EMSC toolbox for MATLAB
Program Manual

Version 0.82,March 31 2003

Harald Martens
Consensus Analysis AS
Harald.Martens@matforsk.no

Software summary

The Matlab program EMSC_Main.m represents an implementation of Extended Multiplicative Signal Correction (EMSC[1,3]) and related methods for model-based pre-processing of multichannel data such as diffuse absorbance spectra and chromatograms. The purpose is to elucidate the causal basis for uncontrolled variations in the multi-channel data, and to make the data more suited for multivariate calibration.

The toolbox provides software for

- EMSC pre-processing
  with its sub-methods
  - Multiplicative Signal Correction (MSC),
  - Spectral Interference Subtraction (SIS[1]),

  as well as its inverted method
- Extended Inverted Scatter Correction (EISC[2,3])
  with its sub-method
  - Inverted Multiplicative Scatter Correction (ISC).

During *calibration* (i.e. definition) of a pre-processing method, the model spectra (the meta-parameters) used for the model-based pre-processing of a given set of data, may be

- read from file as prior information, or
- estimated automatically from available set of data
  by various automatically optimization methods:
  - Re-weighted EMSC
  - Simplex-optimized EMSC
  - Direct Orthogonalization-based EMSC)

During later *prediction* pre-processing, a previously ”calibrated” (defined) pre-processing method is read and applied to pre-process new spectra.

In addition to the many pre-defined standard method combinations, the user may set up new preprocessing method combinations. The various method combinations may be applied one at a time, or run automatically in series in order to search for an optimal pre-processing.

The Matlab program take simple Matlab files as in put and output; the standard format of these files is suitable for export/import in e.g. The Unscrambler(TM).

Method summary

The methodology can identify and separate various effects in multichannel measurements, making the measurements more suitable for e.g. multivariate calibration, reducing the number of PCs required and improving the robustness and predictive ability. It enhances the traditional MSC or ISC by explicit model-based separation of the main contribution sources in the spectra.

For instance, in diffuse light absorbance spectra obtained by transmission as log/(1/T), or by reflectance as log(1/R) or Kubelka-Munk transform (1-R)/2R, it can:

1) Estimate and separate *multiplicative* physical effects (path length, light scattering) from *additive* chemical effects (absorbance of analytes and interferants) and *additive* physical effects (temperature, baseline variations etc).
2) Estimate and remove *identified, but undesired* ”physical” and ”chemical” interference effects, while retaining *identified, but desired* effects as well as *unidentified* effects in the data. This may be done either by mixed multiplicative and additive modelling (EMSC and its inverse, EISC), or by purely additive modelling (SIS).
3) Read *known* analyte and interferant spectra input from file, and/or automatically *estimate* unknown analyte and interferant spectra by Direct Orthogonalization or by Simplex optimization, based on known Y-values.
4) Identify and down-weigh instrument channels (e.g. wavelengths) that destroy the model-based pre-processing, by *reweighting*.

The EMSC/EISC approach appears to be applicable for a number of different types of input data - for different types of spectroscopy, (UV / vis./ NIR / IR / Raman / fluorescence / multichannel imaging), for chromatography and electrophoresis, for micro-array data as well as for sensory data.

Documentation and functionality:

Downloading the software and documentation:
http://www.models.kvl.dk/source/EMSCtoolbox/index.asp

Start: Installation/Getting Started.pdf

This manual: Using the EMSC Toolbox.pdf

Illustrations: Various diffuse vis./NIR/IR applications from Martens et al. (2003)[3)] are demonstrated with the software, (e.g. DataCase=199). They are summarised in file EMSCExamples.ppt

References:

[1] Martens, H. and Stark, E. (1991) Extended multiplicative signal correction and spectral interference subtraction: New preprocessing methods for near infrared spectroscopy. J.Pharmaceutical & Biomedical Analysis **9** (8), 625-635.

[2] Pedersen, D.K., Martens, H., Pram Nielsen, J. and Balling Engelsen, S. (2002): Light absorbance and light scattering separated by Extended Inverted Multiplicative Signal Correction (EIMSC). Analysis of NIT spectra of single wheat seeds.
Applied Spectroscopy. **56** (9) 1206-1214.

[3] Martens, H., Pram Nielsen, J. and Balling Engelsen, S. (2003) Light Scattering and Light Absorbance Separated by Extended Multiplicative Signal Correction. Application to Near-Infrared Transmission Analysis of Powder Mixtures**.**
Anal. Chem. **75** (3), 394 – 404.

Patents

The EMSC/EISC methodology is covered by:

US Patent 5,568,400, E. Stark and H. Martens:
"Multiplicative Signal Correction Method and Apparatus", and
European patent 0415401, E. Stark and H. Martens:
"Improved Multiplicative Signal Correction Method and Apparatus".

