Autoloader Reference Manual

LUSAS Version 15.1 : Issue 1

LUSAS
Forge House, 66 High Street, Kingston upon Thames,
Surrey, KT1 1HN, United Kingdom

Tel: +44 (0)20 8541 1999
Fax +44 (0)20 8549 9399
Email: info@lusas.com
http://www.lusas.com

Distributors Worldwide

Table of Contents

Chapter 1 Introduction

Vehicle Load Optimisation

Vehicle load optimisation (VLO) makes use of influence surfaces and influence analysis to identify the most onerous vehicle loading patterns on bridges for a chosen design code and to apply these loading patterns to LUSAS models. It reduces the amount of time spent generating loadcases to replicate traffic and lane loading on models and leads to more efficient and economic design, assessment or load rating of bridge structures. It can be applied to grillage, line beam and plate/shell models.

For information regarding the general use of the Vehicle Load Optimisation facility within LUSAS please see the online help.

Design codes supported

A vehicle load optimisation wizard, available within LUSAS, provides the means of defining parameters, for a particular design code, to generate the most critical traffic loading pattern for each influence shape under consideration. Note that the actual vehicle load optimisation software that is used to generate this loading depends upon the design code chosen.

LUSAS Traffic Load Optimisation

LUSAS Traffic Load Optimisation software is used for:

- **Australia AS5100-2: 2004**
- **Canada CAN/CSA-S6-06 (Design)**
- **Eurocode EN-1991-2,**
- **New Zealand (Transit New Zealand Bridge Manual)**
- **United States of America AASHTO LRFD 7th (and 6th) Edition**

Autoloader Traffic Load Optimisation

Autoloader is a bridge loading optimisation module for use with LUSAS. Autoloader was developed to interpret and implement the following design loading standards:

- **Australia** - the AUSTROADS standard
- **China** – JTG D60-2004 and Hong Kong extensions to the BD 37/88 standard
- **Finland -** the Finland Highway Loading standard
- **India -** IRC6:2000
- **Malaysia** - the JKR Highway Loading standard
- **Norway-** the Norway Highway Loading standard
- **RAIL** - an Autoloader specific standard, for use with actual train loading
- **South Africa** – TMH7
- **South Korea**
- **Sweden** – BRO standard
- **UK** - Highways Agency Departmental Standard BD 21/97
- **UK** - Highways Agency Departmental Standard BD 21/01
- **UK** - Highways Agency Departmental Standard BD 37/88
- **UK** - Highways Agency Departmental Standard BD 37/01
- **USA** - the AASHTO LFD standard

To achieve the above, Autoloader uses the influence surface technique. In general an influence surface for a particular detail on a structure is generated and submitted to Autoloader. Autoloader interprets the influence surface and produces critical loading patterns according to the specified standard. Autoloader generates loading command files which may be read into LUSAS enabling moments, shears, displacements, etc. to be calculated. For a more detailed explanation of influence surface theory, see Appendix A of this manual, “Influence Surfaces and how to use them”.

BD 37/88 applies highway loading in two basic forms:

- **Normal (HA) loading**;
- **Abnormal (HB) loading**.

Normal (HA) loading comprises a Uniformly Distributed Load (UDL) and a Knife Edge Load (KEL). Abnormal loading comprises numerous point loads which represent the wheels of an abnormal vehicle. The names may change in other standards, but the concepts remain the same.

Autoloader has several vehicles pre-defined, including the HB vehicles from BD 37/88, SV from JKR, HLP320 and HLP400 from AUSTROADS, HS15, HS20 from AASHTO, and you have the ability to define others.

Autoloader performs calculations associated with determining the most adverse pattern of live loading of a structure ***for a particular detail***. Autoloader is customisable allowing you to:

- define the international loading standard to be applied;
- define the number and shape (curved or straight) of the carriageways;
- decide which side of the influence surface to look at, i.e. positive, negative, or both;
- run Autoloader in batch mode, i.e. to run Autoloader for a number of details and collate the results;
- optionally suppress UDL, KEL and abnormal vehicle loading;
- specify a list of abnormal vehicles for investigation, Autoloader will select the vehicle which creates the worst effect;
- customise/define new abnormal vehicles;
- customise/alter the UDL loading curve and the magnitude of the KEL;
- customise/alter the width of notional lanes;
- customise/alter the Beta lane factors used;
- customise/alter the Gamma load factors used;
- substitute a vehicle instead of UDL + KEL loading, for example: T44 instead of L44 in AUSTROADS;
- customise/alter the K factor curve for use within BD 21/97;
- customise/alter the Adjustment Factor curve for use within BD 21/97
- customise/alter the input and output units.

Load type naming

Autoloader implements several international design loading codes. These codes use different names to describe the similar concepts. For instance, BD 37/88 denotes UDL + KEL effects as HA loading, whereas AUSTROADS refers to this as L44 loading, whilst JKR refers to the same as LTAL loading. Similarly, abnormal vehicles in BD 37/88 are referred to as HB vehicles, in JKR as SV vehicles and AUSTROADS as HLP vehicles. Autoloader, along with this manual, attempts to cope with these name changes. Generally, these names are interchangeable, and options such as USE_HB have synonyms such as USE_SV and USE_HLP.

Chapter 2 Using Autoloader

Autoloader Considerations

Autoloader helps to speed up the design or assessment of the structure by reducing the time consuming process involved with implementing bridge loading design standards. It enables more details and configuration to be tested within the same time period, thus ensuring greater levels of confidence in the overall design.

However, Autoloader is only a tool, and should not be used in isolation, therefore the output should be verified to ensure correctness. Autoloader has a number of limitations, and makes a number of assumptions about the influence surface in use.

If the mesh of finite elements is too coarse the resulting quality of the influence shape will be poor resulting in excessive cusping/distortion. Autoloader can produce very different results from LUSAS. This is because Autoloader uses straight line interpolation between deck elements. A finer mesh, or placing elements at critical positions can limit this problem. Generation of the worst loading combinations for shear influence surfaces are particularly prone to this because large discontinuities are involved.

Similarly, if any of the step values defined are too large the Autoloader results can be very different from those generated by LUSAS. The step values are the distances which Autoloader uses to move the vehicle and calculate the values on the influence surface. There is a trade-off between time and accuracy. As the step values decrease in size, so Autoloader becomes more accurate. As the step values increase in size, so Autoloader becomes faster.

Autoloader gives slightly different results from LUSAS. This is because LUSAS and Autoloader estimate values between nodes in slightly different ways. Also the method adopted for generating the influence shape for shear may induce shear deformations locally in the split structure which will not necessarily occur in the complete structure.

Generally, to improve the accuracy of Autoloader, consider increasing the number of elements used and/or decreasing the increment values used.

For BD 37/88, where the carriageway is less than 5m wide, the standard is unclear. With HA loading only, the carriageway has one notional lane with a width of 2.5m. Autoloader positions the notional lane in such a way as to produce the worst effect. The loading on the remainder of the carriageway width is in accordance with clause 6.4.1.1 of BD 37/88, for example 5kN/m2. For abnormal vehicle loading, Autoloader takes the notional lane width as 3.5m (the width of the abnormal vehicle). It loads the remainder of the carriageway with 5kN/m2.

Autoloader only works with continuous lanes. Where gaps are used for example, to model hammer heads in a bridge deck, these should be closed for the Autoloader analysis with dummy members. Then these members should be removed for the final structural analysis.

Vehicle Load Optimisation (Autoloader)

In certain loading situations an upward load may be applied by Autoloader. This can be a result of a HB loading overhanging a lane where HA loading is applied. The upward loads is applied to cancel the excess of HA loading. The loading standard BD 37/88 "Loads for Highway Bridges", clause 6.4 deals with this loading situation in more detail.

Using Grillages with Vehicle Load Optimisation (Autoloader)

When using a grillage with the Vehicle Load Optimisation facility each of the lines representing a section of the deck must be meshed with one element only i.e. each bay of the grillage must have one element assigned. In addition, only lower order elements can be used. Quadratic elements with mid-side nodes are not permitted.

Using Autoloader with LUSAS

This section describes the steps required to use Autoloader with LUSAS. A Vehicle Load Optimisation wizard is also provided in LUSAS to enable automatic creation of worst-case loading patterns and substantially reduces the manual involvement that is described in the following steps:

a. Define the structure (excluding any loading) in LUSAS Modeller noting that Autoloader projects vehicle loads in the -ve Z axis direction. Therefore, models for use with Autoloader must be set up to ensure that loading can be applied in the global Z axis.

b. Define Influence Lines/Surfaces for each design point in the structure. (See Appendix A)
c. Select the **File > LUSAS Datafile** menu command and select the Datafile type as '**Influence Line Analysis'** to generate:
   (a) a LUSAS datafile (.dat) for each Influence Line/Surface defined
   (b) an influence information file (.inf) for each Influence Line/Surface
a. Run each datafile through the LUSAS Solver to generate
   (a) a LUSAS graphical results file (.mys) for each Influence Line/Surface
   (b) an Autoloader displacement file (.dsp)
a. Check the deformed shapes in LUSAS Modeller to ensure correct behaviour of the model for each influence line/surface.
b. Prepare an Autoloader input deck (.inp). This is documented fully in the Reference section of this manual.
c. Run Autoloader on this input deck. This generates four LUSAS Modeller command files, specifying loading for ultimate limit states and serviceability conditions.
d. Re-load the LUSAS Modeller pre-processing model saved in (1) and run the four command files generated in (6). Tabulate the LUSAS datafile without Influence Lines/Surfaces to generate a single input deck with all live load cases defined.
e. Solve and post-process the results as usual.

This sequence will define the Ultimate Limit State and Serviceability loading for the structure being analysed for the influence lines/surfaces under consideration.

Using the VLO/Autoloader Wizard in LUSAS

As an alternative to manually creating the required files for analysis, an Vehicle Load Optimisation wizard is provided in LUSAS to substantially speed-up the creation of the files required. A schematic diagram illustrating the interaction between the VLO/Autoloader Wizard, the Autoloader program and the LUSAS Modeller and Solver follows this section which describes the steps carried out by the VLO/Autoloader Wizard and the steps which need to be subsequently carried out to complete the analysis and view the results.

Step 1

In LUSAS Modeller, define the structure (excluding any loading) and define Influence Lines/Surfaces for each design point in the structure. (See Appendix A) and define or select any lines in the model representing edge of carriageways (kerb lines). Then select the **Analyses** > **Vehicle Load Optimisation** menu item to run the wizard.

Running the Autoloader Wizard for a design code that is supported by the Autloader software causes the following files to be generated from the active LUSAS model:

- deck_1.dat, deck_2.dat, deck_3.dat LUSAS data files for use by Autoloader.
- deck.inp - Input data for Autoloader produced by LUSAS. Contains design information such as carriageway widths, numbers of carriageways, vehicles, etc.
- deck_1.inf, deck_2.inf, deck_3.inf Influence files created by LUSAS for use by Autoloader.

Step 2

The VLO/Autoloader Wizard also invokes the LUSAS Solver for each LUSAS data file which produces displacement files for use by Autoloader and LUSAS results and output files for verification of the influence lines selected. The following files are produced:

- **deck*.dsp** - LUSAS displacement output files for Autoloader. One for each influence surface.
- **deck*.mys** - LUSAS results files produced from the influence line analysis - one for each influence surface. These contain results of the influence line analyses only and are used in LUSAS in-conjunction with a deformed mesh plot to check that the influence lines have been correctly defined. They do not contain results from any loadings and are not used further in this analysis.
- **deck*.out** - LUSAS output files containing details of model data, assigned attributes and selected statistics of the analysis. One file for each influence surface.

Step 3

The VLO/Autoloader Wizard then invokes the Autoloader program to calculate the worst case loading configurations by combining the displacement and influence files with the Autoloader input file. Autoloader then generates the following files:

- **deck.sum** - Summary file produced by Autoloader of the total load applied to the structure for each influence surface.
- **deck.ver** - Verification file produced by Autoloader listing all of the loads applied to the structure.
- **Deck_u1.vbs** - LUSAS visual basic script produced by Autoloader defining the Ultimate Limit State load combinations on the structure for all of the influence surfaces.
- **Deck_u2.vbs -** LUSAS visual basic script produced by Autoloader defining the Ultimate Limit State load combinations on the structure for all of the influence surfaces.
- **Deck_s1.vbs** LUSAS visual basic script produced by Autoloader defining the Serviceability Limit State load combinations on the structure for all of the influence surfaces.

- **Deck_s2.vbs** - LUSAS visual basic script produced by Autoloader defining the Serviceability Limit State load combinations on the structure for all of the influence surfaces.

Note. The suffixes U1, U2, S1 and S2 on the generated LUSAS command files refer to Ultimate Limit State combinations 1 and 2, and Serviceability Limit State combinations 1 and 2 from Table 1 in the design code BD37/88 used in this example. The applicable clauses are 6.2 and 6.3, the load factors are applied as per the entries in this table.

Step 4

The LUSAS visual basic script files generated are automatically run to create the required loading datasets and load combinations for the active model according to the selected design code. Whilst these are being created it is suggested that the following schematic diagram should be studied to fully understand the processes involved in using the Autoloader Wizard and the range of files produced.

