International Atomic Energy Agency
and
United Nations Educational Scientific and Cultural Organization

INTERNATIONAL CENTRE FOR THEORETICAL PHYSICS

ESAA

Environment for Seismic Activity Analysis

Shu-yu ZHANG* Bai-lin HAO†
International Centre for Theoretical Physics, Trieste, Italy

MIRAMARE – TRIESTE
September 1994

*Permanent address: The Institute of Physics, The Chinese Academy of Sciences, P. O. Box 603, Beijing 100080, China.

†Permanent address: The Institute of Theoretical Physics, The Chinese Academy of Sciences, P. O. Box 2735, Beijing 100080, China.

Abstract

ESAA is an X-window based, graphical and interactive, software system for analyzing seismic activity, using the earthquake catalogs of a given region as input. Basic design idea and structure of this system, as well as the progress in its implementation are reported.

1 Introduction

We start with some general considerations on forecasting natural processes. Any forecasting is an attempt to infer the future behavior from the understanding of the past. The understanding and inference must be based on a certain kind of theoretical paradigm. Essentially speaking, there are only three paradigms, namely,

1. Assuming that the process under study is a stochastic one, statistical methods of prediction are employed. The approach centers on the calculation of various moments from as long as possible records and on observation of deviations of these moments from recent data as compared to long-term averages.

2. Assuming that the process under study is deterministic, dynamical informations are extracted from the data. The essence of this approach consists in estimating various periodic components and the dynamical process is approximated by expansion through periodic solutions.

3. Assuming that the process under study is a self-similar one, the prediction makes use of scaling-invariant properties. A key notion here is scaling exponent, e.g., the dimension. In seismology there are many empirical formulae of the type

$$y = a + b \log x,$$

   the very form of which suggests a scaling law.

The progress of nonlinear dynamics in recent years has provided more knowledge and experience for the last two categories of forecasting. However, one must be aware of the following basic points.

First, the three aforementioned assumptions may be clearly distinguished, in principle, only when infinite long measurements have been performed on the natural process and an infinite amount of data are collected. For a finite data set, any one of the three approaches is necessarily an approximate description.

Second, the merit or demerit of one or another approach is dictated by the natural process itself and must be determined from repeated analysis of the real data.

Speaking about earthquake prediction, the most reliable data come from seismic activity itself. In principle, a great amount of data may be obtained from a single earthquake, including records of various types of seismic waves in different directions and distances, as well as in wide frequency ranges. These data may provide detailed information on the source region and on the region the seismic waves have propagated through. However, in the first stage of the ESAA Project, we only take 4 or 5 numbers from one earthquake, namely, time, longitude, latitude, depth, and magnitude. Put in other words, the analysis is based on the earthquake catalog of a given region. This kind of analysis, based on a rather small data set, can only provide very crude results. Our main concern is not to be restricted to the development of a certain specific method, but to create an environment for all possible kinds of analysis, using the earthquake catalog as the only input[1].

In fact, we are now in a good position to carry out the program, because Chinese seismologists have recently compiled a summary earthquake catalog of China, which covers a time span from 768 B. C. to 1985 A. D. and contains more than 300 000 records[2]. In fact, more than 100 000 new records have been accumulated since 1985. Our main motivation to develop the ESAA system is to make maximal use of these precious data.

2 Object-Orientated Programming

The advance of computing technology has led to such a situation that, generally speaking, the development of software can hardly keep up with the progress of hardware. Accordingly, in scientific computing we are facing a transition from the paradigm of "program modules plus libraries" to the creation of computing and research environments where the initiative of human being and the capability of machine is being integrated in an ever more interactive way. In view of programming philosophy, we are facing a transition from Procedure-Oriented Programming (POP) to Object-Oriented Programming (OOP). In fact, this transition has taken place in the mid 1980s.

As the notion of OOP is still not vary popular among those dealing with earthquakes and the ESAA system is based on the X Window System[3], which is an object-oriented software, we explain briefly the idea of OOP[4].