Academic use of this code is free, but commercial use of it requires a license from the patent holders. Please contact: StarkEdw@aol.com or Harald.Martens@matforsk.no.

1. Method overview

1.1 Calibration pre-processing

This consists in the following steps:

1) Defining the file name for input spectra of a set of samples (rows in matrix **Z**) from a Matlab data file <InputFileZ>, as well as other optional input files.
2) Setting up a pre-processing method (certain combination of method control parameters and ”model spectra” meta-parameters) defined by the user by assigning an integer value, e.g.:
    - 0 (interactive definition of inputs and method)
    - 103 (default file names EMSC_Z.mat and EMSC.Y.mat, method=EMSC with default settings)
    - 1000 (a list of all pre-defined inputs and method)

    Hence, once the user has saved the input data to the default file name, the main user interaction consists in (repeatedly) giving one single index, here called DataCase.
3) The program then automatically reads the input data from the specified file(s), and submitting these to the specified pre-processing method(s). Then the program automatically also does the following:

- Saves the corrected spectrum for each of the samples to file EMSCTreated_<InputFileZ>
- Saves the estimated model parameters for each of the samples to file EMSCModParam_<InputFileZ>
- Saves the correction method, including model meta-parameters, to a model file EMSCModel_<InputFileZ> (default: EMSCModel_Z.mat) for later prediction use.
- Saves some files with more technical details
- Plots modelling results at the user’s chosen level of report detail.

1.2 Prediction pre-processing

This consists in the following steps:

1) Defining the file name for input spectra of a set of samples (rows in matrix **Z**) from a Matlab data file < InputFileZPred >, e.g. EMSC_ZNew.mat
2) Defining which pre-processing method to be used, with meta-parameters, previously defined in file EMSCModel_<InputFileZ> (default: EMSCModel_Z.mat), e.g. (conf.
    - -1 (interactive definition of input files and method)
    - -3 (default file names EMSC_Z.mat and EMSC.Y.mat, EMSC method from EMSCModel_Z.mat).
3) Automatically reading the input data from the specified file(s), and submitting these to the specified pre-processing method(s). This automatically also does the following:
    - Saves the corrected spectrum for every sample, to file EMSCTreated_<InputFileZPred>
    - Saves the estimated model parameters for every sample, to file EMSCModParam_<InputFileZPred>
    - Saves some files with more technical details.

2. Program overview

2.1 Program options

Table 1a) outlines the options for interactive input of data and method definitions, while Tables 1b),1c) and 1d) list various pre-defined calibration and prediction methods. Table 1e) summarises the options for interactive program control.

2.2 User-defined options

The user may define new input-and-method calibration combinations in file EMSC_GetUserDefinedDataCases.m, saved in the user's own directory, and run them individually by specify their index, or running all of them and compare the results, by giving index 99. The user may specify new prediction methods in file EMSC_GetUserDefPredDataCases.m

2.3 Starting the program

The full EMSC/EISC program packages is started by

EMSC_Main.m

This includes routines to handle input/output, graphics, method comparisons for a given input set of "spectra", **Z**. If data for a target ("analyte") variable **Y** are given, then it also allows method optimization by estimation of model meta-parameters, based on Direct Orthogonalization or Simplex optimization.

A program for EMSC/EISC pre-processing of a given set of spectra **Z** according to a given pre-processing method, using a given, known set of model meta–parameters, is

function [Zcorrected,...]= EMSCEISC (Z ...)

This function requires no Y-data, but does not allow method optimization. A demo of its use is given in program

TestEMSC.m (Note: TestEMSC is not finished yet)

Program EMSC_Main.m

1. **Start the program EMSC_Main.m as described in *Getting Started*.**

2. Control overview:

The program usually takes only one single number as interacive input - the method number (called DataCase) (see Table 1e):
For DataCase ≤ 1000 or ≥ 1000 allow program controls. For instance,
DataCase = 1000 gives an overview of the pre-defined methods available
DataCase = ±3000 toggles the level of plotting,
DataCase = ±4000 toggles pause between each method, while
DataCase = -1000 stops the program.
For details, see section XXXXa.

DataCase values between –999 and +999 represent actual pre-processing methods:
DataCase ≥ 1 provides pre-defined calibration methods,
DataCase ≤ -1 provide pre-defined calibration methods.
DataCase = 0 provides full interactive input for calibration
DataCase = -1 provides full interactive input for prediction
For details, see section XXXXb.