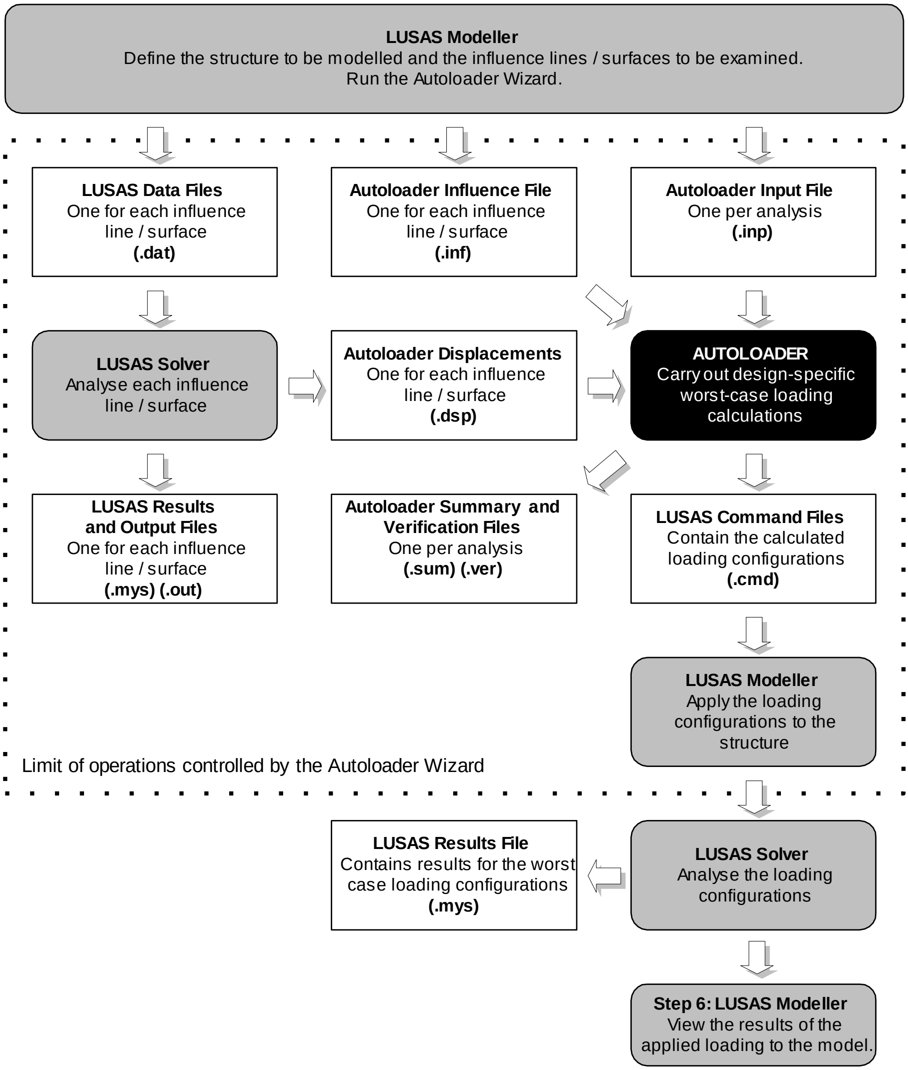
Step 5

An analysis is run using LUSAS Solver to calculate the loading effects on the model.

Step 6

Results are viewed using LUSAS Modeller.

Schematic of LUSAS and Autoloader Interaction



Chapter 3
Reference Manual

Introduction

This chapter contains descriptions of all of the Autoloader Parameters. All default values are in SI units, apart from forces, which are specified in kN, and angles, which are specified in degrees.

A parameter is specified as follows:

`<parameter> = <value>`

where:

`<parameter>`	is replaced by the name of the parameter
`<value>`	is replaced by the value to be assigned to that parameter.

For example:

```
CWSHAPE = straight
CWDIR = 45.0
```

In the following descriptions, each parameter is followed by a value and a field, if appropriate. The value defines which value(s) the parameter accepts. This can take a number of forms:

- a multiple choice, for example *straight* or *curved* - the choices are always in italics, and the value must be one of those specified.

  For example: **`CWSHAPE = straight;`**
- a list - this is a number of values, separated by commas, for example **`CWDIR = 34.0, 35.0, 36.0.`** Any list can be split over more than one line;
- an integer - a whole number;
- a real number - a number that may contain decimal fractions;
- a Boolean - a special type of multiple choice - must be either **`TRUE`** or **`FALSE`**. (**`YES`** or **`NO`** can also be specified)

- a point - two real numbers, separated by a comma, e.g. **`45.0, 46`**;
- a string - a series of letters and numbers. This cannot contain spaces, tab characters or carriage returns;
- a filename - a specification of a filename. This may contain a path specification.

The default field defines the value of the parameter if it is has not been specified.

Some Autoloader parameters require an input value. These are **`CWSHAPE`**, **`CWNUM`**, **`KBPOS`**, **`CWDIR`**, **`DECK`** and **`FILE`**. **`CWDIR`** is only required when **`CWSHAPE`** is specified as straight.

The following sections are set out to provide easy access to the relevant parameters. Parameters from the Required Parameters section will always be used. The Optional Parameters are applicable to all standards, and there are several sections with parameters applicable to the design loading standard in use.

Required Parameters

Autoloader requires the following parameters.

- **CWSHAPE** Carriageway Shape
- **CWNUM** Number of Carriageways
- **KBPOS** Kerb Positions
- **CWDIR** Carriageway Direction
- **FILE** Input Filename
- **ISRCH** LUSAS Search area assignment
- **ORIGIN** Origin to be used for load definition/application
- **OUTPUT_PREFIX** Prefix for concatonated loading (.vbs) files
- **SUMMARY** Generate summary of information + verification file

CWSHAPE Carriageway Shape

Value: *straight* or *curved*

`CWSHAPE` specifies whether the carriageway is curved or straight in plan view. In conjunction with **`CWNUM`**, it affects the data required for **`CWDIR`** and **`KBPOS`**.

The following line specifies a straight carriageway:

```
CWSHAPE = straight
```

and the following a curved carriageway:

```
CWSHAPE = curved
```

Only the first character is significant, and case is ignored, therefore the next three examples all have the same effect:

```
CWSHAPE = S
CWSHAPE = Straight
CWSHAPE = s
```

If neither *curved* nor *straight* is specified, Autoloader reports an **INVALID_DATA** error and aborts.

CWNUM Number of Carriageways

Value: positive integer

CWNUM specifies the number of carriageways to be placed on the deck. Autoloader assumes the kerbs of each carriageway are parallel.

The following line specifies 2 carriageways:

```
CWNUM = 2
```

If a positive integer is not specified, then Autoloader reports an **INVALID_DATA** error, and aborts.

KBPOS Kerb Positions

Value: list of points
Units: metres

KBPOS defines the positions of the kerbs on each carriageway. Points are specified in Cartesian form.

When the carriageway shape is straight, i.e. **CWSHAPE** is Straight, then two points per carriageway are specified. The first of these is the ‘base’ point and is on a kerb at the start of the carriageway. The other point is anywhere on the other kerb. Using this data and the value of **CWDIR**, the width of the carriageway can be calculated.

The following specifies two 10m wide straight carriageways with kerb positions at (0.0, 0.0) and (1.0, 10.0) and (0.0, 11.0) and (1.0, 21.0):

```
CWSHAPE = straight
CWNUM = 2
KBPOS = 0.0, 0.0, 1.0, 10.0,
       0.0, 11.0, 1.0, 21.0
```

When the carriageway shape is curved, that is **CWSHAPE** is Curved specify four points per carriageway. The first of these is the ‘base’ point and is on a kerb at the start of the carriageway. The next two points are on the same kerb, and are used to calculate the centre of curvature and the radius of curvature. The final point is anywhere on the other kerb. Autoloader uses this data to calculate the width of the carriageway.

The following specifies two 10m wide carriageways with centre of curvature (10.0, 10.0), moving in a clockwise direction:

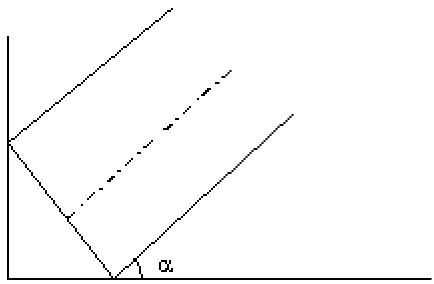
```
CWSHAPE = curved
CWNUM = 2
KBPOS = 0.0, 10.0, 10.0, 20.0, 20.0, 10.0, 10.0, 30.0,
       -1.0, 10.0, 10.0, 21.0, 21.0, 10.0, 10.0, 31.0
```

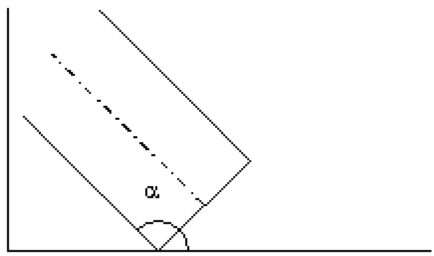
If an incorrect number of kerb positions are specified, then Autoloader reports an **INVALID_DATA** error, and aborts.

CWDIR Carriageway Direction

Value: positive real number between 0 and 180 inclusive
Units: degrees

CWDIR is only required when the carriageway shape is straight. It specifies the angle of inclination in degrees of the carriageway in an anticlockwise direction relative to the positive x axis, as in the diagram:





For example:

```
CWDIR = 30
```

or

```
CWDIR = 120
```

FILE Input Filename

Value: filename

FILE specifies the file containing the influence surface. The filename will have an INF suffix. For more information see Autoloader Output.

For example:

```
FILE = deck_1.INF
```

Multiple runs may be carried out by listing file names;

```
FILE = deck_1.INF,deck_2.INF,deck_3.inp ….deck_n.inp
```

ISRCH LUSAS Search area assignment

Value: INTEGER

Autoloader produces a LUSAS command file which in turn defines and assigns the loading onto the structure. To enable 3 dimensional applications it may be necessary to assign the loading to a particular level or part of the structure. This is achieved by specifying a search area which exists in the model file.

For example:

```
ISRCH = 1
```

If ISRCH is not specified then the default value of ISRCH=0 will be adopted.

ORIGIN Origin to be used for load definition/application

Value: list of co-ordinates of origin x,y,z

For example:

```
ORIGIN = 10,25.2,6
```

All loading will be defined with respect to the x and y co-ordinates and the loading will subsequently be assigned onto a point with the co-ordinates of the origin. Note that the load will be located at the z=6.0 position.

If ORIGIN is not specified then the default values of 0,0,0, will be adopted.

OUTPUT_PREFIX Prefix for concatonated loading (.vbs) files

Value; filename

For example:

```
OUTPUT_PREFIX=deck_
```

Autoloader will define 4 loading vbs files (for BD 37/88), deck_u1.vbs, deck_u2.vbs, deck_s1.vbs, deck_s2.vbs. If 3 influence effects are being investigated;

```
FILE = test_1.INF, test_2.INF, test_3.inf
```

deck_u1.vbs will contain the load configurations for test1.INF, test_2.INF and test_3.INF at the Ultimate limit state for combination 1. Similarly deck_u2.vbs contains the load configurations for test_1.INF, test_2.INF and test_3.INF at the Ultimate limit state for combination 2, etc….

If OUTPUT_PREFIX is not specified then a unique vbs file will be created for each influence effect comprising load configurations for combinations U1, U2, S1, S2 (BD 37/88).

SUMMARY Generate summary of information + verification file

Value: filename

```
SUMMARY = bend
```

Autoloader generates a summary file **bend.sum** which contains the effects for each vehicle considered at each influence point under consideration. Typical output is as follows;

```
Influence - Reaction At support 2     Influence Title
File s3_r_1.inf                       Origin File Of Effects
-----------------------------------------------------------
Effect     ULS1   ULS2         SLS1         SLS2
-----------------------------------------------------------
HA Only positive   2948.068   2456.723   2358.454    1965.378
HB + HA positive   4255.682   3600.962   3600.962    3273.601
HA Only negative   -247.097   -205.914   -197.677    -164.731
HB + HA negative   -287.615   -243.366   -243.366    -221.242
-----------------------------------------------------------
```

Autoloader will also define a verification file **bend.ver** which contains a complete breakdown of load configurations; loaded length, lane intensity, beta factors and global reactions for each loadcase are included as follows;

```
Influence - Reaction At Support 2       Influence Title
File lev_2_1.inf   Origin File Of Effects
-----------------------------------------------------------
! Positive HA Only
! Cway/Lane    Length     Int.      Beta     Effect    Reaction
! 1/1          56.062   24.068    1.000    721.101    1469.280
! 1/2          57.927   23.989    1.000    592.093    1509.611
! 1/3          59.212   23.937    0.600    277.686     922.405
```

```
! 1/4           59.709    23.917    0.600    200.489    928.821
! 1/5           59.120    23.940    0.600    124.181    921.200
! 1/6           55.698    24.083    0.600     49.828    876.844
! Total                                     1965.378   6628.167

! Positive HB + HA
! 1277/12A(for) span 13.69 effect=1948.474 pos=76.044,
! -4.046 skew=351.283 reaction=3947.544
! Cway/Lane     Length      Int.     Beta     Effect   Reaction
! 1/1           146.30    21.866    0.600      2.720    590.120
                     4
! 1/2           57.927    23.989    1.000    485.098   1327.651
! 1/3           59.212    23.937    1.000    462.810   1537.341
! 1/4           59.709    23.917    0.600    200.489    928.821
! 1/5           59.120    23.940    0.600    124.181    921.205
! 1/6           55.698    24.083    0.600     49.828    876.844
! Total                                     3273.601  10129.527

! Negative HA Only
! Cway/Lane     Length      Int.     Beta     Effect   Reaction
! 1/1           53.559    24.178    1.000    -48.171   1414.942
! 1/2           55.332    24.099    1.000    -42.097   1453.460
! 1/3           55.487    24.093    0.600    -21.927    874.094
! 1/4           54.796    24.123    0.600    -19.046    865.098
! 1/5           53.949    24.160    0.600    -17.032    854.059
! 1/6           55.990    24.071    0.600    -16.456    880.639
! Total                                     -164.731   6342.291

! Negative HB + HA
! HB(for) span 6.00 hbunits=45.0 effect=-84.384 pos=41.15,
! 0.06 skew=355.2 reaction=1800.000
! Cway/Lane     Length      Int.     Beta     Effect   Reaction
! 1/1           87.059    23.031    0.600     -11.70    373.926
! 1/2           55.332    24.099    1.000    -36.075   1321.332
! 1/3           55.487    24.093    1.000    -36.546   1456.823
! 1/4           54.796    24.123    0.600    -19.046    865.098
! 1/5           53.949    24.160    0.600    -17.032    854.059
```

```
! 1/6           55.990   24.071    0.600    -16.456     880.639
! Total                                    -221.242    7551.876
```

Typical Calculation For Reaction, consider Negative HA Only;

For Cway/Lane, 1/1
Loaded Length x *intensity* x *beta factor + Knife Edge Load*
(53.559 x 24.178+120) x 1.0 = 1414.942 *Target = 1414.942*

Optional Parameters

The following parameters are applicable to all loading standards.

