Annexes

Annex A. Spreadsheet file with the results of fire simulation by means of CFAST computer code

Table A.1. Samples of fire simulation by means of CFAST computer code

Time, s	No of sample			
	1		2	
	Temperature, °C	Oxygen content, %	Temperature, °C	Oxygen content, %
0	8.170E+00	2.061E+01	8.913E+00	2.061E+01
100	2.043E+01	2.049E+01	3.575E+01	2.039E+01
200	6.342E+01	2.010E+01	1.259E+02	1.964E+01
300	1.532E+02	1.919E+01	2.964E+02	1.788E+01
400	2.496E+02	1.676E+01	4.547E+02	1.444E+01
500	3.933E+02	1.048E+01	5.920E+02	9.049E+00
600	2.921E+02	1.426E+01	3.128E+02	8.489E+00
700	3.311E+02	1.673E+01	2.424E+02	8.023E+00
800	3.962E+02	1.580E+01	6.751E+02	1.060E+01
900	4.693E+02	1.464E+01	8.044E+02	9.087E+00

1000	5.467E+02	1.327E+01	8.167E+02	8.876E+00
1100	6.248E+02	1.170E+01	8.990E+02	6.741E+00
1200	6.497E+02	1.124E+01	9.444E+02	5.485E+00
1300	6.555E+02	1.123E+01	9.505E+02	5.475E+00
1400	6.589E+02	1.123E+01	9.537E+02	5.470E+00
1500	6.611E+02	1.123E+01	9.556E+02	5.467E+00
1600	6.625E+02	1.123E+01	9.568E+02	5.465E+00
1700	6.634E+02	1.122E+01	9.575E+02	5.464E+00
1800	6.390E+02	1.170E+01	9.580E+02	5.463E+00
1900	5.097E+02	1.445E+01	9.024E+02	7.040E+00
2000	3.883E+02	1.638E+01	8.425E+02	9.055E+00
2100	3.171E+02	1.729E+01	7.426E+02	1.144E+01
2200	2.543E+02	1.811E+01	6.129E+02	1.379E+01
2300	2.098E+02	1.848E+01	4.706E+02	1.599E+01
2400	1.774E+02	1.852E+01	3.621E+02	1.735E+01
2500	1.535E+02	1.852E+01	2.907E+02	1.811E+01
2600	1.341E+02	1.852E+01	2.415E+02	1.839E+01
2700	1.177E+02	1.852E+01	2.058E+02	1.839E+01
2800	1.036E+02	1.852E+01	1.788E+02	1.839E+01
2900	9.143E+01	1.852E+01	1.565E+02	1.839E+01
3000	8.082E+01	1.852E+01	1.377E+02	1.839E+01
3100	7.155E+01	1.852E+01	1.215E+02	1.839E+01
3200	6.342E+01	1.852E+01	1.075E+02	1.839E+01
3300	5.627E+01	1.852E+01	9.529E+01	1.839E+01
3400	4.996E+01	1.852E+01	8.468E+01	1.839E+01
3500	4.439E+01	1.852E+01	7.541E+01	1.839E+01
3600	3.945E+01	1.852E+01	6.727E+01	1.839E+01
Time, s	No of sample			
	3		4	
	Temperature, °C	Oxygen content, %	Temperature, °C	Oxygen content, %
0	1.477E+01	2.055E+01	1.070E+01	2.059E+01
100	3.599E+01	2.036E+01	2.022E+01	2.049E+01
200	9.400E+01	1.983E+01	4.132E+01	2.027E+01
300	2.106E+02	1.862E+01	8.917E+01	1.979E+01
400	3.484E+02	1.623E+01	1.597E+02	1.897E+01
500	4.535E+02	1.247E+01	2.469E+02	1.755E+01

600	3.930E+02	8.773E+00	3.181E+02	1.548E+01
700	4.239E+02	1.025E+01	2.079E+02	1.775E+01
800	5.401E+02	1.367E+01	2.376E+02	1.788E+01
900	6.244E+02	1.217E+01	2.802E+02	1.734E+01
1000	6.871E+02	1.124E+01	3.272E+02	1.671E+01
1100	7.716E+02	9.095E+00	3.713E+02	1.611E+01
1200	7.743E+02	9.003E+00	4.174E+02	1.545E+01
1300	7.810E+02	8.688E+00	4.678E+02	1.467E+01
1400	8.452E+02	6.815E+00	5.192E+02	1.382E+01
1500	8.842E+02	5.607E+00	5.728E+02	1.285E+01
1600	8.882E+02	5.599E+00	6.258E+02	1.180E+01
1700	8.901E+02	5.596E+00	6.779E+02	1.066E+01
1800	8.912E+02	5.594E+00	6.929E+02	1.037E+01
1900	8.919E+02	5.593E+00	6.966E+02	1.037E+01
2000	8.923E+02	5.592E+00	6.988E+02	1.036E+01
2100	8.925E+02	5.592E+00	7.002E+02	1.036E+01
2200	8.927E+02	5.592E+00	7.010E+02	1.036E+01
2300	8.916E+02	5.603E+00	7.016E+02	1.036E+01
2400	7.870E+02	8.782E+00	6.892E+02	1.058E+01
2500	7.916E+02	9.089E+00	5.696E+02	1.333E+01
2600	7.000E+02	1.106E+01	4.429E+02	1.555E+01
2700	5.517E+02	1.421E+01	3.512E+02	1.683E+01
2800	4.192E+02	1.627E+01	2.853E+02	1.768E+01
2900	3.270E+02	1.746E+01	2.327E+02	1.828E+01
3000	2.642E+02	1.813E+01	1.939E+02	1.850E+01
3100	2.197E+02	1.838E+01	1.654E+02	1.850E+01
3200	1.874E+02	1.838E+01	1.433E+02	1.850E+01
3300	1.625E+02	1.838E+01	1.250E+02	1.850E+01
3400	1.420E+02	1.838E+01	1.093E+02	1.850E+01
3500	1.246E+02	1.838E+01	9.585E+01	1.850E+01
3600	1.096E+02	1.838E+01	8.412E+01	1.850E+01
Time, s	No of sample			
	5		6	
	Temperature, °C	Oxygen content, %	Temperature, °C	Oxygen content, %
0	1.490E+01	2.055E+01	8.150E+00	2.061E+01
100	2.631E+01	2.044E+01	1.517E+01	2.053E+01

200	6.403E+01	2.008E+01	3.637E+01	2.032E+01
300	1.433E+02	1.927E+01	8.147E+01	1.987E+01
400	2.578E+02	1.768E+01	1.510E+02	1.908E+01
500	3.524E+02	1.510E+01	2.377E+02	1.775E+01
600	4.344E+02	1.138E+01	3.111E+02	1.575E+01
700	3.054E+02	1.691E+01	2.684E+02	1.353E+01
800	3.632E+02	1.633E+01	2.434E+02	1.780E+01
900	4.264E+02	1.537E+01	2.850E+02	1.731E+01
1000	4.956E+02	1.424E+01	3.302E+02	1.670E+01
1100	5.672E+02	1.292E+01	3.722E+02	1.612E+01
1200	6.395E+02	1.143E+01	4.174E+02	1.546E+01
1300	6.603E+02	1.104E+01	4.306E+02	1.529E+01
1400	6.654E+02	1.103E+01	4.342E+02	1.529E+01
1500	6.684E+02	1.103E+01	4.366E+02	1.529E+01
1600	6.703E+02	1.103E+01	4.382E+02	1.529E+01
1700	6.715E+02	1.103E+01	4.394E+02	1.529E+01
1800	6.723E+02	1.103E+01	4.402E+02	1.529E+01
1900	6.728E+02	1.103E+01	4.408E+02	1.529E+01
2000	6.731E+02	1.103E+01	4.413E+02	1.529E+01
2100	6.288E+02	1.199E+01	4.001E+02	1.591E+01
2200	5.029E+02	1.459E+01	3.193E+02	1.711E+01
2300	3.870E+02	1.640E+01	2.665E+02	1.778E+01
2400	3.182E+02	1.728E+01	2.232E+02	1.834E+01
2500	2.566E+02	1.808E+01	1.862E+02	1.876E+01
2600	2.121E+02	1.848E+01	1.567E+02	1.899E+01
2700	1.789E+02	1.855E+01	1.331E+02	1.904E+01
2800	1.545E+02	1.855E+01	1.153E+02	1.904E+01
2900	1.348E+02	1.855E+01	1.008E+02	1.904E+01
3000	1.183E+02	1.855E+01	8.859E+01	1.904E+01
3100	1.041E+02	1.855E+01	7.803E+01	1.904E+01
3200	9.176E+01	1.855E+01	6.883E+01	1.904E+01
3300	8.108E+01	1.855E+01	6.073E+01	1.904E+01
3400	7.172E+01	1.855E+01	5.355E+01	1.904E+01
3500	6.352E+01	1.855E+01	4.720E+01	1.904E+01
3600	5.630E+01	1.855E+01	4.157E+01	1.904E+01
Time, s	No of sample			
	7		8	

	Temperature, °C	Oxygen content, %	Temperature, °C	Oxygen content, %
0	1.620E+01	2.053E+01	1.020E+01	2.059E+01
100	2.855E+01	2.041E+01	2.536E+01	2.045E+01
200	6.568E+01	2.005E+01	6.822E+01	2.005E+01
300	1.419E+02	1.927E+01	1.574E+02	1.913E+01
400	2.523E+02	1.775E+01	2.811E+02	1.731E+01
500	3.469E+02	1.525E+01	3.779E+02	1.441E+01
600	4.271E+02	1.165E+01	4.659E+02	1.026E+01
700	3.172E+02	1.606E+01	2.958E+02	8.660E+00
800	3.638E+02	1.635E+01	4.055E+02	1.533E+01
900	4.224E+02	1.546E+01	4.755E+02	1.461E+01
1000	4.883E+02	1.437E+01	5.499E+02	1.330E+01
1100	5.577E+02	1.312E+01	6.270E+02	1.175E+01
1200	6.275E+02	1.171E+01	7.033E+02	1.001E+01
1300	6.831E+02	1.042E+01	7.429E+02	8.961E+00
1400	6.910E+02	1.039E+01	7.995E+02	7.359E+00
1500	6.947E+02	1.039E+01	8.085E+02	7.173E+00
1600	6.970E+02	1.038E+01	8.117E+02	7.168E+00
1700	6.985E+02	1.038E+01	8.135E+02	7.165E+00
1800	6.994E+02	1.038E+01	8.146E+02	7.163E+00
1900	7.000E+02	1.038E+01	8.153E+02	7.162E+00
2000	7.004E+02	1.038E+01	8.157E+02	7.162E+00
2100	6.325E+02	1.198E+01	8.120E+02	7.220E+00
2200	5.115E+02	1.447E+01	7.322E+02	9.698E+00
2300	3.942E+02	1.633E+01	5.974E+02	1.297E+01
2400	3.239E+02	1.723E+01	4.818E+02	1.499E+01
2500	2.626E+02	1.803E+01	3.726E+02	1.664E+01
2600	2.175E+02	1.848E+01	2.933E+02	1.771E+01
2700	1.832E+02	1.859E+01	2.397E+02	1.826E+01
2800	1.580E+02	1.859E+01	2.005E+02	1.840E+01
2900	1.379E+02	1.859E+01	1.721E+02	1.840E+01
3000	1.211E+02	1.859E+01	1.498E+02	1.840E+01
3100	1.066E+02	1.859E+01	1.311E+02	1.840E+01
3200	9.410E+01	1.859E+01	1.152E+02	1.840E+01
3300	8.328E+01	1.859E+01	1.015E+02	1.840E+01
3400	7.381E+01	1.859E+01	8.951E+01	1.840E+01

