УНИВЕРЗИТЕТ У БЕОГРАДУ

АРХИТЕКТОНСКИ ФАКУЛТЕТ

Милош Т. Манески

# РАЗВОЈ МАКРО КОНАЧНОГ ЕЛЕМЕНТА ПРИМЕНОМ ГРУПНО СУПЕРМАТРИЧНЕ ПРОЦЕДУРЕ ЗА НАБОРАНЕ КОНСТРУКЦИЈЕ

докторска дисертација

Београд, 2018

UNIVERSITY OF BELGRADE

FACULTY OF ARCHITECTURE

Milos T. Maneski

# DEVELOPMENT OF MACRO FINITE ELEMENT BY APLICATION OF GROUP SUPERMATRICE PROCEEDURE FOR FOLDED PLATES STRUCTURES

Doctoral Dissertation

Belgrade, 2018

**ментор**

др Миодраг Несторовић

редовни професор у пензији

Универзитет у Београду Архитектонски факултет

**Чланови комисије**

др Весна Милошевић Митић

редовни професор

Универзитет у Београду Машински факултет

др Растислав Мандић

редовни професор

Универзитет у Београду Грађевински факултет

**ИЗЈАВА ЗАХВАЛНОСТИ**

Желео бих да се захвалим свима који су имали удела при изради докторске дисертације. Свом ментору проф. др Миодрагу Несторовићу који је својим искуством, смерницама и знањем допринео успешном завршетку рада. Члановима комисије проф. др Весни Милошевић-Митић и проф. др Растиславу Мандићу који су активно учествовали у истраживању и помогли да се оствари спој различитих инжењерских наука. Оцу проф. др Ташку Манеском уз кога сам се оформио као инжењер и који је развио програмски пакет KOMIPS у који је интегрисан рачунарски алгоритам. Доц. др Јелени Милошевић на интересовању и разумевању за проблеме у вези са радом. Колегама са посла са којима сам делио заједничке проблеме и који су помогли при форматирању страница и текста. Породици на неизмерној подршци и разумевању.

*Милош Манески*

## ПОДАЦИ О ДОКТОРСКОЈ ДИСЕРТАЦИЈИ

**Наслов дисертације**

РАЗВОЈ МАКРО КОНАЧНОГ ЕЛЕМЕНТА ПРИМЕНОМ ГРУПНО СУПЕРМАТРИЧНЕ ПРОЦЕДУРЕ ЗА НАБОРАНЕ КОНСТРУКЦИЈЕ

**Резиме:**

Предмет истраживања докторске дисертације чини домен пројектовања наборaних конструкција. Код оваквих конструктивних система је изражена међузависност форме и структуре због начина на који преноси оптерећење. Односно, утицај геометрије система у великој мери одређује ефикасност конструкције. Обзиром да конструкција у исто време има носећу функцију и функцију форме у изгледу, сам рад је имао за циљ да сагледа како архитектонске, тако и прорачунске (статичко-динамичке карактеристике) захтеве које конструкција треба да испуни. Мултидисциплинаран приступ проблему је подразумевао израду рачунарских алгоритама за генерисање структуре, затим пројектовање примитива, интеграцију групно суперматричне процедуре у метод коначних елемената, дефинисање макро коначног елемента и прорачун еквиваленте матрице крутости равне плоче са којом се улази у статичко – динамичку анализу конструкције. Приказана је предност параметарског генерисања набора и дефинисање заменског, једноставног, макро коначног елемента погодног за прорачун ГСП процедуром. Имплементација процедуре потврђује иновацију и унапређење процеса архитектонско – инжењерског пројектовања сложених конструктивних система.

**Кључне речи:**
Конструктивни системи, наборане конструкције, генерисање конструкција, примитив, макро коначни елемент, МКЕ, ГСП

**Научна област:**
Архитектура и урбанизам

**Ужа научна област:**
Конструктивни системи

**УДК број:**
624.074:519.6(043.3)

**DATA ON DOCTORAL DISSERTATION**

**Thesis title**

DEVELOPMENT OF MACRO FINITE ELEMENT BY APLICATION OF GROUP SUPERMATRICE PROCEEDURE FOR FOLDED PLATES STRUCTURES

**Abstract:**

The research subject of the doctoral dissertation is within domain of folded plate structures designing. Because of the load distribution way, this type of construction has very strong interdependence between form and structure. Respectively system geometry has significant impact on structure efficiency. Since structure has simultaneously form and bearing function, this paper has a goal to perceive and apply both, architectural and structural (static – dynamic analysis) demands that structure must fulfill. The multidisciplinary problem approach has considered development of software algorithm for generating of structure, then primitive designing, following with integration of the group supermatrices procedure in finite element method, determination of macro finite element and computing of equivalent stiffness matrices of flat plate which is the main input for static- dynamic structure analysis. The advantage of parametric folded plate generator and defining of simple equivalent macro finite element suitable for GPS analysis is presented. Procedure implementation confirms innovation and improvement of architectural – engineering designing of the complex structural systems.

**Key words:**
Structural systems, folded plate structure, generating of structures, primitive, macro finite element, FEM, GSP

**Academic Expertise:**
Architecture and Urbanism

**Field of Academic Expertise:**
Structural Systems

**UDC Number:**
624.074:519.6(043.3)

**САДРЖАЈ:**

# 1. УВОД

## 1.1 Образложење теме - проблематика и предмет истраживања

Уколико покушамо да дефинишемо најшири оквир истраживања, рад се бави проблемом унапређења процеса архитектонског и инжењерског пројектовања. Одређеније, контекст истраживања представља развој и примена научних метода у пројектовању и грађењу високо економичних конструктивних система применом нових технологија. Домен истраживања се ограничава на анализу узајамних веза, принципа, моделирања и статичких прорачуна набораних конструкција. Набориане конструкције се сврставају у групу савремених просторних структура, односно конструктивних система сложених форми и великих распона.

У том смислу истраживање ће се бавити следећим темама и проблемима:

### 1.1.1 Неусаглашеност процеса архитектонско-инжењерског пројектовања, прорачуна и реализације

Веома чест проблем актуелног архитектонског развоја је неконзистентност процеса продукције. Проблеми су јасно уочљиви у тренуцима када идеју концепта треба превести у фазу реализације објеката. Слободоумне архитектонске идеје и комплексне геометријске функције облика често остају без одговарајућег математичке формулације па самим тим и немогућности креирања одговарајућег рачунарског модела, као и избора адекватне нумеричке методе. У циљу превазилажења постојећег проблема неопходно је успоставити дијалог и интерактивни однос различитих аспеката архитектуре и њој сродних научних дисциплина. Специфичан вид изложеног проблема чини питање односа примарног конструктивног система и концепта предложене форме. Класична решења у којима се тежи раздвајању форме објеката од његове носеће структуре применом већ добро познатих, стандардних конструктивних система оправдају став о нерационалности и недовољној ефикасности истих. Сасвим је оправдано сматрати да форму не треба посматрати као неконструктивну презентацију, већ је треба третирати као структуру састављену од одговарајућег материјала. С обзиром да форма, структура и материјал не поседују апсолутно независно садејство, следи да није могуће извршити прецизнију дијагностификацију спровођењем независних анализа. Пре томе потребно је обезбедити усаглашеност комплетног процеса пројектовања уз задовољење свих архитектонско-инжењерских аспеката од примарне форме, преко функције, до завршног конструктивног система.

### 1.1.2 Проблем компјутерског генерисања и структуралне анализе

Архитектонски креиране форме често су засноване на различитим принципима пројектовања, самим тим је потребно урадити одређене апроксимације у циљу рачунарског генерисања и структуралних анализа предметних форми. У процес креирања модела је могуће, па и пожељно, укључити структуралну анализу. Унапређивањем софтвера за генерисање форме и могућност креирања координараних система, тако да они веродостојно прате архитектонску форму, може да омогући истовремени процес пројектовања и напонско-деформацијске анализе елемената структуре. Другим речима могуће је остварити истовремени и међузависни процес који се разликује од уобичајеног приступа где се ови процеси третирају независно. Као уобичајена алатка структуралне анализе користе се програмски пакети засновани на методи коначних елемената (МКЕ). Обзиром на различитост методологија моделовања, тј. описивања геометрије у процесима архитектонског пројектовања и геометрије модела који се користи у анализи коначним елементима (поступак предпроцесирања), укључивање структуралне анализе у процес морфогенезе не представља тривијални задатак. У циљу превазилажења овог проблема развијане су предпроцесорске рутине, модули и генератори са дефинисаним продукционим системима за генерисање геометријских облика заснованим на логици моделовања коначним елементима. Иако напредне такве процедуре, су још увек прилично ограничавајуће у контексту слободнијег архитектонског изражавања.

### 1.1.3 Имплементација групно суперматричне процедуре и методе коначних елемената у процес прорачуна набораних конструкција

Улога архитектуре и њеног креативног размишљања одувек је била сигнификанта у процесу креирања објекта. Имајући у виду да уједно представља и полазну тачку процеса пројектовања, она мора да направи оптимални спој различитих метода, техника, алата и научних достигнућа. Управо пројектант има задатак да успостави везу између процеса пројектовања и унапређења расположивих алата. Увођење дигиталних технологија у процес пројектовања и реализације имплицирало је промену процеса и његово померање у сферу аутоматизованог, симулативног и интерактивног. Интерактивном употребом метода и технологија других дисциплина са циљем унапређења процеса пројектовања, отвара се могућност даљим истраживањима и унапређењима. Развој и примена групно суперматричне процедуре и методе коначних елемената представља квалитативан напредак у процесу пројектовања, тако да пројектанту омогућава већу ефикасност.

### 1.1.4 Проблематика пројектовања набораних конструкција

Предмет истраживања је ограничен на наборане конструктивне системе, обзиром да је код њих, због начина на који преносе утицаје, изражена међузависност форма - структура, односно утицај глобалне геометрије на ефикасност конструкције. Просторне љуске, које су 1950-тих и 1960-тих година развијали Фулер, Ото, Нерви, Ајзлер и др. представљају примере екстремне интеграције конструкције и омотача објекта. Развој ових система је подразумевао коришћење различитих научних метода укључујући и експерименталне метода и израду физичких модела у процесу проналажења адекватних форми које оптерећење преносе претежно мембранским напрезањем. Савремене дигиталне технологије, компјутерско моделовање, анализе, симулације, продукције физичких модела и њихова интеграција у једну целину, нуде могућности за развој нових високо поузданих, прецизних и ефикасних метода.

## 1.2 Циљ истраживања

Из претходно изложеног могу се уочити одређени истраживачки потенцијали са наглашеном могућношћу развоја и унапређења. Може се рећи да предметно истраживање има два циља. Са једне стране рад се фокусира на креирању одговарајућег генератора рачунарских модела путем координраних система који ће бити способан да на адекватан начин креира рачунски модел набора неопходан за структуралну анализу. Он треба да тежи правцу креирања континуално-интегралног, дизајн-анализа процеса за архитектонско и инжењерско пројектовање комплексних форми површинских система у оквиру тродимензионалног просторног аналитичког окружења. При томе треба посебно усмерити истраживање ка проучавању природе набораних конструкција, како линијских тако и површинских. Са друге стране, обзиром да се ради о комплексном просторном конструктивном систему, треба успоставити корелације између познатих чињеница из домена групно суперматричне процедуре и одговарајућег макро коначног елемента наборане конструкције. На том основу се за циљ истраживања поставља развој процедуре за пројектовање и прорачун линијских и површинских конструктивних система наборaних конструкција. Применом теорије група на конструкције које могу бити издељене на коначан број Г-инваријантних подпростора, акценат рада биће усмерен ка смањењу укупног броја рачунских операција у односу на тренутно најзаступљенију методу коначног елемента.

Примена групно суперматричне процедуре на формирање матрице крутости и маса линијских и површинских коначних елемената биће програмски развијена за постојеће програмске пакете. За потребе наборaних конструкција биће развијен макро коначни елемент на бази групно суперматричне процедуре који у себи садржи линијске и површинске конструктивне елементе и који има исто понашање као скуп класичних коначних елемената у истом домену макро континуума.

Основни циљ рада је да укаже на неискоришћене потенцијале имплементације различитих научних дисциплина и технологија у процес пројектовања при креирању архитектонског дискурса. Интегрисани приступ пројектовању захтева дубоко разумевање архитектонско-инжењерске технологије, обзиром да се том технологијом креира, односно да је она алатка пројектантске методологије.

**1.3 Задаци истраживања**

Након утврђених проблема и постављених циљева истраживања, водећи се критеријума крајњег производа, постављају се следећи задаци.

**Задатак 1.**

Истраживање потенцијала моделирања различитих типова наборaних конструкција уз максимално искоришћавање њихових изометријских особина при креирању рачунарских алгоритама за генерисање структура. Такође, применити адекватан начин примене изогеометријске анализе, као универзалне алатке, у области архитектуре, или одређеније у процесу пројектовања конструктивних система набораних конструкција.

**Задатак 2.**

Дефинисање макро коначног елемента са особинама симетрије у циљу смањења броја рачунских операција неопходних за одређивање матрице крутости елемента, односно, напонско-деформацијског стања модела конструкције.

**Задатак 3.**

Развој макро коначног елемента применом групно суперматричне процедуре за добијање матрице крутости елемента па и система који задовољавају услове за примену теорија група. Групно суперматричном процедуром већ је показано да се драстично смањује број рачунских операција неопходних за добијање решења, односно да се одликује значајним квалитативним и квантитативним предностима у односу на конвенцијални метод коначних елемената. Макро елемент ће у себи садржати линијске и површинске коначне елементе.

**Задатак 4.**

Развој групно суперматричне процедуре за решавање система алгебарских и диференцијалних једначина за прорачун набораних конструкција.

**Задатак 5.**

Израда софтвера за генерисање структура набораних конструкција. Израда софтвера за дефинисање матрице крутости и маса макро коначног елемента применом групно суперматричне процедуре. Израда софтвера за прорачун набораних конструкција применом групно суперматричне процедуре.

**Задатак 6.**

Пројектовање набораних конструкција у циљу креирања примера неопходних за валидацију и верификацију оператибилности и ефикасности предложеног решења, као и компарација добијених резултата са резултатима добијених употребом конвенционалне методе коначних елемената.

## 1.4 Радне (полазне) хипотезе

На основу дефинисаног предмета, циљева и задатака наредне претпоставке је потребно проверити истраживањем.

**Хипотеза 1.**

Коришћење поступка којим се интегришу геометријско генерисање и компјутерско моделовање набораних конструкција. Примена оваквог продукционог система и приоритета заснованих на структуралним перформансама у фази концепције би омогућила формална истраживања, артикулацију и адаптацију архитектонских морфологија и унапређење процеса пројектовања.

**Хипотеза 2.**

У изогеометријској анализи почетна геометрија и кинематика се описују истим функцијама. На основу тога је омогућено пројектовање (продукција), анализа (евалуација) и подешавање (рафинација) модела употребом јединственог типа података и елиминисана потреба за конверзијом. Претходно може бити основ за дефинисање алгоритма који ће омогућити једноставније, егзактније и ефикасније пројектовање набораних конструктивних система и анализу коначним елементима.

**Хипотеза 3.**

Обзиром да наборани конструктивни системи спадају у круте конструктивне системе захваљујући својој геометрији и просторном преношењу утицаја, њихово генерисање на основу структуралних перформанси обезбеђује рационално искоришћење конструктивних капацитета форме, односно функционално и структурално јединство компонената система, што уједно омогућава приступ који поседује и креативни потенцијал.

**Хипотеза 4.**

Геометрија набораних конструкција мора да поседује неопходне особине симетричности како би била могућа примена групно суперматричне процедуре. Такође, особине симетричности мора да поседује и изабрани макро коначни елемент.

### 1.5 Научне методе истраживања

Раније дефинисане хипотезе је потребно проверити одређеним научним методама рада. Према томе, у циљу спровођења предметног истраживања предвиђа се употреба метода логичке аргументације, симулације, моделовање и студија случаја. Наведене методе ће имати улогу у одређивању конкретних начина спровођења истраживања, омогућити закључивање о релацијама везаним за дефинисане проблеме, организовање и саопштавање резултата.

#### 1.5.1 Логичка аргументација

Логичка аргументација представља концептуални конструкт, оквир за објашњење, методу која има карактеристике система у погледу: дефиниције, компоненти, начина на које се компоненте повезују, начина на који се систем повезује са осталим системима. Ова метода има за циљ уобличавање кохерентног, логичког, концептуалног система, а као главни задатак има успостављање логичког реда и стварање система од претходно неповезаних или на тај начин неразматраних чињеница. Генерално идеје компоноване од засебних синтетички повезаних јединица припадају структуралистичкој парадигми.

Претпоставке које ће се проверавати предметним истраживањем, подржавају становиште да се коришћењем научних метода може унапредити процес пројектовања такође припадају структуралистичкој парадигми. Претходно наведено није неосновано у контексту архитектонско-грађевинске праксе нити представља новину обзиром да архитекте користе физичке моделе како би итеративно тестирале и модификовале своје идеје још од ренесансе. Транзиција на виртуални модел, његово тестирање и коришћење повратних информација за модификацију у процесу изналажења решења је, у неку руку, логичан континуитет. Без намере да се на овом месту исцрпно разматра тема односа архитектуре и средстава њене продукције, чињеница је да у основи рада технолошка унапређења узрокују квантитативне и квалитативне промене процеса пројектовања.

Истраживање ће се спровести применом интерних и екстерних тактика. Интерна тактика која ће бити примењена у истраживању је математичка репрезентација и компјутерско моделовање. Овом тактиком се математичким везама описује геометријски облик. Логички систем ће бити уоквирен описивањем понашања реалних структура математичким појмовима, односно репрезентација емпиријских реалности системом математичких законитости и релација. Компјутерски програм ће бити развијен на основу претпоставке да је нумеричким изразима могуће дефинисати просторне релације које ће служити описивању различитих конфигурација.

### 1.5.2 Симулација и моделовање

У симулационим истраживањима се сазнање о одрађеном аспекту реалности може добити репродуковањем те реалности посредством одређеног медија. Симулација представља реплику реалности или антиципирану реалност. У конкретном случају методологија је погодна за коришћење, имајући у виду размеру и комплексност проблема који ће се разматрати. Након постављања логичког експланаторног система симулациона истраживања ће помоћи да се концептуални систем тестира, емпиријски провери, али и да се на основу добијених повратних информација, коригује или допуни.

Када је у питању сама стратегија, симулационим истраживањима се прави реплика реалног контекста, у конкретном случају моделује се проблем у циљу спровођења анализе у оквирима такве поставке. Овде је такође битно направити разлику између презентације и симулације. Реч презентација се користи да означи слике које приказују објекат. У том смислу, сви архитектонски цртежи, уколико нису део ширег истраживачког програма, представљају презентације. С друге стране, симулација подразумева претпоставку реалног контекста која је динамички интерактивна у смислу да се манипулацијом одређених фактора добијају резултати у виду корисних повратних података. Симулиране интеракције представљају рефлексије интеракција реалног контекста.

У контексту овог рада термини симулација и моделовање су тесно повезани. Термин модел се користи да означи, не само статичне репрезентације модела објекта у размери, већ представља реплику апстрахованих физичких законитости реалног проблема. У складу са постојећом класификацијом модела, у оквирима овог рада ће се користити математички модели. Математички модели представљају систем нумеричких кодова, тј. апстраховану дескрипцију реалног проблема. Компјутерски модели есенцијално представљају математичке моделе. Под компјутерским моделом се подразумевају 2 D слике на компјутерском екрану које имају илузију треће димензије (дубине). Међутим, под дубином је, између осталог, могуће подразумевати и способност динамичке промене услова (нпр. напрезање грађевинских конструкција у конкретном случају). Моделирање конструкције је креирање идеализоване или поједностављене репрезентације реалног система на ком се интерпретира понашање конструкције за неко дејство. Представља кључни корак у анализи и пројектовању, јер грешке и пропусти у моделирању могу да буду узрок проблема у функционисању конструктивног система.

За рад је од значаја веза између метода симулације и логичке аргументације, обзиром да ће се симулација користити за тестирање логичком аргументацијом предложене концепције. Коришћење компјутерских модела за симулацију, доприноси преплитању ове две методологије, обзиром да је сваки компјутерски програм формално-математички систем и да је процес превођења реалности у програмски језик задатак логичке аргументације.

### 1.5.3 Студија случаја

Студија случаја представља емпиријско испитивање, проверу реалног контекста или феномена са становишта једног одређеног или више аспеката. Карактеристике ове методе су: (1) фокус на једном или више случајева, који се проучавају у свом реалном контексту; (2) капацитет да објасни каузалне везе; (3) ослањање на више извора података; (4) способност генерализације. Генерално ова стратегија омогућава добијање специфичних података комбиновањем више метода и тактика. У конкретном случају ће се користити у циљу да се кроз конкретни пројектни задатак дефинисан за потребе овог рада, који представља симулацију реалне, практичне ситуације, тестирају поставке логичке аргументације коришћењем развијене пројектантске алатке.

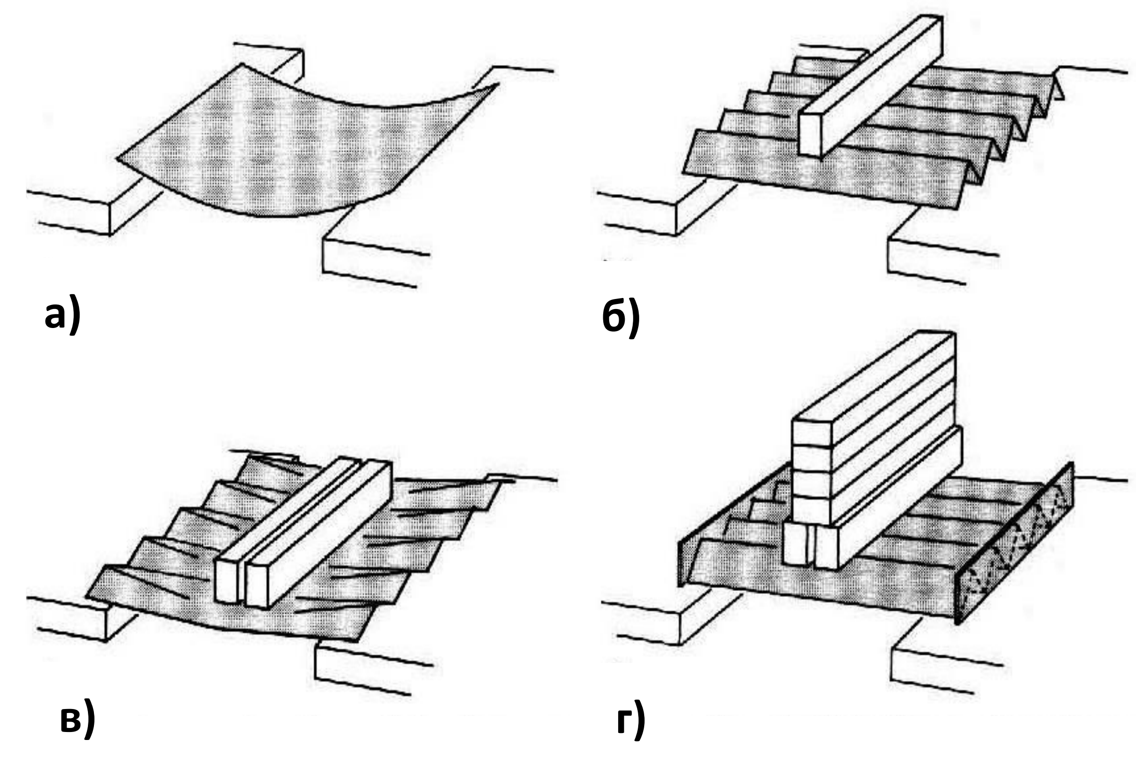
Предност поступка интеграције поменутих метода огледа се у комплементарности и тријангулацији, коришћењу предности и добрих страна односно неутралисању слабости и ограничених карактеристика појединачних метода. На тај начин се обезбеђује богатство информација (ослањањем на више извора података) и повећава се јачина аргумената, уверљивост и валидност добијених резултата.

## 2. НАБОРАНЕ КОНСТРУКЦИЈЕ

### 2.1 Дефиниција набора

Наборане конструкције представљају скуп равних плоча круто повезаних дуж својих ивица на такав начин да су способне да приме оптерећење без формирања греда дуж заједничких ивица. Ивица коју формирају две плоче довољно је крута да преузме функцију појасне греду, уз услов да угао између ових плоча није близак равном углу. Како би се додатно повећала крутост, односно носивост набора на крајеве се могу ставити попречне дијафрагме, такозване чеоне шајбне.

Улогу набора у носивости конструкције је најлакше илустровати на простом практичном примеру папира формата А4 сл. 2.1.




Слика 2.1.1. Модел од папира-носивост набора (а, б, в, г)

а) слободно ослоњен папир може се упоредити са танком равном плочом која није способна да носи ни сопствену тежину

б) савијен папир тако да формира набор, не само да је способан да носи сопствену тежину већ може да прихвати и оптерећење

в) са повећањем оптерећења долази до исправљања набора и губитка носивости г) уколико крајеве набора укрутимо уз помоћ дијафрагми, постиже се вишеструко повећање носивости

Можемо закључити да се формирањем набора повећава статичка висина пресека, што за резултат има већу крутост конструкције. Увођењем дијафрагми, односно додатним укрућењем ивица набора добијају се просторне конструкције које су у стању да уз малу дебљину плоча премосте значајне распоне и прихвате потребно спољно оптерећење.

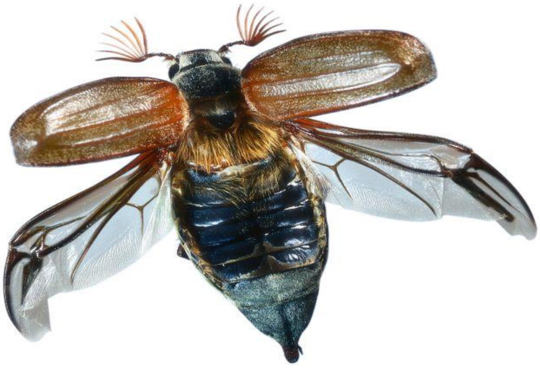
### 2.2 Кратак историјат

Принцип савијања у циљу добијања просторних структуралних форми је познат од давнина. Може се рећи да је идеја набора стара колико и настајање папира. Древна јапанска техника савијања папира оригами представља уједно и креирање првих модела набора. Такође, у природи се могу срести многе форме које на најбољи начин илуструју сродност просторних структура у архитектури и примери флоре и фауне. Наборана просторна структура која се среће код одређених листова биљки, шкољки, крила птица и инсеката је дугим временским периодом полако усвајана и развијана како би се применила у новом, техничком контексту архитектуре (Hachem, Karni, & Hanor, 2004). У том смислу може се рећи да у природи већ постоје одговарајући конструктивни облици које је потребно препознати и применити. Они доприносе већој стабилности и крутости форме и омогућавају лакше извођење комплексних покрета (Vincent, 2000).

Слика 2.2.1. Лист траве

Слика 2.2.2. Лист палме

Слика 2.2.3. Крила птица



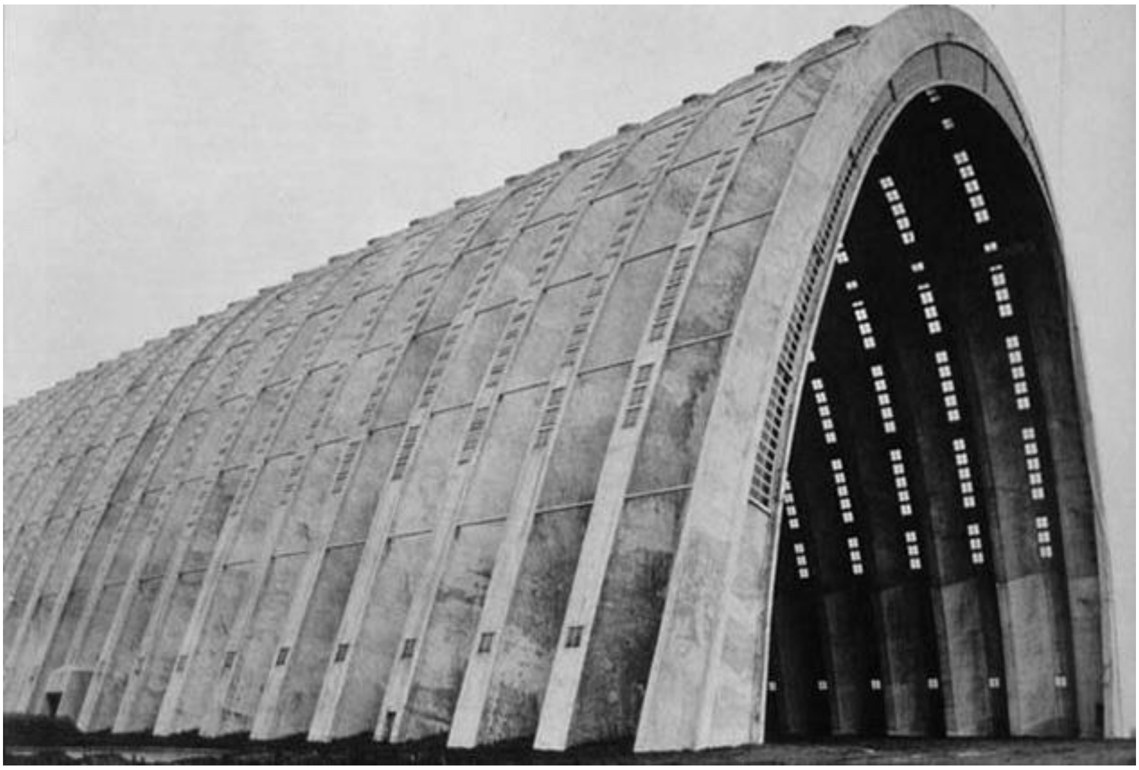
Слика 2.2.4. Крила инсекта

Слика 2.2.5. Морски пуж

Слика 2.2.6. Морска шкољка

И поред чињенице да постоји аналогија узајамних веза набора у природи и архитектури, идеја да се набирањем формирају круте конструкције великих распона јавила се тек почетком XX века. Немогућност раније реализације просторних набора у пракси вероватно лежи у чињеници комплексности прорачуна конструкције и у недовољној развијености прорачунских метода. Напредовањем математике и успостављањем статичких корелација отворила се могућност употребе набора за различита архитектонско-грађевинска решења. Набори поседују потенцијал разноврсног обликовања, па су тиме јако погодни за пројектовање јединствених форми. Иако данас постоје напредни програми за статичке прорачуне и приметан је значајан пораст истраживања и конструисања форми набора код пројектаната, њихова примена у пракси заузима тек незнатан проценат изведених конструкција. Управо у томе лежи велики истраживачки потенцијал.

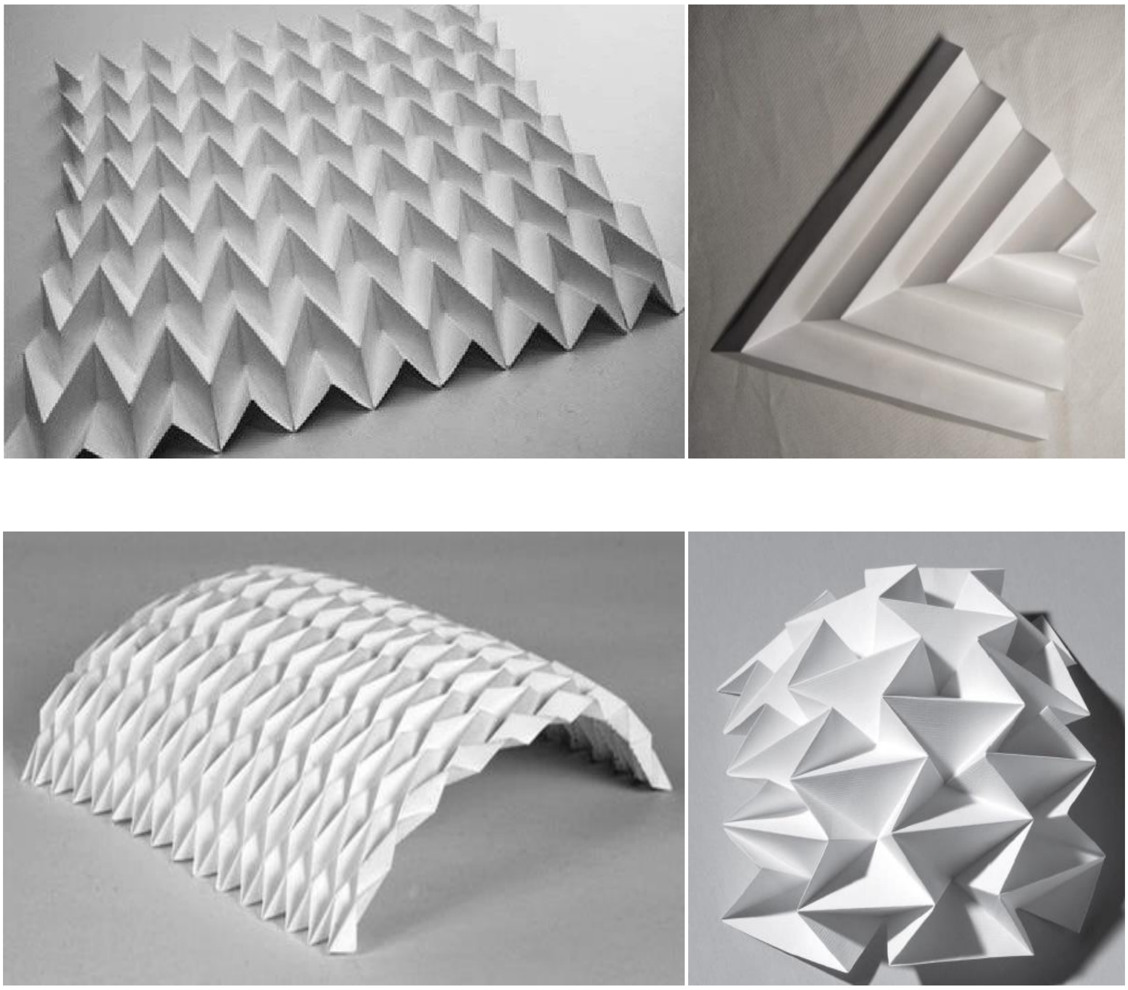
Аеродромски хангар изграђен у Орлију, Француска 1923. године се сматра првом изведеном конструкцијом такве врсте коју је конструисао инжењер Е. Фресине (Freyssinet). Конструкција се састоји од танких параболичних лукова „V“ облика (слика 2.2.7.). Елементи су изведени од преднапрегнутих бетонских елемената при чему је по први пут користио покретну оплату. Дужина хангара је 300 m, распон лукова 80 m и стрела је висине 56 m, док је дебљина зидова лукова свега 9 cm. Обзиром на димензије објекта и тадашња сазнања, мора се признати да се радило о веома смелом архитектско-грађевинском подухвату свога времена.



Слика 2.2.7. Аеродромски хангар Орли, Француска

Након прве успешно изведене грађевине Л. Морети (Morreti) и Р. Флајшман (Flayshman) су почетком педесетих година прошлог века обавили испитивања носивости на папирним моделима. Треба напоменути да је због својих конструктивних карактеристика јако једноставно направити разнолике папирне моделе набора (слика 2.2.8.). Резултати истраживања су недвосмислено потврдили велику крутост, монолитност и значајан капацитет носивости. (Jackson, 2011)

Д. Гајгер и Х. Бергер се сматрају инжењерима који су познати по примени олакшаних кровних конструкција великих распона. Значајан део њихових истраживања био је посвећен и набораним конструкцијама. Иако су познати по примени мешовитих система и такозваног шаблона са радијалним кабловима, ипак се могу сматрати особама које су значајно допринеле развоју свести о њиховој савременој примени.



Слика 2.2.8. Модели од папира

Заједно са развојем науке и напретком технолошких достигнућа, развијала се и примена набора у савременој архитектури. Данас, широм света постоји поприличан број изведених објекта, као и разних студија које потврђују завидан потенцијал примене. Концепт набора је недвосмислено веома привлачан како грађевинској индустрији тако и архитектонској, јер осим што пружа завидне техничке карактеристике, односно способност да премости велике распоне, исто тако поседује велики потенцијал архитектонског обликовања. На слици 2.2.9. је приказан прототип модела од дрвених плоча дебљине 21 mm, распона 6 m и стреле

2,8 m. Лом је настао при оптерећењу од 2,7 KN. Тест је спроведен на универзитету у Швајцарској (Buri & Weinamd, 2008). Управо због своје комплексности форме, изменом димензија набора, коришћењем другог шаблона при савијању, променом висине и дебљине набора могуће је значајно утицати на носивост конструкције. Управо та способност омогућава да се малом изменом геометрије утиче на носивост конструкције без значајнијег ремећења архитектонског решења. Тренутно су доступни разни компјутерски алати уз помоћ којих је могуће моделирати комплексне геометријске облике набора, који се касније уносе у програме за статичке прорачуне, где се врши рационализација геометрије. Имајући у виду напомените чињенице, може се закључити да се у будућности треба очекивати пораст заступљености набораних конструкција у пракси.

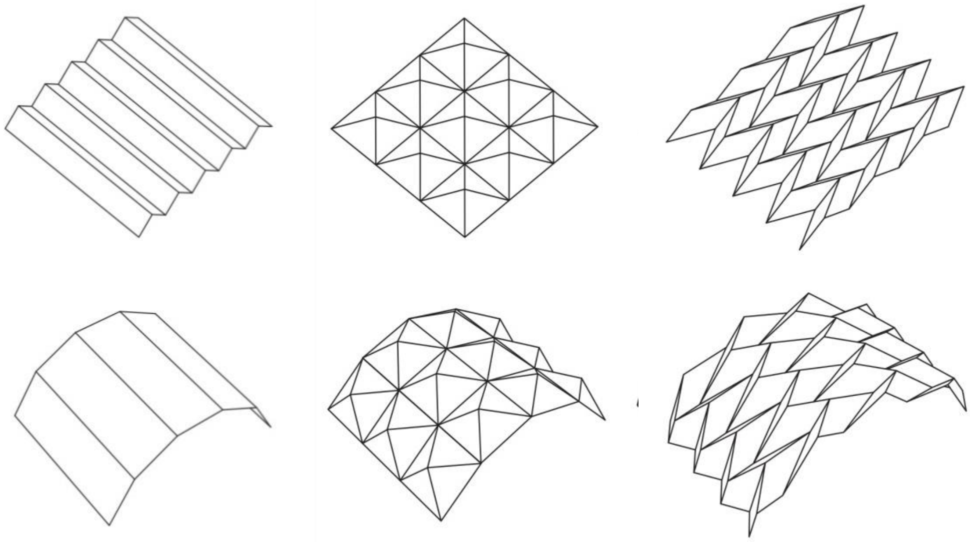
Нажалост, у Србији примена набора у грађевинско-архитектонској техници је још увек недовољно заступљена, па се они могу срести углавном код типских фасада и кровова хала изграђених од легура метала.

Слика 2.2.9. Прототип набора израђен од дрвених шпер плоча

## 2.3 Принципи обликовања и конструисања набора

### 2.3.1 Дефинисање појмова

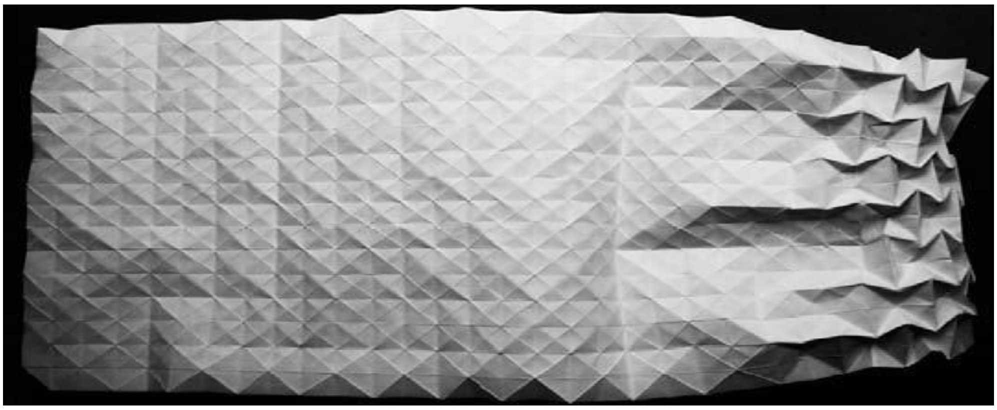
**Набор** можемо дефинисати и као облик који настаје сталним деформацијама -набирањем, а набирање као процес савијања, склапања, гужвања, обликовања и сл. Набирање се може одвијати у једном правцу (слика 2.3.1. лево) или у два - бидирекционо (слика 2.3.1. средина и десно). Термин бидирекциони се односи на правце набирања. Линије набирања које могу имати различите орјентације у три димензије у случају бидирекционих набора орјентисане су најмање у два правца.



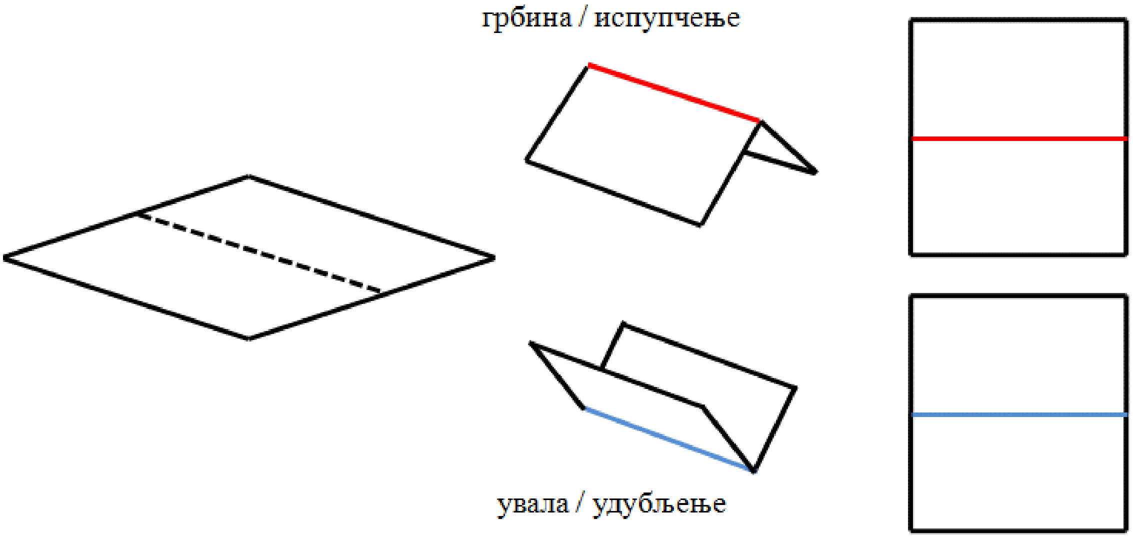
Слика 2.3.1. Набирање у једном правцу и бидирекционо

**Развојна површ** набора се односи на особину набора да се исправи у раван (Гаусова кривина једнака је нули). Развијањем набора добија се површ која се не савија или не увија у односу на раван у којој се налази (слика 2.3.2.). Набрана коначна форма може бити комплексно закривљена, при томе појединачни региони од којих се набор састоји налазе се у једној равни. Постоје и облици набора, као што су елипсоиди или сферичне форме, које је немогуће развити у раван нулте Гаусове кривине, али их је могуће апроксимирати.

**Линија набирања** је дуж (права или у неким случајевима крива) око које се врши набирање. Она се формира на два начина образовањем **увала** или **грбина**, док у специјалном случају она може равна. На слици 2.3.3. црвеном бојом је означена грбина, плавом увала, а испрекиданом линијом равна линија набирања.



Слика 2.3.2. Развојна површ набора

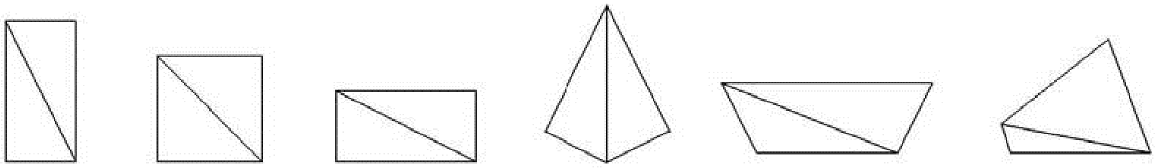



Слика 2.3.3. Линија набирања

Основну јединицу свих развојних бидирекционих набора чини троугао. Основна јединица набора који нису бидирекциони или бидирекционих набора који се не развијају у раван су полигони вишег реда (нпр. пентагони, хексагони или неправилне геометрије).

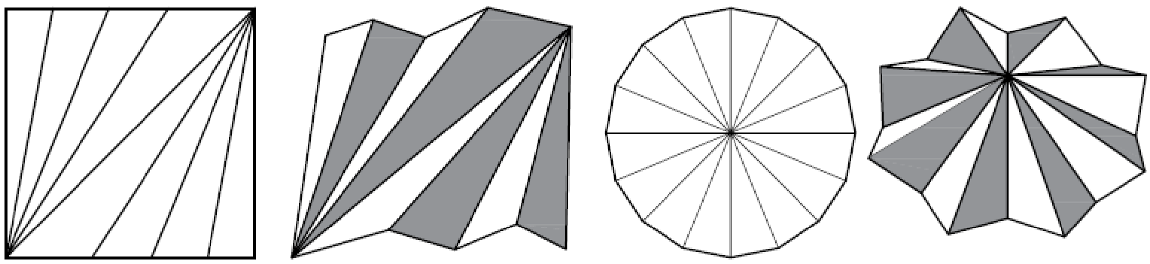
**Аксиом:** све четвороугаоне геометријске фигуре је могуће свести на два троугла, а у случају развојних бидирекционих набора све четвороугле и троуглове је могуће свести на правоугле троуглове.

Две основне јединице међусобно спојене дуж сопствених хипотенуза образују **колено набора**. Оно се налази између две контролне линије на мрежи набора и четвороугаоног је облика. Најчешћи типови колена приказани су на слици 2.3.4.



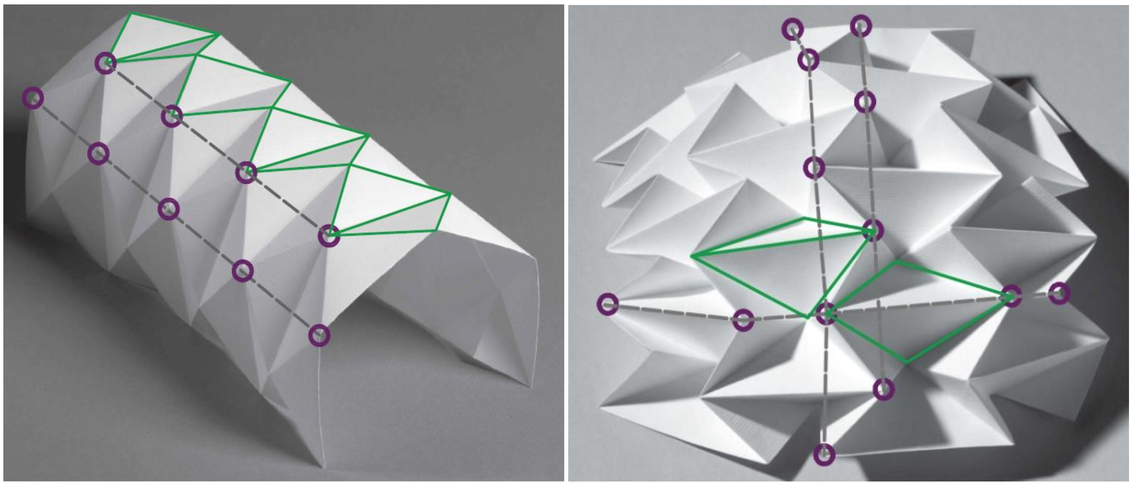
Слика 2.3.4. Колено набора

Растер линија које су повезане у крајњим тачкама и дефинишу колена набора чине **мрежу набора** (слика 2.3.5.) (Jackson, 2011). Такође, мрежа може бити подељена у коначан скуп полигоналних региона сегментима правих линија. Линије могу бити закривљене али се најчешће ради о правим линијама, тј. најкраћим растојањем између тачака. Важна геометријска карактеристика мреже набора је да поседују различите типове симетрије (нпр. могу бити аксијално или ротационо симетрични) јер настају транслацијом, ротацијом и рефлексијом основних јединица.



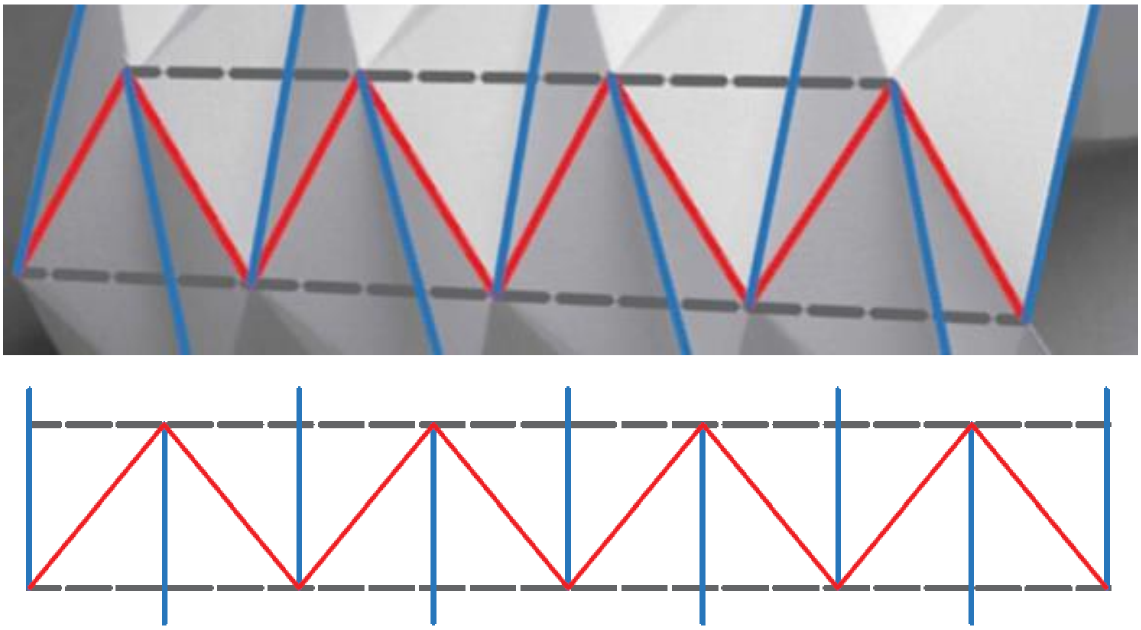
Слика 2.3.5. Мрежа набора

Тачке где се сусрећу четири или више набора називамо чворовима. Линија која повезује чворове образује **контролну линију набора** (слика 2.3.6.). Геометријски облик контролне линије дефинише облик набора у 3D простору. Она може бити права, цик-цак, крива (парабола, елиптична, кружна). Све наборе је могуће представити уз помоћу мрежа контролних линија која је јединствена, односно не постоји мрежа контролних линија којом се дефинишу два различита набора. Контролне линије одређују степен слободе набора који геометријски поседује. Пресек набора обично поседује две контролне линије, унутрашњу и спољашњу, а њиховим растојањем дефинисана је висина (дубина) набора. На слици 2.3.6., сивом испрекиданом линијом приказане су контролне линије набора, а зеленом бојом колена набора.



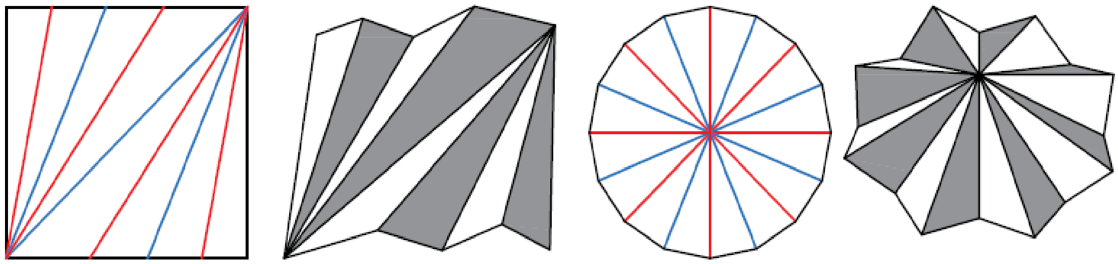
Слика 2.3.6. Контролна линија и колена набора

Континуална трака повезаних колена између две контролне линије на мрежи формира **низ набора** (слика 2.3.7.). Треба напоменути да је код развојних бидирекционих набора постоји само један могући низ.

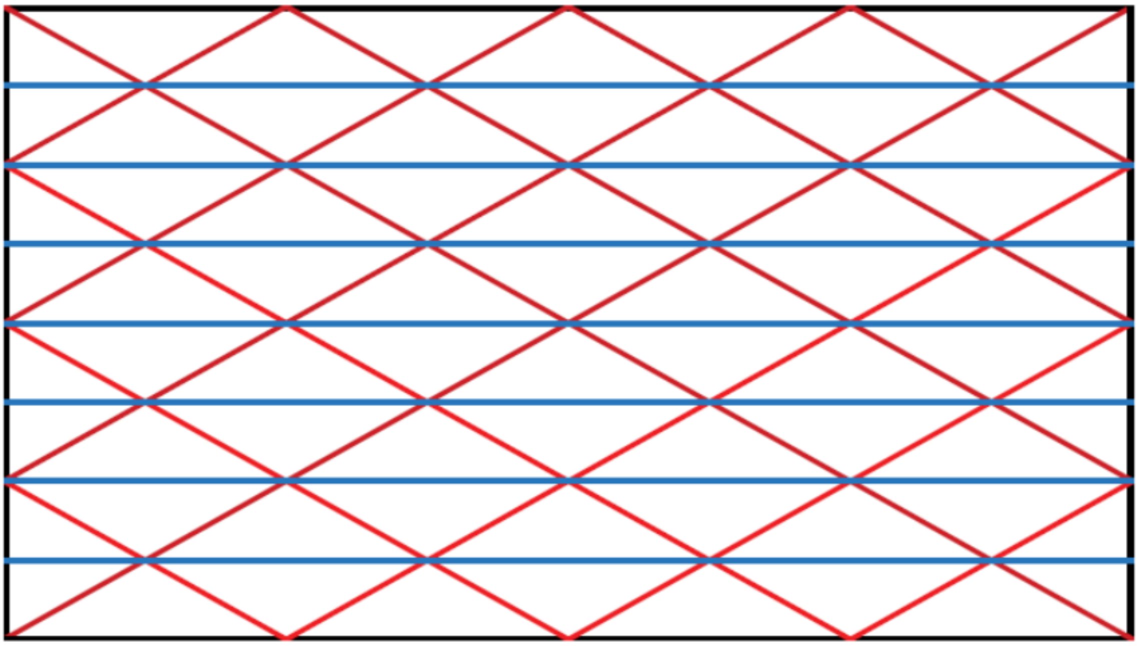


Слика 2.3.7. Низ набора; плава - увале; црвена - грбине; сива -контролна линија

Када су дефинисане сви параметри набора може се приступити изради **дијаграма набирања** (слика 2.3.8. и 2.3.9.), који представља заправо план набирања. На њему је у равни приказано који линију набирања треба савити као испупчење а коју као удубљење. Конвенцијом је усвојено да дијаграм представља горњу површину набора. Дијаграм набирања треба разликовати од мреже набора на којој није јасно дефинисан распоред увала и грбина.

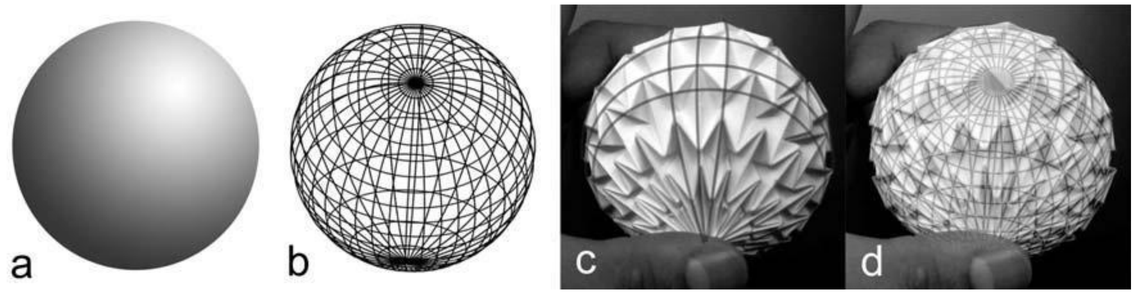


Слика 2.3.8. Дијаграм набирања



Слика 2.3.9. Јошимурин дијаграм набирања

**Пресликавање набора** на изабрану архитектонску геометрију врши се уз помоћ контролних линија и мреже набора. На слици 2.3.10. (а) приказано је пресликавање набора на сферу, где је на слици (б) приказана мрежа набора (жичани модел), (ц) неколико контролних линија, (д) све контролне линије. Уколико је предвиђено да број линија у мрежи сфере и број контролних линија набора буде исти (на слици (б) и (д) су идентични) могуће је пресликавање један на један, односно прилагођавања набора одређеној архитектонској геометрији.




Слика 2.3.10. Пресликавање набора на сферу

### 2.3.1 Аксиоми и правила набирања

Приликом пресликавања набора потребно је задовољити услов изометрије и да не постоји продор (Demaine и O'Rourke, 2007). Под изометријом се подразумева да најкраће растојање између две тачке остаје непромењено, тј. да папир не трпи деформацију. Под условом непостојања продора подразумева се да се делови папира међусобно не секу. Додатно, Хумиаки Хузита (Humiaki Huzita, 1991) је први математички дефинисао шест аксиома набирања површи. При томе важи претпоставка да се све операције набирања врже у равни и да су све линије набора дужи:

**А1.** Уколико су задате две тачке, кроз њих је могуће повући линију набирања.

**А2.** Уколико су задате две тачке, могуће је повући линију набирања дуж њихове симетрале, тако да се након набирања тачке преклапају.

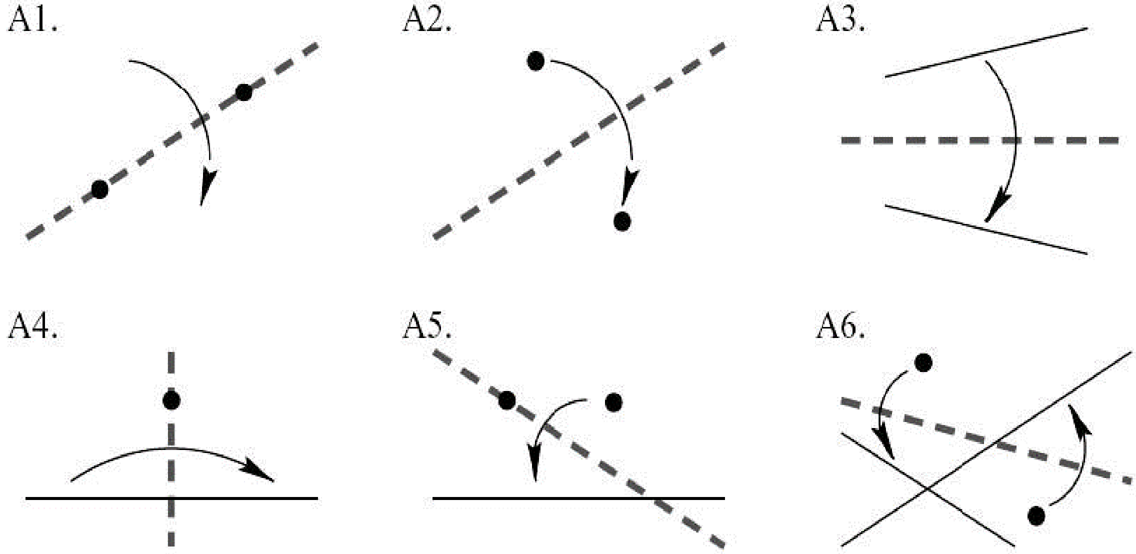
**А3.** Уколико су задате две линије, могуће је повући линију набирања дуж њихове симетрале, тако да се након набирања линије преклапају.

**А4.** Уколико су задате тачка и линија, нормала постављена кроз тачку на задату линију може представљати линију набирања, а након набирања делови задате линије се преклапају.

**A5.** Уколико су задате две тачке и линија, могуће је повући линију набирања кроз једну тачку, тако да се друга задата тачка пресликава на задату линију.

**A6.** Уколико су задате две тачке и две линије, могуће је поставити линију набирања тако да се једновремено једна задата тачка пресликава на једну задату линију, а друга задата тачка на другу задату линију.

Графичка интерпретација Хузитиних аксиома приказана је на слици 2.3.11.

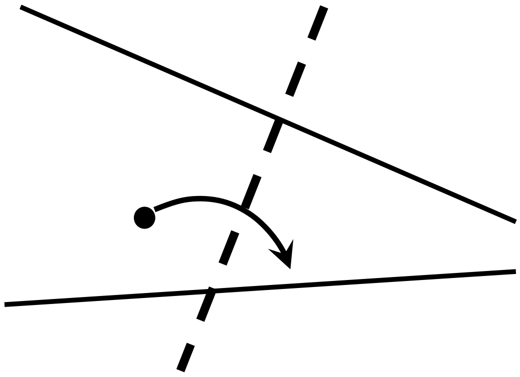



Слика 2.3.11. Хузитини аксиоми

Седми аксиом је предложио Хатори (Koshiro Hatori, 2001), па се често може наћи у литератури да се аксиоми називају Хузита-Хаторијеви аксиоми:

**A7.** Уколико су задате једна тачка и две линије, могуће је поставити линију набирања нормално на једну од задатих линија тако да се тачка преслика на другу задату линију (Hull 1995 (слика 2.3.12.)).

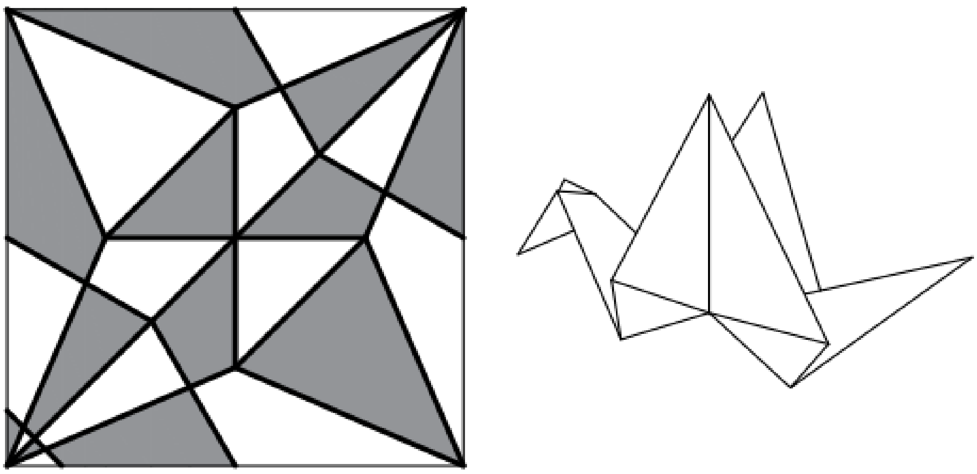
Уз помоћ прва четири аксиома за задате две тачке могуће је конструисати једноставне облике набора који практично могу да се направе уз помоћ лењира и шестара. Следећа два аксиома дефинишу компликованију геометрију, док седми аксиом искључује могућност дефинисања нових аксиома.



Слика 2.3.12. Хаторијев аксиом

Код пресликавања на 3D форму полазна основа нам је равна 2D површ. Уколико је могуће равну површ набирањем трансформисати у просторну форму, онда томе важе четири основна правила набирања које је математички доказао Роберт Ланг.

**П1.** Уколико регионе на дијаграму набирања обојимо на тај начин да ниједна два суседна региона немају исту боју довољне су само две боје како би се направила шема набора који могу бити развијени у 2Д раван (слика 2.3.13.).

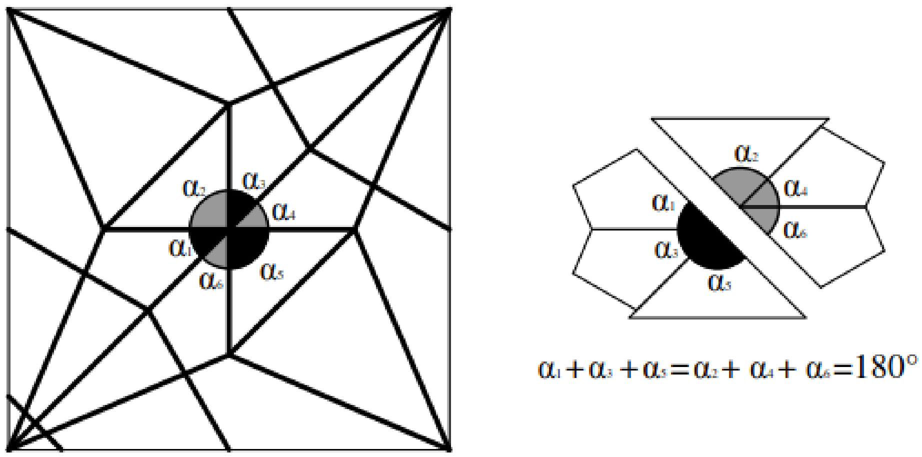


Слика 2.3.13. Правило 1

**П2.** У сваком чвору, број линија набирања мора бити једнак, а сви углови $\alpha_1, ..., \alpha_{2n}$ у чвору морају да задовоље следећи услов:

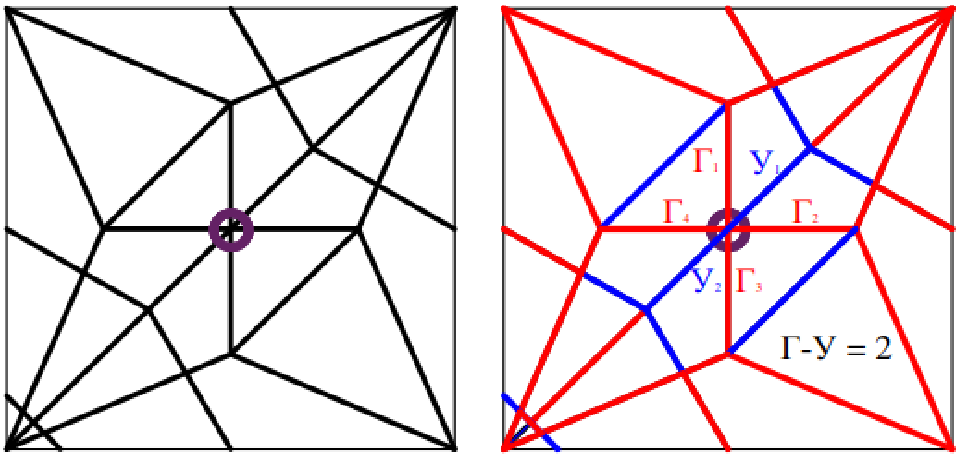
$\alpha_1 + \alpha_3 + \ldots + \alpha_{2n-1} = \alpha_2 + \alpha_4 + \ldots + \alpha_{2n} = 180^o$ или другачије формулисано

$\alpha_1 - \alpha_2 + \alpha_3 - \cdots + \alpha_{2n-1} - \alpha_{2n} = 0$ (Kawasaki-Justin теорема (слика 2.3.14.))




Слика 2.3.14. Правило 2

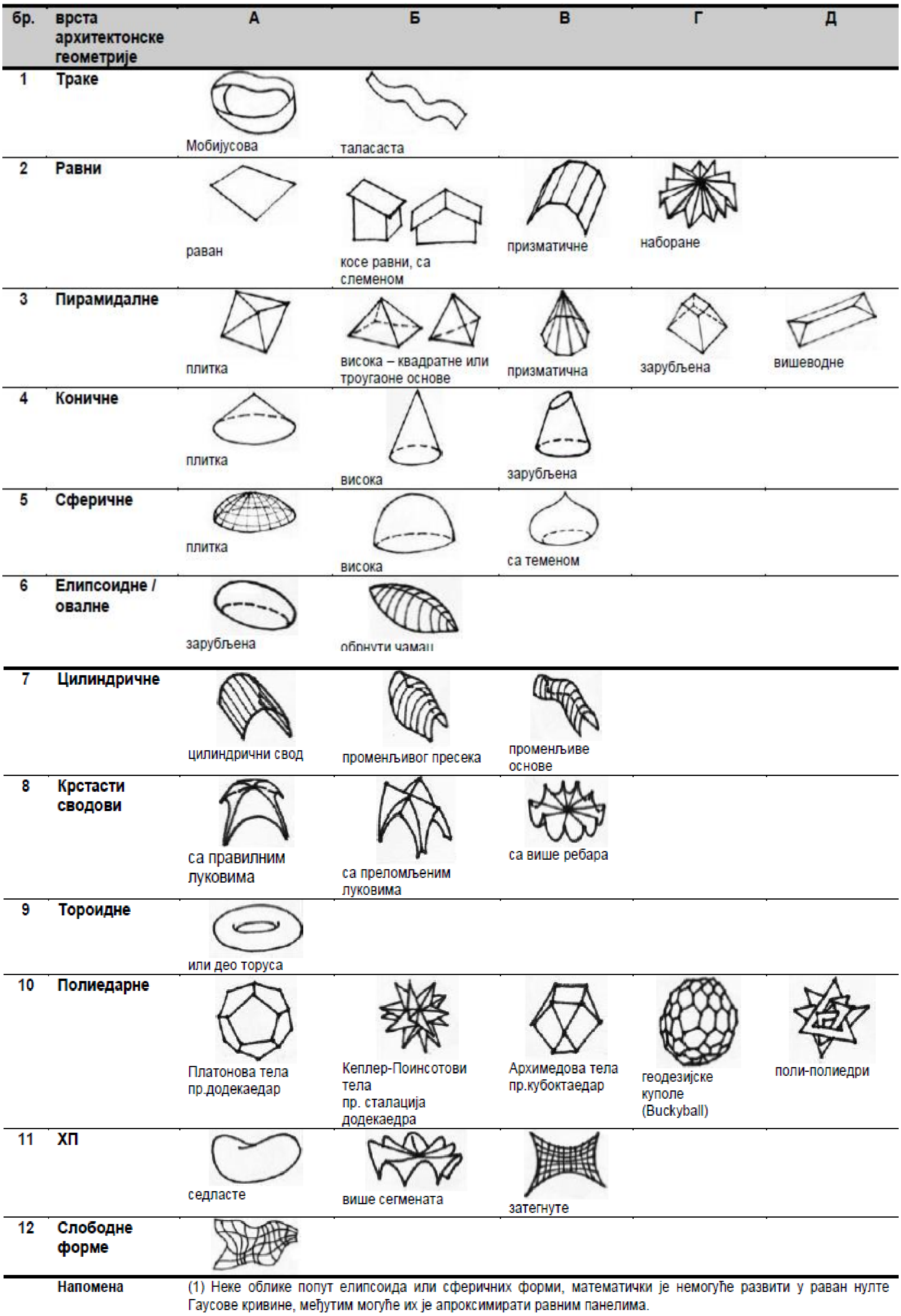
**П3**. У сваком темену број увала (У) и грбина (Г) мора увек да се разликује за два, тј. Г – У = ±2 (Maekawa-Justin теорема (слика 2.3.15.)).




Слика 2.3.15. Правило 3

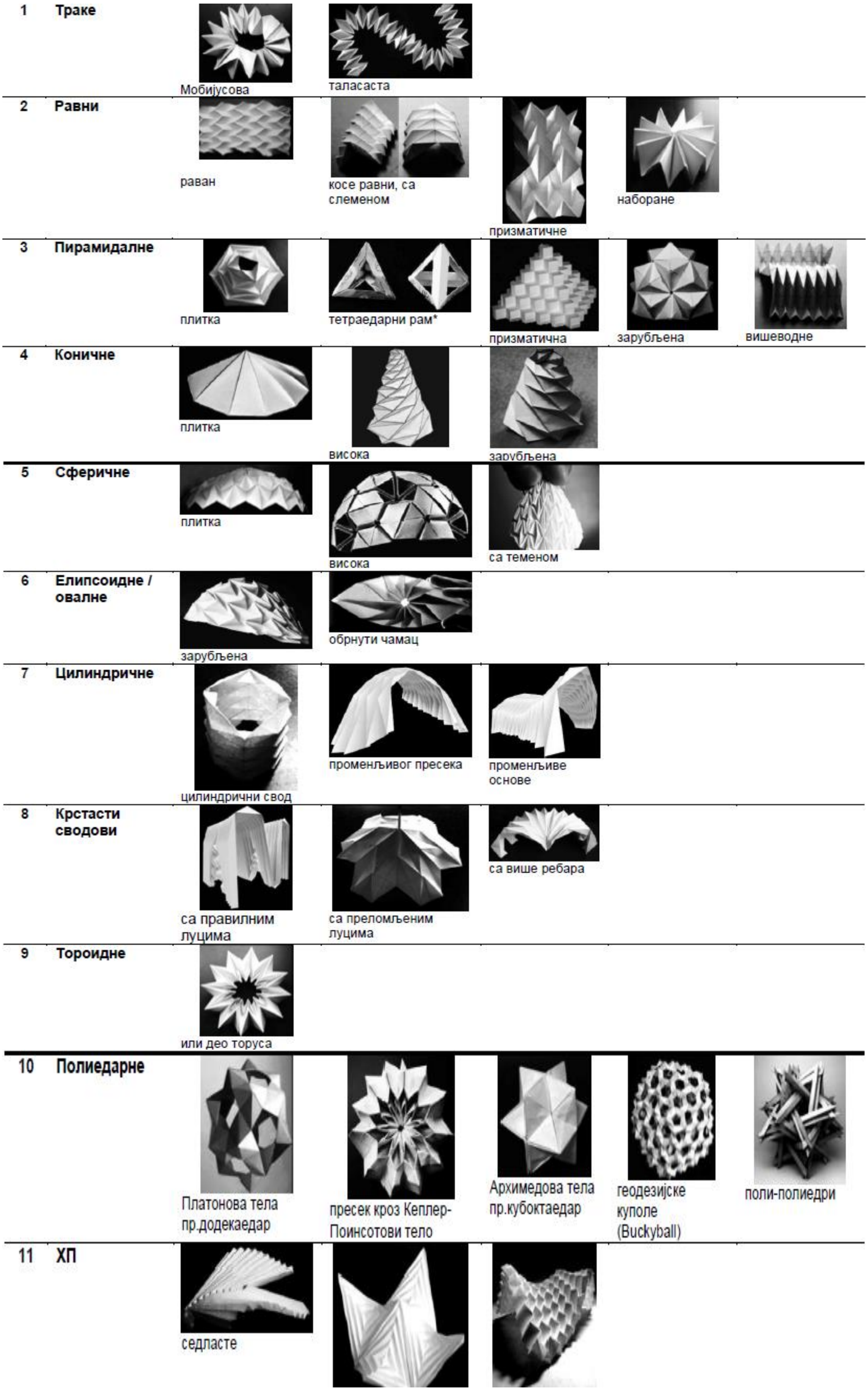
**П4**. Не сме бити никаквих прекида, резова, како би се задовољио услов да нема продора. (Schneider, 2004 (слика 2.3.15.)).

Као што смо показали дијаграм набирања пресликан на одређену архитектонску геометрију даје коначну форму набора. Уобичајене архитектонске геометрије су приказане на слици 2.3.16., а коначна пресликана форма набора на слици 2.3.17. (Mitra, 2009)



| бр. | врста архитектонске геометрије | А | Б | В | Г | Д |
|---|---|---|---|---|---|---|
| 1 | Траке | Мобијусова | таласаста | | | |
| 2 | Равни | раван | косе равни, са слеменом | призматичне | набо特ране | |
| 3 | Пирамидалне | плитка | висока – квадратне или троугаоне основе | призматична | зарубљена | вишеводне |
| 4 | Коничне | плитка | висока | зарубљена | | |
| 5 | Сферичне | плитка | висока | са теменом | | |
| 6 | Елипсоидне / овалне | зарубљена | обрнути чамац | | | |
| 7 | Цилиндричне | цилиндрични свод | променљивог пресека | променљиве основе | | |
| 8 | Крстасти сводови | са правилним луковима | са преломљеним луковима | са више ребара | | |
| 9 | Тороидне | или део торуса | | | | |
| 10 | Полиедарне | Платонова тела пр.додекаедар | Кеплер-Поинсотови тела пр. сталација додекаедра | Архимедова тела пр.кубоктаедар | геодезијске куполе (Buckyball) | поли-полиедри |
| 11 | ХП | седласте | више сегмената | затегнуте | | |
| 12 | Слободне форме | | | | | |
| | Напомена | (1) Неке облике попут елипсоида или сферичних форми, математички је немогуће развити у раван нулте Гаусове кривине, међутим могуће их је апроксимирати равним панелима. | | | | |

Слика 2.3.16. Архитектонске форме




Слика 2.3.17. Пресликана форма набора на архитектонску форму

## 2.4 Оригами као средство за пројектовање набора

### 2.4.1 Увод

Оригами техника савијања папира је настала у Кини, а преузета и детаљно развијена и изучена у Јапану. На Архитектонском факултету, Универзитета у Београду у оквиру предмета Структурни системи спроведено је истраживање потенцијала употребе оригами технике савијања папира као алата при опису, презентовању и структури форме физичког објекта, односно апстрактног алата при пројектовању структурних форми набора. Целокупно истраживање је спроведено под менторством проф. Миодрага Несторовића, објављено је у часопису "Spatium[1]" и уједно представља саставни део овог докторског рада.

Техника савијања папира је коришћена као приступ за стицање когнитивног искуства у циљу лакшег разумевања просторне трансформације и дигиталне интерпретације пројектоване форме (Buri & Weinamd, 2008) (Lister, 2003, 2004). Циљ оваквог приступа проблематици је да омогући студентима анализу геометријских принципа набирања како би се касније применили при развоју и пројектовању нових структура. Генеративни алгоритам процеса је инспирисан техником савијања папира и од користи је при проналаску жељене форме. Добијени облици су физички потврђени производњом скалираних модела - прототипа.

Убрзан развој технологија и повећане у сложености процеса константо утичу на прилагођавање процеса пројектовања. Изазови са којима се суочавамо укључују тежњу ка аутоматизованом процесу продукције, односно вођени су развојем CAD/CAM система. Поменуте технологије омогућавају спровођење тестова на рачунским моделима структура у дигиталној форми, односно олакшавају аплицирање тестова неконвенционалних форми.

Приступ проблематици пројектовању набораних конструкција смо засновали на примени оригами технике како бисмо добили варијантна решења пројектованих геометријских форми. Оригами као вештина је временом еволуирала у интердисциплинарну методу Оригамикс (Origamics) [1] (Stewart, 2007). Са веома широким спектром примене, оригами се наметнуо као напредни алат за

[1] Nestorović, M., Milošević, J., Nestorović, P., Maneski M. (2016) Instrumentalization of origami in construction of folded plate structures - design, researchandeducation; Spatium, No.35, pp. 22-29

развој различитих инжењерских и пројектантских решења. Током последњих деценија истраживања спроведена у домену нанотехнологија, биологије, аутомобилског и структурног дизајна отворила су могућности за истраживањем геометријских корелација, односно креирањем нових форми и структура. Задивљујућа техничка и уметничка унапређења у великој мери су остварена захваљујући већој математичкој и рачунарској свести и разумевању процеса анализе објекта. Генерално у свету постоји тренд растућег броја нових примера, истраживања и презентације резултата, па се може закључити да ова древна техника савијања папира има још много тога да нам понуди.

Истраживање зависности форма - функција - структура у архитектонском смислу је развојем савремених технологија постало веома сложено. Излагање и промовисање нових метода, алата и средства, који би поједноставили комплексне захтеве истраживачког аспекта, постало је веома важно средство за рад. У том смислу ово истраживање је имало за циљ да критички преиспита одрживост оригами технике као средства на основу кога се пројектују наборани конструктивни системи. Додатно, рад скреће пажњу да оригами може бити помоћни медијум пре ефикаснијем проналажењу одговарајуће геометријске форме.

### 2.4.2 Оригами као методологија при архитектонском пројектовању

Савремена примена оригамија није ограничена само на занатство. У архитектури је нашла примену као метода за решавање проблема комплексних геометрија, односно средство за учење и разумевање пројектовања различитих геометријских облика (Hagiwara, 2008). У почетку оригами техника развијана искључиво на инстинкту индивидуе, данас, разумевање и формулација математичких и геометријских шаблона је главни предмет студија (Nestorović, Nestorović, & Milošević, 2012). У математичком контексту оригами форме могу се разматрати преко пресликавања (мапирања) тесалација у 2D и 3D простору (Sorguç, Hagiwara, & Selçuk, 2009).

Развој већине оригами шаблона (код набора дијаграм набора) је заснован на одређеним правилностима. Присутна је зависност облика (линија и углова) при пројектовању геометријских облика. Углавном, геометрија је базирана на Хузитиним (Huzita) аксиомима (Khademzadehand Mazaheri, 2007), Миакавиним фундаменталним теоремама (Maekawa, 2008), Муириним дијаграмима (Miura, 1994, 1997), Кавасакијевим (Kawasaki) теоремама (Hull 2002) и теоремама и аксиомима предложених од осталих математичара. Р. Ланг је математичар (R. Lang) који је први наведене аксиоме и алгоритме имплементирао у математичке алгоритме, што је допринело развоју комплекснијих геометрија и комерцијализацији компјутерских програма (Lang, 1994, 2004)

Овакав приступ управо пружа могућност истраживања естетски привлачних и конструктивно ефикасних решења, па је нашао и своје место у архитектури при тражењу нових зависности форма - структура и истраживању нових алата при пројектовању. Бројна истраживања у домену примене оригамија при структурном пројектовању само су потврдила потенцијал ове методе.

### 2.4.3 Материјал и технике конструисања

Мала дебљина плоче и компликована геометрија набораних структура представља највећи изазов при конструисању, где су традиционалне методе често неадекватне и недовољно развијене. У односу на ортогоналне структуре, анализа комплексне геометрије је увек био тежак задатак и често без примене у пракси. Захтеви економичности често ограничавају конструисање на примену идентичних елемената (јефтинија производња) умножених у простору. Међутим, развој дигиталног пројектовања и процеса фабрикације пружа нове могућности у конструисању набора. Широк спектар приступа укључује употребу префабрикованих елемената оплате, 3D дигиталне моделе, употребу CNC уређаја велике брзине за обликовање елемента и оплате, као и примену оригами метода за добијање сложених структура уз помоћ једноставних раванских елемената различитих димензија.

Потенцијал оригамија је управо у релативно једноставном креирању прототипа и математичком алгоритму геометрије - структуре. Један такав пример чини набор од бетона. Вин (Wheen, 1980) је патентирао идеју о извођењу прототипа набора од бетону. Реализацију ове идеје је први тестирао Бечхолд (Bechthold, 2008). На слици 2.4.1. је приказан Бечхолдов прототип распона 6,6 m, стреле 2 m и дебљине плоче од 22 mm. Прототип модела је показао да је могуће извести компликоване системе набора од цементно-композитних материјала уз употребу префабрикованих елемената и једноставне оплате.

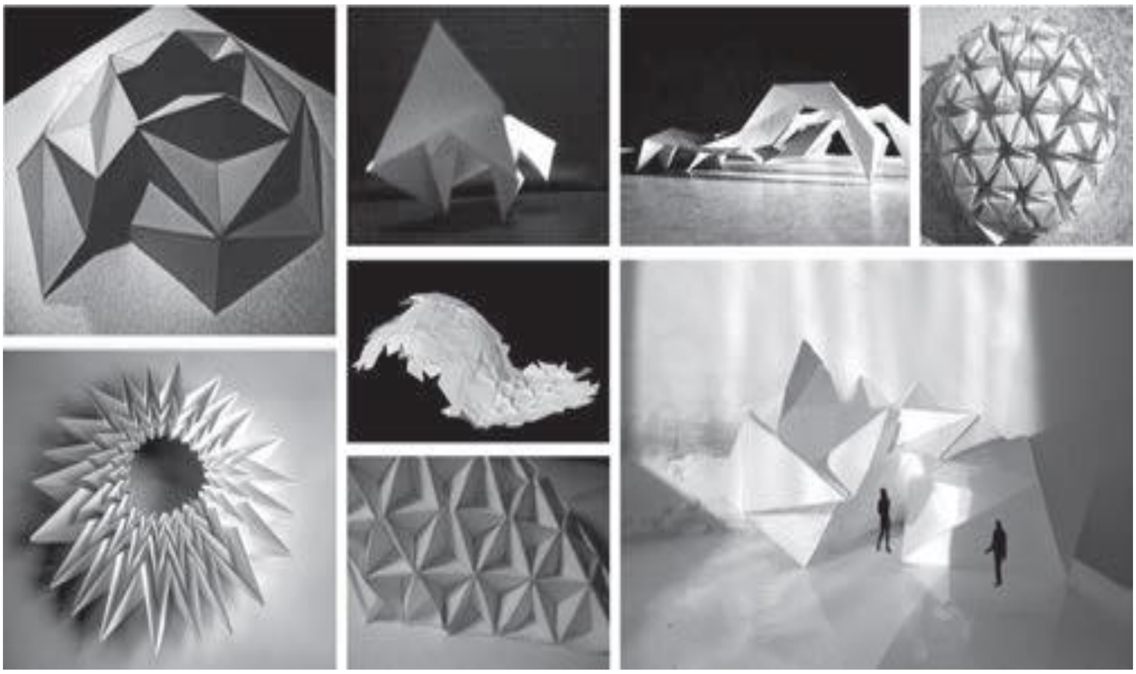
Слика 2.4.1. Бечхолдов прототип

Студија спроведена од стране Бенгхаја (Banghay, 2000) је показала да се оригами може искористити при декомпозицији виртуално направљених објеката у сегменте спремне за CAM производњу, што представља још једно могуће поље примене.

Битан удео у даљем развоју набора свакако је директно повезан са напретком у развоју материјала високих перформанси који би омогућили широк спектар конструктерског обликовања. Стакло, високо вредна влакна бетона и полимер композити су неких од материјала који поседују велики потенцијал примене. Оригами техника савијања је послужила као инспирација за развој нано материјала (Stellman & Greene, 2005) који би могли у будућности наћи примену и у грађевинској индустрији.

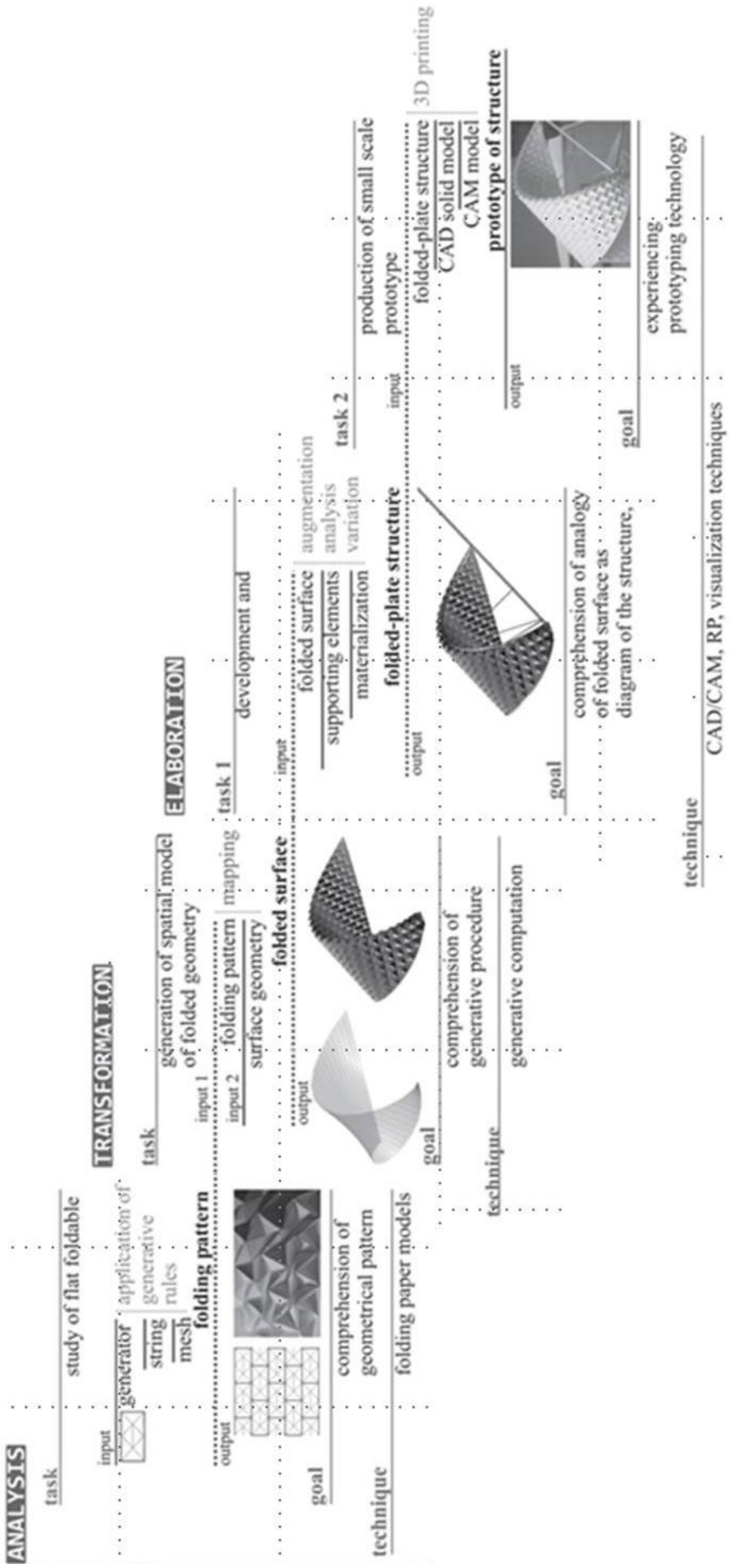
### 2.3.4 Процес конструисања у оквиру предмета Конструктивни системи

Већ смо напоменули да се директан увид и спознаја геометрије набораних структура најједноставније постиже на моделима од савијеног папира. Обзиром на једноставност и приступачност, ова метода се често користи у едукативне сврхе. Фрубел (Fröbel) је савијање папира користио у жељи да деца предшколског узраста стекну општа сазнања о геометрији и естетици (Lister, 2003, 2004). Како би приближио студентима релације између материјала, геометрије и структуре Алберс (Albers, 1952) је применио сличан приступ учењу. Тренутно, постоји велики број предмета у различитим архитектонским образовним установама где се кроз радионице омогућава студентима истраживање односа форма – структура - простор како у стварном, тако и у виртуелном контексту. Инспирисани сличним мотивацијама на Архитектонском факултету, Универзитета у Београду у оквиру предмета Структурни системи изучава се поменути приступ и методологија. Циљ проф. Несторовића (Nestorović, Nestorović, & Milošević, 2012) је био да се тестира метода која би омогућила студентима да анализирају геометријске принципе пројектовања набораних конструкција и да их примене при развоју нових конструктивних решења форми набора (слика 2.4.2.). Комплетан пројекат је био подељен у три сукцесивне фазе: анализа, трансформација и елаборација, где је свака фаза изграђена на основу резултата претходне (слика 2.4.3.). Како би се створила општа слика, фазе примењене процедуре биће илустроване на једном конкретном примеру решења наборане конструкције



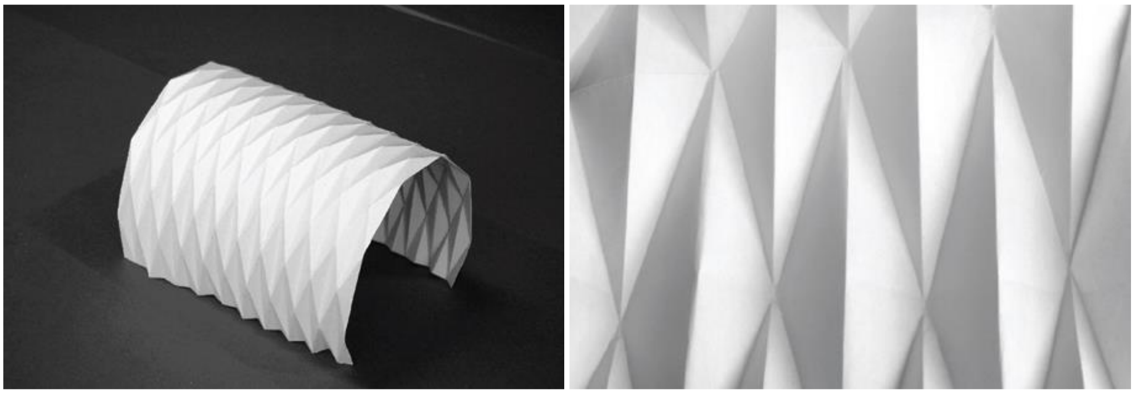
Слика 2.4.2. Конструктивна решења форми набора

Током прве фазе анализирају се геометријски и структурни принципи. Студенти подељени у групе имали су задатак да изуче различите савијене тесалације набора тако што ће направити физичке моделе. Производња различитих серија папирних модела савијањем омогућила је студентима увид у предности технике у погледу просторног сагледавања геометрије и примену при архитектонском пројектовању. Задатак ове фазе је био да се одреде занимљиви дијаграми набирања који поседују потенцијал интеграције у процес пројектовања.



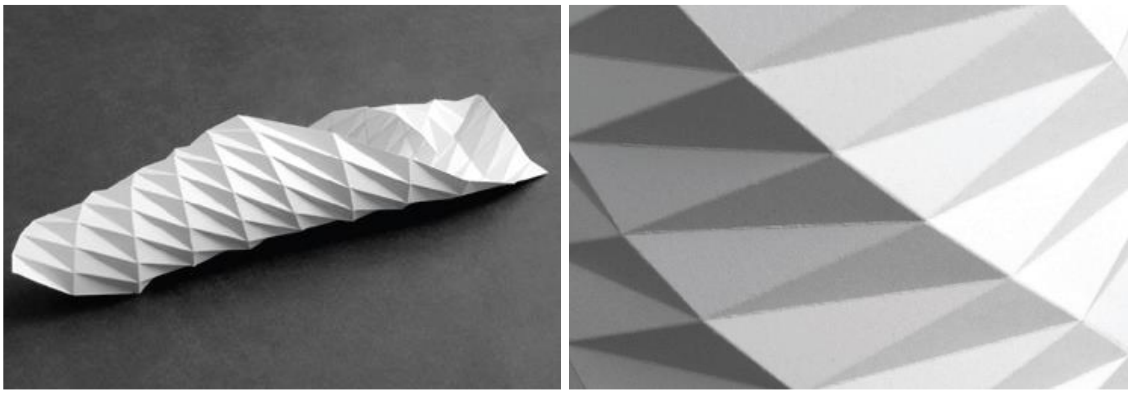

Слика 2.4.3. Фазе истраживања

У складу са тим, примењени су различити дијаграми набирања као што су: Јошимурин (Yoshimurapattern) Дијамантски дијаграм (слика 2.4.4.), Мијура Ори дијаграм (Miura Ori pattern) (Miura) познат као и рибља кост (2.4.5.), дијагонални дијаграм (слика 2.4.6.) и слично.



Слика 2.4.4. Јошимурин шаблон

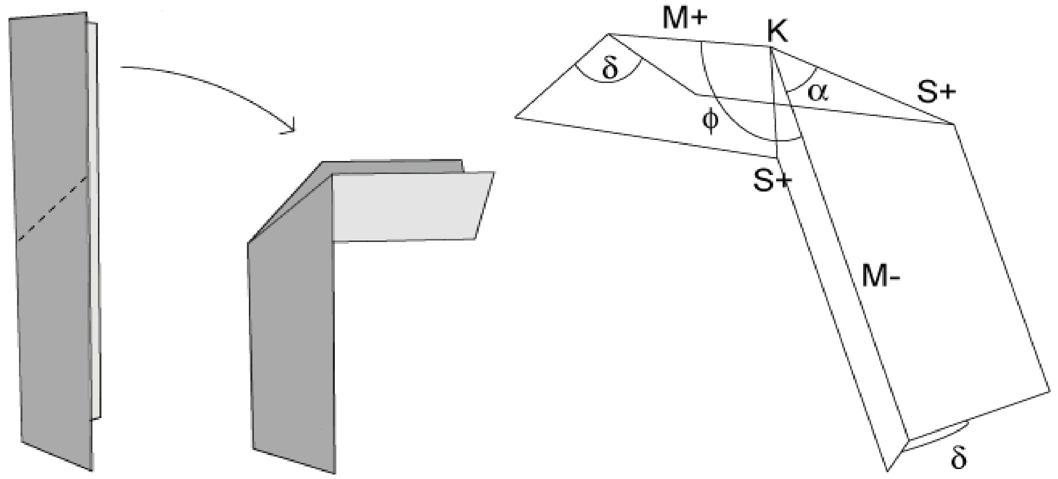
Слика 2.4.5. Мијаурин шаблон



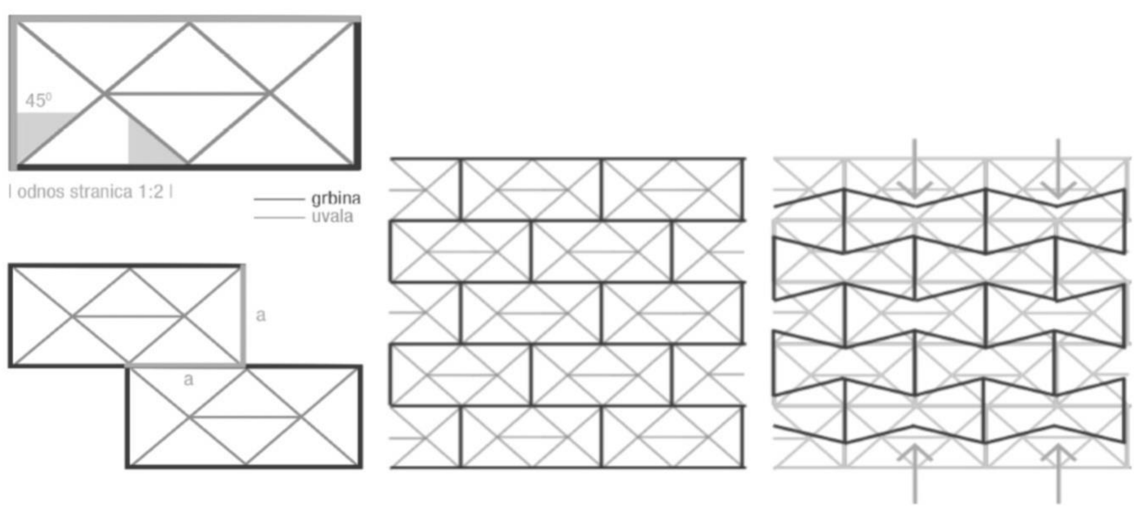
Слика 2.4.6. Дијагонални шаблон

Предметни дијаграми набирања су углавном формирани на основу два основна принципа набирања: основно набирање и обрнуто набирање (слика 2.4.7.), при чему превоји у низу наизменично састављени од увала и грбина, а формирани уз помоћ технике обрнутог савијања, креирају кориговане површине.