2.1 Calibration control

In a calibration pre-processing, a ”method” (combination of parameter settings and meta-parameter values) is defined, either interactively (by giving DataCase=0) or in method default definition files (conf. Table 1a-d):

DataCase 1 to 98: MyDirectory\EMSC_GetUserDefinedDataCases.m
(where the user may define favourite methods)
or
DataCase between 100 to 998: EMSC\EMSCGetInternalDataCases.m
(The Toolbox pre-defined methods)
Some values of DataCase give a sequence of calibration methods, e.g.
DataCase= 99: All calibration cases defined in
MyDirectory\ EMSC_GetUserDefinedDataCases.m
DataCase= 199: All calibration cases from file EMSC\EMSCGetInternalDataCases.m
For details, see section XXXXc.

2.2 Prediction control

In a prediction pre-processing, the file name of the ”method” may be given interactively (DataCase = -1), or read from a file defined in files
MyDirectory\EMSC_GetUserDefPredDataCases.m
(the user’s favourite methods, DataCase between -5 and -98)
or
EMSC\EMSCGetDefaultInputDataForPred.m
(pre-defined methods, DataCase between –2 and -5).
Some values of DataCase give a sequence of prediction methods, e.g.
DataCase= -99: All predictions defined in MyDirectory\
EMSC_GetUserDefPredDataCases.m

3 Calibration based on the user's spectra Z

3.1 Controlling the method and input

Detailed illustration of how calibration methods are defined, see

MyDirectory\EMSC_GetUserDefinedDataCases.m (DataCase 1-98)
and
EMSC\EMSCGetInternalDataCases.m (DataCase 100-998)

In both these programs a method definition structured as :

```
if DataCase== <a number n1>...
 ... define some methods
elseif DataCase==<n2>
        %Comments: define method #n2
        DataCaseName=' MyMethod n2';
        (define method  here, by giving file names or control parameters)
        ...
elseif DataCase==<n3>
        DataCaseName=' MyMethod n3';
        (define method  here, by giving file names or control parameters)
else...
        (nothing)
end
```

A method is controlled in part by which ***file names*** are given, and in part which ***control parameters*** are given. For instance, including the statement

WgtFileName = 'EMSC_Wgt.mat';

forces channel weights to be read explicitly from a weight file by the name of EMSC_Wgt.mat, instead of being set internally to 1 for all wavelength,
while the control statement

nWeightIter=3

lets the weights be re-estimated 3 times in iterative least squares estimation, (see below).

For simplicity, it is recommended that you replace all the *.mat files in c:\Matlab\toolbox \EMSC\MyDirectory by your own corresponding files. Then you may test all the pre-defined method combinations directly without further programming, e.g. all the EMSC methods with DataCase=199.
Alternatively, you may use full interactive control (obtained by DataCase=0).

Method definition options are given in Table 1 for calibration pre-processing.

3.2 File format definition

All the input files follow the same standard Matlab, with a data matrix called Matrix and two character arrays representing names for the rows and columns in the data matrix:

Matrix (data matrix of size nObj x nZVar)
ObjLabels (character array of size nObj)
VarLabels (character array of size nZVar)

A simple way to obtain files with this format is to export data matrix from e.g. The Unscrambler(TM) as Matlab file <FileName>. It is then saved as <FileName>.mat.

Alternatively, the file may be saved from within Matlab itself: When Matrix, ObjLabels and VarLabels have been defined within Matlab, type

```
save <FileName> Matrix   ObjLabels   VarLabels <return>
```

This format is used for all input data files: Spectra to be pre-processed, and the different optional data files, e.g. for target variable Y, for known good or bad constituent spectra, etc etc,.
(conf. Section YYYY).

3.3 Defining the data input file for spectra Z

To be able to use all the pre-defined methods, save each dataset to be pre-processed in a separate directory (<MyDirectory>), with the file name

EMSC_Z.mat

This file should contain **Matrix, ObjLabels** and **VarLabels** as explained in the previous section. By having different datasets in different directories, this will
Alternatively, the user may save spectra for pre-processing in other file names, e.g. <ZFileName>.mat, and define this as input for new methods in file
MyDirectory\ EMSC_GetUserDefinedDataCases.m:

```
if DataCase= =... (other methods)
...
elseif DataCase= =<a new number between 1 and 98>
        DataCaseName='My latest data , EMSC default'
        ZFileName= 'ZFileName.MAT ';

        <Followed by optional statements for this method,
                to define deviations from the default conditions, see below>
elseif DataCase = = ... ( next method)
...
end
```

3.4 Optional input files

Since the various default methods require various additional input options, the following files must also be defined with the same format (with Matrix, ObjLabels and VarLabels). The most important optional file is EMSC_Y.mat, containing a data matrix whose first column is taken as target variable to optimized for.