In order to grasp the essence of OOP, it is instructive to look at POP retrospectively. In a sense, the programming technique, using one or another language, familiar to us since the late 1950s, let it be the early ALGOL and FORTRAN, or the more recent C, all come under POP. A program consists of many procedures (functions or subroutines); the most frequently used procedures are collected into scientific libraries, which are shared among the users. The realization of any procedure is, in turn, based on the idea of structured programming, by making use of sequential, conditional (branching) and repeated (loop) structures. There are various data structures in POP. However, the relation between data structure and procedure is rather loose. For example, from a declaration like

```
INTEGER A(100)
```

one cannot tell what action will be taken upon the array `A`. To run a POP program means basically calling various procedures in a predefined order. It is true that the calling order may be somehow changed in the run time, however, these possibility must be programmed into the procedures before hand, using condition tests or "interruptions". A programmer cannot change the organization of the procedures at his will at run-time.

In a POP system, all data structures are concrete. Formally, there is no restriction on what types of operations may be carried out on a data structure. However, it is just this "flexibility", which makes the running of such programs rather inflexible.

The basic idea of OOP consists in abstraction of data structure. On one hand, many classes and subclasses of data structure are defined in quite general form as frameworks which will be materialized when necessary. On the other hand, in the definition of data structures not only their properties (attributes) are declared as we do in OOP, but also the allowed operations on the data structure must be defined. Any unspecified operation cannot be used in the program. This second aspect is specific to OOP. This kind of data structure is called an *object*, hence object-oriented programming. An object is a data structure with operations defined or a data structure characterized by the operations. From a formal point of view we no longer have the free relation between data structure and operation. However, these objects may provide a rather free environment for the program to run.

Writing an object-oriented program consists in the definition of objects and in the registration of these objects with a dispatcher, called a *notifier*. Running an object-oriented program means to activate and to call the objects dynamically, the number of objects being activated and the order of calling them are quite free. They are activated by the user as the circumstance requires by making various inputs, using mouse buttons, key strokes, etc. These inputs are called *events*. In this sense, we say that an OOP system is an "event-driven" system.

3 Structure of ESAA

At present the ESAA system comprises 4 modules:

1. A "Quake map" module which is essentially a cartographical unit. It draws and erases geographic elements such as a mesh made of longitude and latitude lines, borders between countries and provinces, as well as epicenters of earthquakes, using small polygons of different sizes or a color code for different magnitudes.

2. A "Select quakes" module which selects quakes from one or more earthquake catalogs in accordance with a given region, magnitude range and time span. We call a collection of selected quakes a *quakelist*. The region may be indicated on the map by clicking the mouse buttons or be typed in from the keyboard. All year, month, day in the catalog are converted into Julian days for convenience of using sliding time windows. The hour, minute, and second of a quake is converted into fractional part of the Julian day. Internally, a quakelist is a struct which also keeps a record of the region, magnitude range and time span, according to which the list has been selected.

3. A "Analyze quakes" module, which calculates, e.g., b-value, interspike intervals, or number of main shocks from a given quakelist. Each type of calculation may require some parameters to be given interactively. For example, one can choose the minimal

magnitude, the width of the sliding time window and the sliding step (both in days). The results of calculation are kept in separate files, whose names are derived from the quakelist names in a fixed way. The quakelist information and the calculation parameters are also kept in these files.

4. A “Show results” module which displays the results of calculations from the quakelists. In principle, there two types of results to be displayed: time variation or spatial distribution of one or another quantity. For the time being only the time variations have been implemented. The spatial distribution will make use of colors and will be added in the near future.

These modules may be called in any order by choosing items from a menu or by pressing buttons in a panel. For example, one may select and store a number of quakelists in one session, carry out various analysis on some of them in another session, and, finally, display and compare the results in a third session.

An important aspect of ESAA is its user friendly, interactive interface. It has the general “look and feel” of the X widgets, using which the system is developed. For the time being this is the XView system, based on the OPENLOOK implementation of X Window Systems[5, 6] on the Sun Sparc Workstations.

4 Analysis of Seismic Data

In principle, any method of analysis, which has been proven useful for the understanding of seismic activity, may be incorporated into the ESAA system.

