' ter v drugi smeri akceleroogram zabeležen v smeri 'Direction 2', bo ustvarjeno toliko obtežnih primerov, kolikor parov akceleroogramov z enako identifikacijsko številko, zabeleženih v obeh smereh, je na seznamu izbranih.

Ime ustvarjenih obtežnih primerov je sestavljeno iz identifikacijske številke meritve oziroma akceleroograma iz baze podatkov, oznake 'THC', ki v angleščini predstavlja okrajšavo za Time History Case in iz definirane smeri delovanja obtežbe (primer: 35_THC_U1U2). Če bi želeli definirati več obtežnih primerov z istimi akceleroogrami in smermi obtežbe, lahko v izogib podvajanju imen, le tem dodamo poljubno končnico. Pred začetkom generiranja obtežnih primerov, ki ga pričnemo s klikom na gumb 'Generate TH Cases', v polje 'Case Name Suffix' vpišemo poljubno končnico, ki bo pripeta za avtomatsko generirano ime obtežnih primerov. Polje za vnos končnice generiranih imen je prikazano na sliki 85.



Slika 85: Polje za vnos poljubne končnice, pripete samodejno generiranim imenom obtežnih primerov

V primeru, da vseeno pride do podvojitve imen, kar bi povzelo obstoječe obtežne primere, nas program na to opozori. Opozorilo je prikazano na sliki 86.

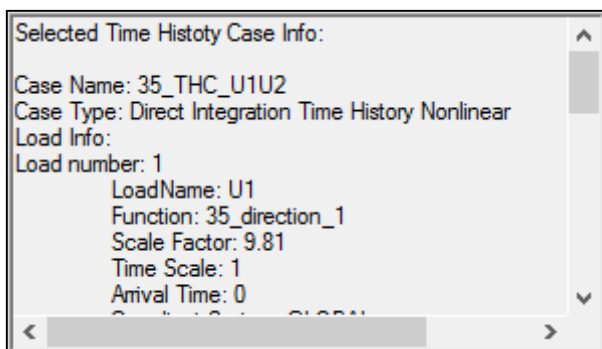


Slika 86: Okno z opozorilom o podvajanju imen obtežnih primerov

S klikom na gumb 'Skip' preskočimo generiranje navedenega obtežnega primera, s klikom na gumb 'Overwrite' pa obstoječi obtežni primer nadomestimo z novim. Če želimo za vse primere s podvojenim imenom uporabiti isto opcijo, lahko obkljukamo izbirno polje z imenom 'Do this for all cases, which name already exists'.

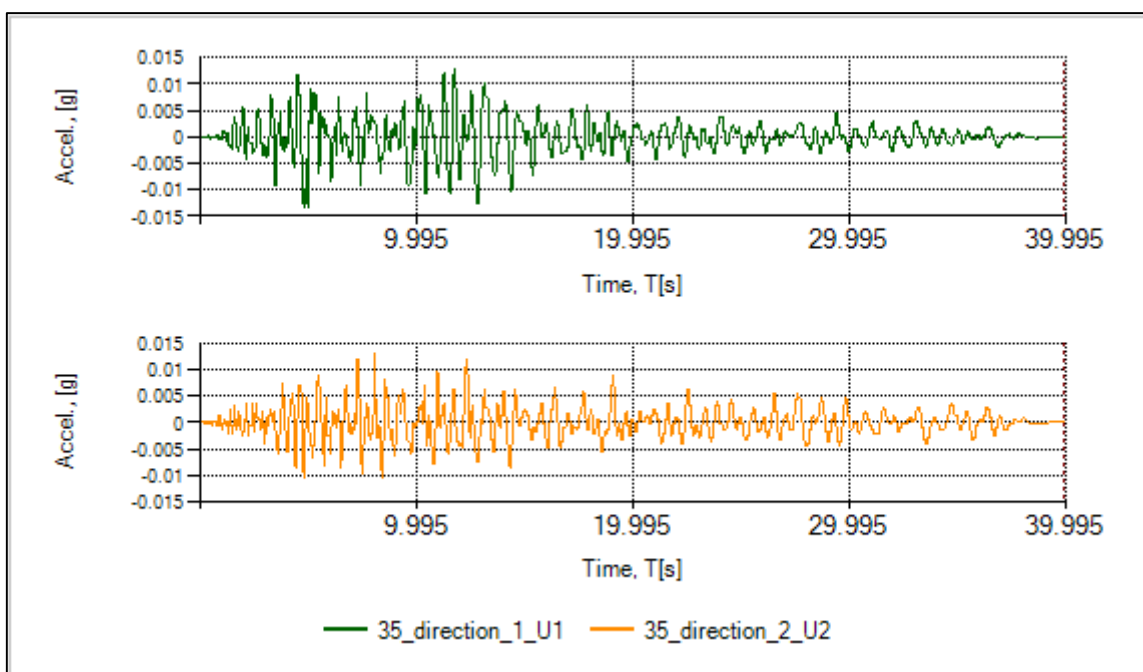
Z ustvarjenimi obtežnimi primeri časovnega odziva je mogoče upravljati v drugem modulu zavihka z imenom 'Manage Time History Cases' –slika 78. Na seznamu z imenom 'Current Model Time History Cases' so naštetni vsi obtežni primeri generirani s pomočjo vtičnika. Če na omenjenem seznam označimo en sam obtežni primer, dobimo o njem naslednje informacije.

V tekstovnem polju na sliki 87, se strukturirano, v urejeni obliki izpišejo vsi parametri, ki obtežni primer definirajo.



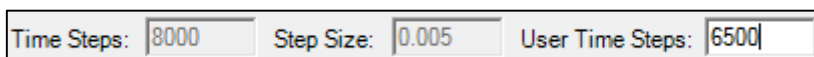
Slika 87: Polje za prikaz strukturirane oblike parametrov, ki določajo izbrani obtežni primer

Na grafu se prikažejo funkcije časovnega odziva oziroma akceleroگرامи, ki predstavljajo obtežbo izbranega obtežnega primera. Ta graf, prikazan na sliki 88, je mogoče, tako kot ostale, kopirati v odložišče ali shraniti na disk.



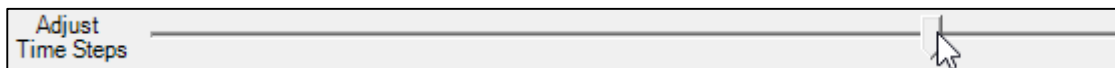
Slika 88: Graf akceleroگرامов izbranega obtežnega primera

Izbranemu obtežnemu primeru je mogoče spremeniti število časovnih korakov, ter s tem skrajšati njegovo trajanje. To lahko storimo na dva načina. Mogoče ga je spremeniti tako, da v polje z imenom 'User Time Steps' na sliki 89, vpišemo novo število časovnih korakov. Pri tem sta, zgolj za informacijo, v poljih z imeni 'Time Steps' in 'Step Size', prikazana število in velikost časovnega koraka akceleroگرامа.



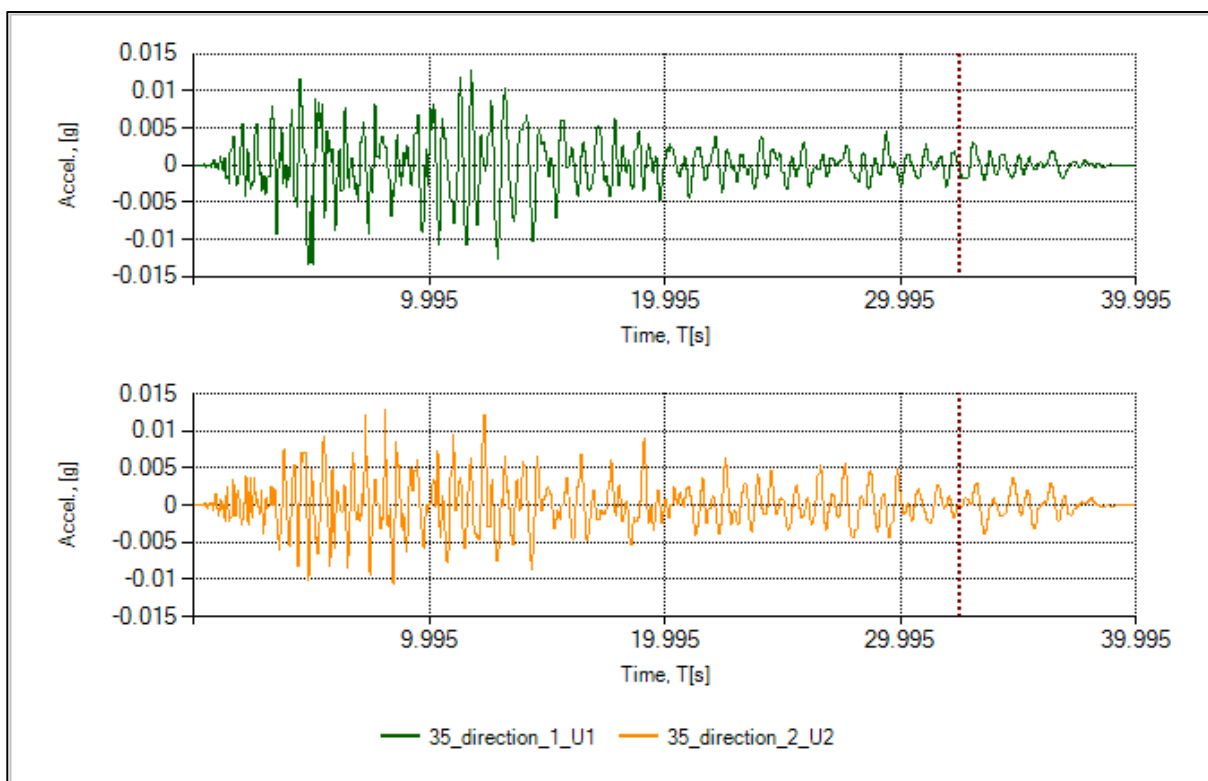
Slika 89: Polje 'User Time Steps', v katerem lahko spremenimo število časovnih korakov izbranega obtežnega primera

Drugi način spreminjanja časovnega koraka in s tem trajanja obtežnega primera, pa je s premikanjem drsnika na sliki 90 z imenom 'Adjust Time Steps', ki se nahaja pod grafom akceleroagramov izbranega obtežnega primera na sliki 88.



Slika 90: Drsnik, s pomočjo katerega lahko spreminjamo število časovnih korakov izbranega obtežnega primera

Pri obeh načinih so spremembe trajanja obtežnega primera vidne tudi na grafu, kjer je z rdečo črtkano črto prikazano, kje se bo račun obtežnega primera, glede na trajanje obtežbe zaključil – slika 91.



Slika 91: Grafična ponazoritev trajanja obtežnega primera glede na trajanje akceleroگرامa

Označene obtežne primere na seznamu 'Current Model Time History Cases' lahko pobrišemo s pritiskom na gumb 'Delete Selected'.

Program po končanem delu lahko zapremo. Vse uvožene funkcije akceleroگرامov in ustvarjeni obtežni primeri so sedaj dostopni tudi preko obstoječega grafičnega vmesnika programa SAP2000. Pred začetkom analize ali zaprtjem programa SAP2000 je potrebno shraniti obstoječi model, v kolikor želimo ohraniti prenesene akceleroگرامe in generirane obtežne primere.

5.4 Zaključni komentarji k programu

Vtičnik SSPA predstavlja zmogljivo orodje, ki zagotavlja kvaliteto integracijo naprednih funkcij v obstoječi grafični vmesnik programa SAP2000. Z njim smo razširili osnovno funkcionalnost programa in tako omogočili nov pristop k oceni potresne odpornosti objektov. Integracijo v obstoječ grafični vmesnik in upravljanje s programom SAP2000 omogoča uporaba programskega vmesnika. Vtičnik je izdelan s pomočjo pogosto uporabljenih programskih jezikov, po principih objektno orientiranega

programiranja. Uporablja standardizirane načine obdelave in prenosa podatkov ter ima modularno zasnovo. Našteto omogoča enostavne posodobitve in implementacijo novih funkcij, ki so že predvidene za prihodnje verzije vtičnika. V prvi fazi je to predvsem izbira funkcij akceleroگرامov na podlagi ciljnega spektra pospeškov, v nadaljevanju pa vgradnja statističnih metod za obdelavo in prikaz rezultatov analize časovnega odziva konstrukcij.

6 DRUGI PRIMERI UPORABE PROGRAMSKEGA VMESNIKA SAP2000

V poglavju štiri in pet smo se osredotočili na izdelavo namenske programske opreme s pomočjo programskega vmesnika za SAP2000, v tem poglavju pa bomo na enostavnih primerih prikazali še nekaj idej oziroma možnosti, ki nam jih ponuja vzpostavitev povezave s pomočjo programskega vmesnika, med programom SAP2000 in drugimi programi.

6.1 Povezava s programom Matlab na primeru generiranja mrež končnih elementov

Generiranje mreže končnih elementov je eden izmed prvih korakov pri analizi ploskovnih konstrukcij po metodi končnih elementov. V programu SAP2000 je izredno enostavno generirati pravokotne ploskovne elemente in jih razdeliti na poljubno končno število v obeh smereh. Generiranje poligonalnih mrež končnih elementov je možno, vendar algoritmi za njihovo samodejno razdelitev na manjše končne elemente ne dajo vedno optimalnih rezultatov. Še posebej zamudno je ročno generiranje ravninskih ukrivljenih mrež končnih elementov, kjer želimo dosegati z zadostnim številom ravnih odsekov končnih elementov. Ročno generiranje ukrivljenih prostorskih ploskovnih mrež končnih elementov praktično ni mogoče. V programu sicer obstaja nekaj orodij, ki so specializirana za generiranje mrež končnih elementov v obliki cilindrov, obokov, hiperboličnih paraboloidov in kupol, vendar to včasih ni dovolj, da bi ustvarili kar se da idealen računski model konstrukcije ali njenega dela.

V ta namen obstaja veliko število namenske programske opreme, ki je specializirana za generiranje mrež končnih elementov. V našem primeru smo izbrali brezplačen program DistMesh, ker je relativno preprost za uporabo, programska koda pa je javno dostopna. To nam omogoča, da algoritme prilagodimo svojim potrebam in generirane mreže s pomočjo programskega vmesnika prenesemo v program SAP2000.

6.1.1 Opis programa DistMesh

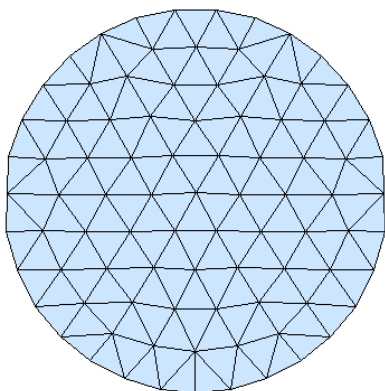
Program DistMesh se uporablja v programu Matlab in služi za generiranje trikotnih in tetraedričnih mrež končnih elementov. Razvila sta ga Per-Olof Persson (UC Berkeley) in Gilbert Strang (MIT). Geometrijo podajamo s pomočjo funkcije območja, ki poda razdaljo poljubne točke do njegovega roba, s predznakom razdalje pa označuje, ali se točka nahaja zunaj ali znotraj območja [56] (angl. signed distance function [57]). Primer take funkcije za enotski krog je $d = r - 1$, kjer je r polmer kroga. Funkcija je pozitivna za vse točke izven, negativna za točke znotraj in enaka nič za točke na meji območja. Samo generiranje mreže končnih elementov temelji na analogiji s paličjem in iterativnem reševanju ravnotežnih enačb pri predpostavljenem odnosu med silo in pomiki [58]. Natančen opis algoritma, program in navodila so na voljo na spletu [23].

