



ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

Alba Regia Műszaki Kar
Geoinformatikai Intézet



SZAKDOLGOZAT

3D-S SAJÁTOS EREDETŰ GEOFIZIKAI ADATOK TÉRINFORMATIKAI MEGJELENÍTÉSE

**OE-AMK
2022**

Hallgató neve: Czírok Lili

Hallgató törzskönyvi száma: T000666/FI12904/A-G



ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

Óbudai Egyetem
Alba Regia Műszaki Kar
Geoinformatikai Intézet

SZAKDOLGOZAT FELADATLAP

Hallgató neve: Czírok Lili

Szaktervező száma: SZD21051109223348

Törzskönyvi száma: T000666/FI12904/A-G

Neptun kódja: XXXXXXXXXX

Szak: geoinformatikai szakmérnök

Specializáció:

A dolgozat címe: 3D-s sajátos eredetű geofizikai adatok térinformatikai megjelenítése

A dolgozat címe angolul: Geospatial representation of 3D geophysical data with specific origin

A feladat részletezése:

A Délkeleti-Kárpátok külső kanyarulatában, az úgynevezett Vráncsa-térségben jelentős földrengés-tevékenység zajlik. A területre jellemző feszültségteret és földtani helyzetet a területen kipattant földrengésekhez kiszámított fészekmechanizmus-megoldások bemutatásával lehet jellemezni. A fészekmechanizmus megoldást a földrengés fészében az eltörő kőzettest két darabjának egymáshoz képesti elmozdulásakor létrejövő vetőfelület csapását és dőlését sztereografikus vetületben ábrázoló ábrával, ("strandlabda diagram"-mal) szokás reprezentálni. Minden földrengéshez (amely földrajzi elhelyezkedése 3D-ban írható le, szélesség, hosszúság - koordináták és fészekmélység km-es dimenzióban megadva) hozzárendelhető egy strandlabda diagram.

A szaktervezőben a jelölt mutassa be, hogy földrengések fészekmechanizmusait bemutató térképek elkészítéséhez milyen szoftvereket és kódokat vagy éppen programcsomagokat használhatunk és hogy miként törekszünk az ábrázolási feladatok mennyiségének csökkentésére és hatékonyságának növelésére.

Jelenleg öt különböző program segítségével történik a bevitt adatokhoz, az eredményekhez és az interpretációkhoz kapcsolódó vizualizációs feladatok és számítások elvégzése. A fészekmechanizmusok sztereografikus projekciója után kapott úgynevezett strandlabdák térképet Generic Mapping Tools-ban állítjuk elő. A számításokat megelőző klaszteranalízises algoritmusok lefuttatása Python-kódok segítségével történik. A kapott kisebb területek (klaszterek) térképeit QGIS-ben állítjuk elő. Ezt követően egy MATLAB-környezetben futtatható feszültség-inverziós kóddal, a STRESSINVERSE-zel dolgozunk, amely raszteres formátumban (PNG-formátum) exportálja a főfeszültségtengelyek orientációját demonstráló sztereogramokat. A sztereogramok kiértékelése szintén QGIS-ben történik és kerül illusztrálásra, azonban a sztereogramokat egy QGIS-ben előállított alaptérképre illesztjük rá LibreOffice-ban (vagy CorelDraw-ban).

A munka célja az, hogy csökkentsük a felhasznált programok, szoftverek mennyiségét és ezzel növelve a megjelenítési feladatok gyorsabb és hatékonyabb elvégzését. A jelölt találjon vagy fejlesszen olyan szoftvert, modult vagy programot, amely a legtöbb részfeladatot egyesíti és alkalmas a sajátos eredetű geofizikai adatok 2D-s, (esetleg 3D-s) megjelenítésére.

Intézményi konzulens neve: Molnár Gábor Péter

Külső konzulens neve: Bozsó István

Munkahelye: ELKH Földfizikai és Űrtudományi Kutatóintézet



ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

Óbudai Egyetem
Alba Regia Műszaki Kar
Geoinformatikai Intézet

A kiadott téma elővülési határideje: 2023. 12. 15.

Beadási határidő: 2022. 05. 15.

A szakdolgozat: Nem titkos.

Kiadva: Székesfehérvár, 2022. 05. 03.



P.H

Intézetigazgató

A dolgozatot beadásra alkalmasnak találom:

belső konzulens

külső konzulens



ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

Alba Regia Műszaki Kar
Geoinformatikai Intézet

HALLGATÓI NYILATKOZAT

Alulírott szigorló hallgató kijelentem, hogy a szakdolgozat saját munkám eredménye, a felhasznált szakirodalmat és eszközöket azonosíthatóan közöltem.

Egyéb jelentős segítséget nem vettem igénybe.

Az elkészült szakdolgozatban található eredményeket az egyetem és a feladatot kiíró intézmény saját céljára térítés nélkül felhasználhatja, a titkosításra vonatkozó esetleges megkötések mellett.

Kelt: ...*Székesfehérvár, 2022. május 5.*...

Gábor Lili

hallgató aláírása

Tartalomjegyzék

Összefoglalás	6
1. Bevezetés	8
1.1. Földtani háttér	8
1.2. Feszültségtér – vizsgálatok a Délkeleti-Kárpátokban	10
2. Felhasznált adatok	12
3. Alkalmazott eszközök	14
3.1. Python	14
3.2. GIS	16
4. Eredmények	20
4.1. Felhasznált adatok ábrázolása QGIS-ben	20
4.2. Térbeli lekérdezések SpatiaLite-ban	22
5. Diskusszió	39
6. Következtetések	41
Irodalomjegyzék	42
Függelék	46

Összefoglalás

A Vrancea-zóna - amely a Délkeleti-Kárpátok alatt helyezkedik el, és amelyet gyakran egy leválás alatt álló óceáni lemezként értelmeznek - jelentős földrengés aktivitással rendelkező terület. A terület feszültségmezeje és tektonikai háttere a térségben bekövetkezett földrengésekre számított fészekmechanizmus-megoldásainak inverziójával tanulmányozható. A fészekmechanizmus-megoldásokat általában a földrengésfészekben a két töredezett közettömeg egymáshoz viszonyított elmozdulásából keletkező vetületi felszín csapásirányának és dőlésének sztereografikus vetülete ("strandlabda-diagram") ábrázolja. Minden egyes földrengéshez (amelynek földrajzi helyét 3D-ben lehet leírni, szélességi, hosszúsági - koordinátákkal és hipocentrummal km dimenzióban) egy úgynevezett "strandlabda" társítható. Önmagában azonban a fészekmechanizmusok nem elegendőek ahhoz, hogy pontosan megismerjük a geológiai környezetet, ezért kell végrehajtani azok inverzióját (más néven feszültséginverzióját).

Jelenleg öt különböző programot használok a bemeneti adatokkal, eredményekkel és értelmezésekkel kapcsolatos vizualizációs feladatok és számítások elvégzésére. A fészekmechanizmusok sztereografikus projekciója után kapott úgynevezett strandlabda-térképet a Generic Mapping Tools programban készítek el. A számításokat megelőző klaszterelemző algoritmusok Python kódok segítségével futnak. Az így kapott kis területek (klaszterek) térképei a QGIS-ben kerülnek előállításra. Ezt követően a MATLAB környezetben futtatható STRESSINVERSE feszültséginverziós kódot használok a fő feszültségtengelyek orientációját bemutató sztereogramok raszteres formátumban (PNG formátumban) történő kimentésére, valamint az eredményül kapott feszültségtenzor mennyiségi értékeit tartalmazó mátrixot is innen töltöm le. A sztereogramok értelmezése szintén a QGIS-ben történik, és különböző színű nyilakkal ábrázolom, amelyek a domináns töréstípusokat jelölik, de a sztereogramokat a QGIS által LibreOffice-ban (vagy CorelDraw-ban) előállított alaptérképre helyezem rá. A munka célja, hogy olyan megoldást találjak, amely csökkenti a felhasznált programok és szoftverek számát, és ez által növeli a vizualizációs feladatok sebességét és hatékonyságát.

Summary

The Vrancea region - located beneath the Southeast Carpathians often interpreted as an oceanic slab undergoing detachment - is a zone of significant earthquake activity. The stress field and geological setting of the area can be characterised by presenting calculated focal mechanism solutions of earthquakes that have occurred in the area. The focal mechanism solutions are usually represented by a stereographic projection of the strike and dip of the seeding surface in the earthquake nest, which is generated by the displacement of two pieces of fractured rock mass relative to each other ("beachball diagram"). For each earthquake (whose geographic location is described in 3D, with latitude, longitude - coordinates and hypocentre in km dimension) a beachball diagram can be associated. In order to gain more information about the tectonic setting, it is also necessary to carry out the inversion of these focal mechanisms (stress inversion).

Currently, five different programs are used to perform visualisation tasks and calculations related to the input data, results and interpretations. The so-called beachball map obtained after stereographic projection of focal mechanisms is produced in Generic Mapping Tools. The cluster analysis algorithms preceding the computations are run using Python codes. The resulting maps of small areas (clusters) are produced in QGIS. Subsequently, the stress inversion code, STRESSINVERSE, which operates in a MATLAB environment, is applied to export stereograms demonstrating the orientation of the principal stress axes in raster format (PNG format). The interpretation of the stereograms is also done and illustrated in QGIS by differently coloured arrows that represent the dominant fault types, but the stereograms are overlaid on a base map produced by QGIS in LibreOffice (or CorelDraw). The aim of this work is to find a solution to reduce the number of programs and software utilized and thus to increase the speed and efficiency of the visualisation tasks.

1. Bevezetés

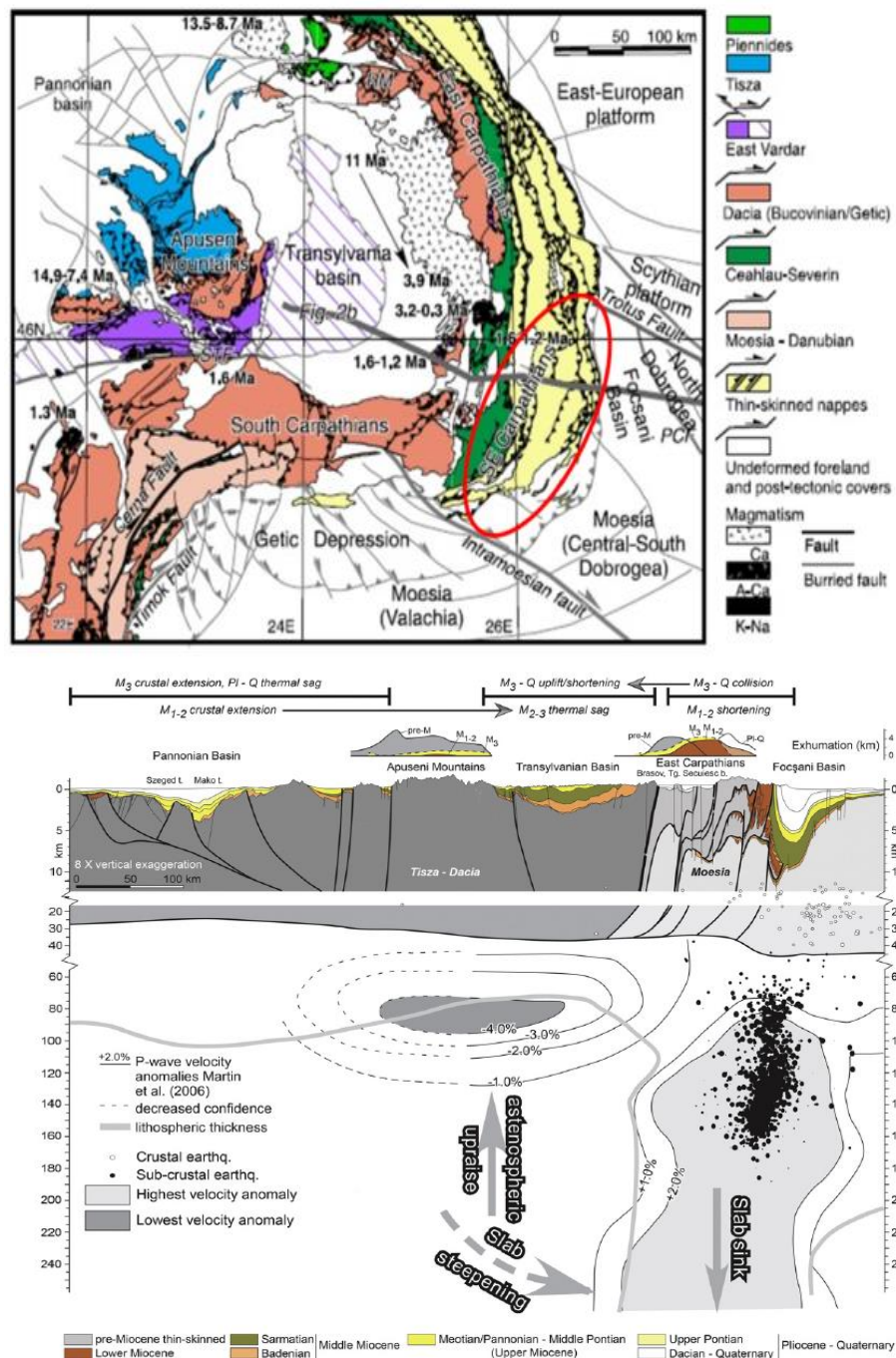
1.1. Földtani háttér

A Délkeleti-Kárpátok egy izgalmas és érdekes terület a földtudományi szakemberek számára, köszönhetően a napjainkban is zajló geodinamikai folyamatoknak. A térség belső (nyugati) részén a Kelemen-Görgényi – havasok és Hargita mentén egy vulkáni hegylánc terül el. Ez a vulkáni lánc a késő jura és korai kréta között lezajlott Neo-Thétisz remanens óceáni lemez lebukásához köthető (Harangi et al, 2015). Ennek legdélebbre található és egyben legfiatalabb tagja a Csomád vulkán, amely csupán 32 000 évvel ezelőtt tört ki utoljára (Harangi et al, 2015). Ezt követően a kora kréta végétől a késő krétáig bezárólag a Ceahlău–Severin óceán bezáródása zajlott, a Kárpátkanyar keleti felében. Ehhez a szubdukciós folyamathoz kapcsolódóan napjainkban is erős földrengés-tevékenység társul. Ez az úgynevezett Vrancea-zóna, ahol a szeizmikus események közepes fészekmélységűek (a kipattanási hely mélysége vagy más néven hipocentrum), 60 és 200 km között történnek ezek a szeizmikus események.

A Kárpátok délkeleti része már több geofizikai és geodéziai mérésorozatnak volt a célpontja. A különböző műholdas geodéziai mérések, mint pl. Globális Helymeghatározó Rendszerek (Global Positioning System, továbbiakban GPS) vagy éppen interferometrikus szintetikus apertúrájú radarok (továbbiakban InSAR) mérései, amelyek leginkább a földfelszínen történő horizontális (észak-déli és kelet-nyugati) -, valamint a vertikális mozgások térképezésére, monitorozására alkalmasak (Heidbach et al, 2007, Schmitt et al, 2007, Szűcs et al, 2018). Ezek a típusú mérések kimutatták a már középső miocénben elkezdődött és a mai napig tartó süllyedést, ennek mértéke jelenleg körülbelül 1 mm/év (Ismail-Zadeh et al, 2012).

Ami a geofizikai mérési módszereket illeti: leginkább itt a különböző szeizmológiai vizsgálati módszereket érdemes kiemelni, de pl. (Ismail-Zadeh et al, 2012) bemutatott geoelektromos, gravitációs és mágneses mérési kampányok eredményeit is. Jelen munka szempontjából a legérdekesebb mégis a P-hullámokon alapuló szeizmikus tomográfia, amellyel egy $110 \times 70 \times 30$ km³-es térfogatú, környezetéhez képest magasabb szeizmikus hullámterjedéssel rendelkező testet találtak meg. Ez a test jó átfedésben van a Vrancea térségében található szeizmogén zónával. Az 1. ábra egy térképet mutat be a Romániai Kárpátok tektonikai egységeiről (felső, Ismail-Zadeh et al, 2012) illetve egy K-Ny-i irányú

szelvényt, amely egy szeizmikus tomografikus mérés értelmezésén alapul (alsó, Matenco and Radivojević, 2012).



1. ábra, felső: a Romániai Kárpátokban fellelhető tektonikai egységek és földtani szerkezetek. Piros ellipszis jelzi a dolgozat fókuszterületét, a Vrancea-zónát. Alsó: Matenco and Radivojević, 2012 publikációjában szereplő értelmezett szeizmikus tomográfiai szelvény látható a Délkeleti-Kárpátok területén.

1.2. Feszültségter – vizsgálatok a Délkeleti-Kárpátokban

A földrengésveszélyes Vrancea-zóna egy szűk régió, amelynek epicentrális területe körülbelül 30*70 km. A hipocentrumok általános eloszlása egy övvel jellemezhető, a 60 és 170 km közötti mélységintervallumban. Ugyanakkor egy vékony átmeneti öv is kimutatható (kb. 100 km magasságban), amely a szeizmogén övezetet két részre osztja: egy felső (UASZ – Upper Active Seismogenic Zone) és egy alsó aktív szeizmogén övezetre (LASZ – Lower Active Seismogenic Zone), feltételezhetően egy kontinentális és egy óceáni töredékre (Carbunar és Radulian 2011).

Ami a fészekmechanizmusokhoz tartozó segédsíkokat illeti, figyelemre méltó, hogy a törési folyamatok fő iránya a Kárpátok ívével párhuzamos (ÉK-DNy-i irányban), és a dőlési szögek általában 40 foknál nagyobbak (Popescu et al, 2018). Mindazonáltal a kisebb szeizmikus események merőleges, ÉNy-DK irányú terjedésre utalnak (Popescu et al, 2018; Radulian et al, 2018). Ezen korábban közzétett információk alapján észrevehető, hogy a fő tektonikai rezsimek feltolódások (vagy más néven reverz vetők). Ezek a szerkezetek alkalmasak a kompressziós tektonikai stílus dominanciájának igazolására, és így az említett geodinamikai folyamatok bizonyítékai is.

A fészekmechanizmus-megoldások invertálásával is lehetőség van az adott terület feszültségterének térképezésére. Ezek az úgynevezett feszültséginverziók. Ahhoz, hogy ilyen számítások elvégzése lehetséges legyen, a következő feltételeknek kell teljesülnie (Michael 1984, Gephart 1990, Angelier 2002, Czirok, 2016):

1. A feszültségmezőnek homogénnek kell lennie.
2. A földrengések már létező törések mentén, változó irányítottsággal alakulnak ki.
3. Wallace-Bott hipotézis: a fészekmechanizmus-megoldás által megadott csúszási vektor iránya megegyezik a keresett feszültségtenzor által a vetősíkon létrehozott nyírási feszültség irányával. (Wallace, 1951, Bott, 1959)

Ahhoz, hogy ezek a feltételek teljesüljenek, az esetek zömében kisebb alterületekre kell felosztani a tanulmányozott régiót (Hardebeck and Michael, 2006, Maury et al, 2013, Kassaras and Kapetanidis, 2018), így pontosabban térképezhetőek a térbeli és akár időbeli változások is.

