We are IntechOpen,
the world's leading publisher of
Open Access books
Built by scientists, for scientists

4,800

Open access books available

122,000

International authors and editors

135M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities

BOOK CITATION INDEX

CLARIVATE ANALYTICS

INDEXED

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com

Ground Motion Estimation During Strong Seismic Events Using Matlab

Margaret Segou
Earthquake Science Center, Menlo Park, United States Geological Survey (USGS),
United States of America

1. Introduction

This chapter presents the implementation of seismological and earthquake engineering principles and the development of innovative computer code using Matlab platform. Since earthquakes remain the greatest natural disaster for modern society, causing loss of life and millions of damages to the urban environment, the efforts of earth scientists and engineers lie in providing the tools for quick and adequate assessment of potential damage.

The foundation of seismic hazard analysis is based on the accurate scientific estimation of anticipated ground motion at a site following the occurrence of a strong earthquake. The seismic parameters involved in this estimation are the magnitude of the earthquake, the distance between the epicenter and the site in question, the description of the site's geological formations and additional characteristics of the earthquake's rupture style. Nowadays, knowledge about the level of the anticipated shaking near cities or villages is directly linked with past observations. In scientific practice the above statement describes the development of elaborate empirical mathematical models, which are based on the available seismological data. Seismological data collection and analysis is a demanding time-consuming task, related with Digital Signal Processing and Data Archiving. The goal of this chapter is two-fold; firstly to provide an insight of the necessary Digital Signal Processing steps, easily performed through Matlab, leading to the derivation of earthquake engineering parameters and secondly testing traditional regression analysis and optimization in order to develop empirical equations modeling the aforementioned parameters.

2. Data processing

According to modern data acquisition practice once an earthquake, exceeding a specific threshold occurs, ground motion time-series recorded by digital seismometers or accelerometers, usually at a sampling frequency equal to 200 samples-per-second, are transmitted to the data analysis center. No matter the progress of modern technology, scientists in earthquake prone countries cannot simply ignore the older analog acceleration time series, which in many cases can be irreplaceable. Matlab through *resample1* function provides the necessary tool to produce an equally sampled time series at a given sampling interval. After this processing stage, the main objective is to remove the undesirable long period and high frequency noise, which can be attributed to various sources like the

mechanical hysteresis of the instrument or exposure to wind gasps and industrial environment (Segou et al., 2008). The subsequent processing steps are related with:

1. Visualization of time series and calculation of the peak ground acceleration parameter
2. Computation of Fourier amplitude spectrum
3. Filter design of the appropriate Infinite Impulse Response filter for the specific time series
4. Phase Preserving Implementation of the -previously designed- filter in the frequency domain
5. Graphical comparison of the Fourier amplitude velocity spectrum for the filtered and unfiltered time series, to determine whether the noise of the record has been successfully removed
6. Computation of the response acceleration spectra
7. Calculation of earthquake engineering parameters, like spectrum intensity (SI), useful to assess potential structural damage.

For the aforementioned steps a number of Matlab functions such as *fft* and *ifft* for domain conversion and *butter* for Infinite Impulse Response (IIR) filter design of Butterworth type have formed the core of Proschema software (Segou & Voulgaris, 2010), developed in Matlab R2009a version.

The visual inspection of the time series at the beginning of strong motion processing allows to determine the quality of the recording, decide whether removal of spurious spikes, known as *despiking*, is needed or if any other pre-processing steps are required. Figure 1 displays the time series of acceleration through the *plot* function using *linspace* function to derive the time line of the horizontal axis based on the sampling rate of the instrument in the field. Peak ground acceleration value (PGA) represents the maximum absolute amplitude of this acceleration time series (Amp) and it can be calculated using the *max* and *abs* functions.
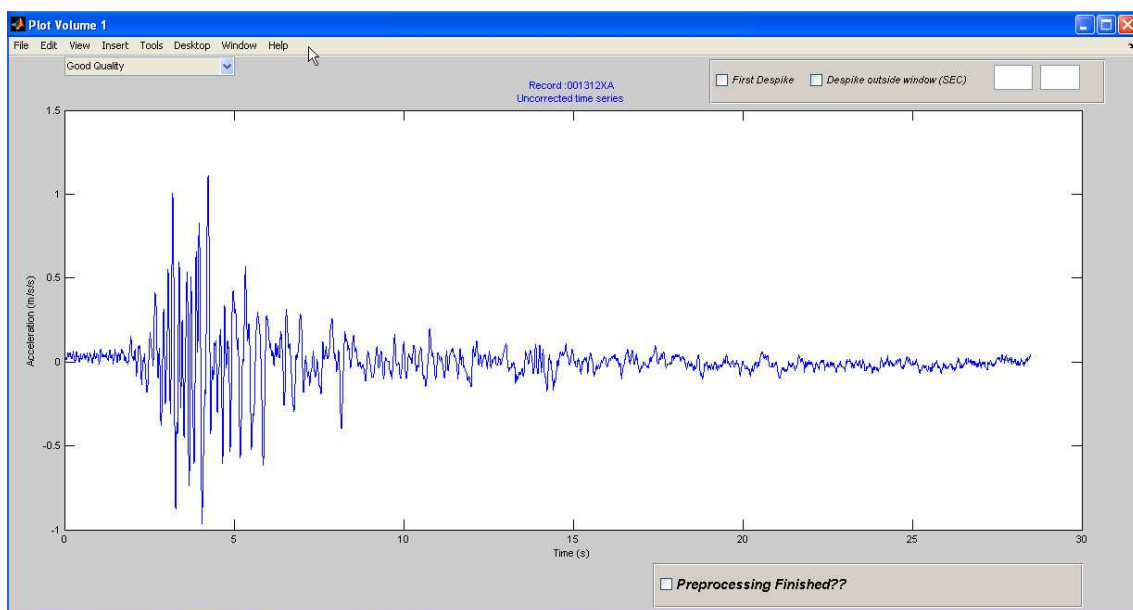

Fig. 1. Time series of acceleration.

