NSPI User's Guide

Brian J. Williams, Thomas J. Santner and Lori Lyn Price

November 2001

1 Introduction

NSPI is a FORTRAN 77 program that computes simultaneous, nonparametric prediction bands for a new curve drawn from the same population as a given set of training curves. NSPI has been compiled and used on HP and SUN workstations as well as PC platforms using a variety of operating systems and FORTRAN compilers. Modifications of the source code may be needed for other operating systems.

This program assumes the training curves have the *same* number of points but are otherwise arbitrary. NSPI uses a batch file, denoted `input.job` below, to input directions to the program. NSPI executes the directions specified in `input.job`, then writes the output information to the files specified in `input.job`.

1.1 Statistical Method

NSPI replaces *ceach each curve* by a set of (curve-specific) Fourier *coefficients* that represent the curve. If θ, $0 \leq \theta \leq 100$, represents the percentage of the time between one foot-strike and the next, then each curve is represented by a set coeficients $(\alpha_0, \alpha_1, \ldots, \alpha_L, \beta_1, \ldots, \beta_L)$ where

$$f(\theta) = \alpha_0 + \sum_{\ell=1}^{L} \left(\alpha_\ell \cos\left(\frac{2\ell}{100}\pi\theta\right) + \beta_\ell \sin\left(\frac{2\ell}{100}\pi\theta)\right)\right) \tag{1}$$

and these coefficients are calculated ordinary least squares (OLS). Here L is the maximum harmonic and α_0 is the mean response across all the phases of the cycle; the input variable `NumFourierTerms` $= 1 + 2L$.

In practice, the process of estimating the Fourier coefficients is slightly more difficult than described above because NSPI can be used with *arbitrary* curves while Fourier series representations of data require that the starting and ending values of the curve be the same. The fitting process is performed on an *augmented* version of the curve that satisfies this starting-ending restriction rather than on the original curve. For each curve, NSPI adds points at both extremes of the data.

If the i^{th} curve consists of points (θ_{ij}, C_{ij}) for $j = i, \ldots, J$, then a set of addtional θ_{ij} points are constructed to the left of θ_{i1} and to the right of θ_{iJ}. The program input `NumPointsPerCurve` $= J$ and the input `NumPointsExtension` is the number of points added on each extreme of the data. To the left end of the original curve, a polynomial is constructed that is "smooth" in the sense that it has the same first derivative at θ_{i1} as does the quadratic thourgh the three points $\{(\theta_{i1}, C_{i1}), (\theta_{i2}, C_{i2}), (\theta_{i3}, C_{i3})\}$ and the polynomial goes through $(C_{i1} + C_{iJ})/2$ at the left most extreme augmented θ_{ij}. The C_{ij} for each augmented θ_{ij} are the points on this polynomial. A similar "smooth" set of points is added on the right of the original data. The resulting OLS coefficients are used to represent the curve.

A bootstrap procedure is applied to the sets of curve-specific coefficients to determine the upper α critical point of the maximum deviation over the standardized difference between a randomly drawn curve from the bootstrap distribution and the estimated mean based on all the curves [Sutherland, Olshen, Biden and Wyatt (1988), Lenhoff, Santner, Otis, Peterson, Williams and Backus (1999), Efron and Tibshirani (1993)]. The simultaneous confidence bands are constructed using this critical point, the estimated mean curve, and the estimated standard deviation of the curve.

2 Overview of Program Execution

Let `nspi.run` denote the executable code obtained by compiling the NSPI source code `nspi.f`. NSPI reads program specifications from a batch file called `input.job` below. The program does some error checking and in case of an error in the batch file, error messages are printed to the terminal screen listing the error, the program will stop, and the user must correct the batch file. The syntax for running NSPI using the commands in `input.job` is

```
unix> nspi.run input.job
```

at the UNIX command prompt.

3 Batch File Description

NSPI batch files contain the information for single *or* multiple jobs. The input parameters include individual values and file names. The following rules govern the basic syntax of the batch file.