The first method implemented is the time and space scanning of the b-value[7, 8] in the empirical Richter-Gutenberg relation of the number of earthquakes N and their magnitude M:

$$N = Ae^{-bM}. \tag{1}$$

It has been shown phenomenologically that b reflects the interplay of averaged stress and strength of a region and, therefore, may be used as an indicator of earthquake risk. When a catastrophic shock is being conceived, a profound lowering of b may be observed. This has been checked on the example of several earthquakes with magnitude 7 and higher. When the time and space scanning method was proposed in the late 1970s[7, 8], its power was limited by the computational and graphical capability of the computers. Now the time has come to explore in depth the potential of this technique.

The 8 or 14 functions, based on “earthquake flow” and introduced by V. I. Keilis-Borok and coworkers[9], may easily be implemented in ESAA. In fact, we have already included a few those functions, for which there was not much ambiguity in choosing the parameters. It should be noted that, being based on the same set of one or two numbers, e.g., time and magnitude, from every earthquake, these functions are not independent on each other. In fact, some of these functions are of the same nature, but not better than the b-value, mentioned before.

Other methods, such as the analysis of interspike intervals between earthquakes[10], may be implemented as well. However, the method suggested in [10] did not take into account the difference of magnitude of events, whose time interval were being examined.

We mention in passing that we are quite sceptical towards using so-called phase space reconstruction technique to earthquake data. In spite of the great success of nonlinear time series analysis in recent years, see, e.g., [11], the method is most applicable to smooth dynamical data in between bifurcations, i.e., in between sudden changes of the dynamical regime in the system due to instabilities. On the other hand, earthquake catalogs are nothing but time records of these bifurcations and instabilities. However, there exist three types of events, which are closer in nature, namely, earthquakes, rock bursts, and sound emissions in material tests, though they happen at very different energy scales. It makes sense to devise common methods for their study. The ESAA system may be of use to all of them. Nontheless, we may implement a few modules of phase space reconstruction techniques in order to compare the results.

Another comment concerning pattern cognition technique may be in order. At our present, rather limited, level of understanding of the seismic processes, it is still better to let human beings to do the job than giving it to the machine, although subjective factors may be reduced in machine cognition. We have not implemented a pattern cognition module yet, but future development in this direction is not excluded.

5 Implementations

There are two main platforms for the ESAA system: the Sun Sparc Workstations using OPENLOOK and XView interface[5, 6] to the Xlib[3], and the SGI Indigo Workstations using Motif interface to X[12]. The implementation on Sun Sparcs has been under way. The work on SGI will be soon carried out in the State Key Laboratory of Scientific and Engineering Computations in Beijing, China.

For the time being, only the X Window System itself is fully in OOP. The quakelists, the results of calculations, and the geographic elements are given as ordinary data structures. In the future, most of these data structures will be defined as objects in OOP. In principle, there may be three basic classes of objects in ESAA.

1. Earth catalogs: we distinguish the original catalogs from lists of earthquakes, which are selected according to a given `Quake_range`. We call the former *catalogs* and the latter — *quakelists*. From one and the same catalog many different quakelists may be selected for various kinds of analysis. For the time being, quakelists are defined as

   ```
   typedef struct {
     int quakenumber;
     Quake_range range;
     Quake *list;
   } Quake_list;
   ```

where Quake_range keeps the record of time span, magnitude range, and geographic region, according to which the quakes have been selected:

```
typedef struct {
  float startday;
  float endday;
  float lowermagnitude;
  float uppermagnitude;
  char  *region_name;
  Polygon region;
} Quake_range;
```

In order to make quakelists a class in the OOP sense, allowed operations on these lists will be defined as part of the data structure. These operations are nothing but procedures for calculating various functions from the quakelists.

2. The results of the aforementioned calculations comprises the second class of objects. Operations defined on them are graphic display of the results. In principle, there are two types of display, namely, $x(t)$ type which is a curve describing the time variation of a certain quantity x, e.g., the b-value, and $f(x, y)$ type which shows spatial distribution of, say, release of energy or deficiency of seismic activity.

