Contents

EXAMPLE / CALIBRATION of WEIGHT MASSES

Viktor Witkovsky (witkovsky@savba.sk) Ver.: '14-Feb-2024 17:57:36'


```
clear
close all
```

CALIBRATION of WEIGHT MASSES

Problem specified by Jaroslav ZUDA / Czech Metrology Institute (CMI) E-MAIL: jzuda@cmi.cz Related EMPIR Project: Improvement of the realisation of the mass scale Short Name: RealMass, Project Number: 19RPT02

The fundamental challenge faced by any primary laboratory is ensuring the accurate calibration of weights. While the procedure for routine calibration of weights of lower accuracy classes is well-documented in OIML R111, procedures for calibration at the highest level are not as extensively covered. This issue is addressed by the RealMass project, which aims to develop such procedures.

Calibration at the highest level often requires the use of the split method for weight units. This procedure involves deriving individual weights from one or more reference weights using a measurement system with known weight values. Typically, these reference weights are prototypes made of a Pt and Ir alloy, although some institutes may use stainless steel weights. The weight is then calculated using the least squares method.

A significant challenge in accurate weight measurement is accounting for buoyancy forces. To address this, it is necessary to determine the density of the surrounding medium, which affects either the density or the volume of the weights. Additionally, the temperature of the environment must be considered since the volume of the weights changes with temperature.

There are various methods for determining the volume of weights, such as measuring the apparent weight in a liquid or comparing the apparent masses with a known volume body in the liquid. In recent years, there has been progress in measuring weights in a vacuum, particularly in light of the new definition of the unit of weight. However, measuring in a vacuum presents challenges due to surface phenomena on the weights. Therefore, it is often more practical to measure at low pressure, approximately corresponding to 50% atmospheric pressure, where surface changes are negligible.

Measurements in air and at low pressure can effectively determine both the volume and weight of weights. Comparisons indicate that this method yields results comparable to determining volume in a liquid. Consequently, there is interest in whether low-pressure measurement can be utilized for current weight and volume determination using the split method for weight units.