- **CLINCR** Increment for Influence Lines
- **DLALLOWANCE** Dynamic Load Allowance
- **INFL** Influence Area to be Loaded
- **KEL** Value of Knife Edge Load
- **LANE_FACTORS** Lane Beta Factors
- **LANE_MOD_FACTORS** Lane Modification Factors
- **LTINCR** Increment FOR longitudinal movement
- **KELNUM** NUMBER OF KEL'S TO APPLY
- **PARAMETER_FILE** Extra Parameter File
- **STANDARD** Design Standard to be applied
- **TRINCR** Increment for TRANSVERSE movement
- **UDL_CURVE** UDL Loading Intensity Curve
- **UDL_LIMIT** Loaded Length Limit
- **UNITS_INPUT** INPUT UNITS
- **UNITS_OUTPUT** OUTPUT UNITS
- **USE_CUSP** Indicates USE OF CUSPING
- **USE_KEL** Indicates USE OF KEL’S
- **VEHDIR** Vehicle Direction
- **VEHICLE_FILE** Autoloader vehicle library
- **VEHICLES** Vehicle Name(s)

CLINCR Increment for Influence Lines

Value:	positive real value
Default:	1.0m
Units:	metres

CLINCR specifies the increment interval used when interpolating influence values to obtain the influence lines for the centrelines of the notional lanes. The smaller the increment, the more accurately the vehicle loads will be positioned on the carriageway. The larger the increment, the more quickly Autoloader runs. An appropriate value must be chosen from experience. If an invalid value is specified, Autoloader gives a warning and uses the default value.

For example:

```
CLINCR = 3.0
```

DLALLOWANCE Dynamic Load Allowance

Value: positive real value
Default: 0.0

DLALLOWANCE specifies the dynamic load allowance for gamma factors. This will generally be used when AUSTROADS is applied, but can also be used within the other standards. If an invalid value is specified, Autoloader gives a warning and uses the default value.

For example:

```
DLALLOWANCE = 0.2
```

INFL Influence Area to be Loaded

Value: *both*, *positive* or *negative*
Default: *positive*

INFL specifies the area of the influence surface to be loaded, either both, positive or negative. Values of -1 and 1 for negative and positive respectively can also be specified.

For example:

```
INFL = negative
```

Only the first character is significant, and case is ignored, therefore the next four examples all have the same effect:

```
INFL = N
INFL = Negative
INFL = n
INFL = -1
```

KEL Value of Knife Edge Load

Value: positive real number

Default: as per design standard used
Units: kN

KEL allows specification of an alternate value for the Knife Edge Load to use when applying KEL loading.

For example:

```
KEL = 100
```

LANE_FACTORS Lane Beta Factors

Value: list of positive real values
Default: as per design standard used

LANE_FACTORS specifies alternate HA lane factors to use when calculating the effects of HA loads. The format is a list of factors. The first factor is used for the lane with the greatest effect, the second is used for the lane with the second highest effect, etc. If there are more lanes than factors, then the last factor in the list is used for any lanes without corresponding factors. It should be noted that for the JKR standard the make up lane is also treated as an HA lane and a factor should be specified for it bearing in mind that it is quite likely to have the least effect.

For example:

```
LANE_FACTORS = 1.0, 1.0, 0.71, 0.67
```

LANE_MOD_FACTORS Lane Modification Factors

Value: list of positive real values
Default: as per design standard used

LANE_MOD_FACTORS allows the specification of alternate lane modification factors to use when calculating effects. The format is a list of factors. The first value is used when there is one lane loaded, the second when there is two, etc. If there are more lanes than factors, then the last factor in the list is used. This will generally be used when AUSTROADS is applied, but can also be used within the other standards.

For example:

```
LANE_MOD_FACTORS = 1.0, 0.9, 0.8, 0.7
```

LTINCR Increment For Longitudinal Movement

Value: positive real value
Default: 1.0m
Units: metres

LTINCR specifies the increment used when moving the abnormal vehicle along the carriageway. The smaller the increment, the more accurately positioned the vehicle on the carriageway. The larger the increment, the quicker Autoloader runs. An appropriate value must be chosen through experience. If an invalid value is specified, Autoloader gives a warning and uses the default value.

For example:

```
LTINCR = 1.0
```

KELNUM Number of KEL'S to Apply

Value: zero or positive integer value
Default: 1

KELNUM specifies the number of KEL's to apply within a specified design lane. This will generally be used when AUSTROADS is applied, but can also be used within the other standards. If an invalid value is specified, Autoloader gives a warning and uses the default value. A value of 0 has the same effect as specifying USE_KEL = FALSE.

For example:

```
KELNUM = 2
```

PARAMETER_FILE Extra Parameter File

Value: filename
Default: none

PARAMETER_FILE specifies another parameter file for Autoloader. This option allows the specification of a generic set of parameters in another file and use of that file in a series of Autoloader runs, for example for a specific project. This additional file can contain anything that the input file can contain. It is read after the entire input file is processed, and any parameter specified within this file will overwrite one specified in the original file.

For example:

```
PARAMETER_FILE = C:\PROJECT\PARA.INP
```

STANDARD Design Standard to be applied

Value: *AASHTO*, *AUSTROADS*, *BD2197*, *BD3788*, *JKR*, *HK*, *RU* or *RAIL*
Default: *BD3788*

STANDARD allows specification of which loading standard to apply.

For example:

```
STANDARD = JKR
```

TRINCR Increment for Transverse Movement

Value: positive real value
Default: 1.0m
Units: metres

TRINCR specifies the increment used when moving the vehicle across the carriageway. The smaller the increment, the more accurately positioned the vehicle on the carriageway. The larger the increment, the more quickly Autoloader runs. An appropriate value must be chosen from experience. If an invalid value is specified, Autoloader gives a warning and uses the default value.

For example:

```
TRINCR = 1.0
```

UDL_CURVE UDL Loading Intensity Curve

Value: list of sets of 2 values
Default: as per design standard used
Units: metres, kN

UDL_CURVE allows changing the relationship between loaded length and intensity of the UDL applied. Each pair of values is a length and an intensity for that length. Autoloader does straight line interpolation between these values. If more accurate loading intensity is as many points as possible should be specified.

For example:

```
UDL_CURVE = 2, 112.2,
            4, 132.7,
            6, 101.2,
            8, 83.4,
            10, 71.8,
            12, 63.4,
            14, 57.2
```

For example, using the table above, the value of load intensity for a loaded length of 13m is calculated as 60.3.

UDL_LIMIT Loaded Length Limit

Value: positive real number
Default: as per design standard used
Units: m

`UDL_LIMIT` allows specification of a loaded length, under which the lane in question is not loaded. This can be used in conjunction with HA_ALTERNATIVE to create a situation such as in BD 21/97, where if the loaded length is below 2m, the lane is loaded with a Single Axle Load instead of a UDL + KEL.

For example:

```
UDL_LIMIT = 2
```

UNITS_INPUT Input Units

Value: one or more of units as specified below
Default: kN, m, deg

`UNITS_INPUT` allows specification of a set of input units. These units only apply to the autoloader input file and the parameter file. The autoloader vehicle library must be in kiloNewtons and metres.

Autoloader recognises the following units:

Type	Unit	Description
force	lb	pounds force
	ton	tons force
	n	Newtons
	kn	kiloNewtons
	kip	thousand lbs force
	kg	kilogram force
	tonne	tonnes force
	pdl	poundal
	mn	megaNewton
	dyne	dyne
rotation	deg	degrees
	rad	radians
length	yd	yards
	ft	feet
	in	inches
	m	metres
	cm	centimetres
	mm	millimetres

For example:

```
UNITS_INPUT = lb, rad, ft
```

UNITS_OUTPUT Output Units

Value: one or more of units as specified below
Default: kN, m, deg

UNITS_OUTPUT allows specification of a set of output units. These units will be used to write the output loading files. They do not affect the Maximum effects displayed by Autoloader, since Autoloader calculates the effect in accordance with the influence surface.

Autoloader recognises the following units:

Type	Unit	Description
force	lb	pounds force
	ton	tons force
	n	Newtons
	kn	kilonewtons
	kip	thousand lbs force
	kg	kilogram force
	tonne	tonnes force
	pdl	poundal
	mn	megaNewton
	dyne	dyne
rotation	deg	degrees
	rad	radians
length	yd	yards
	ft	feet
	in	inches
	m	metres
	cm	centimetres
	mm	millimetres

For example:

```
UNITS_OUTPUT = lb, rad, ft
```

USE_CUSP Indicates Use of Cusping

Value: Boolean
Default: dependent upon the loading standard specified

USE_CUSP specifies whether to apply cusping when working out loadable areas. If TRUE, cusping is applied. If FALSE, cusping is not applied.

For example:

```
USE_CUSP = FALSE
```

Only the first character is significant, and case is ignored, therefore the next three examples all have the same effect:

```
USE_CUSP = F
USE_CUSP = False
USE_CUSP = f
```

USE_KEL Indicates Use of KEL'S

Value: Boolean
Default: TRUE

USE_KEL specifies whether to apply Knife Edge Loads. If TRUE, Knife Edge Loads are applied to areas with UDL loading, according to the standard in use. If FALSE, no KEL'S are applied.

For example:

```
USE_KEL = FALSE
```

Only the first character is significant, and case is ignored, therefore the next three examples all have the same effect:

```
USE_KEL = F
USE_KEL = False
USE_KEL = f
```

VEHDIR Vehicle Direction

Value: *forward*, *reverse* or *both*
Default: *both*

VEHDIR specifies the vehicle direction which is used to calculate the effects of each vehicle. *Forward* means the vehicle is run with axles with the lowest axle positions aligned towards the base point of the carriageway. *Reverse* means the vehicle is run with axles with the highest axle positions aligned towards the base point of the carriageway. *Both* means both cases are run. For an abnormal vehicle, either *forward* or *reverse* could produce the greatest effect, dependent upon the shape of the influence surface and the increments used. The default is both directions.

For example:

```
VEHDIR = forward
```

Only the first character is significant, and case is ignored, therefore the next three examples all have the same effect:

```
VEHDIR = B
VEHDIR = Both
VEHDIR = b
```

VEHICLE_FILE Autoloader vehicle library

Value: filename
Default: autoload.vec

VEHICLE_FILE specifies an alternate file containing the Autoloader Vehicle Library (Autoloader Vehicle Library for more details). This enables definition of another vehicle for a set of autoloader runs. By default, Autoloader looks for autoload.vec in the working directory.

For example:

```
VEHICLE_FILE = C:\PROJECT\PROJECT.VEC
```

VEHICLES Vehicle Name(s)

Value: list of strings
Default: none

VEHICLES defines a list of vehicles in order to calculate the greatest effect. Autoloader tests every vehicle in this list. Each vehicle must be specified in the Autoloader vehicle library. For more information, see Autoloader Vehicle Library. If the field is not specified, HA or equivalent only loading is assumed.

For example:

```
VEHICLES = HB, 1277/12B
```

AASHTO Parameters

The following parameters apply to the AASHTO standard.

LOAD_LEVEL AASHTO LOAD LEVEL

Value: Boolean
Default: FALSE

LOAD_LEVEL specifies the level of AASHTO loading to be applied during the AUTOLOADER run (ref AASHTO cl. 3.7.2 et al).

For example:

```
LOAD_LEVEL = H20
```

EFFECT_TYPE AASHTO LOAD EFFECT

Value: *Moment or Shear*
Default: None

LOAD_EFFECT allows the specification of the type of effect being investigated. AUTOLOADER then applies the relevant KEL in accordance with AASHTO fig 3.7.6B.