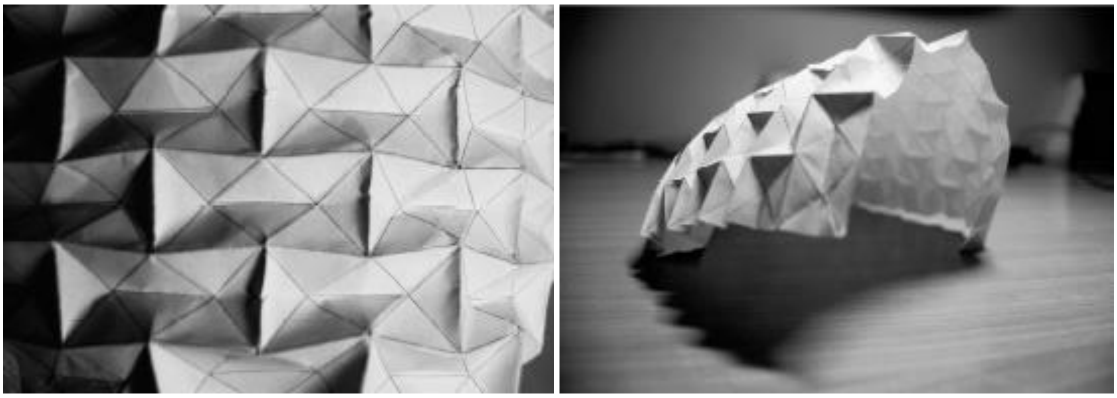
Разумевање математичких и геометријских релација је био најважнији задатак предметних студија. Почетак је подразумевао избор основне јединице шаблона, односно одговарајући генератор. Применом разних генеративних правила резултовало је добијањем раванске мреже и стварањем одговарајућег низа елемената. Раванске мреже се потом преводе у дијаграме набирања, односно неку врсте мапе која приказује увале и грбине набора (слика 2.4.8.). Затим, набори настају избором почетних параметра и генеративних правила (слика 2.4.9.).




Слика 2.4.7. Обрнуто набирање



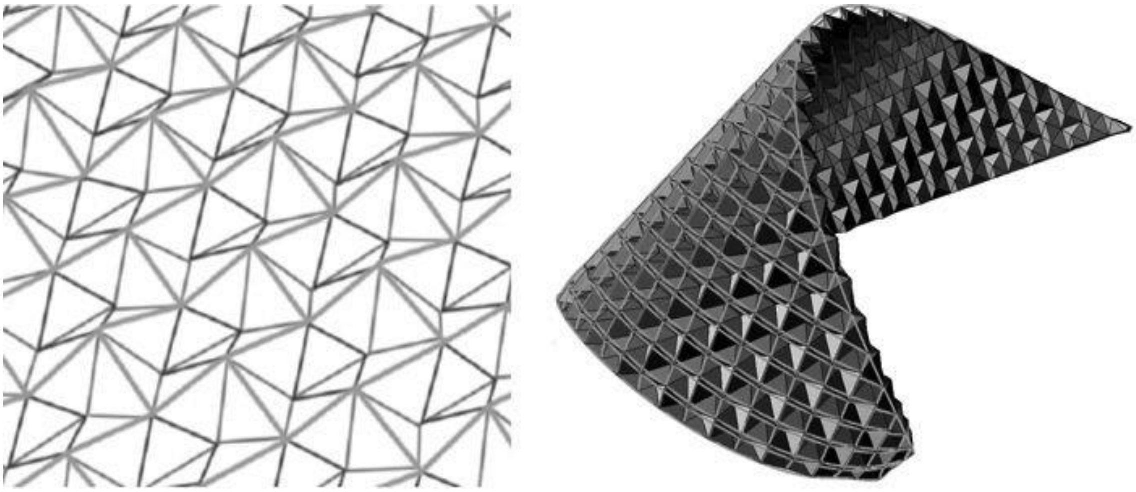

Слика 2.4.8. Дијаграм набирања



Слика 2.4.9. Креирање набора

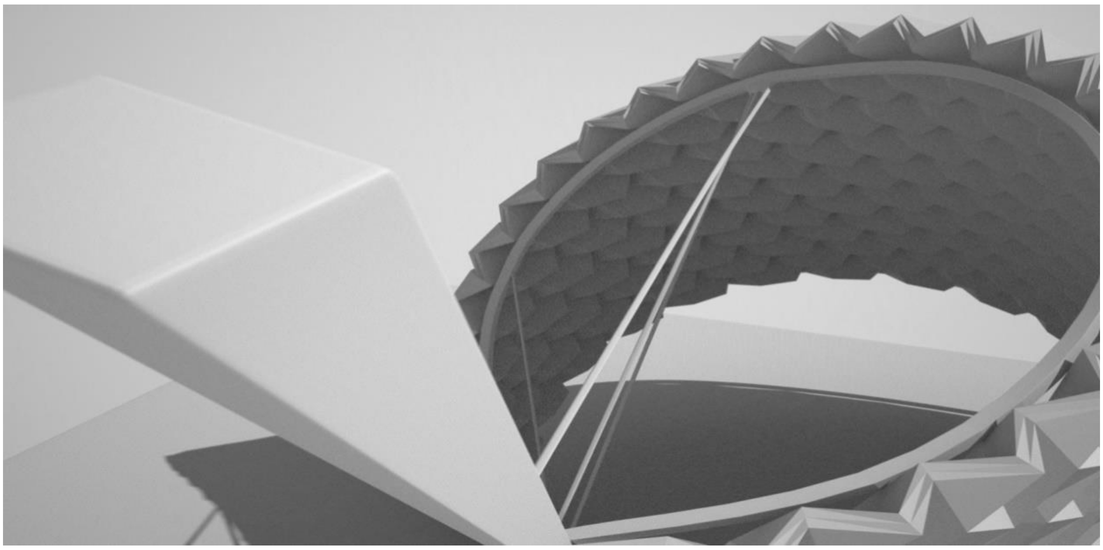
У фази трансформације изабрани дијаграми набора пресликавају се на дефинисану архитектонску геометрију како би се формирали модели просторних структура. У циљу стварања различитих архитектонских облика набирањем као подлоге коришћене су различите класе геометријских типова и њихових композиција (Pottman, Asperl, Hofer, & Kilian, 2007). Задатак овог дела је усмерен ка аналитичком разумевању изабране геометрије како би се генерисао модел употребом CAD компјутерског програма (Khoshnevis). Другим речима, акценат је стављен на истраживање потенцијала оригами технике као алгоритма за проналажење одговарајуће форме.

Пресликавање дијаграма набора на дефинисан геометријски облик је изведено уз помоћ такозваног концепта контролиних линија (Mitra, 2009). Контролне линије повезују нодове (чворове) у којима се сусрећу четири или више линије набора. Оне дефинишу уједно и степене слободе (DOF) коју поседује површина набора. Геометрија површине се може представити уз помоћ линијске мреже (жичани модел). Анализа је показала да неки облици набора имају више степени слободе од других, да могу одговарати за различите геометријске облике, односно да су за неке облике идеални док за друге и нису најбоље решење (слика 2.4.10.). Примењени генеративни метод се показао као веома ефикасан, јер је омогућио брзо генерисање различитих форми и показао висок степен адаптибилности према специфичним захтевима пројекта.



Слика 2.4.10. Форма набора

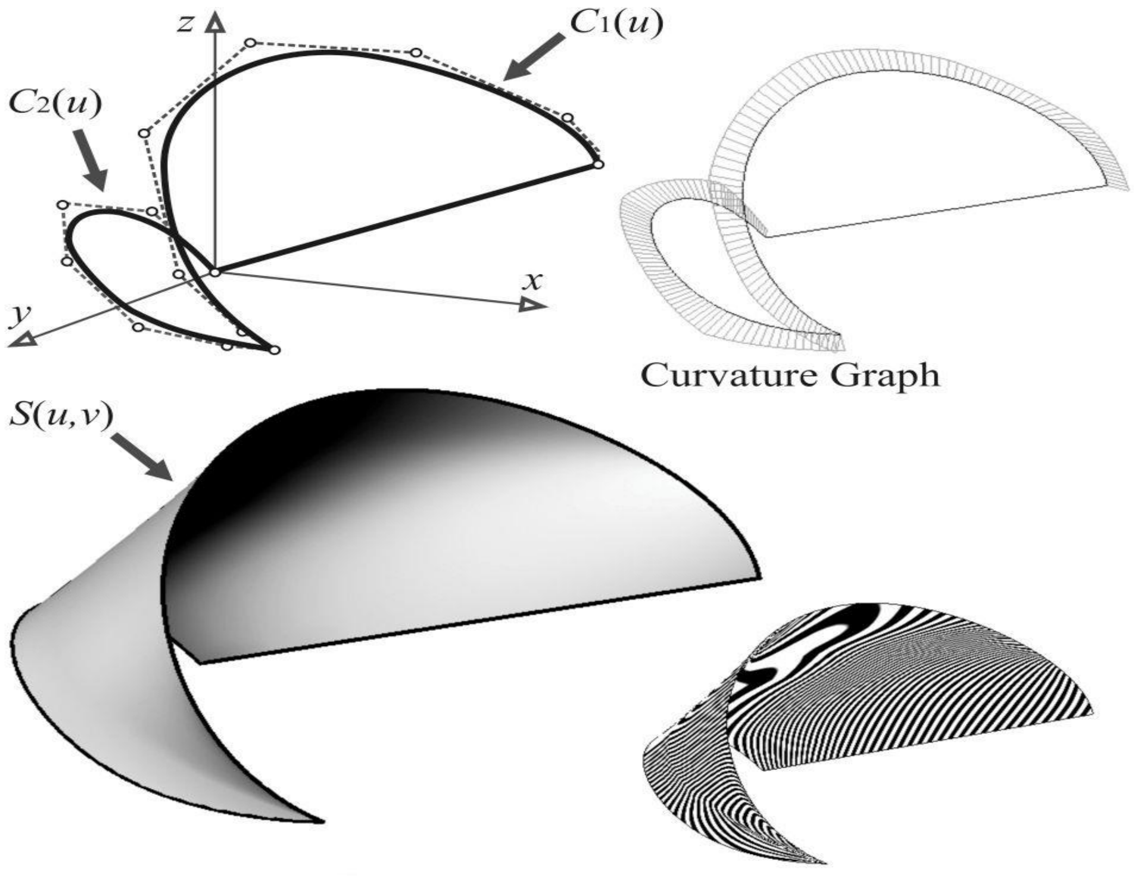
Под фазом елаборације се подразумева физичко представљање пројектованог решења набориране конструкције. Фокус је усмерен ка релацијама форма - структура и конструкција - материјализација. Главни задатак је био трансформација геометрије површина набора у стваран крут модел – прототип. Како би се успешно остварила материјализација генерисане геометрије потребно је детаљније разумети саму конструкцију и везе геометрије, облика, материјала, структурног понашања и сл. Управо је то један од главних разлога за увођење и имплементацију анализе и евалуације структурног понашања у ову фазу пројекта. Коначно, приступа се креирању прототипа у малој размени употребом 3D штампача (слика 2.4.11.)



Слика 2.4.11. 3D прототип набора

### 2.4.5 Генерисање шаблона набора и креирање прототипа

Након избора одговарајућег шаблона набора, приступа се дефинисању референтне геометријске површине на коју ће се пресликати шаблон и формирати слободну форму набора. Геометријска површина се математички дефинише преко NURBS површина (Piegl & Tiller, 1997). Изабрана површина у виду слободне форме се у општем случају дефинише уз помоћ две изводнице (директрисе), такозване NURBS криве C1(u) и C2(u). Референта површина S (u, v) у суштини представља линеарну интерполацију између кривих C1(u) и C2(u) и може се развити у раван (слика 2.4.12.).



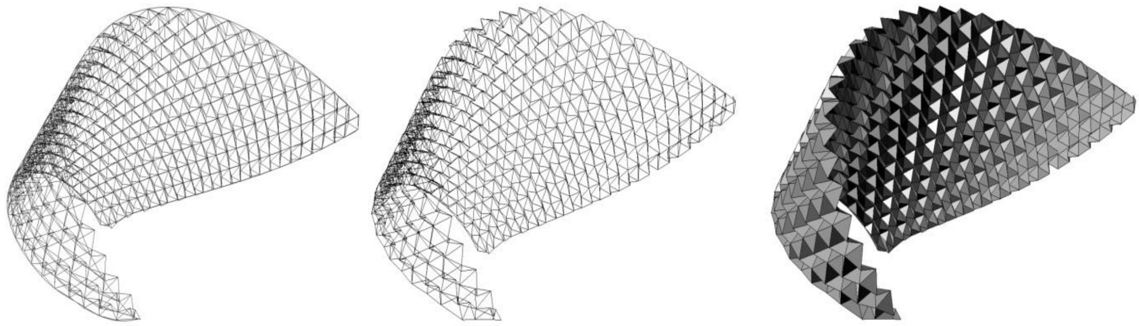

Слика 2.4.12. Линеарна интерполација кривих

Генерисана NURBS површ представља референту геометрију за алгоритам пресликавања шаблона набора, при чему предметни алгоритам укључује следеће кораке:

- Врши се потподела референтне површи у виду мреже интерполационих кривих. Густина мреже зависи од густине пројектоване форме набора.

Креирана мрежа интерполационих крива треба да одговара контролним линијама површи набора.

- Умножавање контролних линија набора и суседних чворова.
- Генерисање низа тачака пресликавањем на референтну површ у пресеку контролних линија.
- Према изабраном дијаграму набора врши се линијско повезивање тачака - ивице троугаоних плоча.
- Генерисање троугаоних плоча површи набора (слика 2.4.13.).



Слика 2.4.13. Генерисање површи набора

При конструисању набора вођено је рачуна о препорученим параметрима за пројектовање набора, као што су угао између суседних плоча и димензије плоча. Ови параметри у многоме утичу на структурно понашање система. Углови између 70 и 110 степени представљају компромисно решење практичности и структурне ефикасности система. Повећање углова доводи до губитка крутости система. За однос висине и распона просто ослоњеног система препоручује се опсег од L/8 до L/15 за бетонске елементе, односно L/7 до L/12 за наборе од дрвених шперплоча. За системи од преднапрегнутог бетона може се допустити већа виткост конструкције. Захваљујући параметарском везивању чворова површи набора, једноставно се може извести модификација параметара као што је стрела набора у циљу испитивања структурног понашања и просторних карактеристика (квалитета) система.

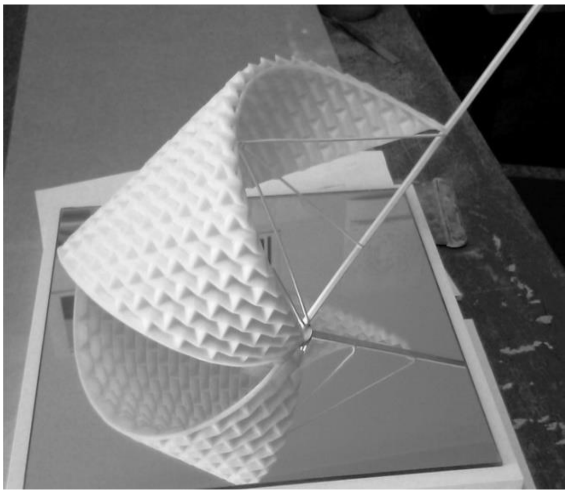
Осим стандардног начина приказивања модела од папира, коришћена је технологија брзе израде модела уз помоћ 3D штампача за креирање физичког модела.. Брза израда модела (Rapid Prototyping - RP) је термин који се односи на брзо продукцију 3D физичког модела на основу претходно дефинисане геометрију у неком од CAD система (Dimitrov, Schreve, & de Beer, 2006) У архитектури РП је нашла примену при визуализацији и тестирању иницијалних идеја у различитим фазама пројектовања конструкције, односно као пожељан корак који претходи било изради модела великих размера или стварних конструкција. У будућности се може очекивати шира примена дигиталне продукцију у пољу грађевинске индустрије.

Задатак израде физичког модела требало је да задовољи следећа очекивања:

- Ефикасна оцена пројектованог концепта
- Да демонстрира изводљивост конструкције на основу предложене геометрије
- Стицање првог искуства са предметном технологијом.

За израду модела коришћен је уређај високе прецизности Spectrum Z$^{TM}$501(слика 2.4.14.) (Z-Corporation, n.d.). Уређај подржава програмске моделе формата STL, VRML и PLY. При изради модела потребно је узети у обзир техничке карактеристике уређаја као што су формат штампе, коришћени материјал, дебљина слоја, резолуција и сл.

Слика 2.4.14. Штампач



Слика 2.4.15. Прототип

Процес израде је укључивао следеће кораке:

- Припрема 3D дигиталног модела набора генерисаног у CAM програму у виду конверзије у STL формат документа. Дебљина плоче набора је дефинисана као стандардна дебљина од 2 mm.

- Модел је изрезан у 2D паралелне слојеве стандардне дебљине од 0.1 mm
- Трансфер модела на уређај за штампу како би се израчунао начин израде
- Пренос информација о сваком слоју до главе уређаја и производња модела додавањем материјала у принципу слој по слој. Потом ласер селективно спаја материјал у праху тако што скенира попречне пресеке генерисане из 3D дигиталног модела
- Након скенирања сваког попречног пресека, лежиште за прах се спушта за дебљину једног слоја и наноси се нови слој на врх. Процес се понавља до завршетка израде модела

Носећи лукови, пилон и затеге су накнадно израђени од челика и уклопљени како би формирали крајњи физички модел (слика 2.4.15.).

**2.3.6 Завршна разматрања**

Оригами као полазна тачка за истраживање и развој архитектонских форми помогла је при дефинисању визуалних алгоритама за израду шаблона набора и резултовала је у стицањем новог спознајног искуства. Као што је илустровано пројектом, оригами дијаграми поседују потенцијал да обезбеде генеративну мрежу као основу за моделирање структурне форме. Такође, пружају добру основу за одређивање структурне стабилности, поготово што постоји могућност анализе како физичког тако и дигиталног модела. Треба имати у виду да нису све форме погодне за примену, па је потребно пажљиво анализирати различите параметре како би се задовољили потребне захтеве стабилности у погледу структура-форма-простор. Ипак, недвосмислено се може рећи да је имплементација оригами методе у процес архитектонског пројектовања моћан алат који архитектама омогућава проширења спектра структурних форми. Представљени примери су показали да оригами може бити разматран као алат за истраживање форме. Као што је претпостављено, на крају студије студенти су постали у потпуности свесни геометрије набора. Наша будућа очекивања на основу свеобухватног искуства су везана за већу примену мултидисциплинарног, интегрисаном приступу пројектовања у циљу добијања иновативних пројектних решења.

Овде бих се захвалио и студентима који су учествовали у семестралном пројекту *Набoране конструкције- Принципи обликовања и конструисања* и својим радом допринели истраживању.

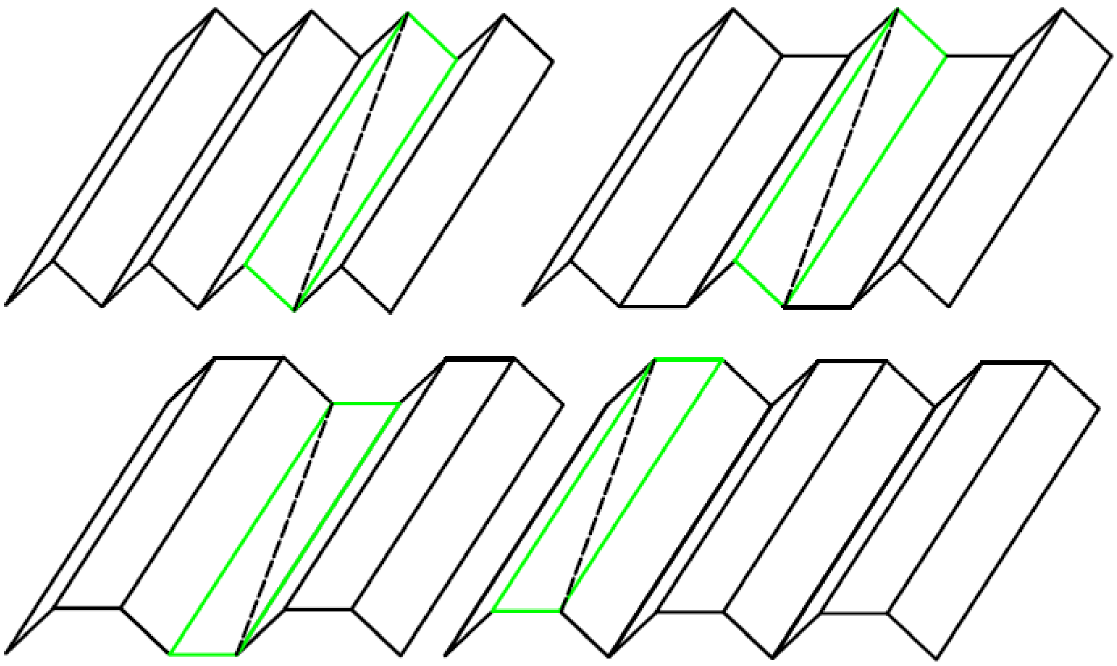
## 2.5 Подела набораних конструкција

У општем случају не постоји јединствена класификација набора, већ се она разликује од аутора до аутора. Ипак, можемо извршити поделу према облику колена набора, форми набора, тесалацији, конструктивном понашању, материјалу, архитектонској геометрији (слика 2.3.17.) и слично.

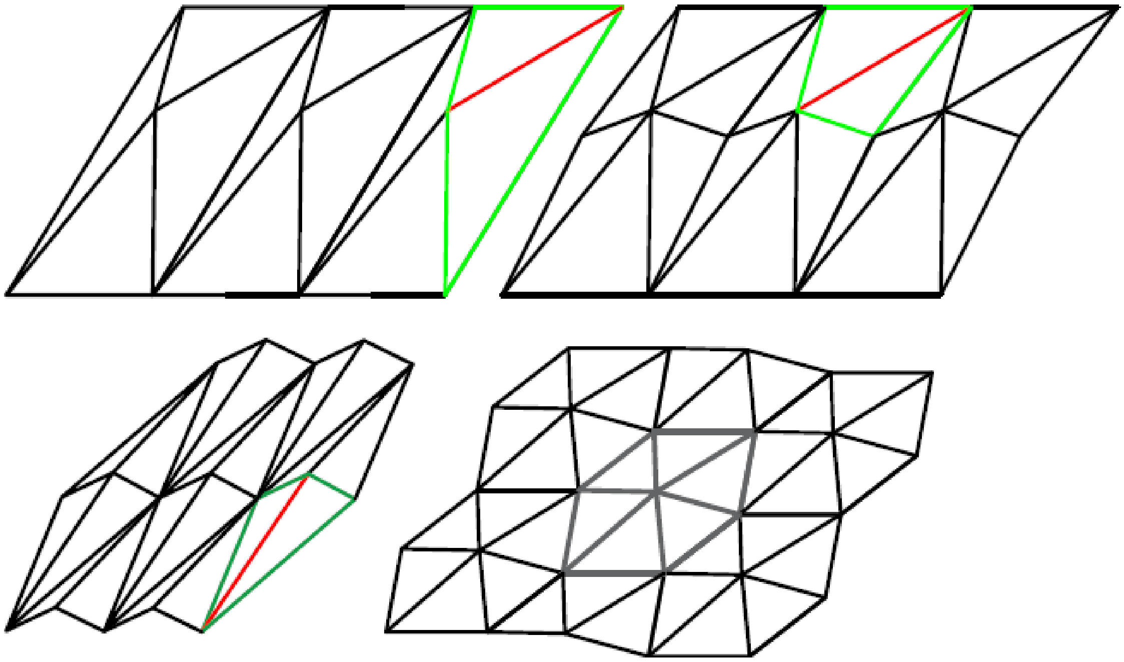
### 2.5.1 Подела набора према облику колена набора

Према облику колена набора усвојена је општа класификација на :

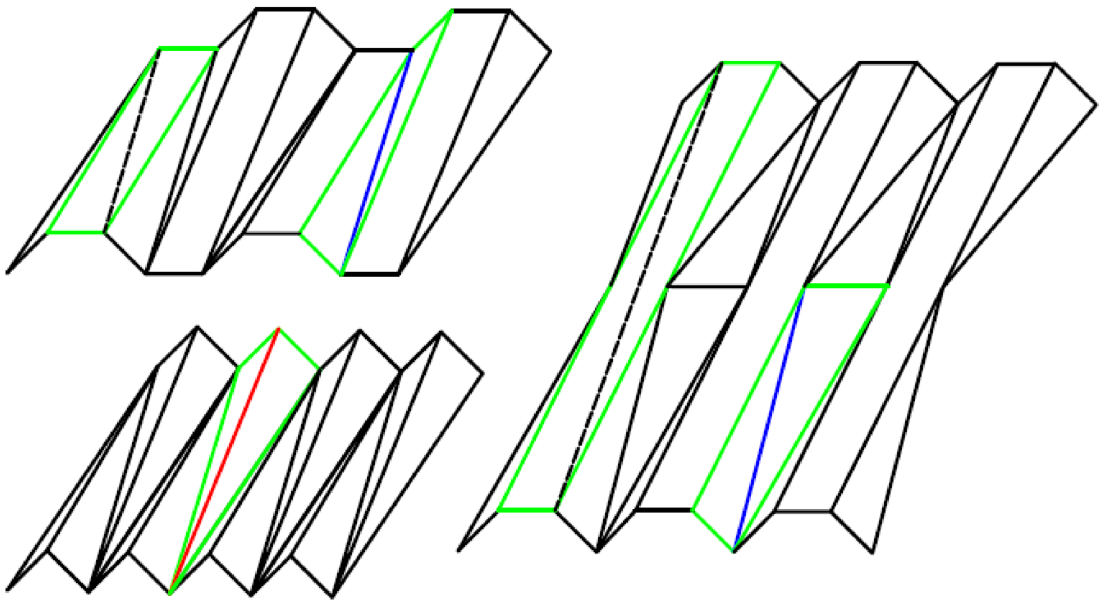
- Призматичне наборе - састављени од правоугаоних плоча (слика 2.5.1.)
- Пирамидалне наборе - састављени од плоча које нису правоугаоне (слика 2.5.2.)
- Полупризматичне - састављене од комбинације троуглова и трапеза (слика 2.5.3.)



Слика 2.5.1. Призматични набори



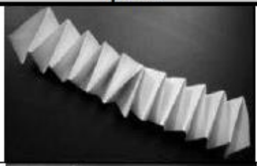
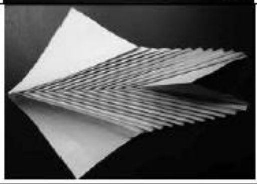
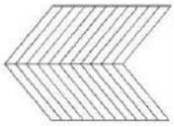
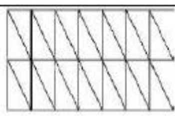
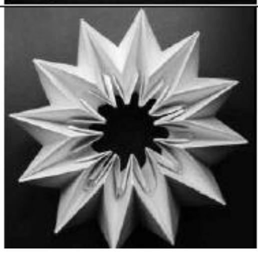
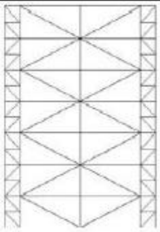
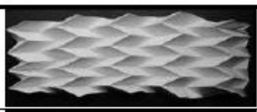
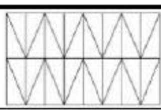
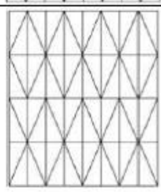
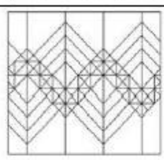
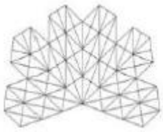
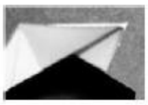
Слика 2.5.2. Пирамидални набори



Слика 2.5.3. Полупризматични набори

### 2.5.2 Подела набора према форми набора

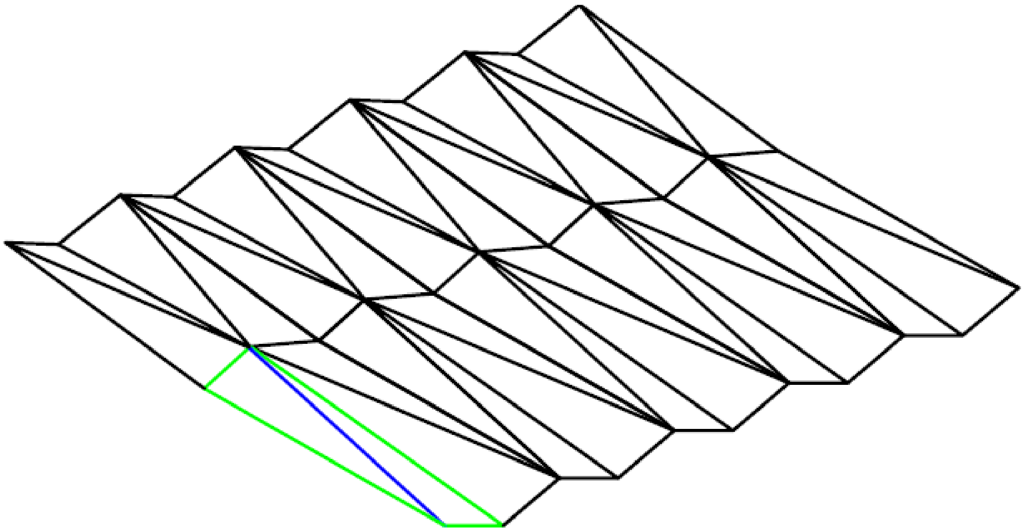
У овом случају не можемо рећи да постоји прецизна подела набора, већ се називи и типови који се могу наћи у литератури углавном разликују. На слици 2.5.4. приказана је подела коју је усвојио проф. Несторовић**.**

| бр. | тип набора | приказ | бочна пројекција набора | мрежа |
|---|---|---|---|---|
| 1 | Централни стуб |  |  |  |
| 2 | Цик-цак или лист дрвета |  |  |  |
| 3 | Опруга |  |  |  |
| 4 | Ватромет |  |  |  |
| 5 | Рибља кост |  |  |  |
| 6 | Магична лопта |  |  |  |
| 7 | Готска река |  |  |  |
| 8 | Лотос |  |  |  |
| 9 | Пирамида |  |  |  |
| | Напомена | Изабрани типова набора покривају велики дијапазон; сваки од њих је могуће развити. Потребно је напоменути да не постоји конзистентност у номенклатури, овом приликом су одабрани често коришћени називи, али је у литертури могуће пронаћи и другачије. | | |

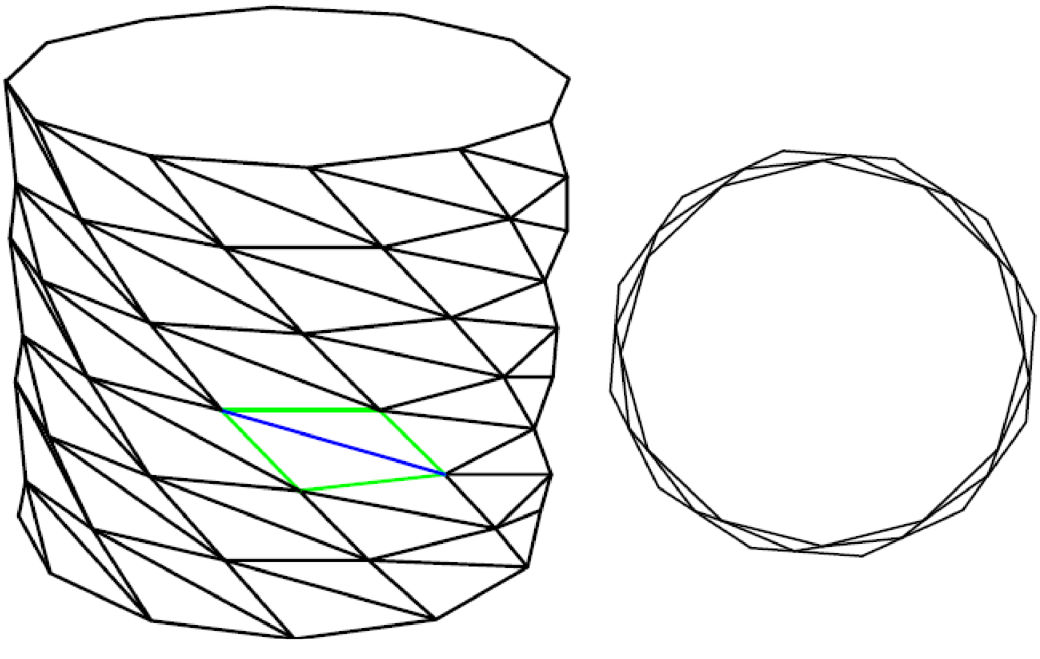
Слика 2.5.4. Подела набора према форми набора

### 2.5.3 Подела набора према тесалацији

Разликујемо наборе са хоризонталном и вертикалном тесалацијом што је приказано на сликама 2.5.5. и 2.5.6. Хоризонтална тесалација је одлика кровних конструкција и рамова уколико се ради о халама, она се одвија у једном или два правца која леже у хоризонталној равни. Вертикална тесалација се врши око вертикалне осе и такав вид набора треба да је способан да прими момент савијања у сва три правца.



Слика 2.5.5. Хоризонтална тесалација - кров



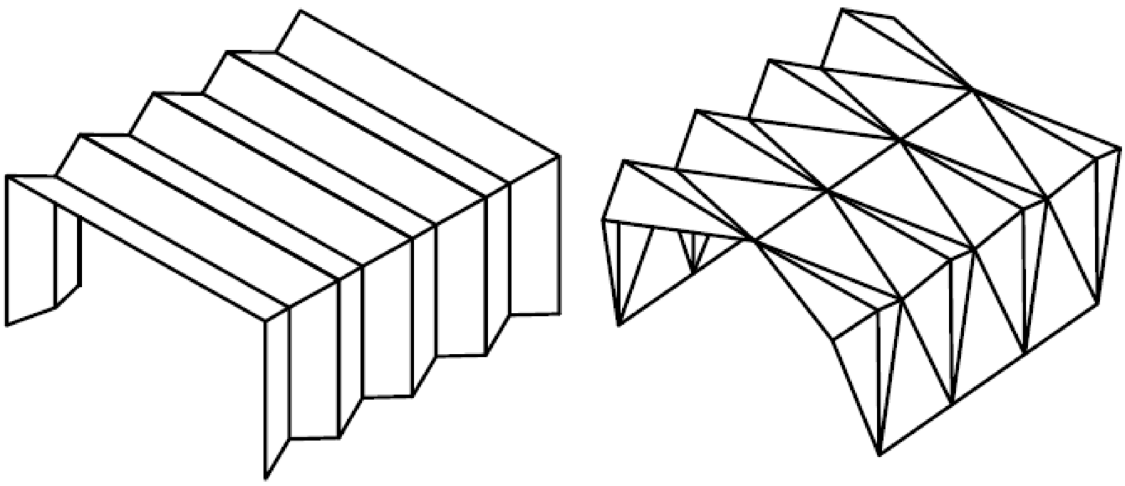
Слика 2.5.6. Вертикална тесалација - фасада уједно и носећа конструкција

### 2.5.4 Подела набора према конструктивном понашању

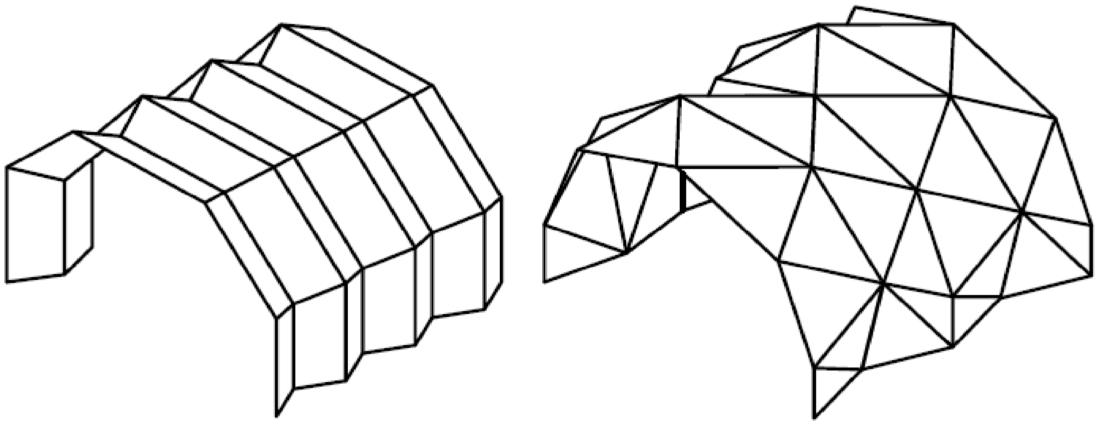
У општем случају, у зависности од начина прихватања и дистрибуције оптерећења, можемо извршити поделу на:

- гредне системе (слика 2.5.7.)
- рамовске (оквирне) системе (слика 2.5.8.)
- лучне (слика 2.5.9.) системе
- куполасте (слика 2.5.10.) системе и
- конзолне (слика 2.5.11.)

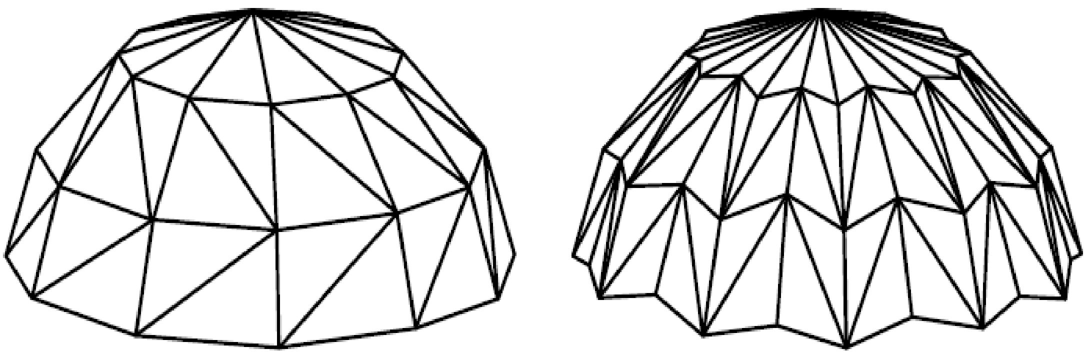
Слика 2.5.7. Гредни систем



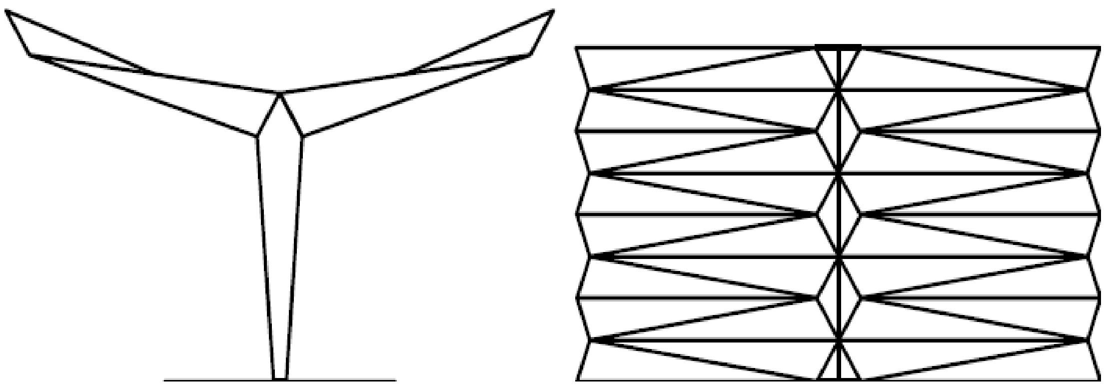
Слика 2.5.8. Оквирни систем



Слика 2.5.9. Лучни систем



Слика 2.5.10. Куполасти систем

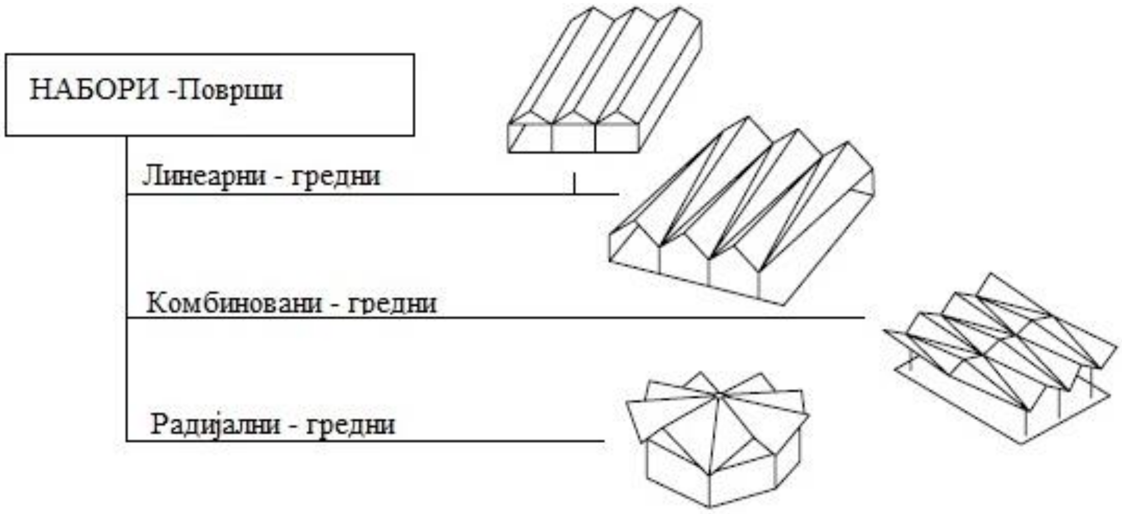


Слика 2.5.11. Конзолни систем

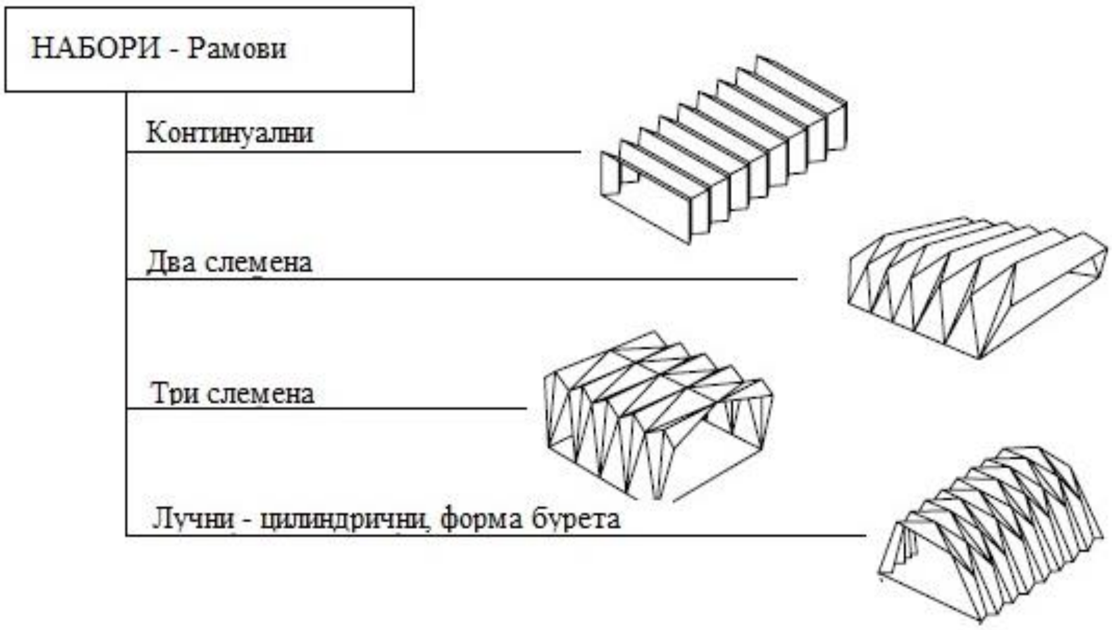
Детаљнију поделу, која се не ослања само на конструктивни систем већ и на просторну форму набора, представио је Шулер (Schueller, 1983). Шулерову поделу је искористио и прихватио и Шекуларац (Šekularac, Ivanović Šekularac, & Čikić

Tovarović, 2012) (слика 2.5.12.). Према Шулеру све наборе према конструктивном понашању могу се сврстати у три велике групе:

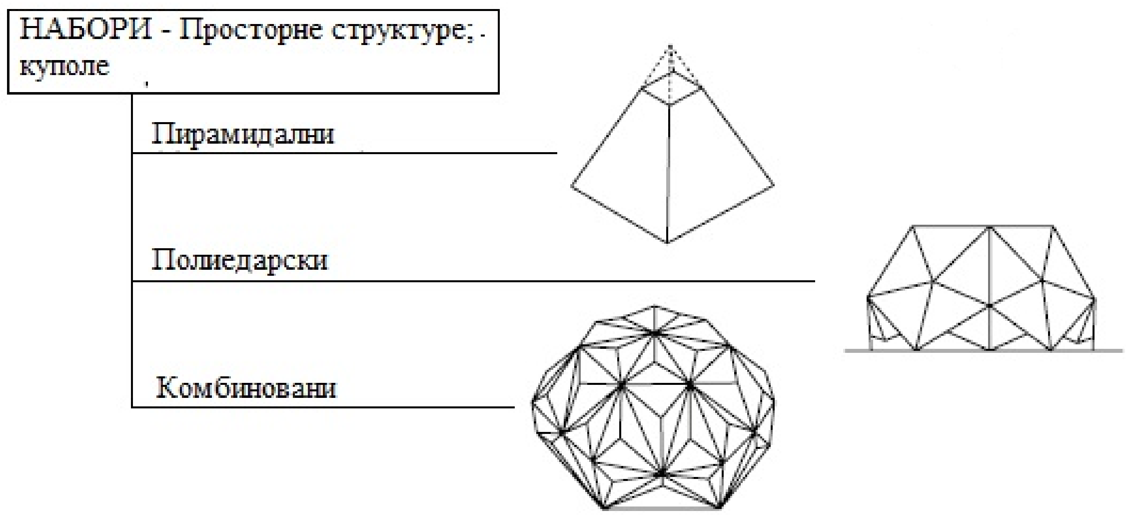
- Површи, односно гредни (слика 2.5.12.)
- Рамови (слика 2.5.13.)
- Просторни - куполасти (слика 2.5.14.)




Слика 2.5.12. Гредни системи




Слика 2.5.13. Рамовски системи




Слика 2.5.14. Просторни системи

**Напомена:** важно је разликовати конструкције набора састављених од плоча (оригинално: folded plate structure) и набора који испуњавају само форму набора док им је конструктивни систем сачињен од штапастих елемената који у суштини чине просторну структуру или решетку (оригинално: folded form structure).

У разним литературама имамо примере који могу да унесу забуну. Наиме, постоје изведени примери који спољашњим изгледом у потпуности испуњавају форму набора, а заправо конструктивни систем је сачињен од штапастих носача који чине просторну структуру и прекривени су лаганим материјалом који има чисто естетску функцију. На могућност уношења забране скреће пажњу и Мулџадината (Muljadinata & Darmawan, 2016), где је отишао чак и корак даље тражећи јасно разграничење ова два појма, односно форме и структуре. На слици 2.5.14. приказан је преглед објеката које имају само форму набора и објеката који се према Шулеровој (Schueller, 1983) подели третирају као структурални набори (folded plate structure) (Muljadinata & Darmawan, 2016). Може се закључити да велики број објеката само визуелно испуњава форму набора.

У складу са претходном констатацијом треба напоменути да је предмет овога рада набор као конструктивни систем састављен од плоча. Проблематика решавања таквог набора се у суштини своди на решавања теорије плоча. Додатну компликацију ствара чињеница да су плоче просторно распоређене под различитим угловима и да су на ивицама круто везане формирајући монолитну конструкцију. Просторни распоред плоча набора има за последицу комплексан прорачун напона конструктивних елемената, што је уједно и главна проблематика рада. Фундаментално разматрање конструктивног понашања набора представљено је у следећем поглављу.

| Building | Picture | Folded Form Building | Folded Plate Structure (System, based on Schueller) |
|---|---|---|---|
| The folded roof over the Verenigd Plastic verkoopkantoor N.V. laboratory building in Zeist, in The Netherlands. (1960) |  | Yes | Yes (Folded Plate Surface – Linear) |
| Precast concrete folded plate façade and roof of the Church in Hoensbroek, Heerlen, in The Netherlands. (1964) |  | Yes | Yes (Folded Plate Frame – Continues) |
| The folded roof, façade, and floor of The Human Resource Development Centre in Jakarta, Indonesia. (1977) |  | Yes | Yes (Folded Plate Frame – Two Hinged) |
| The United States Air Force Academy Cadet Chapel in Colorado, USA. (1962) |  | Yes | No |
| The Space Truss Roof Structure of the Transair Hangar in Gatwick Airport, London, UK. (1958) |  | Yes | No |
| The International Convention and Exposition Centre in Naning, China. (2003) |  | Yes | No |
| The Osanbashi Yokohama International Passenger Terminal in Japan. (2002) |  | Yes | No |
| The Istora Senayan Sport Hall in Jakarta, Indonesia. (1962) |  | Yes | No |
| The Miami Marine Stadium in Florida, USA. (1963) |  | Yes | No |
| The University of Illinois State Farm Center in Illinois, USA. (1964) |  | Yes | Yes (Folded Plate Dome) |

Слика 2.5.15. Преглед изведених објеката

## 2.6 Конструктивно понашање набора

Код површинских носача дебљина носача је мала у односу на друге две димензије, дужину и ширину. Код њих подужни напони у оба правца су истог реда величине, док је напон управан на њих прилично мањи тако да се може занемарити. Површински носачи се могу поделити на плоче и љуске.

Основни елемент набора чини плоча. Скуп плоча спојених дуж својих ивица под различитим нагибима у монолитну целину формира набор. Плоча је тело чија је једна димензија мала у односу на друге две. Ограничена је са две паралелне равни и цилиндричном површином управно на њу. Средња раван плоче је раван која полови дебљину плоче, док је еластична површина средња раван при деформацији плоче. Захваљујући просторном распореду плоча набори поседују велику крутост на савијање што их чини веома повољним за премошћавање великих распона. Чак и једноставан набор (слика 2.6.1.) поседује завидан ниво крутости на савијање.

Иако је у суштини набор тродимензионалан проблем, можемо упростити проблематику и посматрати га као дводимензионални проблем како би уз помоћ основне статике, прерасподеле оптерећења и тригонометрије боље разумели понашање и потенцијал примене набора. На слици 2.6.1. је приказана плоча дебљине 15 cm у хоризонталном положају и иста плоча нагнута под углом од 30 °.

Рецимо да се у конкретном случају ради о армирано-бетонској плочи које су погодне за распоне 3-5 m у зависности од пројектованог оптерећења (Gomez, n.d.). Капацитет носивости плоче се разликује било да се ради о армирано - бетонској плочи или преднапрегнутој. У хоризонталном положају плоча поседује отпорни момент:

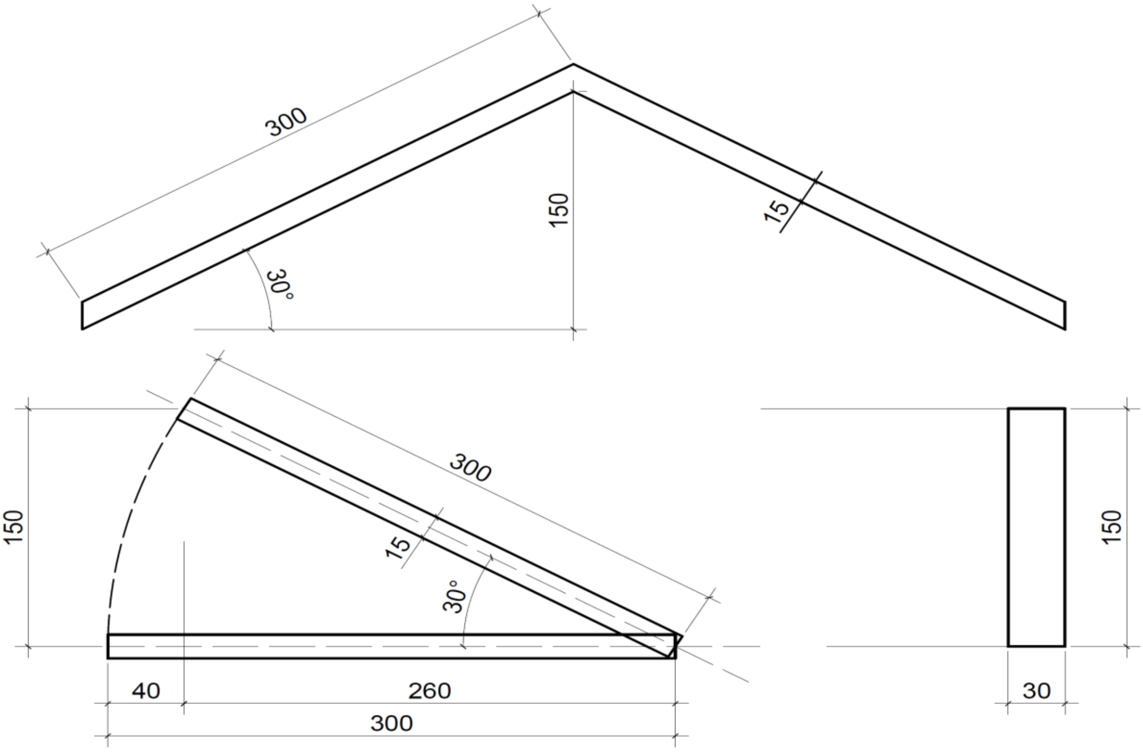
$$W = \frac{bd^2}{6} \tag{1}$$

при чему је b-ширина, а d-дебљина плоче. Њен отпорни момент износи:

$$W = \frac{300 \times 15^2}{6} = 11.250\ cm^3 \tag{2}$$

Уколико сада издигнемо исту ту плочу на висину од 150 cm, аналогни попречни пресек је димензија 30 x 150 cm, а отпорни момент сада износи:

$$W = \frac{30 \times 150^2}{6} = 111.250 \text{ cm}^3 \quad (3)$$

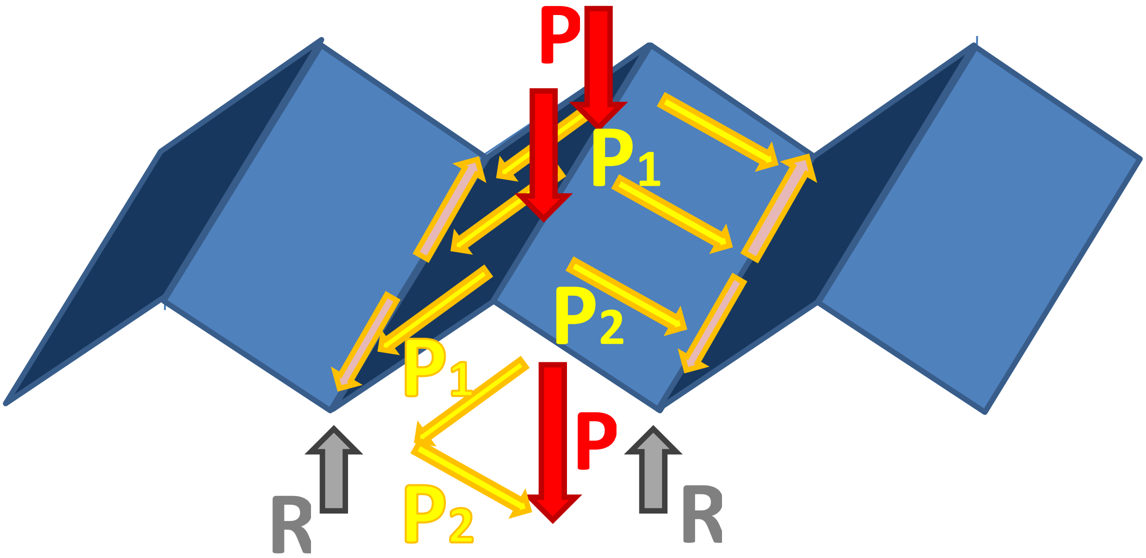



Слика 2.6.1. Геометрија основног набора

Уједно се и смањила покривна површина за 40 / 300 = 0.133. Закључујемо да се у замену за 13.3% већег утрошка материјала добија 10 пута већа крутост конструкције управно на хоризонталну пројекцију набора. Код сложенијих геометрија набора могуће је добити чврстоћу на вертикално оптерећене већу и до 100 пута, што ћемо показати касније на моделима. Треба напоменути да се формирањем набора губи крутост на истезање у хоризонталној равни, па је неопходно предвидети дијафрагме у циљу прихватања оптерећења.

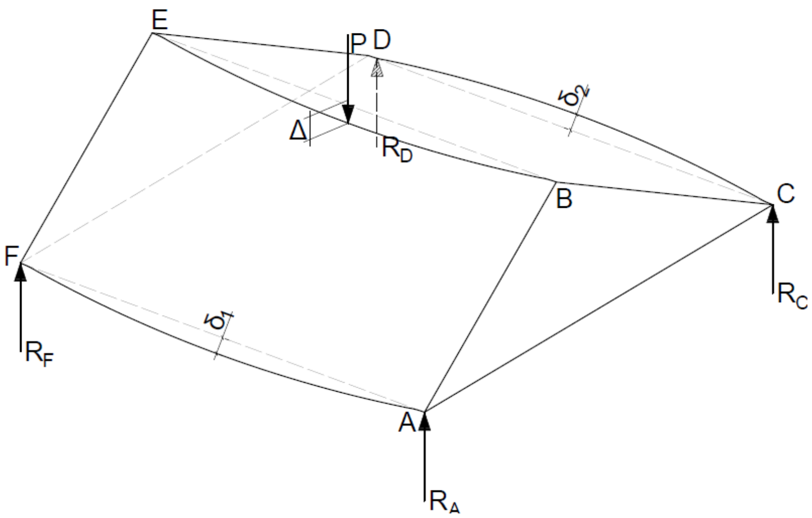
Посматрајмо конструкцију најједноставнијег облика набора састављеног од косих правоугаоних плоча круто спојених дуж својих дужих страница. Другим речима, правоугаоне плоче формирају таласе као код хармонике (слика 2.6.2.). На ослонцима набор је круто везан са попречним дијафрагмама или оквирима тако да формира укљештење. Његове ивице се понашају као ослоначки појасеви.

Уколико набор оптеретимо вертикалним оптерећењем (што је најчешћи случај) прерасподела оптерећења ће се вршити као што је приказано на слици 2.6.2. Прво, спољашње оптерећење се преноси дуж краће стране, потом се дуж споја суседних елемената оптерећење преноси аксијално до ослонаца. Оваква прерасподела сила се остварује захваљујући попречној крутости набора. Ако су суседне плоче нагнуте под истим углом $P_1=P_2=P/2$.



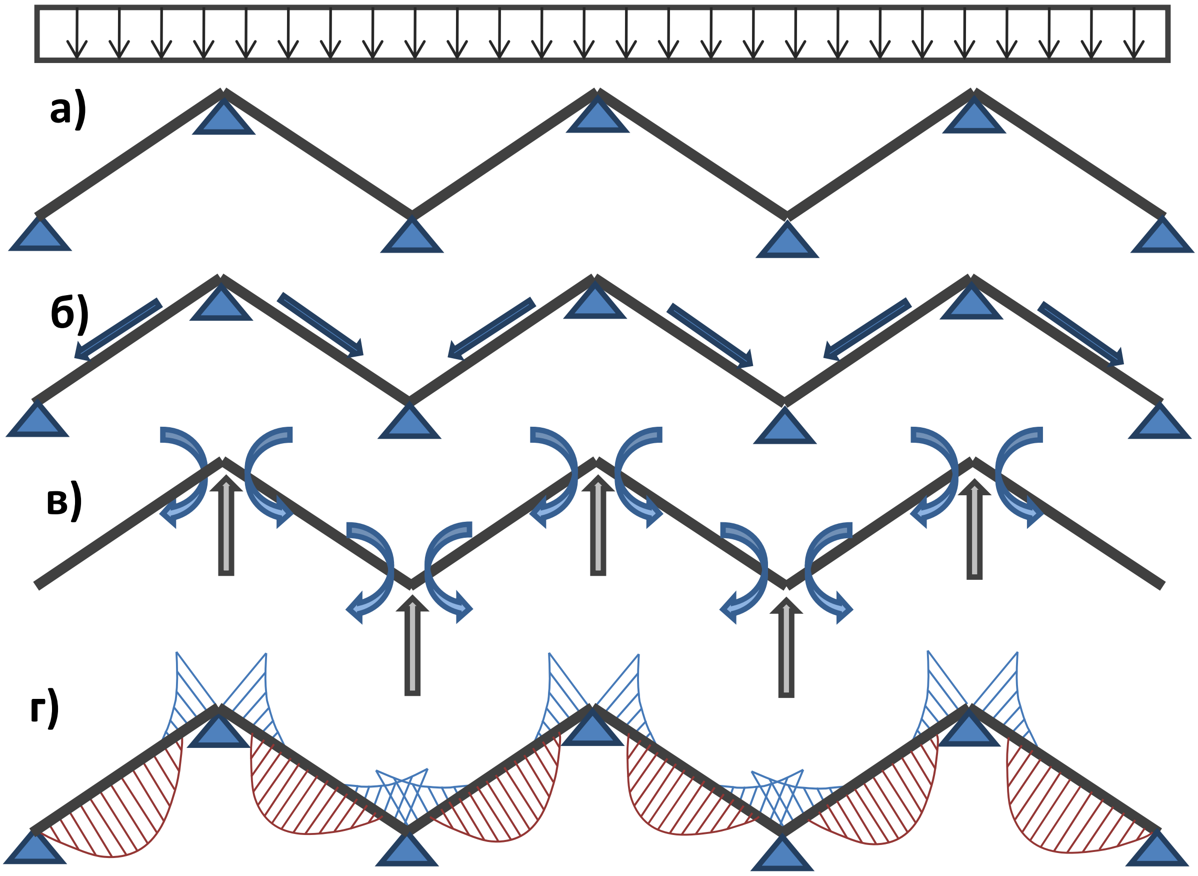

Слика 2.6.2. Прерасподела оптерећења

Даље, на слици 2.6.3. су приказане деформације набора у случају вертикалног оптерећења, при чему је Δ- укупна вертикална деформација у средини распона (угиб), а $\delta_{1,2}$ деформације одговарајућих плоча у сопственој равни. Деформације се могу одредити употребом Вилотовог дијаграма.




Слика 2.6.3. Деформација набора услед вертикалног оптерећења

Реакције ослонца износе $R_A=R_B=R_C=R_D=P/4$.

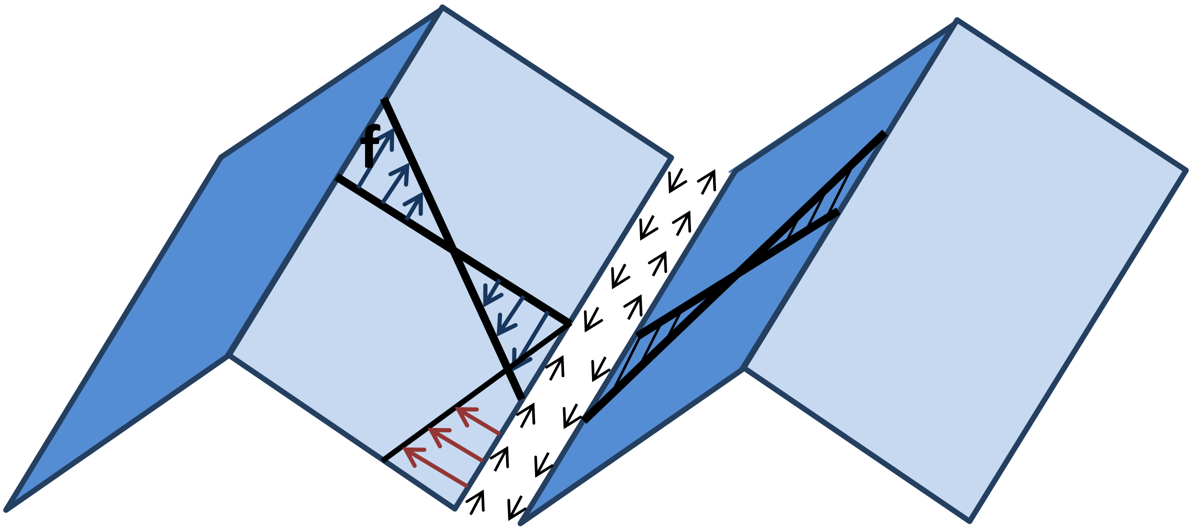



Слика 2.6.4. Статички утицаји (а, б, в, г)

Узмимо да на набор делује континуално вертикално оптерећене (слика 2.6.4. а)). При томе се јављају аксијалне силе, реакције ослонца и моменти савијања као што је приказано на слици 2.6.4. б) и 2.6.4. в). У попречном правцу, због присуства дијафрагми, набор се понаша као континуални носач. Дијаграм момента приказан је на слици 2.6.4. г), при чему су у горњој зони јавља затезање изнад ослонаца, а у доњој зони затезања у пољу.

Напони проузроковани деловањем смичућих сила су веома битни при пројектовању набора. Смичући напони у оба правца свој максимум достижу на ивицама плоча и имају нулту вредност у тежишту плоче (слика 2.6.5.). При чему се вредност напона може срачунати као:

$$\sigma = \frac{M}{W} = \frac{6M}{bd^2} \tag{4}$$




Слика 2.6.5. Напони смицања

Из приказаног произилази да се плоча набора конструктивно третира као танка плоча, односно као сложен скуп просторно распоређених елемената набора (плоча), што значајно компликује прорачун конструкције набора који ће бити описан овим радом.

## 2.7 Примери објеката изведених у пракси

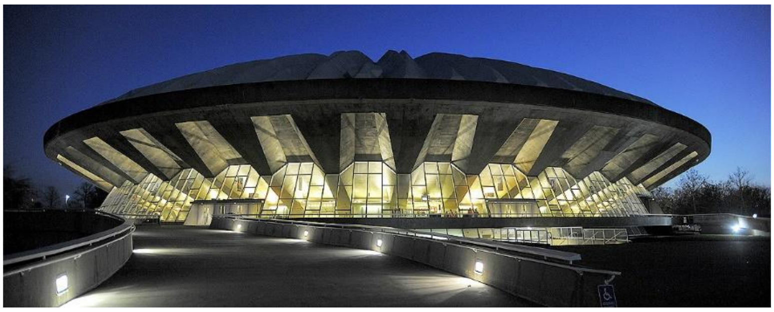
### 2.7.1 State Farm Center

Илиноис, САД, 1963.

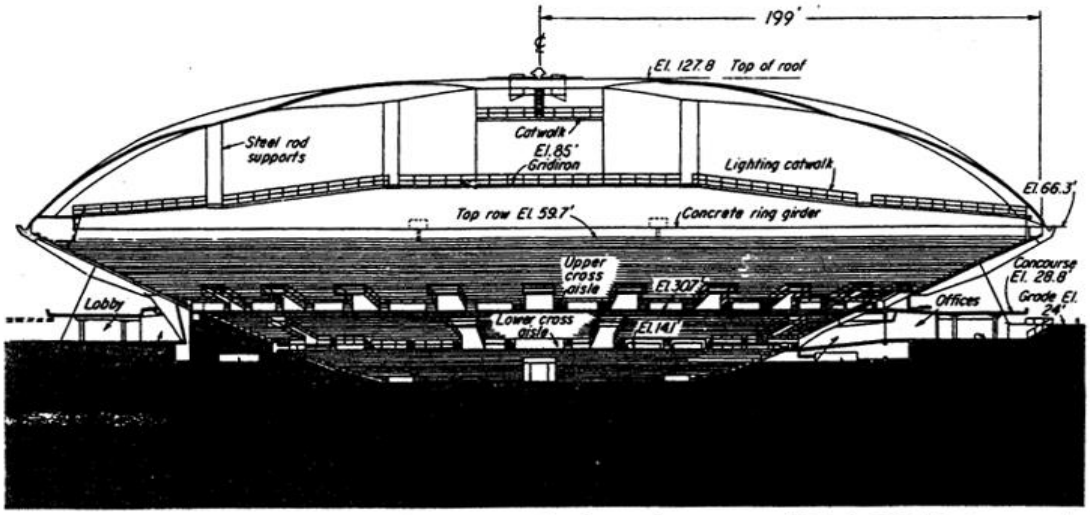
Архитекте:

**Harrison & Abramovitz**

Набoрана купола
распона 122,0m

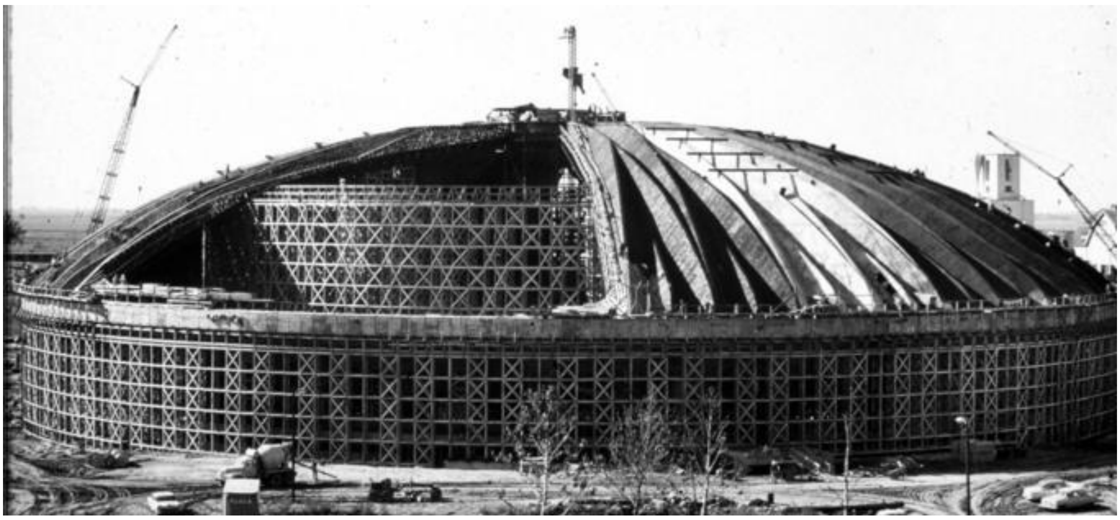


Слика 2.7.1.1. Набoрана купола дворане у Илиноису САД




**Dome of University of Illinois "bowl" rises 128 ft above the arena, which is 24 ft below grade.**

Слика 2.7.1.2. Попречни пресек набoране куполе у Илиноису, САД



Слика 2.7.1.3. Изградња дворане у Илиноису, САД

### 2.7.2 Gunma Music Center

Такасаки, Јапан, 1961.

Архитекта:

**Antonin Raymond**

Конструктор:

**Tsuyashi Okamoto**

Бетонски набор распона 60,0m

Слика 2.7.2.1. Гунма музички центар Такасаки, Јапан

Слика 2.7.2.2. Бетонски набори Такасаки, Јапан

Слика 2.7.2.3. Гунма музички центар Такасаки, Јапан

### 2.7.3 Miami Marine Stadium

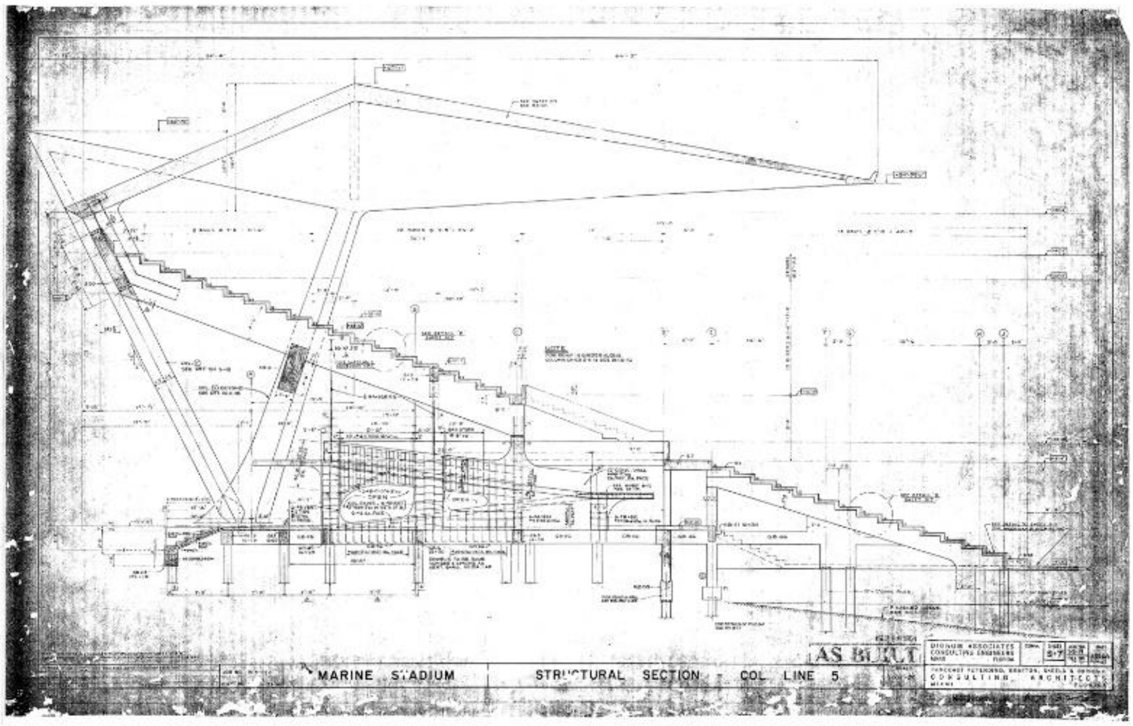
Мајами, САД, 1963.

Архитекта:

**Hilario Candela**

Армиранобетонска конзола у виду набора ослоњена на 8 стубова

Слика 2.7.3.1. Армиранобетонска конзола Мајами, САД




Слика 2.7.3.2. Попречни пресек Мајами, САД

Слика 2.7.3.3. Приказ трибине Miami Marine Stadium

### 2.7.4 Yokohama International Port Terminal

Јапан 2002.

Архитекта:

**Foreign Office Architects (FOA)**

**Farshid Moussavi**

**Alejandro Zaera Polo**

Слика 2.7.4.1. Yokohama International Port Terminal, Јапан

Слика 2.7.4.3. Yokohama International Port Terminal, Јапан

Слика 2.7.4.2. Yokohama International Port Terminal, Јапан

### 2.7.5 American Concrete Institute Building

Детроит, САД, 1958

Архитекта:

**Minoru Yamasaki**

Слика 2.7.5.1. American Concrete Institute Building, Детроит

Слика 2.7.5.2. American Concrete Institute Building, Детроит

Слика 2.7.5.3. American Concrete Institute Building, Детроит

**2.7.6 Електрана у Бирсфелдену**

Бирсфелдену, Немачка,

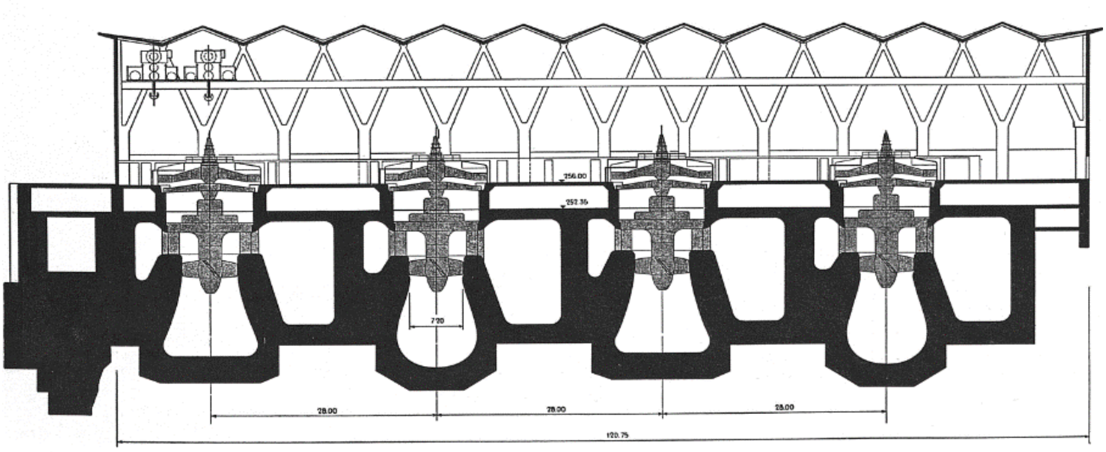
1957

Архитекта:

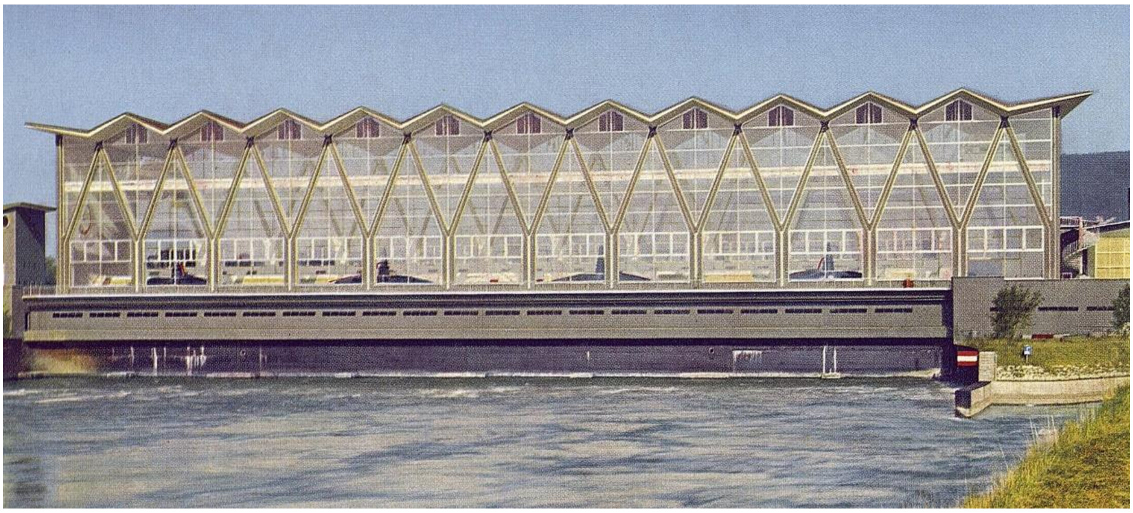
**Hans Hoffman**

Набор распона 30,0m

Слика 2.7.6.1. Електрана у Бирсфелдену, Немачка



Слика 2.7.6.2. Електрана у Бирсфелдену, Немачка



Слика 2.7.6.3 Електрана у Бирсфелдену, Немачка

### 2.7.7 Спортски центар Mülimatt

Windisch, Швајцарска, 2010

Архитекта:

**Studio Vacchini Architetti, Locarno**

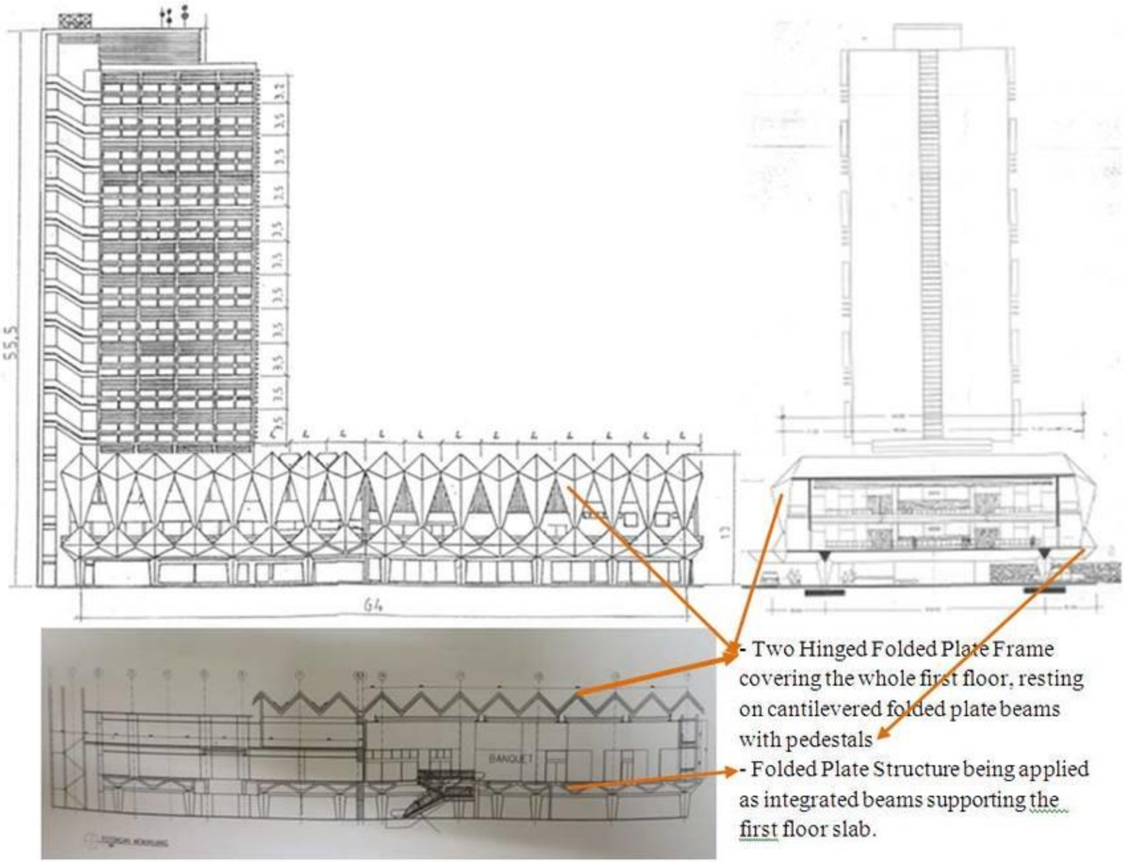
Слика 2.7.7.1. Спортски центар Mülimatt, Швајцарска

Слика 2.7.7.2. Спортски центар Mülimatt, Швајцарска

Слика 2.7.7.3. Спортски центар Mülimatt, Швајцарска

## 2.7.8 Human resource center

Џакарта, Индонезија, 1977




Слика 2.7.8.1. Human resource center Џакарта, Индонезија

Слика 2.7.8.2. Human resource center Џакарта, Индонезија

# 3. ГЕНЕРИСАЊЕ ГЕОМЕТРИЈЕ ПРИМИТИВА И КОНСТРУКЦИЈЕ

## 3.1 Опште о генерисању геометрије - предпроцесирање

Геометријско генерисање комплексних структура у области пројектовања конструктивних система представља систематизован и свеобухватан корак по корак процес. Генерисање представља процес израде математичке репрезентације геометрије реалног система, што уједно представља и полазну тачку при прорачуну конструкције. Оно је веома заступљено, јер је експериментисање на реалном моделу, углавном из економских разлога, немогуће извести. Управо је **модел** замена за реалан систем. Циљ креирања генерисаног модела је упознавање понашања реалног система при промени задатих улазних параметара. Инжењерски посматрано модел треба да што реалније преслика стварност у складу са траженом прецизношћу и ценом израде. Геометрију физичког модела са граничним условима (ослањање, оптерећење) је потребно пресликати у рачунарски модел. Моделирање се врши преко избора типа, броја и величине коначних елемената, степена слободе чворова, граничних услова уз идеализацију и упрошћавање. Може се рећи да је израда модела уједно и најосетљивији и најтежи проблем који је потребно решити. Ваљаност модела у највећој мери зависи од интуитивности инжењера и његовој способности да адекватно представи стварни систем и симулира одговарајуће утицаје.

Геометријско дефинисање структуре представља први корак при креирању модела применом геометријских трансформација и операција. Потребно је познавати основне карактеристике и међусобне зависности елемената. Изабрана просторна структура поседује уређен скуп тачака и елемената. Задатак пројектанта јесте да уочи правилност геометрије структуре (Milošević, 2015). Потом, уз помоћ алгоритама за аутоматско генерисање представи и опише све елементе предметне конструкције (Stošić, 1999) (Rogačev, 2006).

Приликом пројектовања архитектонских облика неретко се дешава да је јако тешко препознати правилности форме, што доводи до немогућности описивања геометријских својстава конструкције постојећим генераторима. Математички апарат предпроцесирања је често недовољно развијен и поприлично крут у домену генерисања разноврсних и комплексних архитектонских форми, уједно је од изузетног значаја. Допринос овог рада је уједно и развој генератора (предпроцесорских рутина и модула) за описивање геометрије набора. Помоћу оваквог приступа и развоју одговарајућих алата омогућава се имплементација описивања дефинисане структуре применом методе коначних елемената у креирању и пројектовању архитектонске форме. Овакав приступ архитектонском и конструктивном решавању и пројектовању конструкције представља неизбежан алат. Од релевантног описа геометрије уједно зависи и прецизност резултата добијених анализом МКЕ, зато је јако важно имати на располагању одговарајући генератор геометрије. (Zloković Đ. , 1969) (Zloković Đ. , Prostorne strukture, 1969)

За потребе овог рада направљен је генератор структура набора и приказан је потенцијал разноврсне употребе и проширења могућности истог генератора за широк спектар произвољне форме набораних структура. Функционалност и примена генератора су приказани кроз примере које следе. Главну проблематику је представљало дефинисање правила у процесу генерисања, као и геометријска дефиниција просторних мрежа тачака. При томе, положај тачака у простору дефинише чворове просторних структура које је потребно описати одговарајућим математичким апаратом.

Комплексан процес генерисања геометрије пресликавањем физичког у рачунски модел применом идеализације реалног система може се спровести коришћењем специјалних програма, односно путем разних алгоритама (процедурално моделирање). У новије време, све популарнија метода постаје скенирање стварног објекта на основу којег се интерпретира рачунски модел.

### 3.2 Генерисање геометрије структуре

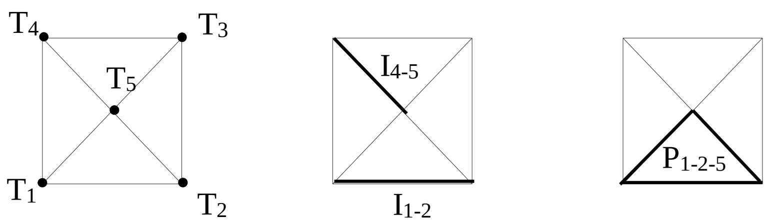
У веома софистицираном делу пројектовања генератор геометрије представља веома важну полазну базу. Имајући у виду чињеницу да процес архитектонског стварања има изражену и компоненту уметничког рада која захтева комплексан процес пројектовања, није тешко закључити да управо генератор геометрије има пресудну улогу при процесу статичко конструктивне анализе структуре. Развој архитектонске мисли директно је у спрези за развојем генератора геометрије који треба веродостојно математичким апаратом да прикаже креирану архитектонску структуру. Стога је неопходно да процес припреме и дефинисање структуре представља скуп корака који се рачунски могу дефинисати и приказати. С обзиром да је углавном за потребе математичке презентације архитектонске форме, односно статичких и динамичких прорачуна применом методе коначних елемената неопходно унети јако велики број улазних параметара, у првом плану, везаних за дефинисање чворова. Основни циљ при генерисању геометрије примитива набора је био да се уз помоћ малог броја улазних параметара генерише пуно података формирајући комплексну просторну структуру како би се омогућило све наведено, неопходно је одлично познавање алгоритма и кораке генерисања геометрије. Припрема улазних података је уједно и најтежи задатак. Њихов одабир представља јако осетљив почетни корак. У општем случају процес генерисања геометрије можемо приказати следећом 3.2.1.).


Слика 3.2.1. Алгоритам генерисања

За потребе овог рада креиран је јединствен генератор геометрије писан у програмском језику FORTRAN који је приложен у прилогу на крају рада. (Прилог 1). Статичка и динамичка анализа су извршени у програму KOMIPS (Maneski, 1998). Програм је отвореног типа и компатибилан је са програмским језиком FORTRAN. Захваљујући модуларно конципираном систему креирани алгоритми за моделирање набораних конструкција коришћењем неког од типа примитива су интегрисани у KOMIPS програмским пакет додатно проширујући његов спектар могућности. Компујтерско моделирање и прорачун структура (KOMIPS) је заснован на примени методе коначних елемената (МКЕ) и омогућава моделирање и прорачун комплексних конструкција, одређивање напона и померања, стварног понашања конструкције и њених елемената (Maneski, 1998). Овакав приступ проблему омогућава и интегрисање групно суперматричне процедуре у циљу смањења неопходних математичких операција. Детаљнији процес моделирања биће представљен уз опис креирања макро коначних елемената набора.

### 3.3 Избор примитива набора

У складу са обимом овог рада, дефинисани су и креирани 15 различитих основних облика набора, у даљем тексту примитиви. Примитиви се математичким операцијама, транслацијом и ротацијом, односно међусобним комбиновањем преводе у комплексне архитектонске форме различитих геометријски и физичких карактеристика. Први корак при креирању примитива је одређивање основних елемената који учествују у формирању примитива (Moussavi, et al., 2009). Да би било могуће једноставно вршити математичке операције у простору над примитивима, неопходан услов је да пројекција контуре примитива има правоугаони облик. Генерисање и вршење математичких операција над примитивима који не испуњавају наведени услов је знатно сложеније и превазилази обим овог рада, самим тим може бити предмет даљег истраживања. (Слика 3.3.1.). Раније смо напоменули да симетричност у процесу моделирања доприноси смањењу времена анализе конструкције, фундаментални циљ дефинисања примитива, односно касније просторне структуре набора, јесте да се генерисање геометрије изврши са што мање улазних параметара.



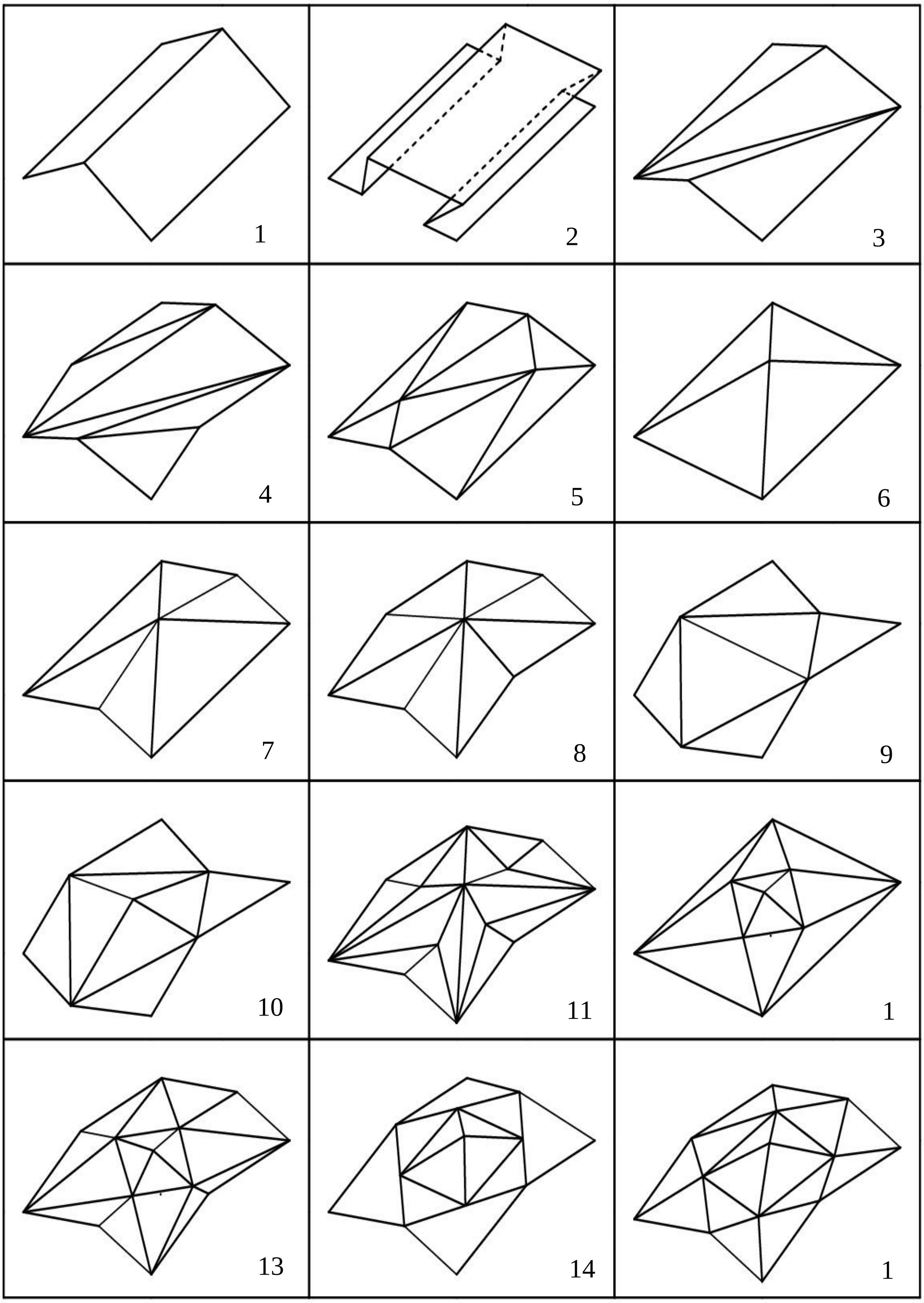

Слика 3.3.1. Основни елементи примитива

Сваки примитив се формира креирањем чворних тачака, њиховим повезивањем у линијске елементе, односно базисно раванске елементе, при чему дефинишемо следеће следеће појмове:

- Чворна тачка – формира се на контури примитива и у пресеку ивица $\mathbf{T_i}$
- Линијски елемент (ивица) – линија спаја две чворне тачке у унапред дефинисаном редоследу. Третира се као линијски коначни елемент. Представља дуж у пресеку суседних плоча - $\mathbf{I_{i\text{-}j}}$
- Плоча – базисни равански елемент, може бити троугаона или четвороугаона равна површина ограничена одговарајућим ивицама $\mathbf{P_{i\text{-}j\text{-}k}}$

Сви примитиви су формирани повезивањем плоча у простору, при томе креирање линијских елемената није обавезно, већ опционо и то у својству дефинисања укрућења дуж пресека плоча. Укрућења додатно доприносе крутости целокупног система. Кориснику је остављена могућност њиховог избора, такође у параметарском облику. Сваки примитив је састављен од најмање две плоче. Додатно, распоред тачака у простору по контури примитива мора бити такав да омогући надовезивање следећег примитива приликом његовог умножавања у простору. Другим речима, тачке у простору на контурним ивицама примитива морају бити подударне са тачкама надовезујућег сродног примитива, односно висине тачака по контури морају бити исте како би се обезбедила непрекидност структуре набора. Сам примитив по себи може бити антисиметричан и не мора поседовати нити једну особину симетрије.

Избор примитива који су коришћени у овом раду приказан је на слици 3.3.2. уз напомену да је за сваки примитив креиран математички алгоритам у параметарском облику за њихово генерисање.




Слика 3.3.2. Примитиви

### 3.4 Улазни параметри за генерисање примитива

Иако је обимом овог рада предвиђено креирање 15 примитива, на исти начин могуће је креирати и дефинисати било коју нову форму новог примитива. Сви примитиви се генеришу у параматарском облику у тродимензионалном кооординатном систему. Хоризонтална пројекција примитива лежи у равни x-z, док су у правцу y дефинисане висине чворних тачака. Важно је напоменити да координате висине у правцу y координате осе може имати и негативне вредности. За угаоне чворне тачке контуре, увек их има 4, важи да је y = 0 .

