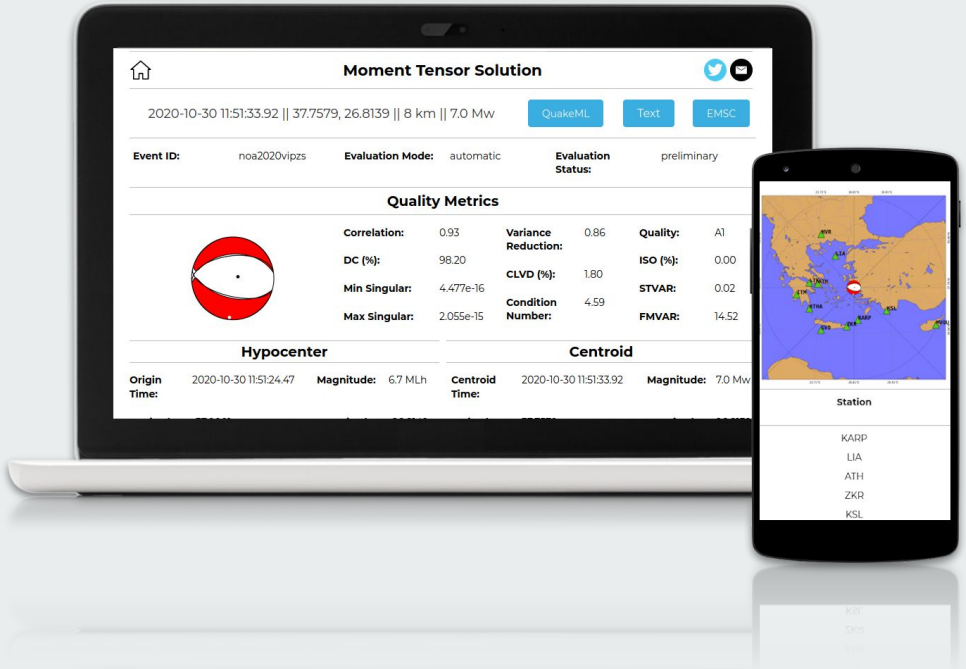
National Observatory of Athens
Institute of Geodynamics

Gisola

Real-Time Moment Tensor Computation



Nikolaos Triantafyllis (triantafyl@noa.gr)

Triantafyllis, N., Venetis, I., Fountoulakis, I., Pikoulis, E.-V., Sokos, E., and Evangelidis, C.: Gisola: Real-Time Moment Tensor computation optimized for multicore and manycore architectures, EGU General Assembly 2021, online, 19–30 Apr 2021, EGU21-15888, https://doi.org/10.5194/egusphere-egu21-15888, 2021.

Automatic Moment Tensor (MT) determination for regional areas is essential for real-time seismological applications such as shakemap generation, and tsunami warning

is an open-source Python-based software for automatic MT calculation of seismic events provided by the FDSN Web Services in real-time, oriented for High-Performance Computing

Process Flowchart

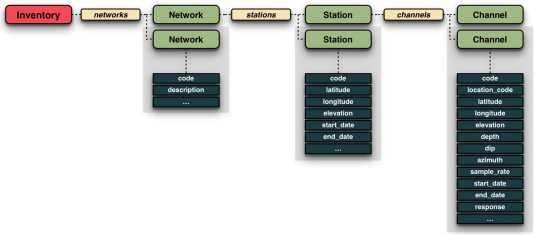
Process Flowchart

1. Check for new Events

- Monitor **FDSNWS-event** for new seismic events
- **Filter** events based on magnitude characteristics, origin quality metrics and geographical restrictions

2. Select Inventory

- Retrieve Inventory from **FDSNWS-station** and/or file in **StationXML** format
- **Filter** inventory based on station priority, specified in configuration
- Select inventory based on **distance rules**



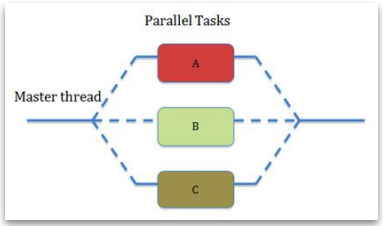
3. Select Waveforms

- Retrieve Waveforms from **SeedLink** and/or **SDS** mseed archiving and/or **FDSNWS-dataselect**
- **Filter** waveforms by quality check modules: clipping, signal-to-noise, “mouse”, psd (power spectral densities), in **parallel** CPU processing
- **Remove instrument response** in **parallel** CPU processing
- Apply **Azimuthal coverage** and **prioritization** rules

Process Flowchart

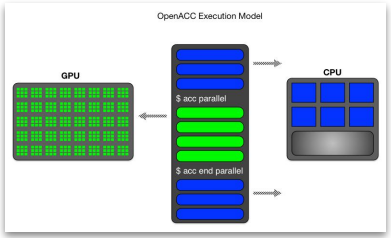
4. Compute Green's Functions

- Calculate Green's Functions of one or more adjustable **3D spatial grids,** specified in configuration, in **parallel** computation with CPU threads



5. Compute Inversions

- Calculate Moment Tensor Inversions for one or more adjustable **4-D spatio-temporal grids**, specified in configuration, in **parallel** computation with **GPU** or **CPU** threads



6. Distribute Results

- Render an **HTML** webpage with the evaluated results
- Publish results in different formats, e.g. **QuakeML**, EMSC text format
- **E-mail Notification** with the best Moment Tensor solution
- Apply user-defined script e.g. Focal Mechanism computation ingestion in **SeisComP**

1.Check for new Events

- ➔ Read Seismic Events from the **FDSNWS-event**
- ➔ **Real-Time**, **Playback** or **Event-specific** operations are supported with adjustable parameters
- ➔ **Filter events** based on magnitude characteristics, origin's quality metrics and geographical restrictions