Za generiranje ravninskih mrež končnih elementov je v splošnem potrebno podati geometrijo območja, želeno dolžino robov končnih elementov, razdaljo med vozlišči za njihovo začetno distribucijo, pravokotno območje, ki zajema celotno podano geometrijo ter koordinate morebitnih fiksnih točk mreže.

V preglednici 6 sta zapisana ukaza za generiranje enakomerne mreže končnih elementov enotskega kroga, katerega območje podamo s funkcijo v obliki $d(x, y) = \sqrt{x^2 + y^2} - 1$, pri čemer je dolžina stranic končnih elementov enaka $h_0 = 0,2$, krog leži v območju $-1 \leq x \leq 1$, $-1 \leq y \leq 1$, mreža končnih elementov pa nima v naprej določenih fiksnih točk.

Preglednica 6: Ukaza za generiranje mreže končnih elementov enotskega kroga s programom Distmesh v programu Matlab

Prvi ukaz, s pomočjo katerega definiramo funkcijo območja za enotski krog	<code>fd=@(p) sqrt(sum(p.^2,2))-1;</code>
Drugi ukaz, v katerem s pomočjo funkcije 'distmesh2d', generiramo mrežo končnih elementov	<code>[p,t]=distmesh2d(fd,@huniform,0.2,[-1,-1;1,1],[]);</code>



Slika 92: Enotski krog razdeljen na enakomerno mrežo trikotnih končnih elementov s stranico $h_0 = 0,2$, s pomočjo programa DistMesh

Program poleg grafične reprezentacije mreže končnih elementov prikazane na sliki 92, vrne matriko p reda $n \times 2$ (n je število vozlišč končnih elementov), kjer so zbrane koordinate x , y za posamezna vozlišča, ter matriko t reda $m \times 3$, (m je število končnih elementov), kjer so zbrana imena vozlišč posameznega končnega elementa.

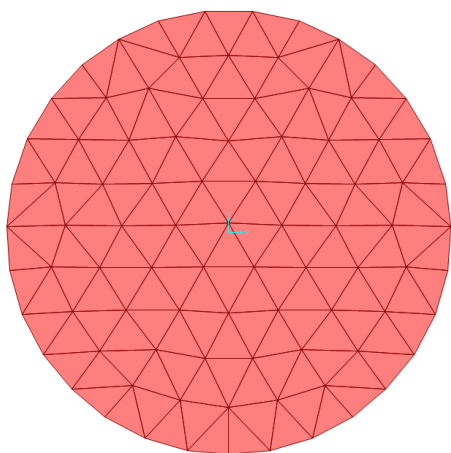
Z unijo, razliko in presekom funkcij lahko sestavimo kompleksnejšo geometrijo. Na podoben način je mogoče generirati prostorske ploskovne mreže končnih elementov. Rezultate vedno dobimo v obliki matrike koordinat vozlišč in matrike imen vozlišč posameznega končnega elementa.

6.1.2 Prenos mrež končnih elementov v program SAP2000

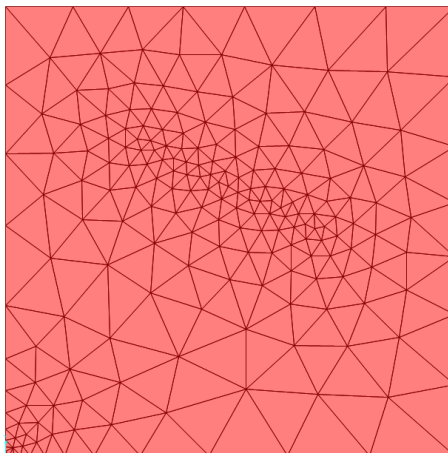
Oblika rezultatov programa DistMesh je izredno primerna za generiranje mrež končnih elementov v programu SAP2000. Vse kar moramo storiti je, da preko programskega vmesnika vzpostavimo povezavo s programom SAP2000, ustvarimo nov model, glede na matriko p generiramo n točk s koordinatami x , y ter ustvarimo m ploskovnih elementov glede na matriko t . Diplomski nalogi je

priložen modificiran program DistMesh, kjer je v datotekah 'distmesh2d.m' in 'distmeshsurface.m' dodana programska koda, ki samodejno prenese generirane ravninske in prostorske mreže ploskovnih končnih elementov v program SAP2000. Izbrana dolžinska enota je meter. Pri tem omenimo, da gre le za demonstracijo ideje, zato je samodejno generiranje omejeno zgolj na geometrijo, brez definicije robnih pogojev, materialnih karakteristik, obtežbe ali drugih parametrov. To lahko naknadno storimo v programu SAP2000. Če bi želeli avtomatsko določiti robne pogoje na celotni mreži končnih elementov, bi to lahko storili iz pogoja, da je vrednost funkcije s katero podamo geometrijo na robovih območja teoretično enaka nič ($|d| = 0$) v resnici pa je zaradi numeričnih napak in približkov manjša od epsilon ($|d| < \varepsilon$, $\varepsilon \ll 0$). Na ta način bi lahko označili oziroma pridobili imena vseh točk na robovih mreže, ter jim določili ustrezne robne pogoje.

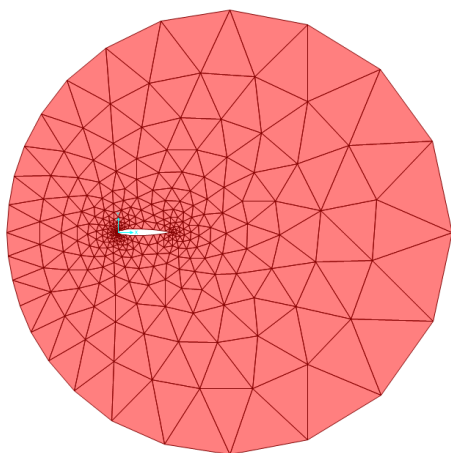
Na slikah 93-100 so prikazani primeri mrež končnih elementov v programu SAP2000. Mreže so bile generirane v programu DistMesh in avtomatsko prenesene v program SAP2000 s pomočjo programske kode, ki temelji na uporabi programskega vmesnika.



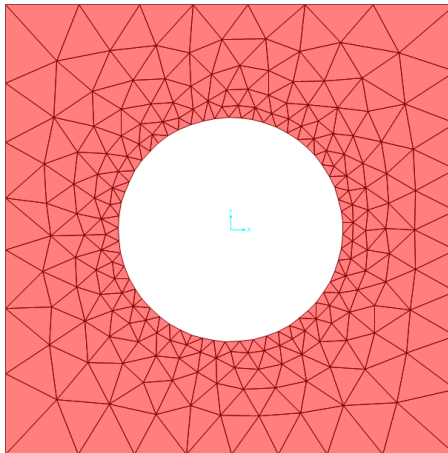
Slika 93: Krog, razdeljen na enakomerno mrežo končnih elementov



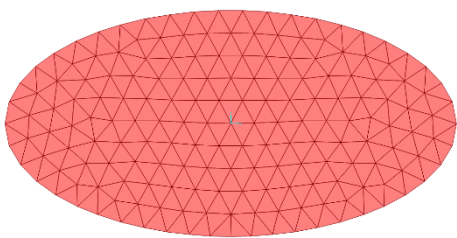
Slika 94: Kvadrat, razdeljen na neenakomerno mrežo končnih elementov



Slika 95: Krog z odprtino, razdeljen na neenakomerno mrežo končnih elementov



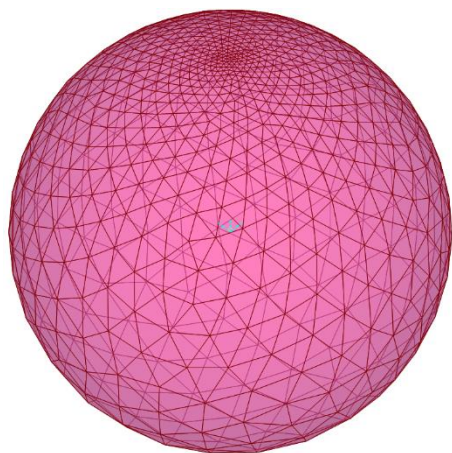
Slika 96: Kvadrat s krožno odprtino, razdeljen na neenakomerno mrežo končnih elementov



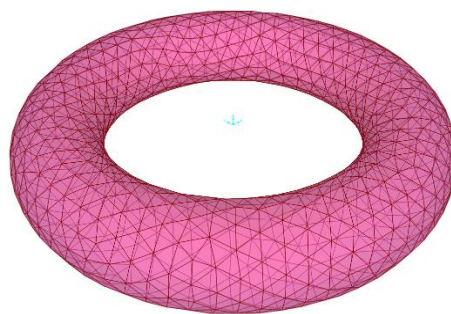
Slika 97: Elipsa, razdeljena na enakomerno mrežo končnih elementov



Slika 98: Poljuben poligon, razdeljen na enakomerno mrežo končnih elementov



Slika 99: Sfera, razdeljena na neenakomerno mrežo končnih elementov

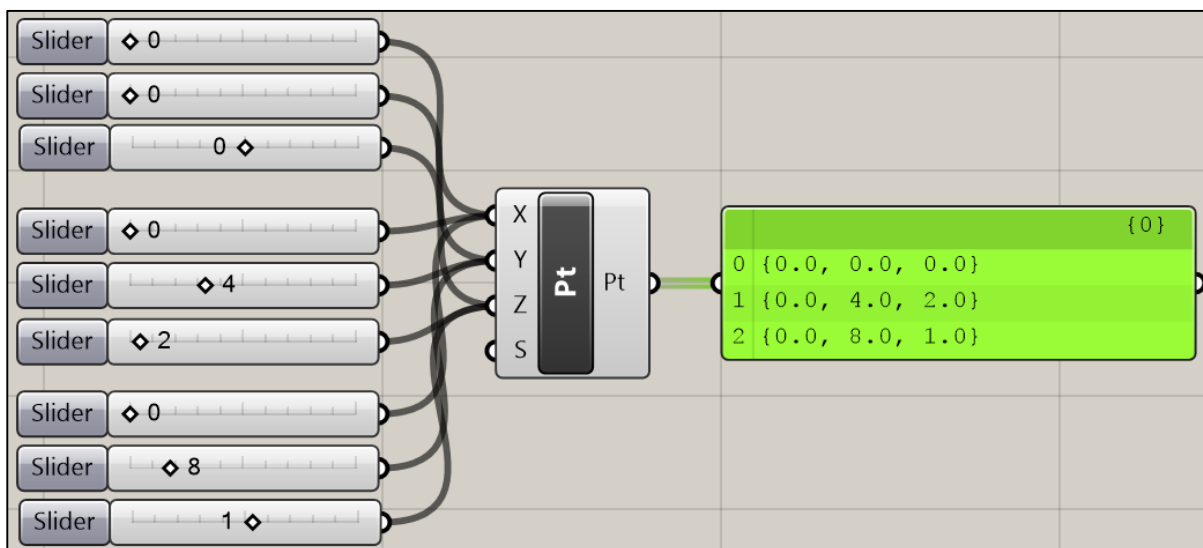


Slika 100: Torus, razdeljen na enakomerno mrežo končnih elementov

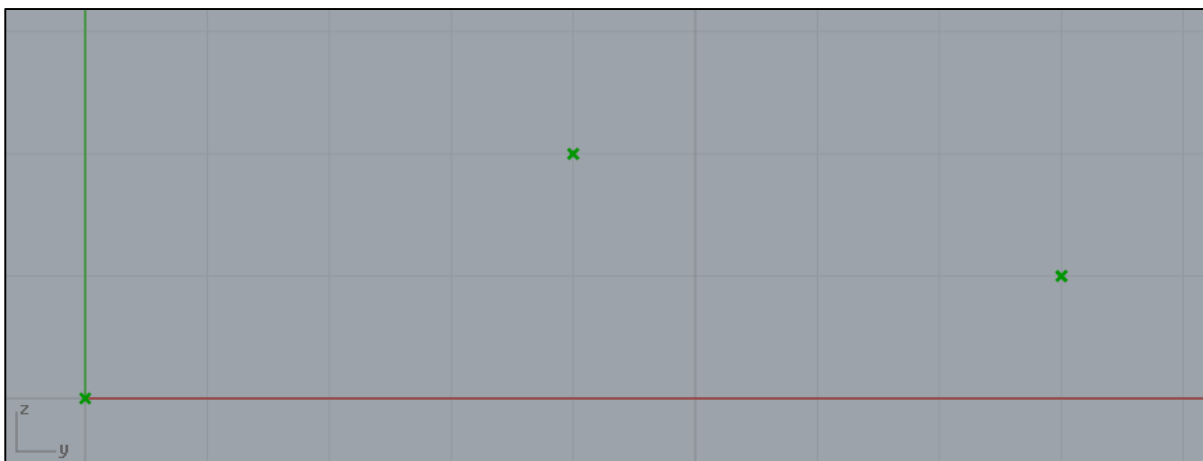
6.2 Povezava s programom Rhinocerus 3D ter vtičnikom Grasshopper 3D

Rhinocerus 3D [14] je program za tridimenzionalno modeliranje, ki temelji na neenakomernih racionalnih osnovnih krivuljah [59] (angl. non-uniform rational basis spline, NURBS), s pomočjo katerih je mogoče ustvariti poljubne in analitične oblike [60]. Program se najpogosteje uporablja v arhitekturnem načrtovanju in industrijskem oblikovanju. Skupaj z vtičnikom Grasshopper 3D [15], ki predstavlja vizualni programski jezik za parametrično modeliranje geometrije, tvori izredno zmogljivo in napredno orodje. Princip delovanja vtičnika Grasshopper 3D temelji na komponentah, katerih izhodne podatke uporabimo kot vhodne parametre v novih komponentah. V program so vgrajene komponente, s katerimi je mogoče generirati geometrijo, izvajati matematične in logične operacije, nadzorovati vhodne podatke in prikazati rezultate. Z vsemi temi orodji je možno v programu Rhinocerus 3D parametrično generirati najrazličnejšo geometrijo. Osnovni princip je prikazan na sliki 101. S pomočjo komponente 'Point XYZ' z imenom 'Pt' smo definirali tri točke v prostoru. Prvi trije vhodni parametri komponente so posamezne koordinate točk, ki jih podamo s komponentami tipa 'Number Slider' oziroma številskimi drsniki, s katerimi lahko enostavno spreminjamo posamezne vrednosti. Četrti vhodni parameter je koordinatni sistem v katerem so definirane koordinate in je v tem primeru konstanten ter enak globalnemu koordinatnemu sistemu z izhodiščem v točki $O(0,0,0)$. Določili smo ga s klikom na parameter 'S'. Rezultati komponente 'Pt' so točke, ki se izrišejo v programu Rhinocerus 3D prikazane na sliki 102, ter izhodni parametri, ki smo jih izpisali v

komponenti 'Panel' na sliki 101. Ta služi tekstovnemu prikazu rezultatov. S sestavljanjem komponent oziroma povezovanjem izhodnih parametrov ene z vhodnimi argumenti druge komponente, lahko parametrično oblikujemo poljubno geometrijo.