- All input description names are *case-insensitive* and spaces (except for a space in the first column) have no effect on the processing of the batch file. The only situations where case is important are the names of the input and output files that the user enters. The UNIX operating system is case-sensitive.

- The maximum length of the lines in the batch file is 64 characters.

- The parameters for each job can be entered in the batch file in any order.

- The user can place data files in separate subdirectories and simply enter the path to the file along with the file name (see example below).

- Comment lines can be used in the batch file by preceeding the command line by the symbol `#` or by a space in the first column.

The program was written in FORTRAN 77 which requires that maximum values for certain work parameters be set prior to compiling. The values of these parameters can be easily modified in the source code and the program recompiled. The default values for all such quantities in the distributed code are listed in Section 5.

3.1 Batch File Input

Unless default values are used, the required items in this file must be included in the batch file. If any are missing, the program will stop executing and report the missing parameters to the user.

- `RunName = Name` where *Name* is a name that the user assigns to this particular job. This command is *optional* but aids in organizing output messages when running multiple jobs in a batch file.

- `NumCurves = Integer` where `Integer` is the number of curves to be analyzed. *Required Input*

- `NumPointsPerCurve = Integer` where `Integer` is the number of points per curve in the input. The number of points per curve must be the same for each individual. *Required Input*

- `NumFourierTerms = Integer` where `Integer` is the number of Fourier terms used to fit each curve. The diagnostics in the diagnostic output file, along with a plot of the curves and the prediction curves, can be used to determine the appropriate number of terms. *Required Input*

- `NumPointsExtension = Integer` where `Integer` is the number of points for *each side* of curve extension. **Warning:** $2\times$ `NumPointsExtension` must be less than or equal to the value of `maxtotaddpts` which is set in the PARAMETER statement of the FORTRAN code. *Required Input*

- `NumBootstrapReps = Integer` where `Integer` is the number of bootstrap draws. *Required Input*

- `CovProb = FloatingPoint` where `FloatingPoint` is the coverage probability for the prediction intervals. This is a value of the range (0,1). *Required Input*

- `Seed = Integer` where `Integer` is a non-zero integer used for random number generation. Integer must be a value -32765,-32764,..,-1,1,32765. *Required Input*

- `DataInputMethod` $=$ `Integer` where `Integer` is equal to 1 or 2. This signifies whether the program utilizes one input file or multiple input files in different subdirectories. *Required Input* with a **default** of 1.
  - If `DataInputMethod` $= 1$, then the data is in a single input file that should contain the curves to be analyzed in column format (i.e. columns represent curves).
  - If `DataInputMethod` $= 2$, then there is a separate file for each curve. The data input files must contain four columns. The first is the X coordinate, and the following three will be the X, Y, and Z dimensions of the curve. Each of these different input files are contained in a different subdirectory defined by `SubDirectoryFile`. Each input file in these subdirectories must have the same name.

- `DataInputFile = FileName` where `FileName` is the name of the file that contains the input data for the program. There are two formats that this file can have. The format that is used is dependent on the `DataInputMethod` stated in the batch file. The **default** is 1. *Required Input*

- `OmitNumOfLines = Integer` where `Integer` is the number of lines to omit from the top of the input files if `DataInputMethod` is equal to 2. This allows that headers may be located at the top of the input files when utilizing the multiple input files method. The **default** is 0, and therefore does not need to be present in the batch file if there are no lines to omit. In addition, this parameter is *not used* if the `DataInputMethod` is equal to 1. This item is *Required Input* if `DataInputMethod`$= 2$

- `ColumnToAnalyze = Integer` where `Integer` is equal to 2, 3, or 4. This the the column number that will be used to perform the calculations for this program in the input files when `DataInputMethod` is equal to 2. These numbers correspond to the X, Y, and Z dimensions. *Required Input*