Többféle inverziós eljárás is létezik, lineáris (Michael, 1984, Hardebeck és Michael, 2006, Vavrycuk, 2014) és nemlineáris (Gephart, 1990, Angelier, 2002). A DK-i Kárpátok

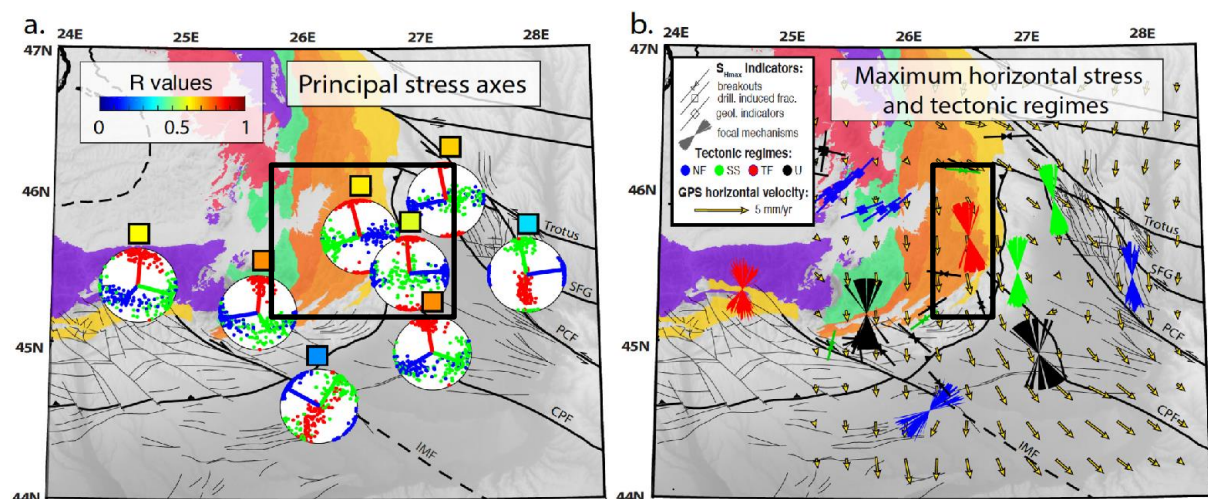
külső ívén található szeizmogén övezetekre (pl. Onescu, 1988, Bala et al, 2021, Petrescu et al, 2021) főként lineáris, csillapított (Hardebeck és Michael, 2006, Martinez-Garzon et al, 2014) illetve ennek a Mohr-Coulomb törési kritérium alkalmazásával továbbfejlesztett (Vavrycuk, 2014) változatával végeztek el ilyen típusú számításokat. Ezeknek az eredményei magában foglalják a kapott főfeszültség-tengelyek irányítottságát és az egymáshoz viszonyított helyzetüket leíró R-számot (Gephart and Forsyth, 1984, Bada et al, 2007). A sztereogramokon is szereplő arányossági tényező a következő egyenlettel fejezhető ki:

$$R = \frac{(\sigma_1 - \sigma_2)}{(\sigma_1 - \sigma_3)} \quad (1)$$

Ez utóbbi értéke 0 és 1 között változhat, 0 tartozik a feltolódásokhoz, 0.5 az oldalelmozdulásokhoz és 1 a normál vetőkhöz. Köztes vetőstílusok is meghatározhatóak ez alapján: ezek a transzpresszió (oldaleltolódás feltolódásos összetevővel) illetve a transztenzió (oldalelmozdulás normál vetős komponenssel).

Az feszültség-inverziók eredményeként kapott úgynevezett feszülsténzor (pontosabban annak deviatorikus része) alkalmas a területre jellemző maximális horizontális feszültségorientációk és a domináns feszültség-rezsimek meghatározására.

A Vráncsa térségében ezek a számítások is igazolják a feltolódások uralkodó szerepét és az így kapott feszültség-irányok is követik azt a trendet, amelyet a segédsíkok alapján meg lehet határozni. A 2. ábrán (Petrescu et al, 2021) publikációjában megjelentett eredmények és azok értelmezése látható. A cikkben az MSATSI (Martinez-Garzon, 2014) MATLAB-környezetében futtatható feszültség-inverziós programmal számított eredmények térképes értelmezése tekinthető meg.



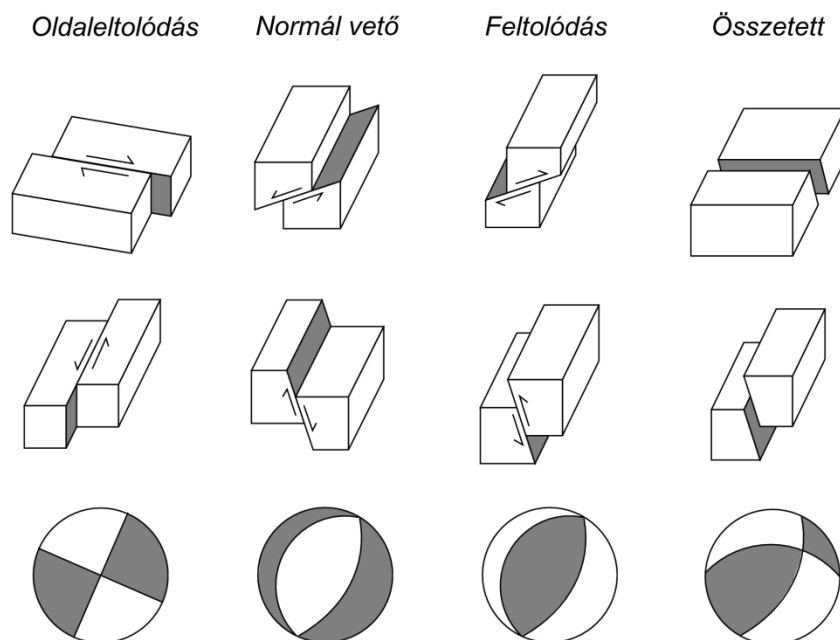
2. ábra: A) A Délkeleti Kárpátokban elvégzett feszültség-inverziók során kapott főfeszültség-sztereogramok (piros a σ_1 , zöld a σ_2 és kék a σ_3) és az R-számok (kék 0 – t, piros az 1-t jelöli) láthatóak. B) A feszülsténzor

alapján meghatározott horizontális maximális kompressziós irányok és a kapcsolódó uralkodó tektonikus rezsimek (piros a reverz vető, zöld az oldaleltolódás, kék a normál vető, fekete pedig az ismeretlen vagy nem meghatározható) vannak megjelenítve a GPS-mérések alapján meghatározott horizontális mozgás-sebességekkel együtt (Petrescu et al, 2021).

A bal oldalon látható sztereogramok szemléltetik a három Anderson-féle főfeszültség (Anderson, 1951) irányát (piros a σ_1 , azaz a legnagyobb, zöld a σ_2 és kék a σ_3 , ami a legkisebb) és azoknak megbízhatóságát (A pontok az úgynevezett realizációs számítások eredményei. Minél nagyobb az eloszlásuk, annál kevésbé egyértelműen írható le a helyzete az adott főfeszültség-tengelynek), a felettük lévő kis négyzet színe pedig az R-szám értékét mutatja (kék a legkisebb, piros a legmagasabb értékeket jelöli). Ami a legközelebb van a sztereogram középpontjához, az a tengely a van a legközelebb a függőleges helyzethez, az irányt pedig a leginkább horizontális feszültségtengely azimutja adja meg. A fekete négyzetek jelölik a számunkra érdekes térséget, itt a σ_3 – tengelynek a legnagyobb a dőlése és a σ_1 – főtengely azimutja alapján ÉÉK-DDNy maximális horizontális kompressziós irány a jellemző. A jobb oldali térkép egy kiértékelése a sztereogramoknak, de más feszültségindikátort is feltüntet (pl. fúrások vagy terepi felmérések eredményeit), valamint GPS-mérésekből meghatározott lemezmozgások is szerepelnek. Ezek az eredmények a World Stress Map Project-hez köthetők (Heidbach et al, 2016), amely világszerte szolgáltat adatokat, információkat a feszültségmezőben lévő térbeli és időbeli változásokról egyaránt mind kutatásokban, mind akár ipari alkalmazásokban (Heidbach et al, 2018). Ez az eredmény azt mutatja, hogy a feltolódások (piros nyilak) töltik be az uralkodó szerepet a térségben, amely indikátora a szubdukciós eseményeknek.

2. Felhasznált adatok

A földrengések regisztrátumából az események földrajzi helyzetét (szélesség és hosszúság koordináták, a hipocentrum mélysége) és magnitúdója (erőssége) mellett a rengéshez kapcsolódó földtani szerkezetről is nyerhetünk információt. Ezeket az információkat az úgynevezett fészekmechanizmus – megoldások (a dőlésszög, dőlésirány és csúszásirány segítségével tudja leírni, milyen földtani vetőhöz kapcsolódhatott egy adott rengés) illetve a belőlük származtatott főfeszültség-tengelyekhez tartozó irányok (azimut és dőlésirány) adják. A 3. ábra egy összefoglaló ábra arról, hogy a fészekmechanizmus-megoldás mely típusa mely tektonikai szerkezethez köthető.



3. ábra: Leginkább jellemző vetőtípusok (felső két sor) illetve az őket reprezentáló fészekmechanizmus-megoldások sztereografikus projekciói, azaz a strandlabdák (alsó sor) Shearer, 2009 alapján

Ebben a munkában a Vrancea-zónában kipattant földrengésekhez tartozó fészekmechanizmusok kerülnek illusztrálásra. A munka során elsősorban a Romániai Földfizikai Intézet által megosztott adatokat használtam, azon belül is az újraszámított (relokalizást) eseményeknek a katalógusából (Romanian Earthquake Focal Mechanism Catalogue), amelyet innen le lehet tölteni: <https://data.mendeley.com/datasets/mykxx4gygy/3> (Radulian et al, 2018, 2019, 2020). A katalógus Excel-táblázat formájában készült el és 1929-től 2012-ig bezárólag tartalmaz adatokat. Ezen a katalóguson felül még lokális mérések alapján meghatározott fészekmechanizmusokat itt találunk: <http://atlas2.infp.ro/~rt/fms/>. A dolgozatban bemutatott megoldások 2014-től napjainkig kipattant szeizmikus eseményekhez meghatározott fészekmechanizmus-megoldásokat tartalmaznak.

Fészekmechanizmus-megoldásokat nem csak helyben lévő szeizmológiai mérőállomások tudnak meghatározni. Nagyobb geológiai és geofizikai szolgálatok a világon, mint pl. az United States Geological Survey (USGS, <https://earthquake.usgs.gov/earthquakes/search/>) vagy az European-Mediterranean Seismological Centre (EMSC, <https://www.emsc-csem.org/Earthquake/emma.php>) a világhálón elérhetővé tesznek olyan listákat, amelyekben az általuk regisztrált, úgynevezett teleszeizmikus (a kipattant földrengésektől nagyobb, mint 1000 km-es távolságban lévő állomások) vagy regionális (250 és 1000 km közötti távolságban lévő állomások)

földrengésekhez tartozó adatok (epicentrum, hipocentrum, fészekmechanizmus-megoldás stb.) szerepelnek. Ehhez a kutatómunkához kapcsolódóan a fészekmechanizmus megoldásokat az International Seismological Centre bulletinjeiből gyűjtöttem ki (Lentas et al, 2018, 2019). Ezeket a teleszeizmikus eseményeket csakis kizárólag a helyi adatokban lévő hiány feltöltésére használtam, tehát mindössze 2012 és 2014. között kialakult események megoldásait ábrázoltam az előbb említett földrengés-katalógusból.

3. Alkalmazott eszközök

3.1. Python

A Python egy nyílt forráskódú, objektumorientált és értelmezett nyelv (nincs külön elválasztott forrás- és tárgykód, a megfelelő értelmezővel rögtön futtatható a kód megírása után). Napjainkban rendkívül sok tudományos projektben használják, amelyekhez kapcsolódóan úgynevezett package-ek (csomagok) is fejlesztésre kerültek (pl. Pedregosa et al, 2011, McInnes et al, 2017, Goulard, 2020), de már a földtudományok területén is egyre nagyobb szerepet kapnak. Ilyen pl. az `Obspy` – csomag, amely kifejezetten szeizmológiai vizsgálatok elvégzésére készült (Beyreuther et al, 2010, Megies et al, 2011, Krischer et al, 2015) vagy `pyGIMLi` (Wagner et al, 2015), ami pedig geofizikai inverziós számításokra és kapcsolódó modellezési feladatokra alkalmas.

Ebben a munkában a Python – nyelv használata a következő feladatkörökben került felhasználásra:

A) A földrengés-adatok feszültség-inverzióra való előkészítéséhez, az alterületek létrehozására is ebben a nyelvben írt klaszterező algoritmusokat használtam: agglomeratív hierarchikus klaszterezés (Pedregosa et al, 2011) és hierarchikus sűrűség-alapú térbeli klaszterezés alkalmazása zajjal terhelt adatokon (HDBSCAN, McInnes et al, 2017), ami a DBSCAN (Ester et al, 1996) továbbfejlesztett változata. Emellett kifejlesztésre került egy teljesen automatizált, csak egy opcionális hiperparaméterrel rendelkező klaszterező algoritmus (Czirok és Kuslits, 2018, Czirok et al, 2022 – elbírálás folyamatban, Kuslits et al, 2022 EGU, Czirok et al, 2022 EGU). Az algoritmusnak már van egy GitHub-repozitóriuma és Python – csomagja is elérhető a következő linken: <https://github.com/lukacskuslits/clu-stress>. A dolgozatban ennek a (majdnem) teljesen automatizált klaszterező algoritmusnak csak az eredményei kerülnek bemutatásra.

B) Az előbb említett programfejlesztéshez kötődően létrehoztunk egy szintetikus fészekmechanizmusokból álló adatrendszer (Czirok et al, 2022 –elbírálás alatt, Kuslits et al, 2022 – EGU). A fészekmechanizmusok alapvetően három szöggel írhatóak le: dőlésszög, amely 0 és 360° között változik, dőlésirány, amely 0 és 90° közötti értéket vehet fel és a csúszásirány, amely 0-360° vagy -180-180° intervallumokban értelmezhető (Cronin, 2010). Ezen az adatokon felül a generált pontokhoz hozzárendelésre került szélesség- és hosszúság koordináta, illetve fészekmélység is. A pontokat és a pontcsoportokat magukba foglaló alakzatokat a `Shapely` – csomaggal (Gillies et al, 2007) gyártottam le. A `Shapely` - programcsomag síkbeli geometrikus objektumok vizsgálatára és kezelésére használható és elsősorban `GEOS` – könyvtárra épül (GEOS contributors, 2021). A szintetikus fészekmechanizmusok szögeit, valamint a hozzátartozó pontok koordinátáit a Python-nyelvhez tartozó “*random*” - csomag (Goulard, 2020) függvényeinek felhasználásával generáltam (Függelék, Czirok et al, 2022, Kuslits et al, 2022). Ehhez létrehozásra került a QGIS-ben egy plugin (SynFMS néven) ami a Clu-Stress repozitóriumban is elérhető lesz.

C) A szintetikus és a valódi fészekmechanizmus-megoldások ábrázolása az `Obspy` nevű Python-csomag `beachball()` függvényével történt (Beyreuther et al, 2010, Megies et al, 2011, Krischer et al, 2015). A valódi szeizmikus eseményeket ezután SVG (Scalable Vector Graphics) – formátumban exportáltam, így ezek már SVG-szimbólumokként bevezethetők voltak QGIS-be.

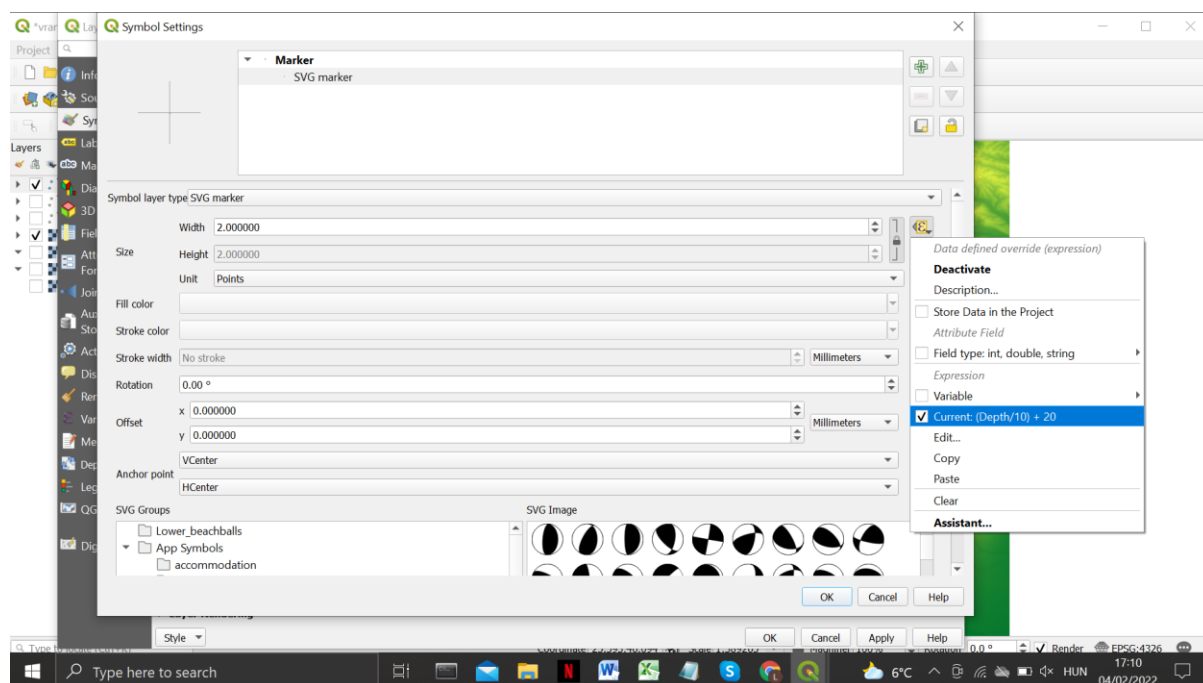
D) Feszültséginverziók lefuttatása: Python-ban a `STRESSINVERSE` forráskódját futtattam le. Eredményekként egyrészt megkaptam raszteres képeket (PNG formátumban) a Mohr-körök, a bevitt fészekmechanizmusok alapján meghatározott kompressziós (P-) – és dilatációs (T-) tengelyek helyzeteit, valamint a főfeszültség-tengelyek bizonytalanságát reprezentáló sztereogramot és az előforduló R-számok hisztogramját. Kvantitatív kiértékelésre alkalmas a `MAT` – formátumban exportált feszültségtenzor. Ez magában foglalja a leginkább megbízható R-számot (ami megfelel a hisztogramon a leggyakrabban előforduló értéknek) és a három főfeszültség-tengelyhez tartozó, leginkább megbízható dölések és azimutok értékeit. A `MAT` – formátum önmagában nem kompatibilis a Python-nal, QGIS-szel vagy bármilyen más szöveges programmal, így Python-ban ezt a `mat4py` (<https://pypi.org/project/mat4py/>) Python-csomag `loadmat()` függvényének a segítségével kaptam egy Python - objektumot, ezt átalakítottam `DataFrame` – típusú objektummá, majd kimentettem Excel-táblázatba és CSV-formátumba is. Innentől már a földrengések helyadatait hozzáadva lehetséges volt QGIS-be is importálni az eredményeket.

3.2. GIS

A térinformatikai alkalmazásokhoz a térképek elkészítése során használtam fel. A térinformatikai szoftverek alkalmasak arra, hogy egy rendszeren belül ábrázoljunk különböző típusú adatokat, elemezzük azokat és ezek eredményeiből (tematikus) térképet hozunk létre. A munka során fontos volt, hogy amellett, hogy minél több feladatot tudjunk elvégezni egy keretrendszerben vagy egy programban, az is számított, hogy az eszköz, amivel dolgozunk, ingyenes elérhető és jól kezelhető legyen. Így került kiválasztásra a Quantum GIS (QGIS.org, 2022). A következő lépéseket hajtottam végre QGIS-ben:

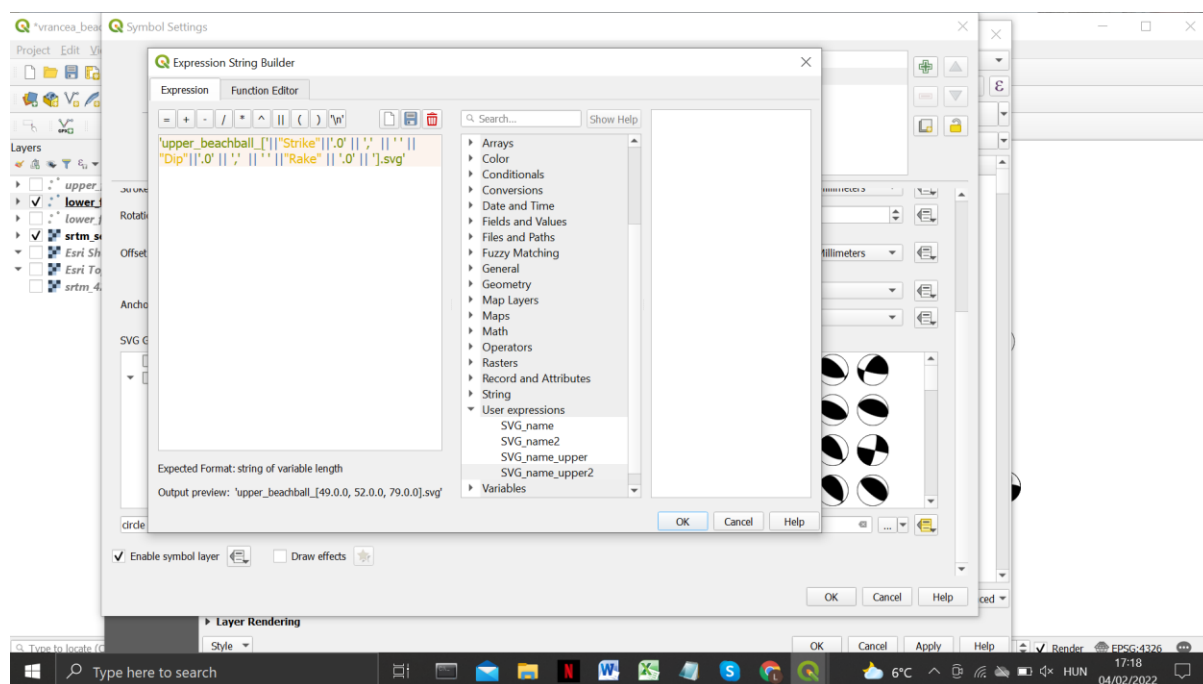
A) A 3.1. alfejezet C) pontjában bemutatott fészekmechanizmusokat itt ábrázoltam domborzatmodellel együtt (a domborzatmodellt a QGIS-en belül található SRTM-Downloader plugin-nal töltöttem le, <https://plugins.qgis.org/plugins/SRTM-Downloader/>) . A 2. ábrán a legyártott térképek láthatók, két mélységtartományban: 60-100 (piros – fehér) és 120-200 km (fekete-fehér) közötti hipocentrumban kipattant eseményekkel.