In strong motion processing it is usual to implement a phase-preserving IIR filter in order to avoid phase delays, which will eventually distort the onset of the earthquake. Figure 2

presents a comparison between causal and phase–preserving filtering during strong motion processing. Figure 3 presents the uncorrected acceleration and filtered acceleration time series using five different pairs of cut-off frequencies combined in a phase preserving pass-band Butterworth filter's implementation.
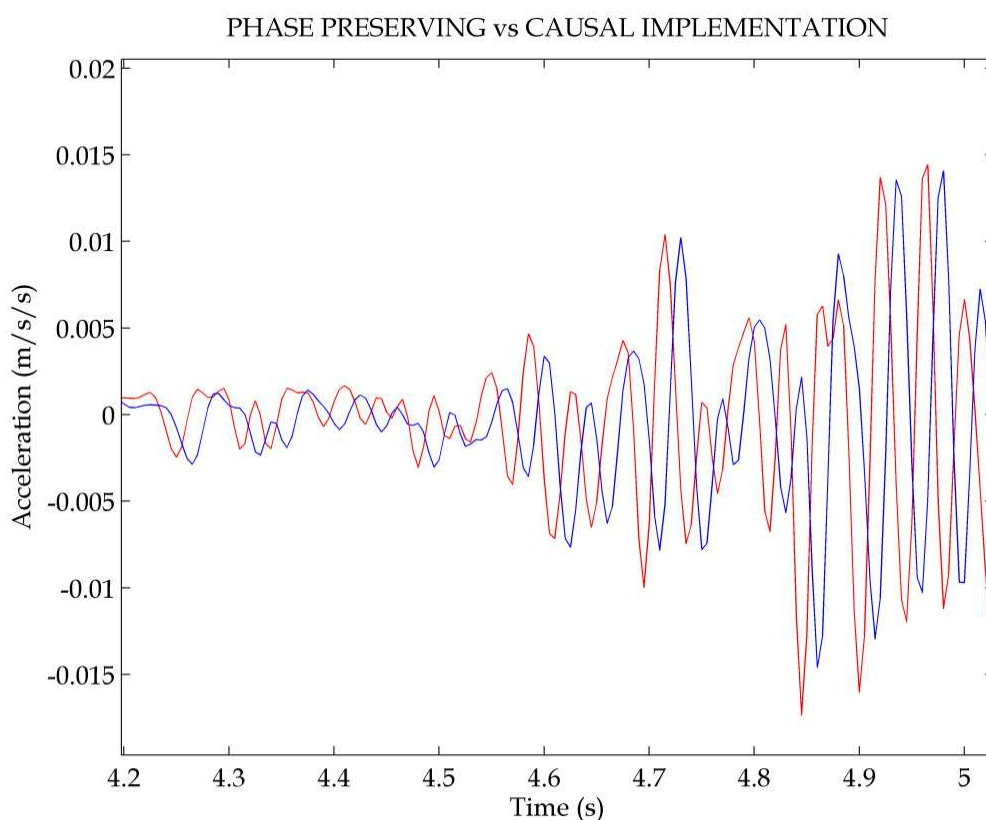

Fig. 2. Comparison between phase-preserving (in red) and causal (in blue) implementation of an IIR filter.

Another important aspect is the integration of a sinusoidal signal, such as the acceleration time series in this case study (Figure 1). This computation in the frequency domain corresponds to the convolution of Fourier amplitude spectrum with the frequency response of the perfect integration operator, equal to $1/i\omega$, whereas for differentiation the frequency response of the operator is just the inverse of the perfect integrator, simply $i\omega$ (Karl, 1989). After convolution the user can easily select the real part of an array of complex numbers, such as the Fourier amplitude spectrum, by using the *real* function.

Immediately after filtering the acceleration time series the inspection of velocity and displacement time series (Figure 4), calculated after single and double integration respectively, is required in order to determine whether the high-frequency and long period noise has been removed adequately from the records.

In more elaborate mathematical calculations, related with the response of a single degree of freedom (SDOF) harmonic oscillator $\ddot{u}$ of a specific damping level ζ subjected to an acceleration time series x (Equation 1a), the computational effort required is greater. In order to calculate the response spectral acceleration S_A at a given period ω, the user should define the maximum of the oscillator time series (Equation 1b) for this specific period.

$$\ddot{x} + 2\zeta\omega\dot{x} + \omega^2 x = \ddot{u}(t) \tag{1a}$$

$$S_A(\omega_n, \zeta) = \max_t |\ddot{x}(t) + \ddot{u}_g(t)| \tag{1b}$$

The calculation of the response spectral acceleration of the damped SDOF harmonic oscillator over a range of periods and various damping levels provides the response acceleration spectrum (Figure 5), which describes the shaking of typical structures during an earthquake. Figure 5 is the output of Proschema software using the *Plot Pseudo Spectral Acceleration* option.
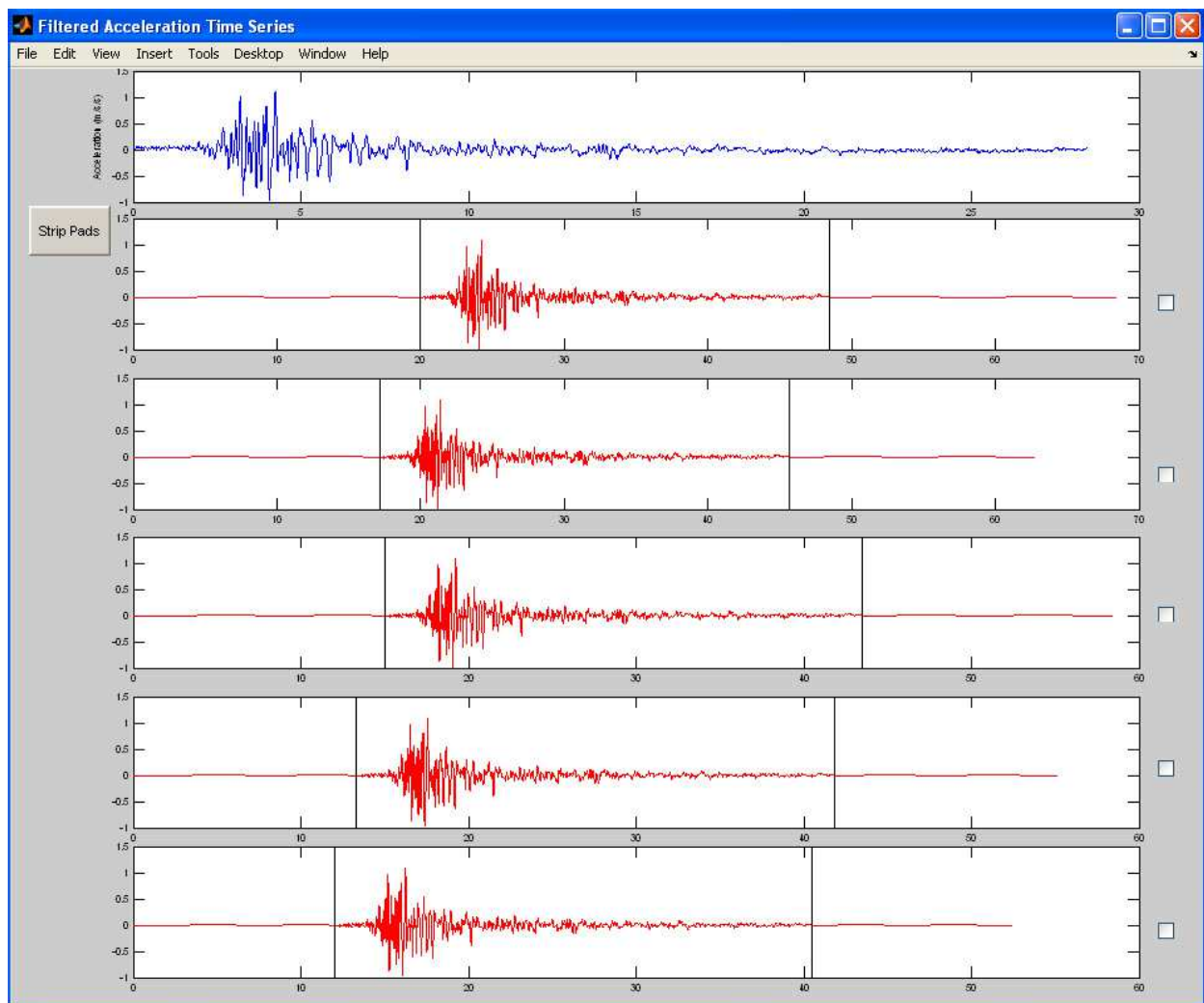

