



In Cooperation with Tufts University

Implementation of the Next Generation Attenuation (NGA) Ground-Motion Prediction Equations in Fortran and R

By James Kaklamanos, David M. Boore, Eric M. Thompson, and Kenneth W. Campbell

U.S. Geological Survey Open-File Report 2010-1296

Version 1.0, November 1, 2010

U.S. Department of the Interior
U.S. Geological Survey

U.S. Department of the Interior
KEN SALAZAR, Secretary

U.S. Geological Survey
Marcia K. McNutt, Director

U.S. Geological Survey, Reston, Virginia 2010

For product and ordering information:
World Wide Web: <http://www.usgs.gov/pubprod>
Telephone: 1-888-ASK-USGS

For more information on the USGS—the Federal source for science about the Earth,
its natural and living resources, natural hazards, and the environment:
World Wide Web: <http://www.usgs.gov>
Telephone: 1-888-ASK-USGS

Suggested citation:
Kaklamanos, J., Boore, D.M., Thompson, E.M., and Campbell, K.W., 2010, Implementation of the Next Generation Attenuation (NGA) ground-motion prediction equations in Fortran and R: U. S. Geological Survey Open-File Report 2010–1296 (v 1.0, revised November 1, 2010).

Any use of trade, product, or firm names is for descriptive purposes only and does not imply endorsement by the U.S. Government.

Although this report is in the public domain, permission must be secured from the individual copyright owners to reproduce any copyrighted material contained within this report.

Contents

Introduction.....	1
Estimation of Input Parameters	2
Depth to Top of Rupture, Z_{TOR}	3
Fault Dip Angle	3
Down-Dip Rupture Width, W	4
Distance Measures, R_X and R_{RUP}	4
Depth Parameters, $Z_{1.0}$ and $Z_{2.5}$	6
Implementation in R.....	6
Installation Instructions.....	7
UNIX Variants.....	8
Microsoft Windows	8
Loading Package nga.....	8
Specific Functions in R.....	9
Function for Obtaining Ground-Motion Predictions from all NGA Models.....	11
Functions for Obtaining Ground-Motion Predictions from Individual NGA Models	16
Functions for Estimating Unknown Input Parameters.....	18
Distance Functions $R_{rup}.calc$ and $R_x.calc$	18
Estimation of Fault Dip using the Function $dip.calc$	18
Estimation of Down-Dip Rupture Width using the Function $W.calc$	18
Estimation of Depth to Top of Rupture using the Function $Z_{tor}.calc$	18
Estimation of Hypocentral Depth using the Function $Z_{hyp}.calc$	19
Estimation of Depth Parameters using the Functions $Z1.calc.as$ and $Z1.calc.cy$	19
Overview of Usage in R	19
Overview of Input/Output in R.....	21
Example Sessions	22
Example 1	23
Example 2	25
Example 3	27
Example 4	28
Implementation in Fortran.....	30
Program Acquisition and Use.....	30
Inputs and Outputs.....	33
Interpolation of Spectral Values for Periods not Corresponding to the Tabulated Coefficients	37
Examples of Input and Output.....	38
Programs to Compute Fault-to-Station Distances.....	41
Selecting Between the R and Fortran Implementations.....	41
Acknowledgments	42
References Cited.....	42

Figures

Figure 1. Illustration of earthquake source and distance measures using a vertical cross-section through a fault rupture plane (from Kaklamanos and others, 2011).....	4
---	---

Figure 2.	Plan view of a fault rupture, giving the definition and sign convention of the source-to-site azimuth (from Kaklamanos and others, 2011).	5
Figure 3.	Comparison of NGA ground-motion predictions for KBflatfile record no. 824 (2010 Baja earthquake recorded in Winterhaven, California).	25
Figure 4.	Comparison of NGA ground-motion predictions: PGA versus distance for the 2008 Chino Hills, California, earthquake.	27
Figure 5.	Example interpolation of spectral values for periods not corresponding to the tabulated coefficients; comparison of median estimates of S_a .	37
Figure 6.	Example interpolation of spectral values for periods not corresponding to the tabulated coefficients; comparison of aleatory uncertainties.	38
Figure 7.	Median ground-motion estimates for the hanging wall case.	39
Figure 8.	Total standard deviations for the hanging wall case.	39
Figure 9.	Median ground-motion estimates for the footwall case.	40
Figure 10.	Total standard deviations for the footwall case.	40

Tables

Table 1.	Input parameters of the Next Generation Attenuation models.	2
Table 2.	Estimation of dip angle from style of faulting.	4
Table 3.	Hierarchy of functions for the nga R package.	9
Table 4.	Arguments for R function $S_a.nga$.	12
Table 5.	Output for R function $S_a.nga$.	14
Table 6.	Additional arguments for R functions $S_a.as$, $S_a.ba$, $S_a.cb$, and $S_a.cy$.	17
Table 7.	Contents of file <code>nga_gm_tmr_zips.zip</code> .	30
Table 8.	Inputs for Fortran control file.	33
Table 9.	Terms in Fortran output file.	35

Implementation of the Next Generation Attenuation (NGA) Ground-Motion Prediction Equations in Fortran and R

By James Kaklamanos¹, David M. Boore², Eric M. Thompson¹, and Kenneth W. Campbell³

Introduction

This report presents two methods for implementing the earthquake ground-motion prediction equations released in 2008 as part of the Next Generation Attenuation of Ground Motions (NGA-West, or NGA) project coordinated by the Pacific Earthquake Engineering Research Center (PEER). These models were developed for predicting ground-motion parameters for shallow crustal earthquakes in active tectonic regions (such as California). Of the five ground-motion prediction equations (GMPEs) developed during the NGA project, four models are implemented: the GMPEs of Abrahamson and Silva (2008), Boore and Atkinson (2008), Campbell and Bozorgnia (2008), and Chiou and Youngs (2008a); these models are abbreviated as AS08, BA08, CB08, and CY08, respectively. Since site response is widely recognized as an important influence of ground motions, engineering applications typically require that such effects be modeled. The model of Idriss (2008) is not implemented in our programs because it does not explicitly include site response, whereas the other four models include site response and use the same variable to describe the site condition (V_{S30}). We do not intend to discourage the use of the Idriss (2008) model, but we have chosen to implement the other four NGA models in our programs for those users who require ground-motion estimates for various site conditions. We have implemented the NGA models by using two separate programming languages: Fortran and R (R Development Core Team, 2010). Fortran, a compiled programming language, has been used in the scientific community for decades. R is an object-oriented language and environment for statistical computing that is gaining popularity in the statistical and scientific community. Derived from the S language and environment developed at Bell Laboratories, R is an open-source language that is freely available at <http://www.r-project.org/> (last accessed 11 January 2011). In R, the functions for computing the NGA equations can be loaded as an add-on user-contributed code, which is referred to as a “package” in R. The details of the `nga` package (Kaklamanos and Thompson, 2010) are presented in this report. In addition, differences between the R and Fortran implementations will be discussed later in this report.

The NGA models have established a new baseline for seismic hazard assessments, and they have been incorporated into the most recent national seismic hazard maps published by the U.S. Geological Survey (Petersen and others, 2008). However, many of the new models are significantly more complicated than previous GMPEs and, therefore, require a substantial investment of time to implement and validate. We hope that the scientific and engineering communities find our implementations to be useful in research and practice. Our implementations may be considered as an alternate to the Microsoft

¹Department of Civil and Environmental Engineering, Tufts University, Medford, Massachusetts 02155
james.kaklamanos@tufts.edu eric.thompson@tufts.edu

²Earthquake Science Center, U.S. Geological Survey, 345 Middlefield Road, Menlo Park, California 94025 boore@usgs.gov

³EQECAT, Inc., 1030 NW 161st Place, Beaverton, Oregon 97006 KCampbell@eqecat.com

Excel spreadsheet implementation available on the PEER NGA project Web site (<http://peer.berkeley.edu/ngawest/index.html>, last accessed 11 January 2011). The implementations in Fortran and R are more appropriate for performing batch calculations than the implementation in Microsoft Excel. Spreadsheets and Fortran code for some of the individual models also are available on the PEER NGA project Web site; our programs implement the four GMPEs simultaneously. Our programs give the same results as the programs on the PEER NGA Web site, but we offer some additional flexibility of input, additional methods of estimating unknown input parameters, and additional options for output.

Although these programs have been used by the U.S. Geological Survey (USGS), Tufts University, and others, no warranty, expressed or implied, is made by Tufts or the USGS as to the accuracy or functioning of the programs and related material, nor shall the fact of distribution constitute any such warranty, and no responsibility is assumed by Tufts or the USGS in connection therewith.

Estimation of Input Parameters

A summary of the input parameters used in the GMPEs is presented in table 1 (modified from Kaklamanos and Baise, 2011).

Table 1. Input parameters of the Next Generation Attenuation models.

Parameter	Model			
	AS08	BA08	CB08	CY08
<i>Source parameters:</i>				
Moment magnitude, M	•	•	•	•
Depth to top of rupture, Z_{TOR}	•		•	•
Down-dip rupture width, W	•			
Fault dip, δ	•		•	•
Style-of-faulting flag (function of rake angle, λ)	•	•	•	•
Aftershock flag	•			•
<i>Path parameters:</i>				
Closest distance to the rupture plane (rupture distance), R_{RUP}	•		•	•
Horizontal distance to the surface projection of the rupture (Joyner-Boore distance), R_{JB}	•	•	•	•
Horizontal distance to top edge of rupture measured perpendicular to the strike (site coordinate), R_X	•			•
Hanging wall flag	•			•
<i>Site parameters:</i>				
Time-averaged shear wave velocity over the top 30 meters of the subsurface, V_{S30}	•	•	•	•
Depth to $V_S = 1.0$ km/s ($Z_{1.0}$)	•			•
Depth to $V_S = 2.5$ km/s ($Z_{2.5}$)			•	
PGA (or Sa) on rock, as baseline for nonlinear site response	•	•	•	•

When employing the GMPEs, users will likely face situations in which some of the required input parameters are unknown. First, we describe the case in which users wish to implement the NGA models on historical earthquake records in the NGA flatfile (the master database used to develop the NGA models, available on the PEER NGA Web site). In the flatfile, there is no column for the site coordinate (R_X), which is needed for the AS08 and CY08 models. In addition, for events in the NGA flatfile without finite fault models, the columns are unpopulated for several of the source and distance parameters necessary for the AS08, CB08, and CY08 models. Although estimates of V_{S30} are included for nearly every station in the flatfile, the depth parameters $Z_{1.0}$ (used in AS08 and CY08) and $Z_{2.5}$ (used in CB08) are not present for many records.

Alternatively, in the frequent case that users wish to implement the NGA models for hypothetical earthquake scenarios in seismic hazard analyses, even fewer input parameters may be available. Our implementations of the NGA models follow a framework for estimating unknown input parameters that is described in detail by Kaklamanos and others (2011). In this report, we outline the framework used in the implementations, but we refer the reader to Kaklamanos and others (2011) for the derivations of the distance equations, further justifications of the framework, and discussion of alternatives.

Our programs require the user to input a bare minimum of

- Moment magnitude, M ,
- Joyner-Boore distance, R_{JB} ,
- Time-averaged shear wave velocity over a subsurface depth of 30 meters, V_{S30} ,
- Style of faulting (*e.g.*, rake angle, λ), and
- Hanging wall flag, F_{HW} .

If the other input parameters are unknown, they are estimated by the programs using the following procedures.

Depth to Top of Rupture, Z_{TOR}

If the depth to top of rupture (Z_{TOR}) is not specified, then the depth is estimated by using the hypocentral depth (Z_{HYP}), down-dip rupture width (W), and dip (δ), assuming that the hypocenter is located 60 percent down the fault width, as suggested by Mai and others (2005). In the case that Z_{HYP} is unknown, we employ the linear relationship between Z_{HYP} and M published in table 1 of Scherbaum and others (2004):

$$Z_{HYP} = \begin{cases} 5.63 + 0.68 M & \text{for strike - slip faults} \\ 11.24 - 0.2 M & \text{for non - strike - slip faults.} \end{cases} \quad (1)$$

Strike-slip faults are taken to be events with rake angles within 30° of horizontal. Once Z_{HYP} is estimated using equation 1, Z_{TOR} can then be estimated using the method of Mai and others (2005).

Fault Dip Angle

When the dip angle (δ) is not specified, it is estimated from the style of faulting (using the rake angle) from the guidelines explained in Kaklamanos and others (2011). These recommendations (table 2) are a modification of the guidelines Chiou and Youngs (2008b) used in developing their NGA model.

Table 2. Estimation of dip angle from style of faulting.

Style of faulting	Dip angle (deg)
Strike-slip	90
Reverse	40
Normal	50

Down-Dip Rupture Width, W

When left unspecified, the down-dip rupture width (W) is estimated from the earthquake magnitude and style of faulting by using the logarithmic relationships in table 2A by Wells and Coppersmith (1994):

$$W = \begin{cases} 10^{-0.76 + 0.27M} & \text{for strike - slip events} \\ 10^{-1.61 + 0.41M} & \text{for reverse events} \\ 10^{-1.14 + 0.35M} & \text{for normal events.} \end{cases} \quad (2)$$

Distance Measures, R_X and R_{RUP}

The vertical cross-section in figure 1 illustrates the three distance measures (R_{JB} , R_{RUP} , and R_X) and the three rupture parameters (Z_{TOR} , δ , and W) for a hypothetical site. Users must enter a value for the Joyner-Boore distance (R_{JB}), because it is required for each of the four models, and it is easily visualized in two dimensions as the horizontal distance to the surface projection of the rupture. If the rupture distance (R_{RUP}) and/or site coordinate (R_X) are unspecified, they are calculated using the geometric equations derived by Kaklamanos and others (2011). Given R_{JB} , the other two distance parameters (R_{RUP} and R_X) are constrained by the value of R_{JB} , the source-to-site azimuth (α), and the geometry of the rupture plane (W , Z_{TOR} , and δ , which may be estimated using the default methods in this report if the values are unknown). First, the code calculates R_X from R_{JB} , α , W , Z_{TOR} , and δ , and then R_{RUP} is calculated from R_X and the other parameters.

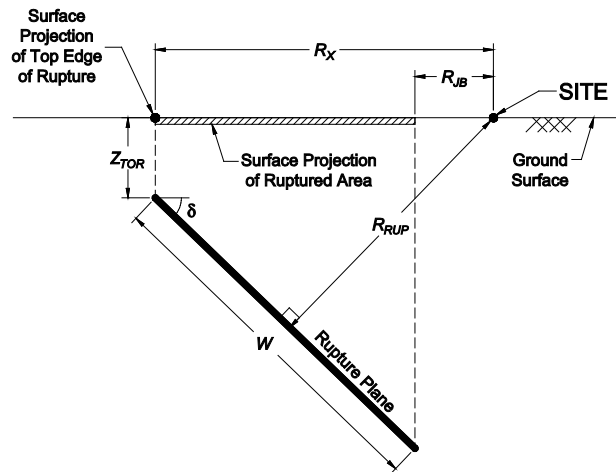


Figure 1. Illustration of earthquake source and distance measures using a vertical cross-section through a fault rupture plane (from Kaklamanos and others, 2011).

The main challenge with this method is determining the source-to-site azimuth, which may not be known. The definition of the source-to-site azimuth is shown in figure 2, which is a plan view of the surface projection of the ruptured area. Five example sites are shown, along with their source-to-site azimuths and site coordinates. Sites 1, 2, and 4 are on the hanging wall side of the fault and have positive azimuths; sites 3 and 5 are on the footwall side of the fault and have negative azimuths. If α is unspecified, we assign α to be 50° for sites on the hanging wall side of the fault ($F_{HW} = 1$) and -50° for sites on the footwall side of the fault ($F_{HW} = 0$). These numbers are approximations based upon the average values in the NGA flatfile: 48.5° for hanging wall sites and -53.1° for footwall sites. The distance calculations are symmetric for sites reflected parallel to the strike about the ruptured area (mathematically, any site with an azimuth equal to $\text{sgn}(\alpha) \cdot (180^\circ - \alpha)$ will have the same distance measures as a site with an azimuth of α). For example, a pair of sites with azimuths of 35° and 145° will have identical distance measures, as will a pair of sites with azimuths of -50° and -130° . In computing the average azimuths for use, azimuths with absolute values greater than 90° were reflected so that $0 \leq |\alpha| \leq 90^\circ$. The reality is that the numerical value of the azimuth for footwall sites does not matter; as long as α is negative, the value of α does not influence the calculated value of R_{RUP} . The calculated value of R_X is affected, but it does not enter the ground-motion calculations because the hanging wall terms (which include R_X) are zero.

