ANNEX D

Obtaining the σ-w Curve from the Inverse Analysis of the Notched Beam Response

OBTAINING THE σ-*w* CURVE FROM THE INVERSE ANALYSIS
OF THE NOTCHED BEAM RESPONSE

José Luis Antunes de Oliveira e Sousa, Ravindra Gettu and Bryan E. Barragán

D.1.   Objectives

The main objective of the work is to establish an objective methodology for determining the tensile stress-crack opening (σ-*w*) relation of concrete using the load versus crack mouth opening displacement behaviour of notched beam specimens. Comparisons are made between the σ-*w* relations obtained through the inverse analysis and directly from uniaxial tensile tests.

D.2.   Determination of the σ−*w* relationship through least-squares based optimisation

In general, most inverse analysis procedures fix the shape of the σ-*w* curve, as exponential, linear or bilinear models, with the latter including drop-sloped, drop-constant and sloped-constant shapes. For plain concrete, the bilinear model has been most widely used (Roelfstra and Wittmann, 1986; Alvaredo and Torrent, 1987; Wittmann et al., 1987; Guinea et al., 1994; Bolzon and Maier, 1998; Stang and Olesen, 1998, 2000; Kooiman et al., 2000). Other curves that appear in the literature are exponential models (Hordijk, 1991; Duda and König, 1992; García-Álvarez et al., 1998). The optimisation based procedures use determine the best fit parameters through the least squares approach (Roelfstra and Wittmann, 1986; Kooiman et al., 2000). In other approaches, data from more than one specimen is used (Guinea et al., 1994; García-Álvarez et al., 1998) or more than one data set from the same specimen is used (Bolzon and Maier, 1998) within the inverse analysis for obtaining a unique solution. Nevertheless, the issue of the uniqueness of the σ-*w* curve obtained through such procedures has not been resolved completely (Alvaredo and Torrent, 1987). Another more recent alternative is the use of procedures that determine polylinear

σ-*w* curves, where the shape of the curve is also free to vary (Kitsutaka, 1995, 1997; Uchida et al., 1995; Kitsutaka and Oh-oka, 1998; Kitsutaka et al., 2001). In these procedures, a step-wise non-iterative inverse analysis is performed using segments of the load-deflection or load-CMOD response to obtain a piece-wise linear or polylinear σ-*w* curve. This approach normally gives inconsistent results in the first part of the σ-*w* curve but seems to be satisfactory in the tail (Planas et al., 1999). In the present work, the former approach is adapted with the use of prescribed shape for the σ-*w* curve. However, a comparison with the results of inverse analysis performed by Kitsutaka (2001) for the polylinear curve is also given.

The present methodology for the quasi-automatic determination of the cohesive stress versus crack opening (σ-*w*) relation uses experimentally-obtained load versus crack mouth opening displacement (P-CMOD) data from notched beam tests (e.g., the beam tests performed in Chapter 3, following the RILEM Recommendation, 2000). The P-CMOD curve obtained for each trial σ-*w* curve, defined by a set of parameters, is compared to the corresponding experimental result in the least square sense. Optimisation algorithms are used to obtain a set of parameters that best fit the experimental results. The methodology has been implemented in C^{++}, with the possibility of implementing new softening models or structural analysis approaches, and other features such as a friendly user interface and coupling to other cohesive crack modelling strategies. The current implementation deals only with three-point-bending specimens and is based on a quasi-analytical formulation developed by Stang and Olesen (1998).

The fitting process can be tailored to vary only some of the parameters, though it appears that the variation of as many parameters as possible should be permitted within the range that guaranties uniqueness of solution. For instance, for plain concrete, the tensile strength (f_t) can be obtained from another test (e.g., the Brazilian splitting test) and fixed within the optimisation process. For a given shape of the σ-*w* curve, the only variable is, then, the fracture energy (G_F) or another complementary parameter. However, both f_t and G_F that may result from an optimisation process that gives both parameters may fit the experimental P-CMOD curve best.

D.3.   Description of the Procedure

The inverse analysis that has been used is implemented in a software program called **Fit3pb**, which is described at the end of the annex. The user chooses the shape of the σ−*w* from various options, such as the Hordijk (1991) model, linear, drop-constant, sloped-constant, bilinear, trilinear and tetralinear. In the present study, the Hordijk and bilinear models have been used for plain concrete, and bilinear and trilinear models for SFRC. Seed values for the parameters are supplied by the user, along with the bounds for the parameters. The experimental load-CMOD curve is specified as data in a separate file as columns of load, CMOD and deflection data (the third column can be dummy data since it is not used when the load-CMOD curve is inverse analysed).

The program first linearizes, if needed, the initial part of the curve to eliminate any nonlinearity due to settling of the beam. Then, the program module based on the formulation of Stang and Olesen (1998) determines the modulus of elasticity from the initial linear part. The same module is used to determine a load-CMOD curve based on the trial model parameters to fit the experimental curve. The error is considered to be the integral of the difference between the two curves over the CMOD interval and an optimisation algorithm is used to determine the parameters that minimise this error. The final σ-*w* curve and the predicted load-CMOD curve are provided as output, along with the error in terms of the area under the curve. Weights can be provided for the different parts of the load-CMOD curve to give unequal importance to different regimes within the optimisation process.

D.4.   Example applications

The approach was applied to experimental data of plain and steel fibre reinforced concrete beams. The comparison of the predicted load-CMOD curve for a notched three-point-bend specimen of plain concrete with the corresponding experimental curve is presented in Figure D.1. The optimised softening curve (σ-*w*) corresponding to Hordijk's model is shown in the inset. The Young's modulus was determined automatically by fitting

a straight line to the initial experimental data, using the Stang-Olesen formulation. In order to further validate results, the optimised material parameters were used in a finite element simulation with DIANA (Witte et al., 2000) to obtain the curve indicated with a dashed line. This simulation was performed using interface elements along the crack path (discrete crack approach), with the σ-*w* curve described by the two model parameters f_t (tensile strength) and G_F (fracture energy). The bulk of the specimen was assumed to be linear elastic. It can be seen that the analytical and finite element simulations correspond closely to the experimental curve.