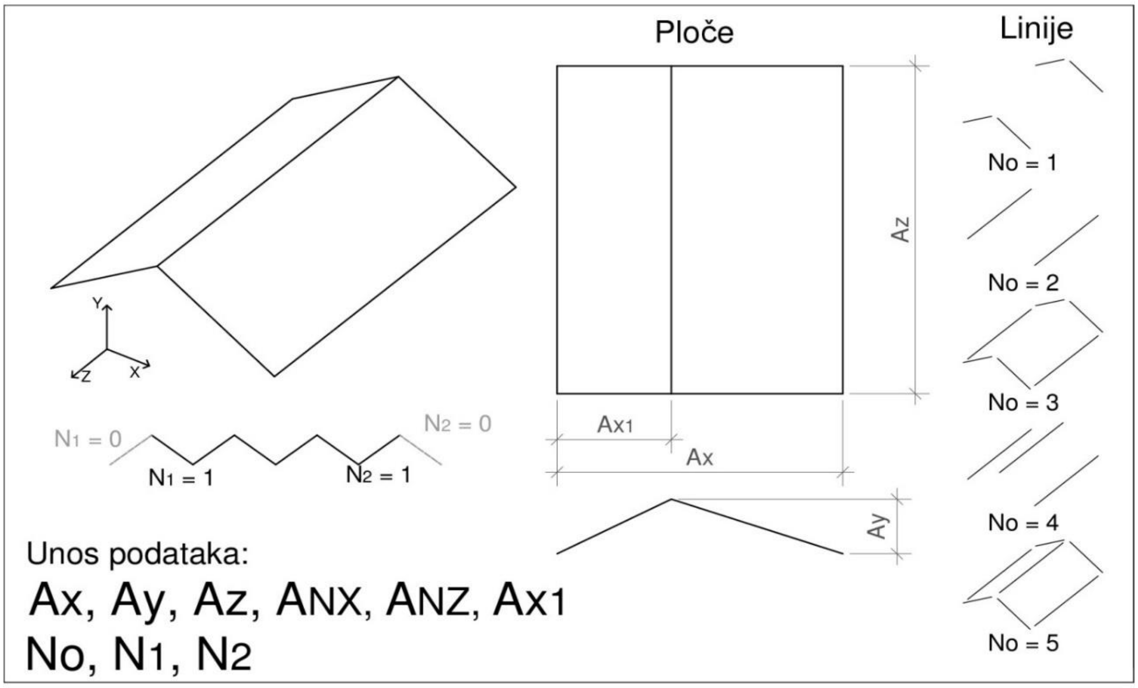
У општем случају дефинишемо следеће параметре:

- **$A_x$** – дужина пројекције контуре примитива у правцу x
- **$A_z$** – дужина пројекције контуре примитива у правцу z
- **$A_y$** – координата висине централне чворне тачке
- **$A_{nx}$** – умножавање примитива у правцу x
- **$A_{nz}$** – умножавање примитива у правцу z
- **$A_{x1}$** – дужина пројекције ивице примитива у x правцу, између прва два чвора са најмањом вредношћу x координате
- **$A_{z1}$** – дужина пројекције ивице примитива у z правцу, између прва два чвора са најмањом вредношћу z координате
- **$A_{yi}$** – координата висине одговарајућих чворних тачака, уз напомену да висине на које се два сродна примитива надовезују морају имати исту величину
- **$N_0$** – избор линијских елемената, укрућења набора**,** **$N_1$** – избор конкавног (0) или конвексног (1) завршетка при умножавању са леве стране
- **$N_2$**– избор конкавног (0) или конвексног (1) завршетка при умножавању са десне стране
- **$X_c$** – центар ротације у правцу x
- **$Y_c$** – центар ротације у правцу y

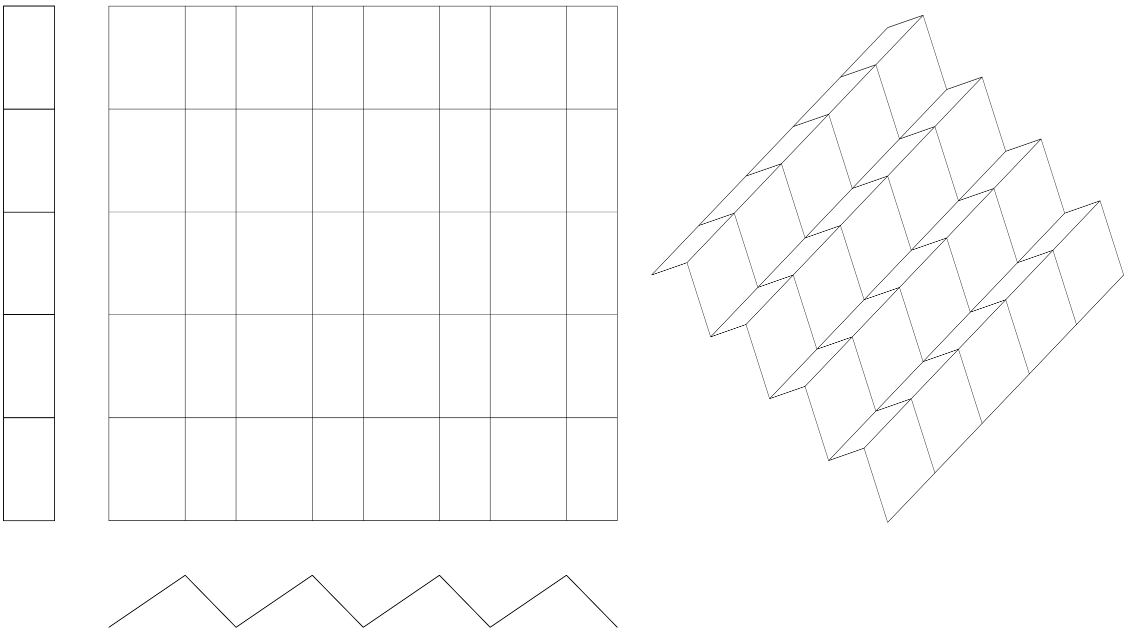
Параметри неопходни за генерисање примитива су приказани на Сликама које следе. Такође, за сваки примитив одрађен је и пример генерисања. У њима параметри **$N_0$, $N_1$**, **$N_2$**, **$X_c$** и **$Y_c$** су идентички једнаки нули.

## 3.5 Примитиви

### 3.5.1 Примитив типа P - 1



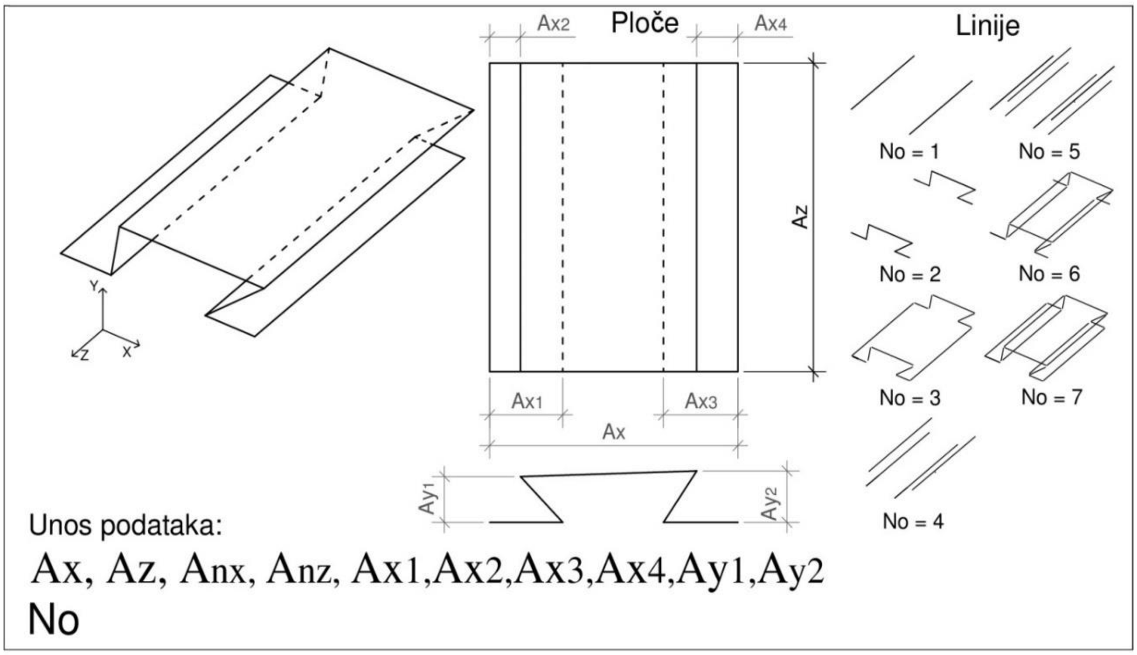

Слика 3.5.1. а) Параметри за генерисање примитива типа 1



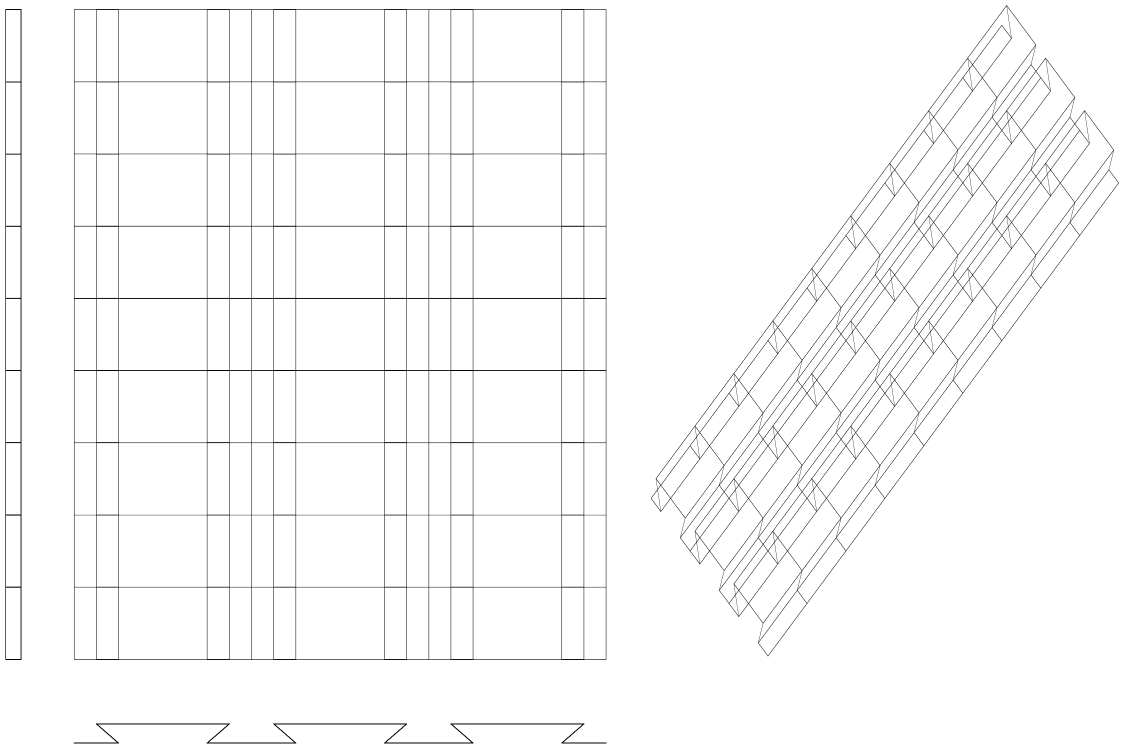
Слика 3.5.1. b) Пример примитива 1, раванске пројекције и изометрија

$A_x = 4; A_z = 3; A_y = 2; A_{nx} = 4; A_{nz} = 5$

### 3.5.2 Примитив типа P - 2



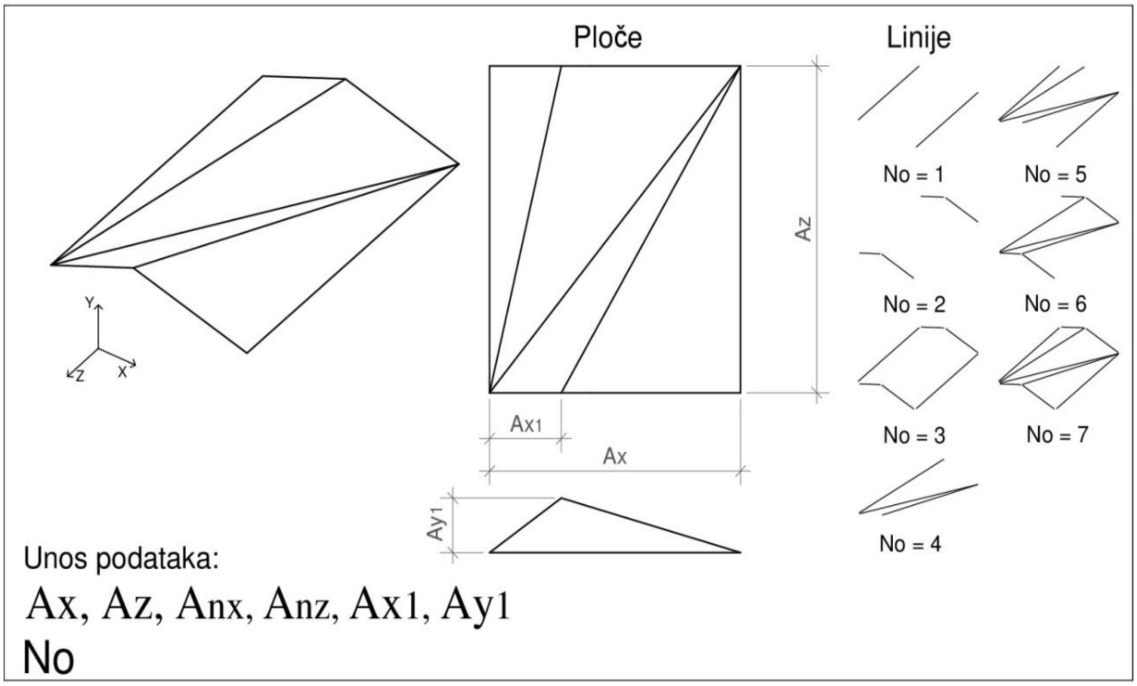

Слика 3.5.2. a) Параметри за генерисање примитива типа 2



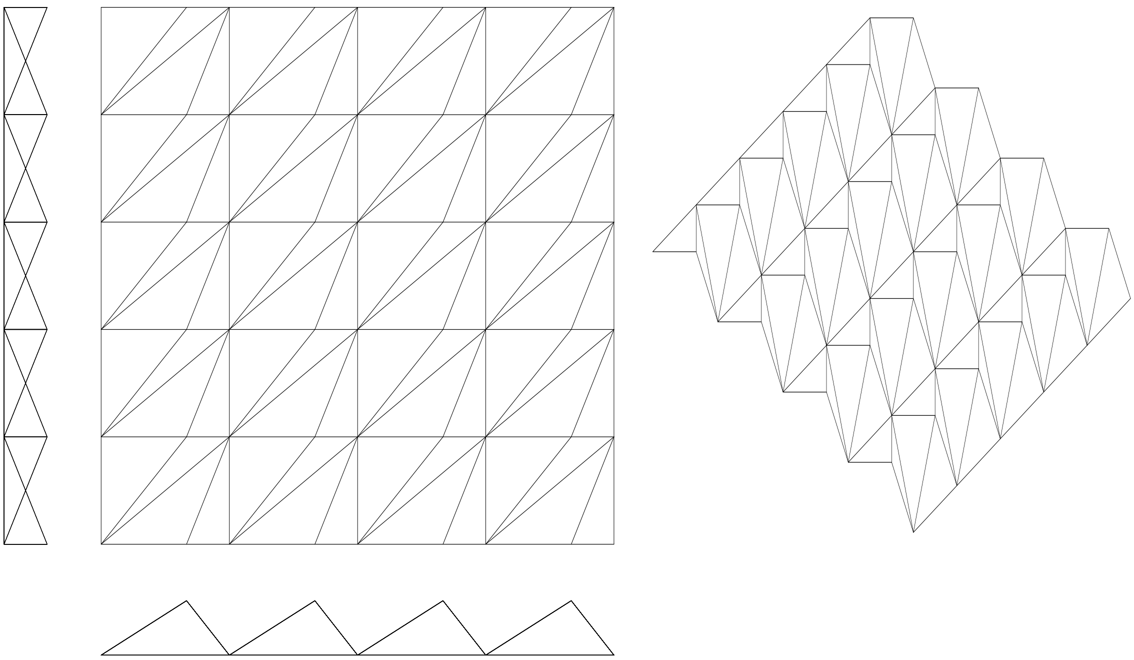
Слика 3.5.2. b) Пример примитива 2, раванске пројекције и изометрија

$A_x = 5$; $A_z = 2$; $A_{y1} = A_{y2} = 0,7$; $A_{nx} = 3$; $A_{nz} = 9$; $A_{x1} = A_{x3} = 0,75$; $A_{x2} = A_{x4} = 0,5$

### 3.5.3 Примитив типа P - 3



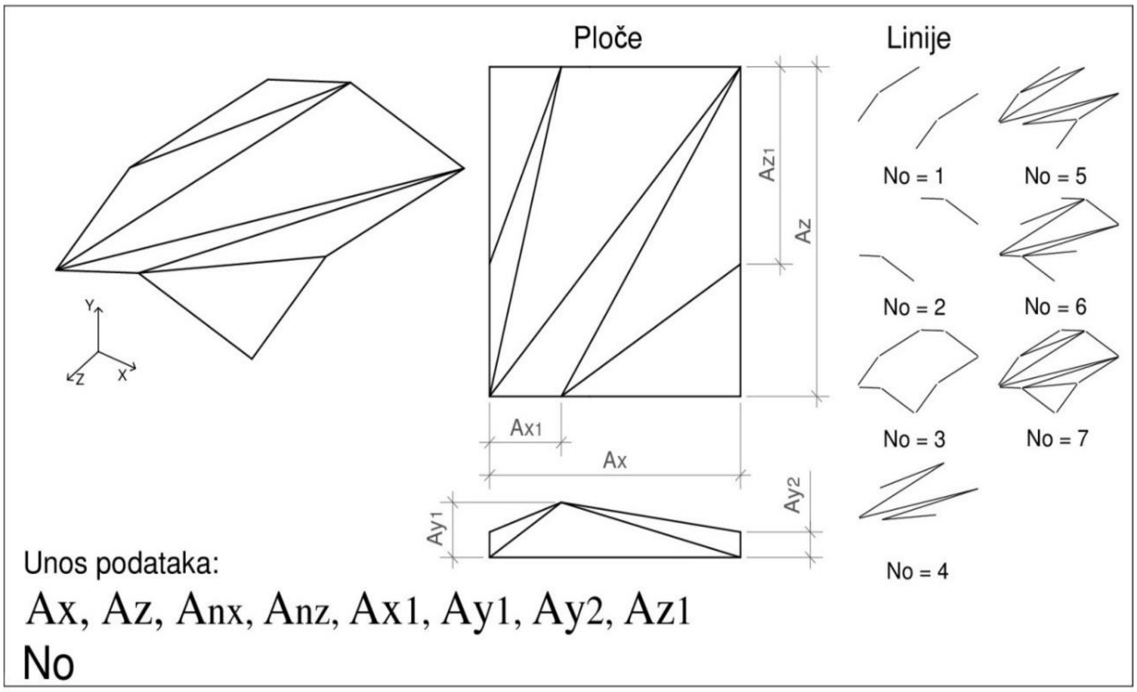

Слика 3.5.3. а) Параметри за генерисање примитива типа 3



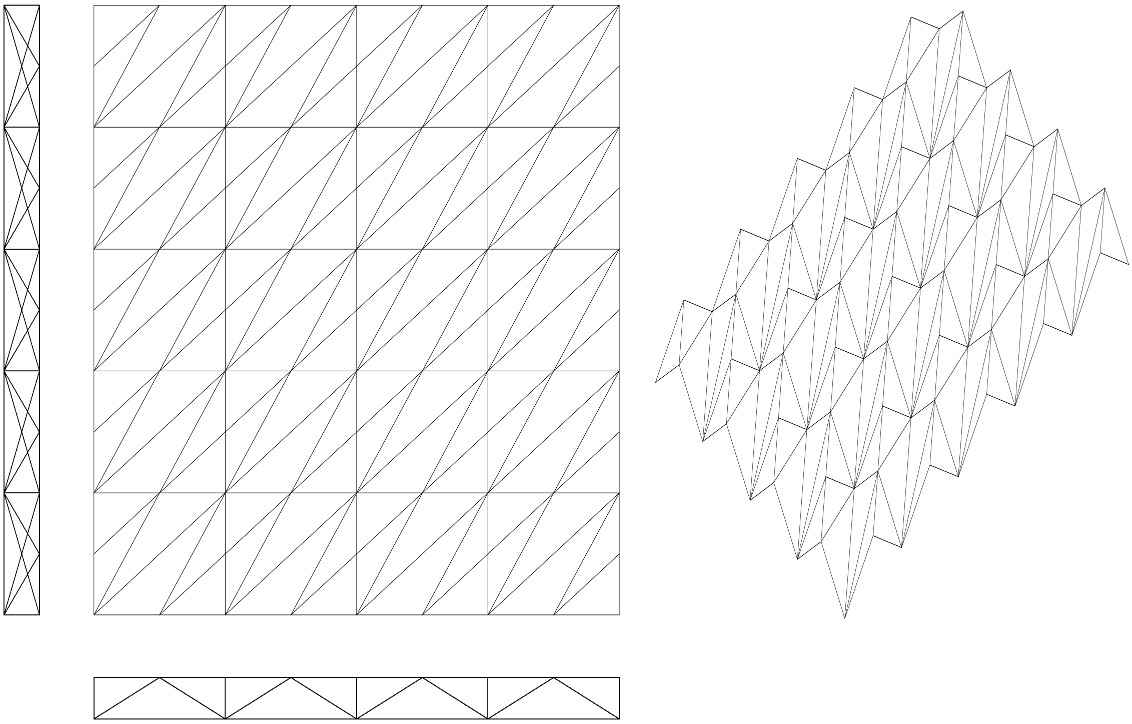
Слика 3.5.3. b) Пример примитива 3, раванске пројекције и изометрија

$A_x = 5$; $A_z = 3$; $A_{y1} = 2$; $A_{nx} = 4$; $A_{nz} = 5$; $A_{x1} = 4$;

### 3.5.4 Примитив типа P - 4



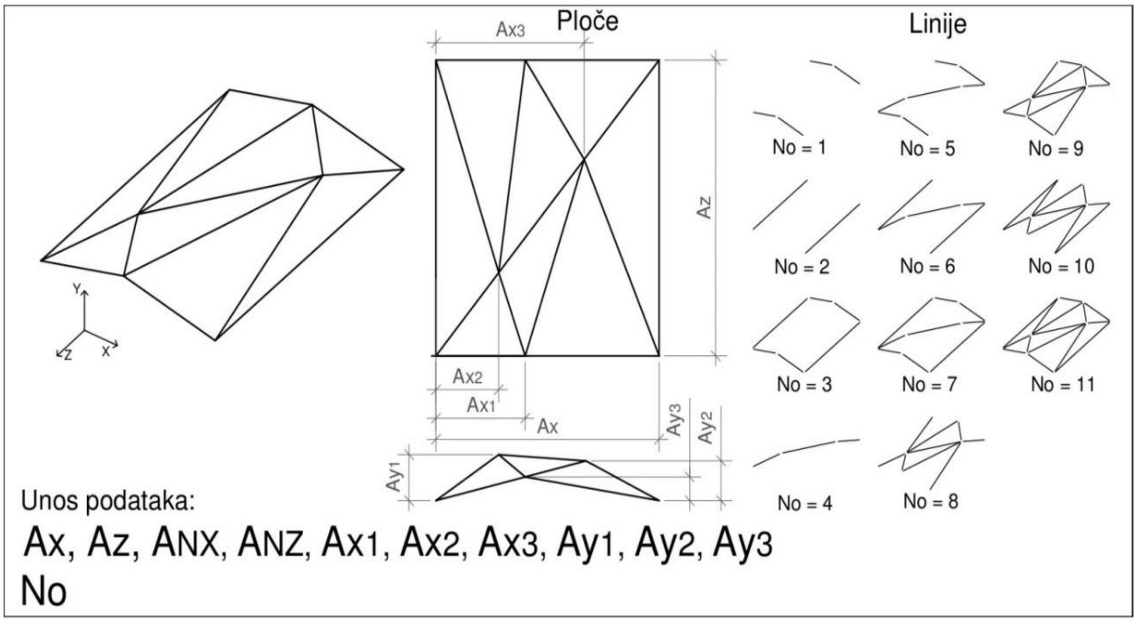

Слика 3.5.4. а) Параметри за генерисање примитива типа 4



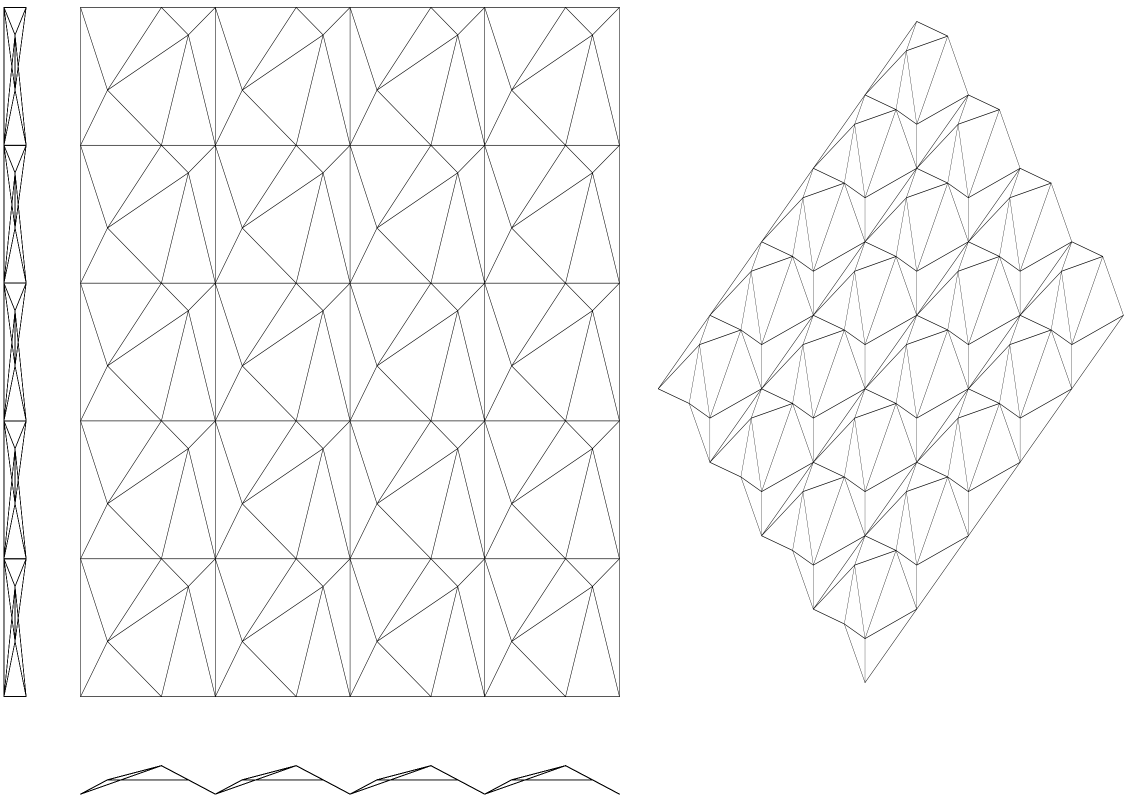
Слика 3.5.4. b) Пример примитива 4, раванске пројекције и изометрија

$A_x = 5$; $A_z = 4$; $A_{y1} = Ay_2 = 2$; $A_{nx} = 4$; $A_{nz} = 5$; $A_{x1} = 2{,}5$; $A_{z1} = 2{,}5$

### 3.5.5 Примитив типа Р - 5



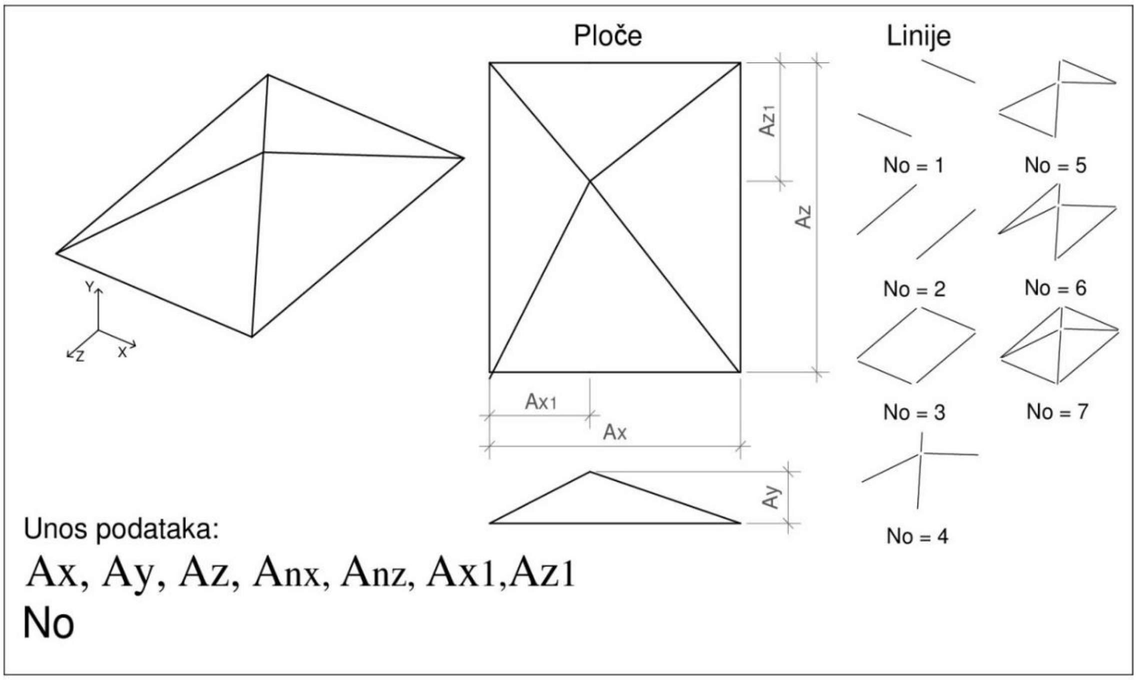

Слика 3.5.5. а) Параметри за генерисање примитива типа 5



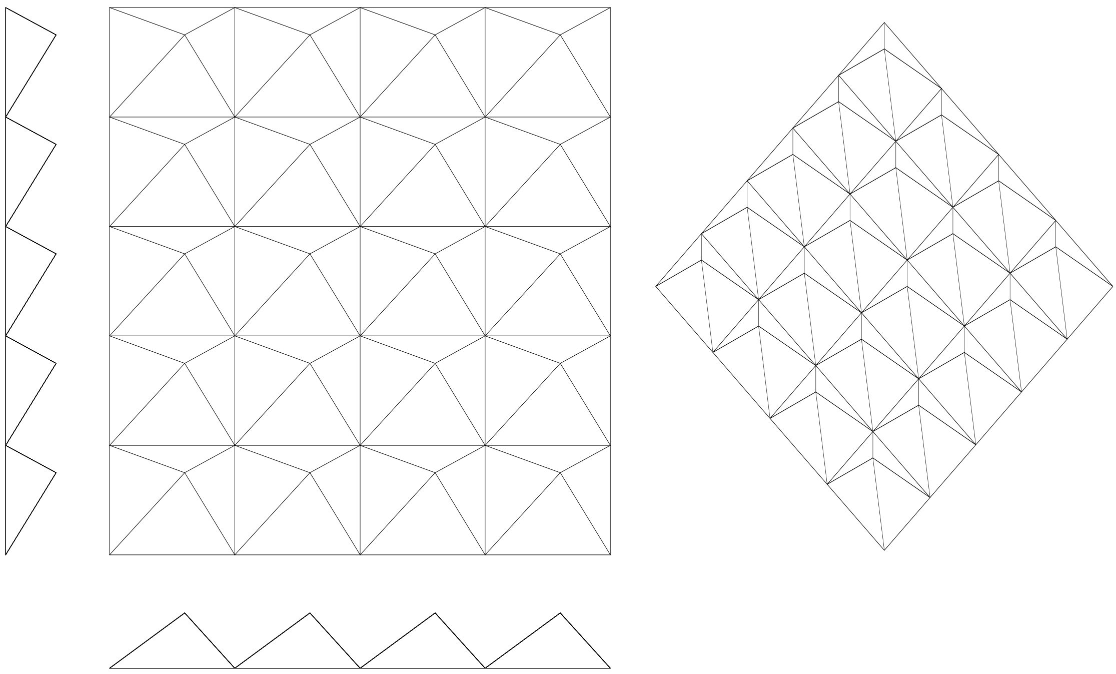
Слика 3.5.5. b) Пример примитива 5, раванске пројекције и изометрија

$A_x = 5$; $A_z = 4$; $A_{y1} = A_{y3}$ 1; $A^{y2} = 2$; $A_{nx} = 4$; $A_{nz} = 5$; $A_{x1} =3$; $A_{x2} = 1$; $A_{x3} = 4$

### 3.5.6 Примитив типа P - 6



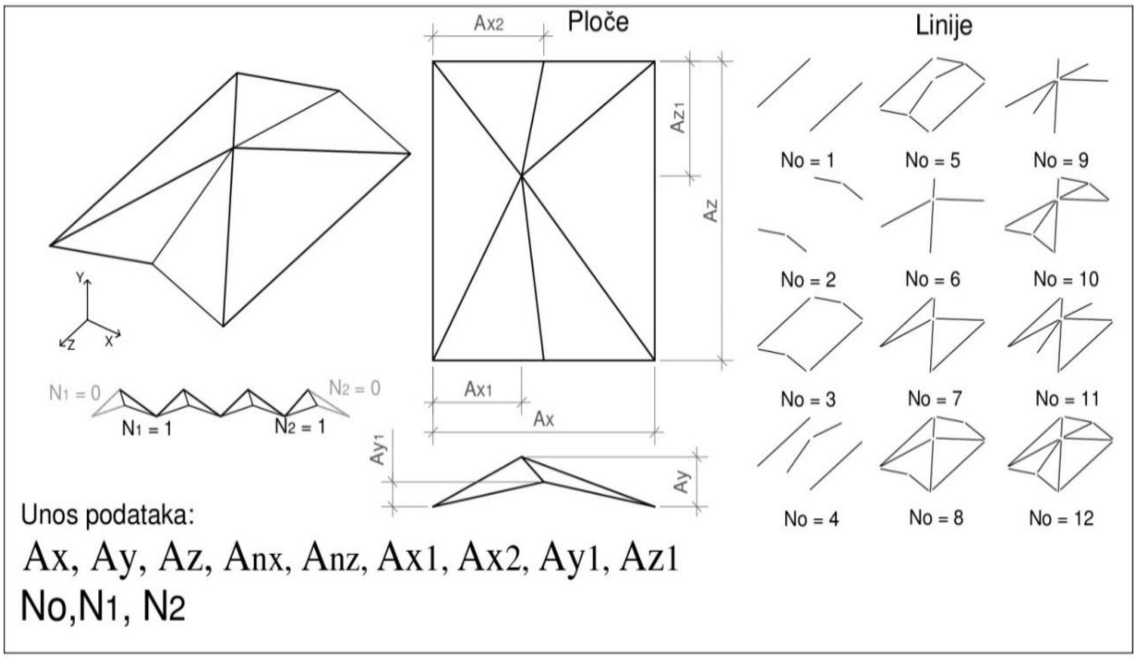

Слика 3.5.6. а) Параметри за генерисање примитива типа 6



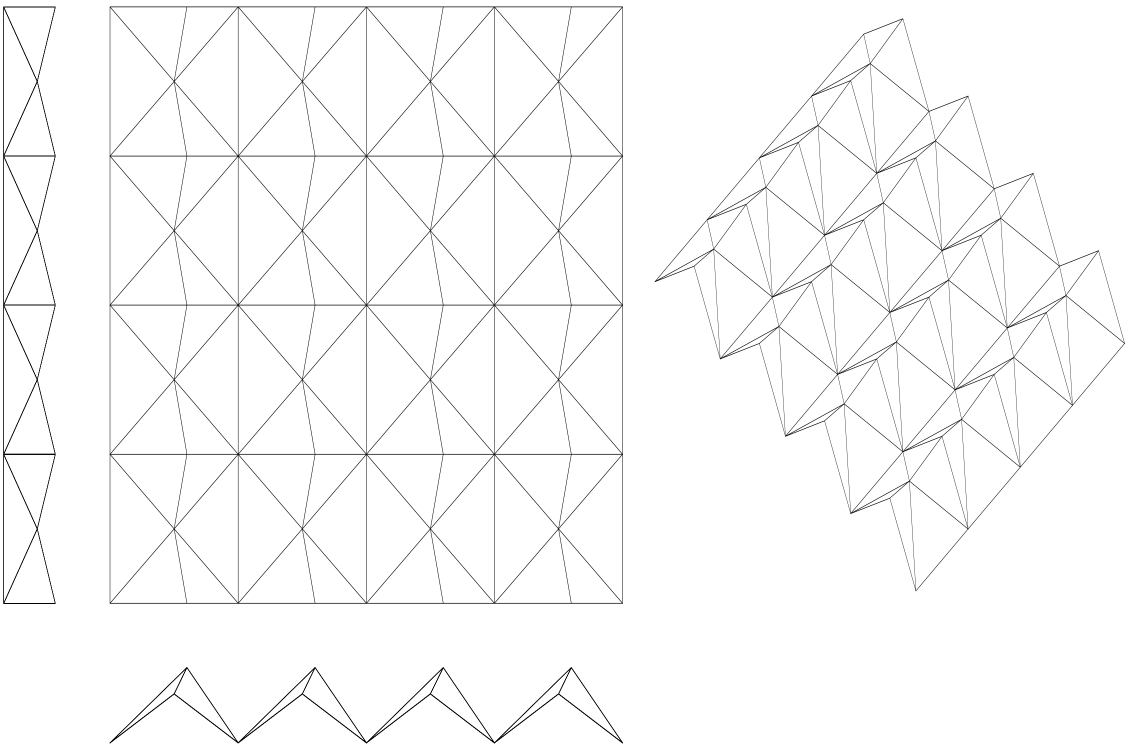
Слика 3.5.6. b) Пример примитива 6, раванске пројекције и изометрија

$A_x = 5$; $A_z = 4$; $A_y = 2$; $A_{nx} = 4$; $A_{nz} = 5$; $A_{x1} = 3$; $A_{z1} = 4$

### 3.5.7 Примитив типа P - 7



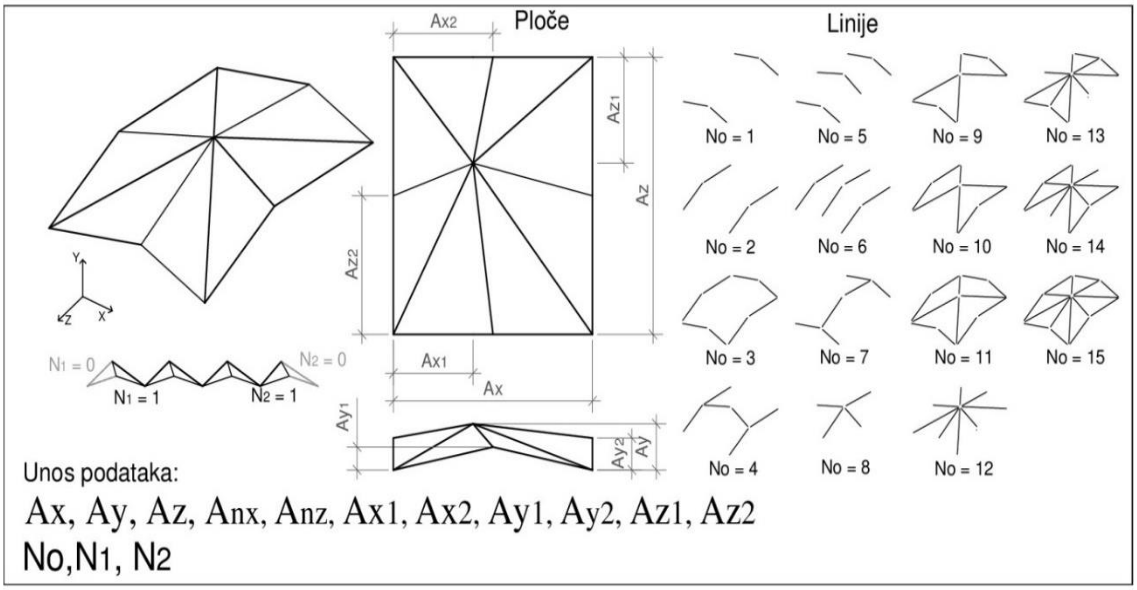

Слика 3.5.7. а) Параметри за генерисање примитива типа 7



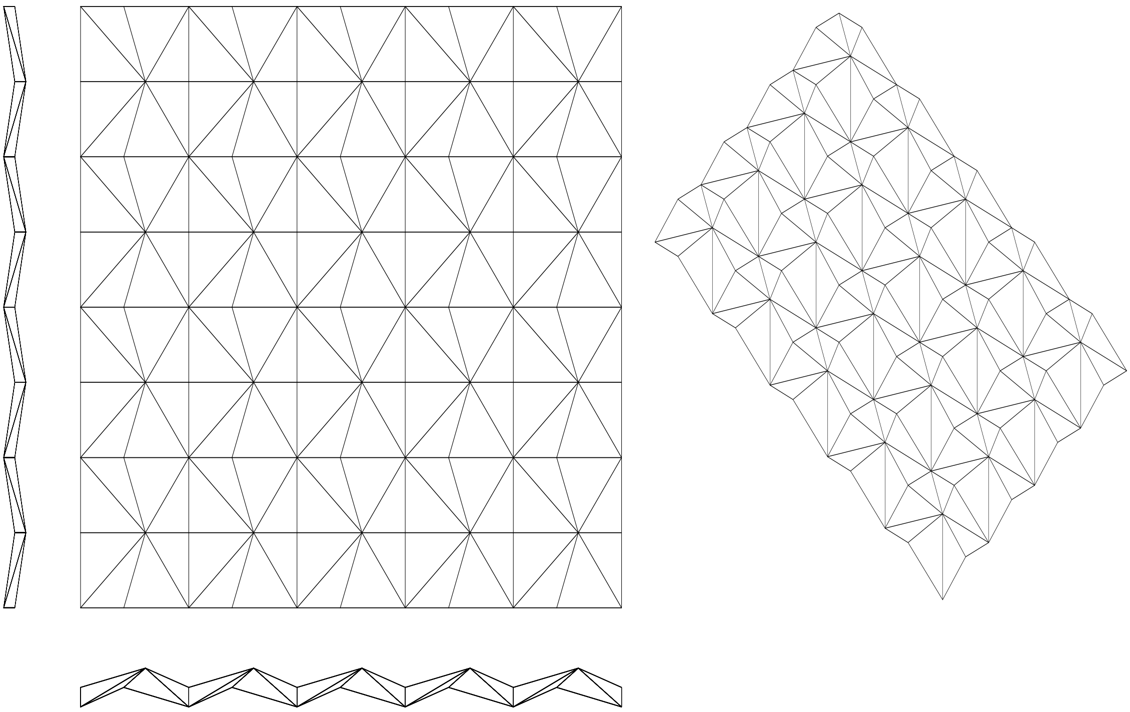
Слика 3.5.7.b) Пример примитива 7, раванске пројекције и изометрија

$A_x = 3$; $A_z = 5$; $A_y = 1{,}5$; $A_{y1} = 2{,}5$; $A_{nx} = 4$; $A_{nz} = 4$; $A_{x1} = 1{,}5$ $A_{x2} = 2$; $A_{z1} = 2{,}5$

### 3.5.8 Примитив типа Р - 8




Слика 3.5.8. а) Параметри за генерисање примитива типа 8

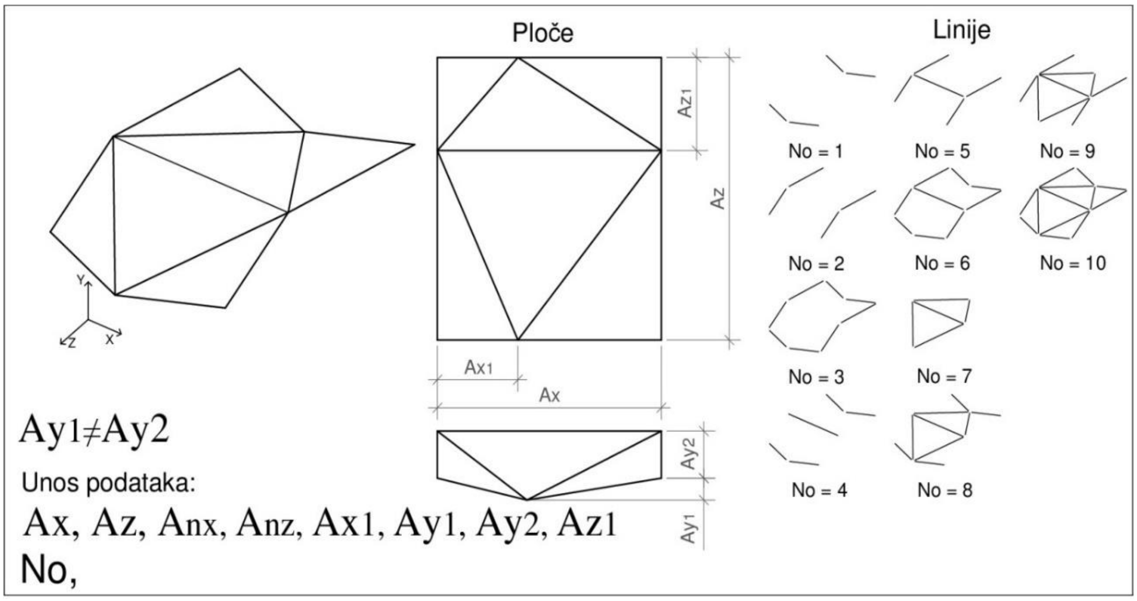


Слика 3.5.8. b) Пример примитива 8, раванске пројекције и изометрија

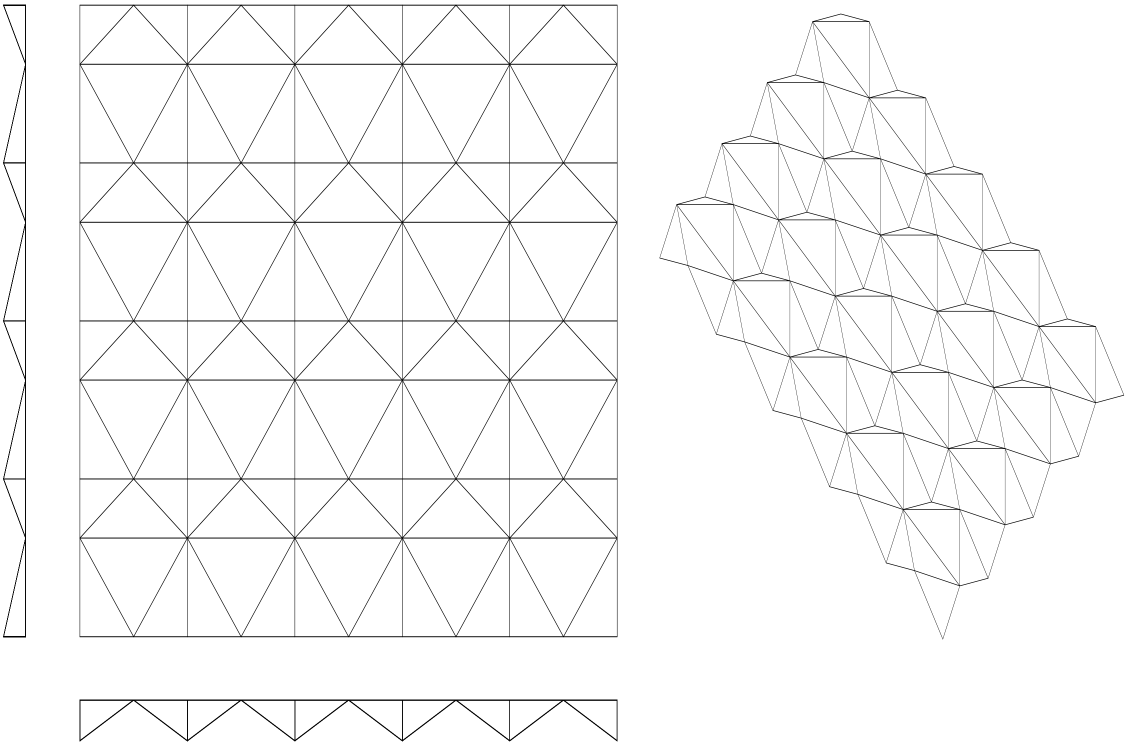
$A_x = 4$; $A_z = 5$; $A_{y1} = 2$; $A_{y1} = Ay_2 = 1$; $A_{nx} = 5$; $A_{nz} = 4$; $A_{x1} = 3$; $A_{x2} = 2$;

$A_{z1} = A_{z2} = 2{,}5$

### 3.5.9 Примитив типа P - 9



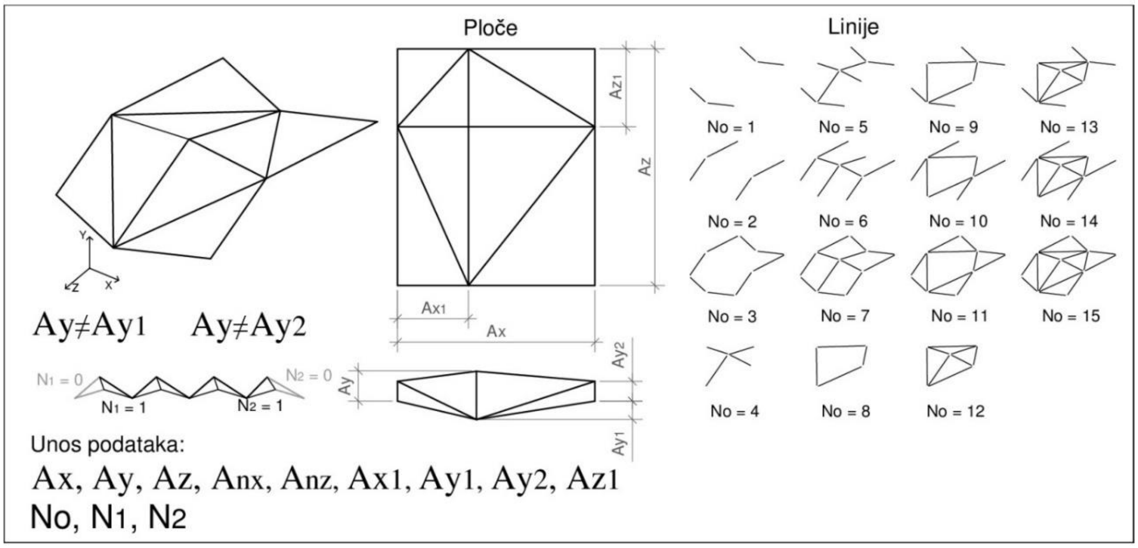

Слика 3.5.9. а) Параметри за генерисање примитива типа 9



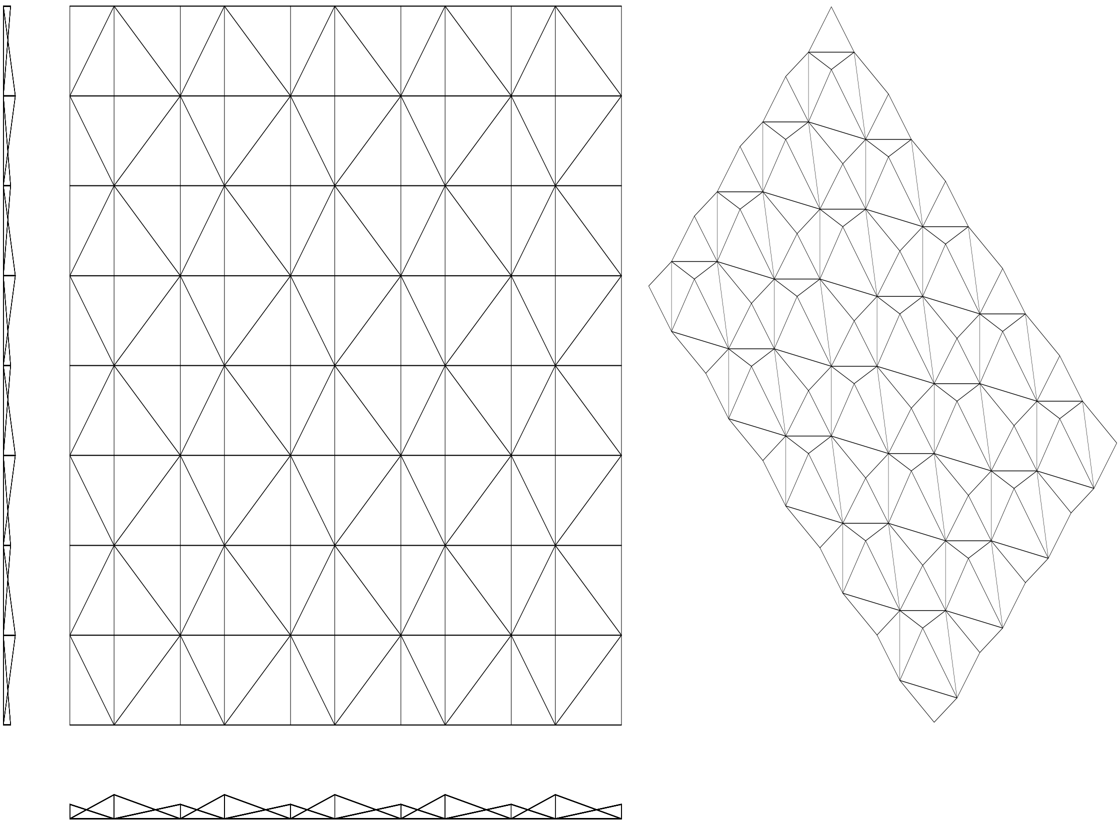
Слика 3.5.9. b) Пример примитива 9, раванске пројекције и изометрија

$A_x = 4$; $A_z = 5$; $A_{y1} = A_{y2} = 2$; $A_{nx} = 5$; $A_{nz} = 4$; $A_{x1} = 2$; $A_{z1} = 3$

## 3.5.10 Примитив типа Р - 10



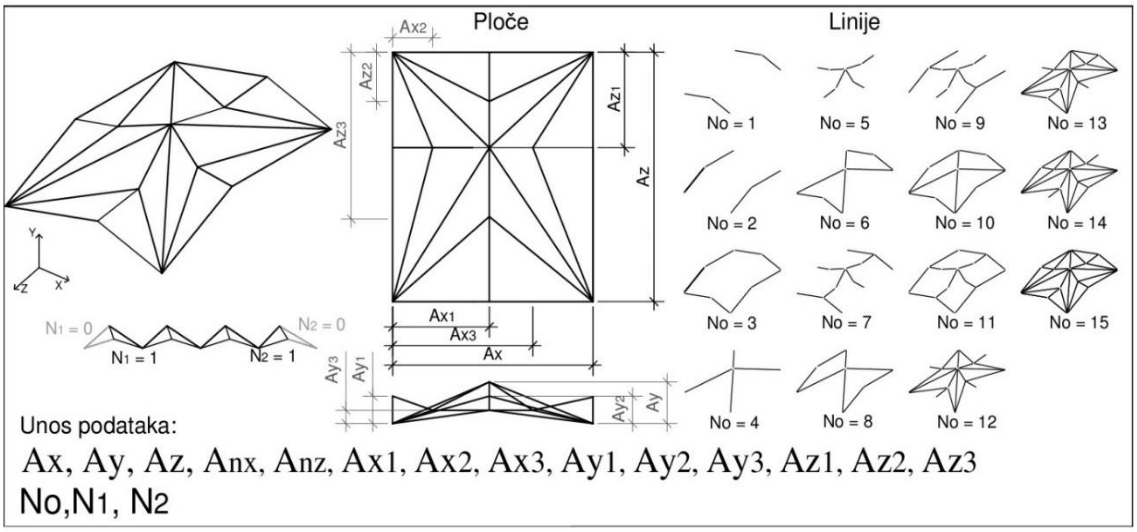

Слика 3.5.10. а) Параметри за генерисање примитива типа 10



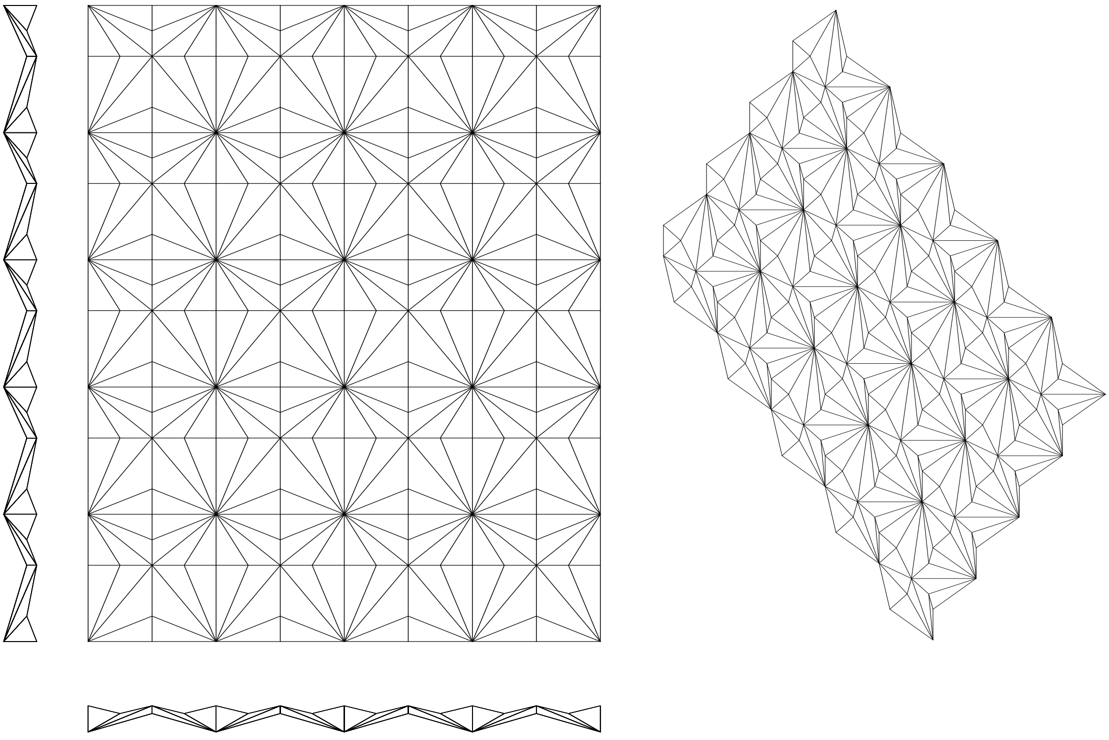
Слика 3.5.10. b) Пример примитива 10, раванске пројекције и изометрија

$A_x = 4$; $A_z = 5$; $A_y = 2$; $A_{y1} = Ay_2 = 0$; $A_{nx} = 4$; $A_{nz} = 5$; $A_{x1} = 1,5$; $A_{z1} = 2,5$

### 3.5.11 Примитив типа P - 11



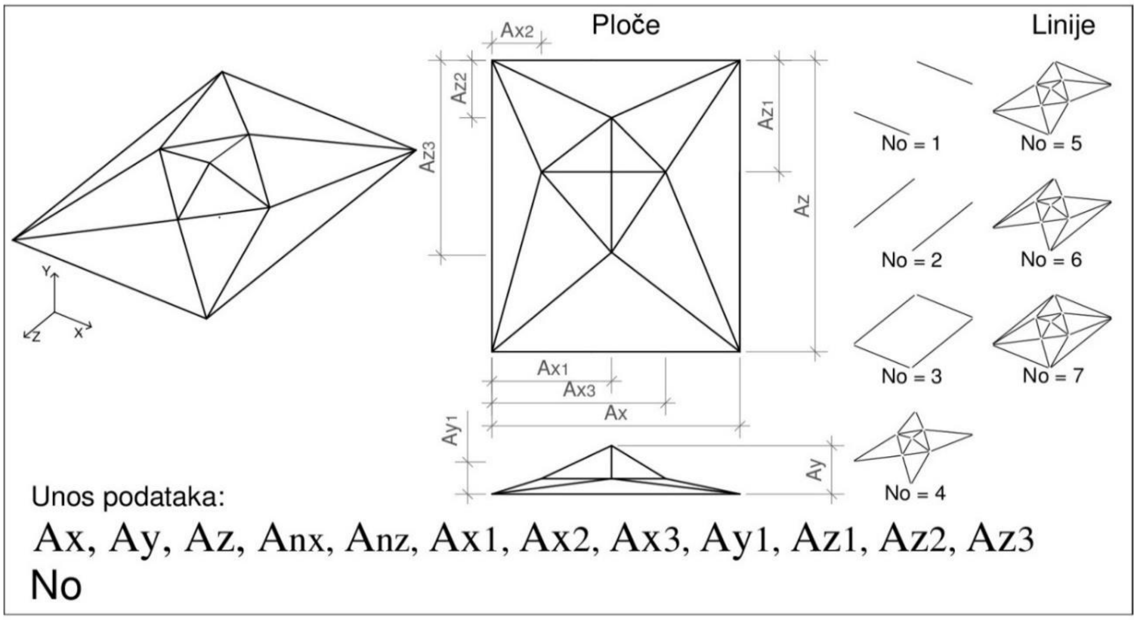

Слика 3.5.11. b) Параметри за генерисање примитива типа 11



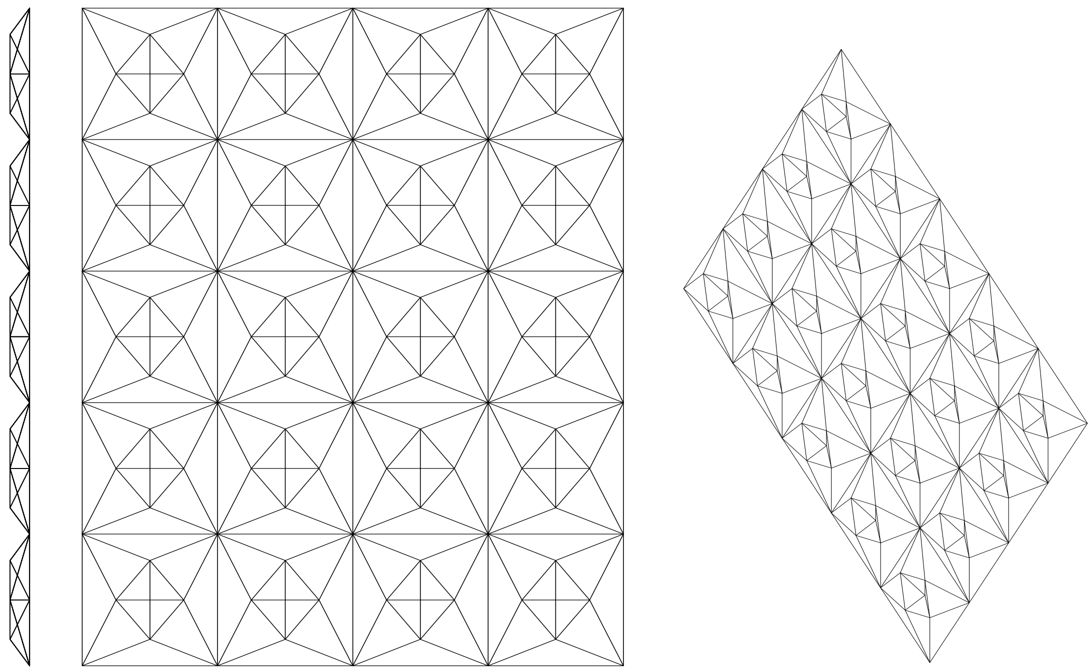
Слика 3.5.11. а) Пример примитива 11, раванске пројекције и изометрија

$A_x = 5$; $A_z = 5$; $A_y = 1$; $A_{y1} = 1$; $A_{y2} = A_{y3}=0,5$; $A_{nx} = 4$; $A_{nz} = 5$; $A_{x1} = 2,5$; $A_{x2} = 1,25$; $A_{x3} = 3,75$; $A_{z1} = 2$; $A_{z2}=1$; $A_{z3}=4$

### 3.5.12 Примитив типа P - 12

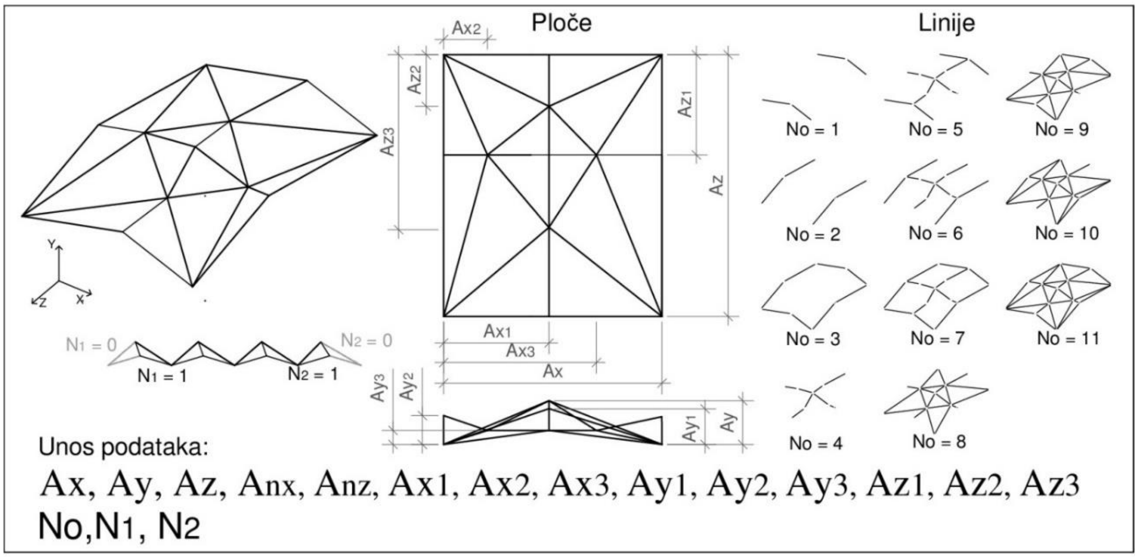



Слика 3.5.12. а) Параметри за генерисање примитива типа 12

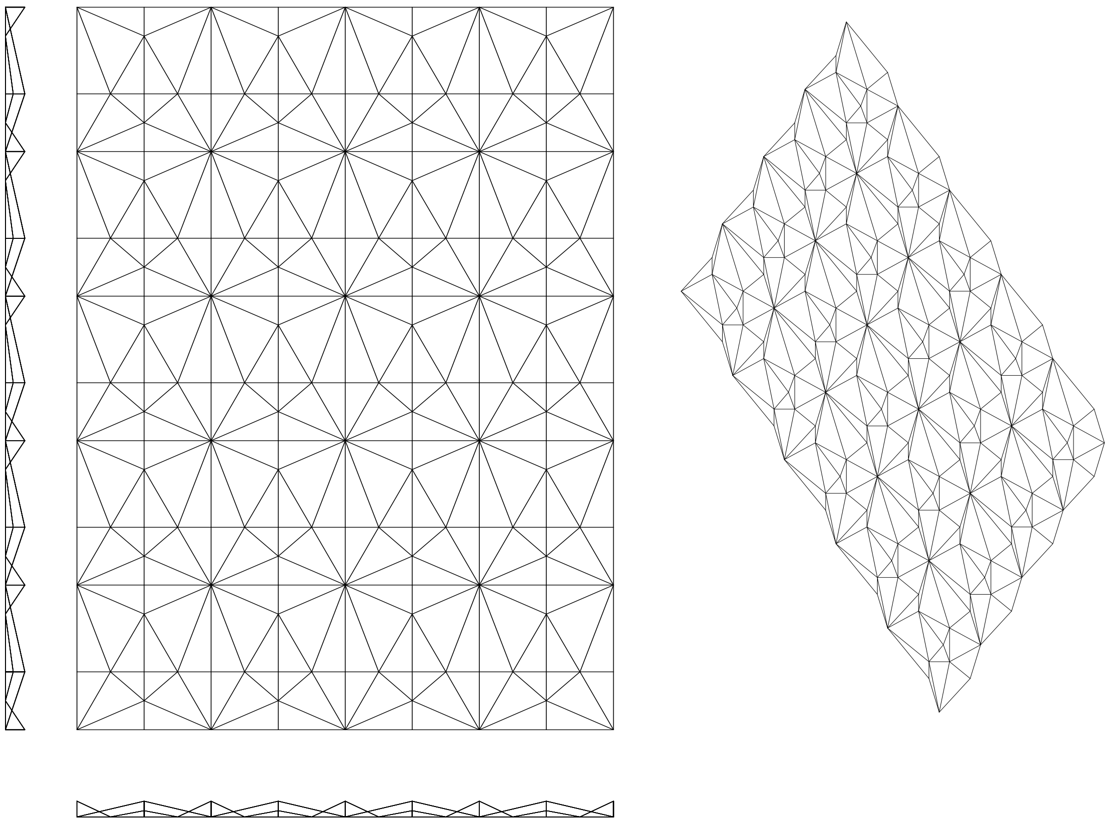


Слика 3.5.12. b) Пример примитива 12, раванске пројекције и изометрија$A_x$ = 5; $A_z$ = 5; $A_y$ = 0; $A_{y1}$ = 1; $A_{nx}$ = 4; $A_{nz}$ = 5; $A_{x1}$ = 2,5; $A_{x2}$ = 1,25; $A_{x3}$ = 3,75; $A_{z1}$ = 2,5; $A_{z2}$=1,25; $A_{z3}$=3,75

### 3.5.13 Примитив типа P - 13




Слика 3.5.13. a) Параметри за генерисање примитива типа 13

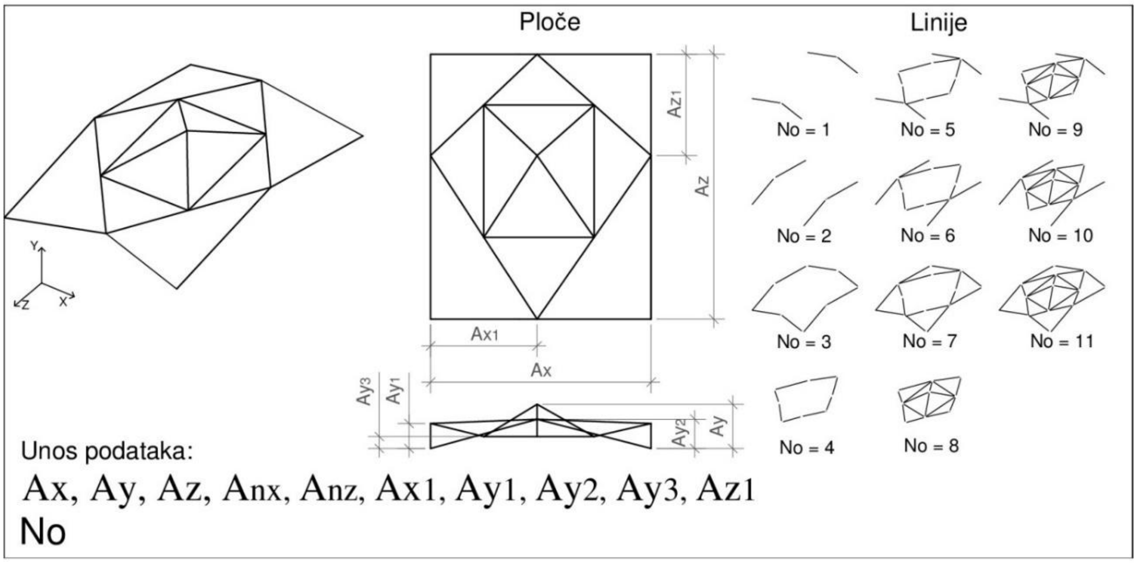


Слика 3.5.13. b) Пример примитива 13, раванске пројекције и изометрија

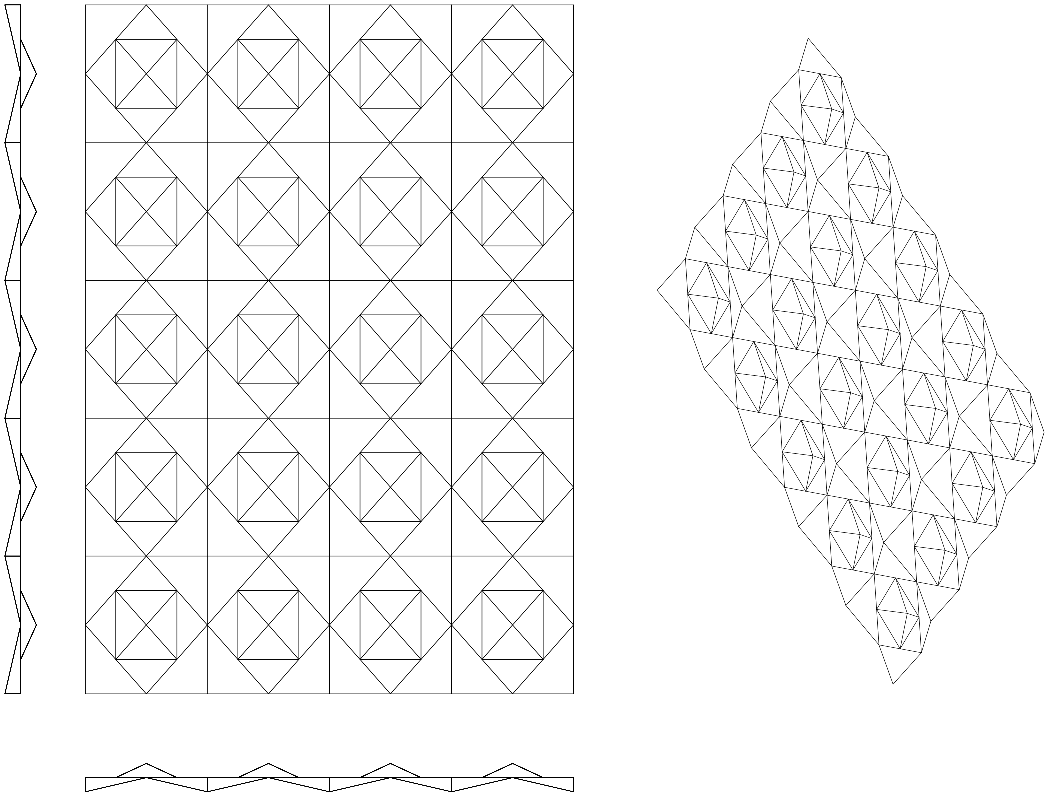
$A_x = 5$; $A_z = 5$; $A_y = 1$; $A_{y1} = 0,5$; $A_{y2} = A_{y3} = 0,5$; $A_{nx} = 4$; $A_{nz} = 5$; $A_{x1} = 2,5$; $A_{x2} = 1,25$;

$A_{x3} = 3,75$; $A_{z1} = 3$; $A_{z2}=1$; $A_{z3}=4$

### 3.5.14 Примитив типа P - 14



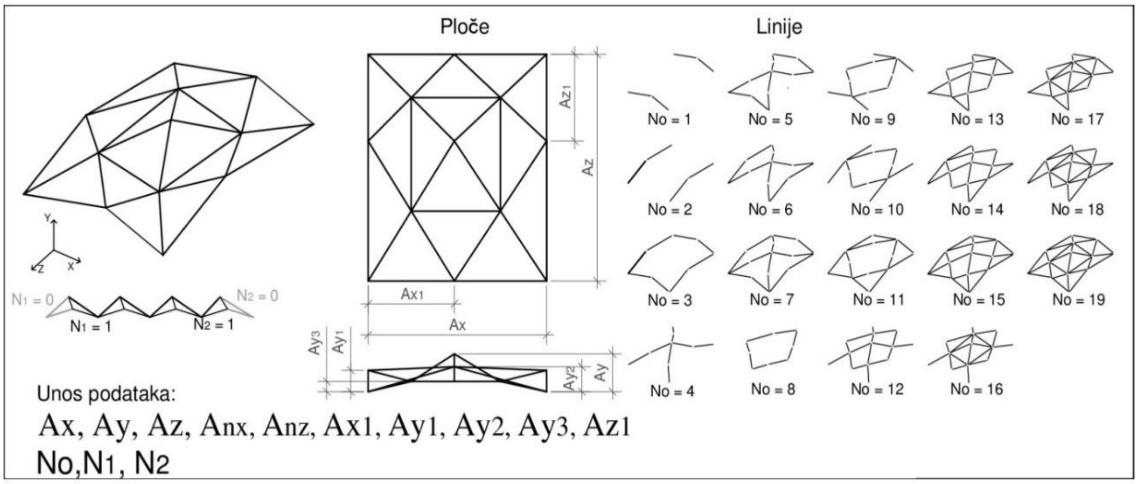

Слика 3.5.14. а) Параметри за генерисање примитива типа 14



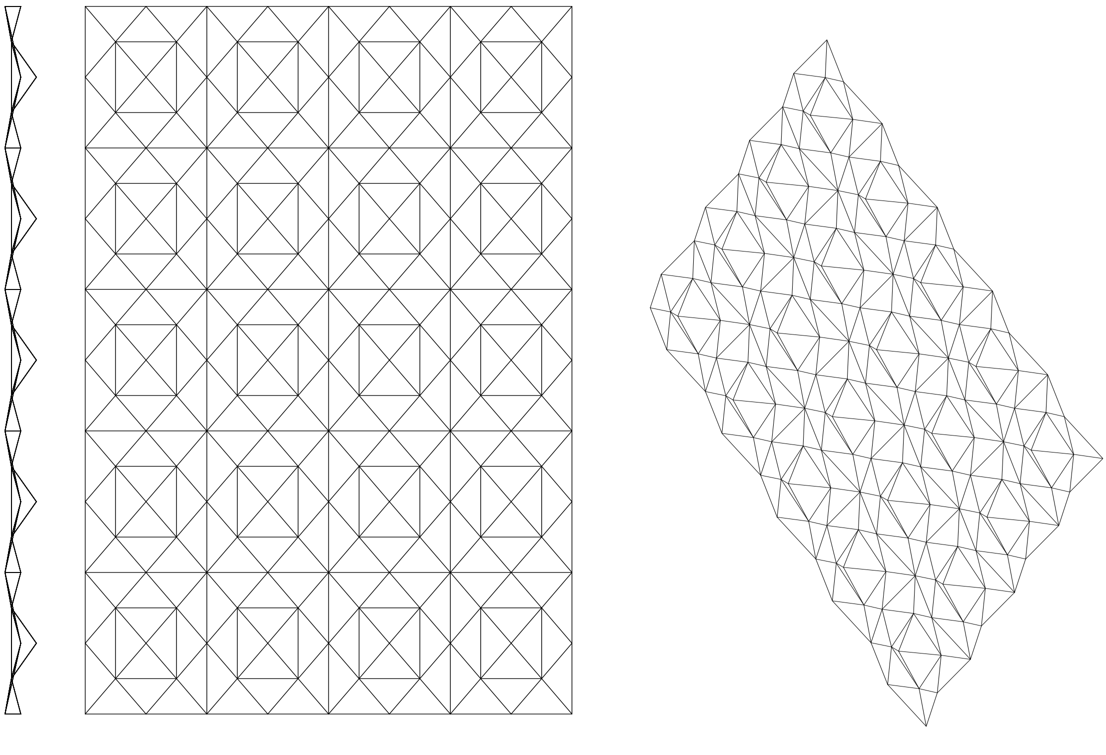
Слика 3.5.14. b) Пример примитива 14, раванске пројекције и изометрија

$A_x = 5$; $A_z = 5$; $A_y = 2$; $A_{y1} = A_{y2} = A_{y3} = 1$; $A_{nx} = 4$; $A_{nz} = 5$; $A_{x1} = 2,5$; $A_{z1} = 2,5$;

### 3.5.15 Примитив типа P - 15




Слика 3.5.15. а) Параметри за генерисање примитива типа 15



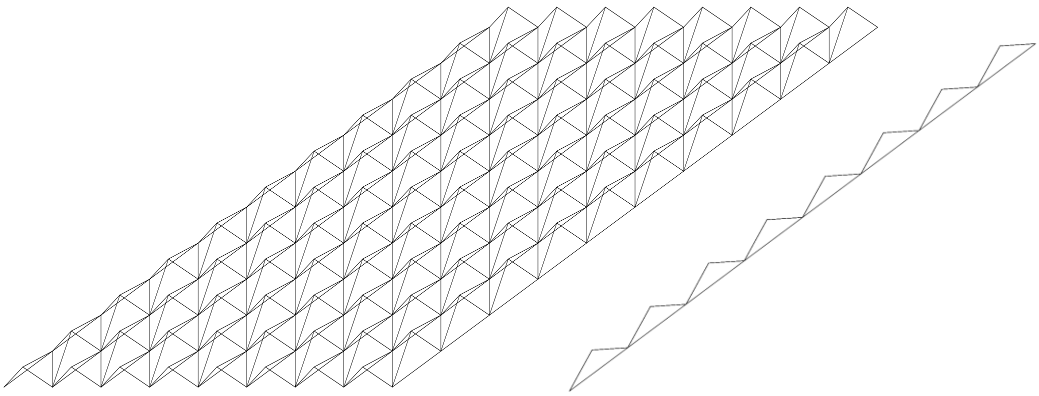
Слика 3.5.15. b) Пример примитива 15, раванске пројекције и изометрија

$A_x = 5$; $A_z = 5$; $A_y = 2$; $A_{y1} = A_{y2} = A_{y3}=1$; $A_{nx} = 4$; $A_{nz} = 5$; $A_{x1} = 2,5$; $A_{z1} = 2,5$

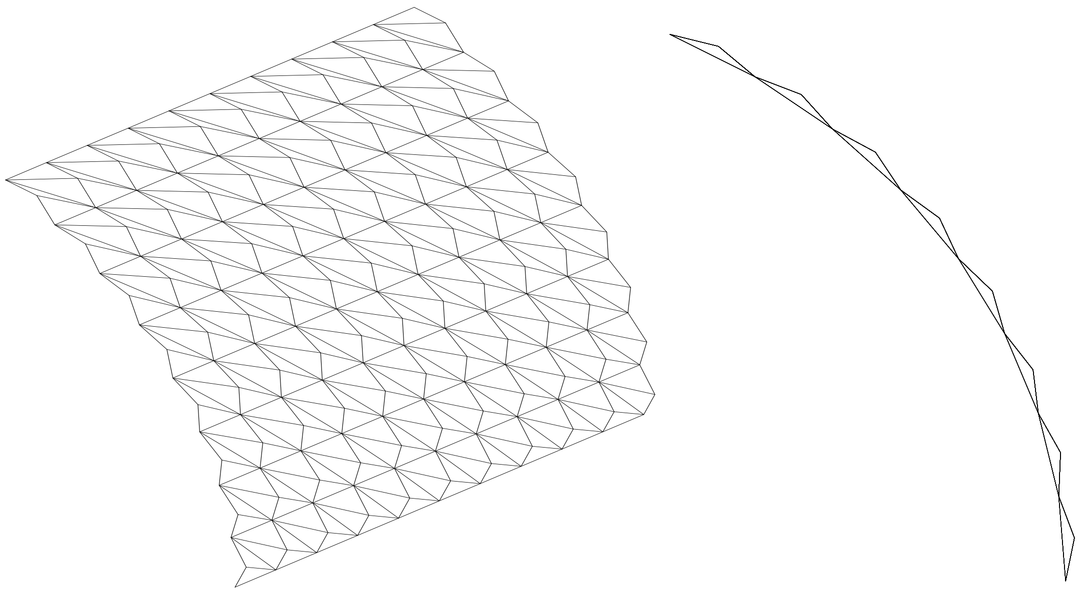
### 3.5.16 Ротација форме набора и ротационо умножавање примитива по контури круга

Набор добијен транслацијом примитива, као што је приказано преко претходних 15 примера, уколико је потребно можемо ротирати за жељени угао. Ротација набора који лежи у равни је неопходна за генерисања сложених форми набора који не леже у равни.

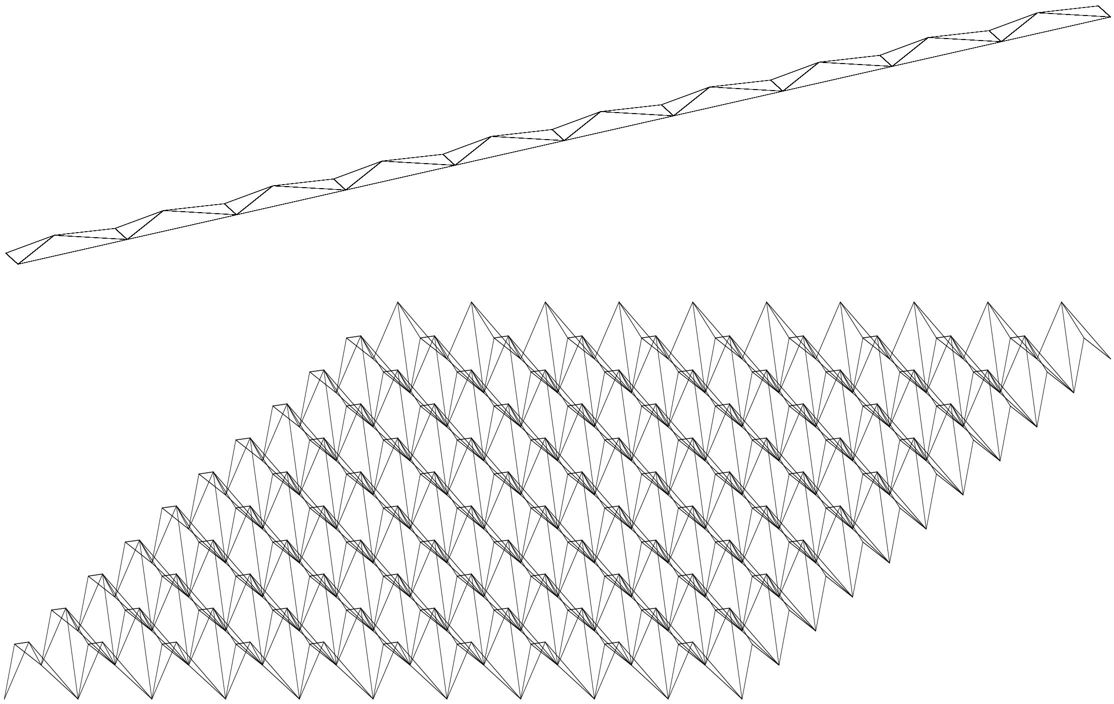
На исти начин као што смо умножавали примитиве транслацијом у равни, направљена је функција генератора која умножава примитиве по контури круга око осе y, а у правцу осе z се транслира. Примена ових функција је приказана преко следећих неколико примера.