In the example, Quality metrics have been specified for restricting seismic events to the desired ones

```
# config.yaml
# This service provides gisola with event info
Event:
  Host: http://eida.gein.noa.gr
  # Trigger Event after Quality characteristics are
reached
  # Uncertainty threshold or null for disabling
  Quality:
    Magnitudetype: MLh # or null if not specifications
is wanted
    Time: 1 # Uncertainty
    Depth: 10 # Uncertainty
    Latitude: 15 # Uncertainty
    Longitude: 15 # Uncertainty
    Magnitude: 10 # Uncertainty
    Timeout: 120 # sec. After timeout, event is
triggered anyway

  # The Watcher Service monitors FDSNWS-event
  # for new events and triggers the auto MT procedure
  Watcher:
    Range: 120 # check interval in sec
    Playback: 0 # set sec for historic run
    Historical: false # set playback option
    Geobox: (20.3054,37.2037), (22.4079,33.6323),
(27.0569,33.8793),
            (30.3113,35.8225), (27.4245,41.5998),
(24.9779,42.0729),
            (21.7796,41.8349), (18.6751,40.6285),
(20.3054,37.2037)
             # null indicates not bounds
```

2.Select Inventory

- ➔ Read Inventory from the **FDSNWS-station** and/or file in **StationXML** format
- ➔ Offers **multiple sources** and prioritization by operator and **sources redundancy** in failover cases
- ➔ **Filter inventory** based on station priority values, specified in file. Higher value indicates higher priority in selection. Zero value indicates blacklisting

In the example, the program will first read inventory from the first source i.e. StationXML, then it will append inventory from the second source, etc. In case of duplicated stations, the first occurrence will be kept

```
# config.yaml
Inventory:
  # Reads inventory from the file. You are responsible
to provide updated
  # XML file for real-time processing (e.g. daily
inventory update cron-job).
  # If set to null or file is not found, most updated
inventory will be
  # retrieved from the FDSNWS hosts. However, more time
will be consumed
  # for retrieval
  # tokens are not used in inventory retrieval,
therefore null is set
  Service:
    - [StationXML, inventory.xml]
    - [FDSNWS, http://eida.gein.noa.gr, null]
    - [FDSNWS, http://eida2.gein.noa.gr, null]
    - [FDSNWS, http://eida.koeri.boun.edu.tr, null]
    - [FDSNWS, http://webservices.ingv.it, null]

  # value of 0, indicates blacklisted station in
./stations file
  WhiteList:
    Filepath: ./stations
    Priority: 2  # default 2
    Rules:
      - [3.5, 4.0, 1]
      - [4.1, 4.5, 5]
      - [4.6, 5.0, 3]
      - [5.1, 5.8, 4]
      - [5.9, 9.0, 5]
```

2.Select Inventory (2)

- ➔ Use only accepted type of **components**, specified in configuration
- ➔ **Filter inventory** based on Distance Rules
- ➔ **Distance Rule** is triggered by the origin's magnitude and indicates the range in km and channel type of the accepted stations. Multiple rules can be appended

In the example, if an event of 5.8 Magnitude occurs, two rules will be triggered. Inventory will be filtered by stations that exist on the range of 50-450 km of BH/HH channel types and on the range of 50-150 km of HN/EN channel types

```yaml
# config.yaml
Inventory:
  # Accepted types of components
  Components: ['Z23', 'Z12', '123', 'ZNE']

  # Retrieve channel types based on distance rules
  # Distance rules has the form of: minMag,
maxMag, minDist, maxDist, channels
  Distance:
    - [3.5, 4.0, [5, 150, ['BH','HH']]]
    - [4.1, 5.0, [10, 250, ['BH','HH']]]
    - [5.1, 5.5, [40, 300, ['BH','HH']]]
    - [5.1, 5.5, [20,  100, ['HN','EN']]]
    - [5.6, 6.0, [50, 450, ['BH','HH']]]
    - [5.6, 6.0, [50, 150, ['HN','EN']]]
    - [6.1, 9.0, [100, 700, ['BH','HH']]]
```

3.Select Waveforms

- ➔ Retrieve Waveforms from **SeedLink** and/or **SDS mseed archiving** and/or **FDSNWS-dataselect**
- ➔ Setting **token** for **restricted** data is also supported
- ➔ Offers **multiple sources** and prioritization by operator and **sources redundancy** in failover cases
- ➔ Filter waveforms based on check modules: **signal-to-noise**, **clipping**, **long-period disturbances (mouse)**, **psd (power spectral densities)**

In the example, the program will first read traces from the first source i.e. SeedLink, then if it cannot find them or the service is down, it will try to retrieve them from the second source and append them, etc. In case of duplicated waveforms, the first occurrence will be kept. Psd check module is not enabled

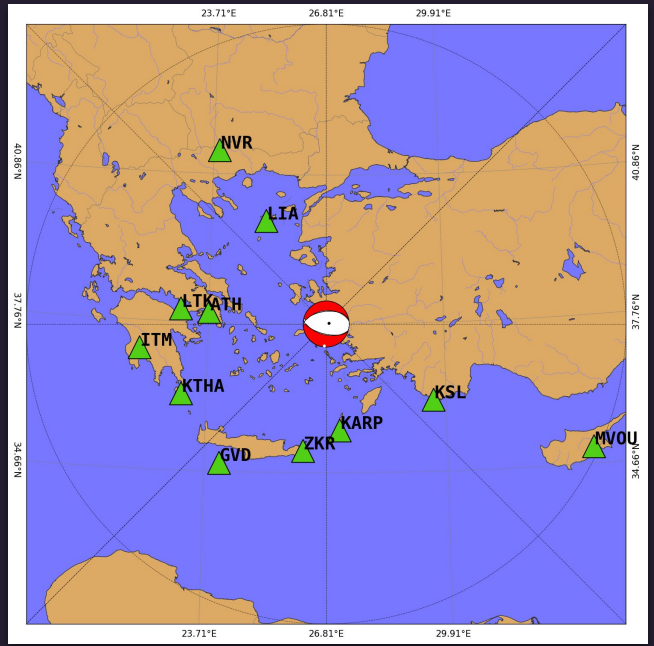
```
# config.yaml
# The Stream Service
Stream:
  # If you add more that one FDSNWS Hosts, results
will be merged according to ObsPy's policy
  # Service can have a value of: FDSNWS, SDS
  # FDSNWS accepts multiple hosts but SDS only one
  Service:
    - [SeedLink, eida.gein.noa.gr:18000]
    - [SeedLink, eida2.gein.noa.gr:18000]
    - [SDS, archive]
    - [FDSNWS, http://eida.gein.noa.gr, null]
    - [FDSNWS, http://eida2.gein.noa.gr,
KGLTIAS123SF]
    - [FDSNWS, http://eida.koeri.boun.edu.tr,
null]

  # filter data by the defined only modules
  Modules: ['clip', 'snr', 'mouse']
```