The σ-*w* curves for three such beams, all from the same batch, are plotted together in Figure D.2, where the scatter in the value of f_t can be appreciated. However, there is not much variability in the rest of the curve. The dashed lines correspond to results from uniaxial tension tests on notched moulded cylinders of the same concrete. The curves from the inverse analysis of the beams and the uniaxial tension tests are practically similar.
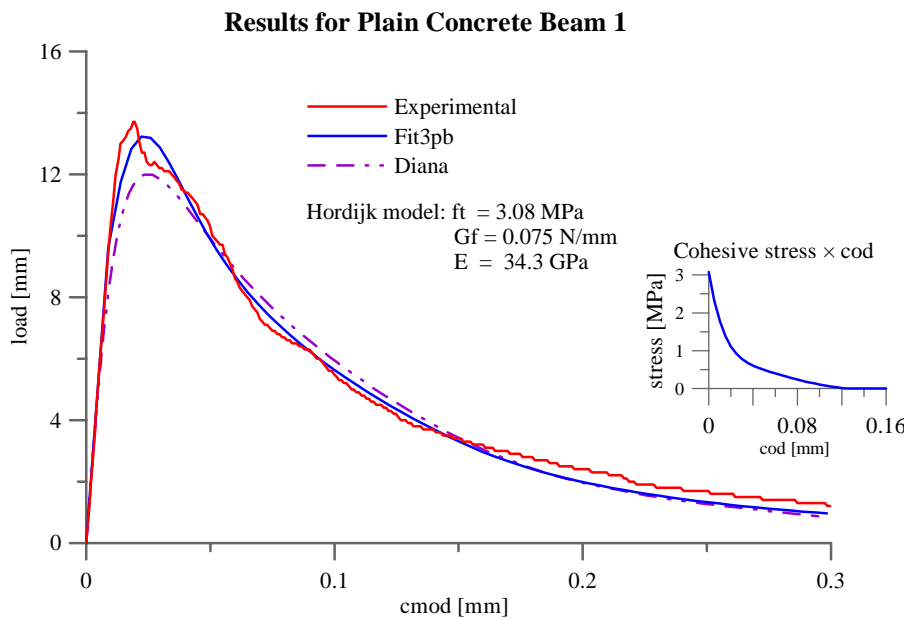

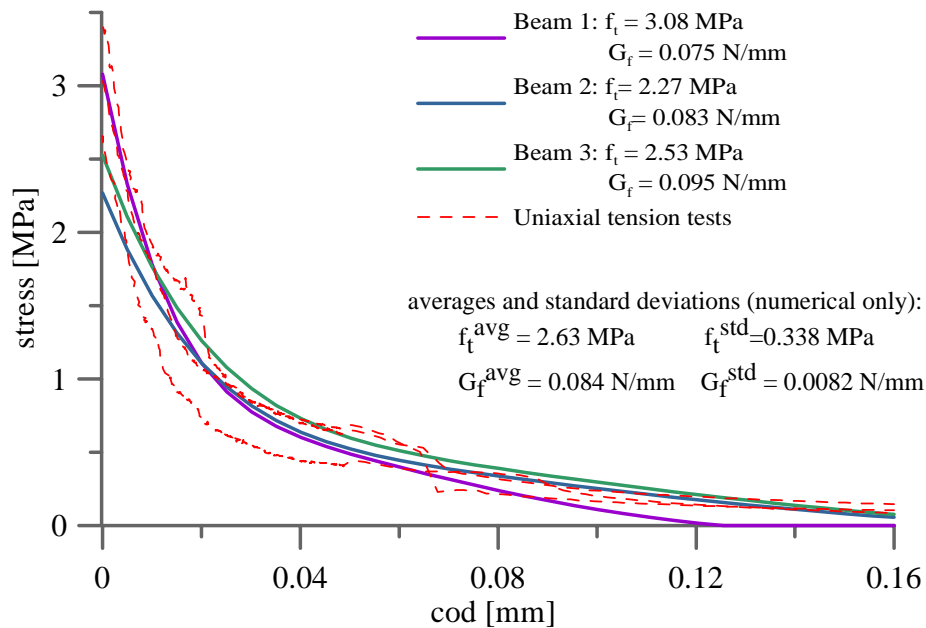**Figure D.1.** Load-CMOD curves for plain concrete beam 1

Figure D.2. Stress versus crack opening displacement curves for the three beams and curves from uniaxial tension tests

An application of the optimisation procedure to a steel fibre reinforced concrete specimen is presented in Figure D.3. Two models of the σ-*w* curve have been used: the bilinear and sloped-constant (with a sloping first part and a constant second part) models. In general, the curve obtained from the analytical prediction with the bilinear curve matches the experimental curve satisfactorily while the sloped-constant model leads to an average fit in the post-peak.

For obtaining a comparison with the polylinear procedure, the experimental data was provided to Kitsutaka who independently determined a σ-*w* curve through inverse analysis of the beam test results. The load-CMOD curve corresponding to Kitsutaka's σ-*w* curve was obtained using finite element analysis with the DIANA code, and is also shown in Figure D.3. The σ-*w* curves obtained by inverse analysis (bilinear, sloped-constant and

polylinear) for beam 1 are presented in Figure D.4. The bilinear curve is close to Kitsutaka's curve while the sloped-constant gives an average final part.
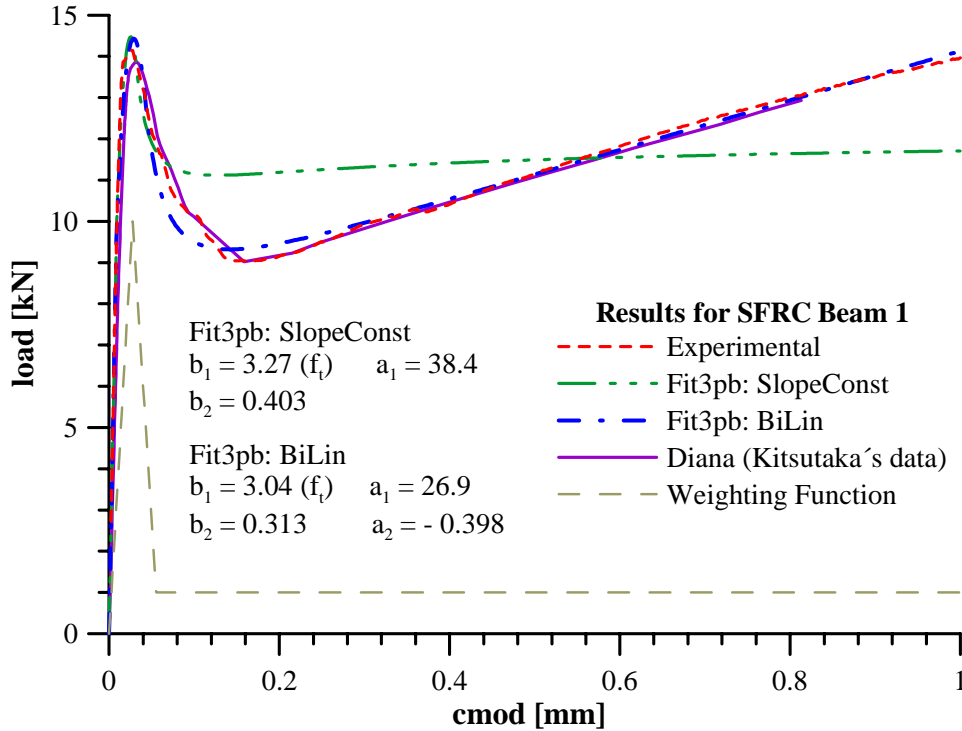