Load data

```
load EXAMPLE_MassCalibration.mat
```

Function of implicite restrictions on the model parameters

```
% function fun = funMassCalibration(mu,beta)
% % funMassCalibration - defines the necessary restrictions on the EIV model
% %  parameters for the calibration of weight masses.
% %
% %  funMassCalibration is an anonymous function that takes arguments mu and
% %  beta and returns fun, a q-dimensional column vector of constraints.
% %  Ideally, these constraints should equal zero. In this context, mu can be
% %  specified as either an m-dimensional column vector or as a cell array
% %  with one column vector {mu}, while beta is a p-dimensional vector. The
% %  output fun from funMassCalibration(mu, beta) represents the q
% %  constraints on the model parameters mu and beta. If the model
% %  restrictions are satisfied, fun = funMassCalibration(mu, beta) = 0.
% %
% %  Variable related to the vector parameter mu:
% %
% %  mu(1)  = X1 = X1R_1  % comparison measurement: M1 - MR in evironment 1
% %  mu(2)  = X2 = X2R_1  % comparison measurement: M2 - MR in evironment 1
% %  mu(3)  = X3 = X21_1  % comparison measurement: M2 - M1 in evironment 1
% %  mu(4)  = X4 = X1R_2  % comparison measurement: M1 - MR in evironment 2
% %  mu(5)  = X5 = X2R_2  % comparison measurement: M2 - MR in evironment 2
% %  mu(6)  = X6 = X21_2  % comparison measurement: M2 - M1 in evironment 2
% %  mu(7)  = X7 = X1R_3  % comparison measurement: M1 - MR in evironment 3
% %  mu(8)  = X8 = X2R_3  % comparison measurement: M2 - MR in evironment 3
% %  mu(9)  = X9 = X21_3  % comparison measurement: M2 - M1 in evironment 3
% %  mu(10) = dMR         % difference from nominal of the reference weight
% %  mu(11) = VR          % volume of the reference weight
% %  mu(12) = G           % gravitational ratio
```

```
% %  mu(13) = h1 = hR     % center of gravity of the reference weight
% %  mu(14) = h2 = h1     % center of gravity of the measured weight 1
% %  mu(15) = h3 = h2     % center of gravity of the measured weight 2
% %  mu(16) = K           % adjustment constant
% %  mu(17) = α           % coefficient of thermal expansion
% %  mu(18) = dT1 = dT_1  % temperature differnce T_1 - 20 in evironment 1
% %  mu(19) = dT2 = dT_2  % temperature differnce T_2 - 20 in evironment 2
% %  mu(20) = dT3 = dT_3  % temperature differnce T_3 - 20 in evironment 3
% %  mu(21) = ρ1  = ρ_1   % density of air in evironment 1
% %  mu(22) = ρ2  = ρ_2   % density of air in evironment 2
% %  mu(23) = ρ3  = ρ_3   % density of air in evironment 3
% %
% %  Variable related to the vector parameter beta:
% %
% %  beta(1) = dM1        % difference from nominal of the measured weight 1
% %  beta(2) = dM2        % difference from nominal of the measured weight 2
% %  beta(3) = V1         % volume of the measured weight 1
% %  beta(4) = V2         % volume of the measured weight 2
% %
% % EXAMPLE
% %  data  = [-9.7550e-08 -9.0350e-08  6.3200e-09 -4.7849e-07 9.4810e-08 ...
% %            3.8381e-07 -3.2604e-07 -9.3040e-08  2.3331e-07 6.4100e-07 ...
% %            1.2544e-04  3.0000e-07  3.6500e-02  3.6500e-02 3.0000e-02 ...
% %            9.9986e-01  4.8000e-05  8.0000e-01  4.0000e-01 9.0000e-01 ...
% %            1.1552e+00  5.8770e-01  8.1380e-01]';
% %  ux    = [ 5.0000e-10  3.0000e-10  2.0000e-10  3.0000e-10 3.0000e-10 ...
% %            3.0000e-10  2.0000e-10  3.0000e-10  2.0000e-10 4.0000e-08 ...
% %            1.0000e-09  3.0000e-08  5.0000e-04  5.0000e-04 5.0000e-04 ...
% %            5.0000e-06  2.0000e-06  1.0000e-01  1.0000e-01 1.0000e-01 ...
% %            5.0000e-04  5.0000e-04  5.0000e-04]';
% %  U     = diag(ux.^2);
% %  fun   = @(mu,beta) funMassCalibration(mu,beta)
% %  mu0   = data;
% %  beta0 = [mu0(10) mu0(11) mu0(10) mu0(11)]';
% %  clear options
% %  options.q = 9;
% %  options.method = 'oefpil2';
% %  options.criterion = 'function';
% %  options.tol       = 1e-15;
% %  result = OEFPIL(data,U,fun,mu0,beta0,options);
%
% % Viktor Witkovsky (witkovsky@savba.sk)
% % Ver.: 17-Feb-2024 13:16:08
%
% %% CHECK THE INPUTS AND OUTPUTS
%
% if iscell(mu)
%     mu = mu{1};
% end
%
% % Set the number of constraints
% q = 9;
% fun = zeros(q,1);
%
% %% Constraints based on equation (8) in [Zuda (2023)] / Environment 1
%
% fun(1) = mu(1)*mu(16) - ((1+beta(1))*(1-mu(12)*mu(14))) ...
%                           + ((1+mu(10))*(1-mu(12)*mu(13))) ...
%                           + (mu(21)*(1+mu(17)*mu(18))*(beta(3)-mu(11)));
%
% fun(2) = mu(2)*mu(16) - ((1+beta(2))*(1-mu(12)*mu(15))) ...
%                           + ((1+mu(10))*(1-mu(12)*mu(13))) ...
%                           + (mu(21)*(1+mu(17)*mu(18))*(beta(4)-mu(11)));
%
% fun(3) = mu(3)*mu(16) - ((1+beta(2))*(1-mu(12)*mu(15))) ...
%                           + ((1+beta(1))*(1-mu(12)*mu(14))) ...
%                           + (mu(21)*(1+mu(17)*mu(18))*(beta(4)-beta(3)));
%
% %% Constraints based on equation (8) in [Zuda (2023)] / Environment 2
%
% fun(4) = mu(4)*mu(16) - ((1+beta(1))*(1-mu(12)*mu(14))) ...
%                           + ((1+mu(10))*(1-mu(12)*mu(13))) ...
%                           + (mu(22)*(1+mu(17)*mu(19))*(beta(3)-mu(11)));
%
% fun(5) = mu(5)*mu(16) - ((1+beta(2))*(1-mu(12)*mu(15))) ...
%                           + ((1+mu(10))*(1-mu(12)*mu(13))) ...
%                           + (mu(22)*(1+mu(17)*mu(19))*(beta(4)-mu(11)));
%
% fun(6) = mu(6)*mu(16) - ((1+beta(2))*(1-mu(12)*mu(15))) ...
%                           + ((1+beta(1))*(1-mu(12)*mu(14))) ...
%                           + (mu(22)*(1+mu(17)*mu(19))*(beta(4)-beta(3)));
%
% %% Constraints based on equation (8) in [Zuda (2023)] / Environment 3
%
% fun(7) = mu(7)*mu(16) - ((1+beta(1))*(1-mu(12)*mu(14))) ...
%                           + ((1+mu(10))*(1-mu(12)*mu(13))) ...
%                           + (mu(23)*(1+mu(17)*mu(20))*(beta(3)-mu(11)));
%
% fun(8) = mu(8)*mu(16) - ((1+beta(2))*(1-mu(12)*mu(15))) ...
%                           + ((1+mu(10))*(1-mu(12)*mu(13))) ...
%                           + (mu(23)*(1+mu(17)*mu(20))*(beta(4)-mu(11)));
%
% fun(9) = mu(9)*mu(16) - ((1+beta(2))*(1-mu(12)*mu(15))) ...
```

```
%                           + ((1+beta(1))*(1-mu(12)*mu(14))) ...
%                           + (mu(23)*(1+mu(17)*mu(20))*(beta(4)-beta(3)));
%
% end
```