Slika 101:Komponenta 'Point XYZ' z imenom 'Pt' in njeni vhodni ter izhodni parametri



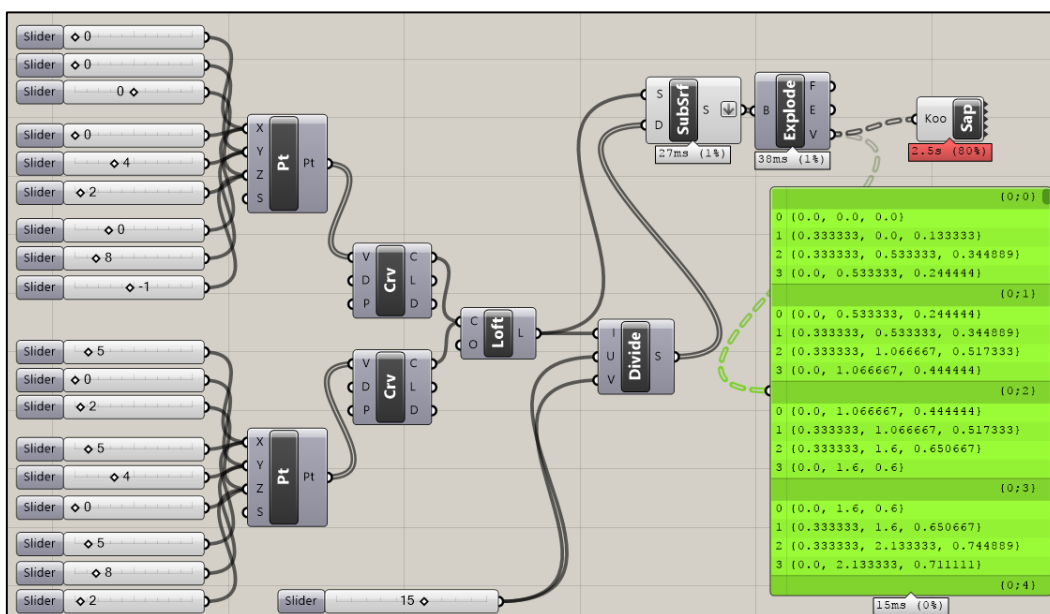
Slika 102:Tri točke v prostoru izrisane v programu Rhinoceros 3D, so rezultat sheme iz slike 101

Poleg že vgrajenih komponent program omogoča, da s pomočjo programskega jezika Visual Basic .Net ali C# .Net ustvarimo tudi lastne komponente. Na ta način nam je uspelo preko programskega vmesnika vzpostaviti povezavo s programom SAP2000 in ustvarjeno geometrijo, skupaj s podatki o ostalih parametrih analize prenesti v program SAP2000. Na dveh idejnih primerih bomo pokazali, kaj nam omogoča omenjena povezava.

V prvem primeru bomo definirali ploskev, ki bo parametrizirana z vsemi tremi koordinatami šestih neodvisnih točk v prostoru. Ploskev bomo nato razdelili na končno število manjših ploskev, ki predstavljajo posamezne končne elemente in jih prenesli v program SAP2000.

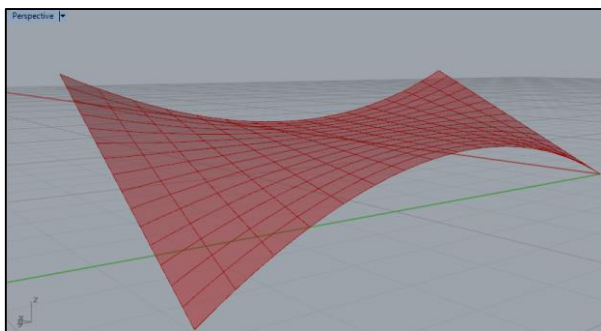
Postopek modeliranja parametrične ploskve je sledeč. Najprej definiramo dva sklopa treh točk v prostoru, katerih koordinate lahko spreminjamo z drsniki. Skozi vsak sklop treh točk nato napnemo

neenakomerno racionalno krivuljo tretjega reda, v naslednjem koraku pa skozi obe krivulji napremo ploskev. To ploskev razdelimo na končno število manjših ploskev, ki je enako v obeh smereh in ga lahko parametrično spreminjamo s pomočjo dodatnega drsnika. Koordinate vozlišč posameznih ploskev predstavljajo vhodne podatke za komponento z imenom 'Sap', ki smo jo napisali sami. Ta komponenta vzpostavi povezavo s programom SAP2000 in v njem, glede na vhodne podatke, ustvari mrežo ploskovnih elementov. Pred-nastavljena dolžinska enota je meter. Celotna shema je prikazana na sliki 103. S premikanjem drsnikov, ki nadzirajo posamezne vrednosti koordinat točk, lahko spreminjamo geometrijo parametrizirane ploskve. Vse spremembe geometrije se avtomatično odražajo tudi v programu SAP2000.

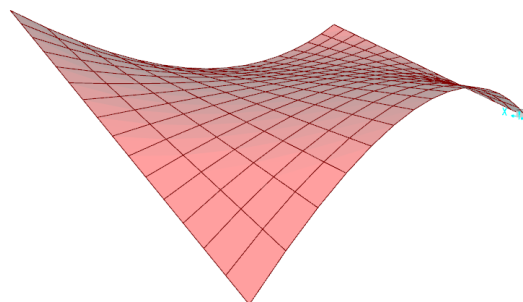


Slika 103: Shema primera parametrizacije ploskve v programu Grasshopper 3D

Na sliki 104 je primer parametrizirane ploskve izrisane v programu Rhinoceros 3D, na sliki 105 pa mreža ploskovnih elementov v programu SAP2000.



Slika 104: Parametrizirana ploskev izrisana v programu Rhinoceros 3D

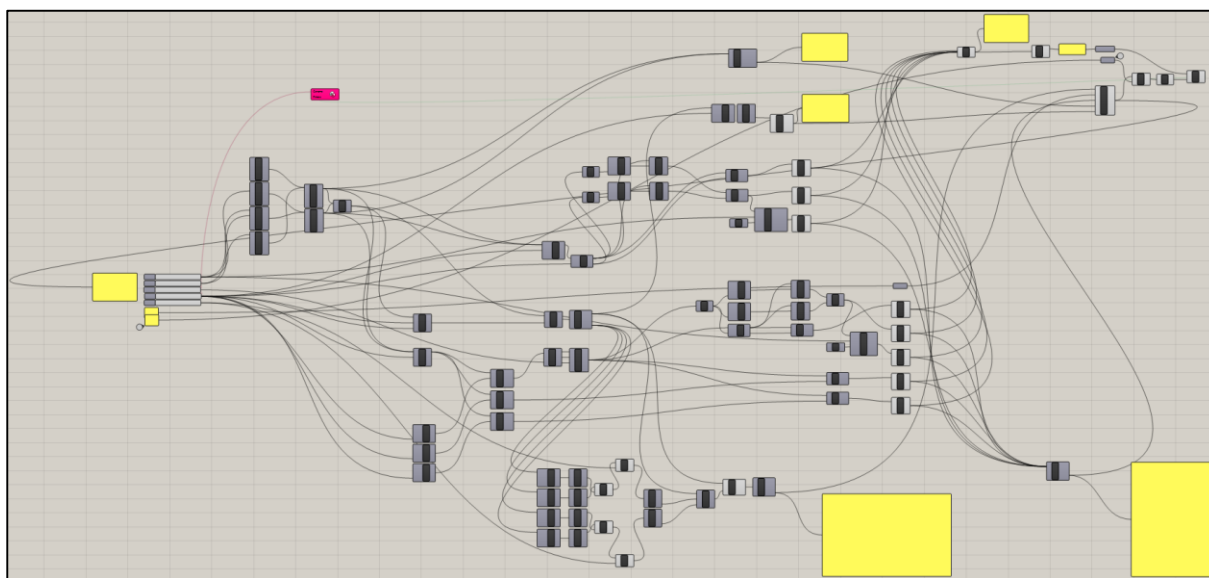


Slika 105: Generirana mreža ploskovnih elementov v programu SAP2000

V drugem primeru smo s pomočjo vtičnika Grasshopper 3D v programu Rhinoceros 3D parametrično zmodelirali preprosto premostitveno konstrukcijo. S pomočjo programskega jezika Visual Basic .Net smo napisali lastno komponento za vtičnik Grasshopper 3D, s pomočjo katere v program SAP2000

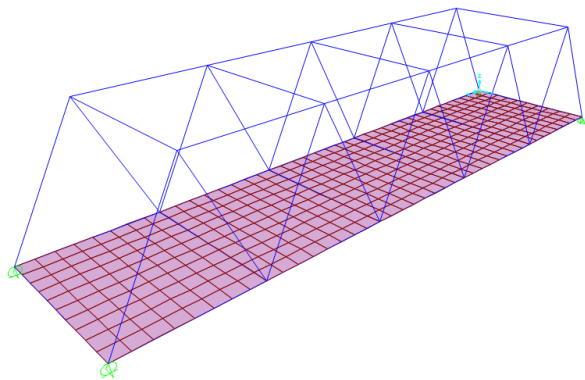
prenesemo geometrijo mostu, ter definiramo parametre obtežbe, materiala, prerezov in ostale parametre potrebne za statično analizo konstrukcije. Po končani analizi komponenta vrne pomik poljubno predhodno izbrane točke v globalni smeri osi Z. Na enostavnem primeru bomo pokazali idejo, kako lahko avtomatiziramo proces optimizacije konstrukcije na podlagi spreminjanja geometrijskih parametrov.

Zasnovali smo idealiziran premostitveni objekt, katerega nosilno konstrukcijo sestavljajo jekleni linijski elementi s cevastim prerezom premera 0,15 m in debeline 5 mm. Most je točkovno podprt v vseh štirih vogalih in sicer členkasto na enem ter drsno na drugem koncu. Vozno konstrukcijo predstavlja betonska plošča debeline 0,30 m. Vsi ostali geometrijski parametri, kot so širina ter dolžina mostu, višina ter število razponov jeklene nosilne konstrukcije in gostota mreže končnih elementov, so poljubni in jih lahko spreminjamo neodvisno glede na ostale parametre. Obtežbo na konstrukcijo predstavlja kombinacija lastne teže, pomnožene z varnostnim faktorjem 1,35 in enakomerno porazdeljene koristne ploskovne obtežbe, pomnožene z varnostnim faktorjem 1,5, ki je nanesena na ploskovne elemente v smeri delovanja gravitacijske sile. Celotna shema izdelave modela v programu Grasshopper 3D je prikazana na sliki 106. Namensko izdelana komponenta s pomočjo programskega jezika Visual Basic.NET z imenom 'Most', ki s pomočjo programskega vmesnika vzpostavi povezavo s programom SAP2000 ter skrbi za dvosmerni tok podatkov ima naslednje vhodne parametre: koordinate vozlišč ploskovnih elementov, koordinate vozlišč linijskih elementov, karakteristično vrednost ploskovne obtežbe v kN/m^2 , koordinate podprtih točk in koordinato opazovane točke. Izhodni parameter komponente je vertikalni pomik opazovane točke v metrih. Vsi ostali parametri, ki so potrebni za analizo konstrukcije, kot so podatki o materialu, varnostni faktorji idr., so v danem primeru definirani znotraj komponente 'Most' in jih ni mogoče spreminjati.

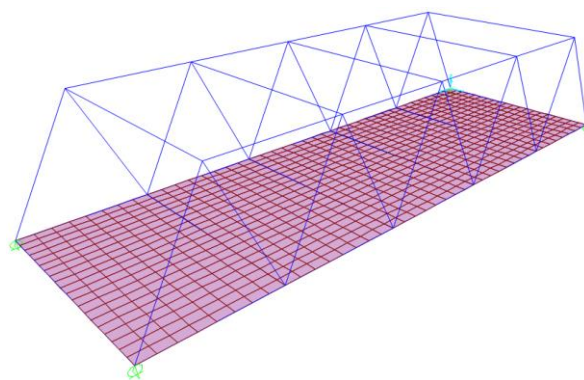


Slika 106: Shema parametrizacije idealizirane premostitvene konstrukcije v programu Grasshopper 3D

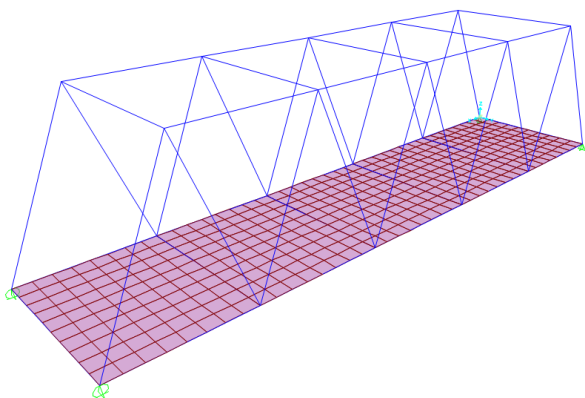
Podobno kot v prvem primeru se s spremembo katerega koli parametra, ki definira geometrijo ali obtežbo konstrukcije, vse spremembe odražajo tudi v programu SAP2000. Glavna razlika pri tem je, da se poleg geometrije prenesejo tudi podatki potrebni za analizo, kar nam omogoča istočasno opravljanje statične analize v programu SAP2000 ter prenos izbranih rezultatov z vsako spremembo parametra. V našem primeru vrnemo podatek o vertikalnem pomiku točke na sredini razpona mostu. Na slikah 108-112 so prikazane možne parametrizacije geometrije konstrukcije, glede na referenčno konstrukcijo na sliki 107.