Fit3pb: SlopeConst
$b_1 = 3.27$ (f_t) $a_1 = 38.4$
$b_2 = 0.403$

Fit3pb: BiLin
$b_1 = 3.04$ (f_t) $a_1 = 26.9$
$b_2 = 0.313$ $a_2 = -0.398$

Results for SFRC Beam 1
Experimental
Fit3pb: SlopeConst
Fit3pb: BiLin
Diana (Kitsutaka´s data)
Weighting Function

Figure D.3. Load-CMOD curves obtained for SFRC beam using different σ-*w* models

Since the bilinear curve is a better model, it was used for inverse analysing two other similar SFRC beams. The resulting σ-*w* curves for all the three beams are presented in Figure D.5 and the fitted and experimental load-CMOD curves corresponding to all three beams are presented in Figure D.6. In Figure D.5, the data from inverse analysis is compared with the curves obtained from uniaxial tension tests of moulded and cored cylinders. The cores were obtained perpendicular to the casting direction. There is considerable scatter in the curves from the inverse analysis of the beams. However, it is clear that they are closer to the experimental results from cored specimens with the moulded specimens giving much lower stresses.
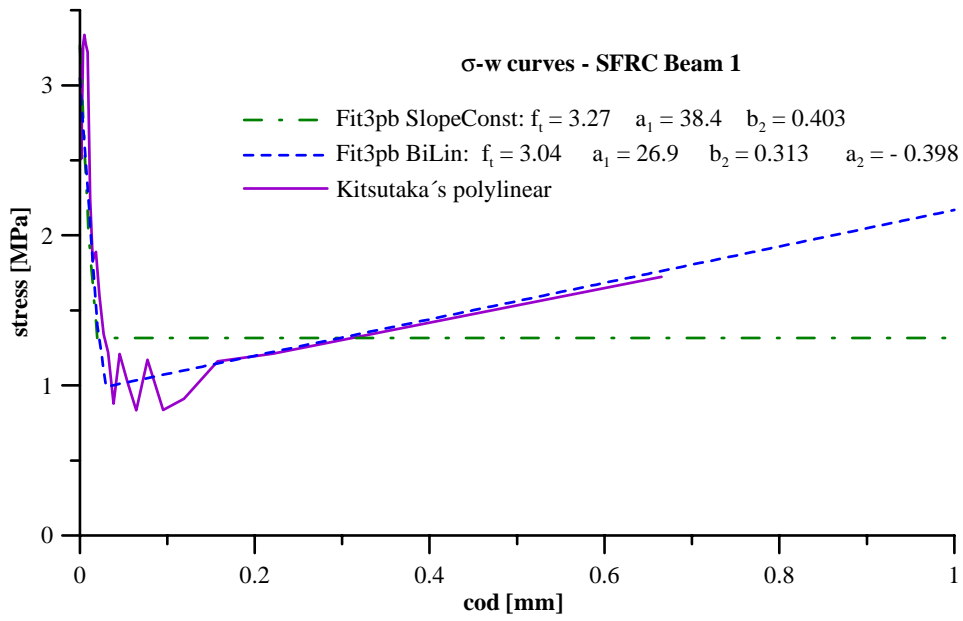
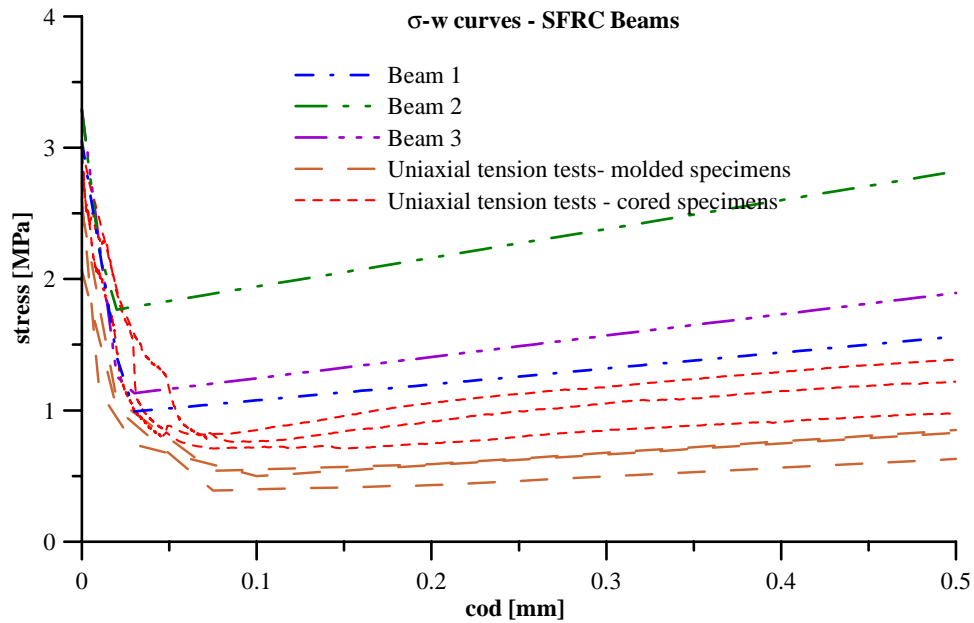**Figure D.4.** The σ-*w* curves obtained from inverse analysis