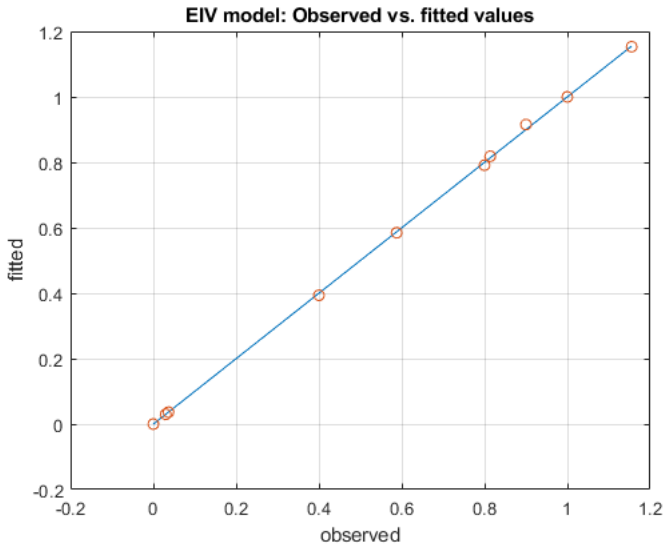
EXAMPLE

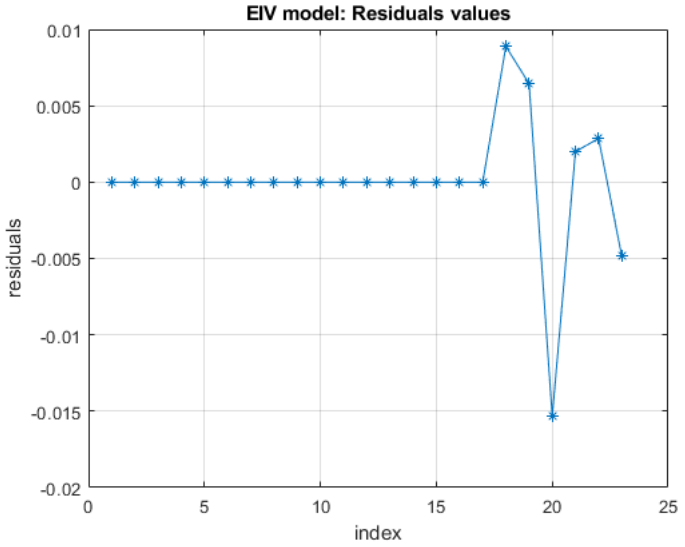
```
%U     = U_diag;
%U     = U_full;
U     = U_full2;
fun   = @(mu,beta) funMassCalibration(mu,beta);
mu0   = data;
% beta0 = [mu0(10) mu0(11) mu0(10) mu0(11)]';
beta0 = [0 0 0 0]';

clear options
options.q = 9;
options.method = 'oefpil2';
options.criterion = 'function';
options.tol       = 1e-15;
options.isEstimatedVariance = false;
```