3500	6.550E+01	1.859E+01	7.909E+01	1.840E+01
3600	5.820E+01	1.859E+01	7.001E+01	1.840E+01
Time, s	No of sample			
	9		10	
	Temperature, °C	Oxygen content, %	Temperature, °C	Oxygen content, %
0	6.087E+00	2.063E+01	6.048E+00	2.063E+01
100	2.681E+01	2.044E+01	3.209E+01	2.044E+01
200	8.613E+01	1.990E+01	9.927E+01	1.994E+01
300	2.047E+02	1.869E+01	2.376E+02	1.879E+01
400	3.460E+02	1.627E+01	4.049E+02	1.658E+01
500	4.547E+02	1.244E+01	5.329E+02	1.307E+01
600	3.901E+02	8.781E+00	5.037E+02	1.031E+01
700	4.558E+02	1.442E+01	5.419E+02	1.532E+01
800	5.378E+02	1.374E+01	6.418E+02	1.397E+01
900	6.117E+02	1.250E+01	7.531E+02	1.219E+01
1000	7.024E+02	1.058E+01	8.501E+02	1.055E+01
1100	7.631E+02	9.072E+00	9.183E+02	9.800E+00
1200	7.663E+02	8.899E+00	9.412E+02	9.035E+00
1300	8.234E+02	7.258E+00	9.388E+02	8.929E+00
1400	8.755E+02	5.616E+00	9.654E+02	8.213E+00
1500	8.808E+02	5.596E+00	1.041E+03	6.337E+00
1600	8.831E+02	5.592E+00	1.073E+03	5.517E+00
1700	8.845E+02	5.589E+00	1.077E+03	5.512E+00
1800	8.853E+02	5.588E+00	1.079E+03	5.510E+00
1900	8.858E+02	5.587E+00	1.080E+03	5.508E+00
2000	8.861E+02	5.586E+00	1.081E+03	5.507E+00
2100	8.726E+02	5.905E+00	1.081E+03	5.507E+00
2200	7.816E+02	8.997E+00	1.081E+03	5.506E+00
2300	7.160E+02	1.077E+01	9.671E+02	8.343E+00
2400	5.882E+02	1.336E+01	9.587E+02	9.070E+00
2500	4.470E+02	1.583E+01	8.278E+02	1.145E+01
2600	3.404E+02	1.727E+01	6.251E+02	1.485E+01
2700	2.717E+02	1.805E+01	4.578E+02	1.686E+01
2800	2.251E+02	1.830E+01	3.567E+02	1.785E+01
2900	1.919E+02	1.830E+01	2.921E+02	1.816E+01
3000	1.664E+02	1.830E+01	2.474E+02	1.816E+01

3100	1.455E+02	1.830E+01	2.139E+02	1.816E+01
3200	1.277E+02	1.830E+01	1.865E+02	1.816E+01
3300	1.125E+02	1.830E+01	1.634E+02	1.816E+01
3400	9.923E+01	1.830E+01	1.435E+02	1.816E+01
3500	8.773E+01	1.830E+01	1.263E+02	1.816E+01
3600	7.767E+01	1.830E+01	1.114E+02	1.816E+01
Time, s	No of sample			
	11		12	
	Temperature, °C	Oxygen content, %	Temperature, °C	Oxygen content, %
0	1.590E+01	2.053E+01	8.818E+00	2.061E+01
100	3.012E+01	2.040E+01	3.256E+01	2.041E+01
200	7.358E+01	1.999E+01	9.753E+01	1.984E+01
300	1.638E+02	1.906E+01	2.269E+02	1.856E+01
400	2.855E+02	1.720E+01	3.765E+02	1.604E+01
500	3.810E+02	1.421E+01	4.918E+02	1.208E+01
600	4.692E+02	9.956E+00	3.918E+02	8.738E+00
700	2.602E+02	8.549E+00	2.679E+02	8.357E+00
800	4.048E+02	1.567E+01	5.851E+02	1.302E+01
900	4.766E+02	1.459E+01	6.903E+02	1.133E+01
1000	5.531E+02	1.320E+01	7.624E+02	1.018E+01
1100	6.300E+02	1.163E+01	8.116E+02	9.064E+00
1200	7.053E+02	9.876E+00	8.117E+02	8.969E+00
1300	7.772E+02	7.958E+00	8.192E+02	8.623E+00
1400	7.880E+02	7.783E+00	8.871E+02	6.722E+00
1500	7.920E+02	7.778E+00	9.247E+02	5.621E+00
1600	7.944E+02	7.774E+00	9.285E+02	5.614E+00
1700	7.958E+02	7.772E+00	9.304E+02	5.611E+00
1800	7.967E+02	7.771E+00	9.315E+02	5.609E+00
1900	7.973E+02	7.770E+00	9.321E+02	5.608E+00
2000	7.811E+02	8.158E+00	9.325E+02	5.607E+00
2100	6.753E+02	1.115E+01	9.328E+02	5.607E+00
2200	5.581E+02	1.371E+01	9.329E+02	5.607E+00
2300	4.388E+02	1.572E+01	8.591E+02	7.737E+00
2400	3.580E+02	1.682E+01	8.255E+02	9.003E+00
2500	2.968E+02	1.761E+01	8.055E+02	9.565E+00
2600	2.433E+02	1.826E+01	6.788E+02	1.236E+01

2700	2.039E+02	1.859E+01	5.379E+02	1.486E+01
2800	1.736E+02	1.864E+01	4.071E+02	1.668E+01
2900	1.506E+02	1.864E+01	3.233E+02	1.766E+01
3000	1.318E+02	1.864E+01	2.642E+02	1.821E+01
3100	1.160E+02	1.864E+01	2.206E+02	1.841E+01
3200	1.023E+02	1.864E+01	1.887E+02	1.841E+01
3300	9.050E+01	1.864E+01	1.638E+02	1.841E+01
3400	8.023E+01	1.864E+01	1.431E+02	1.841E+01
3500	7.124E+01	1.864E+01	1.255E+02	1.841E+01
3600	6.335E+01	1.864E+01	1.103E+02	1.841E+01
Time, s	No of sample			
	13		14	
	Temperature, °C	Oxygen content, %	Temperature, °C	Oxygen content, %
0	1.729E+01	2.056E+01	1.120E+01	2.058E+01
100	5.167E+01	2.032E+01	2.321E+01	2.047E+01
200	1.572E+02	1.955E+01	6.115E+01	2.011E+01
300	3.510E+02	1.779E+01	1.406E+02	1.931E+01
400	5.317E+02	1.435E+01	2.560E+02	1.772E+01
500	6.850E+02	8.976E+00	3.524E+02	1.515E+01
600	3.644E+02	8.461E+00	4.352E+02	1.145E+01
700	2.836E+02	7.993E+00	3.330E+02	8.748E+00
800	5.077E+02	7.515E+00	3.700E+02	1.562E+01
900	9.414E+02	9.112E+00	4.312E+02	1.533E+01
1000	9.555E+02	9.007E+00	5.005E+02	1.417E+01
1100	9.573E+02	8.857E+00	5.727E+02	1.283E+01
1200	1.022E+03	7.382E+00	6.447E+02	1.132E+01
1300	1.098E+03	5.480E+00	6.917E+02	1.023E+01
1400	1.107E+03	5.447E+00	6.984E+02	1.022E+01
1500	1.110E+03	5.442E+00	7.018E+02	1.022E+01
1600	1.112E+03	5.439E+00	7.039E+02	1.022E+01
1700	1.114E+03	5.437E+00	7.053E+02	1.021E+01
1800	1.114E+03	5.436E+00	7.062E+02	1.021E+01
1900	1.115E+03	5.435E+00	7.067E+02	1.021E+01
2000	1.004E+03	8.105E+00	6.514E+02	1.155E+01
2100	9.862E+02	9.029E+00	5.479E+02	1.380E+01
2200	9.040E+02	1.043E+01	4.420E+02	1.561E+01

2300	6.932E+02	1.420E+01	3.583E+02	1.678E+01
2400	5.027E+02	1.655E+01	3.018E+02	1.748E+01
2500	3.880E+02	1.768E+01	2.490E+02	1.817E+01
2600	3.150E+02	1.815E+01	2.086E+02	1.858E+01
2700	2.646E+02	1.816E+01	1.764E+02	1.876E+01
2800	2.280E+02	1.816E+01	1.516E+02	1.876E+01
2900	1.984E+02	1.816E+01	1.322E+02	1.876E+01
3000	1.737E+02	1.816E+01	1.160E+02	1.876E+01
3100	1.525E+02	1.816E+01	1.021E+02	1.876E+01
3200	1.342E+02	1.816E+01	9.021E+01	1.876E+01
3300	1.183E+02	1.816E+01	7.985E+01	1.876E+01
3400	1.045E+02	1.816E+01	7.075E+01	1.876E+01
3500	9.237E+01	1.816E+01	6.276E+01	1.876E+01
3600	8.178E+01	1.816E+01	5.572E+01	1.876E+01
Time, s	No of sample			
	15		16	
	Temperature, °C	Oxygen content, %	Temperature, °C	Oxygen content, %
0	8.790E+00	2.060E+01	9.070E+00	2.060E+01
100	2.481E+01	2.046E+01	3.249E+01	2.038E+01
200	7.227E+01	2.002E+01	1.130E+02	1.964E+01
300	1.695E+02	1.902E+01	2.675E+02	1.785E+01
400	2.976E+02	1.703E+01	4.097E+02	1.438E+01
500	3.970E+02	1.384E+01	5.253E+02	8.987E+00
600	4.912E+02	9.313E+00	2.772E+02	8.489E+00
700	2.770E+02	8.595E+00	2.144E+02	8.024E+00
800	4.317E+02	1.403E+01	6.184E+02	1.150E+01
900	5.118E+02	1.395E+01	7.312E+02	9.096E+00
1000	5.930E+02	1.242E+01	7.531E+02	8.531E+00
1100	6.733E+02	1.069E+01	8.407E+02	5.926E+00
1200	7.366E+02	9.064E+00	8.659E+02	5.216E+00
1300	7.819E+02	7.837E+00	8.706E+02	5.207E+00
1400	8.055E+02	7.181E+00	8.733E+02	5.202E+00
1500	8.094E+02	7.175E+00	8.749E+02	5.199E+00
1600	8.117E+02	7.172E+00	8.758E+02	5.197E+00
1700	8.130E+02	7.169E+00	8.764E+02	5.196E+00
1800	8.139E+02	7.168E+00	8.768E+02	5.195E+00