Note that *errors* may occur if you attempt to use all the default-definitions of methods to analyse your own data in file EMSC_Z.mat *without* first redefined all these extra quantitative model parameter files, because then the program tries to combine your input data EMSC_Z.mat, store in <MyDirectory>, with the demo files for e.g. known constituent spectra, stored in directory ....\EMSC, and they may not have the right formats or contents.

EMSCMakeDefaults reads EMSC_Z.mat and EMSC_Y.mat and generates all the optimal files in a compatible format (see next section).

EMSC_Y.mat (<MyYFile>)

Target variable (usually: Analyte concentration) required for assessment and optimization of the pre-treatment.

Defined by statement in MyDirectory\ EMSC_GetUserDefinedDataCases.m

YFileName= <MyYFile>;
Default: YFileName= 'EMSC_Y.mat';

The values for the nObj objects in EMSC_Z.mat, used for optimizing the EMSC model. Let Matrix (size nObj x 1) contain Y-values, with ObjLabels as in EMSC_Z.mat and VarLabels= e.g.'MyYValues'.

EMSC_Wgt.mat (<MyZWeightFile'>)

Reliability weights (ChannelWeights, between 0 and 1) for the nZVar variables in EMSC_Z.mat, (e.g. 1 for normal channels, 0 or 0.001 for channels with problems).
Defined by statement
WgtFileName = < MyZWeightFile'>;
Default: WgtFileName='EMSC_Wgt.mat';

The weights are used for making the modelling insensitive to variations in certain regions. Let Matrix (size 1 x nZVar) contain weights, e.g. very high noise or very high variations of "unknown" type (like the water peak around 1940nm in NIR spectra, if water spectra are not included in the model), with VarLabels as in EMSC_Z.mat, ObjLabels = e.g. 'MyZWeights'.

The ChannelWeights be optimized automatically, by setting integer nWeightIter>0, e.g.

nWeightIter=3

for three iterations in the iterative re-estimation of the weights based on how well the different wavelengths are fitted by the chosen EMSC model.

EMSC_Ref.mat (<MyZReference'>)

Reference spectrum **m**; to be used for estimating and correcting for the multiplicative interference effects.
Defined by statement
RefFileName = < MyZReference'>;

Default: RefFileName ='EMSC_Ref.mat';
Let Matrix (size 1 x nZVar) contain the reference spectrum, e.g. mean of the spectra in EMSC_Z.mat, with VarLabels as in EMSC_Z.mat, ObjLabels =e.g. ‘MyZReference’.

May be optimized automatically by simplex optimization, by defining
OptPar=1

EMSC_GoodSpectra.mat (<MyGoodSpectra’>)
Spectra of known analytes considered as “good” or desirable, and therefore retained in the corrected spectra. Let Matrix (size nGoodComponents x nZVar) contain one or more constituent spectra or difference spectra, with VarLabels as in EMSC_Z.mat, ObjLabels (size nGoodComponents) =e.g. [‘MyGoodEffect1’;’MyGoodEffect2;...’]

Note: To avoid collinearity problems for the reference spectrum **m**, use difference-spectra in EMSC_GoodSpectra.mat (see discussion in EMSC-Theory and Martens et al 2003).

Defined by the statement
FileNameGood = < MyGoodSpectra’>;
Default: FileNameGood = [ ]
i.e. no good component spectra known.

A new ‘Good component’ spectrum may be estimated automatically by Direct Orthogonalization, (projection of Z on Y), by
the statement
OptPar=-1 or OptPar=-2

A known or new ‘Good component’ vector may be optimized automatically by simplex optimization, by defining
OptPar=3

EMSC_BadSpectra.mat
Spectra of known interferants considered as “bad” or undesirable, and therefore estimated and removed in the corrected spectra. Let Matrix (size nBadComponents x nZVar) contain one or more interferant spectra or difference spectra (see Martens et al 2003), with VarLabels as in EMSC_Z.mat,
ObjLabels (size nBadComponents) =e.g. [‘MyBaddEffect1’;’MyBaddEffect2’;...]

Note: To avoid collinearity problems for the reference spectrum **m**, use difference spectra in EMSC_BadSpectra.mat (see Martens et al 2003).

Defined by the statement
FileNameBad = < MyBadSpectra’>;
Default: FileNameBad = [ ]
i.e. no bad component spectra known.

One or two new ‘Bad component’ spectra may be estimated automatically by Direct Orthogonalization, (SVD of residuals after projection of Z on Y), by the statement

OptPar=-1 or OptPar=-2, respectively.

A known or new ‘Bad component’ vector may be optimized automatically by simplex optimization, by defining

OptPar=2

3.5 Making dummy versions of optional input files.

For simplicity, “dummy” parameter versions of these quantitative parameter files may alternatively be set up by running the program EMSCMakeDefaults.m, which defines them more or less automatically from your data in file EMSC_Z.mat; conf. Appendix ZZZ. This lets you see how the program works with its many options.