OEFPIL fit

```
result = OEFPIL(data,U,fun,mu0,beta0,options);
```

```
-------------------------------------------------------------------------------
   OEFPIL ESTIMATION METHOD = oefpil2
   fun = @(mu,beta)funMassCalibration(mu,beta)
-------------------------------------------------------------------------------
   n    m     p    q    ITERATIONS         CRITERION              FUNCCRIT             FUNCCRIT_LIN            wRSS                RSS
   _    __    _    _    __________    ____________________    ____________________    ____________________    _____________    ____________________

   1    23    4    9         5         1.9412502980686e-16    1.9412502980686e-16    1.53129460506657e-19    208156.88669793    0.000391344976275638

-------------------------------------------------------------------------------
                  ESTIMATE                 STD                 FACTOR                LOWER                  UPPER                 PVAL
                _____________    ____________________    _____________    ____________________    ____________________    ____________________

   beta_1     -1.1279917379422e-07    4.00058100374311e-08    1.95996398454005    -1.91209120639936e-07    -3.43892269485045e-08     0.00480880712305643
   beta_2      7.54278548370493e-07    4.00048732788535e-08    1.95996398454005     6.75870437537851e-07     8.32686659203135e-07    2.69029446861711e-79
   beta_3      0.000124891615012911    1.22208267574956e-09    1.95996398454005      0.00012488921977488      0.000124894010250941                       0
   beta_4      0.000125642839434627    1.18168967328469e-09    1.95996398454005      0.000125640523365427     0.000125645155503828                       0

-------------------------------------------------------------------------------
```





RESULTS

```
mufit = result.mu;

% Estimated (fitted) differences from nominal weight 1kg
dMRfit_51699 = mufit(10);       % Reference weight: 51699
dM1fit_15N   = result.beta(1);  % Measured weight: 15N
dM2fit_15963 = result.beta(2);  % Measured weight: 15936

dmass = [ dMRfit_51699;  dM1fit_15N; dM2fit_15963];
massNominal = 1;
mass = massNominal + dmass;

% Standard uncertainty of the weight
u_MRfit_51699 = result.umu(10);  % Uncertainty of the reference weight: 51699
u_M1fit_15N   = result.ubeta(1); % Uncertainty of the measured weight: 15N
u_M2fit_15963 = result.ubeta(2); % Uncertainty of the measured weight: 15936

u_mass = [ u_MRfit_51699;  u_M1fit_15N; u_M2fit_15963];

% Estimated (fitted) volumes of the weights
VRfit_51699 = mufit(11);        % Volume of the reference weight: 51699
V1fit_15N   = result.beta(3);   % Volume of the measured weight: 15N
V2fit_15963 = result.beta(4);   % Volume of the measured weight: 15936

volume = [ VRfit_51699;  V1fit_15N; V2fit_15963];

% Standard uncertainty of the volumes
u_VRfit_51699 = result.umu(11);  % Uncertainty of the volume of the reference weight: 51699
u_V1fit_15N   = result.ubeta(3); % Uncertainty of the volume of the measured weight: 15N
u_V2fit_15963 = result.ubeta(4); % Uncertainty of the volume of the measured weight: 15936

u_volume = [ u_VRfit_51699;  u_V1fit_15N; u_V2fit_15963];
```

Best Estimates by OEFPIL

```
BestEstimates   = [mass;volume];

disp(table(BestEstimates))
```

```
   BestEstimates
  ________________

     1.00000064100001
    0.999999887200826
     1.00000075427855
  0.000125440000000005
  0.000124891615012911
  0.000125642839434627
```