3.Select Waveforms (2)

- Apply **Azimuthal** coverage filtering. **Number of sectors** (that contain at least one station) in order to continue the MT calculation and the number of accepted **stations per sector**. There are 8 sectors of 45 degrees each
- Apply **prioritization** rules in case of more stations than stations per sector are found. Prioritization order: i) priority value, ii) broadband first then strong-motion, iii) number of QC passed components, iv) lower distance from epicenter

In the example, the program will continue to run only if 3 or more sectors have at least one station each and maximum of 2 stations each

```
# config.yaml
Inventory:
  # sectors, stations per sector # null both
for disabling
  Azimuth: [3,2]
```



4.Compute GFs

- ➔ Green's Functions are computed by a recent version of the Axitra, highlighting the power of **simultaneous processing** in the CPU domain
- ➔ **Adjustable 3-D** spatial MT solution search grid based on Magnitude size
- ➔ Operator can submit **multiple grids** for the same magnitude size range
- ➔ Operator can define **different applications** (sparse, dense matrix based on needs)

In the example, if an event of 5.8 Magnitude occurs, the MT search grid will contain points of range 0-9 km with a step of 4 km and 15-21 km with a step of 5 km (concerning the epicentral distance) and 0-31 km with a step of 2 km from origin's depth estimation above and below

```
# config.yaml

# Green's Functions computation
Green:

  Grid:
    - Rule: [3.5, 5.5] # mag condition
      Distance:     # maxValue is NOT reached (+1
if you want to reach it)
        - [0,9,5] #2  # minValue, maxValue,
stepValue in km
      Depth: # relevant to init depth
        - [0,31,5] #2

    - Rule: [5.6, 6.5]
      Distance:
        - [0,9,4]
        - [15,21,5]
      Depth:
        - [0,31,2]

    - Rule: [6.6, 9.0]
      Distance:
        - [0,31,5]
      Depth:
        - [0,31,2]
```

4.Compute GFs (2)

- ➔ **Crustal models** are set on specific geographical areas or null for any case (default)
- ➔ **Multiple crustal models** can be defined and calculated. In this case, the program will choose to submit the best crustal model, according to the higher variance reduction value of the MT solution that will be produced
- ➔ Multiple crustal models **for multiple grids** is also supported

In the example, if event's location is inside one of the specified geobox polygons, the respective crustal will be triggered. If no polygon is prescribed, null will be used

```
# config.yaml

Crustal:
  - Filepath: crustals/drakatos.vz
    Geobox: (22.6163,37.6465), (23.3432,36.8208),
(24.6418,36.6669), (25.6634,37.5202),
(24.9950,38.6259), (23.6373,39.4235),
(22.3654,39.2644), (21.8251,39.1394), (22.6163,37.6465)

  - Filepath: crustals/hasslinger.vz
    Geobox: (19.2844,39.7272), (19.3869,39.6062),
(20.6181,36.2835), (21.0868,36.3442),
(21.6460,36.3353), (22.4078,36.3616),
(22.4193,36.8072), (22.3387,37.8602),
(22.1245,38.5904), (21.5799,39.8288),
(20.8563,41.2116), (19.2330,40.5713), (19.2844,39.7272)

  - Filepath: crustals/karagianniS.vz
    Geobox: (23.0601,34.3852), (23.6039,34.0317),
(26.8522,33.6953), (29.1178,34.7188),
(29.9628,36.1148), (27.4820,37.5490),
(26.4968,37.6732), (21.4060,37.0584), (23.0601,34.3852)

  - Filepath: crustals/novotny.vz
    Geobox: null # null indicates default: if none of
the above crustal models match use this
```

5.Compute Inversions

- **GPU**-based multiprocessing implementation
- With automatic **fallback to CPU**-based multiprocessing in case of GPU hardware absence
- Newer version of code with sub-programs unification, to **minimize I/O operations**
- 3 Types of **Rules**: i) Time search shifts (4th dimension of MT grid search) ii) Seconds of inversion procedure iii) Frequencies where the inversion procedure occurs

In the example, if an event of 6.2 Magnitude occurs, two frequency ranges will be triggered and 409.6 sec will be used in the inversion. Time search is given in units of dt, where dt=409.6/8192=0.05 s. MT solution with the best variance reduction will be chosen