The one situation in which R_X and R_{RUP} are not constrained by R_{JB} and α is in the special case of $R_{JB} = 0$, when the site is located directly above the ruptured area. In this situation, either R_X or R_{RUP} must be specified in addition to $R_{JB} = 0$ in order to calculate the third distance parameter. If neither R_X nor R_{RUP} can be assumed in this situation, then we assume that the site is located in the middle of the surface projection of the ruptured area.

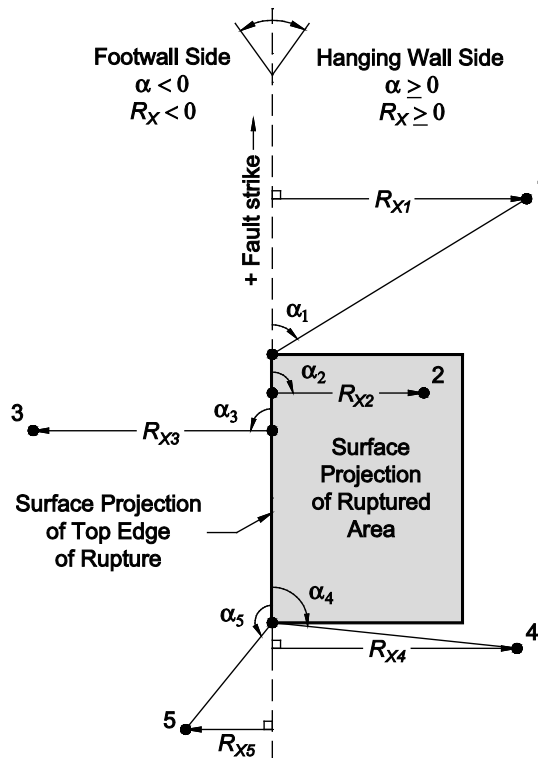


Figure 2. Plan view of a fault rupture, giving the definition and sign convention of the source-to-site azimuth (from Kakkamanos and others, 2011).

Depth Parameters, $Z_{1.0}$ and $Z_{2.5}$

The depth to the $V_S = 1.0$ km/s horizon ($Z_{1.0}$) is a required input parameter for the AS08 and CY08 models, and the depth to the $V_S = 2.5$ km/s horizon ($Z_{2.5}$) is required for the CB08 model. To estimate $Z_{1.0}$ when applying the AS08 and CY08 models, we use the recommendations of the respective model developers. Abrahamson and Silva (2008) recommend using the following median relationship to estimate $Z_{1.0}$ (m) from V_{S30} (m/s):

$$Z_{1.0} = \begin{cases} \exp(6.745) & \text{for } V_{S30} < 180 \text{ m/s} \\ \exp\left[6.745 - 1.35 \cdot \ln\left(\frac{V_{S30}}{180}\right)\right] & \text{for } 180 \leq V_{S30} \leq 500 \text{ m/s} \\ \exp\left[5.394 - 4.48 \cdot \ln\left(\frac{V_{S30}}{500}\right)\right] & \text{for } V_{S30} > 500 \text{ m/s} . \end{cases} \quad (3)$$

Chiou and Youngs (2008a) recommend using the following median relationship to estimate $Z_{1.0}$ from V_{S30} :

$$Z_{1.0} = \exp\left[28.5 - \frac{3.82}{8} \cdot \ln(V_{S30}^8 + 378.7^8)\right] . \quad (4)$$

The estimated values of $Z_{1.0}$ are different for equations 3 and 4 because Abrahamson and Silva (2008) developed equation 3 from analytical site response models, and Chiou and Youngs (2008b) developed equation 4 using an updated velocity model for southern California, which has smaller depth parameters than the previous velocity model reflected in the NGA flatfile. The AS08 model was derived so that equation 3 for $Z_{1.0}$ should be used when $Z_{1.0}$ is unknown, and the CY08 model was derived so that equation 4 for $Z_{1.0}$ should be used. Thus, although this requires having two separate estimates of $Z_{1.0}$ for a single site, equation 3 should be used when estimating $Z_{1.0}$ for the AS08 model, and equation 4 should be used when estimating $Z_{1.0}$ for the CY08 model.

In order to estimate $Z_{2.5}$, Campbell and Bozorgnia (2007) offer guidelines for extrapolating the estimates of $Z_{1.0}$ or $Z_{1.5}$ (depth to $V_S = 1.5$ km/s) if these values are available. If $Z_{1.5}$ is known, then

$$Z_{2.5} = 636 + 1.549Z_{1.5} . \quad (5)$$

If $Z_{1.0}$ is known (but $Z_{1.5}$ is unknown), then

$$Z_{2.5} = 519 + 3.595Z_{1.0} , \quad (6)$$

where all depths are in meters. When $Z_{1.0}$ is unknown, the AS08 equation for $Z_{1.0}$ (equation 3) may be used to estimate $Z_{1.0}$ from V_{S30} , and $Z_{2.5}$ may then be obtained from equation 6. Equation 3 (AS08) is used instead of equation 4 (CY08) because Chiou and Youngs (2008b) used an updated velocity model for southern California when they developed equation 4; the CB08 model is based on an earlier version.

Implementation in R

In this section, we outline the basic commands that are necessary to employ the `nga` package within the statistical language and environment R. In order to use the `nga` package, it is not necessary to invest a great deal of time to become an expert in the language. All that is necessary is a basic understanding of some simple commands and an awareness of the extensive help documentation

available within R. Users who have experience with the commercial software Matlab generally find that the syntax of R is easy to pick up due to the similarities. One of the benefits of R over Matlab is that R is free open-source software. For an additional explanation on the benefits of R, the reader is referred to Thompson (2007).

R is available for download at <http://www.r-project.org/> (last accessed 11 January 2011). In this report, we provide some basic examples on using R (within the context of the `nga` package), but a large amount of general documentation is available on the R-project Web site, including details on how to install and get started with R for the most common operating systems. “An Introduction to R” is an excellent resource for new users in becoming familiar with the syntax of the R language, and “R Data Import/Export” is especially helpful for learning how to use R to read and write data. The concise reference cards in the “Contributed Documentation” are helpful for finding specific functions. The “R-help” mailing list, which is archived online in a number of locations, is helpful for finding answers to specific questions that have already been asked, or for posing new questions to the list (for best results, be sure to read the posting guide prior to submitting your question). The “R Reference Index” is a more extensive resource, but its length limits its usefulness for a beginner. We find that most answers to specific questions in R are available somewhere online; the “R-help” mailing list, as well as simple Internet searches using any engine, will often yield fruitful results. Efficient usage of R need not entail the memorization of every function you plan to use, but rather knowing how to find help documentation on the functions you need. The quality and extent of the R documentation is unsurpassed by many other programs.

Installation Instructions

Once the R software has been installed on your system, the next step is to install the `nga` package. This package can be installed by using the package manager within R, or by typing

```
> install.packages("nga")
```

into the R command prompt. Commands intended to be entered at the R command line are indicated by lines that begin with the greater-than symbol. All functions in R have parentheses, and the function arguments, if any, are placed inside the parentheses; here, the argument is a package name, which is placed in quotes because it is a character.

Alternately, our package may be installed by downloading the appropriate binary file (Windows or MacOS X) from <http://cran.r-project.org/web/packages/nga/index.html>, which also can be accessed through the Web page associated with this report (<http://pubs.usgs.gov/of/2010/1296/>). To install the package locally using the binary file, the relevant files are `nga_<version>.tgz` for Unix variants (including Mac OSX), and `nga_<version>.zip` for Windows operating systems. Download the appropriate file into a directory on your computer (do not decompress the file), and follow the relevant instructions below. Note that an additional file, `nga_<version>.tar.gz`, is the package-source file, which can be viewed or compiled on any operating system. In order for Windows users to unzip this file, we recommend the program 7Zip, which is available at <http://www.7-zip.org/> (last accessed 11 January 2011). An important use of the `.tar.gz` file is to view the source code for the ground-motion calculations; once the file is unzipped, navigate to the R subfolder and open the respective source-code files. The source code contains useful insight into how the ground-motion calculations are performed. Comments in the R language are specified with the number sign (#).

UNIX Variants

To install the package on a Unix-based system at the R command prompt, type

```
> install.packages("DIR/nga_<version>.tar.gz", repos = NULL),
```

where DIR indicates the path to the directory where the .tar.gz file is located. To install a package on Mac OSX, select the pull-down menu Packages & Data from within the Mac R GUI, and then select Package Installer. Next, select the Local Source Package option in the Packages Repository panel, then click on the Install button in the Install Location panel. You will then be prompted to choose the file you wish to install.

Microsoft Windows

To install the `nga` package on a Microsoft Windows system, select the pull down menu Packages from within the R GUI, and then select “Install package(s) from local .zip files.” You will then be prompted to locate the .zip file on your computer.

Loading Package `nga`

Once the package is installed, the commands will be the same for all operating systems. To access the scripts, the package must be loaded with the command

```
> library(nga).
```

The `nga` package only needs to be installed once on a system, but it must be loaded for each new R session. To familiarize yourself with R and the `nga` package, a good place to start is by typing

```
> help.start(),
```

which opens the extensive HTML documentation for R. If you navigate to the Packages hyperlink, you will see a list of the packages currently installed on your system. (Several packages are installed automatically when you download R.) By opening the `nga` link, you will see a list of most of the functions that are available within the `nga` package. Once the package is loaded, another shortcut is to type

```
> ?nga ,
```

or

```
> help(nga)
```

to open the help documentation. The standard R help documentation for a function contains a short description of the function, its usage, the definition for each argument, a description of the value returned by the function, and a series of examples for using the function. To learn more about a particular function, you can copy and paste the example codes into the R command line and see the output.

To search the help documentation for a particular topic, type

```
> help.search("foo"),
```

where `foo` is the name of the topic for which you are searching. (It is necessary to enclose your topic in quotes.) Further details on usage in R will be presented in a later section.

Specific Functions in R

The `nga` package is composed of sets of functions that are used to perform specific calculations. In this section, we provide details about the commonly-used functions we have written for the `nga` package. Table 3 is an extensive listing of the functions contained within the `nga` package, with the most frequently used functions listed first. The table is divided into external and internal functions. Here, we use the term “external” to indicate those functions that are typically called by the user, while we use the term “internal” to refer to those functions that are not commonly called directly by the user; instead, these functions are called by the more comprehensive external functions in performing calculations. However, for completeness, all functions associated with the `nga` R package are listed in the table.

Table 3. Hierarchy of functions for the `nga` R package.

General purpose ¹	Names of functions	Description and notes
External functions		
Ground-motion predictions for all NGA models (<code>nga</code>)	<code>Sa.nga</code>	Employ the four GMPEs and output the results in the format of a data frame. The function <code>Sa.nga</code> is designed to mimic the output from the Fortran output files.
Ground-motion predictions for individual NGA models (<code>Sa</code>)	<code>Sa.as</code> , <code>Sa.ba</code> , <code>Sa.cb</code> , <code>Sa.cy</code>	Employ each individual NGA GMPE. The output is a single number (<code>Sa</code> , <code>PGA</code> , <code>PGV</code> , or <code>PGD</code>), which is more streamlined (but less transparent) than the output from <code>Sa.nga</code> . Each of these respective functions is called by <code>Sa.nga</code> when performing ground-motion calculations.
Estimation of unspecified input parameters	<code>dip.calc</code> , <code>Rrup.calc</code> , <code>Rx.calc</code> , <code>W.calc</code> , <code>Z1.calc.as</code> , <code>Z1.calc.cy</code> , <code>Z2.5.calc</code> , <code>Zhyp.calc</code> , <code>Ztor.calc</code>	Estimation of fault dip, source-to-site distance measures, down-dip rupture width, depth parameters, hypocentral depth, and depth to top of rupture, respectively.
Spectral periods for NGA models and interpolation	<code>getPeriod</code> , <code>modelPeriods</code> , <code>interpolate</code>	Functions that concern the available spectral periods for each model and how interpolation is performed.
Example ground-motion calculations	<code>KBflatfile</code>	This is a dataset (<i>not</i> a function) that is used for examples of data input and output and for performing calculations with the GMPEs.
Internal functions		
Trigonometric functions (<code>trig</code>)	<code>csc</code> , <code>sec</code> , <code>cot</code>	Reciprocal trigonometric functions: cosecant (<code>csc</code>), secant (<code>sec</code>), and cotangent (<code>cot</code>); used in the distance calculations.
Spectral periods for each NGA model (<code>periods</code>)	<code>periods.as</code> , <code>periods.ba</code> , <code>periods.cb</code> , <code>periods.cy</code>	Functions that return a vector of spectral periods at which the period-dependent model coefficients are defined for the different NGA models. Called by <code>modelPeriods</code> .
Subprocedures for the AS08 model (<code>subs.as</code>)	<code>f1.as</code> , <code>f4.as²</code> , <code>f5.as</code> , <code>f6.as</code> , <code>f8.as</code> , <code>f10.as</code> , <code>PGA1100.as</code> , <code>Alpha.as</code> , <code>Sigma0.as</code> , <code>Tau0.as</code> , <code>Sigma.as</code> , <code>Tau.as</code> , <code>SigmaTot.as</code> , <code>SaMedian.as</code>	Functions needed to estimate ground motions using the AS08 model. These functions are called by the main function used for calculations, <code>Sa.as</code> .

General purpose ¹	Names of functions	Description and notes
Subprocedures for the BA08 model (subs.ba)	Fd.ba, Fm.ba, Fs.ba, SaMedian.ba	Functions needed to estimate ground motions using the BA08 model. These functions are called by the main function used for calculations, Sa.ba.
Subprocedures for the CB08 model (subs.cb)	Fmag.cb, Fdist.cb, Fflt.cb, Fhng.cb, Fsite.cb, Fsed.cb, A1100.cb, Alpha.cb, Sigma.cb, Tau.cb, SigmaTot.cb, SigmaArb.cb, SaMedian.cb	Functions needed to estimate ground motions using the CB08 model. These functions are called by the main function used for calculations, Sa.cb. NOTE: The CB08 model is designed with Z2.5 in units of km, and these internal functions require that Z2.5 be input in km. (On the other hand, the more comprehensive Sa.cb and Sa.nga functions require that Z2.5 be input in m, to be consistent with the units of the other depth parameters.)
Subprocedures for the CY08 model (subs.cy)	LNyref.cy, NL.cy, Sigma.cy, Tau.cy, SigmaTot.cy, SaMedian.cy	Functions needed to estimate ground motions using the CY08 model. These functions are called by the main function used for calculations, Sa.cy.
Period-dependent coefficients for the AS08 model (coefs.t.as)	Vlin.as, b.as, a1.as, a2.as, a3.as, a4.as, a5.as, a8.as, a10.as, a12.as, a13.as, a14.as, a15.as, a16.as, a18.as, s1.est.as, s2.est.as, s1.meas.as, s2.meas.as, s3.as, s4.as, rho.as	Functions for computing the period-dependent coefficients for the AS08 model. Called by the main function used for calculations, Sa.as.
Period-dependent coefficients for the BA08 model (coefs.t.ba)	blin.ba, b1.ba, b2.ba, c1.ba, c2.ba, c3.ba, h.ba, e1.ba, e2.ba, e3.ba, e4.ba, e5.ba, e6.ba, e7.ba, Mh.ba, Sigma.ba, TauU.ba, TauM.ba, SigmaTotU.ba, SigmaTotM.ba	Functions for computing the period-dependent coefficients for the BA08 model. Called by the main function used for calculations, Sa.ba.
Period-dependent coefficients for the CB08 model (coefs.t.cb)	c0.cb, c1.cb, c2.cb, c3.cb, c4.cb, c5.cb, c6.cb, c7.cb, c8.cb, c9.cb, c10.cb, c11.cb, c12.cb, k1.cb, k2.cb, k3.cb, Sigma.LnY.cb, Tau.LnY.cb, Sigma.C.cb, rho.cb	Functions for computing the period-dependent coefficients for the CB08 model. Called by the main function used for calculations, Sa.cb.
Period-dependent coefficients for the CY08 model (coefs.t.cy)	c1.cy, c1a.cy, c1b.cy, cn.cy, cM.cy, c5.cy, c6.cy, c7.cy, c7a.cy, c9.cy, c9a.cy, c10.cy, cGamma1.cy, cGamma2.cy, phi1.cy, phi2.cy, phi3.cy, phi4.cy, phi5.cy, phi6.cy, phi7.cy, phi8.cy, tau1.cy, tau2.cy, sigma1.cy, sigma2.cy, sigma3.cy, sigma4.cy	Functions for computing the period-dependent coefficients for the CY08 model. Called by the main function used for calculations, Sa.cy.
Period-independent coefficients for all models (coefs)	coefs.as, coefs.ba, coefs.cb, coefs.cy	Define the period-independent coefficients for each model. Called by Sa.as, Sa.ba, Sa.cb, and Sa.cy.