For example:

```
LOAD_EFFECT = Moment
```

UDL_ALTERNATIVE Vehicle to be used as ALTERNATIVE TO UDL + KEL loading

Value: list of strings
Default: none

UDL_ALTERNATIVE defines a list of vehicles to be tried as an alternative to UDL + KEL loading. When loading a lane, Autoloader tries to place vehicles from this list within the lane, and if the effect is greater than UDL + KEL loading, it uses that vehicle instead. Each vehicle must be specified in the Autoloader vehicle library. For more information see Autoloader Vehicle Library

For example:

```
UDL_ALTERNATIVE = HS20
```

LF_GAMMA LOAD FACTOR partial load factors

Value: list of positive real values
Default: as per design loading standard used

LF_GAMMA allows specification of the partial load factors for the Load Factor Design. The list is ordered as follows:

LF I
LF IA
LF III

LF IV
LF VI
LF VIII
LF X

For example:

```
LF_GAMMA = 1.0, 1.1, 1.2, 1.3, 1.4, 1.5, 1.6
```

SL_GAMMA SERVICE LOAD partial load factors

Value: list of positive real values
Default: as per design loading standard used

SL_GAMMA allows specification of the partial load factors for the Service Load State. The list is ordered as follows:

SL I
SL IA
SL III
SL IV
SL VI
SL VIII
SL X

For example:

```
SL_GAMMA = 1.0, 1.1, 1.2, 1.3, 1.4, 1.5, 1.6
```

AUSTROADS Parameters

The following parameters apply to the AUSTROADS standard.

AUST_TYPE Type of road

Value: *urban* or *rural*
Default: *rural*

AUST_TYPE allows specification of either an urban road or a rural road, as defined in the AUSTROADS standard. This option has no effect in any of the other standards. If an invalid value is specified, Autoloader gives a warning and the default value is used.

For example:

```
AUST_TYPE = urban
```

Only the first character is significant, and case is ignored, therefore the next three examples all have the same effect:

```
AUST_TYPE = U
AUST_TYPE = Urban
AUST_TYPE = u
```

L44_ALTERNATIVE Vehicle to be used as ALTERNATIVE L44 loading

Value: list of strings
Default: none

L44_ALTERNATIVE defines a list of vehicles to be tried as an alternative to L44 loading. When loading an L44 lane, Autoloader tries to place vehicles from this list within the lane, and if the effect is greater than L44 loading, uses that vehicle instead. Each vehicle must be specified in the Autoloader vehicle library. For more information see Autoloader Vehicle Library

For example:

```
L44_ALTERNATIVE = T44
```

LANE_INCR Increment for moving lanes

Value: positive real value
Default: 0.25m
Units: metres

LANE_INCR specifies the increment interval used when moving lanes across the carriageway within the AUSTROADS standard. The smaller the increment, the more accurately the vehicle loading is positioned on the carriageway. The larger the increment, the more quickly Autoloader runs. An appropriate value must be chosen from experience. If an invalid value is specified, Autoloader gives a warning and uses the default value.

For example:

```
LANE_INCR = 1.0
```

LANE_WIDTH L44 LANE WIDTH

Value: positive real value
Default: 2.5m
Units: metres

LANE_WIDTH specifies the L44 lane width. If an invalid value is specified, Autoloader gives a warning and uses the default value.

For example:

```
LANE_WIDTH = 3.0
```

SLS_GAMMA SLS partial load factors

Value: list of positive real values
Default: as per design loading standard used

SLS_GAMMA specifies the partial load factors for the Serviceability Limit State. The list is ordered as follows:

L44/T44 alone
HLP with L44/T44 or HLP alone

For example:

```
SLS_GAMMA = 2.0, 1.5
```

TRAFFIC_LANES SPECIFY MARKED TRAFFIC LANES

Value: list of sets of 5 values

TRAFFIC_LANES defines marked traffic lanes in relation to the carriageway(s). Each set of values consists of a carriageway number, and two sets of x and y values, through which points the lane edges pass. The traffic lane is assumed to be parallel to the carriageway kerbs.

For example:

```
TRAFFIC_LANES = 1, 0.0, 0.0, 0.0, 3.0,
                     2, 2.0, 3.5, 0.0, 6.5,
                     1, 0.0, 3.0, 0.0, 6.0
```

This defines three traffic lanes, the first in carriageway one, with one side of the traffic lane (0, 0), and the other passing through (0, 3). The second defines a lane on carriageway 2, the edge of which passes through (2, 3.5) and the other edge passes through (0, 6.5). The third defines a lane in carriageway 1, the edge of which passes through (0, 3) and the other edge through (0, 6).

ULS_GAMMA ULS partial load factors

Value: list of positive real values
Default: as per design loading standard used

ULS_GAMMA specifies the partial load factors for the Ultimate Limit State. The list is ordered as follows:

L44/T44 alone

HLP with L44/T44 or HLP alone

For example:

```
ULS_GAMMA = 2.0, 1.5
```

USE_HLP IndicateS USE OF abnormal vehicle

Value: Boolean
Default: If VEHICLES has been specified, TRUE, otherwise FALSE

USE_HLP specifies whether to apply an abnormal vehicle. If TRUE, the abnormal vehicles are applied to the carriageway. If FALSE, then L44/T44 only loading is applied. This option is the same as USE_HB.

For example:

```
USE_HLP = FALSE
```

Only the first character is significant, and case is ignored, therefore the next three examples all have the same effect:

```
USE_HLP = F
USE_HLP = False
USE_HLP = f
```

USE_L44 IndicateS USE OF L44/T44 LOADING

Value: Boolean
Default: TRUE

USE_L44 specifies whether to apply the L44/T44 loading (which may include KEL loading).

For example:

```
USE_L44 = FALSE
```

Only the first character is significant, and case is ignored, therefore the next three examples all have the same effect:

```
USE_L44 = F
USE_L44 = False
USE_L44 = f
```

BD 21/97 Parameters

These parameters are for use with BD 21/97.

ULS_GAMMA_F3 γ_{f3} ULS PARTIAL LOAD FACTOR

Value: List of positive real values
Default: None

ULS_GAMMA_F3 specifies the γf3 partial load factors for the Ultimate Limit State. The list is ordered as follows:-

ULS effect

For example:

```
ULS_GAMMA_F3 = 1.10
```

USE_GAMMA_F3 INDICATES USE OF γ_{f3} LOAD FACTOR

Value: Boolean
Default: FALSE

USE_GAMMA_F3 specifies whether or not to apply the γ_{f3} partial load factors to the overall effect. If TRUE, γ_{f3} is applied, if FALSE, γ_{f3} is not applied.

For example:

```
USE_GAMMA_F3 = TRUE
```

Only the first character is significant, and case is ignored, therefore the next three examples all have the same effect:

```
USE_ GAMMA_F3 = F
USE_ GAMMA_F3 = False
USE_ GAMMA_F3 = f
```

ADJFACTOR_CURVE ADJustment FACTOR Curve

Value: list of sets of 2 values
Default: as per design standard used
Units: metres, no units

ADJFACTOR_CURVE allows changing the relationship between loaded length and the Adjustment factor. Each pair of values is a length and a factor for that length. Autoloader performs linear interpolation between these values. If a more accurate value is required as many points as possible should be defined.

For example:

```
ADJFACTOR_CURVE = 0, 1.46,
                  20, 1.46,
                  40, 1,
                  50, 1
```

For example, using the table above, the Adjustment factor for a loaded length of 25m is calculated as 1.345.

HA_ALTERNATIVE Vehicle To be used as ALTERNATIVE HA loading

Value: list of strings
Default: none

HA_ALTERNATIVE defines a list of vehicles to be tried as an alternative to HA loading. When loading an HA lane, Autoloader tries to place vehicles from this list within the lane, and if the effect is greater than HA loading, uses that vehicle instead. Each vehicle must be specified in the Autoloader vehicle library. For more information, see Autoloader Vehicle Library.

This could be used along with UDL_LIMIT to implement the Single Axle loading when the loaded length for one of the lanes is below 2m.

For example:

```
HA_ALTERNATIVE = SAL-HP
```

KFACTOR_CURVE K FACTOR Curve

Value: list of sets of 2 values
Default: as per design standard used
Units: metres, no units

KFACTOR_CURVE allows changing the relationship between loaded length and the K factor. Each pair of values consists of a length and a factor for that length. Autoloader performs linear interpolation between these values. If a more accurate value is required as many points as possible should be defined.

For example:

```
KFACTOR_CURVE = 0, 0.79,
                10, 0.79,
                15, 0.78,
                30, 0.78,
                40, 0.79,
                50, 0.78
```

For example, using the table above, the K factor for a loaded length of 35m is calculated as 0.785.

LANE_WIDTH_TABLE Notional lane widths

Value: list of sets of three real values

Default: as per design standard used
Units: metres

LANE_WIDTH_TABLE alters the relationship between carriageway width and the number of notional lanes. Each 3 values are taken as a lower limit, an upper limit and a number of notional lanes. After calculating the carriageway width, Autoloader works its way down the table, checking the calculated width against the values in the table. If the width is greater than or equal to the lower limit, and less than or equal to the upper limit, then the number of notional lanes is given by the third value.

For example:

```
LANE_WIDTH_TABLE = 5, 7, 2,
                      7, 10.5, 3,
                      10.5, 14, 4,
                      14, 18.25, 5,
                      18.25, 22, 6
```

In this case a width of 12m will have 4 notional lanes of width 3m each.

SURFACE Quality of road surface

Value: *good* or *poor*
Default: *poor*

SURFACE specifies the quality of road surface, as defined in the BD 21/97 standard. This option has no effect in any of the other standards. If an invalid value is specified, Autoloader gives a warning and uses the default value.

For example:

```
SURFACE = good
```

Only the first character is significant, and case is ignored, therefore the next three examples all have the same effect:

```
SURFACE = G
SURFACE = Good
SURFACE = g
```

TRAFFIC_LANES SPECIFY MARKED TRAFFIC LANES

Value: list of sets of 5 values

TRAFFIC_LANES defines marked traffic lanes in relation to the carriageway(s). Each set of values consists of a carriageway number, and two sets of x and y values, through which points the lane edges pass. The traffic lane is assumed to be parallel to the carriageway kerbs.

For example:

```
TRAFFIC_LANES = 1, 0.0, 0.0, 0.0, 3.0,
                      2, 2.0, 3.5, 0.0, 6.5,
                      1, 0.0, 3.0, 0.0, 6.0
```

This defines three traffic lanes, the first in carriageway one, with one side of the traffic lane (0, 0), and the other passing through (0, 3). The second defines a lane on carriageway 2, the edge of which passes through (2, 3.5) and the other edge passes through (0, 6.5). The third defines a lane in carriageway 1, the edge of which passes through (0, 3) and the other edge through (0, 6).

TRAFFIC_LEVEL TRAFFIC_LEVEL

Value: *heavy, medium* or *low*
Default: *low*

TRAFFIC_LEVEL specifies the traffic level, as defined in the BD 21/97 standard. This option has no effect in any of the other standards. If an invalid value is specified, Autoloader gives a warning and uses the default value.

For example:

```
TRAFFIC_LEVEL = medium
```

Only the first character is significant, and case is ignored, therefore the next three examples all have the same effect:

```
TRAFFIC_LEVEL = M
TRAFFIC_LEVEL = Medium
TRAFFIC_LEVEL = m
```

ULS_GAMMA ULS partial load factors

Value: list of positive real values
Default: as per design loading standard used

ULS_GAMMA specifies the partial load factors for the Ultimate Limit State. The list is ordered as follows:

Assessment factor

For example:

```
ULS_GAMMA = 1.6
```

BD 37/88 Parameters

These parameters are for use with BD 37/88.

ULS_GAMMA_F3 γ_{f3} ULS PARTIAL LOAD FACTOR

Value: List of positive real values
Default: None

ULS_GAMMA_F3 specifies the γf3 partial load factors for the Ultimate Limit State. The list is ordered as follows:-

1. HA alone (combination 1)
2. HA alone (combination 2)
3. HA/HB or HB alone (combination 1)
4. HA/HB or HB alone (combination 2)

For example:

```
ULS_GAMMA_F3 = 1.10, 1.10, 1.10, 1.10
```

SLS_GAMMA_F3 γ_{f3} ULS PARTIAL LOAD FACTOR

Value: List of positive real values
Default: None

SLS_GAMMA_F3 specifies the γf3 partial load factors for the Ultimate Limit State. The list is ordered as follows:-

1. HA alone (combination 1)
2. HA alone (combination 2)
3. HA/HB or HB alone (combination 1)
4. HA/HB or HB alone (combination 2)

For example:

```
SLS_GAMMA_F3 = 1.10, 1.10, 1.10, 1.10
```

USE_GAMMA_F3 INDICATES USE OF γ_{f3} LOAD FACTOR

Value: Boolean
Default: FALSE

USE_GAMMA_F3 specifies whether or not to apply the γ_{f3} partial load factors to the overall effect. If TRUE, γ_{f3} is applied, if FALSE, γ_{f3} is not applied.

For example:

```
USE_GAMMA_F3 = TRUE
```

Only the first character is significant, and case is ignored, therefore the next three examples all have the same effect:

```
USE_ GAMMA_F3 = F
USE_ GAMMA_F3 = False
USE_ GAMMA_F3 = f
```

HA_ALTERNATIVE Vehicle To be used as ALTERNATIVE HA loading

Value: list of strings
Default: none

HA_ALTERNATIVE defines a list of vehicles to be tried as an alternative to HA loading. When loading an HA lane, Autoloader tries to place vehicles from this list within the lane, and if the effect is greater than HA loading, uses that vehicle instead. Each vehicle must be specified in the Autoloader vehicle library. For more information Autoloader Vehicle Library

For example:

```
HA_ALTERNATIVE = T44
```

HBUNITS Number of HB Units

Value: positive real value
Default: as per standard used

HBUNITS specifies the number of HB Units to use. In cases where an abnormal vehicle uses HB Units, the values of the weights of the axles are multiplied by the number of HB Units. See Autoloader Vehicle Library for more details.