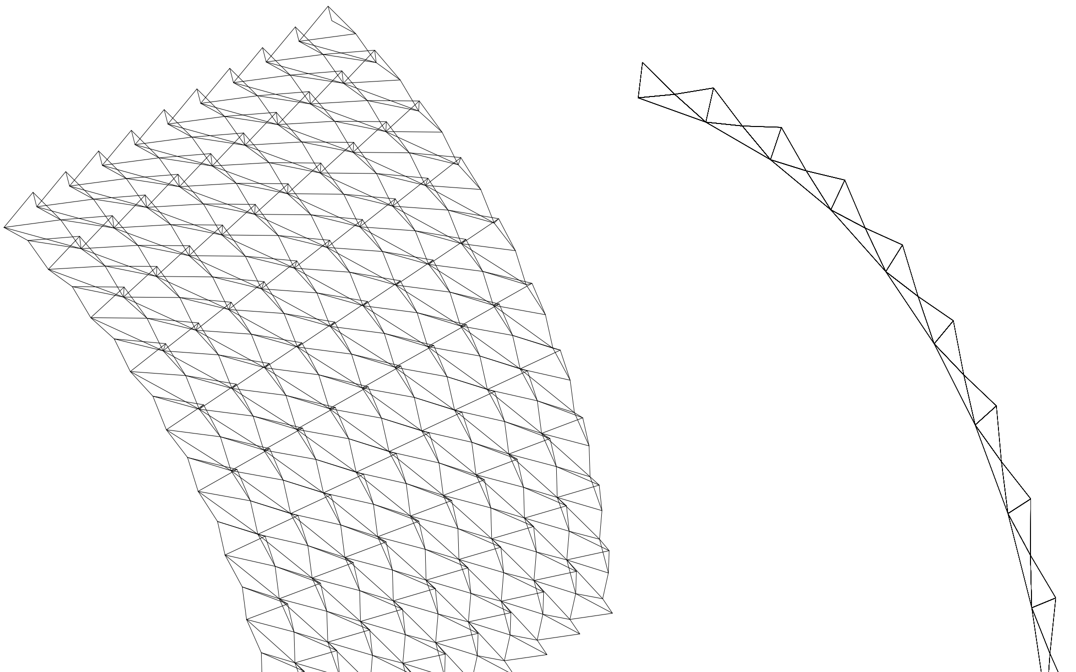
Слика 3.5.16.1. Ротација форме набора настала од примитива P - 3



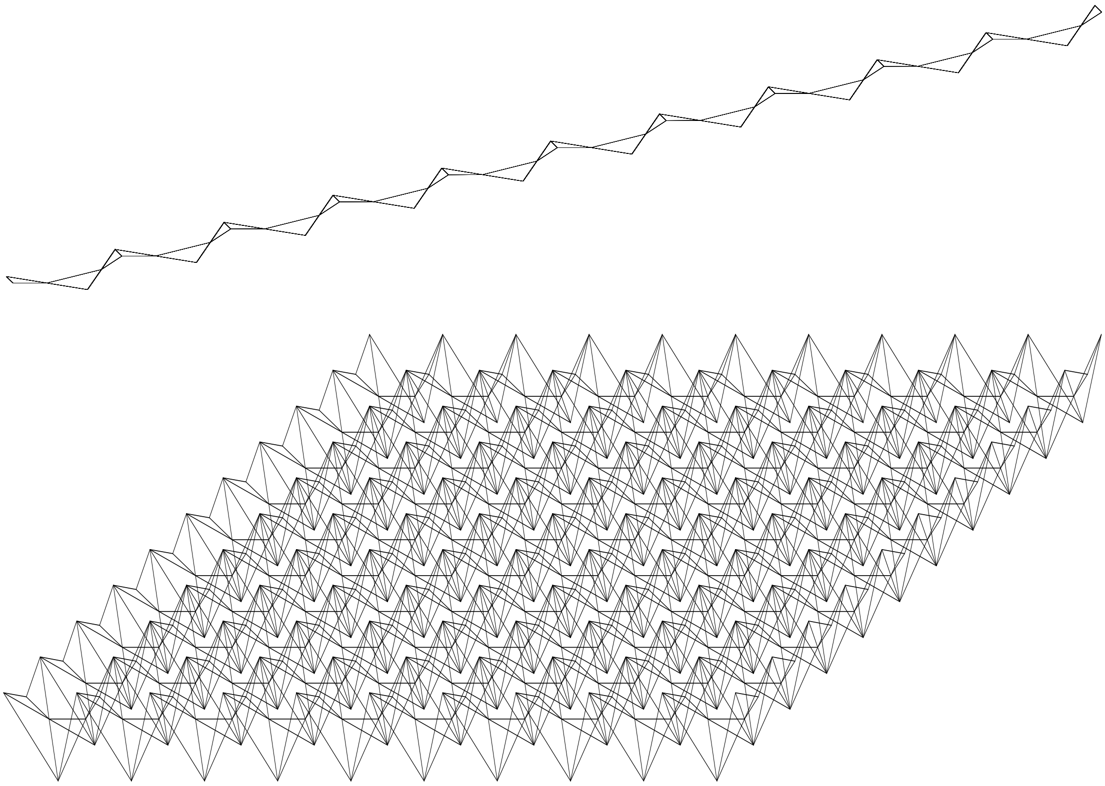
Слика 3.5.16.2. Ротација примитива P – 3 на део круга



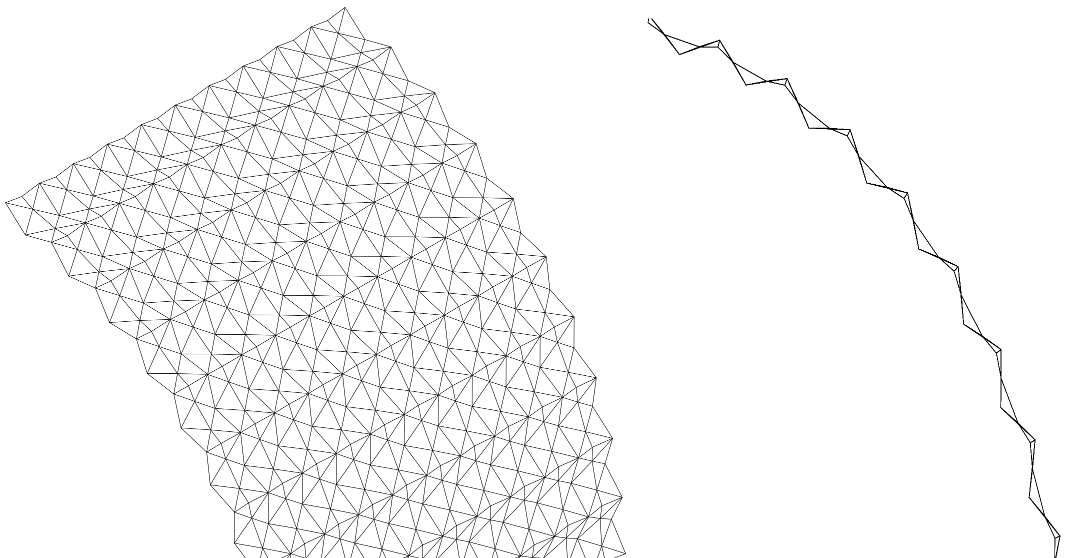
Слика 3.5.16.3. Ротација форме набора настала од примитива P - 4



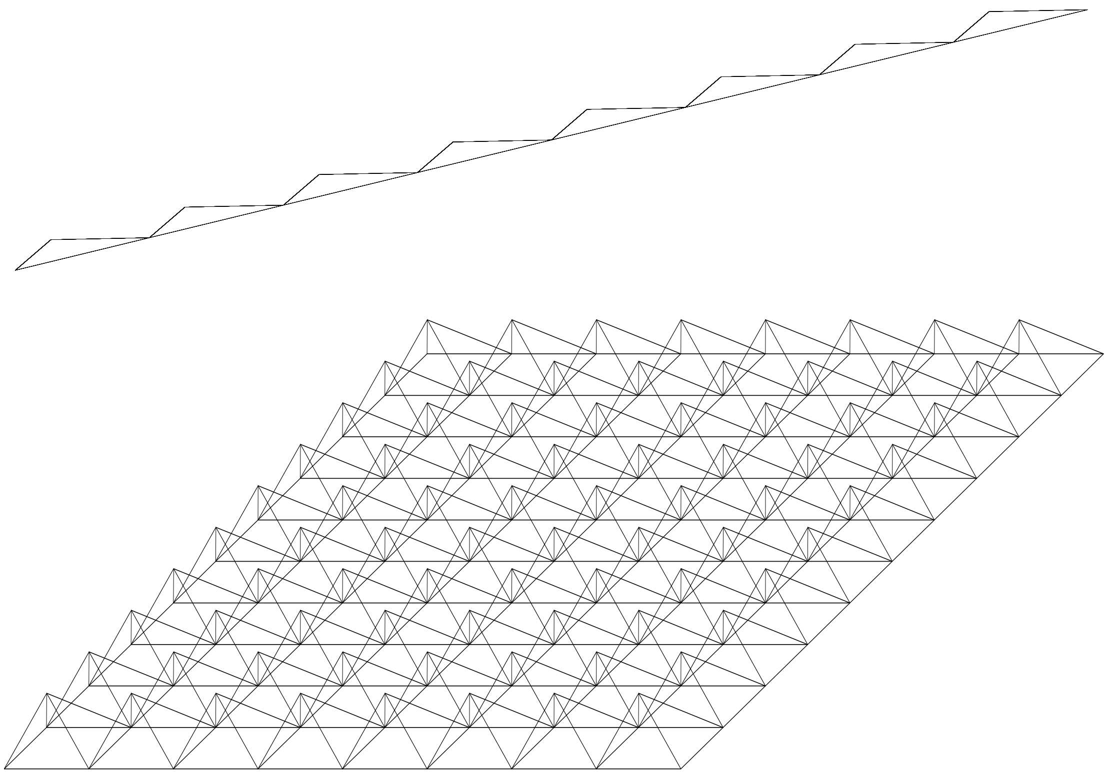
Слика 3.5.16.4. Ротација примитива P – 4 на део круга



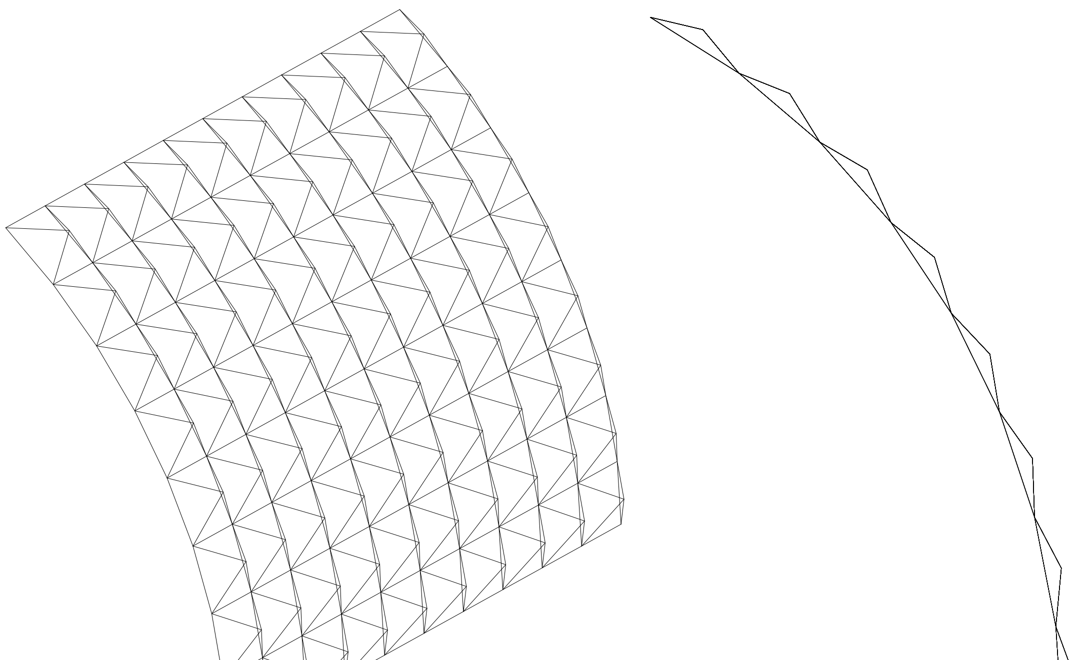
Слика 3.5.16.5. Ротација форме набора настала од примитива P - 5



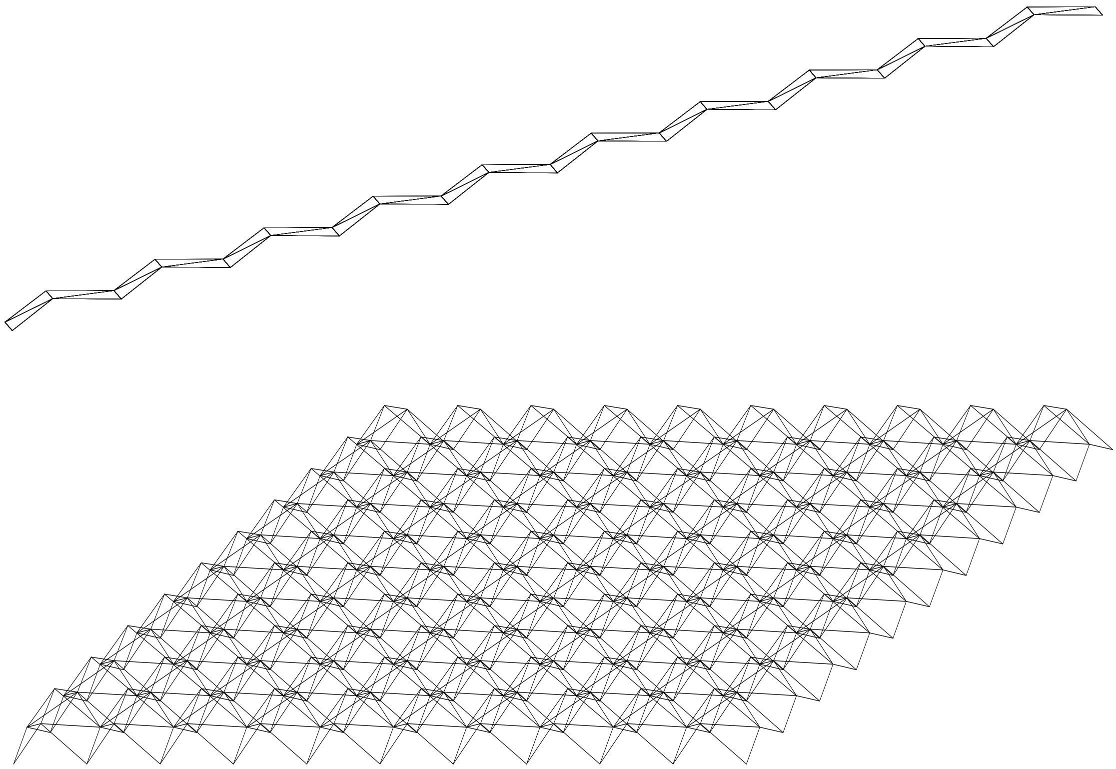
Слика 3.5.16.6. Ротација примитива P – 5 на део круга



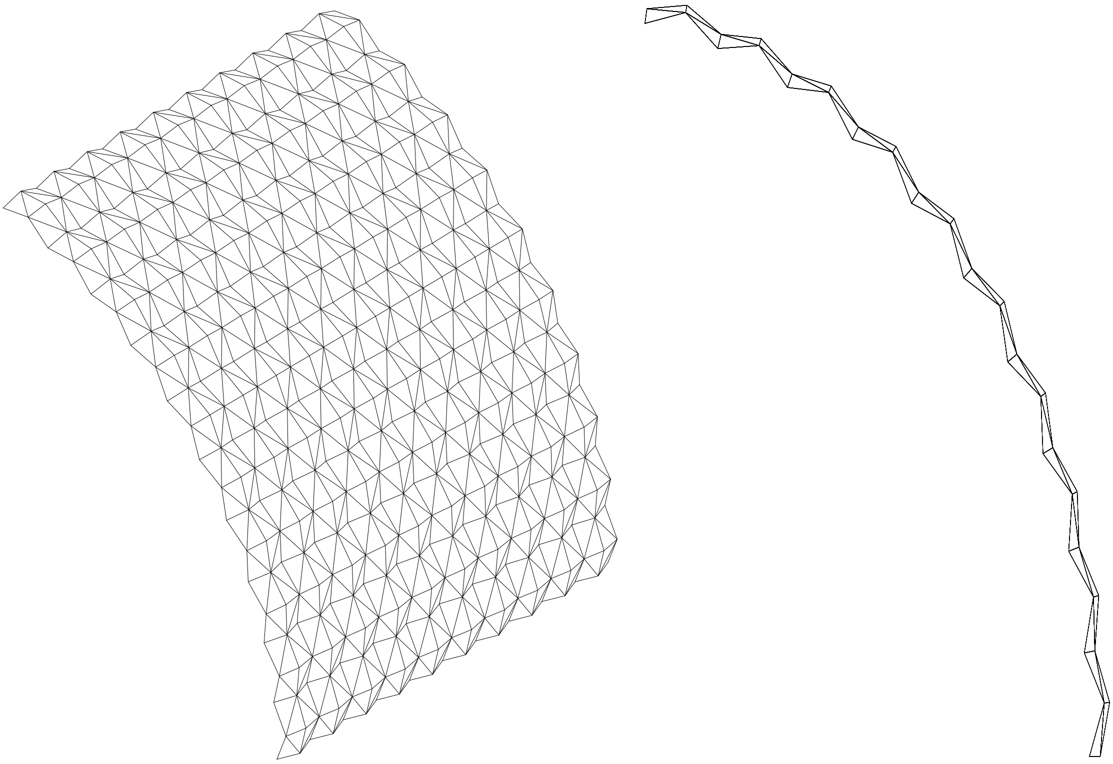
Слика 3.5.16.7. Ротација форме набора настала од примитива P - 6



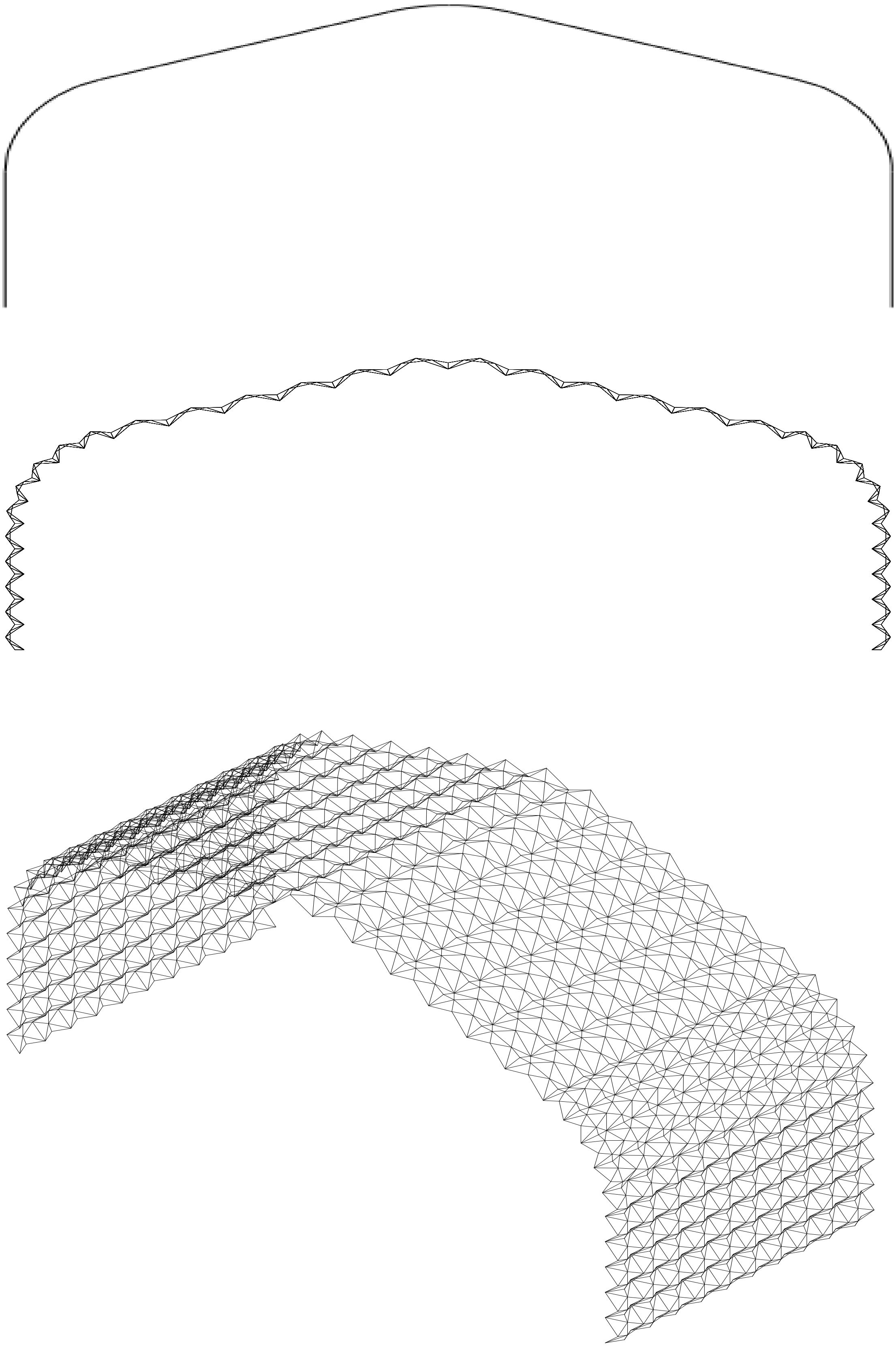
Слика 3.5.16.8. Ротација примитива P – 6 на део круга



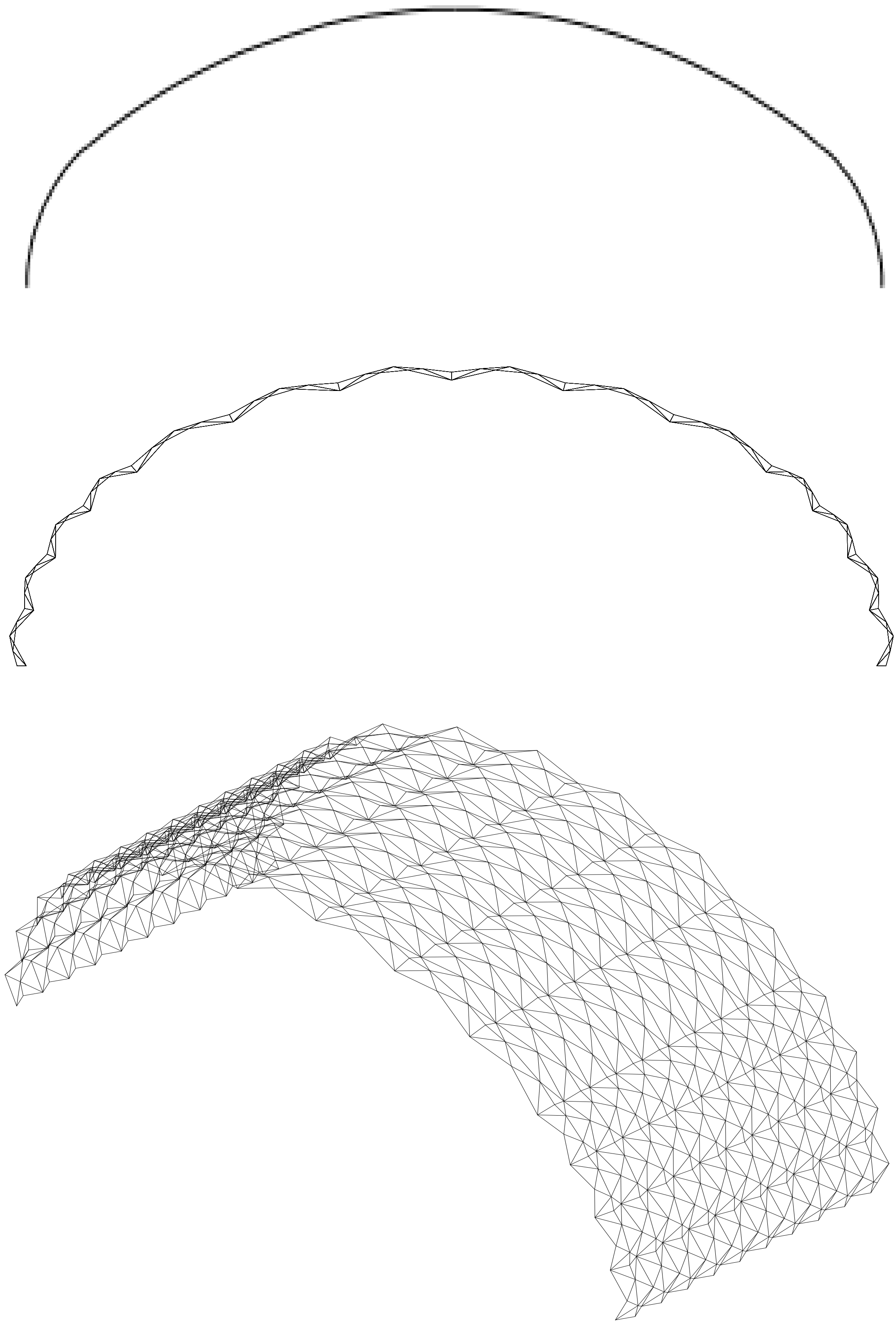
Слика 3.5.16.9. Ротација форме набора настала од примитива Р - 10



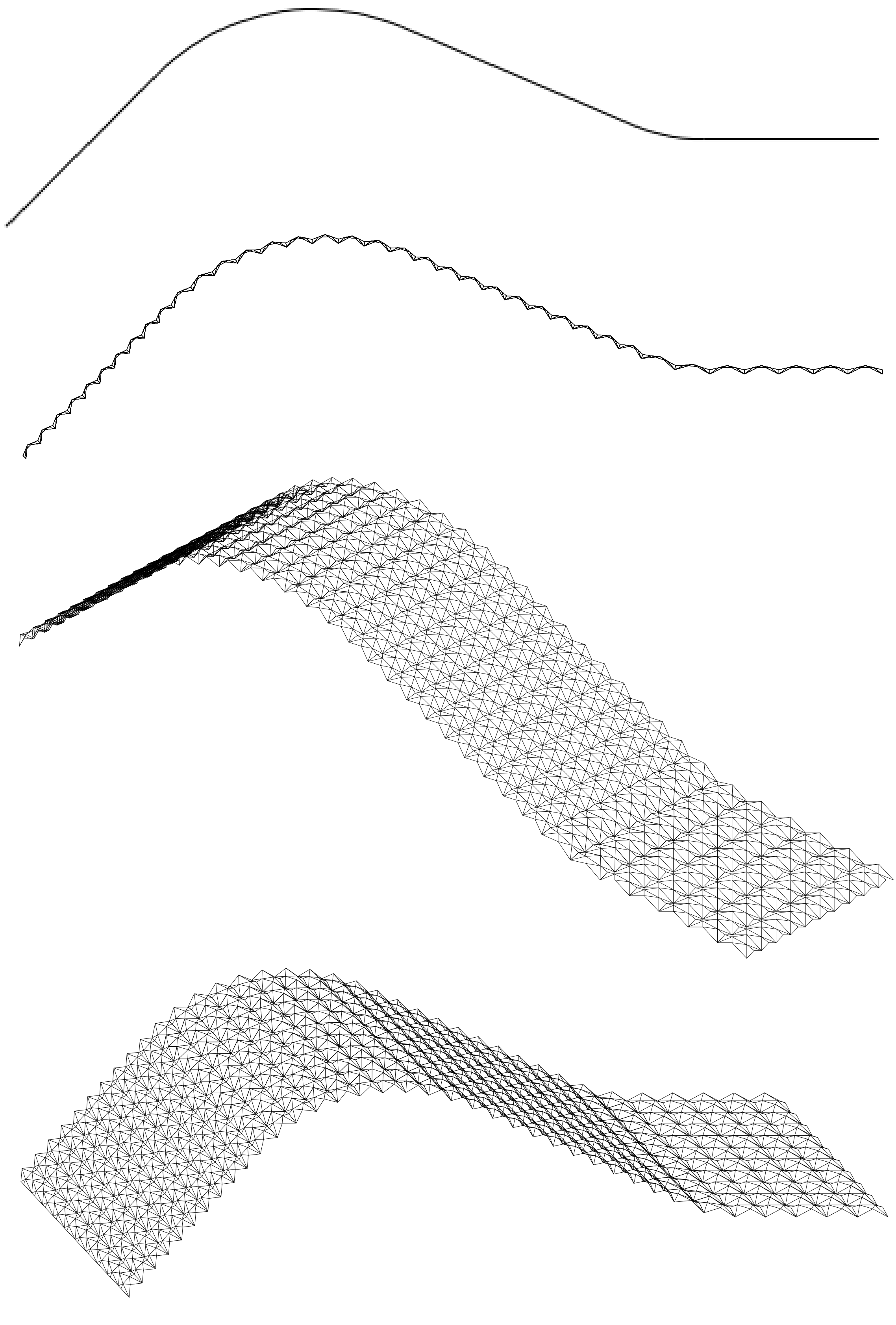
Слика 3.5.16.10. Ротација примитива Р – 10 на део круга



Слика 3.5.16.11. Сложена форма набора настала од примитива P – 5



Слика 3.5.16.12. Сложена форма набора настала од примитива P – 5

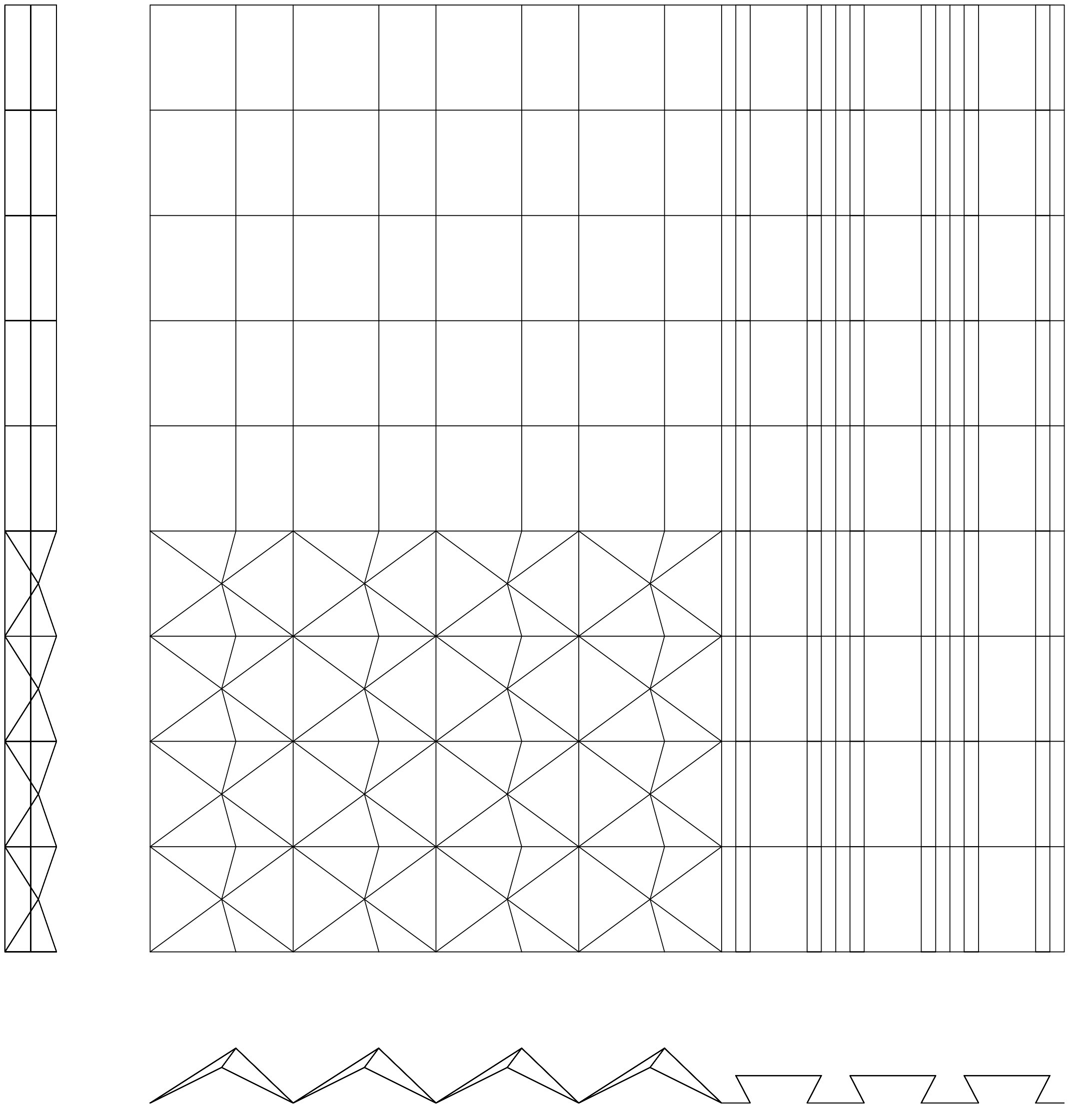


Слика 3.5.16.13. Сложена форма набора настала од примитива Р – 5

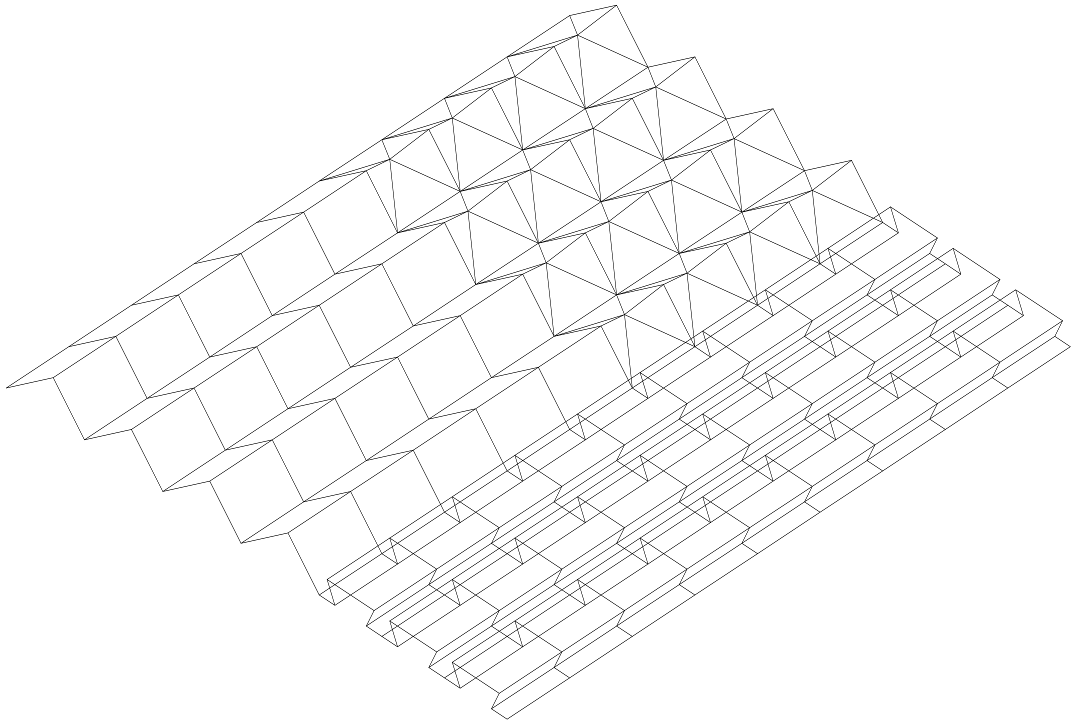
### 3.6 Набори добијени комбинацијом више примитива - сложени набори

Сложени набори се добијају сабирањем сродних примитива. Сродни примитиви су они који по контури ивице на коју се настављају поседују исту вредност ординате у правцу у што омогућава њихово уклапање. Сложени набори су интерпретирани кроз примере

### 3.6.1 Сложени набор P 1 + P 2 + P 7

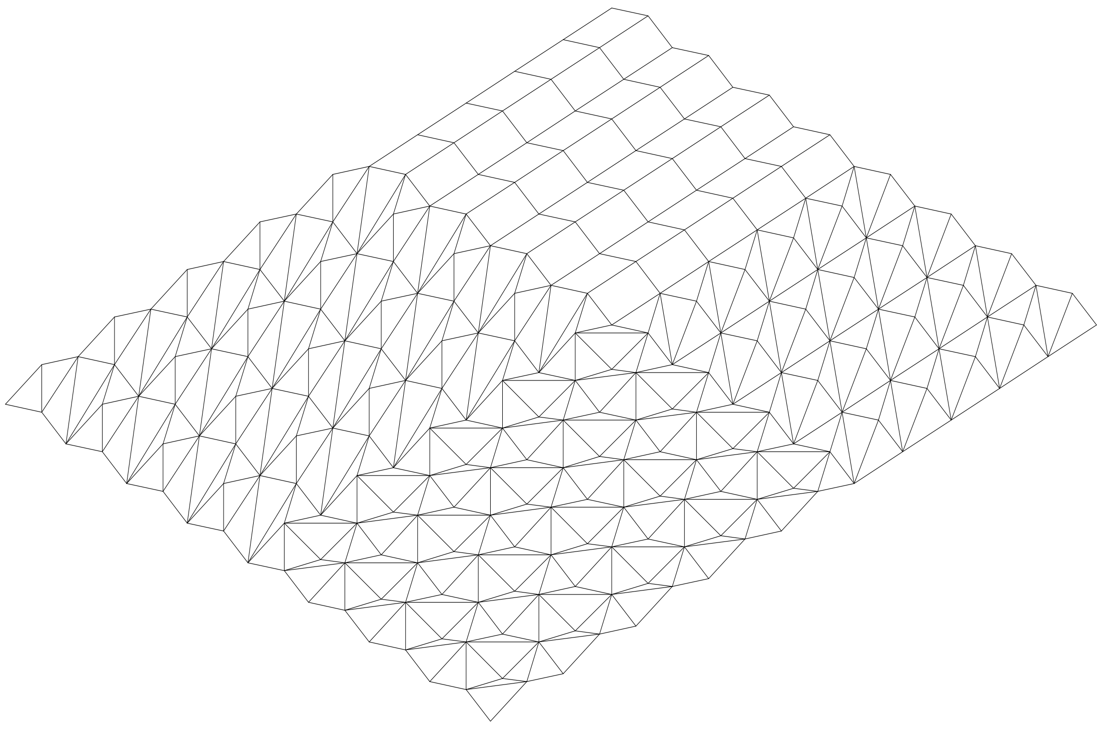


Слика 3.6.1. а) Пример сложеног набора 1, раванске пројекције

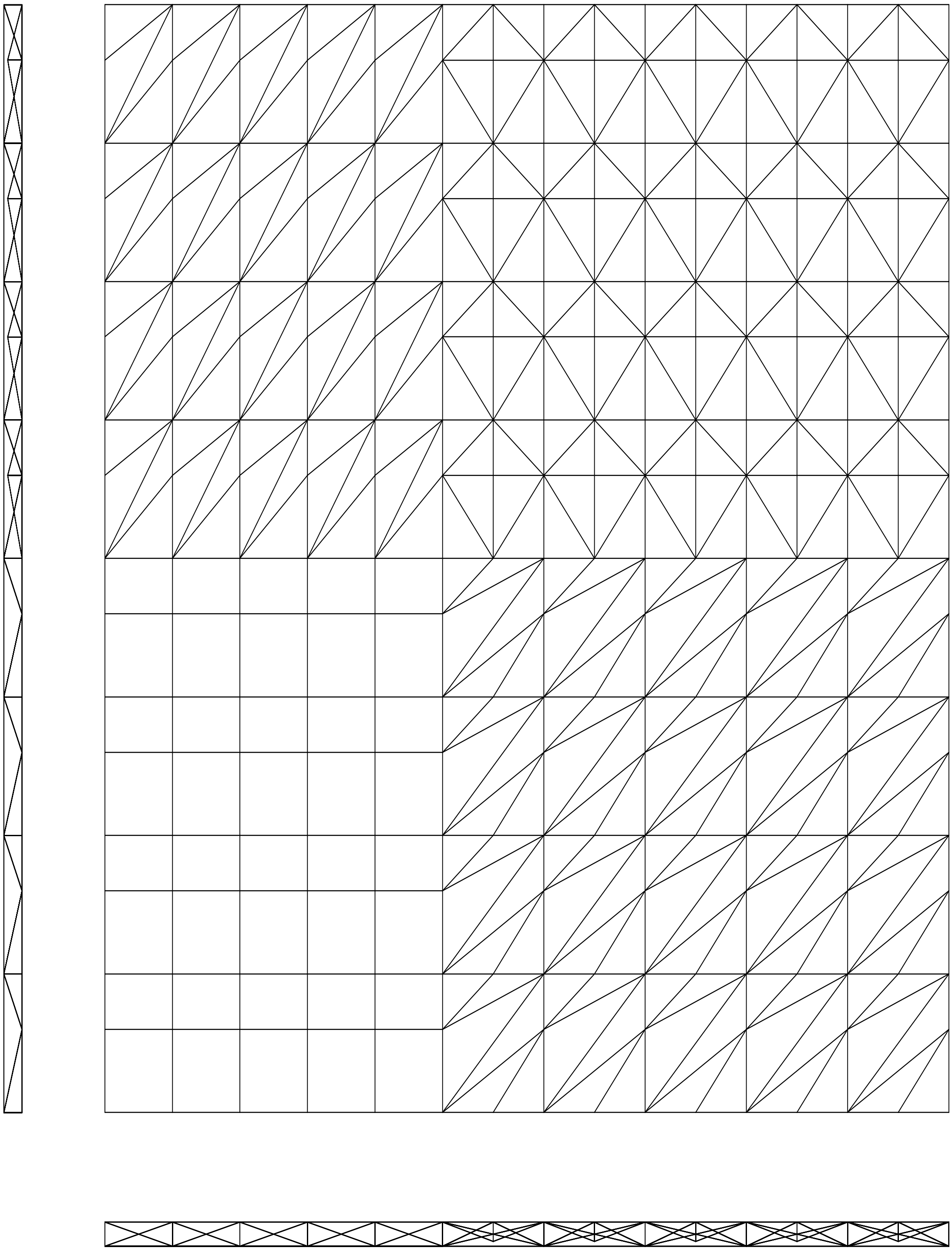


Слика 3.6.1.b) Пример сложеног набора 1, изометрија

### 3.6.2 Сложен набор P 1 + P 3 + P 4 + P 10

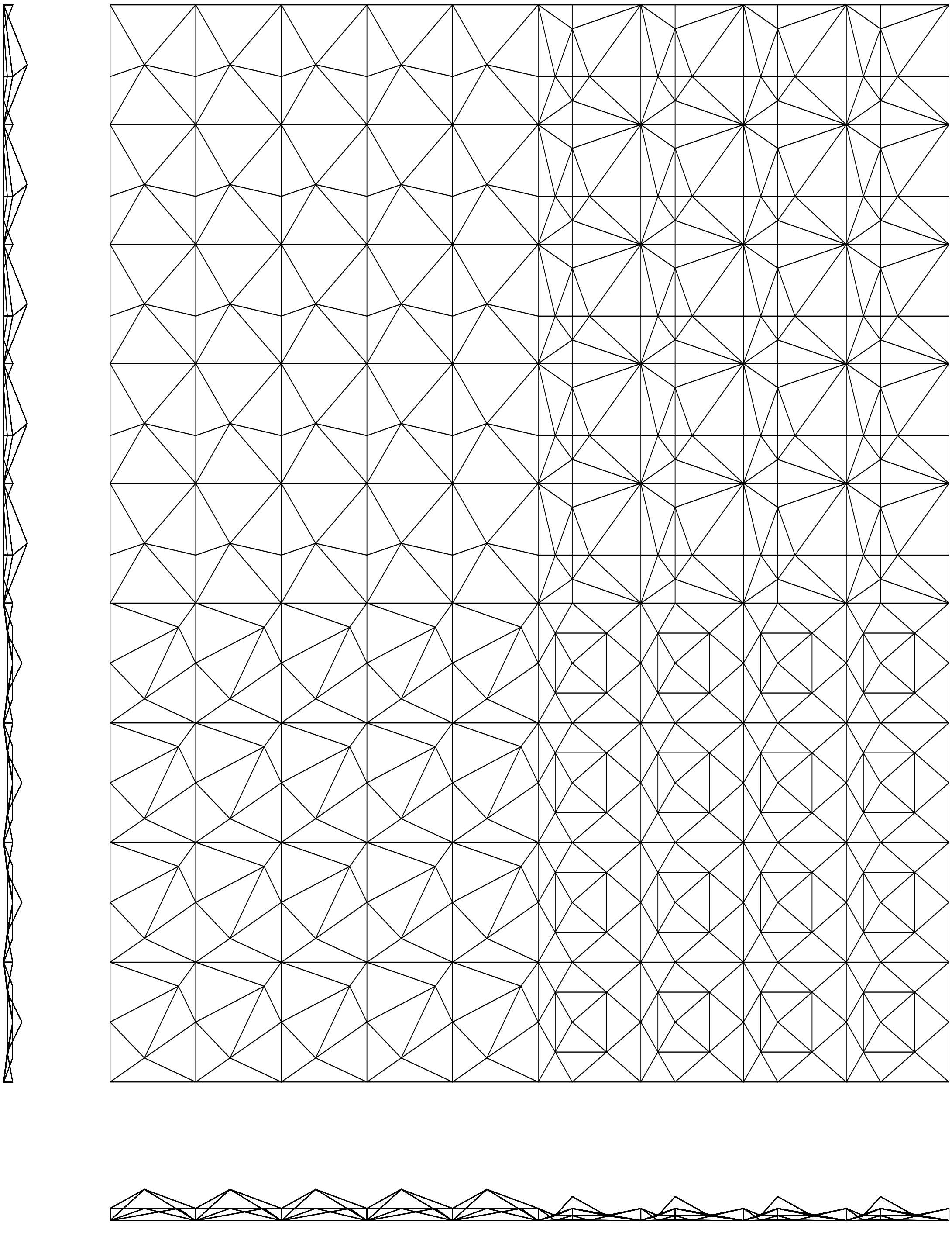


Слика 3.6.2 b) Пример сложеног набора 2, изометрија

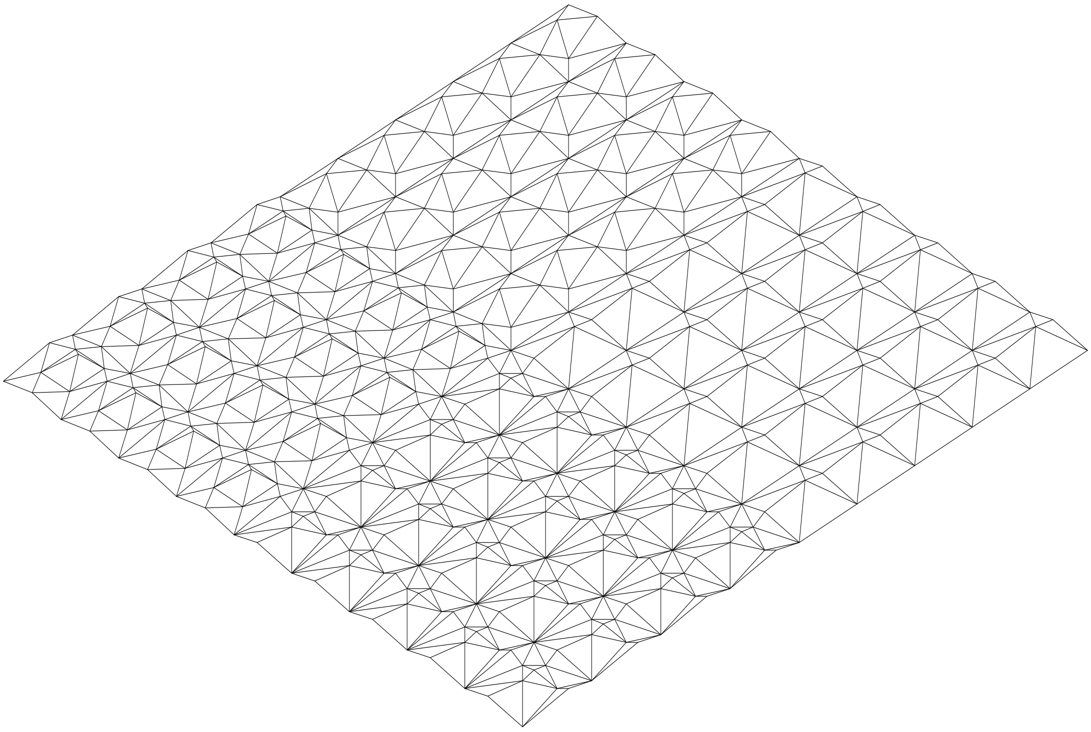


Слика 3.6.2. а) Пример сложеног набора2, раванске пројекције

### 3.6.3 Сложен набор P 5 + P 7 + P 13 + P 13



Слика 3.6.3. а) Пример сложеног набора 3, раванске пројекције



Слика 3.6.3. b) Пример сложеног набора 3, изометрија

# 4. ОСНОВИ ТЕОРИЈЕ ЕЛАСТИЧНОСТИ И МЕТОД КОНАЧНИХ ЕЛЕМЕНАТА – МКЕ

## 4.1 Уводна разматрања

Смеле архитектонске идеје често су током времена биле осуђене на немогућност реализације из разлога недовољне развијености прорачунских метода, односно чињеница које би поткрепиле сигурност и оправданост реализације. У садашњем временском периоду постало је уобичајено да су инжењери најчешће само корисници одговарајућих програмских пакета креираних од стране експерата из области рачунарских технологија, базираних на стеченим знањима и потребама одређене струке. Међутим, неоспорно је да постоји потреба за укључивањем различитих научних дисциплина, па самим тим и инжењера у процесе креирања програмских пакета, а све у циљу добијања ефикасног алата прилагођеног конкретним потреба струке. Употреба уско специјализованих знања подразумева сарадњу са експертима из различитих научних области. Свакако, стручњаци из области архитектуре не смеју бити изостављени. Архитектура као научна дисциплина суочена са оваквим развојем технологије мора дати пун допринос при развоју генеративних алгоритама у процесу компјутерског пројектовања.

Применом рачунара инжењерство је подигнуто на значајно виши технолошки ниво. Сложени прорачуни пружају нам увид у комплексно понашање конструкције. Математичком формулацијом и дигитализацијом континуума отварају се потпуно нове могућности приступу архитектонског пројектовања и изградњи објеката. Креирање модела и решавање компликованих нумеричких метода уз помоћ рачунара омогућили су разноликост истраживања, односно обезбеђена је анализа компјутерски генерисаних конструкција произвољне геометрије и њихове динамичке трансформације. Паралелно са развојем рачунарске технике развијао се и метод коначних елемената, тренутно најзаступљенији метод коришћена у области прорачуна инжењерских конструкција. Занимљиво је да првобитни приступ проблему био чисто инжењерски, да би недуго после добио и математичку формулацију.

У зависности од дефинисаних критеријума, могуће су различите класификације конструктивних система. Према начину преношења утицаја конструктивне елементе је начелно могуће поделити на: линијске (1D проблем), површинске (2D проблем) и просторне (3D проблем). Линијски конструктивни елементи поседују једну димензију израженију у односу на остале и сва разматрања везана за њихово понашање се врше у односу на ту димензију. Код површинских конструктивних система једна димензија (дебљина) је знатно мања од друге две димензије у односу на које се врше сва разматрања везана за њихово понашање. У површинске конструктивне системе се убрајају танке еластичне плоче и љуске, као и мембране које представљају специјални случај танких љуски. За разлику од плоча које су равне, љуске представљају закривљене носеће структуре. Код просторних система трећа димензија, дебљина, се не може занемарити.

Набoране конструкције представљају један тип просторних структура и као такве припадају области конструктивних система. Преношење оптерећења код набораних конструкција врши се површином планарних плоча и преко греда (укрућења) које се налазе на месту прелома плоча. Од пресудне важности за овај рад је геометријска карактеристика мреже набора која се огледа у поседовању различитих типова аксијалне и ротационе симетрије. Стога, закључујемо да набори чине комбинацију линијских и површинских конструктивних елемената. Одлике симетрије набораних конструкција (основни елемент или шаблон, углавном не поседује ни једну симетрију) нису повољне за примену групно суперматричне процедуре, па самим тим имплементација групно суперматричне процедуре у метод коначног елемента захтева одређење кораке који ће бити приказани у поглављима која следе. При томе је коришћена следећа литература у којој су детаљније приказана извођења и докази (Maneski, 1998) (Sekulović, Matrična analiza konstrukcija, 1991)

## 4.2 Основи линеарне теорије еластичности

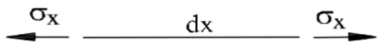
У линеарној теорији еластичности једначине везе између сила, померања и деформације штапа су линеарне. Сматрамо да се набори понашају у потпуности према теорији еластичности, тако да важе следеће претпоставке:

- Претпоставка о малим померањима
- Претпоставка о малим деформацијама
- Хуков (Hook) закон о физичкој линеарности

С обзиром да се сматрају познатим, сви изрази су приказани у њиховом коначном облику и приказани су само изрази који су релевантни за овај рад. Теорија за задатак има успостављање једначина за добијање величина померања, деформације и напона тела. Дефинишемо следеће појмове:

- Померање – промена димензија услед деловања оптерећења на тело
- Деформација – релативно померање у односу на недеформисано тело
- Напон – унутрашња сила по јединици површине посматраног пресека тела и не представља физичку величину. Имамо нормалан (σ) и смичући напон (γ)

На слици 4.2.1 графички су приказани вектори напона линијског елемента:




Слика 4.2.1. Вектори напона линијског елемента

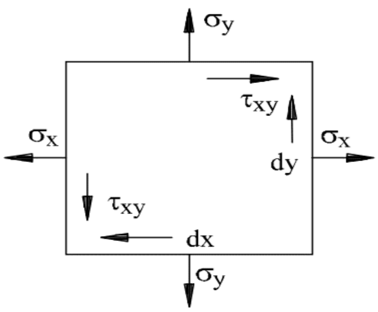
При том дефинишемо непознате величине било које тачке. Број непознатих величина је 3:

Померање {f}: $\{u\} = \{u\ (x)\}$

Деформација {ε}: $\{\varepsilon_x\}$ (5)

Напон {σ}: $\{\sigma_x\}$

На слици 4.2.2 графички су приказани вектори напона површинског елемента:




Слика 4.2.2. Вектори напона површинског елемента

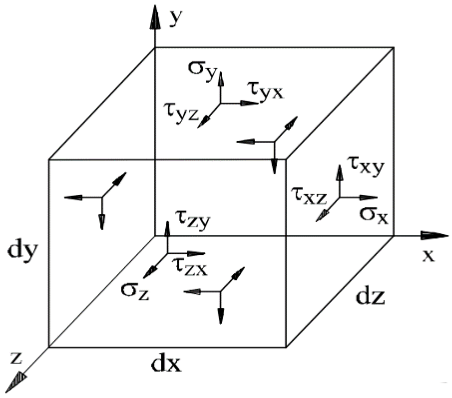
При том дефинишемо непознате величине било које тачке. Број непознатих величина је 8:

Померање {f}: $\begin{Bmatrix} u \\ v \end{Bmatrix} = \begin{Bmatrix} u\,(x,y) \\ v\,(x,y) \end{Bmatrix}$

Деформација {ε}: $\{\varepsilon_x, \varepsilon_y, \gamma_{xy}\}^T$ $\quad \gamma_{xy} = \gamma_{yx}$ (6)

Напон {σ}: $\{\sigma_x, \sigma_y, \tau_{xy}\}^T$ $\quad \tau_{xy} = \tau_{yx}$

На слици 4.2.3. графички су приказани вектори напона просторног елемента:




Слика 4.2.3. Вектори напона запреминског елемента

При том дефинишемо непознате величине било које тачке. Број непознатих величина је 15:

Померање {f}: $\begin{Bmatrix} u \\ v \\ w \end{Bmatrix} = \begin{Bmatrix} u\,(x,y,z) \\ v\,(x,y,z) \\ w\,(x,y,z) \end{Bmatrix}$

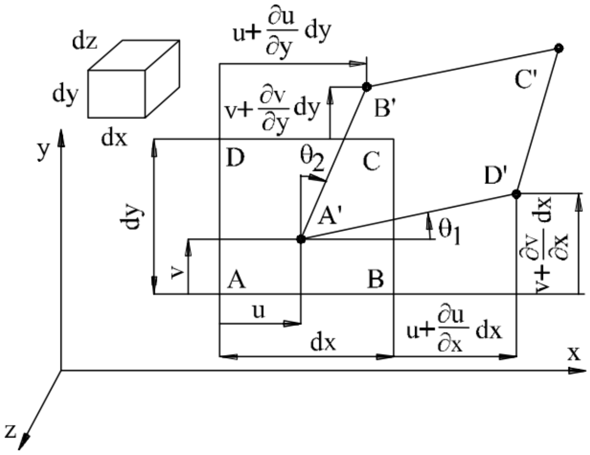
Деформација {ε}: $\{\varepsilon_x, \varepsilon_y, \varepsilon_z, \gamma_{xy}, \gamma_{yz}, \gamma_{zx}\}^T$ $\quad \gamma_{xy} = \gamma_{yx}$ $\quad \gamma_{yz} = \gamma_{zy}$ $\quad \gamma_{zx} = \gamma_{xz}$ (7)

Напон {σ}: $\{\sigma_x, \sigma_y, \sigma_z, \tau_{xy}, \tau_{yz}, \tau_{zx}\}^T$ $\quad \tau_{xy} = \tau_{yx}$ $\quad \tau_{yz} = \tau_{zy}$ $\quad \tau_{zx} = \tau_{xz}$

### 4.2.1 Успостављање веза деформација-померање

Настале деформације у телу се могу изразити преко померања u, v и w. Посматрајмо мали правоугаони елемент ABCD чије су дужине ивица dx и dy.

Напоменули смо већ да важи претпоставка о малим померањима и деформацијама. Корелација између нормалних компоненти деформација $\varepsilon_x, \varepsilon_y$ и смичуће деформације $\gamma_{xy}$ приказани су на слици 4.2.4.




Слика 4.2.4. Деформација раванског елемента

Односно, преко израза :

$$\varepsilon_x = \frac{\partial u}{\partial x} \qquad \varepsilon_y = \frac{\partial v}{\partial y} \qquad \gamma_{xy} = \gamma_{yx} = \frac{\partial v}{\partial x} + \frac{\partial u}{\partial y}$$

$$\varepsilon_z = \frac{\partial w}{\partial z} \qquad \gamma_{yz} = \gamma_{zy} = \frac{\partial v}{\partial z} + \frac{\partial w}{\partial y} \qquad \gamma_{zx} = \gamma_{xz} = \frac{\partial u}{\partial z} + \frac{\partial w}{\partial x} \tag{8}$$

### 4.2.2 Услови компатибилности

Посматрано тело је континуално и сматрамо да после деформисања и поље померања остаје континуално. Услови компатибилности обезбеђују континуалност и представљају допунске једначине за решавање диференцијалних једначина унутрашње равнотеже:

За површинске елементе оне гласе:

$$\frac{\partial^2 \varepsilon_x}{\partial y^2} + \frac{\partial^2 \varepsilon_y}{\partial x^2} = \frac{\partial^2 \gamma_{xy}}{\partial x \partial y}$$

$$\frac{\partial^2 \varepsilon_z}{\partial x^2} = 0 \qquad \frac{\partial^2 \varepsilon_z}{\partial y^2} = 0 \qquad \frac{\partial^2 \varepsilon_z}{\partial x \partial y} = 0 \tag{9}$$

За просторне елементе оне гласе:

$$\frac{\partial^2 \varepsilon_x}{\partial y^2} + \frac{\partial^2 \varepsilon_y}{\partial x^2} = \frac{\partial^2 \gamma_{xy}}{\partial x\, \partial y} \qquad \frac{\partial^2 \varepsilon_y}{\partial z^2} + \frac{\partial^2 \varepsilon_z}{\partial y^2} = \frac{\partial^2 \gamma_{yz}}{\partial y\, \partial z} \qquad \frac{\partial^2 \varepsilon_z}{\partial x^2} + \frac{\partial^2 \varepsilon_x}{\partial z^2} = \frac{\partial^2 \gamma_{zx}}{\partial z\, \partial x}$$

$$\frac{\partial}{\partial x}\left(\frac{\partial \gamma_{xy}}{\partial z} - \frac{\partial \gamma_{yz}}{\partial x} + \frac{\partial \gamma_{zx}}{\partial y}\right) = 2\frac{\partial^2 \varepsilon_x}{\partial y\, \partial z}$$

$$\frac{\partial}{\partial y}\left(\frac{\partial \gamma_{xy}}{\partial z} + \frac{\partial \gamma_{yz}}{\partial z} - \frac{\partial \gamma_{zx}}{\partial y}\right) = 2\frac{\partial^2 \varepsilon_y}{\partial z\, \partial x} \qquad (10)$$

$$\frac{\partial}{\partial z}\left(-\frac{\partial \gamma_{xy}}{\partial z} + \frac{\partial \gamma_{yz}}{\partial z} + \frac{\partial \gamma_{zx}}{\partial y}\right) = 2\frac{\partial^2 \varepsilon_z}{\partial x\, \partial y}$$

### 4.2.3 Једначине спољашње равнотеже

Уколико је тело у равнотежи, збир свих унутрашњих и спољашњих сила (F) и момената (M) у сваком правцу је једнак нули. Успостављамо следеће једначине:

линијски елемент: $\Sigma F_x = 0 \quad \Sigma M_x = 0$

површински елемент: $\Sigma F_x = 0 \quad \Sigma F_y = 0 \quad \Sigma M_z = 0$ (11)

запремински елемент: $\Sigma F_x = 0 \quad \Sigma F_y = 0 \quad \Sigma F_z = 0$

$\Sigma M_x = 0 \quad \Sigma M_y = 0 \quad \Sigma M_z = 0$

### 4.2.4 Једначине унутрашње равнотеже

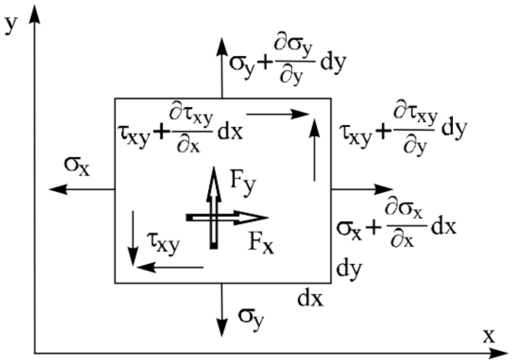
Основни проблем теорије еластичности може се формулисати на следећи начин: за дату геометрију, дати материјал и дато спољне оптерећене тела одредити напоне, деформације и померања која се јављају у телу. Напонско стање било које тачке дефинисано је тензором напона:

$$\sigma = \begin{bmatrix} \sigma_x & \tau_{xy} & \tau_{xz} \\ \tau_{yx} & \sigma_y & \gamma_{yz} \\ \gamma_{zx} & \tau_{zy} & \sigma_z \end{bmatrix} \qquad (12)$$

Применом става о коњугованости смичућих напона добијамо да је шест од девет компоненти тензора напона међусобно независно, односно:

$$\tau_{xy} = \tau_{yx} \qquad \tau_{xz} = \tau_{zx} \qquad \tau_{yz} = \tau_{zy} \qquad (13)$$

На слици 4.2.5 приказане су компоненте напона са њиховим прираштајем по јединици запремине за површински елемент. Једначине услова равнотеже се изводе из чињенице да је збир свих унутрашњих запремински сила и прираштаја напона за одговарајуће осе идентички једнак нули. Оне се још и називају диференцијалне једначине равнотеже.




Слика 4.2.5. Компоненте напона са прираштајима

Сада исписујемо једначине услова равнотеже:

линијски елемент:

$$\frac{\partial \sigma_x}{\partial x} + F_x = 0 \tag{14}$$

површински елемент:

$$\frac{\partial \sigma_x}{\partial x} + \frac{\partial \tau_{xy}}{\partial y} + F_x = 0 \qquad \frac{\partial \tau_{xy}}{\partial x} + \frac{\partial \sigma_y}{\partial y} + F_y = 0$$

$$\frac{\partial \sigma_x}{\partial x} + \frac{\partial \tau_{xy}}{\partial y} + \frac{\partial \tau_{xz}}{\partial z} + F_x = 0 \tag{15}$$

запремински елемент:

$$\frac{\partial \tau_{xy}}{\partial x} + \frac{\partial \sigma_y}{\partial y} + \frac{\partial \tau_{zy}}{\partial z} + F_y = 0$$

$$\frac{\partial \tau_{xz}}{\partial x} + \frac{\partial \tau_{yz}}{\partial y} + \frac{\partial \sigma_z}{\partial z} + F_z = 0 \tag{16}$$

### 4.2.5 Успостављање релација између напона и деформација

Претпостављамо да се ради о линеарно еластично изотропном телу, што значи да важи Хуков закон о релацији деформације и напона. Релацију у матричном облику представљамо као:

$$|\varepsilon| = [C]|\sigma| + |\varepsilon_o| \tag{17}$$

- $[C]$ – матрица еластичних коефицијената

- $|\varepsilon_o|$ – вектор иницијалне деформације

Сада можемо и напон изразити преко деформације:

$$|\sigma| = [C]^{-1}(|\varepsilon| - |\varepsilon_o|) = [D](|\varepsilon| - |\sigma_o|) \tag{18}$$

Успостављају се релације:

| | истезање | савијање | |
|---|---|---|---|
| линијски елемент: | $[C] = \frac{1}{E}$ <br> $[D] = E$ | $[C] = \frac{1}{G}$ <br> $[D] = E$ | (19) |

површински елемент - равно стање напона:

$$[C] = \frac{1}{E}\begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & 2(1+\nu) \end{bmatrix} [D] = \frac{E}{1-2\nu^2}\begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{bmatrix} \tag{20}$$

површински елемент - равно стање деформације:

$$[C] = \frac{1+\nu}{E}\begin{bmatrix} 1-\nu & -\nu & 0 \\ -\nu & 1-\nu & 0 \\ 0 & 0 & 2 \end{bmatrix}$$

$$[D] = \frac{E}{(1+\nu)(1-2\nu)}\begin{bmatrix} 1-\nu & \nu & 0 \\ \nu & 1-\nu & 0 \\ 0 & 0 & \frac{1-2\nu}{2} \end{bmatrix} \tag{21}$$

запремински елемент:

$$[C] = \frac{1}{E}\begin{bmatrix} 1 & -\nu & -\nu & 0 & 0 & 0 \\ -\nu & 1 & -\nu & 0 & 0 & 0 \\ -\nu & -\nu & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2(1+\nu) & 0 & 0 \\ 0 & 0 & 0 & 0 & 2(1+\nu) & 0 \\ 0 & 0 & 0 & 0 & 0 & 2(1+\nu) \end{bmatrix} \tag{22}$$

$$[D] = \frac{E}{(1+\nu)(1-2\nu)}\begin{bmatrix} 1-\nu & \nu & \nu & 0 & 0 & 0 \\ \nu & 1-\nu & \nu & 0 & 0 & 0 \\ \nu & \nu & 1-\nu & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1-2\nu}{\nu} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1-2\nu}{\nu} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1-2\nu}{\nu} \end{bmatrix}$$

- $[D]$ – матрица еластичности материјала

- $|\sigma_o|$ – вектор иницијалног напона
- $\nu$ – Поасонов (Poisson) коефицијент
- E – Јангов (Young) модул еластичности материјала
- G – Модул клизања материјала

$$G = \frac{E}{2(1+\nu)} \tag{23}$$

Матрица еластичности линерано еластичног анизотропно - ортотропног материјала за површински елемент при равном стању напона:

$$[D] = \begin{bmatrix} D_{11} & D_{12} & 0 \\ D_{21} & D_{22} & 0 \\ 0 & 0 & D_{66} \end{bmatrix}$$

$$D_{11} = \frac{E_1}{1-\nu_{12}\nu_{21}} \qquad D_{22} = \frac{E_2}{1-\nu_{12}\nu_{21}} \qquad D_{66} = 2G_{12} \tag{24}$$

$$D_{12} = D_{21}\frac{\nu_{21}E_1}{1-\nu_{12}\nu_{21}} = \frac{\nu_{12}E_2}{1-\nu_{12}\nu_{21}}$$

Аналогно матрица еластичности линерано еластичног анизотропно - ортотропног материјала за запремински елемент гласи:

$$[D] = \begin{bmatrix} D_{11} & D_{12} & D_{13} & 0 & 0 & 0 \\ D_{21} & D_{22} & D_{23} & 0 & 0 & 0 \\ D_{31} & D_{32} & D_{33} & 0 & 0 & 0 \\ 0 & 0 & 0 & D_{44} & 0 & 0 \\ 0 & 0 & 0 & 0 & D_{55} & 0 \\ 0 & 0 & 0 & 0 & 0 & D_{66} \end{bmatrix} \tag{25}$$

### 4.2.6 Основи теорије танких плоча

У општем случају набор представља управо скуп танких плоча. У теорији танких плоча разликујемо две врсте напрезања, савијање и равно напрезање плоча. У оба случаја сматрамо да материјал изотропан и линеарно еластичан.

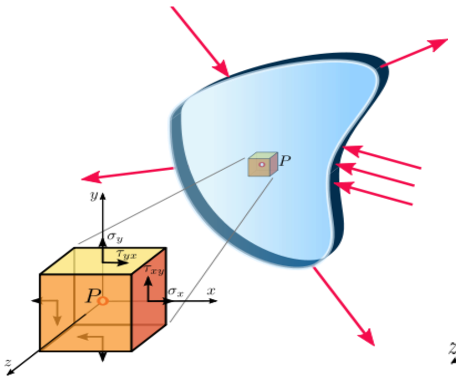
Код савијања плоча важе Кирхохове (Kirchoff) претпоставке:

- Линеаран елемент управан на средњу раван плоче пре деферомације, остаје прав, непромењене дужине и управан на деформисану средњу раван (еластичну површину) и после деформације

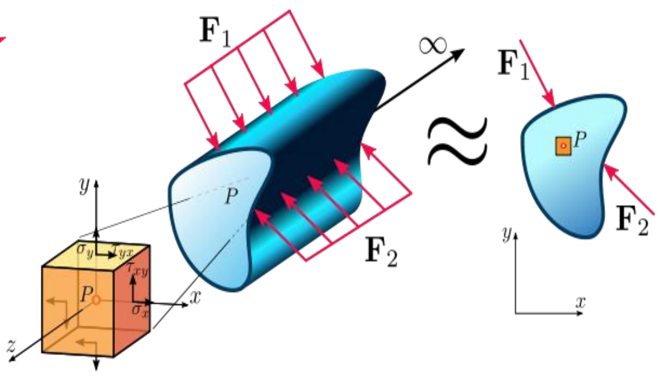
- Дужина и угао између линијских елемената средње равни се не мењају при деформацији
- Нормални напон $\boldsymbol{\sigma_z}$ за равни паралелне са средњом равни плоче сматрају се занемарљиво малим у поређењу са осталим компоненталним напонима

Код равног напрезање важе следеће претпоставке (слика 4.2.6.):

- Деформација се одвија без кривљења, односно средња раван плоче остаје равна и после деформације
- Оптерећење плоче је равномерно распоређено по њеној дебљини
- Напони $\boldsymbol{\sigma_x}$, $\boldsymbol{\sigma_y}$ и $\boldsymbol{\tau_{xy}}$ су равномерно распоређени по дебљини плоче
- Напони $\boldsymbol{\sigma_z}$, $\boldsymbol{\tau_{zx}}$ и $\boldsymbol{\tau_{zy}}$ су једнаки нули




Слика 4.2.6. Равно стање напона




Слика 4.2.7. Равно стање деформација

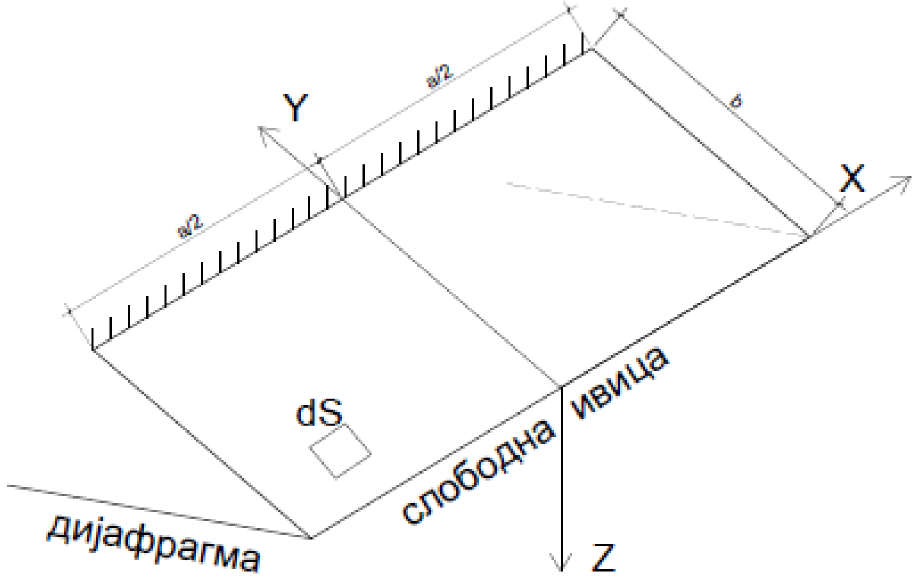
Код равног стања деформације важи следећа претпоставка(слика 4.2.7):

- Оно представља такво стање деформације где су нормална деформација $\boldsymbol{\varepsilon_z}$, и клизања $\boldsymbol{\gamma_{zx}}$ и $\boldsymbol{\gamma_{zy}}$ су једнаки нули. Другим речима, померања и деформација у правцу z осе имају само компоненте померања у равни x0y.

$$u = u\,(x,\ y)$$

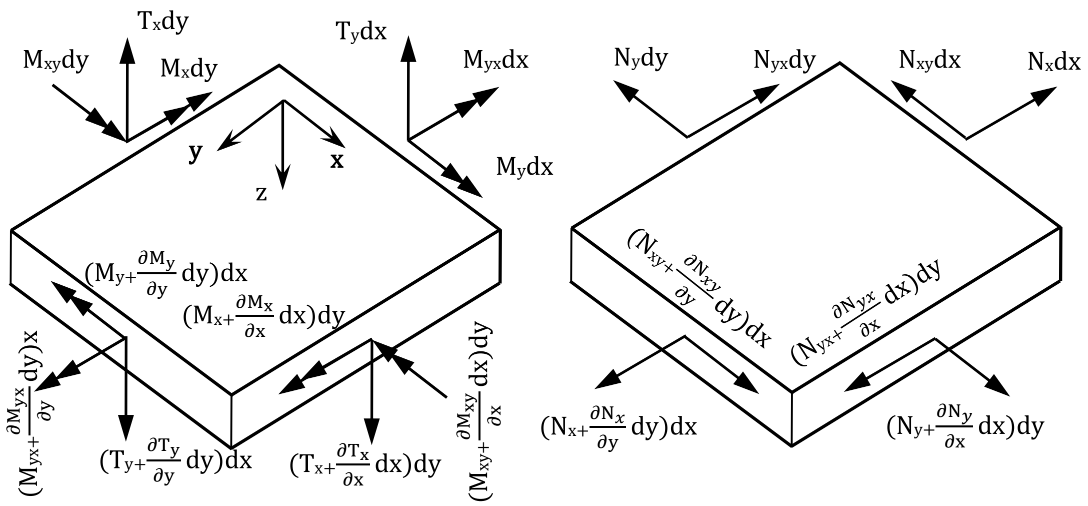
$$v = v\,(x,\ y)$$

У овом делу биће још постављени услови равнотеже на основу принципа минимума потенцијалне енергије - енергетски приступ. На слици 4.2.8 приказана је једна плоча набора. Плоча је укљештена у дијафрагме, такође и подужни спој две плоче се може третирати као укљештење, док је друга подужна ивица слободно ослоњена. Нека су u, v и w померања плоче у x,y и правцу.




Слика 4.2.8. Једна плоча набора

Унутрашње силе (на делу плоче dS) у случају савијања плоче су приказане на слици 4.2.9. лево, док у случају аксијалног напрезања на слици 4.2.9. десно.




Слика 4.2.9. Унутрашње силе у плочи

Укупна потенцијална енергија је дата изразом:

$$U = U_1 + U_2 \tag{26}$$

где је $U_1$ потенцијална енергија услед истезања плоча као мембране (равно напрезање), а $U_2$ потенцијална енергија услед савијања, следи:

$$
\begin{aligned}
U_1 = \frac{Et}{2(1-\nu^2)} \iint \Bigg\{ & \left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial u}{\partial x}\right)\left(\frac{\partial w}{\partial x}\right)^2 + \left(\frac{\partial v}{\partial y}\right)^2 + \left(\frac{\partial v}{\partial y}\right)\left(\frac{\partial w}{\partial y}\right)^2 \\
& + \frac{1}{4}\left[\left(\frac{\partial w}{\partial x}\right)^2 + \left(\frac{\partial w}{\partial y}\right)^2\right]^2 \\
& + 2\nu\left[\left(\frac{\partial u}{\partial x}\right)\left(\frac{\partial v}{\partial y}\right) + \frac{1}{2}\left(\frac{\partial v}{\partial y}\right)\left(\frac{\partial w}{\partial x}\right)^2 + \frac{1}{2}\left(\frac{\partial u}{\partial x}\right)\left(\frac{\partial w}{\partial y}\right)^2\right] \\
& + \frac{1-\nu}{2}\left[\left(\frac{\partial u}{\partial y}\right)^2 + 2\left(\frac{\partial u}{\partial y}\right)\left(\frac{\partial v}{\partial x}\right) + \left(\frac{\partial v}{\partial x}\right)^2 + 2\left(\frac{\partial u}{\partial y}\right)\left(\frac{\partial w}{\partial x}\right)\left(\frac{\partial w}{\partial y}\right) \right. \\
& \left. + 2\left(\frac{\partial v}{\partial x}\right)\left(\frac{\partial w}{\partial x}\right)\left(\frac{\partial w}{\partial y}\right)\right]\Bigg\} dxdy
\end{aligned}
\tag{27}
$$

Са обзиром да се при савијању компонентална померања u и v могу изразити преко померања *w*, у изразу са потенцијалну енергију фигурише само померање *w*:

$$
U_2 = \frac{1}{2} D \iint \left\{ \left(\frac{\partial^2 w}{\partial x^2} + \frac{\partial^2 w}{\partial y^2}\right)^2 - 2(1-\nu)\left[\frac{\partial^2 w}{\partial x^2}\frac{\partial^2 w}{\partial y^2} - \left(\frac{\partial^2 w}{\partial x\, \partial y}\right)^2\right]\right\} dxdy
$$

$$
D = \frac{Et^3}{12(1-\nu^2)} \tag{28}
$$

где су: D цилиндрична крутост плоче , E модул еластичности , t дебљина плоче, ν Поасонов коефицијент. Варирање потенцијалне енергије по чворним параметрима доводи до матрице крутости плоче.

Како би се одредили изрази за померања u, v и w, неопходно је поставити граничне услове. Померања се могу одредити принципом виртуелног померања. Гранични услови су дефинисани следећим изразима:

Укљештена ивица:

$$
(w)_{x=b} = 0, \qquad \left(\frac{\partial w}{\partial x}\right)_{x=b} = 0 \tag{29}
$$

Слободно ослоњена ивица:

$$
(w)_{y=\pm a/2} = 0, \qquad \left(\frac{\partial^2 w}{\partial y^2}\right)_{y=\pm a/2} = 0 \tag{30}
$$

Слободна ивица:

$$\left[\frac{\partial^3 w}{\partial x^3} + (2-\nu)\frac{\partial^3 w}{\partial x\,\partial y^2}\right]_{x=0} = 0, \qquad \left[\frac{\partial^2 w}{\partial x^2} + \nu\frac{\partial^2 w}{\partial y^2}\right]_{x=0} = 0 \qquad (31)$$

Потенцијална енергија се може изразити и преко момената савијања, момената торзије и аксијалних сила. На основу теорије еластичности познате су следеће корелације:

$$M_x = -D\left(\frac{\partial^2 w}{\partial x^2} + \nu\frac{\partial^2 w}{\partial y^2}\right) \qquad M_y = -D\left(\frac{\partial^2 w}{\partial x^2} + \nu\frac{\partial^2 w}{\partial y^2}\right)$$

$$M_{xy} = -D(1-\nu)\frac{\partial^2 w}{\partial x\,\partial y}$$

$$T_x = -D\left(\frac{\partial^3 w}{\partial x^3} + \frac{\partial^3 w}{\partial x\,\partial y^2}\right) \qquad T_y = -D\left(\frac{\partial^3 w}{\partial y^3} + \frac{\partial^3 w}{\partial y\,\partial x^2}\right)$$

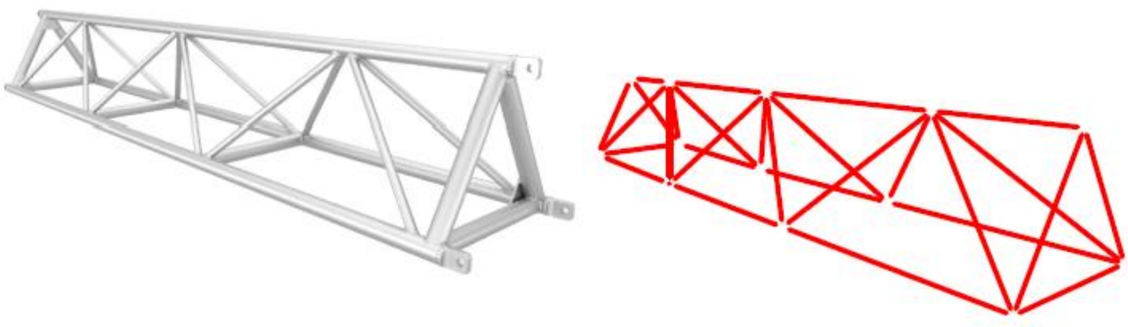
$$N_x = K\left(\frac{\partial u}{\partial x} + \nu\frac{\partial v}{\partial y}\right) N_y = K\left(\frac{\partial v}{\partial y} + \nu\frac{\partial u}{\partial x}\right)$$

$$N_{xy} = \frac{1}{2}K(1-v)\left(\frac{\partial u}{\partial y} + \nu\frac{\partial v}{\partial x}\right)$$

$$K = \frac{Et}{(1-\nu^2)} \qquad (32)$$

### 4.3 Уводна разматрања - метод коначних елемената МКЕ

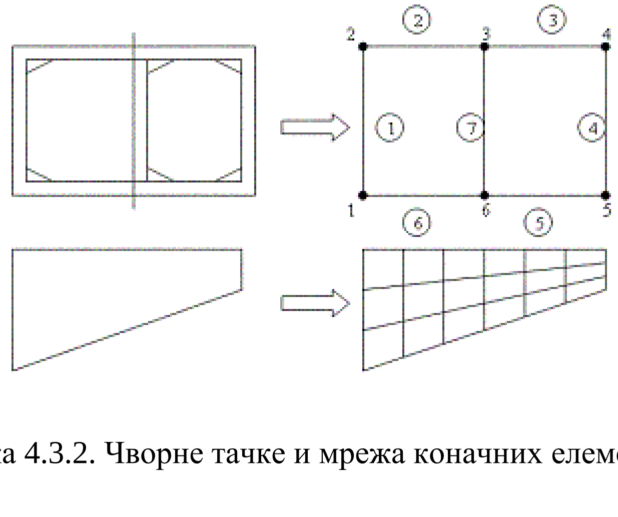
Опште је познато да се метод коначних елемената заснива на физичкој дискретизацији разматраног домена, односно континуума, а основу за сва разматрања представља део домена коначних димензија, поддомен или коначни елемент. Са становишта физичке интерпретације, разматрани домен, као континуум са бесконачно много степени слободе, замењује се дискретним моделом међусобно повезаних коначних елемената, са коначним бројем степени слободе. Основни задатак је да се изабере онај модел који најбоље апроксимира одговарајући гранични проблем (слика 4.3.1). (Sekulović, Metoda konačnih elemenata, 1988), (Sekulović, Matrična analiza konstrukcija, 1991), (Maneski, 1998), (Zienkiewicz, 1977)



Слика 4.3.1. Дискретизација модела

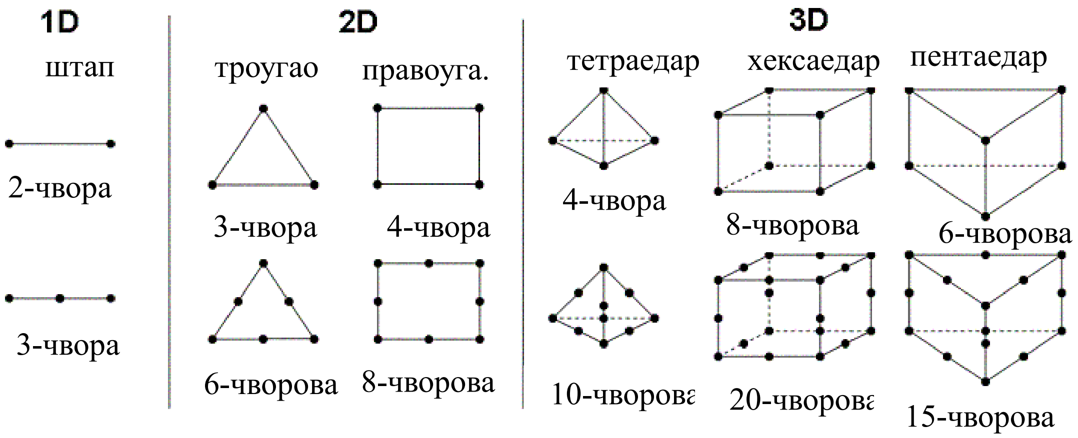
Суштина апроксимације континуума по методу коначних елемената, састоји се у следећем:

- Разматрани домен континуума, помоћу замишљених линија или површи, дели се на одређен број поддомена коначних димензија. Поједини поддомени се називају коначни елементи, а њихов скуп за цео домен систем или мрежа коначних елемената (слика 4.3.2)
- Претпоставља се да су коначни елементи међусобно повезани у коначном броју тачака, које се усвајају на контури елемента. Те тачке се називају чворне тачке или чворови (слика 4.3.2)
- Стање у сваком коначном елементу описује се помоћу интерполационих функција и коначног броја параметара у чворовима који представљају основне непознате величине у методу коначних елемената
- За анализу и прорачун система коначних елемената важе сви принципи и поступци који важе за класичне дискретне методе




Слика 4.3.2. Чворне тачке и мрежа коначних елемената

Избор одговарајућег коначног елемента треба извршити у зависности од предметне проблематике. Елементи могу бити конципирани по линији, у равни и у простору. Најчешће примењени облици коначних елемената приказани су на слици 4.3.3. На сваком коначном елементу усвајамо одређен број тачака на контури, које називамо чворови. Као основне непознате усвајају се параметри померања у чворовима, а њихов број зависи од природе разматраног проблема. Ове параметре често називамо и степени слободе. Параметри померања у чворовима у оквиру елемента немају улогу основних непознатих, већ се уводе да омогуће што бољу апроксимацију поља померања, односно деформација у елементу. Сваки тип елемента поседује особине са којима се физички модел може пресликати уз одговарајућу идеализацију




Слика 4.3.3. Типови коначних елемената

Према начину на који се изводе и формулишу основне једначине методе коначних елемената, односно једначине за поједине коначне елементе, постоје четири основна вида метода коначних елемената и то директна метода (аналогна методи деформације у прорачуну линијских носача), варијациона метода (заснива се на принципу о стационарности потенцијала), метода резидуума (заснива се на диференцијалним једначинама разматраног проблема) и метода енергетског баланса (заснива се на балансу различитих видова енергије).

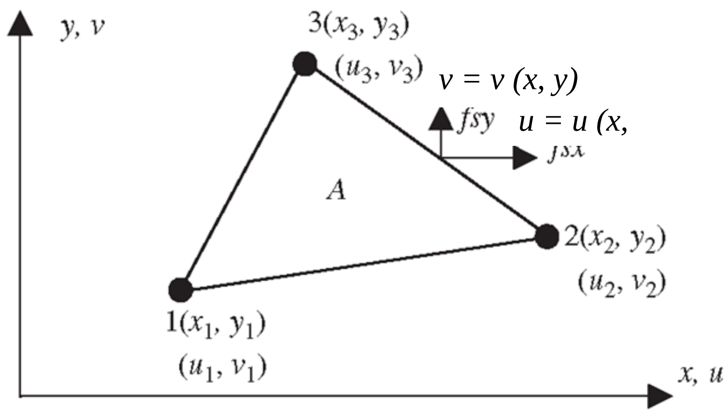
Анализа и решавање проблема у методи коначних елемената увек се своди на такозван процес корак по корак, са следећим корацима (Maneski, 1998):

- Моделирање, идеализација и дискретизација структуре, избор типа, броја величине и расподела елемената представљају најважније функције моделирања проблема
- Избор правог интерполациониг модела за поље померања. Модели се углавном дају у форми полинома. Они треба да обезбеде претпостављени модел померања, деформације и напона коначног елемента
- Формирање матрице крутости и вектора оптерећења коначних елемената. Све величине коначног елемента се израчунавају у координатном систему елемента (локални) и координатном систему структуре (глобални)
- Састављање укупне, односно глобалне, матрице крутости, вектора оптерећења и граничних услова структуре. Свака матрица крутости и вектор оптерећења коначног елемента додаје се на местима деф. глобалним редним бројем чворова елемента, као и гранични (контурни) услови
- Решавање непознатих померања чворова из једначине статичке равнотеже - систем линеарних алгебарских једначина
- Израчунавање деформације и напона коначних елемената
- Израчунавање напона структуре

### 4.3.1 Дводимензионални коначни елемент – троугаони коначни елемент плоче

У суштини дводимензионални проблеми теорије еластичности су равно стање напона и равно стање деформација. Разлика у њиховом решавању ова два проблема је само у матрици еластичности материјала.

Раније смо напоменули да напрезање елемента плоче може да се разложи на напрезање у равни плоче, односно мембранско и управно на раван плоче, односно савијање плоче (слика 4.2.8). Напрезање у равни плоче поседује два степена слободе транслације, две силе у равни и три компоненте деформације и напона. Код напрезања управно на раван плоче имамо један степен слободе транслације, једну силу у правцу нормале плоче и три компоненте деформације и напона. Стања напрезања се могу разматрати као два независна случаја. Дискретизацијом плоче на троугаоне елементе добија се скуп елемента са чворовима i, j, k, односно 1,2,3 и одговарајућим координатама (x, y). Разматрајмо предметни троугаони мембрански елемент у свом локалном координатном систему (слика 4.3.4).




Слика 4.3.4. Троугаони елемент

Усвајамо основне непознате у чворовима у виду компоненти померања u и v. Вектор померања непознатих $f_i$ у чвору i има две компоненте и једнак је:

$$f = \begin{Bmatrix} u_i \\ v_i \end{Bmatrix}$$

а вектор померања у свим чворовима:

$$f = \begin{Bmatrix} f_1 \\ f_2 \\ f_3 \end{Bmatrix}$$

Према томе, усвојени коначни елемент троугла са три чвора у теменима има шест спољашњих степени слободе, у сваком чвору по два. Претпостављамо да се померања у елементу линеарно мењају, тако да их можемо приказати у облику полинома са укупно шест непознатих коефицијената:

$$u = a_1 + a_2x + a_3y$$
$$v = a_4 + a_5x + a_6y \qquad (33)$$

Ове једначине у матричном облику гласи:

$$f = n\alpha$$

$$n = \begin{bmatrix} 1 & x & y & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & x & y \end{bmatrix} \qquad (34)$$

$$a = [a_1 \quad a_2 \quad a_3 \quad a_4 \quad a_5 \quad a_6]^T$$

Вектори деформације и напона гласе:

$$\{\varepsilon\} = \begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{Bmatrix} = \begin{Bmatrix} \dfrac{\partial u}{\partial x} \\ \dfrac{\partial v}{\partial x} \\ \dfrac{\partial v}{\partial x} + \dfrac{\partial u}{\partial y} \end{Bmatrix} \qquad \{\sigma\} = \begin{Bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{Bmatrix} = [D]\{\varepsilon\} \tag{35}$$

$$\{\varepsilon\} = [L]\{f\} = [L][N]\{\delta\}_e \qquad \{\sigma\} = [L]\{f\} = [D][B]\{\delta\}_e$$

$$[N] = \begin{bmatrix} N_1 & 0 & N_2 & 0 & N_3 & 0 \\ 0 & N_1 & 0 & N_2 & 0 & N_3 \end{bmatrix}, N_i = \frac{1}{2A}(a_i + b_i x + c_i y) \ ,$$

$$[B] = [[B_1][B_2][B_3]] \ , \ [B_i] = \frac{1}{2A}\begin{bmatrix} b_i & 0 \\ 0 & c_i \\ c_i & b_i \end{bmatrix}, \ A = \frac{1}{2}\begin{vmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{vmatrix}, \tag{36}$$

$a_i = x_j y_k - x_k y_j \ , \ b_i = y_j - y_k \ , \ c_i = x_k - x_i$ za

$i = 1,2,3 \qquad j = 2,3,1 \qquad k = 3,1,2$ респективно

Матрица еластичности D помоћу које се успоставља веза између компонената напона и деформације за ортотропна и изотропна тела гласи:

$$[D] = \begin{bmatrix} d_{11} & d_{12} & 0 \\ d_{21} & d_{22} & 0 \\ 0 & 0 & d_{33} \end{bmatrix} \tag{37}$$

Равно стање напона:

$$d_{11} = d_{22} = \frac{E}{1-\nu^2}, d_{12} = d_{21} = \nu d_{11}, d_{33} = \frac{E}{2(1+\nu)} = G \tag{38}$$

Равно стање деформације:

$$d_{11} = d_{22} = \frac{E(1-\nu)}{d}, d_{12} = d_{21} = \frac{\nu E}{d}, d_{33} = \frac{E}{2(1+\nu)} = G \tag{39}$$

Троуагаони елемент назива се троугао са константним деформацијама, из разлога што су чланови матрице B константни па самим тим и компоненте деформације у елементу. Сада дефинише се матрица крутости троугла константе дебљине t:

$$[k]_e = \int_V [B]^T[D]\,[B]dV = t\int_A [B]^T[D]\,[B]dA = tA\,[B]^T[D][B]$$

(40)

Ред матрице крутости троугаоног елемента је 6x6 и она се може декомпоновати на блокове реда 2x2 ради лакшег срачунавања елемената матрице:

$$[k]_e = \begin{bmatrix} [k_{11}]_e & [k_{12}]_e & [k_{13}]_e \\ [k_{21}]_e & [k_{22}]_e & [k_{23}]_e \\ [k_{31}]_e & [k_{32}]_e & [k_{33}]_e \end{bmatrix} \quad , \quad \begin{matrix} [k_{ij}]_e = [k_{ji}]_e^T \text{ za } i \neq j \\ i, j = 1,2,3 \end{matrix} \quad . \tag{41}$$

Сада исписујемо:

$$[k_{ij}]_e = tA[B_i]^T [D][B_j] = \frac{t}{4A} \begin{bmatrix} b_i d_{11} b_j + c_i d_{33} c_j & b_i d_{12} c_j + c_i d_{33} b_j \\ c_i d_{21} b_j + b_i d_{33} c_j & c_i d_{22} c_j + b_i d_{33} b_j \end{bmatrix} \tag{42}$$

Силе у чворовима при спољашњем оптерећењу се одређују преко израза:

$$\{F\}_e = \{F\}_e^i + \{F\}_e^V + \{F\}_e^A = \int_V [B]^T [D] \{\varepsilon\}_0 dV + \int_V [N]^T \{F\} dV + \int_A [N]^T \{\sigma\} dA \tag{43}$$

Вектор $\{F\}_e^i$ представља вектор почетне термичке деформације. Вектор спољшњег запреминског оптерећења елемента се развија у израз:

$$\{F\}_e^V = \int_V [N]^T \{F\} dV = t \int_A \begin{bmatrix} N_1 & 0 \\ 0 & N_1 \\ N_2 & 0 \\ 0 & N_2 \\ N_3 & 0 \\ 0 & N_3 \end{bmatrix} \begin{Bmatrix} F_x \\ F_y \end{Bmatrix} dxdy = \frac{At}{3} \begin{Bmatrix} F_x \\ F_y \\ F_x \\ F_y \\ F_x \\ F_y \end{Bmatrix}. \tag{44}$$

Уколико се координате x и y мере у односу на тежиште троугла $x_c$ и $y_c$ добијамо:

$$\int_A x dxdy = x_C \cdot A \quad i \quad \int_A y dxdy = y_C \cdot A \tag{45}$$

Коначно добија се и вектор спољашњег површинског оптерећења:

$$\{F\}_e^A = \begin{Bmatrix} F_{x1} \\ F_{y1} \\ F_{x2} \\ F_{y2} \\ F_{x3} \\ F_{y3} \end{Bmatrix} = \frac{t}{2} \begin{Bmatrix} L^{12}\sigma_x^{12} + L^{13}\sigma_x^{13} + 0 \\ L^{12}\sigma_y^{12} + L^{13}\sigma_y^{13} + 0 \\ L^{12}\sigma_x^{12} + 0 + L^{23}\sigma_x^{23} \\ L^{12}\sigma_y^{12} + 0 + L^{23}\sigma_y^{23} \\ 0 + L^{13}\sigma_x^{13} + L^{23}\sigma_x^{23} \\ 0 + L^{13}\sigma_y^{13} + L^{23}\sigma_y^{23} \end{Bmatrix}$$

$$\{F\}_e^{A^{12}} = \int_{A^{12}} \begin{bmatrix} N_1 & 0 \\ 0 & N_1 \\ N_2 & 0 \\ 0 & N_2 \\ N_3 & 0 \\ 0 & N_3 \end{bmatrix} \begin{Bmatrix} \sigma_x^{12} \\ \sigma_y^{12} \end{Bmatrix} dA^{12} = \frac{t \cdot L^{12}}{2} \begin{Bmatrix} \sigma_x^{12} \\ \sigma_y^{12} \\ \sigma_x^{12} \\ \sigma_y^{12} \\ 0 \\ 0 \end{Bmatrix} \tag{46}$$

$$\{F\}_e^{A^{13}} = \frac{t \cdot L^{13}}{2} \begin{Bmatrix} \sigma_x^{13} \\ \sigma_y^{13} \\ 0 \\ 0 \\ \sigma_x^{13} \\ \sigma_y^{13} \end{Bmatrix}, \quad \{F\}_e^{A^{23}} = \frac{t \cdot L^{23}}{2} \begin{Bmatrix} 0 \\ 0 \\ \sigma_x^{23} \\ \sigma_y^{23} \\ \sigma_x^{23} \\ \sigma_y^{23} \end{Bmatrix}$$

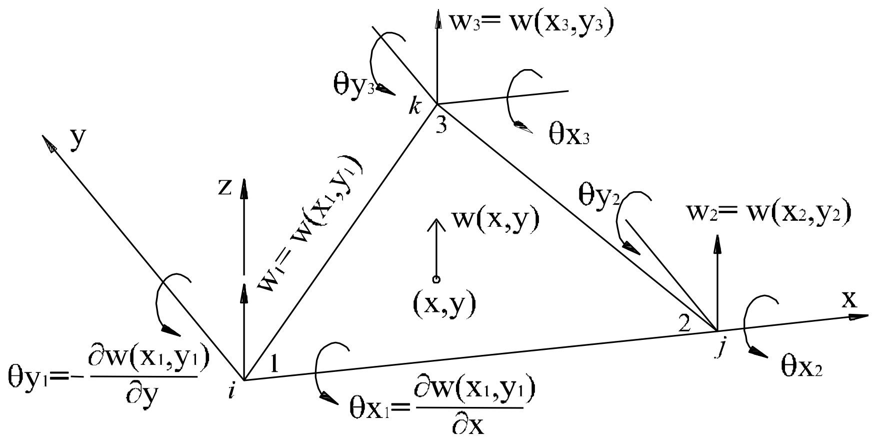
За i, j = 1, 2 кажемо да повезује чвор 1 са чвором 2 и има дужину $L^{12}$ и површину пресека $A^{12} = t\,L^{12}$. По истом принципу се се везују и остали чворови.

Из приложеног се види да је троугаони елемент са чворовима у теменима погодан и једноставан за примену. Промена померања је линеарна, а напон и деформација константи. Такође, све дефинисане величине не зависе од транслације координатоног система. Следи, при раванском проблему дефинише се само заједнички координати систем, уколико се равански елемент примењује при решавању запреминског проблема потребно је локални координатни систем пребацити у глобални.

### 4.3.2 Троугаони коначни елемент танке плоче напрегнут на савијање

Троугаони елементи су јако погодни за решавање проблематике савијања плоча. Наједноставнији облик је троугао са чворовима у теменима, при чему свака тачка поседује три степена слободе, што укупно чини 9 степени слободе. Степене слободе идентификујемо као попречно померање и ротације око оса које леже у равни плоче, респективно ћемо их означити са w, $\theta_x$ i $\theta_y$. Троугаони елемент и степени слободе су приказани на слици 4.3.5. Поље померања сада има 9 степени слободе и може се приказати са 9 независних функција, односно нешто компликованијим полиномом трећег реда са 9 чланова. Постоје неколико предлога за формулисање полинома. У раду је коришћен Тохеров полином T-9 (Tocher).

$$w(x,y)= \alpha_1 + \alpha_2 x + \alpha_3 y + \alpha_4 x^2 + \alpha_5 xy + \alpha_6 y^2 + \alpha_7 x^3 + \alpha_8 ( x^2 y+xy^2 ) + \alpha_9 y^3$$




Слика 4.3.5. Троугаони елемент са степенима слободе

Изрази за ротације тачке су:

$$\theta x = \frac{\partial w}{\partial y} \qquad \theta y = -\frac{\partial w}{\partial x} \tag{47}$$

Померања тачке елемента изражавамо у следећем облику:

$$\{f(x,y)\} = \begin{Bmatrix} w(x,y) \\ \theta_x(x,y) \\ \theta_y(x,y) \end{Bmatrix} = \begin{Bmatrix} w(x,y) \\ \partial w(x,y)/\partial y \\ -\partial w(x,y)/\partial x \end{Bmatrix} = [n]\cdot\{a\} \tag{48}$$

$$[n] = \begin{bmatrix} 1 & x & y & x^2 & xy & y^2 & x^3 & (x^2 y + xy^2) & y^3 \\ 0 & 0 & 1 & 0 & x & 2y & 0 & (x^2 + 2xy) & 3y^2 \\ 0 & -1 & 0 & -2x & -y & 0 & -3x^2 & -(2xy + y^2) & 0 \end{bmatrix} \tag{49}$$

$$\{a\} = \{a_1 \ a_2 \ a_3 \ a_4 \ a_5 \ a_6 \ a_7 \ a_8 \ a_9\}^T \ .$$

Затим преко вектора деформације и померања може се извести матрица диференцијалних оператора:

$$\{\varepsilon\}=\begin{Bmatrix}-z\dfrac{\partial^2 w}{\partial x^2}\\ -z\dfrac{\partial^2 w}{\partial y^2}\\ -2z\dfrac{\partial^2 w}{\partial x\partial y}\end{Bmatrix}=[L]\cdot\{f\}=-z\cdot\begin{bmatrix}\dfrac{\partial}{\partial x^2} & 0 & 0\\ 0 & \dfrac{\partial}{\partial y} & 0\\ 0 & \dfrac{\partial}{\partial x} & -\dfrac{\partial}{\partial y}\end{bmatrix}\begin{Bmatrix}w\\ \dfrac{\partial w}{\partial y}\\ -\dfrac{\partial w}{\partial x}\end{Bmatrix} \tag{50}$$

Величине степени слободе тачака троугла у локалном координатном систему гласе:

$$\{\delta\}_e=\begin{Bmatrix}w_1\\ \theta_{x1}\\ \theta_{y1}\\ w_2\\ \theta_{x2}\\ \theta_{y2}\\ w_3\\ \theta_{x3}\\ \theta_{y3}\end{Bmatrix}_e=\begin{Bmatrix}w(x_1=0,y_1=0)\\ \partial w(x_1=0,y_1=0)/\partial y\\ -\partial w(x_1=0,y_1=0)/\partial x\\ w(x_2,y_2=0)\\ \partial w(x_2,y_2=0)/\partial y\\ -\partial w(x_2,y_2=0)/\partial x\\ w(x_3,y_3)\\ \partial w(x_3,y_3)/\partial y\\ -\partial w(x_3,y_3)/\partial x\end{Bmatrix}_e=[N]\{a\} \tag{51}$$

$$[N]=\begin{bmatrix}1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\ 1 & x_2 & 0 & x_2^2 & 0 & 0 & x_2^3 & 0 & 0\\ 0 & 0 & 1 & 0 & x_2 & 0 & 0 & x_2^2 & 0\\ 0 & -1 & 0 & -2x_2 & 0 & 0 & 3x_2^2 & 0 & 0\\ 1 & x_3 & y_3 & x_3^2 & x_3y_3 & y_3^2 & x_3^3 & (x_3^2y_3+x_3y_3^2) & y_3^3\\ 0 & 0 & 1 & 0 & x_3 & 2y_3 & 0 & (x_3^2+2x_3y_3) & 3y_3^2\\ 0 & -1 & 0 & -2x_3 & -y_3 & 0 & -3x_3^2 & -(2x_3y_3+y_3^2) & 0\end{bmatrix} \tag{52}$$

Матрица извода се приказује путем следећих производа:

$$[\underline{B}]=[L]\cdot[n]=-z\cdot\begin{bmatrix}0&0&0&2&0&0&6x&2y&0\\0&0&0&0&0&2&0&2x&6y\\0&0&0&0&2&0&0&4(x+y)&0\end{bmatrix} \tag{53}$$

Матрица везе је:

$$[B]=[\underline{B}]\cdot[\underline{N}]^{-1} \quad , \quad [N]=[n][\underline{N}]^{-1} \tag{54}$$

Сада можемо исписати израз за одређивање матрице крутости у локалном координатном систему троугаоног елемента:

$$[k]_e=\int_V[B]^T[D]\cdot[B]dV=([\underline{N}]^{-1})^T\left\{\int_A dA(\int_{-t/2}^{t/2}[\underline{B}]^T[D]\cdot[\underline{B}]dz)\right\}[\underline{N}]^{-1} \tag{55}$$

Видимо да матрица [N] не зависи од подинтегралних величина, веч само од координата тачака. Решавајући интеграл по површини и дебљини добија се:

$$\int_A dA\int_{-t/2}^{t/2}[\underline{B}]^T[D][\underline{B}]\cdot dz \tag{56}$$

$$\frac{Et^3}{12(1-\nu^2)}\int_A dx\cdot dy\begin{bmatrix}0&&&&&&&&\\0&0&&&&&&&\\0&0&0&&&&&&\\0&0&0&4&&&\text{симетрично}&&\\0&0&0&0&2(1-\nu)&&&&\\0&0&0&4\nu&0&4&&&\\0&0&0&12x&0&12\nu x&36x^2&&\\0&0&0&4\cdot(\nu x+y)&4(1-\nu)\cdot(x+y)&4\cdot(x+\nu y)&12x\cdot(\nu x+y)&(12-8\nu)\cdot(x+y)^2-8xy(1-\nu)&\\0&0&0&12\nu&0&12y&36\nu xy&12y(x+\nu y)&36y^2\end{bmatrix}$$

Вредности интеграла у локалном координатном систему се добијају преко израза:

$$\int_A dxdy=A=x_2y_3/2 \qquad \int_A xdxdy=x_2y_3(x_2+x_3)/6$$

$$\int_A ydxdy=x_2y_3^2/6 \qquad \int_A x^2dxdy=x_2y_3(x_2^2+x_2x_3+x_3^2)/12$$

$$\int_A xydxdy = x_2 y_3^3/12 \qquad \int_A y^2 dxdy = x_2 y_3^3/12 \tag{57}$$

Сада су познати сви параметри за израчунавање матрице крутости која је реда 9 x 9 и има три подматрице, односно једна подматрица по чвору реда 3 x 3. Овако дефинисани коначни елементи су веома повољни за имплементацију у прорачун плоча, тј. у нашем случају набораних конструкција. На овим основама у које се касније имплементира групно суперматрична процедура базира се прорачун напона, деформација и сопствених фреквенци генерисаних примитива.

# 5. ГРУПНО СУПЕРМАТРИЧНА ПРОЦЕДУРА (ГСП)

## 5.1 Уводна разматрања

Током друге половине прошлог века, у периоду док дигиталне технологије нису имале значајног утицаја на процес пројектовања и реализације, академик Ђорђе Злоковић радио је на примени теорије група и Г-векторских простора у прорачуну осцилација, стабилности и статике конструкција. Својим радом недвосмислено је представио и доказао огромне потенцијале и вредности практичне примене математичког апарата из домене теорије група и Г-векторских простора за решавање проблема теорије конструкција.

Основи проблем сваке конструкције представља њена математичка формулација и креирање одговарајућег модела. Самим тим, развој геометрије као научне дисциплине у многоме утиче и на развој архитектонске мисли у функцији њене реализације. Моделирање конструкције представља стварање идеализоване или поједностављене репрезентације понашања конструкције за неко дејство. Посматрајући конструкцију на овај начин, можемо је представити као просте или сложене планове тачака са прецизно дефинисаним вредностима њихових растојања. Математичке формулације узимају за почетни положај решетку формирану од системских тачака. Затим се приступа, помоћу теорије група, разматрању особина симетрије тако образоване решетке. Наиме, просторни објекат, односно конструкција може да садржи идентичне тачке, тј. чворове који заузимају физички идентичне положаје у чворној решетки. Ротацијом око неке осе чворне решетке могуће је довести чворну решетку у нови положај који се у потпуности поклапа са почетним положајем и физички се од њега не разликује. Потпуно исто може да се постигне рефлексијом чворне решетке или комбинацијом ротација и рефлексија. Из тога се закључује да операција симетрије не мења ни једну физичку особину конструкције, као и на почетно стање у погледу сила и померања. Применом репрезентације група остварује се значајно упрошћење и смањење обима целокупног прорачуна. Овакав алат изискује мултидисциплинаран приступ проблему. Употреба рачунарске технике је данас незаобилазно средство при решавању различитих инжењерских проблема. Размена знања и примена различитих метода, процеса и процедура других, сродних дисциплина постали су незаобилазан фактор при имплементацији и реализацију архитектонских идеја.