¹When functions are grouped together within the HTML help documentation, the name of the function group is listed in parentheses.

²Both the R and Fortran programs use the modified hanging wall term presented in Abrahamson and Silva (2009).

There exist several methods of accessing the help documentation for any of these functions (or groups of functions). First, the user may type

```
> help(foo),
```

where `foo` is the name of a specific function (such as `Sa.as`) or the name of a group of functions (such as `Sa`).

Alternately, the user may type

```
> help.start()
```

and navigate to the help documentation for the `nga` package, as described in the previous section. This page displays a comprehensive list of the external groups of functions available.

Finally, the user may navigate to the documentation for the comprehensive `Sa.nga` function by typing

```
> help(nga),
```

or

```
> help(Sa.nga)
```

and scrolling to the “See Also” portion of the help file where there are links to each of the help files for the other external and internal groups of functions. Although all help files contain a “See Also” section that has links to other relevant help files, the “See Also” section for `Sa.nga` is the most comprehensive.

We will now provide details on the input and output of the most commonly-used functions in the `nga` package: (1) ground-motion predictions for all models (`Sa.nga`), (2) ground-motion predictions for individual models (`Sa.as`, `Sa.ba`, `Sa.cb`, and `Sa.cy`), and (3) estimation of unknown input parameters (`dip.calc`, `Rrup.calc`, `Rx.calc`, `W.calc`, `Ztor.calc`, `Zhyp.calc`, and `Z1.calc.as`, `Z1.calc.cy`). These details also are available in the HTML help documentation, but because these are the functions that are most often used, we find it necessary to include these details within the body of this report. For the other functions not included in this explanation, the user is referred to the HTML help documentation described in the preceding paragraphs.

Function for Obtaining Ground-Motion Predictions from all NGA Models

The function `Sa.nga` is a comprehensive function that estimates ground-motion parameters by using the AS08, BA08, CB08, and CY08 models. This function, which returns the results as a data frame, is designed to mimic the output from the Fortran program to be discussed later in this report. The usage statement for the function is

```
Sa.nga(M, Rjb, Vs30, T, Rrup = NA, Rx = NA, dip = NA, W = NA, Ztor = NA, Z1.0 = NA,
      Z1.5 = NA, Z2.5 = NA, rake = NA, Frv = NA, Fnm = NA, Fhw = NA, azimuth = NA,
      Zhyp = NA, Fas = 0, epsilon = 1).
```

Mandatory arguments do not have default values; optional arguments, if unspecified, assume the respective default values in the usage statement above. If unspecified, arguments with default values of “NA” will be calculated using the methodologies described in table 4, and as previously discussed in the Estimation of Input Parameters section. If the user enters invalid or inconsistent input parameters, the program will print an error message and the calculations will not proceed. (For example, the user cannot enter a positive value for `Rx` and a negative value for `azimuth`.) In table 4, we present the details of the arguments.

Table 4. Arguments for R function Sa.nga.

Argument	Units	Description
M	–	Moment magnitude of earthquake.
Rjb	kilometers	Joyner-Boore distance: the horizontal distance to the surface projection of the rupture plane.
Vs30	meters/ second	Time-averaged shear-wave velocity over a subsurface depth of 30 meters.
T	seconds	Spectral period. Use 0 for PGA and –1 for PGV. For spectral acceleration, T must be in the range $0.01 \leq T \leq 10$ sec. If the specified period is within the allowable range and does not have defined equations, the program uses log-log interpolation (using <code>interpolate</code>) between the next-highest and next-lowest spectral periods with defined equations. Note that T may be a vector, while all other arguments must be scalars.
Rrup	kilometers	Rupture distance: the closest distance to the rupture plane; if left empty, Rrup is calculated from Rx, the source-to-site azimuth, and the geometric rupture parameters (Ztor, W, and dip) using <code>Rrup.calc</code> .
Rx	kilometers	Site coordinate: The horizontal distance to the surface projection of the top edge of the rupture plane, measured perpendicular to the strike. If left empty, Rx is calculated from Rjb, the source-to-site azimuth, and the geometric rupture parameters (Ztor, W, and dip) using <code>Rx.calc</code> . When only Rjb and the azimuth are assumed, Rjb is used to calculate Rx, which is then used to calculate Rrup.
dip	degrees	Dip angle of the rupture plane. If left empty, the dip is estimated using <code>dip.calc</code> .
W	kilometers	Down-dip width of rupture plane. If left empty, W is estimated using <code>W.calc</code> .
Ztor	kilometers	Depth to top of rupture. If left empty, Ztor is estimated using <code>Ztor.calc</code> .
Z1.0	meters	Depth to Vs=1.0 km/s. If left empty, Z1.0 is estimated using <code>Z1.calc.as</code> for the AS08 model and <code>Z1.calc.cy</code> for the CY08 model.
Z1.5	meters	Depth to Vs=1.5 km/s. Z1.5 is not used in ground-motion calculations, but if available, it may be used to estimate Z2.5 for the CB08 model.
Z2.5	meters	Depth to Vs=2.5 km/s. If left empty, Z2.5 is estimated from Z1.5 or Z1.0 if available, using the recommendations in Campbell and Bozorgnia (2007). If neither Z1.5 nor Z1.0 is available, then Vs30 is used to estimate Z1.0 using <code>Z1.calc.as</code> , which is in turn used to estimate Z2.5.
rake	degrees	Rake angle of fault movement. Either the rake angle or the style-of-faulting flag variables (Frv and Fnm) must be specified.
Frv	–	Reverse style-of-faulting flag (1 for reverse faulting, 0 otherwise). Either (a) the rake angle, or (b) both Frv and Fnm, must be specified. Reverse faulting is characterized by rake angles in the range $30^\circ \leq \text{rake} \leq 150^\circ$ for the AS08, BA08, and CY08 models; and in the range $30^\circ < \text{rake} < 150^\circ$ for the CB08 model.
Fnm	–	Normal style-of-faulting flag (1 for normal faulting, 0 otherwise). Either (a) the rake angle, or (b) both Frv and Fnm, must be specified. Normal faulting is characterized by rake angles in the range of $-120^\circ \leq \text{rake} \leq -60^\circ$ for the AS08 and CY08 models, $-150^\circ \leq \text{rake} \leq -30^\circ$ for the BA08 model, and $-150^\circ < \text{rake} < -30^\circ$ for the CB08 model.

Argument	Units	Description
Fhw	–	Hanging wall flag; equal to 1 for sites on the hanging wall side of the fault ($R_x \geq 0$; azimuth ≥ 0), and 0 otherwise. Either Fhw, Rx, or the azimuth must be specified.
azimuth	degrees	Source-to-site azimuth. Used by <code>Rx.calc</code> and <code>Rrup.calc</code> for distance calculations. Either Fhw, Rx, or the azimuth must be specified.
Zhyp	kilometers	Hypocentral depth of the earthquake. <code>Zhyp</code> is not used directly in ground-motion calculations, but it may be used to estimate <code>Ztor</code> . See <code>Ztor.calc</code> for details.
Fas	–	Aftershock flag; equal to 1 for aftershocks and 0 for mainshocks (the default).
epsilon	–	Number of standard deviations to be considered in the calculations (default value is 1). The function <code>Sa.nga</code> automatically outputs the median estimates (corresponding to <code>epsilon = 0</code>), as well as the estimates corresponding to the median estimate plus and minus <code>epsilon · σ_T</code> , where <code>epsilon</code> may be specified in the function call.

The output for function `Sa.nga` is a data frame of 62 elements, described in table 5. The data frame is divided into inputs and outputs. In the nomenclature of the output section of this data frame, “Y” refers to the ground-motion parameter of interest, which can be

- Spectral acceleration, S_a (g),
- Peak ground acceleration, PGA (g), or
- Peak ground velocity, PGV (cm/s).

Because only the CB08 model has coefficients for PGD (peak ground displacement), the CB08-specific function `Sa.cb` must be used to obtain predictions for PGD. In addition, “sd” refers to the standard deviation of the ground-motion estimate, which is presented in natural logarithmic space.

In the output list, the indicator variables `VsFlag` and `arb` refer to

- `VsFlag` = Flag variable indicating how `Vs30` is obtained (AS08 and CY08 models only); equal to 1 if `Vs30` is measured, and 0 if `Vs30` is estimated or inferred.
- `arb` = flag variable indicating the method of determining aleatory uncertainty for the CB08 model; equal to 1 if the standard deviation should be calculated for the arbitrary horizontal component of ground motion, and 0 if the standard deviation should be calculated for the geometric mean horizontal ground motion.

These two indicator variables represent model-specific options for output: AS08 and CY08 have different standard-deviation terms for measured and inferred `Vs30` (specified by `VsFlag`), and CB08 is the only model that offers predictions for the arbitrary horizontal component of ground motion (`arb`). For each case (0 and 1) of each of these two indicator variables, `Sa.nga` provides the estimated ground motion. This output is consistent with that of the Fortran program described later in this report. The model-specific functions `Sa.as`, `Sa.ba`, `Sa.cb`, and `Sa.cy` allow the user to directly specify the values of the indicator variables in the arguments to the functions.

The median BA08 estimate is presented in terms of the original GMPE (Boore and Atkinson, 2008), as well as the modified BA08 GMPE given by Atkinson and Boore (2011). The small-magnitude modification affects ground-motion estimates for $M \leq 5.75$. Predictions using the updated model require that the flag variable `AB11` in function `Sa.ba` be set equal to 1.

Table 5. Output for R function Sa.nga.

Name	Units	Description
Input variables		
T	seconds	Spectral period [input]
M	–	Moment magnitude [input]
Rjb	kilometers	Joyner-Boore distance [input]
Rrup.in	kilometers	Rupture distance [input]
Rrup.out	kilometers	Rupture distance [calculated if Rrup.in is not specified]
Rx.in	kilometers	Site coordinate [input]
Rx.out	kilometers	Site coordinate [calculated if Rx.in is not specified]
azimuth.in	degrees	Source-to-site azimuth [input]
azimuth.out	degrees	Source-to-site azimuth [calculated if azimuth.in is not specified]
Fhw	–	Hanging wall flag
Zhyp.in	kilometers	Hypocentral depth [input]
Zhyp.out	kilometers	Hypocentral depth [calculated if Zhyp.in is not specified]
rake.in	degrees	Rake angle of fault movement [input]
rake.out	degrees	Rake angle of fault movement [calculated if rake.in is not specified]
Frv1	–	Reverse style-of-faulting flag for AS08, BA08, and CY08 [input]
Frv2.cb	–	Reverse style-of-faulting flag for CB08
Fnm1	–	Normal style-of-faulting flag for AB08 and CY08
Fnm2.ba	–	Normal style-of-faulting flag for BA08
Fnm3.cb	–	Normal style-of-faulting flag for CB08
dip.in	degrees	Fault dip angle [input]
dip.out	degrees	Fault dip angle [calculated if dip.in is not specified]
W.in	kilometers	Down-dip rupture width [input]
W.out	kilometers	Down-dip rupture width [calculated if W.in is not specified]
Ztor.in	kilometers	Depth to top of rupture [input]
Ztor.out	kilometers	Depth to top of rupture [calculated if Ztor.in is not specified]
Vs30	meters/ second	Time-averaged shear wave velocity over 30 m subsurface depth [input]
Z1.0in	meters	Depth to Vs of 1.0 km/s [input]
Z1.0as	meters	Depth to Vs of 1.0 km/s [calculated for use in AS08 model, if Z1.0in unknown]
Z1.0cy	meters	Depth to Vs of 1.0 km/s [calculated for use in CY08 model, if Z1.0in unknown]
Z1.5in	meters	Depth to Vs of 1.5 km/s [input]
Z2.5in	meters	Depth to Vs of 2.5 km/s [input]
Z2.5out	meters	Depth to Vs of 2.5 km/s [calculated from Z1.0 for use in CB08 model]
Fas	–	Aftershock flag [input]
epsilon	–	Number of standard deviations considered in the calculations [input]

Name	Units	Description
Output variables		
<i>AS08 Model:</i>		
Y50.as		Median AS08 ground-motion prediction ($\epsilon_{\text{psilon}} = 0$)
YplusEpsilon.meas.as		Upper ¹ AS08 ground-motion prediction, for measured V_{S30} ($\text{VsFlag} = 1$)
YplusEpsilon.est.as		Upper AS08 ground-motion prediction, for estimated V_{S30} ($\text{VsFlag} = 0$)
YminusEpsilon.meas.as		Lower ² AS08 ground-motion prediction, for measured V_{S30} ($\text{VsFlag} = 1$)
YminusEpsilon.est.as		Lower AS08 ground-motion prediction, for estimated V_{S30} ($\text{VsFlag} = 0$)
sdMeas.as		Total AS08 standard deviation, for measured V_{S30} ($\text{VsFlag} = 1$)
sdEst.as		Total AS08 standard deviation, for estimated V_{S30} ($\text{VsFlag} = 0$)
<i>BA08 Model:</i>		
Y50.ba		Median BA08 ground-motion prediction ($\text{AB11} = 0$)
Y50mod.ba		Median BA08 ground-motion prediction with small- M correction ($\text{AB11} = 1$)
YplusEpsilon.ba		Upper BA08 ground-motion prediction ($\text{AB11} = 0$)
YplusEpsilon.mod.ba		Upper BA08 ground-motion prediction with small- M correction ($\text{AB11} = 1$)
YminusEpsilon.ba		Lower BA08 ground-motion prediction ($\text{AB11} = 0$)
YminusEpsilon.mod.ba		Lower BA08 ground-motion prediction with small- M correction ($\text{AB11} = 1$)
sd.ba		Total BA08 standard deviation (same for ($\text{AB11} = 0$ and $\text{AB11} = 1$))
<i>CB08 Model:</i>		
Y50.cb		Median CB08 ground-motion prediction ($\epsilon_{\text{psilon}} = 0$)
YplusEpsilon.GM.cb		Upper CB08 ground-motion prediction for the geometric mean horiz. component ($\text{arb} = 0$)
YplusEpsilon.arb.cb		Upper CB08 ground-motion prediction for the arbitrary horiz. component ($\text{arb} = 1$)
YminusEpsilon.GM.cb		Lower CB08 ground-motion prediction for the geometric mean horiz. component ($\text{arb} = 0$)
YminusEpsilon.arb.cb		Lower CB08 ground-motion prediction for the arbitrary horiz. component ($\text{arb} = 1$)
sdGM.cb		Total CB08 standard deviation for the geometric mean horiz. component ($\text{arb} = 0$)
sdArb.cb		Total CB08 standard deviation for the arbitrary horiz. component ($\text{arb} = 1$)
<i>CY08 Model:</i>		
Y50.cy		Median CY08 ground-motion prediction ($\epsilon_{\text{psilon}} = 0$)
YplusEpsilon.meas.cy		Upper CY08 ground-motion prediction, for measured V_{S30} ($\text{VsFlag} = 1$)
YplusEpsilon.est.cy		Upper CY08 ground-motion prediction, for estimated V_{S30} ($\text{VsFlag} = 0$)
YminusEpsilon.meas.cy		Lower CY08 ground-motion prediction, for measured V_{S30} ($\text{VsFlag} = 1$)
YminusEpsilon.est.cy		Lower CY08 ground-motion prediction, for estimated V_{S30} ($\text{VsFlag} = 0$)
sdMeas.cy		Total CY08 standard deviation, for measured V_{S30} ($\text{VsFlag} = 1$)
sdEst.cy		Total CY08 standard deviation, for estimated V_{S30} ($\text{VsFlag} = 0$)

¹Upper ground-motion predictions correspond to the median estimate plus $\epsilon \cdot \sigma_T$.