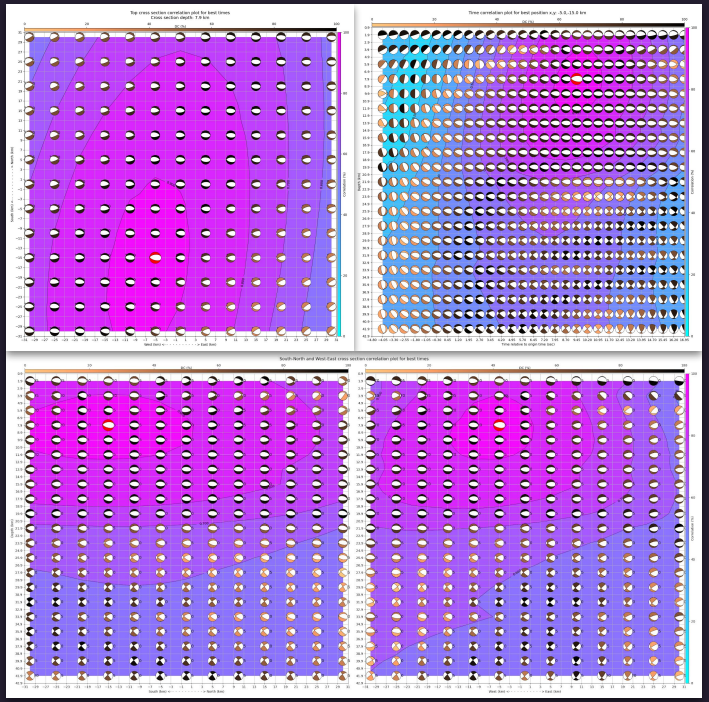
```yaml
# config.yaml

TimeShift:
  - [3.5, 4.5, [-67, 10, 167]]
  - [4.6, 5.5, [-51, 10, 161]]
  - [5.6, 9.0, [-81, 15, 341]]

Window:
  - [3.5, 4.5, 245.76]
  - [4.6, 5.5, 327.68]
  - [5.6, 9.0, 409.6]

Frequency:
  - [3.5, 5.5, [0.04, 0.05, 0.08, 0.09]]
  - [5.6, 5.9, [0.02, 0.03, 0.06, 0.07]]
  - [6.0, 7.0, [0.01, 0.02, 0.05, 0.06]]
  - [6.0, 7.0, [0.007, 0.008, 0.02, 0.03]]
  - [7.1, 9.0, [0.007, 0.008, 0.02, 0.03]]
```

5.Compute Inversions (2)

- ➔ **4-D** spatio-temporal MT grid search
- ➔ **Map View** of best correlations (for best depth and best correlated time search)
- ➔ **South-to-North** and **East-to-West** Vertical Cross-Section view of best correlations (for best correlated time search)
- ➔ **Time-Depth** view of best correlations (for best centroid position)

Map View

Time-Depth

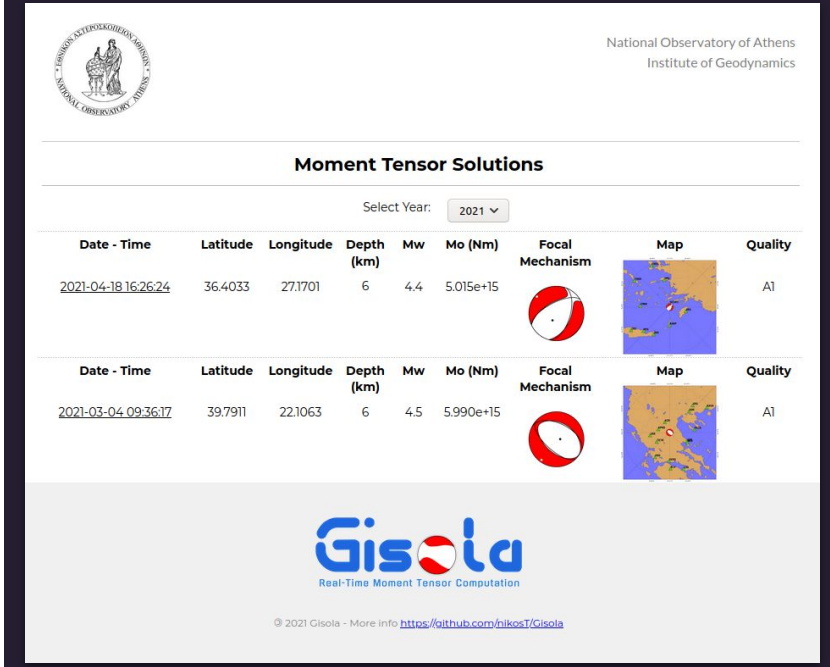


S/N vert. spatial Cross-Section

W/E vert. spatial Cross-Section

6.Distribute Results

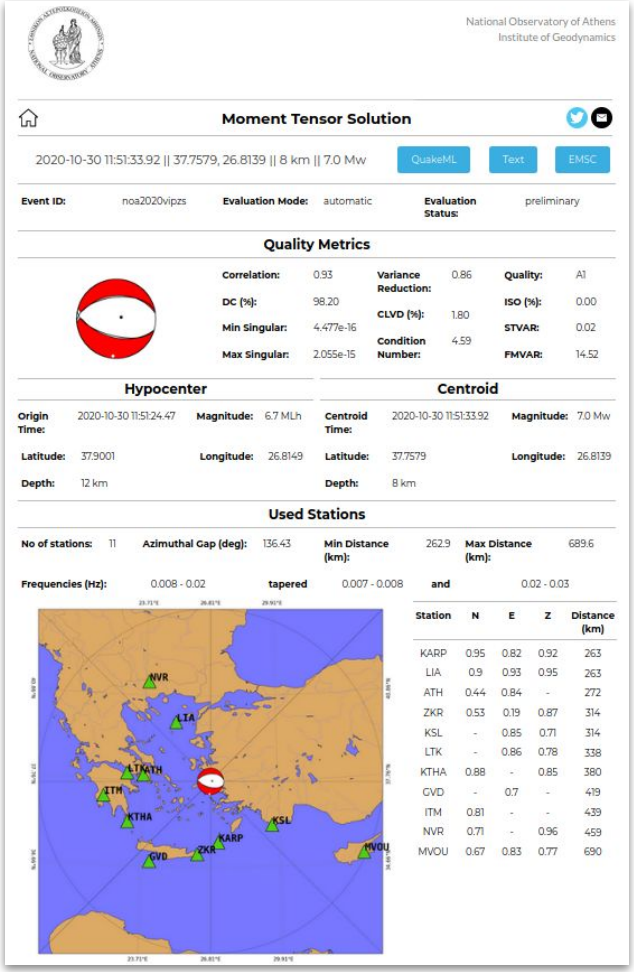
- ➔ Render an **HTML** webpage with the evaluated results
- ➔ Publish results in different formats e.g. **QuakeML**, ASCII text format etc
- ➔ **E-mail Notification** with the best Moment Tensor solution
- ➔ Apply user-defined script e.g. Focal Mechanism computation ingestion in **SeisComP**