Fig. 3. Uncorrected (in blue) and filtered acceleration (in red) time series.
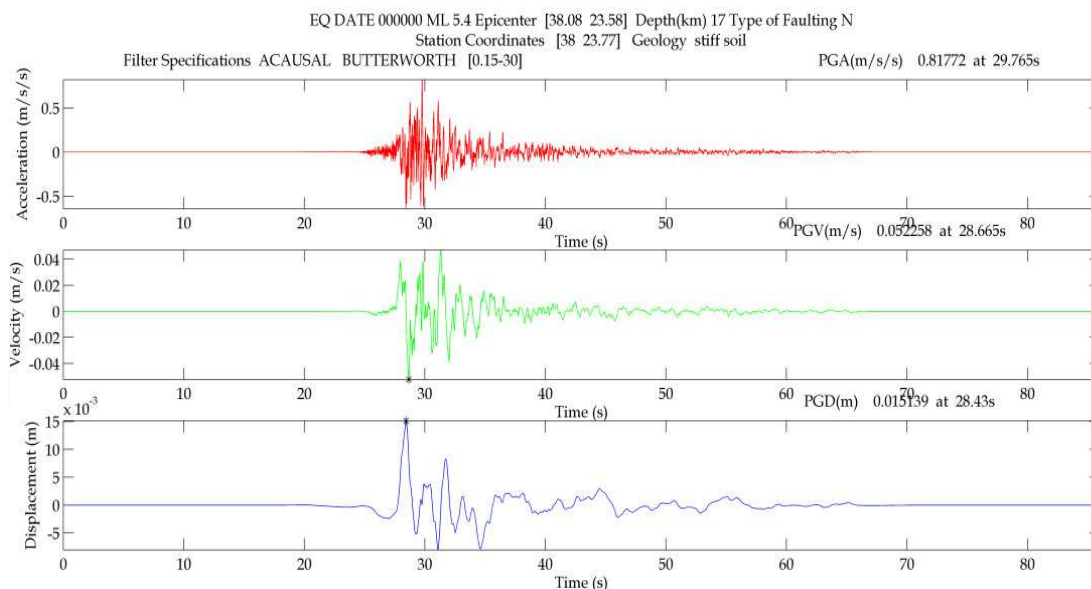
Fig. 4. Filtered acceleration, velocity and displacement time series.

Once the response acceleration spectrum is computed the calculation of more sophisticated earthquake engineering parameters, such as spectrum intensity (SI) follows. Spectrum intensity is defined as the integral of pseudo-velocity spectrum of 5% damping level between 0.1 s and 2.5 s. The critical issue behind the derivation of SI, and other parameters, is related with the common problem of calculation the area under the graph of an unknown function. Matlab makes possible the approximation of this unknown mathematical function, corresponding to the graph, using the boundary integral method (Liggett & Salmon, 1981) through a spline curve using chord-length parametrization and cubic spline interpolation. The above can be implemented through the combined use of *diff* function for calculating differences and approximate derivatives, at points ($SV_{(\omega,\zeta)}$, T) between 0.1 s and 2.5 s (Figure 6), the cumulative sum function *cumsum* and the cubic spline approximation function *csapi*. After the determination of the unknown function, through cubic spline approximation, it is straightforward to calculate the area of interest under response spectrum by evaluating the cubic spline function in the interval of interest. In Figure 6 the example illustrates the cubic spline approximation for computing the engineering parameter SI, using the response spectrum of a corrected strong motion record after the removal of high frequency and long period noise.

After computing so many parameters, either single value (1X1), such spectrum intensity (SI) and peak ground acceleration (PGA), or one dimensional arrays (1XN) like the acceleration time series (Amp) or even multi-dimensional arrays (NXM), the problem of minimizing storage requirements arises. To overcome this problem the desirable parameters e.g. PGA, can be assigned as fields of a structure array. Assigning fields in a structure array called e.g. Output Data Structure (OPD) can be achieved through Command Prompt lines e.g OPD. pga=[pga].