To start the program, type

EMSCMakeDefaults<return>

and follow the primitive dialogue. More details are given in xxxx. You may later replace some of these “dummy” quantitative parameter files from e.g. The Unscrambler(TM), by a more consciously chosen contents, to tailor the methodology.

Once the default files for quantitative parameters have been saved, e.g. in EMSCMakeDefaults, one can start RunEMSC.m as described in XXX., testing e.g. all pre-defined EMSC methods by giving DataCase=199.

However, the user should try to replace the quantitative files, like EMSC_GoodSpectra, EMSC_BadSpectra

3.6 Define new calibration pre-processing *method* settings

Note that this re-programming is only required if other file names than the above-mentioned defaults names have been chosen, and/or if now parameter combinations are desired.

3.7 Output files

For any DataCase (calibration or prediction):

3.7.1 Pre-processed spectra

The pre-processed version of the input spectra , Zcorrected, are saved in standard format to file

EMSCTreated_<ZFileName>.mat, e.g. EMSCTreated_Z.mat

(Note that since this <ZFileName>=’**EMSC_**Z.mat’starts with **EMSC_**, these are dropped in order to avoid excessively long names like

EMSCTreated **_EMSC_** Z.mat').

3.7.2 Pre-processing parameter estimates

The estimated parameters used for pre-processing the input spectra for the individual samples are saved in standard format to file

EMSCModParam_<ZFileName>.mat, e.g. EMSCModParam_Z.mat

3.7.3 Pre-processing residual spectra

The lack of fit between the input regressand spectra and the EMSC/EISC regressor model is saved in standard format to file

EMSCRes _<ZFileName>.mat, e.g. EMSCRes _Z.mat

3.7.4 Corrected component concentrations.

If spectra for good or bad component spectra were used as part of the linear EMSC or EISC models, then the "concentration" estimates of these components are corrected for multiplicative effect and saved in standard format to file

EMSCEstCompConc_<ZFileName>.mat, e.g. EMSCEstCompConc _Z.mat

For non-negative DataCase (calibration only):

3.7.5 Pre-processing model spectra

The meta-parameters (model spectra) used for estimating the pre-processing parameters for this set of samples samples are saved in standard format to file

EMSCModSpectra_<ZFileName>.mat, e.g. EMSCModSpectra_Z.mat

3.7.6 The Model output file

Each time a calibration pre-processing has been run, a new, complete method definition file ("ModelFile") has been saved. This ModelFile contains a complete method definition file may later be used for prediction pre-processing of new data sets. Its filename is 'EMSCModel_<ZFileName>.mat'
With the default input file name <ZFileName>='EMSC_Z.mat', the automatic model file name is

ModelFile ='EMSCModel_Z.mat';

4 Prediction pre-processing: Define input data and pre-processing method

Prediction pre-processing means to submit spectra in a data file, e.g. EMSC_Z.mat, to pre-processing according to a previously stored method definition file, e.g. EMSCModel_Z.mat. This is obtained with DataCase<0.

4.1 **Defining the files**

DataCase= -3 is the simplest prediction default. It reads:
Z-data from file EMSC_Z.mat and
Y-data from file EMSC_Y.mat, and
the EMSC model from file EMSCModel_Z.mat .

Other defaults:

DataCase= -1	Interactive input for data Z and Y,	Interactive input model EMSCModel__Z.
DataCase= -2	Interactive input for data Z and Y,	Default model EMSCModel_Z.
DataCase= -3	Default EMSC_Z.mat, no Y,	Default model EMSCModel_Z.mat
DataCase= -4	Default Z.mat, Y.mat	Default: old model EMSCModel_Z.mat
DataCase= -5	Default Z.mat, Y.mat ,Reduced plot,	Default: old model EMSC_ZModel_Z.mat

The basic defaults for prediction pre-processing are defined in file
EMSC\EMSCGetDefaultInputDataForPred.m (DataCases –1,-2,-3,-4-,5)
MyDirectory\EMSC_GetUserDefPredDataCases.m (DataCases≤-6, but >-98)

5. Plots

(not finished yet)

Table 1a – Interactive data inputs and methods

Interactive inputs:
DataCase= 0: Interactive input for calibration
DataCase=-1: Interactive input of prediction data and old cal. model file
DataCase=-2: Interactive input of prediction data, default cal. file EMSC_ZModel

Table 1b – Pre-defined data inputs and EMSC/EISC methods
DataCase= between 1 and 98: User-defined calibration cases, specified in file EMSC_GetUserDefinedDataCases.m
DataCase= 99: All user-defined calibration cases