Интеграцијом и применом знања различитих струка инсистира се на схватању концепта објекта као рационалне, комплексне структуре способне да задовољи инжењерске критеријуме и архитектонске захтеве остварујући сложену интеракцију форме, функције и конструкције.

Академик Ђорђе Злоковић крајем 20-тог века је у својим књигама (Zloković G. , 1992), (Zloković G. , 1989) представио концепт примене групних суперматрица у формулацији функције облика и израза крутости на коначном елементу који поседује особине симетричности. У поређењу са конвенционалним приступом, дефинисање проблема путем репрезентације група у анализа коначног елемента остварује значајну предност у погледу квалитативних и квантитативних вредности. Концепт облика функције у Г-инваријантном подпростору одликује се битним смањењем укупног броја једначина при прорачуну. Она омогућава максимално искоришћење особина симетрије, што доводи до мањег броја математичких израза које је потребно решити. Сама процедура се лако имплементира у метод коначних елемената и примењива је за решавање свих врста проблема. (статичка и динамичка анализа, линерани и нелинеарне теорије и сл.). При изради овог поглавља додатно је коришћена литература (Mathiak & Stingl, 1967), (Zloković Đ. , 1973), (Zloković, Maneski, & Nestorović, 1993), (Zloković, Maneski, & Nestorović, 1995), (Zloković, Maneski, & Nestorović, 1994), (Zloković, Maneski, & Nestorović, 1998), (Zloković, Maneski, & Nestorović, 1999), (Maneski, 1998)

Значи, уколико коначни елемент са чворном структуром поседује одређену симетрију која се може приписати групи, групно суперматрична процедура може обезбедити формулацију са максималним искоришћењем симетрије. Примена особина симетрије путем групно суперматричне процедуре при анализи коначног елемента поседује много предности у односу на конвенционално искоришћавање особине симетричности. Предности произилазе из декомпозиције Г-векторскох простора система у Г-инваријантни подпростор. За разлику од конвенционалне директне методе одређивања крутости елемената, групно суперматрична процедура примењује другачију конвенцију при нумерисању чворова, позиционирању првобитних положаја и праваца оса локалног и глобалног координатног система. Такође, поставке позитивних смерова генерализованих померања су адаптирани тако да се одговарају типовима симетрије прве несводљиве репрезентације припадајуће групе. Са овако дефинисаним почетним параметрима, могуће је извршити формулацију система једначина крутости применом групно суперматричне процедуре у нормалном и дијагоналном облику, као и да се изведу изрази за крутост система у Г-инваријантном подпростору. Групно суперматрична процедура примењена на директан метод одређивања чврстоће обухвата следеће кораке:

- Одређивање групе симетрије конструкције
- Нумерисање чворова и њихових промењивих
- Одређивање базичног вектора Г-инваријантног подпростора
- Примена групних суперматрица изабране групе
- Успостављање релација између позитивном смера генералисаног померања чвора са одговарајућим померањем у првом подпростору
- Формулација матрице чврстоће елемента
- Формулација матрице чврстоће система са групним суперматрицама у нормалној и дијагоналној форми.

Са овако постављеним теоријским основама представљен је групно суперматрични поступак за нумеричко описивање система са особинама симетрије у зависности од групе симетрије коју елемент поседује. Решавањем независних система једначина у појединим Г – инваријантним подпросторима добијају се резултати чија линеарна комбинација даје тражене величине. Добијање групне суперматрице у нормалном облику врши се помоћу таблица карактера групе у суперматричном облику. Суперматрица карактера је ортогонална. Њеним транспоновањем добија се инверзна матрица, на основу које се изводе изрази суперматартице у дијагоналном облику. Из таблице карактера изводи се суперматрица T, преко које се групна суперматрица у нормалном облику трансформише у групну суперматрицу у дијагоналном облику.

Устаљен начин употребе симетрије најчешће подразумева сагледавање једне равни симетрије, при чему се спољни утицаји деле на симетричне и антисиметричне, што омогућава да се анализа изврши на једној половини конструкције. Суперпозицијом резултата симетрично и антисиметрично оптерећење конструкције добијају се резултати несиметрично оптерећење.

Према томе, директном суперпозицијом матрице крутости елемента, поштујући поставке које одговарају типу симетрије групе, може се формирати матрица крутости целокупног система. Применом овакве прорачунске процедуре из израза за крутост система проналази се тражено решење за сваки од Г-инваријантних подпростора, док се коначно решење добија као линеарна комбинација решења подпростора, што се постиже трансформацијом суперматрице.

Уколико геометрија конструкције поседује особину симетрије која се може описати неком групом G, онда једначина крутости гласи:

$$K\,\Phi = F \tag{58}$$

K – групна суперматрица у нормалном облику

Φ – генералисана померања

F – генералисана оптерећења

Затим, трансформацијом горње једначине преводимо у:

$$T\,K\,T^{-1}\,T\,\Phi = T\,F \tag{59}$$

T – суперматрица трансформације.

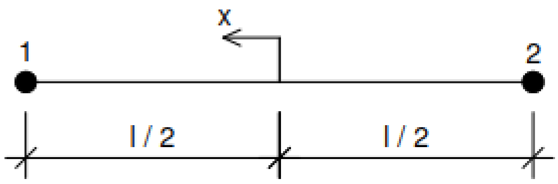
Сада формирамо групну суперматрицу крутости у дијагоналном облику, успостављајући везу између скупа базисних вектора и чворних функција која гласи:

$$\overline{K}\,\overline{\Phi} = \overline{F} \quad \text{ili} \quad \begin{bmatrix} \overline{K}_1 & & & \\ & \overline{K}_2 & & \\ & & \ddots & \\ & & & \overline{K}_k \end{bmatrix} \begin{bmatrix} \overline{\Phi}^{(1)} \\ \overline{\Phi}^{(2)} \\ \vdots \\ \overline{\Phi}^{(k)} \end{bmatrix} = \begin{bmatrix} \overline{F}^{(1)} \\ \overline{F}^{(2)} \\ \vdots \\ \overline{F}^{(k)} \end{bmatrix}, \quad \begin{aligned} \overline{K} &= TKT^{-1} \\ \overline{F} &= TF \\ \Phi &= kT\overline{\Phi} \end{aligned} . \tag{60}$$

Овако описана глобална једначина крутости поседује к независних скупова једначина крутости. Сваки скуп крутости је дефинисан у свом Г – инваријантном подпростору $U_i$ заједно са генералисаним померањима и оптерећењима. Свака од група G поседује свој тип симетрије, свог подпростора. За потребе овог рада користиће се групне суперматрице са једном и две равни симетрије које се обележавају са $C_2$ i $C_{2v}$.

### 5.2 Групне суперматрице и матрица крутости линијског елемента ($C_2$)

Узмимо линијски елемент као на слици 5.2.1. Позиција првог чвора, је нумерисана са 1 и налази се на позитивној координати x осе, јединственим нумерисањем чворова по правилима трансформације групних суперматрица, применом операција симетрије E (идентитет) и ротације од 180° око z осе означену са $C_2$, на почетни чвор 1, добијамо скуп чворова 1, 2 у овом низу.




Слика 5.2.1. Линијски елемент за два чвора – група $C_2$

Јединствена нумерација чворова са слике 5.2.1 може се и табеларно приказати у следећој форми:

**Табела 5.2.1.** Приказ нумерације чворова линијског елемента

| чвор | x оса |
|---|---|
| 1 | + |
| 2 | - |

Таблица карактера, скупова и продукта групе $C_2$ је приказана у табели:

**Табела 5.2.2.** Приказ карактера, скупова и продукта групе $C_2$

| $C_2$ | E | $C_2$ | | | |
|---|---|---|---|---|---|
| A | 1 | 1 | z | 1 , $x_2$ , $y_2$ , $z_2$ | $U_1$ |
| B | 1 | -1 | x , y | $y_z$ , $x_z$ | $U_2$ |

Резултат операција симетрије на почетном низу [1,2] се приказује кроз изразе:

$$E = \begin{bmatrix} 1 & 2 \\ 1 & 2 \end{bmatrix} \qquad C_2 = \begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix} \tag{61}$$

Применом операције симетрије, поштујући редослед и правила операција у складу са таблицом карактера, на скупове чворова добијају се њихове пермутације.

Заменом бројева скупа чворова 1, 2 матрицама A, B добија се 2 x 2 групна суперматрица G у нормалном облику:

$$G = \begin{bmatrix} A & B \\ B & A \end{bmatrix} \tag{62}$$

Идемпотенти $\pi_i$ (i = 1 , 2) центра групе се изражавају преко израза:

$$\begin{bmatrix} \pi_1 \\ \pi_2 \end{bmatrix} = \frac{1}{2}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}\begin{bmatrix} E \\ C_2 \end{bmatrix} \qquad \text{или} \qquad \Pi = T\,\Sigma \tag{63}$$

Где је матрица трансформације

$$T = \frac{1}{2}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \tag{64}$$

Уколико јединичне вредности матрице T заменимо n x n јединичном матрицом E, поменути израза може се представити и као:

$$T = \frac{1}{2}\begin{bmatrix} E & E \\ E & -E \end{bmatrix} \tag{65}$$

Матрица T представља матрицу трансформације и њена инверзна матрица је :

$$T^{-1} = 2\,T \tag{66}$$

И сада идемпотенти $\pi i$ (i = 1 , 2) се изражавају као:

$$\begin{bmatrix} \pi_1 \\ \pi_2 \end{bmatrix} = \frac{1}{2}\begin{bmatrix} E & E \\ E & -E \end{bmatrix}\begin{bmatrix} E \\ C_2 \end{bmatrix} \tag{67}$$

Уколико сада применимо идемпотенте на скупове чворних функција $\phi$ , добијамо скупове базисних вектора G инваријантних подпростора $U_1$, $U_2$ које можемо приказати преко следећих израза:

$$\begin{bmatrix} U_1 \\ U_2 \end{bmatrix} = \begin{bmatrix} \Phi^{(1)} \\ \Phi^{(2)} \end{bmatrix} = \begin{bmatrix} \phi_1^{(1)} & \phi_2^{(1)} & \cdots & \phi_n^{(1)} \\ \phi_1^{(2)} & \phi_2^{(2)} & \cdots & \phi_n^{(2)} \end{bmatrix} = \begin{bmatrix} \overline{\Phi}^{(1)} + \overline{\Phi}^{(2)} \\ \overline{\Phi}^{(1)} - \overline{\Phi}^{(2)} \end{bmatrix} = 2T\overline{\Phi} \tag{68}$$

Односно G базисних инваријантних подпростора $U_1$, $U_2$:

$$\begin{bmatrix} \overline{U}_1 \\ \overline{U}_2 \end{bmatrix} = \begin{bmatrix} \overline{\Phi}^{(1)} \\ \overline{\Phi}^{(2)} \end{bmatrix} = \begin{bmatrix} \overline{\phi}_1^{(1)} & \overline{\phi}_2^{(1)} & \cdots & \overline{\phi}_n^{(1)} \\ \overline{\phi}_1^{(2)} & \overline{\phi}_2^{(2)} & \cdots & \overline{\phi}_n^{(2)} \end{bmatrix} = \frac{1}{2}\begin{bmatrix} \Phi^{(1)} + \Phi^{(2)} \\ \Phi^{(1)} - \Phi^{(2)} \end{bmatrix} = T\Phi \tag{69}$$

Затим, једначину крутости К Ф = F са групном суперматрицом крутости у нормалном облику применом групног суперматричног поступка преводимо у следеће изразе:

$$K\Phi = \begin{bmatrix} A & B \\ B & A \end{bmatrix} \begin{bmatrix} \Phi^{(1)} \\ \Phi^{(2)} \end{bmatrix} = \begin{bmatrix} F^{(1)} \\ F^{(2)} \end{bmatrix} = F \ , \ \begin{matrix} TKT^{-1}\, T\Phi = TF \\ \overline{K}\overline{\Phi} = \overline{F} \ , T^{-1} = 2F \end{matrix} , \tag{70}$$

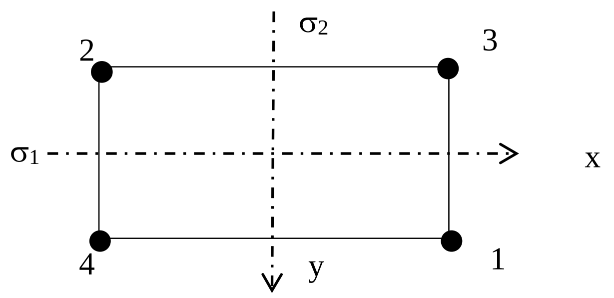
$$\begin{bmatrix} A+B & 0 \\ 0 & A-B \end{bmatrix} \begin{bmatrix} \overline{\Phi}^{(1)} \\ \overline{\Phi}^{(2)} \end{bmatrix} = \begin{bmatrix} \overline{F}^{(1)} \\ \overline{F}^{(2)} \end{bmatrix} \ , \ \begin{bmatrix} \overline{F}^{(1)} \\ \overline{F}^{(2)} \end{bmatrix} = \frac{1}{2} \begin{bmatrix} F^{(1)} + F^{(2)} \\ F^{(1)} - F^{(2)} \end{bmatrix} \tag{71}$$

Матрица крутости је сада пресликана у две квазидијагоналне матрице, односно постигнуто је да се систем једначина решава два пута уз помоћ дупло мање једначина, што значи да је матрица мање димензије и потребно време компутације једначина је вишеструко краће. Померања се добијају решавањем генералисаних померања по подпросторима и гласи:

$$\begin{bmatrix} \Phi^{(1)} \\ \Phi^{(2)} \end{bmatrix} = \begin{bmatrix} \overline{\Phi}^{(1)} + \overline{\Phi}^{(2)} \\ \overline{\Phi}^{(1)} - \overline{\Phi}^{(2)} \end{bmatrix} \tag{72}$$

## 5.3 Групне суперматрице и матрица крутости правоугаоног елемента ($C_{2v}$)

Посматрајмо правоугаони елемент на слици 5.3.1. Позиција првог чвора, је нумерисана са 1 и налази се на позитивној координати x и y осе, јединственим нумерисањем чворова по правилима трансформације групних суперматрица, применом операција симетрије E (идентитет) и ротације од 180° око z осе означену са $C_2$, затим рефлексије $\sigma_1$ и $\sigma_2$ респективно у равнима xz и yz, на почетни чвор 1, добијамо скуп чворова 1, 2, 3, 4 у овом низу.




Слика 5.3.1. Правоугаони елемент са четири чвора – група $C_{2v}$

Јединствена нумерација чворова са слике 5.3.1 може се и Табеларно приказати у следећој форми:

**Табела 5.3.1.** Приказ нумерације чворова површинског елемнта

| чвор | x оса | y оса |
|---|---|---|
| 1 | + | + |
| 2 | - | - |
| 3 | + | - |
| 4 | - | + |

Таблица карактера, скупова и продукта групе $C_{2v}$ је приказана у табели:

**Табела 5.3.2.** Приказ карактера, скупова и продукта групе $C_{2v}$

| $C_{2v}$ | E | $C_2$ | $\sigma_1$ (xz) | $\sigma_2$ (yz) | | | |
|---|---|---|---|---|---|---|---|
| $A_1$ | 1 | 1 | 1 | 1 | z | 1 , $x_2$ , $y_2$ , $z_2$ | $U_1$ |
| $A_2$ | 1 | 1 | -1 | -1 | | $x_y$ | $U_2$ |
| $B_1$ | 1 | -1 | 1 | -1 | x | $x_z$ | $U_3$ |
| $B_2$ | 1 | -1 | -1 | 1 | y | $y_z$ | $U_4$ |

Изражен преко пермутације чворова, резултат операција симетрије на почетном низу [1,2] се приказује кроз изразе:

$$E = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 \end{bmatrix} \qquad C_2 = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 2 & 1 & 4 & 3 \end{bmatrix}$$

$$\sigma_1 = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 3 & 4 & 1 & 2 \end{bmatrix} \qquad \sigma_1 = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 4 & 3 & 2 & 1 \end{bmatrix} \tag{73}$$

Применом операције симетрије, поштујући редослед и правила операција у складу са таблицом карактера, на скупове чворова добијају се њихове пермутације. Заменом бројева скупа чворова 1, 2, 3, 4 матрицама A, B, C, D добија се 4 x 4 групна суперматрица G у нормалном облику:

$$G = \begin{bmatrix} A & B & C & D \\ B & A & D & C \\ C & D & A & B \\ D & C & B & A \end{bmatrix} \tag{74}$$

Идемпотенти $\pi_i$ (i = 1 , 2, 3, 4) центра групе се изражавају преко израза:

$$G=\begin{bmatrix} A & B & C & D \\ B & A & D & C \\ C & D & A & B \\ D & C & B & A \end{bmatrix}, \quad \begin{bmatrix} \pi_1 \\ \pi_2 \\ \pi_3 \\ \pi_4 \end{bmatrix} = \frac{1}{4}\begin{bmatrix} E & E & E & E \\ E & E & -E & -E \\ E & -E & E & -E \\ E & -E & -E & E \end{bmatrix} \qquad \text{или} \qquad \Pi = T\,\Sigma \tag{75}$$

Где је матрица трансформације:

$$T = \frac{1}{4}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \tag{76}$$

Уколико јединичне вредности матрице T заменимо n x n јединичном матрицом E, поменути израза може се представити и као:

$$T = \frac{1}{4}\begin{bmatrix} E & E & E & E \\ E & E & -E & -E \\ E & -E & E & -E \\ E & -E & -E & E \end{bmatrix} \tag{77}$$

Матрица T представља матрицу трансформације и њена инверзна матрица је :

$$T^{-1}=2T \tag{78}$$

Уколико сада применимо идемпотенте на скупове чворних функција $\phi$ , добијамо скупове базисних вектора G инваријантних подпростора $U_1$, $U_2$, $U_3$, $U_4$ које можемо приказати преко следећих израза:

$$\begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{bmatrix} = \begin{bmatrix} \phi_1^{(1)} & \cdots & \phi_n^{(1)} \\ \phi_1^{(2)} & \cdots & \phi_n^{(2)} \\ \phi_1^{(3)} & \cdots & \phi_n^{(3)} \\ \phi_1^{(4)} & \cdots & \phi_n^{(4)} \end{bmatrix} = \begin{bmatrix} \overline{\Phi}^{(1)} + \overline{\Phi}^{(2)} + \overline{\Phi}^{(3)} + \overline{\Phi}^{(4)} \\ \overline{\Phi}^{(1)} + \overline{\Phi}^{(2)} - \overline{\Phi}^{(3)} - \overline{\Phi}^{(4)} \\ \overline{\Phi}^{(1)} - \overline{\Phi}^{(2)} + \overline{\Phi}^{(3)} - \overline{\Phi}^{(4)} \\ \overline{\Phi}^{(1)} - \overline{\Phi}^{(2)} - \overline{\Phi}^{(3)} + \overline{\Phi}^{(4)} \end{bmatrix} = 4T\overline{\Phi} \tag{79}$$

Односно G базисних инваријантних подпростора $U_1$, $U_2$, $U_3$, $U_4$:

$$\begin{bmatrix} \overline{U}_1 \\ \overline{U}_2 \\ \overline{U}_3 \\ \overline{U}_4 \end{bmatrix} = \begin{bmatrix} \overline{\phi}_1^{(1)} & \cdots & \overline{\phi}_n^{(1)} \\ \overline{\phi}_1^{(2)} & \cdots & \overline{\phi}_n^{(2)} \\ \overline{\phi}_1^{(3)} & \cdots & \overline{\phi}_n^{(3)} \\ \overline{\phi}_1^{(4)} & \cdots & \overline{\phi}_n^{(4)} \end{bmatrix} = \frac{1}{4}\begin{bmatrix} \Phi^{(1)} + \Phi^{(2)} + \Phi^{(3)} + \Phi^{(4)} \\ \Phi^{(1)} + \Phi^{(2)} - \Phi^{(3)} - \Phi^{(4)} \\ \Phi^{(1)} - \Phi^{(2)} + \Phi^{(3)} - \Phi^{(4)} \\ \Phi^{(1)} - \Phi^{(2)} - \Phi^{(3)} + \Phi^{(4)} \end{bmatrix} = T\Phi \tag{80}$$

Сада једначину крутости K Ф = F са групном суперматрицом крутости у нормалном облику применом групног суперматричног поступка преводимо у следеће изразе:

$$K\Phi = \begin{bmatrix} A & B & C & D \\ B & A & D & C \\ C & D & A & B \\ D & C & B & A \end{bmatrix} \begin{bmatrix} \Phi^{(1)} \\ \Phi^{(2)} \\ \Phi^{(3)} \\ \Phi^{(4)} \end{bmatrix} = \begin{bmatrix} F^{(1)} \\ F^{(2)} \\ F^{(3)} \\ F^{(4)} \end{bmatrix} = F \ , \ \frac{TKT^{-1}\ T\Phi = TF}{\overline{K\Phi} = \overline{F}\ ,\ T^{-1} = 4F} , \tag{81}$$

$$\text{dijag} \begin{bmatrix} A+B+C+D \\ A+B-C-D \\ A-B+C-D \\ A-B-C+D \end{bmatrix} \begin{bmatrix} \overline{\Phi}^{(1)} \\ \overline{\Phi}^{(2)} \\ \overline{\Phi}^{(3)} \\ \overline{\Phi}^{(4)} \end{bmatrix} = \begin{bmatrix} \overline{F}^{(1)} \\ \overline{F}^{(2)} \\ \overline{F}^{(3)} \\ \overline{F}^{(4)} \end{bmatrix} \tag{82}$$

$$\begin{bmatrix} \overline{F}^{(1)} \\ \overline{F}^{(2)} \\ \overline{F}^{(3)} \\ \overline{F}^{(4)} \end{bmatrix} = \frac{1}{4} \begin{bmatrix} F^{(1)} + F^{(2)} + F^{(3)} + F^{(4)} \\ F^{(1)} + F^{(2)} - F^{(3)} - F^{(4)} \\ F^{(1)} - F^{(2)} + F^{(3)} - F^{(4)} \\ F^{(1)} - F^{(2)} - F^{(3)} + F^{(4)} \end{bmatrix} \tag{83}$$

Матрица крутости је сада пресликана у четири квазидијагоналне матрице, односно постигнуто је да се систем једначина решава четири пута уз помоћ четири пута мање једначина, што опет значи да је матрица мање димензије и потребно време компутације једначина је вишеструко краће. Померања се добијају решавањем генералисаних померања по подпросторима и гласе:

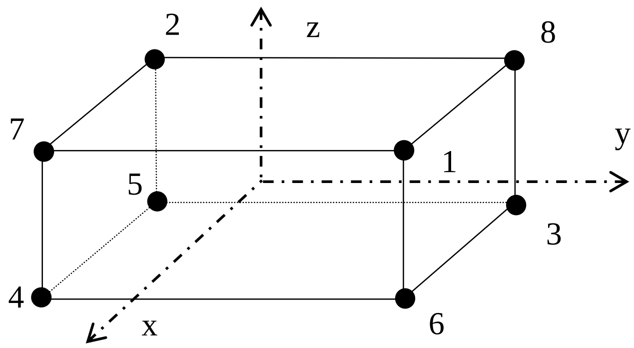
$$\begin{bmatrix} \Phi^{(1)} \\ \Phi^{(2)} \\ \Phi^{(3)} \\ \Phi^{(4)} \end{bmatrix} = \begin{bmatrix} \overline{\Phi}^{(1)} + \overline{\Phi}^{(2)} + \overline{\Phi}^{(3)} + \overline{\Phi}^{(4)} \\ \overline{\Phi}^{(1)} + \overline{\Phi}^{(2)} - \overline{\Phi}^{(3)} - \overline{\Phi}^{(4)} \\ \overline{\Phi}^{(1)} - \overline{\Phi}^{(2)} + \overline{\Phi}^{(3)} - \overline{\Phi}^{(4)} \\ \overline{\Phi}^{(1)} - \overline{\Phi}^{(2)} - \overline{\Phi}^{(3)} + \overline{\Phi}^{(4)} \end{bmatrix} \tag{84}$$

По истом принципу могу се приказати и остале групе симетрије.

### 5.4 Групне суперматрице и матрица крутости просторног елемента ($D_{2h}$)

Посматрајмо правоугаони елемент на слици 5.4.1. Позиција првог чвора, је нумерисана са 1 и налази се на позитивној координати x, s и z осе, јединственим нумерисањем чворова по правилима трансформације групних суперматрица, применом операција симетрије E (идентитет) и ротација $C_{2z}$, $C_{2y}$, $C_{2x}$ од 180°

респективно око оса z, y, и x, затим рефлексије $\sigma_{xy}$, $\sigma_{xz}$ и $\sigma_{yz}$ респективно у равнима xy, xz и yz, на почетни чвор 1, добијамо скуп чворова 1, 2, 3, 4, 5, 6, 7, 8 у овом низу.




Слика 5.4.1. Правоугаони хексаедарски елемент са осам чворова – група $D_{2h}$

Јединствена нумерација чворова са слике 5.4.1 може се и Табеларно приказати у следећој форми:

**Табела 5.4.1.** Приказ нумерације чворова запреминског елемента

| чвор | x оса | y оса | z оса |
|---|---|---|---|
| 1 | + | + | + |
| 2 | - | - | + |
| 3 | - | + | - |
| 4 | + | - | - |
| 5 | - | - | - |
| 6 | + | + | - |
| 7 | + | - | + |
| 8 | - | + | + |

Таблица карактера, скупова и продукта групе $D_{2h}$ је приказана у табели:

**Табела 5.4.2.** Приказ карактера, скупова и продукта групе $D_{2h}$

| $D_{2v}$ | E | $C_{2z}$ | $C_{2y}$ | $C_{2x}$ | i | $\sigma_{xy}$ | $\sigma_{xz}$ | $\sigma_{yz}$ | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $A_g$ | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | | $1, x_2, y_2, z_2$ | $U_1$ |
| $B_{1g}$ | 1 | 1 | -1 | -1 | 1 | 1 | -1 | -1 | | xy | $U_2$ |
| $B_{2g}$ | 1 | -1 | 1 | -1 | 1 | -1 | 1 | -1 | | xz | $U_3$ |
| $B_{3g}$ | 1 | -1 | -1 | 1 | 1 | -1 | -1 | 1 | | yz | $U_4$ |
| $A_u$ | 1 | 1 | 1 | 1 | -1 | -1 | -1 | -1 | xyz | | $U_5$ |
| $B_{1u}$ | 1 | 1 | -1 | -1 | -1 | -1 | 1 | 1 | z | | $U_6$ |
| $B_{2u}$ | 1 | -1 | 1 | -1 | -1 | 1 | -1 | 1 | y | | $U_7$ |
| $B_{3u}$ | 1 | -1 | -1 | 1 | -1 | 1 | 1 | -1 | x | | $U_8$ |

Изражен преко пермутације чворова, резултат операција симетрије на почетном низу [1,2] се приказује кроз изразе:

$$
\begin{aligned}
&\mathrm{E} = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \end{bmatrix} \qquad && \mathrm{i} = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 5 & 6 & 7 & 8 & 1 & 2 & 3 & 4 \end{bmatrix} \\
&\mathrm{C}_{2z} = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 2 & 1 & 4 & 3 & 6 & 5 & 8 & 7 \end{bmatrix} && \sigma_{xy} = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 6 & 5 & 8 & 7 & 2 & 1 & 4 & 3 \end{bmatrix} \\
&\mathrm{C}_{2y} = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 3 & 4 & 1 & 2 & 7 & 8 & 5 & 6 \end{bmatrix} && \sigma_{xz} = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 7 & 8 & 5 & 6 & 3 & 4 & 1 & 2 \end{bmatrix} \\
&\mathrm{C}_{2x} = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 4 & 3 & 2 & 1 & 8 & 7 & 6 & 5 \end{bmatrix} && \sigma_{yz} = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 8 & 7 & 6 & 5 & 4 & 3 & 2 & 1 \end{bmatrix}
\end{aligned}
\tag{85}
$$

Применом операције симетрије, поштујући редослед и правила операција у складу са таблицом карактера ове групе за добијање матрице крутости идентичан је као код осталих група, па се аналогно може добити.

Као што смо показали применом групно суперматричне процедуре драстично се смањује укупан број неопходних математичких операција, што се огледа у чињеници смањења димензије матрица добијених применом групно суперматричне процедуре. Може се закључити да примена методе групно суперматричне процедуре при анализи коначног елемента омогућава најоптималније квалитативне и квантитативне предности уколико је посматрани коначни елемент поседује комплексну симетрију, при чему је могуће извршити декомпозицију на велики број одговарајућих подпростора.

# 6. РЕДУКОВАНИ ПОВРШИНСКИ МАКРО КОНАЧНИ ЕЛЕМЕНТИ ПРИМИТИВА

## 6.1 Поставка проблема

Из претходног смо видели да генерисани облици примитива не морају да поседују одлику симетрију. Самим тим нису погодни за прорачун применом групно суперматричне процедуре. Такође, умеју да буду поприлично комплексан скуп више плоча у простору, што додатно отежава прорачун. У те сврхе развија се макро коначни елемент у виду танке плоче који треба да поседује идентичне статичке и данимичке карактеристике као и одговарајући примитив. Поступак формирања макро коначног елемента је јединствен и важи само за предметни примитив. Уколико се било који параметар примитива промени, поступак је потребно урадити поново.

## 6.2 Равна танка плоча

Све примитиве преводимо у заменску равну изотропну танку плочу, коју називамо еквивалентном плочом. Ради једноставности приказа узећемо да су пројекције свих примитива на хоризонталну раван димензија 100 x 100 cm ($A_x$, $A_z$) и да поседују константну јединичну дебљину од 1 cm. Исте димензије усвајамо и за еквивалентну плочу. За материјал се усваја челик, па су модул еластичности и Поасонов коефицијент респективно:

$$E = 2{,}1 \text{ x } 10^4 \text{ kN/cm}^2 \qquad \nu = 0{,}3$$

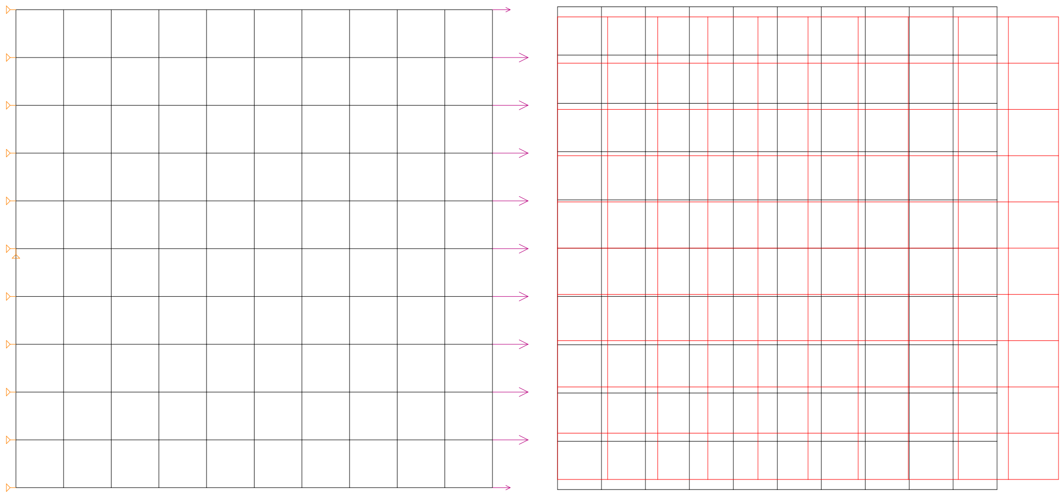
Равну танку плочу делимо у оба правца на по 10 коначних елемената. Ако сада једну ивицу плоче ослонимо дуж ивице у правцу z и извршимо њено истезање дуж наспрамне ивице јединичном континуалном силом 10 kN/cm, добијамо деформацију плоче у x правцу (слика 6.2.1).

Укупна сила истезања износи:

$$F = 9 \text{ x } 1 \text{ x } 10^2 + 2 \text{ x } 0{,}5 \text{ x } 10^2 = 10^3 \text{ kN}$$

Анализом тако дефинисаних улазних параметара у програму KOMIPS добија се величина деформације и напона који износе:

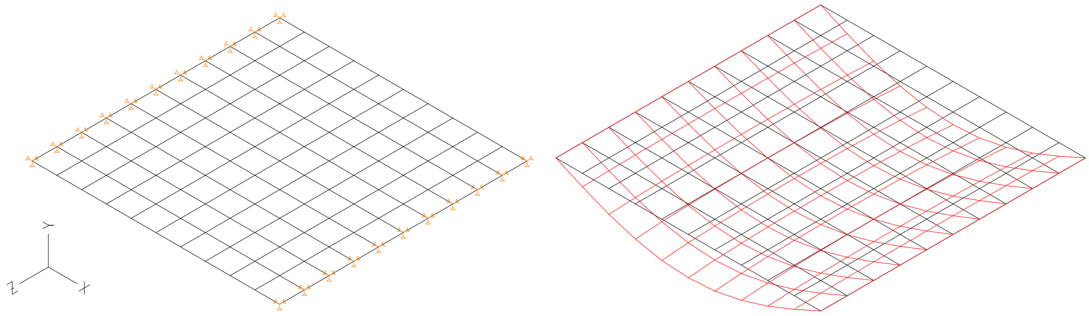
$$f_{zmaxp} = f_{xmaxp} = 0{,}048 \text{ cm} \qquad \sigma = 10 \text{ kN/cm}^2$$



Слика 6.2.1. Оптерећење равне танке плоче у правцу x и њена деформација

С обзиром да је плоча изотропна потпуно исти вредности се добијају и у другом правцу.

Слично можемо применити и при савијању плоче. Формирајмо сада слободно ослоњене наспрамне ивице плоче у z правцу. Затим плочу оптерећујемо само сопственом тежином. Рачунски модел и његова деформација дати су на слици 6.2.2.




Слика 6.2.2. Савијање плоче и њена деформација

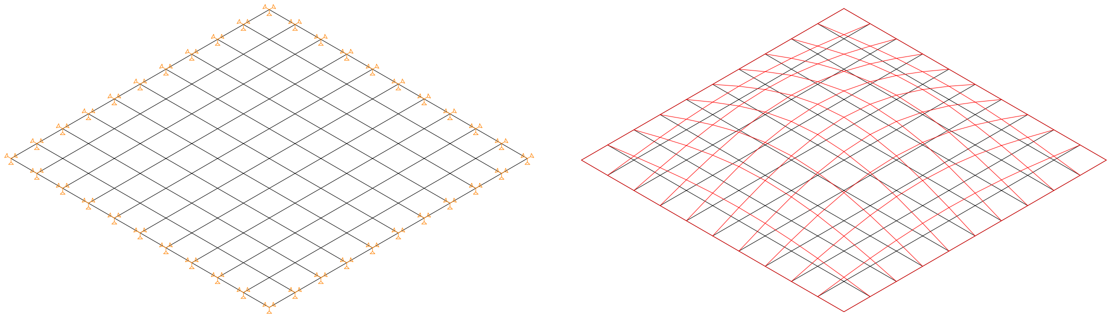
Континуално оптерећење услед сопствене тежине износи:

$$q = 8 \text{ x } 10^{-5} \text{ kN/cm}^2 \text{ (тежина 0,8 kN)}$$

Добијају се величине угиба и напона која износи:

$$f_{uxmax} = f_{uzmax} = 0{,}062 \text{ cm} \qquad \sigma = 0{,}6 \text{ kN/cm}^2$$

Затим, динамички прорачун сопствених фреквенци са слободно ослоњену плочу по целој. На слици 6.2.3 приказан је рачунски модел плоче и први тон осциловања плоче.



Слика 6.2.3. Први тон осциловања

Добија се вредност сопствене фреквенце првог тона осциловања и она износи:

$$f_{01} = 48{,}8\ Hz$$

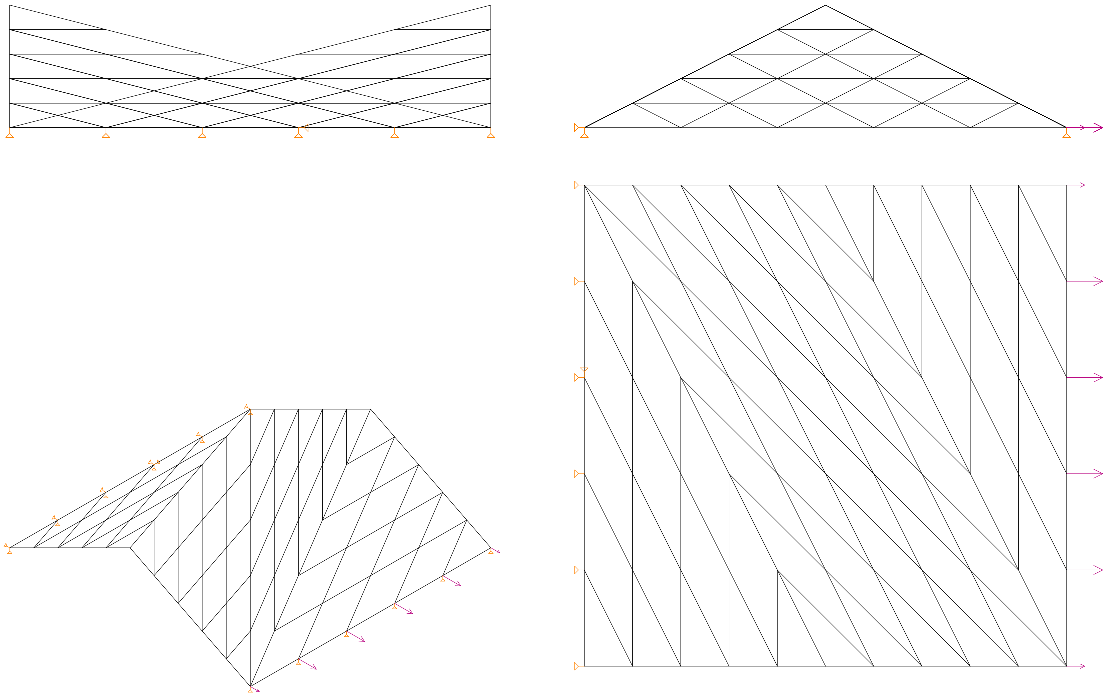
**6.3 Одређивање еквивалентне матрице крутости за примитив P – 3**

Потпуно исти поступак примењујемо и на изабрани примитив. Посматрајмо примитив типа 3 чија је једна ивица слободно ослоњена, а наспрамна се истеже силама које су у збиру једнаке као и за плочу, што је приказано на слици 6.3.1 за x правац, односно слици 6.3.3 за z правац.

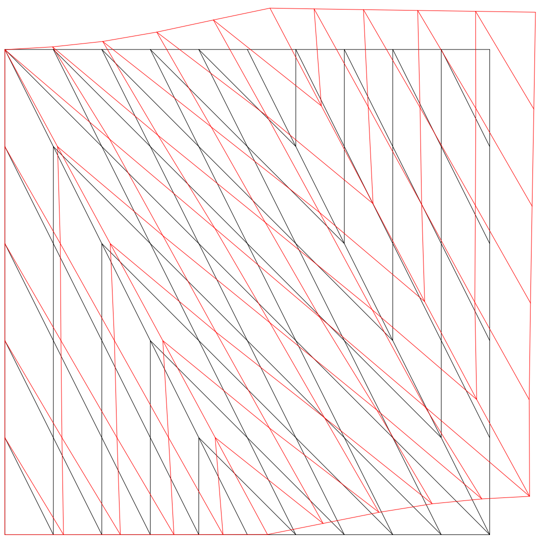
$$F = 4 \text{ x } 2 \text{ x } 10^2 + 2 \text{ x } 1 \text{ x } 10^2 = 10^3\ kN$$

Сада добијамо вредност максималну вредност деформације предметног примитива услед оптерећења у x правцу (слика 6.3.2):

$$f_{xmaxpr} = 14\ cm$$



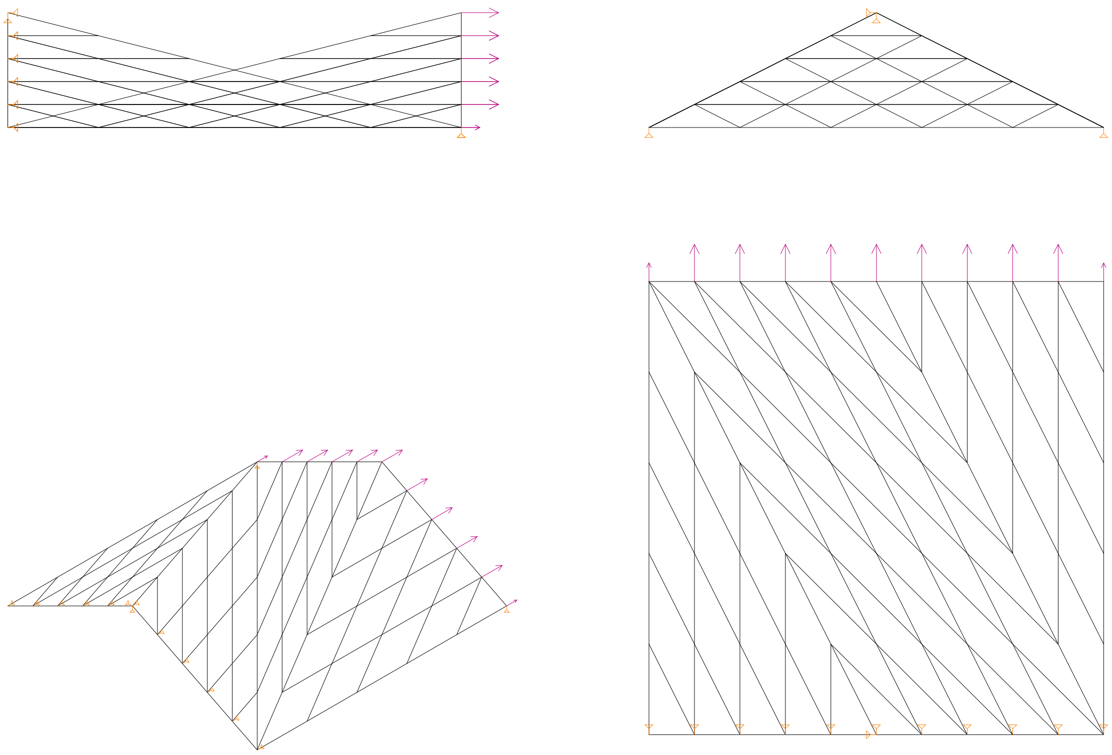
Слика 6.3.1. Оптерећење примитива у x правцу



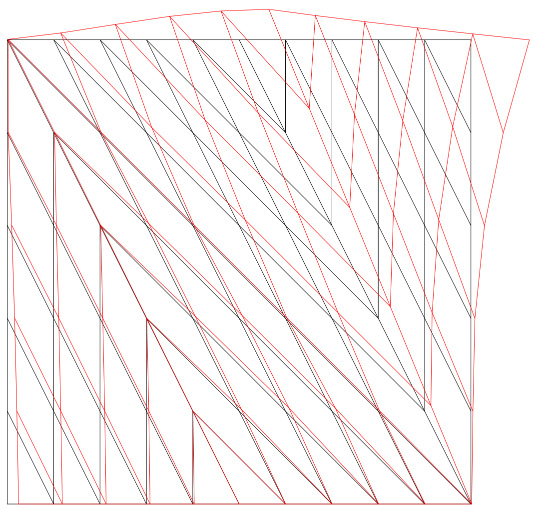
Слика 6.3.2. Деформација примитива од оптерећења у правцу x

Приметно је да је деформација примитива на истезање вишеструко већа него плоче истих димензија и карактеристика. Сада је потребно одредити еквиваленти модул еластичности $E_{xx}^{m}$. То постижемо тако што изједначавамо деформације, односно њиховог односа:

$$E_{xx}^{m} = f_{zmaxpr} / f_{zmaxp} \ E = 14 / 0{,}48 \text{ x } 2{,}1 \text{ x } 10^4 = 0{,}0072 \text{ x } 10^4 \text{ kN/cm}^2$$



Слика 6.3.3. Оптерећење примитива у z правцу



Слика 6.3.4. Деформација примитива од оптерећења у правцу z

Добија се вредност максималне деформације од оптерећења у z правцу (слика 6.3.2):

$$f_{zmaxpr} = 7\ \text{cm}$$

Обзиром да максимална деформација у оба правца није идентична и еквивалентнаи модул еластичности неће бити исти:

$$E^{m}_{zz} = f_{zmaxpr} \;/\; f_{zmaxp} \;\; E = 7 \,/\, 0{,}48 \text{ x } 2{,}1 \text{ x } 10^{4} = 0{,}0144 \text{ x } 10^{4} \text{ kN/cm}^{2}$$

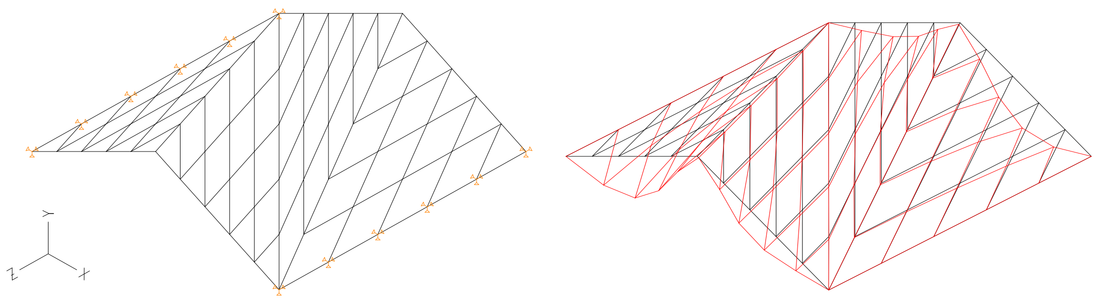
За матрицу еластичности изотропног материјала довољно је одредити вредности еластичности и Поасонов коефицијент. Пошто су одређене све потребне вредности, формирамо еквивалентну матрицу еластичности за мембранско напрезање $D^{m}_{e}$:

$$[D^{m}_{e}] = \begin{bmatrix} D^{m}_{xx} & D^{m}_{xz} & 0 \\ D^{m}_{zx} & D^{m}_{zz} & 0 \\ 0 & 0 & D^{m}_{yy} \end{bmatrix}$$

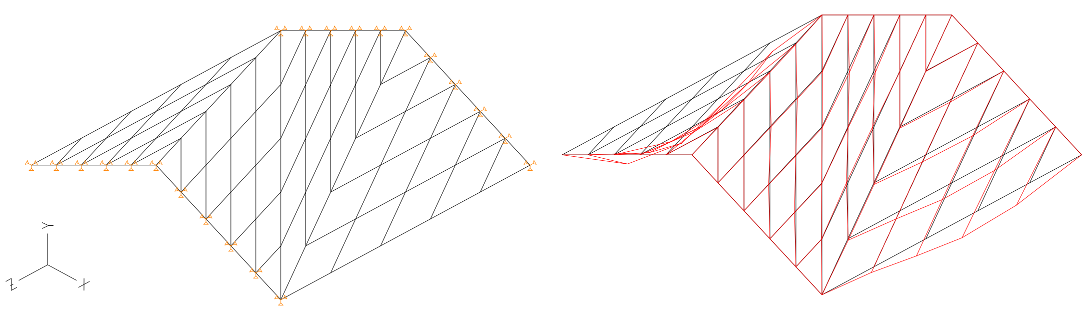
Где су:

$$D^{m}_{xx} = D^{m}_{zz} = \frac{E^{m}_{xx}}{1-\nu^{2}} \qquad D^{m}_{xz} = D^{m}_{zx} = \nu\, D^{m}_{xx} \qquad D^{m}_{yy} = G = \frac{E^{m}_{z}}{2(1+\nu)}$$

Аналогно савијању плоче формирамо сада слободно ослоњене наспрамне ивице плоче у z, односно x правцу. Затим плочу оптерећујемо вертикалним континуалним оптерећењем само од сопствене тежине. Рачунски модели и његова деформација дати су на сликама 6.3.5 и 6.3.6



Слика 6.3.5. Савијање плоче и њена деформација $f_{uzmax}$



Слика 6.3.6. Савијање плоче и њена деформација $f_{uxmax}$

Укупно оптерећење од сопствене тежине  износи 0,94 кN:

$$q = 9{,}4 \text{ x } 10^{-5} \text{ kN/cm}^2$$

Због нејаднакости сопствених тежина уводи се фактор усаглашавања оптерећења од тежине који износи 0,8/0,94 = 0,85 и представља однос тежине равне плоче и примитива. Након редуковања оптерећења добијају се величине угиба:

$$f_{uzmaxpr} = 0{,}00234 \text{ cm}$$

$$f_{uxmaxpr} = 0{,}00143 \text{ cm}$$

Еквивалентни савојни модули еластичности сада гласе:

$$E_{xx}^{s} = 103\text{x}10^{4}\text{kN/cm}^2 \qquad E_{zz}^{s} = 51\text{x}10^{4}\text{ kN/cm}^2$$

$$E_{xz}^{s} = E_{xz}^{s} = 30\text{x}10^{4}\text{ kN/cm}^2 \qquad E_{yy}^{s} = 30\text{x}10^{4}\text{ kN/cm}^2$$

Пошто су одређене све потребне вредности, формирамо еквивалентну матрицу еластичности за савојно напрезање $D_e^s$:

$$[D_e^s] = \begin{bmatrix} D_{xx}^s & D_{xz}^s & 0 \\ D_{zx}^s & D_{zz}^s & 0 \\ 0 & 0 & D_{yy}^s \end{bmatrix}$$

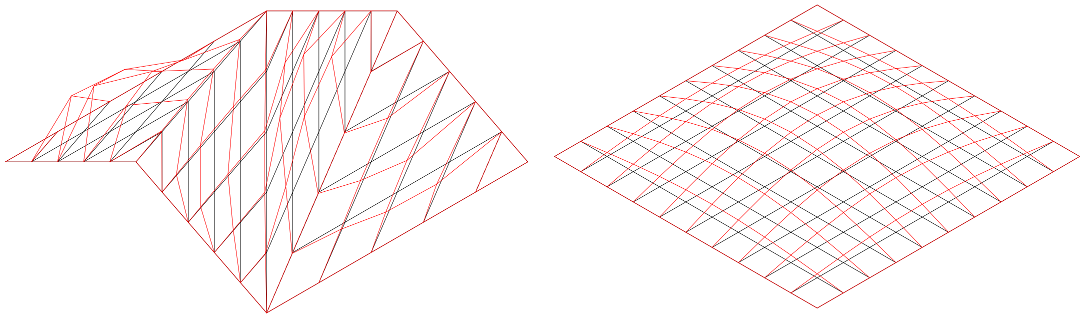
Где су:

$$D_{xx}^s = \frac{E_{xx}^s}{1-\nu^2} \qquad D_{xz}^s = D_{zx}^s = \nu_{xz} D_{xx}^s = \nu_{zx} D_{zz}^s \qquad D_{zz}^s = \frac{E_{zz}^s}{1-\nu^2}$$

Видимо да се при савијању јављају анизотропне карактеристике па модул клизања можемо приближно одредити. Грешка која се јавља при одређивању не утиче значајно на квалитет прорачуна.

$$D_{yy}^s = G = \frac{E}{2(1+\nu_{xz})}$$

У циљу верификације и провере тачности комплетног поступка урађена је динамичка анализа примитива и одговарајуће еквиваленте плоче којој су сада додељене вредности еквивалентних матрица еластичности за савојно напрезање. Услови ослањања по контури су идентични како за примитив тако и за еквивалентну плочу и цела контура је слободно ослоњена. Имајући у виду да на резултат динамичке анализе значајан утицај има маса примитива, извршена је редукција специфичне тежине примитива у циљу изједначавања маса примитива и еквивалентне плоче. Када се постигне једнакост маса вредност сопствене фреквенце зависи само од матрице еластичности. Облици првих тонова осциловања су приказани на слици 6.3.7.



Слика 6.3.7. Први тон осциловања примитива и еквивалентне плоче

Добијена је вредност првог тона осциловања примитива у износу од:

$$f_{01pr} = 299 \text{ Hz}$$

Односно вредност првог тона осциловања еквивалентне плоче у износу од:

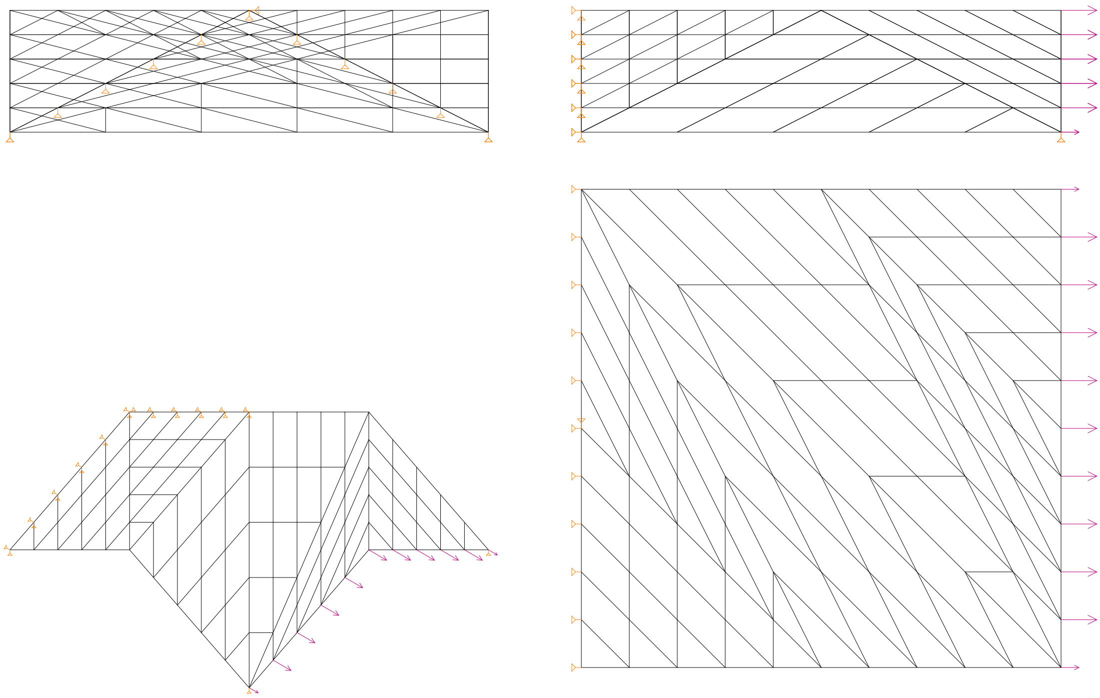
$$f_{01ep} = 294{,}3 \text{ Hz}$$

Разлика у вредностима у износу од 1,6 % се може сматрати занемарљивом, тако да можемо да константујемо исправност и коректност примењеног поступка.

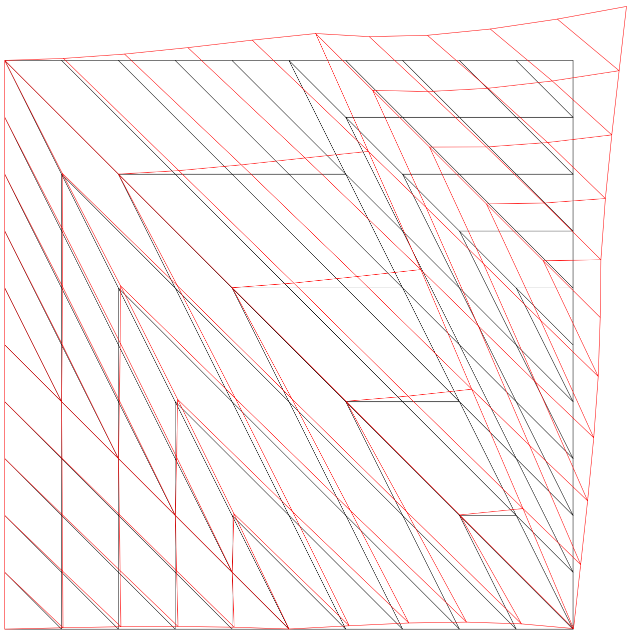
Вредност сопствене фреквенце првог тона осциловања је такође и значајан параметар при компарацији примитива. Већа вредност фреквенце значи и повећан капацитет носивости, односно већу крутост примитива.

Аналогогно спроводимо исти поступак за све примитиве. У наредним поглављима приказани су само резултати статичко – динамичке анализе и њихова визуелна интерпретација, што је довољно за описивање понашања примитива.

## 6.4 Одређивање еквивалентне матрице крутости за примитив Р – 4

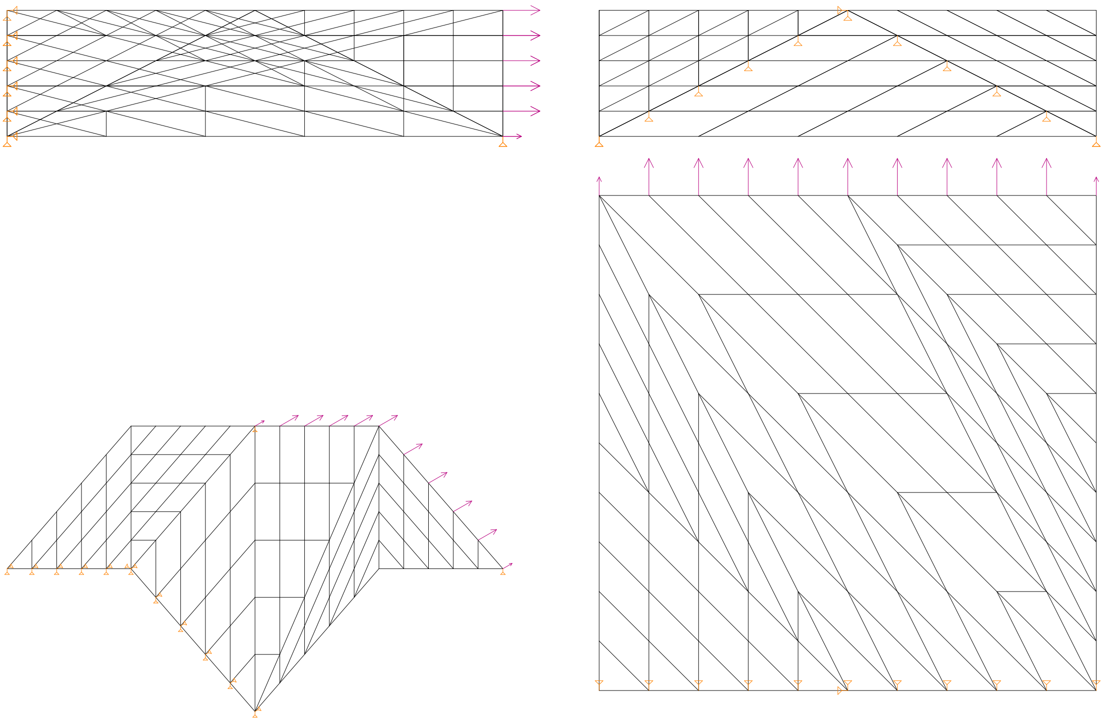


Слика 6.4.1. Оптерећење примитива у х правцу

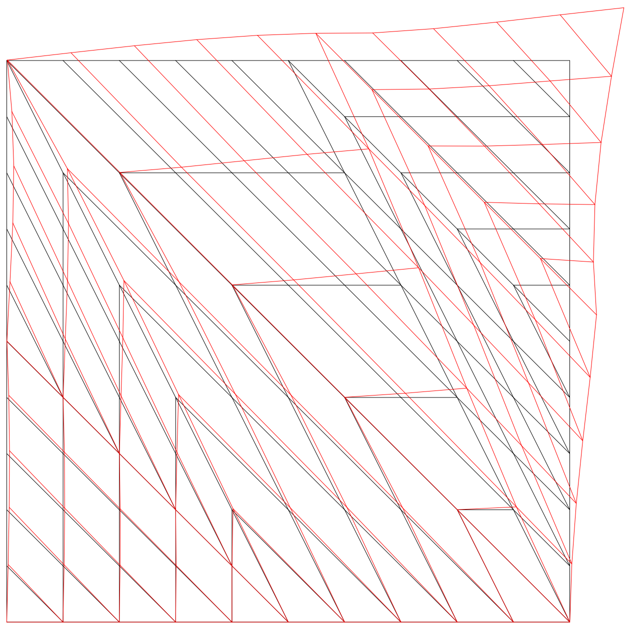


Слика 6.4.2. Деформација примитива од оптерећења у правцу х

$f_{xmaxpr} = 9{,}2$ cm $\quad E_{xx}^{m} = 0{,}011 \times 10^4$ kN/cm$^2$

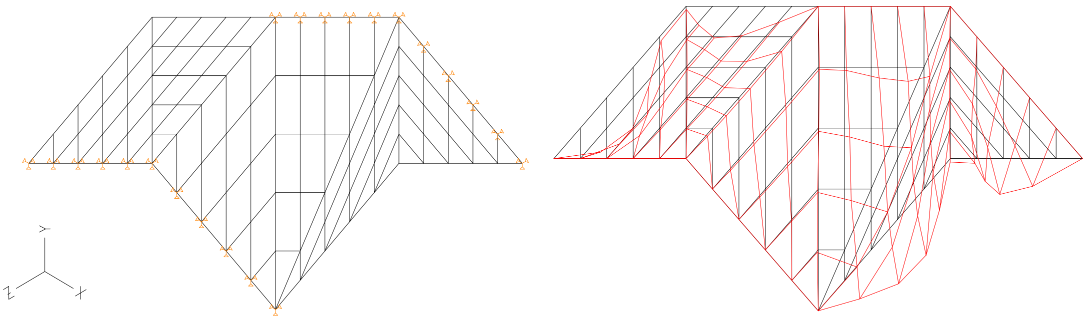


Слика 6.4.3. Оптерећење примитива у z правцу



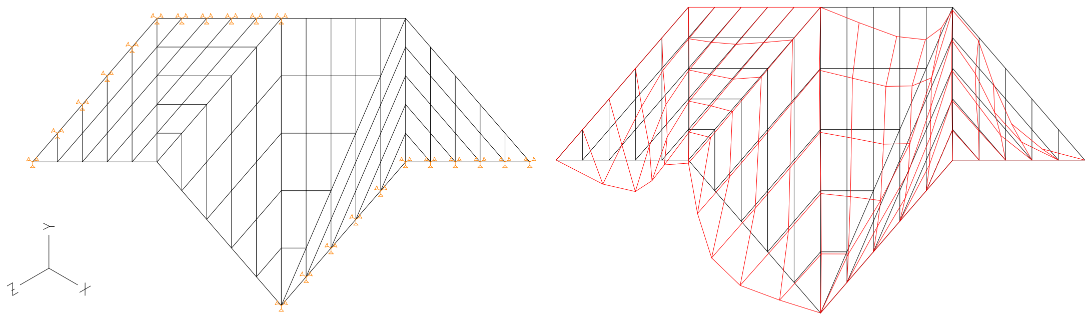
Слика 6.4.4. Деформација примитива од оптерећења у правцу z

$f_{zmaxpr}$ = 9,2 cm $\qquad$ $E^{m}_{zz}$ = 0,011x10$^4$ kN/cm$^2$



Слика 6.4.5. Савијање плоче и њена деформација $f_{uzmax}$

$f_{uzmax}$ =0,00056 cm

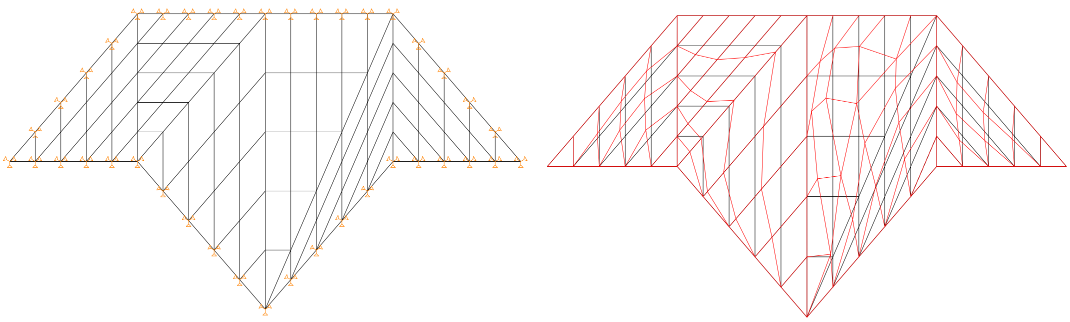


Слика 6.4.6. Савијање плоче и њена деформација $f_{uxmax}$

$f_{uxmax}$ = 0,00069 cm

$E^{s}_{xx} = 250\text{x}10^{4}$ kN/cm$^2$ $E^{s}_{xz} = 69\text{x}10^{4}$ kN/cm$^2$

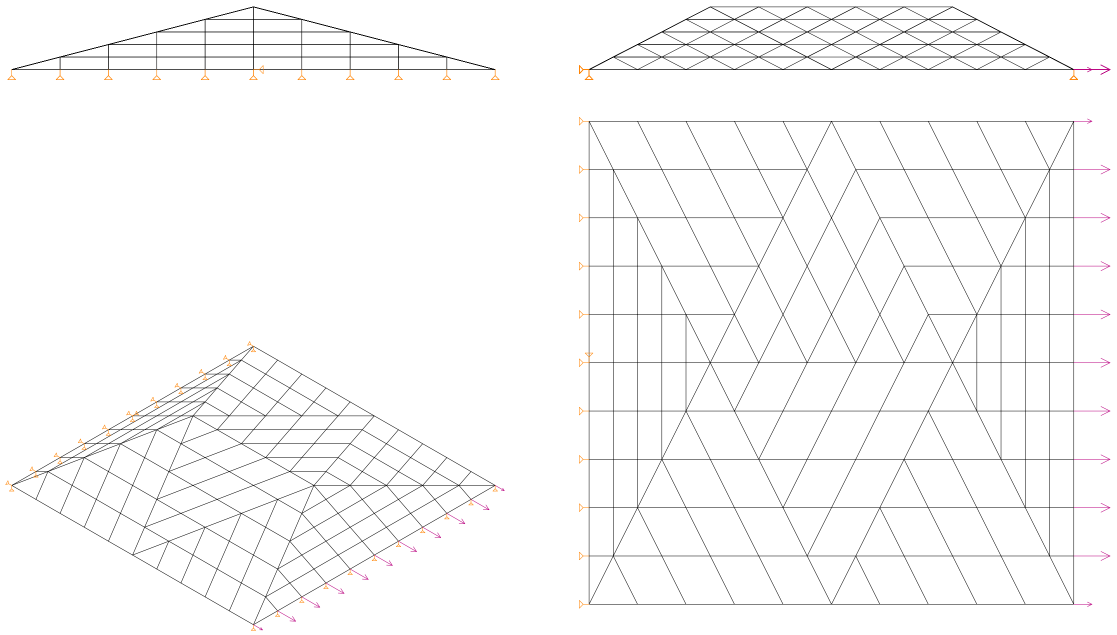
$E^{s}_{zz} = 209\text{x}10^{4}$ kN/cm$^2$ $E^{s}_{yy} = 88 \text{ x } 10^{4}$ kN/cm$^2$



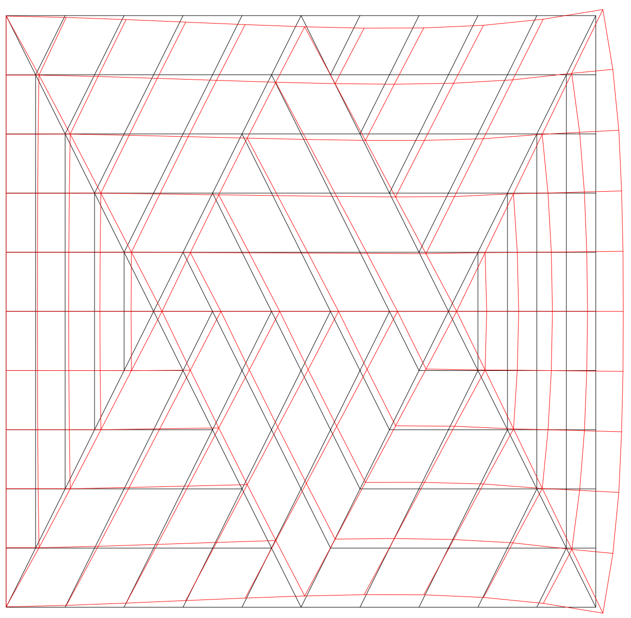
Слика 6.4.7. Први тон осциловања примитива

$f_{01}$ = 209,1 Hz

## 6.5 Одређивање еквивалентне матрице крутости за примитив P – 5

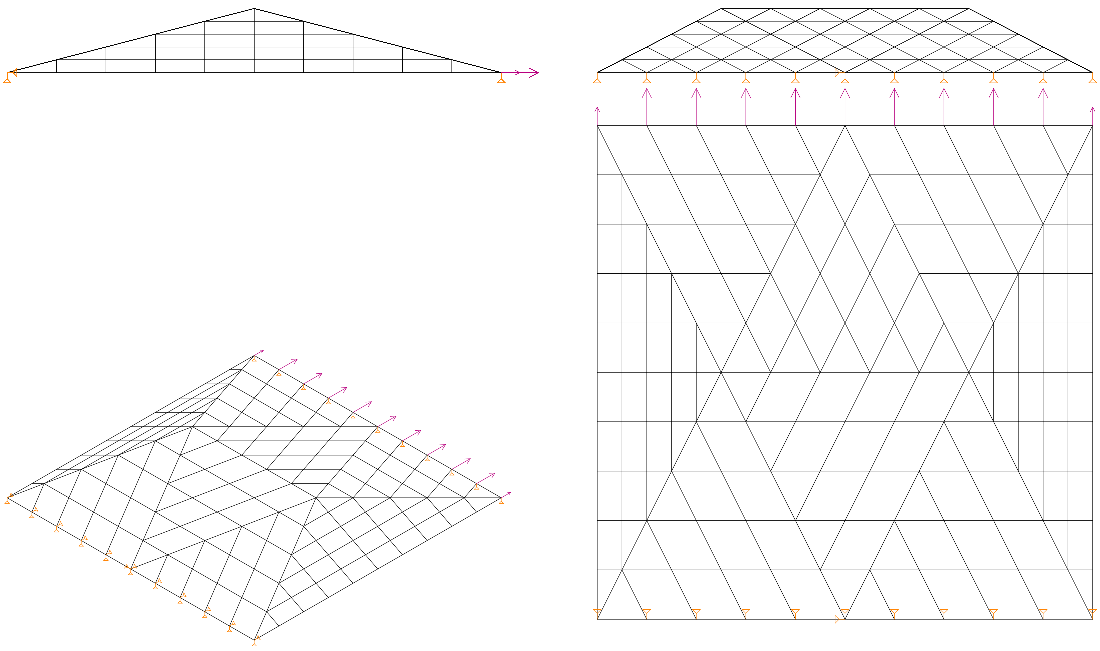


Слика 6.5.1. Оптерећење примитива у x правцу

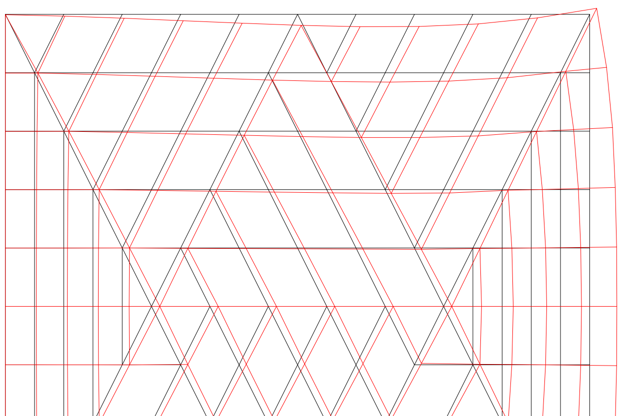


Слика 6.5.2. Деформација примитива од оптерећења у правцу x

$f_{xmaxpr} = 0,74$ cm $\qquad E^{m}_{xx} = 0,15x10^{4}$ kN/cm$^{2}$

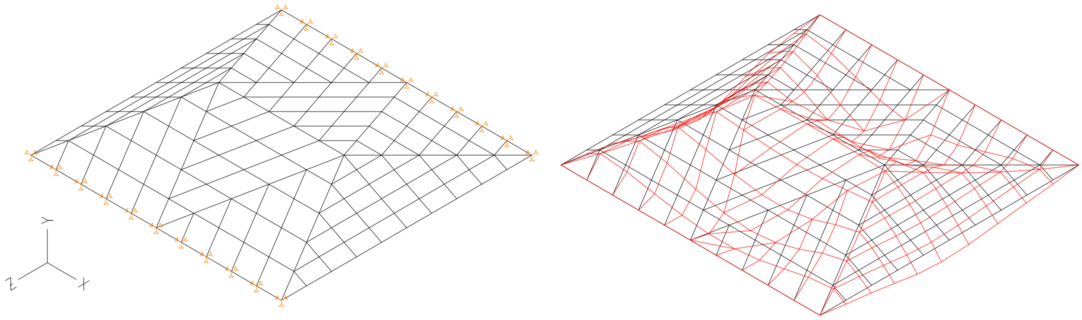


Слика 6.5.3. Оптерећење примитива у z правцу



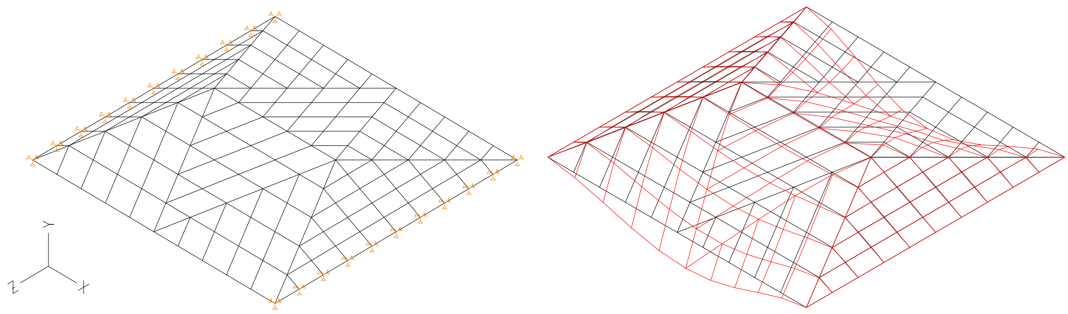
Слика 6.5.4. Деформација примитива од оптерећења у правцу z

$f_{zmaxpr} = 0{,}57$ cm $\qquad E_{xx}^{m} = 0{,}194 \times 10^{4}$ kN/cm$^2$



Слика 6.5.5. Савијање плоче и њена деформација $f_{uzmax}$

$f_{uzmax}$ = 0,0061 cm

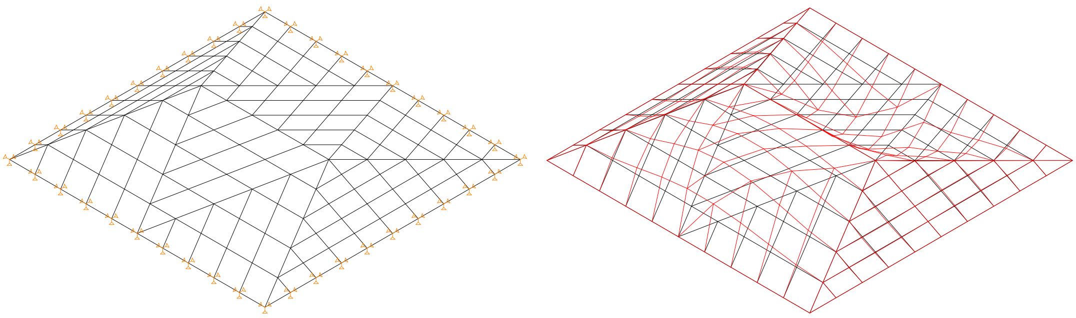


Слика 6.5.6. Савијање плоче и њена деформација $f_{uxmax}$

$f_{uxmax}$ = 0,063 cm

$E^{s}_{xx}$ = 198,72x10$^{4}$ kN/cm$^{2}$ $\qquad$ $E^{s}_{xz}$ = 33x10$^{4}$ kN/cm$^{2}$

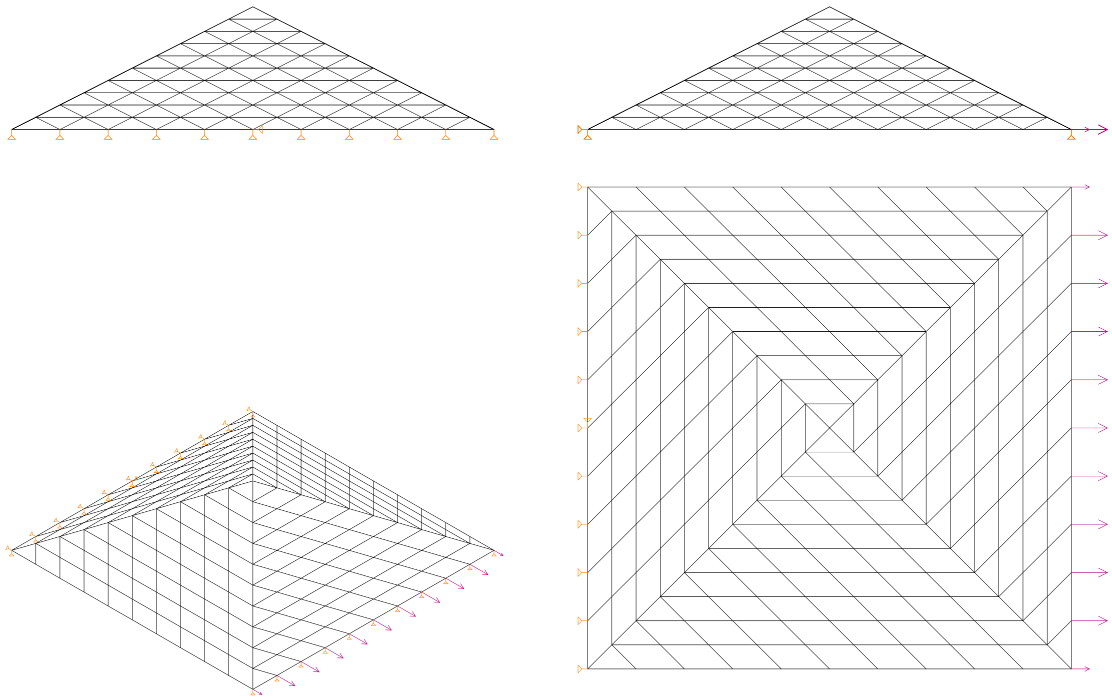
$E^{s}_{zz}$ = 17,68x10$^{4}$ kN/cm$^{2}$ $\qquad$ $E^{s}_{yy}$ = 42,3x10$^{4}$ kN/cm$^{2}$



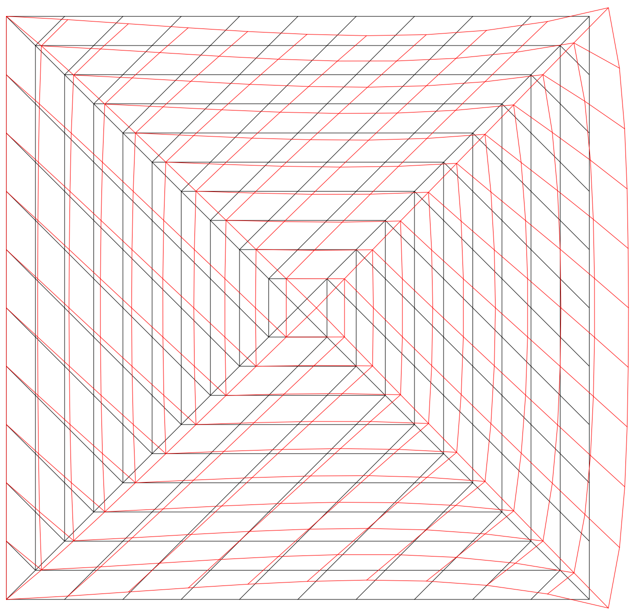
Слика 6.5.7. Први тон осциловања примитива

$f_{01}$ = 163,8 Hz

## 6.6 Одређивање еквивалентне матрице крутости за примитив Р – 6

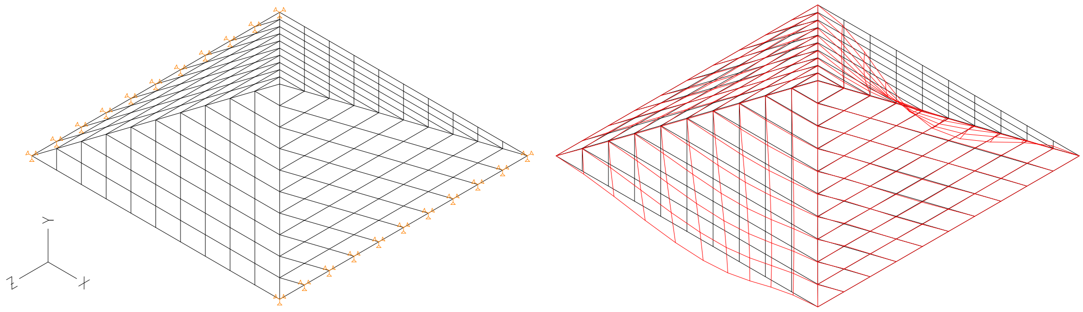


Слика 6.6.1. Оптерећење примитива у x правцу



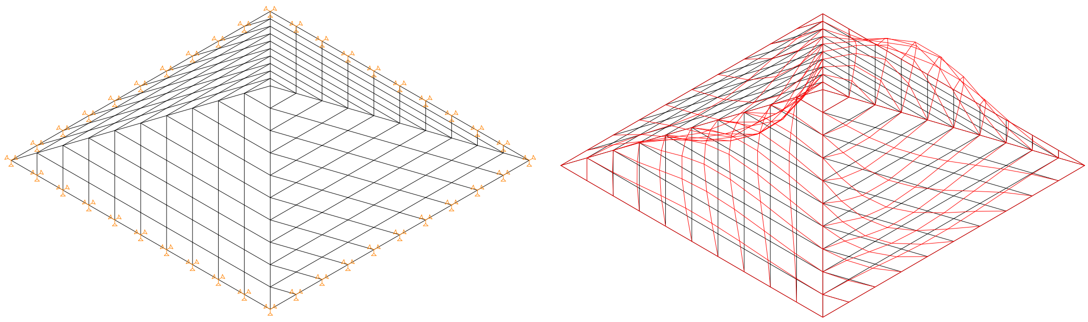
Слика 6.6.2. Деформација примитива од оптерећења у правцу x

$f_{xmaxpr} = 0{,}635$ cm



Слика 6.6.3. Савијање плоче и њена деформација $f_{uzmax}$

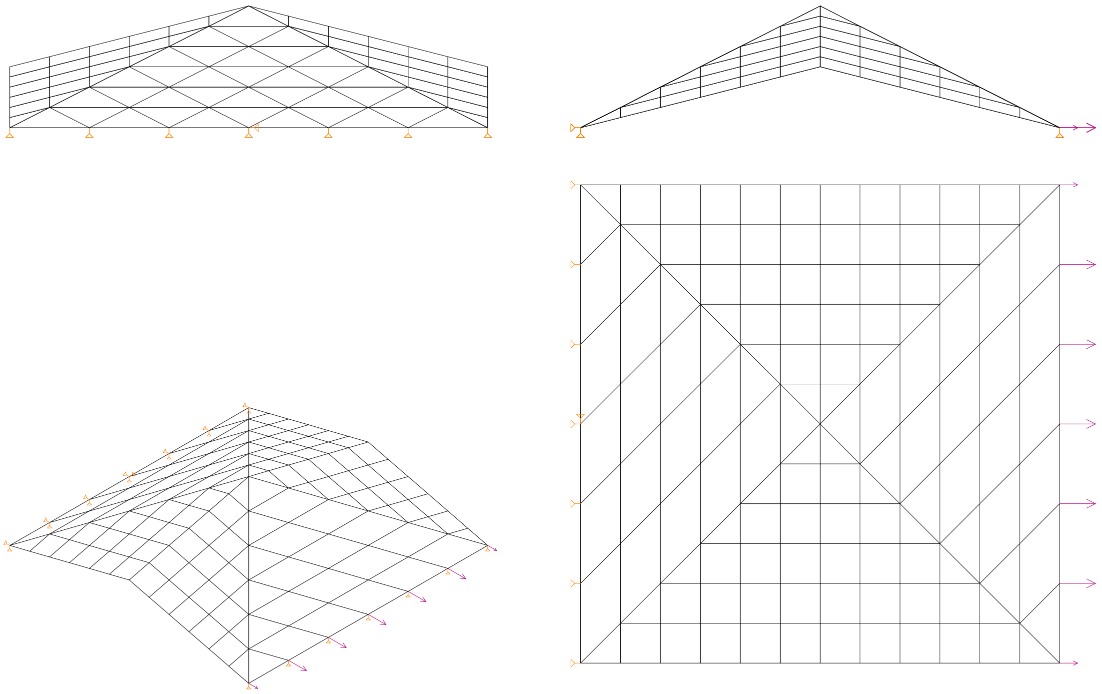
$f_{uzmax}$ = 0,283 cm



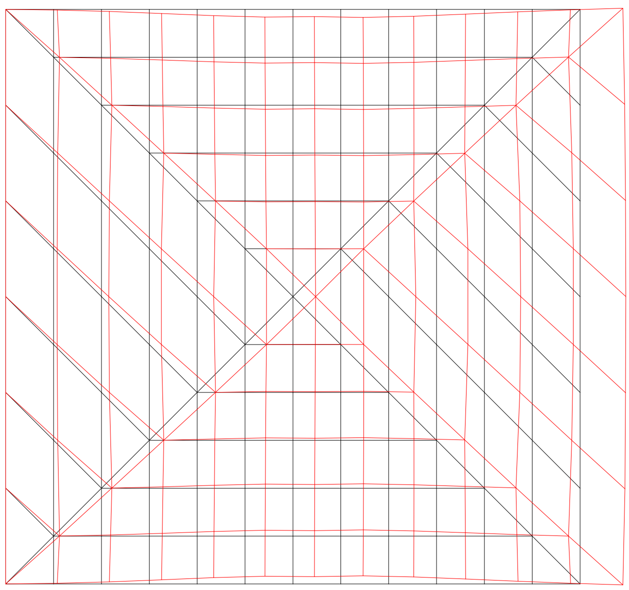
Слика 6.6.4. Први тон осциловања примитива

$f_{01}$ = 214 Hz

## 6.7 Одређивање еквивалентне матрице крутости за примитив Р – 7

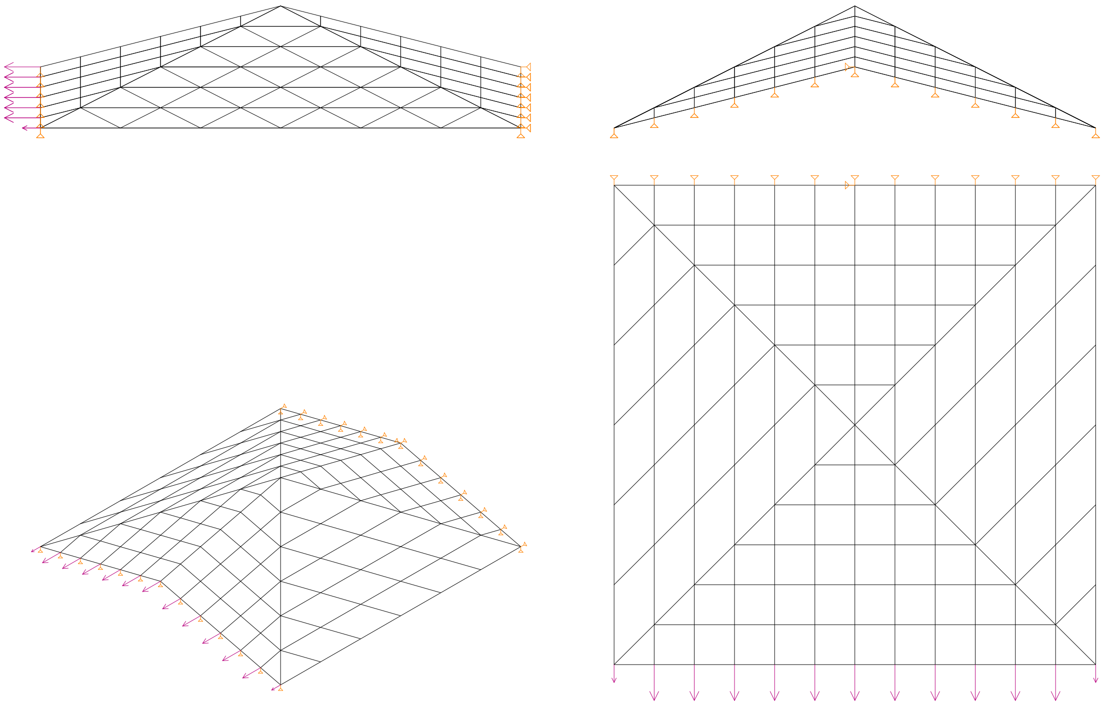


Слика 6.7.1. Оптерећење примитива у x правцу

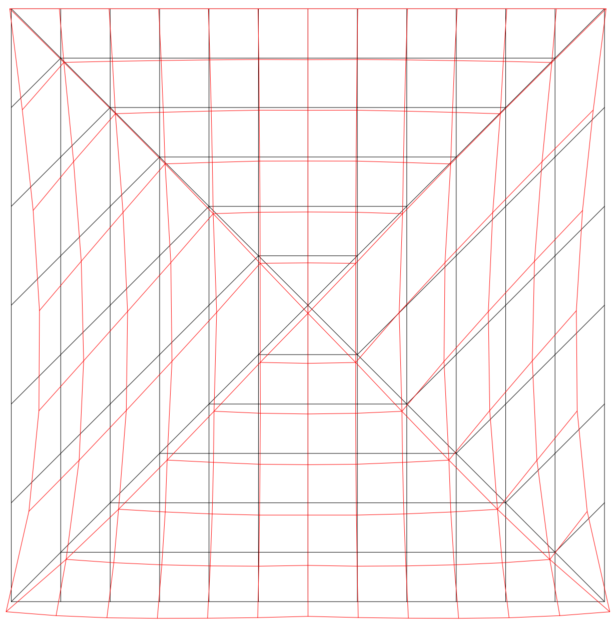


Слика 6.7.2. Деформација примитива од оптерећења у правцу x

$f_{xmaxpr}$ = 15,1 cm

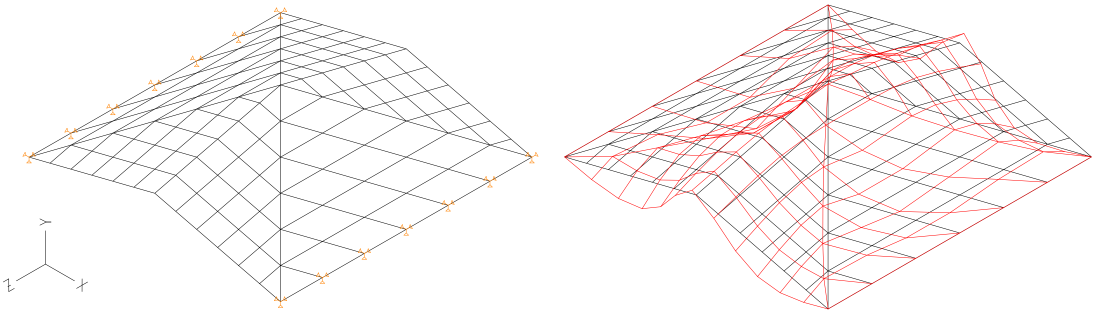


Слика 6.7.3. Оптерећење примитива у z правцу



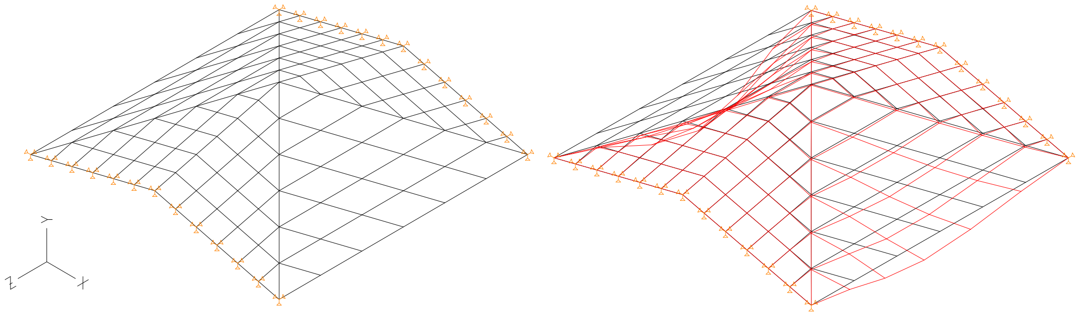
Слика 6.7.4. Деформација примитива од оптерећења у правцу z

$f_{zmaxpr} = 0{,}264$ cm



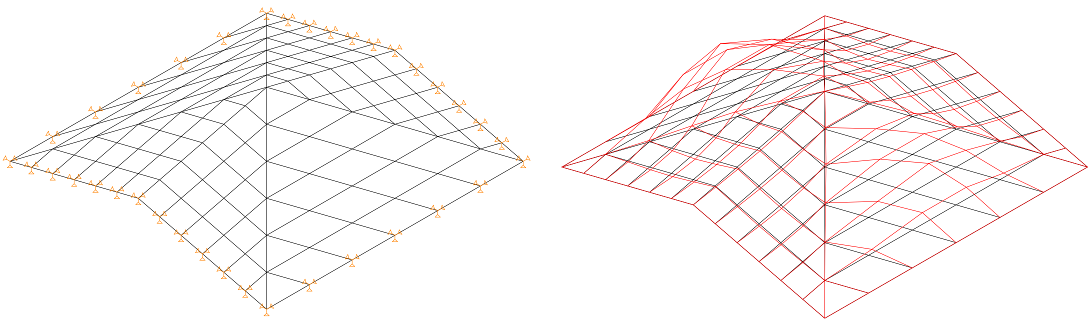
Слика 6.7.5. Савијање плоче и њена деформација $f_{uzmax}$

$f_{uzmax}$ = 0,00419 cm



Слика 6.7.6. Савијање плоче и њена деформација $f_{uxmax}$

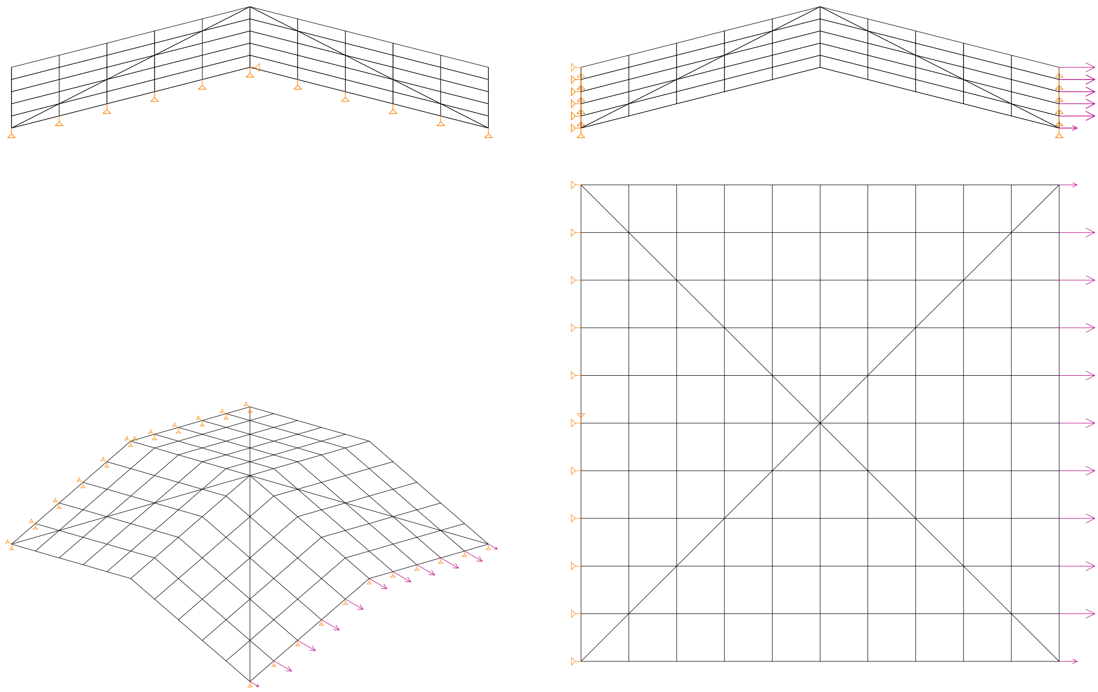
$f_{uxmax}$ = 0,0316 cm



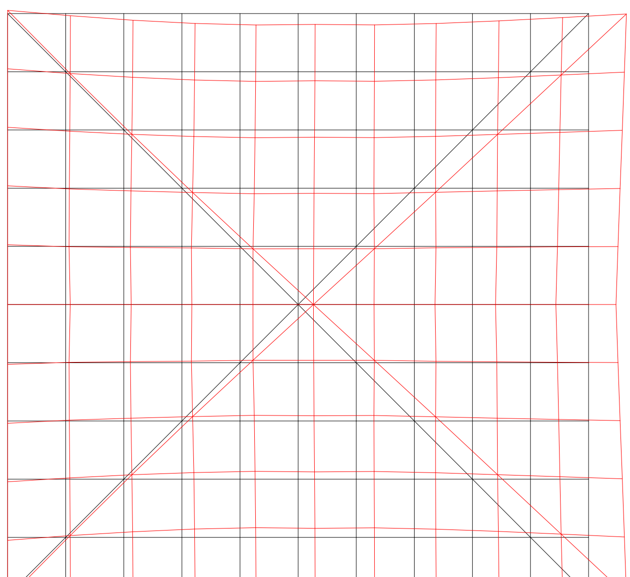
Слика 6.7.7. Први тон осциловања примитива