Event HTML Page

The User can review the MT solution of a particular Event

The User can easily share it through auto-generated buttons for e-mail or tweet



National Observatory of Athens
Institute of Geodynamics

Moment Tensor Solution

2020-10-30 11:51:33.92 || 37.7579, 26.8139 || 8 km || 7.0 Mw

QuakeML | Text | EMSC

Event ID: noa2020vipzs **Evaluation Mode:** automatic **Evaluation Status:** preliminary

Quality Metrics

Correlation:	0.93	**Variance Reduction:**	0.86	**Quality:**	A1
DC (%):	98.20	**CLVD (%):**	1.80	**ISO (%):**	0.00
Min Singular:	4.477e-16	**Condition Number:**	4.59	**STVAR:**	0.02
Max Singular:	2.055e-15			**FMVAR:**	14.52

Hypocenter

Origin Time: 2020-10-30 11:51:24.47 **Magnitude:** 6.7 MLh
Latitude: 37.9001 **Longitude:** 26.8149
Depth: 12 km

Centroid

Centroid Time: 2020-10-30 11:51:33.92 **Magnitude:** 7.0 Mw
Latitude: 37.7579 **Longitude:** 26.8139
Depth: 8 km

Used Stations

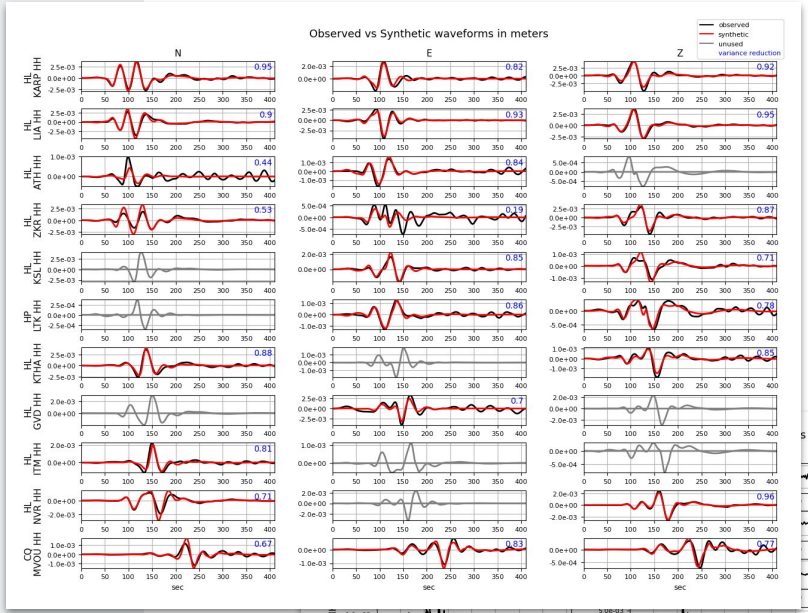
No of stations: 11 **Azimuthal Gap (deg):** 136.43 **Min Distance (km):** 262.9 **Max Distance (km):** 689.6

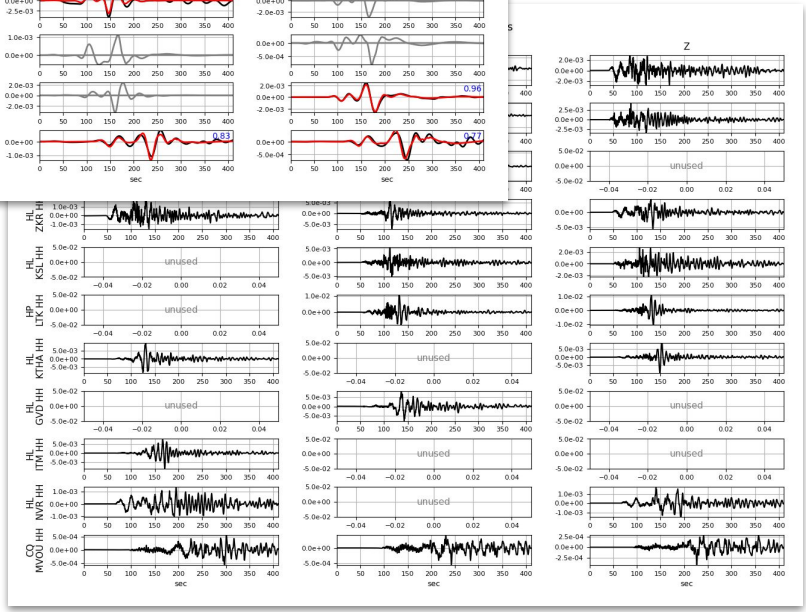
Frequencies (Hz): 0.008 - 0.02 **tapered** 0.007 - 0.008 **and** 0.02 - 0.03

Station	N	E	Z	Distance (km)
KARP	0.95	0.82	0.92	263
LIA	0.9	0.93	0.95	263
ATH	0.44	0.84	-	272
ZKR	0.53	0.19	0.87	314
KSL	-	0.85	0.71	314
LTK	-	0.86	0.78	338
KTHA	0.88	-	0.85	380
GVD	-	0.7	-	419
ITM	0.81	-	-	439
NVR	0.71	-	0.96	459
MVOU	0.67	0.83	0.77	690