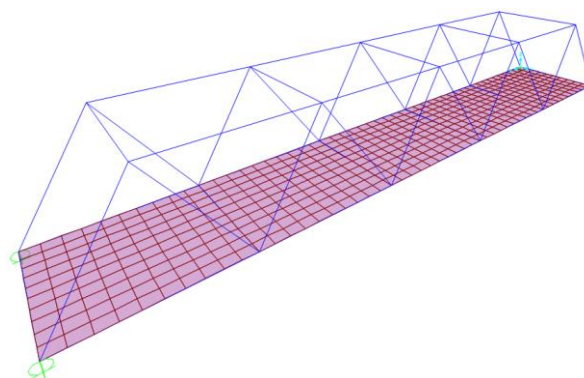
Slika 107: Izbrano referenčno stanje premostitvene konstrukcije



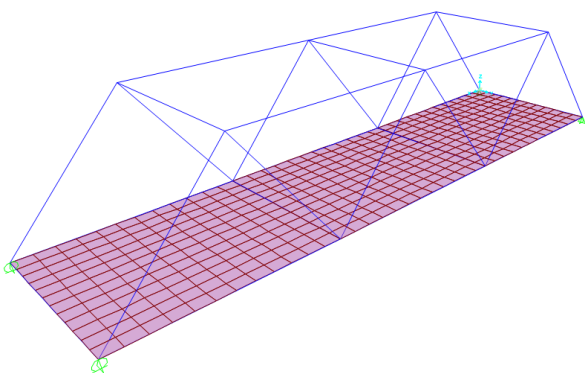
Slika 108: Sprememba širine premostitvene konstrukcije, glede na referenčno stanje



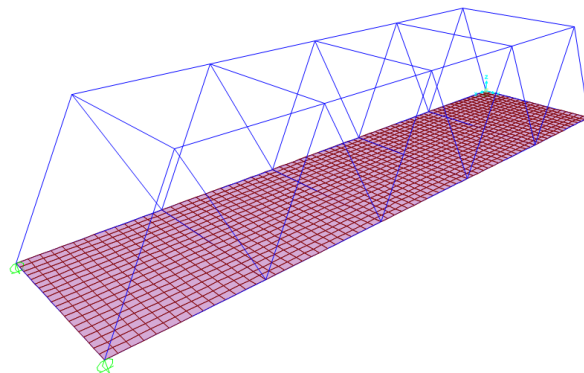
Slika 109: Sprememba višine nosilne konstrukcije, glede na referenčno stanje



Slika 110: Sprememba dolžine premostitvene konstrukcije, glede na referenčno stanje



Slika 111: Sprememba števila razponov nosilne konstrukcije, glede na referenčno stanje

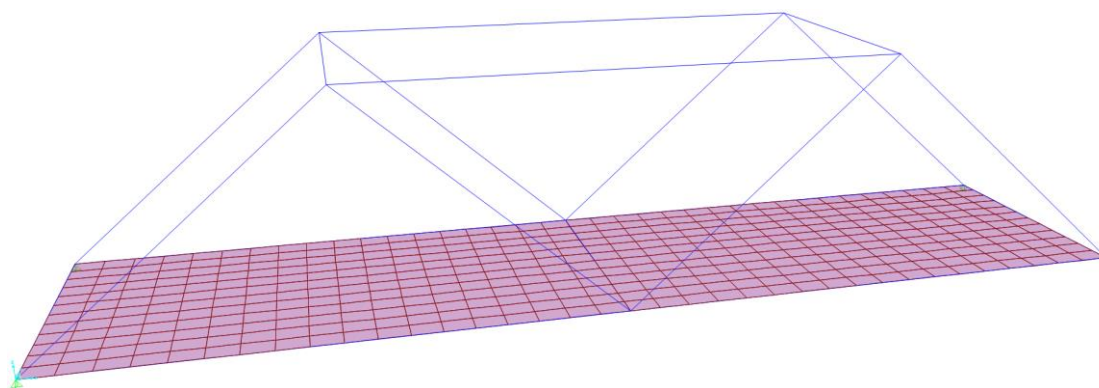


Slika 112: Sprememba dimenzij končnih elementov, glede na referenčno stanje

Parametrična zasnova konstrukcije nam v povezavi z analizo v programu SAP2000 omogoča njeno optimizacijo. Program Grasshopper 3D skupaj z vgrajeno komponento 'Galapagos' nudi možnost samodejnega spreminjanja želenih parametrov in iskanja optimalne rešitve glede na vrednost poljubne namenske funkcije. V kolikor je le ta ustrezno podana, njeni rezultati konvergirajo k eni sami rešitvi.

Za demonstracijo ideje smo dano konstrukcijo optimizirali tako, da je pri njeni konstantni dolžini $d=20$ m in širini $\bar{s}=5$ m poraba materiala nosilne konstrukcije najmanjša, pri tem pa zadostimo podani omejitvi pomika na sredini razpona $u_z > -0,02$ m. Zasnovan algoritem v vsakem koraku spremeni parametra, ki določata višino nosilne konstrukcije v na intervalu $2,2 \text{ m} \leq v \leq 5,2 \text{ m}$ ter število njenih razponov n na intervalu $1 \leq n \leq 10$, opravi statično analizo v programu SAP2000 in vrne vertikalni pomik na sredini razpona mostu. S pomočjo vrednosti pomika in količine porabljenega materiala nosilne konstrukcije se izračuna vrednost namenske funkcije, na podlagi katere se samodejno nastavijo nove vrednosti parametrov. Na podlagi teh se začne nov korak algoritma in zanka se ponovi. V kolikor vrednosti namenske funkcije konvergirajo k enemu samemu maksimumu oziroma minimumu, dobimo optimalno rešitev v relativno kratkem času.

Rešitev prikazane optimizacije predstavljata parametra, ki določata število razponov nosilne konstrukcije in njeno višino. Najmanjšo porabo materiala nosilne konstrukcije, ki jo v našem primeru zaradi enotnih prerezov lahko določimo kot skupno dolžino linijskih elementov, pri kateri še zadostimo omejitvam nastavljenega pomika $u_z > -0,02$ m, dosežemo pri dveh razponih nosilne konstrukcije ter njeni višini $v = 4,4$ m. Pomik opazovane točke pri tem znaša $-0,01926$ m, skupna dolžina linijskih elementov pa je približno enaka 138,3 m. Optimalna premostitvena konstrukcija je prikazana na sliki 113.



Slika 113: Optimizirana premostitvena konstrukcija

Programska koda namenskih komponent in ustrezne datoteke s parametriziranimi modeli v programu Grasshopper 3D so priložene diplomski nalogi.

7 ZAKLJUČEK

V diplomski nalogi smo predstavili programski vmesnik za program SAP2000 in raziskali koncepte njegove uporabe. Če povzamemo, nudi programski vmesnik vzpostavitev dvosmerne komunikacije med novo ali obstoječo programsko opremo in programom SAP2000. Ta zagotavlja drugim programom dostop do njegovih funkcij in numeričnih metod ter možnost implementacije in združevanja funkcionalnosti drugih aplikacij s programom SAP2000. Hkrati omogoča integracijo razvitih aplikacij v njegov obstoječ grafični vmesnik.

Vsi našeti koncepti so bili uporabljeni pri razvoju programov, ki temeljijo na uporabi programskega vmesnika za SAP2000. Osredotočili smo se predvsem na področje potresne analize. Najprej smo napisali program za analizo lastnega nihanja ter modalno analizo, ki ga je z nekaj dodatki mogoče uporabiti za oceno potresne ogroženosti širšega urbanega območja. V nadaljevanju smo izdelali vtičnik za dinamično analizo na osnovi potresnega scenarija, ki je definiran glede na magnitudo potresa, oddaljenost potresa od lokacije objekta, tip tal ter nekatere ostale parametre. Vtičnik, ki smo ga poimenovali 'Scenario-Based Seismic Performance Assessment Plug-in', je neposredno integriran v grafični vmesnik programa SAP2000 in omogoča pripravo analize, s pomočjo katere je mogoče oceniti ranljivost konstrukcije na izbran potresni scenarij. V okviru tega programa, smo na strežniku Fakultete za gradbeništvo in geodezijo Univerze v Ljubljani vzpostavili bazo podatkov potresnega delovanja z več kot 10.000 akceleroگرامi in spektri pospeškov, ki jo je mogoče uporabiti pri nadaljnjem raziskovalnem delu. Pripravili smo tudi primere uporabe programskega vmesnika, ki združujejo funkcionalnosti obstoječih programov, kot so Matlab, DistMesh, Rhinocerus 3D in Grasshopper 3D, s funkcionalnostjo programa SAP2000.

Z razvojem aplikacij, ki temeljijo na uporabi programskega vmesnika za SAP2000, lahko prilagodimo vmesnik za interakcijo s programom, avtomatiziramo procese načrtovanja in analize, nadgradimo funkcionalnost programa SAP2000 in izvajamo parametrične študije. Na ta način, lahko izboljšamo procese za projektiranje objektov ter omogočimo bolj detaljne študije, ki jih zaradi časovno zamudne priprave podatkov ni mogoče izvajati v praksi.

Sama uporaba programskega vmesnika programa SAP2000 je relativno nezahtevna, predvsem v primeru, ko ima uporabnik nekaj predhodnega znanja programiranja, veliko zahtevnejši pa je lahko celostno izpeljan in avtomatiziran proces upravljanja s programom SAP2000. Ta navadno ne odraža privajenih postopkov upravljanja s programom preko grafičnega vmesnika. Posledice tega so dodatni a potrebni logični procesi, ki se jih je potrebno privaditi pri razvoju programske opreme. Programski vmesnik je praviloma posodobljen oziroma razširjen z vsako novo izdano verzijo programa SAP2000, kljub temu pa še vedno ni mogoče dostopati do popolnoma vseh funkcij, kot to lahko storimo preko

grafičnega vmesnika. Na tem mestu omenimo še programski vmesnik programa ETABS [61]. Program ETABS je razvit na ogrodju programa SAP2000 in pri računu uporablja enake numerične metode, vgrajene pa ima funkcije, ki so prilagojene za analizo gradbenih konstrukcij. V mesecu maju leta 2013 je izšla nova verzija programa ETABS 2013, v kateri je prvič vključen tudi programski vmesnik. V prihodnje je smiselno preučiti morebitne dodatne možnosti, ki jih ta ponuja pri upravljanju s programom ETABS. Pri tem lahko za osnovo služi ta diplomska naloga, saj sta oba programska vmesnika zasnovana na enak in konsistenten način, vsebujeta pa lahko različne funkcije, prilagojene za upravljanje s posameznim programom.

VIRI

Uporabljeni viri:

- [1] Computers and Structures, Inc. SAP2000: računalniški program za analizo in zasnovu konstrukcij po metodi končnih elementov.
<http://www.csiamerica.com/sap2000> (Pridobljeno 28. 08. 2013.)
- [2] Microsoft Support. 2007. What is DLL?.
<http://support.microsoft.com/kb/815065> (Pridobljeno 14. 08. 2013)
- [3] Microsoft MSDN. 2013. Dynamic-Link Libraries.
<http://msdn.microsoft.com/en-us/library/windows/desktop/ms682589%28v=vs.85%29.aspx>
(Pridobljeno 14. 08. 2013.)
- [4] Microsoft MSDN. 2013. Component Object Model (COM).
[http://msdn.microsoft.com/en-us/library/windows/desktop/ms680573\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/windows/desktop/ms680573(v=vs.85).aspx)
(Pridobljeno 21. 08. 2013.)
- [5] TechTerms. API.
<http://www.techterms.com/definition/api> (Pridobljeno 14. 08. 2013.)
- [6] 3scale. 2011. What is an API? Your guide to the Internet Business (R)evolution.
<http://www.3scale.net/wp-content/uploads/2012/06/What-is-an-API-1.0.pdf>
(Pridobljeno 14.08.2013.)
- [7] Computers and Structures, Inc. SAP2000 Open Application Programming Interface (OAPI).
<http://www.csiamerica.com/sap2000/open-api> (pridobljeno 11. 08. 2013).
- [8] Sextos, A. G., Balafas, G.K. 2011. Using the new sap2000 open application programming interface to develop an interactive front-end for the modal pushover analysis of bridges: p. 6.
http://congress.cimne.com/eccomas/proceedings/compdyn2011/compdyn2011_full/532.pdf
(Pridobljeno 12. 08. 2013.)
- [9] Computers and Structures, Inc. 2011. SAP2000 v15.0.1. Csi_OAPI_Documentation.chm: datoteka v kateri so našteje in opisane funkcije programskega vmesnika SAP2000.
- [10] Visual Basic for Applications. 2013.
https://en.wikipedia.org/wiki/Visual_Basic_for_Applications (Pridobljeno 15. 08. 2013.)
- [11] SearchWinDevelopment. 2007. Visual Basic .NET.
<http://searchwindevelopment.techtarget.com/definition/Visual-Basic-NET>
(Pridobljeno 15. 08. 2013.)
- [12] Microsoft MSDN. Common Language Runtime (CLR).
<http://msdn.microsoft.com/en-us/library/8bs2ecf4.aspx> (Pridobljeno 15. 08. 2013.)
- [13] Microsoft MSDN. Overview of the .NET Framework.
<http://msdn.microsoft.com/en-us/library/zw4w595w.aspx> (Pridobljeno 15. 08. 2013.)
- [14] McNeel, R. et al. Rhinoceros: računalniški program za tridimenzionalno modeliranje
<http://www.rhino3d.com/> (Pridobljeno 22. 07. 2013.)

- [15] Davidson, S., Grasshopper: vtičnik programa Rhinocerus za parametrično zasnov geometrije.
<http://www.grasshopper3d.com> (Pridobljeno 22. 07. 2013.)
- [16] Database Directory. What is SQL?.
<http://www.databasedir.com/what-is-sql> (Pridobljeno 16. 08. 2013.)
- [17] SQLCourse. What is SQL?.
<http://www.sqlcourse.com/intro.html> (Pridobljeno 16. 08. 2013.)
- [18] Klinc, R. Program in namenski programski vmesnik na strežniku Fakultete za gradbeništvo in geodezijo Univerze v Ljubljani, ki zagotavlja komunikacijo z relacijsko bazo podatkov o potresnem delovanju. Osebna komunikacija. (10. 06. 2013.)
- [19] Oracle. MySQL: sistem za upravljanje z relacijskimi podatkovnimi bazami.
<https://www.mysql.com> (Pridobljeno 16. 08. 2013.)
- [20] Freeservers. What is PHP?.
<http://www.freeservers.com/WebHosting101/WhatIsPHP.html> (Pridobljeno 16.08. 2013.)
- [21] MathWorks. MATLAB: računalniški program za numerično analizo in obdelavo podatkov.
<http://www.mathworks.com/products/matlab/> (Pridobljeno 07. 08. 2013)
- [22] MathWorks Documentation Center. MATLAB COM Integration.
http://www.mathworks.com/help/matlab/matlab_external/introducing-matlab-com-integration.html (Pridobljeno 17. 08. 2013.)
- [23] Persson, P. O., Strang, G. DistMesh - A Simple Mesh Generator in MATLAB.
<http://persson.berkeley.edu/distmesh/> (Pridobljeno 16. 07. 2013.)
- [24] Microsoft MSDN. What is XAML?.
<http://msdn.microsoft.com/en-us/library/cc295302.aspx> (Pridobljeno 17. 08. 2013.)
- [25] JSON. Introducing JSON.
<http://www.json.org> (Pridobljeno 26. 07. 2013.)
- [26] Fajfar, P. 1984. Dinamika gradbenih konstrukcij. Ljubljana, Fakulteta za arhitekturo, gradbeništvo in geodezijo: 550 str.
- [27] Fajfar, P. 1984. Dinamika gradbenih konstrukcij: Ljubljana, Fakulteta za arhitekturo, gradbeništvo in geodezijo: str. 132
- [28] Computers and Structures, Inc. SAP2000 - 25 Open Application Programming Interface: Watch & Learn.
<https://www.youtube.com/user/computersNstructures?feature=watch>
(Pridobljeno 02. 04. 2013.)
- [29] Computers and Structures, Inc. Update linked bridge model using Excel API.
<https://wiki.csiamerica.com/display/tp/Update+linked+bridge+model+using+Excel+API>
(Pridobljeno 02. 04. 2013.)
- [30] Computers and Structures, Inc. Sample Plugin 1.
<https://wiki.csiamerica.com/display/kb/Sample+Plugin+1> (Pridobljeno 28. 05. 2013.)