²Lower ground-motion predictions correspond to the median estimate minus $\epsilon \cdot \sigma_T$.

The benefits of the comprehensive `Sa.nga` function are the ability to (1) compute ground-motion predictions for the four NGA models simultaneously, (2) review the input parameters to see what values the program has calculated for the unknown arguments, and (3) easily compare with the output from the Fortran program. However, there may be some situations in which the user desires a single value for output, perhaps as an end result or as an intermediate step in later calculations. Individual elements of the output data frame generated by `Sa.nga` are easily extracted (as shown in the examples), but the new R user may be uncomfortable working with data frames. Therefore, we also provide the functions `Sa.as`, `Sa.ba`, `Sa.cb`, and `Sa.cy`, which compute ground-motion predictions for individual models and output results as a single number, rather than as a data frame.

Functions for Obtaining Ground-Motion Predictions from Individual NGA Models

In this section, we describe the usage of the four R functions `Sa.as`, `Sa.ba`, `Sa.cb`, and `Sa.cy`, which are available in the package as group `Sa`. These functions compute ground-motion predictions for individual models and output the results as a single number (Sa, PGA, PGV, or PGD). Only the CB08 model may be used to estimate peak ground displacement (PGD), with units of cm, by setting `T` equal to `-2`. Note that `Sa.nga` calls each of these functions when performing ground-motion predictions. An added benefit of these functions is that the input options are more flexible and model-specific than the more comprehensive `Sa.nga` function. The usage statements for the functions are as follows:

```
Sa.as(M, Rjb, Vs30, VsFlag, epsilon, T, Rrup = NA, Rx = NA, dip = NA, W = NA,
      Ztor = NA, Z1.0 = NA, rake = NA, Frv = NA, Fnm = NA, Fhw = NA, azimuth = NA,
      Zhyp = NA, Fas = 0)
```

```
Sa.ba(M, Rjb, Vs30, epsilon, T, rake = NA, U = NA, SS = NA, NS = NA, RS = NA,
      AB11 = 0)
```

```
Sa.cb(M, Rjb, Vs30, epsilon, T, Rrup = NA, dip = NA, W = NA, Ztor = NA, Z1.0 = NA,
      Z1.5 = NA, Z2.5 = NA, rake = NA, Frv = NA, Fnm = NA, Fhw = NA, azimuth = NA,
      Zhyp = NA, arb = 0)
```

```
Sa.cy(M, Rjb, Vs30, VsFlag, epsilon, T, Rrup = NA, Rx = NA, dip = NA, W = NA,
      Ztor = NA, Z1.0 = NA, rake = NA, Frv = NA, Fnm = NA, Fhw = NA, azimuth = NA,
      Zhyp = NA, AS = 0)
```

Unlike the input for `Sa.nga`, the value of `epsilon` is now a mandatory parameter because only the estimated ground motion corresponding to a specific value of `epsilon` is output. Most of the arguments in the above statements are the same as those in table 4, but additional details of the new arguments are shown in the following table.

Table 6. Additional arguments for R functions Sa.as, Sa.ba, Sa.cb, and Sa.cy.

Argument	Models	Description
<i>Style-of-faulting parameters¹:</i>		
Frv	AS08 CB08 CY08	Reverse style-of-faulting flag (1 for reverse faulting, 0 otherwise) for the AS08, CB08, and CY08 models. Either (a) the rake angle, or (b) both Frv and Fnm, must be specified. Reverse faulting is characterized by rake angles in the range $30^\circ \leq \text{rake} \leq 150^\circ$ for the AS08 and CY08 models, and in the range $30^\circ < \text{rake} < 150^\circ$ for the CB08 model.
Fnm	AS08 CB08 CY08	Normal style-of-faulting flag (1 for normal faulting, 0 otherwise) for the AS08, CB08, and CY08 models. Either (a) the rake angle, or (b) both Frv and Fnm, must be specified. Normal faulting is characterized by rake angles in the range of $-120^\circ \leq \text{rake} \leq -60^\circ$ for the AS08 and CY08 models and $-150^\circ < \text{rake} < -30^\circ$ for the CB08 model.
U	BA08	Unspecified style-of-faulting flag for the BA08 model; equal to 1 if the user wishes to perform a generic ground-motion calculation when the style of faulting is unspecified, and 0 otherwise.
RS	BA08	Reverse style-of-faulting flag for the BA08 model; equal to 1 for reverse faulting ($30^\circ \leq \text{rake} \leq 150^\circ$), and 0 otherwise.
NS	BA08	Normal style-of-faulting flag for the BA08 model; equal to 1 for normal faulting ($-150^\circ \leq \text{rake} \leq -30^\circ$), and 0 otherwise.
SS	BA08	Strike-slip style-of-faulting flag for the BA08 model; equal to 1 for strike-slip faulting (when the rake is not in either of the ranges specified for RS or NS), and 0 otherwise.
<i>Other parameters:</i>		
VsFlag	AS08 CY08	Flag variable indicating how vs30 is obtained; equal to 1 if vs30 is measured, and 0 if vs30 is estimated or inferred.
Fhw	AS08 CB08 CY08	Hanging wall flag; equal to 1 for sites on the hanging wall side of the fault ($R_x > 0$; $\text{azimuth} \geq 0$), and 0 otherwise. For AS08 and CY08, either Fhw, Rx, or the azimuth must be specified. For CB08, the parameters Fhw and azimuth are optional, and they are only used to estimate Rrup when Rrup is unknown; if neither Fhw nor azimuth is specified, the site is assumed to be located on the footwall, and Rrup is easily estimated as $\sqrt{R_{JB}^2 + Z_{TOR}^2}$.
Fas	AS08	Aftershock flag for AS08; equal to 1 for aftershocks and 0 for mainshocks (the default).
AS	CY08	Aftershock flag for CY08; equal to 1 for aftershocks and 0 for mainshocks (the default).
arb	CB08	Flag variable indicating the method of determining aleatory uncertainty for the CB08 model; equal to 1 if the standard deviation should be calculated for the arbitrary horizontal component of ground motion, and 0 if the standard deviation should be calculated for the geometric mean horizontal ground motion (the default).
AB11	BA08	Flag variable equaling 1 if the Atkinson and Boore (2011) small-magnitude correction factor should be applied to the BA08 model, and 0 otherwise.

¹Note that either the rake angle or the style-of-faulting flag variables (Frv and Fnm for AS08, CB08, and CY08; or U, RS, NS, and SS for BA08) must be specified.

Functions for Estimating Unknown Input Parameters

The methodology for estimating unknown input parameters was described in the Estimation of Input Parameters section of this report, and is more greatly explained by Kaklamanos and others (2011). Here, we present the usage of the R functions for estimating unknown input parameters. These functions are used by `Sa.as`, `Sa.ba`, `Sa.cb`, `Sa.cy`, and `Sa.nga`, and also may be used separately. The definitions for the arguments of these functions (`M`, `Rjb`, `Rrup`, `Rx`, `Vs30`, `Ztor`, `W`, `dip`, `rake`, `azimuth`, and `Zhyp`) are the same as those in table 4.

Distance Functions `Rrup.calc` and `Rx.calc`

The distance functions `Rrup.calc` and `Rx.calc` compute the values of the rupture distance (`Rrup`) and site coordinate (`Rx`) from the other distance parameters and the geometric source characteristics of the fault rupture. The usage statements for the functions are

```
Rx.calc(Rjb, Ztor, W, dip, azimuth, Rrup = NA)
Rrup.calc(Rx, Ztor, W, dip, azimuth, Rjb = NA).
```

A general strategy for calculating distances is to first calculate `Rx`, and then to calculate `Rrup` using `Rx`. In order to calculate `Rx` using the function `Rx.calc`, the argument `Rrup` is necessary only when the site is located directly over the ruptured area (`Rjb = 0`). If `Rrup` is unknown in this case, then the function assumes that the site is located in the middle of the surface projection of the ruptured area. In the function `Rrup.calc`, the argument `Rjb` is necessary only in the rare case that the site is located directly on the surface projection of fault strike (`azimuth = 0, 180, or -180`).

Estimation of Fault Dip using the Function `dip.calc`

This function estimates the fault dip angle from the style of faulting (using the rake angle, in degrees), from the guidelines Chiou and Youngs (2008b) used in developing their NGA model (table 2 in this report):

```
dip.calc(rake).
```

Estimation of Down-Dip Rupture Width using the Function `W.calc`

The function `w.calc` estimates the down-dip rupture width (`W`) from magnitude using the empirical correlations published by Wells and Coppersmith (1994) for strike-slip, normal, and reverse-faulting mechanisms (equation 2 in this report):

```
W.calc(M, rake).
```

Estimation of Depth to Top of Rupture using the Function `Ztor.calc`

To estimate `Ztor`, estimates of `w`, `dip`, and `Zhyp` must be provided. The resulting calculation assumes that the hypocenter is located 60 percent down the fault width, as suggested by Mai and others (2005).

```
Ztor.calc(W, dip, Zhyp).
```

Estimation of Hypocentral Depth using the Function `Zhyp.calc`

This function, which may be used to obtain an estimate of `zhyp` in order to employ `Ztor.calc`, uses the correlations presented by Scherbaum and others (2004), as presented in equation 1 of this report:

```
Zhyp.calc(M, rake).
```

Estimation of Depth Parameters using the Functions `Z1.calc.as` and `Z1.calc.cy`

The estimation of the depth parameter `z1.0` (m) for the AS08 and CY08 models, shown in equations 3 and 4, respectively, requires `Vs30` (m/s):

```
Z1.calc.as(Vs30)  
Z1.calc.cy(Vs30).
```

Overview of Usage in R

Here we present a brief overview of usage in R, aimed primarily at users new to R. After the `nga` package has been loaded, the user may employ any of the functions in table 3. For these examples, we assume the following earthquake parameters: $M=7$, $R_{JB}=50$ km, $R_{RUP}=51$ km, $V_{S30}=300$ m/s, $Z_{2.5}=3500$ m, $Z_{TOR}=0$ km, $\text{dip}=90^\circ$, and $\text{rake}=180^\circ$. To assign these variables in an R session, we use the assignment operator: `<-` (an arrow-like symbol). Alternately, the equal sign (`=`) may be used for assignment, but the equal sign typically is used only for assigning arguments within function calls. The assignment of these earthquake parameters in R would be:

```
> M <- 7  
> Rjb <- 50  
> Rrup <- 51  
> Vs30 <- 300  
> Z2.5 <- 3500  
> Ztor <- 0  
> dip <- 90  
> rake <- 180.
```

If we wish to obtain a median estimate of PGA using the CB08 model, we could also assign appropriate values of `epsilon`, `T`, and `arb`:

```
> epsilon <- 0  
> T <- 0  
> arb <- 0.
```

(Technically, the value of `arb` does not matter, because we only compute the median in this example. The value of `arb` only influences the standard deviations.)

To obtain a ground-motion prediction using `Sa.cb`, the user would type

```
> Sa.cb(M = M, Rjb = Rjb, Rrup = Rrup, Vs30 = Vs30, Z2.5 = Z2.5, Ztor = Ztor,  
>       dip = dip, rake = rake, epsilon = epsilon, T = T, arb = arb)
```

into the R terminal. The order of the arguments within the function call does not matter if each argument is named. One can save space by omitting the argument name, as in `Sa.cb(M, ...)`, rather than `Sa.cb(M=M, ...)`; however, when the argument names are omitted, the arguments are matched by their

order. In this example, we have defined variables with the same names as those of the arguments to the function. This is not necessary. For example, let us instead define M by typing

```
> Mw = 7.
```

Now, the function call would read

```
> Sa.cb(M = Mw, Rjb = Rjb, Rrup = Rrup, Vs30 = Vs30, Z2.5 = Z2.5, Ztor = Ztor,  
>       dip = dip, rake = rake, epsilon = epsilon, T = T, arb = arb).
```

In fact, the first step of defining variables is not necessary, although it is often convenient to do so, especially for repeated calculations. The function call

```
> Sa.cb(M = 7, Rjb = 50, Rrup = 51, Vs30 = 300, Z2.5 = 3500, Ztor = 0, dip = 90,  
>       rake = 180, epsilon = 0, T = 0, arb = 0)
```

also is valid syntax.

In creating the `nga` R package, we have designed the inputs to the functions to be as flexible as possible. For example, the user may enter the style-of-faulting flag variables (`Frv` and `Fnm`) in place of the rake:

```
> Sa.cb(M = 7, Rjb = 50, Rrup = 51, Vs30 = 300, Z2.5 = 3500, Ztor = 0, dip = 90,  
>       Frv = 0, Fnm = 0, epsilon = 0, T = 0, arb = 0).
```

The values passed to the functions must be single values, with the exception of spectral period (T), which may be a vector. We have designed the functions in this manner because many users will wish to obtain ground-motion estimates for a suite of spectral periods. A vector in R may be easily defined using the “c” operator. For example, a vector `T.list` of spectral periods 0.1, 0.5, 1.0, and 2.0 s could be defined in R as

```
> T.list = c(0.1, 0.5, 1.0, 2.0).
```

The corresponding ground-motion prediction would be obtained by the command

```
> Sa.cb(M = M, Rjb = Rjb, Rrup = Rrup, Vs30 = Vs30, Z2.5 = Z2.5, Ztor = Ztor,  
>       dip = dip, rake = rake, epsilon = epsilon, T = T.list, arb = arb).
```

When a parameter is unknown, the user has two options: (1) omit the unknown parameter from the function call, or (2) specify the unknown parameter to be `NA`. For example, let us assume that we did not have an estimate of $Z_{2.5}$, and we prefer that the program calculate $Z_{2.5}$ automatically. Either of the following two function calls would achieve the same result:

```
> Sa.cb(M = 7, Rjb = 50, Rrup = 51, Vs30 = 300, Ztor = 0, dip = 90, Frv = 0,  
>       Fnm = 0, epsilon = 0, T = 0, arb = 0),
```

or

```
> Sa.cb(M = 7, Rjb = 50, Rrup = 51, Vs30 = 300, Z2.5 = NA, Ztor = 0, dip = 90,  
>       Frv = 0, Fnm = 0, epsilon = 0, T = 0, arb = 0).
```

More than one argument may be unknown. The following command would generate a CB08 prediction using the bare minimum required input parameters: `M`, `Rjb`, `Vs30`, `epsilon`, `T`, and `rake`:

```
> Sa.cb(M = 7, Rjb = 50, Vs30 = 300, epsilon = 0, T = 0, rake = 180).
```

The examples in this report, as well as more extensive examples in the HTML help documentation, provide further insight into using the `nga` package within R.

Overview of Input/Output in R

In this section, we provide a brief example of a typical calculation session in R. For batch calculations, a common method of storing data for calculations in R is to use a comma-delimited or tab-delimited file. The functions `scan`, `read.table`, `read.csv`, and `file` are all included in the R base package and are frequently used for reading data into R. The R Data Import/Export manual is an excellent reference on importing and exporting data and on the functions used in the process. As described in the manual, there are custom packages in R that read data from specific formats other than those listed above, such as spreadsheets (using the RODBC package by Ripley and Lapsley (2009)).

To read the data to R, you must be sure that your working directory is set to the location of your input file, or use full paths in the file name. To set your working directory in R, use the `setwd` function. For example, the command

```
> setwd("C:/Jim")
```

sets the working directory to “C:\Jim.” For Windows systems, note that forward slashes should be used in the function call to R, instead of the typical backslashes used to represent file paths in Windows. To check your current directory, type

```
> getwd().
```

The command

```
> EQdata <- read.csv(file = "EQdata.csv", head = TRUE)
```

reads data from a file named “EQdata.csv” (with a header row) into a newly-defined R object entitled `EQdata`. Now, you are ready to perform calculations in R using this data.