- `SubDirectoryFile = FileName` where `FileName` is the name of the file that contains the subdirectory names where the input files are located when `DataInputMethod` equals 2. This file should contain one subdirectory name per line. *Required Input* if `DataInputMethod=`2

- `ConfLimitOutputFile = FileName` where `FileName` is the name of the file where the program will output the confidence bands and estimated mean curve. *Required Input*

- `DiagnosticOutputFile = FileName` where `FileName` is the name of the file that will contain summary and diagnostic output data. *Required Input*

- `Stop` signifies the end of one job. This command must follow every job to separate it from the job after it. *Required Input*

- `End` signifies the end of the *batch file*. The program will stop reading the batch file and complete execution. Nothing written after this command will be read by the program. *Required Input*

3.2 Batch File Output

Unless otherwise stated, the following items are *required* to be included in the batch file. If any are missing, the program will stop executing and report the missing parameters to the user.

- `ConfLimitOutputFile`: The first column lists the points at which the mean curve and the confidence limits were evaluated. The last three columns give the values of the lower confidence limit, mean curve, and upper confidence limit, respectively. *Required Input*

- `DiagnosticOutputFile`: This output file contains the maximum absolute deviation between the observed and fitted response for each curve, the maximum overall absolute deviation over all curves and points for each curve, the maximum relative deviation, and several other pieces of input information. *Required Input*

  The maximum absolute deviation for each curve allows the user to see if there are one or a small number of curves that are ill-fit by the Fourier series expansion while all other curves are well fit. The latter situation would suggest examination of the poorly fit curve to make sure it is representative of the population of curves that is desired to be studied; if not either an external cause for this discrepancy can be sought or the curve possibly removed.

  The *maximum relative deviation* is calculated by dividing the maximum absolute deviation by the range of the largest and smallest data points among all the curves. We suggest as a rule of thumb that the maximum relative deviation be no larger then 0.05. In this case, the curves are well approximated by the Fourier terms. Using a model with more Fourier coefficients and/or adding more extension points will lead to a smaller maximum relative deviation.

4 Examples

4.1 Sample Batch File Using a Single Data File

```
RunName = Example1
NumCurves = 11
NumPointsPerCurve = 128
NumFourierTerms = 21
NumPointsExtension = 25
NumBootstrapReps = 400
CovProb = .90
Seed = 18853
DataInputFile = test/in.dat
ConfLimitOutputFile = Couttest.1
DiagnosticOutputFile = Douttest.1

stop
end
```

Note that by default this program uses `DataInputMethod`$=1$, i.e. each column contains the data for one curve. All curves are in the same file. In this batch file `Example1` is the job name, there are 11 curves with 128 points each, 21 Fourier terms are used to represent each curve based on the following model.

$$f(\theta) = \alpha_0 + \sum_{j=1}^{10}\left(\alpha_j\cos\left(\frac{2j}{100}\pi\theta\right) + \beta_j\sin\left(\frac{2j}{100}\pi\theta)\right)\right)$$

The program generates 90% confidence bands based on the input data using 400 bootstrap replications. The input file is in the subdirectory `test` and is called `in.dat`. The first few lines of `in.dat` corresponding to 6 curves are listed below. Each column contains the data for one curve.

```
0.19593 -0.04342 0.26366 -0.05593  0.12076  0.11118
0.20544 -0.0126  0.28319 -0.03694  0.14749  0.14176
0.21494  0.01894 0.30271 -0.01764  0.17536  0.17245
0.22341  0.05306 0.3229   0.00423  0.20819  0.20078
0.23115  0.08801 0.34333  0.02696  0.24264  0.22685
0.23827  0.12419 0.3629   0.0501   0.27886  0.25116
0.24468  0.15851 0.3818   0.07355  0.31356  0.27389
```