Default EMSC data cases for calibration, defined in file EMSCGetDefaultInputData.m:
DataCase=100: No pre -treatment
DataCase=101: Spectral Interference Subtraction (SIS)
DataCase=102: MSC
DataCase=103: EMSC physical,default
DataCase=106: EMSC, physical & input Good Spectra from file
DataCase=107: EMSC, physical & input Bad Spectra from file
DataCase=108: EMSC, physical & Good & Bad Spectra from file
DataCase=109: Default EMSC with modelling of Ref^2
DataCase=110: Default EMSC with modelling of Ref^2,but not subtraction
DataCase=111: Default EMSC, but no subtraction of physical effects
DataCase=121: EMSC,Automatically estimated 1 GoodSpectra and 1 BadSpectra
DataCase=122: EMSC,Automatically estimated 1 GoodSpectra and 2 BadSpectra
DataCase=131: MSC, Weights from file Wgt.mat
DataCase=132: EMSC default, Weights from file Wgt.mat
DataCase=133: 3 x Reweighted MSC,
DataCase=134: 5 x ReWeighted EMSC
DataCase=135: 5 x ReWeighted EMSC with GoodSpectra and BadSpectra
DataCase=151: EMSC, opt. the Ref.spectrum, starting from the mean spectrum
DataCase=152: EMSC, opt. a Bad comp.
DataCase=153: EMSC, opt. a Good comp.
DataCase=154: EMSC, opt.the Ref.spectrum, starting from the file Ref.mat
DataCase=155: EMSC, opt.an extra Bad spectrum, in addition to input BadSpectra
DataCase=156: EMSC, opt. one good spectrum, in addition to input GoodSpectra
DataCase=157: EMSC, opt. one bad spectrum, in addition to input GoodSpectra
DataCase=158: EMSC, opt. one good spectrum, in addition to input BadSpectra
DataCase=159: EMSC, opt. one good spectrum, in addition to input BadSpectra
DataCase=160: EMSC, opt. 2D one good spectrum, in addition to input GoodSpectra and BadSpectra
DataCase=161: EMSC, opt. 3D one good spectrum, in addition to input GoodSpectra and BadSpectra
DataCase=162: EMSC, opt. 3D ref spectrum, in addition to input GoodSpectra and BadSpectra
DataCase=163: EMSC, opt. 3D ref spectrum, in addition to input GoodSpectra and BadSpectra
DataCase= 199: All demos of EMSC calibration

Default EISC data cases for calibration, defined in file EMSCGetDefaultInputData.m:
DataCase=201: ISC
DataCase=202: EISC physical,default
DataCase=203: EISC physical & opt. Good spectrum
DataCase=204: EISC physical & opt. mean Ref spectrum
DataCase=205: EISC, physical & input Good Spectra from file
DataCase=206: EISC, physical & input Bad Spectra from file
DataCase=207: EISC, physical & Good & Bad Spectra from file
DataCase=208: Default ESC with modelling of Ref^2
DataCase=209: Default EISC with modelling of Ref^2,but not subtraction
DataCase=210: Default EISC, but no subtraction of physical effects
DataCase=211: ISC, Weights from file Wgt.mat
DataCase=212: EISC default, Weights from file Wgt.mat
DataCase= 299: All demos of EISC calibration
DataCase= 599: All demos of EMSC/EISC calibration,
from files EMSCGetUserDefinedDataCases.m and EMSCGetInternalDataCases

Table 1c – Default prediction methods

Default data cases for prediction, defined in file EMSCGetDefaultInputDataForPred.m:

DataCase= -99: Some prediction cases,
DataCase=-3: PredCase=Pred. Defaults, DataCaseCal= <whatever whas run last>, e.g. EMSC physical,default,
DataCase=-4: Pred.Defaults + Y from EMSC_Y.mat
DataCase=-5: PredCase=Pred. Default EMSC_Z.mat, EMSC_Y.mat, use old EMSC_Z model
DataCase= -99: Some prediction cases

DataCase between –6 and –98 may be defined by the user as new prediction cases
in file GetUserDefPredDataCases.m

Table 1d – Illustration of how the methods were used in Martens et al. 2003, 75 (3), 394 – 404

Cases published by Martens et al. 2003, defined in file EMSCGetDefaultInputData.m:
DataCase=301: MSC, as published by Martens et al. in Analytical Chemistry 2003
DataCase=302: EMSC, as published by Martens et al. in Analytical Chemistry 2003
DataCase=303: EMSC, as published by Martens et al. in Analytical Chemistry 2003
DataCase=304: Spectral Interference Subtraction (SIS)
DataCase= 399: All MSC/EMSC/EISC calibration examples from Martens et al. 2003