1900	8.144E+02	7.167E+00	8.494E+02	6.013E+00
2000	8.147E+02	7.167E+00	7.614E+02	9.026E+00
2100	8.149E+02	7.167E+00	6.836E+02	1.108E+01
2200	7.655E+02	8.562E+00	5.724E+02	1.338E+01
2300	6.627E+02	1.151E+01	4.474E+02	1.556E+01
2400	5.170E+02	1.452E+01	3.450E+02	1.701E+01
2500	4.025E+02	1.626E+01	2.751E+02	1.789E+01
2600	3.111E+02	1.753E+01	2.270E+02	1.831E+01
2700	2.509E+02	1.818E+01	1.911E+02	1.839E+01
2800	2.093E+02	1.832E+01	1.644E+02	1.839E+01
2900	1.797E+02	1.832E+01	1.430E+02	1.839E+01
3000	1.565E+02	1.832E+01	1.250E+02	1.839E+01
3100	1.372E+02	1.832E+01	1.095E+02	1.839E+01
3200	1.208E+02	1.832E+01	9.615E+01	1.839E+01
3300	1.066E+02	1.832E+01	8.450E+01	1.839E+01
3400	9.434E+01	1.832E+01	7.433E+01	1.839E+01
3500	8.365E+01	1.832E+01	6.543E+01	1.839E+01
3600	7.433E+01	1.832E+01	5.762E+01	1.839E+01
Time, s	No of sample			
	17		18	
	Temperature, °C	Oxygen content, %	Temperature, °C	Oxygen content, %
0	6.880E+00	2.062E+01	9.790E+00	2.060E+01
100	1.792E+01	2.051E+01	2.497E+01	2.045E+01
200	4.743E+01	2.023E+01	7.333E+01	2.000E+01
300	1.088E+02	1.961E+01	1.720E+02	1.898E+01
400	2.032E+02	1.847E+01	3.002E+02	1.697E+01
500	2.994E+02	1.656E+01	4.002E+02	1.374E+01
600	3.737E+02	1.378E+01	4.975E+02	9.123E+00
700	3.101E+02	1.381E+01	3.585E+02	9.922E+00
800	3.130E+02	1.701E+01	4.643E+02	1.469E+01
900	3.630E+02	1.632E+01	5.348E+02	1.353E+01
1000	4.152E+02	1.554E+01	5.992E+02	1.241E+01
1100	4.722E+02	1.463E+01	6.805E+02	1.064E+01
1200	5.333E+02	1.358E+01	7.423E+02	9.084E+00
1300	5.952E+02	1.240E+01	7.453E+02	9.033E+00
1400	6.479E+02	1.125E+01	7.487E+02	9.033E+00

1500	6.558E+02	1.121E+01	7.508E+02	9.033E+00
1600	6.595E+02	1.121E+01	7.521E+02	9.033E+00
1700	6.618E+02	1.121E+01	7.529E+02	9.033E+00
1800	6.632E+02	1.121E+01	7.534E+02	9.032E+00
1900	6.642E+02	1.121E+01	7.538E+02	9.032E+00
2000	6.648E+02	1.121E+01	7.540E+02	9.032E+00
2100	6.652E+02	1.121E+01	7.047E+02	1.043E+01
2200	6.654E+02	1.121E+01	6.024E+02	1.254E+01
2300	6.068E+02	1.251E+01	4.658E+02	1.527E+01
2400	5.121E+02	1.440E+01	3.535E+02	1.692E+01
2500	4.188E+02	1.591E+01	2.800E+02	1.785E+01
2600	3.400E+02	1.701E+01	2.298E+02	1.831E+01
2700	2.872E+02	1.766E+01	1.929E+02	1.840E+01
2800	2.386E+02	1.828E+01	1.659E+02	1.840E+01
2900	2.010E+02	1.866E+01	1.444E+02	1.840E+01
3000	1.705E+02	1.885E+01	1.264E+02	1.840E+01
3100	1.466E+02	1.885E+01	1.110E+02	1.840E+01
3200	1.279E+02	1.885E+01	9.762E+01	1.840E+01
3300	1.122E+02	1.885E+01	8.600E+01	1.840E+01
3400	9.892E+01	1.885E+01	7.586E+01	1.840E+01
3500	8.744E+01	1.885E+01	6.700E+01	1.840E+01
3600	7.745E+01	1.885E+01	5.923E+01	1.840E+01

Time, s	No of sample			
	19		20	
	Temperature, °C	Oxygen content, %	Temperature, °C	Oxygen content, %
0	6.140E+00	2.062E+01	1.430E+01	2.055E+01
100	1.894E+01	2.050E+01	2.437E+01	2.045E+01
200	5.825E+01	2.014E+01	5.878E+01	2.012E+01
300	1.404E+02	1.931E+01	1.296E+02	1.940E+01
400	2.596E+02	1.767E+01	2.342E+02	1.804E+01
500	3.585E+02	1.502E+01	3.297E+02	1.576E+01
600	4.435E+02	1.121E+01	4.069E+02	1.247E+01
700	3.139E+02	9.145E+00	3.249E+02	1.430E+01
800	3.772E+02	1.611E+01	3.468E+02	1.657E+01
900	4.420E+02	1.515E+01	3.993E+02	1.580E+01
1000	5.141E+02	1.392E+01	4.591E+02	1.485E+01

1100	5.880E+02	1.252E+01	5.234E+02	1.375E+01
1200	6.613E+02	1.095E+01	5.363E+02	1.357E+01
1300	7.251E+02	9.365E+00	5.406E+02	1.357E+01
1400	7.340E+02	9.302E+00	5.433E+02	1.357E+01
1500	7.379E+02	9.298E+00	5.451E+02	1.357E+01
1600	7.403E+02	9.295E+00	5.463E+02	1.357E+01
1700	7.417E+02	9.293E+00	5.471E+02	1.357E+01
1800	7.427E+02	9.292E+00	5.477E+02	1.357E+01
1900	7.432E+02	9.291E+00	4.872E+02	1.466E+01
2000	7.436E+02	9.291E+00	4.063E+02	1.599E+01
2100	6.853E+02	1.078E+01	3.285E+02	1.709E+01
2200	5.693E+02	1.343E+01	2.797E+02	1.769E+01
2300	4.491E+02	1.552E+01	2.354E+02	1.826E+01
2400	3.620E+02	1.672E+01	1.979E+02	1.868E+01
2500	2.999E+02	1.751E+01	1.678E+02	1.893E+01
2600	2.449E+02	1.820E+01	1.432E+02	1.900E+01
2700	2.043E+02	1.854E+01	1.242E+02	1.900E+01
2800	1.732E+02	1.861E+01	1.088E+02	1.900E+01
2900	1.497E+02	1.861E+01	9.577E+01	1.900E+01
3000	1.307E+02	1.861E+01	8.456E+01	1.900E+01
3100	1.146E+02	1.861E+01	7.481E+01	1.900E+01
3200	1.007E+02	1.861E+01	6.626E+01	1.900E+01
3300	8.868E+01	1.861E+01	5.870E+01	1.900E+01
3400	7.825E+01	1.861E+01	5.203E+01	1.900E+01
3500	6.909E+01	1.861E+01	4.612E+01	1.900E+01
3600	6.105E+01	1.861E+01	4.089E+01	1.900E+01

Time, s	No of sample			
	21		22	
	Temperature, °C	Oxygen content, %	Temperature, °C	Oxygen content, %
0	1.290E+01	2.057E+01	1.000E+01	2.059E+01
100	3.130E+01	2.039E+01	2.191E+01	2.048E+01
200	8.499E+01	1.989E+01	5.365E+01	2.017E+01
300	1.938E+02	1.873E+01	1.181E+02	1.952E+01
400	3.258E+02	1.642E+01	2.142E+02	1.833E+01
500	4.270E+02	1.276E+01	3.101E+02	1.631E+01

600	3.968E+02	8.810E+00	2.638E+02	1.339E+01
700	2.522E+02	8.460E+00	2.766E+02	1.743E+01
800	4.610E+02	1.213E+01	3.292E+02	1.678E+01
900	5.525E+02	1.313E+01	3.808E+02	1.605E+01
1000	6.407E+02	1.141E+01	4.361E+02	1.521E+01
1100	7.260E+02	9.451E+00	4.949E+02	1.424E+01
1200	7.421E+02	8.956E+00	5.570E+02	1.312E+01
1300	7.973E+02	7.375E+00	6.206E+02	1.185E+01
1400	8.414E+02	6.005E+00	6.549E+02	1.113E+01
1500	8.463E+02	5.994E+00	6.607E+02	1.112E+01
1600	8.487E+02	5.990E+00	6.639E+02	1.112E+01
1700	8.501E+02	5.987E+00	6.659E+02	1.112E+01
1800	8.510E+02	5.985E+00	6.672E+02	1.112E+01
1900	8.515E+02	5.984E+00	6.680E+02	1.112E+01
2000	8.519E+02	5.984E+00	6.686E+02	1.112E+01
2100	8.521E+02	5.983E+00	6.305E+02	1.197E+01
2200	7.594E+02	8.991E+00	5.438E+02	1.383E+01
2300	6.769E+02	1.121E+01	4.555E+02	1.537E+01
2400	5.518E+02	1.368E+01	3.702E+02	1.664E+01
2500	4.125E+02	1.610E+01	3.091E+02	1.743E+01
2600	3.163E+02	1.742E+01	2.612E+02	1.804E+01
2700	2.542E+02	1.809E+01	2.190E+02	1.855E+01
2800	2.113E+02	1.825E+01	1.861E+02	1.884E+01
2900	1.807E+02	1.825E+01	1.591E+02	1.895E+01
3000	1.568E+02	1.825E+01	1.382E+02	1.895E+01
3100	1.369E+02	1.825E+01	1.213E+02	1.895E+01
3200	1.200E+02	1.825E+01	1.072E+02	1.895E+01
3300	1.054E+02	1.825E+01	9.506E+01	1.895E+01
3400	9.268E+01	1.825E+01	8.458E+01	1.895E+01
3500	8.165E+01	1.825E+01	7.544E+01	1.895E+01
3600	7.199E+01	1.825E+01	6.739E+01	1.895E+01

Time, s	No of sample			
	23		24	
	Temperature, °C	Oxygen content, %	Temperature, °C	Oxygen content, %
0	1.360E+01	2.056E+01	1.290E+01	2.057E+01