NSPI is asked to generate two output data files. `Couttest.1` contains the data to form the confidence bands. The first column shows the points at which the mean curve and prediction limits were evaluated. The second column contains the values of the lower confidence limit, the third column contains the values of the mean curve and the final column lists the values of the upper prediction limit. The following output shows the first few lines of `Couttest.1` for the example above.

```
          0.00000          -0.09810            0.14887            0.39584
          0.26042          -0.08128            0.16828            0.41785
          0.52083          -0.06635            0.18637            0.43909
          0.78125          -0.05328            0.20306            0.45941
          1.04167          -0.04207            0.21828            0.47864
          1.30208          -0.03270            0.23198            0.49666
          1.56250          -0.02515            0.24409            0.51332
          1.82292          -0.01938            0.25458            0.52854
```

The second output data file produced by NSPI is called `Douttest.1` and contains diagnostics. The first few lines give details of the analysis, then the maximum absolute deviation between the fitted response for each curve, then the maximum overall absolute deviations and finally the maximum relative deviation. The following shows a selection of lines from `Douttest.1`. Most curves have maximum absolute deviations over 0.15 so we should consider adding more Fourier terms to better represent the curves.

```
Input File: in.dat
Confidence Limit Output File: Couttest.1
Seed:   18853

Simultaneous Prediction Intervals
Coverage Probability is          90.0000 %
Number of Bootstrap Reps =        400

Number of Fourier Terms =         11
Number of Curve Extension Points =         25

Maximum absolute deviation between observed and
fitted response for each curve:

    1            0.17260
    2            0.21792
    3            0.17701
    4            0.16225
    5            0.23357
    6            0.27730
           .
           .
           .
   23            0.17250
   24            0.21677
   25            0.21114
   26            0.18776
   27            0.23306
   28            0.25729

Maximum absolute deviation over curves =          0.27730
Maximum relative deviation over curves =          0.19054
```

4.2 Sample Batch File Using Multiple Data Files

```
RunName = Example2
NumCurves = 4
NumPointsPerCurve = 128
NumFourierTerms = 21
NumPointsExtension = 25
NumBootstrapReps = 400
CovProb = .90
Seed = 18853
DataInputMethod = 2
OmitNumOfLines = 2
ColumnToAnalyze = 3
SubDirectoryFile = Directories.txt
DataInputFile = data.dat
ConfLimitOutputFile = Couttest.2
DiagnosticOutputFile = Douttest.2

stop

end
```

Note this program uses the second method of data input. The data for each curve is in a separate file. Each data input file contains four columns as described in `DataInputMethod` in Section 2. Other than the input files, all else remains the same as in the previous example. The output files are the same as in Section 4.1.

5 Compilation Hints

A typical set of commands used to create compiled code is as follows:

```
unix>f77 -o nspi.run nspi.f
```

assuming the compiled code is named `nspi.run`. Here `f77` is the command to invoke the FORTRAN77 compiler.

We reiterate that the user can vary the *maximum* size of the problems that the compiled code can process by changing the values in the PARAMETER statement of the FORTRAN code prior to compiling. Of course, increasing the maximum problem size increases the memory required to run the program. These statements are located at the beginning of the code. The current defaults are:

Maximum Parameter Values of Distributed Code

- `maxnrespin = 200`
- `maxsubj = 70`
- `maxfourtrm = 50`
- `maxboot = 1000`
- `maxtotaddpts = 55`

6 Disclaimer

The source code for NSPI is made available in good faith. It has been tested on a number of different platforms and compilers to assure reproducibility of results. However, none of the authors or distributors warrants its accuracy nor can be held accountable for the consequences of its use.

References

Efron, B. and Tibshirani, R. J. (1993). *An Introduction to the Bootstrap (ISBN 0412042312)*. Chapman & Hall.

Lenhoff, M., Santner, T. J., Otis, J. O., Peterson, M., Williams, B. J. and Backus, S. (1999). Bootstrap prediction and confidence bands: a superior statistical method for analysis of gait data. *Gait and Posture* **9**, 10–17.

Sutherland, D. H., Olshen, R. A., Biden, E. N. and Wyatt, M. P. (1988). *The Development of Mature Walking*. MacKeith Press, London.