Waveform Analysis

Part of the waveform figures produced and shown in the Event Page





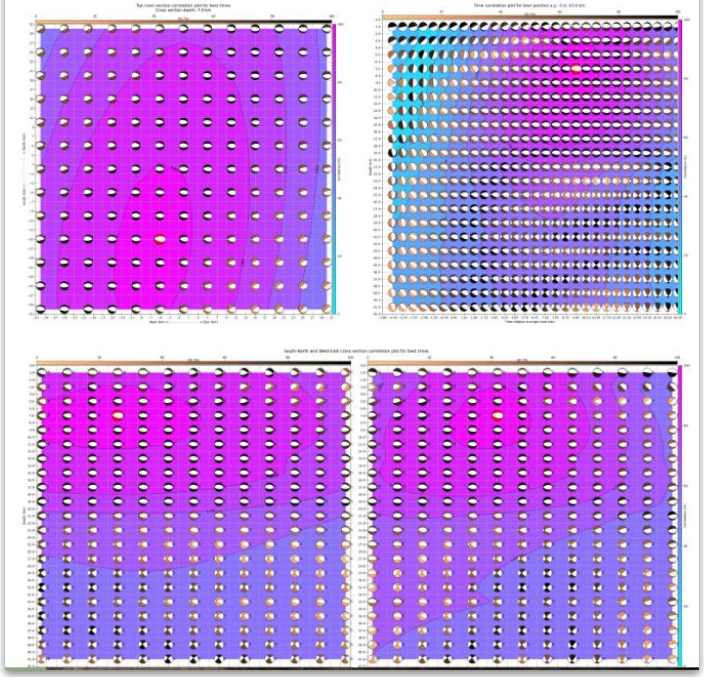
Cross Section Plots
and
best MT solution

Part of the correlation figures produced and shown in the Event Page

4-D Moment Tensor Grid Search

Best Moment Tensor Solution

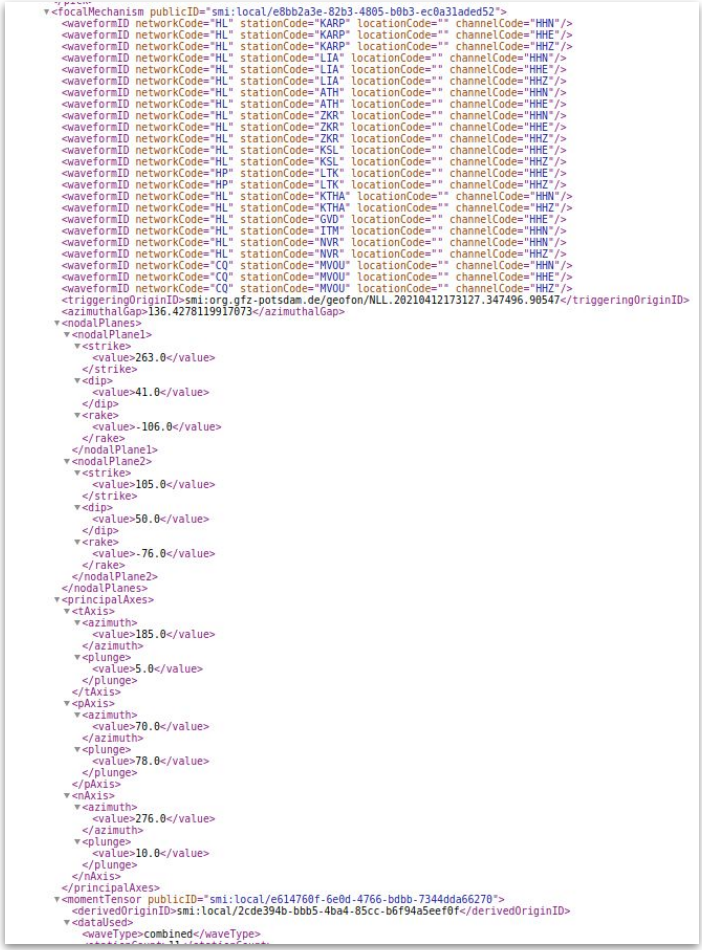
NP1	**Strike:**	263	**Dip:**	41	**Rake:**	-106
NP2	**Strike:**	105	**Dip:**	50	**Rake:**	-76
P-Axis	**Azimuth:**	70	**Plunge:**	78		
T-Axis	**Azimuth:**	185	**Plunge:**	5		
B-Axis	**Azimuth:**	276	**Plunge:**	10		
Six Tensor Components	**Mrr:**	-3.008e+19	**Mtt:**	3.129e+19	**Mpp:**	-1.210e+18
	Mrt:	-4.890e+18	**Mrp:**	6.245e+18	**Mtp:**	-2.474e+18



QuakeML v.2 XML format

The User is able to view/download the MT solution in QuakeML v.2 XML format

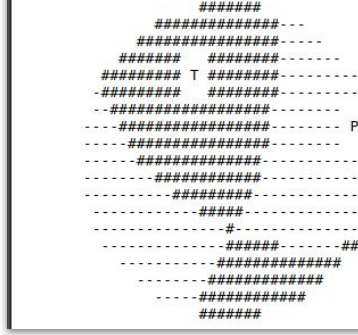
The User is also able to submit the generated QuakeML with the particular Focal Mechanism back to SeisComP



ASCII text format

The User is able to view/download the MT solution in EMSC accepted text format

```
=====================================
======= Moment Tensor Solution =======
=====================================

Hypocenter Solution (NOA)
Origin Time :  20210424 16:17:37.30
Lat:  36.8033     Lon:  20.9116
Depth (km) : 11
Mw :  4.0

=====================================
Centroid Solution
Centroid Time :  +3.99 (sec) relative to origin time
Centroid Lat:  36.9050       Lon:  20.9133
Centroid Depth : 3

=====================================
No of Stations:  8    (VLS-LTK-FSK-KTHA-ITM-VLI-PDO-KLV)
Freq band (Hz)
0.05-0.08  tapered 0.04-0.05 and 0.08-0.09
Variance Reduction (%): 47

Moment Tensor (Nm):  Exponent 10**14
  Mrr      Mtt      Mpp
2.786   3.342   -6.128
  Mrt      Mrp      Mtp
5.677   4.138   4.891
VOL (%)      : 0.0
DC (%)       : 53.2
CLVD (%)     : 46.8
Quality      : B2

Computed by: Gisola

Best Double Couple: Mo= 1.008e+15  Nm
NP1:   Strike   Dip   Rake
        121      53     22
NP2:   Strike   Dip   Rake
        17       71     141
```