In the following section (and in the HTML help documentation, particularly for the `KBflatfile` data), we provide examples of working with data objects in R. On a basic level, it is important to understand that the dollar sign (`$`) is used to access elements of a list or columns of a data frame. In R, a list is a flexible data structure that is similar to a spreadsheet in which the columns can have unequal lengths, whereas the columns of a data frame must all have the same length (a data frame is a special case of a list). A common use of the dollar sign is to access individual columns of an object you have read into R (using the `read.table` family of functions), where the returned R object is a data frame. The object produced by `Sa.nga` is a data frame, and, therefore, the dollar sign is used to access the individual elements. For example, say that the result of some ground-motion calculation is stored in an object entitled `x`:

```
> x <- Sa.nga(M = 6, Rjb = 10, Vs30 = 500, rake = 90, Fhw = 0, T = 0).
```

To see the names of the components of the data frame `x`, type

```
> names(x).
```

From reading the help file, you should be aware that `Sa.nga` returns a data frame. If you forgot, then you can learn more about the type and structure of an R object by typing:

```
> str(x).
```

The object `x` is a data frame containing 62 column vectors, each of unit length (that is, one row), and having the names given by the command `names(x)`. Let us assume we wish to access the numerical value of the BA08 median prediction. To do so, we use the dollar sign (`$`) followed by the name of the element:

```
> x$Y50.ba.
```

To access particular elements of a vector or matrix, brackets should be used. For example, for the following vector,

```
> T.list = c(0.1, 0.5, 1.0, 2.0)
```

typing `T.list[1]` would access 0.1, `T.list[2]` would access 0.5, and so on. If the ground-motion calculation were repeated with the vector `T.list` instead of `T`,

```
> x <- Sa.nga(M = 6, Rjb = 10, Vs30 = 500, rake = 90, Fhw = 0, T = T.list).
```

The structure of `x` is a data frame containing 62 column vectors, each of length four (that is, four rows). Now, the command

```
> x$Y50.ba
```

returns four elements. To access the first element (corresponding to $T=0.1$), we would type

```
> x$Y50.ba[1]
```

After you have performed calculations in R, you may wish to output the results of your calculations. Your data should be stored in some type of structure, such as a list, vector, matrix, or data frame. Say your data is stored in a matrix called `Results` (note that a matrix is a special case of a data frame in which all elements are of the same data type). To export the contents of `Results` to a file, you may use the functions `write`, `write.table`, `write.csv`, or `file`, all of which are part of the R base package. For example, the command

```
> write.csv(Results, file = "ResultsFile.csv")
```

would output the data to a .csv file titled `ResultsFile.csv`. Alternately, the command

```
> write.table(Results, file = "ResultsFile.txt")
```

outputs the data to a space-delimited text file titled `ResultsFile.txt`. If you wish to use R for later calculations, it might be advantageous to save the data in binary format using the `save` function, which saves the R-object to a file. The data may then be read back into R by using the `load` function. The R Data Import/Export manual describes additional methods of exporting data.

The `KBflatfile` page within the HTML help documentation, as well as the examples to follow in this report, provides additional examples of inputting and outputting data to and from R using an actual dataset.

Example Sessions

The following four examples use the `KBflatfile` dataset, which is included with the R package. In Kaklamanos and Baise (2011), this dataset is available as table S1 in the electronic supplement. It contains the values of PGA and S_a for 1,060 ground-motion records from seven recent earthquakes recorded in California: (1) the 2003 M 6.5 San Simeon, (2) the 2004 M 6.0 Parkfield, (3) the 2005 M 5.2 Anza, (4) the 2007 M 5.4 Alum Rock, (5) the 2008 M 5.4 Chino Hills, (6) the 2010 M 7.2 Baja, and (7)

the 2010 *M* 5.7 Ocotillo earthquakes. Because none of these earthquakes were present in the database used to develop the NGA models, these records were used in a blind comparison test of the NGA models, as presented in Kaklamanos and Baise (2011). The headers of this data frame are designed to be similar to those in the NGA flatfile; this data frame is termed the “KB flatfile” (“KB” stands for “Kaklamanos and Baise”).

In the first example, we illustrate how to generate a plot of observed versus predicted response spectra for a single ground-motion record. In the second example, we generate a plot of peak ground acceleration versus distance for the 2008 *M* 5.4 Chino Hills earthquake. Third, we explain how to generate a table of observed and predicted peak ground accelerations by using the ground-motion records of the 2003 *M* 6.5 San Simeon earthquake. Finally, in the fourth example, we generate matrices of median predicted response spectra for the same earthquake in example 3. These examples are also documented in the HTML help page for the `KBflatfile` dataset.

Example 1

To access the `KBflatfile` dataset, type the command

```
> data(KBflatfile).
```

To see the column names of the dataset, type

```
> names(KBflatfile).
```

To see additional information about the type of data in each column, type

```
> str(KBflatfile).
```

More details on the columns in the dataset are available in the HTML help documentation for this dataset and in the electronic supplement to Kaklamanos and Baise (2011).

For this example, we will create a plot of observed versus predicted spectral acceleration for the first ground-motion record for the 2010 *M* 7.2 Baja earthquake (record no. 824 in the dataset). To see a summary of these values quickly, type

```
> KBflatfile[824,].
```

First, we extract the data from the 824th row:

```
> # Input parameters
> n <- 824
> M <- KBflatfile$M[n]
> dip <- KBflatfile$Dip[n]
> rake <- KBflatfile$Rake[n]
> Ztor <- KBflatfile$Ztor[n]
> W <- KBflatfile$W[n]
> Rjb <- KBflatfile$Rjb[n]
> Rrup <- KBflatfile$Rrup[n]
> Rx <- KBflatfile$Rx[n]
> azimuth <- KBflatfile$Azimuth[n]
> Vs30 <- KBflatfile$Vs30[n]

> # Observed response spectral values
> PGA <- KBflatfile$PGA[n]
> Sa0.1 <- KBflatfile$T0.1S[n]
> Sa0.2 <- KBflatfile$T0.2S[n]
> Sa0.3 <- KBflatfile$T0.3S[n]
```

```

> Sa0.5 <- KBflatfile$T0.5S[n]
> Sa1.0 <- KBflatfile$T1.0S[n]
> Sa2.0 <- KBflatfile$T2.0S[n].

```

For the purposes of plotting, we vectorize the observed spectral acceleration and the corresponding periods. In generating the plot, PGA is assumed to have a spectral period of 0.01 s.

```

> T.obs <- c(0.01, 0.1, 0.2, 0.3, 0.5, 1.0, 2.0)
> Sa.obs <- c(PGA, Sa0.1, Sa0.2, Sa0.3, Sa0.5, Sa1.0, Sa2.0)

```

In order to generate the predicted response spectra, we must select the desired spectral periods for the calculation. In this example, we will use the periods with defined model coefficients (although any spectral period in the range $0.01 \leq T \leq 10$ sec will suffice):

```

> T.list <- c(0.01, 0.02, 0.03, 0.04, 0.05, 0.075, 0.10, 0.15, 0.20, 0.25, 0.30,
>            0.40, 0.50, 0.75, 1.0, 1.5, 2.0, 3.0, 4.0, 5.0, 7.5, 10.0) .

```

Note that the command

```

> modelPeriods(model = "AS08", positive = TRUE)

```

would have also generated identical results, since the periods in `T.list` are the periods at which the AS08 model has defined coefficients. (The AS08 and CY08 models differ from the BA08 and CB08 models in that the AS08 and CY08 models specify model coefficients for a spectral period of 0.04 s, whereas the BA08 and CB08 models do not.)

To compute ground-motion predictions, we use the `Sa.nga` function and store the results in a data frame entitled `ResultsDF`:

```

> ResultsDF <- Sa.nga(M = M, dip = dip, rake = rake, Ztor = Ztor, W = W,
>                   Rjb = Rjb, Rrup = Rrup, Rx = Rx, azimuth = azimuth,
>                   Vs30 = Vs30, epsilon = 1, T = T.list).

```

We access individual columns of the data frame using the `$` symbol:

```

> SaAS08 <- ResultsDF$Y50.as
> SaBA08 <- ResultsDF$Y50.ba
> SaCB08 <- ResultsDF$Y50.cb
> SaCY08 <- ResultsDF$Y50.cy.

```

Finally, the results may be plotted using the commands shown below. The resulting figure (fig. 3) is generated using this code.

```

> plot(T.obs, Sa.obs, type = "p", log = "xy", col = "black", pch = 19, lwd = 4,
>       xlim = c(0.01, 10), ylim = c(0.001, 1), xaxs = "i", yaxs = "i",
>       xlab = "Spectral Period, T [s]", ylab = "Spectral Acceleration, Sa [g]")
> lines(T.list, SaAS08, lwd = 2, col = "blue")
> lines(T.list, SaBA08, lwd = 2, col = "red")
> lines(T.list, SaCB08, lwd = 2, col = "darkgreen")
> lines(T.list, SaCY08, lwd = 2, col = "purple")
> legend(x = "bottomleft", inset = 0.02, bty = "n", lty = c(-1,1,1,1,1),
>        pch = c(19,-1,-1,-1,-1), lwd = c(-1,2,2,2,2),
>        col = c("black", "blue", "red", "darkgreen", "purple"),
>        legend = c("Observed", "AS08", "BA08", "CB08", "CY08"))

```

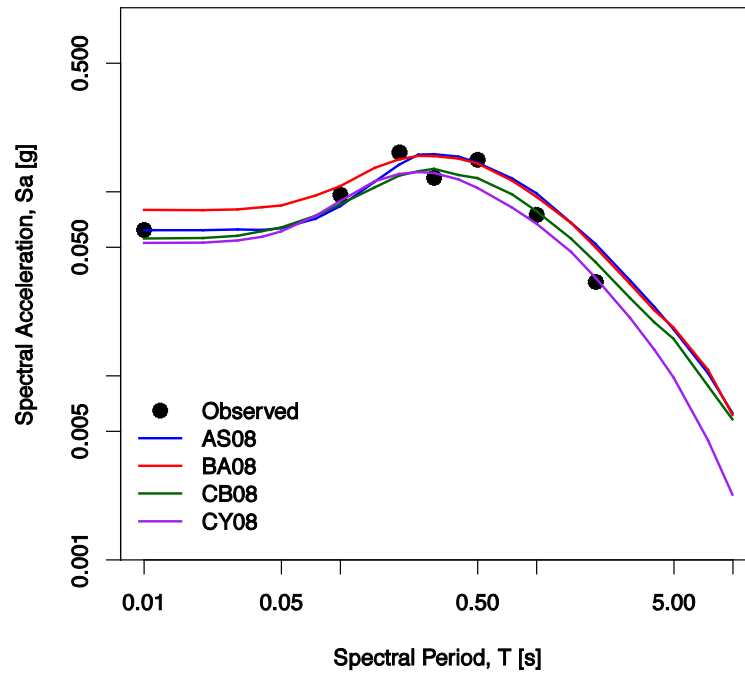


Figure 3. Comparison of NGA ground-motion predictions for KBflatfile record no. 824 (2010 Baja earthquake recorded in Winterhaven, California).

Example 2

In example 2, we generate a plot of peak ground acceleration versus distance for the 2008 M 5.4 Chino Hills earthquake. The relevant ground-motion records are present in rows 447 to 823 of the `KBflatfile` dataset. Because a finite fault model was not developed for this earthquake, some of the source and distance parameters are unknown and must be estimated by the program. First, we extract the relevant data:

```
> # Define starting and ending rows for the Chino Hills earthquake
> start <- 447
> end <- 823

> # Define the sequence {447, 448, 449, ..., 823}
> n <- seq(from = start, to = end, by = 1)

> # Create sub-vectors relevant for the Chino Hills earthquake
> M <- KBflatfile$M[n]
> rake <- KBflatfile$Rake[n]
> dip <- KBflatfile$Dip[n]
> Zhyp <- KBflatfile$Zhyp[n]
> Repi <- KBflatfile$Repi[n]
> Vs30 <- KBflatfile$Vs30[n]
> PGA <- KBflatfile$PGA[n].
```

The source parameters M , $rake$, dip , and Z_{hyp} (used for estimating Z_{tor}) are constant for a single earthquake; thus, we may extract the values from any row of these vectors:

```

> M.value <- M[1]
> rake.value <- rake[1]
> dip.value <- dip[1]
> Zhyp.value <- Zhyp[1].

```

For the purpose of drawing the plots, assume the average value of v_{s30} among the 337 records:

```

> Vs30.value <- mean(Vs30).

```

Because the earthquake has such a low magnitude, hanging wall effects are not likely to be significant; thus, it is not unreasonable to neglect hanging wall effects and assume that the site is located on the footwall:

```

> Fhw <- 0.

```

First, illustrate the calculation for one point ($R_{jb} = 0$):

```

> ResultsMatrix1 <- Sa.nga(M = M.value, dip = dip.value, rake = rake.value,
>                           Rjb = 0, Fhw = 0, Vs30 = Vs30.value, epsilon = 0, T = 0).

```

For the purposes of plotting, we will generate ground-motion predictions for a sequence of distances from 0 to 200 km, in increments of 4 km:

```

> Rjb.plot <- seq(from = 0, to = 200, by = 4).

```

To perform calculations for all locations in `Rjb.plot`, we first define data frame entitled `ResultsMatrix2`, which has the same column names as `ResultsMatrix1`:

```

> ResultsMatrix2 <- matrix(nrow = length(Rjb.plot), ncol = length(ResultsMatrix1))
> ResultsMatrix2 <- as.data.frame(ResultsMatrix2)
> names(ResultsMatrix2) <- names(ResultsMatrix1).

```

It is necessary to place the calculations in a loop, because we are varying R_{jb} . In each iteration of the loop, we assign the output of `Sa.nga` to a row in `ResultsMatrix2`, with each row corresponding to a particular value of R_{jb} .

```

> for(i in 1:length(Rjb.plot)){
>   ResultsMatrix2[i,] <- Sa.nga(M = M.value, dip = dip.value, rake = rake.value,
>                               Rjb = Rjb.plot[i], Fhw = 0, Vs30 = Vs30.value,
>                               epsilon = 0, T = 0)
> }

```

Like Example 1, we access individual columns of the data frame using the `$` symbol:

```

> pgaAS08 <- ResultsMatrix2$Y50.as
> pgaBA08 <- ResultsMatrix2$Y50.ba
> pgaCB08 <- ResultsMatrix2$Y50.cb
> pgaCY08 <- ResultsMatrix2$Y50.cy.

```

Finally, we generate figure 4 using the code below. Note that `Repi` has been used in place of R_{jb} for the purpose of plotting the observed data; for small earthquakes without finite fault models, this is a reasonable approximation.

```

> plot(Repi, PGA, type = "p", log = "y", pch = 1, xlab = "Joyner-Boore Distance,
>   Rjb [km]", ylab = "Peak Ground Acceleration, PGA [g]")
> lines(Rjb.plot, pgaAS08, lwd = 2, col = "blue")
> lines(Rjb.plot, pgaBA08, lwd = 2, col = "red")

```

```

> lines(Rjb.plot, pgaCB08, lwd = 2, col = "darkgreen")
> lines(Rjb.plot, pgaCY08, lwd = 2, col = "purple")
> legend(x = "bottomleft", inset = 0.02, pch = c(1,-1,-1,-1,-1),
>       lwd = c(-1,2,2,2,2), lty = c(-1,1,1,1,1), bty = "n",
>       col = c("black", "blue", "red", "darkgreen", "purple"),
>       legend = c("Observed", "AS08", "BA08", "CB08", "CY08"))

```

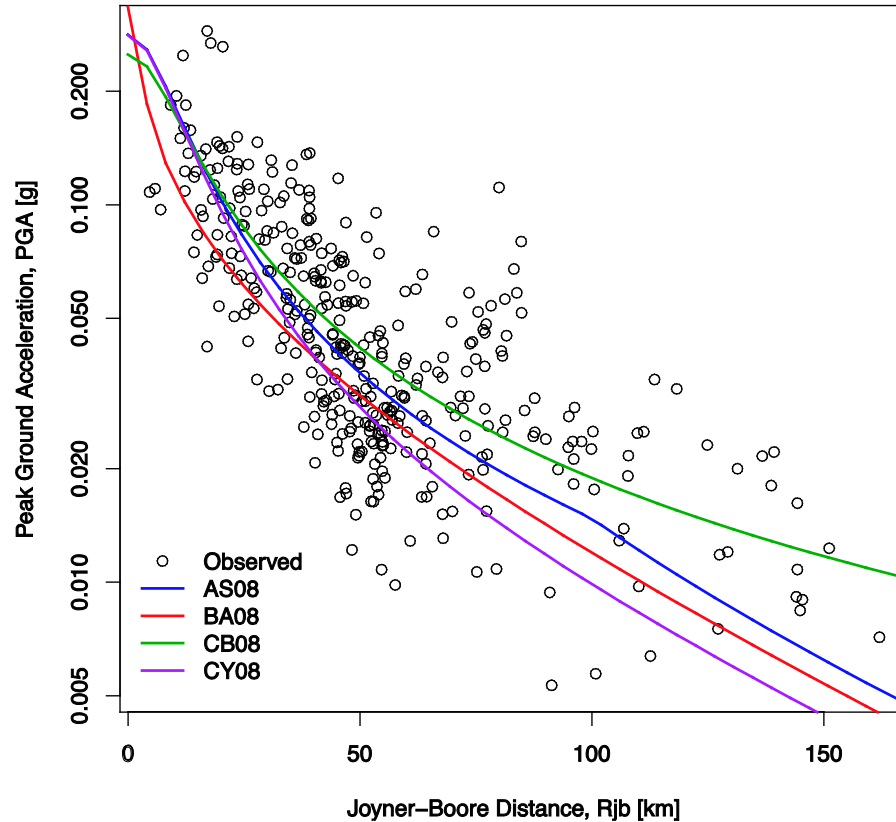


Figure 4. Comparison of NGA ground-motion predictions: PGA versus distance for the 2008 Chino Hills, California, earthquake.