For example:

```
HBUNITS = 30
```

LANE_WIDTH_TABLE Notional lane widths

Value: list of sets of three real values
Default: as per design standard used
Units: metres

LANE_WIDTH_TABLE permits altering the relationship between carriageway width and the number of notional lanes. Each 3 values are taken as a lower limit, an upper limit and a number of notional lanes. After calculating the carriageway width, Autoloader works its way down the table, checking the calculated width against the values in the table. If the width is greater than or equal to the lower limit, and less than or equal to the upper limit, then the number of notional lanes is given by the third value.

For example:

```
LANE_WIDTH_TABLE = 5, 7, 2,
                      7, 10.5, 3,
                      10.5, 14, 4,
                      14, 18.25, 5,
                      18.25, 22, 6
```

In this case a width of 12m will have 4 notional lanes of width 3m each.

ONEWAY One Way Traffic

Value: Boolean
Default: FALSE

BD 37/88 specifies that a bridge carrying traffic in one direction only would result in the doubling of the N value in the calculation of the HA lane factors (ref. BD 37/88, Table 14, Note 2). If **ONEWAY** is **TRUE**, then the value of N used in the table is doubled. This only applies to loaded lengths in excess of 50m.

For example:

```
ONEWAY = TRUE
```

Only the first character is significant, and case is ignored, therefore the next three examples all have the same effect:

```
ONEWAY = F
ONEWAY = False
ONEWAY = f
```

SLS_GAMMA SLS partial load factors

Value: list of positive real values
Default: as per design loading standard used

SLS_GAMMA specifies the partial load factors for the Serviceability Limit State. The list is ordered as follows:

1. HA alone (combination 1)
2. HA alone (combination 2)
3. HB with HA or HB alone (combination 1)
4. HB with HA or HB alone (combination 2)

For example:

```
SLS_GAMMA = 1.4, 1.3, 1.2, 1.1
```

ULS_GAMMA ULS partial load factors

Value: list of positive real values

Default: as per design loading standard used

ULS_GAMMA specifies the partial load factors for the Ultimate Limit State. The list is ordered as follows:

1. HA alone (combination 1)
2. HA alone (combination 2)
3. HB with HA or HB alone (combination 1)
4. HB with HA or HB alone (combination 2)

For example:

```
ULS_GAMMA = 1.4, 1.3, 1.2, 1.1
```

USE_HA Indicates USE OF HA LOADING

Value: Boolean
Default: TRUE

USE_HA specifies whether to apply the HA loading (which includes KEL loading).

For example:

```
USE_HA = FALSE
```

Only the first character is significant, and case is ignored, therefore the next three examples all have the same effect:

```
USE_HA = F
USE_HA = False
USE_HA = f
```

USE_HB Indicates USE OF abnormal vehicle

Value: Boolean
Default: If VEHICLES has been specified, TRUE, otherwise FALSE

USE_HB specifies whether to apply an abnormal vehicle. If TRUE, the abnormal vehicles are applied to the carriageway. If FALSE, then HA only loading is applied.

For example:

```
USE_HB = FALSE
```

Only the first character is significant, and case is ignored, therefore the next three examples all have the same effect:

```
USE_HB = F
USE_HB = False
USE_HB = f
```

HK Parameters

HA_ALTERNATIVE Vehicle To be used as ALTERNATIVE HA loading

Value: list of strings
Default: none

HA_ALTERNATIVE defines a list of vehicles to be tried as an alternative to HA loading. When loading an HA lane, Autoloader tries to place vehicles from this list within the lane, and if the effect is greater than HA loading, uses that vehicle instead. Each vehicle must be specified in the Autoloader vehicle library. For more information Autoloader Vehicle Library

For example:

```
HA_ALTERNATIVE = T44
```

HBUNITS Number of HB Units

Value: positive real value
Default: as per standard used

HBUNITS specifies the number of HB Units to use. In cases where an abnormal vehicle uses HB Units, the values of the weights of the axles are multiplied by the number of HB Units. See Autoloader Vehicle Library for more details.

For example:

```
HBUNITS = 30
```

LANE_WIDTH_TABLE Notional lane widths

Value: list of sets of three real values
Default: as per design standard used
Units: metres

LANE_WIDTH_TABLE permits altering of the relationship between carriageway width and the number of notional lanes. Each 3 values are taken as a lower limit, an upper limit and a number of notional lanes. After calculating the carriageway width, Autoloader works its way down the table, checking the calculated width against the values in the table. If the width is greater than or equal to the lower limit, and less than or equal to the upper limit, then the number of notional lanes is given by the third value.

For example:

```
LANE_WIDTH_TABLE = 5, 7, 2,
                   7, 10.5, 3,
```

```
10.5, 14, 4,
14, 18.25, 5,
18.25, 22, 6
```

In this case a width of 12m will have 4 notional lanes of width 3m each.

ONEWAY One Way Traffic

Value: Boolean
Default: FALSE

HK specifies that a bridge carrying traffic in one direction only would result in the doubling of the N value in the calculation of the HA lane factors (ref. HK, Table 4.2.8, Note 2). If **ONEWAY** is **TRUE**, then the value of N used in the table is doubled. This only applies to loaded lengths in excess of 50m.

For example:

```
ONEWAY = TRUE
```

Only the first character is significant, and case is ignored, therefore the next three examples all have the same effect:

```
ONEWAY = F
ONEWAY = False
ONEWAY = f
```

SLS_GAMMA SLS partial load factors

Value: list of positive real values
Default: as per design loading standard used

SLS_GAMMA specifies the partial load factors for the Serviceability Limit State. The list is ordered as follows:

1. HA alone (combination 1)
2. HA alone (combination 2)
3. HB with HA or HB alone (combination 1)
4. HB with HA or HB alone (combination 2)

For example:

```
SLS_GAMMA = 1.4, 1.3, 1.2, 1.1
```

ULS_GAMMA ULS partial load factors

Value: list of positive real values
Default: as per design loading standard used

ULS_GAMMA specifies the partial load factors for the Ultimate Limit State. The list is ordered as follows:

1. HA alone (combination 1)
2. HA alone (combination 2)
3. HB with HA or HB alone (combination 1)
4. HB with HA or HB alone (combination 2)

For example:

```
ULS_GAMMA = 1.4, 1.3, 1.2, 1.1
```

USE_HA IndicateS USE OF HA LOADING

Value: Boolean
Default: TRUE

USE_HA specifies whether to the HA loading (which includes KEL loading).is to be applied.

For example:

```
USE_HA = FALSE
```

Only the first character is significant, and case is ignored, therefore the next three examples all have the same effect:

```
USE_HA = F
USE_HA = False
USE_HA = f
```

USE_HB IndicateS USE OF abnormal vehicle

Value: Boolean
Default: If VEHICLES has been specified, TRUE, otherwise FALSE

USE_HB specifies whether to apply an abnormal vehicle. If TRUE, the abnormal vehicles are applied to the carriageway. If FALSE, then HA only loading is applied.

For example:

```
USE_HB = FALSE
```

Only the first character is significant, and case is ignored, therefore the next three examples all have the same effect:

```
USE_HB = F
USE_HB = False
USE_HB = f
```

JKR Parameters

JKR_TYPE CONTROLLED MOVEMENT

Value: *controlled* or *uncontrolled*
Default: *uncontrolled*

JKR_TYPE specifies either controlled or uncontrolled vehicle movement, as defined in the JKR standard. This option has no effect in any of the other standards. If an invalid value is specified, Autoloader gives a warning and uses the default value.

For example:

```
JKR_TYPE = controlled
```

Only the first character is significant, and case is ignored, therefore the next three examples all have the same effect:

```
JKR_TYPE = U
JKR_TYPE = Uncontrolled
JKR_TYPE = u
```

LANE_INCR Increment for moving lanes

Value: positive real value
Default: 0.25m
Units: metres

LANE_INCR specifies the increment interval used when moving lanes across the carriageway within the JKR standard. The smaller the increment, the more accurately the vehicle loading on the carriageway is positioned. The larger the increment, the more quickly Autoloader runs. An appropriate value must be chosen from experience. If an invalid value is specified, Autoloader gives a warning and uses the default value.

For example:

```
LANE_INCR = 3.0
```

LANE_WIDTH LTAL LANE WIDTH

Value: positive real value
Default: 2.5m
Units: metres

LANE_WIDTH specifies the LTAL lane width for use with the JKR standard. If an invalid value is specified, Autoloader gives a warning and uses the default value.

For example:

```
LANE_WIDTH = 3.0
```

LTAL_ALTERNATIVE Vehicle to be used as ALTERNATIVE LTAL loading

Value: list of strings
Default: none

LTAL_ALTERNATIVE defines a list of vehicles to be tried as an alternative to LTAL loading. When loading an LTAL lane, Autoloader tries to place vehicles from this list within the lane, and if the effect is greater than LTAL loading, uses that vehicle instead. Each vehicle must be specified in the Autoloader vehicle library. For more information Autoloader Vehicle Library

For example:

```
LTAL_ALTERNATIVE = T44
```

PED_LOAD Value of Pedestrian load

Value: positive real number
Default: as per design standard used
Units: kN/m^2

PED_LOAD specifies an alternate value for the Pedestrian Load.

For example:

```
PED_LOAD = 10
```

SLS_GAMMA SLS partial load factors

Value: list of positive real values
Default: as per design loading standard used

SLS_GAMMA allows the user to specify the partial load factors for the Serviceability Limit State. The list is ordered as follows:

1. LTAL alone (combination 1)
2. LTAL alone (combination 2)
3. SV with LTAL or SV alone (combination 1)
4. SV with LTAL or SV alone (combination 2)

For example:

```
SLS_GAMMA = 1.4, 1.3, 1.2, 1.1
```

SVUNITS Number of SV Units

Value: positive real value
Default: as per standard used

SVUNITS specifies the number of HB Units to use. In cases where an abnormal vehicle uses SV Units, the values of the weights of the axles are multiplied by the number of SV Units. See Autoloader Vehicle Library for more details. SVUNITS and HBUNITS are synonymous.

For example:

```
SVUNITS = 30
```

ULS_GAMMA ULS partial load factors

Value: list of positive real values
Default: as per design loading standard used

ULS_GAMMA specifies the partial load factors for the Ultimate Limit State. The list is ordered as follows:

1. LTAL alone (combination 1)
2. LTAL alone (combination 2)
3. SV with LTAL or SV alone (combination 1)
4. SV with LTAL or SV alone (combination 2)

For example:

```
ULS_GAMMA = 1.4, 1.3, 1.2, 1.1
```

USE_LTAL IndicateS USE OF LTAL LOADING

Value: Boolean
Default: TRUE

USE_LTAL specifies whether to apply the LTAL loading (which includes KEL loading).

For example:

```
USE_LTAL = FALSE
```

Only the first character is significant, and case is ignored, therefore the next three examples all have the same effect:

```
USE_LTAL = F
USE_LTAL = False
USE_LTAL = f
```

USE_SV IndicateS USE OF abnormal vehicle

Value: Boolean
Default: If VEHICLES has been specified, TRUE, otherwise FALSE

USE_SV specifies whether an abnormal vehicle is to be applied. If TRUE, the abnormal vehicles are applied to the carriageway. If FALSE, then LTAL only loading is applied. This option has the same effect as USE_HB.

For example:

```
USE_SV = FALSE
```

Only the first character is significant, and case is ignored, therefore the next three examples all have the same effect:

```
USE_SV = F
USE_SV = False
USE_SV = f
```

RAIL & RU Parameters

HBUNITS Number of HB Units

Value: positive real value
Default: as per standard used

HBUNITS specifies the number of HB Units to use. In cases where an abnormal vehicle uses HB Units, the values of the weights of the axles are multiplied by the number of HB Units. See Autoloader Vehicle Library for more details.

For example:

```
HBUNITS = 30
```

SLS_GAMMA SLS partial load factors

Value: list of positive real values
Default: as per design loading standard used

SLS_GAMMA allows the user to specify the partial load factors for the Serviceability Limit State. The list is ordered as follows:

1. UDL alone (combination 1)
2. UDL alone (combination 2)
3. RU/ RAIL with UDL or RU/ RAIL alone (combination 1)
4. RU/RAIL with UDL or RU/RAIL alone (combination 2)

For example:

```
SLS_GAMMA = 1.4, 1.3, 1.2, 1.1
```

ULS_GAMMA ULS partial load factors

Value: list of positive real values
Default: as per design loading standard used

ULS_GAMMA allows the user to specify the partial load factors for the Ultimate Limit State. The list is ordered as follows:

1. UDL alone (combination 1)
2. UDL alone (combination 2)
3. RU/RAIL with UDL or RU/ RAIL alone (combination 1)
4. RU/RAIL with UDL or RU/RAIL alone (combination 2)

For example:

```
ULS_GAMMA = 1.4, 1.3, 1.2, 1.1
```

USE_RU IndicateS USE OF abnormal vehicle

Value: Boolean
Default: If VEHICLES has been specified, TRUE, otherwise FALSE

USE_RU specifies whether to apply an abnormal vehicle. If TRUE, the abnormal vehicles are applied to the carriageway. If FALSE, then UDL only loading is applied.