Table 1e – Interactive control of program operation

1000=list options,	-1000=stop
2000=turn on printing of plots,	-2000=turn off printing of plots
3000= more plots,	-3000=less plots
4000=pause between DataCases,	-4000=no pause between DataCases

Table 2b – Mandatory non-default settings for new calibration methods

The default parameter settings are defined in files
EMSCGetDefaults.m and
EMSCGetOptimizationDefaults.m

The pre-defined method settings from Table 1 are specified in file
EMSCGetInternalDataCases.m

User-defined method settings:
For each DataCase (between 1 and 98) in file
EMSC_GetUserDefinedDataCases.m
the user may specify all non-default parameter settings,
as examplified e.g. in file
EMSCGetInternalDataCases.m

The following minimum information is mandatory for a new Method:

if DataCase== <NUMBER>

DataCaseName=' SOME TEXT;
Note that text variable **DataCaseName** MUST be defined for each DataCase, otherwise the method will be ignored.

ZFileName(char): Name of input file containing spectral data to be pre-processed

YFileName(char): Name of file containing target variable data, e.g. analyte concentrations.
(default=EMSC_Y.mat, i.e. Y= column 1 in the data matrix in file EMSC_Y.mat)
Note: Data in this file is only **mandatory** if model optimization is required (OptPar~=0), otherwise it is optional.

Table 2b - Optional non-default settings for new calibration methods

Optional input file information:

WgtFileName (char) name of weight file (default=[ ]; all elements in ChannelWeights =1),
The ChannelWeights may be modified automatically if nWeightIter>0
FileNameGood (char): name of file containing spectral data of the good components to be modelled but
not subtracted (default=[ ], i.e. no good spectra to be modelled)
FileNameBad (char): name of file containing spectral data of the bad components to be modelled
and subtracted; (default=[ ], i.e. no bad spectra to be modelled)
RefFileName (char): name of file containing reference spectrum (default=[ ]; i.e. defines reference
spectrum internally, as mean(Z))

Optional EMSC/EISC method control:

MscOrIsc (scalar) 1=MSC (or EMSC), -1=ISC (or EISC) (Default=1)
ModRef (scalar) (Default=1)
MSC/ISC/EMSC/EISC:ModRef=1.
SIS: MscOrIsc=1,ModRef=0

ModOffset(scalar): offset in the model(Default=1)
0= no offset modelling,
1= modell, and subtract the estimated effect,
-1= model, but do not subtract the estimated effect,
ModChannel(scalar), channel vector (-1:1) in the model(Default=1)
0= no channel vector modelling
1= model and subtract the estimated effect
-1= model, but do not subtract the estimated effect
ModSqChannel(scalar),squared channel vector (-1:1)^2 in the model(Default=1)
0= no squared channel vector modelling
1= model and subtract the estimated effect
-1= model, but do not subtract the estimated effect
ModSqSpectrum(scalar) a squared spectrum in the model (Default=0)
MSC/EMSC: squared reference, ISC/EISC: squared individual spectrum
0= no squared spectrum modelling
1= model and subtract the estimated effect
-1= model, but do not subtract the estimated effect

Optimization controls:

OptPar=(scalar) defines optimization (Default: 0, i.e. no optimzation)
OptPar=1 : optimize Reference spectrum by simplex search in the AsearchDim first PCs from Z
OptPar=2 : optimize a new “good” vector by simplex search in the AsearchDim first PCs from Z,
OptPar=3 : optimize a new “bad” vector by simplex search in the AsearchDim first PCs from Z,
OptPar=-1: estimate a new “good” and a new ”bad” spectrum, by direct orthogonalization (projection of Z on Y)
OptPar=-2: estimate one new “good” and two new ”bad” spectra, by direct orthogonalization (projection of Z on Y)

nWeightIter (scalar) # of reweighted estimates of the parameters,
using updated ChannelWeigths (default=0)

Parameters not yet tested enough, so don’t use them yet:

ASearchDim number of svd dimensions in the raw data to be combined
in SIMPLEX optimization (OptPar>0)
CondNumber (scalar) condition number, e.g. 10^10 (default CondNumber=10^12)
FactorNeeded(scalar) conservatism factor for finding optimal model rank,
PunishLongC, PunishHighA, SamplingPrec, ToleranceRMSEP,
MaxIter maximum SIMPLEX iterations
PlotIt (scalar) 1= plot all, 0=reduced plot
AMax (max # of PCs in regression)
jY (scalar) Index if there are more than one Y-variable read
(ModRef=2: Estimate the scaling factor b(i) from the sum of the estimated concentrations of good chem.components)
(ModRef=3: Estimate the scaling factor b(i) from the sum of the estimated concentrations of all chem. components)