Figure D.5. The σ-*w* curves obtained from inverse analysis of 3 beams compared with results from uniaxial tension tests
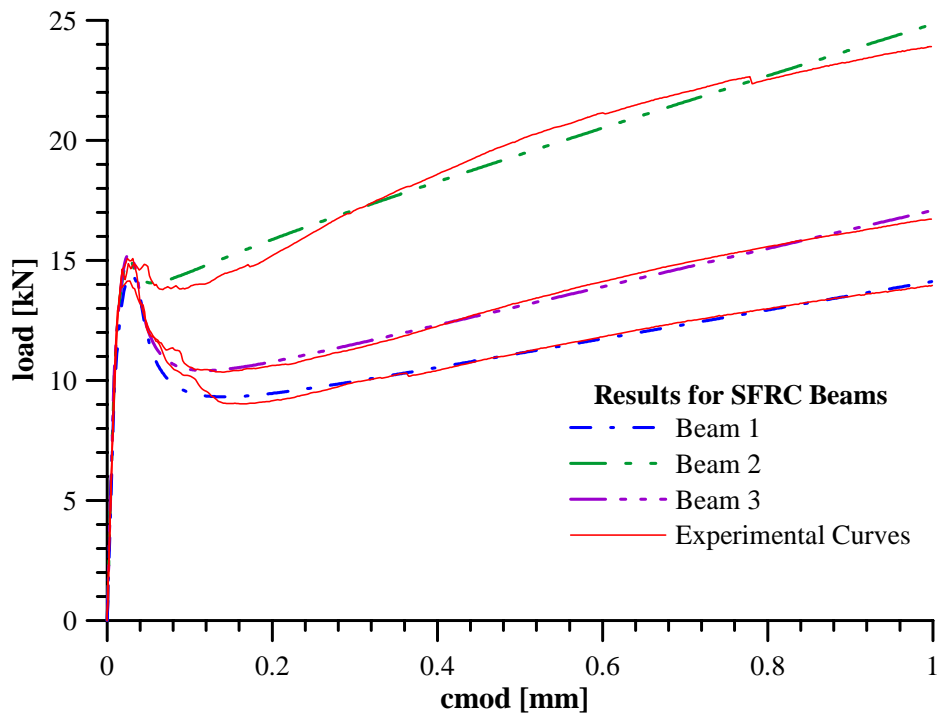
Figure D.6. Load-CMOD curves corresponding to the beam specimens presented in
Figure D.5

The load-displacement curves of the SFRC beam 1 corresponding to the different
σ-*w* curves have been determined and are shown in Figure D.7, along with the
experimental curve. The simulated behaviour is close to the experimental response for the
bilinear model and with an average post-peak for the sloped-constant model.
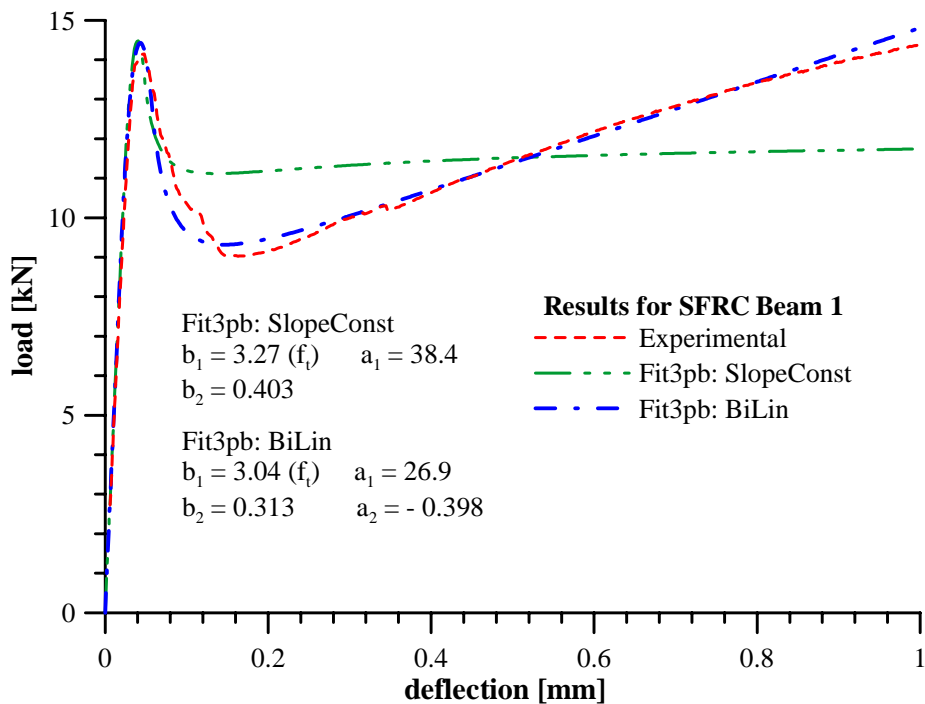
Figure D.7. Load-deflection curves from simulations and test

In general the inverse analysis with the **Fit3pb** worked well, with reasonably fast convergence for the analyses related to plain concrete, using the exponential Hordijk model, as well as with the other linear and bilinear models. However, some difficulties were found for trilinear or tetralinear models, where the optimisation algorithms converged to different solutions depending on the initial trial parameters. It appears that there may be some non-uniqueness in the values of the parameters when the models are complex (i.e., with more than two independent parameters). In such cases, it has been found that the use of parameters from a model with a lower number of parameters as the trial parameters is useful.

D.5. Use of weighting functions in the inverse analysis

In the previous sections, a strategy for inverse analysis to obtain the stress-crack opening curve based on a least square fitting using experimental load-CMOD curves has

been applied. In general, the strategy leads to good agreement for plain and fibre reinforced concretes. The implementation allows the use of arbitrary weighting functions to improve the fit of the experimental results in prescribed portions of the fitting interval according to the structural application. For example, for the use of SFRC, to improve shear strength in a bridge beam web, where the crack openings are small, more weight would be given to the response at the peak. On the other hand, in the design of a fibre reinforced concrete pavement, the correct description of the material behaviour beyond 2 or 3 mm is probably more important than the peak load capacity.

The use of a weighting function for the determination of material properties is debatable since a good model should be able to describe the material behaviour under all circumstances. However, this may only be possible through the use of complex models and inverse analysis procedures and/or the use of data from other tests to keep the fitting error within acceptable limits. Assuming that a weighting function should be introduced, the second issue that arises is the choice of the appropriate function. In the examples given in section D.4, higher weight is given to the peak response, as seen in the same program command file at the end of the annex.