For example:

```
USE_RU = FALSE
```

Only the first character is significant, and case is ignored, therefore the next three examples all have the same effect:

```
USE_RU = F
USE_RU = False
USE_RU = f
```

USE_UDL IndicateS USE OF UDL LOADING

Value: Boolean
Default: TRUE

USE_UDL specifies whether to apply the UDL loading.

For example:

```
USE_UDL = FALSE
```

Only the first character is significant, and case is ignored, therefore the next three examples all have the same effect:

```
USE_UDL = F
USE_UDL = False
USE_UDL = f
```

Chapter 4 Error Messages

Whilst using Autoloader, a number of error messages may be encountered. The following section is a description of these error messages and, where appropriate, a workaround or fix is provided.

Autoloader breaks errors down into several types. These are:

- **Failed To Open A File**
- **Incorrect Data File**
- **Invalid Data**
- **Invalid Interpolation Line**
- **Memory Allocation Failure**
- **Missing Data File**
- **Value Not There**
- **Internal Error**

Failed To Open A File

This indicates that Autoloader has attempted to open a new file for writing, and failed. Files which Autoloader will attempt to open for writing are, for example, U1, U2, S1 and S2.

This error could be generated due to an incorrect pathname being specified, access rights to the directory being incorrect, etc.

If Autoloader encounters errors such as these, it will print a **`FAILED_OPENFILE`** error, and abort.

Incorrect Data File

This error indicates that Autoloader is reading a file which is not in the expected format. The format of the Autoloader input file and parameter file are set out in the

Reference Manual. The format of the Autoloader Vehicle Library is set out in the Autoloader Vehicle Library, and the format of the Influence Surface files are defined in LUSAS documentation. Typical mistakes include:

- placing a full stop '.' instead of a comma ',' in a field;
- specifying an unknown variable name;
- specifying an invalid value for a field within a data file, such as a string when a number is required;
- not enough nodes or elements are specified within the Influence surface file, e.g. an element refers to a non-existent node.

If Autoloader encounters errors such as these, it will print an **INCORRECT_DATAFILE** error, and abort.

Invalid Data

This error indicates that the data read was in the correct format, but was not valid. For instance, when examining kerb points specified in the Autoloader input file, if any of the points are coincident, then this would generate an INVALID_DATA error (the carriageway width or the centre or curvature cannot be calculated if any kerb points are coincident).

If Autoloader encounters errors such as these, it will print an **INVALID_DATA** error, and abort.

Invalid Interpolation Line

This occurs when Autoloader attempts to create an interpolation line along the carriageway, and the line will does cross the finite element deck at any time. Workarounds include defining more elements or changing the kerb positions.

If Autoloader encounters errors such as these, it will print an **INVALID_INTERP_LINE** error, and abort.

Memory Allocation Failure

This occurs when Autoloader attempts to access more memory and fails. Autoloader will abort, printing how much memory is already allocated and how much it was attempting to allocate. The permanent solution to this is to install more memory, but workarounds include closing any other applications which may be running, not defining as many elements or nodes or using less load cases in the URF.

If Autoloader encounters errors such as these, it will print a **MALLOC_FAILURE** error, and abort.

Missing Data File

This indicates that Autoloader attempted to open a file for reading, and failed. This error message will be preceded by the name of the file. There are four files which Autoloader requires in order to function correctly:

- Autoloader input file;
- Autoloader parameter file, when specified with PARAMETER_FILE;
- Autoloader Vehicle Library, when required;
- File containing the influence surface.

This error could be generated due to the file not existing, an incorrect pathname being specified, access rights to the file being incorrect, etc.

If Autoloader fails to open any of these files, it will print a **`MISSING_DATAFILE`** error, and abort.

Value Not There

This occurs when Autoloader cannot locate a value required, such as a loading intensity for a particular loaded length. This can occur when the user has specified a different HA UDL curve which does not cover a relevant section or if a carriageway width is not covered in the LANE_WIDTH_TABLE.

If Autoloader encounters errors such as these, it will print a **`NOT_THERE`** error, and abort.

Internal Error

This occurs when an internal Autoloader error occurs. If this error does occur, then please forward a copy of relevant files and description to your technical support representative.

If Autoloader encounters errors such as these, it will print a **`INTERNAL_ERROR`** error, and abort.

Chapter 5 Autoloader Output

Autoloader outputs loading **.vbs** files, verification **.ver** and summary **.sum** files for each influence surface(s).

AASHTO

When the loading standard is AASHTO, fourteen files are created with suffixes LF*x* and SL*x*. These contain the Load factor design combinations and Service load combination respectively.

The loadcase .vbs files are called test_lf*x*.vbs and test_sl*x*.vbs respectively, where *x* is replaced by 1, 1a, 3, 4, 6, 8 and 10. See Output Prefix.

AUSTROADS

When the loading standard is AUSTROADS, two files are created with suffixes ULS and SLS. These contain the Ultimate Limit State combination and Serviceability Limit State combination respectively.

The loadcase vbs files are called test_uls.vbs and test_sls.vbs and contain the ULS and SLS combinations respectively.

BD 21/97

When the loading standard is BD 21/97, one file is created with suffixes AS. This contains the Assessment loading.

The loadcase vbs file is called test_as.vbs and contains the assessment load combinations.

BD 37/88, JKR, RU, RAIL and HK

When the loading standard is one of the above, four files are created one for each of the Ultimate Limit State (ULS) combinations 1 and 2 and one for each of the Serviceability Limit State (SLS) combinations 1 and 2. These files have the suffix u1, u2, s1 and s2.

The loadcase vbs files are called test_u1.vbs, test_u1.vbs, test_s1.vbs and test_s2.vbs, and contain the ULS 1, ULS 2, SLS 1 and SLS 2 combinations respectively.

File Structure

Each file is structured as follows:

- Definitions of loads with greatest effect for the relevant combination.
- Positions of loads with greatest effect for the relevant combination.
- Combination definition containing above loads.

All loading definitions, positions of loads and combination definitions are output to the vbs files. Only relevant loads appear in the file. For instance, if Autoloader is directed to use only HB loading, then only HB loading will be calculated and be written to the file.

Chapter 6 Autoloader Vehicle Library

When evaluating critical loading patterns containing abnormal vehicles, Autoloader must be able to read the Autoloader Vehicle Library. This contains a number of common vehicle definitions, and can be extended by the user when required. This section explains the format of the library file, and how new vehicles are defined.

The library contains a series of vehicle definitions, all of which have the same basic format:

```
[name]
{
definition
}
```

where **`name`** is replaced by the name of the vehicle, and **`definition`** is replaced by a series of parameters, defined as follows. Parameters are in the same format as normal Autoloader parameters, as defined in the Reference Manual. For examples for vehicle definitions, see Appendix B Example Autoloader Vehicle Definitions.

Note.

Units specified with UNITS_INPUT have no effect in this file. All values must be in either metres or kiloNewtons.

AXLENUM Number of axles

Value: A positive integer

AXLENUM specifies the number of axles on the vehicle. This value must be the same as the number of weights (**AXLEWEIGHT**), the number of axle positions (**AXLEPOS**) and the number of wheel definitions (**WHEELS**).

For example:

```
AXLENUM = 4
```

AXLEPOS Position Of Axles

Value: list of real numbers
Units: metres

AXLEPOS specifies the position of each axle along the vehicle. The values are in metres, and can be specified in either ascending or descending order. The values do not have to be relative to the centre point, but can be relative to any point.

For example, the following two lines have the same effect:

```
AXLEPOS = -4.8, -3.0, 3.0, 4.8
AXLEPOS = 0.0, 1.8, 7.8, 9.6
```

AXLEWEIGHT Weight Of Axles

Value: list of positive real numbers
Units: kN

AXLEWEIGHT specifies the individual weights of each axle. If **HBUNITS** is **TRUE**, then all non-tractor axles are multiplied by the number of HB Units. The weight of each axle is divided equally between each wheel on the axle.

For example:

```
AXLEWEIGHT = 10, 10, 12.5, 12.5
```

BEHIND Unloaded Length Behind Vehicle

Value: positive real value
Default: 10m
Units: metres

BEHIND defines the length behind the vehicle which will not have any other loading applied.

For example:

```
BEHIND = 25.0
```

FRONT Unloaded Length In Front Of Vehicle

Value: positive real value
Default: 10m
Units: metres

FRONT defines the length in front of the vehicle which will not have any other loading applied. For example:

```
FRONT = 25.0
```

HBUNITS Multiply axle weights by HB Units

Value: Boolean
Default: FALSE

If **HBUNITS** is **TRUE**, non-tractor axle weights are multiplied by the number of HB Units. If **HBUNITS** is **FALSE**, then they are not.

For example:

```
HBUNITS = TRUE
```

LENGTH Overall Length Of Vehicle

Value: positive real value greater than or equal to the maximum axle length
Default: maximum axle length
Units: metres

LENGTH defines the overall length of the vehicle so that the overhang of the vehicle can be determined, and HA loading, etc. be placed correctly. For vehicles with variable spans and vehicles with variable axles, the length is adjusted. In this case, the length specified is the length when the vehicle is in the configuration specified in the library. For example:

```
LENGTH = 10
```

TRACTOR Axles that make up the tractor

Value: list of positive integers

TRACTOR is a list of axles that define the tractor (as opposed to the trailer) of the vehicle. This has two effects. Firstly, when **HBUNITS** is **TRUE**, the axles defined as the tractor are NOT multiplied by the number of HB Units.

For example, on a 23 axle vehicle with the following **TRACTOR** definition:

```
TRACTOR = 1, 2, 3
HBUNITS = TRUE
```

Axles 4-23 inclusive would have their weights multiplied by the number of HB Units.

Secondly, when the vehicle has a variable number of axles, tractor axles are not varied. For example, on a 23 axle vehicle with the following definition:

```
TRACTOR = 1, 2, 3
VARSTEP = 1
```

The effect would be calculated with axles 1, 2, 3, 4 then 1, 2, 3, 4, 5, then 1, 2, 3, 4, 5, 6, etc. The tractor can be at either end of the vehicle.

VARSPACE Define a variable axle spacing

Value: list of two integers followed by appropriate number of real numbers
Units: metres

VARSPACE allows the user to vary the spacing between axles. Specifying **VARSPACE** in a vehicle definition defines the vehicle as a variable span vehicle. If **VARSTEP** has been previously specified, then Autoloader returns an error. The format is:

- number of hspan to be varied;
- number of lengths to be considered;
- lengths

The number of the span is counted from the first axle specified. If the number of lengths is 0, then the next two values are defined as the minimum spacing length, the maximum spacing length and a value to increase the spacing by each time. For example:

```
VARSPACE = 2, 0, 6.0, 26.0, 5.0
```

This specifies that span 2 (between axles 2 and 3) will have the effect calculated with span lengths of 6, 11, 16, 21 and 26 metres.

If the number of lengths is greater than 0, the number specified is taken as the number of values that follow, and each value is tried separately.

For example:

```
VARSPACE = 2, 5, 6.0, 11.0, 16.0, 21.0, 26.0
```

has exactly the same effect as the above **VARSPACE** definition.

VARSTEP Define a number of variable axles

Value: A positive integer

VARSTEP defines the number of axles to remove each time to calculate the worst effect. Specifying **VARSTEP** in a vehicle definition defines the vehicle as a variable axle vehicle. If **VARSPACE** has previously been specified, then Autoloader returns an error.

For example:

```
VARSTEP = 2
```

WHEELS Position Of Wheels On Axles

Value: list of integers followed by appropriate real numbers
Units: metres

WHEELS defines the wheel positions on each axle relative to the centreline of the vehicle. The list is a series of integers followed by that number of wheel positions.

For example:

```
WHEELS = 4, -1.5, -0.5, 0.5, 1.5,
    4, -1.5, -0.5, 0.5, 1.5,
    2, -0.5, 0.5,
    2, -0.5, 0.5
```

This defines wheel positions for 4 axles, the first and second having 4 wheels each, the third and fourth having 2 wheels. The weight of each axle is divided equally between each wheel.

WIDTH Overall Width Of Vehicle

Value: positive real value greater than or equal to the maximum axle width
Default: maximum axle width
Units: metres

WIDTH defines the overall width of the vehicle so that the overhang of the vehicle can be determined, and HA loading, etc. be placed correctly.

For example:

```
WIDTH = 3.5
```

Appendices

Appendix A - Influence Surfaces and How to Use Them

This section provides an explanation of the influence surface theory underlying Autoloader and the Bridge Design Loading Standards, and provides the engineer with a guide as to which influence surfaces should be used when assessing a bridge.

An influence surface relates the value of a parameter at a given point in the structure to the position of a unit point load anywhere on that structure. An influence line is a 2-dimensional form of the 3-dimensional surface, and may be considered as a section through the influence surface or an influence surface for a line beam.

Maxwell's reciprocal theorem states:

> *"The deflection at point A due to the application of a load at point B is the same as the deflection at point B due to the application of the same load at point A".*

The theorem can be applied to both single span and multiple span structures and is equally applicable to rotations and moments.

An influence surface can be generated by applying a point load at a number of positions and calculating the deflection due to the applied loads. This approach however is computationally expensive. Using Maxwell's Theorem influence surfaces are generated simply and quickly by applying a single distortion at each point on the structure to be considered in the design. From the influence surfaces Autoloader calculates the loading pattern which generates the worst effect at each design point.