Table 3 - Options for non-default settings for prediction

For each DataCase (between -6 and -98)
in file EMSC\MyDirectory\EMSC_GetUserDefPredDataCases.m
define:

Mandatory information for each DataCase:

PredCaseName= some text" ;

ZFileName(char): Name of file containing spectral data

YFileName(char): Name of file containing spectral data
(default=EMSC_Y.mat , i.e. Y= column 1 in the data matrix in file EMSC_Y.mat)

ModelFile=(char) Name of the old calibration pre-processing file to be used
(default='EMSCModel_Z.mat')

In addition, calls to the system, such as

```
!copy MyDataFile.mat   EMSC_Z.mat
!copy MyPreviousEMSCModelfile.mat  EMSCModel_Z.mat
```

may be included and will be executed prior to reading any data or model parameters.

Appendix 1

File contents, which may be modified by the user:

EMSC_SetPath.m Sets your Matlab path so that software files and demonstration data files in directory ...Matlab\toolbox \EMSC can be accessed from the present directory,
...Matlab\toolbox \EMSC\MyDirectory. To be run at the start of every Matlab session.

EMSC_GetUserDefinedDataCases.m Allows you to define your own standard input- and method combinations for ”calibration pre-processing” – i.e. for setting up models that may later be used for ”prediction pre-processing” with setups defined in file EMSC_GetUserDefPredDataCases.m. See Appendix 1.

EMSC_GetUserDefPredDataCases.m Allows youto define your own standard input- and method combinations for ”prediction pre-processing”. See Appendix 1

RunEMSC.m Starts the EMSC pre-processing program. See Appendix 2.

Default input data-files:

<u>*Required*</u>:

EMSC_Z.mat <InputFile> Spectra to be pre-processed, containing the three
Matlab elements:
Matrix (objects x variables), ObjLabels, VarLabels

<u>*Mandatory if all default methods are to be tested, otherwise optional*</u>:

EMSC_Y.mat Y-values for objects in <InputFile>;
may be used for optimizing the EMSC model.

EMSC_Wgt.mat Reliability weights for the variables in<InputFile>;
may be used for making the modelling insensitive to variations
in certain regions, e.g. extreme water peaks in NIR (weight = 0)

EMSC_Ref.mat Default reference spectrum **m**;
may be used for defining a reference spectrum different from
the mean of the input spectra in <InputFile>

EMSC_GoodSpectra Spectra of known analytes considered as “good” or desirable, and therefore retained in the corrected spectra.

EMSC_BadSpectra Spectra of known interferants considered as “bad” or undesirable, and therefore estimated and removed in the corrected spectra.

In addition, some output files from previous runs follow;
they will be overwritten when the program is started:

EMSCTreated_Z.mat EMSCTreated_<InputFile>
EMSC-treated versions of the “spectra”
from file _<InputFile> =EMSC_Z.mat

EMSCModSpectra_Z.mat EMSCModSpectra_<InputFile>
EMSC-correction model used

EMSCModParam_ Z.m EMSCModParam_<InputFile>
EMSC-correction parameter estimates

EMSCModel_Z EMSCModel_<InputFile>
Complete EMSC-correction model, may be
used for “prediction pre-processing” of new spectra,
without changing
EMSCModSpectra_<InputFile> or
EMSCModel_<InputFile>

EMSCRes_Z EMSCRes_<InputFile> EMSC Residuals

OptEMSC OptEMSC (optimized EMSC model; will probably be changed later)

EMSCMakeDefaults.m Optional quick-and-dirty set-up of the above “dummy” parameter files for all the possible quantitative parameters, in order to allow you to test all default method definitions automatically. Useful for preparing defaults from the user’s own input file. See Appendix 3.

Appendix 2

EMSC_GetUserDefinedDataCases.m Allows you to define your own standard input- and method combinations for ”calibration pre-processing” – i.e. for setting up models that may later be used for ”prediction pre-processing” with setups defined in file EMSC_GetUserDefPredDataCases.m. See Appendix 1.

EMSC_GetUserDefPredDataCases.m Allows youto define your own standard input- and method combinations for ”prediction pre-processing”. See Appendix 1

Appendix 3

EMSCMakeDefaults.m Quick-and-dirty set-up of “dummy” parameter files for all the possible quantitative parameters, in order to allow you to test all default method definitions automatically.

EMSCMakeDefaults.m has sub-programs

EMSCMakeRef.m, (Suggests to use the mean as reference spectrum)
EMSCMakeWeights.m, (allows a noise limit to be entered,
to weigh down channels above this)
EMSCMakeGoodBadSpectra.m (uses a method akin to Direct Orthogonalization)

Subroutine EMSCEISC
and
program TestEMSC.m
(Doc. not finished yet)