Contents

SIMULATIONS / Monte Carlo Method

Viktor Witkovsky (witkovsky@savba.sk) Ver.: "17-Feb-2024 10:46:35'

```
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
```

Set the uncertainty matrix of the input variables

```
U       = U_full2;
%SqrtmU  = sqrtm(U);
CholU   = chol(U);
Sqrtm   = CholU';
```

Monte Carlo simulations

```
% Set the seed of random number generator for reproducibility
rng(0)

% Set N - number of Monte Carlo simulations
N = 1000000;
mu = zeros(23,N);
MassVolume = zeros(6,N);

for i = 1:N
 MassVolume(:,i) = funMonteCarloMethod(data,Sqrtm);
end
```

Mean / BestEstimates of (MR M1 M2 VR V1 V2)

```
BestEstimate = mean(MassVolume,2);
NominalMass = 1;
BestEstimate(1:3) = BestEstimate(1:3)+ NominalMass;

disp(BestEstimate)

% BestEstimates =
%
%    1.000000641030607
%    0.999999876385788
%    1.000000758794277
%    0.000125440001730
%    0.000124870801688
%    0.000125637594583
```

```
   1.000000641030607
   0.999999876385788
   1.000000758794277
   0.000125440001730
   0.000124870801688
   0.000125637594583
```

Uncertainty (covariance) matrix of BestEstimates

```
U_BestEstimate = cov(MassVolume');

disp(U_BestEstimate)

% U_BestEstimates =
%
%    1.0e-14 *
%
%    0.160006730895750   0.160010039260497   0.160009778727043   0.000116622371968   0.000122279219908   0.000119318196852
%    0.160010039260497   0.160062615213230   0.160029921595559   0.000116577503689   0.000172651826353   0.000143044586418
%    0.160009778727043   0.160029921595559   0.160044688496514   0.000116680381491   0.000146119609758   0.000154169574439
%    0.000116622371968   0.000116577503689   0.000116680381491   0.000099918194857   0.000099886438516   0.000099996109220
%    0.000122279219908   0.000172651826353   0.000146119609758   0.000099886438516   0.000164072695936   0.000130896691300
%    0.000119318196852   0.000143044586418   0.000154169574439   0.000099996109220   0.000130896691300   0.000141876710344
```

```
   1.0e-14 *

  Columns 1 through 3

   0.160006730895750   0.160010039260497   0.160009778727043
   0.160010039260497   0.160062615213230   0.160029921595559
   0.160009778727043   0.160029921595559   0.160044688496514
   0.000116622371968   0.000116577503689   0.000116680381491
   0.000122279219908   0.000172651826353   0.000146119609758
   0.000119318196852   0.000143044586418   0.000154169574439

  Columns 4 through 6

   0.000116622371968   0.000122279219908   0.000119318196852
   0.000116577503689   0.000172651826353   0.000143044586418
   0.000116680381491   0.000146119609758   0.000154169574439
   0.000099918194857   0.000099886438516   0.000099996109220
   0.000099886438516   0.000164072695936   0.000130896691300
   0.000099996109220   0.000130896691300   0.000141876710344
```

Standard uncertainties of BestEstimates

```
u_BestEstimate = sqrt(diag(U_BestEstimate));

disp(u_BestEstimate)

% u_BestEstimates =
%
%    1.0e-07 *
%
%    0.400008413531204
%    0.400078261360487
%    0.400055856720676
%    0.009995908905979
%    0.012809086459850
%    0.011911201045409
```

```
   1.0e-07 *

   0.400008413531204
   0.400078261360487
   0.400055856720676
   0.009995908905979
   0.012809086459850
   0.011911201045409
```