A strandlabdák mérete a mélységükkel arányosak, a 4. ábrán a 60 és 100 km-es mélységek között lévő földrengésekre leírt összefüggés látható.



4. ábra: Az importált fészekmechanizmusok strandlabdáinak méretezése a mélység függvényében történt. A fekete-fehér strandlabdák a 60 és 100 km közötti mélységben kipattant, úgynevezett felső mélységtartományban(angolul Upper Active Seismic Zone) lévő földrengésekhez tartoznak.

A strandlabdákat pedig kategorizált módon párosítottuk a földrengések helyéhez (pontjaihoz). Ehhez gyártottam egy “Name” nevű mezőt, amely a dőlésszög (strike), dőlésirány (dip) és csúszásirány (rake) mezőkből a következő képlettel töltöttem fel (ld. 5. ábra, szintén az alacsonyabb mélységekben kipattant rengések esetében).



5. ábra: A 60 – 100 km-es mélységtartományban kipattant események esetében a strandlabdák SVG – fájlainak nevét tartalmazó mező (Name) generálása az ábrán látható a képlettel történt. Miután ezeket létrehoztam, utána ezek alapján az értékek alapján sikerült kategorizálni az eseményeket és így hozzárendelni a megfelelő strandlabdának a képét az adott ponthoz.

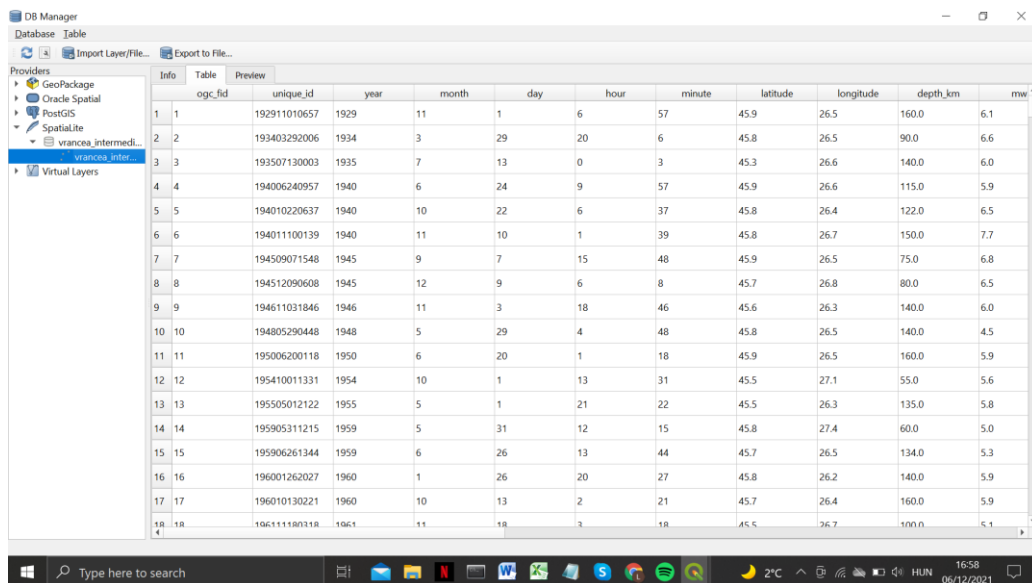
A 120 és 200 km-es mélységtartományban kipattant rengések esetében a képlet ugyanez, csak az “upper” előtagot a “lower” – szóra cseréltem.

B) Egy másik fontos aspektusa jelen munkának a meglévő katalógusokból egy egységes adatbázis létrehozása és ehhez kapcsolódó térbeli lekérdezések végrehajtása, az eredmények megjelenítése térképes és táblázatos formában. A QGIS-ben lehetőség van úgynevezett SpatiaLite - fájlok (https://docs.qgis.org/3.16/en/docs/training_manual/databases/spatialite.html) generálására. A SpatiaLite nyílt forráskódú könyvtár az SQLite – könyvtárnak (<https://www.gaia-gis.it/fossil/libspatialite/index>) egy térbeli kiterjesztése, és ami a legfontosabb, hogy egy önálló fájlban, méretkorlátozás nélkül, tárolható benne a létrehozott adatbázis.

Jelenleg a Romániai Földfizikai Kutatóintézet által újraszámított szeizmikus események fészekmechanizmusainak katalógusa a fő adatforrás

(<https://data.mendeley.com/datasets/mykx4gygy/3> (Radulian et al 2018, 2019, 2020). Külön munkalapban szerepelnek a kéregrengések (0 és 50 km közötti hipocentrummal rendelkező események), valamint a közepes fészekmélységű események (50.1 km-től 200 km-ig). A feszültséginverziók tekintetében az utóbbi csoportba tartozó földrengések a mérvadóak, így a továbbiakban az erre a munkalapra elvégzett lekérdezések kerülnek bemutatásra.

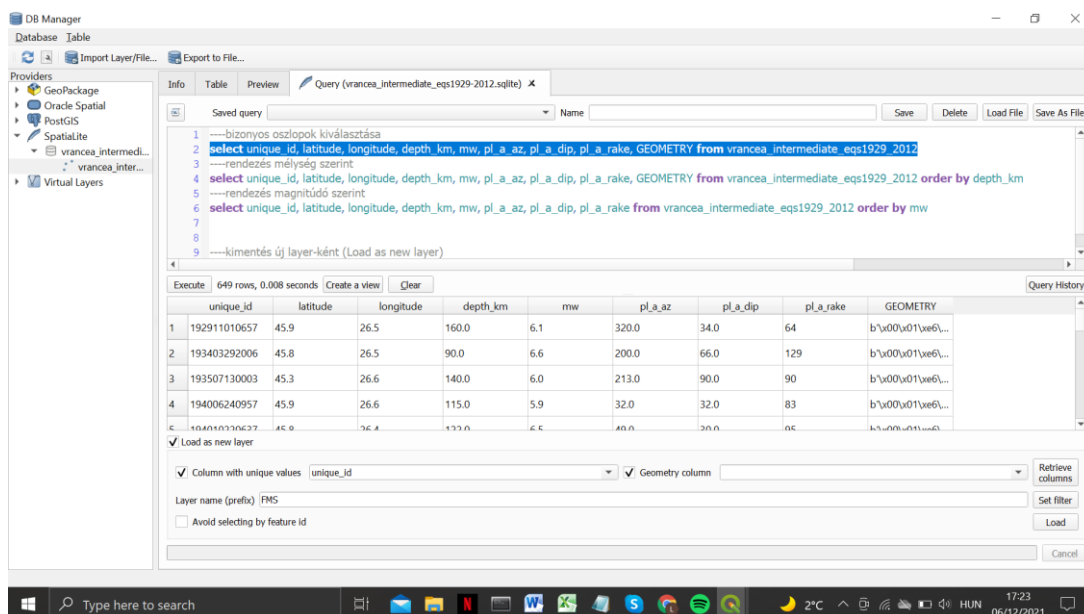
Még mielőtt magát a térbeli adatbázist létrehoztam QGIS-ben, a közepes hipocentrumú rengések munkalapját külön kimentettem CSV – formátumban, majd ezt importáltam QGIS-be. Az eseményeket alapvetően pontszerűnek tekinthetjük, azaz pontobjektumként kezeltem őket. (Ez a geometria megadásánál játszik fontos szerepet.) Az importálás után már lehetőség volt elmenteni ezt SpatiaLite – formátumban, majd behívni egy SpatiaLite-réteggént. Az így előállított tábla látható az 6. ábrán, már a Database Manager – panelban megjelenítve és beolvasva.



	ogc_fid	unique_id	year	month	day	hour	minute	latitude	longitude	depth_km	mw
1	1	192911010657	1929	11	1	6	57	45.9	26.5	160.0	6.1
2	2	193403292006	1934	3	29	20	6	45.8	26.5	90.0	6.6
3	3	193507130003	1935	7	13	0	3	45.3	26.6	140.0	6.0
4	4	194006240957	1940	6	24	9	57	45.9	26.6	115.0	5.9
5	5	194010220637	1940	10	22	6	37	45.8	26.4	122.0	6.5
6	6	194011100139	1940	11	10	1	39	45.8	26.7	150.0	7.7
7	7	194509071548	1945	9	7	15	48	45.9	26.5	75.0	6.8
8	8	194512090608	1945	12	9	6	8	45.7	26.8	80.0	6.5
9	9	194611031846	1946	11	3	18	46	45.6	26.3	140.0	6.0
10	10	194805290448	1948	5	29	4	48	45.8	26.5	140.0	4.5
11	11	195006200118	1950	6	20	1	18	45.9	26.5	160.0	5.9
12	12	195410011331	1954	10	1	13	31	45.5	27.1	55.0	5.6
13	13	195505012122	1955	5	1	21	22	45.5	26.3	135.0	5.8
14	14	195905311215	1959	5	31	12	15	45.8	27.4	60.0	5.0
15	15	195906261344	1959	6	26	13	44	45.7	26.5	134.0	5.3
16	16	196001262027	1960	1	26	20	27	45.8	26.2	140.0	5.9
17	17	196010130221	1960	10	13	2	21	45.7	26.4	160.0	5.9
18	18	196111180318	1961	11	18	3	18	45.5	26.7	100.0	5.1

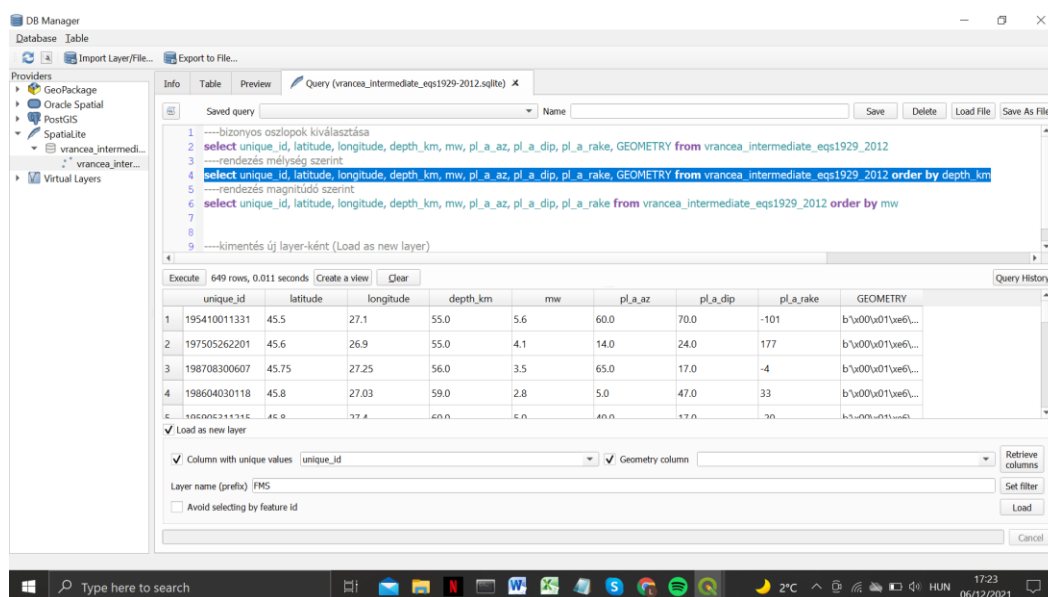
6. ábra: a Romániai Földfizikai Kutatóintézete által publikált relokalizált eseményeinek katalógusából látható egy részlet, már a SpatiaLite-ba beolvasott formában.

Ez a táblázat szűkítésre került, hogy elősegítse a felhasználói oldalról a minél egyszerűbb lekérdezéseket? A következőekben csak a következő adatokra volt szükségem az események kiértékeléséhez: egyedi azonosító (unique_id), szélesség és hosszúság és mélység (latitude, longitude, depth_km), momentum magnitúdó (mw) és az egyik síkja a fészekmechanizmus-megoldásoknak (pl_a_az, pl_a_dip, pl_a_rake – alapvetően két síkot számítanak ki), valamint a geometriát leíró adatok (GEOMETRY). A lekérdezés eredményének egy kivágata megtekinthető a 7. ábrán.

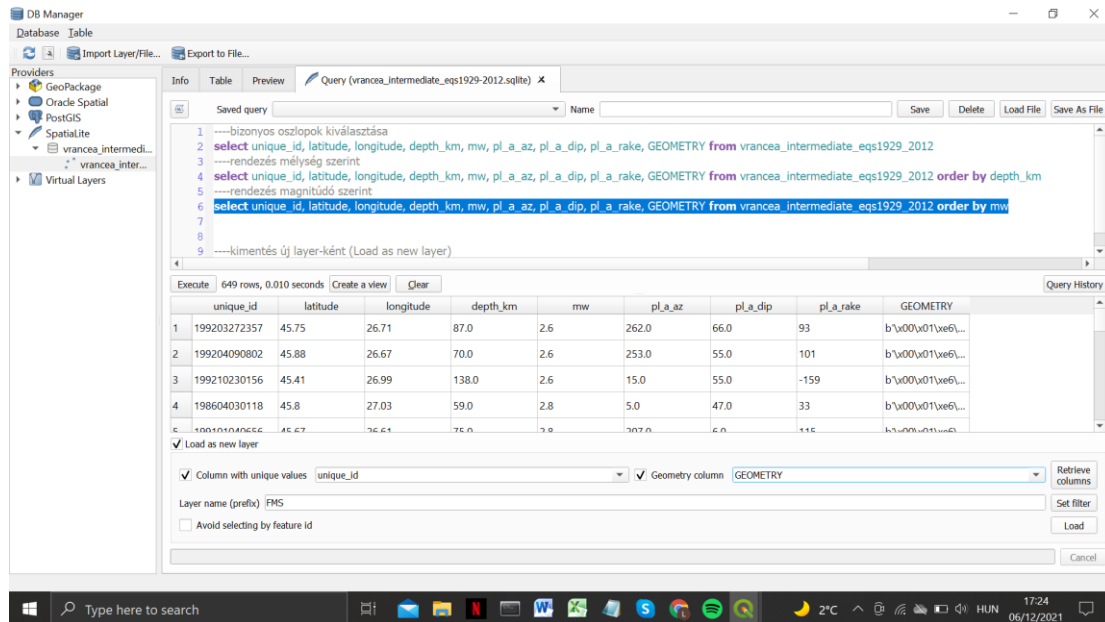


7. ábra: A leszűkített adatbázis létrehozása és néhány kapcsolódó lekérdezés . A kijelölt rész jelzi a megfelelő mezők kiválasztását, alatta mélység, majd az alatt pedig momentum magnitúdó szerint rendeztem a sorokat.

A 7. ábrán már nemcsak a szűkített adattábla lekérése olvasható, de ezeket az eseményeket sorba rendeztem momentum magnitúdójuk nagysága és a fészekmélységek szerint is (alapesetben az ORDER BY növekvő sorrendben írja ki az adatsorokat, de az ASCENDING/DESCENDING kulcsszavak beírásával mi közvetlenül meg is határozhatjuk a sorrend típusát is). A 8. és 9. ábra ezen lekérdezések első 4 sorát mutatja be.



8. ábra: Mélység szerint történő sorba rendezésének felső 4 sora. Ez alapján kijelenthető, hogy a legkisebb mélység az 55 km volt, utána az 56 km, majd az 59 km.



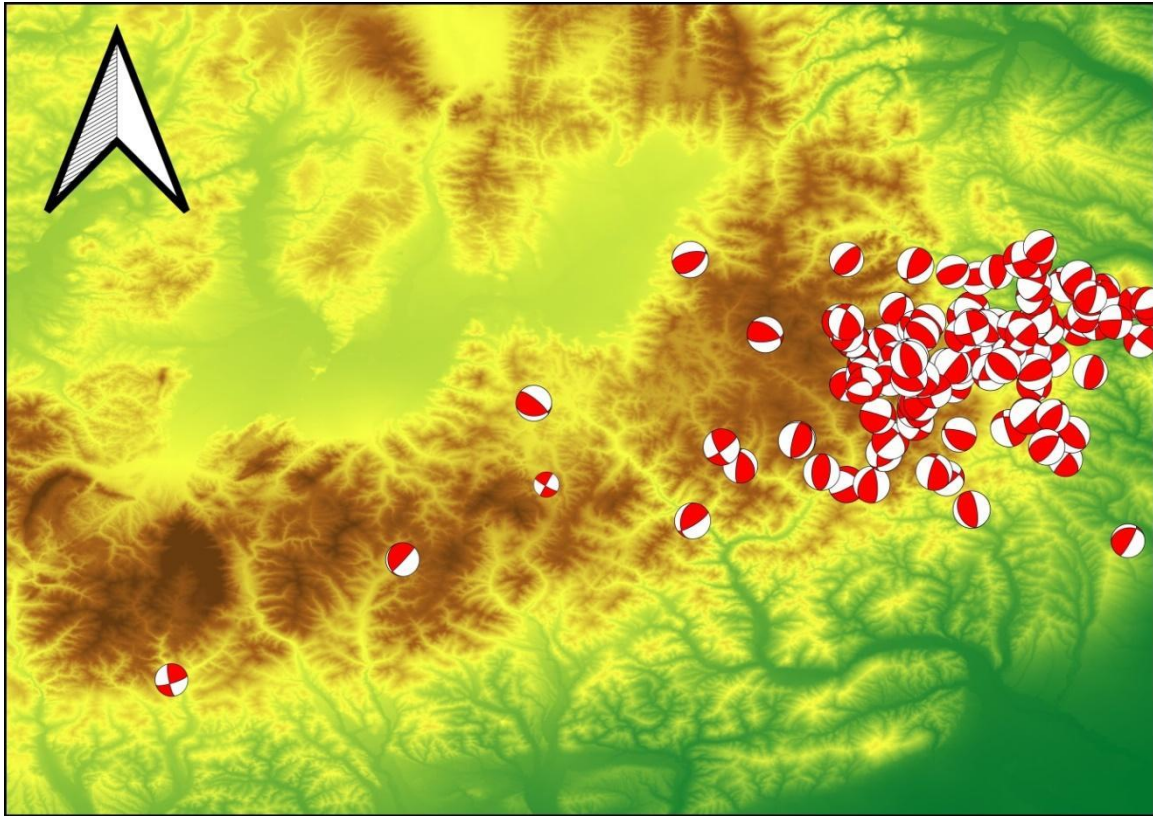
9. ábra: Magnitúdó szerint történő sorba rendezésének felső 4 sora . A térségben 2.6 –os magnitúdónál kisebb rengések nem fedezhetők fel ebben a közepes fészekmélységű földrengések között.

Az új adattáblát ezután elmentettem egy új réteggént, FMS néven (lásd 6-8. ábrák) és meg is jelenítettem a benne lévő pontokat. A további lekérdezések eredményeit már nemcsak táblázatos formában, hanem új rétegekként térképes nézetben is megjelenítettem. Ehhez külön még egyszer kimentettem a táblát, FMS2 néven. A továbbiakban a lekérdezéseket és a különböző matematikai műveleteket erre az adattáblára végeztem el.