100	2.634E+01	2.043E+01	2.754E+01	2.043E+01
200	6.129E+01	2.009E+01	8.688E+01	1.988E+01
300	1.352E+02	1.933E+01	2.087E+02	1.858E+01
400	2.456E+02	1.784E+01	3.459E+02	1.602E+01
500	3.398E+02	1.544E+01	4.511E+02	1.194E+01
600	4.181E+02	1.197E+01	3.474E+02	8.731E+00
700	3.348E+02	1.321E+01	4.295E+02	1.512E+01
800	3.481E+02	1.653E+01	5.146E+02	1.392E+01
900	4.034E+02	1.572E+01	6.052E+02	1.216E+01
1000	4.666E+02	1.471E+01	6.491E+02	1.125E+01
1100	5.354E+02	1.352E+01	6.562E+02	1.125E+01
1200	6.048E+02	1.219E+01	6.601E+02	1.125E+01
1300	6.730E+02	1.071E+01	6.626E+02	1.124E+01
1400	7.406E+02	9.057E+00	6.642E+02	1.124E+01
1500	7.664E+02	8.434E+00	6.652E+02	1.124E+01
1600	7.710E+02	8.429E+00	6.659E+02	1.124E+01
1700	7.735E+02	8.425E+00	6.646E+02	1.126E+01
1800	7.751E+02	8.423E+00	5.752E+02	1.324E+01
1900	7.761E+02	8.421E+00	4.809E+02	1.497E+01
2000	7.767E+02	8.420E+00	4.016E+02	1.615E+01
2100	7.604E+02	8.773E+00	3.270E+02	1.718E+01
2200	6.330E+02	1.211E+01	2.673E+02	1.795E+01
2300	4.940E+02	1.481E+01	2.221E+02	1.846E+01
2400	3.836E+02	1.643E+01	1.871E+02	1.874E+01
2500	3.119E+02	1.735E+01	1.595E+02	1.880E+01
2600	2.512E+02	1.809E+01	1.384E+02	1.880E+01
2700	2.083E+02	1.840E+01	1.212E+02	1.880E+01
2800	1.767E+02	1.840E+01	1.067E+02	1.880E+01
2900	1.527E+02	1.840E+01	9.416E+01	1.880E+01
3000	1.330E+02	1.840E+01	8.333E+01	1.880E+01
3100	1.162E+02	1.840E+01	7.387E+01	1.880E+01
3200	1.018E+02	1.840E+01	6.554E+01	1.880E+01
3300	8.928E+01	1.840E+01	5.821E+01	1.880E+01
3400	7.831E+01	1.840E+01	5.175E+01	1.880E+01
3500	6.873E+01	1.840E+01	4.604E+01	1.880E+01
3600	6.034E+01	1.840E+01	4.099E+01	1.880E+01
Time, s	No of sample			

	25		26	
	Temperature, °C	Oxygen content, %	Temperature, °C	Oxygen content, %
0	9.910E+00	2.060E+01	8.250E+00	2.061E+01
100	2.370E+01	2.046E+01	1.855E+01	2.051E+01
200	6.639E+01	2.006E+01	5.210E+01	2.019E+01
300	1.548E+02	1.915E+01	1.225E+02	1.949E+01
400	2.776E+02	1.735E+01	2.296E+02	1.812E+01
500	3.753E+02	1.445E+01	3.267E+02	1.588E+01
600	3.173E+02	1.383E+01	4.053E+02	1.263E+01
700	3.291E+02	1.675E+01	2.769E+02	1.734E+01
800	3.919E+02	1.586E+01	3.336E+02	1.671E+01
900	4.629E+02	1.474E+01	3.904E+02	1.589E+01
1000	5.396E+02	1.340E+01	4.531E+02	1.492E+01
1100	6.175E+02	1.187E+01	5.196E+02	1.380E+01
1200	6.939E+02	1.016E+01	5.878E+02	1.251E+01
1300	7.490E+02	8.755E+00	6.038E+02	1.227E+01
1400	7.565E+02	8.734E+00	6.087E+02	1.226E+01
1500	7.601E+02	8.730E+00	6.117E+02	1.226E+01
1600	7.623E+02	8.727E+00	6.136E+02	1.226E+01
1700	7.637E+02	8.725E+00	6.149E+02	1.226E+01
1800	7.646E+02	8.723E+00	6.157E+02	1.226E+01
1900	7.652E+02	8.723E+00	6.163E+02	1.226E+01
2000	7.655E+02	8.722E+00	5.872E+02	1.280E+01
2100	7.653E+02	8.725E+00	4.726E+02	1.503E+01
2200	6.540E+02	1.161E+01	3.686E+02	1.661E+01
2300	5.238E+02	1.429E+01	3.054E+02	1.740E+01
2400	4.037E+02	1.619E+01	2.482E+02	1.815E+01
2500	3.325E+02	1.708E+01	2.057E+02	1.856E+01
2600	2.684E+02	1.793E+01	1.732E+02	1.866E+01
2700	2.218E+02	1.837E+01	1.492E+02	1.866E+01
2800	1.863E+02	1.849E+01	1.302E+02	1.866E+01
2900	1.601E+02	1.849E+01	1.141E+02	1.866E+01
3000	1.393E+02	1.849E+01	1.004E+02	1.866E+01
3100	1.217E+02	1.849E+01	8.849E+01	1.866E+01
3200	1.067E+02	1.849E+01	7.817E+01	1.866E+01
3300	9.370E+01	1.849E+01	6.910E+01	1.866E+01

3400	8.238E+01	1.849E+01	6.113E+01	1.866E+01
3500	7.249E+01	1.849E+01	5.412E+01	1.866E+01
3600	6.383E+01	1.849E+01	4.792E+01	1.866E+01
Time, s	No of sample			
	27		28	
	Temperature, °C	Oxygen content, %	Temperature, °C	Oxygen content, %
0	9.790E+00	2.060E+01	6.400E+00	2.062E+01
100	2.435E+01	2.046E+01	1.555E+01	2.053E+01
200	6.733E+01	2.006E+01	3.683E+01	2.032E+01
300	1.567E+02	1.915E+01	8.416E+01	1.985E+01
400	2.801E+02	1.732E+01	1.554E+02	1.904E+01
500	3.776E+02	1.441E+01	2.424E+02	1.766E+01
600	4.657E+02	1.024E+01	3.158E+02	1.563E+01
700	2.944E+02	8.661E+00	3.776E+02	1.287E+01
800	2.196E+02	8.351E+00	2.874E+02	1.480E+01
900	4.580E+02	1.301E+01	2.935E+02	1.728E+01
1000	5.443E+02	1.325E+01	3.373E+02	1.668E+01
1100	6.235E+02	1.170E+01	3.796E+02	1.608E+01
1200	6.999E+02	9.958E+00	4.262E+02	1.538E+01
1300	7.750E+02	8.010E+00	4.747E+02	1.461E+01
1400	8.068E+02	7.161E+00	5.264E+02	1.373E+01
1500	8.120E+02	7.153E+00	5.790E+02	1.276E+01
1600	8.149E+02	7.149E+00	6.276E+02	1.175E+01
1700	8.167E+02	7.146E+00	6.353E+02	1.168E+01
1800	8.177E+02	7.144E+00	6.387E+02	1.168E+01
1900	8.184E+02	7.143E+00	6.408E+02	1.168E+01
2000	8.188E+02	7.142E+00	6.421E+02	1.168E+01
2100	8.058E+02	7.447E+00	6.430E+02	1.168E+01
2200	6.996E+02	1.057E+01	6.435E+02	1.168E+01
2300	5.813E+02	1.327E+01	6.439E+02	1.168E+01
2400	4.590E+02	1.542E+01	5.657E+02	1.333E+01
2500	3.698E+02	1.667E+01	4.539E+02	1.535E+01
2600	3.068E+02	1.747E+01	3.571E+02	1.678E+01
2700	2.507E+02	1.818E+01	2.957E+02	1.755E+01
2800	2.095E+02	1.853E+01	2.413E+02	1.824E+01
2900	1.779E+02	1.861E+01	2.011E+02	1.860E+01

3000	1.540E+02	1.861E+01	1.702E+02	1.868E+01
3100	1.346E+02	1.861E+01	1.471E+02	1.868E+01
3200	1.183E+02	1.861E+01	1.284E+02	1.868E+01
3300	1.042E+02	1.861E+01	1.127E+02	1.868E+01
3400	9.207E+01	1.861E+01	9.916E+01	1.868E+01
3500	8.152E+01	1.861E+01	8.749E+01	1.868E+01
3600	7.228E+01	1.861E+01	7.733E+01	1.868E+01

Time, s	No of sample			
	29		30	
	Temperature, °C	Oxygen content, %	Temperature, °C	Oxygen content, %
0	1.110E+01	2.059E+01	6.990E+00	2.062E+01
100	3.174E+01	2.042E+01	2.510E+01	2.045E+01
200	9.262E+01	1.990E+01	8.792E+01	1.987E+01
300	2.161E+02	1.872E+01	2.150E+02	1.851E+01
400	3.649E+02	1.636E+01	3.545E+02	1.584E+01
500	4.795E+02	1.263E+01	4.619E+02	1.160E+01
600	4.297E+02	8.798E+00	3.343E+02	8.718E+00
700	4.472E+02	1.122E+01	4.381E+02	1.067E+01
800	5.301E+02	1.451E+01	5.285E+02	1.356E+01
900	6.261E+02	1.300E+01	6.234E+02	1.169E+01
1000	7.233E+02	1.122E+01	7.165E+02	9.511E+00
1100	8.184E+02	9.214E+00	7.995E+02	7.201E+00
1200	8.271E+02	8.907E+00	8.107E+02	7.088E+00
1300	8.965E+02	7.103E+00	8.153E+02	7.082E+00
1400	9.363E+02	6.014E+00	8.180E+02	7.077E+00
1500	9.410E+02	6.007E+00	8.197E+02	7.074E+00
1600	9.434E+02	6.003E+00	8.207E+02	7.073E+00
1700	9.448E+02	6.000E+00	8.214E+02	7.072E+00
1800	9.457E+02	5.999E+00	8.218E+02	7.071E+00
1900	9.462E+02	5.998E+00	8.220E+02	7.071E+00
2000	9.450E+02	6.013E+00	7.605E+02	8.871E+00
2100	8.441E+02	9.023E+00	6.443E+02	1.193E+01
2200	7.489E+02	1.128E+01	5.218E+02	1.439E+01
2300	6.218E+02	1.352E+01	4.059E+02	1.621E+01
2400	4.711E+02	1.589E+01	3.366E+02	1.708E+01
2500	3.623E+02	1.724E+01	2.744E+02	1.790E+01

2600	2.896E+02	1.801E+01	2.276E+02	1.839E+01
2700	2.395E+02	1.829E+01	1.914E+02	1.858E+01
2800	2.032E+02	1.829E+01	1.643E+02	1.858E+01
2900	1.755E+02	1.829E+01	1.431E+02	1.858E+01
3000	1.528E+02	1.829E+01	1.254E+02	1.858E+01
3100	1.335E+02	1.829E+01	1.103E+02	1.858E+01
3200	1.168E+02	1.829E+01	9.720E+01	1.858E+01
3300	1.024E+02	1.829E+01	8.590E+01	1.858E+01
3400	8.985E+01	1.829E+01	7.602E+01	1.858E+01
3500	7.883E+01	1.829E+01	6.736E+01	1.858E+01
3600	6.920E+01	1.829E+01	5.976E+01	1.858E+01

Annex B. Results of the analysis of extreme winds recorded in Lithuania in the last 48 years

Lithuanian classification of extreme winds corresponds to the international classification based on the Beaufort scale (e.g., Liu 1991, Allaby 2007). The two most severe categories of the wind speed v in the Beaufort scale are violent storm (28 m/s < v < 32 m/s or 64 mph < v < 73 mph) and hurricane ($v \geq 33$ m/s or $v \geq 73$ mph). These two categories of winds are used in the Lithuanian classification of extreme natural and man-made phenomena (Lietuvos Vyriausybė 2006).