The type of weighting used here is illustrated in Figure D.8, which corresponds to the plain concrete beam treated in section D.4. The chosen weighting function is shown in under the data, where a relatively higher weight (i.e., 10) is given at the peak with a linear increase in the pre-peak and a symmetric decrease in the post-peak until a constant value of 1. The Hordijk model is fitted through the inverse analysis of the load-CMOD data. The σ-w curves obtained with and without weighting are shown in the inset and the numerical predictions of load-CMOD in the main plot. It can be seen that there is practically no difference in the fitting of the curve. The variations in the σ-w curve due to the weighting are also negligible.
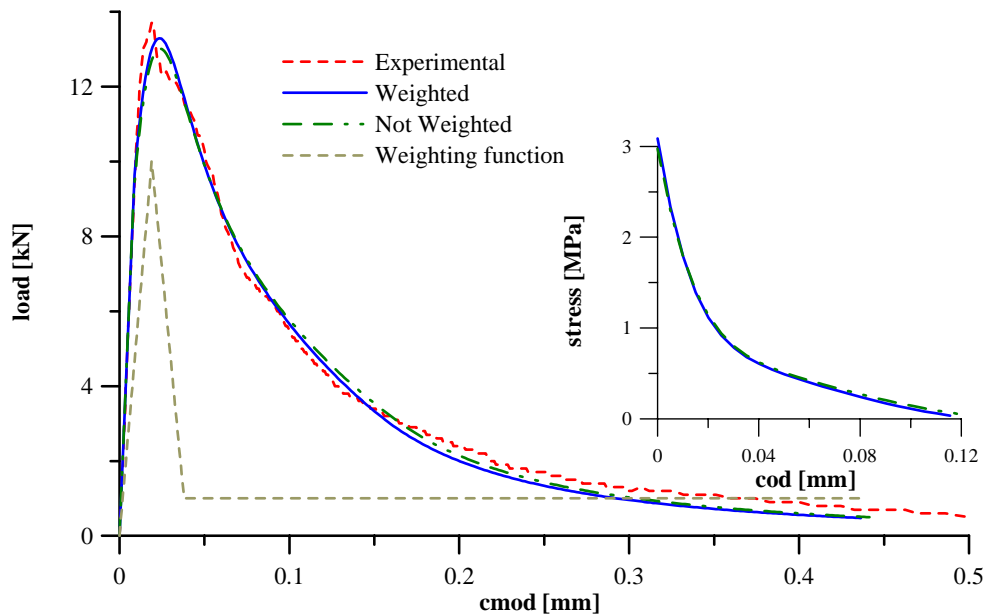
Figure D.8. Effect of weighting function for plain concrete

In the case of SFRC, for the inverse analysis of a trilinear model for the σ-w curve, the importance of the weighting is related to the fitting interval (of CMOD). In Figure D.9, the experimental load-CMOD curve used in the inverse analysis is shown, along with the predictions obtained using the σ-w curves with and without weighting. The interval specified is [0.0;1.0 mm] and the weighting function is shown under the load-CMOD curve. Note that the weighting function has a peak value of 10 and a plateau value of 3. The two σ-w curves are shown in the inset. It can be seen that the differences in both the σ-w curve and the fit of the load-CMOD curve due to weighting are small.

However, when the fitting interval is larger, say [0.0;2.0 mm], as in Figure D.9, the fits of the load-CMOD curve vary significantly, especially around the peak. The σ-w curve obtained using weighting is compared in Figure D.11 with that obtained without weighting. The main differences are in the first and final parts of the curve, with the tensile strength being significantly different.
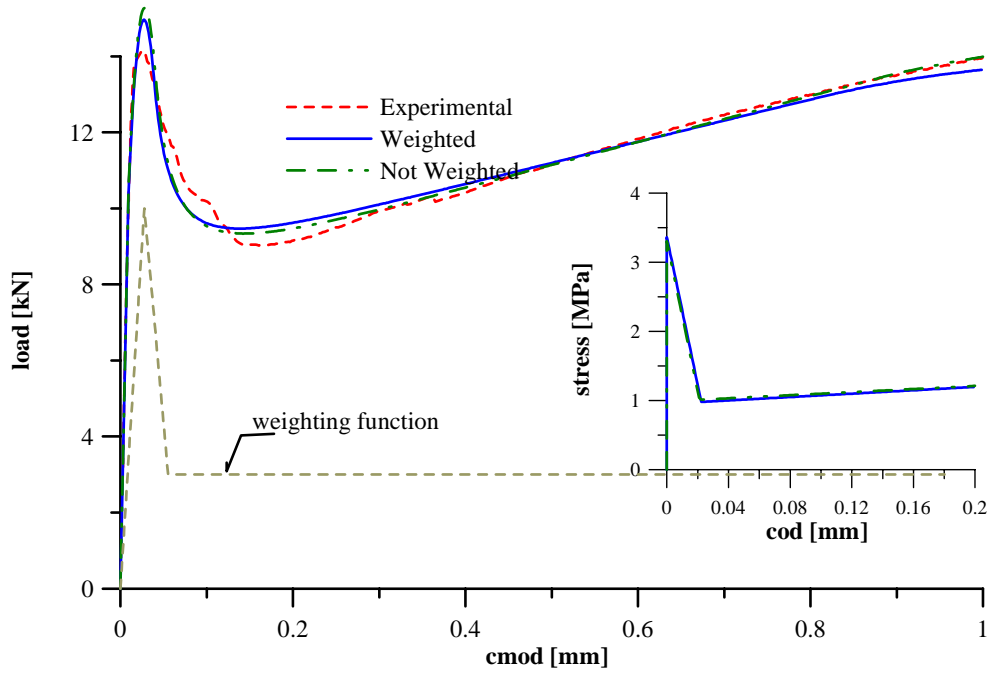
Figure D.9. Effect of weighting function for SFRC, using a trilinear model and fitting interval of [0.0; 1.0 mm]
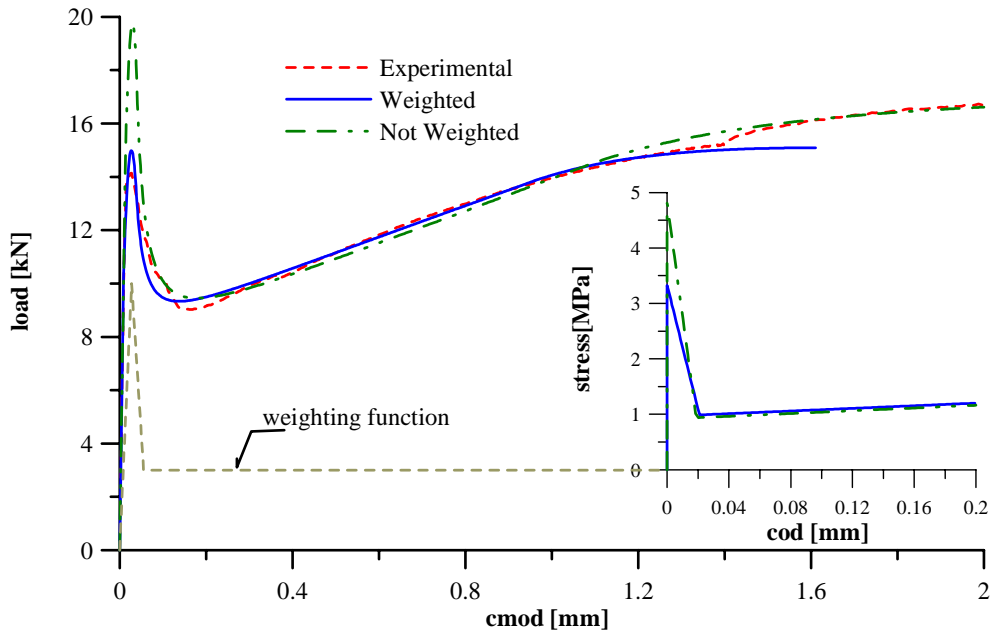