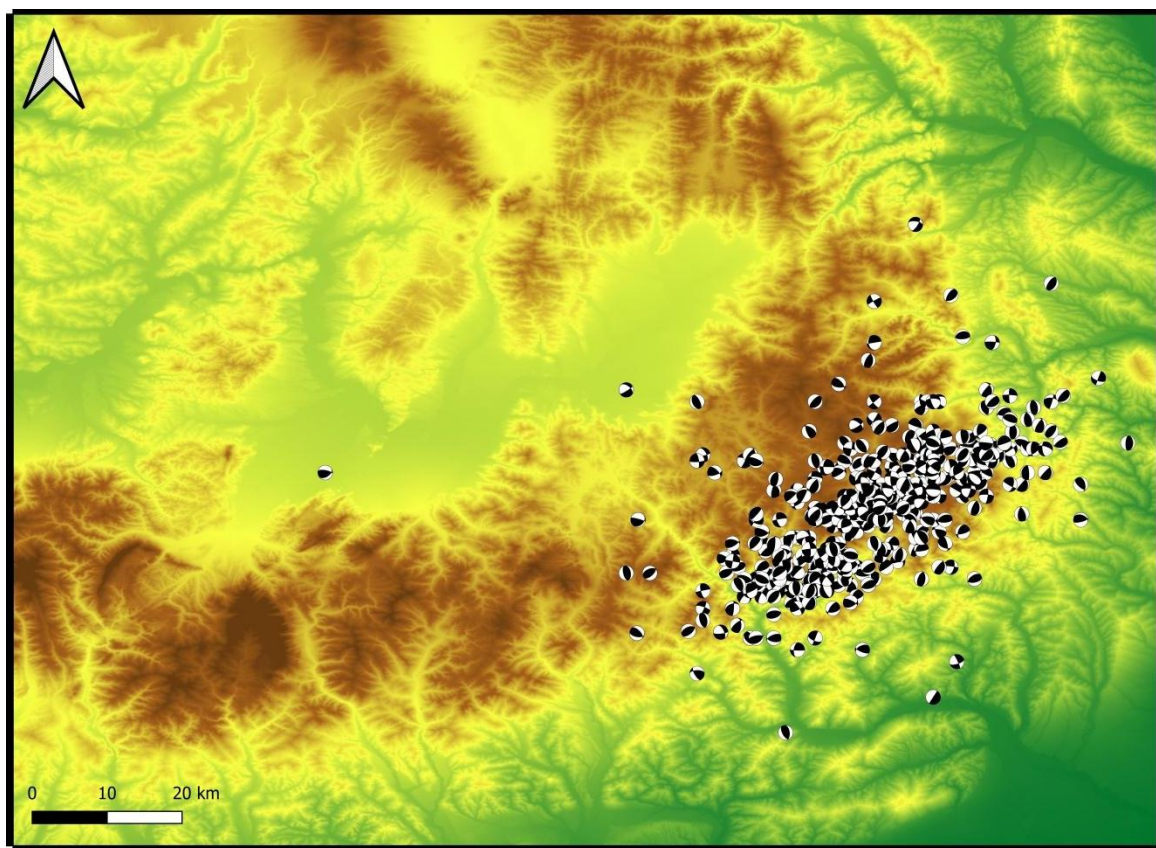
4. Eredmények

4.1. Felhasznált adatok ábrázolása QGIS-ben

A 10. és 11. ábrán a 60 és 100 km (piros-fehér strandlabdák) illetve a 120 és 200 km közötti mélységben (fekete-fehér strandlabdák) kialakult földrengések fészekmechanizmus-megoldásai láthatóak, QGIS Layout Manager-éből kimentett térképeken.



10. ábra: A felső zónában (60 – 100 km) kipattant szeizmikus események fészekmechanizmusai. A strandlabdák nagysága a mélységgel egyenes arányban áll.



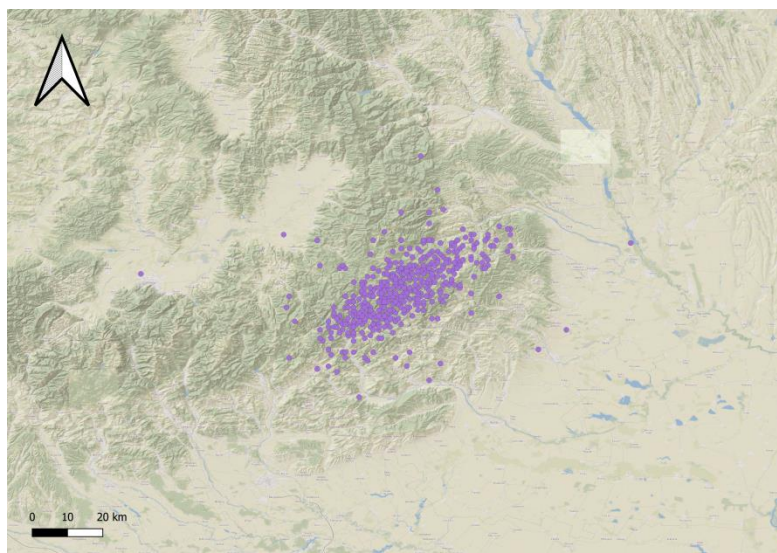
11. ábra: Az alsó zónában (120 – 200 km) kipattant szeizmikus események fészekmechanizmusai. A strandlabdák nagysága a mélységgel egyenes arányban áll.

Az események eltérő száma – kisebb mélységben néhány 10, a nagyobb mélységben néhány száz – miatt különböző összefüggéseket alkalmaztam a strandlabdák nagyságára, de mindkét esetben a hipocentrumukkal áll arányossági kapcsolatban.

4.2. Térbeli lekérdezések SpatiaLite-ban

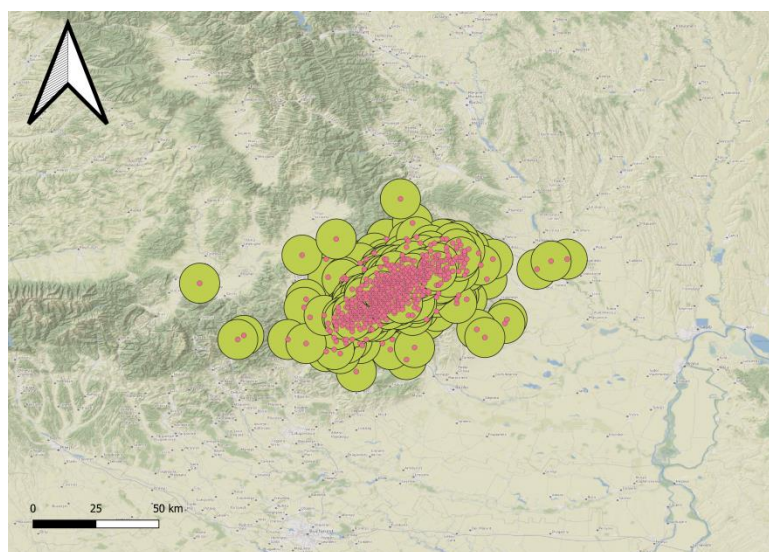
Milyen szempontok szerint kerültek szűrésre az eseményeket?

1. lekérdezés: Mely események pattantak ki 100 km-nél nagyobb mélységben? A térképes megjelenítés mellett sorba rendeztem őket erősségük szerint is. A legkisebb nagyságú esemény 2,6-os magnitúdójú volt, a szelektált események a 12. ábrán láthatóak.

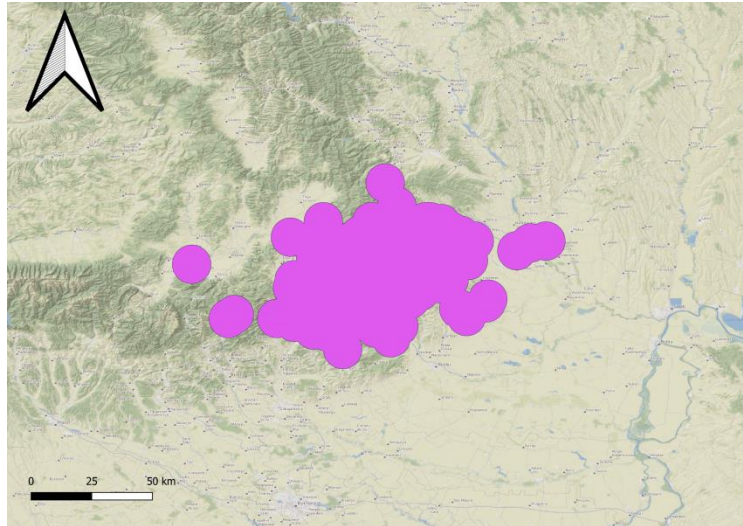


22. ábra: a 100 km-nél mélyebben kipattant földrengések lekérdezés utáni megjelenítése.

2-4. lekérdezés: Ezek a lekérdezések, műveletek arra irányultak, hogy kirajzoljam megközelítőleg a szeizmogén zónát. Ehhez először egyesítettem az `ST_UNION` parancs segítségével a földrengésekhez tartozó pontokat, amelyekből egy úgynevezett Multipont – objektum keletkezett. Utána erre az alakzatra övezetet generáltam az `st_buffer` - paranccsal, 0.1-es “sugárral” (azaz 0.1 fokos körben terjesztettem ki az övezetet). A multipontok övezeteinek generálása után ezeket egyesítve kaptam meg a formáját ennek a szeizmikusan aktív térségnek (természetesen ez csak első közelítésként vehető figyelembe). A 13. és 14. ábra illusztrálja ezeknek a kalkulációknak az eredményeit.

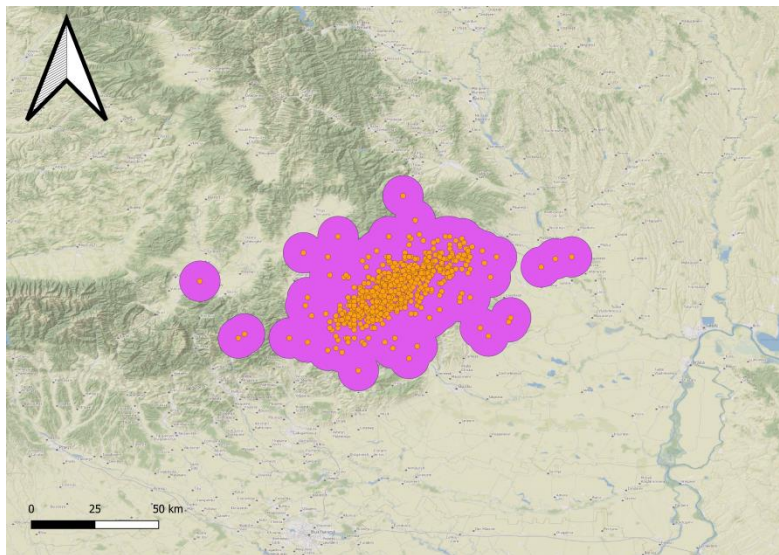


13. ábra: Az unió és az övezetgenerálás műveletének eredménye, a lila pontok a földrengések helyeit, a zöld körök pedig a hozzájuk tartozó övezetet jelölik.



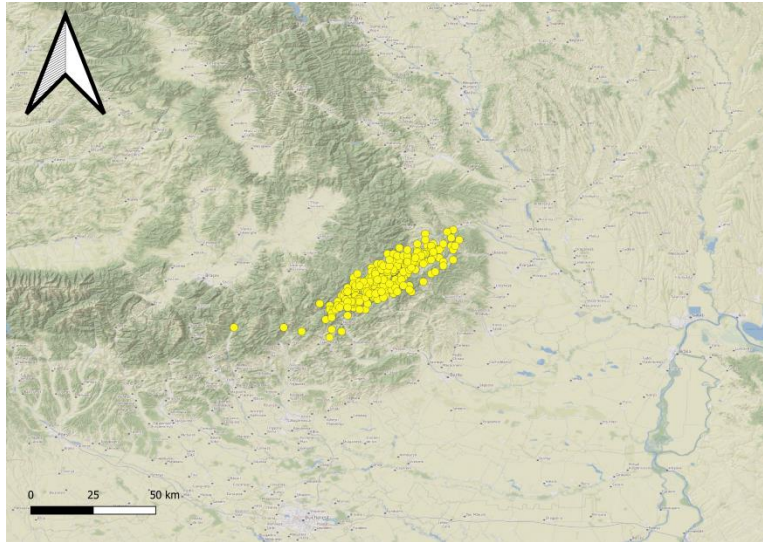
14. ábra: A multipontokra létrehozott övezetek egyesítéséből létrejött alakzat (lila). Ez egy első közelítésként jól demonstrálja a szeizmikusan aktív területet.

A tanulmányozott eseményekkel együttes megjelenítése látható a 15. ábrán.



15. ábra: A generált szeizmogén zóna (lila) és a területén kipattant földrengések (narancssárga) helyei.

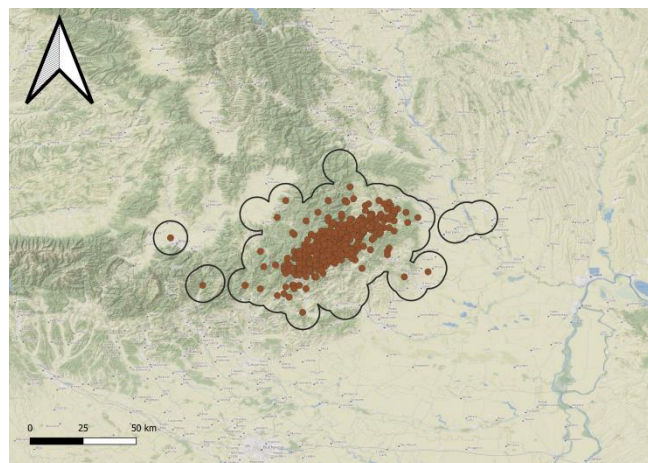
5-6. lekérdezés: Az egyedi azonosítók alapvetően a földrengések időpontjait indikálják. Ezek a lekérdezések a 21. században kipattant szeizmikus események kigyűjtésére és megjelenítésére szolgáltak. A 16. ábrán ezen lekérdezések eredményei láthatóak és jól megfigyelhető, hogy egy – két izolált földrengést leszámítva ezek mind a Kárpátkanyar külső ívére fókuszálnak.



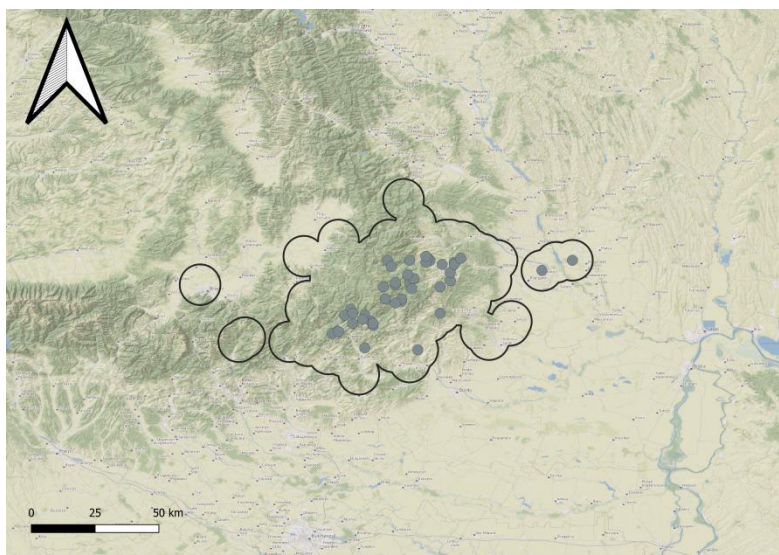
16. ábra: A földrengésekhez tartozó egyedi azonosítók időbeli információt foglalnak magukba. Ez alapján kérdeztem le azokat a szeizmikus eseményeket, amelyek 2000. január 1. és 2012. 12. 31. között (citromsárga pontok).

Ezek hipocentruma 64 és 171 km között változott, tehát elég széles mélységtartományt ölel fel.

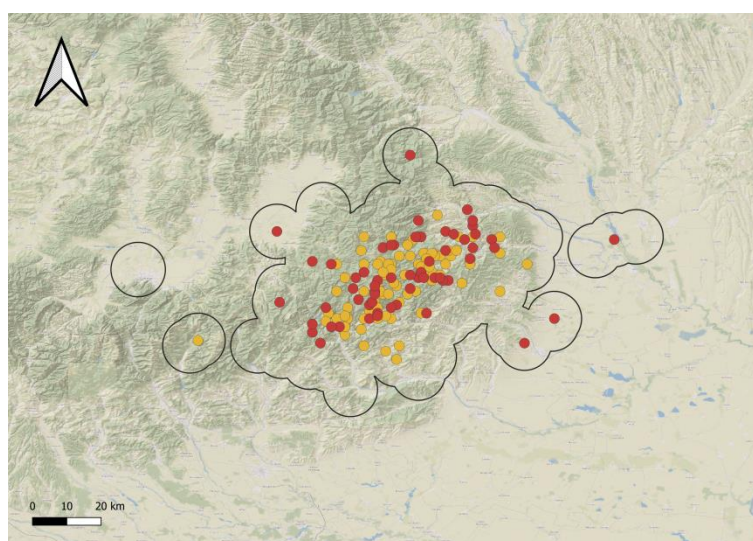
7-9. lekérdezés: Ezek a lekérdezések a fészekmechanizmus-megoldásokra, azokon belül is a csúszásirányokra koncentráltak. A csúszásirány nagysága (pl. Cronin, 2010) leírja, milyen vetőtípushoz kapcsolható az adott földrengés. A 17-19. ábrán a feltolódásokhoz, normál vetőkhöz és oldaleltolódásokhoz kapcsolható események kerültek megjelenítésre.



17. ábra: Feltolódásokhoz kapcsolható földrengések (barna) a fészekmechanizmus-megoldásuk csúszásirányai alapján. Körülötte a fekete kontúr a már korábban létrehozott szeizmogén zónát jelöli.



18. ábra: Normál vetőkhöz kapcsolható földrengések (szürke körök) a fészekmechanizmus-megoldásuk csúszásirányai alapján. Körülötte a fekete kontúr a már korábban létrehozott szeizmogén zónát jelöli.



19. ábra: Oldalelmozdulásokhoz kapcsolható földrengések (piros és sárga pontok) a fészekmechanizmus-megoldásuk csúszásirányai alapján. A sárga pontok mutatják a 90-180 fokos csúszásiránnyal rendelkező, jobbos oldalelmozdulásokat, piros pontok pedig a -180 és -90 fok közötti, balos oldaleltolódásokat. Körülötte a fekete kontúr a már korábban létrehozott szeizmogén zónát jelöli.

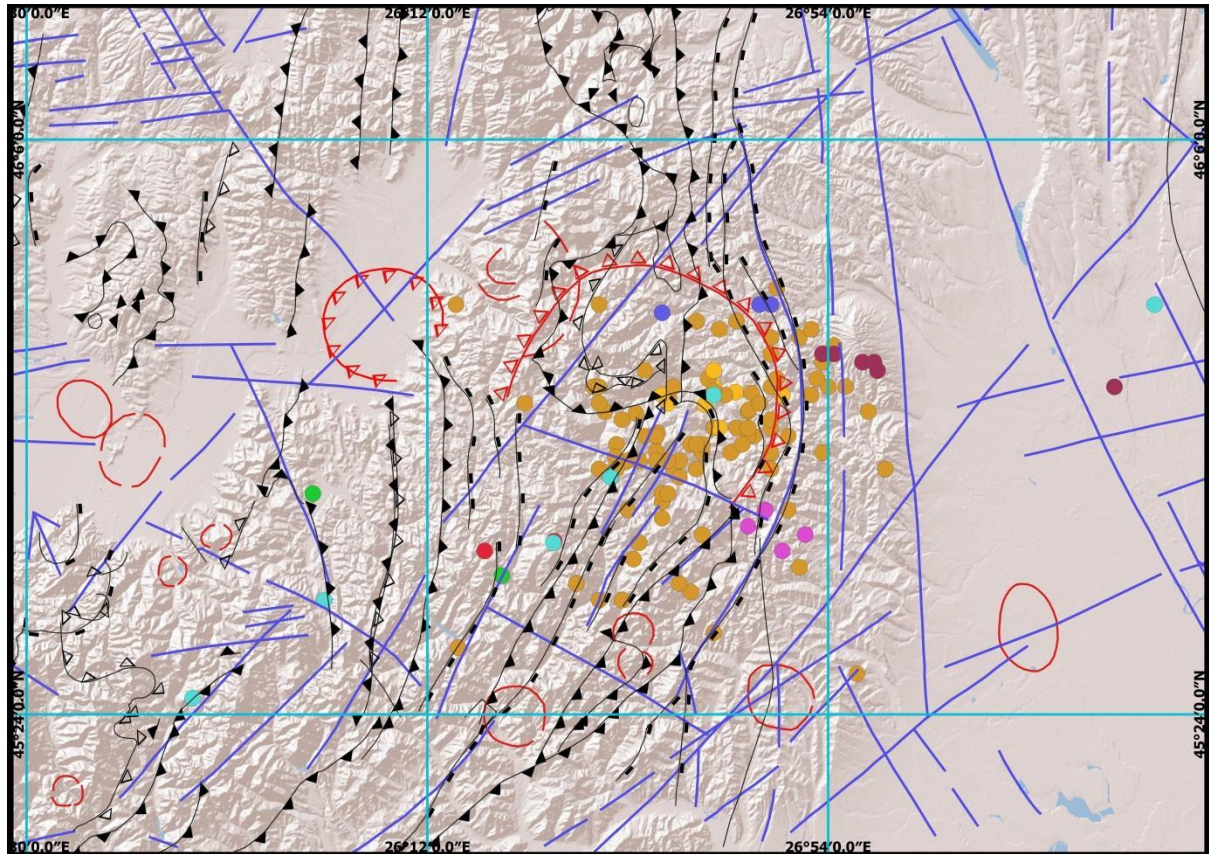
4.3. “Teljesen” automatizált klaszterező algoritmusunk által generált csoportok

Ez az alfejezet már tulajdonképpen korábbi eredményeket mutat be, csak a kontextusba helyezés miatt kerül sor bemutatására.