$f_{01}$ = 257 Hz

## 6.8 Одређивање еквивалентне матрице крутости за примитив P – 8

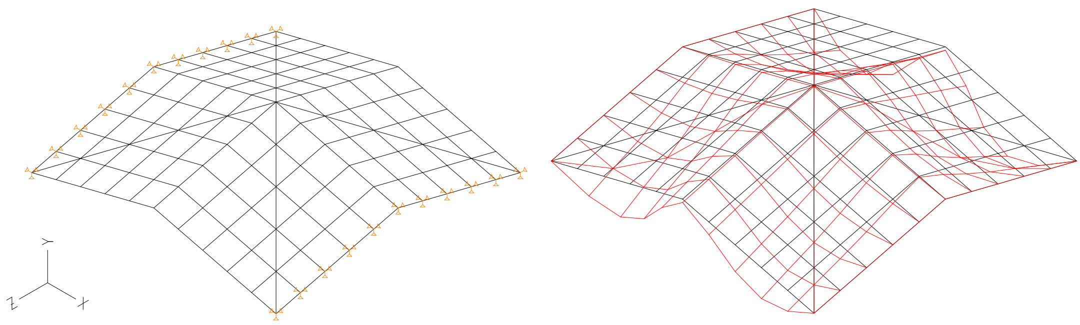


Слика 6.8.1. Оптерећење примитива у x правцу



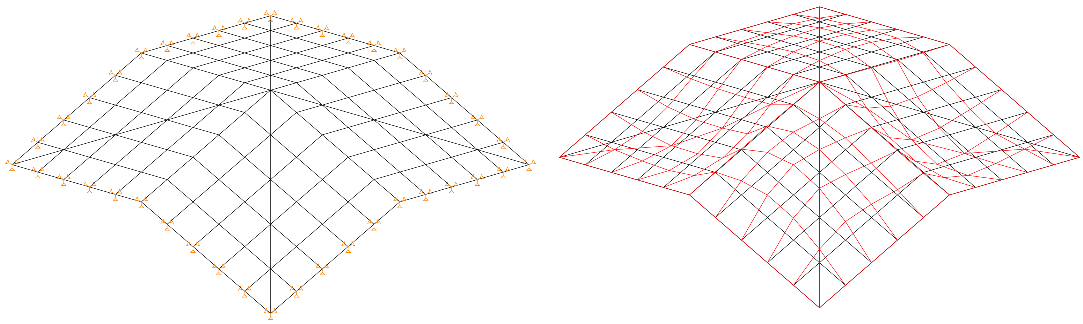
Слика 6.8.2. Деформација примитива од оптерећења у правцу x

$f_{zmaxpr}$ = 1,52 cm



Слика 6.8.3. Савијање плоче и њена деформација $f_{uzmax}$

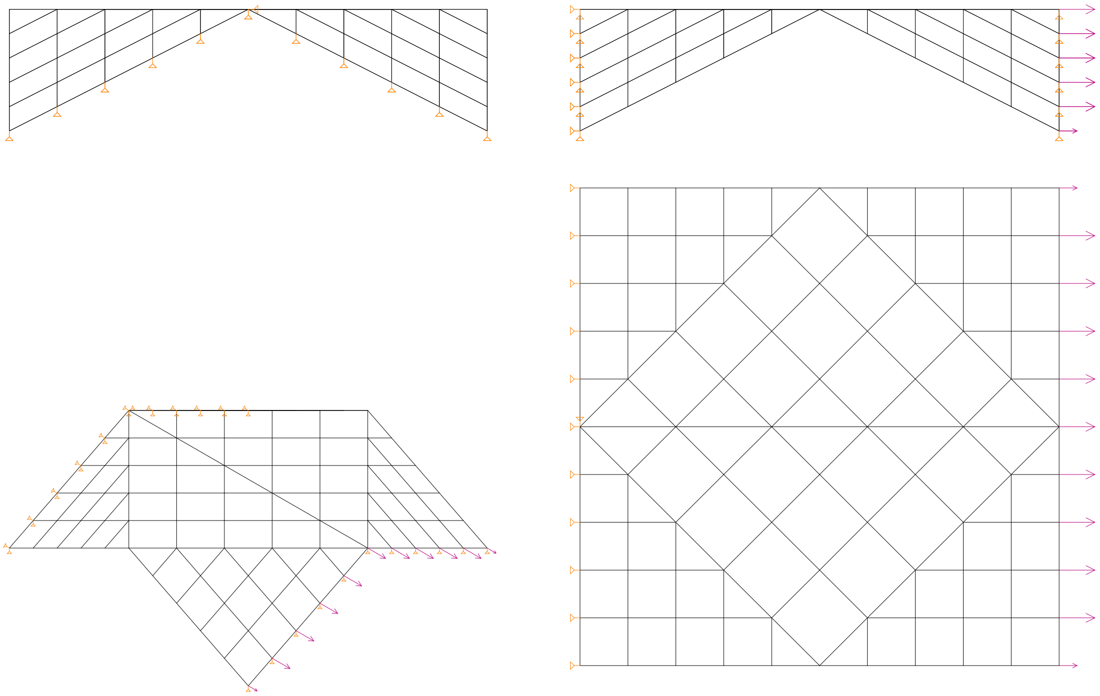
$f_{uzmax}$ = 0,0149 cm



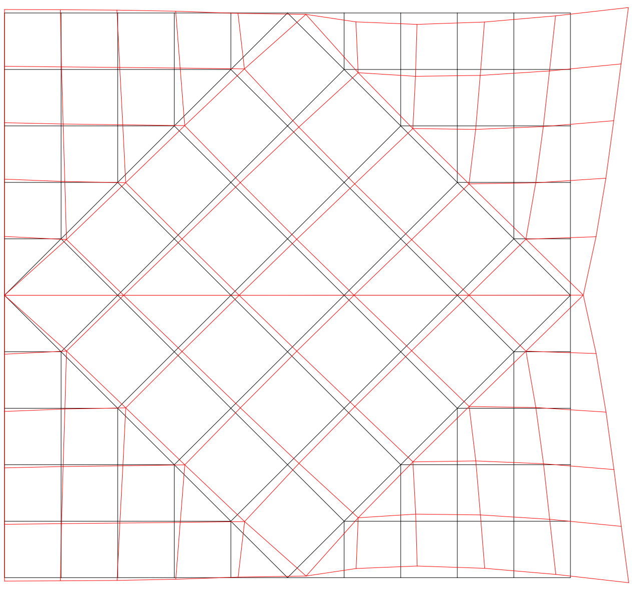
Слика 6.8.4. Први тон осциловања примитива

$f_{01}$ = 280 Hz

### 6.9 Одређивање еквивалентне матрице крутости за примитив P – 9

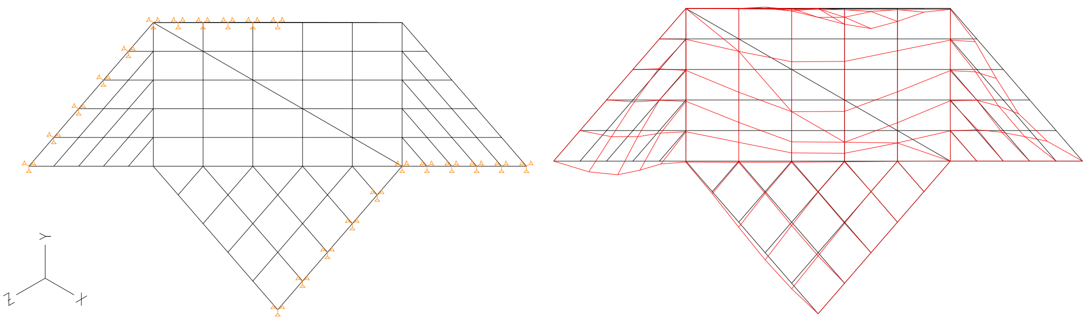


Слика 6.9.1. Оптерећење примитива у x правцу



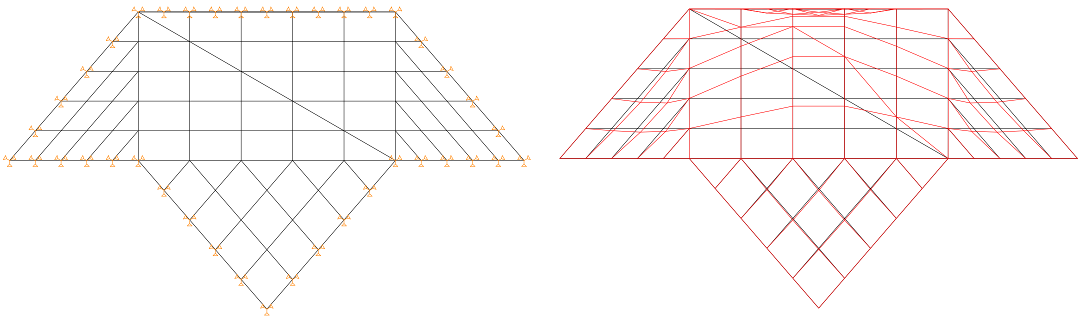
Слика 6.9.2. Деформација примитива од оптерећења у правцу x

$f_{xmaxpr}$ = 0,373 cm



Слика 6.9.3. Савијање плоче и њена деформација $f_{uzmax}$

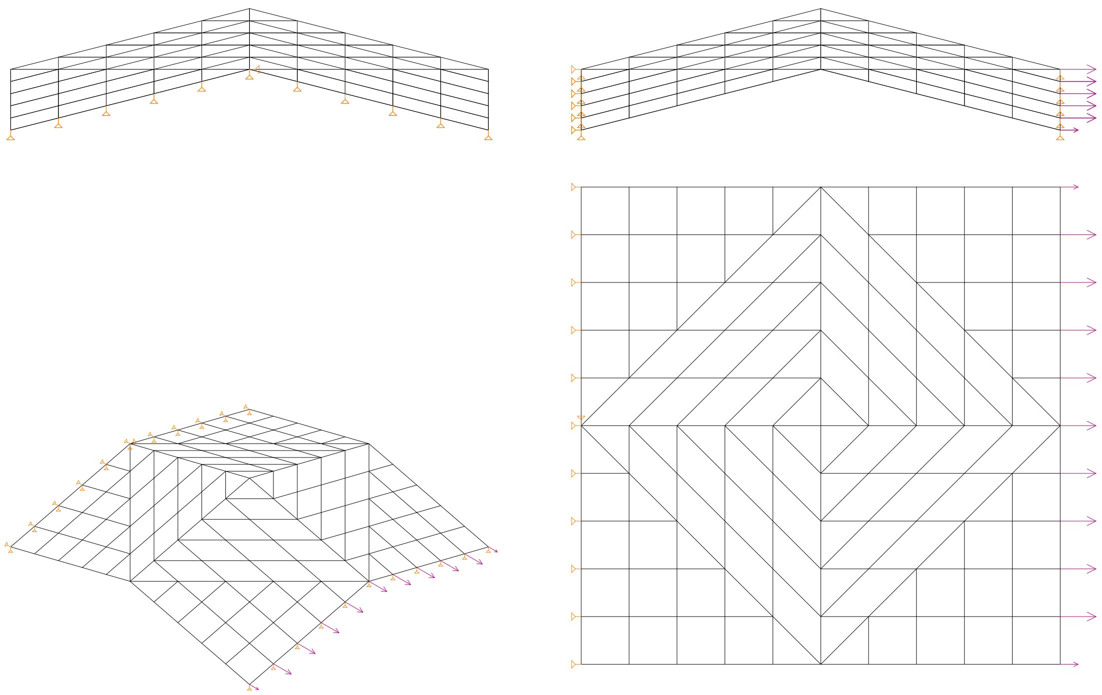
$f_{uzmax}$ = 0,0132 cm



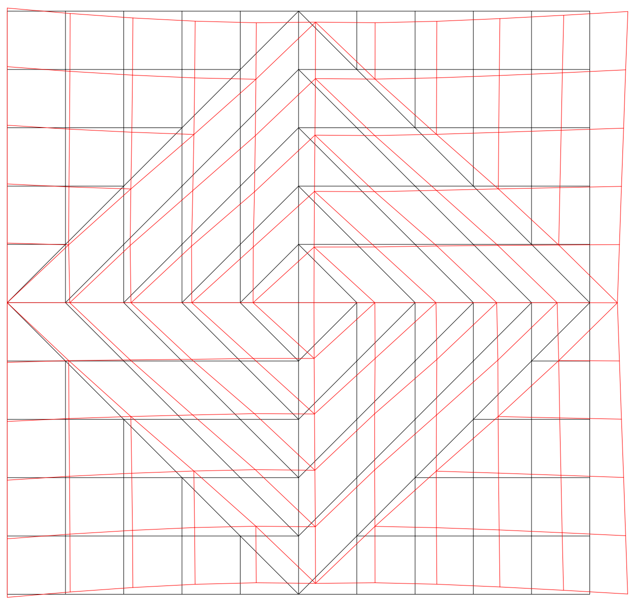
Слика 6.9.4. Први тон осциловања примитива

$f_{01}$ = 242 Hz

## 6.10 Одређивање еквивалентне матрице крутости за примитив P – 10

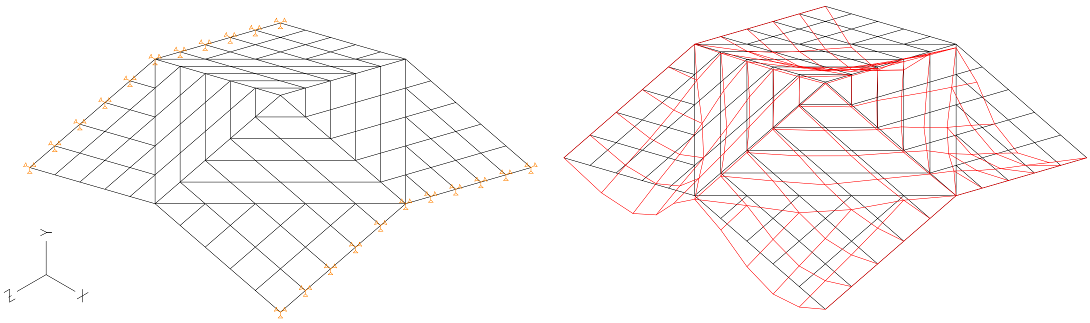


Слика 6.10.1. Оптерећење примитива у x правцу



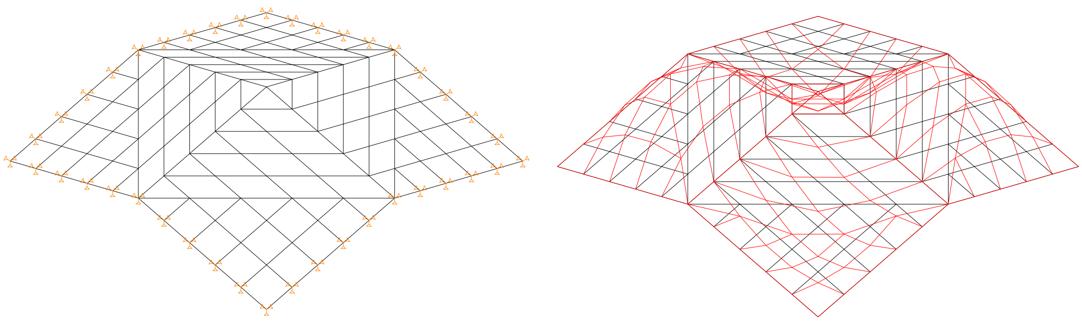
Слика 6.10.2. Деформација примитива од оптерећења у правцу x

$f_{xmaxpr} = 1,63$ cm




Слика 6.10.3. Савијање плоче и њена деформација $f_{uzmax}$

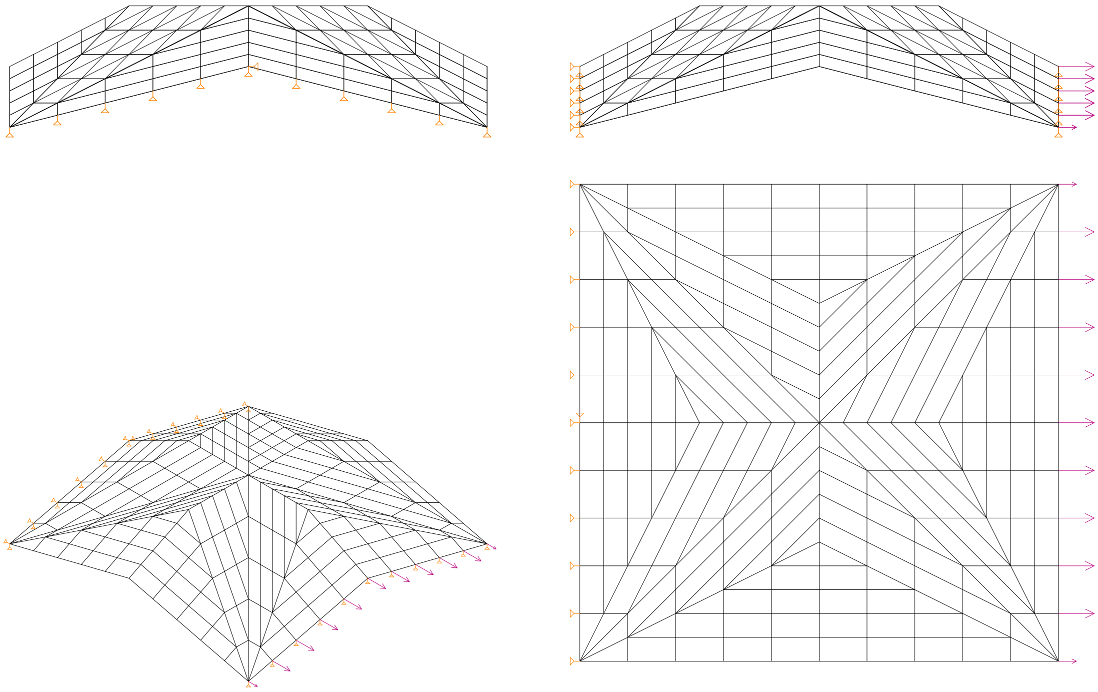
$f_{uzmax}$ = 0,0134 cm



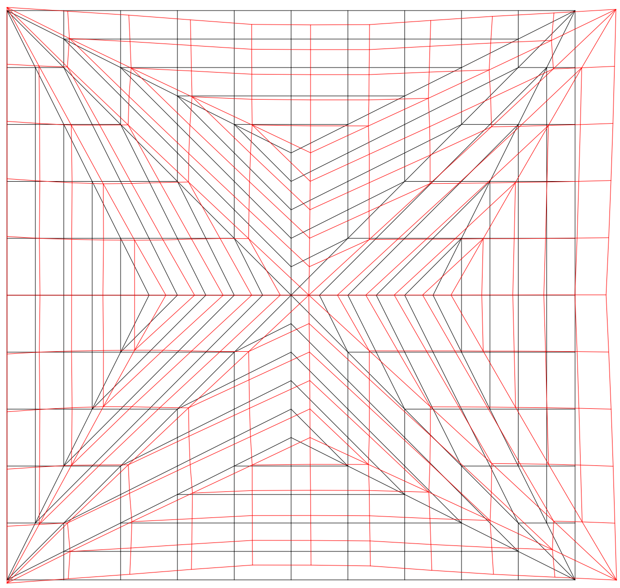
Слика 6.10.4. Први тон осциловања примитива

$f_{01}$ = 188 Hz

### 6.11 Одређивање еквивалентне матрице крутости за примитив P – 11

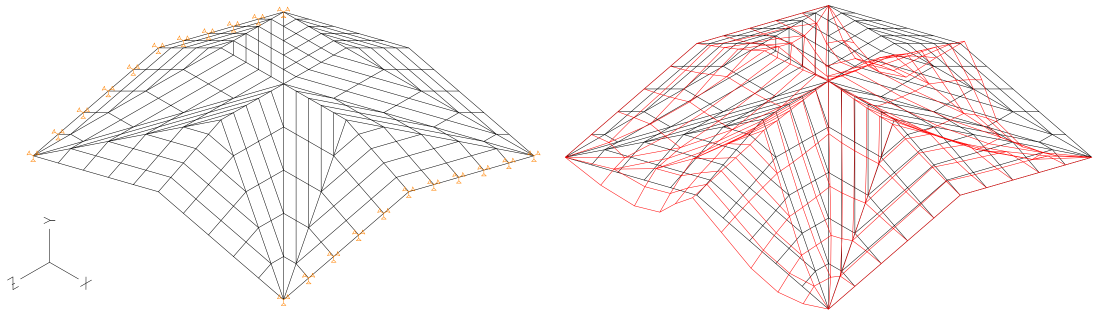


Слика 6.11.1. Оптерећење примитива у x правцу



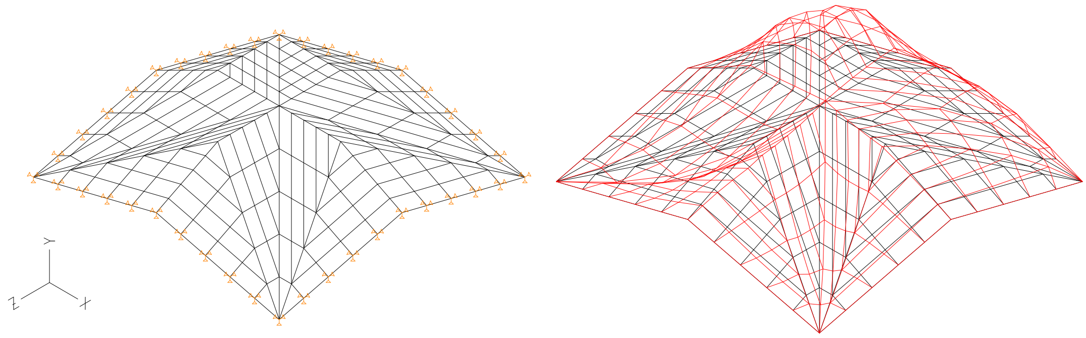
Слика 6.11.2. Деформација примитива од оптерећења у правцу x

$f_{xmaxpr} = 2,3$ cm




Слика 6.11.3. Савијање плоче и њена деформација $f_{uzmax}$

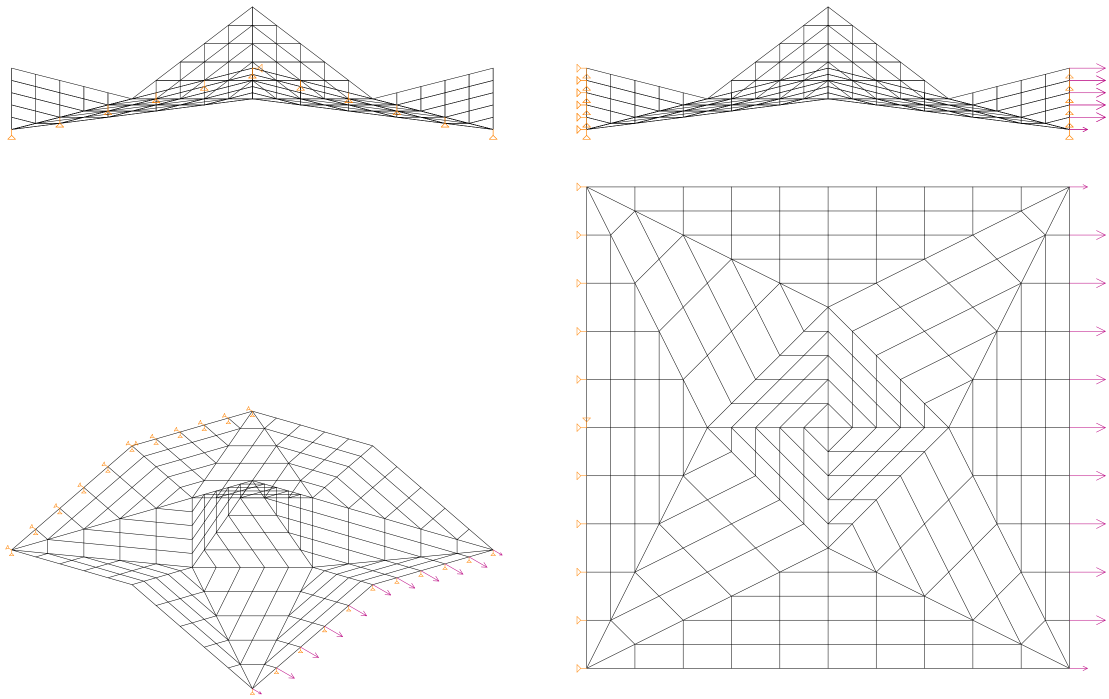
$f_{uzmax} = 0,00344$ cm



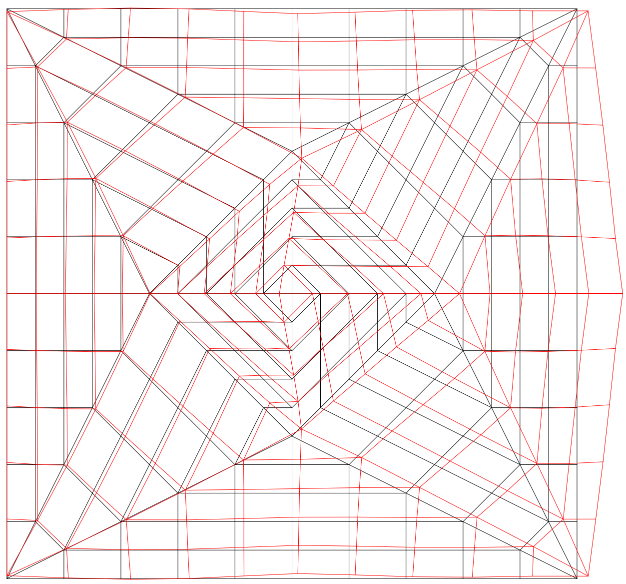

Слика 6.11.4. Први тон осциловања примитива

$f_{01} = 385$ Hz

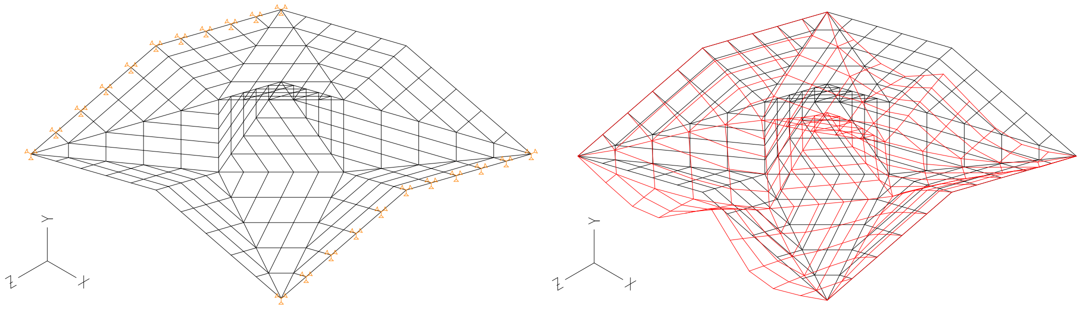
## 6.12 Одређивање еквивалентне матрице крутости за примитив P – 13



Слика 6.12.1. Оптерећење примитива у x правцу

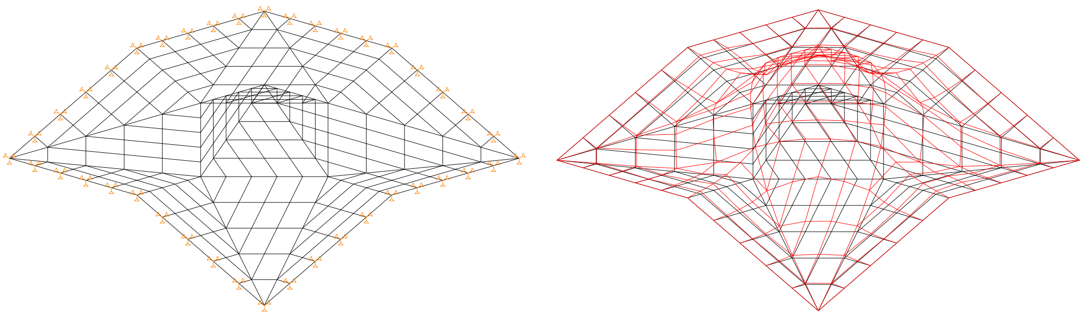


Слика 6.12.2. Деформација примитива од оптерећења у правцу x



Слика 6.12.3. Савијање плоче и њена деформација $f_{uzmax}$

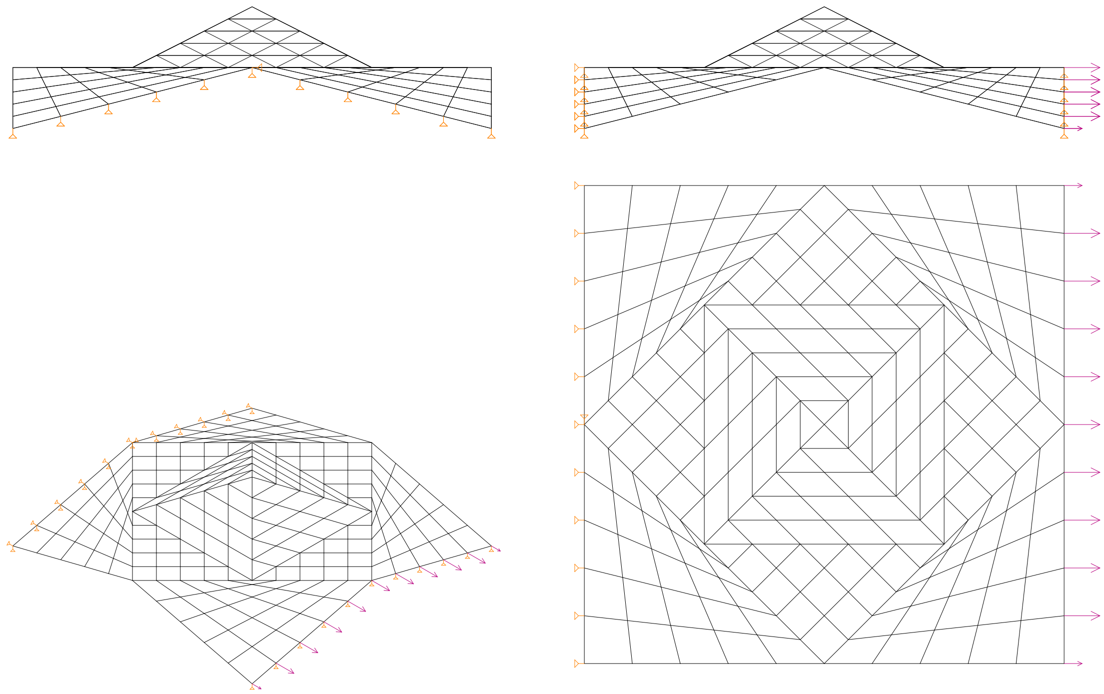
$f_{uzmax}$ = 0,0069 cm



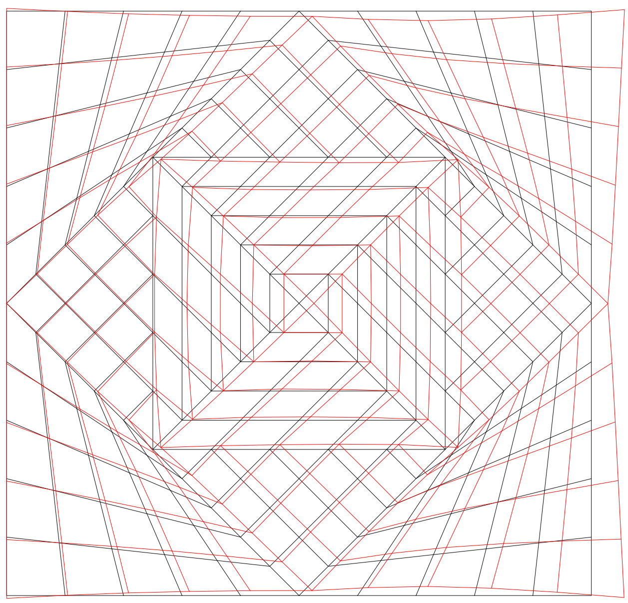
Слика 6.12.4. Први тон осциловања примитива

$f_{01}$ = 440 Hz

### 6.13 Одређивање еквивалентне матрице крутости за примитив Р – 14

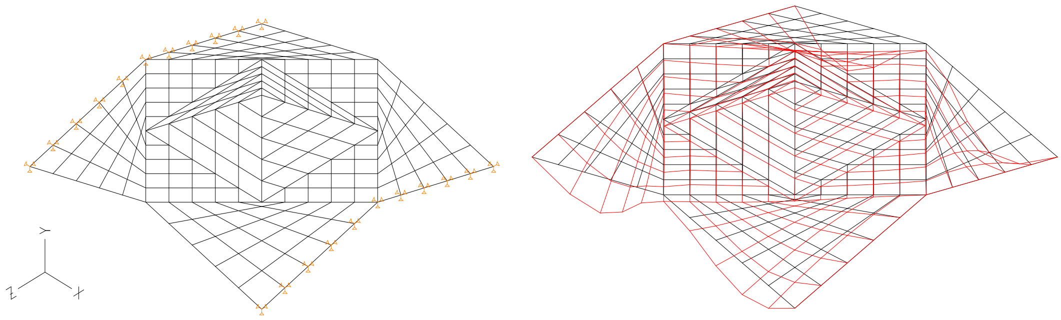


Слика 6.12.5. Оптерећење примитива у х правцу



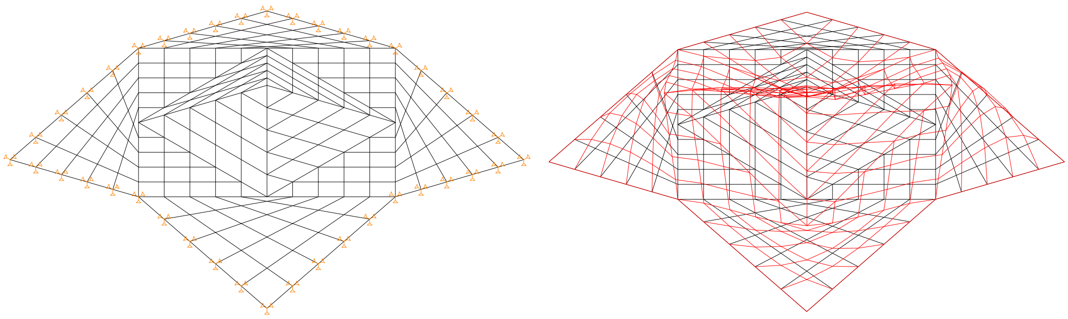
Слика 6.12.6. Деформација примитива од оптерећења у правцу х

$f_{xmaxpr}$ = 0,7 cm




Слика 6.12.7. Савијање плоче и њена деформација $f_{uzmax}$

$f_{uzmax}$ = 0,0069



Слика 6.12.8. Први тон осциловања примитива

$f_{01}$ = 266 Hz

# 7. КОНЦЕПТУАЛНИ ПРИКАЗ ФАЗА ИСТРАЖИВАЊА

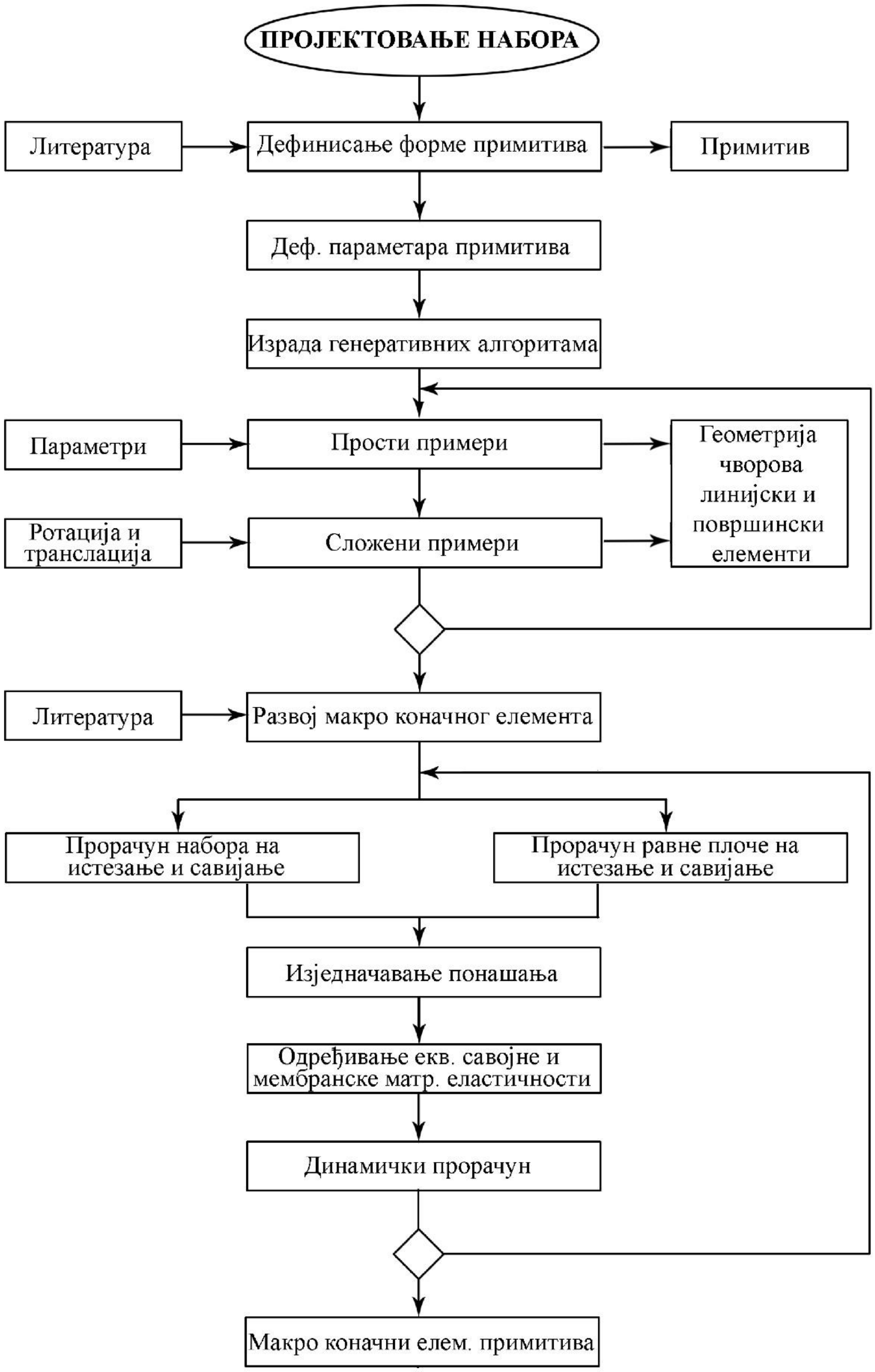
Пројектовање набораних конструкција је комплексан процес који изискује мултидисциплинаран приступ. Први корак се састојао од изучавања геометрије набора коришћењем расположиве литературе. Затим у складу са постављеним хипотезама и претпоставкама извршено је дефинисање форми примитива. На основу дефинисаних циљева врши се избор параметара потребних за дефинисање примитива. Са одређеним параметрима се улази у израду генеративног алгоритма. Програмски алгоритам је написан у Фортран програмском језику и као такав интегрисан у програмски пакет Комипс.

Након израде алгоритма уносом различитих параметра формирају су примери настали само од транслације примитива. Затим, комбинацијом транслације и ротације добијамо сложене примитиве. Крајњи резултат генерисања је стварање чворних тачака, линијских и површинских елемената неопходних за прорачун методом коначних елемената. Представљени поступак понављамо за све примитиве и неколико сложених форми набора по избору аутора.

Следећи корак представља развој макро коначног елемента, при чему се ослањамо на већ познате чињенице из литературе о теорији еластичности, методи коначних елемената и групно суперматричној процедури. Врши се прорачун примитива набора и равне плоче на савијање и истезање. Потом се изједначавају понашања оба елемената, тако да деформације имају исту вредност. На основу добијених вредности срачунавамо савојну и мембранску еквивалентну матрицу еластичности. Примитиви углавном поседују анизотропне карактеристике. На крају ради верификације резултата, као и поређења капацитета носивости примитива, врши се динамичка анализа и одређује се први тон осциловања сопствене фреквенце. Овај поступак је неопходно поновити за сваки дефинисани примитив.

Када су сви кораци успешно извршени можемо сматрати да је формиран макро коначни елемент примитива у виду анизотропне равне плоче са еквивалетним матрицама еластичности.

Графички приказ фаза истраживања је приказан следећим алгоритмом:




Слика 7. 1. Алгоритам концептуалног приказа фаза истраживања

# 8. ЗАКЉУЧНА РАЗМАТРАЊА

## 8.1 Анализа резултата

Уколико сагледамо најшири оквир истраживања, докторска дисертација ставља акценат на унапређење процеса архитектонско – инжењерског пројектовања. Другим речима, домен истраживања чини развој и примена научних метода и њихово интегрисање у пројектовање високо економичних конструктивних система. Може се рећи да је у ради успешно остварена имплементација савремених научних метода, дигиталне технологије, рачунског моделовања, прорачуна, анализе и симулације. Поље истраживања се ограничава на развој узајамних веза моделирања и статичко – динамичког прорачуна набораних конструкција.

Иако су принципи набирања преко оригами технике познати од давнина, као и присуство аналогије узајамних веза набора у природи и архитектури, идеја да се набирањем формира конструктивни систем великог распона јавила се тек почетком XX века. Техника набирања поседује потенцијал разноврсног обликовања, што их чини погодним за пројектовање комплексних крутих форми. Тренутно, њихова примена у пракси заузима тек незнатан проценат изведених објеката, што имплицира на постојање великог истраживачког потенцијала.

Генерално рад можемо поделити у три фазе:

- дефинисање форме примитива
- генерисање геометрије набора
- развој макро коначног елемента

Под **дефинисањем форме примитива** се подразумева изучавање геометрије форме набора. У овом делу је описан кратак историјат развоја и приступ пројектовања набора, затим изложене су теоријске основе за обликовање набора и поделе набора по различитим параметрима. На крају је само констатован потенцијал примене набора и преко отпорног момента приближно упоређена носивост са равном плочом.

Неконзистентност процеса пројектовања је веома чест проблем архитектонског развоја. Проблеми су јасно уочљиви у тренутку када је потребно идеју концепта превести у фазу реализације објекта. Комплексне геометријске функције облика често је јако тешко или чак немогуће описати одговарајућим рачунарским моделом. Такође, избор адекватне нумеричке методе може представљати несавладиву препреку. У циљу превазилажења поменутог проблема успостављен је интерактивни однос различитих дисциплина инжењерства, односно архитектуре и њој сродних наука. Обезбеђена је усаглашеност комплетног процеса пројектовања, при чему су задовољени сви архитектонско – инжењерски од почетне геометријске форме, преко функције, до завршног конструктивног система набора.

За предмет истраживања су изабрани наборани конструктивни системи јер је код њих изражена међузависност форме и структуре. Такође, повољне геометријске карактеристике омогућавају велики капацитет носивости. Наборане конструкције се сврставају у групу савремених просторних система и чине конструктивне системе сложених форми и великих распона. Геометрија набора је таква да веома повољно утиче на ефикасност конструкције при савијању, што је радом и показано. Набор смо дефинисали као скуп међусобно повезаних равних плоча у простору.

Набори, као комплексни површински системи још увек недовољно заступљени у пракси, свакако представљају конструкције будућности. Видели смо да је основни проблем креирање одговарајућег генератора, путем ког се описује геометрија наборане конструкције. Интегрисање генеративног приступа пројектовању подразумевала је поставку процедуре, израду програмског алгоритма, повезивање са већ постојећим програмским пакетом и евалуацију кроз примере.

Пројектоване форме набора, посебно слободне, често су засноване на различитим принципима пројектовања, што условљава примену одређених апроксимација при генерисању геометрије. Саставни део рада чини и истраживање спроведено на Архитектонском факултету[2] и објављено у часопису. У раду је као основа за пројектовање форме и стицање когнитивног искуства коришћења техника савијања папира. Овакав приступ је имао за циљ лакше разумевање просторне трансформације и дигиталне интерпретације, односно геометријске принципе набирања који касније доприносе развоју пројектовања нових форми. Комплетан процес остварен је захваљујући математичкој и рачунарској формулацији уз разумевање процеса анализе објекта.

Мала дебљина плоче и компликована геометрија набора је уједно и највећи изазов при конструисању. Конвенционалне методе уз недовољну развијеност су и често неадекватне за примену.

У првој фази врши се анализа принципа геометрије и структуре, тј. Задатак чини изучавање различитих тесалација набора. Уз прављење физичких модела од папира. Занимљиви дијаграми набирања се затим интегришу у процес пројектовања. Током фазе трансформације изабрани дијаграми набирања пресликани су на одређену архитектонску форму (Mitra, 2009). На крају је извршена елаборација у виду израде физичког модела пројектованог набора - прототип. Пажња је усмерена на међузависности форма – структура и конструкција – материјализација.

Студија је показала да оригами, као полазна тачка истраживања, је допринела развоју визуелних алгоритама неопходних за израду дијаграма набирања. Уједно, стечено је ново спознајно искуство у виду развоја архитектонских форми. Показан је потенцијал оригами технике као средства за израду генеративне мреже. Закључујемо и да нису све форме погодне за примену, па се пажљивим анализирањем модела од папира већ у раној фази могу издвојити форме које задовољавају захтеве стабилности у погледу структура – форма – простор. Свакако, имплементација оригами технике је моћан алат и полазиште за пројектовање набора. Закључујемо да мултидисциплинарни приступ интегрисан у процес пројектовања чини важан корак за добијање иновативних пројектних решења.

---

[2]Nestorović, M., Milošević, J., Nestorović, P., Maneski M. (2016)Instrumentalization of origami in construction of folded plate structures -design, researchandeducation; Spatium, No.35, pp. 22-29

У конструктивном смислу наборе третирамо као скуп плоча. Видели смо да су подужни напони у оба правца истог реда величине, док су управни напони занемарљиви. И поред чињенице да је набор тродимензионални проблем, одређеним апроксимацијама га можемо третирати и превести у равански проблем, што је касније у раду и показано.

На једном простом примеру приказано је како се повећава капацитет носивосту плоче нагнуте под углом од 30˚ реда величине 10 пута уз повећање утрошка материјала за 13,3%. Из приказаног је произашло да плочу набора можемо конструктивно третирати као танку плочу.

На основу представљених примера изведених објеката приметно је да набори могу премостити и распоне преко 100 m чиме их сврставамо у групу конструктивних система великих распона.

У **фази генерисања геометрије** конструисано је 15 различитих примитива уз услов да пројекција контуре на хоризонталну раван мора бити правоугаони елемент. Обимом рада је дефинисано да генерисање и вршење комплекснијих математичких операција превазилази обим истраживања и може бити предмет даљег изучавања. Генерисање примитива представљало је систематизован и свеобухватан корак по корак процес, уједно веома осетљив и најобимнији део рада. Ваљаност генерисане геометрије у великој мери зависи од интуитивности инжењера и његовој способности да реално представи стварни систем. Крај је резултовао креирањем новог генеративног алата којим је приказан само мали део могућности оваквог приступа.

Израда генеративног алгоритма изискивала је израду математичког апарата способног да репрезентује геометрију реалног система. Како би генерисање било што једноставније одлучено је да се унос врши преко неколико унапред дефинисаних параметара који у потпуности описују геометрију набора. Резултат креирања оваквог алгоритма се огледа у једноставном упознавању понашања реалног система само на основу промене неколико параметара. Овакав приступ доприноси једноставности при компарацији примитива.

Један део доприноса овог истраживања се огледа у развоју предпроцесорских рутина и модула преко којих је креиран генератор за описивање геометрије набора. Генератором је обезбеђено формирање чворних тачака линијских и површинских елемената чиме се добија математичка формулација погодна за интеграцију у МКЕ. Од прецизности интерпретације реалног модела зависи и тачност напонско деформацијских величина добијених прорачуном путем МКЕ.

Функционалност генератора је приказана преко примера – примитива. Основни проблем се огледао у дефинисању геометријских правила у процесу генерисања, односно способности алгоритма да генерише положај тачака у простору. Након пројектовања форме примитива, према правилима обликовања, врши се дефинисање улазних параметара за сваки примитив. при чему сам примитив не мора да поседује ни једну особину симетрије. Висина примитива може бити и позитивна и негативна величина, односно остављена је могућност пројектовања било конкавне или конвексне форме. Параметарским дефинисањем геометрије и умножавањем примитива транслацијом и ротацијом, остварен је постављен циљ да се помоћу малог броја улазних параметра генерише велики број података формирајући комплексну просторну структуру. Овакав приступ омогућава релативно једноставно пројектовање набора у простору, које иако нуди веома разнолико обликовање и даље поседује ограничења обликовања. Ограничења се огледају у немогућности пресликавања примитива набора на све архитектонске форме, о чему ће бити више речи у поглављу о потенцијалу даљег истраживања.

Генератор врши повезивање плоча примитива у простору, тако да је врло битно усвојити јединствену нумерацију чворова примитива. Уколико је потребно могу се дефинисати и оребрења у пресеку плочу преко параметра $N_0$ у форми линијских елемената. При томе сваки примитив је састављен од најмање две плоче. Координате висине чворова на ивицама примитива који се настављају у простору морају бити исте. Овај услов је неопходан за остваривање континуалности и непрекидности форме. Форме генерисаних примитива су приказане на слици 3.3.2 Описан поступак се може поновити за било коју нову пројектовану форму примитива.

Параметри потребни за генерисање примитива су приказани у поглављу 3.4.

На крају ове фазе израђени су примери настали транслацијом примитива у оба правца, сложени примери настали комбинацијом транслације и ротације и сложени примери комбиновањем форми више сродних примитива.

Последњи део докторске дисертације чини **развој макро коначног елемента**. У садашњости је постала пракса да су инжењери најчешће само корисници програмских пакета развијених од стране експерата из области рачунарских технологија заснованих на стеченим знањима из струке. Да би то било могуће неопходно је у процес креирања укључити експерте из различитих инжењерских струка. Управо архитектура као једна од научних дисциплина мора дати свој допринос при развоју специјализованих софтвера.

Према начину преношења утицаја конструктивне елементе делимо на линијске, површинске и просторне. Раније смо напоменули да се генератором дефинишу површински и линијски елементи и да примитив набора третирамо као систем танких плоча. Преношење оптерећења код набора врши се површином планарних плоча и дуж ивица на месту прелома плоча.

Сматрамо да се наш материјал понаша еластично. Према теорији еластичности важи претпоставка о малим померањима и деформацијама, као и Хуков закон о физичкој линеарности. У раду су приказани основни изрази теорије еластичности за штап и равну плочу. Незаобилазан поступак при прорачуну примитива чини метода коначних елемената. При анализи се користи троугаони коначни елемент са 3 чворне тачке и 9 степени слободи (поглавље 4.3.2). Померања чворова уједно представљају и основне непознате које је потребно одредити. Анализа методом коначних елемената се своди на корак по корак процес.

У поглављу 5 приказан је прорачун методом групно суперматричне процедуре. Неопходан услов за примену ГСП-а је да примитив поседује одређену групу симетрије. Ротацијом око неке осе чворне решетке могуће је довести чворну решетку у нови положај који се у потпуности поклапа са почетним положајем и физички се од њега не разликује. Имајући у види да хоризонтална пројекција наших примитива правоугаоног облика, закључујемо да је најпогоднија примена правоугаоног елемента са четири чворне тачке која припада групи симетрије $C_{2v}$. Применом репрезентације група остварује се значајно упрошћење и смањење обима прорачуна. Како би основни елемент био погодан за примену групно суперматричне анализе, неопходно је одредити еквиваленти елемент.

На теоријским основама теорије еластичности и методе коначних елемената остварено је превођење несиметричне форме примитива у равну анизотропну танку плочу коју називамо еквивалентном плочом. У суштини разматрамо два независна случаја напрезања, мембранско и савојно. Дефинишемо услов да еквивалентна плоча има исте димензије као и примитив. Усваја се челик за материјал при прорачуну чврстоће. Затим, врши се избор коначних елемената и примитиви и еластична плоча се истежу и савијају континуалним јединичним оптерећењем. Одређивање еквивалентне матрице еластичности се врши изједначавањем деформација. Са обзиром да је плоча анизотропна јавља се проблем немогућности тачног одређивања вредности модула клизања. У прорачуну се јавља грешка при одређивању модула клизања пошто поасонови коефицијент у x и z правцу нису исти. Сматрамо да поменута грешка не утиче значајно на крајњи резултат. Тако добијена матрица еластичности се приписује еквивалентној плочи.

Преко динамичке анализе одређивањем сопствене фреквенце првог тона осциловања за савијање извршена је провера валидности поступка. И примитив и еквивалента плоча су при прорачуну слободно ослоњени дуж контуре. Проблем неједнакости маса решен је смањењем специфичне густине материјала примитива, чиме је остварена зависност вредности сопствене фреквенце само од матрице еластичности, што је и услов да компарација сопствених фреквенција буде упоредива. Вредност првог тона осциловања примитива је добар параметар за компарацију њихових чврстоћа. Примећује се да све форме примитива имају знатно већу деформацију при истезању него равна плоча. Из тог разлога треба предвидети употребу оребрења или крутих дијафрагми које би допринели повећању крутости при мембранском стању. Са друге стране, примитиви оптерећени на савијање поседују чврстоћу већу и преко 100 пута у односу на равну плочу. Закључујемо да примитив 11 поседује највећу крутост на савијање.

Овако дефинисан елемент еквивалентне плоче је сада погодан за прорачун путем групно суперматричне процедуре коришћењем одговарајуће групе симетрије.

## 8.2 Смернице за даља истраживања

Могући правци даљег развоја предложеног концепта пројектовања набора су:

- унапређење процеса пројектовања набора – истраживање форме (оригами)
- развој генератора
- израда физичког модела.

**Истраживање форме** набора преко оригами технике поседује огроман истраживачки потенцијал. Она је нашла примену у архитектури као метода за решавање проблема комплексних геометрија. Значајно олакшава разумевање процеса пројектовања различитих просторних геометријских форми. С обзиром да архитектонске форме имају комплексну математичку интерпретацију, управо оригами, као полазиште за анализу геометрије, треба да буде будући правац развоја овог истраживања. Развој оригами шаблона и идентификација мреже набора треба да буде основа при описивању одговарајуће математичке функције за пресликавање геометрије на сложене архитектонске форме. Као највећи проблем за решавање издваја се сложеност активности у фази описивања конструкције, односно генерисања физичког модела објекта коначним елементима генератора. Управо овакав приступ омогућава лакше успостављање математичког апарата уз помоћ којег описујемо геометрију конструкције. Већ у овој фази до изражаја долази искуство и интуитивност инжењера да препозна форме које поседују употребни потенцијал како у естетском контексту, тако и у контексту способности да конструктивни систем премости велики распон.

**Развој генератора** и одговарајућег програмског кода и нових модула је свакако могућ правац даљег деловања. Напоменули смо већ да је најосетљивији део програма генератор. Параметарско генерисање је алат који омогућава пројектовање примитива и просторне структуре уз мали утрошак времена. Израђен генератор поседује лимитиране функције које дозвољавају само примену операција транслације и ротације, и то тако да хоризонтална пројекција просторне форме набора увек чини правоугаоник. Другим речима, примитив дуж једне осе се може умножавати по произвољној функцији апроксимацијом сегмената на линије и делове кружнице, док у другом смеру може се само умножавати транслацијом

Свакако, треба уложити додатне напоре на развоју програмских модулима који ће бити способни пресликати примитив на сложену архитектонску геометријску форму. Описивање просторних тачака могуће неком математичком функцијом, која се вероватно најлакше може математички описати различитим полиномским функцијама у одговарајућим правцима оса, односно изогеометријском анализом (Milošević, 2015). Предмет даљег развоја генеративног алгоритма свакако има огроман потенцијал и неопходан је корак како би се омогућило генерисање разнородних архитектонских форми. Основни циљ генератора треба да буде унос малог броја података, чији резултат се огледа у аутоматској производњи скупа података, тачака и елемената. Генерисање мора да испуни услов и задовољи принципе архитектонског обликовања и конструисања. У контексту аутоматизације, као коначни резултат ове фазе формиран је програмски алгоритам. Програм поседује скуп функција, којима се за дефинисане параметре тачно описује геометрија система, а као излаз добија се скуп података у виду датотеке, односно формирају се тачке и елементи у складу са захтевима за МКЕ прорачун. Закључујемо да истраживање геометрије набора и развој програма за генерисање треба да буде део даљег истраживања.

**Израда физичког модела** обавезно мора бити један од следећих корака. Видели смо да је у раду коришћена израда модела од папира у контексту истраживања геометрије набора. Међутим, ти модели никако нису репрезентативни у смислу верификације прорачунатог напонског стања и матрица еластичности. Сам поступак прорачуна лежи на познатим теоријским основама и поступак одређивања еквивалентног макро коначног елемента је увек иста. Другим речима, сматрамо да је нумеричка анализа конструкције задовољавајуће развијена. Заснована на методи МКЕ, добија се напонско деформација слика примитива. Свака апроксимација при прорачуну вуче за собом могућност одређене грешке. Иако смо путем верификације резултата добили поклапање и тим констатовали валидност примењеног поступка, увек је пожељно резултате потврдити извођењем експеримента на реалном моделу. Испитивањем се долази до емпиријских знања и података који непобитно обезбеђују довољну сигурност добијених резултата. У нашем конкретном случају то би подразумевало испитивање примитива на савијање и истезање и мерење деформацијских величина. Додатно треба размотрити испитивање на физичком моделу форме набора (Buri & Weinamd, 2008) чиме би сматрали да је процес истраживања заокружен. На крају констатујемо да би у циљу верификације валидности поступка било пожељно извршити испитивање на реалном моделу.

## ПРЕГЛЕД АКРОНИМА

| | | |
|---|---|---|
| 1D, 2D, 3D | Једно-, дво-, тродимензионално | |
| FEM, МКЕ | Метод коначних елемената | *Finite Element Method* |
| GSP, ГСП | Групно суперматрична процедура | *Group Supermatrices Procedure* |
| CAD | | *Computer Aided Design* |
| CAM | | *Comupet Aided Manufacturing* |
| NURBS | Неравномеран разионални базни сплајн | *Non Uniform Rational B-Spline* |

**ПРЕГЛЕД ОЗНАКА**

| | |
|---|---|
| $\alpha_i$ | угао у чвору набора |
| У | увала |
| Г | грбина |
| $Ci(u)$ | NURBS кривине |
| $S(u,v)$ | референтна површина |
| $b$ | ширин |
| $d$ | дебљина |
| $W$ | отпорни моменат |
| $R_A$ | реакција ослонца |
| $\Delta$ | укуна вертикална деформација у средини распона |
| $\delta$ | деформација плоче набора у равни |
| $M$ | момент савијања |
| $F$ | укупна сила |
| $T_i$ | чворна тачка набора |
| $I_{i-j}$ | линијски елемент |
| $\mathbf{P_{i-j-k}}$ | равански елемент |
| $f$ | померање |
| $u, v, w$ | компонентална померања |
| $\varepsilon$ | деформација |
| $\sigma$ | напон |
| $dx$, $dy$ | диференцијалне величине правоугаоног елемента |
| $[C]$ | матрица еластичних коефицијената |
| $[D]$ | матрица еластичности материјала |

| | |
|---|---|
| $\lvert\varepsilon_o\rvert$ | вектор иницијалне деформације |
| $\lvert\sigma_o\rvert$ | вектор иницијалног напона |
| $\nu$ | Поасонов (Poisson) коефицијент |
| E | Јангов (Young) модул еластичности материјала |
| G | модул клизања материјала |
| dS, dA | диференцијална површина средње површи |
| $U_1$ | потенцијална енергија - равно стање напона |
| $U_2$ | потенцијална енергија – равно стање деформације |
| $a_i$ | коефицијенти полинома |
| [n] | матрица коефицијената полинома |
| t | дебљина |
| $[k]_e$ | матрица крутости елемента |
| $\{F\}_e^i$ | вектор почетне термичке деформације елемента |
| $\{F\}_e^V$ | Вектор спољшњег запреминског оптерећења елемената |
| $\{F\}_e^P$ | Вектор спољшњег површинског оптерећења елемента |
| $\theta x, \theta y$ | ротације тачака |
| [B] | матрица извода |
| К | групна суперматрица у нормалном облику |
| Ф | генералисана померања |
| F | генералисана оптерећења |
| T | суперматрица трансформације |
| $U_i$ | подпростор |
| $C_2$, $C_{2v}$, $D_{2h}$ | група симетрије |
| G | групна суперматрица |

| | |
|---|---|
| $\pi_i$ | идемпотенти |
| $f_{maxp}$ | максимална деформација плоче - дилатација |
| q | континуално оптерећење |
| $f_{uxmax}$ | максимална деформација плоче – угиб |
| $f_{01}$ | први тон осциловања |
| $f_{xmaxpr}$ | максимална деформација примитива - дилатација |
| $f_{uzmaxpr}$ | максимална деформација примитива – угиб |
| $E_{xx}^{m}$ | еквивалентни модул еластичности – мембрански |
| $[D_e^m]$ | еквивалента матрица еластичности за мембранско напрезање |
| $E_{xx}^{s}$ | еквивалентни савојни модули еластичности |
| $[D_e^s]$ | еквивалентна матрица еластичности за савојно напрезање |

**ПРЕГЛЕД ИЛУСТРАЦИЈА**

**Извори коришћених илустрација**

Слика 2.2.10. https://gobotany.newenglandwild.org/species/sorghum/halepense/

Слика 2.2.11. http://www.thetablebellingham.org/2014/04/table-worship-april-13th-530pm-at-the-fairhaven-library-basement/palm-2/

Слика 2.2.12. https://www.pinterest.com/pin/241998179960524863/

Слика 2.2.13. https://poduckun.net/razgovor-boris-tomulic-%E2%80%93-strastveni-kolekcionar-morskih-skoljaka/

Слика 2.2.14. https://www.123rf.com/photo_20751296_scallop-seashell.html

Слика 2.2.15. http://arquiscopio.com/archivo/2013/02/02/hangares-para-dirigibles-de-orly/?lang=en

Слика 2.2.16. (Jackson, 2011)

Слика 2.2.17. (Buri & Weinamd, 2008)

Слика 2.3.18. (Mitra, 2009)

Слика 2.3.19. (Mitra, 2009)

Слика 2.3.20. (Mitra, 2009)

Слика 2.3.21. (Hagiwara, 2008)

Слика 2.3.22. (Hagiwara, 2008)

Слика 2.3.23. (Mitra, 2009)

Слика 2.3.24. (Mitra, 2009)

Слика 2.3.25. (Mitra, 2009)

Слика 2.3.26. (Mitra, 2009)

Слика 2.3.27. (Mitra, 2009)

Слика 2.4.16. (Buri & Weinamd, 2008)

Слика 2.4.17. (Buri & Weinamd, 2008)

Слика 2.4.18. (Buri & Weinamd, 2008)

Слика 2.4.19. (Buri & Weinamd, 2008)

Слика 2.5.16 (Mitra, 2009)

Слика 2.5.17. (Schueller, 1983)

Слика 2.5.18. (Schueller, 1983)

Слика 2.5.19 (Schueller, 1983)

Слика 2.5.20. (Muljadinata & Darmawan, 2016)

Слика 2.7.1.1. http://www.statefarmcenter.com/arena-info/about-state-farm-center

Слика 2.7.1.2. https://www.illinoisloyalty.com/Forums/showthread.php?p=1124951

Слика 2.7.1.3. http://www.news-gazette.com/image/2015-09-11/20150910-195547-pic-692047293jpg.html

Слика 2.7.1.1. http://www.worldarchitecturemap.org/buildings/gunma-music-center

Слика 2.7.1.1. http://www.japan-photo.de/mod-au39-08-gunma-music-center-takasaki.htm

Слика 2.7.2.3. http://www.japan-photo.de/mod-au39-08-gunma-music-center-takasaki.htm

Слика 2.7.3.1. https://www.floridamemory.com/items/show/58828

Слика 2.7.3.2. https://miami.curbed.com/2013/12/9/10169544/marine-stadium-marvin-aguilar

Слика 2.7.3.3. https://www.wmf.org/project/miami-marine-stadium

Слика 2.7.4.1. https://www.archdaily.com/554132/ad-classics-yokohama-international-passenger-terminal-foreign-office-architects-foa

Слика 2.7.4.2. https://www.archdaily.com/554132/ad-classics-yokohama-international-passenger-terminal-foreign-office-architects-foa

Слика 2.7.4.2. https://www.archdaily.com/554132/ad-classics-yokohama-international-passenger-terminal-foreign-office-architects-foa

Слика 2.7.5.1. http://www.worldarchitecturemap.org/buildings/american-concrete-institute-building

Слика 2.7.5.2. http://www.worldarchitecturemap.org/buildings/american-concrete-institute-building

Слика 2.7.5.3. http://www.worldarchitecturemap.org/buildings/american-concrete-institute-building

Слика 2.7.6.1. http://atlasofplaces.com/Kraftwerk-Birsfelden-Hans-Hofmann

Слика 2.7.6.2. http://atlasofplaces.com/Kraftwerk-Birsfelden-Hans-Hofmann

Слика 2.7.6.3. http://atlasofplaces.com/Kraftwerk-Birsfelden-Hans-Hofmann

Слика 2.7.7.1. https://inspiration.detail.de/muelimatt-sports-education-and-training-centre-in-windisch-brugg-106156.html?lang=en

Слика 2.7.7.2 https://inspiration.detail.de/muelimatt-sports-education-and-training-centre-in-windisch-brugg-106156.html?lang=en

Слика 2.7.7.3 https://inspiration.detail.de/muelimatt-sports-education-and-training-centre-in-windisch-brugg-106156.html?lang=en

Слика 4.2.10. (Maneski, 1998)

Слика 4.2.11. (Maneski, 1998)

Слика 4.2.12. (Maneski, 1998)

Слика 4.2.13. https://en.wikipedia.org/wiki/Plane_stress

Слика 4.2.14. https://en.wikipedia.org/wiki/Plane_stress

Слика 4.3.6. https://www.xsftruss.com/triangle-truss/

Слика 4.3.7. (Maneski, 1998)

Слика 4.3.8. (Maneski, 1998)

Слика 5.2.2. (Zloković G. , 1992)

Слика 5.3.2. (Zloković G. , 1992)

Слика 5.4.2. (Zloković G. , 1992)

**ПРЕГЛЕД ТАБЕЛА**

## ЛИТЕРАТУРА


Albers, J. (1952). Concerning Fundamental Design, in Bayer. У G. W. H. (Ур.), *Bauhaus 1919-1928* (стр. 114-121). Boston.

Banghay, S. (2000). From Virtual to Physical Reality with Paper Folding. *Computational Geometry Theory and Applications*, 161-174.

Bechthold, M. (2008). *Innovative Surface Structures: Technologies and Applications.* Taylor & Francis.

Buri, H., & Weinamd, Y. (2008). Origami - Folded Plate Structures,. Преузето са http://www.ewpa.com/archive/2008/june/paper_286.pdf

Dimitrov, D., Schreve, K., & de Beer, N. (2006). Advances in Three Dimensional Printing - State of the Art and Future Perspectives. *Rapid Prototyping Journal, 12*(3), 136-147.

Ebara, M., & Kawaguchi, K. (2003). Deployable Solid: A New Folding Structure. *Forma*(18), 187-195. Преузето Feb 23, 2014 са http://www.scipress.org/journals/forma/pdf/1803/18030187.pdf

Engel, H. (1967). *Structure Systems.* Sttutgart: Deutche verlan anstalt GmbH.

Gaćeša, B., Maneski, T., Milošević-Mitić, V., Nestorović, M., & Petrovic, A. (2014). Influence of Furnace Tube Shape on Thermal Strain on Fire-Tube Boilers. *Thermal Science*, 39-47.

Gomez, R. (.). *Design of Folded Plates*. Преузето са PDHonline Course S275: www.pdhonline.org

Hachem, C., Karni, E., & Hanor, A. (2004). Deployable Structures in Nature: Examples, Analysis and Realizations. *Journal of IASS, 45*(3), 190-198.

Hagiwara, I. (2008). From Origami to Origamics, The Japan. *The Japan, 5*(3), 22-25.

Hull, T. (2002). The Combinatorics of Flat Folds: A Survey. У T. Hull (Ур.), *Origami3: Third International Meeting of Origami Science, Mathematics, and Education* (стр. 29-38). Natick, MA: AK Peters.

Jackson, P. (2011). Folding Techniques for Designers: From Sheet to Form. UK: Laurence King Publishers.

Khademzadeh, H. R., & Mazaheri, H. (2007). Some Results to the Huzita’s Theorems. *International Mathematical Forum 2*(14), 699-704. Преузето Jan 8, 2014 са http://www.m-hikari.com/imf-password2007/13-16-2007/mazaheriIMF13-16-2007-1.pdf

Khoshnevis, B. (.). Automated Construction by Contour Crafting - Related Robotics and Information Technology. *Journal of Automation in Construction, 13*(1), 5-19.

Lang, R. J. (1994). Mathematical Algorithms for origami Design. *Symmetry: Culture and Science, 5*(2), 115-152.

Lang, R. J. (2004). *Robert J. Lang Origami*. Преузето Dec 07, 2011 са http://www.langorigami.com

Lister, D. (2003, 2004). Die Geschichte des Papierfaltens – Eine Deutsche Perspektive 1 und 2. *Der Falte*(35&37). Преузето Feb 23, 2014 са https://www.papierfalten.de/

Maekawa, J. (2008). *Genuine Origami: 43 Mathematically-Based Models, From Simple to Complex* (1st Edition изд.). Tokyo: Japan Publications Trading.

Maneski, T. (1998). *Kompjutersko modeliranje i proračun struktura.* Beograd: Mašinski fakultet univerziteta u Beogradu.

Mathiak, K., & Stingl, P. (1967). *Gruppentheorie.* Braunschweig: Friedr. viewg & Sohn Braunschweig.

Milošević, J. (2015). *Izogeometrijska analiza u morfogenezi površinskih konstruktivnih sistema.* Beograd: Arhitektonski fakultet univeriziteta u Beogradu.

Mitra, A. (2009). *The Grammar of Developable Double Corrugation (for Formal Architectural Applications).* University College London.

Miura, K. (1997). Folds - Its Physical and Mathematical Principles. У K. Miura (Ур.), *Origami Science and Art: Proceedings of the Second International Meeting of Origami Science and Scientific Origami*, (стр. 41-50). Otsu, Japan.

Miura, K. (.). Folds - the Basis of Origami. *Symmetry: Culture and Science, 5*(1), 13-22.

Moussavi, F., Lopez, D., G, A., B, F., R, L., & A, S. (2009). *The function of form.* Actar and Harvard University Graduate Scool of Design.

Muljadinata, A., & Darmawan, S. (2016). Redifining folded plate structure as a form-resistant structure. *ARPN Journal of Engeeniring and Applied Sciences*, 4782-4792.

Nestorović, M. (2000). *Konstruktivni sistemi: principi konstruisanja i oblikovanja.* Beograd: Arhitektonski fakultet univerziteta u Beogradu.

Nestorović, M., Milošević, J., Nestorović, P., & Maneski, M. (2016). Instrumentalization of Origami in Construction of Folded Plate Structures - Design, Research and Education. *Spatium*, 22-29.

Nestorović, M., Nestorović, P., & Milošević, J. (2012). Instrumental Role of Geometry in Design Process of Folded Architectural Structures - Research and Education. *Applied Geometry and Graphics - The Interdepartmental Collection of Proceedings, 90*, стр. 397-402.

Piegl, L., & Tiller, W. (1997). *The NURBS Book* (2nd Edition изд.). Berlin, Heildelberg, New York: Springer–Verlag.

Pottman, H., Asperl, A., Hofer, M., & Kilian, A. (2007). *Architectural Geometry* (1st Edition изд.). Exton, PA: Bentley Institute Press.

Rogačev, O. (2006). *Prilog utvrđivanju veza, principa i analogija prostornih struktura u arhitekturi i prirodi.* Beograd: Arthitektonski fakultet univerziteta u Beogradu.

Schueller, W. (1983). *Horizontal-Span Building* . New York: John Welley & Sons.

Šekularac, N., Ivanović Šekularac, J., & Čikić Tovarović, J. (2012). FOLDED STRUCTURES IN MODERN ARCHITECTURE. *Architecture and Civil Engineering Vol. 10, No 1*, 1-16.

Sekulović, M. (1988). *Metoda konačnih elemenata.* Beograd: Građevinska knjiga.

Sekulović, M. (1991). *Matrična analiza konstrukcija.* Beograd: Građevinska knjiga.

Sorguç, A. G., Hagiwara, I., & Selçuk, S. A. (2009). Origami in Architecture: A Medium of Inquiry for Design in Architecture. *METU JFA, 26*(2). Преузето Dec 5, 2011 са http://www.metu.edu.tr/archive/0258-5316/2009/cilt26/sayi_2/235-247.pdf

Stellman, A., & Greene, J. (2005). *Apllied Software Project Management.* O'Reilly Media, Inc.

Steward, I. (2007). Mathematics: Some Assembly Needed. *Nature, 448*(419). Преузето Dec 5, 2011 са http://www.nature.com/nature/journal/v448/n7152/pdf/448419a.pdf

Stošić, N. (1999). *Generisanje i modelovanje koordiniranih sistema konstrukcija.* Beograd: Arhitektonski fakultet univerziteta u Beogradu.

Vincent, J. (2000). Deployable Structures Found in Nature: Potential for Biomimicking. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science,, 214*, стр. 1-10.

Wheen, R. J. (1980, Apr 17). *Australia Patent Application Бр. патента PE 3167.*

Z-Corporation. (.). *Spectrum ZTM510 - 3D Printing System*. Преузето Dec 5, 2011 са http://www.zcorp.com

Zienkiewicz. (1977). *The Finite Element Method.* Maidenhead: McGraw-Hill Book Company.

Zloković, Đ. (1969). *Koordinirani sistem konstrukcija.* Beograd: Građevinska knjiga.

Zloković, Đ. (1969). *Prostorne strukture.* Beograd: Građevinska knjiga.

Zloković, Đ. (1973). *Teorija gurpa i G-vektorski prostori u oscilacijama, stabilnosti i statici konstrukcija.* Beograd: Centar za nalizu i projektovanje prostornih sistema ispu.

Zloković, Đ., Maneski, T., & Nestorović, M. (1993). Analiza grupno supermatričnim postupkom implementiran u CAD i GSP primenom grupnih supermatrica u metodi konačnih elemenata. *XX JUMEH.* Kragujevac.

Zlokovic, D., Maneski, T., & Nestorovic, M. (1994). CAD i GSP primenom grupnih supermatrica u metodi konačnih. *I CAD forum.* Novi Sad.

Zloković, Đ., Maneski, T., & Nestorović, M. (1995). Grupno teorijska analiza ostvarena simetrizovanjem. *XXI JUMEH.* Niš.

Zloković, G. (1989). *Group theory and G-vector spaces in structural analysis.* Chichester: Elis Horwood limited.

Zloković, G. (1992). *Group supermatrices in finite element analysis.* 1992: Ellis Horwood limited.

Zloković, G., Maneski, T., & Nestorović, M. (1994). Group Supermatrix Procedure in Computing of Engineering Structures. *Structural Engineering Review*, 39-50.

Zloković, G., Maneski, T., & Nestorović, M. (1998). Analysis of Nonsymetrical Structures by the Group Supermatrix Procedure Developed on Symmetry Groups. *Strengt Theory: Application, Development & Prospects for 21st Century*, 1149-1153.

Zloković, G., Maneski, T., & Nestorović, M. (1999). Group Theoretical Formulation of Nonsymmetrical Systems by the Group Supermatrix Procedure. *Computers and Structures*, 637-649.

Zloković, G., Maneski, T., & Nestorović, M. (2004). Group theoretical formulation of quadrilateral and hexahedral isoparametric finite elements. *Computers and Structures*, 883-899.

## БИОГРАФИЈА АУТОРА

**Милош Манески**, дипломирани грађевински инжењер за конструкције рођен је 18. марта 1983. г. у Београду. Након завршене XIII београдске гимназије, уписао је 2002. г. Грађевински факултет Универзитета у Београду на коме је дипломирао 2009. г. са просечном оценом 8,02 и оценом 10 на дипломском раду. Током студирања, као један од студената који се истицао успехом, остварио је право на стручну праксу у иностранству. Стручну праксу у трајању од 14 месеци провео је усавршавајући се у Бечу, Аустрија у једном од највећих светских грађевинских концерна, фирми Штрабаг. За време трајања стручне праксе искористио је права мировања обавеза на факултету.

Докторске студије на архитектонском факултету уписује школске 2009/10. г. Као усмерење докторских студија кандидат је одабрао студије научног карактера, област истраживања архитектура, ужа научна област конструктивни системи. 2013. године добија сагласност Већа научних области грађевинско-урбанистичких наука Универзитета у Београду на предлог докторске дисертације под насловом „Развој макроконачног елемента применом групно суперматричне процедуре за наборане конструкције“ (бр. 61206-6023/2-13 од 10.12.2013.). За ментора је именован проф. др Миодраг Несторовић.

Стручни испит је положио 2014. Године (Уверење бр. бр. 11479-3/9927 од 15.11.2014.;), поседује лиценцу одговорног извођача радова грађевинских конструкција и грађевинско-занатских радова на објектима високоградње, нискоградње и хидроградње (бр. 410 Ј121 15 од 12.03.2015). Новембра 2015. године поднео је захтев за издавање лиценце одговорног пројектанта грађевинских конструкција објеката високоградње, нискоградње и хидроградње (бр. 310)

Течно говори енглески и немачки језик, а служи се руским и бугарским језиком.

### Професионално искуство:

Досадашњи професионални рад кандидата карактерише разноврсно искуство у различитим сегментима струке. Након дипломирања наставио је сарадњу са фирмом Штрабаг где је и данас запослен. Рад у фирми Штрабаг допринео је стручном усавршавању у области управљањем пројектима, организацији и технологије изградње. Такође, учествовао је као сарадник на пројекту у изради неколико генералних пројекта, односно статичког прорачуна конструкција.

Поред рада у пракси кандидат је учествовао као студент докторских студија у настави на Архитектонском факултету од 2009. до 2015. године. Истовремени практични рад у области изградње инфраструктуре, пројектовању и прорачуну конструкција уз научно ангажовање и волонтерски рад у настави на Архитектонском факултету Универзитета у Београду, омогућили су му јединствен спој науке и праксе и додатно су допринели комплетном професионалном развоју кандидата.

