README – A method for determining the performance of formulas for calculating the crack width and spacing in reinforced concrete elements

Introduction
This README describes how to import, process data and generate results for the study "A method for determining the performance of formulas for calculating the crack width and spacing in reinforced concrete elements".

Please read this file, and the licence, carefully.

Metadata:
- Title: A method for determining the performance of formulas for calculating the crack width and spacing in reinforced concrete elements
- Version: final
- Date: 2025-08-20 (year-month-day)
- Country: the Netherlands
- Institution: Eindhoven University of Technology
- Author: Anton (I.A.) van der Esch, i.a.v.d.esch@tue.nl

Files
Especially the following files are relevant:
- 1) Research/Method_comparing_CWFs/Data_processing_for_GitHuB_2025_08_20 → contains the data processing of the database "Crack width and crack spacing in reinforced and prestressed concrete elements: database". The database is available open-source: (https://doi.org/10.1016/j.dib.2024.110305 [1] and https://doi.org/10.5281/zenodo.16903588 [2]) keep in mind that the file Data_processing_for_GitHuB_2025_08_20 uses the database in a format where the targets are expanded, i.e., the number of cracks, crack widths and crack spacings are listed in separate columns.
- 2) Research/Method_comparing_CWFs/CWF_analysis_for_GitHuB_2025_08_20 → contains the evaluation of the formulas and generation of the results

Workflow for data processing
- Go to Github: https://github.com/iavdesch/Research/tree/main/Method_comparing_CWFs
- Download the database "Crack width and crack spacing in reinforced and prestressed concrete elements: database" from Github (Datasets_used/df_mastersorted.csv) and save it on your computer; for instance, under downloads → this is also done in the Jupyter Notebook.
- Also, download Data_processing_for_GitHuB_2025_08_20.ipynb to your computer, for instance, under downloads.
- Go to the folder "Datasets_used" and copy the file "df_master_sorted" to your computer, to the same folder as the .ipynb folder.
- Run the file Data_processing_for_GitHuB_2025_08_20.ipynb → the data processing is executed and enriches the database with additional variables. The output file is called "Working_database_no_hordijk_initshrink.csv".
- Run the file "Working_database_no_hordijk_initshrink.csv" to re-generate the figures and results in the paper. The workflow for the generation of the figures is presented below.

The file used for data processing and generating the results offers various options or possibilities. For instance, you have the option to calculate the steel stress including tension softening effects according to Hordijk [3, 4]. Additionally, you have the option to discard the computation of uniform targets; thus, no maximum crack widths and spacings are then derived from mean ones in case maximum ones are not provided. Finally, the effect of shrinkage occurring prior to testing (thus, the effect of shrinkage strains developing prior to cracking when the element is placed in a load setup; in the notebook called: initial shrinkage) can be included.

To generate the results in the paper, each of these settings was turned off except for the uniform target functions.

Workflow to reproduce results presented in figures and tables
To make the code easy to read and reproduce, all source code was created in two separate Jupyter Notebooks (Python). For reproducing the results, per the figure (as indicated in the paper), specific steps need to be taken. These steps are summarised below. To run the notebook directly for directly obtaining the numerical results, without changing any settings, the notebook imports various datasets which were created in the various steps indicated.

Figure 2 – error distribution of formulas for calculating maximum crack width; crack formation stage and stabilised cracking stage
File 2: no additional steps required.

Figure 3 – error distribution of formulas for calculating maximum crack spacings; stabilised cracking stage
File 2: no additional steps required.

Figure 4 – conversion factors
File 1: no additional steps required.

Figure 5 – ratio of maximum crack widths
File 2: imports a separate dataset. The dataset is based on small data analysis, using measurements of Gilbert et al. [5, 6]. The dataset and separate data analysis are uploaded to GitHub. The file containing the data to generate the figure is called "Data_Ncrack_ratio.csv". To obtain the relevant data, see "Comparison_wk_wmax.xlsx".

Figure 6 – Influence of the implementation of uniform targets on the error distribution
File 2: imports a separate dataset. This dataset was created by taking the following steps:

- Run file 1) with default settings (as published); save the export as a separate processed dataset.
- Run file 1) a change in the settings: uniformtargets = "no"; save the export as a separate processed dataset.
- For processed datasets: import them File 2, for each import, a separate run and save the resulting error distributions in two distinct datasets.
- Use the two distinct datasets to generate the figure.

Figure 7- performance of the formula of Menn
See the steps of figure 6.

Figure 8 – interpretation of metrics for the maximum crack width
File 2: no additional steps required.

Figure 9 – type of loading versus cracking stages
File 2: no additional steps required.

Figure 10 – error distribution of formulas for calculating maximum crack width; crack formation stage only
File 2:

- At: "switch settings for output": enable #df = df[df["CS"] == 1]; # CFS

Figure 11 – error distribution of formulas for calculating maximum crack width; stabilised cracking stage only
File 2:

- At: "switch settings for output": enable #df = df[df["CS"] == 2]; # SCS

Table 2a: metrics for the overall error distribution for the crack width – CFS + SCS

- The table will then be written towards your computer. If you have chosen the download folder as default folder, the tables will be automatically written towards that folder.
- Alternatively: the metrics can be found in the output: "func_plot_results_per_group"

Table 2a: metrics for the overall error distribution for the crack width – CFS

- At: "switch settings for output": enable #df = df[df["CS"] == 1]; # CFS
- For the remaining steps; see Table 2a – CFS + SCS

Table 2b: metrics for the overall error distribution for the crack width – SCS

- At: "switch settings for output": enable #df = df[df["CS"] == 2]; # SCS
- For the remaining steps; see Table 2a – CFS + SCS

Table 2b: metrics for the overall error distribution for the crack spacing – SCS

- See Table 2a – CFS + SCS

Closure
Please address questions to i.a.v.d.esch@tue.nl

Bibliography

[1] A. van der Esch, R. Wolfs and S. Wijte, "Crack width and crack spacing in reinforced and prestressed concrete elements: Data description and acquisition," *Data in Brief,* vol. 54, 2024.

[2] I. van der Esch, "Crack width and crack spacing in reinforced and prestressed concrete elements: database," 19 August 2025. [Online]. Available: https://doi.org/10.5281/zenodo.16903588. [Accessed 19 August 2025].

[3] D. A. Hordijk, "Local approach to fatigue of concrete," W.D. Meinema b.v., Delft, 1991.

[4] H. Cornelissen, D. Hordijk and H. Reinhardt, "Experimental determination of crack softening characteristics of normalweight and lightweight," *Heron,* vol. 31, no. 2, pp. 45-56, 1986.

[5] R. I. Gilbert and S. Nejadi, "An Experimental Study of Flexural Cracking in Reinforced Concrete Members under Short Term Loads," University of New South Wales, Sydney, 2004.

[6] R. Gilbert, *Personal Communication.*