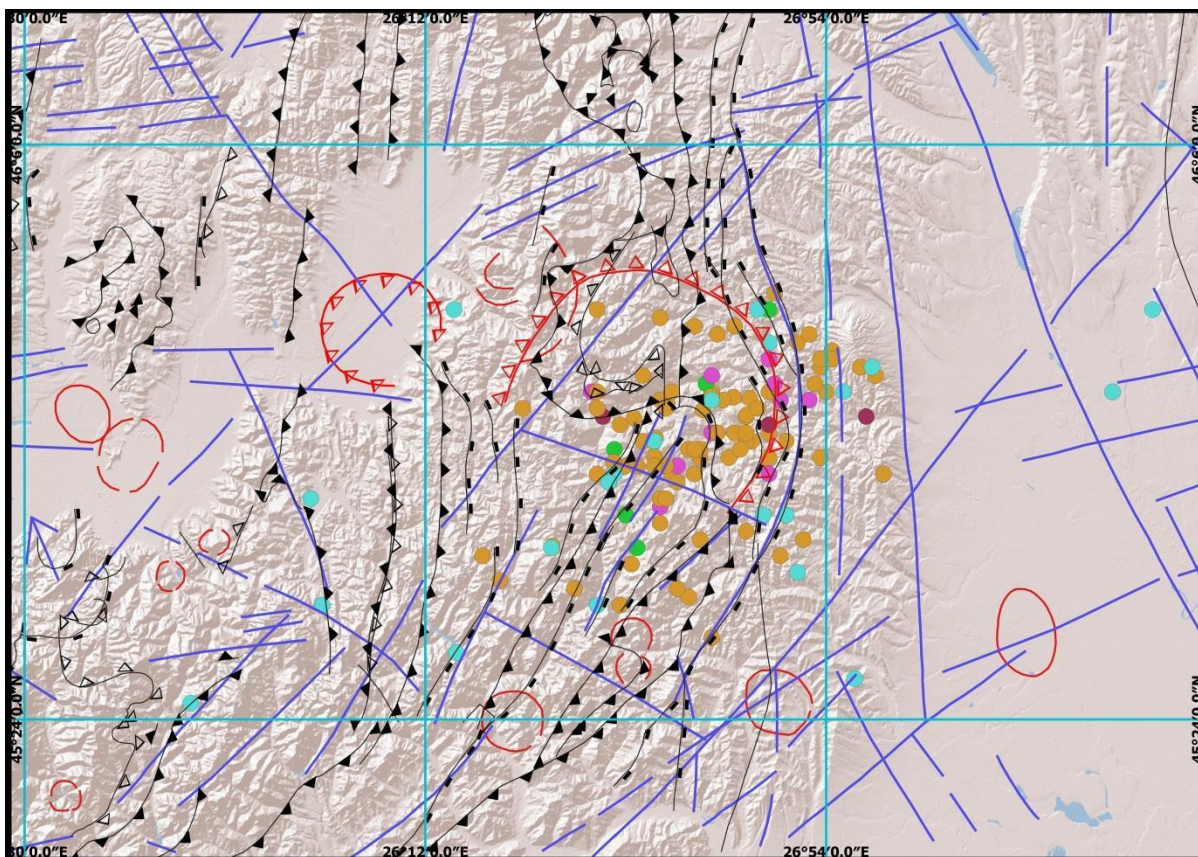
A 20-23. ábrákon az általunk fejlesztett “teljesen” automatizált klaszterező algoritmus eredmények tekinthetőek meg. Ezeket 3- illetve 4 dimenzióban hajtottuk végre. 3 dimenzió esetén a földrengésekhez kapcsolódó helyadatok, tehát a szélesség – és hosszúság koordináták, valamint azok fészekmélysége lett figyelembe véve. A negyedik dimenzió a

földrengések fészekmechanizmusaihoz kapcsolódó csúszásirány, amely alapján feltételezhető, hogy egy adott földrengés milyen típusú földtani szerkezethez kötődően alakult ki.

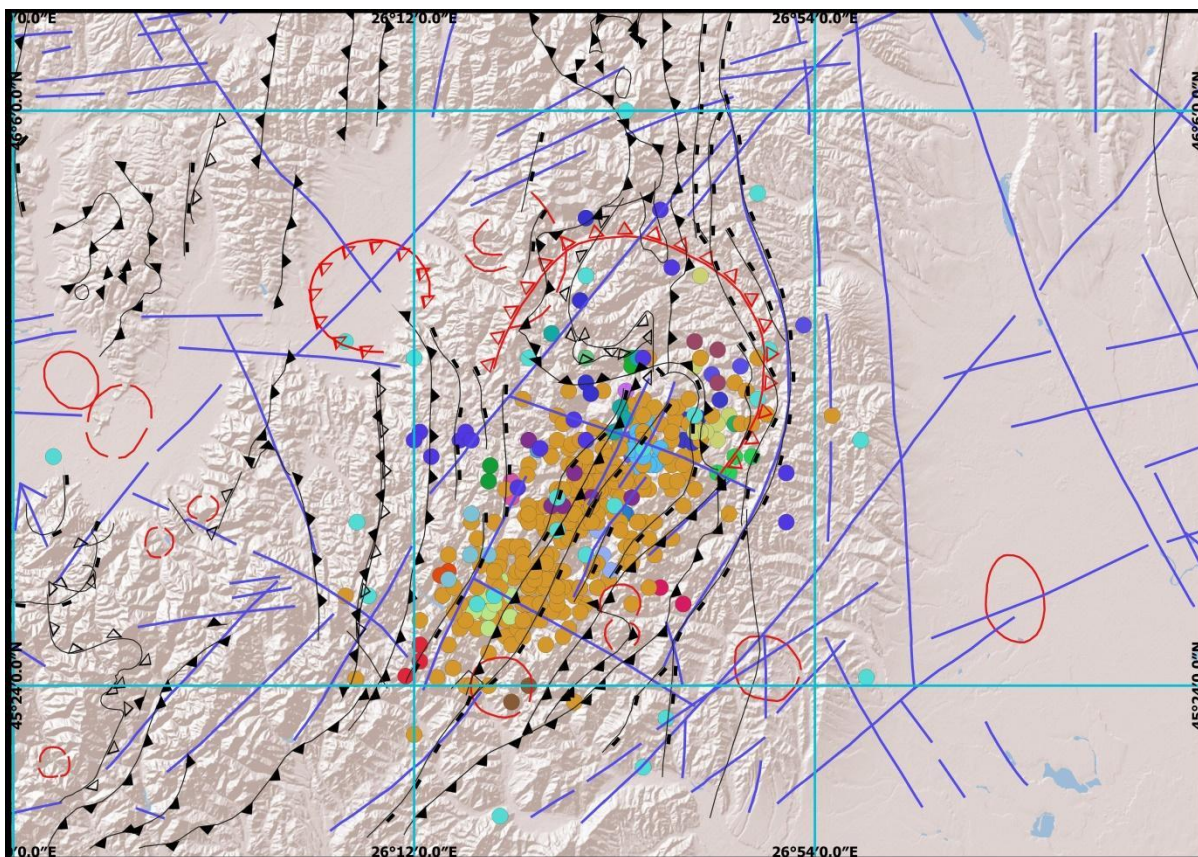
A 20. és 21. ábra a 60 és 100 km – es mélységintervallumban kipattant földrengések 3-, valamint 4 dimenzióban történt klaszterezés eredményeit mutatják.



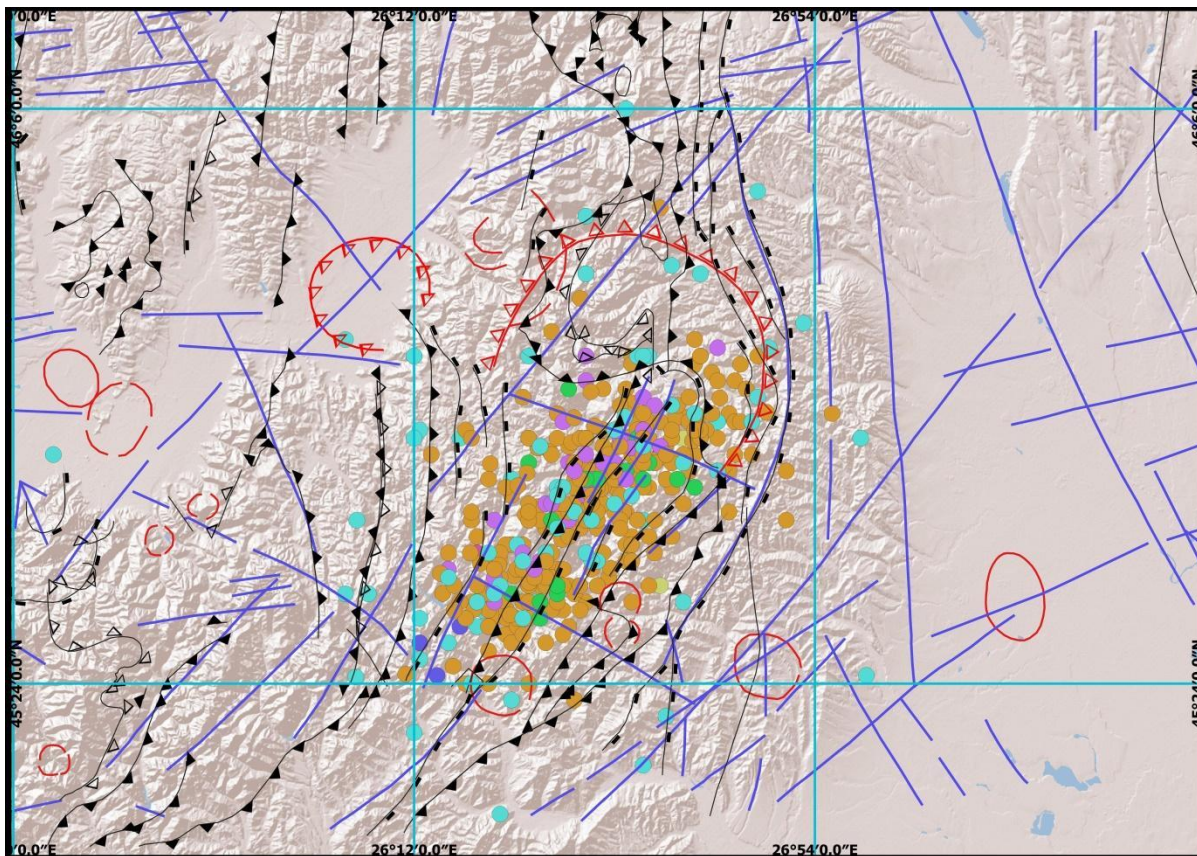
20. ábra: A földrengések térbeli adatai alapján történt klaszterezés eredményeként kapott alcsoportok a feszültség-inverziókhoz a felső mélységtartományban. Különböző színek a különböző csoportokat jelölik.



21. ábra: A földrengések térbeli adatai illetve a csúszásirányok alapján történt klaszterezés eredményeként kapott alcsoportok a feszültség-inverziókhöz a felső mélységtartományban. Különböző színek a különböző csoportokat jelölik.



22. ábra: A földrengések térbeli adatai alapján történt klaszterezés eredményeként kapott alcsoportok a feszültség-inverziókhöz az alsó mélység-tartományban. Különböző színek a különböző csoportokat jelölik.



23. ábra: A földrengések térbeli adatai illetve a csúszásirányok alapján történt klaszterezés eredményeként kapott alcsoportok a feszültség-inverzióhoz az alsó mélységtartományban. Különböző színek a különböző csoportokat jelölik.

Azon földrengéscsoportok esetén ahol a földrengések száma nem érte el az ötöt, nem hajtottam végre a feszültség-inverziót az eredmények várhatóan alacsony pontossága és megbízhatósága miatt.

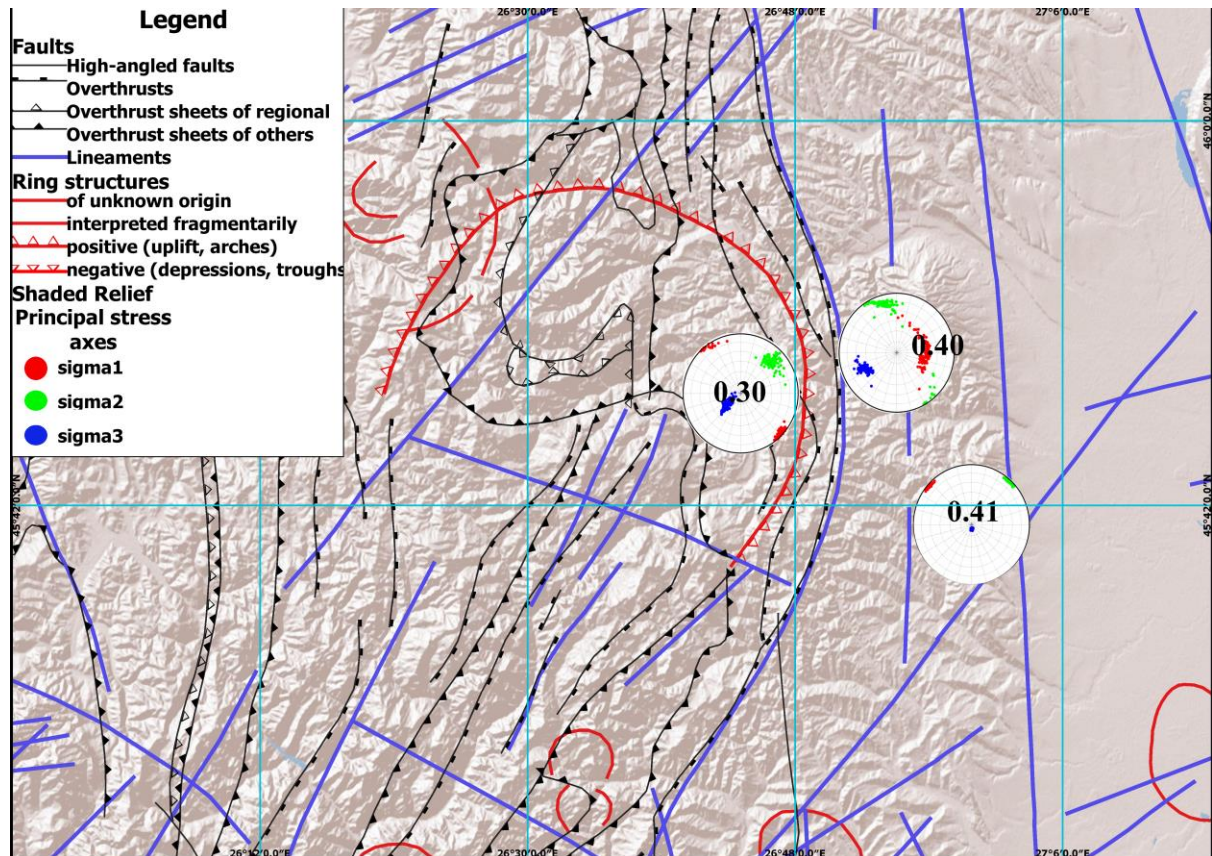
4.4. Feszültség-inverziók eredménye QGIS-ben

A szeizmotektonikai kiértékelés szempontjából a legfontosabb eredmény a főfeszültségekre kapott úgynevezett sztereogramok, amely a főfeszültség-tengelyek azimutját és dőlését mutatja be, így azok egymáshoz viszonyított helyzete is vizsgálható és megállapítható, hogy milyen tektonikai szerkezetek és milyen stílus jellemzi az adott területet.

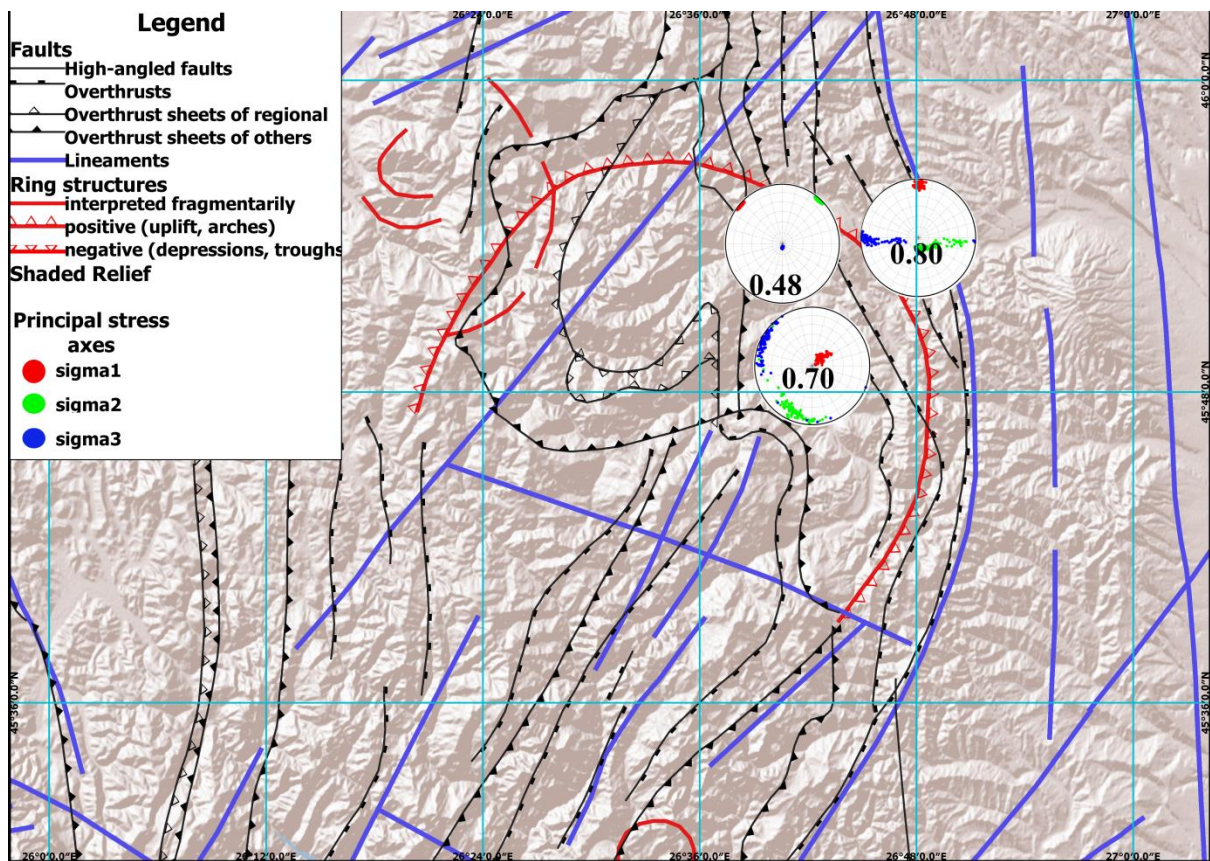
Egy tengelyhez több pont tartozik, ezek a realizációk során kerültek kiszámításra, így azok bizonytalanságát is láthatjuk. Az azimut azt mutatja meg, hogy melyik irányba tart a maximális horizontális kompressziós vagy extenziós feszültségmező. Ennek értéke 0 és 360 fok között változik. A dőlés (angolul plunge) a főfeszültség-tengelyek vertikális helyzetéhez való viszonyát írja le, ez a szög 0-tól 90 fokig vehet fel értéket, ahol 90 fok a tökéletesen

függőleges helyzetet jelenti. Kívülről befelé nő ennek értéke, és amelyik főfeszültség közelebb van a sztereogram közepéhez, az a leginkább vertikális helyzetű.