At this point the calculation of engineering parameters, such as PGA, corresponds to the observations (OBS) of the natural system, reaches to an end. In the next section the development of an empirical model, aiming to predict the anticipated peak ground acceleration (PGA) during a strong earthquake, is briefly described.
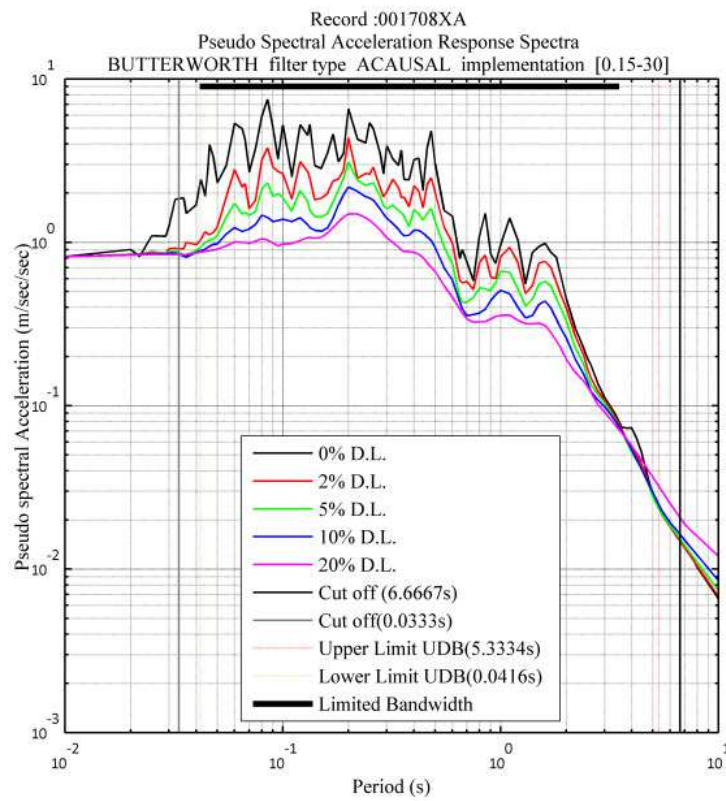
Fig. 5. Response acceleration spectrum for various damping levels (0%, 2%, 5%, 10% and 20% of the critical damping) over two hundred period estimators ranging between 0.01 s and 10.00 s.
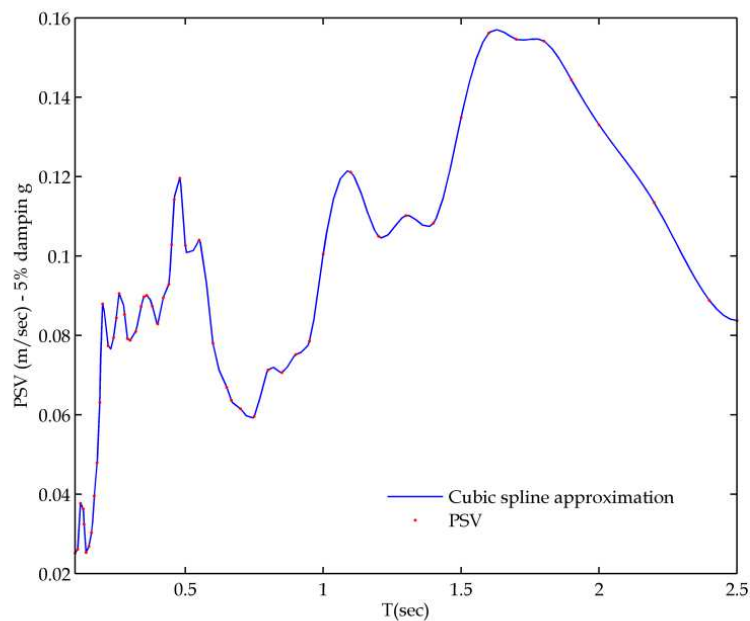

Fig. 6. Engineering parameter calculation after an important earthquake aided by cubic spline approximation.

3. Regression analysis versus stochastic optimization

The mathematical expression of an empirical model that is frequently used in ground motion modelling is given below:

$$\log_{10}(\text{PGA}) = a + bM + cM^2 + (d + eM)\log_{10}\sqrt{R^2 + (H - h)^2} + e + f \qquad (2)$$

In Equation (2) the seismic magnitude M, distance R (km) and depth H (km) are considered to be the independent variables of the model (Equation 2). Random variables were introduced in Equation (2) for modeling soil site conditions (e) and style of faulting (f). From this point on the scientific effort focuses in solving equation (2), corresponding to the determination of the coefficients [a, b, c, d, e, f, h].

The traditional method to determine the coefficients of Equation (2) corresponds to regression analysis whereas modern techniques of mathematical optimization are developed over the last decades. The contribution of optimization to geophysics has been interestingly growing the last decades due to its efficiently in modelling complex natural systems by determining the best solution from a set of alternative solutions (Goldberg, 1989). But which is the main advantage of optimization? The answer to this question would be its ability to reach the optimal solution for any system even under extreme computational environments, either when data-sets are limited but also when vast data-sets of high diversity require the determination of a solution describing sufficiently all the samples given. In the following section the advantages of optimization versus regression would be analysed and the best solver for optimization process would be selected through a test involving a number of important deterministic and stochastic algorithms.

3.1 Regression analysis

Using the *nonlinfit* function the implementation of mixed effects technique can be a first approach to modelling using regression analysis, leading to the determination of the coefficients of Equation (2). The results however revealed that this method was not successful in determining coefficients e and f, corresponding to the random effects terms of the model, due to the poor representation –in the database- of different types of soils and styles-of-faulting, respectively.

3.2 Optimization

In theory, traditional regression analysis -discussed in previous paragraph- is expected to provide one possible solution for any given equation. Nowadays optimization addresses the necessity for determining the best solution for data-sets of complex physical systems. As described previously the effort lies in determining the optimal solution for the mathematical model of Equation (2), which leads to the implementation of constrained optimization techniques. The mathematical problem corresponds to the minimization of misfit represented by the sum of squares of the residuals, between the logarithms of observed and predicted values (Equation 3). Constrained minimization problems have some basic pillars which are briefly given as: (1) the existence of a candidate theoretical solution, to initiate optimization (2) the existence of an objective function, to evaluate whether minimizing the misfit is achieved (3) a set of linear constraints serving as bounds for the coefficients' determination and (4) the determination of convergence criteria (Rothlauf, 2006).

$$OF = \sum_i (pred_i - obs_i)^2 \qquad\qquad (3)$$

It should be noted that for consistency in this example the same objective function, linear constraints and convergence criteria have been used during the implementation of the aforementioned solvers.

Techniques used during optimization to exhaust the search space are classified generally in three classes: (1) Calculus based techniques (2) Guided Random search techniques and (3) Enumerative techniques (Filho et al., 1994). In this paper calculus based versus guided random search techniques will be test through comparison of different solvers whereas enumerative algorithms will be discarded since "they cannot compete to the robustness race" when compared with the aforementioned techniques mainly due to the characteristics of their search domains (Said, 2005).