A further Lithuanian categorisation of extreme winds distinguishes between "elemental" or strong wind (30 m/s ≤ v < 35 m/s) and "catastrophic" wind ($v \geq 35$ m/s) (Korkutis 2000). Although the term "elemental" is a more precise translation of Lithuanian "*stichinis*", the term "strong wind" is more natural and will be used in the subsequent text. The categorisation of extreme winds into strong and catastrophic ones does not agree strictly with the Beaufort scale (see, e.g., the definitions of the scale presented by Liu (1991) and Allaby (2007)).

A reliable recording of wind speeds in Lithuania began in early 1960s when weather stations started to use adequate instrumentation for obtaining and recording wind data. The data on extreme winds recorded in the period 1962 to 2005 is summarised in Table B.4. In the subsequent years 2006 to 2009, no mean wind speeds exceeding 30 m/s were recorded. The data given in Table B.4 are either 10-min means recorded at 10 m elevation of rotating cups anemometers or records adjusted to these instrumentation parameters. The network of Lithuanian weather stations with the records of strong and catastrophic winds is shown in Fig. B.1.

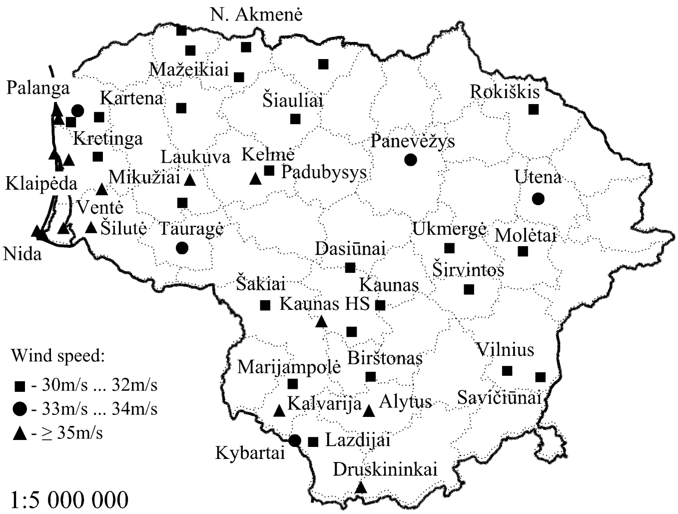

Fig. B.1. Map showing the network of Lithuanian weather stations with 10-min wind speeds of at least 30 m/s recorded in 1962-2005 (the geographic coordinates the stations were provided by Lithuanian Hydrometeorological Service, http://www.meteo.lt/)

In the period 1962 to 2009, a total of 33 wind storms reached the level of extreme winds. This results in the annual frequency of occurrence equal to 0,688 per annum. Four storms (18 Oct 1967, 23 Nov 1973, 14 Jan 1993, and 4 Dec 1999) covered more that one third of Lithuanian territory (were recorded in at least seven weather stations). The largest storm recorded in 24 weather stations occurred on 14 Jan 1993.

The mean wind speeds of the extreme winds ranged between 30 m/s to 40 m/s. The frequencies of the wind speeds belonging to this range are shown in Fig. B.2. These frequencies were calculated for a sample of wind speeds which includes only one value of all speeds recorded in each individual storm, no matter how many whether stations measured this speed. Thus 30 m/s wind speeds were recorded in 25 storms, 31 m/s wind speeds were recorded in 6 storms and so on. In counting the frequencies from the data given in Table B.4, the values “over 30” and the like were replaced by a non-conservative “30” and the values “approx. 30” and similar by a fixed “30”. The sample consists of 70 wind speeds.

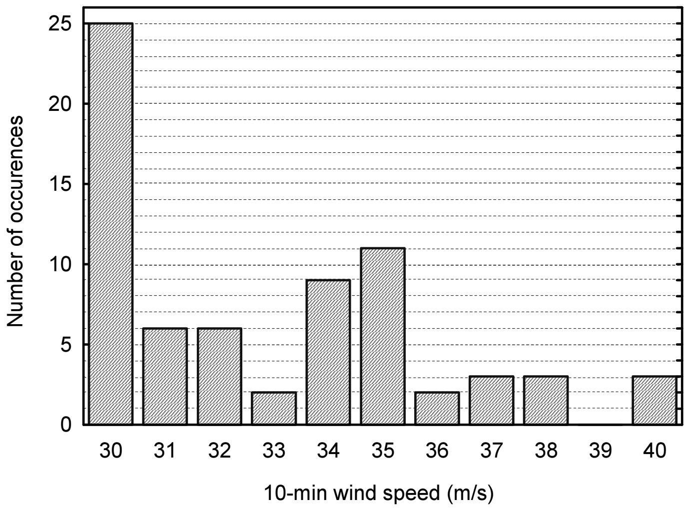

Fig. B.2. Absolute frequencies of wind speeds recorded in 33 storms with the wind speed of at least 30 m/s in the 44 years period 1962 to 2002

The same sample of 70 winds speeds recorded in 33 storms was used to calculate annual frequencies of a non-strict exceedance of given wind speed values (Fig. B.3). These frequencies are defined by

$$fr(v) = \frac{\text{Number of records greater or equal to } v}{\text{Period of record (years)}} \quad \text{(B.1)}$$

The frequencies $fr(v)$ were calculated for the 48 year period 1962 to 2009.

The catastrophic winds with $v \geq 35$ m/s occurred in 15 storms in the period 1962-2005 (Table B.4). During the storm, they were recorded in one weather station 9 times, in two stations 3 times, in three stations 2 times, and once the storm with catastrophic winds was observed in 5 stations. In total, the catastrophic winds were recorded in 16 weather stations (Fig. B.4). The most severe storms with the mean wind speeds of 40 m/s were recorded three times, namely, on 18 Oct 1967, 31 Oct 1970, and 4 Dec 1999. All of these wind speeds were recorded in coastal wind stations.

Reliability and micrometeorological homogeneity of data on extreme winds

The instrumentation of used for obtaining the wind data in Lithuania was of two types (Korkutis 1996: 10):

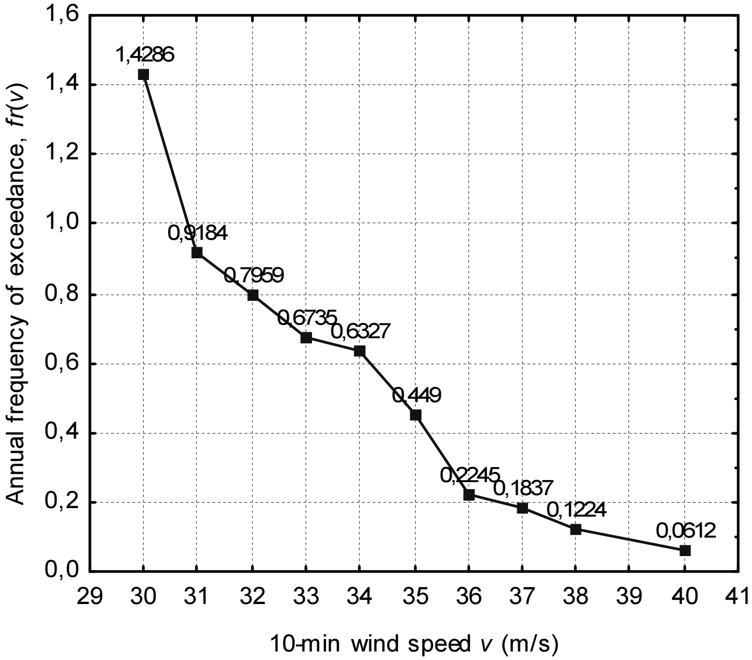

Fig. B.3. Annual frequencies of a non-strict exceedance of the wind speed *v* calculated for the period 1962 to 2009

- In the years 1961 to 1976, the wind speeds were measured by means of swinging-plate (pressure-plate) anemometers;
- Since 1976, the swinging-plate anemometers were replaced by windmill anemometers (the names of the devices were adopted from Allaby (2007)).

Information is available to adjust the wind records taken by swinging-plate anemometers to agree with the readings of the windmill anemometers (Korkutis 1996). The wind speeds recorded in Table B.4 and covering the period 1960 to 1975 are adjusted readings of the swinging-plate anemometers. The set of wind speed data given in Table B.4 may be considered to be a relatively homogenous one because all the data belonging to the set have been obtained under equivalent conditions. These conditions were determined by the following factors: averaging time, height of anemometers above ground, roughness of terrain surrounding a weather station (e.g., Simiu and Scanlan 1996). As applied to the Lithuanian weather stations, the following statements can be made in favour of or against data homogeneity:

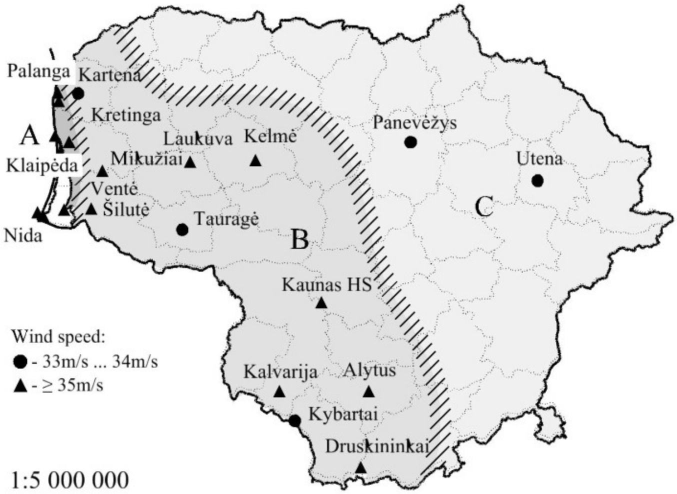

Fig. B.4. Weather stations with the records of hurricane and catastrophic winds as well as proposed hurricane wind regions A, B, and C

- The same averaging time of 10 minutes has been used in all weather stations during the period of record;
- The 10.2 m elevation of anemometers has been used in all weather stations since 1986; prior to this year, elevations above 10.2 m have been used in a small number of stations (e.g., the costal station in Nida); all speeds from these stations included in Table B.4 were corrected to agree with the readings at 10.2 m height;
- In some weather stations listed in Table B.4, anemometer locations have been changed during the period of record; for instance, the stations in Laukuva, Šilutė and Panevėžys were relocated to more open areas and this had increased the quality of data recorded in these stations (Korkutis 1996); however, it doesn't seem that the records given in Table B.4 have been adjusted to a common terrain roughness.

With one exception, the sensors of wind speed in all Lithuanian stations were exposed in such a way that they were not influenced by local flow effects. In recent years, a land development around the Klaipėda costal weather station has changed the roughness of terrain surrounding its anemometer; however, these changes have not affected the terrain roughness in the prevailing directions of extreme winds. Consequently, there is no need to adjust wind speeds 38 m/s and 32 m/s recorded in this station during the storms 31 and 32 included in Table B.4 (Galvonaitė 2009).