The engineer needs to decide which influence surfaces to generate. This is an area in which this manual can be of limited help, and the engineer must rely on sound engineering judgement. However, we give a few pointers here that may help. For example for a multiple span bridge with similar spans, a good start would be to generate influence surfaces for:

- bending moments at mid-span;
- bending moments at supports;
- shear at supports;
- bending moments at any section changes / splice connections;
- shear at any section changes / splice connections;

- support reactions;
- support rotations.

This definition should be repeated for any dissimilar spans. Further investigation may be required, depending on the results of these investigations.

Appendix B - Sample Autoloader Vehicles

BD 37/88 abnormal vehicle (HB)

```
[HB]
{
 AXLENUM = 4
 AXLEWEIGHT = 10, 10, 10, 10
 AXLEPOS = -4.8, -3.0, 3.0, 4.8
 WHEELS =    4, -1.5, -0.5, 0.5, 1.5,
             4, -1.5, -0.5, 0.5, 1.5,
             4, -1.5, -0.5, 0.5, 1.5,
             4, -1.5, -0.5, 0.5, 1.5
 WIDTH = 3.5
 LENGTH = 10.0
 FRONT = 25.0
 BEHIND = 25.0
 VARSPACE = 2, 5, 6.0, 11.0, 16.0, 21.0, 26.0
 HBUNITS = TRUE
}
```

JRK Special Vehicle (SV)

```
[SV]
{
      AXLENUM = 23
      TRACTOR = 21, 22, 23
      VARSTEP = 1
      AXLEWEIGHT = 10, 10, 10, 10, 10, 10, 10, 10,
                   10, 10, 10, 10, 10, 10, 10,
                   120, 120, 60
      AXLEPOS =    -19.5, -18, -16.5, -15, -13.5,
                   -12, -10.5, -9, -7.5, -6,
                   -4.5, -3, -1.5, 0, 1.5,
                   3, 4.5, 6, 7.5, 9,
                   14.5, 15.850, 19.250
      WIDTH = 3.5
      LENGTH = 40.0
      FRONT = 25.0
      BEHIND = 25.0
```

```
    WHEELS =    4, -1.2, -0.5, 0.5, 1.2,
                4, -1.2, -0.5, 0.5, 1.2,
                4, -1.2, -0.5, 0.5, 1.2,
                4, -1.2, -0.5, 0.5, 1.2,
                4, -1.2, -0.5, 0.5, 1.2,
                4, -1.2, -0.5, 0.5, 1.2,
                4, -1.2, -0.5, 0.5, 1.2,
                4, -1.2, -0.5, 0.5, 1.2,
                4, -1.2, -0.5, 0.5, 1.2,
                4, -1.2, -0.5, 0.5, 1.2,
                4, -1.2, -0.5, 0.5, 1.2,
                4, -1.2, -0.5, 0.5, 1.2,
                4, -1.2, -0.5, 0.5, 1.2,
                4, -1.2, -0.5, 0.5, 1.2,
                4, -1.2, -0.5, 0.5, 1.2,
                4, -1.2, -0.5, 0.5, 1.2,
                4, -1.2, -0.5, 0.5, 1.2,
                4, -1.2, -0.5, 0.5, 1.2,
                4, -1.2, -0.5, 0.5, 1.2,
                4, -1.2, -0.5, 0.5, 1.2,
                2, -1.45, 1.45,
                2, -1.45, 1.45,
                2, -1.1, 1.1
    HBUNITS = TRUE
}
```

AUSTROADS HEAVY LOADING PLATFORM 400

```
[HLP400]
{
FRONT = 0.0
BEHIND = 0.0
AXLEPOS = 0.0, 1.8, 3.6, 5.4, 7.2, 9.0, 10.8, 12.6, 14.4,
          16.2, 18.0, 19.8, 21.6, 23.4, 25.2, 27.0
AXLEWEIGHT = 250, 250, 250, 250, 250, 250, 250, 250,
                250, 250, 250, 250, 250, 250, 250, 250
WIDTH = 4.5
LENGTH = 27
VARSPACE = 8, 11, 1.8, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15
WHEELS =  8, -2.15, -1.85, -1.25, -0.95, 0.95, 1.25, 1.85, 2.15,
          8, -2.15, -1.85, -1.25, -0.95, 0.95, 1.25, 1.85, 2.15,
          8, -2.15, -1.85, -1.25, -0.95, 0.95, 1.25, 1.85, 2.15,
          8, -2.15, -1.85, -1.25, -0.95, 0.95, 1.25, 1.85, 2.15,
          8, -2.15, -1.85, -1.25, -0.95, 0.95, 1.25, 1.85, 2.15,
          8, -2.15, -1.85, -1.25, -0.95, 0.95, 1.25, 1.85, 2.15,
          8, -2.15, -1.85, -1.25, -0.95, 0.95, 1.25, 1.85, 2.15,
          8, -2.15, -1.85, -1.25, -0.95, 0.95, 1.25, 1.85, 2.15,
          8, -2.15, -1.85, -1.25, -0.95, 0.95, 1.25, 1.85, 2.15,
          8, -2.15, -1.85, -1.25, -0.95, 0.95, 1.25, 1.85, 2.15,
          8, -2.15, -1.85, -1.25, -0.95, 0.95, 1.25, 1.85, 2.15,
          8, -2.15, -1.85, -1.25, -0.95, 0.95, 1.25, 1.85, 2.15,
          8, -2.15, -1.85, -1.25, -0.95, 0.95, 1.25, 1.85, 2.15,
```

```
            8, -2.15, -1.85, -1.25, -0.95, 0.95, 1.25, 1.85, 2.15,
            8, -2.15, -1.85, -1.25, -0.95, 0.95, 1.25, 1.85, 2.15,
            8, -2.15, -1.85, -1.25, -0.95, 0.95, 1.25, 1.85, 2.15
AXLENUM = 16
HBUNITS = NO
}
```

Appendix C - Quick Reference Guide

Required Parameters

CWSHAPE	Carriageway Shape
CWNUM	Number of Carriageways
KBPOS	Kerb Positions
CWDIR	Carriageway Direction
FILE	Input Filename
DECK	LUSAS deck name

Optional Parameters

CLINCR	Increment for Influence Lines
DLALLOWANCE	DYNAMIC LOAD ALLOWANCE
INFL	INFLUENCE AREA TO BE LOADED
KEL	Value of Knife Edge Load
LANE_FACTORS	LANE BETA FACTORS
LANE_MOD_FACTORS	LANE MODIFICATION FACTORS
LOADCASES	LOAD CASES TO PROCESS
LTINCR	Increment FOR longitudinal movement
KELNUM	NUMBER OF KEL'S TO APPLY
PARAMETER_FILE	Extra Parameter File
STANDARD	Design Standard to be applied
TRINCR	Increment for TRANSVERSE movement
UDL_CURVE	UDL Loading Intensity Curve
UDL_LIMIT	LOADED LENGTH LIMIT
UNITS_INPUT	INPUT UNITS
UNITS_OUTPUT	OUTPUT UNITS
USE_CUSP	IndicateS USE OF CUSPING
USE_KEL	IndicateS USE OF KEL’S
VEHDIR	Vehicle Direction
VEHICLE_FILE	Autoloader vehicle library
VEHICLES	Vehicle Name(s)

AASHTO PARAMETERS

UDL_ALTERNATIVE	Vehicle to be used as ALTERNATIVE TO UDL loading
SL_GAMMA	SERVICE LOAD partial load factors

LF_GAMMA	LOAD FACTOR partial load factors

AUSTROADS PARAMETERS

AUST_TYPE	TYPE OF ROAD (AUSTROADS)
HLP_TRRANGE	TRANSVERSE RANGE OF ABNORMAL VEHICLE
L44_ALTERNATIVE	Vehicle to be used as ALTERNATIVE L44 loading
LANE_INCR	Increment for moving lanes
LANE_WIDTH	L44 LANE WIDTH
SLS_GAMMA	SLS partial load factors
TRAFFIC_LANES	SPECIFY MARKED TRAFFIC LANES
ULS_GAMMA	ULS partial load factors
USE_HLP	IndicateS USE OF abnormal vehicle
USE_L44	IndicateS USE OF L44/T44 LOADING

BD 21/97 PARAMETERS

ADJFACTOR_CURVE	Adjustment factor curve
HA_ALTERNATIVE	Vehicle To be used as ALTERNATIVE HA loading
KFACTOR_CURVE	K factor curve
LANE_WIDTH_TABLE	Notional lane widths
SURFACE	QUALITY OF ROAD SURFACE
TRAFFIC_LANES	SPECIFY MARKED TRAFFIC LANES
TRAFFIC_LEVEL	Traffic level
ULS_GAMMA	ULS partial load factors

BD 37/88 PARAMETERS

HA_ALTERNATIVE	Vehicle To be used as ALTERNATIVE HA loading
HB_TRRANGE	TRANSVERSE RANGE OF ABNORMAL VEHICLE
HBUNITS	Number of HB Units
LANE_WIDTH_TABLE	Notional lane widths
ONEWAY	One Way Traffic
SLS_GAMMA	SLS partial load factors
ULS_GAMMA	ULS partial load factors
USE_HA	IndicateS USE OF HA LOADING
USE_HB	IndicateS USE OF abnormal vehicle

HK PARAMETERS

HA_ALTERNATIVE	Vehicle To be used as ALTERNATIVE HA loading
HB_TRRANGE	TRANSVERSE RANGE OF ABNORMAL VEHICLE
HBUNITS	Number of HB Units
LANE_WIDTH_TABLE	Notional lane widths
ONEWAY	One Way Traffic
SLS_GAMMA	SLS partial load factors
ULS_GAMMA	ULS partial load factors

USE_HA	IndicateS USE OF HA LOADING
USE_HB	IndicateS USE OF abnormal vehicle

JKR PARAMETERS

JKR_TYPE	CONTROLLED MOVEMENT
LANE_INCR	Increment for moving lanes
LANE_WIDTH	LTAL LANE WIDTH
LTAL_ALTERNATIVE	Vehicle to be used as ALTERNATIVE LTAL loading
SLS_GAMMA	SLS partial load factors
SV_TRRANGE	TRANSVERSE RANGE OF ABNORMAL VEHICLE
SVUNITS	Number of SV Units
ULS_GAMMA	ULS partial load factors
USE_LTAL	IndicateS USE OF LTAL LOADING
USE_SV	IndicateS USE OF abnormal vehicle

RAIL & RU PARAMETERS

HBUNITS	Number of HB Units
SLS_GAMMA	SLS partial load factors
ULS_GAMMA	ULS partial load factors
USE_RU	INDICATES USE OF abnormal vehicle
USE_UDL	IndicateS USE OF UDL LOADING