3. The third class of objects consists of geographic elements, which in turn are subdivided into line objects and pointwise objects. State and province boundaries are examples of line elements; their attributes include line type, line width, color, and the filename where the original geographic data are stored. Cities and epicenters of earthquakes are examples of pointwise objects. They are in fact small circles or polygons, whose size and color reflect the population or magnitude range. The operations, common to all these objects, are to draw and to erase. Since maps are drawn in the X Window environment, they may be easily amplified or reduced, as well as scrolled in horizontal and vertical directions. In practice, a map is drawn on a canvas, whose size are much bigger than the visible screen, so significant resolution is guaranteed.

There are many functions in ESAA that require a suitable algorithm or even a trick or two to implement. For example, the functions to realize the cartographic projections and their inverse transformation, the function to decide whether a point falls in a given region or not, the transformation between Julian days and the usual year-month-day with leap years taken into account. We will give a detailed description of these functions in our final report on the ESAA system.

As we have said, the system has been written in C, but we plan to transform it into a full OOP system, using the C++ language. This will be done when we gain more experience in running the present system.

Acknowledgements

The work described in this report is part of a long-term project. The present authors are responsible for the overall design of the system and the implementation of the software, while the collection and calibration of the earthquake data are conducted by Li Quan-lin, Chen Jing-biao, and their colleagues in the State Seismological Bureau of China. We thank Li and Chen for continuous discussions in our collaboration ever since 1976.

The first version of ESAA was implemented by using Sunview tools on a Sun 3/260C Workstation, donated by Sun Microsystems, Inc., to the Institute of Theoretical Physics in Beijing. The rewriting of the system using the XView interface to the X Window System has been started in ITP, Beijing, and continued in ICTP, Trieste, using the Sun Sparc Workstations at the two institutions. We are grateful to the above Institutions for the use of computing facilities. The authors sincerely thank Professor Abdus Salam, the International Atomic Energy Agency and UNESCO for hospitality at the International Centre for Theoretical Physics, Trieste, during two summer visits in 1992 and 1994.

References

[1] Bai-lin Hao, "Earthquake prediction in China", *OCPA Newsletter*, *1* (1992), No. 6, 12-13.

[2] Li Quan-lin *et al.*, *Earthquake Catalog of China — From 768 B.C. to 1985 A.D.*, Seismic Press, Beijing, 1990 (in Chinese).

[3] A. Nye, *Xlib Programming Manual for Version 11*, O'Reilly & Associates, 1990; *Xlib Reference Manual*, O'Reilly & Associates, 1992.

[4] Zhang Shu-yu, and Hao Bai-lin, "Object-Orientated Programming", *Chinese J. Comp. Phys.* **9** (1992), 343-345 (in Chinese).

[5] D. Heller, *XView Programming Manual for XView Version 3*, O'Reilly & Associates, 1991.

[6] T. V. Raalte, *XView Reference Manual*, O'Reilly & Associates, 1991.

[7] Li Quan-lin, Chen Jin-biao, Yu Lu, and Hao Bai-lin, "Time and space scanning of the b-value: a method for monitoring the development of catastrophic earthquakes", *Acta Geophysica Sinica*, **21** (1978), 101-124 (in Chinese).

[8] Li Quan-lin, Yu Lu, Hao Bai-lin, and Chen Jin-biao, *Time and space scanning of the magnitude-frequency relation of earthquakes*, Seismic Press, Beijing, 1979, iii + 102 (in Chinese).

[9] A. M. Gabrielov *et al.*, "Algorithms of long-term earthquakes prediction", CERESIS, September 1986, Lima, Peru.

[10] G. P. Pavlos, L. Karakatsanis, J. B. Latoussakis, D. Dialetis, and G. Papaioannou, "Chaotic analysis of a time series composed of seismic events recorded in Japan", *Int. J. Bifur. & Chaos*, **4** (1994), 87-98.

[11] A. S. Weigend, and N. A. Gershenfeld, eds., *Time Series Prediction — Forecasting the Future and Understanding the Past*, Addison-Wesley, 1994, xvii + 643.

[12] Open Software Foundation, *OSF/Motif Programmer's Guide*, Prentice Hall, 1991; *OSF/Motif Programmer's Reference*, Prentice Hall, 1991.