Correlation Matrix of BestEstimates

```
Corr_BestEstimate = U_BestEstimate ./ (u_BestEstimate * u_BestEstimate');

disp(Corr_BestEstimate)

% Corr_BestEstimates =
%
%    1.000000000000000   0.999846087385071   0.999900454471953   0.029166912209070   0.023865216373058   0.025042748991460
%    0.999846087385071   1.000000000000000   0.999851737163817   0.029150600626223   0.033690547250360   0.030017251028595
%    0.999900454471953   0.999851737163817   1.000000000000000   0.029177959554372   0.028514759553513   0.032353590611810
%    0.029166912209070   0.029150600626223   0.029177959554372   1.000000000000000   0.780128388043332   0.839856828296582
%    0.023865216373058   0.033690547250360   0.028514759553513   0.780128388043332   1.000000000000000   0.857936128622859
%    0.025042748991460   0.030017251028595   0.032353590611810   0.839856828296582   0.857936128622859   1.000000000000000
```

```
  Columns 1 through 3

   1.000000000000000   0.999846087385071   0.999900454471953
   0.999846087385071   1.000000000000000   0.999851737163817
   0.999900454471953   0.999851737163817   1.000000000000000
   0.029166912209070   0.029150600626223   0.029177959554372
   0.023865216373058   0.033690547250360   0.028514759553513
   0.025042748991460   0.030017251028595   0.032353590611810

  Columns 4 through 6

   0.029166912209070   0.023865216373058   0.025042748991460
   0.029150600626223   0.033690547250360   0.030017251028595
   0.029177959554372   0.028514759553513   0.032353590611810
   1.000000000000000   0.780128388043332   0.839856828296582
   0.780128388043332   1.000000000000000   0.857936128622859
   0.839856828296582   0.857936128622859   1.000000000000000
```

TABLE / Estimated parameters (MR M1 M2 VR V1 V2) / MCM - Monte Carlo Method

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

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

disp(TABLE_MCM)

%                ESTIMATE [kg | m^3]     STD [kg | m^3]          FACTOR        LOWER BOUND [kg | m^3]    UPPER BOUND [kg | m^3]
%                ____________________    ____________________    ________________    ______________________    ______________________
%
% M_R (51699)        1.00000064103061    4.00008413531204e-08    1.95996398454005        1.0000005626304         1.00000071943082
% M_1 (15N)         0.999999876385788    4.00078261360487e-08    1.95996398454005       0.99999979797189        0.999999954799687
% M_2 (15963)        1.00000075879428    4.00055856720675e-08    1.95996398454005       1.00000068038477         1.00000083720378
% V_R (51699)    0.000125440001729943    9.99590890597917e-10    1.95996398454005    0.000125438042567798    0.000125441960892087
% V_1 (15N)        0.00012487080168844    1.28090864598496e-09    1.95996398454005    0.000124868291153627    0.000124873312223254
% V_2 (15963)    0.000125637594583042    1.19112010454089e-09    1.95996398454005    0.000125635260030536    0.000125639929135548
```

```
                   ESTIMATE [kg | m^3]     STD [kg | m^3]          FACTOR        LOWER BOUND [kg | m^3]    UPPER BOUND [kg | m^3]
                   ____________________    ____________________    ________________    ______________________    ______________________

    M_R (51699)        1.00000064103061    4.00008413531204e-08    1.95996398454005        1.0000005626304         1.00000071943082
    M_1 (15N)         0.999999876385788    4.00078261360487e-08    1.95996398454005       0.99999979797189        0.999999954799687
    M_2 (15963)        1.00000075879428    4.00055856720675e-08    1.95996398454005       1.00000068038477         1.00000083720378
    V_R (51699)    0.000125440001729943    9.99590890597917e-10    1.95996398454005    0.000125438042567798    0.000125441960892087
```

```
V_1 (15N)       0.00012487080168844    1.28090864598496e-09    1.95996398454005     0.000124868291153627     0.000124873312223254
V_2 (15963)    0.000125637594583042    1.19112010454089e-09    1.95996398454005     0.000125635260030536     0.000125639929135548
```

Published with MATLAB® R2023b