Example 3

We will generate tables of observed and predicted PGA for the ground-motion records of the 2003 M 6.5 San Simeon earthquake. The relevant ground-motion records are in rows 1 to 30 of the KBflatfile, and are read with the following code:

```

> start <- 1
> end <- 30
> n <- seq(from = start, to = end, by = 1)
> M <- KBflatfile$M[n]
> dip <- KBflatfile$Dip[n]
> rake <- KBflatfile$Rake[n]
> Ztor <- KBflatfile$Ztor[n]
> W <- KBflatfile$W[n]
> Rjb <- KBflatfile$Rjb[n]
> Rrup <- KBflatfile$Rrup[n]
> Rx <- KBflatfile$Rx[n]
> azimuth <- KBflatfile$Azimuth[n]

```

```
> Vs30 <- KBflatfile$Vs30[n]
> PGA.obs <- KBflatfile$PGA[n].
```

Let us create matrices to store the calculated values:

```
> pgaAS08 <- matrix(nrow = length(n), ncol = 1)
> pgaBA08 <- matrix(nrow = length(n), ncol = 1)
> pgaCB08 <- matrix(nrow = length(n), ncol = 1)
> pgaCY08 <- matrix(nrow = length(n), ncol = 1).
```

The ground-motion calculations are performed with the following code. The input parameters for each successive iteration of the loop are located in successive rows in the `KBflatfile` data frame (as extracted in the first part of this example). After the ground-motion calculations are performed using `Sa.nga`, the four desired numbers (median PGA from the AS08, BA08, CB08, and CY08 models) are stored in the corresponding matrices `pgaAS08`, `pgaBA08`, `pgaCB08`, and `pgaCY08`.

```
> for(i in 1:length(n)){
>   ResultsMatrix <- Sa.nga(M = M[i], dip = dip[i], rake = rake[i], Ztor = Ztor[i],
>   W = W[i], Rjb = Rjb[i], Rrup = Rrup[i], Rx = Rx[i],
>   azimuth = azimuth[i], Vs30 = Vs30[i], epsilon = 0,
>   T = 0)
>   pgaAS08[i] <- ResultsMatrix$Y50.as
>   pgaBA08[i] <- ResultsMatrix$Y50.ba
>   pgaCB08[i] <- ResultsMatrix$Y50.cb
>   pgaCY08[i] <- ResultsMatrix$Y50.cy
> }
```

Combine the observed and predicted results into a single data frame, and name the columns of the data frame accordingly:

```
> Ex3 <- cbind(PGA.obs, pgaAS08, pgaBA08, pgaCB08, pgaCY08)
> colnames(Ex3) <- c("pgaObs", "pgaAS08", "pgaBA08", "pgaCB08", "pgaCY08").
```

Display the results:

```
> Ex3.
```

The first five rows of output are replicated here:

	pgaObs	pgaAS08	pgaBA08	pgaCB08	pgaCY08
[1,]	0.012908338	0.01133973	0.01292844	0.01923762	0.009633979
[2,]	0.139227123	0.06986381	0.10275051	0.09595295	0.087869308
[3,]	0.018965459	0.02574668	0.04171581	0.03464685	0.023029267
[4,]	0.005000487	0.01082196	0.01099226	0.01923376	0.008327123
[5,]	0.026162640	0.03629257	0.05963713	0.04787469	0.037672263.

We could now use functions such as `write.csv` or `write.table` to export `Ex3` to a `.csv` or text file.

Example 4

In the final example, we generate matrices of the median predicted response spectra for the 2003 *M* 6.5 San Simeon earthquake. This example is similar to the previous example in that we use the same data, but we illustrate the ground-motion calculations using the individual functions `Sa.as`, `Sa.ba`, `Sa.cb`, and `Sa.cy`, which are faster and generate less output than `Sa.nga`. The code for reading the input parameters is the same as in Example 3, except we also read `VsFlag`, which is necessary for `Sa.as` and `Sa.cy`, but not for `Sa.nga` (as in Example 3):

```
> VsFlag <- KBflatfile$VsFlag[n].
```

The observed response spectra may be read as follows:

```
> PGA <- KBflatfile$PGA[n]
> Sa0.1 <- KBflatfile$T0.1S[n]
> Sa0.2 <- KBflatfile$T0.2S[n]
> Sa0.3 <- KBflatfile$T0.3S[n]
> Sa0.5 <- KBflatfile$T0.5S[n]
> Sa1.0 <- KBflatfile$T1.0S[n]
> Sa2.0 <- KBflatfile$T2.0S[n].
```

To combine the individual vectors into a data frame, use the `cbind` function:

```
> Obs <- cbind(PGA, Sa0.1, Sa0.2, Sa0.3, Sa0.5, Sa1.0, Sa2.0).
```

The corresponding spectral periods for analysis are

```
> T.list <- c(0, 0.1, 0.2, 0.3, 0.5, 1.0, 2.0).
```

The matrices for storing the calculated values are created here, and their column headings are declared to be the same as those of `Obs`:

```
> PredAS08 <- matrix(nrow = length(n), ncol = length(T.list))
> PredBA08 <- matrix(nrow = length(n), ncol = length(T.list))
> PredCB08 <- matrix(nrow = length(n), ncol = length(T.list))
> PredCY08 <- matrix(nrow = length(n), ncol = length(T.list))
> colnames(PredAS08) <- colnames(Obs)
> colnames(PredBA08) <- colnames(Obs)
> colnames(PredCB08) <- colnames(Obs)
> colnames(PredCY08) <- colnames(Obs)
```

The ground-motion calculations are performed in the following loop. In each of the 30 iterations of the loop, a response spectrum of 7 periods is stored in a row of each of the `PredAS08`, `PredBA08`, `PredCB08`, and `PredCY08` matrices.

```
> for(i in 1:length(n)){
>   PredAS08[i,] <- Sa.as(M = M[i], dip = dip[i], rake = rake[i], Ztor = Ztor[i],
>                       W = W[i], Rjb = Rjb[i], Rrup = Rrup[i], Rx = Rx[i],
>                       azimuth = azimuth[i], Vs30 = Vs30[i], VsFlag = VsFlag[i],
>                       Fas = 0, epsilon = 0, T = T.list)
>   PredBA08[i,] <- Sa.ba(M = M[i], rake = rake[i], Rjb = Rjb[i], Vs30 = Vs30[i],
>                        epsilon = 0, T = T.list)
>   PredCB08[i,] <- Sa.cb(M = M[i], dip = dip[i], rake = rake[i], Ztor = Ztor[i],
>                        Rjb = Rjb[i], Rrup = Rrup[i], Vs30 = Vs30[i],
>                        epsilon = 0, T = T.list)
>   PredCY08[i,] <- Sa.cy(M = M[i], dip = dip[i], rake = rake[i], Ztor = Ztor[i],
>                        W = W[i], Rjb = Rjb[i], Rrup = Rrup[i], Rx = Rx[i],
>                        azimuth = azimuth[i], Vs30 = Vs30[i], VsFlag = VsFlag[i],
>                        AS = 0, epsilon = 0, T = T.list)
> }
```

Display the results:

```
> Obs
> PredAS08
> PredBA08
> PredCB08
```

> PredCY08.

The first five rows of output for PredAS08 are replicated here.

	PGA	Sa0.1	Sa0.2	Sa0.3	Sa0.5	Sa1.0	Sa2.0
[1,]	0.01133973	0.01469149	0.02522003	0.02951421	0.02702908	0.01697693	0.007025224
[2,]	0.06986381	0.11742307	0.16164464	0.15270118	0.11466047	0.06018999	0.023054770
[3,]	0.02574668	0.03543278	0.06091671	0.06999945	0.06358901	0.04301014	0.022244179
[4,]	0.01082196	0.01300136	0.02403820	0.03089545	0.03158505	0.02281920	0.010993266
[5,]	0.03629257	0.05440992	0.08553026	0.09033717	0.07528177	0.04539984	0.020515025

As with example 3, we could now use functions such as `write.csv` or `write.table` to export these matrices to .csv or text files.

Implementation in Fortran

The Fortran program `nga_gm_tmr` evaluates four of the five PEER NGA ground-motion prediction equations (AS08, BA08, CB08, and CY08). The program uses a control file to specify the values of the input variables, including the magnitude, distance, and period. This is a more general way of generating output for plotting than special purpose programs that evaluate the GMPEs versus distance or versus magnitude. The user interface for the Fortran program is quite different from that of the R program, and there are some minor differences in input and output, but the fundamental calculations are the same.

Program Acquisition and Use

This Fortran program is available by downloading the compressed binary file `nga_gm_tmr_zips.zip` from David Boore's Web page (<http://www.daveboore.com/>, last accessed 11 January 2011) under "Online Software," and also may be accessed from the Web site associated with this report (<http://pubs.usgs.gov/of/2010/1296/>). Within the .zip file, there are 18 files described in detail in the following table.

Table 7. Contents of file `nga_gm_tmr_zips.zip`.

Name of file	Description	Necessary to employ the program?
AS08_COEFS.TXT	Model coefficients for AS08	Yes
BA08_COEFS.TXT	Model coefficients for BA08	Yes
CB08_COEFS.TXT	Model coefficients for CB08	Yes
CY08_COEFS.TXT	Model coefficients for CY08	Yes
hanging_wall_example_m7_vs30_760_test3.out	Example output file	No
nga_gm_tmr.ct1	Default control file for the program	Yes, unless another control file is specified when <code>nga_gm_tmr.exe</code> is run
nga_gm_tmr.exe	File used to execute the program	Yes

Name of file	Description	Necessary to employ the program?
<code>nga_gm_tmr.for</code>	Source code for the <code>nga_gm_tmr</code> program	No, unless the user chooses to recompile the program
<code>nga_gm_tmr_prepare_ctl.xls</code>	Spreadsheet used to generate the data in the control file	No
<code>nga_gm_tmr_subs.for</code>	Source code of subroutines called by the <code>nga_gm_tmr</code> program	No, unless the user chooses to recompile the program
<code>rjb2rrup_gen_m_5_6.75.txt</code>	Scherbaum and others (2004) coefficients for converting R_{JB} to R_{RUP} ; general fault type, low magnitude	Yes
<code>rjb2rrup_gen_m_6.75_7.5.txt</code>	Scherbaum and others (2004) coefficients for converting R_{JB} to R_{RUP} ; general fault type, high magnitude	Yes
<code>rjb2rrup_shd_m_5_6.75.txt</code>	Scherbaum and others (2004) coefficients for converting R_{JB} to R_{RUP} ; shallow-dipping fault type, low magnitude	Yes
<code>rjb2rrup_shd_m_6.75_7.5.txt</code>	Scherbaum and others (2004) coefficients for converting R_{JB} to R_{RUP} ; shallow-dipping fault type, high magnitude	Yes
<code>rjb2rrup_ss_m_5_6.75.txt</code>	Scherbaum and others (2004) coefficients for converting R_{JB} to R_{RUP} ; strike-slip fault type, low magnitude	Yes
<code>rjb2rrup_ss_m_6.75_7.5.txt</code>	Scherbaum and others (2004) coefficients for converting R_{JB} to R_{RUP} ; strike-slip fault type, high magnitude	Yes
<code>some_comments_on_nga_gm_tmr_v(#).doc</code>	Details and instructions for the Fortran program <code>nga_gm_tmr</code>	No
<code>some_comments_on_nga_gm_tmr_v(#).pdf</code>	Details and instructions for the Fortran program <code>nga_gm_tmr</code>	No

The program is executed from a command-prompt window. At the start of program execution, the user must enter the name of the control file; the default is `nga_gm_tmr.ctl`, but any control file may be used (provided the user has created it). The control file should be located in the same directory as the executable file, `nga_gm_tmr.exe`. The 10 coefficient files in table 7 are necessary for the program to function, and they can be stored anywhere on the user's computer. For simplicity, we recommend that the coefficient files be stored in the same directory as the control and executable files. If the coefficient files are stored elsewhere, then their exact location must be noted in the control file. (Instructions on how to edit the control file will follow.) The files ending with `.ctl`, `.for`, `.out`, and `.txt`, are text files that may be opened with any text editor. For the proper functioning of the program, we recommend that only the control file be modified by the user.

To build the control file, we find that a spreadsheet is useful; nga_gm_tmr_prepare_ctl.xls is provided with the program. In addition, a text editor that allows block operations may also be helpful; we recommend TextPad, which is available at <http://www.textpad.com/> (last accessed 11 January 2011). The top portion of the default control file nga_gm_tmr_ctl is displayed below.

```

! Control file for program nga_gm_tmr.for
! Revision of program involving a change in the control file on this date:
  09/09/10
!Header to add to output file (no "!" at beginning!)
! "[blank]" means that no header is printed!
! [blank]
! Sample header here
!name of path in which these coefficient files are stored:
!
!       AS08_COEFS.TXT
!       BA08_COEFS.TXT
!       CB08_COEFS.TXT
!       CY08_COEFS.TXT
! ** DO NOT FORGET CLOSING "\" IN PATH **
! C:\gm_predictions\nga_files\
!name of path in which these coefficient files for rjb2rrup are stored:
!
!       rjb2rrup_gen_m_5_6.75.txt
!       rjb2rrup_gen_m_6.75_7.5.txt
!       rjb2rrup_shd_m_5_6.75.txt
!       rjb2rrup_shd_m_6.75_7.5.txt
!       rjb2rrup_ss_m_5_6.75.txt
!       rjb2rrup_ss_m_6.75_7.5.txt
! ** DO NOT FORGET CLOSING "\" IN PATH **
! C:\gm_predictions\nga_files\
!name of output file:
! test absrake gt 180.out
!PSA, PGA values in cm/s/s (gals)? (Y,N):
! Note: N = units in g:
! N
! hanging_wall_example_m7_vs30_760.out
! foot_wall_example_m7_vs30_760.out
! foot_wall_example_m7_vs30_760.out
! available periods, BA08: -1.0 0.0 0.01 0.02 0.03 0.05 0.075 0.1 0.15 0.2 0.25 0.3 0.4 0.5 0.75 1 1.5 2 3 4 5 7.5 10
! force the program to derive Ztor from RHyp and Wells and Coppersmith W by setting Ztor<0.0
! If dip < 0, the program uses generic values of 90 for SS, 55 for N, 40 for r (after CY08)
! abs(rake) > 180.0 will result in motion for an undefined fault type for BA08 and null values for the other GMPES
! Note: Fhw=1 for hw; Rjb, Rrup, Rx, Zhyp, Ztor in km; Vs30 in m/s; Zsed1.0, Zsed2.5 in m (CB use km for Zsed2.5, but
! the NGA flatfile uses m).
!
! Undefined values (values will be assigned, unless pairs of values are
! inconsistent, as noted below):
!
!       Fhw < 0;
!       abs(az)>180;
!       rrup<0;
!       zhyp<0;
!       dip<0; w<0;
!       ztor<0;
!       Zsed1p0as08<0;
!       Zsed2p5<0
!
! Inconsistent values (program will stop):
!
!       Fhw < 0.0 .and. abs(az) > 180.0;
!       Fhw == 0.0 .and. 0<=az<=180;
!       Fhw == 1.0 .and. -180<az<0
!
!       Rx is computed from provided information, rather than being an input parameter.
!
! Minimum required input parameters:
!
!       T
!       M
!       Rjb
!       Fhw or az
!       rake (0.0, 90.0, -90.0 for SS, RS, NS)
!       Vs30
!
!hanging wall example
!T      M      Fhw      Az      Rjb(km)      Rrup(km)      Zhyp(km)      rake      Dip      W(km)      Ztor(km)      V30(m/s)      Zsed1p0(m)      Zsed2p5(m)      as(1=aftershock)
0.200 7 +1 +90 0      -1      5.00      90      -1      -1      -1      760      -1.0      -1.0      0
0.200 7 +1 +90 0      -1      5.00      -190 -1      -1      -1      760      -1.0      -1.0      0
0.200 7 +1 +90 0      -1      5.00      +190 -1      -1      -1      760      -1.0      -1.0      0
0.200 7 +1 +90 0      -1      -5.00      +190 -1      -1      -1      760      -1.0      -1.0      0
!stop

```

The portions of the control file that the user typically will modify are highlighted in bold, italics, and underline. Comment lines begin with an exclamation point (!), and all data below the “stop” line are ignored by the program. First, the user may wish to print a header to the output file (this is optional). To do so, the user should modify the line “Sample header here”; if no header is desired, the user should enter “[blank]” (without quotes). Second, the user must enter the paths to the two sets of coefficient files (the path to the model coefficient files, and the path to the coefficients for the conversion from R_{JB} to R_{RUP}). The results of the empirical distance conversions of Scherbaum and others (2004) are provided as output, but the geometric distance conversions of Kaklamanos and others (2011) are used in the ground-motion calculations, as explained by Kaklamanos and others (2011). If the files are located in the same directory as the control and executable files, then no paths need to be entered; the two relevant lines in the control file may be left blank. Third, the user should enter the desired name of the output file (which ends in “.out”), and specify the desired units of output (cm/s² as “Y”, and g as “N”). Finally, the user will enter the data for the input parameters. The details of the inputs and outputs will be discussed in the next section.