Appendix D - Sample LUSAS Modeller Command File

3 span grillage analysis - (3span.cmd)

```
# define deck - straight deck
!...
! Autoloader Example of a 3 span steel composite
continuous
  ! structure
  ! For Example purposes only
! The calculation of Section properties for the grillage
   ! Members is beyond the scope of this example
!...
! Geometry and Modelling
! The structure is composed of 3 equal spans of 50m.
Each
! span comprises 7 longitudinal steel beams equally
spaced
! at 4m, making a total deck width of 24m.
```

```
! A 300mm concrete deck connects the steel beams
transversely
!
! The section properties of the steel beams are as
follows;
! I section; Width Of Top & Bottom Flange        - 0.3m
!             Thickness of Top & Bottom Flanges   - 0.02m
!             Overall Depth Of Beam               - 0.8m
!             Thickness Of Web                    -
0.025m
!             E steel                             -
205Knm-2
!
!             Concrete Slab                       - 0.3m
!             E concrete (Short Term)             -
30Knm-2
!
!....
!....Commence Model generation
MODEL INITIALISE

!....Define Line 1 by coordinates of length 50 m
DEFINE LINE BY_COORDINATES LN=* ...
  X=0 Y=0 Z=0 ...
  X=50 Y=0 Z=0
SET COORDINATES OFF

!...Define mesh 1 comprising 12 number Grillage Elements
!...
DEFINE MESH IMSH=* FEATYP=Line ELTYPE="ENGINEERING
GRILLAGE" ...
       THREED=2D PARABL=LINEAR NMBELM=12 THYS=0 THYE=0
THZS=0 THZE=0

!...Assign mesh 1 to line 1
!...Visualise Mesh Divisions
ASSIGN MESH FEATYP=Line IMSH=1 LN=1
```

```
DRAW MESH VISUALISE SYMBOL=11 SIZE=5.00000 ANGLE=0.0
COLOUR=100.100.100
DRAW FEATURES FN=ACTIVE

!...Define Material 1 containing steel properties and
assign to line 1
!...E=205kNm-2 nu=0.3
DEFINE MATERIAL IMAT=* MATTYP=ELASTIC LPTPF=ISOTROPIC
E=205E6 NU=0.3
ASSIGN MATERIAL FEATYP=Line IMAT=1 LN=1

!...
!...Define a transformation equal to the longitudinal
!...beam spacing 24/6m
!...copy line 1, 6 times using transformation 1, to
!...create 6 additional
!...beams (7 in total). All mesh and Material
!...information will be copied
DEFINE TRANSFORMATION TRANSLATION ITSET=* X=0 Y=24/6
COPY FEATURE_LINE LNINC=* NTIMES=6 ITSET=1 LN=1
SET COORDINATES OFF

!...
!...Visualise the Nodal positions to be created by
mystro
DRAW MESH VISUALISE SYMBOL=11 SIZE=5.00000 ANGLE=0.0
DRAW FEATURES FN=ACTIVE

!...
!...Define Geometric datasets 1 and 2 containing the
!...following properties
!...The effective width of the grillage has been set to
zero as this
!...is only applicable for Wood/Armer analysis
!...                for longitudinal elements
!...        Dataset 1 - Edge Beam    Dataset 2 -
Internal Beam
```

```
!...Area        0.1303                  0.2158
!...Iyy         0.1066e-1         0.1221e-1
!...Izz         0.1066e-1         0.1221e-1
!...Jxx         0.1297e-2         0.2515e-2
!...Sz          0.9696e-1         0.9696e-1
!...Efw         0.0               0.0
!...
DEFINE GEOMETRY IGMP=1 LGTPF=GRILLAGE A=0.1303 …
IYY=.1066e-1 IZZ=.1066e-1 JXX=.1297e-2 SZ=.9696e-1 EFW=0
DEFINE GEOMETRY IGMP=2 LGTPF=GRILLAGE A=0.2158 …
IYY=.1221e-1 IZZ=.1221e-1 JXX=.2515e-2 SZ=.9696e-1 EFW=0

!...
!...Assign Properties to Lines 1 to 7
ASSIGN GEOMETRY FEATYP=Line IGMP=1 LN=1;7
ASSIGN GEOMETRY FEATYP=Line IGMP=2 LN=2;3;4;5;6

!...
!...Erase screen And Draw Mesh
ERASE
DRAW MESH COLOUR=100.100.100 LTYPE=FULL SYMBOL="LETTER
Z"

!...
!...Draw Features
DRAW FEATURES FN=ACTIVE

!...
!...Define transformation between points 1 & 2
DEFINE TRANSFORMATION TRANSLATION BETWEEN_POINTS ITSET=*
PN1=1 PN2=2

!...
!...Copy lines 1t7 twice using transformation 2 creating
the
```

```
!...Longitudinal elements for spans 2 and 3, all
geometric/material
!...and mesh information will be copied
COPY FEATURE_LINE LNINC=* NTIMES=2 ITSET=2
LN=1;2;3;4;5;6;7

!...
!...Draw Mesh
DRAW MESH COLOUR=100.100.100 LTYPE=FULL SYMBOL="LETTER
Z"

!...
!...Define a line between points 1 and 13 to model the
!...transverse elements
DEFINE LINE BY_POINTS LN=* PN=1;13

!...
!...Define mesh 2 comprising 6 number Grillage Elements
DEFINE MESH IMSH=* FEATYP=Line ELTYPE="ENGINEERING
GRILLAGE" …
THREED=2D PARABL=LINEAR NMBELM=6 THYS=0 THYE=0 THZS=0
THZE=0

!...Assign mesh 2 to line 22
!...
ASSIGN MESH FEATYP=Line IMSH=2 LN=22

!...Define Material 2 containing concrete properties
!...and assign to line 22
!...E=30kNm-2 nu=0.15
DEFINE MATERIAL IMAT=* MATTYP=ELASTIC LPTPF=ISOTROPIC
E=30E6 NU=0.15
ASSIGN MATERIAL FEATYP=Line IMAT=2 LN=22

!...
!...Define transformation 3 equal to the transverse
!...beam spacing 50/12
```

```
!...and copy line 22, (transverse beam) 36 times
DEFINE TRANSFORMATION TRANSLATION ITSET=* X=50/12
COPY FEATURE_LINE LNINC=* NTIMES=36 ITSET=3 LN=22

!...
!...Define Geometric datasets 3 and 4 containing the
!...following properties
!...The effective width of the grillage has been set to
zero as this
!...is only applicable for Wood/Armer analysis
!...                 For transverse elements
!...         Dataset 3 - Edge Beam    Dataset 4 -
Internal Beam
!...Area        0.75               1.50
!...Iyy         5.625e-3           11.25e-3
!...Izz         5.625e-3           11.25e-3
!...Jxx         11.25e-3           22.50e-3
!...Sz          0.625                    1.25
!...Efw         0.0                0.0
!...
DEFINE GEOMETRY IGMP=3 LGTPF=GRILLAGE A=0.75 IYY=5.625e-
3 …
IZZ=5.625e-3 JXX=11.25e-3 SZ=0.625 EFW=0
DEFINE GEOMETRY IGMP=4 LGTPF=GRILLAGE A=1.50 IYY=11.25e-
3 …
IZZ=11.25e-3 JXX=22.50e-3 SZ=1.250 EFW=0

!...
!...Select all transverse elements by selecting material
2
!...Note. Longitudinal Beams are material 1
!...       Transverse Beams are material 2
SELECT MATERIAL 2
ERASE
DRAW FEATURES FN=ACTIVE
```

```
!...
!...Assign geometry 3, edge transverse beam to lines 22
and 52
ASSIGN GEOMETRY FEATYP=Line IGMP=3 LN=22;58

!...
!...Remove geometry 3, assign geometry 4
!...to the remaining active elements
REMOVE GEOMETRY 3
ERASE
DRAW FEATURES FN=ACTIVE
ASSIGN GEOMETRY FEATYP=Line LN=ACT IGMP=4

!...
!...Add all features
ADD ALL
ERASE
DRAW FEATURES FN=ACTIVE

!...
!...Define support dataset 1 restrained in the global z
direction
!...Assign support 1 to lines 22;34;46;58
DEFINE SUPPORTS ISUP=* U=0 V=0 W=1 THX=0 THY=0 THZ=0
IFTYPE=ALL
ASSIGN SUPPORT FEATYP=Line ISUP=1 LCID=1 LN=22;34;46;58

!...
!...Define equivalence dataset 1 and assign to all lines
to connect
!...all nodes and remesh model
DEFINE EQUIVALENCE IEQV=* EQTOL=0.100000E-02
ASSIGN EQUIVALENCE FEATYP=Line LN=ALL IEQV=1
DRAW MESH COLOUR=100.100.100 LTYPE=FULL SYMBOL="LETTER
Z"
```

```
!...
!...Verify Geometric assignments by visualising geometry
ERASE
DRAW ASSIGNMENTS IFACE="Line Elements" ATTRIB=GEOMETRY
...
 IGMP1=1 COL1=100.000.000 IGMP2=2 COL2=010.050.100 ...
 IGMP3=3 COL3=100.100.000 IGMP4=4 COL4=000.100.000

!...
!...Annotate Geometric assignments on screen and draw
supports
ANNOTATE ASSIGNMENTS HEIGHT=4.0 SCRCRD=0019.8;0211.8;
DRAW SUPPORT ISUP=ALL VECMAX=6.00000 SIZE=5.00000
COLOUR=020.100.050

!...
!...Verify Material Assignments by visualising materials
ERASE
DRAW ASSIGNMENTS IFACE="Line Elements" ATTRIB=MATERIAL
...
 IMAT1=1 COL1=100.000.000 IMAT2=2 COL2=010.050.100

!...
!...Annotate Material assignments on screen
ANNOTATE ASSIGNMENTS HEIGHT=4.0 SCRCRD=0019.8;0211.8;

!...
!...Transform the model to 3d and redraw the mesh and
supports
TRANSFORM ROTATE ALPHA=-30 BETA=0 GAMMA=-60
ERASE
DRAW MESH COLOUR=100.100.100 LTYPE=FULL SYMBOL="LETTER
Z"
DRAW SUPPORT ISUP=ALL VECMAX=6.00000 SIZE=5.00000
COLOUR=020.100.050

!...
```

```
!...Define Influence Lines
DEFINE INFLUENCE LINE INF=* TYPE=MTHY IAXTYP=GLOBAL_AXES
...
                    LOCCRD=0 IDSDIR=POSITIVE NODE=5 EN=4
...
                    TITLE="1/4 point bending on 1st
span"
DEFINE INFLUENCE LINE INF=* TYPE=MTHY IAXTYP=GLOBAL_AXES
...
                    LOCCRD=0 IDSDIR=POSITIVE NODE=8 EN=7
...
                    TITLE="mid point bending on 1st
span"
DEFINE INFLUENCE LINE INF=* TYPE=MTHY IAXTYP=GLOBAL_AXES
...
                    LOCCRD=0 IDSDIR=POSITIVE NODE=11
EN=10 ...
                    TITLE="3/4 point bending on 1st
span"
DEFINE INFLUENCE LINE INF=* TYPE=MTHY IAXTYP=GLOBAL_AXES
...
                    LOCCRD=0 IDSDIR=POSITIVE NODE=2
EN=85 ...
                    TITLE="bending at first support"
DRAW INFLUENCE LABEL INF=ALL HEIGHT=4.00000 ROTATION=0.0
DRAW INFLUENCE SYMBOL INF=ALL SYMBOL=CIRCLE SIZE=5.00000
ANGLE=0.0
SET LUSAS_OUTPUT NODE DISPLACEMENTS FN=ALL

!...
!...Tabulate data deck
TABULATE INFLUENCE ...
         ACTIVE=0 DEFORM=0 PLTFIL=1 RSTFIL=0 FILINF=1
INF=ALL ...
         FILENAME=3SPAN_

!...
!...Save the current Model as 3span.mdl
MODEL SAVE FILENAME=3span.mdl
```

```
!...
!...Done in LUSAS Modeller
printf("Now you must exit LUSAS Modeller.")
printf("Run LUSAS with 3span_1, 3span_2, 3span_3,
3span_4.")
printf("This produces influence line results files")
printf("Then run Autoloader, by typing ATL 3SPAN")
```

Appendix E - Sample Autoloader Input Files

Straight carriageway - BD 37/88

```
# define deck - straight deck
# two carriageways (4 kerbs), along with direction
#(CWDIR) and position (KBPOS) the following 7 parameters
# are the only ones required - the rest have defaults,
or
# are defined as in the relevant standard
CWSHAPE = Straight
CWNUM = 1
KBPOS = 0.0, 0.0,0.0,12.0
CWDIR = 0.0
FILE=s3str_a1.inf,s3str_a2.inf,s3str_a3.inf,s3str_a4.inf
,s3str_a5.inf,s3str_a6.inf,s3str_a7.inf
OUTPUT_PREFIX=BEND_
SUMMARY=BEND

# can be BD3788 or JKR
STANDARD = BD3788

# vehicles to run along deck, and produce greatest
effect
VEHICLES = HB

# run vehicles in both directions, forward and reverse
VEHDIR = both
```

```
# place where vehicles are defined
VEHICLE_FILE =autoload.vec

# restrict to one vehicle on the deck (NB You can only
# have a maximum of one vehicle per carriageway)
# POSSIBLE_VEHICLES = 1

# 45 units of HB
HBUNITS = 45

# examine negative side of influence surface
INFL = negative

# values that define how accurate autoloader is
# LTINCR - longditudinal increment - how much to move
vehicle
# TRINCR - transverse increment - how much to move
vehicle
# CLINCR - how much to increment when calculating
# influence lines at centres of lanes, etc.
# CLINCR = 0.5
# LTINCR = 0.5
TRINCR = 0.25

# one way as defined in BD 37/88
ONEWAY = FALSE

# load cases, could be comma seperated list such as L1,
L3
# SELECT_CASES = ALL

# Knife Edge Load value of 120 kN
HAKEL = 120
```

```
# whether to use HA loading, HB loading and Knife Edge
Loads
USE_HA = TRUE
USE_KEL = TRUE
USE_HB = TRUE

# define new ULS and SLS gamma values
# ULS_GAMMA = 1.5, 1.3, 1.25, 1.1
# SLS_GAMMA = 1.2, 1.1, 1.0, 1.0
```

CURVED carriageway - BD 37/88

```
CWSHAPE = Curved
CWNUM = 1
KBPOS = 0,500,0,512,52.264,497.261,154.509,475.529
CWDIR = 0.0
FILE=s3cur4.inf,s3cur5.inf
OUTPUT_PREFIX=REAC_
SUMMARY=REAC

# can be BD3788 or JKR
STANDARD = BD3788

# vehicles to run along deck, and produce greatest
effect
VEHICLES = HB

# run vehicles in both directions, forward and reverse
VEHDIR = both

# place where vehicles are defined
VEHICLE_FILE =autoload.vec

# restrict to one vehicle on the deck (NB You can only
# have a maximum of one vehicle per carriageway)
# POSSIBLE_VEHICLES = 1
```

```
# 45 units of HB
HBUNITS = 45

# examine negative side of influence surface
INFL = negative

# values that define how accurate autoloader is
# LTINCR - longditudinal increment - how much to move vehicle
# TRINCR - transverse increment - how much to move vehicle
# CLINCR - how much to increment when calculating influence lines at centres of lanes, etc.
# CLINCR = 0.5
# LTINCR = 0.5
TRINCR = 0.25

# one way as defined in BD 37/88
ONEWAY = FALSE

# load cases, could be comma seperated list such as L1, L3
# SELECT_CASES = ALL

# Knife Edge Load value of 120 kN
HAKEL = 120

# whether to use HA loading, HB loading and Knife Edge Loads
USE_HA = TRUE
USE_KEL = TRUE
USE_HB = TRUE

# define new ULS and SLS gamma values
# ULS_GAMMA = 1.5, 1.3, 1.25, 1.1
# SLS_GAMMA = 1.2, 1.1, 1.0, 1.0
```