Figure D.10. Effect of weighting function for SFRC, using a trilinear model and fitting interval of [0.0; 2.0 mm]
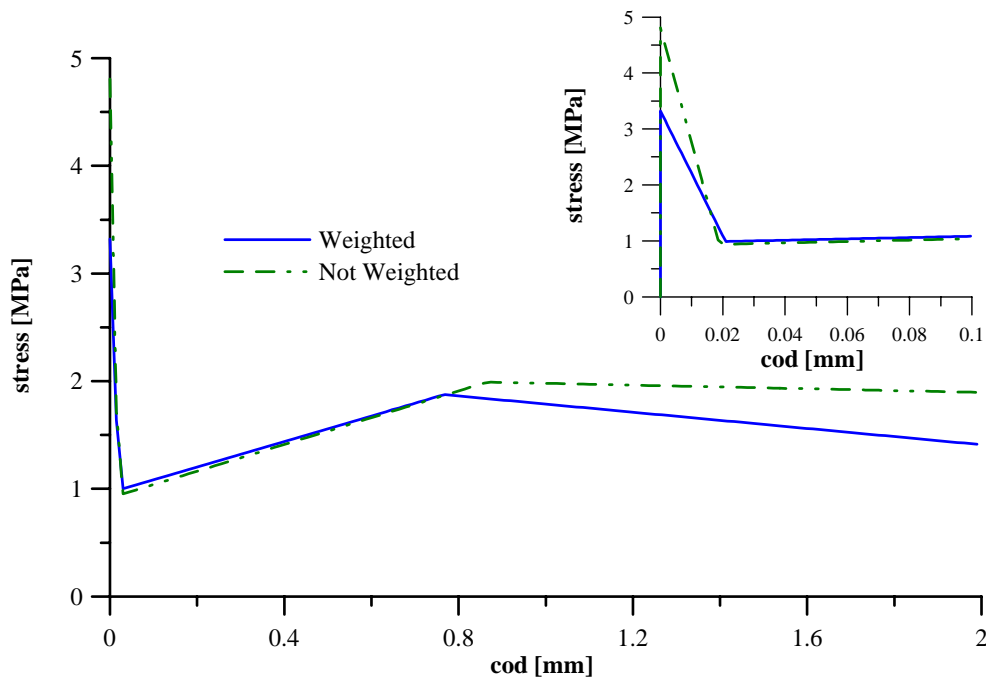
Figure D.11. Effect of weighting function on the stress-crack opening curve obtained from inverse analysis for a fitting interval [0.0; 2.0 mm]

The use of a weighting function appears to be important for the inverse analysis of SFRC. However, comparing the numerical prediction of the load-CMOD curve with the experimental results should be done to check the appropriateness of the selected function.

D.6.   Conclusions

The development of a system of algorithms for the inverse analysis of the stress-crack opening curve under tension using the load-CMOD response of notched beams is presented. The structural analysis is based on the model of Stang and Olesen, which uses the trial σ-*w* curve to obtain the numerical load-CMOD response that fits the experimental result. Example applications are presented to validate the approach for plain and fibre reinforced concrete. It is seen that a weighting function is needed in the case of a large fitting interval for SFRC. Comparisons with finite element calculations and independent

inverse analysis by Kitsutaka are satisfactory. The implementation of the set of algorithms with a software program makes the procedure easy to apply with only the need to create two short header files with the choice of the model and vector space dimension.

References

Alvaredo, A.M., and Torrent, R.J. (1987) "The effect of the shape of the strain-softening diagram on the bearing capacity of concrete beams", *Mater. Struct.*, V. 20, pp. 448-454.

Barenblatt, G.I. (1962) "The mathematical theory of equilibrium of cracks in brittle fracture", *Advances in Appl. Mech.*, V. 7, p. 55.

Bittencourt, T.N. (1993) *Computer simulation of linear and nonlinear crack propagation in cementitious materials*, Ph.D. Thesis, Cornell University.

Bolzon, G., and Maier, G. (1998) "Identification of cohesive crack models for concrete on the basis of three-point-bending-tests*", Computational Modelling of Concrete Structures*, Eds. de Borst, Bicanic, Mang and Meschke, Balkema, pp. 301-309.

Duda, H., and König, G. (1992) "Stress-crack opening relation and size effect in concrete", *Applications of Fracture Mechanics to Reinforced Concrete* (Proc. Intnl. Workshop, Turin, 1990), Ed. A.Carpinteri, Elsevier, pp. 45-61.

Gettu, R., García-Álvarez, V.O., and Aguado, A. (1998) "Effect of aging on the fracture characteristics and briitleness of a high-strength concrete", *Cem. Concr. Res.*, V. 28, No. 3, pp. 349-355.

Guinea, G., Planas, J., and Elices, M. (1994) "A general bilinear fit for the softening curve of concrete", *Mater. Struct.*, V. 27, pp. 99-105.

Hillerborg, A., Modeer, M., and Petersson, P-E. (1976) "Analysis of crack formation and crack growth in concrete by means of fracture mechanics and finite elements", *Cem. Concr. Res.*, V. 6, pp.773-782.

Hordijk, D.A. (1991) *Local approach to fatigue of concrete*, Doctoral thesis, Delft University of Technology, The Netherlands.

Kitustaka, Y. (1995) "Fracture parameters for concrete based on poly-linear approximation analysis of tension softening diagram", *Fracture Mechanics of Concrete Structures*, Ed. F.H.Wittmann, Aedificatio Publ., Freiburg, Germany, pp. 199-208.

Kitsutaka, Y. (1997) "Fracture parameters by polylinear tension-softening analysis", *J. Engng. Mech.*, V. 123, No. 5, pp. 444-450.

Kitsutaka, Y., and Oh-oka, T. (1998) "Fracture parameters of high-strength fiber reinforced concrete based on poly-linear tension softening analysis", *Fracture Mechanics of Concrete Structures*, Eds. H.Mihashi and K.Rokugo, Aedificatio Publ., Freiburg, Germany, pp. 455-464.

Kitsutaka, Y., Uchida, Y., Mihashi, H., Kaneko, Y., Nakamura, S., and Kurihara, N. (2001) "Draft on the JCI standard test method for determining tension softening properties of concrete", *Fracture Mechanics of Concrete Structures*, Eds. de Borst et al., Swets & Zeitlinger, Lisse, THe Netherlands, pp. 371-376.