- [31] Kalny, O., Abell, M. 2012. Tendon.
<https://wiki.csiamerica.com/display/kb/Tendon> (Pridobljeno 14. 04. 2013.)
- [32] Kalny, O., Abell, M. 2012. Tendon.
<https://wiki.csiamerica.com/display/kb/Mass> (Pridobljeno 14. 04. 2013.)
- [33] Beg, D. (ur.), Pogačnik, (ur.). 2009. Priročnik za projektiranje gradbenih konstrukcij po Evrokod standardih: Evrokod 8 projektiranje potresno odpornih konstrukcij. Ljubljana, Inženirska zbornica Slovenije: str. 8-7 – 8-18
- [34] SIST EN 1998-1:2005 - Evrokod 8: Projektiranje potresnoodpornih konstrukcij - 1. del: Splošna pravila, potresni vplivi in pravila za stavbe.
- [35] SIST EN 1998-1:2005/A101 - Evrokod 8: Projektiranje potresnoodpornih konstrukcij - 1. del: Splošna pravila, potresni vplivi in pravila za stavbe - Nacionalni dodatek.
- [36] The European Strong-Motion Database.
http://www.isesd.hi.is/ESD_Local/frameset.htm (Pridobljeno 08. 05. 2013.)
- [37] Pacific Earthquake Engineering Research Center. Ground Motion Database.
http://peer.berkeley.edu/peer_ground_motion_database (Pridobljeno 18. 09. 2013.)
- [38] Center for Engineering Strong Motion Data.
<http://www.strongmotioncenter.org/> (Pridobljeno 09. 05. 2013.)
- [39] Istituto Nazionale di Geofisica e Vulcanologia. ITACA.
<http://scrooge.mi.ingv.it> (Pridobljeno 09. 05. 2013.)
- [40] Swiss Seismological Service.
<http://arclink.ethz.ch/> (Pridobljeno 09. 05. 2013.)
- [41] GeoNet. Earthquake Resources.
<http://info.geonet.org.nz/display/appdata/Earthquake+Resources> (Pridobljeno 09. 05. 2013.)
- [42] National Research Institute for Earth Science and Disaster Prevention. Strong-motion Seismograph Networks (K-NET, KiK-net). <http://www.kyoshin.bosai.go.jp>
(Pridobljeno 09. 05. 2013.)
- [43] Pacific Earthquake Engineering Research Center. PEER NGA Database Flatfile.
<http://peer.berkeley.edu/nga/flatfile.html> (Pridobljeno 16. 05. 2013.)
- [44] Pacific Earthquake Engineering Research Center. NGA Flatfile Documentation.
http://peer.berkeley.edu/nga/NGA_Documentation.pdf (Pridobljeno 16. 05. 2013.)
- [45] Stanford University. Baker Research Group.
<http://www.stanford.edu/~bakerjw> (Pridobljeno 26. 05. 2013.)
- [46] Baker Research Group. A computationally efficient ground motion selection algorithm for matching a target response spectrum mean and variance.
http://www.stanford.edu/~bakerjw/gm_selection.html (Pridobljeno 17. 05. 2013.)
- [47] Microsoft. Visual Studio: integrirano razvojno okolje za razvoj aplikacij na ogrodju .NET
www.microsoft.com/visualstudio/eng (Pridobljeno 32. 06. 2013.)

- [48] Microsoft. The Bing Maps Platform.
<https://www.microsoft.com/maps/> (Pridobljeno 12. 07. 2013.)
- [49] Microsoft. Microsoft Expression Blend: računalniški program za načrtovanje uporabniških grafičnih vmesnikov
<https://www.microsoft.com/expression/eng/#blend> (Pridobljeno 14. 07. 2013.)
- [50] Newton-King, J., 2013. JSON.NET.
<https://json.codeplex.com/> (Pridobljeno 12. 06. 2013.)
- [51] CodePlex. 2011. DotNetZip Library.
<https://dotnetzip.codeplex.com/> (Pridobljeno 17. 06. 2013.)
- [52] Microsoft. 2012. Bing Maps Windows Presentation Foundation (WPF) Control, Version 1.0.
<https://www.microsoft.com/en-us/download/details.aspx?displaylang=en&id=27165>
(Pridobljeno 05. 07. 2013.)
- [53] Northwind, M. 2005. Links with arbitrary text in a RichTextBox.
<http://www.codeproject.com/Articles/9196/Links-with-arbitrary-text-in-a-RichTextBox>
(Pridobljeno 27. 08. 2013.)
- [54] Gagne, M. 2007. How to write a loading circle animation in .NET?.
<http://www.codeproject.com/Articles/14841/How-to-write-a-loading-circle-animation-in-NET> (Pridobljeno 23. 08. 2013.)
- [55] Microsoft MSDN. .NET Framework 4.5
<http://msdn.microsoft.com/en-us/library/vstudio/w0x726c2.aspx> (Pridobljeno 14. 09. 2013.)
- [56] Persson, P. O., Strang, G. 2004. A SIMPLE MESH GENERATOR IN MATLAB: str. 1.
<http://persson.berkeley.edu/distmesh/persson04mesh.pdf> (Pridobljeno 16. 07. 2013.)
- [57] Signed distance function.
https://en.wikipedia.org/wiki/Signed_distance_function (Pridobljeno 19. 07. 2013.)
- [58] Persson, P. O. 2005. Mesh Generation for Implicit Geometries: p.16, 20-22.
<http://persson.berkeley.edu/thesis/persson-thesis-color.pdf> (Pridobljeno 19. 07. 2013.)
- [59] NURBS.
<http://sl.wikipedia.org/wiki/NURBS> (Pridobljeno 17. 09. 2013.)
- [60] Altman, M. About Nonuniform Rational B-Splines – NURBS.
<http://web.cs.wpi.edu/~matt/courses/cs563/talks/nurbs.html> (Pridobljeno 29. 07. 2013.)
- [61] Computers and Structures, Inc. 2013. ETABS: računalniški program s poudarkom na analizi in zasnovi gradbenih konstrukcij po metodi končnih elementov.
<http://www.csiamerica.com/etabs2013> (Pridobljeno 26. 09. 2013)

Ostali viri:

Albright, S. C. 2007. VBA for modelers: developing decision support systems with Microsoft Excel. Mason, South-Western Cengage Learning, cop.: 700 str.

Box, D. 1998. Essential COM. Boston, Addison Wesley, cop.: 440 str.

Cheng, H. 2010. C for engineers and scientists: an interpretive approach. Boston, McGraw-Hill Higher Education, cop.: 911 str.

Dolenc, M. 2001. Programske komponente za analizo konstrukcij po metodi končnih elementov. Doktorska disertacija. Ljubljana, Univerza v Ljubljani, Fakulteta za gradbeništvo in geodezijo: 205 str.

Fajfar, P. 1977. Račun večetažnih konstrukcij pri seizmični obtežbi. Ljubljana, RSS: 228 str.

Fajfar, P., Dolšek, M. 2012. A practice-oriented estimation of the failure probability of building structures. Earthquake Engineering Structural Dynamics. 41, 3: 531-547

Hart-Davis, G. 2005. Mastering Microsoft VBA. Hoboken, Wiley Publishing, cop.: 707 str.

Irvine, K., Gaddis, T. 2012. Advanced visual basic 2010. Boston, Addison Wesley: 675 str.

Kuntjoro, W. 2005. An introduction to the finite element method. Singapore, McGraw-Hill, cop.: 248 str.

Larsen, R. W. 2009. Engineering with Excel. Upper Saddle River, Pearson/Prentice Hall, cop.: 552 str.

Löffelmann, K., Purohit, S. C. 2011. Microsoft Visual Basic 2010 developer's handbook. Sebastopol, O'Reilly Media, cop.: 987 str.

Nagle, T., Saff, E. B., Snider, A. D., West, B. 2004. Fundamentals of differential equations and boundary value problems. Boston, Pearson Addison-Wesley: 862 str.

Slovensko društvo informatike. islovar: terminološki slovar informatike.
<http://www.islovar.org> (Pridobljeno 02. 09. 2013.)

»Ta stran je namenoma prazna.«

SEZNAM PRILOG

PRILOGA A	PRIMER STRUKTURIRANIH PODATKOV V FORMATU JSON	A1
PRILOGA B	IZPIS NEUSPELO PRENEŠENIH DATOTEK AKCELEROGRAMOV	B1
B.1	Manjkajoči akceleroگرامи zabeleženi v prvi smeri horizontalne ravnine	B1
B.2	Manjkajoči akceleroگرامи zabeleženi v drugi smeri horizontalne ravnine	B2
B.3	Manjkajoči akceleroگرامи zabeleženi v vertikalni ravnini	B3
PRILOGA C	NAVODILA IN POSTOPEK IZDELAVE ENOSTAVNEGA PROGRAMA S POMOČJO PROGRAMSKEGA VMESNIKA ZA SAP2000	C1

»Ta stran je namenoma prazna.«

PRILOGA A: PRIMER STRUKTURIRANIH PODATKOV V FORMATU JSON

```
[
  {
    "Earthquake Name": "Helena. Montana-01",
    "Station Name": "Carroll College",
    "Record Sequence Number": 1,
    "Magnitude": 6,
    "Mtype": "MW",
    "EpiD": 6.31,
    "HypD": 8.71,
    "JB-Dist": null,
    "ClstD": null,
    "Vs30": 659.6,
    "YEAR": 1935,
    "MODY": 1031,
    "Lat": 46.61,
    "Lon": -111.96,
    "SLat": 46.58,
    "SLon": -112.03,
    "Sa_FN": [
      0.153925907085873,
      0.153925907085873,
      0.152080381577912,
      .
      .
      .
      0.000663044826623,
      0.00058904850348,
      0.000526766397056
    ],
    "Sa_FP": [
      0.18736740694633,
      0.18736740694633,
      0.179021130682159,
      .
      .
      .
      0.001217986794697,
      0.001079922870224,
      0.000964352947885
    ]
  }
]
```

»Ta stran je namenoma prazna.«

PRILOGA B: IZPIS NEUSPELO PRENEŠENIH DATOTEK AKCELEROGRAMOV

Priloga B.1: Manjkajoči akceleroگرامi zabeleženi v prvi smeri horizontalne ravnine (5 od skupno 3551)

Datum\Ura	Num	ImeDatoteke
27-May-2013 23:05:20	260	260_MAMMOTH_D-HCF_H1XXX.at2

Datum\Ura	Num	ImeDatoteke
27-May-2013 23:22:57	930	930_BIGBEAR_SB2-H1XXX.at2

Datum\Ura	Num	ImeDatoteke
27-May-2013 23:30:14	1171	1171_KOCAELI_SKR-H1XXX.at2

Datum\Ura	Num	ImeDatoteke
27-May-2013 23:45:23	1298	1298_CHICHI_HWA053-H1XXX.at2

Datum\Ura	Num	ImeDatoteke
28-May-2013 00:12:17	2136	2136_BEARCTY_0551c_H1XXX.at2

Priloga B.2: Manjkajoči akcelorogrami zabeleženi v drugi smeri horizontalne ravnine (18 od skupno 3551)

Datum\Ura	Num	ImeDatoteke
28-May-2013 00:50:00	29	29_PARKF_C02-H2XXX.at2
28-May-2013 00:51:16	168	168_IMPVALL_H-QKP-H2XXX.at2
28-May-2013 00:51:24	177	177_IMPVALL_H-E02_H2XXX.at2
28-May-2013 00:52:17	260	260_MAMMOTH_D-HCF_H2XXX.at2
28-May-2013 00:52:20	267	267_VICT_QKP-H2XXX.at2
28-May-2013 00:53:34	361	361_COALINGA_H-PGD-H2XXX.at2
28-May-2013 00:53:36	365	365_COALINGA_H-VC5-H2XXX.at2
28-May-2013 00:53:44	378	378_COALINGA_A-MIT-H2XXX.at2
28-May-2013 00:54:44	473	473_MORGAN_WVE-H2XXX.at2
28-May-2013 00:54:44	474	474_MORGAN_WNE-H2XXX.at2
28-May-2013 00:54:44	475	475_MORGAN_WSE-H2XXX.at2
28-May-2013 00:55:00	500	500_HOLLISTR_D-HD4-H2XXX.at2
28-May-2013 00:56:03	599	599_WHITTIER_A-FLO-H2XXX.at2
28-May-2013 00:56:33	660	660_WHITTIER_A-MAN-H2XXX.at2
28-May-2013 00:58:19	805	805_LOMAP_SPG-H2XXX.at2
28-May-2013 01:20:29	1635	1635_MANJIL_190-H2XXX.at2
28-May-2013 01:21:24	1753	1753_NWCHINA3_X411N-H2XXX.at2
28-May-2013 01:21:45	1774	1774_HECTOR_0535a_H2XXX.at2

Priloga B.3: Manjkajoči akcelorogrami zabeleženi v vertikalni ravnini (57 od skupno 3551)

Datum\Ura	Num	ImeDatoteke
28-May-2013 02:09:17	43	43_LYTLECR_CSMXXX.at2
28-May-2013 02:09:27	61	61_SFERN_C08XXX.at2
28-May-2013 02:09:36	78	78_SFERN_PDLXXX.at2
28-May-2013 02:09:47	98	98_HOLLISTR_A-G01XXX.at2
28-May-2013 02:09:50	104	104_NCALIF_D-SCAXXX.at2
28-May-2013 02:09:50	105	105_NCALIF_D-SCPXXX.at2
28-May-2013 02:09:50	106	106_OROVILLE_A-ORVXXX.at2
28-May-2013 02:11:22	256	256_MAMMOTH_C-XMGXXX.at2
28-May-2013 02:11:30	275	275_MAMMOTH_H-XMMXXX.at2
28-May-2013 02:12:42	361	361_COALINGA_H-PGDXXX.at2
28-May-2013 02:12:58	390	390_COALINGA_A-YUBXXX.at2
28-May-2013 02:13:22	425	425_SMART1_25C0XXX.at2
28-May-2013 02:13:24	436	436_BORAH.MS_CPPAXXX.at2
28-May-2013 02:13:24	437	437_BORAH.MS_CPPBXXX.at2
28-May-2013 02:13:24	438	438_BORAH.MS_PBFXXX.at2
28-May-2013 02:13:24	439	439_BORAH.MS_TANXXX.at2
28-May-2013 02:13:24	442	442_BORAH.AS_BORXXX.at2
28-May-2013 02:13:24	443	443_BORAH.AS_CEMXXX.at2
28-May-2013 02:13:24	444	444_BORAH.AS_HAUXXX.at2