3.2.2 Optimization using calculus based techniques

Calculus based techniques are further divided in Direct and Indirect Search methods (Filho et al., 1994). Indirect Search methods as Non Linear Least Squares -in this example- is the most common approach in data fitting problems in earth sciences corresponding to the implementation of the maximum likelihood criterion (Draper & Smith, 1987) for determining the best solution setting the value of the objective function in Equation (3) to zero.

Pattern Search, on the other hand, is a Direct Search method used broadly in its generalized form in optimization of non-continuous and non-differentiable functions (Hookes & Jeeves; 1961; Dolan et al., 2003). An initial population of possible solutions serves as set of starting points. During optimization the available search space is either increasing or decreasing, depending on a gradient, in the effort to improve the solutions suggested previously (Audet & Dennis, 2003). The latter are then evaluated, for their effectiveness to minimize the misfit between predicted and observed values, using an objective function (Equation 3). Direct search methods are considered to be the simplest variation of deterministic algorithms used in optimization criticized for their efficacy to search sufficiently large solution spaces (Goldberg, 1989).

3.2.1 Optimization using guided random search techniques

Guided random search techniques are classified in Genetic Algorithms and Simulated Annealing (Filho et al., 1994). Both algorithms use information in order to guide their search for the optimal solution of the system. The development however of Genetic Algorithms is based on natural selection principles whereas Simulated Annealing relies on thermodynamic processes.

Fogel et al. (1966) developed Genetic Algorithms (GA), alternatively known as evolutionary programming, as "a technique in which candidate solutions to given tasks were represented as finite-state machines, which were evolved by randomly mutating their state-transition diagrams". Holland (1975) focused on how genetic operators observed in nature, such as survival-of-the-fittest, crossover and mutation could be introduced into evolutionary computing. During the last decades Genetic Algorithms applications has been described in the works of De Jong (1975), Grefenstette (1986), Goldberg (1989), Davis (1991), discussed in Mitchell (1996) thourougly, pointing out their strength in determining solutions for complex natural systems. The robustness of Genetic Algorithms in geophysics has been only recently

described by researchers (Stoffa & Sen, 1991; Tavakoli & Pezeshk, 2005), making the aforementioned solvers known for their application in constrained minimization problems (Goldberg, 1985). GA's are not limited by restrictive assumptions, concerning the continuity, the existence of derivatives and the unimodality of the function in terms of computational geometry. This is an advantage over regression analysis which often determines local minima as the solution of a given equation (Goldberg, 1989), while at the same time the suggested solution is highly dependent on a single point of initialization.

In the following paragraph we focus on the initialization, the process of improvement and the destination of constrained minimization by using stochastic solvers such as GAs. By keeping the analogy to biological systems a number of chromosomes/strings form the genetic prescription for the development and the operation of the organism. In our case the chromosomes/strings are composed by six genes/characters, representing the set of coefficients of Equation (2).

GAs start with a possible solution or a set of possible solutions corresponding to a theoretic attenuation curve and continues with optimization in order to determine the optimal solution for the given data set. In this study both initialization options have been tested, corresponding either to a single starting point (Simple Genetic Algorithm) or multiple random-generated starting points (Genetic Algorithm with initial Population Develoment), forming an initial population of candidate solutions each one satisfying the given linear constraints for the determination of the coefficients of Equation (2). In order to ensure well-dispersed and random initial population development, for the adequate representation of the search space, Latin Hypercube sampling was used (Diaz-Gomez & Hougen, 2006). The technique was elaborated by Iman et al. (1981) as stratified sampling without replacement, whereas in recent years risk analysis software employs Latin Hypercube sampling in preference of Monte Carlo approach for population development.

During optimization every candidate solution/string satisfying the given linear constraints is evaluated, through the objective function of Equation (3), for its effectiveness in minimizing the misfit, hence bringing the response value of the system near a desired value. Within a generation (group of solutions) fitness scaling serves the purpose of ranking each candidate solution to facilitate the selection of the best solutions that should surviving in the next generation. In that way, mimicking nature, the fitter solution survives and can be a parent individual to the next generation whereas worst fit solutions are penalized.

In the present study a number of 200 generations is considered, each one with population size of 600 individuals/solutions. Stochastic operators like the cross-over, mutation and survival-of-the-fittest guarantee diversity of the population (Pan, 1995) forcing the GA to search the solution space intensively, thereby reducing the possibility that the algorithm will return a local minimum (Goldberg, 1985). In terms of survival, 2 elite individuals/solutions are guaranteed to survive to the next generation in this study. A crossover fraction equal to 0.8 specifies the percentage of individual/solutions, other than elite children, produced by crossover in the next generation. Crossover mimics natural recombination between two parent chromosomes/solutions during which the offspring chromosome/solution has changed values of specific genes/numbers with respect to the parent genes/numbers. In this example two point cross-over has been implemented where the selected string is divided into 3 segments and then 2 segments are exchanged with the corresponding segments of another string. Mutation, on the other hand, alters a randomly selected character/coefficient within a string/solution to create a new possible solution. Adaptive mutation, used in this example, generates new directions in the search space, with respect to

the last successful or unsuccessful generation, bounded by the linear constraints set for the coefficients.

By combining the aforementioned stochastic operators, three versions of GAs, aim to test the relation between diversity of the population and performance, corresponding to

1. Simple Genetic Algorithm (SGA)
2. Genetic Algorithm with initial Population Development (GADP)
3. Hybrid Genetic Algorithm (HGA).

It is noted that Hybrid Genetic Algorithm (HGA) introduces the solution, determined a priori by a Simple Genetic Algorithm (SGA), to a deterministic solver, which requires the existence of derivatives, in order to provide local optima once the Genetic Algorithm has determined the neighborhood of the global optima (Il-Seok et al., 2004). The author included this enhanced evolutionary algorithm in the comparison to test the efficiency of the interaction between stochastic and deterministic solvers.

Simulated Annealing is a meta-heuristic algorithm proposed by Kirkpatrick, et al. (1983) and Cerny (1985) for the determination of global minima. It mimics the physical process where metals are slowly cooled so that eventually their crystal structure is frozen. The latter state corresponds to the determination of the optimal solution, using a minimum energy configuration during its implementation (Bertsimas and Tsitsiklis, 1993).

In this study optimization starts from a randomly generated initial population and a hypothesis for a parameter, known as Temperature, slowly decreasing from 100° C by a factor of 0.0059° C in the process of determining the optimal solution. During implementation each new possible solution is evaluated, for its effectiveness in minimizing the objective function (Equation 3), then in case of a lower misfit value the suggested solution is adapted. Simulated Annealing, an important solver in stochastic minimization problems (Bohachevsky et al., 1986), has been reported to successfully determine global minima however the author agree with the results of Ingber (1993) that this algorithm requires fine tuning to specific problems relative to other solvers.