Uncertainty (covariance) matrix of the OEFPIL Best Estimates

Estimated parameters of interest (M_R, M_1, M_2, V_R, V_1, V_2):

```
id_BestEstimates = [10 24 25 11 26 27];
U_BestEstimates = result.Umubeta(id_BestEstimates,id_BestEstimates);

disp(U_BestEstimates)

% U_BestEstimates =
%
%    1.0e-14 *
%
%    0.159999999999368   0.159999999999082   0.159999999688647   0.000116400000000   0.000116399997098   0.000116399996269
%    0.159999999999082   0.160046483675102   0.160017224365120   0.000116399999999   0.000156928897815   0.000137367873159
%    0.159999999688647   0.160017224365120   0.160038988605713   0.000116399999773   0.000137232202826   0.000149667262589
%    0.000116400000000   0.000116399999999   0.000116399999773   0.000100000000000   0.000099999999998   0.000099999999997
%    0.000116399997098   0.000156928897815   0.000137232202826   0.000099999999998   0.000149348606637   0.000126375525402
%    0.000116399996269   0.000137367873159   0.000149667262589   0.000099999999997   0.000126375525402   0.000139639048395
```

```
    1.0e-14 *

   Columns 1 through 3

    0.159999999999368   0.159999999999082   0.159999999688647
    0.159999999999082   0.160046483675102   0.160017224365120
    0.159999999688647   0.160017224365120   0.160038988605713
    0.000116400000000   0.000116399999999   0.000116399999773
    0.000116399997098   0.000156928897815   0.000137232202826
    0.000116399996269   0.000137367873159   0.000149667262589

   Columns 4 through 6

    0.000116400000000   0.000116399997098   0.000116399996269
    0.000116399999999   0.000156928897815   0.000137367873159
    0.000116399999773   0.000137232202826   0.000149667262589
    0.000100000000000   0.000099999999998   0.000099999999997
    0.000099999999998   0.000149348606637   0.000126375525402
    0.000099999999997   0.000126375525402   0.000139639048395
```

Standard uncertainties of the OEFPIL Best Estimates

Estimated parameters of interest (M_R, M_1, M_2, V_R, V_1, V_2):

```
u_BestEstimates = sqrt(diag(U_BestEstimates));

disp(table(u_BestEstimates))

% u_BestEstimates =
%
%    1.0e-07 *
%
%    0.399999999999210
%    0.400058100374311
%    0.400048732788535
%    0.010000000000000
%    0.012220826757496
%    0.011816896732847
```

```
    u_BestEstimates
  ____________________

    3.9999999999921e-08
   4.00058100374311e-08
   4.00048732788535e-08
                  1e-09
   1.22208267574956e-09
   1.18168967328469e-09
```

Correlation matrix of the OEFPIL Best Estimates

Estimated parameters of interest (M_R, M_1, M_2, V_R, V_1, V_2):

```
Corr_BestEstimates = U_BestEstimates ./ (u_BestEstimates * u_BestEstimates');

disp(Corr_BestEstimates)

% Corr_BestEstimates =
%
%    1.000000000000000   0.999854770155231   0.999878180926137   0.029100000000065   0.023811809014198   0.024625753888923
%    0.999854770155231   1.000000000000000   0.999840594256046   0.029095773811460   0.032098095769199   0.029057528035923
%    0.999878180926137   0.999840594256046   1.000000000000000   0.029096455064893   0.028070008747368   0.031659968335108
%    0.029099999999991   0.029095773811460   0.029096455064893   1.000000000000000   0.818275244238725   0.846245865205206
%    0.023811809014198   0.032098095769199   0.028070008747368   0.818275244238725   1.000000000000000   0.875102543877616
%    0.024625753888923   0.029057528035923   0.031659968335108   0.846245865205206   0.875102543877616   1.000000000000000
```

```
  Columns 1 through 3

   1.000000000000000   0.999854770155231   0.999878180926137
   0.999854770155231   1.000000000000000   0.999840594256046
   0.999878180926137   0.999840594256046   1.000000000000000
   0.029099999999991   0.029095773811460   0.029096455064893
   0.023811809014198   0.032098095769199   0.028070008747368
   0.024625753888923   0.029057528035923   0.031659968335108

  Columns 4 through 6

   0.029100000000065   0.023811809014198   0.024625753888923
   0.029095773811460   0.032098095769199   0.029057528035923
   0.029096455064893   0.028070008747368   0.031659968335108
   1.000000000000000   0.818275244238725   0.846245865205206
   0.818275244238725   1.000000000000000   0.875102543877616
   0.846245865205206   0.875102543877616   1.000000000000000
```