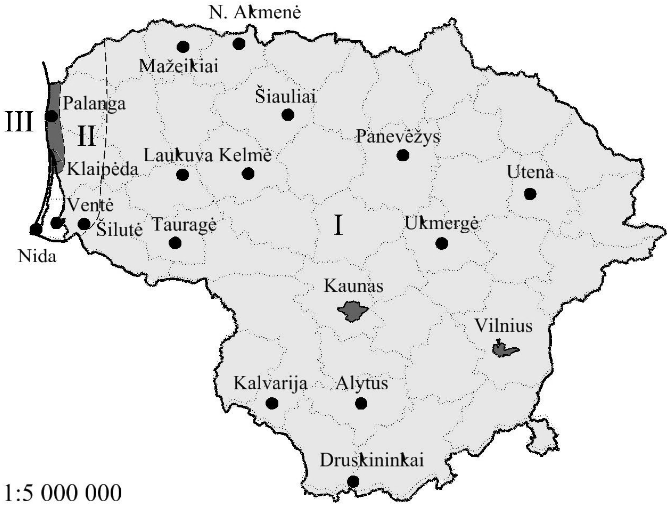

Fig. B.5. The map of basic reference wind speeds $v_{ref,d}$ specified in the design code STR 2.05.04:2003 (see also Table B.1)

The maximum wind speeds of 40 m/s recorded in 1967 and 1970 by means of swinging-plate anemometers are of limited reliability. These instruments have the range of wind speeds limited by 40 m/s. It might be that the mean wind speeds of 40 m/s recorded in 1967 and 1970 have been exceeded, to say nothing about gust wind speeds. Another fact which may also negatively influence the reliability of the data set given in Table B.4 is a complete absence of the 39 m/s wind speed values (Fig. B.2). It is difficult to explain what has caused such a result, faulty observations or, simply, a pure coincidence.

One can conclude that the set of the wind speeds presented in Table B.4 is not unquestionable in terms of reliability and meteorological homogeneity. On the other hand, this data set does not have grave and obvious faults which could prevent it from being used for assessing the risks posed by extreme winds.

Estimation of data on extreme winds in relation to structural design

The set of wind speed data given in Table B.4 can be estimated by comparing these data with the wind speeds specified in the loading codes STR 2.05.04:2003 and SNiP 2.01.07-85. They explicitly define serviceability and ultimate limit sates for structural design and provide either basic reference wind speeds (the case of STR) or characteristic wind pressures (the case of SNiP) for the regions of Lithuania shown in Figs B.5 and B.6.

The provisions of STR and SNiP are summarised in Tables B.1 and B.2, respectively. The basic reference wind speeds $v_{ref,0}$ specified in the STR code are

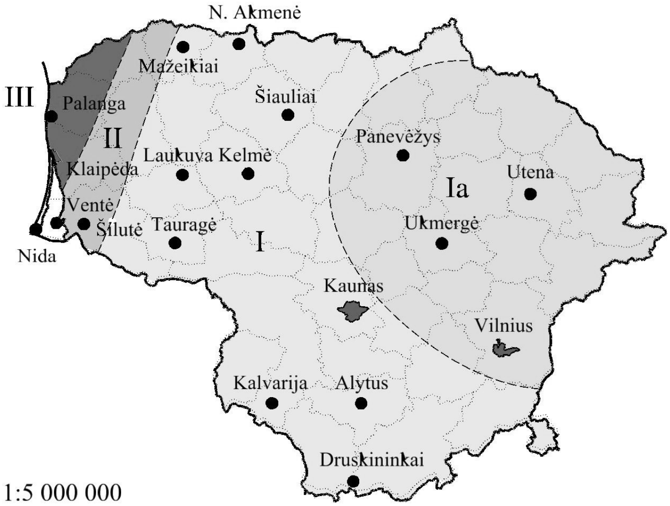

Fig. B.6. The map of characteristic wind pressures w_0 specified in the design code SNiP 2.01.07-85 (see also Table B.2)

10-min mean velocities at 10 m height in open county terrain with an annual probability of exceedance of 0,02 (50-year return period) (Col. 2, Table B.1). The characteristic wind pressures w_0 provided in the SNiP code were defined analogously with the exception of the return period. This period was set to be equal to 5 years although the code allowed to specify longer return periods (Col. 2, Table B.2). The reason why such a short basic return period was specified in the SNiP code is not known to us.

A natural approach to an estimation of the wind data given in Table B.4 is a comparison of these data with the design wind speeds $v_{ref,d}$ and v_d which were obtained by applying partial factors for the wind specified in STR and SNiP codes (Cols. 5 and 4 in Tables B.1 and B.2, respectively). As the basic reference wind speed $v_{ref,0}$ is defined using the annual exceedance probability of 0,02, one can expect that such a probability for $v_{ref,d}$ will be much smaller.

The design wind speeds $v_{ref,d}$ and v_d (or corresponding dynamic wind pressures) have been applied in the years 1987 to 2009 to the design of structures for ultimate limit states. The frequency and magnitude of exceedance of these values by the mean wind speeds recorded during wind storms are obvious criteria for assessing the hazard to structural property posed by the storms listed in Table B.4.

One can say in this regard that the design wind speeds specified in the SNiP code, v_d, were exceeded during all 33 storms recorded in all regions shown in Fig. B.6. After the introduction of the SNiP code in 1987 and until the last storm recorded in 2005, largest design value v_d = 25,9 m/s specified in SNiP for the coastal region III was exceeded four times by the winds ranging from 30 m/s to 40 m/s. This yields the annual exceedance frequency of 0,21 per annum. The 30 m/s to 40 m/s winds were 1,16 to 1,54 times faster than the wind with the speed v_d. The situation in the remaining wind speed regions I, Ia, and II is even more risky.

The design wind speeds provided in the STR code, $v_{ref,d}$, are more conservative that those specified in the SNiP code (compare Tables B.1 and B.2). Despite the fact that the "coastal" design wind speed $v_{ref,d}$ = 36,5 m/s is relatively high, this speed was already reached and even slightly exceeded once after the introduction of the STR code in 2003 (the storm of 2005, Table B.4). If we consider a long-term exceedance frequency related to the entire period of record in region III, 1962 to 2009, this frequency amounts to 0,10. Table B.3 summarises numbers of exceeding and annual exceedance frequencies calculated for the three wind speed regions shown in Fig. B.5 (see Col. 3). The annual frequencies given in Table B.3 exceed by far the annual probability 0,02 used in the STR code to specify the basic reference wind speeds $v_{ref,0}$ for all regions. This difference is particularly grave in case of the largest region I. This region appears to be most problematic also in terms on exceedance magnitude expressed by mean values and maxima of the differences between the winds speed given in Table B.4, v_{ij}, and the design wind speed $v_{ref,d}$ calculated for all three regions (Cols. 4 and B.5, Table B.3). Although the magnitude of exceedance of the design values $v_{ref,d}$ is not dramatic, it can be a cause of concern due to the following reasons:

- A relatively often exceedance of the design values (once every 1 to 10

Table B.1. Wind speed regions in Lithuania specified in the structural design code STR 2.05.04:2003[4] used since 2003

Wind speed region (see Fig B.5)	The basic reference wind speed $v_{ref,0}$, m/s[1]	Characteristic wind pressure q_{ref}, kPa	Design wind pressure $\gamma_Q\, q_{ref}$, kPa[2]	Basic design wind speed $v_{ref,d}$, m/s[3]
1	2	3	4	5
I	24	0.36	0.468	27.4
II	28	0.49	0.637	31.9
III	32	0.64	0.832	36.5

[1] A 10-min mean velocity at 10 m height in open country terrain with 50-year return period

[2] The partial factor for the wind action $\gamma_Q = 1.3$

[3] Calculated by the standard formula $v_{ref,d} = (2\, \gamma_Q\, q_{ref}/\rho)^{1/2}$ with the air density ρ = 1.25 kg/m^3; q_{ref} in Pa

[4] STR 2.05.04:2003 Poveikiai ir apkrovos. Vilnius, 2003. 103 p.

Table B.2. Wind speed regions specified in the structural design code SNiP 2.01.07-85[4] used in Lithuania in the years 1987 to 2002

Wind speed region (see Fig B.6)	Characteristic wind pressure w_0, kPa[1]	Design wind pressure $\gamma_f w_0$, [2]	Design wind speed v_d, m/s[3]
1	2	3	4
Ia	0.17	0.238	19.5
I	0.23	0.322	22.7
II	0.30	0.420	25.9
III	0.38	0.532	29.2

[1] 5-year return pressures with a 10-min averaging time at 10 m height in open country terrain

[2] The partial factor for the wind action $\gamma_f = 1.4$

[3] Calculated by the standard formula $v_d = (2\,\gamma_f\, w_0/\rho)^{1/2}$ with the air density $\rho = 1.25$ kg/m^3; w_0 in Pa

[4] СНиП 2.01.07-85 Нарузки и воздействия, [SNiP 2.01.07-85 Apkrovos ir poveikiai] Москва: ГКСПДС, 1986, 35 с.

years, depending on the region) can lead to an accumulation of wind-induced damage in a wind-sensitive structure;

- A gradual deterioration of structural resistance due to, say, corrosion may become dangerous during the wind loading peaks occurring as an exceedance of the design wind speeds for which the structure was dimensioned;
- Older wind-sensitive structures designed and built in 1983 to 2002 in line with the SNiP code (or codes used in previous years) may be prone to the peak loading induced by the winds exceeding the design wind pressures specified in these codes.

The exceedance data shown in Table B.3 lead to an obvious conclusion: the provisions of the STR design code are not sufficiently conservative, especially, with regard to the largest, inland wind speed region I. To the best of our knowledge, the basic reference wind speeds $v_{ref,0}$ were determined for all three regions by a standard extreme value procedure based on annual fastest wind series, although this was not documented in the open literature.

In our opinion, the exceedance of the design wind speeds $v_{ref,d}$ highlights the need to separate data expressed as the annual fastest wind series and data recorded during wind storms. The storm data can be grouped into the so-called partial duration series (e.g., Liu 1991, Ben-Zvi 2009). The need for such grouping arises from the fact that no storms were recorded in Lithuania during

Table B.3. Numbers of exceeding of the design wind speeds $v_{ref,d}$ specified in STR 2.05.04:2003 and corresponding frequencies of exceedance calculated for the 48 years period 1962 to 2009

Wind speed region j (see Fig. B.5)	Basic design wind speed $v_{ref,d}$, m/s	Numbers of exceeding N_j / annual frequency $N_j/48$	Mean value of exceeding $v_{ij} - v_{ref,d}$, m/s[1]	Maximum of exceeding $v_{ij} - v_{ref,d}$, m/s[1]
1	2	3	4	5
I	27.4	47 / 0.98 per annum	4.17	10.6
II	31.9	15 / 0.31 per annum	3.83	8.10
III	36.5	5 / 0.10 per annum	1.90	3.50

(1) The symbol v_{ij} denotes the ith 10-min wind speed from Annex A recorded in a weather station located in a respective wind speed region j ($i = 1, 2, \dots, N_j$)

27 years spread within the period 1962 to 2009, whereas the years 1967, 1969, 1971, and 1981 had more than one wind storm. The creation of the partial duration series evokes the question of dividing the territory of Lithuania into storm regions and combining storm data from several weather stations in each region.