Inputs and Outputs

Input parameters should be entered directly into the control file. Details on the input parameters are shown in table 8.

Table 8. Inputs for Fortran control file.

Argument	Units	Description	Required?	How to specify if unknown
T	seconds	Spectral period. Use 0 for PGA and -1 for PGV.	Yes	-- ¹
M	–	Moment magnitude of earthquake.	Yes	--
Fhw	–	Hanging wall flag; equal to 1 for sites on the hanging wall side of the fault and 0 otherwise. Either Fhw or Az must be specified.	No	Fhw < 0
Az	degrees	Source-to-site azimuth. Either Az or Fhw must be specified.	No	Az > 180
Rjb	kilometers	Joyner-Boore distance: the horizontal distance to the surface projection of the rupture plane.	Yes	--
Rrup	kilometers	Rupture distance: the closest distance to the rupture plane; if entered as a negative number, R_{rup} is calculated from R_x (which is calculated from R_{jb}), the source-to-site azimuth, and the geometric rupture parameters (Z_{tor} , W , and dip).	No	$R_{rup} < 0$
Zhyp	kilometers	Hypocentral depth of the earthquake. Z_{hyp} is not used in ground-motion calculations, but it may be used to estimate Z_{tor} .	No	$Z_{hyp} < 0$
rake	degrees	Rake angle of fault movement. Reasonable values of rake angle for strike-slip, reverse, and normal faulting events are 0, 90, and -90, respectively.	Yes	--
Dip	degrees	Dip angle of the rupture plane.	No	Dip < 0
W	kilometers	Down-dip width of rupture plane.	No	W < 0

Argument	Units	Description	Required?	How to specify if unknown
z _{tor}	kilometers	Depth to top of rupture.	No	z _{tor} < 0
v ₃₀	meters/ second	Time-averaged shear wave velocity over a subsurface depth of 30 meters.	Yes	--
z _{sed1p0}	meters	Depth to V _s = 1.0 km/s (Z _{1,0}).	No	z _{sed1p0} < 0
z _{sed2p5}	meters	Depth to V _s = 2.5 km/s (Z _{2,5}).	No	z _{sed2p5} < 0
as	–	Aftershock flag; equal to 1 for aftershocks and 0 for mainshocks. For BA08 and CB08, only as = 0 is acceptable input, because these models are not designed for implementation on aftershocks. If as = 1 is specified, then null values are returned for the BA08 and CB08 relationships.	Yes	--

¹Required input parameters have "--" in this column.

Additional instructions on the input parameters are available as comments in the control file, but they are replicated here. As seen in table 8, the minimum required input parameters are T, M, either F_{hw} or A_z, R_{jb}, rake, v₃₀, and as. When parameters are unknown, they should be assigned negative values in the control file, with the exception of A_z, which should be specified as |A_z| > 180 if unknown (since A_z may take values from –180 to 180). The BA08 model is the only model that can generate a ground-motion prediction for an unspecified faulting mechanism. To obtain a prediction using BA08 for the case of unspecified faulting, the user should specify the rake angle such that |rake| > 180. In this case, the program will return null values for the other GMPEs.

The user must ensure that F_{hw} and A_z are entered consistently; the program will return an error message if any of the following conditions are met:

- F_{hw} < 0 and |A_z| > 180
- F_{hw} = 0 and 0 ≤ A_z ≤ 180
- F_{hw} = 1 and –180 < A_z < 0.

Unlike the R program, the site coordinate (R_x) is not an independent input parameter in the Fortran program, but it is calculated from R_{jb} and the other geometric rupture parameters as described in the Distance Measures section of this report, and further in Kaklamanos and others (2011). The methodology for calculating unknown input parameters is the same for both the R and Fortran programs.

The parameters in the output file are presented in table 9. Like table 5 for the R output, table 9 is divided into inputs and outputs. In the nomenclature of the output section of this list, Y refers to the ground-motion parameter of interest, which can be:

- Spectral acceleration, Sa (g), also denoted as PSA;
- Peak ground acceleration, PGA (g); or
- Peak ground velocity, PGV (cm/s).

Like the R output, the Fortran output provides the total standard deviations (in natural logarithmic space). In addition to the total standard deviation, the intra- and inter-event components of standard deviation also are presented for the various models. The intra-event standard deviation (σ) and the inter-event standard deviation (τ) are used to calculate the total standard deviation (σ_T) by the formula

$$\sigma_T = \sqrt{\sigma^2 + \tau^2} . \quad (7)$$

Unlike the R output, the Fortran output does not present the upper ($\mu + \varepsilon \sigma_T$) and lower ($\mu - \varepsilon \sigma_T$) ground-motion estimates. However, the user may easily calculate these values by selecting a value of ε , and adding $\pm \varepsilon \sigma_T$ from the median ground-motion estimate in logarithmic space.

Table 9. Terms in Fortran output file.

Name	Units	Description
Input variables		
T	seconds	Spectral period [input]
M	–	Moment magnitude [input]
Az_in	degrees	Source-to-site azimuth [input]
Az_use	degrees	Source-to-site azimuth for use [calculated if Az_in is unknown]
Rjb	kilometers	Joyner-Boore distance [input]
Rrup_in	kilometers	Rupture distance [input]
Rrup_calc	kilometers	Rupture distance [calculated using the physically-derived equations in Kaklamanos and others (2011)]
Rrup_scherbaum	kilometers	Rupture distance [calculated using the empirical equations in Scherbaum and others (2004)]
Rrup_use	kilometers	Rupture distance used in calculations [if Rrup_in is unknown, use Rrup_calc]
Rx_calc	kilometers	Site coordinate [calculated using the equations of Kaklamanos and others (2011)]
Zhyp_in	kilometers	Hypocentral depth [input]
Zhyp_use	kilometers	Hypocentral depth [calculated if Zhyp_in is negative]
Fhw_in	–	Hanging wall flag [input]
Fhw	–	Hanging wall flag used in calculations [determined from Az_in if Fhw_in < 0]
rake	degrees	Rake angle of fault movement [input]
Dip_in	degrees	Fault dip angle [input]
Dip_use	degrees	Fault dip angle [calculated if Dip_in is negative]
W_in	kilometers	Down-dip rupture width [input]
W_use	kilometers	Down-dip rupture width [calculated if W_in is negative]
Ztor_in	kilometers	Depth to top of rupture [input]
Ztor_use	kilometers	Depth to top of rupture [calculated if Ztor_in is negative]
Vs30	meters/second	Time-averaged shear wave velocity over 30 m subsurface depth [input]
Zsd1p0_in	meters	Depth to Vs of 1.0 km/s [input]
Zsd1p0_AS	meters	Depth to Vs of 1.0 km/s [calculated by AS08 equation]
Zsd1p0_CY	meters	Depth to Vs of 1.0 km/s [calculated by CY08 equation]

Name	Units	Description
zsd2p5_in	meters	Depth to Vs of 2.5 km/s [input]
zsd2p5 (km)	kilometers	Depth to Vs of 2.5 km/s [calculated for use in CB08 model if zsd2p5_in < 0, using either zsd1p0_in (if provided) or zsd1p0_AS]. Note that the units are km, and not m.
AS	–	Aftershock flag [input]
Output variables		
<i>AS08 Model:</i>		
Y_as		Median AS08 ground-motion prediction
Sigestv_as		Intra-event AS08 standard deviation, using coefficients s_1 and s_2 for estimated V_{S30} in equation 24 of Abrahamson and Silva (2008)
Sigmeav_as		Intra-event AS08 standard deviation, using coefficients s_1 and s_2 for measured V_{S30} in equation 24 of Abrahamson and Silva (2008)
Tau_as		Inter-event AS08 standard deviation, calculated by equation 25 of Abrahamson and Silva (2008)
SigTest_as		Total AS08 standard deviation, for estimated V_{S30}
SigTmea_as		Total AS08 standard deviation, for measured V_{S30}
<i>BA08 Model:</i>		
Y_ba		Median BA08 ground-motion prediction
Y-bamod		Median BA08 ground-motion prediction using the small-magnitude correction published in Atkinson and Boore (2011)
Sigma_ba		Intra-event BA08 standard deviation; from table 8 in Boore and Atkinson (2008), first column
Tau_U_ba		Inter-event BA08 standard deviation when the fault type is not specified; from table 8 in Boore and Atkinson (2008), second column
Sig_TU_ba		Total BA08 standard deviation, when fault type is unspecified
Tau_M_ba		Inter-event BA08 standard deviation when the fault type is specified; from table 8 in Boore and Atkinson (2008), fourth column
Sig_TM_ba		Total BA08 standard deviation, when fault type is specified
<i>CB08 Model:</i>		
Y_cb		Median CB08 ground-motion prediction
Sigma_cb		Intra-event CB08 standard deviation, calculated from equation 15 in Campbell and Bozorgnia (2008)
Tau_cb		Inter-event CB08 standard deviation, calculated from equation 14 in Campbell and Bozorgnia (2008)
Sig_Arb_cb		Total CB08 standard deviation for the arbitrary horizontal component, calculated from equation 18 of Campbell and Bozorgnia (2008)
SigT_cb		Total CB08 standard deviation for the geometric mean horizontal component

Name	Units	Description
<i>CY08 Model:</i>		
Y_cy		Median CY08 ground-motion prediction
Tau_cy		Inter-event CY08 standard deviation, calculated by equation 19 of Chiou and Youngs (2008)
SigInfr_cy		Inter-event CY08 standard deviation for inferred V_{S30} , calculated by equation 20 of Chiou and Youngs (2008) with $F_{Inferred} = 1$ and $F_{Measured} = 0$
SgTinfr_cy		Total CY08 standard deviation, for estimated V_{S30} ; calculated by equation 21 of Chiou and Youngs (2008)
SigMeas_cy		Inter-event CY08 standard deviation for measured V_{S30} , calculated by equation 20 of Chiou and Youngs (2008) with $F_{Inferred} = 0$ and $F_{Measured} = 1$
SgTmeas_cy		Total CY08 standard deviation, for measured V_{S30} ; calculated by equation 21 of Chiou and Youngs (2008)

Interpolation of Spectral Values for Periods not Corresponding to the Tabulated Coefficients

The program uses straight-line interpolation of $\ln Sa$ versus $\ln T$. Here are two graphs showing a comparison of Sa and sigma values from an older version of `nga_gm_tmr` (for which the periods were required to be the tabulated periods) and given in the distributed version of the program. Figures 5 and 6 confirm that the interpolation is being done correctly.

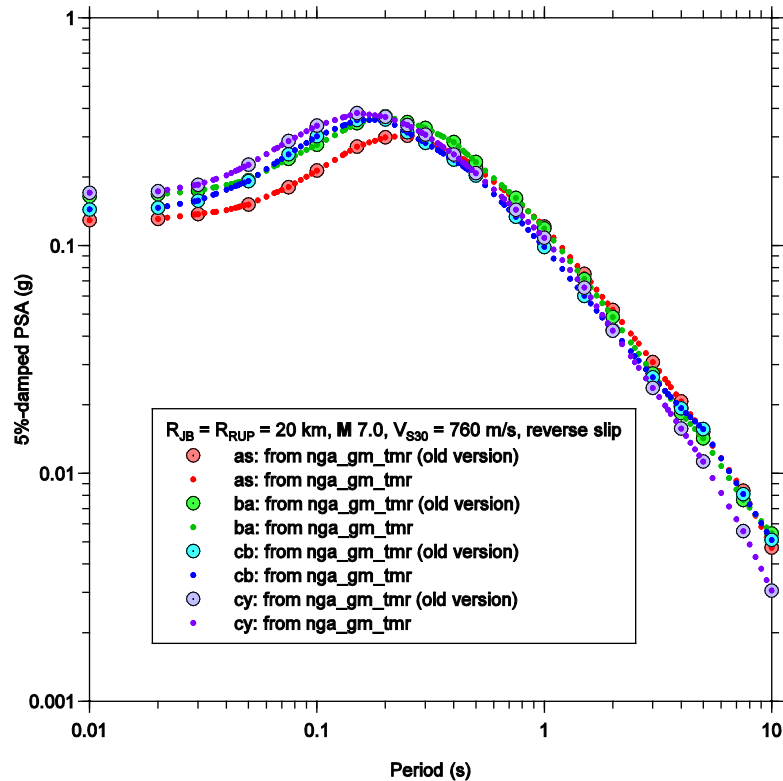


Figure 5. Example interpolation of spectral values for periods not corresponding to the tabulated coefficients; comparison of median estimates of Sa .

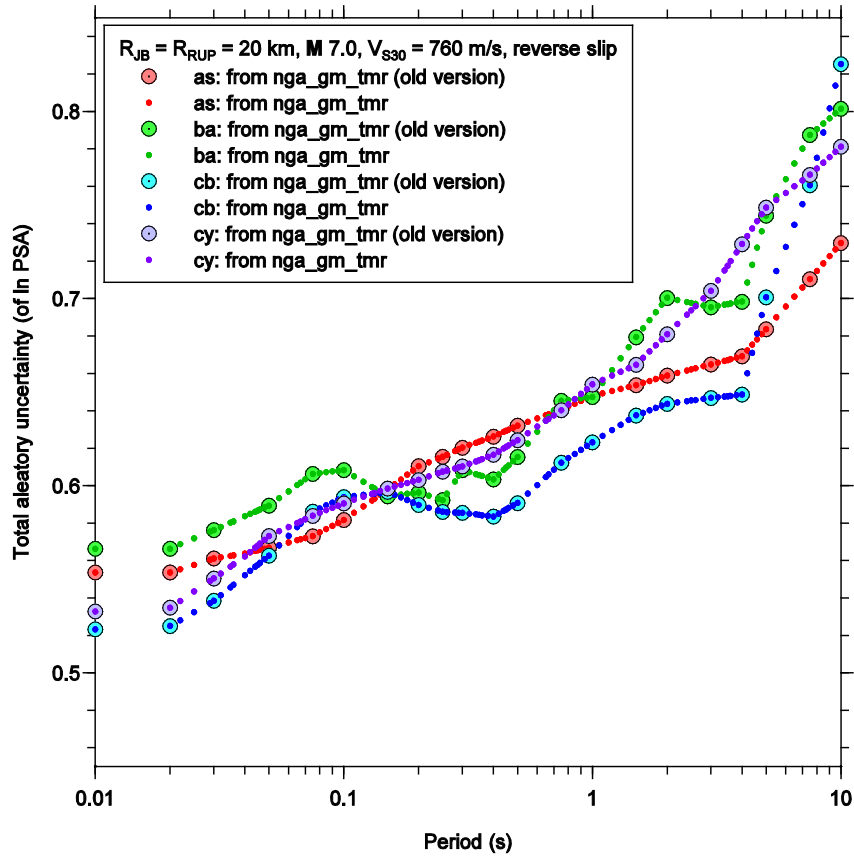


Figure 6. Example interpolation of spectral values for periods not corresponding to the tabulated coefficients; comparison of aleatory uncertainties.