TABLE OEFPIL Best Estimates of Mass

Estimated parameters of interest (M_R, M_1, M_2):

```
alpha = 0.05;
coverageFactor = norminv(1-alpha/2);

TABLE_mass= table;
TABLE_mass.Properties.Description = 'Estimated Mass of Weights by OEFPIL';
TABLE_mass.ESTIMATE = mass;
TABLE_mass.STD      = u_mass;
TABLE_mass.FACTOR   = coverageFactor*ones(size(mass));
TABLE_mass.LOWER    = mass - coverageFactor*u_mass;
TABLE_mass.UPPER    = mass + coverageFactor*u_mass;
TABLE_mass.PVAL     = 2*normcdf(-abs((1-mass)./u_mass));
TABLE_mass.Properties.RowNames = {'M_R (51699)' 'M_1 (15N)' 'M_2 (15963)'};
TABLE_mass.Properties.VariableNames = {'ESTIMATE [kg]'  'STD [kg]'  'FACTOR'  'LOWER BOUND [kg]'  'UPPER BOUND [kg]'  'PVAL [H0: Mass = 1 kg]'};

disp(TABLE_mass)

%                  ESTIMATE [kg]           STD [kg]              FACTOR         LOWER BOUND [kg]     UPPER BOUND [kg]     PVAL [H0: Mass = 1 kg]
%                _________________    ____________________    _____________    ________________    _________________    ______________________
%
% M_R (51699)     1.00000064100001    3.99999999999833e-08    1.95996398454005     1.00000056260145     1.00000071939857     8.54906569271744e-58
% M_1 (15N)      0.999999887200826     4.0005810037432e-08    1.95996398454005    0.999999808790879    0.999999965610773      0.00480880712865863
% M_2 (15963)     1.00000075427855     4.0004873278121e-08    1.95996398454005     1.00000067587044     1.00000083268666     2.69029455690485e-79
```

```
                  ESTIMATE [kg]           STD [kg]              FACTOR         LOWER BOUND [kg]     UPPER BOUND [kg]     PVAL [H0: Mass = 1 kg]
                _________________    ____________________    _____________    ________________    _________________    ______________________

    M_R (51699)     1.00000064100001     3.9999999999921e-08    1.95996398454005     1.00000056260145     1.00000071939857     8.54906568928706e-58
    M_1 (15N)      0.999999887200826    4.00058100374311e-08    1.95996398454005    0.999999808790879    0.999999965610773      0.00480880712865768
    M_2 (15963)     1.00000075427855    4.00048732788535e-08    1.95996398454005     1.00000067587044     1.00000083268666     2.69029457446623e-79
```

TABLE OEFPIL Best Estimates of Volume

Estimated parameters of interest (V_R, V_1, V_2):

```
TABLE_volume= table;
TABLE_volume.Properties.Description = 'Estimated Volume of Weights by OEFPIL';
TABLE_volume.ESTIMATE = volume;
TABLE_volume.STD      = u_volume;
TABLE_volume.FACTOR   = coverageFactor*ones(size(volume));
TABLE_volume.LOWER    = volume - coverageFactor*u_volume;
TABLE_volume.UPPER    = volume + coverageFactor*u_volume;
TABLE_volume.PVAL     = 2*normcdf(-abs((0.000125-volume)./u_volume));
TABLE_volume.Properties.RowNames = {'V_R (51699)' 'V_1 (15N)' 'V_2 (15963)'};
TABLE_volume.Properties.VariableNames = {'ESTIMATE [m^3]'  'STD [m^3]'  'FACTOR'  'LOWER BOUND [m^3]'  'UPPER BOUND [m^3]'  'PVAL [H0: Volume = 125 cm^3]'};

disp(TABLE_volume)
```

```
%                 ESTIMATE [m^3]          STD [m^3]            FACTOR          LOWER BOUND [m^3]       UPPER BOUND [m^3]      PVAL [H0: Volume = 125 cm^
%                 ____________________    ____________________    ____________    ____________________    ____________________    ____________________________
%
% V_R (51699)    0.000125440000000005                 1e-09    1.95996398454005     0.00012543804003602    0.000125441959963989                    0
% V_1 (15N)      0.000124891615012911    1.22208267574956e-09    1.95996398454005     0.00012488921977488    0.000124894010250941                    0
% V_2 (15963)    0.000125642839434627    1.18168967328469e-09    1.95996398454005    0.000125640523365427    0.000125645155503828                    0
```