4. Selecting the optimum solver

In order to test the efficiency of the optimization solvers a subset of the database was used, corresponding to rock site conditions, with the purpose of determining the coefficients of Equation (2) for peak ground acceleration and unspecified style-of-faulting. The solvers implemented for this test correspond to: (1) Simple Genetic Algorithm (SGA), (2) Genetic Algorithm with initial Population Development through Latin Hypercube sampling (GADP) (3) Hybrid Genetic Algorithm (HGA), (4) Simulated Annealing algorithm (SA) (5) Non Linear Least Squares (NLLSQ) (6) Non Linear Least Squares with initial population development (NLLSQDP) and (7) Pattern Search algorithm (PS). Although the author provided some basic principles of the solvers in the previous section, details and theoretical comparison between the solvers can be found in recent literature (Wetter and Wright, 2003; Gabere, 2007; Alander, 2009; El-Mihoub et al, 2006; Solomatine, 1998 among others).

Before evaluating the performance of these solvers it is meaningful to describe their relative computational efficiencies. The major difference between the two main categories of stochastic solvers (GAs, SA) is that GAs can either automatically produce a starting point (SGA) or they can be enhanced by initial population development (GADP), whereas Simulated Annealing requires a priori definition of initial conditions (Davis, 1987). The latter requirement applies for deterministic algorithms (NLLSQ, NLLSQDP, PS) as well. The

difference however between Simulated Annealing (SA) and deterministic solvers (NLLSQ, NLLSQDP, PS) is that the latter depend on the existence of derivatives in order to continue their iterations.

The evaluation of the performance of optimization solvers follows a qualitative and quantitative method. Analytically by qualitative criteria the authors refers to inability of the solver to determine a possible solution (1) within the given number of generations (2) satisfying the linear constraints and (3) in a timely manner. When the above criteria failed, optimization was implemented again with different starting points, which is acknowledged to be the main source of error that could lead to a solver's failure. The alternative solution, that of relaxing convergence criteria in the case of a solver's fail, was not considered since it would jeopardize the final comparison between different solvers. In the event of a solver's failure to produce a possible solution for second time, it was excluded from the quantitative comparison. In that sense Non Linear Least Squares (NLLSQ) solver failing the (2) criterion has been implemented again using this time multiple starting points (NLLSQDP). After the second failure to determine a feasible solution by returning the initial conditions -describing only the theoretical ground motion prediction equation, which was subjectively set by the programmer- Non Linear Least Squares (NLLSQ, NLLSQDP) have been excluded from the comparison from this point forward.

Table 1 presents the results of the quantitative comparison of the optimization solvers together with the coefficients of Equation (2) together with the numerical details used during optimization, such as the linear constraints introduced in the form of lower and upper bounds. It is noted that convergence criterion, alternatively known as tolerance, was set to 1E-06 for the purpose of this test. The quantitative comparison has been based in two criteria (1) the standard error, in logarithm base 10, and (2) the average sample log-likelihood (LLH) value (Scherbaum et al., 2009) of the resulting ground motion prediction equation as derived by a specific solver.

The standard error (σ_k) has been calculated as the mean of absolute residuals between observed and predicted by the model g_k (where k denotes the index of the solver) ground motion values described in the equation below:

$$\sigma_k = \frac{1}{N}\left|obs_i g_k\left(x_i\right)\right| \tag{4}$$

Assuming that the set of observations adequately describes nature, the likelihood $L(g_k|x)$, of the model g_k given the set of observations x, would represent how close model g_k describes reality. According to Scherbaum et al. (2009) the average sample log-likelihood (LLH) estimator has been calculated as the mean of log-likelihood values over N number of x samples

$$\left\langle\log\left(L\left(g_k|x\right)\right)\right\rangle = \frac{1}{N}\sum_{i=1}^{N}\log\left(g_k\left(x_i\right)\right) \tag{5}$$

It is noted that the sigma value is calculated as the standard deviation of the residuals in log 10 units, for consistency with Equation 2, using the *std* function. It is noted that the calculation of log-likelihood estimator of Equation 5 has been made through the function *normlike*.

	a	b	c	d	e	h	σ	LLH
Initial Points	1.00000	0.15000	0.00300	-0.50000	0.01000	0.01000		
Lower Bounds	1.00000	0.01000	0.00100	-1.50000	0.00010	0.00100		
Upper Bounds	3.00000	0.50000	1.00000	-0.00010	1.00000	2.00000		
NLLSQ	1.70166	0.35714	0.00100	-1.22962	0.00010	0.00100		
NLLSQDP	1.70166	0.35714	0.00100	-1.22962	0.00010	0.00100		
GA	1.82357	0.32641	0.00279	-1.27552	0.00755	0.00116	0.3464	1.8184
GADP	2.62122	0.11622	0.01568	-1.47212	0.03988	0.00136	0.3426	1.8044
HGA	1.70170	0.35713	0.00100	-1.22961	0.00010	0.00100		
PS	1.50000	0.01000	0.00100	-0.00010	0.00010	1.00100		
SA	1.73819	0.32360	0.00608	-1.25514	0.00264	0.55508	0.3484	1.8259

Table 1. Initial and boundary conditions for the seven solvers used in constrained minimization problem. Successful solvers are presented in bold.

The graphical comparison of the most successful solvers of Table 1 is shown in Figure 7. Matlab through the Optimization Toolbox provides core functions for solving minimization problems, using a number of suggested solvers such as the deterministic non-linear least squares (*lsqcurvefit* function) and Pattern Search (*patternsearch* function), the stochastic Genetic Algorithm (*ga* function) and Simulated Annealing (*simulannealbnd* function). Especially in the case of Genetic Algorithms the development of initial population of possible solutions is achieved through Latin Hypercube sampling using the *lhsdesign* function. The adjustment of stochastic operators, such as the survival-of–the fittest, mutation and cross-over, for Genetic Algorithms' implementation can be achieved through the *gaoptimset* function whereas for Pattern Search *psoptimset* function is required.