Datum\Ura	Num	ImeDatoteke
28-May-2013 02:13:24	446	446_MORGAN_A1EXXX.at2
28-May-2013 02:13:46	473	473_MORGAN_WVEXXX.at2
28-May-2013 02:13:46	474	474_MORGAN_WNEXXX.at2
28-May-2013 02:13:46	475	475_MORGAN_WSEXXX.at2
28-May-2013 02:13:59	496	496_NAHANNI_S2XXX.at2
28-May-2013 02:14:10	517	517_PALMSPR_DSPXXX.at2
28-May-2013 02:14:18	530	530_PALMSPR_PSAXXX.at2
28-May-2013 02:14:19	532	532_PALMSPR_CLJXXX.at2
28-May-2013 02:15:31	651	651_WHITTIER_A-L01XXX.at2
28-May-2013 02:15:36	660	660_WHITTIER_A-MANXXX.at2
28-May-2013 02:16:09	719	719_SUPERST_B-BRAXXX.at2
28-May-2013 02:16:09	720	720_SUPERST_B-CALXXX.at2
28-May-2013 02:16:10	722	722_SUPERST_B-KRNXXX.at2
28-May-2013 02:16:10	723	723_SUPERST_B-PTSXXX.at2
28-May-2013 02:16:10	724	724_SUPERST_B-PLSXXX.at2
28-May-2013 02:16:10	725	725_SUPERST_B-POEXXX.at2
28-May-2013 02:16:10	726	726_SUPERST_B-WLFXXX.at2
28-May-2013 02:16:10	727	727_SUPERST_B-SUPXXX.at2
28-May-2013 02:17:19	803	803_LOMAP_WVCXXX.at2
28-May-2013 02:17:20	805	805_LOMAP_SPGXXX.at2

Datum\Ura	Num	ImeDatoteke
28-May-2013 02:17:34	822	822_ROERMOND_GSHXXX.at2
Datum\Ura	Num	ImeDatoteke
28-May-2013 02:18:48	973	973_NORTHR_GARXXX.at2
Datum\Ura	Num	ImeDatoteke
28-May-2013 02:19:21	1048	1048_NORTHR_STCXXX.at2
Datum\Ura	Num	ImeDatoteke
28-May-2013 02:20:04	1122	1122_KOZANI_EDEXXX.at2
Datum\Ura	Num	ImeDatoteke
28-May-2013 02:20:11	1132	1132_KOZANI_B-GR1XXX.at2
Datum\Ura	Num	ImeDatoteke
28-May-2013 02:20:11	1134	1134_KOZANI_C-GRPXXX.at2
Datum\Ura	Num	ImeDatoteke
28-May-2013 02:26:58	1299	1299_CHICHI_HWA054XX.at2
Datum\Ura	Num	ImeDatoteke
28-May-2013 02:38:48	1591	1591_CHICHI_TTN047XX.at2
Datum\Ura	Num	ImeDatoteke
28-May-2013 02:39:31	1628	1628_STELIAS_059v27UP.at2
Datum\Ura	Num	ImeDatoteke
28-May-2013 02:39:31	1629	1629_STELIAS_059v20UP.at2
Datum\Ura	Num	ImeDatoteke
28-May-2013 02:39:54	1694	1694_NORTH392_MU2XXX.at2
Datum\Ura	Num	ImeDatoteke
28-May-2013 02:39:54	1696	1696_NORTH392_HOWXXX.at2
Datum\Ura	Num	ImeDatoteke
28-May-2013 02:39:55	1701	1701_NORTH392_WILXXX.at2
Datum\Ura	Num	ImeDatoteke
28-May-2013 02:40:00	1720	1720_NORTH392_MCSXXX.at2
Datum\Ura	Num	ImeDatoteke
28-May-2013 02:40:03	1732	1732_NORTH392_KATXXX.at2
Datum\Ura	Num	ImeDatoteke
28-May-2013 03:31:47	3549	3549_XXX.at2
Datum\Ura	Num	ImeDatoteke
28-May-2013 03:31:47	3550	3550_XXX.at2
Datum\Ura	Num	ImeDatoteke
28-May-2013 03:31:47	3551	3551_XXX.at2

»Ta stran je namenoma prazna.«

PRILOGA C: NAVODILA IN POSTOPEK IZDELAVE ENOSTAVNEGA PROGRAMA S POMOČJO PROGRAMSKEGA VMESNIKA ZA SAP2000

Na primeru bomo po korakih opisali postopek izdelave enostavnega programa, ki temelji na uporabi programskega vmesnika za SAP2000, v programu Microsoft Excel s pomočjo programskega jezika Visual Basic for Applications. Pri tem se ne bomo spuščali v podrobno razlago osnov programiranja kot so deklaracija spremenljivk ali funkcij ter uporabe pogojnih odvisnosti in zank. Osredotočili se bomo na vzpostavitev povezave s programom SAP2000, funkcije programskega vmesnika in logični proces. Sam program bo omogočal račun velikosti in mesta maksimalnega pomika poljubno podprte pravokotne betonske plošče izbranih dimenzij.

Pred začetkom programiranja, je potrebno na delovnem listu zasnovati vmesnik za podajanje vhodnih parametrov in predvideti mesto rezultatov. V tem primeru se bomo sklicevali na vmesnik prikazan na sliki C.1. Za vhodne podatke smo izbrali dimenzije pravokotne plošče, način podpiranja posameznih robov, število končnih elementov, podatke o elastičnem modulu in Poissonovem številu, vrednost enakomerno porazdeljene koristne obtežbe ter varnostna faktorja za lastno in koristno obtežbo.

A	B	C	D	E	F	G	H	I	J	K
1	Enostaven program za izračun maksimalnega pomika poljubno podprte pravokotne betonske plošče s pomočjo programa Sap2000									
2										
3	Vhodni podatki:									
4	Pravokotna betonska plošča:			Robni pogoji:		Mreža končnih elementov:		Velikost enega KE		
5	Dimenzija v smeri X	10.00 m	Rob A	Prosto	Št. KE v smeri X:	40	0.2500 m			
6	Dimenzija v smeri Y	8.00 m	Rob B	Togo	Št. KE v smeri Y:	32	0.2500 m			
7	Debelina plošče	0.25 m	Rob C	Prosto						
8			Rob D	Togo						
9										
10	Material:		Enakomerno porazdeljena koristna ploskovna obtežba q:				Obtežba:			
11	Elastični modul	3.E+07 kN/m2					5.00 kN/m2			
12	Poissonovo število	0.2					Varnostni faktor za lastno težo g:		1.35	
13							Varnostni faktor za koristno obtežbo q:		1.5	
14									Izračunaj	
15										
16	Rezultat:		Koordinata X	Koordinata Y	Točka					
17	Maksimalen pomik:	-0.0106 m	0.0000 m	-4.0000 m	663					
18			0.0000 m	4.0000 m	695					
19										
20										
21										
22										
23										
24										

Rob C

g, q

Y

Z ⊗

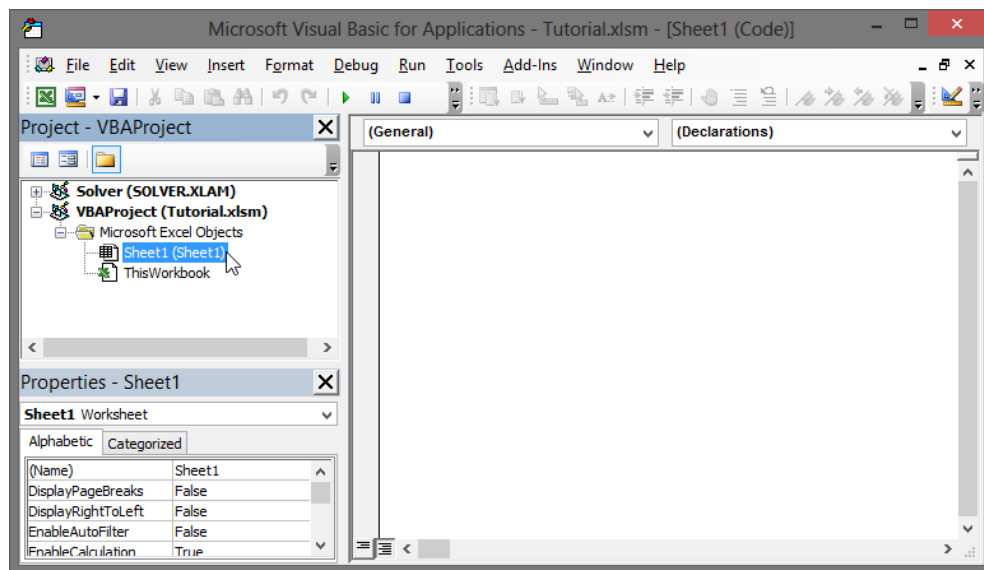
X

Rob B

Rob A

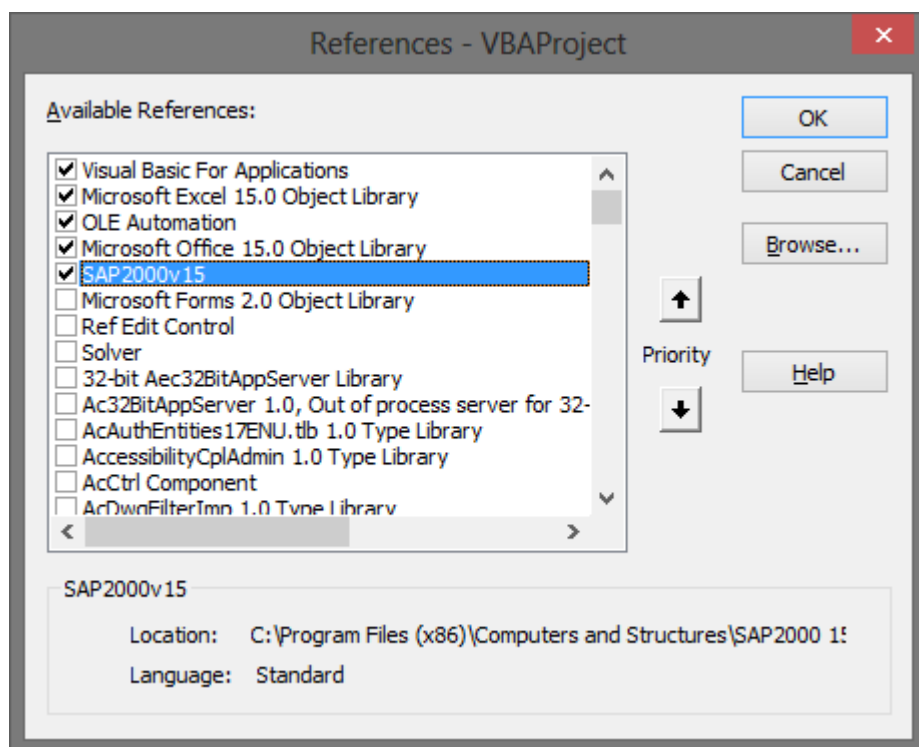
Slika C.1: Vmesnik za vnos podatkov in prikaz rezultatov

Okno za vnos programske kode lahko odpremo tako, da na zavihku Razvijalec (angl. Developer) kliknemo na gumb Visual Basic ali pritisnemo kombinacijo tipk alt+F11. Kodo pišemo v delovni zvezek, kjer se nahaja vmesnik za vnos podatkov – slika C.2.



Slika C.2: Okno za vnos programske kode

Sedaj je potrebno izbrati datoteko, v kateri se nahajajo javno dostopne metode za delo s programom SAP2000, do katerih dostopamo s pomočjo programskega vmesnika. To storimo s klikom na meni Orodja (angl. Tools) in klikom na gumb Reference (angl. References). V novo odprtem oknu prikazanem na sliki C.3, med naštetimi poiščemo ter obkličujemo dinamično povezovalno knjižnico z imenom SAP2000v15, oziroma SAP2000, če uporabljamo starejšo verzijo programa. V kolikor je ni na seznamu jo lahko ročno dodamo. Knjižnica z imenom SAP2000v15.dll se nahaja v namestitvenem direktoriju programa SAP2000.



Slika C.3: Okno za dodajanje referenc

Program začnemo z deklaracijo objektov SapObject, preko katerega so dostopne metode za upravljanje s samim programom SAP2000 in SapModel preko katerega dostopamo do metod za upravljanje z modelom konstrukcij.

```
Set SapObject = New SAP2000v15.SapObject  
'Set SapObject = New SAP2000.SapObject 'če uporabljamo verzijo programa < 15  
  
Set SapModel = SapObject.SapModel
```

S funkcijo SapObject.ApplicationStart zaženemo program SAP2000.

```
SapObject.ApplicationStart
```

Večina funkcij programskega vmesnika, predvsem tiste s katerimi upravljamo z modelom konstrukcije, po izvajanju nalog vrnejo vrednost, s katero povedo uspešnost realizacije svojih nalog. Če so te uspešno izvršene, funkcija vrne vrednost 0, v nasprotnem primeru pa vrednost, ki je od 0 različna. Ta podatek z vsako uporabo funkcije programskega vmesnika shranimo v spremenljivko. V vsakem koraku lahko preverimo njeno vrednost in s tem nadziramo proces ali opozorimo uporabnika na mesto, kjer je prišlo do napake. V tem primeru se ne bomo osredotočili na sprotno preverjanje izvršenih nalog, kljub temu pa je potrebno vrnjene vrednosti funkcije, ki odražajo njihovo uspešnost realizacije shraniti v spremenljivko. Zato najprej definiramo spremenljivko podatkovnega tipa Long z imenom 'ret' (krajše za angl. return). Ime spremenljivke je v splošnem poljubno.

```
Dim ret As Long
```

Nato pa funkcije programskega vmesnika uporabljamo v obliki:

ret = funkcija_programskega_vmesnika(njeni argumenti)

S funkcijo SapModel.InitializeNewModel inicializiramo nov model, pri tem pa z dodatnim parametrom določimo izbrane enote. Enote kN, m, C podamo s številom 6. Kode ostalih enot so podane v originalnih navodilih programskega vmesnika. Funkcija SapModel.NewBlank generira nov prazen model.

```
ret = SapModel.InitializeNewModel (6)  
  
ret = SapModel.File.NewBlank
```

Pred uporabo funkcij s katerimi bomo definirali material v programu SAP2000, iz delovnega lista za vnos podatkov preberemo vrednost elastičnega modula iz celice C11 in Poissonovega števila iz celice C12, ter ju shranimo v spremenljivki.

```
Dim elastičniModul As Double  
Dim poissonovoSt As Double  
  
elastičniModul = Range("C11").Value  
poissonovoSt = Range("C12").Value
```

Nov material ustvarimo s funkcijo SapModel.PropMaterial.SetMaterial, kjer s prvim argumentom podamo njegovo ime, z drugim pa njegov tip. S številom 2 definiramo material tipa beton, kar mu

samodejno pripiše specifično težo z vrednostjo 23.56 kN/m^3 . Slednjo vrednost bi lahko spremenili z uporabo funkcije `SapModel.PropMaterial.SetWeightAndMass`, vendar bomo v tem primeru uporabili pred-nastavljeno vrednost. S funkcijo `SapModel.PropMaterial.SetMPIsotropic`, ustvarjenemu materialu z imenom 'Bet', pripišemo vrednost elastičnega modula, Poissonovega števila in toplotnega koeficienta. Zadnjega v našem primeru ne potrebujemo, vendar ga moramo kljub temu podati.

```
ret = SapModel.PropMaterial.SetMaterial("Bet", 2)

ret = SapModel.PropMaterial.SetMPIsotropic("Bet", elasticniModul, poissonovoSt,
0.0000001)
```