	ESTIMATE [m^3]	STD [m^3]	FACTOR	LOWER BOUND [m^3]	UPPER BOUND [m^3]	PVAL [H0: Volume = 125 c
V_R (51699)	0.000125440000000005	1e-09	1.95996398454005	0.00012543804003602	0.000125441959963989	0
V_1 (15N)	0.000124891615012911	1.22208267574956e-09	1.95996398454005	0.00012488921977488	0.000124894010250941	0
V_2 (15963)	0.000125642839434627	1.18168967328469e-09	1.95996398454005	0.000125640523365427	0.000125645155503828	0

TABLE / Estimated parameters / OEFPIL

```
p = length(BestEstimates);
TABLE_OEFPIL = table;
TABLE_OEFPIL.Properties.Description = 'Estimated Mass and Volume of the Weights by OEFPIL';
TABLE_OEFPIL.ESTIMATE = BestEstimates;
TABLE_OEFPIL.STD      = u_BestEstimates;
TABLE_OEFPIL.FACTOR   = coverageFactor*ones(size(BestEstimates));
TABLE_OEFPIL.LOWER    = BestEstimates - coverageFactor*u_BestEstimates;
TABLE_OEFPIL.UPPER    = BestEstimates + coverageFactor*u_BestEstimates;
TABLE_OEFPIL.Properties.RowNames = {'M_R (51699)' 'M_1 (15N)' 'M_2 (15963)' 'V_R (51699)' 'V_1 (15N)' 'V_2 (15963)'};
TABLE_OEFPIL.Properties.VariableNames = {'ESTIMATE [kg | m^3]'  'STD [kg | m^3]'  'FACTOR'  'LOWER BOUND [kg | m^3]'  'UPPER BOUND [kg | m^3]' };

disp(TABLE_OEFPIL)
```

```
%                 ESTIMATE [kg | m^3]       STD [kg | m^3]           FACTOR          LOWER BOUND [kg | m^3]    UPPER BOUND [kg | m^3]
%                 ____________________    ____________________    ____________    ____________________    ____________________
%
% M_R (51699)        1.00000064100001     3.9999999999921e-08    1.95996398454005        1.00000056260145        1.00000071939857
% M_1 (15N)         0.999999887200826    4.00058100374311e-08    1.95996398454005       0.999999808790879       0.999999965610773
% M_2 (15963)        1.00000075427855    4.00048732788535e-08    1.95996398454005        1.00000067587044        1.00000083268666
% V_R (51699)    0.000125440000000005                   1e-09    1.95996398454005     0.00012543804003602     0.000125441959963989
% V_1 (15N)      0.000124891615012911    1.22208267574956e-09    1.95996398454005     0.00012488921977488     0.000124894010250941
% V_2 (15963)    0.000125642839434627    1.18168967328469e-09    1.95996398454005    0.000125640523365427     0.000125645155503828
```

	ESTIMATE [kg | m^3]	STD [kg | m^3]	FACTOR	LOWER BOUND [kg | m^3]	UPPER BOUND [kg | m^3]
M_R (51699)	1.00000064100001	3.9999999999921e-08	1.95996398454005	1.00000056260145	1.00000071939857
M_1 (15N)	0.999999887200826	4.00058100374311e-08	1.95996398454005	0.999999808790879	0.999999965610773
M_2 (15963)	1.00000075427855	4.00048732788535e-08	1.95996398454005	1.00000067587044	1.00000083268666
V_R (51699)	0.000125440000000005	1e-09	1.95996398454005	0.00012543804003602	0.000125441959963989
V_1 (15N)	0.000124891615012911	1.22208267574956e-09	1.95996398454005	0.00012488921977488	0.000124894010250941
V_2 (15963)	0.000125642839434627	1.18168967328469e-09	1.95996398454005	0.000125640523365427	0.000125645155503828

Save the results

```
save RESULTS_MassCalibration.mat
```

Published with MATLAB® R2023b