Table B.4. Storms with 10-min. wind speeds of at least 30 m/s recorded in Lithuania in the 44 year period of 1962-2005 (extracted from Korkutis (2000) and Lietuvos Hidrometeorologijos Tarnyba [5])

No	Date	Weather station	10-min wind speed, m/s	Wind directi on	Start of storm, hh:mm	Storm duration
1	2	3	4	5	6	7
			1960 – 1969			
1	1962 01 23	Klaipėda	34	SW	~ 07:00	not rec.
2	1962 02 17	Klaipėda	30	NW	~ 19:00	not rec.
3	1964 01 27	Kaunas AS[1]	30	NW	02:30	37min
4	1965 12 06	Klaipėda AS	30	W	16:12	16h 7min
5	1967 10 18	Klaipėda Nida	40 40	SW SW	00:25 morning	12h 30min not rec.

		Palanga AS	35	SW	02:00	12h 40min
		Kybartai	34	W-SW	02:07	not rec.
		Laukuva	34	S	morning	not rec.
		Rokiškis	30	S-SW	morning	not rec.
		Šiauliai	30	S-SW	morning	not rec.
		Šilutė	30	SW	morning	not rec.
6	1967 11 12	Klaipėda	30	SW, S-SW/N W	21:00/02:40	2h 45min/ 4h 40min
7	1969 03 09	Kaunas HS[(1)]	38	SW	08:00	12h 30min
		Klaipėda AS	34	SW	13:20	10h 40min
		Širvintos	~ 31	SW	~ 08:00	not rec.
		Padubisys	30	SW	10:35	8h 25min
8	1969 11 02	Klaipėda	36	W-NW	03:30	8h 40min
		Klaipėda AS	35	W-NW	00:41	12h 19min
	1969 11 01	Klaipėda	34	W	22:35	1h 25min
		Utena	30	W-SW, W-NW	02:30	1h 35min
	1969 11 03	Birštonas	~ 31	not rec.	not rec.	not rec.
9	1969 11 20	Klaipėda	30	SW	03:55	1h 20min
	1969 11 21	Klaipėda AS	33	SW	23:05	5h 55min
1970 – 1979						
10	1970 07 21	Klaipėda AS, Tubausiai	34	SW	13:05	3h 55min
11	1970 10 20	Klaipėda	35	SW	13:40	21h 20min
12	1970 10 31	Klaipėda	40	W	03:15	5h 40min
	1970 11 01	Klaipėda	30	W	18:10	8h 10min
	1970 11 02	Klaipėda	30	W	06:30	1h 25min
		Kaunas AS	30	NW	10:15	1h
13	1970 11 10	Klaipėda	30	NW	14:10	41min
14	1971 10 22	Šilutė	34	W	20:40	7h 20min
		Palanga AS	30	SW	~ 17:05	not rec.
		Garliava	30	W	20:00	10h 30min
		Kybartai	30	SW	21:00	6h 00min
		Birštonas	30	NW	20:30	8h 30min

15	1971 12 08	Palanga AS	35	NW	11:00	19h 45min
16	1971 12 22	Palanga AS	30	NW	06:35	1h 10min
17	1972 03 23	Druskininkai	38	NW	~ 20:00	not rec.
		Palanga AS	37	NW	16:25	20min
		Kalvarija	35	W	~ 19:00	not rec.
		Lazdijai	> 30	NW	~ 20:00	not rec.
18	1973 11 23	Klaipėda	35	NW	~ 15:00	not rec.
		Panevėžys	34	W-NW	afternoon	not rec.
		Utena	34	NW	~ 18:00	not rec.
		Šilutė	34	NW	afternoon	not rec.
		Ventė	34	W-NW	afternoon	not rec.
		Kybartai	34	W-NW	11:00	9h
		Birštonas	34	W	afternoon	not rec.
		Mikužiai	> 33	NW	~ 15:00	not rec.
19	1975 01 06	Klaipėda	32	W	~ 06:00	not rec.
20	1975 12 31	Klaipėda	30	W-PW	~ 12:00	not rec.
21	1976 01 12	Alytus	35	W	~ 21:00	not rec.
		Kalvarija	30	W	~ 21:00	not rec.
			1980 – 1989			
22	1977 09 14	Klaipėda	30	SW	~ 21:00	not rec.
		Šilutė	30	W-SW	~ 00:00	not rec.
23	1981 11 21	Šilutė	30	W-SW	daytime	not rec.
		Klaipėda	30	SW	~ 13:00	not rec.
24	1981 11 25	Kelmė	35	SW	night	not rec.
		Mikužiai	35	SW	night	not rec.
		Šilutė	35	S-SW	06:36	4h 54min
		Nida	34	S-SW	morning	not rec.
		Kybartai	32	SW	morning	not rec.
		Klaipėda	30	SW	02:00	4h
25	1982 12 16	Klaipėda	35	W	16:35	1h 40min
		Šilutė	30	W-SW	afternoon	not rec.
		Nida	30	S	afternoon	not rec.
26	1983 01 18	Palanga	35	W	16:00	2h
		Šilutė	31	W-SW	daytime	not rec.
		Klaipėda	30	W	21:25	2h 20min
	1983 01 19	Šilutė	31	WNW	daytime	not rec.
		Nida	30	W	daytime	not rec.

27	1983 01 29	Šilutė	31	W-SW	in the	not rec.
28	1984 01 02	Klaipėda	30	W	~ 18:00	not rec.
29	1986 11 06	Nida	30	NW	20:05	55min
			1990 – 1999			
30	1993 01 14	Nida, coastal	36	SW	09:00	2 h
		Laukuva	35	W	09:05	2h 55min
		Telšiai	32	W-SW	07:25	6h 35min
		Ukmergė	31	W-SW	10:18	1h 07min
		Kybartai	31	W-SW	11:26	7h 34min
		Leckava	> 30	W-SW	morning	not rec.
		Mažeikiai	> 30	W-SW	morning	not rec.
		N. Akmenė	> 30	W-SW	09:45	9h 15min
		Joniškis	> 30	W-SW	morning	not rec.
		Papilė	> 30	W-SW	morning	not rec.
		Paakmenis	> 30	W	10:10	1h 50min
		Vilnius AS	30	W	~ 11:00	not rec.
		Panevėžys	30	W-SW	09:35	3h 50min
		Vėžaičiai	30	SW	07:15	3h 15min
		Kelmė	30	W-SW	~ 10:00	not rec.
		Molėtai	30	SW	morning	not rec.
		Dasiūnai	30	W-SW	11:00	7h
		Šakiai	30	W-SW	11:00	3h
		Savičiūnai	30	W	11:00	8h
		Laukuva	30	W	night	not rec.
		Kartena	~ 30	W-SW	morning	not rec.
		Mikužiai	~ 30	SW	07:15	3h 15min
		Kretinga	~ 30	W-SW	morning	not rec.
	1993 01 15	Marijampolė	31	W-SW	11:30	8h
31	1999 12 04	Nida	40	S, SW(3)	06:33	10h 20min
		Ventė	40		08:00	8h
		Mikužiai	38		03:30	5h 30min
		Klaipėda	38	S, SE(2)	04:30	11h 45min
		Šilutė	37	SW, W(4)	06:53	8h 02min
		Vėžaičiai	32		10:30	6h 30min

		Marijampolė Birštonas	31 30		~ 6:00 not rec.	not rec. not rec.
32	2002 01 29	Tauragė Klaipėda Mažeikiai	34 32 30	W-SW	8:15 16:20 17:30	15min 10h 25min not rec.
33	2005 01 09	Klaipėda, seaport Palanga	37 32	W W-SW	02:05 00:00	3h 2h

(1) AS = station at the airport; HS = hydrological station

(2) Prevailing direction in the evening of 1999 12 03, all stations

(3) Prevailing direction during the night and the 1st half of the day of 1999 12 04, all stations

(4) Prevailing direction during the night and the 2nd half of the day of 1999 12 04, all stations

(5) Lietuvos Hidrometeorologijos Tarnyba 2008. *1996-2008 m. stichinių meteorologinių reiškinių apžvalga* [A Review of Extreme Meteorological Phenomena in 1996-2008]. Vilnius: Lietuvos hidrometeorologijos tarnyba. 117 p.