Kitsutaka, Y. (2001), Personal communication to R.Gettu.

Llorca, J., and Elices, M. (1990) "A simplified model to study fracture behavior in cohesive materials", Cem. Concr. Res., V. 20, pp. 92-102.

Kooiman, A.G., van der Veen, C., and Walraven, J.C. (2000) "Modelling the post-cracking behaviour of steel fibre reinforced concrete for structural design purposes", *Heron*, V. 45, No. 4, pp. 275-307.

Luenberger, D.G. (1984) *Linear and Nonlinear Programing*, Addison-Wesley Publishing Company, 491 p.

Martha, L.F.C.R., and Bittencourt T.N. (1999) *Quebra2D - Two-dimensional Fracture Simulation*, http://www.tecgraf.puc-rio.br/~lfm/.

Nanakorn, P., and Horii, H. (1996) "Back analysis of tension-softening relationship of concrete", *J. Materials, Conc. Struct. Pavements* (JSCE), V. 32, No. 544, pp. 265-275.

Planas, J., Guinea, G.V., and Elices, M. (1999) "Size effect and inverse analysis in concrete fracture", *Int. J. Fracture*, V. 95, pp. 367-378.

Press, W.H., Teukolsky, S.A., Vetterling, W.T. & Flannery, B.P. (1993) *Numerical Recipes in C: The art of scientific computing*, Cambridge University Press, 994 p.

Roelfstra, P.E., and Wittmann, F.H. (1986) "Numerical method to link strain softening with failure of concrete", *Fracture Toughness and Fracture Energy of Concrete*, Ed. F.H.Wittmann, Elsevier Science, pp. 163-175.

Stang, H., and Olesen, J.F. (1998) "On the interpretation of bending tests on FRC-materials", *Fracture Mechanics of Concrete Structures*, Eds. H.Mihashi and K.Rokugo, Aedificatio Publ., Freiburg, Germany, pp. 511-520.

Stang, H., and Olesen, J.F. (2000) "A fracture mechanics based design approach to FRC", *Proc. Fifth RILEM Symposium on Fibre-Reinforced Concretes* (Lyon), Eds. P.Rossi and G.Chanvillard, RILEM, Cachan, France, pp. 315-324.

Uchida, Y., Kurihara, N., Rokugo, K., and Koyanagi, W. (1995) "Determination of tension softening diagrams of various kinds of concrete by means of numerical analysis", Fracture Mechanics of Concrete Structures, Ed. F.H.Wittmann, Aedificatio Publ., Freiburg, Germany, pp. 17-30.

Witte, F.C., and Kikstra, W.P., Eds. (2000) *DIANA - Finite Element Analysis, User's Manual release 7.2, Nonlinear Analysis*, TNO Building and Construction Research, (digital version in HTML).

Wittmann, F.H., Roelfstra, P.E., Mihashi, H., Huang, Y.-Y., Zhang, X.-H., and Nomura, N., 1987, Influence of age of loading, water-cement ratio and rate of loading on fracture energy of concrete, *Mater. Struct.*, V. 20, pp. 103-110.

STRUCTURE OF THE SOFTWARE

The software **Fit3pb** has been developed according to the Object Oriented Programming (OOP) philosophy, using C^{++} language. In order to describe the implementation, some basic concepts of OOP are presented next.

Code development is usually performed by following a sequence of steps, in general grouped in routines, managing data according to the needs to accomplish a specific task. This is known as the procedural, or traditional, approach. An alternative approach, that is becoming popular nowadays, is based on the Object Oriented Programming (OOP) philosophy. According to this philosophy, an application consists of a set of objects interacting with each other to perform the required task. An **object** is responsible for keeping **member data**, not accessible to other objects, and the corresponding handling routines. The other objects see only a set of **member functions** that are necessary to interact with it. For example, in the code described here, a PDelta object handles data and the computations required to produce a P-CMOD curve for a given set of softening parameters. The other objects do not need to know how the curves are produced. The only thing they need to know is that given a set of parameters, the PDelta object returns a P-CMOD curve. In the current implementation a quasi-analytical model for the notched three-point-bending test is used. If a Finite Element model is implemented for the same purpose, only the module corresponding to PDelta has to be developed, provided the member functions returns a P-CMOD curve for the same set of parameters.

The class to which it belongs describes the object. A **class** describes the form and behaviour of the objects. A PDelta object is an instance of the class PDelta.

The OOP philosophy encompasses the following key features:

- **Abstraction**: creation of a well-defined interface for the object, i. e., which are the public **member functions** for the interaction with the other objects;
- **Encapsulation**: keeping all the implementation details private, i. e., other objects cannot access **member data** or routines that are not in the interface;

- **Hierarchy**: a class can be derived from another (**subclass** and **superclass** are used in this text to refer to this interrelationship). In **Fit3pb** the class OptMngr receives a seed to fire up the error minimization process and keeps, privately, all the optimization routines. The subclass OptMngrBiLin inherits these features from OptMngr, adding data and routines necessary to handle the bilinear softening model, which requires 4 parameters to be determined from the minimization process. The same applies to OptMngrHordijk, which requires 2 parameters to handle Hordijk´s model. Normally a root class is created to handle general capabilities. In **Fit3pb** the root class is named **Object**, which is a superclass of all other classes in the application;

- **Polymorphism**: a code is said to be polymorphic if it can be transparently used on instances of different classes. An example is given by the PDelta private routines that handle softening model parameters without asking to which classes the objects belong (SigwHordijk, SigwBiLin, SigwTriLin, etc.). In Fit3pb, this feature allows the implementation of a new softening model XXX by creating only the pieces of code corresponding to the SigwXXX and OptMngrXXX.

Object Oriented Programming is independent of the programming language. However, some languages are prepared to handle directly the above features. Perhaps the most popular is C^{++}, which corresponds to the C language with the addition of **class**, the most important, and other features. C^{++} was chosen for the development of **Fit3pb**.

The OOP philosophy favors code modularization and reuse. The overhead introduced by the manipulation of objects is compensated by the higher efficiency in the implementation of new capabilities or code reuse in new applications.