5. Conclusions

Two major conclusions can be drawn from the results of this study: firstly, deterministic algorithms, such as Non Linear Least Squares (NLLSQ, NLLSQDP) fail to determine the whole set of coefficients since the values of the coefficients c, e and h (Table 1) remain fixed to their lower boundary value. The above remark emphasizes the weakness of deterministic algorithms leading to the determination of local minima instead of returning a global solution for the minimization problem. Secondly, the effectiveness of GAs in solving minimization problems, even in their simpler parameterization (SGA), is supported by their ranking following the LLH criterion. Thus, the use of Genetic Algorithms aided by initial Population Development (GADP) by Latin Hypercube sampling for the constrained minimization problem set in Equation (2) is suggested.

Once the final set of coefficients is determined the seismologist can assess the expected ground motion at any given site, located at distance R from the epicenter of a strong earthquake with magnitude M. Figure 8 presents the estimated ground motion after a magnitude M5 and M6 earthquake at a rock site for various distances using stochastically derived ground motion prediction equation using Genetic Algorithm with initial Population Developement.
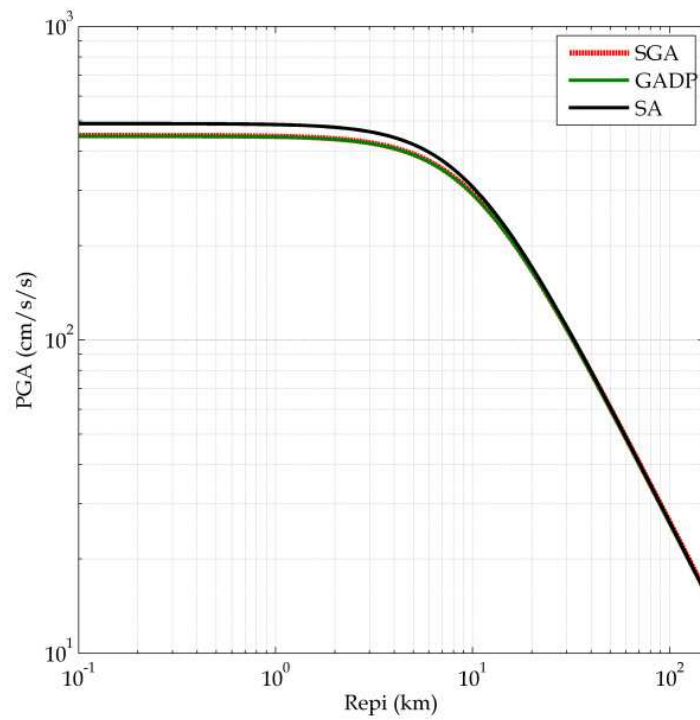
Fig. 7. Graphical representation of ground motion prediction equations (see Equation 2) as a result of constrained minimization using different solvers for the case of an M6 event at rock site for unspecified style of faulting.
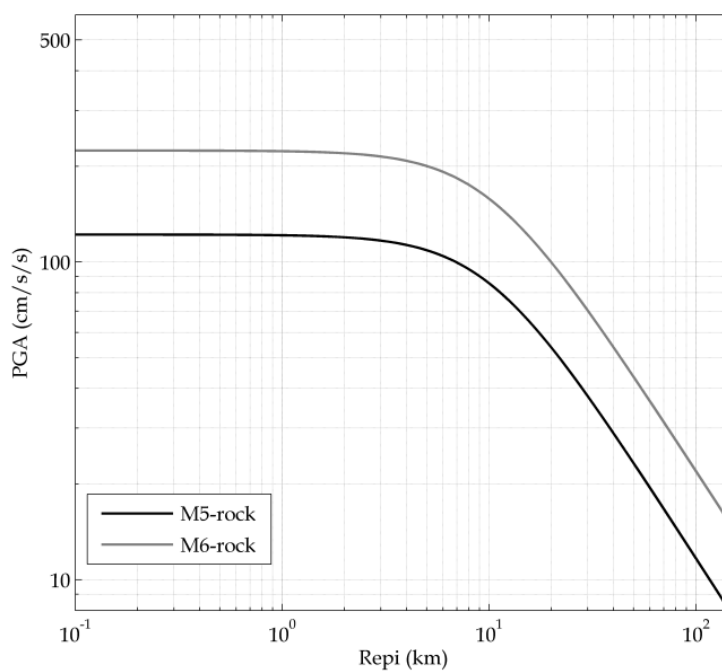

Fig. 8. Attenuation of peak ground acceleration (PGA) over a wide range of distances from the seismic source.

Ground motion prediction equations are an important tool for the scientific and engineering community to forecast anticipated ground motion and predict potential economical losses and structural damages. Another important application of ground motion prediction equations lies in developing possible scenarios for the planning short and long term emergency response.

This chapter describes how Matlab can be used in scientific research for Digital Signal Processing, Data Archiving but also for modeling complex natural systems through Optimization. Since the number of graduate students writing computer codes from scratch, in order to expand the frontiers of their research, continue to grow, core Matlab functions can be used to develop new software packages in the future. The application examples of this chapter clearly show that in seismological practice, demanding mathematical procedures can be implemented and their results can be easily visualized through Matlab.

6. References

Alander, J.T. (2009). *An Indexed Bibliography of Genetic Algorithms and Simulated Annealing: Hybrids and Comparisons*, Report for the Department of Electrical Engineering and Automation, Vaasa, Finland.

Audet, C. & Dennis, J.E. (2003) Analysis of generalized pattern searches. *SIAM Journal on Optimization*, Vol. 13, No. 3, pp. 889–903.

Bertsimas, D. & Tsitsiklis, J. (1993). Simulated Annealing, *Statistical Review*, Vol. 8, No. 1, pp. 10-15.

Bohachevsky, I.O., Johnson, M.E. & Stein, M. L. (1986). Generalized Simulated Annealing for Function Optimization, *Technometrics*, Vol. 28, No 3, pp. 209-217.

Cerny, V. (1982). A thermodynamical approach to the travelling salesman problem: An efficient simulation algorithm, *Journal of Optimization Theory and Applications*, Vol. 45, pp. 41-51.

Davis, L. (1991). *Handbook of Genetic Algorithms*, Van Nostrand Reinhold, New York.

Goldberg, D. E. (1989). *Genetic Algorithms in search, optimization and machine learning*, Addison Wesley Longman, Reading, Massachusetts.

De Jong, K. A. (1975). An analysis of the behavior of a class of genetic adaptive systems. Dissertation Abstracts International, Vol. 36, No. 10, 5140B.

Diaz-Gomez, P.A. & Hougen, D.F. (2006). Genetic Algorithms for hunting snakes in hypercubes: Fitness function and open questions, *Proceedings of the Seventh ACIS International Conference on Software Engineering, Aritificial Intelligence, Networking and Parallel/Distributed Computing*, Las Vegas, 2006.