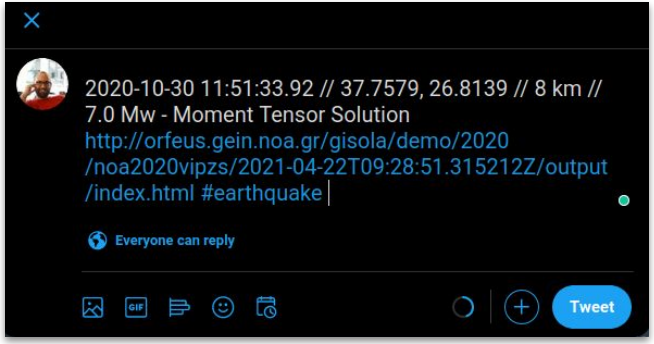
Auto-generated “Share” buttons

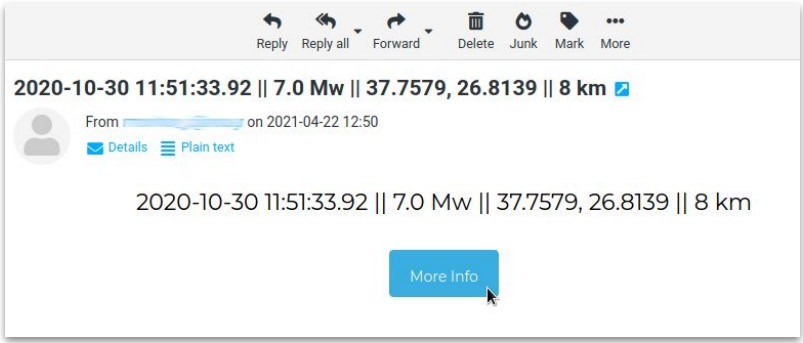
and

E-mail Notification

Recipients receive an auto-generated e-mail with the MT computation

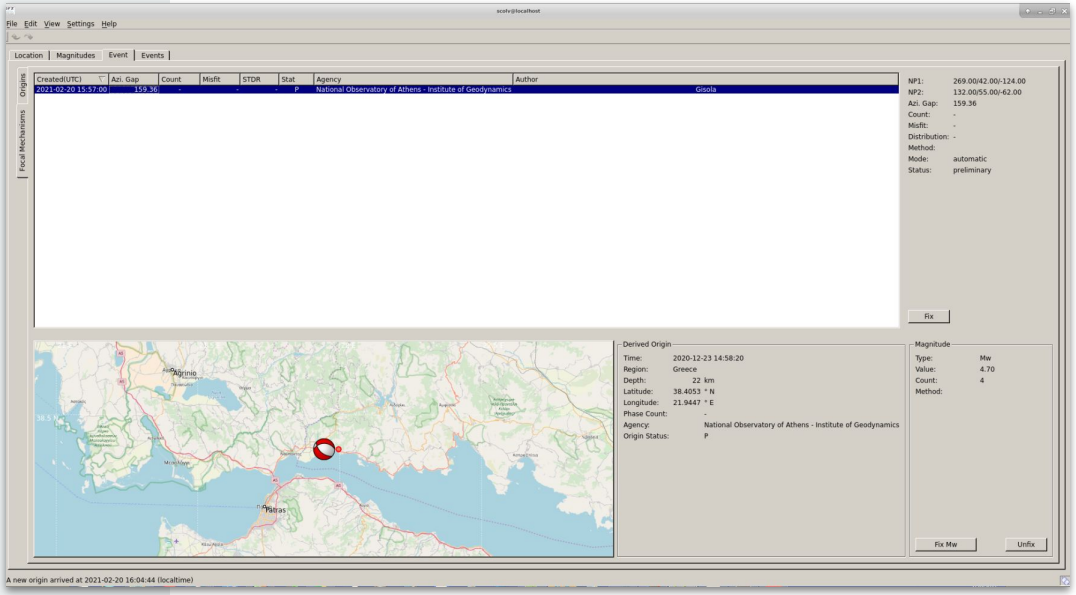
Easily share results to web and social media





Dispatch MT solution to SeisComP

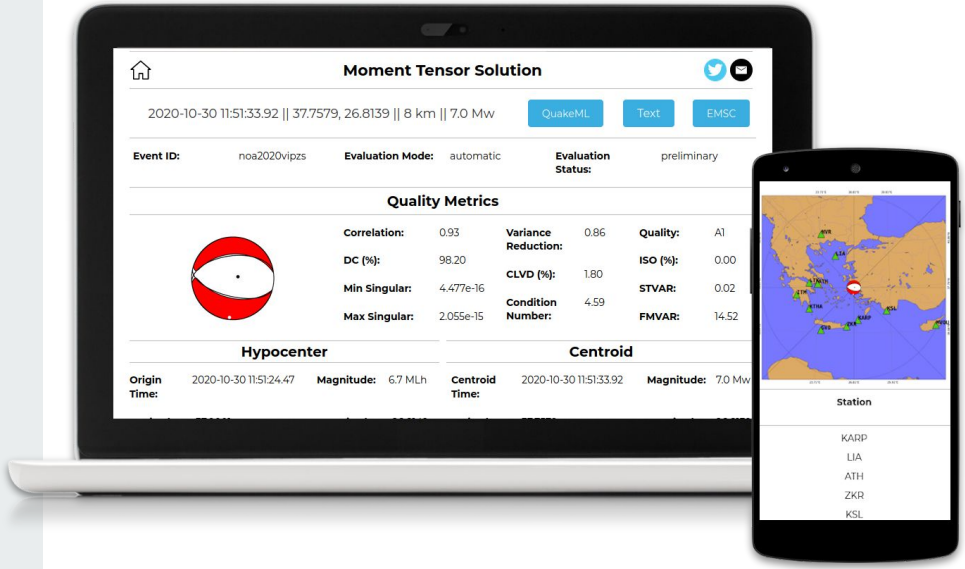
View MT solution from “scolv” module of the SeisComP software