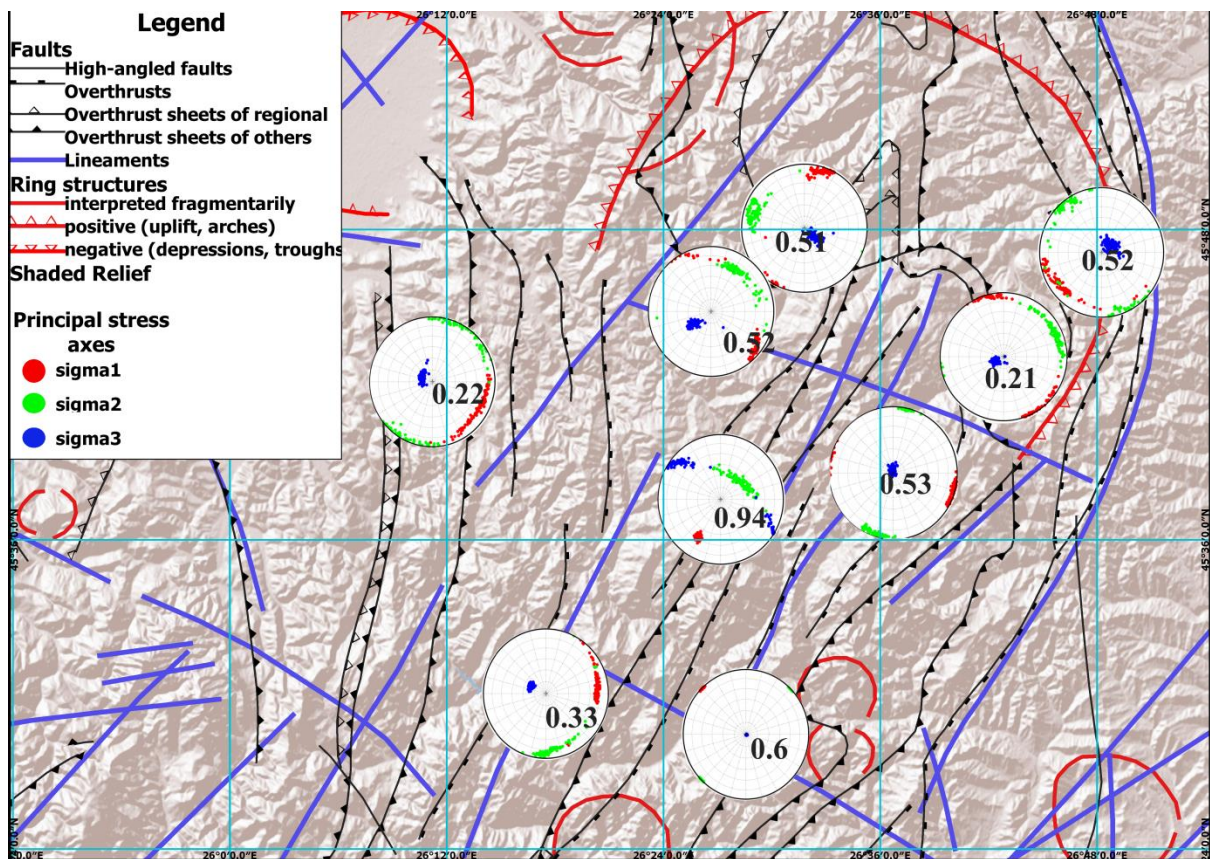
A következő négy ábrán (24-27. ábra) a 60 és 100 km közötti térségben a 3- és 4D-ben elvégzett, valamint a 120 és 200 km – es mélységtartományban elvégzett számítások során kapott sztereogramok láthatóak.



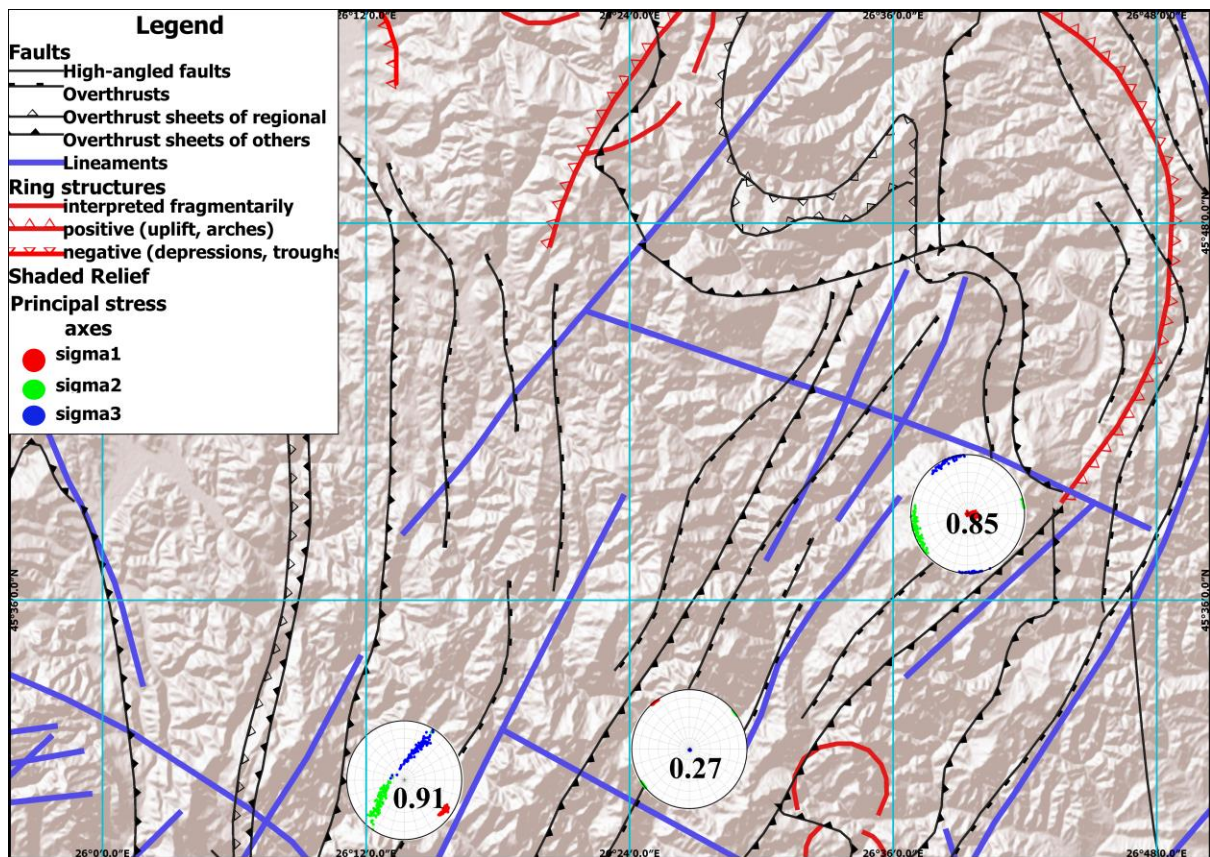
24. ábra: a felső mélységtartományban a 3D-s klaszterezéssel létrehozott, ötnél több eseménnyel rendelkező alterületekre elvégzett feszültséginverziók eredményei. A sztereogram mutatja a főfeszültség-tengelyek dőlésének és azimutjának bizonytalanságát, a piros szín a σ_1 , zöld a σ_2 és kék a σ_3 – tengelyt indikálják.



25. ábra: a felső mélységtartományban a 4D-s klaszterezéssel létrehozott, ötnél több eseménnyel rendelkező alterületekre elvégzett feszültség-inverziók eredményei. A sztereogram mutatja a főfeszültség-tengelyek dőlésének és azimutjának (helyzeteiknek) bizonytalanságát, a piros szín a σ_1 , zöld a σ_2 és kék a σ_3 – tengelyt indikálja.



26. ábra: az alsó mélységtartományban a 3D-s klaszterezéssel létrehozott, ötnél több eseménnyel rendelkező alterületekre elvégzett feszültség-inverziók eredményei. A sztereogram mutatja a főfeszültség-tengelyek dőlésének és azimutjának (helyzeteiknek) bizonytalanságát, a piros szín a σ_1 , zöld a σ_2 és kék a σ_3 – tengelyt indikálják.



27. ábra: az alsó mélységtartományban a 4D-s klaszterezéssel létrehozott, ötnél több eseménnyel rendelkező alterületekre elvégzett feszültség-inverziók eredményei. A sztereogram mutatja a főfeszültség-tengelyek dőlésének és azimutjának (helyzeteiknek) bizonytalanságát, a piros szín a σ_1 , zöld a σ_2 és kék a σ_3 – tengelyt indikálják.

A piros szín jelöli a legnagyobb, σ_1 , a zöld a második legnagyobb, a σ_2 és kék a legkisebb, σ_3 főfeszültség-tengelyt. A sztereogramok legnagyobb részén a kék színnel jelölt σ_3 rendelkezik közel vertikális dőléssel, de néhány esetben a piros σ_1 dőlése közel 90 fok (ld. 25. és 27. ábra).

A főfeszültség-tengelyekhez tartozó legjobb azimutális és dőlési szögértékek illetve a legjobban illeszkedő R-szám egy MAT fájlban kerül tárolásra, ez alkalmas a számszerű kiértékelésre és a következő fejezetben látható maximális horizontális feszültségirány – térképek elkészítéséhez innen nyertem ki az adatokat.

4.4. Feszültségmező orientációja és uralkodó tektonikai szerkezetek

A sztereogramok interpretációja során a (Zoback, 1992) publikációban szereplő táblázat volt a segítségemre. Ennek magyar fordítását lehet megtekinteni az 1. táblázatban.

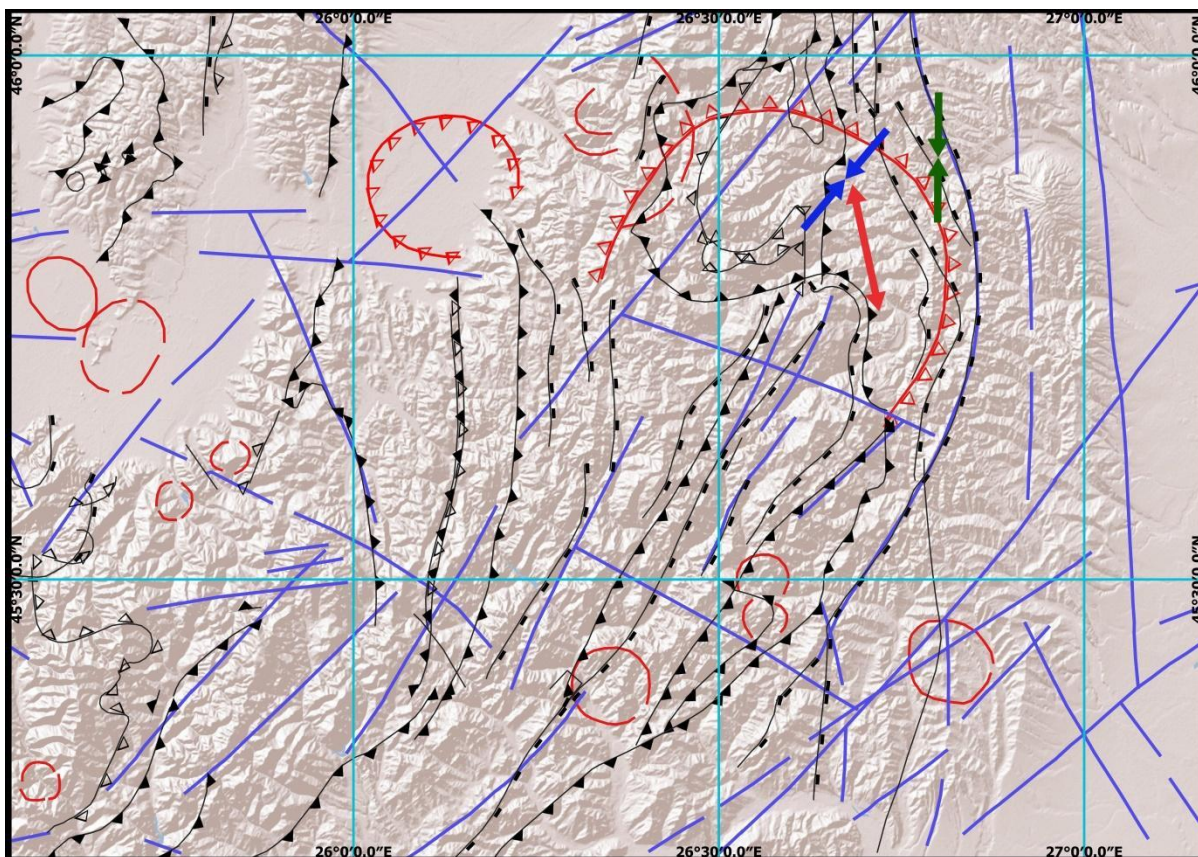
Ha tudjuk, hogy melyik a legnagyobb dőléssel rendelkező főfeszültség-tengely, akkor a táblázatban megkereshetjük a hozzá tartozó azimutális – vagyis a leginkább horizontális – főfeszültség-tengelyt. Az 1. táblázatban (Zoback, 1992) publikációban megtalálható klasszifikáció magyar fordítása látható.

1. táblázat: Zoback, 1992 alapján meghatározott uralkodó tektonikus rezsim a három főfeszültség-tengely egymáshoz viszonyított helyzete alapján.

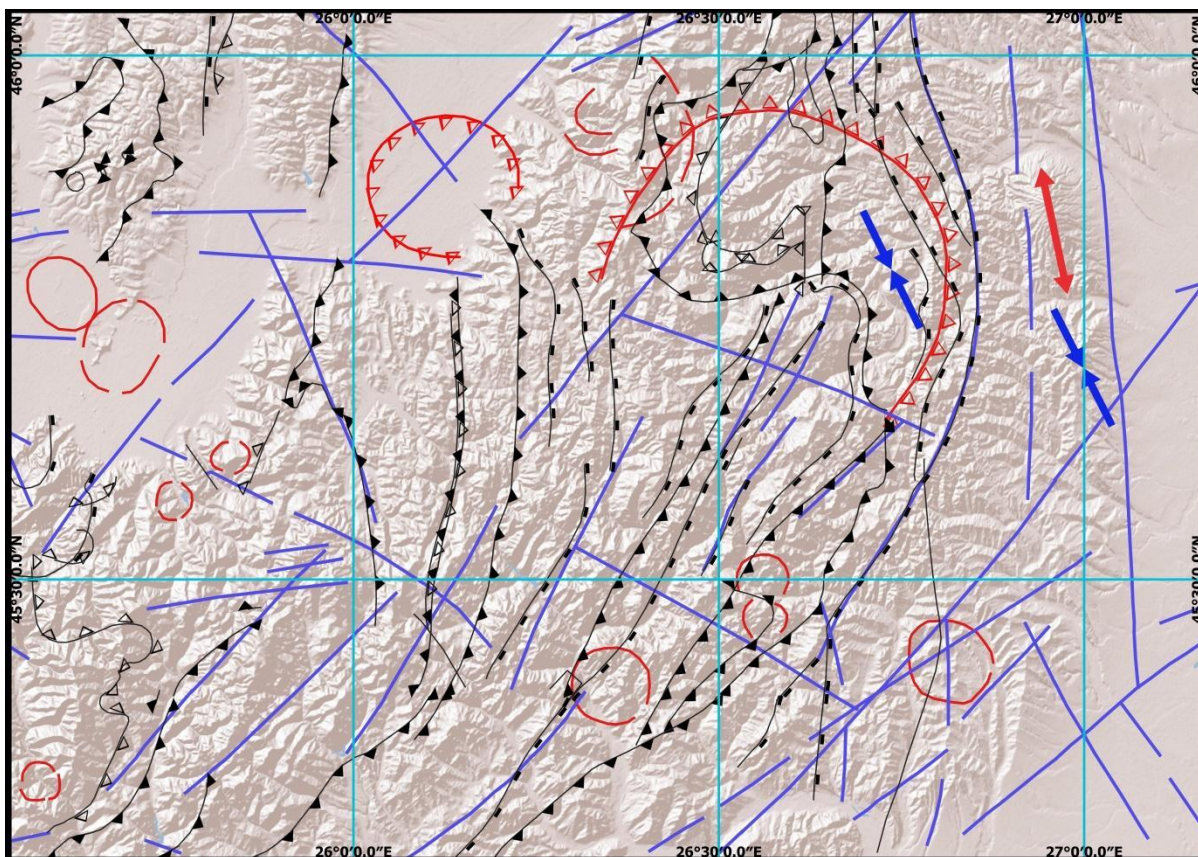
σ_1 -tengely	σ_2 -tengely	σ_3 -tengely	S_{hmax} (azimut)	Rezsim
dőlés $> 52^\circ$	dőlés $< 35^\circ$		σ_2	normál vető
$40^\circ < \text{dőlés} < 52^\circ$	dőlés $< 20^\circ$		$\sigma_3 + 90^\circ$	transztenzió
dőlés $< 40^\circ$	dőlés $> 45^\circ$	dőlés $< 20^\circ$	$\sigma_3 + 90^\circ$	oldalelmozdulás
dőlés $< 20^\circ$	dőlés $> 45^\circ$	dőlés $< 40^\circ$	σ_1	oldalelmozdulás
dőlés $< 20^\circ$	$40^\circ < \text{dőlés} < 52^\circ$		σ_1	transzpresszió
dőlés $< 35^\circ$	dőlés $> 52^\circ$		σ_1	feltolódás

A 4.4. fejezetben már említett MAT – formátumú eredménytábla lesz a segítségünkre abban, hogy számszerűsítsük a legnagyobb dőlésű főfeszültség-tengelyhez tartozó horizontális tengelyt.

A 28. és a 29. ábra a 60 és 100 km közötti térségben elvégzett feszültséginverziók interpretációját mutatja a három- illetve négydimenziós csoportosítás után kapott alrészekre. A nyilak színét a domináns tektonikus szerkezet határozza meg, irányítottságát pedig az ezekhez tartozó maximális horizontális kompresszió iránya (azaz a leginkább vízszintes helyzetű főfeszültség-tengely). Kék szín a feltolódást, piros szín a normál vetőt és zöld szín az oldaleltolódást jelképezi.



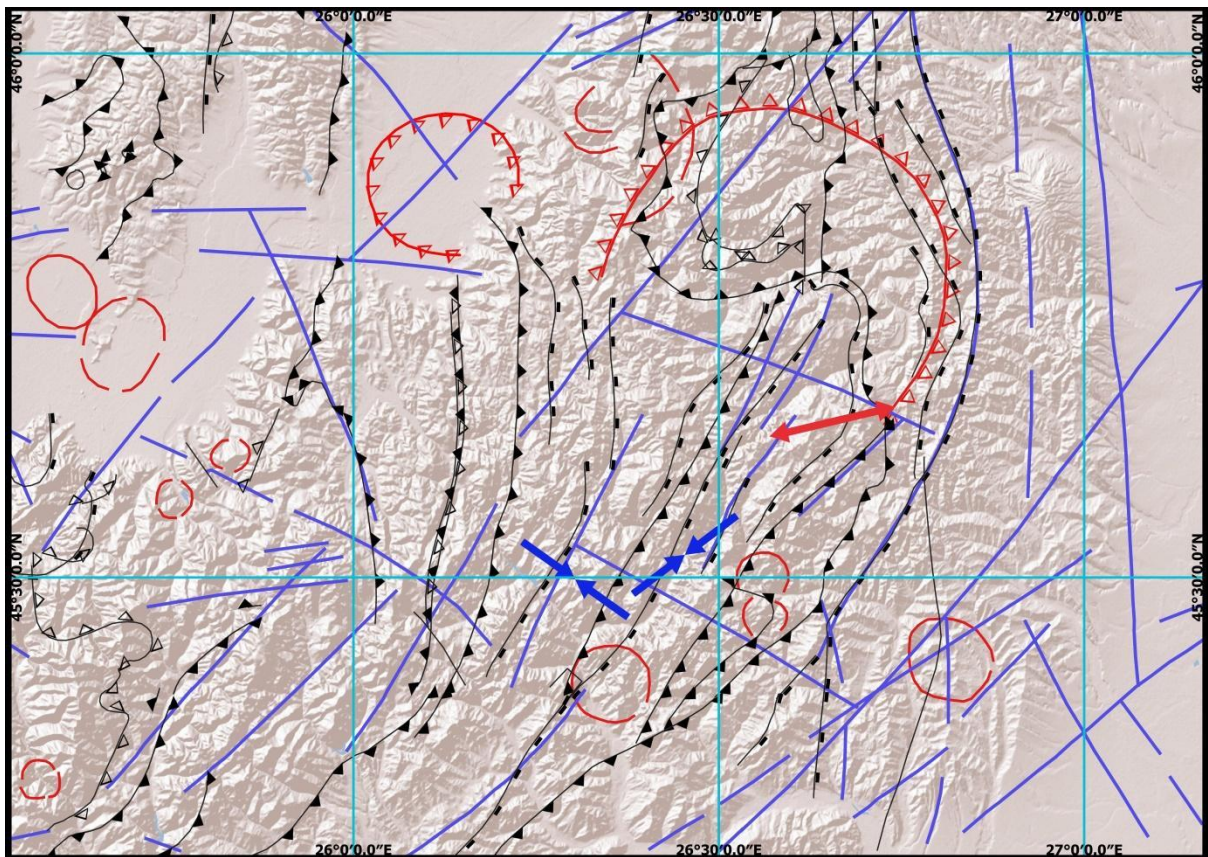
28. ábra: a 60 és 100 km közötti mélységtartományban lévő, 3D-s klaszterezés után kapott alterületekre értelmezett jellemző feszültségirányok és tektonikus rezsimek. Kék színű nyíl jelzi a feltolódásokat, zöld az oldalelmozdulásokat és piros a normál vetőket.