Dolan, E.D., Lewis, R. M. & Torczon, V. (2003). On the local convergence of pattern search, *SIAM Journal on Optimization*, Vol. 14, No. 2, 567-583.

El-Mihoub, T.A., Hopgood, A.A.,Nolle L. & Battersby, A. (2006). Hybrid genetic algorithms: A review, *Engineering Letters*, Vol. 13, No. 2, pp. 124-137.

Filho, J.L., Treleaven, P.C. & Alippi, C. (1994). Genetic-Algorithm programming envirinments, *Computer*, Vol. 27, No. 6, pp. 28-43.

Fogel, L.J., Owens, A.J., & Walsh, M. J. (1966). *Artificial intelligence through simulated evolution,* John Wile, New York.

Gabere, M.N. (2007). *Simulated Annealing Driven Pattern Search.* Algorithms for Global Optimization, Msc Thesis, School of Computational and Applied Mathematics, University of the Witwatersrand, Witwatersrand.

Grefenstette, J.J. (1986). Optimization of control parameters for genetic algorithms, *IEEE Transactions SMC*, Vol. 16, pp. 122-128.

Holland, J. H. (1975). *Adaptation in natural and artificial Systems: An introductory analysis with applications to biology, control, and artificial intelligence*, A. Arbor (editor), University of Michigan Press, MIT press, Cambridge.

Hookes, R. & Jeeves, T.A. (1961). 'Direct search' solution of numerical and statistical problems, *Journal of the Association for Computing Machinery* (ACM), Vol. 8, No.2, pp. 212–229.

Il-Seok O., Jin-Seon L. & Byung-Ro M., Hybrid Genetic Algorithms for Feature Selection, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 26, No.11, pp. 1424-1437.

Iman, R. L., Helton, J. C., & Campbell, J. E. (1981). An approach to sensitivity analysis of computer models, Part 1. Introduction, input variable selection and preliminary variable assessment, *Journal of Quality Technology*, Vol. 31, No.3,pp. 174-183.

Ingber, L. (1983). Simulated annealing: Practice versus theory, *Journal of Mathematical Computational Modelling*, Vol. 18, No.11,pp. 29-57.

Karl, J. H. (1989). *An introduction to digital signal processing*. Academic Press Inc., San Diego, California.

Kirkpatrick, S., Gelatt, Jr. C.D, & Vecchi, M.P. (1983). Optimization by simulated annealing, *Science, Vol.* 220, No.4598,pp. 671-680.

Kramer, S. L., 1996. Geotechnical Earthquake Engineering. Prentice Hall International Inc., New Jersey, pp652.

Liggett, J.A. & Salmon, J.R. (1981). Cubic spline boundary elements, *International Journal for Numerical Methods in Engineering* , Vol.17,pp. 543-556.

Mitchell, M. (1996). *An Introduction to Genetic Algorithms*, MIT Press, Cambridge.

Rothlauf, F. (2006). *Representations for Genetic and Evolutionary Algorithms*, Springer Verlag, Berlin Heidelberg New York.

Pan, Z., Chen, Y., Kang, L. & Zhang, Y. (1995). Parameter estimation by Genetic Algorithms For nonlinear regression, *Proceedings of International Conference on Optimization Technique and Applications*, G.Z.Liu (editor).

Said, Y. H. (2005). Genetic Algorithms and their applications. *Handbook of Statistics 24*, Calyampudi Radhakrishna Rao, Edward J. Wegman, Jeffrey L. Solka (editors),pp. 359-390.

Scherbaum, F., Delavaud, E. & Riggelsen, C. (2009). Model selection in seismic hazard analysis: An information-theoretic perspective, *Bulletin of Seismological Society of America*, Vol.99,pp. 3234–3247.

Segou, M., Voulgaris, N., Makropoulos, K. C., & Stavrakakis, G. N. (2008). A review of the Greek strong motion database: needs, improvements and future development, in *Proceedings of the 31st general Assembly of the European Seismological Commission*, Hersonissos, Crete, Greece, 7-12 September 2008.

Segou, M. and Voulgaris, N. (2010). Proschema: A Matlab application for processing strong motion records and estimating earthquake engineering parameters, *Computers & Geosciences*, Vol.10,pp. 977-986.

Solomatine, D. P. (1998). Genetic and other global optimization algorithms - comparison and use in calibration problems, *Proceedings of the 3rd International Conference on Hydroinformatics*, Balkema Publishers, Copenhagen, Denmark.

Stoffa, P. L., & Sen, M. K. (1991). Nonlinear multiparameter optimization using genetic algorithms: Inversion of plane wave seismograms, *Geophysics*, Vol. 56, No.11,pp. 1794-1810.

Tavakoli, B., & Pezeshk, S. (2005). Empirical-stochastic ground-motion prediction for eastern North America, *Bulletin of Seismological Society of America,* Vol. 95,pp. 2283-2296.

Wetter, M. & Wright, J. (2003). Comparison of generalized pattern search and a genetic algorithm optimization method, *Proceedings of the Eighth International Conference on Building Simulation (IBPSA)*, Eindhoven, Netherlands.
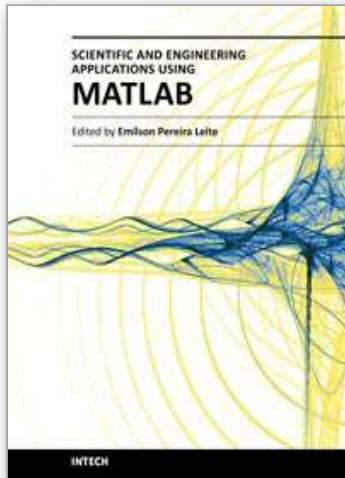
Scientific and Engineering Applications Using MATLAB

Edited by Prof. Emilson Pereira Leite

The purpose of this book is to present 10 scientific and engineering works whose numerical and graphical analysis were all constructed using the power of MATLAB® tools. The first five chapters of this book show applications in seismology, meteorology and natural environment. Chapters 6 and 7 focus on modeling and simulation of Water Distribution Networks. Simulation was also applied to study wide area protection for interconnected power grids (Chapter 8) and performance of conical antennas (Chapter 9). The last chapter deals with depth positioning of underwater robot vehicles. Therefore, this book is a collection of interesting examples of where this computational package can be applied.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

INTECH
open science | open minds