Preden lahko izrišemo ploščo, je potrebno iz podanih dimenzij izračunati štiri koordinate, ki definirajo njene vogale. Dimenzije preberemo iz celic C5 in C6, ter na podlagi teh definiramo koordinate vozlišč tako, da bo izhodišče globalnega koordinatnega sistema vedno na sredini plošče, ta pa bo ležala v ravnini X-Y.

```
Dim dimenzijaX As Double
Dim dimenzijaY As Double

dimenzijaX = Range("C5").Value
dimenzijaY = Range("C6").Value

Dim x(3) As Double
Dim y(3) As Double
Dim z(3) As Double

x(0) = -dimenzijaX / 2
x(1) = dimenzijaX / 2
x(2) = x(1)
x(3) = x(0)

y(0) = -dimenzijaY / 2
y(1) = y(0)
y(2) = dimenzijaY / 2
y(3) = y(2)

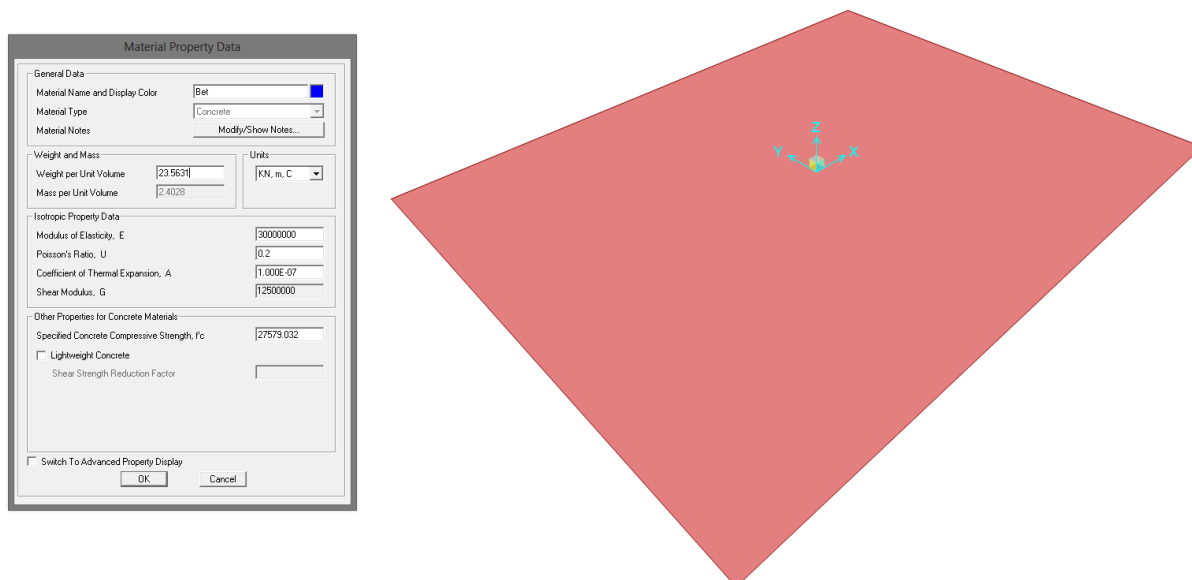
z(0) = 0
z(1) = z(0)
z(2) = z(0)
z(3) = z(0)
```

S funkcijo `SapModel.AreaObj.AddByCoord` lahko sedaj izrišemo ploščo, na podlagi koordinat njenih vogalov. S prvim argumentom podamo število točk, s pomočjo katerih definiramo ploskev, naslednji trije argumenti pa predstavljajo polja oziroma vektorje posameznih koordinat teh točk. Peti argument predstavlja spremenljivko, v katero se po izvršitvi funkcije shrani ime ustvarjene plošče.

```
Dim Ime As String

ret = SapModel.AreaObj.AddByCoord(4, x, y, z, Ime)
```

Na sliki C.4 je prikazan model konstrukcije in material, ki smo ga do te faze ustvarili v programu SAP2000.



Slika C.4: Model in material, ki smo ga do te faze ustvarili v programu SAP2000

Za določitev prečnega prereza, iz delovnega lista za vnos podatkov iz celice C7, najprej shranimo podatek o debelini plošče.

```
Dim debelinaPlosce As Double  
debelinePlosce = Range("C7").Value
```

Nato s funkcijo SapModel.PropArea.SetShell_1 ustvarimo prečni prerez tipa Shell z imenom 'Prerez', kar povemo s prvim argumentom. Z drugim, ki ima vrednost 2 povemo, da gre za tip končnega elementa Shell-thick, s četrtem argumentom podamo ime predhodno ustvarjenega materiala, s šestim in sedmim argumentom pa debelino prereza. Ostali argumenti in kode tipov končnih elementov so opisane v originalnih navodilih programskega vmesnika. S funkcijo SapModel.AreaObj.SetProperty nato ustvarjeni prerez z imenom 'Prerez' pripišemo predhodno ustvarjeni plošči z imenom, ki je shranjeno v spremenljivki 'Ime'.

```
ret = SapModel.PropArea.SetShell_1("Prerez", 2, True, "Bet", 0, debelinaPlosce, debelinaPlosce)  
ret = SapModel.AreaObj.SetProperty(Ime, "Prerez")
```

Ploščo bomo sedaj razdelili na večje število končnih elementov. Iz celic J5 in J6 najprej shranimo podatke o številu končnih elementov v smeri globalne osi X in Y.

```
Dim steviloKeX As Integer  
Dim steviloKeY As Integer  
steviloKeX = Range("J5").Value  
steviloKeY = Range("J6").Value
```

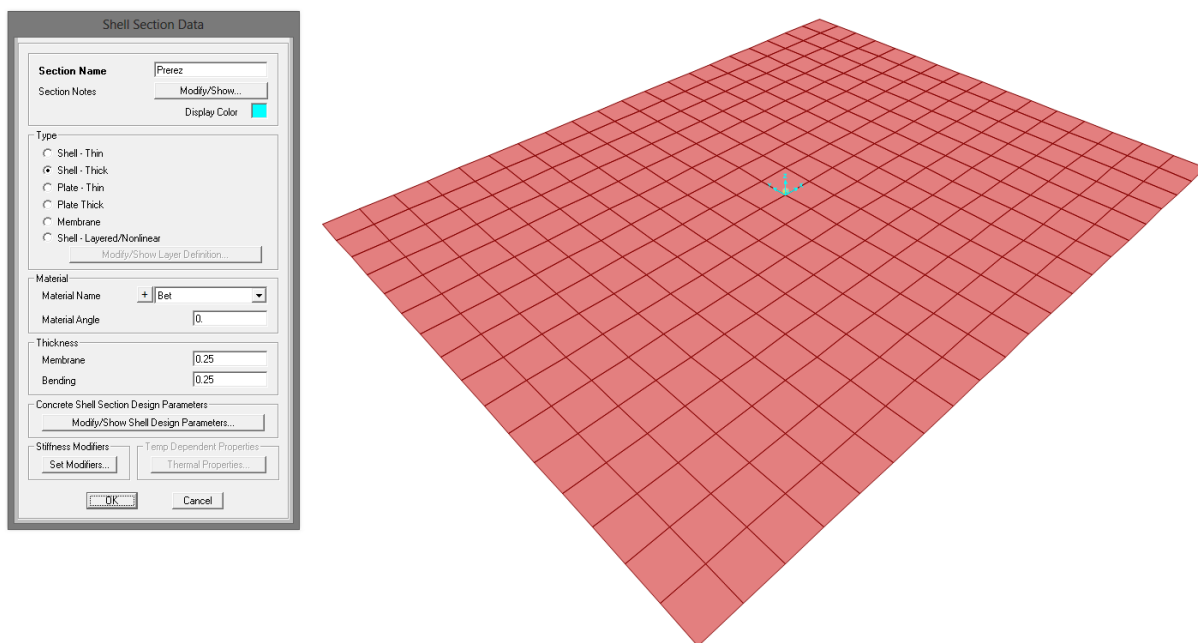
V funkciji SapModel.EditArea.Divide podamo ime plošče, ki jo želimo razdeliti. S številom 1 povemo, da jo želimo razdeliti na določeno število končnih elementov. Ostale opcije so opisane v originalnih navodilih programskega vmesnika. Naslednja dva parametra predstavljata spremenljivki, v

kateri se po delitvi shrani število ustvarjenih končnih elementov in njihova imena. Z zadnjima dvema parametroma podamo število razdelitev plošče z imenom, ki je shranjeno v spremenljivki Ime, v smeri osi X ter Y.

```
Dim steviloKE As Long
Dim imenaKE() As String

ret = SapModel.EditArea.Divide(Ime, 1, steviloKE, imenaKE, steviloKeX,
steviloKeY)
```

Na sliki C.5 je prikazana razdelitev plošče na končne elemente in definiran prerez v program SAP2000.



Slika C.5: Definiran prerez plošče in njena razdelitev na končne elemente v programu SAP2000

Naslednji korak je določitev robnih pogojev za vsak rob plošče posebej. Prikazali bomo postopek določitve za rob A, analogno pa določimo tudi robne pogoje vseh ostalih robov.

Robove podpremo v vseh vozliščih končnih elementov. Za to potrebujemo njihova imena ali pa morajo biti ta vozlišča označena, ter vektor oziroma polje šestih elementov, ki definira način podpiranja. Vsak izmed šestih elementov vektorja označuje stanje ene prostostne stopnje, katerih vrstni red je enak U1, U2, U3, R1, R2, R3. V kolikor je posamezna vrednost enaka 'True', je pripadajoča prostostna stopnja podprta, če pa je vrednost enaka 'False', je prostostna stopnja sproščena. Za točke na robu plošče najprej določimo vektor podpiranja. V začetku predpostavimo, da je vsaka točka togo podprta in vsem elementom vektorja pripišemo vrednost 'True'. Glede na vrednost celice F5 pa naknadno, po potrebi, sprostimo ustrezne prostostne stopnje. V primeru vrtljivo vpetega roba sprostimo vse zasuke, v primeru prostega roba pa vseh šest prostostnih stopenj sprostimo tako, da vsem elementom vektorja pripišemo vrednost 'False'.

```
Dim robA(5) As Boolean
Dim i As Integer
```

```
For i = 0 To 5
    robA(i) = True
Next

Select Case Range("F5").Value
    Case "Vrtljivo"
        For i = 3 To 5
            robA(i) = False
        Next
    Case "Prosto"
        For i = 0 To 5
            robA(i) = False
        Next
    Next
End Select
```

Nato s funkcijo `SapModel.SelectObj.CoordinateRange` označimo vse točke, ki se nahajajo v definiranem območju. Parametri funkcije so meje območja, ki jih vnesemo v obliki minimalnih in maksimalnih koordinat za posamezno smer. Te podamo z ustreznimi koordinatami vogalov plošče, ki smo jih definirali na začetku. Z ostalimi parametri povemo, da želimo izbrati le točke, ki ležijo v danem območju, ne pa tudi ostalih elementov konstrukcije. Vsi parametri funkcije so opisani v originalnih navodilih programskega vmesnika.

```
ret = SapModel.SelectObj.CoordinateRange(x(0), x(1), y(0), y(0), 0, 0, False, ,
True, True, False, False, False, False)
```

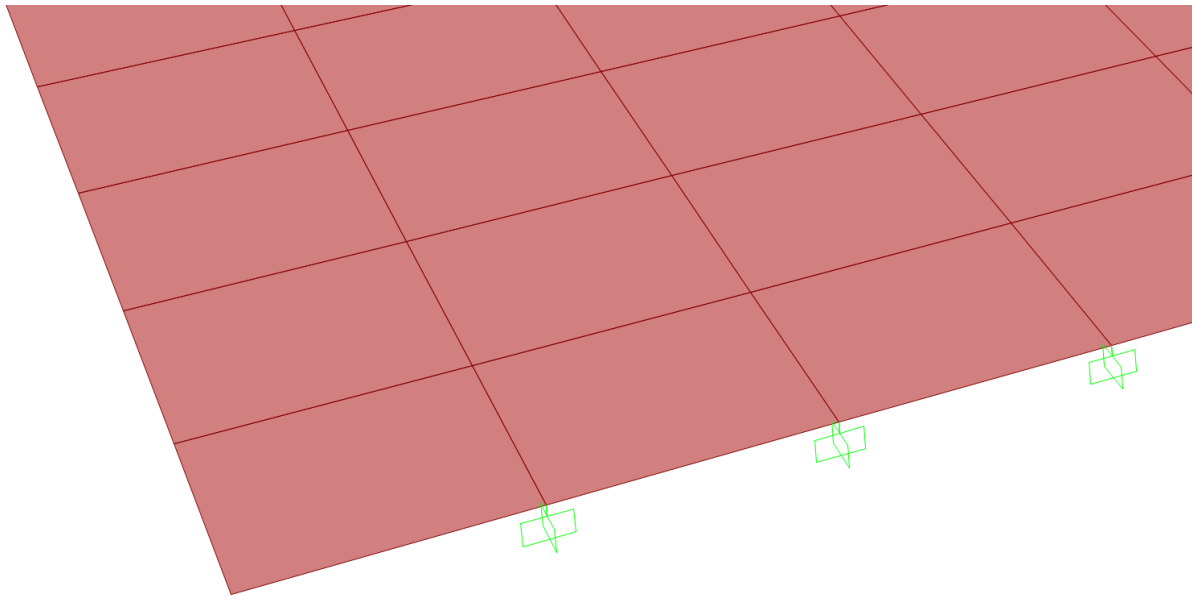
S funkcijo `SapModel.PointObj.SetRestraint` nato vsem označenim točkam pripišemo vektor podpiranja, ki ga predstavlja spremenljivka `robA`. V kolikor podpiramo označene točke, kar povemo z zadnjim argumentom z vrednostjo 2, je lahko prvi argument, ki predstavlja ime točke poljuben, saj ga funkcija ne upošteva. S funkcijo `SapModel.SelectObj.ClearSelection` nato odznačimo predhodno označene točke.

```
ret = SapModel.PointObj.SetRestraint("", robA, 2)

ret = SapModel.SelectObj.ClearSelection
```

Princip določitve robnih pogojev za ostale tri robove je enak, prilagoditi je potrebno le vhodne parametre ter koordinate območja za izbiro točk.

Ker vogalna točka pripada dvema robovoma, se lahko zgodi, da ob izbiri nepodprtega roba ostane nepodprta tudi vogalna točka, slika C.6. V kolikor je sosednji rob podprt, to v večini primerov ne odraža realnega stanja, zato bomo program zasnovali tako, da bo imela vogalna točka ob enem nepodprtem robu enake robne pogoje kot podprt rob, kateremu prav tako pripada. Problem bi sicer lahko rešili tako, da bi uporabnik dodatno podal robne pogoje za vse vogalne točke, vendar bomo raje napisali preprost algoritem s katerim bomo to rešili enostavneje, pri tem pa prikazali nekaj dodatnih funkcij programskega vmesnika.