Examples of Input and Output

Here, we present two examples having input available in the default control file `nga_gm_tmr.ct1` (scroll down below the word “stop” to view the input parameters for these examples). Two cases are considered for a magnitude 7.0 earthquake: a site on the hanging wall (over a dipping reverse fault) and a site on the foot wall at a distance corresponding to the distance from the surface projection of the fault to the site over the hanging wall. The fault has a rake of 90° and a hypocentral depth of 5 km. Because fault dip was not specified, and the rake angle is characteristic of reverse faulting, a generic dip of 40° was assigned by the program. In addition, the fault width was not specified but was assigned using the Wells and Coppersmith (1994) empirical relations. The vertical distance from the surface to the top of the rupture surface (Z_{TOR}) was not specified, but was calculated using the fault width and the hypocentral depth. With the fault type and geometry determined, the next task is to specify the site location at which the GMPEs will be evaluated. The hanging wall case was specified first, with $R_{JB} = 0.0$ km but without specifying R_{RUP} . An azimuth of 90° was specified (along a line normal to the strike of the fault). In this case the program assumes that the site is in the middle of the surface projection of the fault and calculates R_{RUP} and R_X . See the sample program output file for the values assigned by the program. Plots of spectral acceleration and total standard deviation are given in figures 7 and 8, respectively.

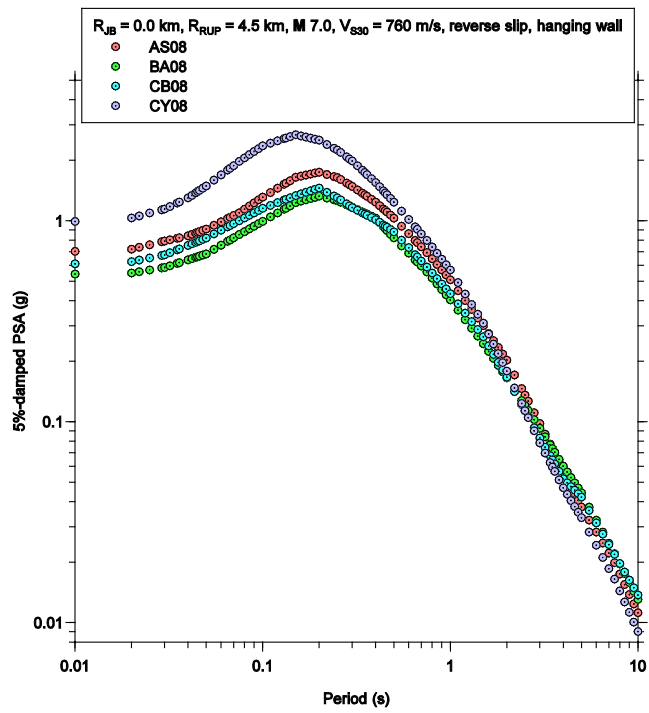


Figure 7. Median ground-motion estimates for the hanging wall case.

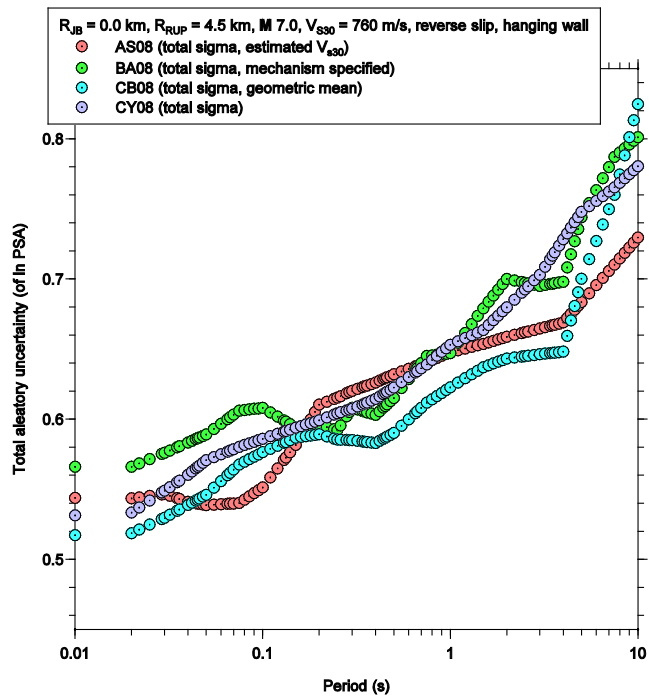


Figure 8. Total standard deviations for the hanging wall case.

Once the hanging wall case was run, the footwall case was run, using R_{JB} from the hanging wall case. Plots of Sa and the total standard deviations for the footwall case are given in figures 9 and 10, respectively.

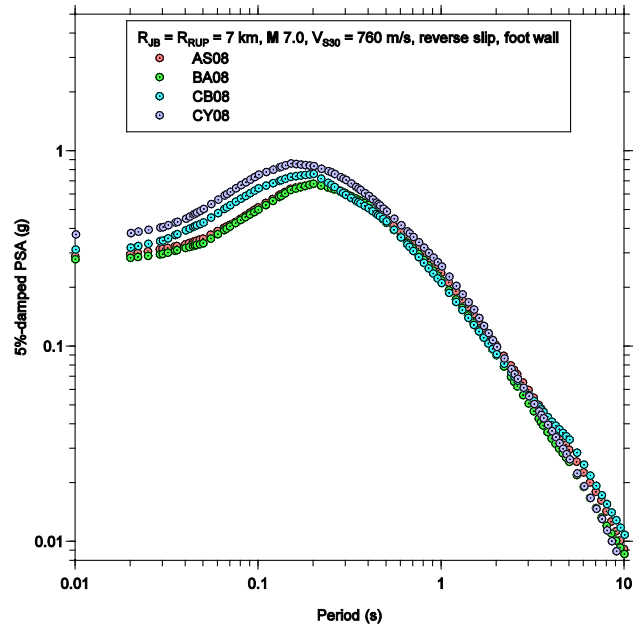


Figure 9. Median ground-motion estimates for the footwall case.

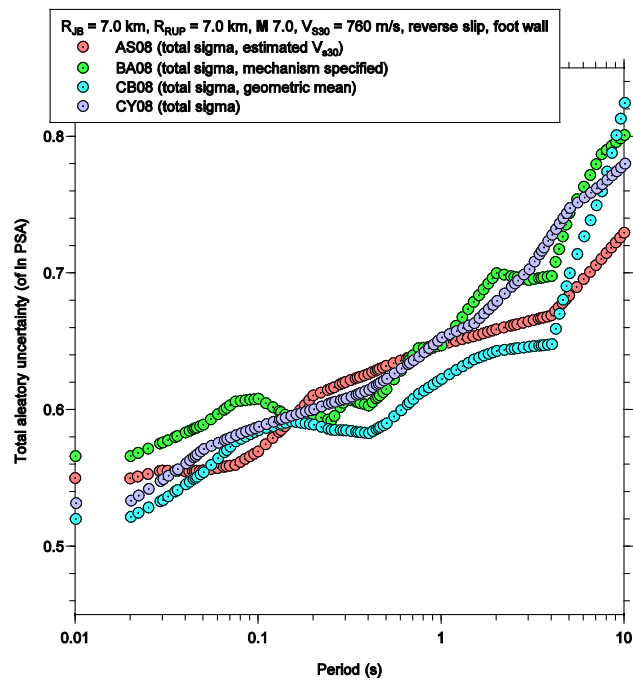


Figure 10. Total standard deviations for the footwall case.

Programs to Compute Fault-to-Station Distances

Some companion programs to `nga_gm_tmr`, available at David Boore's Web page (<http://www.daveboore.com/> under "Online Software", last accessed 11 January 2011), are useful for determining the appropriate distances and azimuths for input to `nga_gm_tmr`. The file "`dist_programs.zip`" contains programs for computing various distance measures between a station and a finite fault, programs to compute the distance between two points specified by their latitudes and longitudes, as well as other useful programs, such as those for converting between kilometers and degrees (useful when the vertices of a fault's surface projection are in terms of kilometers on a Cartesian grid, rather than latitude and longitude, and vice versa). These and other programs are also documented in Boore (2010).

The program `dist_3d`, which computes various distance measures from a point on the Earth's surface to a rectangle with arbitrary orientation and location in space, may be used to obtain the distance from a station to a finite fault. Initially, the program did not compute the azimuths needed for `nga_gm_tmr` if the fault was nonvertical. To increase its compatibility with the `nga_gm_tmr` program, the `dist_3d` program has been modified to calculate the azimuth (called "`az_jb`") in these cases. The program computes distances and azimuths for single- and multi-segment faults. For multisegment faults, the value of `az_jb` is computed for each individual fault segment, not for a single-fault representation of a multisegment fault. The output from `dist_3d` may be used as input to `nga_gm_tmr`.

Selecting Between the R and Fortran Implementations

We have independently verified the R and Fortran implementations using a validation set of 128,000 calculations designed to test all reasonable ranges of the input parameters. This extensive dataset consists of various combinations of magnitude, distance, site conditions, hanging wall flag, style of faulting, and spectral period. The implementations in R and Fortran generate identical results, consistent with the formulation of the NGA models. There are occasional roundoff differences between the Fortran and R output, but these roundoff differences are very small and rare.

The selection between the R and Fortran implementations is largely a matter of personal preference. The key advantage of Fortran over R is its computational speed. If the user desires to perform a large number of calculations (on the order of tens of thousands), and if computational speed is a concern, then Fortran is advantageous over R. For small numbers of calculations, the difference between Fortran and R is not noticeable. The Fortran program automatically reads the data from the control file, performs the calculations, and writes an output file. In the R terminal, the user must enter commands to read the input data, perform the calculations, and write the output. Many R users type their commands directly into a text editor, and copy and paste them into the R command line. Such text files can act as a useful resource for quickly reproducing results at a later time, or modifying existing code for future similar calculations. Although the examples in this report (and in the HTML help documentation) provide a framework for this type of analysis, a user new to R may find this inconvenient.

On the other hand, the input to the R program is more flexible than that of the Fortran program, and the user has access to the internal functions used in the ground-motion calculations (which is not possible in Fortran). The intermediate functions used in the estimation of unknown input parameters and the subfunctions used in each of the models are readily available. Also, for users wishing to perform quick ground-motion calculations, they may simply enter the function calls into the R terminal, without having to go through the process of developing a control file. In situations where a greater degree of precision is required, the user may prefer output from R over the `.out` file of the Fortran program. Also,

it is much easier to quickly visualize the results in R; plotting in Fortran requires other software, whereas plotting can be performed in tandem with the calculations in R. However, we do not wish to promote one type of implementation over another, and we hope that the scientific and engineering communities find both of these programs useful.

Acknowledgments

We thank John Douglas and Erol Kalkan for providing helpful reviews that improved the quality of this report. In addition, we would like to acknowledge the researchers involved in the NGA project and their work that went into the creation of the NGA flatfile, the NGA relations, and the supporting NGA projects and documentation; we commend them for their efforts.

References Cited

- Abrahamson, N.A., and Silva, W.J., 2008, Summary of the Abrahamson & Silva NGA ground-motion relations: *Earthquake Spectra*, vol. 24, no. 1, p. 67–97.
- Abrahamson, N.A., and Silva, W.J., 2009, Errata for “Summary of the Abrahamson & Silva NGA ground-motion relations” by Abrahamson, N.A. and W.J. Silva (2008): Pacific Earthquake Engineering Research Center, University of California, Berkeley, http://peer.berkeley.edu/products/nga_project.html, last accessed 11 January 2011, 3 p.
- Atkinson, G.M., and Boore, D.M., 2011, Modifications to existing ground-motion prediction equations in light of new data: *Bulletin of the Seismological Society of America*, in press.
- Boore, D.M., 2010, TSPP—A collection of FORTRAN programs for processing and manipulating time series: U.S. Geological Survey Open-File Report 2008-1111, 56 p.
- Boore, D.M., and Atkinson, G.M., 2008, Ground-motion prediction equations for the average horizontal component of PGA, PGV, and 5%-damped PSA at spectral periods between 0.01 s and 10.0 s: *Earthquake Spectra*, vol. 24, no. 1, p. 99–138.
- Campbell, K.W., and Bozorgnia, Y., 2007, Campbell-Bozorgnia NGA ground motion relations for the geometric mean horizontal component of peak and spectral ground motion parameters: PEER Report No. 2007/02, Pacific Earthquake Engineering Research Center, University of California, Berkeley, 246 p.
- Campbell, K.W., and Bozorgnia, Y., 2008, NGA ground motion model for the geometric mean horizontal component of PGA, PGV, PGD and 5% damped linear elastic response spectra for periods ranging from 0.01 to 10 s: *Earthquake Spectra*, vol. 24, no. 1, p. 139–171.
- Chiou, B.S.-J., and Youngs, R.R., 2008a, An NGA model for the average horizontal component of peak ground motion and response spectra: *Earthquake Spectra*, vol. 24, no. 1, p. 173–215.
- Chiou, B.S.-J., and Youngs, R.R., 2008b, NGA model for the average horizontal component of peak ground motion and response spectra: PEER Report No. 2008/09, Pacific Earthquake Engineering Research Center, University of California, Berkeley, 293 p.
- Idriss, I.M., 2008, An NGA empirical model for estimating the horizontal spectral values generated by shallow crustal earthquakes: *Earthquake Spectra*, vol. 24, no. 1, p. 217–242.
- Kaklamanos, J., and Baise, L.G., 2011, Model validations and comparisons of the Next Generation Attenuation of Ground Motions (NGA-West) project: *Bulletin of the Seismological Society of America*, vol. 101, no. 1, p. 160–175.
- Kaklamanos, J., Baise, L.G., and Boore, D.M., 2011, A framework for estimating unknown input parameters when implementing the NGA ground motion prediction equations in engineering practice: *Earthquake Spectra*, vol. 27, no. 4, p. 1219–1235.

- Kaklamanos, J., and Thompson, E.M., 2010, nga: NGA earthquake ground motion prediction equations, R package version 1.4, <http://CRAN.R-project.org/package=nga>, last accessed 1 December 2011.
- Mai, P.M., Spudich, P., and Boatwright, J., 2005, Hypocenter locations in finite-source rupture models: *Bulletin of the Seismological Society of America*, vol. 95, no. 3, p. 965–980.
- Petersen, M.D., Frankel, A.D., Harmsen, S.C., Mueller, C.S., Haller, K.M., Wheeler, R.L., Wesson, R.L., Zeng, Y., Boyd, O.S., Perkins, D.M., Luco, N., Field, E.H., Wills, C.J., and Rukstales, K.S., 2008, Documentation of the 2008 update of the United States national seismic hazard maps: U.S. Geological Survey Open-File Report 2008-1128, 61 p.
- R Development Core Team, 2010, R: a language and environment for statistical computing, R Foundation for Statistical Computing, Vienna, Austria, ISBN 3-900051-07-0, <http://www.R-project.org>, last accessed 11 January 2011.
- Ripley, B., and from 1999 to Oct. 2002 Lapsley, M., 2009, RODBC: ODBC database access, R package version 1.3-1., <http://CRAN.R-project.org/package=RODBC>, last accessed 11 January 2011.
- Scherbaum, F., Schmedes, J., and Cotton, F., 2004, On the conversion of source-to-site distance measures for extended earthquake source models: *Bulletin of the Seismological Society of America*, vol. 94, no. 3, p. 1053–1069.
- Thompson, E.M., 2007, Surface-source downhole seismic analysis in R: U.S. Geological Survey Open-File Report 2007-1124, 17 p.
- Wells, D.L., and Coppersmith, K.J., 1994, New empirical relationships among magnitude, rupture length, rupture width, rupture area, and surface displacement: *Bulletin of the Seismological Society of America*, vol. 94, no. 4, p. 974–1002.