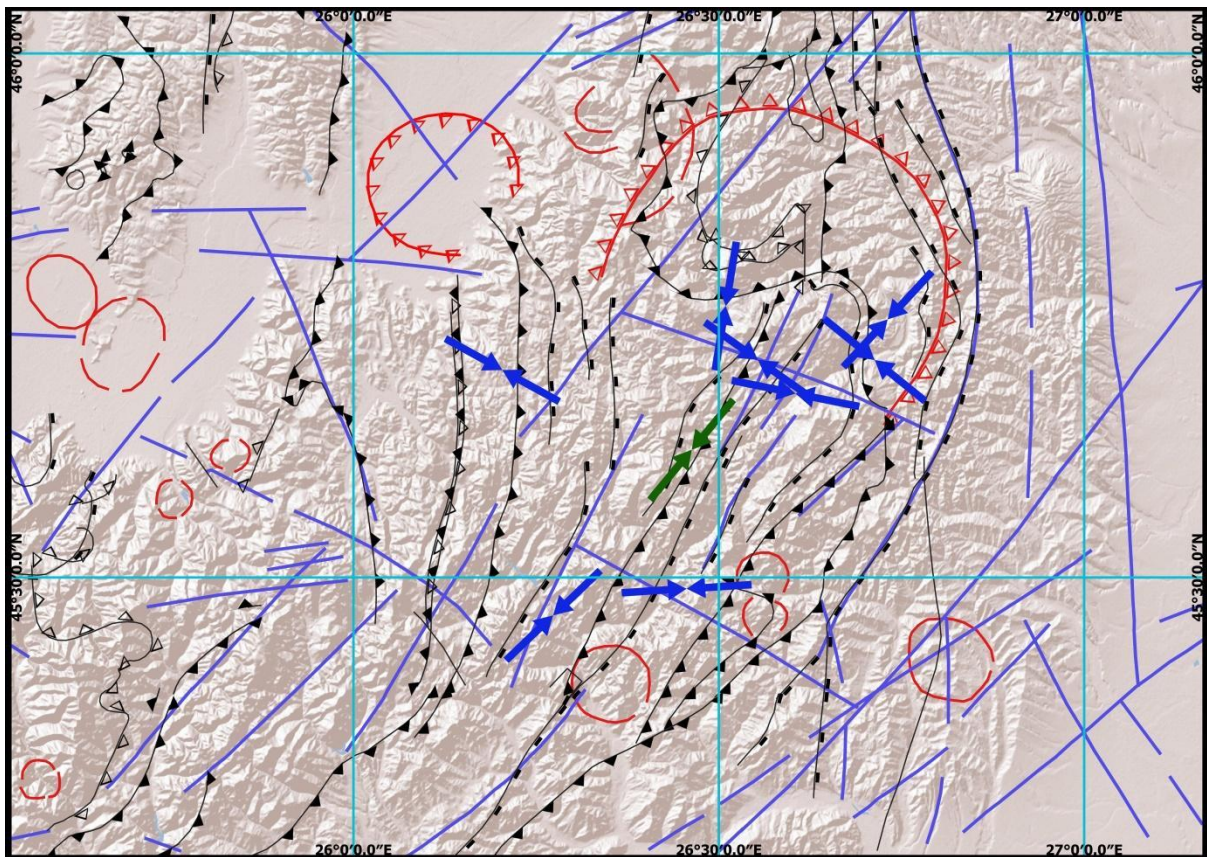
29. ábra: a 60 és 100 km közötti mélységtartományban lévő, 4D-s klaszterezés után kapott alterületekre értelmezett jellemző feszültségirányok és tektonikus rezsimek. Kék színű nyíl jelzi a feltolódásokat, zöld az oldalelmozdulásokat és piros a normál vetőket.

A 30. és 31. ábra alapján az mondható el, hogy a jellemző földtani szerkezet a feltolódás, de normál vetők és oldalelmozdulás is jelentkezik az eredmények között. A maximális horizontális kompressziós irány pedig követi a Kárpátoknak az ívét.

A következő két (ld. 30. és 31. ábra) térképen a 120 és 200 km közötti mélységben lévő földrengésekre elvégzett inverziók értelmezése látható.



30. ábra: a 120 és 200 km közötti mélységtartományban lévő, 3D-s klaszterezés után kapott alterületekre értelmezett jellemző feszültségirányok és tektonikus rezsimek. Kék színű nyíl jelzi a feltolódásokat, zöld az oldalelmozdulásokat és piros a normál vetőket.



31. ábra: a 120 és 200 km közötti mélységtartományban lévő, 4D-s klaszterezés után kapott alterületekre értelmezett jellemző feszültségirányok és tektonikus rezsimek. Kék színű nyíl jelzi a feltolódásokat, zöld az oldalelmozdulásokat és piros a normál vetőket.

Amíg a felső mélységtartományban csak a terület ÉK-i részéről vannak eredmények, addig itt a teljes Vráncsa-zónára meghatározhatóak voltak a feszültségmező irányítottsága és az uralkodó tektonikus rezsimek. Itt még inkább kiemelkedik a feltolódások főszerepe, de normál vető és oldaleltolódás is előfordul. Ami a feszültségmező orientációját illeti, két eset különböztethető meg: vagy követi a Kárpátkanyar és a szerkezeteknek a vonalát vagy éppen ezekre merőleges (elsősorban a terület középső részének északi felén).

5. Diszkusszió

A feszültséginverziós számításokhoz kapcsolódó adatok gyűjtése illetve a különböző feladatokhoz, mint pl. a strandlabdák térképen történő illusztrálása vagy éppen a fészekmechanizmusok segédsíkjainak kiírása, Excelben kerültek elvégzésre. Továbbá ha szükséges volt valamilyen adatmódosítást végrehajtani vagy éppen bővíteni az adatrendszert, azt is ott hajtottam végre. A szakdolgozat írása során azonban már miután QGIS-ben elmentettem a kapcsolódó rétegeket vektoros (SHAPE) formátumban, már magában az attribútum-táblákban is el tudom ezeket végezni.

Ami a fészekmechanizmusok sztereografikus projekcióit illeti, eredetileg a Generic Mapping Tools – szoftvert használtam a térképek elkészítéséhez. Mindazonáltal maga a program meglehetősen bonyolult. A munka során azonban sikerült áttérni Obspy nevű Python-csomagra, amelyben a `beachball()` függvény alkalmas arra, hogy a bevitt adatokból kirajzolja a strandlabdákat. Ezeket aztán kimentettem SVG – formátumban, amiket SVG – markereként be tudtam hívni QGIS-be és ráilleszteni a földrengések helyeire (pontjaira).

A feszültséginverziós számítások esetében is sikerült áttérni a MATLAB – programozási nyelvről Python-ba, köszönhetően annak, hogy a STRESSINVERSE 2020 óta már elérhető ezen a nyelven. Azonban belefutottam abba a problémába, hogy bár Python-ban futnak a kódok, a feszülsténzor elemeit tartalmazó mátrix továbbra is a MATLAB-nyelvhez köthető MAT formátumban mentődik ki. Ezt a `mat4py` nevű csomagnak a segítségével sikerült kezelhetőbb formátumba átmenteni és így módon folytatni tovább a munkát QGIS-ben és elkészíteni az eredmények és azok kiértékelését szolgáló összefoglaló térképeket (ld. 24-31. ábrák).

A kapott klaszterek és a feszültség- sztereogramok interpretációinak térképei a dolgozat előtt is már QGIS-ben kerültek előállításra, azonban a sztereogramokat egy alaptérképre helyezni már csak a LibreOffice – rajzoló programjában tudtam. Itt találtam lehetőséget arra, hogy az eredetileg téglalap alakú képen lévő kör alakú sztereogramot kivágjam és utána ráhelyezzem az amúgy QGIS-ből kimentett alaptérképre. A feladat első része – azaz a sztereogramok testreszabása – még mindig LibreOffice-ban történik, azonban ezeket raszteres kép – jelölőként (Raster Image Symbol) behívhatóak QGIS-be, így az SVG-vel vagy más típusú markerekkel egyenértékű szimbólumokat kaptam. Ezekre aztán a QGIS-ban feliratok (Labels) illesztésével az R-számokat is sikerült megjelenítenem, ugyanis azok is szerepelnek az eredmények kiértékelésére szolgáló attribútum-táblában. Így most már elmondható, hogy a térképi interpretációk összes lépése QGIS – ben történik.

Az adatok gyűjtéséhez továbbra is Microsoft Excel – t használom, amiből ki tudom menteni a szükséges szöveges fájlokat a munkákhoz Pythonban és QGIS-ben. Utóbbiban, ha valamilyen hiba történt az adatok bevitelénél (pl. elírás), akkor annak javítását vagy egy új rekord hozzáadását már QGIS-ben teszem meg. Mindemellett a különböző szimbólumok átalakítására különböző rajzoló programokat használom (elsősorban LibreOffice Draw-t a

térképekhez és a sztereogramokhoz és InkScape-t az SVG-szimbóluk esetében) még. A 32. ábrán a munkafolyamatok változását mutatja be.



32. ábra: A feszültségviszonyok vizsgálatához kapcsolódó munkafolyamat főbb lépései láthatók. Kék résszel a szakdolgozat előtti, zöld résszel a szakdolgozat során elvégzett feladatok olvashatóak. Vörös kerettel kiemelttem azokat a lépéseket, ahol más alkalmazást használtam.

6. Következtetések

Sikerült az eddig felhasznált programok, szoftverek számát ötről kettőre csökkenteni a kutatási munka során, csak kiegészítő lépések esetén (pl. adatgyűjtés vagy szimbólumok kezelése) használtam más alkalmazást. Ez jelentősen javítja a számításigényt, a memóriakapacitás felhasználását és a feladatok elvégzésére szánt időt is. Mindemellett ezek a módszertani lépések a további kutatási tevékenységben – pl. tézisek doktori disszertációmban vagy éppen publikáció formájában történő megjelentetése – is fontos szerepet játszanak a jövőbeni munkában.

További tervek között szerepel PostGIS lekérdezések illetve a feszültség inverziókhöz kapcsolódó sztereogramok tényleges kör alakban (sztereonet/sztereoplot), valamint kapcsolódó geológiai szerkezetekhez vagy éppen tektonikai folyamatokhoz kapcsolódó jelek ábrázolásának átvitele Python-környezetbe, ezzel a rajzoló programok is “kiesnének a praxisból”. Emellett QGIS-plugin-t is tervezek létrehozni a CluStress – klaszterező

algoritmushoz kapcsolódóan, valamint a már meglévő szintetikus fészekmechanizmus-generátort is tovább fejleszteném.

Ami a földfizikai aspektusát illeti jelen munkának: a feszültséginverziós számítások eredménye jó egyezést mutat korábbi, már megjelent publikációk eredményeivel. A feltolódások uralkodó szerepe a Vrancea-zónában alátámasztja a kompressziós feszültségmező jelenlétét, amely összekapcsolható a még napjainkban is zajló szubdukciós (lemezlebukásos) folyamatokkal.

Irodalomjegyzék

- Anderson, E. M. (1951). The dynamics of faulting and dike formation with application to Britain. Oliver and Boyd, 2nd Edition, Edinburgh, 206.
- Angelier, J. (2002). Inversion of earthquake focal mechanisms to obtain the seismotectonic stress IV – a new method free of choice among nodal planes, *Geophys. J. Int.*, 150, 588–609.
- Bala, A., Radulian, M., and Toma-Danila, D. (2021). Present-day stress field pattern in the Vrancea seismic zone (Romania) deduced from earthquake focal mechanism inversion. *Annals of Geophysics*, 64(6), PE660-PE660.
- Bott, M. H. (1959), The mechanics of oblique slip faulting. *Geol. Mag.*, 96, 109–117.
- Beyreuther, M., Barsch, R., Krischer, L., Megies, T., Behr, Y., and Wassermann, J. (2010). ObsPy: A Python toolbox for seismology. *Seismological Research Letters*, 81(3), 530-533.
- Cronin, V. (2010). A primer on focal mechanism solutions for geologists. Science Education Resource Center, Carleton College.
- Czirok, L. (2016). Analysis of stress relations using focal mechanism solutions in the Pannonian Basin. *Geosciences and Engineering: A publication of the University of Miskolc*, 5(8), 65-84.
- Czirok, L., and Kuslits, L. (2018). Effects of earthquake data clustering on the results of stress inversion. *Geosciences and Engineering: A publication of the University of Miskolc*, 6(9), 127-141.
- Czirok, L., Kuslits, L., Bozsó, I., and Gribovszki, K. (2022): Studying the stress field variations in the Vrancea-zone using clustering-based stress inversions, EGU General Assembly 2022, Vienna, Austria, 23–27 May 2022, EGU22-9181, <https://doi.org/10.5194/egusphere-egu22-9181>.
- Ester, M., Kriegel, H. P., Sander, J., and Xu, X. (1996, August). A density-based algorithm for discovering clusters in large spatial databases with noise. In *kdd* (Vol. 96, No. 34, pp. 226-231).
- GEOS contributors (2021). GEOS computational geometry library. Open Source Geospatial Foundation. URL <https://libgeos.org/>.

- Gephart, J.W. and D.W. Forsyth (1984). An improved method for determining the regional stress tensor using earthquake focal mechanism data: application to the San Fernando earthquake sequence, *J. Geophys. Res.*, 89, 9305–9320.
- Goualard, F. (2020, June). Generating Random Floating-Point Numbers by Dividing Integers: a Case Study. In *International Conference on Computational Science* (pp. 15-28). Springer, Cham.
- Harangi, S., Novák, A., Kiss, B., Seghedi, I., Lukács, R., Szarka, L., Wesztergom, V., Metwaly, M., and Gribovszki, K. (2015). Combined magnetotelluric and petrologic constraints for the nature of the magma storage system beneath the Late Pleistocene Ciomadul volcano (SE Carpathians). *Journal of Volcanology and Geothermal Research*, 290, pp. 82–96.
- Hardebeck, J.L. and A.J. Michael (2006). Damped regional-scale stress inversions: Methodology and examples for southern California and the Coalinga aftershock sequence, *J. Geophys. Res., Solid Earth*, 111, B11310, doi 10.1029/2005JB004144.
- Heidbach, O., Ledermann, P., Kurfeß, D., Peters, G., Buchmann, T., Matenco, L., Negut, M., Sperner, B., Müller, B., Nuckelt, A. and Schmitt, G. (2007, October). Attached or not attached: slab dynamics beneath Vrancea, Romania. In *Proceedings of the International Symposium on Strong Vrancea*.
- Heidbach, Oliver; Rajabi, Mojtaba; Reiter, Karsten; Ziegler, Moritz; WSM Team (2016): World Stress Map Database Release 2016. V. 1.1. GFZ Data Services. <https://doi.org/10.5880/WSM.2016.001>
- Heidbach, O., Rajabi, M., Cui, X., Fuchs, K., Müller, B., Reinecker, J., ... and Zoback, M. (2018). The World Stress Map database release 2016: Crustal stress pattern across scales. *Tectonophysics*, 744, 484-498.
- Ismail-Zadeh, A., Matenco, L., Radulian, M., Cloetingh, S., and Panza, G. (2012). Geodynamics and intermediate-depth seismicity in Vrancea (the south-eastern Carpathians): Current state-of-the art. *Tectonophysics*, 530, 50–79.
- Kassaras, I. G., and Kapetanidis, V. (2018). Resolving the tectonic stress by the inversion of earthquake focal mechanisms. Application in the region of Greece. A tutorial. In *Moment Tensor Solutions* (pp. 405-452). Springer, Cham.
- Krischer, L., Megies, T., Barsch, R., Beyreuther, M., Lecocq, T., Caudron, C., and Wassermann, J. (2015). ObsPy: A bridge for seismology into the scientific Python ecosystem. *Computational Science & Discovery*, 8(1), 014003.
- Kuslits, L., Czirok, L., and Bozsó, I. (2022). “Fully-automated” clustering method for stress inversions (CluStress) (No. EGU22-12489). *Copernicus Meetings*.
- Lentas, K. (2018). Towards routine determination of focal mechanisms obtained from first motion P-wave arrivals. *Geophys. J. Int.*, 212(3), 1665-1686. doi:10.1093/gji/ggx503
- Lentas, K., Di Giacomo, D., Harris, J., and Storchak, D. A. (2019). The ISC Bulletin as a comprehensive source of earthquake source mechanisms. *Earth Syst. Sci. Data*, 11, 565-578. doi: <https://doi.org/10.5194/essd-11-565-2019>
- Matenco, L., and Radivojević, D. (2012). On the formation and evolution of the Pannonian Basin: Constraints derived from the structure of the junction area between the Carpathians and Dinarides. *Tectonics*, 31(6).

- Maury, J., Cornet, F. H., and Dorbath, L. (2013). A review of methods for determining stress fields from earthquakes focal mechanisms; Application to the Sierentz 1980 seismic crisis (Upper Rhine graben). *Bulletin de la Societe Geologique de France*, 184(4-5), 319-334.
- Martínez-Garzón, P., Kwiątek, G., Ickrath, M., and Bohnhoff, M. (2014). MSATSI: A MATLAB package for stress inversion combining solid classic methodology, a new simplified user-handling, and a visualization tool. *Seismological Research Letters*, 85(4), 896-904.
- McInnes, L., Healy, J., and Astels, S. (2017): hdbscan: Hierarchical density based clustering. *Journal of Open Source Software*, 2(11), 205.
- Megies, T., Beyreuther, M., Barsch, R., Krischer, L., and Wassermann, J. (2011). ObsPy—What can it do for data centers and observatories?. *Annals of Geophysics*, 54(1), 47-58.
- Michael, A.J. (1984). Determination of stress from slip data: Faults and folds, *J. Geophys. Res.*, 89, B13, 11517–11526.
- Onescu, M. C. (1988). On the stress tensor in Vrancea region. *Journal of Geophysics*, 62(1), 62-65.
- Onescu M.C., Bonjer K.P., A note on the depth recurrence and strain release of large intermediate Vrancea earthquakes. *Tectonophysics* 272, 291-302, 1997.
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, É. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12(Oct), 2825–2830
- Petrescu, L., Borleanu, F., Radulian, M., Ismail-Zadeh, A., and Mațenco, L. (2021). Tectonic regimes and stress patterns in the Vrancea Seismic Zone: Insights into intermediate-depth earthquake nests in locked collisional settings. *Tectonophysics*, 799, 228688.
- Popescu, E., Radulian, M., Bala, A., and Toma-Danila, D. (2018). Earthquake mechanism in the Vrancea subcrustal source and in the adjacent crustal seismogenic zones of the south-eastern Romania. *Romanian Rep Phys*, 704.
- Radulian, M., Bala, A., Popescu, E., and Toma-Danila, D. (2018). Earthquake mechanism and characterization of seismogenic zones in south-eastern part of Romania. *Annals of Geophysics*, 61 (1), 108.
- Radulian M., Bala A., Ardeleanu L., Toma-Danila D., Petrescu L., and Popescu E. (2019). Revised catalogue of earthquake mechanisms for the events occurred in Romania until the end of twentieth century: REFMC. *Acta Geodaetica et Geophysica*, 54(1): 3-18.
- Radulian M., Bala A., Toma-Danila D. (2020): “REFMC 1929 2012 (Romanian Earthquake Focal Mechanism Catalogue)”, Mendeley Data, V3, doi: 10.17632/mykx4gygy.3
- Schmitt, G., Nuckelt, A., Knöpfler, A., and Marcu, C. (2007). Three dimensional plate kinematics in Romania. In *Proceedings of the International Symposium on Strong Vrancea Earthquakes and Risk Mitigation* (pp. 34-45). Romania: Bucharest.
- Shearer, P. M. (2009). *Introduction to seismology*. Cambridge University Press.