Slika C.6: Nepodprta vogalna točka plošče

Princip bomo pokazali na vogalu A, ki je hkrati del roba A ter D. Analogno velja tudi za vse ostale vogale. Ker bomo v algoritmu večkrat preverjali podprtost posameznih točk, napišemo funkcijo z imenom 'AliJeTockaPodprta' v izogib neprestanemu ponavljanju iste metode. Vhodni parameter funkcije je ime izbrane točke, funkcija pa vrne vrednost 'True' v kolikor je točka podprta oziroma vrednost 'False', če ni. V njej, s pomočjo funkcije SapModel.PointObj.GetRestraint, pridobimo vektor podpiranja za izbrano točko. Nato preverimo vsak element vektorja in le v primeru, da so sproščene vse prostostne stopnje, funkcija vrne vrednost 'False'.

```
Function AliJeTockaPodprta (ByVal imeTocke As String) As Boolean
    Dim Podpore(5) As Boolean
    Dim JePodprto As Boolean
    Dim j As Integer

    JePodprto = False
    ret = SapModel.PointObj.GetRestraint(imeTocke, Podpore)

    For j = 0 To UBound(Podpore)
        If Podpore(j) = True Then JePodprto = True
    Next

    AliJeTockaPodprta = JePodprto
End Function
```

Algoritem pričnemo s preverjanjem podprtosti vogalne točke A, za katero v našem primeru vemo, da je njeno ime enako '1'. V kolikor je ta podprta, preskočimo na preverjanje naslednje vogalne točke, drugače pa nadaljujemo z algoritmom in s funkcijo SapModel.SelectObj.CoordinateRange, podobno kot pri označitvi točk na robu ploskve, označimo sosednjo točko, ki leži na enem izmed robov vogalne točke. Funkcija SapModel.SelectObj.GetSelected nam v argumentu imeObjekta vrne ime označene točke, ki ga shranimo v spremenljivko. S funkcijo SapModel.SelectObj.ClearSelection nato izbrano točko odznačimo. Postopek ponovimo za sosednjo točko vogala, ki leži na drugem robu.

Z algoritmom nadaljujemo zgolj v primeru, ko je podprta samo ena izmed obeh sosednjih točk, kar za vsako preverimo s funkcijo 'AliJeTockaPodprta'. V tem primeru nato s funkcijo SapModel.PointObj.GetRestraint pridobimo vektor podpiranja podprte sosednje točke in ga s funkcijo SapModel.PointObj.SetRestraint pripišemo vogalni točki.

```
Dim sosednjaTocka1 As String
Dim sosednjaTocka2 As String
Dim steviloObjektov As Long
Dim tipObjekta() As Long
Dim imeObjekta() As String
Dim Podpore(5) As Boolean

'Vogala
If AliJeTockaPodprta("1") = False Then
    ret = SapModel.SelectObj.CoordinateRange(x(0) + dimenzijaX / steviloKeX, x(0) + dimenzijaX / steviloKeX, y(0), y(0), 0, 0, False, , True, True, False, False, False, False)
    ret = SapModel.SelectObj.GetSelected(steviloObjektov, tipObjekta, imeObjekta)
    sosednjaTocka1 = imeObjekta(0)
    ret = SapModel.SelectObj.ClearSelection

    ret = SapModel.SelectObj.CoordinateRange(x(0), x(0), y(0) + dimenzijaY / steviloKeY, y(0) + dimenzijaY / steviloKeY, 0, 0, False, , True, True, False, False, False, False)
    ret = SapModel.SelectObj.GetSelected(steviloObjektov, tipObjekta, imeObjekta)
    sosednjaTocka2 = imeObjekta(0)
    ret = SapModel.SelectObj.ClearSelection

    If AliJeTockaPodprta(sosednjaTocka1) = True Xor
    AliJeTockaPodprta(sosednjaTocka2) = True Then
        If AliJeTockaPodprta(sosednjaTocka1) = True Then
            ret = SapModel.PointObj.GetRestraint(sosednjaTocka1, Podpore)
            ret = SapModel.PointObj.SetRestraint("1", Podpore)
        Else
            ret = SapModel.PointObj.GetRestraint(sosednjaTocka2, Podpore)
            ret = SapModel.PointObj.SetRestraint("1", Podpore)
        End If
    End If
End If
```

Postopek ponovimo za ostale vogalne točke. Prilagoditi je potrebno zgolj koordinate območja za izbiro sosednjih in imena vogalnih točk.

Sedaj bomo definirali še obtežbo in ustvarili statični obtežni primer, ter linearno kombinacijo stalne in koristne obtežbe. Najprej s funkcijo SapModel.LoadPatterns.Add definiramo obtežbo tipa LIVE z imenom 'Koristna'. Ostali tipi obtežb so naštet v originalnih navodilih programskega vmesnika.

```
ret = SapModel.LoadPatterns.Add("Koristna", LTYPE_LIVE)
```

Nato iz celice J10 v spremenljivko shranimo vrednost podane enakomerne ploskovne obtežbe, ki jo nato s funkcijo SapModel.AreaObj.SetLoadUniform pripišemo ploskovnim elementom. Obtežbo lahko pripišemo ploskovnim elementom glede na njihova imena, označenim ploskovnim elementom ali ploskovnim elementom združenim v skupine. Program SAP2000 samodejno ustvari skupino z imenom 'ALL', v kateri so združeni vsi, ne samo ploskovni elementi modela.

Funkcija `SapModel.AreaObj.SetLoadUniform` lahko vpliva zgolj na ploskovne elemente, zato v našem primeru kljub temu uporabimo slednjo možnost, saj nanašamo enakomerno obtežbo na celotno ploščo. S prvim argumentom podamo ime, z zadnjim, ki ima vrednost 'Group' pa povemo, da je to ime skupine in ne elementa. Z drugim argumentom pripišemo obtežbi tipa LIVE z imenom 'Koristna', ki smo jo ustvarili v prejšnjem koraku, vrednost obtežbe, ki je podana s tretjim parametrom. S parametrom, ki ima vrednost 10 povemo, da obtežba deluje v smeri gravitacije, v predzadnjem parametru pa podamo ime koordinatnega sistema. Vsi ostali parametri so opisani v originalnih navodilih programskega vmesnika.

```
Dim obtezba As Double

obtezba = Range("J10").Value

ret = SapModel.AreaObj.SetLoadUniform("ALL", "Koristna", obtezba, 10, True, "Global", Group)
```

Tako smo definirali koristno obtežbo z imenom 'Koristna', obtežbo lastne teže z imenom 'DEAD', pa določi program SAP2000 samodejno. Ti dve obtežbi bomo sedaj združili v obtežno kombinacijo, ki jo najprej definiramo s funkcijo `SapModel.RespCombo.Add`, ter ji pripišemo ime 'Kombinacija'. S parametrom 0 povemo, da gre za linearno kombiniranje obtežb.

```
ret = SapModel.RespCombo.Add("Kombinacija", 0)
```

Tej kombinaciji nato dodamo obe obtežbi s funkcijo `SapModel.RespCombo.SetCaseList`. S prvim argumentom navedemo ime kombinacije v katero dodajamo obtežbe, drugi argument z vrednostjo LoadCase pa pomeni, da v kombinacijo združujemo obtežbe in ne drugih kombinacij. S tretjim argumentom podamo ime obtežbe, ki jo želimo dodati, s četrtem pa vrednost varnostnega faktorja. Tega smo predhodno prebrali iz celic J11 ter J12, ter ju shranili v spremenljivki.

```
Dim gamaG As Double
Dim gamaQ As Double

gamaG = Range("J11").Value
gamaQ = Range("J12").Value

ret = SapModel.RespCombo.SetCaseList("Kombinacija", LoadCase, "DEAD", gamaG)
ret = SapModel.RespCombo.SetCaseList("Kombinacija", LoadCase, "Koristna", gamaQ)
```

Program SAP2000 pri definiciji novega modela samodejno ustvari tudi obtežni primer za določitev nihajnih časov konstrukcije z imenom 'MODAL'. Za hitrejšo analizo lahko ta obtežni primer s funkcijo `SapModel.Analyze.SetRunCaseFlag` odstranimo iz seznama obtežnih primerov za analizo ali pa ga s funkcijo `SapModel.LoadCases.Delete` pobrišemo, kar smo storili v tem primeru.

```
ret = SapModel.LoadCases.Delete("MODAL")
```

Pred analizo je potrebno model shraniti na disk s funkcijo `SapModel.File.Save`. V edinem parametru funkcije je potrebno podati pot in ime datoteke s končnico sdb.

Pri tem je treba opozoriti, da mora mapa, kamor želimo shraniti datoteko, predhodno obstajati.

Da bi se v danem primeru izognili težavam z delovanjem programa na različnih računalnikih, smo model shranili v začasno mapo z imenom 'TEMP', ki se nahaja na vsakem računalniku, za njen dostop pa ne potrebujemo administratorskih pravic. V kolikor slednje ni ovira, je priporočljivo mapo na disku ustvariti s pomočjo programske kode in vanjo shraniti model.

```
Dim zacasnaMapa As String  
  
zacasnaMapa = IIf(Environ$("tmp") <> "", Environ$("tmp"), Environ$("temp"))  
  
ret = SapModel.File.Save(zacasnaMapa + "\Primer.sdb")
```

Analizo poženemo s pomočjo funkcije SapModel.Analyze.RunAnalysis.

```
ret = SapModel.Analyze.RunAnalysis
```

Po končani analizi najprej pobrišemo območje namenjeno izpisu rezultatov.

```
Range("C17:F999").Clear
```

Dobra praksa je, da s funkcijo SapModel.Results.Setup.DeselectAllCasesAndCombosForOutput najprej odstranimo morebitne obtežne primere ali kombinacije, ki so izbrane za branje rezultatov.

```
ret = SapModel.Results.Setup.DeselectAllCasesAndCombosForOutput
```

Nato s funkcijo SapModel.Results.Setup.SetComboSelectedForOutput izberemo kombinacijo obtežb z imenom 'Kombinacija', katere rezultate bomo uvozili iz programa SAP2000.

```
ret = SapModel.Results.Setup.SetComboSelectedForOutput("Kombinacija")
```

Poiskati želimo vrednost in mesta maksimalnega pomika v globalni smeri Z. Najprej s funkcijo SapModel.Results.JointDisplAbs prenesemo vse absolutne pomike in zasuke izbranih točk za obtežni primer ali kombinacijo, ki smo jo izbrali v prejšnjem koraku. Enako kot velja za funkcijo SapModel.AreaObj.SetLoadUniform, s katero smo ploskovnim elementom pripisali obtežbo, lahko s funkcijo SapModel.Results.JointDisplAbs prenesemo pomike točk, za kar moramo poznati njihovo ime oziroma morajo biti točke označene ali pa združene v skupine. Izbrali bomo zadnjo možnost, saj želimo prenesti pomike vseh točk in nato med njimi poiskati maksimalno vrednost. Z drugim argumentom GroupElm povemo, da želimo prenesti pomike točk združene v skupino z imenom 'ALL', katerega podamo s prvim argumentom funkcije. Vsi ostali parametri predstavljajo spremenljivke, v katere funkcija shrani prenesene rezultate. V našem primeru nas zanima zgolj spremenljivka U3, v kateri so shranjeni pomiki vseh točk v smeri Z. S preprosto zanko nato v polju U3 poiščemo maksimalno vrednost in jo izpišemo v celico C17. V naslednjem koraku bomo potrebovali še spremenljivko imeTock, kjer so shranjena vsa imena točk, katerih pomike smo prenesli.

```
Dim imeTock() As String  
Dim Elm() As String  
Dim LCase() As String  
Dim steviloRezultatov As Long  
Dim StepType() As String  
Dim StepNum() As Double
```

```

Dim U1() As Double
Dim U2() As Double
Dim U3() As Double
Dim R1() As Double
Dim R2() As Double
Dim R3() As Double

ret = SapModel.Results.JointDisplAbs("ALL", GroupElm, steviloRezultatov, imeTock,
Elm, LCase, StepType, StepNum, U1, U2, U3, R1, R2, R3)
Dim k As Integer
Dim indeks() As Integer
Dim Maksimum As Double
Dim Stevec As Integer

Maksimum = 0
Stevec = 0

For k = 0 To UBound(U3)
    If U3(k) < Maksimum Then
        Maksimum = U3(k)
    End If
Next

Range("C17").NumberFormat = "0.0000\ \m"
Range("C17").Value = Round(Maksimum, 5)

```

Izpisali bomo še koordinate maksimalnega pomika. Ni nujno, da ta nastopi zgolj v eni točki, zato bomo poiskali vse, kjer se ta pojavi. To lahko storimo tako, da se z zanko sprehodimo čez vse vrednosti pomikov shranjenih v spremenljivki U3. V njej primerjamo posamezne vrednosti z maksimalno vrednostjo pomika, ki smo jo poiskali v prejšnjem koraku. V kolikor sta vrednosti enaki, shranimo zaporedno mesto trenutnega pomika v polju U3 oziroma njegov indeks v polju U3. Na podlagi tega indeksa, lahko v spremenljivki imeTock, v kateri so shranjena imena vseh točk, katerih pomike smo prenesli s funkcijo SapModel.Results.JointDisplAbs, poiščemo imena točk z maksimalnimi pomikom. To storimo v zadnji zanki, kjer s pomočjo funkcije SapModel.PointObj.GetCoordCartesian prenesemo koordinate točk. Prvi argument funkcije je ime točke, zadnji trije parametri pa predstavljajo spremenljivke v katere se shranijo posamezne koordinate. Koordinati X, Y ter ime točke maksimalnega pomika izpišemo v celice D17, E17 in F17. V kolikor je točk več, zamaknemo zapis naslednje za eno vrstico.

```

For k = 0 To UBound(U3)
    If Round(U3(k), 5) = Round(Maksimum, 5) Then
        ReDim Preserve indeks(Stevec)
        indeks(Stevec) = k
        Stevec = Stevec + 1
    End If
Next

Dim zamikVrstice As Integer
Dim kooX As Double
Dim kooY As Double
Dim kooZ As Double
Dim tockaMaksPomika As String
zamikVrstice = 0

For k = 0 To UBound(indeks)
    tockaMaksPomika = imeTock(indeks(k))
    ret = SapModel.PointObj.GetCoordCartesian(tockaMaksPomika, kooX, kooY, kooZ)
    Range("D17").Offset(zamikVrstice, 0).NumberFormat = "0.0000\ \m"

```

```
Range("D17").Offset(zamikVrstice, 0).Value = kooX  
Range("E17").Offset(zamikVrstice, 0).NumberFormat = "0.0000\ \m"  
Range("E17").Offset(zamikVrstice, 0).Value = kooY  
Range("F17").Offset(zamikVrstice, 0).Value = tockaMaksPomika  
zamikVrstice = zamikVrstice + 1
```

Next

Po končanem prenosu rezultatov zapremo program SAP2000 s funkcijo SapObject.ApplicationExit, kjer z argumentom True povemo, da želimo model pred zapiranjem ponovno shraniti.

```
SapObject.ApplicationExit (True)
```

»Ta stran je namenoma prazna.«