**ПРИЛОЗИ**

```
      INCLUDE 'MSFLIB.FD'

      INCLUDE 'ZAJZONM.FOR'
      INCLUDE 'ZAJZON1M.FOR'
      include 'ZAJZON4M.FOR'
      integer*2 dummy2,dummy

      EXTERNAL    MouseEvent

      CHARACTER*32  NA1,NA3,NA4
      CHARACTER*15 NASL7,UUD,IME2(50)
      CHARACTER*71 TEXT(14)
      CHARACTER*80
NASLL,NSSLL,NASL3,NASLD,NASL6
      CHARACTER*1  NA2
      CHARACTER*80 IME,IME1
        CHARACTER*20 IMEM
        CHARACTER*256 IME4
        CHARACTER*1 IME3(128)
        LOGICAL*4 CH
          LOGICAL               modestatus
            TYPE (windowconfig)   vc

        integer*2 numfonts
        character*2 fnt
      record /fontinfo/ fontdata

      call savuns(0,11)
        IF(PW1.EQ.0.) PW1=0.1
        IF(PW2.EQ.0.) PW2=0.1
        IF(PW3.EQ.0.) PW3=0.1
      NTAC=20000
        L20=50
        L25=50
        LEN=10
        NOPT=20
        NPRIM=20
        IME1=IMEDIR

        MTOT1= 600000
        MTOT2= NTAC*(NOPT*NOPT+73)+NTAC/2
        MTOTX=MTOT2

        PI=4.000*ATAN(1.000)
        PIRADST=180.000/PI
        PISTRAD=PI/180.000

          vc.numxpixels=-1
          vc.numypixels=-1
          vc.numtextcols=-1
          vc.numtextrows=-1
          vc.numcolors=-1
          vc.fontsize=-1
          vc.title = " Tasko Maneski "C  !
blank

          modestatus=SETWINDOWCONFIG(vc)
          IF(.not.modestatus)
modestatus=SETWINDOWCONFIG(vc)

          modestatus=GETWINDOWCONFIG(vc)

        KKK=0
   21     mmx = vc.numxpixels  -1
          mmy = vc.numypixels  -1
           mx = vc.numtextrows -1
           my = vc.numtextcols -1
      dummy=displaycursor($gcursoroff )
        IIII=1
        IF(mmx.GT.1000) IIII=2
        IF(mmx.GT.1200) IIII=3
      numfonts =
registerfonts('msfonts.fon')
        fnt='n1'
      i=setfont(fnt)
        dummy =getfontinfo(fontdata)
      ah=1.195
        fontdata.pixheight=float(mmy/mx)*ah

fontdata.pixwidth=fontdata.pixheight*0.422
        anh=fontdata.pixheight
      ans=fontdata.pixwidth
      if(ans.eq.0) ans=anh*0.422
        nkol=(my-2)/2-1

   23 CALL SAVUNS(0,11)
        DO 221 I=1,MTOT1
  221 AA(I)=0.0
        DO 222 I=1,MTOT2
  222 BB(I)=0.0
        NQQ(1)=0
        NQQ(2)=0

      OPEN
(11,FILE='\KOMIPS\BOJE.TXT',STATUS='OLD',ER
R=22)
      READ (11,'(10I10)')
NN1,NN2,NN3,NN4,NN5,NN6,NN7
        READ (11,'(10F10.0)')
anh1,ans1,PW1,PW2,PW3
      CLOSE(11)

   22 KKK=1
        iny=15
         iny=0
       iak=setcolor  (iny)
       ak=RECTANGLE(
$GFILLINTERIOR,0,0,mMX,mMY)
      NSSLL=' '
        IPROZ=0
        II8=0
        if (NN1.eq.0) NN1=7
        if (NN2.eq.0) NN2=3
        if (NN3.eq.0) NN3=14
        if (NN4.eq.0) NN4=15
        if (NN5.eq.0) NN5=1
        if (NN6.eq.0) NN6=0
        if (NN7.eq.0) NN7=15

        if (anh1.ne.0) anh=anh1
        if (ans1.ne.0) ans=ans1
        IF(PW1.EQ.0.) PW1=0.10
        IF(PW2.EQ.0.) PW2=0.10
        IF(PW3.EQ.0.) PW3=0.10
        fontdata.pixheight=anh
      fontdata.pixwidth =ans

      CALL MASKA(0)

        CH=CHANGEDIRQQ(IMEDIR)
        NASLD=IMEDIR
        IND1=INDEX(IMEDIR,' ')
        IND2=INDEX(ULIZL,' ')
        DO 103 I=IND2,15
  103 NASLD(IND1+I-IND2+2:IND1+I-
IND2+2)=ULIZL(I-IND2+1:I-IND2+1)

        IRAD=3

      IRID=19
        IKOL=5

      NASL3    ='K  O  M  I  P  S
        +     www.mas.bg.ac.rs'
      TEXT( 1) =  'F1  Preprocesor -
Lokalno Generisanje'
      TEXT( 2) =  'F2  Preprocesor -
Globalno Generisanje'
```

```
      TEXT( 3) =  'F3  Graficki Prikaz Pre
- Post procesor'
      TEXT( 4) =  'F4  Procesor MKE-GSP -
Statika - Dinamika - Termika'
      TEXT( 5) =  'F5  Postprocesor
'
      TEXT( 6) =  'F6  Radni Direktorijum'
      TEXT( 7) =  'F7  UltraEdit
Editovanje Datoteka Primitiva'
      TEXT( 8) =  'F8  UltraEdit
Editovanje Ulazno-Izlaznih Datoteka'
      TEXT( 9) =  'F9  Wordpad Editovanje
Ulazno-Izlaznih Datoteka'
      TEXT(10) = '  0  Kopiranje-
Arhiviranje-Brisanje-Dos Komanda-Setova
       +nje OsnovnoIme'
      NASL6='Q / Esc > Izlaz , Funk.Taster
, Broj , G-gore / D-dole + En
       +ter'
      CALL
OKVIR(5,3,16,61,4,NASL3,4,NASL6,NA1,NA2,NA3
,NA4)
      call okvir1(0,64,2,IND1+2,'
',0.3,0.6)
      CALL ZAPIS (1,66,IMEDIR,-NN5)
      CALL ZAPIS (2,66,ULIZL ,-NN3)
      CALL MENI (6,4,10,61,TEXT,IRAD)
      IF   (IRAD.EQ.0) GOTO 100
       CALL
MENI1(6,4,10,61,TEXT,IRAD,NASL3,NASL6)
       GOTO (2,2,4,2,6,1,8,7,7,9) IRAD
    1 IDIR=0
   11 IME1=' '
       II8=II8+1
      IF(IDIR.NE.1)
IIT=GETDRIVEDIRQQ(IMEDIR)
      NASL3='Radni Direktorijum > Enter
tekuci / . visi Direktorijum / i
       +me Poddirektor.'
      IF(IDIR.EQ.1) NASL3='Ne Postoji Radni
Direktorijum'
      IF(IDIR.EQ.2) NASL3='Ne Mozete Raditi
u ovom Direktorijum'
      NASLL='Enter > Izlaz   '
      CALL OKVIR1(21,3,26,61,NASLL,.5,.0)
      NASLL='Unesi Ime Radnog
Direktorijuma'
       IRID=21
       IKOL=5
      CALL ZAPIS(IRID,IKOL,NASL3 ,NN5)
      CALL ZAPIS(IRID,IKOL,IMEDIR,NN3)
      IF(IDIR.GE.1) CALL GRESKA(1)
      IF(IDIR.GE.1) IRID=IRID+1
       call READR(0)
       IF(terk     .EQ.' ') GOTO 17
       LKX1=1
   15  IF(terk(LKX1:LKX1).EQ.'.') THEN
       INDX2=INDEX(IMEDIR,' ')
       DO 20 I=INDX2,1,-1
      IF(IMEDIR(I:I).NE.'\') GOTO 20
       INDX1=I
      GOTO 25
   20 CONTINUE
   25  DO 19 I=INDX1,INDX2
   19 IMEDIR(I:I)=' '
       LKX1=LKX1+1
       IF(terk(LKX1:LKX1).EQ.'.') GOTO 15
      GOTO 17
       END IF
       IF(terk(2:2).EQ.':') GOTO 12
      IMEDIR(1:1)=NASLD(1:1)
       INDE1=INDEX(IMEDIR,' ')
       INDE2=INDEX(terk,' ')
       IMEDIR(INDE1:INDE1)='\'
       DO 18 I=1,INDE2
   18 IMEDIR(INDE1+I:INDE1+I)=terk(I:I)
      GOTO 17
   12  IMEDIR=terk
   17   INDE=INDEX(IMEDIR,'\')

       IF(INDE.EQ.0.AND.(IMEDIR(1:1).EQ.NA
SLD(1:1))) THEN
       IDIR=2
       GOTO 11
       END IF
       IF(IMEDIR.NE.' ')
CH=CHANGEDIRQQ(IMEDIR)
       IRAD=2
      IF(IMEDIR.NE.' ')
IIT=GETDRIVEDIRQQ(IME1)
       CALL SAVUNS(1,11)
      IF(IMEDIR.EQ.IME1) GOTO 23
   16   IDIR=1
       GOTO 11
    2   call geoMilos(NTAC,LEN)
      call GRAFTM(NTAC,LEN,NOPT,IIII)
       goto 23
c    3  call geotmb(NTAC,LEN,NOPT,NPRIM)
      goto 23
    4 call GRAFTM(NTAC,LEN,NOPT,IIII)
      GOTO 23
    5 continue
      IRRD=1
       NASL3    =        'Procesor MKE-
GSP'
      NASL6   =    'Q / Esc> Izlaz'
      TEXT( 1) =  'F1  Osnovni Stat / Dinam
- icki Proracun'
      TEXT( 2) =  'F2  Dinamicki -
Vremenski Domen'
      TEXT( 3) =  'F3  Dinamicki -
Frekventni Domen'
      TEXT( 4) =  'F4  Dinamicki -
Spektralna Analiza'
      TEXT( 5) =  'F5  Termicki -
Prostiranje Toplote'
      CALL
OKVIR(10,17,16,35,28,NASL3,49,NASL6,NA1,NA2
,NA3,NA4)
      CALL MENI (11,18, 5,35,TEXT,IRRD)
      IF(IRRD.EQ.0) GOTO 23
      dummy=displaycursor($gcursoron )
C      CALL PROCESOR(IRRD,NTAC)
C      IF(NQQ(2).NE.-1) call
POSTT(0,LEN,NTAC,NOPT)
      dummy=displaycursor($gcursoroff )
       goto 23
    6 CONTINUE
c    6 call POSTT(1,LEN,NTAC,NOPT)
      GOTO 23
    7 NASL3    =        'Editovanje Ulazno-
Izlaznih Dat'
      NASL6   =    'Q / Esc> Izlaz'
      TEXT( 1) =  'F1  Datoteka Tacaka
'
      TEXT( 2) =  'F2  Datoteka Elemenata
'
      TEXT( 3) =  'F3  Datoteka
Opterecenja'
      TEXT( 4) =  'F4  Datoteka Izlaza
'
      TEXT( 5) =  'F5  Datoteka Pomeranja
'
      TEXT( 6) =  'F6  Datoteka Napona
Elemenata'
      TEXT( 7) =  'F7  Datoteka Napona
Tacaka'
      TEXT( 8) =  'F8  Komponente Napona
```

```
Grede'
      TEXT( 9) =  'F9  Proizvoljna
Datoteka'
      CALL
OKVIR(5,17,15,32,27,NASL3,49,NASL6,NA1,NA2,
NA3,NA4)
       CALL MENI (6,18,9,32,TEXT,IRR7)
      IF(IRR7.EQ. 0) GOTO 23
      IF(IRR7.EQ. 9) GOTO 69
         CALL SAVUNS(2,12)
       CALL DAT (4,1,2,INS,ULIZL)
        CLOSE(4)
       IF(INS.EQ.0) IME=ULIZL
       IF(INS.EQ.0) GOTO 60
      NASL3='Editovanje Datoteke'
      NASL7='Enter > Izlaz'
      CALL OKVIR1(21,3,25,61,NASL7,.5,.0)
      NASLL='Osnovno Ime Dat Racunskog
Modela'
        IRID=23
        IKOL=5
      CALL ZAPIS(21, 5,NASL3,NN3)
      CALL ZAPIS(22, 5,NASLL,NN5)
        call READR(0)
        READ(terk,'(A15)') UUD
        IF(UUD.NE.' ') ULIZL=UUD
   60  IME=ULIZL
         NT=INDEX(IME,' ')
         IME(NT:NT)='.'
      GOTO (61,62,63,64,65,66,67,68) IRR7
   61    IME(NT+1:NT+1)='T'
         IME(NT+2:NT+2)=' '
         IME(NT+3:NT+3)=' '
      GOTO 24
   62    IME(NT+1:NT+1)='E'
         IME(NT+2:NT+2)=' '
         IME(NT+3:NT+3)=' '
      GOTO 24
   63    IME(NT+1:NT+1)='O'

         IME(NT+2:NT+2)=' '
         IME(NT+3:NT+3)=' '
      GOTO 24
   64    IME(NT+1:NT+1)='I'
         IME(NT+2:NT+2)='Z'
         IME(NT+3:NT+3)='L'
      GOTO 24
   65    IME(NT+1:NT+1)='P'
         IME(NT+2:NT+2)='O'
         IME(NT+3:NT+3)='M'
      GOTO 24
   66    IME(NT+1:NT+1)='N'
         IME(NT+2:NT+2)='A'
         IME(NT+3:NT+3)='P'
      goto 24
   67    IME(NT+1:NT+1)='N'
         IME(NT+2:NT+2)='C'
         IME(NT+3:NT+3)='T'
      goto 24
   68    IME(NT+1:NT+1)='N'
         IME(NT+2:NT+2)='I'
         IME(NT+3:NT+3)='D'
      GOTO 24
   69 NASL7='Enter > Izlaz'
        IRID=22
        IKOL=5
      CALL OKVIR1(21,3,25,61,NASL7,.5,.0)
      NASLL='Ime Proizvoljne Datoteke'
      CALL ZAPIS(21,5,NASLL,NN5)
        CALL READR(0)
        READ(terk,'(A80)') IME
        IF(IME.EQ.' ') GOTO 21
        goto 24
   24   if(IRAD.EQ. 7.OR.IRAD.EQ.8)
NASLL='c:\Komips\uedit32.exe'
        if(IRAD.EQ. 9              )
NASLL='c:\Komips\wordpad.exe'
      dummy2=runqq(NASLL,IME)
  104   GOTO 23
    8 NASL3    =        'Editovanje
Datoteka Primitiva'
      NASL6   =    'Q / Esc> Izlaz'
      TEXT( 1) =  'F1  Datoteke Tacaka
'
      TEXT( 2) =  'F2  Datoteke Linija'
      TEXT( 3) =  'F3  Datoteke Povrsina'
      TEXT( 4) =  'F4  Datoteke Zapremina'
      TEXT( 5) =  'F5  Datoteke Generisanja
LGP  '
      TEXT( 6) =  'F6  Datoteke Generisanja
GGT  '
      TEXT( 7) =  'F7  Datoteke Generisanja
GGE  '
      TEXT( 8) =  'F8  Datoteke Generisanja
GGO  '
      TEXT( 9) =  'F9  Datoteke
Proizvoljne'
      CALL OKVIR(
5,17,15,32,27,NASL3,49,NASL6,NA1,NA2,NA3,NA
4)
      CALL MENI ( 6,18, 9,32,TEXT,IRR8)
      IF(IRR8.EQ.0) GOTO 23
      IF(IRR8.EQ.9) GOTO 69
       CALL SAVUNS(2,12)
       CALL DAT (4,1,2,INS,ULIZL)
        CLOSE(4)
       IF(INS.EQ.0) IME=ULIZL
       IF(INS.EQ.0) GOTO 80
      NASL3='Editovanje Datoteke'
      NASL7='Enter > Izlaz'
      CALL OKVIR1(21,3,25,61,NASL7,.5,.0)
      NASLL='Osnovno Ime Dat Racunskog
Modela'
        IRID=23
        IKOL=5
      CALL ZAPIS(21, 5,NASL3,NN3)
      CALL ZAPIS(22, 5,NASLL,NN5)
        call READR(0)
        READ(terk,'(A15)') UUD
        IF(UUD.NE.' ') ULIZL=UUD
   80  IME=ULIZL
         NT=INDEX(IME,' ')
         IME(NT:NT)='.'
      GOTO (81,82,83,84,85,86,87,88) IRR8
   81    IME(NT+1:NT+1)='T'
         IME(NT+2:NT+2)=' '
         IME(NT+3:NT+3)=' '
      GOTO 24
   82    IME(NT+1:NT+1)='L'
         IME(NT+2:NT+2)=' '
         IME(NT+3:NT+3)=' '
      GOTO 24
   83    IME(NT+1:NT+1)='P'
         IME(NT+2:NT+2)=' '
         IME(NT+3:NT+3)=' '
        goto 24
   84    IME(NT+1:NT+1)='Z'
         IME(NT+2:NT+2)=' '
         IME(NT+3:NT+3)=' '
        goto 24
   85    IME(NT+1:NT+1)='L'
         IME(NT+2:NT+2)='G'
         IME(NT+3:NT+3)='P'
      GOTO 24
   86    IME(NT+1:NT+1)='G'
         IME(NT+2:NT+2)='G'
         IME(NT+3:NT+3)='T'
      GOTO 24
```

```
   87    IME(NT+1:NT+1)='G'
         IME(NT+2:NT+2)='G'
         IME(NT+3:NT+3)='E'
      GOTO 24
   88    IME(NT+1:NT+1)='G'
         IME(NT+2:NT+2)='G'
         IME(NT+3:NT+3)='O'
      GOTO 24
    9  IRR9=1
        NASL3    =        'Korisnicke
Funkcije'
      NASL6   =    'Q / Esc> Izlaz'
      TEXT( 1) =  'F1  Sabiranje -
Kopiranje Datoteka'
      TEXT( 2) =  'F2  Kopiranje Ulaznih
Datoteka'
      TEXT( 3) =  'F3  Arhiviranje
Datoteka'
      TEXT( 4) =  'F4  Brisanje svih
Nepotrebnih Dat'
      TEXT( 5) =  'F5  Brisanje Rezultata
Proracuna'
      TEXT( 6) =  'F6  Brisanje
Pojedinacnih Datoteka'
      TEXT( 7) =  'F7  Setovanje Osnovnog
ImenaDat'
      TEXT( 8) =  'F8  Dos Komanda'
      CALL OKVIR(
6,17,15,31,28,NASL3,49,NASL6,NA1,NA2,NA3,NA
4)
      CALL MENI ( 7,18, 8,31,TEXT,IRR9)
      IF(IRR9.EQ.0) GOTO 23
        IRID=21
        IKOL=5
        CALL OKVIR1(IRID,IKOL       -2,mx-
1,nkol,' ',.5,-0.0)
        goto (91,92,93,94,95,96,97,98) IRR9

   91 NASL3='Ulazne Datoteke'
      CALL ZAPIS(IRID,IKOL,NASL3,NN5)
        ITEX=0
  910 CALL READR(0)
      IF(terk.EQ.' ') GOTO 911
        READ(terk,'(A15)') IME2(ITEX+1)
        ITEX=ITEX+1
      GOTO 910
  911 IF(ITEX.EQ.0) GOTO 23
      NASL3='Izlazna Datoteka'
  919 CALL ZAPIS(IRID,IKOL,NASL3,NN5)
      CALL READR(0)
      IF(terk.EQ.' ') GOTO 23
        READ(terk,'(A80)') IME

OPEN(12,FILE=IME,STATUS='NEW',ERR=917)
        GOTO 918
  917 NASL3='Datoteka Postoji - Ponovi'
      GOTO 919
  918 DO 912 I=1,ITEX
        write (NASL3,'(i5,2X,A15)')
I,IME2(I)
      CALL ZAPIS(IRID,IKOL,NASL3,NN5)

OPEN(13,FILE=IME2(I),STATUS='OLD',ERR=912)
  920   READ(13,'(128A1)',END=913)
(IME3(K),K=1,128)
       DO 914 J=128,1,-1
        IF(IME3(J).NE.' ') GOTO 915
  914 CONTINUE
  915  IF(J.EQ.0) J=1
      WRITE(12,'(128A1)') (IME3(L),L=1,J)
        GOTO 920
  913 CLOSE(13)
  912 CONTINUE
      CLOSE(12)
        GOTO 23

   92 NASL3='Osnovno Ime Ulazne Datoteke'
      CALL ZAPIS(IRID,IKOL,NASL3,NN5)
      CALL READR(0)
      IF(terk.EQ.' ') GOTO 23
        READ(terk,'(A15)') IME
         NT=INDEX(IME,' ')
         IME(NT:NT)='.'
      NASL3='Osnovno Ime Izlazne Datoteke'
      CALL ZAPIS(IRID,IKOL,NASL3,NN5)
      CALL READR(0)
      IF(terk.EQ.' ') GOTO 23
        READ(terk,'(A15)') IME1
         NT1=INDEX(IME1,' ')
         IME1(NT1:NT1)='.'
       DO I=1,3
               IF(I.EQ.1)  IME (NT +1:NT
+1)='T'
           IF(I.EQ.1)
IME1(NT1+1:NT1+1)='T'
               IF(I.EQ.2)  IME (NT +1:NT
+1)='E'
           IF(I.EQ.2)
IME1(NT1+1:NT1+1)='E'
               IF(I.EQ.3)  IME (NT +1:NT
+1)='O'
           IF(I.EQ.3)
IME1(NT1+1:NT1+1)='O'
      OPEN(13,FILE=IME
,STATUS='OLD',ERR=927)

OPEN(12,FILE=IME1,STATUS='NEW',ERR=927)
      WRITE (NASL3,'(A15,2X,A15)') IME,IME1
      CALL ZAPIS(IRID,IKOL,NASL3,NN5)
  929  READ(13,'(128A1)',END=923)
(IME3(K),K=1,128)
       DO J=128,1,-1
        IF(IME3(J).NE.' ') GOTO 928
        end do
  928  IF(J.EQ.0) J=1
      WRITE(12,'(128A1)') (IME3(L),L=1,J)
        GOTO 929
  923 CLOSE(13)
      CLOSE(12)
      END DO
      GOTO 23
  927 NASL3='Ulaz ne Postoji ili Izlaz
Postoji'
      CALL ZAPIS(IRID,IKOL,NASL3,NN5)
      CALL READR(0)
        GOTO 23

   93 NASL3='Datoteka Arhiviranja'
      CALL ZAPIS(IRID,IKOL,NASL3,NN5)
      CALL READR(0)
      IF(terk.EQ.' ') GOTO 23
        READ(terk,'(A20)') IMEM
        NASLL='\KOMIPS\ARJ.EXE'
        IME4=' A                     *.T
*.L *.P *.Z *.E *.O
        + *.PO1 *.NA2 *.ENE *.LGP *.GGT
*.GGE *.GGO'
      DO 921 I=1,20
  921 IME4(3+I:3+I)=IMEM(I:I)
      dummy2=runqq(NASLL,IME4)
        GOTO 23
   94 CONTINUE
        ia=DELFILESQQ('*.DAT')
        ia=DELFILESQQ('*.NI3')
        ia=DELFILESQQ('*.NC4')
        ia=DELFILESQQ('*.OPN')
        ia=DELFILESQQ('*.GEN')
        ia=DELFILESQQ('*.IZL')
```

```
        ia=DELFILESQQ('*.CVT')
        ia=DELFILESQQ('*.ZAZ')
        ia=DELFILESQQ('*.JED')
        ia=DELFILESQQ('*.SIL')
        ia=DELFILESQQ('*.MAS')
        ia=DELFILESQQ('*.IZP')
        ia=DELFILESQQ('*.IZN')
        ia=DELFILESQQ('*.POM')
        ia=DELFILESQQ('*.NAP')
        ia=DELFILESQQ('*.NID')
        ia=DELFILESQQ('*.NCT')
        ia=DELFILESQQ('*.HGL')
        ia=DELFILESQQ('*.BAK')
        ia=DELFILESQQ('*.ANP')
        ia=DELFILESQQ('*.AN4')
        ia=DELFILESQQ('*.AN5')
        ia=DELFILESQQ('*.AN6')
        ia=DELFILESQQ('*.AN7')
        GOTO 23
   95  CONTINUE
        ia=DELFILESQQ('*.PO1')
        ia=DELFILESQQ('*.NA2')
        ia=DELFILESQQ('*.ENE')
        GOTO 23
   96 CONTINUE
      NASLL='Ime Dat za Brisanje - Enter >
Izlaz'
      CALL ZAPIS(IRID,IKOL,NASLL,NN5)
  941   CALL READR(0)
        READ(terk,'(A80)') IME
        IF(IME.EQ.' ') GOTO 23
        ia=DELFILESQQ(IME)
      goto 23
   97 NASL3='Setovanje Dat'
      NASL7='Enter > Izlaz'
      CALL OKVIR1(21,3,26,61,NASL7,.5,.0)
      NASLL='Osnovno Ime Dat Racunskog
Modela'
        IRID=23
        IKOL=5
      CALL ZAPIS(21, 5,NASL3,NN3)
      CALL ZAPIS(22, 5,NASLL,NN5)
        IF(IRR8.EQ.9) WRITE(NASLL,'(A15,''
Postojece Ime'')') ULIZL
      IF(IRR8.EQ.9) CALL
ZAPIS(IRID,IKOL,NASLL,NN3)
        call READR(0)
        READ(terk,'(A15)') UUD
        IF(UUD.EQ.' ') GOTO 23
        CALL DAT (1,1,2,INS,UUD)
        IF(INS.EQ.0) CLOSE(1)
        IF(INS.EQ.1) CALL GRESKA(1)
        IF(INS.EQ.1) GOTO 98
        ULIZL=UUD
      CALL SAVUNS(1,11)
        GOTO 23
  942 NASLL='Ne Postoji Datoteka - Ponovi'
      CALL ZAPIS(IRID,IKOL,NASLL,NN5)
        GOTO 941
 98
dummy2=runqq('\komips\command.com',' ')
      if(dummy2.ne.0)
dummy2=runqq('\KOMIPS\cmd.exe',' ')
      if(dummy2.ne.0)
dummy2=runqq('\KOMIPS\cmd32.exe',' ')
      goto 23
  100   NASLL='     ***   KOMIPS - Hvala na
paznji   ***'
      CALL ZAPIS(25,2,NASLL,NN5)
      CALL SAVUNS(1,11)

        end
```

```
        subroutine geoMilos(NLT,LEN)
      include 'msflib.fd'
      include  'ZAJZON1M.FOR'
      include  'ZAJZON3M.FOR'
      CHARACTER*63 NASLL,NSSLL
      CHARACTER*256 TEXX
      NSSLL=' '
         NPOD =NLT
        LX = NLT
       LY = NLT
      LW = NLT
      N1=1
       N2=N1+NLT
        N3=N2+NLT
         N4=N3+NLT
          N5=N4+NPOD
           N6=N5+NPOD
             N7=N6+NPOD
             N8=N7+NPOD*3
             J1=1
            J2=J1+LX*2
           J3=J2+LY*4
          J4=J3+LW*8
         J5=J4+NPOD*4
        UDAT='               '

OPEN(8,FILE='GEN.GEO',STATUS='UNKNOWN')
    9 CONTINUE
       CALL
PRIMIT(NLT,X,Y,Z,LEN,LG,LL,LP,LZ,T,DEB,PRIT
,XYZ,
        +              NPOD,LX,LY,LW)
      IF(INP.EQ.4) GOTO 10
      REWIND 8
        INP=5
        IE1=1
         IPROZ=0
        II8=0
        IRID=21
        IKOL=5
   52 L=0
      NLIN=0
      NPOV=0
      NZAP=0
        IVRAT =0
        IPVRAT=0
   51 IUL=0
      IUP=0
      IUZ=0
      IE=0
      IPL=0
      IPP=0
      IPZ=0
      L=L+1
      NLIN=NLIN+1
      NPOV=NPOV+1
      NZAP=NZAP+1
      NXT=0
        NYT=0
         NZT=0
      IP=L
            N1=1
      IEX=0
      IEY=0
      IE5=0
        IF(IVRAT.GE.1) IE=3
        IF(IVRAT.GE.1) GOTO 35
   62 IF(INP.NE.5) READ(INP,'(I5)') IE1
```

```
      IF(UDAT.EQ.' ') UDAT='GEO.GEN'
      CALL DAT (9,26,0,INS,UDAT)
   13 READ (8,4,END=11) TEXX
      WRITE(9,4       ) TEXX
    4 FORMAT(A256)
      GOTO 13
   11 CLOSE(9)
   10 CLOSE(8,STATUS='DELETE')
      IF(INP.EQ.4) CLOSE(4)
      NASLL='                          ***
GEOTMA - Hvala na paznji   ***'
      END

      SUBROUTINE
PRIMIT(NLT,XL,YL,ZL,LEN,LG,LL,LP,LZ,DX,DY,D
Z,DXYZ,
     +                  NPOD,LX,LY,LW)
      include 'msflib.fd'
      INCLUDE 'ZAJZON1M.FOR'
      INCLUDE 'ZAJZON5M.FOR'
      INTEGER*2
LG(NLT,LEN),LL(NLT,LEN),LP(NLT,LEN),LZ(NLT,
LEN*2)
      DIMENSION
XL(NLT),YL(NLT),ZL(NLT),DX(NLT),DY(NLT),DZ(
NLT),
     +          DXYZ(NLT,3)
      CHARACTER*37 TEXT(10)
      CHARACTER*29 NA1,NA3,NA4
        CHARACTER*1  NA2
      CHARACTER*21 NASL2
      CHARACTER*17 NASL3,NASL33
      CHARACTER*29 NASL6
      CHARACTER*25 NASL8
      CHARACTER*18 NASL9
      CHARACTER*23 NASL10
        character*74 NASLL,NASL4,TE1
      IF(INP.NE.5) GOTO 66
        IX1=27
      NASL3   =         'P R I M I T I V I'
    TEXT(1) =  'F1  Povrsinski Primitivi 1'
    TEXT(2) =  'F2  Povrsinski Primitivi 2'
    TEXT(3) =  'F3  Zapreminski Primitivi '
    TEXT(4) =  'F4  Nastavak Primitiva '
    TEXT(5) =  'F5  Operacije Primitiva '
    TEXT(6) =  'F6  Izbor Lin i Pov Elem '
    TEXT(7) =  'F7  Umnozavanje Primitiva'
      TEXT(8) =  'F8  Ime Dat Generisanja '
   TEXT(9) =  'F9  Pov sa otvorom na
Lopti'TEXT(10)= ' 0  Naborane konstrukcije'
      NASL6   ='Q / Esc > Izlaz'
      CALL OKVIR(
5,18,16,27,28,NASL3,21,NASL6,NA1,NA2,NA3,NA
4)
        IF(II8.NE.0) GOTO 4
        II8=1
      CALL OKVIR1(21,3,25,61,'',.5,0.)
      NASL33='Model Primitiva'
      CALL ZAPIS(21,5,NASL33,NN3)
      WRITE(NASLL, 2) NLT,NPOD
    2 FORMAT(  'Ntac = ',I5,'    Npod =
',I5)
      CALL ZAPIS(22,5,NASLL,NN5)
      WRITE(NASLL, 3) LX,LY,LW
    3 FORMAT(  'Nlin  = ',I5,'    Npov =
',I5,'    Nzap = ',I5)
      CALL ZAPIS(23,5,NASLL,NN5)
    4 CALL MENI (  6,19,10,27,TEXT,IE1)
   66 IF(IE1.EQ.0) WRITE(  8,'(I5,'' ;
Izlaz'')') IE1
      IF(IE1.EQ.1) WRITE(  8,'(I5,'' ;
Povrsinski Primitivi 1'')'  ) IE1
      IF(IE1.EQ.2) WRITE(  8,'(I5,'' ;
Povrsinski Primitivi 2'')'  ) IE1
      IF(IE1.EQ.3) WRITE(  8,'(I5,'' ;
Zapreminski Primitiv'')' ) IE1
      IF(IE1.EQ.4) WRITE(  8,'(I5,'' ;
Nastavak Primitiva'')'   ) IE1
      IF(IE1.EQ.5) WRITE(  8,'(I5,'' ;
Operacije Primitiva'')'  ) IE1
      IF(IE1.EQ.6) WRITE(  8,'(I5,'' ;
Izbor Lin i Pov Elem'')' ) IE1
      IF(IE1.EQ.7) WRITE(  8,'(I5,'' ;
Umnozavanje Primitiva'')') IE1
      IF(IE1.EQ.10) WRITE( 8,'(I5,'' ;
Naborane konstrukcije'')') IE1
        IF(IE1.LE.3.AND.IEY.LT.IE1) IEY=IE1
      IF(IE1.EQ.0. AND.IPP.EQ.10) RETURN
      IF(IE1.EQ.0. AND.IEX.EQ.6 ) GOTO 74
      IF(IE1.EQ.0. AND.IE5.EQ.5 ) GOTO 79
      IF(IE1.EQ.0               ) GOTO 70
      IF(IE1.EQ.IE              ) GOTO 62
      iF(IE1.eq.7) goto 701
      iF(IE1.eq.8) goto 80
        if(NXT.eq.0) then
   46 NASL10  = 'Pocetna Tacka Primitiva'
        te1='             Enter > Izlaz'
        CALL OKVIR1(IRID,IKOL       -
2,mx+1,81,' ',.5,-0.0)
      NASL4 = 'Kordinate Poc.Tacke  XP , YP
, ZP , Temperatura , Sistem'
        CALL ZAPIS(IRID,IKOL,NASL4,NN5)
        IF(INP.EQ.5) call READR(1)
      IF(INP.EQ.5) READ (terk,'(5F10.0)')
XL(L),YL(L),ZL(L),TEMP,SIST
      IF(INP.NE.5) READ ( INP,'(5F10.0)')
XL(L),YL(L),ZL(L),TEMP,SIST
      IF(IE1.EQ.9.AND.INP.EQ.5) WRITE
(8,'(''    9'')')
      WRITE(8,'(5F10.3,'' ; Pocetna
tacka'')')
        +
XL(L),YL(L),ZL(L),TEMP,SIST
         CALL
SISTEM(XL(L),YL(L),ZL(L),SIST)
        end if
  701  CALL MENI1( 6,19,
9,28,TEXT,IE1,NASL3,NASL6)

GOTO(10,20,30,70,70,70,700,80,90,910) IE1
      GOTO 62
   10 IF(INP.NE.5) READ(INP,'(I5)') IPRL
      IF(INP.NE.5) GOTO 67
      NASL2   =         'POVRSINSKI
PRIMITIVI 1'
      TEXT(1) =  'F1  Primitiv  1'
      TEXT(2) =  'F2  Primitiv  2'
      TEXT(3) =  'F3  Primitiv  3'
      TEXT(4) =  'F4  Primitiv  4'
      TEXT(5) =  'F5  Primitiv  5'
      TEXT(6) =  'F6  Primitiv  6'
      TEXT(7) =  'F7  Primitiv  7'
      TEXT(8) =  'F8  Primitiv  8'
      TEXT(9) =  'F9  Primitiv  9'
      TEXT(10)= '  0  Primitiv 10'
      NASL6   ='Q / Esc > Izlaz'
      CALL OKVIR(
5,25,16,29,45,NASL2,38,NASL6,NA1,NA2,NA3,NA
4)
       CALL MENI( 6,26,10,29,TEXT,IPRL)
   67 WRITE(  8,'(I5)'       ) IPRL
      IF(IPRL.EQ.0) CALL MENI1( 7,26,
9,28,TEXT,IPRL,NASL2,NASL6)
      IF(IPRL.EQ.0) GOTO 62
        IPL =IPRL
        IPLX=IPRL
        IF(IE.EQ.0) IE=1
      IF(IPL.EQ.1) call
```

```
     +              TALAS1
(NLT,LEN,XL,YL,ZL,LG,LL,LP,L,NLIN,NPOV,IDT)
      IF(IPL.EQ.2) CALL
     +              TALAS2
(NLT,LEN,XL,YL,ZL,LG,LL,LP,L,NLIN,NPOV,IDT)
      IF(IPL.EQ.3) CALL
     +           TALAS3
(NLT,LEN,XL,YL,ZL,LG,LL,LP,L,NLIN,NPOV,IDT)
      IF(IPL.EQ.4) CALL
     +           TALAS4
(NLT,LEN,XL,YL,ZL,LG,LL,LP,L,NLIN,NPOV,IDT)
      IF(IPL.EQ.5) CALL
     +           TALAS5
(NLT,LEN,XL,YL,ZL,LG,LL,LP,L,NLIN,NPOV,IDT)
      IF(IPL.EQ.6) CALL
     +           TALAS6
(NLT,LEN,XL,YL,ZL,LG,LL,LP,L,NLIN,NPOV,IDT)
      IF(IPL.EQ.7) CALL
     +           TALAS7
(NLT,LEN,XL,YL,ZL,LG,LL,LP,L,NLIN,NPOV,IDT)
      IF(IPL.EQ.8) CALL
     +           TALAS8
(NLT,LEN,XL,YL,ZL,LG,LL,LP,L,NLIN,NPOV,IDT)
      IF(IPL.EQ.9) CALL
     +           TALAS9
(NLT,LEN,XL,YL,ZL,LG,LL,LP,L,NLIN,NPOV,IDT)
      IF(IPL.EQ.10)CALL
     +
TALAS10(NLT,LEN,XL,YL,ZL,LG,LL,LP,L,NLIN,NP
OV,IDT)

        CALL
ZICTAC(NLT,L,XL,YL,ZL,1,NXT,TEMP)
         CALL
ZICLIN(NLT,NLIN,LL,2,1,2,LEN)
        CALL ZICPOV(NLT,NPOV,LP,2,1,2,LEN)
       RETURN
      GOTO 62

   20 IF(IE.EQ.1) GOTO 21
   44 IF(INP.NE.5) READ(INP,'(I5)') IPRP
      IF(INP.NE.5) GOTO 68
      NASL2   =         'POVRSINSKI
PRIMITIVI 2'
      TEXT(1) =  'F1  Primitiv 11'
      TEXT(2) =  'F2  Primitiv 12'
      TEXT(3) =  'F3  Primitiv 13'
      TEXT(4) =  'F4  Primitiv 14'
      TEXT(5) =  'F5  Primitiv 15'
      TEXT(6) =  'F6  Primitiv 16 (od
Primitva 1)'
      TEXT(7) =  'F7                 '
      TEXT(8) =  'F8                 '
      TEXT(9) =  'F9                 '
      TEXT(10)= '  0  Primitiv 0     '
      NASL6   ='Q / Esc> Izlaz'
      CALL
OKVIR(5,25,16,29,45,NASL2,38,NASL6,NA1,NA2,
NA3,NA4)
       CALL MENI (6,26,10,29,TEXT,IPRP)
   68  WRITE(  8,'(I5)'       ) IPRP
      IPP =IPRP
      IPPX=IPRP
      IF(IE.EQ.0.OR.IE.EQ.1) IE=2
      IF(IPP.EQ.1) call
     +
TALAS11(NLT,LEN,XL,YL,ZL,LG,LL,LP,L,NLIN,NP
OV,IDT)
      IF(IPP.EQ.2) CALL
     +
TALAS12(NLT,LEN,XL,YL,ZL,LG,LL,LP,L,NLIN,NP
OV,IDT)
      IF(IPP.EQ.3) CALL
     +           F0=180.*PISTRAD
TALAS13(NLT,LEN,XL,YL,ZL,LG,LL,LP,L,NLIN,NP
OV,IDT)
      IF(IPP.EQ.4) CALL
     +
TALAS14(NLT,LEN,XL,YL,ZL,LG,LL,LP,L,NLIN,NP
OV,IDT)
      IF(IPP.EQ.5) CALL
     +
TALAS15(NLT,LEN,XL,YL,ZL,LG,LL,LP,L,NLIN,NP
OV,IDT)
      IF(IPP.EQ.6) CALL
     +
TALAS16(NLT,LEN,XL,YL,ZL,LG,LL,LP,L,NLIN,NP
OV,IDT)
      IF(IPP.EQ.7) CALL
     +           TALAS3
(NLT,LEN,XL,YL,ZL,LG,LL,LP,L,NLIN,NPOV,IDT)
      IF(IPP.EQ.8) CALL
     +           TALAS4
(NLT,LEN,XL,YL,ZL,LG,LL,LP,L,NLIN,NPOV,IDT)
      IF(IPP.EQ.9) CALL
     +           TALAS5
(NLT,LEN,XL,YL,ZL,LG,LL,LP,L,NLIN,NPOV,IDT)
      IF(IPP.EQ.10) CALL
     +
TALAS0(NLT,LEN,XL,YL,ZL,XYZ,LG,LL,LP,L,NLIN
,NPOV,IDT,LZ)

        CALL
ZICTAC(NLT,L,XL,YL,ZL,1,NXT,TEMP)
         CALL
ZICLIN(NLT,NLIN,LL,2,1,2,LEN)
        CALL ZICPOV(NLT,NPOV,LP,2,1,2,LEN)
       RETURN

       IF(IPP.EQ.10) return
       IF(IPZ.EQ. 6) GOTO 35
       CALL
MENI1(6,26,10,29,TEXT,IPRP,NASL2,NASL6)
        GOTO 62
       GOTO 62
   79   IF(L.LE.1) RETURN
         NASL8   = 'Zapis Tac-Elem
PRIMITIVA'
        CALL
ZICTAC(NLT,L,XL,YL,ZL,1,NXT,TEMP)
      NASL8   = 'Zapis Elemenata PRIMITIVA'
      IF(IE.EQ.1.OR.IE.EQ.2.OR.IE.EQ.3)
       +                              CALL
ZICLIN(LX,NLIN,LL,2,IUL,IE,LEN)
      IF(       IE.EQ.2.OR.IE.EQ.3) CALL
ZICPOV(LY,NPOV,LP,2,IUP,IE,LEN)
      IF(                 IE.EQ.3) CALL
ZICZAP(LW,NZAP,LZ,2,IUZ,LEN)
         return
          END
```

**SUBROUTINE UGAO**

```
(X1,X2,Y1,Y2,R0,F0,PISTRAD)

          R0=SQRT((X2-X1)**2 + (Y2-Y1)**2)
          IF(R0.EQ.0) CALL GRESKA(1)
          IF(R0.EQ.0) RETURN
      IF(X2.NE.X1.AND.Y2.NE.Y1) THEN
          F0=ATAN2((Y2-Y1)/R0,(X2-X1)/R0)
          IF(F0.LT.0) F0=360.*PISTRAD+F0
       END IF
      IF(X2.EQ.X1.AND.Y2.GT.Y1) F0=
90.*PISTRAD
      IF(X2.EQ.X1.AND.Y1.GT.Y2)
F0=270.*PISTRAD
      IF(Y2.EQ.Y1.AND.X2.GT.X1) F0=0.0
      IF(Y2.EQ.Y1.AND.X1.GT.X2)
END
```

```
      SUBROUTINE ROT(MR,FR)
      INCLUDE 'ZAJZON1M.FOR'

      DO 742 I=1,3
      DO 742 J=1,3
  742 AR(I,J)=0.0
      AR(MR,MR)=1.0
      M1=MOD(MR,3)+1
      M2=MOD(M1,3)+1
      C=COS(FR)
      S=SIN(FR)
      AR(M1,M1)=C
      AR(M2,M2)=C
      AR(M1,M2)=-S
      AR(M2,M1)=S
      END

      SUBROUTINE XYE(XA,X1,YA,Y1,EX,EY,I)
      INCLUDE 'ZAJZON1M.FOR'
      CHARACTER*57 TEXT
      CHARACTER*57 TEXT1

      TEXT1= 'Centar Kruga i Ekscentriteti'
      TEXT = 'Koordinate centra Kruga XC ,
YC , Ekscentar X , Y'
      IF(I.EQ.2)
     +TEXT = 'Koordinate centra Kruga XC ,
ZC , Ekscentar X , Z'
      IF(I.EQ.3) TEXT=
     + 'Koordinate centra Kruga XC , YC ,
0-1/4 / 1-1/2 / 2- ceo krug'
    3  CALL ZAPIS(IRID,IKOL,TEXT,NN5)
      GOTO 2
    1 CALL GRESKA(1)
      GOTO 3
    2 IF(INP.EQ.5) CALL READR(1)
        IF(INP.EQ.5)
READ(terk,'(4F10.0)',ERR=1) XA,YA,EX,EY
        IF(INP.NE.5) READ
(INP,'(4F10.0)',ERR=1) XA,YA,EX,EY
        if(i.eq.2) goto 4
           R0=SQRT((XA-X1)**2 + (YA-Y1)**2)
           IF(R0.NE.0) GOTO 4
      TEXT1= 'Radius Rotacije = 0.0'
        call zapbris(text1)
         IRID=IRID-1
           goto 3
    4 WRITE(8,'(4F10.3,'' ; Centar Xc , Yc
, Ex , Ey'')' ) XA,YA,EX,EY
      END


      SUBROUTINE
ZICLIN(LX,NLIN,LL,N,IUL,IE,LEN)

c       SUBRUTINA ZA CITANJE (N=1) I ZAPIS
(N=2) DAT LINIJA
c       POZIVA SUBRUTINE :  .DAT  .ZAPIS

      INCLUDE 'ZAJZON1M.FOR'
      INTEGER*2 LL(LX,LEN)
      CHARACTER*15 U1
      CHARACTER*40 TEXT

      U1=UDAT
      INP1=INP

IF(N.EQ.2.AND.(IE.EQ.2.OR.IE.EQ.3).AND.NLIN
.LE.1) RETURN
  102 IF(N.EQ.1) CALL DAT (1,11,2,INS,U1)
      IF(N.EQ.2) CALL DAT (1,11,0,INS,U1)
      IF(INS.EQ.1) GOTO 101
      IF(N.EQ.1) GOTO 11
      IF(N.EQ.2) GOTO 16
  101 INP1=5
      IF(N.EQ.1) TEXT =  'Ime Dat Linija -
Citanje'
      IF(N.EQ.2) TEXT =  'Ime Dat Linija -
Zapis'
      CALL ZAPIS(IRID,IKOL,TEXT,NN5)
   99 IF(INP1.EQ.5) CALL READR(0)
      IF(INP1.EQ.5) READ (terk,'(A15
)') U1
      IF(INP1.NE.5) READ (INP1,'(A15
)') U1
      WRITE(   8,'(A15,'' ; Ime Dat'')') U1
      IF( U1.EQ.'                    ')
RETURN
      GOTO 102
   11 TEXT =  'Citanje LIN Elemenata'
C     CALL ZAPIS(IRID,IKOL,TEXT,NN5)
      READ(1,'(14I5)')
NIE,NL1,(IMX(I),I=1,12)
      IF(NLIN.EQ.1.AND.IUL.EQ.1) NLIN=0
       IF(IUL.EQ.1) NLIN=NLIN+NL1
       DO 13 I=1,12
       DO 13 J=1,IMX(I)
       IF(I.EQ.4) READ(1,*)
   13             READ(1,*)
       IF(IMX(1).EQ.0) GOTO 17
       DO 18 I=1,5
   18  READ(1,*)
   17  DO 12 I=NLIN-NL1+1,NLIN
   12  READ(1,'(6I5)')
J,LL(I,1),LL(I,2),LL(I,3),LL(I,4),LL(I,5)
        GOTO 15
   16   NIE=12
      TEXT =  'Zapis LIN Elemenata'
C     CALL ZAPIS(IRID,IKOL,TEXT,NN5)
      N1=1
      N2=-3
      EE=20000.
      ENI=0.3
        GAMARO=7.8E-5
        ALFA=12.E-6
        WRITE(1,19) NIE,NLIN,N1
        WRITE(1,21) N1,EE,ENI,GAMARO,ALFA
        DO 20 I=1,5
   20   WRITE(1,19)
        DO 14 I=1,NLIN
   14   WRITE(1,19)
I,LL(I,1),LL(I,2),N2,N1,LL(I,3)
   19   FORMAT(10I5)
   21   FORMAT(I5,2F10.1,2(1PG10.3))
   15   CLOSE(1)
        END

      SUBROUTINE
ZICPOV(LY,NPOV,LP,N,IUP,IE,LEN)

c       SUBRUTINA ZA CITANJE (N=1) I ZAPIS
(N=2) DAT POVRSINA
c       POZIVA SUBRUTINE :  .DAT  .ZAPIS

      INCLUDE 'ZAJZON1M.FOR'
      INCLUDE 'ZAJZON2M.FOR'
      CHARACTER*15 U1
      CHARACTER*43 TEXT
      INTEGER*2 LP(LY,LEN)

      U1=UDAT
      INP1=INP
      IF(N.EQ.2.AND.IE.EQ.3.AND.NPOV.LE.1)
RETURN
  102 IF(N.EQ.1) CALL DAT (1,12,2,INS,U1)
      IF(N.GT.1) CALL DAT (1,12,0,INS,U1)
      IF(INS.EQ.1) GOTO 101
```

```
      IF(N.EQ.1) GOTO 11
      IF(N.GT.1) GOTO 16
  101 INP1=5
      IF(N.EQ.1) TEXT = 'Ime Dat Pov -
Citanje'
      IF(N.GT.1) TEXT = 'Ime Dat Pov -
Zapis'
      CALL ZAPIS(IRID,IKOL,TEXT,NN5)
      TEXT=' '
C      CALL ZAPIS(23,3,TEXT,NN6)
   99 IF(INP1.EQ.5) CALL READR(0)
      IF(INP1.EQ.5) READ (terk,'(A15
)') U1
      IF(INP1.NE.5) READ (INP1,'(A15
)') U1
      WRITE(   8,'(A15,'' ; Ime Dat'')') U1
      IF( U1.EQ.'                      ')
RETURN
      GOTO 102
   11 TEXT =  'Citanje POV Elemenata'
C      CALL ZAPIS(22,3,TEXT,NN5)
       READ(1,'(3I5,15X,10I5)')
NIE,NP1,(IMX(I),I=1,11)
       DO 122 I=1,IMX(1)
       DO 122 J=1,3
  122  READ(1,*)
       DO 125 I=1,IMX(2)
  125  READ(1,'(I5,F10.3  )') II,DEB1
       DO 126 I=3,10
       DO 126 J=1,IMX(I)
  126  READ(1,*)
       IF(IMX(1).EQ.0) GOTO 121
  123  DO 124 I=1,5
  124  READ(1,*)
  121 IF(NPOV.EQ.1.AND.IUP.EQ.1) NPOV=0
      IF(IUP.EQ.1) NPOV=NPOV+NP1
       DO 12 I=NPOV-NP1+1,NPOV
   12  READ(1,'(5I5)') L,(LP(I,J),J=1,4)
        IF(LP(I,4).EQ.0) LP(I,4)=LP(I,3)
              GOTO 15
   16  NIE=21
       IF(N.EQ.3) GOTO 19
       TEXT = 'Debljina , Pritisak ,
Temperatura Primitiva'
      CALL ZAPIS(IRID,IKOL,TEXT,NN5)
      TEXT=' '
      GOTO 17
   13 CALL GRESKA(1)
C   17 CALL ZAPIS(IRID,IKOL,TEXT,NN5)
   17 IF(INP1.EQ.5) CALL READR(1)
        IF(INP1.EQ.5) READ
(terk,'(3F10.0)') DEB1,PRIT1,TEMP1
        IF(INP1.NE.5) READ
(INP1,'(3F10.0)') DEB1,PRIT1,TEMP1
      WRITE(  8,'(3F10.3,'' ; Debljina
Primitiva'')') DEB1,PRIT1,TEMP1
      TEXT =  'Zapis POV Elemenata'
C      CALL ZAPIS(22,3,TEXT,NN5)
   19 I1=1
      I2=0
        I3=0
        IPOV1=1
C       CLOSE(1)
C       OPEN(1,FILE='M.P',STATUS='UNKNOWN')
        IF(N.EQ.3) IPOV1=NPOV
      IF(PRIT1.NE.0.0) I2=1
      IF(TEMP1.NE.0.0) I3=1
      EE=20000.
      ENI=0.3
        GAMARO=7.8E-5
        ALFA=12.E-6
      IF(N.EQ.3) NIE=20
       WRITE(1,'(3I5,15X,3I5)')
NIE,NPOV,I1,IPOV1,I2,I3
       WRITE(1,'(2I5,1PG10.3,10X,1PG10.3)')
I1,I1,GAMARO,ALFA
       WRITE(1,'(2F10.1)') EE,ENI
       WRITE(1,'('' ''     )')
      IF(N.EQ.2) THEN
         WRITE(1,'(I5,F10.3  )') I1,DEB1
       IF(PRIT1.NE.0.0) WRITE(1,'(I5,F10.3
)') I1,PRIT1
       IF(TEMP1.NE.0.0) WRITE(1,'(I5,F10.3
)') I1,TEMP1
      ELSE
        DO 20 I=1,NPOV
   20  WRITE(1,'(I5,F10.3  )') I,EPRES(1,I)
      END IF
       DO 18 I=1,5
   18  WRITE(1,'('' ''     )')
        DO 14 I=1,NPOV
      IF(N.EQ.3) WRITE(1,'(5I5,5X,43I5)')
I,(LP(I,J),J=1,4),I1,I
   14 IF(N.EQ.2) WRITE(1,'(5I5,5X,43I5)')
I,(LP(I,J),J=1,4),I1,I1,I2,I3
   15   CLOSE(1)
        END
```

**SUBROUTINE**
**TACKEM**(NLT,X,Y,Z,L,L1,N1,AX,AY,AZ,XC,YC,ZC)

```
      INCLUDE 'ZAJZON1M.FOR'
      DIMENSION X(NLT),Y(NLT),Z(NLT)

        IF(ZC.EQ.-1.) THEN
          DO 10 I=1,N1-1
          DO 10 J=1,L1
          X(I*L1+J)=X(J)+AX
          Y(I*L1+J)=Y(J)+AY
          Z(I*L1+J)=Z(J)+AZ*I
   10 CONTINUE
      L=L+L1*(N1-1)
      END IF

        IF(ZC.EQ.0.0) THEN
          DO 11 I=1,N1-1
          DO 11 J=1,L1
          X(L+I*L1+J)=X(L+(I-1)*L1+J)+AX
          Y(L+I*L1+J)=Y(L+(I-1)*L1+J)+AY
          Z(L+I*L1+J)=Z(L+(I-1)*L1+J)+AZ
   11 CONTINUE
      END IF

      IF(ZC.EQ.1) THEN
        FR=AX*PISTRAD
         MR=3
      DO 442 I=1,3
      DO 442 J=1,3
  442 AR(I,J)=0.0
      AR(MR,MR)=1.0
      M1=MOD(MR,3)+1
      M2=MOD(M1,3)+1
      C=COS(FR)
      S=SIN(FR)
      AR(M1,M1)=C
      AR(M2,M2)=C
      AR(M1,M2)=-S
      AR(M2,M1)=S
        ZZZ=0
           DO 12 I=1,N1-1
           DO 12 J=1,L1
      XXX=X(L+(I-1)*L1+J)-XC
      YYY=Y(L+(I-1)*L1+J)-YC
      X(L+I*L1+J)=AR(1,1)*XXX + AR(1,2)*YYY
+ AR(1,3)*ZZZ + XC
      Y(L+I*L1+J)=AR(2,1)*XXX + AR(2,2)*YYY
+ AR(2,3)*ZZZ + YC
   12 Z(L+I*L1+J)=Z(L+(I-1)*L1+J)+AZ
      END IF
```

```
      L=L+L1*N1
          RETURN
          END

        SUBROUTINE
PLOCEM(NLT,LEN,LL,LP,L,N1,NLIN,NPOV,NLI1,NP
O1)
       INCLUDE 'ZAJZON1M.FOR'
        INTEGER*2 LL(NLT,LEN),LP(NLT,LEN)

          DO 10 I=1,N1-1

          DO 11 J=1,NLI1
        NLIN=NLIN+1
          DO 11 K=1,2
   11 LL(NLIN,K)=LL(NLIN-NLI1,K)+L

          DO 12 J=1,NPO1
        NPOV=NPOV+1
          DO 13 K=1,4
   13 LP(NPOV,K)=LP(NPOV-NPO1,K)+L
   12 CONTINUE

   10 CONTINUE
          RETURN
          END

        SUBROUTINE
IZBTAC(NLT,X,Y,Z,L,A1,IDT,N)
      INCLUDE 'ZAJZON1M.FOR'
      DIMENSION X(NLT),Y(NLT),Z(NLT)
        INTEGER*2 IDT(NLT)
          DO 9 I=1,L
        IDT(I)=I
    9 CONTINUE

        L1=0
      K=0
          DO 10 I=1,L
        XZ=X(I)
        IF(N.EQ.3.OR.N.EQ.4) XZ=Z(I)
      IF(N.EQ.1.OR.N.EQ.3) THEN

      IF(XZ.LT.A1) THEN
        L1=L1+1
        IDT(I)=0
      ELSE
      K=K+1
        IDT(I)=K
        END IF

        ELSE

      IF(XZ.GT.A1) THEN
        L1=L1+1
        IDT(I)=0
      ELSE
      K=K+1
        IDT(I)=K
        END IF

      END IF
   10 CONTINUE

      DO 12 I=1,L
      IF(IDT(I).EQ.0) GOTO 12
        X(IDT(I))=X(I)
        Y(IDT(I))=Y(I)
        Z(IDT(I))=Z(I)
   12 CONTINUE
        L=L-L1
          RETURN
    END
```

```
        SUBROUTINE
IZBELE(NLT,LEN,LL,LP,NLIN,NPOV,IDT)
      INCLUDE 'ZAJZON1M.FOR'
        INTEGER*2
LL(NLT,LEN),LP(NLT,LEN),IDT(NLT)

        DO 1 I=1,NLIN
        DO 1 J=1,2
        IF(IDT(LL(I,J)).EQ.0) LL(I,J)=0
        IF(IDT(LL(I,J)).NE.0)
LL(I,J)=IDT(LL(I,J))
    1 CONTINUE
      L1=0
        DO 2 I=1,NLIN
      DO 4 K=1,2
    4   IF(LL(I,K).EQ.0) GOTO 5
      GOTO 2
    5 CONTINUE
        L1=L1+1
        DO 3 J=I,NLIN-L1
        DO 3 K=1,2
        LL(J,K)=LL(J+1,K)
    3 CONTINUE
    2 CONTINUE
      NLIN=NLIN-L1

         DO 11 I=1,NPOV
        DO 11 J=1,4
        IF(IDT(LP(I,J)).EQ.0) LP(I,J)=0
        IF(IDT(LP(I,J)).NE.0)
LP(I,J)=IDT(LP(I,J))
   11 CONTINUE
      L1=0
        DO 12 I=1,NPOV
      DO 14 K=1,4
   14   IF(LP(I,K).EQ.0) GOTO 15
      GOTO 12
   15 CONTINUE
        L1=L1+1
        DO 13 J=I,NPOV-L1
        DO 13 K=1,4
        LP(J,K)=LP(J+1,K)
   13 CONTINUE
   12 CONTINUE
      NPOV=NPOV-L1

        RETURN
        END

        SUBROUTINE ALFAXYZ(AX,AY,ALFAXY)
      INCLUDE 'ZAJZON1M.FOR'
        IF(AX.GT.0.0.AND.AY.EQ.0.0)
ALFAXY=0.0
        IF(AX.EQ.0.0.AND.AY.GT.0.0)
ALFAXY=PI/2.0
        IF(AX.LT.0.0.AND.AY.EQ.0.0)
ALFAXY=PI
        IF(AX.EQ.0.0.AND.AY.LT.0.0)
ALFAXY=3.0*PI/2.0
        IF(AX.NE.0.0.AND.AY.NE.0.0)
ALFAXY=ATAN(AY/AX)
        RETURN
        END
```

```
SUBROUTINE TALAS1
(NLT,LEN,X,Y,Z,LG,LL,LP,L,NLIN,NPOV,IDT)
      INCLUDE 'ZAJZON1M.FOR'
      CHARACTER*70  TEXT
      DIMENSION X(NLT),Y(NLT),Z(NLT)
INTEGER*2 LG(NLT,LEN),LL(NLT,LEN),
LP(NLT,LEN),IDT(NLT)
       GOTO 9
   88 CALL GRESKA(1)
      RETURN
    9 continue
      L=0
       NLIN=0
       NPOV=0
      AAX=0.
      TEXT = '+-AZ , NZ'
      CALL ZAPIS(IRID,IKOL,TEXT,NN5)
       IF(INP.EQ.5) CALL READR(1)
IF(INP.EQ.5) READ(terk,'(2F10.0)') AZ,ANZ
       IF(INP.EQ.5) WRITE(  8,'(2F10.3,'';
Talas Podela'')') AZ,ANZ
IF(INP.NE.5) READ( INP,'(2F10.0)') AZ,ANZ
       NZ=ANZ
   10 TEXT = '+-X , +-Y , +-X1 , +-Y1 , NX
, Krug(0/1) , Xc , Yc'
      CALL ZAPIS(IRID,IKOL,TEXT,NN5)
      TEXT = 'Linije , N1 , N2'
      CALL ZAPIS(IRID,IKOL,TEXT,NN5)
      IF(INP.EQ.5) THEN
      CALL READR(1)
      READ(terk,'(8F10.0)',ERR=88)
AX,AY,AX1,AY1,ANX,ZC,XC,YC
      WRITE(  8,'(8F10.3       )')
AX,AY,AX1,AY1,ANX,ZC,XC,YC
      IF(AX.EQ.0.0.AND.AY.EQ.0.0)   GOTO 11
      CALL READR(1)
      READ(terk,'(3F10.0)',
ERR=88) B,B1,B2
      WRITE(  8,'(3F10.3,''; Talas
Podela'')') B,B1,B2
       ELSE
      READ( INP,'(8F10.0)',ERR=88)
AX,AY,AX1,AY1,ANX,ZC,XC,YC
      IF(AX.EQ.0.0.AND.AY.EQ.0.0)   GOTO 11
     READ( INP,'(3F10.0)',ERR=88)   B,B1,B2
       END IF
       IF(AAX.EQ.0.0) AAX=AX
         IF(AX1.EQ.0.0) AX1=AX/2.
         IF(AY1.EQ.0.0) AY1=AX/4.
         IF(ANX.EQ.0  ) ANX=1
      IF(ZC.EQ.1.0) AY=0.0
         NX=ANX
         N0=B
         N1=B1
         N2=B2
         Z(L+3)=Z(L+1)
         Z(L+2)=Z(L+1)
         IF(ZC.EQ.0.0) THEN
       AXY=SQRT(AX*AX+AY*AY)
      CALL ALFAXYZ(AX,AY,ALFAXY)
  X(L+2)=X(L+1)+AX1/AX*AXY*COS(ALFAXY)-
AY1*SIN(ALFAXY)
Y(L+2)=Y(L+1)+AX1/AX*AXY*SIN(ALFAXY)+AY1*CO
S(ALFAXY)
         X(L+3)=X(L+1)+AX
         Y(L+3)=Y(L+1)+AY
       END IF
         IF(ZC.EQ.1.0) THEN
CALL
UGAO(XC,X(L+1),YC,Y(L+1),R0,F0,PISTRAD)
X(L+2)=(R0+AY1)*COS(F0+AX1*PISTRAD)+XC
Y(L+2)=(R0+AY1)*SIN(F0+AX1*PISTRAD)+YC
```

```
     END IF
     DO 1 I=1,3
         X(L+I+3)=X(L+I)
         Y(L+I+3)=Y(L+I)
    1    Z(L+I+3)=Z(L+I)+AZ
         L1=6
       NLI1=NLIN
       NPO1=NPOV
IF(N0.EQ.2.OR.N0.EQ.3.OR.N0.EQ.4.OR.N0.EQ.5
) THEN
      DO 2 I=1,2
       NLIN=NLIN+1
       LL(NLIN,1)=1+(I-1)*2+L
    2 LL(NLIN,2)=4+(I-1)*2+L
      IF(N0.EQ.4.OR.N0.EQ.5) THEN
       NLIN=NLIN+1
       LL(NLIN,1)=2+L
       LL(NLIN,2)=5+L
      NLIN=3
       END IF
       END IF
     IF(N0.EQ.1.OR.N0.EQ.3.OR.N0.EQ.5) THEN
       DO 3 I=1,2
       DO 3 J=1,2
       NLIN=NLIN+1
       LL(NLIN,1)=(I-1)*3+J  +L
    3 LL(NLIN,2)=(I-1)*3+J+1+L
       END IF
      DO 4 I=1,2
       NPOV=NPOV+1
       LP(NPOV,1)=I  +L
       LP(NPOV,2)=I+1+L
       LP(NPOV,3)=I+4+L
    4  LP(NPOV,4)=I+3+L
       NLI1=NLIN-NLI1
       NPO1=NPOV-NPO1
      IF(NX.GT.1) THEN
CALL TACKEM
(NLT,X,Y,Z,L,L1,NX,AX,AY,0.0,XC,YC,ZC)
CALL PLOCEM
(NLT,LEN,LL,LP,L1,NX,NLIN,NPOV,NLI1,NPO1)
      ELSE
       L=6
       END IF
       X(L+1)=X(L-L1+3)
       Y(L+1)=Y(L-L1+3)
       Z(L+1)=Z(L-L1+3)
       GOTO 10
   11 CONTINUE
       IF(NZ.GT.1) THEN
       NLI1=NLIN
       NPO1=NPOV
       L2=L
CALL TACKEM
(NLT,X,Y,Z,L,L2,NZ,0.0,0.0,AZ,XC,YC,-1.)
CALL PLOCEM
(NLT,LEN,LL,LP,L2,NZ,NLIN,NPOV,NLI1,NPO1)
       END IF
CALL ISTTAC
(L,NLT,X,Y,Z,LG,LL,LP,NLIN,NPOV,LEN,AAX)
      IF(N1.EQ.1) THEN
       CALL IZBTAC(NLT,X,Y,Z,L,AX1,IDT,1)
       CALL
IZBELE(NLT,LEN,LL,LP,NLIN,NPOV,IDT)
       END IF
       IF(N2.EQ.1) THEN
CALL IZBTAC(NLT,X,Y,Z,L,NX*AX-AX+AX1,IDT,2)
CALL IZBELE(NLT,LEN,LL,LP,NLIN,NPOV,IDT)
       END IF
       RETURN
       END
```

```
X(L+3)= R0      *COS(F0+AX *PISTRAD)+XC
Y(L+3)= R0      *SIN(F0+AX *PISTRAD)+YC
       SUBROUTINE TALAS2
(NLT,LEN,X,Y,Z,LG,LL,LP,L,NLIN,NPOV,IDT)
       INCLUDE 'ZAJZON1M.FOR'
      CHARACTER*70  TEXT
      DIMENSION X(NLT),Y(NLT),Z(NLT)
       INTEGER*2 LG(NLT,LEN),LL(NLT,LEN),
LP(NLT,LEN),IDT(NLT)
       GOTO 9
   88 CALL GRESKA(1)
      RETURN
    9 continue
      L=0
       NLIN=0
       NPOV=0
      AAX=0.0
      TEXT = '+-AZ , NZ'
      CALL ZAPIS(IRID,IKOL,TEXT,NN5)
       IF(INP.EQ.5) CALL READR(1)
       IF(INP.EQ.5)
READ(terk,'(2F10.0)',ERR=88) AZ,ANZ
       IF(INP.EQ.5) WRITE(  8,'(2F10.3,'';
Talas Podela'')') AZ,ANZ
       IF(INP.NE.5) READ(
INP,'(2F10.0)',ERR=88) AZ,ANZ
       NZ=ANZ
   10 TEXT = '+-X , +-Y , +-Y1 , NX
,X1,X2,X3,X4, Krug (0/1),XC,YC'
      CALL ZAPIS(IRID,IKOL,TEXT,NN5)
      TEXT = 'Linije'
      CALL ZAPIS(IRID,IKOL,TEXT,NN5)
      IF(INP.EQ.5) THEN
      CALL READR(1)
      READ(terk,'(9F10.0)',ERR=88)
AX,AY,A1,ANX,ANZ,A1,A2,A3,A4
      IF(AX.EQ.0.0.AND.AY.EQ.0.0)   GOTO 11
      CALL READR(1)
      READ(terk,'( F10.0)',ERR=88)   B
       ELSE
      READ( INP,'(9F10.0)',ERR=88)
AX,AY,AZ,ANX,ANZ,A1,A2,A3,A4
      IF(AX.EQ.0.0.AND.AY.EQ.0.0)   GOTO 11
      READ( INP,'( F10.0)',ERR=88)   B
       END IF
      WRITE(   8,'(9F10.3)')
AX,AY,AZ,ANX,ANZ,A1,A2,A3,A4
 WRITE(8,'( F10.3,''; Talas Podela'')') B
         NX=ANX
         NZ=ANZ
         N0=B
       IF(AAX.EQ.0.0) AAX=AX
         IF(AZ.EQ.0.0) AZ=AX
         IF(AY.EQ.0.0) AY=AX/4.
         IF(A1.EQ.0.0) A1=AX/4.
         IF(A2.EQ.0.0) A2=AX/8
         IF(A3.EQ.0.0) A3=A1
         IF(A4.EQ.0.0) A4=A2
         IF(NX.EQ.0  ) NX=1
         IF(NZ.EQ.0  ) NZ=1
      IF(ZC.EQ.1.0) AY=0.0
         X(2)=X(1)+A1
         Y(2)=Y(1)
         Z(2)=Z(1)
         X(3)=X(1)+A2
         Y(3)=Y(1)+AY
         Z(3)=Z(1)
         X(4)=X(1)+AX-A4
         Y(4)=Y(1)+AY
         Z(4)=Z(1)
         X(5)=X(1)+AX-A3
         Y(5)=Y(1)
         Z(5)=Z(1)
         X(6)=X(1)+AX
___________________________________________
IF(ZC.NE.0.) THEN
CALL UGAO
(XC,X(L+1),YC,Y(L+1),R0,F0,PISTRAD)
       X(2)=(R0+AY)*COS(F0+AX1*PISTRAD)+XC
       Y(2)=(R0+AY)*SIN(F0+AX1*PISTRAD)+YC
       X(3)= R0     *COS(F0+AX *PISTRAD)+XC
       Y(3)= R0     *SIN(F0+AX *PISTRAD)+YC
       END IF
      DO I=1,6
       X(I+6)=X(I)
       Y(I+6)=Y(I)
       Z(I+6)=AZ
       END DO
      L1=12
       NLI1=NLIN
       NPO1=NPOV
IF(N0.EQ.2.OR.N0.EQ.3.OR.N0.EQ.6.OR.N0.EQ.7
) THEN
       do I=1,5
               NLIN=NLIN+1
      LL(NLIN,1)=I+L
      LL(NLIN,2)=I+1+L
      END DO
       do I=7,11
               NLIN=NLIN+1
      LL(NLIN,1)=I+L
      LL(NLIN,2)=I+1+L
      END DO
      END IF
IF(N0.EQ.1.OR.N0.EQ.3.OR.N0.EQ.5.OR.N0.EQ.7
)THEN
       NLIN=NLIN+1
      LL(NLIN,1)=1+L
      LL(NLIN,2)=7+L
       NLIN=NLIN+1+L
      LL(NLIN,1)=6+L
      LL(NLIN,2)=12+L
      END IF
       IF(N0.EQ.4.OR.N0.EQ.5.OR.N0.EQ.6.OR
.N0.EQ.7) THEN
      DO I=2,5
       NLIN=NLIN+1
      LL(NLIN,1)=I+L
      LL(NLIN,2)=I+6+L
       END DO
      END IF
      DO 6 I=1,5
         LP(I,1)=I+L
       LP(I,2)=I+1+L
       LP(I,3)=I+7+L
    11  LP(I,4)=I+6
       NPOV=NPOV+5
       NLI1=NLIN-NLI1
       NPO1=NPOV-NPO1
      IF(NX.GT.1) THEN
CALL TACKEM
(NLT,X,Y,Z,L,L1,NX,AX,AY,0.0,XC,YC,ZC)
CALL PLOCEM
(NLT,LEN,LL,LP,L1,NX,NLIN,NPOV,NLI1,NPO1)
      ELSE
       L=12
       END IF
       X(L+1)=X(L-L1+3)
       Y(L+1)=Y(L-L1+3)
       Z(L+1)=Z(L-L1+3)
       GOTO 10
   11 CONTINUE
      IF(NZ.GT.1) THEN
       NLI1=NLIN
       NPO1=NPOV
       L2=L
```

```
         Y(6)=Y(1)
         Z(6)=Z(1)
(NLT,LEN,LL,LP,L2,NZ,NLIN,NPOV,NLI1,NPO1)
      END IF
CALL ISTTAC
(L,NLT,X,Y,Z,LG,LL,LP,NLIN,NPOV,LEN,AAX)
      IF(N1.EQ.1) THEN
        CALL IZBTAC(NLT,X,Y,Z,L,A1,IDT,1)
        CALL IZBELE
(NLT,LEN,LL,LP,NLIN,NPOV,IDT)
CALL TACKEM
(NLT,X,Y,Z,L,L2,NZ,0.0,0.0,AZ,XC,YC,-1.)
CALL PLOCEM
END IF
        IF(N2.EQ.1) THEN
CALL IZBTAC(NLT,X,Y,Z,L,NX*AX-AX+A1,IDT,2)
        CALL
IZBELE(NLT,LEN,LL,LP,NLIN,NPOV,IDT)
        END IF
        RETURN
END
```

---

```
SUBROUTINE TALAS3
(NLT,LEN,X,Y,Z,LG,LL,LP,L,NLIN,NPOV,IDT)
        INCLUDE 'ZAJZON1M.FOR'
      CHARACTER*70  TEXT
      DIMENSION X(NLT),Y(NLT),Z(NLT)
INTEGER*2 LG(NLT,LEN),LL(NLT,LEN),
LP(NLT,LEN),IDT(NLT)
        GOTO 9
   88 CALL GRESKA(1)
      RETURN
    9 continue
      L=0
        NLIN=0
        NPOV=0
      AAX=0.0
      TEXT = '+-AZ , NZ'
      CALL ZAPIS(IRID,IKOL,TEXT,NN5)
       IF(INP.EQ.5) CALL READR(1)
       IF(INP.EQ.5)
READ(terk,'(2F10.0)',ERR=88) AZ,ANZ
       IF(INP.EQ.5) WRITE(  8,'(2F10.3,'';
Talas Podela'')') AZ,ANZ
IF(INP.NE.5) READ(INP,'(2F10.0)')AZ,ANZ
        NZ=ANZ
   10 TEXT = '+-X , +-Y , +-X1 , +-Y1 , NX
, Krug(0/1) , Xc , Yc'
      CALL ZAPIS(IRID,IKOL,TEXT,NN5)
      TEXT = 'Linije'
      CALL ZAPIS(IRID,IKOL,TEXT,NN5)
      IF(INP.EQ.5) THEN
      CALL READR(1)
      READ(terk,'(8F10.0)',ERR=88)
AX,AY,AX1,AY1,ANX,ZC,XC,YC
      WRITE(  8,'(8F10.3        )')
AX,AY,AX1,AY1,ANX,ZC,XC,YC
      IF(AX.EQ.0.0.AND.AY.EQ.0.0)   GOTO 11
      CALL READR(1)
      READ(terk,'( F10.0)', ERR=88) B
WRITE(  8,'( F10.3,''; Talas Podela'')') B
        ELSE
      READ( INP,'(8F10.0)',ERR=88)
AX,AY,AX1,AY1,ANX,ZC,XC,YC
      IF(AX.EQ.0.0.AND.AY.EQ.0.0)   GOTO 11
      READ( INP,'( F10.0)',ERR=88)    B
        END IF
      IF(AAX.EQ.0.0) AAX=AX
         IF(AX1.EQ.0.0) AX1=AX/2.
         IF(AY1.EQ.0.0) AY1=AX/4.
         IF(ANX.EQ.0) ANX=1
      IF(ZC.EQ.1.0) AY=0.0
         NX=ANX
         N0=B
        AXY=SQRT(AX*AX+AY*AY)
         Z(L+2)=Z(L+1)
         Z(L+3)=Z(L+1)
         IF(ZC.EQ.0.0) THEN
         X(L+2)=X(L+1)+AX1-ABS(AY/AXY)*AY1
      END IF
         IF(ZC.EQ.1.0) THEN
CALLUGAO(XC,X(L+1),YC,Y(L+1),R0,F0,PISTRAD)
 X(L+2)=(R0+AY1)*COS(F0+AX1*PISTRAD)+XC
 Y(L+2)=(R0+AY1)*SIN(F0+AX1*PISTRAD)+YC
X(L+3)= R0      *COS(F0+AX *PISTRAD)+XC
Y(L+3)= R0      *SIN(F0+AX *PISTRAD)+YC
        END IF
      DO 1 I=1,3
         X(L+I+3)=X(L+I)
         Y(L+I+3)=Y(L+I)
    1    Z(L+I+3)=Z(L+I)+AZ
         L1=6
        NLI1=NLIN
        NPO1=NPOV
        IF(N0.EQ.1.OR.N0.EQ.3.OR.N0.EQ.5.OR
.N0.EQ.7) THEN
         NLIN=NLIN+1
       LL(NLIN,1)=1+L
       LL(NLIN,2)=4+L
         NLIN=NLIN+1
       LL(NLIN,1)=3+L
       LL(NLIN,2)=6+L
      END IF
IF(N0.EQ.2.OR.N0.EQ.3.OR.N0.EQ.6.OR.N0.EQ.7
)THEN
         NLIN=NLIN+1
       LL(NLIN,1)=1+L
       LL(NLIN,2)=2+L
         NLIN=NLIN+1
       LL(NLIN,1)=2+L
       LL(NLIN,2)=3+L
          NLIN=NLIN+1
       LL(NLIN,1)=4+L
       LL(NLIN,2)=5+L
         NLIN=NLIN+1
       LL(NLIN,1)=5+L
       LL(NLIN,2)=6+L
      END IF
        IF(N0.EQ.4.OR.N0.EQ.5.OR.N0.EQ.6.OR
.N0.EQ.7) THEN
         NLIN=NLIN+1
       LL(NLIN,1)=5+L
       LL(NLIN,2)=1+L
         NLIN=NLIN+1
       LL(NLIN,1)=1+L
       LL(NLIN,2)=6+L
            NLIN=NLIN+1
       LL(NLIN,1)=6+L
       LL(NLIN,2)=2+L
      END IF
      DO 6 I=1,2
        NPOV=NPOV+1
        LP(NPOV,1)=1+L
        LP(NPOV,2)=7-I+L
        LP(NPOV,3)=6-I+L
        LP(NPOV,4)=LP(NPOV,3)
        NPOV=NPOV+1
        LP(NPOV,1)=6+L
```

```
Y(L+2)=Y(L+1)+AX1*ABS(AY/AX)+ABS(AX/AXY)*
AY1
          X(L+3)=X(L+1)+AX
          Y(L+3)=Y(L+1)+AY
NPO1=NPOV-NPO1
      IF(NX.GT.1) THEN
CALL TACKEM
(NLT,X,Y,Z,L,L1,NX,AX,AY,0.0,XC,YC,ZC)
CALL PLOCEM
(NLT,LEN,LL,LP,L1,NX,NLIN,NPOV,NLI1,NPO1)
      ELSE
        L=6
        END IF
        X(L+1)=X(L-L1+3)
        Y(L+1)=Y(L-L1+3)
        Z(L+1)=Z(L-L1+3)
        GOTO 10
```

```
        LP(NPOV,2)=I+L
        LP(NPOV,3)=I+1+L
    6 LP(NPOV,4)=LP(NPOV,3)
        NLI1=NLIN-NLI1
   11 CONTINUE
      IF(NZ.GT.1) THEN
        NLI1=NLIN
        NPO1=NPOV
        L2=L
CALL TACKEM
(NLT,X,Y,Z,L,L2,NZ,0.0,0.0,AZ,XC,YC,-1.)
CALL PLOCEM
(NLT,LEN,LL,LP,L2,NZ,NLIN,NPOV,NLI1,NPO1)
        END IF
CALL ISTTAC
(L,NLT,X,Y,Z,LG,LL,LP,NLIN,NPOV,LEN,AAX)
        RETURN
        END
```

---

```
      SUBROUTINE TALAS4
(NLT,LEN,X,Y,Z,LG,LL,LP,L,NLIN,NPOV,IDT)
        INCLUDE 'ZAJZON1M.FOR'
      CHARACTER*70  TEXT
      DIMENSION X(NLT),Y(NLT),Z(NLT)
        INTEGER*2 LG(NLT,LEN),LL(NLT,LEN),
LP(NLT,LEN),IDT(NLT)
        GOTO 9
   88 CALL GRESKA(1)
      RETURN
    9 continue
      L=0
        NLIN=0
        NPOV=0
      AAX=0.0
      TEXT = '+-AZ , NZ'
      CALL ZAPIS(IRID,IKOL,TEXT,NN5)
       IF(INP.EQ.5) CALL READR(1)
       IF(INP.EQ.5)
READ(terk,'(2F10.0)',ERR=88)    AZ,ANZ
       IF(INP.EQ.5) WRITE(  8,'(2F10.3,'';
Talas Podela'')') AZ,ANZ
  IF(INP.NE.5) READ( INP,'(2F10.0)) AZ,ANZ
        NZ=ANZ
   10 TEXT= '+-X , +-Y , +-X1 , +-Y1 , NX ,
+-Y2 , Krug(0/1) , Xc , Yc'
      CALL ZAPIS(IRID,IKOL,TEXT,NN5)
      TEXT = 'Linije'
      CALL ZAPIS(IRID,IKOL,TEXT,NN5)
    8 IF(INP.EQ.5) THEN
        CALL READR(1)
      READ(terk,'(10F10.0)',ERR=88)
AX,AY,AX1,AY1,ANX,AY2,AZ1,ZC,XC,YC
      WRITE(  8,'(10F10.3       )')
AX,AY,AX1,AY1,ANX,AY2,AZ1,ZC,XC,YC
       IF(AX.EQ.0.0.AND.AY.EQ.0.0)  GOTO 11
      CALL READR(1)
     READ(terk,'(  F10.0)',ERR=88) B
WRITE(  8,'( F10.3,''; Talas Podela'')') B
        ELSE
      READ( INP,'(10F10.0)',ERR=88)
AX,AY,AX1,AY1,ANX,AY2,AZ1,ZC,XC,YC
       IF(AX.EQ.0.0.AND.AY.EQ.0.0)  GOTO 11
      READ( INP,'(  F10.0)',ERR=88)  B
        END IF
        IF(AAX.EQ.0.0) AAX=AX
          IF(AX1.EQ.0.0) AX1=AX/2.
          IF(AY1.EQ.0.0) AY1=AX/4.
          IF(AY2.EQ.0.0) AY2=AX/4.
          IF(AZ1.EQ.0.0) AZ1=AZ/2.
          IF(ANX.EQ.0  ) ANX=1
      IF(ZC.EQ.1.0) AY=0.0
```

```
  Z(L+8)=Z(L+1)+AZ1
      IF(ZC.EQ.0.0) THEN
X(L+2)=X(L+1)+AX1-ABS(AY/AXY)*AY1
Y(L+2)=Y(L+1)+AX1*ABS(AY/AX)+ABS(AX/AXY)*
AY1
          X(L+3)=X(L+1)+AX
          Y(L+3)=Y(L+1)+AY
          X(L+7)=X(L+1)-ABS(AY/AXY)*AY2
          Y(L+7)=Y(L+1)+ABS(AX/AXY)*AY2
          X(L+8)=X(L+1)+AX-ABS(AY/AXY)*AY2
          Y(L+8)=Y(L+1)+AY+ABS(AX/AXY)*AY2
        END IF
          IF(ZC.EQ.1.0) THEN
CALL UGAO
(XC,X(L+1),YC,Y(L+1),R0,F0,PISTRAD)
X(L+2)=(R0+AY1)*COS(F0+AX1*PISTRAD)+XC
Y(L+2)=(R0+AY1)*SIN(F0+AX1*PISTRAD)+YC
X(L+3)= R0     *COS(F0+AX *PISTRAD)+XC
Y(L+3)= R0     *SIN(F0+AX *PISTRAD)+YC
X(L+ 7)=(R0+AY2)*COS(F0             )+XC
Y(L+ 7)=(R0+AY2)*SIN(F0             )+YC
X(L+8)=(R0+AY2)*COS(F0+AX *PISTRAD)+XC
Y(L+8)=(R0+AY2)*SIN(F0+AX *PISTRAD)+YC
        END IF
      DO 1 I=1,3
          X(L+I+3)=X(L+I)
          Y(L+I+3)=Y(L+I)
    1     Z(L+I+3)=Z(L+I)+AZ
      L1=8
        NLI1=NLIN
        NPO1=NPOV
        IF(N0.EQ.1.OR.N0.EQ.3.OR.N0.EQ.5.OR
.N0.EQ.7) THEN
         NLIN=NLIN+1
        LL(NLIN,1)=1+L
        LL(NLIN,2)=7+L
         NLIN=NLIN+1
        LL(NLIN,1)=7+L
        LL(NLIN,2)=4+L
         NLIN=NLIN+1
        LL(NLIN,1)=3+L
        LL(NLIN,2)=8+L
         NLIN=NLIN+1
        LL(NLIN,1)=8+L
        LL(NLIN,2)=6+L
      END IF
        IF(N0.EQ.2.OR.N0.EQ.3.OR.N0.EQ.6.OR
.N0.EQ.7)THEN
         NLIN=NLIN+1
        LL(NLIN,1)=1+L
        LL(NLIN,2)=2+L
         NLIN=NLIN+1
```

```
          NX=ANX
          N0=B
        AXY=SQRT(AX*AX+AY*AY)
          Z(L+2)=Z(L+1)
          Z(L+3)=Z(L+1)
          Z(L+7)=Z(L+1)+AZ1
               LL(NLIN,1)=5+L
       LL(NLIN,2)=6+L
      END IF
        IF(N0.EQ.4.OR.N0.EQ.5.OR.N0.EQ.6.OR
.N0.EQ.7) THEN
        NLIN=NLIN+1
       LL(NLIN,1)=5+L
       LL(NLIN,2)=1+L
        NLIN=NLIN+1
       LL(NLIN,1)=1+L
       LL(NLIN,2)=6+L
          NLIN=NLIN+1+L
       LL(NLIN,1)=6+L
       LL(NLIN,2)=2+L
          NLIN=NLIN+1
       LL(NLIN,1)=5+L
       LL(NLIN,2)=7+L
          NLIN=NLIN+1
       LL(NLIN,1)=8+L
       LL(NLIN,2)=2+L
      END IF
      NPOV=NPOV+1
        LP(NPOV,1)=7+L
        LP(NPOV,2)=5+L
        LP(NPOV,3)=4+L
      NPOV=NPOV+1
        LP(NPOV,1)=1+L
        LP(NPOV,2)=5+L
        LP(NPOV,3)=7+L
      NPOV=NPOV+1
        LP(NPOV,1)=1+L
        LP(NPOV,2)=6+L
        LP(NPOV,3)=5+L
      NPOV=NPOV+1
        LP(NPOV,1)=1+L
        LP(NPOV,2)=2+L
        LP(NPOV,3)=6+L
       LL(NLIN,1)=2+L
       LL(NLIN,2)=3+L
          NLIN=NLIN+1
       LL(NLIN,1)=4+L
       LL(NLIN,2)=5+L
        NLIN=NLIN+1
      NPOV=NPOV+1
        LP(NPOV,1)=2+L
        LP(NPOV,2)=8+L
        LP(NPOV,3)=6+L
      NPOV=NPOV+1
        LP(NPOV,1)=2+L
        LP(NPOV,2)=3+L
        LP(NPOV,3)=8+L
      DO 2 I=1,6
    2   LP(NPOV-I+1,4)=LP(NPOV-I+1,3)
        NLI1=NLIN-NLI1
        NPO1=NPOV-NPO1
      IF(NX.GT.1) THEN CALL TACKEM
(NLT,X,Y,Z,L,L1,NX,AX,AY,0.0,XC,YC,ZC)
CALL PLOCEM
(NLT,LEN,LL,LP,L1,NX,NLIN,NPOV,NLI1,NPO1)
      ELSE
        L=8
        END IF
        X(L+1)=X(L-L1+3)
        Y(L+1)=Y(L-L1+3)
        Z(L+1)=Z(L-L1+3)
        GOTO 10
   11 CONTINUE
      IF(NZ.GT.1) THEN
        NLI1=NLIN
        NPO1=NPOV
        L2=L
CALL TACKEM
(NLT,X,Y,Z,L,L2,NZ,0.0,0.0,AZ,XC,YC,-1.)
CALL PLOCEM
(NLT,LEN,LL,LP,L2,NZ,NLIN,NPOV,NLI1,NPO1)
        END IF
CALL ISTTAC
(L,NLT,X,Y,Z,LG,LL,LP,NLIN,NPOV,LEN,AAX)
        RETURN
        END
```

---

**SUBROUTINE TALAS5**

```
(NLT,LEN,X,Y,Z,LG,LL,LP,L,NLIN,NPOV,IDT)
        INCLUDE 'ZAJZON1M.FOR'
      CHARACTER*80  TEXT
      DIMENSION X(NLT),Y(NLT),Z(NLT)
        INTEGER*2 LG(NLT,LEN),LL(NLT,LEN),
LP(NLT,LEN),IDT(NLT)
        GOTO 9
   88 CALL GRESKA(1)
      RETURN
    9 continue
      L=0
        NLIN=0
        NPOV=0
      AAX=0.0
      TEXT = '+-AZ , NZ'
      CALL ZAPIS(IRID,IKOL,TEXT,NN5)
       IF(INP.EQ.5) CALL READR(1)
       IF(INP.EQ.5)
READ(terk,'(2F10.0)',ERR=88)       AZ,ANZ
       IF(INP.EQ.5) WRITE(  8,'(2F10.3,'';
Talas Podela'')') AZ,ANZ
       IF(INP.NE.5) READ(
INP,'(2F10.0)',ERR=88)             AZ,ANZ
        NZ=ANZ
   10 TEXT='+-X , +-Y , +-X1 , +-Y1 , NX ,
+-X2 , +-Y2 ,+-X3,+-Y3,+-Z1,+
ANX,AX2,AY2,AX3,AY3,AZ1,AZ2,ZC,XC,YC
WRITE(8,'(14F10.3)')AX,AY,AX1,AY1,ANX,
AX2,AY2,AX3,AY3,AZ1,AZ2,ZC,XC,YC
        IF(AX.EQ.0.0.AND.AY.EQ.0.0)GOTO 11
      CALL READR(1)
      READ(terk,'( F10.0)',  ERR=88) B
WRITE(  8,'( F10.3,''; Talas Podela'')') B
        ELSE
      READ(NP,'(14F10.0)',ERR=88)AX,AY,AX1,
AY1,ANX,AX2,AY2,AX3,AY3,AZ1,AZ2,ZC,XC,YC
        IF(AX.EQ.0.0.AND.AY.EQ.0.0)GOTO 11
      READ( INP,'( F10.0)',ERR=88)   B
        END IF
        IF(AAX.EQ.0.0) AAX=AX
          IF(AX1.EQ.0.0) AX1=AX/2.
          IF(AX2.EQ.0.0) AX2=AX/4.
          IF(AX3.EQ.0.0) AX3=AX2
          IF(AY2.EQ.0.0) AY2=AX/8.
          IF(AZ1.EQ.0.0) AZ1=AZ/2.
          IF(AZ2.EQ.0.0) AZ2=AZ1
          IF(ANX.EQ.0  ) ANX=1
      IF(ZC.EQ.1.0) AY=0.0
          NX=ANX
          N0=B
        AXY=SQRT(AX*AX+AY*AY)
          Z(L+ 2)=Z(L+1)
          Z(L+ 3)=Z(L+1)
```

```
      +-Z2,Krug(0/1),Xc,Yc'
       CALL ZAPIS(IRID,IKOL,TEXT,NN5)
       TEXT = 'Linije'
       CALL ZAPIS(IRID,IKOL,TEXT,NN5)
       IF(INP.EQ.5) THEN
       CALL READR(1)
READ(terk,'(14F10.0)',ERR=88)AX,AY,AX1,AY1,
AY1
          X(L+ 3)=X(L+1)+AX
          Y(L+ 3)=Y(L+1)+AY
         X(L+ 7)=X(L+1)-ABS(AY/AXY)*AY3
          Y(L+ 7)=Y(L+1)+ABS(AX/AXY)*AY3
  X(L+ 8)=X(L+1)+AX2-ABS(AY/AXY)*AY2
Y(L+8)=Y(L+1)+AX2*ABS(AY/AX)+ABS(AX/AXY)*
AY2
X(L+ 9)=X(L+1)+(AX-AX3)-ABS(AY/AXY)*AY2
Y(L+ 9)=Y(L+1)+(AX-
AX3)*ABS(AY/AX)+ABS(AX/AXY)*AY2
X(L+10)=X(L+1)+AX         -ABS(AY/AXY)*AY3
Y(L+10)=Y(L+1)+AY         +ABS(AX/AXY)*AY3
       END IF
        IF(ZC.EQ.1.0) THEN
        CALL
UGAO(XC,X(L+1),YC,Y(L+1),R0,F0,PISTRAD)
X(L+2)=(R0+AY1)*COS(F0+AX1*PISTRAD)+XC
Y(L+2)=(R0+AY1)*SIN(F0+AX1*PISTRAD)+YC

 X(L+3)= R0      *COS(F0+AX *PISTRAD)+XC
 Y(L+3)= R0      *SIN(F0+AX *PISTRAD)+YC
X(L+ 7)=(R0+AY3)*COS(F0              )+XC
Y(L+ 7)=(R0+AY3)*SIN(F0              )+YC
X(L+ 8)=(R0+AY2)*COS(F0+AX2*PISTRAD)+XC
Y(L+ 8)=(R0+AY2)*SIN(F0+AX2*PISTRAD)+YC
X(L+ 9)=(R0+AY2)*COS(F0+(AX-
AX3)*PISTRAD)+XC
Y(L+ 9)=(R0+AY2)*SIN(F0+(AX-
AX3)*PISTRAD)+YC
X(L+10)=(R0+AY3)*COS(F0+AX *PISTRAD)+XC
Y(L+10)=(R0+AY3)*SIN(F0+AX *PISTRAD)+YC
        END IF
       DO 1 I=1,3
          X(L+I+3)=X(L+I)
          Y(L+I+3)=Y(L+I)
    1     Z(L+I+3)=Z(L+I)+AZ
       L1=10
        NLI1=NLIN
        NPO1=NPOV
        IF(N0.EQ.2.OR.N0.EQ.3.OR.N0.EQ.6.OR
.N0.EQ.7.OR.N0.EQ.10.OR.N0.EQ.11) THEN
         NLIN=NLIN+1
        LL(NLIN,1)=1+L
        LL(NLIN,2)=7+L
         NLIN=NLIN+1
        LL(NLIN,1)=3+L
        LL(NLIN,2)=10+L
         NLIN=NLIN+1
        LL(NLIN,1)=7+L
        LL(NLIN,2)=4+L
         NLIN=NLIN+1
        LL(NLIN,1)=10+L
        LL(NLIN,2)=6 +L
       END IF
        IF(N0.EQ.1.OR.N0.EQ.3.OR.N0.EQ.5.OR
.N0.EQ.7.OR.N0.EQ.9.OR.N0.EQ.11)THEN
         NLIN=NLIN+1
        LL(NLIN,1)=1+L
        LL(NLIN,2)=2+L
         NLIN=NLIN+1
        LL(NLIN,1)=2+L
        LL(NLIN,2)=3+L
           NLIN=NLIN+1
        LL(NLIN,1)=4+L
        LL(NLIN,2)=5+L
         NLIN=NLIN+1
```

```
         Z(L+ 7)=Z(L+1)+AZ2
         Z(L+ 8)=Z(L+1)+AZ1
         Z(L+ 9)=Z(L+1)+AZ1
         Z(L+10)=Z(L+1)+AZ2
        IF(ZC.EQ.0.0) THEN
         X(L+ 2)=X(L+1)+AX1-ABS(AY/AXY)*AY1
  Y(L+2)=Y(L+1)+AX1*ABS(AY/AX)+ABS(AX/AXY)*
       LL(NLIN,1)=1+L
       LL(NLIN,2)=8+L
        NLIN=NLIN+1
       LL(NLIN,1)=8+L
       LL(NLIN,2)=9+L
          NLIN=NLIN+1
       LL(NLIN,1)=9+L
       LL(NLIN,2)=6+L
        NLIN=NLIN+1
       LL(NLIN,1)=7+L
       LL(NLIN,2)=8+L
        NLIN=NLIN+1
       LL(NLIN,1)=9 +L
       LL(NLIN,2)=10+L
      END IF
IF(N0.EQ.8.OR.N0.EQ.9.OR.N0.EQ.10.OR.N0.EQ.
11) THEN
        NLIN=NLIN+1
       LL(NLIN,1)=8+L
       LL(NLIN,2)=2+L
        NLIN=NLIN+1
       LL(NLIN,1)=8+L
       LL(NLIN,2)=4+L
          NLIN=NLIN+1
       LL(NLIN,1)=8+L
       LL(NLIN,2)=5+L
        NLIN=NLIN+1
       LL(NLIN,1)=5+L
       LL(NLIN,2)=9+L
          NLIN=NLIN+1
       LL(NLIN,1)=8+L
       LL(NLIN,2)=2+L
        NLIN=NLIN+1
       LL(NLIN,1)=9+L
       LL(NLIN,2)=2+L
          NLIN=NLIN+1
       LL(NLIN,1)=9+L
       LL(NLIN,2)=3+L
        NLIN=NLIN+1
       LL(NLIN,1)=9+L
       LL(NLIN,2)=6+L
      END IF
      NPOV=NPOV+1
        LP(NPOV,1)=1+L
        LP(NPOV,2)=2+L
        LP(NPOV,3)=8+L
      NPOV=NPOV+1
        LP(NPOV,1)=2+L
        LP(NPOV,2)=9+L
        LP(NPOV,3)=8+L
      NPOV=NPOV+1
        LP(NPOV,1)=2+L
        LP(NPOV,2)=3+L
        LP(NPOV,3)=9+L
      NPOV=NPOV+1
        LP(NPOV,1)= 3+L
        LP(NPOV,2)=10+L
        LP(NPOV,3)=9+L
      NPOV=NPOV+1
        LP(NPOV,1)=10+L
        LP(NPOV,2)= 6+L
        LP(NPOV,3)= 9+L
      NPOV=NPOV+1
        LP(NPOV,1)=6+L
        LP(NPOV,2)=5+L
        LP(NPOV,3)=9+L
      NPOV=NPOV+1
```

```
        LL(NLIN,1)=5+L
        LL(NLIN,2)=6+L
       END IF
         IF(N0.EQ.4.OR.N0.EQ.5.OR.N0.EQ.6.OR
.N0.EQ.7.OR.N0.EQ.8.OR.N0.EQ.9.OR.N0.EQ.10.
OR.N0.EQ.11) THEN
         NLIN=NLIN+1

       NPOV=NPOV+1
         LP(NPOV,1)=4+L
         LP(NPOV,2)=7+L
         LP(NPOV,3)=8+L
       NPOV=NPOV+1
         LP(NPOV,1)=7+L
         LP(NPOV,2)=1+L
         LP(NPOV,3)=8+L
         DO 2 I=1,10
     2   LP(NPOV-I+1,4)=LP(NPOV-I+1,3)
         NLI1=NLIN-NLI1
         NPO1=NPOV-NPO1
       IF(NX.GT.1) THEN
CALL TACKEM
(NLT,X,Y,Z,L,L1,NX,AX,AY,0.0,XC,YC,ZC)
CALL PLOCEM
(NLT,LEN,LL,LP,L1,NX,NLIN,NPOV,NLI1,NPO1)
       ELSE
         L=10
         LP(NPOV,1)=9+L
         LP(NPOV,2)=5+L
         LP(NPOV,3)=8+L
       NPOV=NPOV+1
         LP(NPOV,1)=5+L
         LP(NPOV,2)=4+L
         LP(NPOV,3)=8+L
         END IF
         X(L+1)=X(L-L1+3)
         Y(L+1)=Y(L-L1+3)
         Z(L+1)=Z(L-L1+3)
         GOTO 10
    11 CONTINUE
       IF(NZ.GT.1) THEN
         NLI1=NLIN
         NPO1=NPOV
         L2=L
CALL TACKEM
(NLT,X,Y,Z,L,L2,NZ,0.0,0.0,AZ,XC,YC,-1.)
CALL PLOCEM
(NLT,LEN,LL,LP,L2,NZ,NLIN,NPOV,NLI1,NPO1)
         END IF
CALL ISTTAC
(L,NLT,X,Y,Z,LG,LL,LP,NLIN,NPOV,LEN,AAX)
         RETURN
         END
```

**SUBROUTINE TALAS6**

```
(NLT,LEN,X,Y,Z,LG,LL,LP,L,NLIN,NPOV,IDT)
         INCLUDE 'ZAJZON1M.FOR'
       CHARACTER*70  TEXT
       DIMENSION X(NLT),Y(NLT),Z(NLT)
         INTEGER*2 LG(NLT,LEN),LL(NLT,LEN),
LP(NLT,LEN),IDT(NLT)
         GOTO 9
    88 CALL GRESKA(1)
       RETURN
     9 continue
       L=0
         NLIN=0
         NPOV=0
       AAX=0.0
       TEXT = '+-AZ , NZ'
       CALL ZAPIS(IRID,IKOL,TEXT,NN5)
        IF(INP.EQ.5) CALL READR(1)
        IF(INP.EQ.5)
READ(terk,'(2F10.0)',ERR=88)        AZ,ANZ
        IF(INP.EQ.5) WRITE(  8,'(2F10.3,'';
Talas Podela'')') AZ,ANZ
        IF(INP.NE.5) READ(
INP,'(2F10.0)',ERR=88)              AZ,ANZ
         NZ=ANZ
    10 TEXT='+-X , +-Y , +-X1 , +-Y1 , NX ,
+-Z1 , Krug(0/1),Xc,Yc'
       CALL ZAPIS(IRID,IKOL,TEXT,NN5)
       TEXT = 'Linije'
       CALL ZAPIS(IRID,IKOL,TEXT,NN5)
     8 IF(INP.EQ.5) THEN
       CALL READR(1)
       READ(terk,'(9F10.0)',ERR=88)
AX,AY,AX1,AY1,ANX,AZ1,ZC,XC,YC
       WRITE(  8,'(9F10.3       )')
AX,AY,AX1,AY1,ANX,AZ1,ZC,XC,YC
       IF(AX.EQ.0.0.AND.AY.EQ.0.0)   GOTO 11
       CALL READR(1)
       READ(terk,'(2F10.0)' ,  ERR=88) B
 WRITE(  8,'(2F10.3,''; Talas Podela'')') B
         ELSE
       READ( INP,'(9F10.0)',ERR=88)
AX,AY,AX1,AY1,ANX,AZ1,ZC,XC,YC
       IF(AX.EQ.0.0.AND.AY.EQ.0.0)   GOTO 11
       READ( INP,'(2F10.0)',ERR=88)   B
           IF(AX1.EQ.0.0) AX1=AX/2.
           IF(AZ1.EQ.0.0) AZ1=AZ/2.
           IF(ANX.EQ.0  ) ANX=1
       IF(ZC.EQ.1.0) AY=0.0
           NX=ANX
           N0=B
         AXY=SQRT(AX*AX+AY*AY)
           Z(L+2)=Z(L+1)
           Z(L+5)=Z(L+1)+AZ1
         IF(ZC.EQ.0.) THEN
           X(L+2)=X(L+1)+AX
           Y(L+2)=Y(L+1)+AY
  X(L+5)=X(L+1)+AX1-ABS(AY/AXY)*AY1
Y(L+5)=Y(L+1)+AX2*ABS(AY/AX)+ABS(AX/AXY)*
AY1
       END IF
       IF(ZC.EQ.1.0) THEN
         CALL
UGAO(XC,X(L+1),YC,Y(L+1),R0,F0,PISTRAD)
 X(L+2)= R0      *COS(F0+AX *PISTRAD)+XC
 Y(L+2)= R0      *SIN(F0+AX *PISTRAD)+YC
 X(L+5)=(R0+AY1)*COS(F0+AX1*PISTRAD)+XC
 Y(L+5)=(R0+AY1)*SIN(F0+AX1*PISTRAD)+YC
         END IF
        DO 5 I=1,2
           X(L+I+2)=X(L+I)
           Y(L+I+2)=Y(L+I)
     5    Z(L+I+2)=Z(L+I)+AZ
       L1=5
         NLI1=NLIN
         NPO1=NPOV
IF(N0.EQ.1.OR.N0.EQ.3.OR.N0.EQ.5.OR.N0.EQ.7
) THEN
         DO 1 I=1,2
         NLIN=NLIN+1
           LL(NLIN,1)=(I-1)*2+I+L
     1    LL(NLIN,2)=I+1+L
       END IF
IF(N0.EQ.2.OR.N0.EQ.3.OR.N0.EQ.6.OR.N0.EQ.7
) THEN
         DO 2 I=1,2
         NLIN=NLIN+1
           LL(NLIN,1)=I+L
     2    LL(NLIN,2)=I+2+L
       END IF
```

```
        END IF
        IF(AAX.EQ.0.0) AAX=AX
          IF(AY1.EQ.0.0) AY1=AX/2.
        NLIN=NLIN+1
         LL(NLIN,1)=5+L
    3    LL(NLIN,2)=I+L
      END IF
      DO 4 I=1,4
        NPOV=NPOV+1
         LP(NPOV,1)=I+L
         LP(NPOV,2)=2*I+L
      IF(I.EQ.3) LP(NPOV,2)=1+L
      IF(I.EQ.4) LP(NPOV,2)=3+L
         LP(NPOV,3)=5+L
    4    LP(NPOV,4)=5+L
        NLI1=NLIN-NLI1
        NPO1=NPOV-NPO1
      IF(NX.GT.1) THEN
CALL TACKEM
(NLT,X,Y,Z,L,L1,NX,AX,AY,0.0,XC,YC,ZC)
CALL PLOCEM
(NLT,LEN,LL,LP,L1,NX,NLIN,NPOV,NLI1,NPO1)
IF(N0.EQ.4.OR.N0.EQ.5.OR.N0.EQ.6.OR.N0.EQ.7
) THEN
      DO 3 I=1,4
      ELSE
        L=5
      END IF
        X(L+1)=X(L-L1+3)
        Y(L+1)=Y(L-L1+3)
        Z(L+1)=Z(L-L1+3)
        GOTO 10
   11 CONTINUE
      IF(NZ.GT.1) THEN
        NLI1=NLIN
        NPO1=NPOV
        L2=L
CALL TACKEM
(NLT,X,Y,Z,L,L2,NZ,0.0,0.0,AZ,XC,YC,-1.)
CALL PLOCEM
(NLT,LEN,LL,LP,L2,NZ,NLIN,NPOV,NLI1,NPO1)
        END IF
CALL ISTTAC
(L,NLT,X,Y,Z,LG,LL,LP,NLIN,NPOV,LEN,AAX)
        RETURN
        END
```

---

```
      SUBROUTINE TALAS7
(NLT,LEN,X,Y,Z,LG,LL,LP,L,NLIN,NPOV,IDT)
        INCLUDE 'ZAJZON1M.FOR'
      CHARACTER*70  TEXT
      DIMENSION X(NLT),Y(NLT),Z(NLT)
        INTEGER*2 LG(NLT,LEN),LL(NLT,LEN),
LP(NLT,LEN),IDT(NLT)
        GOTO 9
    8 CALL GRESKA(1)
      RETURN
    9 continue
      L=0
        NLIN=0
        NPOV=0
        AAX=0.0
      TEXT = '+-AZ , NZ'
      CALL ZAPIS(IRID,IKOL,TEXT,NN5)
       IF(INP.EQ.5) CALL READR(1)
       IF(INP.EQ.5)
READ(terk,'(2F10.0)',ERR=8)    AZ,ANZ
       IF(INP.EQ.5) WRITE(  8,'(2F10.3,'';
Talas Podela'')') AZ,ANZ
IF(INP.NE.5) READ( INP,'(2F10.0)') AZ,ANZ
        NZ=ANZ
   10 TEXT='+-X , +-Y , +-X1 , +-Y1 , NX ,
+-X2 , +-Y2 , +-Z1 , Krug(0/1),Xc,Yc'
      CALL ZAPIS(IRID,IKOL,TEXT,NN5)
      TEXT = 'Linije , An1 , An2'
      CALL ZAPIS(IRID,IKOL,TEXT,NN5)
      IF(INP.EQ.5) THEN
       CALL READR(1)
READ(terk,'(11F10.0)',ERR=8)AX,AY,AX1,AY1,A
NX,AX2,AY2,AZ1,ZC,XC,YC
      WRITE(  8,'(11F10.3
)')AX,AY,AX1,AY1,ANX,AX2,AY2,AZ1,ZC,XC,YC
       IF(AX.EQ.0.0.AND.AY.EQ.0.0)GOTO 11
       CALL READR(1)
  READ(terk,'(3F10.0)',ERR=8) B,B1,B2
WRITE(  8,'( 3F10.3,''; Talas Podela'')')
B,B1,B2
        ELSE
READ( INP,'(11F10.0)',ERR=8)
AX,AY,AX1,AY1,ANX,AX2,AY2,AZ1,ZC,XC,YC
       IF(AX.EQ.0.0.AND.AY.EQ.0.0)GOTO 11
      READ( INP,'( 3F10.0)',ERR=8)B,B1,B2
          IF(AX2.EQ.0.0) AX2=AX/2.
          IF(AY1.EQ.0.0) AY1=AY/4.
          IF(AY2.EQ.0.0) AY2=AY/8.
          IF(AZ1.EQ.0.0) AZ1=AZ/2.
          IF(NX.EQ.0  ) NX=1
          IF(NZ.EQ.0  ) NZ=1
      IF(ZC.EQ.1.0) AY=0.0
          NX=ANX
          N0=B
          N1=B1
          N2=B2
        AXY=SQRT(AX*AX+AY*AY)
          Z(L+2)=Z(L+1)
          Z(L+3)=Z(L+1)
          Z(L+7)=Z(L+1)+AZ1
         IF(ZC.EQ.0.0) THEN
    X(L+2)=X(L+1)+AX2-ABS(AY/AXY)*AY2
Y(L+2)=Y(L+1)+AX2*ABS(AY/AX)+ABS(AX/AXY)*
AY2
          X(L+3)=X(L+1)+AX
          Y(L+3)=Y(L+1)+AY
          X(L+7)=X(L+1)+AX1-ABS(AY/AXY)*AY1
Y(L+7)=Y(L+1)+AX1*ABS(AY/AX)+ABS(AX/AXY)*
AY1
        END IF
        IF(ZC.EQ.1.0) THEN
CALL UGAO
(XC,X(L+1),YC,Y(L+1),R0,F0,PISTRAD)
X(L+2)=(R0+AY2)*COS(F0+AX2*PISTRAD)+XC
Y(L+2)=(R0+AY2)*SIN(F0+AX2*PISTRAD)+YC
X(L+3)= R0     *COS(F0+AX *PISTRAD)+XC
Y(L+3)= R0     *SIN(F0+AX *PISTRAD)+YC
X(L+7)=(R0+AY1)*COS(F0+AX1*PISTRAD)+XC
Y(L+7)=(R0+AY1)*SIN(F0+AX1*PISTRAD)+YC
        END IF
       DO 7 I=1,3
           X(L+I+3)=X(L+I)
           Y(L+I+3)=Y(L+I)
    7    Z(L+I+3)=Z(L+I)+AZ
      L1=7
        NLI1=NLIN
        NPO1=NPOV
IF(N0.EQ.1.OR.N0.EQ.3.OR.N0.EQ. 4.OR.N0.EQ.
5.OR.N0.EQ.7.OR.N0.EQ.8.OR.N0.EQ.10.OR.N0.E
Q.12    ) THEN
```

```
        END IF
       IF(AAX.EQ.0.0) AAX=AX
          IF(AX1.EQ.0.0) AX1=AX/2.
        NLIN=NLIN+1
       LL(NLIN,1)=3+L
       LL(NLIN,2)=6+L
IF(N0.EQ.4.OR.N0.EQ.5.OR.N0.EQ.9.OR.N0.EQ.1
0.OR.N0.EQ.11.OR.N0.EQ.12) THEN
        DO 4 I=1,2
        NLIN=NLIN+1
        LL(NLIN,1)=7+L
    4  LL(NLIN,2)=3*I-1+L
       END IF
       END IF
IF(N0.EQ.2.OR.N0.EQ.3.OR.N0.EQ.4.OR.N0.EQ.4
.OR.N0.EQ.7.OR.N0.EQ.11.OR.N0.EQ.12) THEN
        DO 5 I=1,2
        K=2
        IF(I.EQ.2) K=5
        M=0
        IF(I.EQ.1) M=-1
        DO 5 J=1,2
        NLIN=NLIN+1
          LL(NLIN,1)=2*I+(J-1)*2+M+L
    5   LL(NLIN,2)=K+L
       END IF
IF(N0.EQ.6.OR.N0.EQ.7.OR.N0.EQ.8.OR.N0.EQ.9
.OR.N0.EQ.10.OR.N0.EQ.11.OR.N0.EQ.12) THEN
        DO 2 I=1,4
        K=1
        IF(I.EQ.1) K=0
        IF(I.EQ.4) K=2
        NLIN=NLIN+1
        LL(NLIN,1)=7+L
    2  LL(NLIN,2)=I+K+L
       END IF
       DO 6 I=1,6
        NPOV=NPOV+1
       I1=I
        IF(I.GE.4) I1=10-I
        I2=I+1
        IF(I.GE.3) I2=10-I
        IF(I.EQ.6) I2=1
        LP(NPOV,2)=I1+L
        NLIN=NLIN+1
       LL(NLIN,1)=1+L
       LL(NLIN,2)=4+L
        LP(NPOV,2)=I2+L
        DO 3 J=3,4
    3   LP(NPOV,J)=7+L
    6  CONTINUE
        NLI1=NLIN-NLI1
        NPO1=NPOV-NPO1
       IF(NX.GT.1) THEN
CALL TACKEM
(NLT,X,Y,Z,L,L1,NX,AX,AY,0.0,XC,YC,ZC)
CALL PLOCEM
(NLT,LEN,LL,LP,L1,NX,NLIN,NPOV,NLI1,NPO1)
       ELSE
        L=7
        END IF
        X(L+1)=X(L-L1+3)
        Y(L+1)=Y(L-L1+3)
        Z(L+1)=Z(L-L1+3)
        GOTO 10
   11  CONTINUE
        IF(NZ.GT.1) THEN
        NLI1=NLIN
        NPO1=NPOV
        L2=L
CALL TACKEM
(NLT,X,Y,Z,L,L2,NZ,0.0,0.0,AZ,XC,YC,-1.)
CALL PLOCEM
(NLT,LEN,LL,LP,L2,NZ,NLIN,NPOV,NLI1,NPO1)
        END IF
CALL ISTTAC
(L,NLT,X,Y,Z,LG,LL,LP,NLIN,NPOV,LEN,AAX)
       IF(N1.EQ.1) THEN
        CALL IZBTAC(NLT,X,Y,Z,L,A1,IDT,1)
CALL IZBELE(NLT,LEN,LL,LP,NLIN,NPOV,IDT)
        END IF
        IF(N2.EQ.1) THEN
CALL IZBTAC(NLT,X,Y,Z,L,NX*AX-AX+A1,IDT,2)
CALL IZBELE(NLT,LEN,LL,LP,NLIN,NPOV,IDT)
        END IF
        RETURN
        END
```