The operator is able to submit an automatic script in order to send the MT solution back to SeisComP, using the “scdispatch” SeisComP module

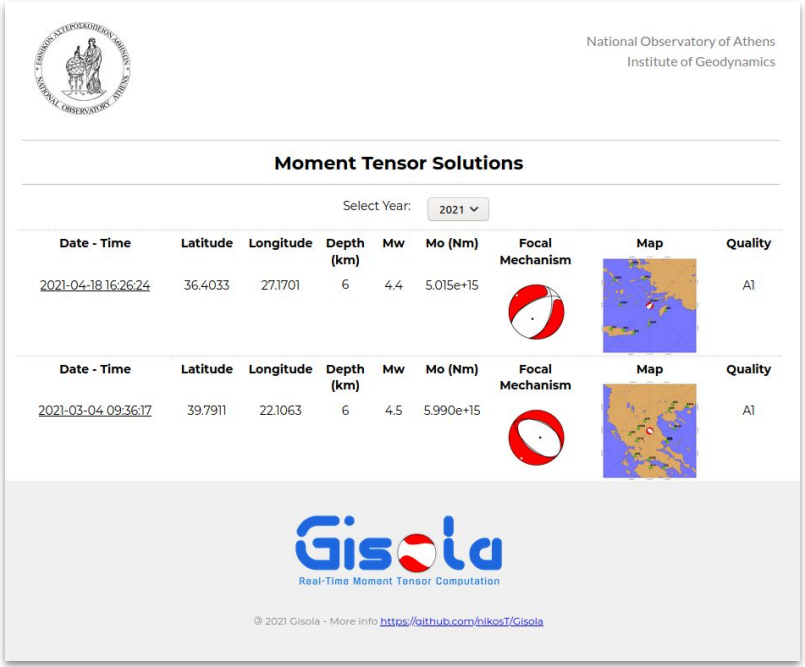
Event Page for mobile

Apart from PC, user is able to monitor the MT Solution from tablet or smartphone



Home Page of MT Solutions

The User can review/share MT solutions, packed in yearly sessions



Applications

Geo-specific Applications

- ➔ **Regional** Area (FDSNWS node/nodes specific region)
- ➔ **European** Area (EIDA Federator)
- ➔ **World-Wide** Area (EIDA Federator + IRIS Federator)

Quality-specific Applications

- ➔ **Quick** Magnitude (Mw) estimation
  Sparse 4-D spatial temporal Grid Search with minimum search criteria {Urgent Warning application}
- ➔ **Normal** Quality MT computation {Real-Time application}
- ➔ **Best** Quality MT computation
  Dense 4-D spatio-temporal Grid Search with multiple search criteria {Research application}

Catalog-specific Applications

- ➔ Monitor different **FDSNWS-event** providers **simultaneously**
- ➔ **Restrict Events** based on:
  - ❖ Geolocation
  - ❖ Magnitude threshold
  - ❖ Magnitude Type
  - ❖ Origin's Quality (time, depth, latitude, longitude, magnitude uncertainty)

Command Line Interface

Event-based Commands

```
# calculate MT for two particular events (rest info retrieved from FDSNWS-event)
# use specific configuration file according to needs
python3 ./gisola.py --event noa2020diego --event noa2021asfg -c config_realtime.yaml

# calculate MTs for all events found in datetime range (rest info retrieved from FDSNWS-event)
python3 ./gisola.py --datetime-range 2020-01-01T00:00:00 2021-01-01T00:00:00

# calculate MT for particular event (without FDSNWS-event)
# use default configuration file (./config.yaml)
python3 ./gisola.py --event 2021-04-25T15:13:39 3.0 ML 37.24 20.49 4.1

# calculate MT for particular event (rest info retrieved from FDSNWS-event)
# use specific configuration file according to needs
# override station selection and use the respective stations,
# from last station given, ignore the Z component
python3 ./gisola.py --event noa2020diego --event-c config_dense.yaml --station ATH KLV APE LIA..NE
```

Event-based Commands (2)

```
# calculate MT for particular event (rest info retrieved from FDSNWS-event)
# use specific configuration file according to needs
# remove respective stations and components from procedure,
# from last station given,remove only the Z component
# override crustal model and apply user-given
python3 ./gisola.py --event noa2020diego -c config_realtime.yaml --station KLV ATH..Z --remove
--crustal novotny.vz

# quick revision of an already calculated MT for particular event (rest info retrieved from
FDSNWS-event)
# runs only inversions and results distributions (all previous step are ignored)
# use specific configuration file according to needs
# remove respective stations and components from procedure,
# from last station given, remove only the Z component
# compute inversions on specific frequencies range
python3 ./gisola.py --event noa2020diego -c config_realtime.yaml --station KLV ATH..Z --remove
--frequency 0.04 0.05 0.08 0.09 --revise
```

Simultaneous Cron-Job Examples

```
# monitor every minute for new events and perform a normal european MTs computation
# (e.g. events >= 5MLh)
* * * * * python3 ./gisola.py -c config_europe.yaml -l log_europe --real-time

# monitor every minute for new events and perform a normal regional MTs computation
* * * * * python3 ./gisola.py -c config_greece.yaml -l log_greece --real-time

# monitor every minute for new events and perform a quick Magnitude(Mw) estimation
# (e.g. sparse spatio-temporal grid and minimum configuration)
* * * * * python3 ./gisola.py -c config_mw.yaml -l log_mw --real-time
```

Sources (e.g. FDSNWS) are defined in the configuration file (yaml)

Future Steps

- ➔ Green’s Functions Pre-Computation
- ➔ Neural Networks and deep learning for event-driven configuration
- ➔ Improved mechanism for characterizing waveforms quality

Live Demo

- ➔ Visit: http://orfeus.gein.noa.gr/gisola/demo/2020
- ➔ Select any of the 2 available years
- ➔ Click on the date-time field of any of the events to review MT solution

Gisola

Real-Time Moment Tensor Computation

Source Code

https://github.com/nikosT/Gisola

GitHub