- Szűcs, E., Bozsó, I., Kovács, I.J., Bányai, L., Gál, Á., Szakács S., and Wesztergom, V. (2018). Probing tectonic processes with space geodesy in the south Carpathians: insights from archive SAR data. *Acta Geodaetica et Geophysica*, 53, 331–345.
- Wallace, R. E. (1951). Geometry of Shearing Stress and Relation to Faulting. *J. Geol.*, 59(2), 118–130.

Függelék

A szintetikus fészekmechanizmus – generáló forráskódja

```
#!/usr/bin/env python
# coding: utf-8

get_ipython().run_line_magic('reset', '')

# Import packages
import random
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from shapely.geometry import Point, Polygon, MultiPoint, LineString
import shapely.affinity
import geopandas
from sklearn.datasets import make_blobs
from obspy.imaging.beachball import beach
from sympy import symbols
from pprint import pprint

# ## Synthetic Focal Mechanisms Inside Different-formed features (SynFMID)
# Synthetic database with the next attributes:
# - Epicentre - coordinates: Longitude (X), Latitude (Y) in Cartesian coordinate
system
# - Focal depth: [0, 200] (km)
# - Strike: [0, 360] (°)
# - Dip: [0, 90] (°)
# - Rake: [-180,180] (°)

# ## 1st solution - using Class

class FocalMechanismGenerator:

    def __init__(self, kwargs):
        self.d1 = kwargs['depth_h']
        self.d2 = kwargs['depth_l']
        self.s1 = kwargs['strike_l']
        self.s2 = kwargs['strike_h']
        self.dip1 = kwargs['dip_l']
        self.dip2 = kwargs['dip_h']
        self.r1 = kwargs['rake_l']
        self.r2 = kwargs['rake_h']

    def create_polygon(self, polygon_type, polygon_points):
        """Creation of the polygons"""
        polygon = getattr(shapely.geometry, polygon_type)
        self.polygon_object = polygon(polygon_points)

    def generate_fms_within_polygon(self, num):
        """Generation of reproducible synthetic focal mechanisms within the
polygons"""
        minx, miny, maxx, maxy = self.polygon_object.bounds
        points = []
        depths = []
        strikes = []
        dips = []
        rakes = []
        n = 0
        while n<=num:
            point = Point((random.uniform(minx, maxx)),
(random.uniform(miny,maxy)))
            depth = random.randint(self.d1,self.d2)
            strike = random.randint(self.s1,self.s2)
```

```

        dip = random.randint(self.dip1, self.dip2)
        rake = random.randint(self.r1, self.r2)
        if point.within(polygon):
            points.append(point)
            depths.append(depth)
            strikes.append(strike)
            dips.append(dip)
            rakes.append(rake)
            n=n+1;
    depths = np.array(depths)
    depths = np.transpose(depths)
    depths = depths.flatten()
    depths.sort()
    strikes = np.array(strikes)
    strikes = np.transpose(strikes)
    strikes = strikes.flatten()
    strikes.sort()
    dips = np.array(dips)
    dips = np.transpose(dips)
    dips = dips.flatten()
    dips.sort()
    rakes = np.array(rakes)
    rakes = np.transpose(rakes)
    rakes = rakes.flatten()
    rakes.sort()
    return points, depths, strikes, dips, rakes

def generate_fms_around_the_line(self, num):
    """Generation of reproducible synthetic focal mechanisms around the line"""
    minx, miny, maxx, maxy = self.polygon_object.bounds
    a,b = self.polygon_object.boundary
    ax = a.x
    bx = b.x
    ay = a.y
    by = b.y
    slope = (ay-by)/(ax-bx)
    depths = []
    strikes = []
    dips = []
    rakes = []
    points = []
    while True:
        c = Point(random.uniform(ax, ay), random.uniform(bx, by))
        cx = c.x
        cy = c.y
        cy_e = ay + slope*(cx-ax)
        if len(points) == num:
            break
        if abs(cy_e-cy) <= 1:
            points.append(c)
    depth = random.sample(range(self.d1, self.d2), num)
    depth = np.transpose(depth)
    depth = depth.flatten()
    depth.sort()
    strike = random.sample(range(self.s1, self.s2), num)
    strike = np.transpose(strike)
    strike = strike.flatten()
    strike.sort()
    dip_1 = random.sample(range(self.dip1, self.dip2), int(num/2))
    dip_2 = random.sample(range(self.dip1, self.dip2), int(num/2))
    dip = [*dip_1, *dip_2]
    dip = np.transpose(dip)
    dip = dip.flatten()
    dip.sort()
    rake = random.sample(range(self.r1, self.r2), num)
    rake = np.transpose(rake)
    rake = rake.flatten()
    rake.sort()

```

```

        return points, depth, strike, dip, rake

    @staticmethod
    def convert_to_dataframe(fms_data):
        pontok = pd.DataFrame(fms_data)
        pontok = np.transpose(pontok)
        return pontok

fms_generator = FocalMechanismGenerator({'depth_h': 70, 'depth_l': 120, 'strike_l':
0, 'strike_h': 360, 'dip_l': 0, 'dip_h': 90, 'rake_l': 50, 'rake_h': 65})

polygon_points = [Point(50, 75), Point(20, 90)]
fms_generator.create_polygon('LineString', polygon_points)

fms_data = fms_generator.generate_fms_around_the_line(10)

fms_df = fms_generator.convert_to_dataframe(fms_data)

fms_df

fms_df.columns = ['point', 'z', 'strike', 'dip', 'rake']
fms_df

#Separation of the X and Y coordinates
def separate_x(input_point):
    x_coordinate = input_point.x
    return x_coordinate

def separate_y(input_point):
    y_coordinate = input_point.y
    return y_coordinate

fms_df['x'] = fms_df['point'].apply(separate_x)
fms_df['y'] = fms_df['point'].apply(separate_y)
fms_df

fms_df.drop('point', axis='columns', inplace = True)
fms_df

##Figures
fig = plt.figure(1, figsize=(5,5), dpi=300)
x,y = fms_generator.polygon_object.xy
ax = fig.add_subplot(111)
ax.plot(x, y, color='#6699cc', alpha=0.7,
        linewidth=3, solid_capstyle='round', zorder=2)
plt.savefig('class_line090521.png')
plt.show()

fig2 = plt.figure(2, figsize=(5,5), dpi=300)
plt.scatter(fms_df['x'], fms_df['y'])
plt.savefig('class_line_points090521.png')
plt.show()

fig3 = plt.figure(3, figsize=(5,5), dpi=600)
plt.plot([50,75], [62,80], "rv", ms=10)
ax = plt.gca()
focal_mechanisms = fms_df[['strike', 'dip', 'rake']]
for i in range(len(focal_mechanisms)):
    beachball = beach(focal_mechanisms.iloc[i], xy=(fms_df['x'].iloc[i],
fms_df['y'].iloc[i]), width=3, facecolor='black')
    ax.add_collection(beachball)
plt.savefig('class_line_fms090521.png')
plt.show()

# ## 2nd solution - using only functions

#Creation of the square
polygon2 = Polygon([(0, 0), (0, 200), (200,200), (200, 0)]) #square

```

```

x2,y2 = polygon2.exterior.xy
fig = plt.figure(1, figsize=(5,5), dpi=300)
ax = fig.add_subplot(111)
ax.plot(x2, y2, color='#6699cc', alpha=0.7,
        linewidth=3, solid_capstyle='round', zorder=2)
ax.set_title('Areas for the test')

#Creation of the features inside the square
circle = Point(25,25).buffer(15) #Circle
dx1,dyl = circle.exterior.xy
fig = plt.figure(1, figsize=(5,5), dpi=300)
ax = fig.add_subplot(111)
ax.plot(dx1,dyl, color='#1111bb', alpha=0.7,
        linewidth=3, solid_capstyle='round', zorder=2)
# -----
line = LineString([Point(50, 75), Point(20, 90)]) #Line
dx1,dyl = line.xy
fig = plt.figure(1, figsize=(5,5), dpi=300)
ax = fig.add_subplot(111)
ax.plot(dx1,dyl, color='#1111bb', alpha=0.7,
        linewidth=3, solid_capstyle='round', zorder=2)
# -----
district2 = Polygon([(140,50), (140, 85), (105,85), (105, 50)]) #Polygon2
dx1,dyl = district2.exterior.xy
fig = plt.figure(1, figsize=(5,5), dpi=300)
ax = fig.add_subplot(111)
ax.plot(dx1,dyl, color='#1111bb', alpha=0.7,
        linewidth=3, solid_capstyle='round', zorder=2)
# -----
district3 = Polygon([(140,140), (140, 180), (180,180), (180, 140)]) #Polygon3
dx1,dyl = district3.exterior.xy
fig = plt.figure(1, figsize=(5,5), dpi=300)
ax = fig.add_subplot(111)
ax.plot(dx1,dyl, color='#1111bb', alpha=0.7,
        linewidth=3, solid_capstyle='round', zorder=2)
# -----
circle2 = Point(127,150).buffer(2)
ellipse = shapely.affinity.scale(circle2, 4, 5) #Ellipse
dx1,dyl = ellipse.exterior.xy
fig = plt.figure(1, figsize=(5,5), dpi=300)
ax = fig.add_subplot(111)
ax.plot(dx1,dyl, color='#1111bb', alpha=0.7,
        linewidth=3, solid_capstyle='round', zorder=2)
# -----
district4 = Polygon([(10,190), (30, 175), (45,145), (50, 130)]) #Polygon4
dx1,dyl = district4.exterior.xy
fig = plt.figure(1, figsize=(5,5), dpi=300)
ax = fig.add_subplot(111)
ax.plot(dx1,dyl, color='#1111bb', alpha=0.7,
        linewidth=3, solid_capstyle='round', zorder=2)
# # -----
plt.savefig('square_190521.jpg')
plt.show()

def convert_to_dataframe(pontok_arr, variables):
    pontok = pd.DataFrame({})
    for var_ind in range(len(variables)):
        pontok[variables[var_ind]] = pontok_arr[0][:,var_ind].reshape(-
1,1).flatten().copy()
    return pontok

def blob(num, center_loc, std, kwargs):
    d1 = kwargs['depth_h']
    d2 = kwargs['depth_l']
    s1 = kwargs['strike_l']
    s2 = kwargs['strike_h']
    dip1 = kwargs['dip_l']
    dip2 = kwargs['dip_h']

```

```

    r1 = kwargs['rake_l']
    r2 = kwargs['rake_h']
    sample_number = num
    data = make_blobs(n_samples=sample_number, n_features=2, centers=center_loc,
random_state=0, cluster_std=std)
    pontok = convert_to_dataframe(data, ['x', 'y'])
    depth = [random.randint(d1,d2) for i in range(num)]
    depth = np.transpose(depth)
    depth = depth.flatten()
    depth.sort()
    strike = [random.randint(s1,s2) for i in range(num)]
    strike = np.transpose(strike)
    strike = strike.flatten()
    strike.sort()
    dip = [random.randint(dip1,dip2) for i in range(num)]
    dip = np.transpose(dip)
    dip = dip.flatten()
    dip.sort()
    rake = [random.randint(r1,r2) for i in range(num)]
    rake = np.transpose(rake)
    rake = rake.flatten()
    rake.sort()
    pontok['z'] = depth
    pontok['strike'] = strike
    pontok['dip'] = dip
    pontok['r'] = rake
    return pontok

def random_repr_fms_within_polygon(polygon, num, kwargs): #random FMSs within the
Polygon
    d1 = kwargs['depth_h']
    d2 = kwargs['depth_l']
    s1 = kwargs['strike_l']
    s2 = kwargs['strike_h']
    dip1 = kwargs['dip_l']
    dip2 = kwargs['dip_h']
    r1 = kwargs['rake_l']
    r2 = kwargs['rake_h']
    minx, miny, maxx, maxy = polygon.bounds
    points = []
    depths = []
    strikes = []
    dips = []
    rakes = []
    n = 0
    while n<=num:
        point = Point((random.uniform(minx, maxx)), (random.uniform(miny,maxy)))
        depth = random.randint(d1,d2)
        strike = random.randint(s1,s2)
        dip = random.randint(dip1,dip2)
        rake = random.randint(r1,r2)
        if point.within(polygon):
            points.append(point)
            depths.append(depth)
            strikes.append(strike)
            dips.append(dip)
            rakes.append(rake)
            n=n+1;
    depths = np.array(depths)
    depths = np.transpose(depths)
    depths = depths.flatten()
    depths.sort()
    strikes = np.array(strikes)
    strikes = np.transpose(strikes)
    strikes = strikes.flatten()
    strikes.sort()
    dips = np.array(dips)
    dips = np.transpose(dips)

```

```

dips = dips.flatten()
dips.sort()
rakes = np.array(rakes)
rakes = np.transpose(rakes)
rakes = rakes.flatten()
rakes.sort()
return points, depths, strikes, dips, rakes

def random_fms_within_polygon(polygon, num, kwargs): #random FMSs within the Square
    d1 = kwargs['depth_h']
    d2 = kwargs['depth_l']
    s1 = kwargs['strike_l']
    s2 = kwargs['strike_h']
    dip1 = kwargs['dip_l']
    dip2 = kwargs['dip_h']
    r1 = kwargs['rake_l']
    r2 = kwargs['rake_h']
    minx, miny, maxx, maxy = polygon.bounds
    point = []
    while True:
        point = [Point((random.uniform(minx, maxx)), (random.uniform(miny, maxy)))]
    for i in range(num):
        depth = random.sample(range(d1, d2), num)
        depth = np.transpose(depth)
        depth = depth.flatten()
        depth.sort()
        strike = random.sample(range(s1, s2), num)
        strike = np.transpose(strike)
        strike = strike.flatten()
        strike.sort()
        dip_1 = random.sample(range(dip1, dip2), int(num/2))
        dip_2 = random.sample(range(dip1, dip2), int(num/2))
        dip = [*dip_1, *dip_2]
        dip = np.transpose(dip)
        dip = dip.flatten()
        dip.sort()
        rake = random.sample(range(r1, r2), num)
        rake = np.transpose(rake)
        rake = rake.flatten()
        rake.sort()
        return point, depth, strike, dip, rake

def random_fms_on_the_line(polygon, num, kwargs): #random FMSs within the Square
    d1 = kwargs['depth_h']
    d2 = kwargs['depth_l']
    s1 = kwargs['strike_l']
    s2 = kwargs['strike_h']
    dip1 = kwargs['dip_l']
    dip2 = kwargs['dip_h']
    r1 = kwargs['rake_l']
    r2 = kwargs['rake_h']
    minx, miny, maxx, maxy = polygon.bounds
    a, b = polygon.boundary
    ax = a.x
    bx = b.x
    ay = a.y
    by = b.y
    slope = (ay-by)/(ax-bx)
    depths = []
    strikes = []
    dips = []
    rakes = []
    points = []
    while True:
        c = Point(random.uniform(ax, ay), random.uniform(bx, by))
        cx = c.x
        cy = c.y
        cy_e = ay + slope*(cx-ax)

```

```

        if len(points) == num:
            break
        if abs(cy_e-cy) <= 1:
            points.append(c)
    depth = random.sample(range(d1,d2),num)
    depth = np.transpose(depth)
    depth = depth.flatten()
    depth.sort()
    strike = random.sample(range(s1, s2),num)
    strike = np.transpose(strike)
    strike = strike.flatten()
    strike.sort()
    dip_1 = random.sample(range(dip1,dip2), int(num/2))
    dip_2 = random.sample(range(dip1,dip2), int(num/2))
    dip = [*dip_1, *dip_2]
    dip = np.transpose(dip)
    dip = dip.flatten()
    dip.sort()
    rake = random.sample(range(r1, r2),num)
    rake = np.transpose(rake)
    rake = rake.flatten()
    rake.sort()
    return points, depth, strike, dip, rake

# Generated random FMS noise within the entire area
argin = {'depth_h': 0, 'depth_l': 200, 'strike_l': 0, 'strike_h': 360,
'dip_l': 0, 'dip_h': 90, 'rake_l': -180, 'rake_h': 180}
fms_district0=random_repr_fms_within_polygon(polygon2, 20, argin)
fms_district0=pd.DataFrame(fms_district0)
fms_district0=np.transpose(fms_district0)

#Generated FMSs in the subareas inside the square
#-----1st
argin = {'depth_h': 70, 'depth_l': 120, 'strike_l': 0, 'strike_h': 360,
'dip_l': 0, 'dip_h': 90, 'rake_l': 50, 'rake_h': 65}
fms_district1=random_repr_fms_within_polygon(circle, 50, argin)
fms_district1=pd.DataFrame(fms_district1)
fms_district1=np.transpose(fms_district1)
#-----2nd
argin = {'depth_h': 10, 'depth_l': 20, 'strike_l': 0, 'strike_h': 360,
'dip_l': 0, 'dip_h': 90, 'rake_l': 0, 'rake_h': 15}
fms_district2=random_repr_fms_within_polygon(ellipse, 15, argin)
fms_district2=pd.DataFrame(fms_district2)
fms_district2=np.transpose(fms_district2)
#-----3rd
argin = {'depth_h': 20, 'depth_l': 50, 'strike_l': 0, 'strike_h': 360,
'dip_l': 0, 'dip_h': 90, 'rake_l': 60, 'rake_h': 110}
fms_district3=random_repr_fms_within_polygon(district2, 10, argin)
fms_district3=pd.DataFrame(fms_district3)
fms_district3=np.transpose(fms_district3)

#fms_district2 = pd.concat([fms_district2a, fms_district2b, fms_district2c])

#-----4th
argin = {'depth_h': 140, 'depth_l': 170, 'strike_l': 0, 'strike_h': 360,
'dip_l': 0, 'dip_h': 90, 'rake_l': -170, 'rake_h': -135}
fms_district4=random_repr_fms_within_polygon(district3, 200, argin)
fms_district4=pd.DataFrame(fms_district4)
fms_district4=np.transpose(fms_district4)

#-----5th
argin = {'depth_h': 130, 'depth_l': 150, 'strike_l': 0, 'strike_h': 360,
'dip_l': 0, 'dip_h': 90, 'rake_l': -180, 'rake_h': -165}
fms_district5=random_repr_fms_within_polygon(district4, 100, argin)
fms_district5=pd.DataFrame(fms_district5)
fms_district5=np.transpose(fms_district5)
#-----6th

```

```

argin = {'depth_h': 130, 'depth_l': 150, 'strike_l': 0,          'strike_h': 360,
'dip_l': 0, 'dip_h': 90, 'rake_l': -150, 'rake_h': -120}
fms_district6=random_fms_on_the_line(line, 20, argin)
fms_district6=pd.DataFrame(fms_district6)
fms_district6=np.transpose(fms_district6)

fms_districts = pd.concat([fms_district0, fms_district1, fms_district2,
fms_district3, fms_district4, fms_district5, fms_district6])

#Rename of the columns
fms_districts.columns = ['point', 'z', 'strike', 'dip', 'r']

#Generated noises
argin = {'depth_h': 10, 'depth_l': 60, 'strike_l': 0,          'strike_h': 360,
'dip_l': 0, 'dip_h': 90, 'rake_l': 115, 'rake_h': 145}
fms_blobs = blob(20, np.array([[175,175],[130,175]]).T, 4.8, argin)

#Full synthetic dataset
fms_districts = fms_districts.drop(columns=['point'], axis=1)
synthetic_fms = pd.concat([fms_districts, fms_blobs])

#Location of the focal mechanisms within the generated synthetic features
fig2 = plt.figure(2, figsize=(15,15), dpi=600)
plt.scatter(synthetic_fms['x'], synthetic_fms['y'])
plt.savefig('synthethic_points090521.png')
plt.show()

#Maps of the synthetic focal mechanis solutions represented by beachballs
fig_fms = plt.figure(1, figsize=(5,5), dpi=600)
plt.plot([0, 200], [0, 200], "rv", ms=10)
ax = plt.gca()
focal_mechanisms = synthetic_fms[['strike', 'dip', 'r']]
for i in range(len(focal_mechanisms)):
    beachball = beach(focal_mechanisms.iloc[i], xy=(synthetic_fms['x'].iloc[i],
synthetic_fms['y'].iloc[i]), width=10, facecolor='black')
    ax.add_collection(beachball)
plt.savefig('synthetic_beachballs090521.png')
plt.show()

#Save as a comma separated values - file
fms_df.to_excel('fms_data_090521.xlsx')
synthetic_fms.to_csv('synthetic_fms_090521.csv')

```