---

```
       SUBROUTINE TALAS8
(NLT,LEN,X,Y,Z,LG,LL,LP,L,NLIN,NPOV,IDT)
        INCLUDE 'ZAJZON1M.FOR'
       CHARACTER*70  TEXT
       DIMENSION X(NLT),Y(NLT),Z(NLT)
        INTEGER*2 LG(NLT,LEN),LL(NLT,LEN),
LP(NLT,LEN),IDT(NLT)
        GOTO 9
   88 CALL GRESKA(1)
       RETURN
    9 continue
       L=0
        NLIN=0
        NPOV=0
       TEXT = '+-AZ , NZ'
       CALL ZAPIS(IRID,IKOL,TEXT,NN5)
        IF(INP.EQ.5) CALL READR(1)
        IF(INP.EQ.5)
READ(terk,'(2F10.0)',ERR=88)    AZ,ANZ
        IF(INP.EQ.5) WRITE(  8,'(2F10.3,'';
Talas Podela'')') AZ,ANZ
IF(INP.NE.5) READ( INP,'(2F10.0)') AZ,ANZ
        NZ=ANZ
10 TEXT='+-X , +-Y , +-X1 , +-Y1 , NX , +-
X2,+-Y2,+-Y3,+-Z1,+-Z2,Krug(0/1),Xc,Yc'
       CALL ZAPIS(IRID,IKOL,TEXT,NN5)
       TEXT = 'Linije , An1 , An2'
       CALL READR(1)
READ(terk,'(13F10.0)',ERR=88) AX,AY,AX1,
AY1,ANX,AX2,AY2,AY3,AZ1,AZ2,ZC,XC,YC
WRITE(  8,'(13F10.3       )') AX,AY,AX1,
AY1,ANX,AX2,AY2,AY3,AZ1,AZ2,ZC,XC,YC
        IF(AX.EQ.0.0.AND.AY.EQ.0.0) GOTO 11
       CALL READR(1)
      READ(terk,'(3F10.0)', ERR=88) B,B1,B2
WRITE(  8,'(3F10.3,''; Talas Podela'')')
B,B1,B2
        ELSE
READ (INP,'(13F10.0)',ERR=88)AX,AY,AX1,AY1,
ANX,AX2,AY2,AY3,AZ1,AZ2,ZC,XC,YC
        IF(AX.EQ.0.0.AND.AY.EQ.0.0) GOTO 11
 READ(  INP,'(4F10.0)',ERR=88) B,B1,B2,ZC
        END IF
       IF(AAX.EQ.0.0) AAX=AX
          IF(AX1.EQ.0.0) AX1=AX/2.
          IF(AX2.EQ.0.0) AX2=AX/2.
          IF(AY1.EQ.0.0) AY1=AX/4.
          IF(AZ1.EQ.0.0) AZ1=AZ/2.
          IF(AZ2.EQ.0.0) AZ2=AZ1
          IF(ANX.EQ.0  ) ANX=1

          NX=ANX
          N0=B
          N1=B1
```

```
      CALL ZAPIS(IRID,IKOL,TEXT,NN5)
        IF(INP.EQ.5) THEN
          Z(2+L)=Z(1+L)
          Z(3+L)=Z(1+L)
          Z(7+L)=Z(1+L)+AZ2
          Z(8+L)=Z(1+L)+AZ1
          Z(9+L)=Z(1+L)+AZ2
        IF(ZC.EQ.0.0) THEN
     X(L+2)=X(L+1)+AX2-ABS(AY/AXY)*AY2
Y(L+2)=Y(L+1)+AX2*ABS(AY/AX)+ABS(AX/AXY)*
AY2
          X(L+3)=X(L+1)+AX
          Y(L+3)=Y(L+1)+AY
          X(L+7)=X(L+1)-ABS(AY/AXY)*AY3
          Y(L+7)=Y(L+1)+ABS(AX/AXY)*AY3
          X(L+8)=X(L+1)+AX1-ABS(AY/AXY)*AY1
Y(L+8)=Y(L+1)+AX1*ABS(AY/AX)+ABS(AX/AXY)*
AY1
          X(L+9)=X(L+1)+AX-ABS(AY/AXY)*AY3
          Y(L+9)=Y(L+1)+AY+ABS(AX/AXY)*AY3
        END IF
        IF(ZC.EQ.1.0) THEN
CALL UGAO(XC,X(1),YC,Y(1),R0,F0,PISTRAD)
X(L+2)=(R0+AY2)*COS(F0+AX2*PISTRAD)+XC
Y(L+2)=(R0+AY2)*SIN(F0+AX2*PISTRAD)+YC
X(L+3)= R0      *COS(F0+AX *PISTRAD)+XC
Y(L+3)= R0      *SIN(F0+AX *PISTRAD)+YC
        X(L+7)=(R0+AY3)*COS(F0)+XC
        Y(L+7)=(R0+AY3)*SIN(F0)+YC
X(L+8)=(R0+AY1)*COS(F0+AX1*PISTRAD)+XC
Y(L+8)=(R0+AY1)*SIN(F0+AX1*PISTRAD)+YC
X(L+9)=(R0+AY3)*COS(F0+AX *PISTRAD)+XC
Y(L+9)=(R0+AY3)*SIN(F0+AX *PISTRAD)+YC
        END IF
        DO 7 I=1,3
           X(L+I+3)=X(L+I)
           Y(L+I+3)=Y(L+I)
    7     Z(L+I+3)=Z(L+I)+AZ
      L1=9
        NLI1=NLIN
        NPO1=NPOV
        IF(N0.EQ.2.OR.N0.EQ.3.OR.N0.EQ.4.OR
.N0.EQ.6.OR.N0.EQ.10.OR.N0.EQ
        +.11.OR.N0.EQ.14.OR.N0.EQ.15)THEN
         NLIN=NLIN+1
        LL(NLIN,1)=1+L
        LL(NLIN,2)=7+L
         NLIN=NLIN+1
        LL(NLIN,1)=7+L
        LL(NLIN,2)=4+L
          NLIN=NLIN+1
        LL(NLIN,1)=3+L
        LL(NLIN,2)=9+L
         NLIN=NLIN+1
        LL(NLIN,1)=9+L
        LL(NLIN,2)=6+L
      END IF
        IF(N0.EQ.1.OR.N0.EQ.3.OR.N0.EQ.5.OR
.N0.EQ.7.OR.N0.EQ.9.OR.N0.EQ.11
        +.OR.N0.EQ.13.OR.N0.EQ.15) THEN
         NLIN=NLIN+1
        LL(NLIN,1)=1+L
        LL(NLIN,2)=2+L
         NLIN=NLIN+1
        LL(NLIN,1)=2+L
        LL(NLIN,2)=3+L
          NLIN=NLIN+1
        LL(NLIN,1)=4+L
        LL(NLIN,2)=5+L
         NLIN=NLIN+1
        LL(NLIN,1)=5+L
        LL(NLIN,2)=6+L
      END IF
```

```
          N2=B2
        AXY=SQRT(AX*AX+AY*AY)
         NLIN=NLIN+1
        LL(NLIN,1)=7+L
        LL(NLIN,2)=8+L
         NLIN=NLIN+1
        LL(NLIN,1)=8+L
        LL(NLIN,2)=9+L
       END IF
IF(N0.EQ.6.OR.N0.EQ.7.OR.N0.EQ.8.OR.N0.EQ.9
.OR.N0.EQ.10.OR.N0.EQ.11.OR.N0.EQ.12.OR.N0.
EQ.13.OR.N0.EQ.14.OR.N0.EQ.15)THEN
         NLIN=NLIN+1
        LL(NLIN,1)=2+L
        LL(NLIN,2)=8+L
         NLIN=NLIN+1
        LL(NLIN,1)=8+L
        LL(NLIN,2)=5+L
       END IF
IF(N0.EQ.12.OR.N0.EQ.13.OR.N0.EQ.14.OR.N0.E
Q.15) THEN
       DO I=1,4
       K=1
        IF(I.EQ.1) K=0
        IF(I.EQ.4) K=2
          NLIN=NLIN+1
        LL(NLIN,1)=I+K+L
        LL(NLIN,2)=8+L
       END DO
       END IF
       DO 6 I=1,8
        NPOV=NPOV+1
        I1=I
        IF(I.EQ.4) I1=9
        IF(I.GE.5) I1=11-I
        IF(I.EQ.8) I1=7
        I2=I+1
        IF(I.EQ.3) I2=9
        IF(I.GE.4) I2=11-I
        IF(I.EQ.7) I2=7
        IF(I.EQ.8) I2=1
        LP(NPOV,1)=I1+L
        LP(NPOV,2)=I2+L
        LP(NPOV,3)= 7+L
    6   LP(NPOV,4)=LP(NPOV,3)
        NLI1=NLIN-NLI1
        NPO1=NPOV-NPO1
       IF(NX.GT.1) THEN
CALL TACKEM
(NLT,X,Y,Z,L,L1,NX,AX,AY,0.0,XC,YC,ZC)
CALL PLOCEM
(NLT,LEN,LL,LP,L1,NX,NLIN,NPOV,NLI1,NPO1)
        ELSE
        L=9
        END IF
        X(L+1)=X(L-L1+3)
        Y(L+1)=Y(L-L1+3)
        Z(L+1)=Z(L-L1+3)
        GOTO 10
   11 CONTINUE
        IF(NZ.GT.1) THEN
        NLI1=NLIN
        NPO1=NPOV
        L2=L
CALL TACKEM
(NLT,X,Y,Z,L,L2,NZ,0.0,0.0,AZ,XC,YC,-1.)
CALL PLOCEM
(NLT,LEN,LL,LP,L2,NZ,NLIN,NPOV,NLI1,NPO1)
        END IF
CALL ISTTAC
(L,NLT,X,Y,Z,LG,LL,LP,NLIN,NPOV,LEN,AAX)
       IF(N1.EQ.1) THEN
        CALL IZBTAC(NLT,X,Y,Z,L,A1,IDT,1)
CALL IZBELE(NLT,LEN,LL,LP,NLIN,NPOV,IDT)
```

```
IF(N0.EQ.4.OR.N0.EQ.5.OR.N0.EQ.8.OR.N0.EQ.9
.OR.N0.EQ.10.OR.N0.EQ.11.OR.N0.EQ.12.OR.N0.
EQ.13.OR.N0.EQ.14.OR.N0.EQ.15)THEN
CALL IZBTAC(NLT,X,Y,Z,L,NX*AX-AX+A1,IDT,2)
       END IF
       RETURN
       END
```

```
       END IF
       IF(N2.EQ.1) THEN
CALL IZBELE(NLT,LEN,LL,LP,NLIN,NPOV,IDT)
```

```
      SUBROUTINE TALAS9
(NLT,LEN,X,Y,Z,LG,LL,LP,L,NLIN,NPOV,IDT)
       INCLUDE 'ZAJZON1M.FOR'
      CHARACTER*70  TEXT
      DIMENSION X(NLT),Y(NLT),Z(NLT)
       INTEGER*2 LG(NLT,LEN),LL(NLT,LEN),
LP(NLT,LEN),IDT(NLT)
       GOTO 9
   88 CALL GRESKA(1)
      RETURN
    9 continue
      L=0
       NLIN=0
       NPOV=0
      AAX=0.0
      TEXT = '+-AZ , NZ'
      CALL ZAPIS(IRID,IKOL,TEXT,NN5)
       IF(INP.EQ.5) CALL READR(1)
       IF(INP.EQ.5)
READ(terk,'(2F10.0)',ERR=88)   AZ,ANZ
       IF(INP.EQ.5) WRITE(  8,'(2F10.3,'';
Talas Podela'')') AZ,ANZ
       IF(INP.NE.5) READ(
INP,'(2F10.0)',ERR=88)              AZ,ANZ
       NZ=ANZ
   10 TEXT='+-X , +-Y , +-X1 , +-Y1 , NX ,
+-Y2 , +-Z1 ,Krug(0/1),Xc,Yc'
      CALL ZAPIS(IRID,IKOL,TEXT,NN5)
      TEXT = 'Linije , An2'
      CALL ZAPIS(IRID,IKOL,TEXT,NN5)
      IF(INP.EQ.5) THEN
       CALL READR(1)
      READ(terk,'(10F10.0)',ERR=88)
AX,AY,AX1,AY1,ANX,AY2,AZ1,ZC,XC,YC
      WRITE(  8,'(10F10.3       )')
AX,AY,AX1,AY1,ANX,AY2,AZ1,ZC,XC,YC
        IF(AX.EQ.0.0.AND.AY.EQ.0.0) GOTO 11
      CALL READR(1)
READ(terk,'( 2F10.0)',    ERR=88) B,B2
WRITE(  8,'( 2F10.3,''; Talas Podela'')')
B,B2
       ELSE
      READ( INP,'(10F10.0)',ERR=88)
AX,AY,AX1,AY1,ANX,AY2,AZ1,ZC,XC,YC
        IF(AX.EQ.0.0.AND.AY.EQ.0.0) GOTO 11
      READ( INP,'( 2F10.0)',ERR=88) B,B2
       END IF
      IF(AAX.EQ.0.0) AAX=AX
         IF(AX1.EQ.0.0) AX1=AX/2.
         IF(AY1.EQ.0.0) AY1=AX/4.
         IF(AY2.EQ.0.0) AY2=AX/4.
         IF(AZ1.EQ.0.0) AZ1=AZ/2.
         IF(ANX.EQ.0  ) ANX=1
       IF(ZC.EQ.0.0) AY=0.0
         NX=ANX
         N0=B
         N2=B2
       AXY=SQRT(AX*AX+AY*AY)
      Z(2+L)=Z(1+L)
       Z(3+L)=Z(1+L)
         Z(7+L)=Z(1+L)+AZ1
         Z(8+L)=Z(3+L)+AZ1
       IF(ZC.EQ.0.0) THEN
         X(2+L)=X(1+L)+AX1-ABS(AY/AXY)*AY1
        Y(3+L)=Y(1+L)+AY
  X(L+7)=X(L+1)            -ABS(AY/AXY)*AY2
  Y(L+7)=Y(L+1)            +ABS(AX/AXY)*AY2
  X(L+8)=X(L+1)+AX         -ABS(AY/AXY)*AY2
  Y(L+8)=Y(L+1)+AY         +ABS(AX/AXY)*AY2
       END IF
       IF(ZC.EQ.1.) THEN
CALL UGAO(XC,X(1),YC,Y(1),R0,F0,PISTRAD)
X(2+L)=(R0+AY1)*COS(F0+AX1*PISTRAD)+XC
Y(2+L)=(R0+AY1)*SIN(F0+AX1*PISTRAD)+YC
X(3+L)= R0     *COS(F0+AX *PISTRAD)+XC
Y(3+L)= R0     *SIN(F0+AX *PISTRAD)+YC
X(L+7)=(R0+AY2)*COS(F0             )+XC
Y(L+7)=(R0+AY2)*SIN(F0             )+YC
X(8+L)=(R0+AY2)*COS(F0+AX *PISTRAD)+XC
Y(8+L)=(R0+AY2)*SIN(F0+AX *PISTRAD)+YC
       END IF
       DO 7 I=1,3
          X(I+3+L)=X(I+L)
          Y(I+3+L)=Y(I+L)
    7     Z(I+3+L)=Z(I+L)+AZ
      L1=8
       NLI1=NLIN
       NPO1=NPOV
       IF(N0.EQ.2.OR.N0.EQ.3.OR.N0.EQ.5.OR
.N0.EQ.6.OR.N0.EQ.9.OR.N0.EQ.10.)THEN
        NLIN=NLIN+1
       LL(NLIN,1)=1+L
       LL(NLIN,2)=7+L
        NLIN=NLIN+1
       LL(NLIN,1)=7+L
       LL(NLIN,2)=4+L
          NLIN=NLIN+1
       LL(NLIN,1)=3+L
       LL(NLIN,2)=8+L
        NLIN=NLIN+1
       LL(NLIN,1)=8+L
       LL(NLIN,2)=6+L
      END IF
IF(N0.EQ.1.OR.N0.EQ.3.OR.N0.EQ.4.OR.N0.EQ.6
.OR.N0.EQ.7.OR.N0.EQ.9. OR.N0.EQ.10) THEN
        NLIN=NLIN+1
       LL(NLIN,1)=1+L
       LL(NLIN,2)=2+L
        NLIN=NLIN+1
       LL(NLIN,1)=2+L
       LL(NLIN,2)=3+L
          NLIN=NLIN+1
       LL(NLIN,1)=4+L
       LL(NLIN,2)=5+L
        NLIN=NLIN+1
       LL(NLIN,1)=5+L
       LL(NLIN,2)=6+L
      END IF
  IF(N0.EQ.6.OR.N0.EQ.7.OR.N0.EQ.8.
       +  OR.N0.EQ.9.OR.N0.EQ.10) THEN
        NLIN=NLIN+1
       LL(NLIN,1)=7+L
       LL(NLIN,2)=8+L
      END IF
IF(N0.EQ.7.OR.N0.EQ.8.OR.N0.EQ.9.OR.N0.EQ.1
0) THEN
      DO 1 I=1,2
      DO 1 J=1,2
```

```
Y(2+L)=Y(1+L)+AX1*ABS(AY/AX)+ABS(AX/AXY)*
AY1
         X(3+L)=X(1+L)+AX
      END IF
       NPOV=NPOV+1
        LP(NPOV,1)=1+L
        LP(NPOV,2)=2+L
         LP(NPOV,3)=7+L
         LP(NPOV,4)=LP(NPOV,3)
       NPOV=NPOV+1
        LP(NPOV,1)=2+L
        LP(NPOV,2)=3+L
         LP(NPOV,3)=8+L
         LP(NPOV,4)=LP(NPOV,3)
       NPOV=NPOV+1
        LP(NPOV,1)=8+L
        LP(NPOV,2)=6+L
         LP(NPOV,3)=5+L
         LP(NPOV,4)=LP(NPOV,3)
       NPOV=NPOV+1
        LP(NPOV,1)=5+L
        LP(NPOV,2)=4+L
         LP(NPOV,3)=7+L
         LP(NPOV,4)=LP(NPOV,3)
       NPOV=NPOV+1
        LP(NPOV,1)=2+L
        LP(NPOV,2)=8+L
         LP(NPOV,3)=7+L
         LP(NPOV,4)=LP(NPOV,3)
       NPOV=NPOV+1
        LP(NPOV,1)=8+L
        LP(NPOV,2)=5+L
         LP(NPOV,3)=7+L
         LP(NPOV,4)=LP(NPOV,3)
```

```
        NLIN=NLIN+1
       LL(NLIN,1)=I*2+J-1+L
       LL(NLIN,2)=J+6+L
    1 CONTINUE
        NLI1=NLIN-NLI1
        NPO1=NPOV-NPO1
      IF(NX.GT.1) THEN
CALL TACKEM
(NLT,X,Y,Z,L,L1,NX,AX,AY,0.0,XC,YC,ZC)
CALL PLOCEM
(NLT,LEN,LL,LP,L1,NX,NLIN,NPOV,NLI1,NPO1)
      ELSE
        L=8
        END IF
        X(L+1)=X(L-L1+3)
        Y(L+1)=Y(L-L1+3)
        Z(L+1)=Z(L-L1+3)
        GOTO 10
   11 CONTINUE
        IF(NZ.GT.1) THEN
        NLI1=NLIN
        NPO1=NPOV
        L2=L
CALL TACKEM
(NLT,X,Y,Z,L,L2,NZ,0.0,0.0,AZ,XC,YC,-1.)
CALL PLOCEM
(NLT,LEN,LL,LP,L2,NZ,NLIN,NPOV,NLI1,NPO1)
        END IF
CALL ISTTAC
(L,NLT,X,Y,Z,LG,LL,LP,NLIN,NPOV,LEN,AAX)
        IF(N2.EQ.1) THEN
CALL IZBTAC(NLT,X,Y,Z,L,NX*AX-AX+AX1,IDT,2)
CALL IZBELE(NLT,LEN,LL,LP,NLIN,NPOV,IDT)
        END IF
        RETURN
        END
```

---

**SUBROUTINE TALAS10**

```
(NLT,LEN,X,Y,Z,LG,LL,LP,L,NLIN,NPOV,IDT)
        INCLUDE 'ZAJZON1M.FOR'
      CHARACTER*70  TEXT
      DIMENSION X(NLT),Y(NLT),Z(NLT)
        INTEGER*2 LG(NLT,LEN),LL(NLT,LEN),
LP(NLT,LEN),IDT(NLT)
        GOTO 9
    8 CALL GRESKA(1)
    9 continue
      L=0
        NLIN=0
        NPOV=0
       AAX=0.0
      TEXT = '+-AZ , NZ'
      CALL ZAPIS(IRID,IKOL,TEXT,NN5)
       IF(INP.EQ.5) CALL READR(1)
       IF(INP.EQ.5)
READ(terk,'(2F10.0)',ERR=8)   AZ,ANZ
       IF(INP.EQ.5) WRITE(  8,'(2F10.3,'';
Talas Podela'')') AZ,ANZ
IF(INP.NE.5) READ(INP,'(2F10.0)') AZ,ANZ
        NZ=ANZ
10 TEXT = '+-X , +-Y , +-X1 , +-Y1 , NX ,
+-Y2 , +-Y3 , +-Z1 , Krug(0/1) , Xc , Yc'
      CALL ZAPIS(IRID,IKOL,TEXT,NN5)
      TEXT = 'Linije , An1 , An2'
      CALL ZAPIS(IRID,IKOL,TEXT,NN5)
      IF(INP.EQ.5) THEN
       CALL READR(1)
READ(terk,'(11F10.0)',ERR=8)AX,AY,AX1,AY1,A
NX,AY2,AY3,AZ1,ZC,XC,YC
WRITE(  8,'(11F10.3
)')AX,AY,AX1,AY1,ANX,AY2,AY3,AZ1,ZC,XC,YC
       IF(AX.EQ.0.0.AND.AY.EQ.0.0)GOTO 11
```

```
        ELSE
      READ(
INP,'(11F10.0)',ERR=8)AX,AY,AX1,AY1,ANX,AY2
,AY3,AZ1,ZC,XC,YC
       IF(AX.EQ.0.0.AND.AY.EQ.0.0)GOTO 11
      READ( INP,'( 3F10.0)',ERR=8)B,B1,B2
        END IF
      IF(AAX.EQ.0.0) AAX=AX
          IF(AX1.EQ.0.0) AX1=AX/2.
          IF(AY1.EQ.0.0) AY1=AX/4.
          IF(AZ1.EQ.0.0) AZ1=AZ/2.
          IF(ANX.EQ.0.0) ANX=1
          NX=ANX
          N0=B
          N1=B1
          N2=B2
        AXY=SQRT(AX*AX+AY*AY)
          Z(L+2)=Z(L+1)
          Z(L+3)=Z(L+1)
          Z(L+7)=Z(L+1)+AZ1
          Z(L+8)=Z(L+1)+AZ1
          Z(L+9)=Z(L+1)+AZ1
          IF(ZC.EQ.0.0) THEN
          X(L+2)=X(L+1)+AX1-ABS(AY/AXY)*AY2
Y(L+2)=Y(L+1)+AX1*ABS(AY/AX)+ABS(AX/AXY)*
AY2
          X(L+3)=X(L+1)+AX
          Y(L+3)=Y(L+1)+AY
          X(L+7)=X(L+1)-ABS(AY/AXY)*AY3
          Y(L+7)=Y(L+1)+ABS(AX/AXY)*AY3
          X(L+8)=X(L+1)+AX1-ABS(AY/AXY)*AY1
Y(L+8)=Y(L+1)+AX1*ABS(AY/AX)+ABS(AX/AXY)*
AY1
          X(L+9)=X(L+1)+AX-ABS(AY/AXY)*AY3
          Y(L+9)=Y(L+1)+AY+ABS(AX/AXY)*AY3
```

```
      CALL READR(1)
     READ(terk,'( 3F10.0)',ERR=8)   B,B1,B2
     WRITE(  8,'( 3F10.3,''; Talas
Podela'')') B,B1,B2
X(L+2)=(R0+AY2)*COS(F0+AX1*PISTRAD)+XC
Y(L+2)=(R0+AY2)*SIN(F0+AX1*PISTRAD)+YC
X(L+3)= R0     *COS(F0+AX *PISTRAD)+XC
Y(L+3)= R0     *SIN(F0+AX *PISTRAD)+YC
X(L+7)=(R0+AY3)*COS(F0             )+XC
Y(L+7)=(R0+AY3)*SIN(F0             )+YC
X(L+8)=(R0+AY1)*COS(F0+AX1*PISTRAD)+XC
Y(L+8)=(R0+AY1)*SIN(F0+AX1*PISTRAD)+YC
X(L+9)=(R0+AY3)*COS(F0+AX *PISTRAD)+XC
Y(L+9)=(R0+AY3)*SIN(F0+AX *PISTRAD)+YC
        END IF
      DO 1 I=1,3
          X(L+I+3)=X(L+I)
          Y(L+I+3)=Y(L+I)
    1     Z(L+I+3)=Z(L+I)+AZ
      L1=9
        NLI1=NLIN
        NPO1=NPOV
IF(N0.EQ.2.OR.N0.EQ.3.OR.N0.EQ.6.OR.N0.EQ.7
.OR.N0.EQ.10.OR.N0.EQ.11.OR.N0.EQ.14.OR.N0.
EQ.15)THEN
        NLIN=NLIN+1
       LL(NLIN,1)=1+L
       LL(NLIN,2)=7+L
        NLIN=NLIN+1
       LL(NLIN,1)=7+L
       LL(NLIN,2)=4+L
          NLIN=NLIN+1
       LL(NLIN,1)=3+L
       LL(NLIN,2)=9+L
        NLIN=NLIN+1
       LL(NLIN,1)=9+L
       LL(NLIN,2)=6+L
      END IF
IF(N0.EQ.1.OR.N0.EQ.3.OR.N0.EQ.5.OR.N0.EQ.7
.OR.N0.EQ.9.OR.N0.EQ.11.OR.N0.EQ.13.OR.N0.E
Q.15) THEN
        NLIN=NLIN+1
       LL(NLIN,1)=1+L
       LL(NLIN,2)=2+L
        NLIN=NLIN+1
       LL(NLIN,1)=2+L
       LL(NLIN,2)=3+L
          NLIN=NLIN+1
       LL(NLIN,1)=4+L
       LL(NLIN,2)=5+L
        NLIN=NLIN+1
       LL(NLIN,1)=5+L
       LL(NLIN,2)=6+L
      END IF
IF(N0.EQ.4.OR.N0.EQ.5.OR.N0.EQ.6.OR.N0.EQ.7
.OR.N0.EQ.12.OR.N0.EQ.13.OR.N0.EQ.14.OR.N0.
EQ.15) THEN
      DO I=2,8,2
      K=1
        IF(I.EQ.2) K=0
          NLIN=NLIN+1
       LL(NLIN,1)=8+L
       LL(NLIN,2)=I+K+L
      END DO
      END IF
IF(N0.EQ.8.OR.N0.EQ.9.OR.N0.EQ.10.OR.N0.EQ.
11.OR.N0.EQ.12.OR.N0.EQ.13.OR.N0.EQ.14.OR.N
0.EQ.15) THEN
      DO I=2,8,2
        K1=I+5
        IF(I.EQ.6) K1=5
        END IF
          IF(ZC.EQ.1.0) THEN
CALL UGAO
(XC,X(L+1),YC,Y(L+1),R0,F0,PISTRAD)
        IF(I.EQ.8) K1=2
      K2=1
        IF(I.EQ.2) K2=0
          NLIN=NLIN+1
       LL(NLIN,1)=  K1+L
       LL(NLIN,2)=I+K2+L
      END DO
      END IF
      DO 6 I=1,2
      NPOV=NPOV+1
          LP(NPOV,1)=I+L
        LP(NPOV,2)=I+1+L
          LP(NPOV,3)=(I-1)*2+7+L
        LP(NPOV,4)=LP(NPOV,3)
      NPOV=NPOV+1
          LP(NPOV,1)=I+4+L
        LP(NPOV,2)=I+3+L
          LP(NPOV,3)=(I-1)*2+7+L
    6   LP(NPOV,4)=LP(NPOV,3)
        DO 7 I=3,9,2
      NPOV=NPOV+1
         K1=I
        IF(I.EQ.3) K1=2
        IF(I.EQ.3) K2=9
        IF(I.EQ.5) K2=7
        IF(I.EQ.7) K2=2
        IF(I.EQ.9) K2=5
         LP(NPOV,1)=K1+L
         LP(NPOV,2)=K2+L
          LP(NPOV,3)=8+L
    7   LP(NPOV,4)=LP(NPOV,3)
        NLI1=NLIN-NLI1
        NPO1=NPOV-NPO1
      IF(NX.GT.1) THEN
CALL TACKEM
(NLT,X,Y,Z,L,L1,NX,AX,AY,0.0,XC,YC,ZC)
CALL PLOCEM
(NLT,LEN,LL,LP,L1,NX,NLIN,NPOV,NLI1,NPO1)
      ELSE
        L=9
        END IF
        X(L+1)=X(L-L1+3)
        Y(L+1)=Y(L-L1+3)
        Z(L+1)=Z(L-L1+3)
        GOTO 10
   11 CONTINUE
        IF(NZ.GT.1) THEN
        NLI1=NLIN
        NPO1=NPOV
        L2=L
CALL TACKEM
(NLT,X,Y,Z,L,L2,NZ,0.0,0.0,AZ,XC,YC,-1.)
CALL PLOCEM
(NLT,LEN,LL,LP,L2,NZ,NLIN,NPOV,NLI1,NPO1)
        END IF
CALL ISTTAC
(L,NLT,X,Y,Z,LG,LL,LP,NLIN,NPOV,LEN,AAX)
      IF(N1.EQ.1) THEN
        CALL IZBTAC(NLT,X,Y,Z,L,A1,IDT,1)
CALL IZBELE(NLT,LEN,LL,LP,NLIN,NPOV,IDT)
        END IF
        IF(N2.EQ.1) THEN
CALL IZBTAC(NLT,X,Y,Z,L,NX*AX-AX+A1,IDT,2)
CALL IZBELE(NLT,LEN,LL,LP,NLIN,NPOV,IDT)
        END IF
        RETURN
        END
```

```
      SUBROUTINE TALAS11
(NLT,LEN,X,Y,Z,LG,LL,LP,L,NLIN,NPOV,IDT)
        INCLUDE 'ZAJZON1M.FOR'
        INTEGER*2 LG(NLT,LEN),LL(NLT,LEN),
LP(NLT,LEN),IDT(NLT)
        GOTO 9
   88 CALL GRESKA(1)
    9 continue
      L=0
        NLIN=0
        NPOV=0
      AAX=0.0
      TEXT = '+-AZ , NZ'
      CALL ZAPIS(IRID,IKOL,TEXT,NN5)
       IF(INP.EQ.5) CALL READR(1)
       IF(INP.EQ.5)
READ(terk,'(2F10.0)',ERR=88)   AZ,ANZ
       IF(INP.EQ.5) WRITE(  8,'(2F10.3,'';
Talas Podela'')')  AZ,ANZ
       IF(INP.NE.5)
READ(INP,'(2F10.0)',ERR=88) AZ,ANZ
        NZ=ANZ
   10 TEXT = '+-X , +-Y , +-X1 , +-Y1 , NX
, +-X2,+-X3,+-Y1,+-Y2,+-Y3,+-
     +Y4,+-Z1,+-Z2,Z3 , Krug(0/1),Xc,Yc'
      CALL ZAPIS(IRID,IKOL,TEXT,NN5)
TEXT = 'Linije , An1 , An2 , An3 , An4'
      CALL ZAPIS(IRID,IKOL,TEXT,NN5)
      IF(INP.EQ.5) THEN
       CALL READR(1)
      READ(terk,'(16F10.0)',ERR=88)
AX,AY,AX1,AY1,ANX,AX2,AX3,AY2,AY3,
        +AY4,AZ1,AZ2,AZ3,ZC,XC,YC
      WRITE(  8,'(14F10.3       )')
AX,AY,AX1,AY1,ANX,AX2,AX3,AY2,AY3,
        +AY4,AZ1,AZ2,AZ3,ZC,XC,YC
        IF(AX.EQ.0.0.AND.AY.EQ.0.0)GOTO 11
      CALL READR(1)
      READ(terk,'( 5F10.0)',ERR=88)
B,B1,B2,B3,B4
      WRITE(  8,'( 5F10.3,''; Talas
Podela'')') B,B1,B2,B3,B4
        ELSE
      READ( INP,'(14F10.0)',ERR=88)
AX,AY,AX1,AY1,ANX,AX2,AX3,AY2,AY3,
        +AY4,AZ1,AZ2,AZ3,ZC,XC,YC
       IF(AX.EQ.0.0.AND.AY.EQ.0.0)GOTO 11
      READ( INP,'( 5F10.0)',ERR=88)
B,B1,B2,B3,B4
        END IF
      IF(AAX.EQ.0.0) AAX=AX
          IF(AX1.EQ.0.0) AX1=AX/2.
          IF(AX2.EQ.0.0) AX2=AX/4.
          IF(AX3.EQ.0.0) AX3=AX*3./4.
          IF(AY1.EQ.0.0) AY1=AX/4.
C         IF(AY2.EQ.0.0) AY2=AY1
C         IF(AY3.EQ.0.0) AY3=AY1
          IF(AY4.EQ.0.0) AY4=AY1
          IF(AZ1.EQ.0.0) AZ1=AZ/2.
          IF(AZ2.EQ.0.0) AZ2=AZ/4.

          IF(AZ3.EQ.0.0) AZ3=AZ/4.*3.
          IF(ANX.EQ.0.0) ANX=1
          NX=ANX
          N0=B
        N1=B1
        N2=B2
        N3=B3
        N4=B4
        AXY=SQRT(AX*AX+AY*AY)
          Z(L+ 2)=Z(L+1)
          Z(L+ 3)=Z(L+1)
```

```
      CHARACTER*70  TEXT
      DIMENSION X(NLT),Y(NLT),Z(NLT)
          Z(L+12)=Z(L+1)+AZ3
          Z(L+13)=Z(L+1)+AZ1
          IF(ZC.EQ.0.0) THEN
         X(L+ 2)=X(L+1)+AX1-ABS(AY/AXY)*AY2
Y(L+2)=Y(L+1)+AX1*ABS(AY/AX)+ABS(AX/AXY)*
AY2
          X(L+ 3)=X(L+1)+AX
          Y(L+ 3)=Y(L+1)+AY
          X(L+ 7)=X(L+1)-ABS(AY/AXY)*AY3
          Y(L+ 7)=Y(L+1)+ABS(AX/AXY)*AY3
  X(L+ 8)=X(L+1)+AX1-ABS(AY/AXY)*AY1
Y(L+8)=Y(L+1)+AX1*ABS(AY/AX)+ABS(AX/AXY)*
AY1
          X(L+ 9)=X(L+1)+AX-ABS(AY/AXY)*AY3
          Y(L+ 9)=Y(L+1)+AY+ABS(AX/AXY)*AY3
         X(L+10)=X(L+1)+AX1-ABS(AY/AXY)*AY4
Y(L+10)=Y(L+1)+AX1*ABS(AY/AX)+ABS(AX/AXY)*
AY4
X(L+11)=X(L+1)+AX3-ABS(AY/AXY)*AY4
Y(L+11)=Y(L+1)+AX3*ABS(AY/AX)+ABS(AX/AXY)*
AY4
X(L+12)=X(L+1)+AX1-ABS(AY/AXY)*AY4
Y(L+12)=Y(L+1)+AX1*ABS(AY/AX)+ABS(AX/AXY)*
AY4
X(L+13)=X(L+1)+AX2-ABS(AY/AXY)*AY4
Y(L+13)=Y(L+1)+AX2*ABS(AY/AX)+ABS(AX/AXY)*
AY4
        END IF
          IF(ZC.EQ.1.0) THEN
CALL UGAO
(XC,X(L+1),YC,Y(L+1),R0,F0,PISTRAD)
 X(L+ 2)=(R0+AY2)*COS(F0+AX1*PISTRAD)+XC
 Y(L+ 2)=(R0+AY2)*SIN(F0+AX1*PISTRAD)+YC
 X(L+ 3)= R0     *COS(F0+AX *PISTRAD)+XC
 Y(L+ 3)= R0     *SIN(F0+AX *PISTRAD)+YC
 X(L+ 7)=(R0+AY3)*COS(F0            )+XC
 Y(L+ 7)=(R0+AY3)*SIN(F0            )+YC
 X(L+ 8)=(R0+AY1)*COS(F0+AX1*PISTRAD)+XC
 Y(L+ 8)=(R0+AY1)*SIN(F0+AX1*PISTRAD)+YC
 X(L+ 9)=(R0+AY3)*COS(F0+AX *PISTRAD)+XC
 Y(L+ 9)=(R0+AY3)*SIN(F0+AX *PISTRAD)+YC
 X(L+10)=(R0+AY4)*COS(F0+AX1*PISTRAD)+XC
 Y(L+10)=(R0+AY4)*SIN(F0+AX1*PISTRAD)+YC
 X(L+11)=(R0+AY4)*COS(F0+AX3*PISTRAD)+XC
 Y(L+11)=(R0+AY4)*SIN(F0+AX3*PISTRAD)+YC
 X(L+12)=(R0+AY4)*COS(F0+AX1*PISTRAD)+XC
 Y(L+12)=(R0+AY4)*SIN(F0+AX1*PISTRAD)+YC
 X(L+13)=(R0+AY4)*COS(F0+AX2*PISTRAD)+XC
 Y(L+13)=(R0+AY4)*SIN(F0+AX2*PISTRAD)+YC
        END IF
      DO 1 I=1,3
          X(L+I+3)=X(L+I)
          Y(L+I+3)=Y(L+I)
    1     Z(L+I+3)=Z(L+I)+AZ
      L1=13
        NLI1=NLIN
        NPO1=NPOV
IF(N0.EQ.1.OR.N0.EQ.2.OR.N0.EQ.8.OR.N0.EQ.9
.OR.N0.EQ.10.OR.N0.EQ.
11.OR.N0.EQ.14.OR.N0.EQ.15)THEN
        NLIN=NLIN+1
       LL(NLIN,1)=1
       LL(NLIN,2)=7
        NLIN=NLIN+1
       LL(NLIN,1)=7
       LL(NLIN,2)=4
          NLIN=NLIN+1
       LL(NLIN,1)=3
       LL(NLIN,2)=9
```

```
          Z(L+ 7)=Z(L+1)+AZ1
          Z(L+ 8)=Z(L+1)+AZ1
          Z(L+ 9)=Z(L+1)+AZ1
          Z(L+10)=Z(L+1)+AZ2
          Z(L+11)=Z(L+1)+AZ1
.N0.EQ.7.OR.N0.EQ.10.OR.N0.EQ.
       +11.OR.N0.EQ.13.OR.N0.EQ.15) THEN
        NLIN=NLIN+1
       LL(NLIN,1)=1
       LL(NLIN,2)=2
        NLIN=NLIN+1
       LL(NLIN,1)=2
       LL(NLIN,2)=3
           NLIN=NLIN+1
       LL(NLIN,1)=4
       LL(NLIN,2)=5
        NLIN=NLIN+1
       LL(NLIN,1)=5
       LL(NLIN,2)=6
      END IF
       IF(N0.EQ.4.OR.N0.EQ.6.OR.N0.EQ.8.OR
.N0.EQ.10.OR.N0.EQ.12.OR.N0.EQ.
       +13.OR.N0.EQ.14.OR.N0.EQ.15) THEN
      DO I=1,7,2
        NLIN=NLIN+1
       LL(NLIN,1)=8
       LL(NLIN,2)=I
      END DO
      END IF
       IF(N0.EQ.7.OR.N0.EQ.9.OR.N0.EQ.11.O
R.N0.EQ.12.OR.N0.EQ.
       +13.OR.N0.EQ.14.OR.N0.EQ.15) THEN
      DO I=10,13
        NLIN=NLIN+1
       LL(NLIN,1)=8
       LL(NLIN,2)=I
      END DO
       DO I=10,13
        NLIN=NLIN+1
       IF(I.EQ.10) M=2
       IF(I.EQ.11) M=9
       IF(I.EQ.12) M=5
       IF(I.EQ.13) M=7
       LL(NLIN,1)=I
       LL(NLIN,2)=M
      END DO
      END IF
       IF(N0.EQ.12.OR.N0.EQ.13.OR.N0.EQ.14
.OR.N0.EQ.15) THEN
       DO I=10,13
        NLIN=NLIN+1
       IF(I.EQ.10) M=1
       IF(I.EQ.11) M=3
       IF(I.EQ.12) M=6
       IF(I.EQ.13) M=4
       LL(NLIN,1)=I+L
       LL(NLIN,2)=M+L
      END DO
       DO I=10,13
        NLIN=NLIN+1
       IF(I.EQ.10) M=3
       IF(I.EQ.11) M=6
       IF(I.EQ.12) M=4
       IF(I.EQ.13) M=1
       LL(NLIN,1)=I+L
       LL(NLIN,2)=M+L
      END DO
      END IF
      K=0
      DO 6 I=10,13
      DO 6 J=1,2
       K=K+1
       LP(NPOV+K,3)=I+L
    6  LP(NPOV+K,4)=LP(NPOV+K,3)
        NLIN=NLIN+1
       LL(NLIN,1)=9
       LL(NLIN,2)=6
      END IF
       IF(N0.EQ.1.OR.N0.EQ.3.OR.N0.EQ.6.OR
       LP(NPOV+I+4,1)=-(I-1)  +6+L
       LP(NPOV+I+4,2)=-(I-1)  +5+L
       LP(NPOV+I+6,1)= (I-1)*3+4+L
       IF(I.EQ.1) LP(NPOV+I+6,2)=7+L
       IF(I.EQ.2) LP(NPOV+I+6,2)=1+L
    7 CONTINUE
      NPOV=NPOV+8
      K=0
      DO 8 I=1,4
       K1=I
       IF(I.EQ.2) K1=3
       IF(I.EQ.3) K1=6
       IF(I.EQ.4) K1=4
      DO 8 J=1,2
       K=K+1
       LP(NPOV+K,3)=K1+L
    8  LP(NPOV+K,4)=LP(NPOV+K,3)
       K8=8
       K10=10
      DO 12 I=1,2
       LP(NPOV+I  ,1)=K10
       LP(NPOV+I  ,2)=K8
       IF(I.EQ.2) LP(NPOV+I  ,1)=K8+L
       IF(I.EQ.2) LP(NPOV+I  ,2)=K10+1+I+L
       LP(NPOV+I+2,1)= K8
       LP(NPOV+I+2,2)= K10
       IF(I.EQ.2) LP(NPOV+I+2,1)=K10+I-1+L
       IF(I.EQ.2) LP(NPOV+I+2,2)=K8+L
       LP(NPOV+I+4,1)= K10+2
       LP(NPOV+I+4,2)= K8
       IF(I.EQ.2) LP(NPOV+I+4,1)=K8+L
       IF(I.EQ.2) LP(NPOV+I+4,2)=K10+1+L
       LP(NPOV+I+6,1)= K8
       LP(NPOV+I+6,2)= K10+2
       IF(I.EQ.2) LP(NPOV+I+6,1)=K10+3+L
       IF(I.EQ.2) LP(NPOV+I+6,2)=K8+L
   12 CONTINUE
      NPOV=NPOV+8
       NLI1=NLIN-NLI1
       NPO1=NPOV-NPO1
      IF(NX.GT.1) THEN
CALL TACKEM
(NLT,X,Y,Z,L,L1,NX,AX,AY,0.0,XC,YC,ZC)
CALL PLOCEM
(NLT,LEN,LL,LP,L1,NX,NLIN,NPOV,NLI1,NPO1)
      ELSE
       L=13
       END IF
       X(L+1)=X(L-L1+3)
       Y(L+1)=Y(L-L1+3)
       Z(L+1)=Z(L-L1+3)
       GOTO 10
   11 CONTINUE
       IF(NZ.GT.1) THEN
       NLI1=NLIN
       NPO1=NPOV
       L2=L
CALL TACKEM
(NLT,X,Y,Z,L,L2,NZ,0.0,0.0,AZ,XC,YC,-1.)
CALL PLOCEM
(NLT,LEN,LL,LP,L2,NZ,NLIN,NPOV,NLI1,NPO1)
       END IF
CALL ISTTAC
(L,NLT,X,Y,Z,LG,LL,LP,NLIN,NPOV,LEN,AAX)
      IF(N1.EQ.1) THEN
CALL IZBTAC(NLT,X,Y,Z,L,A1,IDT,1)
CALL IZBELE(NLT,LEN,LL,LP,NLIN,NPOV,IDT)
       END IF
       IF(N2.EQ.1) THEN
```

```
      DO 7 I=1,2
        LP(NPOV+I  ,1)=  I+L
        LP(NPOV+I  ,2)=  I+1+L
        LP(NPOV+I+2,1)= (I-1)*6+3+L
        LP(NPOV+I+2,2)=-(I-1)*3+9+L
```

```
CALL IZBTAC(NLT,X,Y,Z,L,NX*AX-AX+A1,IDT,2)
CALL IZBELE(NLT,LEN,LL,LP,NLIN,NPOV,IDT)
        END IF
        RETURN
        END
```

```
      SUBROUTINE TALAS12
(NLT,LEN,X,Y,Z,LG,LL,LP,L,NLIN,NPOV,IDT)
        INCLUDE 'ZAJZON1M.FOR'
      CHARACTER*70  TEXT
      DIMENSION X(NLT),Y(NLT),Z(NLT)
        INTEGER*2 LG(NLT,LEN),LL(NLT,LEN),
LP(NLT,LEN),IDT(NLT)
        GOTO 9
   88 CALL GRESKA(1)
    9 CONTINUE
      L=0
        NLIN=0
        NPOV=0
      AAX=0.0
10 TEXT='AX,AY,AZ,
ANX,ANZ,AX1,AX2,AX3,AY1,AZ1,AZ2,AZ3'
      CALL ZAPIS(IRID,IKOL,TEXT,NN5)
      TEXT = 'Linije'
      CALL ZAPIS(IRID,IKOL,TEXT,NN5)
      IF(INP.EQ.5) THEN
       CALL READR(1)

READ(terk,'(12F10.0)',ERR=88)AX,AY,AZ,ANX,
ANZ,AX1,AX2,AX3,AY1,
        +AZ1,AZ2,AZ3
      WRITE(  8,'(12F10.3)')    AX,AY,AZ,
ANX,ANZ,AX1,AX2,AX3,AY1,AZ1,AZ2,AZ3
     IF(AX.EQ.0.0.AND.AY.EQ.0.0)    GOTO 11
       CALL READR(1)
READ(terk,'( 5F10.0)',ERR=88)  B,B1,B2
      WRITE(  8,'( 5F10.3,''; Talas
Podela'')')  B,B1,B2,B3,B4
        ELSE
READ(
INP,'(12F10.0)',ERR=88)AX,AY,AZ,ANX,ANZ,AX1
,AX2,AX3,AY1,AZ1,AZ2,AZ3
     IF(AX.EQ.0.0.AND.AY.EQ.0.0)    GOTO 11
READ( INP,'( 5F10.0)',ERR=88) B,B1,B2,B3,B4
        END IF
      IF(AAX.EQ.0.0) AAX=AX
          NX=ANX
          NZ=ANZ
          N0=B
        N1=B1
        N2=B2
        N3=B3
        N4=B4
          IF(AY.EQ.0.0) AY=AX/4.
          IF(AZ.EQ.0.0) AZ=AX
          IF(AX1.EQ.0.0) AX1=AX/2.
          IF(AX2.EQ.0.0) AX2=AX/4.
          IF(AX3.EQ.0.0) AX3=AX*3./4.
          IF(AY1.EQ.0.0) AY1=AY/2.
          IF(AZ1.EQ.0.0) AZ1=AZ/2.
          IF(AZ2.EQ.0.0) AZ2=AZ/4.
          IF(AZ3.EQ.0.0) AZ3=AZ/4.*3.
          IF(NX.EQ.0  ) NX=1
          IF(NZ.EQ.0  ) NZ=1
        IF(ZC.EQ.0.0) THEN
          X(2)=X(1)+AX
          Y(2)=Y(1)
          Z(2)=Z(1)
          X(3)=X(1)+AX
          Y(3)=Y(1)
          Z(3)=Z(1)+AZ
          X(4)=X(1)
```

```
          X(6)=X(1)+AX1
          Y(6)=Y(1)+AY1
          Z(6)=Z(1)+AZ2
          X(7)=X(1)+AX3
          Y(7)=Y(1)+AY1
          Z(7)=Z(1)+AZ1
          X(8)=X(1)+AX1
          Y(8)=Y(1)+AY1
          Z(8)=Z(1)+AZ3
          X(9)=X(1)+AX2
          Y(9)=Y(1)+AY1
          Z(9)=Z(1)+AZ1
        END IF

IF(ZC.EQ.1.0) THEN CALL UGAO
(XC,X(L+1),YC,Y(L+1),R0,F0,PISTRAD)
X(L+2)=(R0+AY2)*COS(F0+AX2*PISTRAD)+XC
Y(L+2)=(R0+AY2)*SIN(F0+AX2*PISTRAD)+YC
X(L+8)=(R0+AY1)*COS(F0+AX1*PISTRAD)+XC
Y(L+8)=(R0+AY1)*SIN(F0+AX1*PISTRAD)+YC
        END IF
      L1=9
        NLI1=NLIN
        NPO1=NPOV
        IF(N0.EQ.1.OR.N0.EQ.3.OR.N0.EQ.5.OR
.N0.EQ.7)THEN
         NLIN=NLIN+1
       LL(NLIN,1)=1+L
       LL(NLIN,2)=2+L
         NLIN=NLIN+1
       LL(NLIN,1)=3+L
       LL(NLIN,2)=4+L
      END IF
        IF(N0.EQ.2.OR.N0.EQ.3.OR.N0.EQ.6.OR
.N0.EQ.7)THEN
         NLIN=NLIN+1
       LL(NLIN,1)=1+L
       LL(NLIN,2)=4+L
         NLIN=NLIN+1
       LL(NLIN,1)=2+L
       LL(NLIN,2)=3+L
      END IF
        IF(N0.EQ.4.OR.N0.EQ.5.OR.N0.EQ.6.OR
.N0.EQ.7) THEN
      DO I=6,9
         NLIN=NLIN+1
       LL(NLIN,1)=5+L
       LL(NLIN,2)=I+L
      END DO
      DO I=6,9
         NLIN=NLIN+1
       LL(NLIN,1)=I+L
       LL(NLIN,2)=I+1+L
       IF(I.EQ.9) THEN
       LL(NLIN,2)=6+L
       END IF
       END DO
       DO I=1,4
         NLIN=NLIN+1
       LL(NLIN,1)=I+L
       LL(NLIN,2)=I+5+L
       END DO
       DO I=1,4
         NLIN=NLIN+1
       LL(NLIN,1)=I+L
       LL(NLIN,2)=I+4+L
```

```
          Y(4)=Y(1)
          Z(4)=Z(1)+AZ
          X(5)=X(1)+AX1
          Y(5)=Y(1)+AY
          Z(5)=Z(1)+AZ1
         K=0
      DO 6 I=1,4
       K=K+1
       LP(K,1)=I+L
       LP(K,2)=I+1+L
       LP(K,3)=I+5+L
       LP(K,4)=I+5+L
      IF(I.EQ.4) LP(K,2)=1+L
    6 CONTINUE
      DO 7 I=1,4
       K=K+1
       LP(K,1)=I+L
       LP(K,2)=I+5+L
       LP(K,3)=I+4+L
       LP(K,4)=I+4+L
       IF(I.EQ.1.) THEN
      LP(K,3)=9+L
      LP(K,4)=9+L
      END IF
    7 CONTINUE
      DO 12 I=6,9
       K=K+1
       LP(K,1)=I+L
       LP(K,2)=I+1+L
       LP(K,3)=5+L
       LP(K,4)=5+L
      IF(I.EQ.9) LP(K,2)=6+L
   12 CONTINUE

      NPOV=NPOV+12
       NLI1=NLIN-NLI1
       NPO1=NPOV-NPO1
      IF(NX.GT.1) THEN
        IF(I.EQ.1) THEN
        LL(NLIN,2)=9+L
        END IF
        END DO
       END IF
CALL TACKEM
(NLT,X,Y,Z,L,L1,NX,AX,AY,0.0,XC,YC,ZC)
CALL PLOCEM
(NLT,LEN,LL,LP,L1,NX,NLIN,NPOV,NLI1,NPO1)
       L=9
        END IF
        X(L+1)=X(L-L1+3)
        Y(L+1)=Y(L-L1+3)
        Z(L+1)=Z(L-L1+3)
        GOTO 10
   11 CONTINUE
        IF(NZ.GT.1) THEN
        NLI1=NLIN
        NPO1=NPOV
        L2=L
CALL TACKEM
(NLT,X,Y,Z,L,L2,NZ,0.0,0.0,AZ,XC,YC,-1.)
CALL PLOCEM
(NLT,LEN,LL,LP,L2,NZ,NLIN,NPOV,NLI1,NPO1)
        END IF
CALL ISTTAC
(L,NLT,X,Y,Z,LG,LL,LP,NLIN,NPOV,LEN,AAX)
       IF(N1.EQ.1) THEN
        CALL IZBTAC(NLT,X,Y,Z,L,A1,IDT,1)
CALL IZBELE(NLT,LEN,LL,LP,NLIN,NPOV,IDT)
        END IF
        IF(N2.EQ.1) THEN
CALL IZBTAC(NLT,X,Y,Z,L,NX*AX-AX+A1,IDT,2)
CALL IZBELE(NLT,LEN,LL,LP,NLIN,NPOV,IDT)
        END IF
        RETURN
        END
```

**SUBROUTINE TALAS13**

```
(NLT,LEN,X,Y,Z,LG,LL,LP,L,NLIN,NPOV,IDT)
       INCLUDE 'ZAJZON1M.FOR'
      CHARACTER*70  TEXT
      DIMENSION X(NLT),Y(NLT),Z(NLT)
       INTEGER*2 LG(NLT,LEN),LL(NLT,LEN),
LP(NLT,LEN),IDT(NLT)
       GOTO 9
   88 CALL GRESKA(1)
    9 CONTINUE
      L=0
       NLIN=0
       NPOV=0
      AAX=0.0
   10 TEXT =
'AX,AY,AZ,ANX,ANZ,AX1,AX2,AX3,AY1,AY2,AY3,A
Z1,AZ2,AZ3'
      CALL ZAPIS(IRID,IKOL,TEXT,NN5)
      TEXT = 'Linije'
      CALL ZAPIS(IRID,IKOL,TEXT,NN5)
      IF(INP.EQ.5) THEN
       CALL READR(1)
      READ(terk,'(14F10.0)',ERR=88)
AX,AY,AZ,ANX,ANZ,AX1,AX2,AX3,AY1,
       +AY2,AY3,AZ1,AZ2,AZ3
      WRITE(   8,
     +'(14F10.3                   )')
AX,AY,AZ,ANX,ANZ,AX1,AX2,AX3,AY1,
     +AY2,AY3,AZ1,AZ2,AZ3
     IF(AX.EQ.0.0.AND.AY.EQ.0.0)    GOTO 11
       CALL READR(1)
READ(terk,'( 5F10.0)',ERR=88)   B,B1,B2
     IF(AX.EQ.0.0.AND.AY.EQ.0.0)    GOTO 11
      READ( INP,'( 5F10.0)',ERR=88)
B,B1,B2,B3,B4
        END IF
        IF(AAX.EQ.0.0) AAX=AX
          NX=ANX
          NZ=ANZ
          N0=B
        N1=B1
        N2=B2
        N3=B3
        N4=B4
          IF(AY.EQ.0.0) AY=AX/4.
          IF(AZ.EQ.0.0) AZ=AX
          IF(AX1.EQ.0.0) AX1=AX/2.
          IF(AX2.EQ.0.0) AX2=AX/4.
          IF(AX3.EQ.0.0) AX3=AX*3./4.
          IF(AY1.EQ.0.0) AY1=AY/2.
          IF(AY2.EQ.0.0) AY2=AY/2.
          IF(AY3.EQ.0.0) AY3=AY/4.
          IF(AZ1.EQ.0.0) AZ1=AZ/2.
          IF(AZ2.EQ.0.0) AZ2=AZ/4.
          IF(AZ3.EQ.0.0) AZ3=AZ/4.*3.
          IF(NX.EQ.0  ) NX=1
          IF(NZ.EQ.0  ) NZ=1
       IF(ZC.EQ.0.0) THEN
          X(2)=X(1)+AX1
          Y(2)=Y(1)+AY1
          Z(2)=Z(1)
          X(3)=X(1)+AX
          Y(3)=Y(1)
          Z(3)=Z(1)
          X(4)=X(1)+AX
```

```
WRITE(   8,'( 5F10.3,''; Talas Podela'')')
B,B1,B2,B3,B4
       ELSE
      READ( INP,'(14F10.0)',ERR=88)
AX,AY,AZ,ANX,ANZ,AX1,AX2,AX3,AY1,
       +AY2,AY3,AZ1,AZ2,AZ3
         X(6)=X(1)+AX1
         Y(6)=Y(1)+AY1
         Z(6)=Z(1)+AZ
         X(7)=X(1)
         Y(7)=Y(1)
         Z(7)=Z(1)+AZ
         X(8)=X(1)
         Y(8)=Y(1)+AY2
         Z(8)=Z(1)+AZ1
         X(9)=X(1)+AX1
         Y(9)=Y(1)+AY
         Z(9)=Z(1)+AZ1
         X(10)=X(1)+AX1
         Y(10)=Y(1)+AY3
         Z(10)=Z(1)+AZ2
         X(11)=X(1)+AX3
         Y(11)=Y(1)+AY3
         Z(11)=Z(1)+AZ1
         X(12)=X(1)+AX1
         Y(12)=Y(1)+AY3
         Z(12)=Z(1)+AZ3
         X(13)=X(1)+AX2
         Y(13)=Y(1)+AY3
         Z(13)=Z(1)+AZ1
       END IF
      L1=13
       NLI1=NLIN
       NPO1=NPOV
IF(N0.EQ.1.OR.N0.EQ.3.OR.N0.EQ.5.OR.N0.EQ.7
.OR.N0.EQ.9.OR.N0.EQ.11)THEN
        NLIN=NLIN+1
       LL(NLIN,1)=1+L
       LL(NLIN,2)=2+L
        NLIN=NLIN+1
       LL(NLIN,1)=2+L
       LL(NLIN,2)=3+L
        NLIN=NLIN+1
       LL(NLIN,1)=5+L
       LL(NLIN,2)=6+L
        NLIN=NLIN+1
       LL(NLIN,1)=6+L
       LL(NLIN,2)=7+L
      END IF
IF(N0.EQ.2.OR.N0.EQ.3.OR.N0.EQ.6.OR.N0.EQ.7
.OR.N0.EQ.10.OR.N0.EQ.11)THEN
        NLIN=NLIN+1
       LL(NLIN,1)=3+L
       LL(NLIN,2)=4+L
        NLIN=NLIN+1
       LL(NLIN,1)=4+L
       LL(NLIN,2)=5+L
        NLIN=NLIN+1
       LL(NLIN,1)=7+L
       LL(NLIN,2)=8+L
        NLIN=NLIN+1
       LL(NLIN,1)=8+L
       LL(NLIN,2)=1+L
      END IF
IF(N0.EQ.8.OR.N0.EQ.9.OR.N0.EQ.10.OR.N0.EQ.
11) THEN
      DO I=10,13
        NLIN=NLIN+1
       LL(NLIN,1)=I+L
       LL(NLIN,2)=I+1+L
       IF(I.EQ.13) THEN
       LL(NLIN,2)=10+L
       END IF
       END DO
         Y(4)=Y(1)+AY2
         Z(4)=Z(1)+AZ1
         X(5)=X(1)+AX
         Y(5)=Y(1)
         Z(5)=Z(1)+AZ
       END DO
        M=9
      DO I=1,7,2
        NLIN=NLIN+1
       M=M+1
         LL(NLIN,1)=I+2+L
       LL(NLIN,2)=M+L
        IF(I.EQ.7) LL(NLIN,1)=1+L
       END DO
      END IF
        IF(N0.EQ.4.OR.N0.EQ.5.OR.N0.EQ.6.OR
.N0.EQ.7.OR.N0.EQ.8.OR.N0.EQ.9
        +.OR.N0.EQ.10.OR.N0.EQ.11) THEN
      DO I=10,13
        NLIN=NLIN+1
       LL(NLIN,1)=9+L
       LL(NLIN,2)=I+L
      END DO
        M=9
      DO I=2,8,2
        NLIN=NLIN+1
       M=M+1
         LL(NLIN,1)=I+L
       LL(NLIN,2)=M+L
       END DO
      END IF
         K=0
        M=9
      DO 6 I=1,7,2
        K=K+1
       M=M+1
        LP(K,1)=I+L
        LP(K,2)=I+1+L
        LP(K,3)=M+L
        LP(K,4)=LP(K,3)
    6 CONTINUE
        M=9
      DO 7 I=2,8,2
        K=K+1
       M=M+1
        LP(K,1)=I+L
        LP(K,2)=I+1+L
        LP(K,3)=M+L
        LP(K,4)=LP(K,3)
        IF(I.EQ.8) THEN
       LP(K,2)=1+L
       END IF
    7 CONTINUE
      DO 13 I=10,13
        K=K+1
        LP(K,1)=I+L
        LP(K,2)=I+1+L
        LP(K,3)=9+L
        LP(K,4)=LP(K,3)
      IF(I.EQ.13) LP(K,2)=10+L
   13 CONTINUE
      M=1
      DO 12 I=10,13
        K=K+1
      M=M+2
        LP(K,1)=I+L
        LP(K,2)=M+L
        LP(K,3)=I+1+L
      IF(I.EQ.13) LP(K,2)=1+L
      IF(I.EQ.13) LP(K,3)=10+L
        LP(K,4)=LP(K,3)
   12 CONTINUE
      NPOV=NPOV+16
        NLI1=NLIN-NLI1
```

```
        M=9
       DO I=1,7,2
         NLIN=NLIN+1
        M=M+1
         LL(NLIN,1)=I+L
        LL(NLIN,2)=M+L
CALL PLOCEM
(NLT,LEN,LL,LP,L1,NX,NLIN,NPOV,NLI1,NPO1)
       ELSE
        L=13
        END IF
        X(L+1)=X(L-L1+3)
        Y(L+1)=Y(L-L1+3)
        Z(L+1)=Z(L-L1+3)
        GOTO 10
    11 CONTINUE
        IF(NZ.GT.1) THEN
        NLI1=NLIN
        NPO1=NPOV
        L2=L
CALL TACKEM
        NPO1=NPOV-NPO1

      IF(NX.GT.1) THEN
CALL TACKEM
(NLT,X,Y,Z,L,L1,NX,AX,AY,0.0,XC,YC,ZC)
(NLT,X,Y,Z,L,L2,NZ,0.0,0.0,AZ,XC,YC,-1.)
CALL PLOCEM
(NLT,LEN,LL,LP,L2,NZ,NLIN,NPOV,NLI1,NPO1)
        END IF
CALL ISTTAC
(L,NLT,X,Y,Z,LG,LL,LP,NLIN,NPOV,LEN,AAX)
      IF(N1.EQ.1) THEN
CALL IZBTAC(NLT,X,Y,Z,L,A1,IDT,1)
CALL IZBELE(NLT,LEN,LL,LP,NLIN,NPOV,IDT)
        END IF
        IF(N2.EQ.1) THEN
CALL IZBTAC(NLT,X,Y,Z,L,NX*AX-AX+A1,IDT,2)
CALL IZBELE(NLT,LEN,LL,LP,NLIN,NPOV,IDT)
        END IF
        RETURN
        END
```

```
       SUBROUTINE TALAS14
(NLT,LEN,X,Y,Z,LG,LL,LP,L,NLIN,NPOV,IDT)
        INCLUDE 'ZAJZON1M.FOR'
       CHARACTER*70  TEXT
       DIMENSION X(NLT),Y(NLT),Z(NLT)
        INTEGER*2 LG(NLT,LEN),LL(NLT,LEN),
LP(NLT,LEN),IDT(NLT)
        GOTO 9
    88 CALL GRESKA(1)
     9 CONTINUE
        L=0
        NLIN=0
        NPOV=0
       AAX=0.0
    10 TEXT =
'AX,AY,AZ,ANX,ANZ,AX1,AY1,AY2,AZ1'
       CALL ZAPIS(IRID,IKOL,TEXT,NN5)
       TEXT = 'Linije'
       CALL ZAPIS(IRID,IKOL,TEXT,NN5)
       IF(INP.EQ.5) THEN
        CALL READR(1)
       READ(terk,'(9F10.0)',ERR=88)
AX,AY,AZ,ANX,ANZ,AX1,AY1,AY2,AZ1
       WRITE(   8,'(9F10.3
)') AX,AY,AZ,ANX,ANZ,AX1,AY1,AY2,AZ1
      IF(AX.EQ.0.0.AND.AY.EQ.0.0)    GOTO 11
        CALL READR(1)
 READ(terk,'( 5F10.0)',ERR=88)   B,B1,B2
 WRITE(   8,'( 5F10.3,''; Talas Podela'')')
B,B1,B2,B3,B4
        ELSE
       READ( INP,'(9F10.0)',ERR=88)
AX,AY,AZ,ANX,ANZ,AX1,AY1,AY2,AZ1
      IF(AX.EQ.0.0.AND.AY.EQ.0.0)    GOTO 11
       READ( INP,'( 5F10.0)',ERR=88)
B,B1,B2,B3,B4
        END IF
        IF(AAX.EQ.0.0) AAX=AX
          NX=ANX
          NZ=ANZ
          N0=B
        N1=B1
        N2=B2
        N3=B3
        N4=B4
          IF(AY.EQ.0.0) AY=AX/4.
          IF(AZ.EQ.0.0) AZ=AX
          IF(AX1.EQ.0.0) AX1=AX/2.
          IF(AY1.EQ.0.0) AY1=AY/2.
          IF(AY2.EQ.0.0) AY2=AY/2.
          Y(2)=Y(1)+AY1
          Z(2)=Z(1)
          X(3)=X(1)+AX
          Y(3)=Y(1)
          Z(3)=Z(1)
          X(4)=X(1)+AX
          Y(4)=Y(1)+AY2
          Z(4)=Z(1)+AZ1
          X(5)=X(1)+AX
          Y(5)=Y(1)
          Z(5)=Z(1)+AZ
          X(6)=X(1)+AX1
          Y(6)=Y(1)+AY1
          Z(6)=Z(1)+AZ
          X(7)=X(1)
          Y(7)=Y(1)
          Z(7)=Z(1)+AZ
          X(8)=X(1)
          Y(8)=Y(1)+AY2
          Z(8)=Z(1)+AZ1
          X(9)=X(1)+AX1
          Y(9)=Y(1)+AY
          Z(9)=Z(1)+AZ1
          X(10)=(X(2)+X(8))/2.
          Y(10)=(Y(2)+Y(8))/2.
          Z(10)=(Z(2)+Z(8))/2.
          X(11)=(X(2)+X(4))/2.
          Y(11)=(Y(2)+Y(4))/2.
          Z(11)=(Z(2)+Z(4))/2.
          X(12)=(X(4)+X(6))/2.
          Y(12)=(Y(4)+Y(6))/2.
          Z(12)=(Z(4)+Z(6))/2.
          X(13)=(X(6)+X(8))/2.
          Y(13)=(Y(6)+Y(8))/2.
          Z(13)=(Z(6)+Z(8))/2.
        END IF
      L1=13
        NLI1=NLIN
        NPO1=NPOV
        IF(N0.EQ.1.OR.N0.EQ.3.OR.N0.EQ.5.OR
.N0.EQ.7.OR.N0.EQ.9.OR.N0.EQ.11)THEN
         NLIN=NLIN+1
       LL(NLIN,1)=1+L
       LL(NLIN,2)=2+L
         NLIN=NLIN+1
       LL(NLIN,1)=2+L
       LL(NLIN,2)=3+L
       NLIN=NLIN+1
       LL(NLIN,1)=5+L
       LL(NLIN,2)=6+L
```

```
          IF(AZ1.EQ.0.0) AZ1=AZ/2.
          IF(NX.EQ.0  ) NX=1
          IF(NZ.EQ.0  ) NZ=1
      IF(ZC.EQ.0.0) THEN
          X(2)=X(1)+AX1
.N0.EQ.7.OR.N0.EQ.10.OR.N0.EQ.11)THEN
        NLIN=NLIN+1
       LL(NLIN,1)=1+L
       LL(NLIN,2)=8+L
        NLIN=NLIN+1
       LL(NLIN,1)=8+L
       LL(NLIN,2)=7+L
       NLIN=NLIN+1
       LL(NLIN,1)=3+L
       LL(NLIN,2)=4+L
        NLIN=NLIN+1
       LL(NLIN,1)=4+L
       LL(NLIN,2)=5+L
      END IF
        IF(N0.EQ.4.OR.N0.EQ.5.OR.N0.EQ.6.OR
.N0.EQ.7.OR.N0.EQ.8.OR.N0.EQ.9.OR.N0.EQ.10.
OR.N0.EQ.11)THEN
        NLIN=NLIN+1
       LL(NLIN,1)=2+L
       LL(NLIN,2)=11+L
        NLIN=NLIN+1
       LL(NLIN,1)=11+L
       LL(NLIN,2)=4+L
       NLIN=NLIN+1
       LL(NLIN,1)=4+L
       LL(NLIN,2)=12+L
        NLIN=NLIN+1
       LL(NLIN,1)=12+L
       LL(NLIN,2)=6+L
        NLIN=NLIN+1
       LL(NLIN,1)=6+L
       LL(NLIN,2)=13+L
        NLIN=NLIN+1
       LL(NLIN,1)=13+L
       LL(NLIN,2)=8+L
       NLIN=NLIN+1
       LL(NLIN,1)=8+L
       LL(NLIN,2)=10+L
        NLIN=NLIN+1
       LL(NLIN,1)=10+L
       LL(NLIN,2)=2+L
        END IF
IF(N0.EQ.8.OR.N0.EQ.9.OR.N0.EQ.10.OR.N0.EQ.
11)THEN
      DO I=10,13
        NLIN=NLIN+1
       LL(NLIN,1)=9+L
       LL(NLIN,2)=I+L
      END DO
      DO I=10,13
        NLIN=NLIN+1
       LL(NLIN,1)=I+L
       LL(NLIN,2)=I+1+L
       IF(I.EQ.13) LL(NLIN,2)=10+L
       END DO
      END IF
        K=0
      DO 6 I=1,7,2
       K=K+1
       LP(K,1)=I+L
        NLIN=NLIN+1
       LL(NLIN,1)=6+L
       LL(NLIN,2)=7+L
      END IF
       IF(N0.EQ.2.OR.N0.EQ.3.OR.N0.EQ.6.OR
       LP(K,2)=I+1+L
       LP(K,3)=K+9+L
       LP(K,4)=I-1+L
      IF(I.EQ.1) THEN
      LP(K,4)=8+L
      END IF
    6 CONTINUE
      DO 7 I=2,8,2
       K=K+1
       LP(K,1)=I+L
       LP(K,2)=K+6+L
       LP(K,3)=K+5+L
       LP(K,4)=K+5+L
      IF(I.EQ.8) LP(K,2)=10+L
    7 CONTINUE
      DO 12 I=10,13
       K=K+1
       LP(K,1)=I+L
       LP(K,2)=I+1+L
       LP(K,3)=9+L
       LP(K,4)=9+L
      IF(I.EQ.13) LP(K,2)=10+L
   12 CONTINUE

      NPOV=NPOV+13
       NLI1=NLIN-NLI1
       NPO1=NPOV-NPO1
      IF(NX.GT.1) THEN
CALL TACKEM
(NLT,X,Y,Z,L,L1,NX,AX,AY,0.0,XC,YC,ZC)
CALL PLOCEM
(NLT,LEN,LL,LP,L1,NX,NLIN,NPOV,NLI1,NPO1)
      ELSE
       L=13
       END IF
       X(L+1)=X(L-L1+3)
       Y(L+1)=Y(L-L1+3)
       Z(L+1)=Z(L-L1+3)
       GOTO 10
   11 CONTINUE
       IF(NZ.GT.1) THEN
       NLI1=NLIN
       NPO1=NPOV
       L2=L
CALL TACKEM
(NLT,X,Y,Z,L,L2,NZ,0.0,0.0,AZ,XC,YC,-1.)
CALL PLOCEM
(NLT,LEN,LL,LP,L2,NZ,NLIN,NPOV,NLI1,NPO1)
       END IF
CALL ISTTAC
(L,NLT,X,Y,Z,LG,LL,LP,NLIN,NPOV,LEN,AAX)
      IF(N1.EQ.1) THEN
       CALL IZBTAC(NLT,X,Y,Z,L,A1,IDT,1)
CALL IZBELE(NLT,LEN,LL,LP,NLIN,NPOV,IDT)
       END IF
       IF(N2.EQ.1) THEN
CALL IZBTAC(NLT,X,Y,Z,L,NX*AX-AX+A1,IDT,2)
CALL IZBELE(NLT,LEN,LL,LP,NLIN,NPOV,IDT)
       END IF
       RETURN
       END
```

---

```
      SUBROUTINE TALAS15
(NLT,LEN,X,Y,Z,LG,LL,LP,L,NLIN,NPOV,IDT)
       INCLUDE 'ZAJZON1M.FOR'
   88 CALL GRESKA(1)
    9 CONTINUE
      L=0
```

```
      CHARACTER*70  TEXT
      DIMENSION X(NLT),Y(NLT),Z(NLT)
       INTEGER*2 LG(NLT,LEN),LL(NLT,LEN),
LP(NLT,LEN),IDT(NLT)
       GOTO 9
      CALL ZAPIS(IRID,IKOL,TEXT,NN5)
      TEXT = 'Linije'
      CALL ZAPIS(IRID,IKOL,TEXT,NN5)
      IF(INP.EQ.5) THEN
       CALL READR(1)
      READ(terk,'(10F10.0)',ERR=88)
AX,AY,AZ,ANX,ANZ,AX1,AY1,AY2,AY3,AZ1
      WRITE(   8,
     +'(10F10.3                    )')
AX,AY,AZ,ANX,ANZ,AX1,AY1,AY2,AY3,AZ1
     IF(AX.EQ.0.0.AND.AY.EQ.0.0)    GOTO 11
      CALL READR(1)
  READ(terk,'( 5F10.0)',ERR=88)   B,B1,B2
      WRITE(   8,'( 5F10.3,''; Talas
Podela'')') B,B1,B2,B3,B4
       ELSE
      READ( INP,'(10F10.0)',ERR=88)
AX,AY,AZ,ANX,ANZ,AX1,AY1,AY2,AY3,AZ1
    IF(AX.EQ.0.0.AND.AY.EQ.0.0)    GOTO 11
      READ( INP,'( 5F10.0)',ERR=88)
B,B1,B2,B3,B4
       END IF
      IF(AAX.EQ.0.0) AAX=AX
         NX=ANX
         NZ=ANZ
         N0=B
       N1=B1
       N2=B2
       N3=B3
       N4=B4
         IF(AY.EQ.0.0) AY=AX/4.
         IF(AZ.EQ.0.0) AZ=AX
         IF(AX1.EQ.0.0) AX1=AX/2.
         IF(AY1.EQ.0.0) AY1=AY/2.
         IF(AY2.EQ.0.0) AY2=AY/2.
         IF(AY3.EQ.0.0) AY3=AY/4.
         IF(AZ1.EQ.0.0) AZ1=AZ/2.
         IF(NX.EQ.0  ) NX=1
         IF(NZ.EQ.0  ) NZ=1
      IF(ZC.EQ.0.0) THEN
         X(2)=X(1)+AX1
         Y(2)=Y(1)+AY1
         Z(2)=Z(1)
         X(3)=X(1)+AX
         Y(3)=Y(1)
         Z(3)=Z(1)
         X(4)=X(1)+AX
         Y(4)=Y(1)+AY2
         Z(4)=Z(1)+AZ1
         X(5)=X(1)+AX
         Y(5)=Y(1)
         Z(5)=Z(1)+AZ
         X(6)=X(1)+AX1
         Y(6)=Y(1)+AY1
         Z(6)=Z(1)+AZ
         X(7)=X(1)
         Y(7)=Y(1)
         Z(7)=Z(1)+AZ
         X(8)=X(1)
         Y(8)=Y(1)+AY2
         Z(8)=Z(1)+AZ1
         X(9)=X(1)+AX1
         Y(9)=Y(1)+AY
         Z(9)=Z(1)+AZ1
         X(10)=(X(2)+X(8))/2.
         Y(10)=AY3
         Z(10)=(Z(2)+Z(8))/2.
         X(11)=(X(2)+X(4))/2.
         Y(11)=AY3
```

```
       NLIN=0
       NPOV=0
      AAX=0.0
   10 TEXT =
'AX,AY,AZ,ANX,ANZ,AX1,AY1,AY2,AY3,AZ1'
         Y(13)=AY3
         Z(13)=(Z(6)+Z(8))/2.
      END IF
      L1=13
       NLI1=NLIN
       NPO1=NPOV
       IF(N0.EQ.1.OR.N0.EQ.3.OR.N0.EQ.5.OR
.N0.EQ.7.OR.N0.EQ.9.OR.N0.EQ.11.OR.N0.EQ.13
.OR.N0.EQ.15.OR.N0.EQ.17.OR.N0.EQ.19)THEN
        NLIN=NLIN+1
       LL(NLIN,1)=1+L
       LL(NLIN,2)=2+L
        NLIN=NLIN+1
       LL(NLIN,1)=2+L
       LL(NLIN,2)=3+L
       NLIN=NLIN+1
       LL(NLIN,1)=5+L
       LL(NLIN,2)=6+L
        NLIN=NLIN+1
       LL(NLIN,1)=6+L
       LL(NLIN,2)=7+L
      END IF
       IF(N0.EQ.2.OR.N0.EQ.3.OR.N0.EQ.6.OR
.N0.EQ.7.OR.N0.EQ.10.OR.N0.EQ.11.OR.N0.EQ.1
4.OR.N0.EQ.15.OR.N0.EQ.18.OR.N0.EQ.19)THEN
        NLIN=NLIN+1
       LL(NLIN,1)=1+L
       LL(NLIN,2)=8+L
        NLIN=NLIN+1
       LL(NLIN,1)=8+L
       LL(NLIN,2)=7+L
       NLIN=NLIN+1
       LL(NLIN,1)=3+L
       LL(NLIN,2)=4+L
        NLIN=NLIN+1
       LL(NLIN,1)=4+L
       LL(NLIN,2)=5+L
      END IF
       IF(N0.EQ.8.OR.N0.EQ.9.OR.N0.EQ.10.O
R.N0.EQ.11.OR.N0.EQ.12.OR.N0.EQ.13.OR.N0.EQ
.14.OR.N0.EQ.15.OR.N0.EQ.16.OR.N0.EQ.17.
       +OR.N0.EQ.18.OR.N0.EQ.19)THEN
        NLIN=NLIN+1
       LL(NLIN,1)=2+L
       LL(NLIN,2)=11+L
        NLIN=NLIN+1
       LL(NLIN,1)=11+L
       LL(NLIN,2)=4+L
       NLIN=NLIN+1
       LL(NLIN,1)=4+L
       LL(NLIN,2)=12+L
        NLIN=NLIN+1
       LL(NLIN,1)=12+L
       LL(NLIN,2)=6+L
        NLIN=NLIN+1
       LL(NLIN,1)=6+L
       LL(NLIN,2)=13+L
        NLIN=NLIN+1
       LL(NLIN,1)=13+L
       LL(NLIN,2)=8+L
       NLIN=NLIN+1
       LL(NLIN,1)=8+L
       LL(NLIN,2)=10+L
        NLIN=NLIN+1
       LL(NLIN,1)=10+L
       LL(NLIN,2)=2+L
       end if
IF(N0.EQ.4.OR.N0.EQ.5.OR.N0.EQ.6.OR.N0.EQ.7
.OR.N0.EQ.12.OR.N0.EQ.13.OR.N0.EQ.14.OR.N0.
```

```
          Z(11)=(Z(2)+Z(4))/2.
          X(12)=(X(4)+X(6))/2.
          Y(12)=AY3
          Z(12)=(Z(4)+Z(6))/2.
          X(13)=(X(6)+X(8))/2.
       LL(NLIN,2)=I+L
      END DO
      NLIN=NLIN+1
       LL(NLIN,1)=1+L
       LL(NLIN,2)=10+L
        NLIN=NLIN+1
       LL(NLIN,1)=12+L
       LL(NLIN,2)=5+L
       NLIN=NLIN+1
       LL(NLIN,1)=3+L
       LL(NLIN,2)=11+L
        NLIN=NLIN+1
       LL(NLIN,1)=13+L
       LL(NLIN,2)=7+L
      END IF
IF(N0.EQ.16.OR.N0.EQ.17.OR.N0.EQ.18.OR.N0.E
Q.19) THEN
        NLIN=NLIN+1
       LL(NLIN,1)=10+L
       LL(NLIN,2)=11+L
        NLIN=NLIN+1
       LL(NLIN,1)=11+L
       LL(NLIN,2)=12+L
       NLIN=NLIN+1
       LL(NLIN,1)=12+L
       LL(NLIN,2)=13+L
        NLIN=NLIN+1
       LL(NLIN,1)=13+L
       LL(NLIN,2)=10+L
        end if
        K=0
      DO 6 I=1,7,2
        K=K+1
        LP(K,1)=I+L
        LP(K,2)=I+1+L
        LP(K,3)=K+9+L
        LP(K,4)=K+9+L
    6 CONTINUE
      DO 7 I=2,8,2
        K=K+1
        LP(K,1)=I+L
        LP(K,2)=I+1+L
        LP(K,3)=K+6+L
        LP(K,4)=K+6+L
      IF(I.EQ.8) THEN
        LP(K,2)=1+L
      LP(K,3)=10+L
      LP(K,4)=10+L
       END IF
    7 CONTINUE
      DO 12 I=2,8,2
        K=K+1
EQ.15.OR.N0.EQ.16.OR.N0.EQ.17.OR.N0.EQ.18.O
R.N0.EQ.19)THEN
      DO I=10,13
        NLIN=NLIN+1
       LL(NLIN,1)=9+L
        LP(K,1)=I+L
        LP(K,2)=K+2+L
        LP(K,3)=K+1+L
        LP(K,4)=K+1+L
      IF(I.EQ.8) LP(K,2)=10+L
   12 CONTINUE
      DO 13 I=10,13
        K=K+1
        LP(K,1)=I+L
        LP(K,2)=I+1+L
        LP(K,3)=9+L
        LP(K,4)=9+L
      IF(I.EQ.13) LP(K,2)=10+L
   13 CONTINUE
       NPOV=NPOV+16
        NLI1=NLIN-NLI1
        NPO1=NPOV-NPO1

      IF(NX.GT.1) THEN
        CALL TACKEM
(NLT,X,Y,Z,L,L1,NX,AX,AY,0.0,XC,YC,ZC)
        CALL PLOCEM
(NLT,LEN,LL,LP,L1,NX,NLIN,NPOV,NLI1,NPO1)
        ELSE
        L=13
        END IF
        X(L+1)=X(L-L1+3)
        Y(L+1)=Y(L-L1+3)
        Z(L+1)=Z(L-L1+3)
        GOTO 10
    11 CONTINUE
        IF(NZ.GT.1) THEN
        NLI1=NLIN
        NPO1=NPOV
        L2=L
        CALL TACKEM
(NLT,X,Y,Z,L,L2,NZ,0.0,0.0,AZ,XC,YC,-1.)
        CALL PLOCEM
(NLT,LEN,LL,LP,L2,NZ,NLIN,NPOV,NLI1,NPO1)
        END IF

        CALL ISTTAC
(L,NLT,X,Y,Z,LG,LL,LP,NLIN,NPOV,LEN,AAX)
      IF(N1.EQ.1) THEN
        CALL IZBTAC(NLT,X,Y,Z,L,A1,IDT,1)
CALL IZBELE(NLT,LEN,LL,LP,NLIN,NPOV,IDT)
        END IF
        IF(N2.EQ.1) THEN
CALL IZBTAC(NLT,X,Y,Z,L,NX*AX-AX+A1,IDT,2)
        CALL
IZBELE(NLT,LEN,LL,LP,NLIN,NPOV,IDT)
        END IF
        RETURN
        END
```

```
c       ime datoteke : ZAJRUT.FOR

        SUBROUTINE SAVUNS (MOD,ND)
      INCLUDE 'ZAJZONM.FOR'
              IF(MOD.EQ. 1) GOTO 12
      OPEN  (ND, FILE='\KOMIPS\ZAZ.DAT',
        +
STATUS='OLD',FORM='UNFORMATTED',ERR=11)
      READ (ND,err=10)
NPAR,NUMNP,MBAND,NELTYP,N1,N2,N3,N4,N5,MTOT
,NEQ
      READ (ND,err=10)
NQQ,NUI,ULIZL,HED,IMEDIR
      READ (ND,err=10)
MOD1,MOD2,MOD3,N10SV,KEQB,NUMEL,TL
      READ (ND,err=10)
NBLOCK,NEQB,NANAL,NOS,IDUM,NEIG,NAD,ANORM,N
VV,NFO
        +                 ,SDTOL
      READ (ND,err=10) NDUM,DUM,MDUM,NDYN,
        +
SKAL,IFSS,NITEM,RTOL,COFQ,IFPR,NFF,SDTOL
      READ (ND,err=10) L20,L25,STR
      READ (ND,err=10) NGVA,GVA,IOS,XYZOS
   10 CLOSE (ND)
   11 RETURN
   12 IZAP=0
      OPEN  (ND, FILE='\KOMIPS\ZAZ.DAT',
        +
STATUS='UNKNOWN',FORM='UNFORMATTED')
      WRITE (ND)
NPAR,NUMNP,MBAND,NELTYP,N1,N2,N3,N4,N5,MTOT
,NEQ
      WRITE (ND) NQQ,NUI,ULIZL,HED,IMEDIR
      WRITE (ND)
MOD1,MOD2,MOD3,N10SV,KEQB,NUMEL,TL
      WRITE (ND)
NBLOCK,NEQB,NANAL,NOS,IDUM,NEIG,NAD,ANORM,N
VV,NFO,SDTOL
      WRITE (ND) NDUM,DUM,MDUM,NDYN,
        +
SKAL,IFSS,NITEM,RTOL,COFQ,IFPR,NFF,SDTOL
      WRITE (ND) L20,L25,STR
      WRITE (ND) NGVA,GVA,IOS,XYZOS
      CLOSE (ND)
      END

        SUBROUTINE VREME(A)
        INTEGER*2 I1,I2,I3,I4
        I1=0
        I2=0
        I3=0
        I4=0
        CALL GETTIM(I1,I2,I3,I4)
        A=FLOAT(3600*I1+60*I2+I3)
        END

      SUBROUTINE DAT(N,NN,IN,INS,UL)

      CHARACTER*15 UI,UL
      CHARACTER*1 UL1(3,44),ULL(15)
      EQUIVALENCE (UI,ULL(1))
      DATA
UL1/'T','A','C','E','L','E','O','P','T','P'
,'O','1',
     +
'N','A','2','N','I','3','N','C','4','O','P'
,'N',
     +         'G','E','N','I','Z','L',
     +
'L','I','N','P','O','V','Z','A','P','G','R'
,'A',
     +
'A','N','4','A','N','5','A','N','6','A','N'
,'7',
     +         'C','V','T','Z','A','Z',
     +
'J','E','D','S','I','L','M','A','S','I','Z'
,'P',
     +
'I','Z','N','L','G','P','G','G','T','G','G'
,'E',
     +         'G','G','O','T','A','C',
     +
'P','O','M','N','A','P','N','I','D','N','C'
,'T',
        +         'E','N','E',' ',' ',' ','
',' ',' ',' ',' ',' ',
     +         ' ',' ',' ',' ',' ',' ',
     +
'A','L','G','S','9','0','A','N','S','I','N'
,'P'/
        INP=0
      INS=0
      UI=UL
      IF(NN.EQ.0) GOTO 12
      NT=INDEX(UI,'.')
        NNT=NT
      IF(NT.GT.0) GOTO 12
       NT=INDEX(UI,' ')
       ULL(NT)='.'
      NTR=3
        IF((IN.LE. 2).AND.
     +   (NN.EQ.1.OR.NN.EQ.2.OR.NN.EQ.3.OR.
        +
NN.EQ.11.OR.NN.EQ.12.OR.NN.EQ.13)) NTR=1
       DO 10 J=NT+1,NT+NTR
   10 ULL(J)=UL1(J-NT,NN)
       IF(NTR.EQ.3) GOTO 12
         DO 16 J=NT+2,NT+3
   16 ULL(J)=' '
   12 IF(IN.EQ.0)
OPEN(N,FILE=UI,STATUS='UNKNOWN'       )
      IF(IN.EQ.1)
OPEN(N,FILE=UI,STATUS='NEW'    ,ERR=11)
      IF(IN.EQ.2)
OPEN(N,FILE=UI,STATUS='OLD'    ,ERR=11)
      IF(IN.EQ.3)
OPEN(N,FILE=UI,STATUS='OLD'
,ERR=11,ACCESS='APPEND')
      IF(IN.EQ.4)
OPEN(N,FILE=UI,FORM='UNFORMATTED',STATUS='U
NKNOWN')
      IF(IN.EQ.5)
OPEN(N,FILE=UI,FORM='UNFORMATTED',STATUS='O
LD',ERR=11)
      IF(IN.EQ.6)
OPEN(N,FILE=UI,FORM='UNFORMATTED',STATUS='U
NKNOWN',
     +
ACCESS='DIRECT',RECL=45)
      IF(IN.EQ.7)
OPEN(N,FILE=UI,FORM='UNFORMATTED',STATUS='O
LD',ERR=11,
     +
ACCESS='APPEND')
      IF(NN.EQ.0) GOTO 14
        IF(NNT.GT.0) GOTO 14
       DO 13 J=NT,NT+3
   13 ULL(J)=' '
      UL=UI
   14 RETURN
   11
IF((NN.GE.4.AND.NN.LE.10).OR.NN.GE.14) GOTO
```

```
15
        IF(IN.EQ.1) GOTO 15    INP=INP+1
        IF(INP.GE.2) GOTO 15
       ULL(NT+2)=UL1(2,NN)
        ULL(NT+3)=UL1(3,NN)
        GOTO 12
   15 INS=1
      END
      SUBROUTINE OMK(NJ2,NS,N,N1,N2)
       CHARACTER*15 GMK
       CHARACTER*1  U10(10),UU(10)
       DATA
U10/'0','1','2','3','4','5','6','7','8','9'
/
       DATA UU
/'A','B','C','D','E','F','G','H',':',' '/
       GMK = '  GMK000.DAT  '
       NFO=1
       IF(NJ2.GT.N2) NFO=0
        I1= NJ2     /100
        I2=(NJ2  -I1*100  )/10
        I3= NJ2  -I1*100-I2*10
         IF(NFO.EQ.0) GMK(1:1)=UU(10)
         IF(NFO.EQ.0) GMK(2:2)=UU(10)
         IF(NFO.EQ.1) GMK(1:1)=UU(N1)
         IF(NFO.EQ.1) GMK(2:2)=UU( 9)
         GMK(6:6)=U10(I1+1)
         GMK(7:7)=U10(I2+1)
         GMK(8:8)=U10(I3+1)

IF(N.EQ.4)OPEN(NS,FILE=GMK,FORM='UNFORMATTE
D',STATUS='UNKNOWN')

IF(N.EQ.5)OPEN(NS,FILE=GMK,FORM='UNFORMATTE
D',STATUS='OLD',ERR=11)
       RETURN
   11  STOP ' !!!  Ne postoji Matrica
krutosti  !!!'
       END

      SUBROUTINE TEO(NJ2,NDAT)

       CHARACTER*15 GMK
       CHARACTER*1  U10(10)
       DATA
U10/'0','1','2','3','4','5','6','7','8','9'
/
       DATA GMK/'ELE000.DAT     '/
        I1= NJ2     /100
        I2=(NJ2  -I1*100  )/10
        I3= NJ2  -I1*100-I2*10
         GMK(4:4)=U10(I1+1)
         GMK(5:5)=U10(I2+1)
         GMK(6:6)=U10(I3+1)
       CALL DAT (NDAT,0,4,INS,GMK)
      END
      SUBROUTINE ZAGLAV(I)
      WRITE (I,1)
    1 FORMAT(4x,
     1'--------    K      O      M      I
P         S  -----',
     2'------------'/
     14X,'            KOMPJUTERSKO
MODELIRANJE  I  PRORACUN  STRUKTURA',
     2'           '/
     14X,'Autor: Prof.dr Tasko Maneski ,
Masinski fakultet Beograd , ',
     2'tel 33-02-237'/
     14X,'MODELIRANJE      PREPROCESOR
PROCESOR           POSTP',
     2'ROCESOR     '/
     14X,'------------------    1D  ,  2D
I  3D    PROBLEMI    ------',
2'------------'/' ')      END
      SUBROUTINE MASKA1(I)
      do 4 L=1,11
    4 WRITE(*,*) ' '
      WRITE(*,1)
    1 FORMAT(
     14X,'--------    K      O      M
I        P         S  -------
     2----------'/
     14X,'            KOMPJUTERSKO
MODELIRANJE  I  PRORACUN  STRUKTURA
     2           '/
     14X,'
     2           '/
     14X,'Autor: Prof.dr Tasko Maneski ,
Masinski fakultet Beograd , tel
     2 33-02-237'/
     14X,'
     2           '/
     14X,' MODELIRANJE     PREPROCESOR
PROCESOR           POSTPRO
     2CESOR     '/
     14X,'-Preslikavanje  -Lok.generisanje
-Staticki          -Analize
     2 rezultata'/
     14X,'-Defin.problema -Glob. generisan
-Dinamicki         -Specif.
     2 proracuni'/
     14X,'-Izbor KE       -Grafika
-Termicki          -Grafika
     2           '/
     14X,'-Diskretizacija -Opt.poveziv.cv.
-Linear.i nelin.   -Element
     2i optimiz.')
      WRITE(*,2)
    2 FORMAT(
     14X,'-Gran.usl,Opter -Konverzije
-Stac.  i nestac.  -Konverz
     2ije       '/
     14X,'
     2           '/
     14X,'------------------    1D  ,  2D
I  3D    PROBLEMI    --------
     2----------'/' ')
      IF(I.EQ.0) RETURN
      READ(*,'(A)')
      do 3 L=1,26
    3 WRITE(*,*) ' '
      RETURN
      END

      SUBROUTINE KOMLIN(ULAZ,N,istat,ii)
       INCLUDE 'ZAJZON1M.FOR'
      character*(*) ULAZ
      integer*2 stat,i
       NN=N
        istat=nargs()
       if(istat.ne.1) goto 11
ULAZ(1)
       if(   ii.eq.0) CALL READR(0)
       if(   ii.eq.0) ULAZ=terk
       if(   ii.eq.1) ULAZ=' '
      GOTO 12
 11   do 10 i=1,istat-1
   10  call getarg(i,ULAZ,stat)
   12  istat=istat-1
       end
```

**Прилог 2**

**Образац 5.**

**Изјава о ауторству**

Име и презиме аутора Милош Манески

Број индекса Д-2009/80

**Изјављујем**

да је докторска дисертација под насловом

РАЗВОЈ МАКРО КОНАЧНОГ ЕЛЕМЕНТА ПРИМЕНОМ ГРУПНО СУПЕРМАТРИЧНЕ ПРОЦЕДУРЕ ЗА НАБОРАНЕ КОНСТРУКЦИЈЕ

- резултат сопственог истраживачког рада;
- да дисертација у целини ни у деловима није била предложена за стицање друге дипломе према студијским програмима других високошколских установа;
- да су резултати коректно наведени и
- да нисам кршио/ла ауторска права и користио/ла интелектуалну својину других лица.

**Потпис аутора**

У Београду, _________________ ____________________________

**Прилог 3**

**Образац 6.**

**Изјава о истоветности штампане и електронске верзије докторског рада**

Име и презиме аутора _Милош Манески

Број индекса __Д-2009/80

Студијски програм _Архитектура и урбанизам__

Наслов рада _Развој макро коначног елемента применом групно суперматричне процедуре за набоpане конструкције

Ментор _др. Миодраг Несторовић

Изјављујем да је штампана верзија мог докторског рада истоветна електронској верзији коју сам предао/ла ради похрањена у **Дигиталном репозиторијуму Универзитета у Београду.**

Дозвољавам да се објаве моји лични подаци везани за добијање академског назива доктора наука, као што су име и презиме, година и место рођења и датум одбране рада.

Ови лични подаци могу се објавити на мрежним страницама дигиталне библиотеке, у електронском каталогу и у публикацијама Универзитета у Београду.

**Потпис аутора**

У Београду, ________________________ ________________________

**Прилог 4**

**Образац 7.**

**Изјава о коришћењу**

Овлашћујем Универзитетску библиотеку „Светозар Марковић“ да у Дигитални репозиторијум Универзитета у Београду унесе моју докторску дисертацију под насловом:

РАЗВОЈ МАКРО КОНАЧНОГ ЕЛЕМЕНТА ПРИМЕНОМ ГРУПНО СУПЕРМАТРИЧНЕ ПРОЦЕДУРЕ ЗА НАБОРАНЕ КОНСТРУКЦИЈЕ

која је моје ауторско дело.

Дисертацију са свим прилозима предао/ла сам у електронском формату погодном за трајно архивирање.

Моју докторску дисертацију похрањену у Дигиталном репозиторијуму Универзитета у Београду и доступну у отвореном приступу могу да користе сви који поштују одредбе садржане у одабраном типу лиценце Креативне заједнице (Creative Commons) за коју сам се одлучио/ла.

1. Ауторство (CC BY)

2. Ауторство – некомерцијално (CC BY-NC)

3. Ауторство – некомерцијално – без прераде (CC BY-NC-ND)

4. Ауторство – некомерцијално – делити под истим условима (CC BY-NC-SA)

5. Ауторство – без прерада (CC BY-ND)

6. Ауторство – делити под истим условима (CC BY-SA)

(Молимо да заокружите само једну од шест понуђених лиценци.
Кратак опис лиценци је саставни део ове изјаве).

**Потпис аутора**

У Београду, __________________________ ______________________

1. **Ауторство**. Дозвољавате умножавање, дистрибуцију и јавно саопштавање дела, и прераде, ако се наведе име аутора на начин одређен од стране аутора или даваоца лиценце, чак и у комерцијалне сврхе. Ово је најслободнија од свих лиценци.

2. **Ауторство – некомерцијално**. Дозвољавате умножавање, дистрибуцију и јавно саопштавање дела, и прераде, ако се наведе име аутора на начин одређен од стране аутора или даваоца лиценце. Ова лиценца не дозвољава комерцијалну употребу дела.

3. **Ауторство – некомерцијално – без прерада**. Дозвољавате умножавање, дистрибуцију и јавно саопштавање дела, без промена, преобликовања или употребе дела у свом делу, ако се наведе име аутора на начин одређен од стране аутора или даваоца лиценце. Ова лиценца не дозвољава комерцијалну употребу дела. У односу на све остале лиценце, овом лиценцом се ограничава највећи обим права коришћења дела.

4. **Ауторство – некомерцијално – делити под истим условима**. Дозвољавате умножавање, дистрибуцију и јавно саопштавање дела, и прераде, ако се наведе име аутора на начин одређен од стране аутора или даваоца лиценце и ако се прерада дистрибуира под истом или сличном лиценцом. Ова лиценца не дозвољава комерцијалну употребу дела и прерада.

5. **Ауторство – без прерада**. Дозвољавате умножавање, дистрибуцију и јавно саопштавање дела, без промена, преобликовања или употребе дела у свом делу, ако се наведе име аутора на начин одређен од стране аутора или даваоца лиценце. Ова лиценца дозвољава комерцијалну употребу дела.

6. **Ауторство – делити под истим условима**. Дозвољавате умножавање, дистрибуцију и јавно саопштавање дела, и прераде, ако се наведе име аутора на начин одређен од стране аутора или даваоца лиценце и ако се прерада дистрибуира под истом или сличном лиценцом. Ова лиценца дозвољава комерцијалну употребу дела и прерада. Слична је софтверским лиценцама, односно лиценцама отвореног кода.