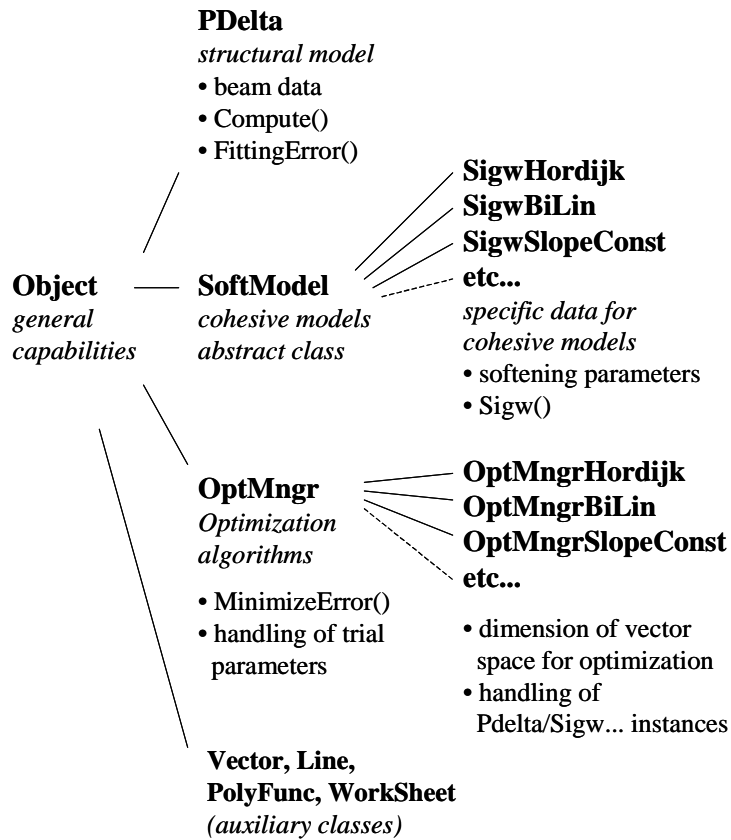The software **Fit3pb** is structured in the classes described in Fig. 1.

PDelta
structural model
• beam data
• Compute()
• FittingError()

SigwHordijk
SigwBiLin
SigwSlopeConst
etc...
specific data for
cohesive models
• softening parameters
• Sigw()

Object
general
capabilities

SoftModel
cohesive models
abstract class

OptMngr
Optimization
algorithms
• MinimizeError()
• handling of trial
  parameters

OptMngrHordijk
OptMngrBiLin
OptMngrSlopeConst
etc...
• dimension of vector
  space for optimization
• handling of
  Pdelta/Sigw... instances

Vector, Line,
PolyFunc, WorkSheet
(auxiliary classes)

Figure 1. Structure of classes in **Fit3pb**

The main classes can be described as:

♦ **Object**: root class, with general capabilities, superclass of all other classes in the system

♦ **PDelta**: corresponds basically to the analytical formulation developed by Stang and Olesen (1998), using their C code with some adaptations to create a C^{++} class. The basic member functions are:

▪ *Compute*(): generates the load versus CMOD data for the current set of softening parameters

- *FitYoungModulus*(): corrects the Young´s modulus based on the initial linear portion of the experimental data (P-CMOD or P-δ, where δ is the load line deflection), using the formulation of Stang and Olesen (1998)

- *YoungModFGB*(): corrects the Young´s modulus based on the initial linear portion of the experimental data (P-CMOD or P-δ, where δ is the load line deflection), using the formulation of Ferreira et al. (2001)

♦ **SoftModel**: Abstract class to handle softening model information. The derived subclasses are:

- **SigwHordijk**: Hordijk model (2 parameters)
- **SigwLinA1**: Linear with varying post peak slope (2 parameters)
- **SigwLinGf**: Linear with varying fracture energy (2 parameters)
- **SigwDropConstant**: Drop-constant (2 parameters)
- **SigwSlopeConstant:** Sloped-constant (3 parameters)
- **SigwBiLin**: Bilinear (4 parameters)
- **SigwTriLin**: Trilinear (6 parameters)
- **SigwTetraLin**: Tetralinear (8 parameters)
- **SigwPolyLin** (multiple parameters) - implemented for the only purpose of analysing a single model with the polylinear function generated by Kitsutaka´s code. No OptMngr subclass is associated to this softening model, i. e., Fit3pb does not handle error minimisation for this model.

The **member data** are the softening parameters of the corresponding model and the basic **member functions** are

- *sigw*(): returns cohesive stress for given crack opening displacement (w)
- *sigww*(): returns w × sigw()

♦ **OptMngr**: class that handles the error minimisation

The basic **member functions** are:

- *MinimizeError*(): handles algorithms to determine partial derivatives of the error objective function, and optimisation algorithms (search direction and line search minimisation). Several complementary non-member functions

are implemented to handle first and second partial derivatives, conjugate gradient and line search algorithms.

The derived subclasses, to handle different softening models are

- **OptMngrHordijk**
- **OptMngrLinA1**
- **OptMngrLinGf**
- **OptMngrDrropConst**
- **OptMngrSlopeConst**
- **OptMngrBiLin**
- **OptMngrTriLin**
- **OptMngrTetraLin**

The **member data** are the dimension of the vector space for the optimisation process (i. e., the number of parameters describing the softening model), the corresponding instances of the SoftModel subclass, target experimental data (PolyFunc class), type of data (P-CMOD or P-δ).

Auxiliary Classes:

BeamData: handles specimen geometric and physical data

PolyFunc: handles description of functions by a polylinear function

> **member data**:
>
> number and list of points
>
> **member functions**:
>
> DiffSqr() integrates the squared difference between current and another instance
>
> EvaluateAt() evaluates the function for a given abscissa
>
> LineFit() fits a line through points selected via a bounding box

Vector, Point, Line: handle operation in n-dimensional vector space

> **WorkSheet, Summary**: handle data collection for output during the program execution

CommandMngr: implements a parser that allows execution through command lines or, as it is still under development, a graphical user interface that favours data consistency check and the execution of multiple analyses. The commands corresponding to the CommandMngr class methods are described in a separate document intended to be a User Guide (available through FTP). Table 1 illustrates the command lines to be supplied in a file with extension *.3pb*. Besides this file, the user should provide a file with the experimental results (with extension *.exp*), described by crack mouth opening displacement (CMOD), deflection (δ) and corresponding load (P), created from the sequential data generated by the testing machine software. All the output files have the same default name defined by the user. Spreadsheet files are generated with data in a format directly readable by most plotting software.

Table 1 Example of command lines to perform the inverse analysis

```
echo
defaultFileName beam1cmod
openResultsMngr
beamData 500.0 150 150 25 0.0 43020.0
setCurveType defl load
softModel TriLin 3.0 1.2 1.4 56.05 0.32 -0.36
limitCmod 1.0
enterTargetTriple beam1.exp
correctYoungMod
setOptAlgor Tangent 20 Quadr 20
setInterval 400   0   0.5
minimizeError 3.0 1.2 1.4 56.05 0.32 -0.36
softModelOpt
plotSigw 0.5
computeResponse
compareToTarget
printResponse
savePolyFunc current
closeResultsMngr
bye
```