Annex C. Computer code "Timber beam"

```
      PROGRAM Prior_main
*
     INCLUDE 'constants.for'
*
* The "seed" of simulation
     INTEGER seed
* Number of repetitions of outer loop
* Number of repetitions of inner loop
     INTEGER n0, nr
* The required time to failure
     REAL*8  treq
* The number of time steps in the fire history
      INTEGER ntau
* The number of fire under consideration
      INTEGER nfire
* Basic variables used in the problem
      REAL*8  z0,z1,z2,z3,z4,z5,z6,z7,z8,z9,z10
* Resistance and action effect
      REAL*8  resistance, action_effect, safety_margin
```

```
* Charring depth
     REAL*8  d_char
*
* Time and temperature for fire history
      REAL*8
t(TIMESTEPMAX),temp(TIMESTEPMAX),oxygen(TIMESTEPMAX)
     REAL*8 temp_oxy(2,TIMESTEPMAX,MAXFIRES)
* Vectors of parameters used in the carring model
     REAL*8 pi(MODELMAX,OUTMODELMAX)
* Auxiliaries
     REAL*8  w,m_crit,sigma_crit,sigma_max
      INTEGER max_k
     REAL*8  y1,y2,y3,psi(10),p(10)
      REAL*8  pfe,pf_ll,pf_ul,prob1,prob2,dens,prob,i_binar
      REAL*8  sums(20)
      REAL*8  means(20),stds(20),mean1,std1,mean2,std2,tau
*
      REAL*8 URAND, GRAND, LGRAND, GAMMA1
* Variables and arrays used to compute char depth values
*
     OPEN(1,FILE='actiondata.txt',STATUS='OLD')
     OPEN(2,FILE='Fire_histories.txt',STATUS='OLD')
      OPEN(5,FILE='PRIOR_RESULTS.TXT',STATUS='OLD')
      OPEN(6,FILE='PRIOR_CURRENT_LOOP.TXT',STATUS='OLD')
      OPEN(7,FILE='CRITICAL SECTION.TXT',STATUS='OLD')
     OPEN(11,FILE='History_out.txt',    STATUS='OLD')
     OPEN(12,FILE='Current_history.txt',  STATUS='OLD')
     OPEN(14,FILE='One column output.txt',STATUS='OLD')

      DATA sums/20*0.d0/
*
      WRITE (*,*) 'Prior is running...'
*
* READ THE DATA
*
      CALL READDATA(seed,n0,nr,treq,ntau,nfire)
* Read the time-history of fire
      DO it=1,ntau
        READ(2,'(E11.5,60E12.4)') t(it),
    &       (temp_oxy(1,it,ifire),temp_oxy(2,it,ifire),ifire=1,30)
         t(it)=t(it)/60
```

```
      END DO
*
* ASSIGN & WRITE THE INPUT DATA RELATED TO FIRES
*
      DO it=1,ntau
        temp(it)=temp_oxy(1,it,nfire)
        oxygen(it)=temp_oxy(2,it,nfire)
       WRITE(12,'(F5.2,2E12.4)') t(it), temp(it), oxygen(it)
        WRITE(11,'(F5.2,60E12.4)') t(it),
    &      (temp_oxy(1,it,ifire),temp_oxy(2,it,ifire),ifire=1,30)
      END DO
*
* OUTER LOOP OF MC SIMULATION FOR HANDLING EPISTEMIC
UNCERTAINTIES
      DO i=1,nr
* Generate values of uncertain parameters in the prior action model
        psi(1)=LGRAND(seed,0.05d0,1.d0)
        psi(2)=LGRAND(seed,0.16d0,0.8d0)
* Assign values of parameters for the distribution of snow load
        psi(3)=3.289d0
        psi(4)=0.6045d0
* Generate values of uncertain parameters of the char depth distribution
        psi(5)=GRAND(seed,0.004d0,0.04d0)
        psi(6)=GAMA1(seed,278,0.01d0)
        psi(6)=DSQRT(1.d0/psi(6))
*       WRITE(14,'(I5,2D12.5)') i, psi(5), psi(6)
* Assign values to parameters (sample values of parameters) of timber charring
**      CALL PARAMETERS(seed,1,pi)
* Assign mean values of the basic random variables
        means(1)=3.5d0    ! Density
        means(2)=0.14d0   ! Width
        means(3)=0.40d0   ! Height at supports
        means(4)=1.00d0   ! Height at midspan
        means(5)=0.0d0    ! Height at critical section
        means(6)=1.53d0   ! Line load from roof cover
        means(7)=0.0d0    ! Snow (not used here)
        means(8)=37.4d0   ! Timber strength
        means(9)=8.1d0   ! Wood moisture content
        means(10)=0.75d0  ! Beam surface emmisivity
* Assign standard deviations of the basic random variables
        stds(1)=0.245d0   ! Density
```

```
        stds(2)=0.005d0   ! Width
        stds(3)=0.005d0   ! Height at supports
        stds(4)=0.01d0    ! Height at midspan
        stds(5)=0.01d0    ! Height at critical section
        stds(6)=0.0765d0  ! Line load from roof cover
        stds(7)=0.0d0     ! Snow (not used here)
        stds(8)=4.862d0   ! Timber strength
        stds(9)=0.486d0   ! Wood moisture content
        stds(10)=0.0375d0 ! Beam surface emmisivity

* Determine the position of the critical section by means of the mean values
*       IF(i.EQ.1) THEN
          max_k=0
          DO k=1,581
            IF(k.EQ.1) sigma_max=0.d0
            means(5)=means(3)+k*(means(4)-means(3))/581
           w=0.16667d0*means(2)*means(5)**2
           m_crit=0.5d0*(means(6)+means(1)*means(2)*means(3))*
     &           0.01*k*(11.62d0-0.01*k)+
     &           (means(4)-means(3))*means(2)*means(1)*11.62d0/2*
     &           0.01*k*(0.5d0-(2*(0.01*k)**2)/(3*(11.62d0)**2))
           sigma_crit=m_crit/w
           IF(sigma_crit.GT.sigma_max) max_k=k
            sigma_max=DMAX1(sigma_max,sigma_crit)
          END DO
*       Compute the critical section height
          means(5)=means(3)+max_k*(means(4)-means(3))/581
*       END IF
        WRITE(*,*) i, means(5)
* Zero values of the sums used for the computation of descriptive statistics
       DO k=1,20
          sums(k)=0.d0
        END DO
* NESTED LOOP FOR HANDLING ALEATORY UNCERTAINTIES
        j=0
        DO j=1,n0
*   Generate values of arguments of the prior action model
*         Generate timber density
             z1=LGRAND(seed,stds(1),means(1))
*         Generate cross-sectional width
          z2=LGRAND(seed,stds(2),means(2))
```

```
*        Generate cross-sectional height at the ends, initial = 0.47d0
         z3=LGRAND(seed,stds(3),means(3))
*        Generate cross-sectional height at the midspan, initial = 1.24d0
         z4=LGRAND(seed,stds(4),means(4))
*        Generate cross-sectional height at the critical section, initial = 0.76d0
         z5=LGRAND(seed,0.01d0*means(5),means(5))
*        Generate the line load from the rooof cover
         z6=LGRAND(seed,stds(6),means(6))
*        Generate the line load from snow
        z0=-DLOG(URAND(seed))
           IF(z0.LE.0.d0) THEN
          j_chaltura=j_chaltura+1
             WRITE(*,'(2I10,D20.10)') j,j_chaltura,z0
*          STOP
         END IF
        z0=DMAX1(1.d-200,z0)
*
         z7=4.0d0*psi(2)*(psi(4)-DLOG(z0)/psi(3))
*        Generate the timber strength
         z8=LGRAND(seed,stds(8),means(8))
*        Generate the value of wood misture content
         z9=GRAND(seed,stds(9),means(9))
*        Generate the surface emmisivity
         z10=GRAND(seed,stds(10),means(10))
*
* Compute the simulated value of the char depth
**          CALL ACCIDENTMODEL(i,j,seed,pi,z1,z9,z10,
**     &                  treq,ntau,t,temp,oxygen,d_char)
*     WRITE(5,'(E15.7)') d_char
*    Assign value to a charring depth
*        d_char=0.000d0
* Convert the charring depth from mm to m
**       d_char=(d_char/1000)*2
**       d_char=0.03d0
         d_char=GRAND(seed,psi(6),psi(5))
*
         resistance=psi(1)*
    &           (0.166667d0*z8*(z2-2*d_char)*(z5-2*d_char)**2)*1.d3
        action_effect=0.5d0*(z6+z1*z2*z3+z7)*
    &          0.01*max_k*(11.62d0-0.01*max_k)+
    &          (z4-z3)*z2*z1*11.62d0/2*
```

```
    &          0.01*max_k*(0.5d0-(2*(0.01*max_k)**2)/(3*(11.62d0)**2))
*
            y1=resistance
            y2=action_effect
          safety_margin = resistance - action_effect
          sums(1)=sums(1)+y1
          sums(2)=sums(2)+y2
          i_binar=0.d0
          IF(safety_margin.LE.0.d0) i_binar=1.d0
            prob=i_binar
*           prob=prob1*prob2
* Accumulate information for calculating descriptive measures
          sums(9)=sums(9)+prob
          IF (j.EQ.1) THEN
              sums(3)=y1
              sums(4)=y2
              sums(5)=y1
              sums(6)=y2
              sums(12)=prob
              sums(15)=prob
          ELSE
* Calculate the corrent minimum and maximum values and store them on the
array "sums"
              sums(3)=DMIN1(sums(3),y1)
              sums(4)=DMIN1(sums(4),y2)
              sums(5)=DMAX1(sums(5),y1)
              sums(6)=DMAX1(sums(6),y2)
              sums(12)=DMIN1(sums(12),prob)
              sums(15)=DMAX1(sums(15),prob)
          END IF
*
*  THE END OF THE NESTED LOOP
        END DO
*
       pfe=sums(9)/n0
       pf_ll=pfe-1.96d0*DSQRT(pfe*(1.d0-pfe)/n0)
       pf_ul=pfe+1.96d0*DSQRT(pfe*(1.d0-pfe)/n0)
***    WRITE(5,'(/)')
*      WRITE(5,'(A30,G12.5,"[",2G12.5,"]")')
*    &
       WRITE(5,'(A4,G12.5,"[",2G12.5,"]",F9.2," % ",
```

```
     &            A8,E11.3,A8,E11.3)')
     &                  'Pfe:',
     &                    pfe,pf_ll,pf_ul,(pf_ul-pf_ll)*100/pfe,
     &                 ' Mean r:',sums(1)/n0, ' Mean e:',sums(2)/n0
        WRITE(14,'(I5,2D12.5)') i, pfe
*
*  THE END OF THE OUTER LOOP
       END DO
*
       STOP
       END

       SUBROUTINE ACCIDENTMODEL(outrepnum,inrepnum,seed,pi,
     &                    z_dens,z_moist,z_emis,
     &                     treq,ntau,t,temp,oxygen,char_depth_final)
*
      INCLUDE 'constants.for'
* current number of repetition of the outer loop
* current number of repetition of the inner loop
      INTEGER outrepnum,inrepnum
* the "seed"
      INTEGER seed
*  vectors of parameters of MODELMAX models (OUTMODELMAX
parameters for each model)
      REAL*8 pi(MODELMAX,OUTMODELMAX)
* Stochastic input variables into the carring model: wood density and moisture,
beam surface emmisivity
      REAL*8 z_dens,z_moist,z_emis
       REAL*8 z(MODELMAX,OUTMODELMAX)
* The required time to failure
      REAL*8  treq
* Arays and variables used to model the charring of timber
       INTEGER ntau
      REAL*8
t(TIMESTEPMAX),temp(TIMESTEPMAX),oxygen(TIMESTEPMAX)
      REAL*8  fksi02(TIMESTEPMAX),psi(TIMESTEPMAX)
       REAL*8
hflux(TIMESTEPMAX),chrate(TIMESTEPMAX),cdepth(TIMESTEPMAX)

* The final value of the char depth
       REAL*8  char_depth_final
```

```
* The duration of the time step
     REAL*8  deltat
      REAL*8  f, te, td, qtd, tempe
* AUXILIARLY VARIABLES
      REAL*8  sumdepth
      INTEGER it
*
     REAL*8 GRAND
*
* Compute the duration of the time step
      deltat=treq/(ntau-1)
      sumdepth=0.d0
*
* Loop for the computation of the char depth
      DO it=1,ntau
* Compute the factor expressing dependance of charring on oxygen concentration
       fksi02(it)=pi(1,3)+(1.d0-pi(1,3))*
    &                       (oxygen(it)/oxygen(1))**0.737d0
* Compute the value of the "psi(t)" function
        psi(it)=pi(2,1)*DEXP(t(it)/pi(2,2))
* Compute the heat flux
        hflux(it)=
    &       pi(3,1)*(1d-3)*(temp(it)-temp(1))+
    &       5.67d-11*0.75d0*((temp(it)+273.d0)**4-(temp(1)+273.d0)**4)
      IF(hflux(it).LT.0.d0) THEN
        CONTINUE
      END IF
* Compute the charring rate in m/s
        chrate(it)=
    &       fksi02(it)*psi(it)*pi(4,4)*hflux(it)**pi(4,5)/
    &           ((z_dens+pi(4,1))*(pi(4,2)+pi(4,3)*z_moist))
* Compute the charring rate in mm/min
        chrate(it)=chrate(it)*6.d4
* Compute the charr depth in mm/min
        sumdepth=sumdepth+chrate(it)*deltat
        cdepth(it)=sumdepth
*       WRITE(OUTUNIT9,'(F15.3,2E15.3)') t(it),temp(it),oxygen(it)
*       WRITE(OUTUNIT10,'(I7,E15.7,7E15.7)')
*     &      it,t(it),oxygen(it),fksi02(it),psi(it),temp(it),hflux(it),
*     &      chrate(it),cdepth(it)
```

```
      END DO
      char_depth_final=cdepth(it-1)
*
      RETURN
      END
*
* OUTMAX  = maximal munber of repetitions of the outer loop
* INTMAX  = maximal munber of repetitions of the inner loop
* CHARMAX = maximal munber of repetitions of the inner loop
     INTEGER OUTMAX, INMAX, CHARMAX, STRINGMAX
*
     PARAMETER ( OUTMAX      = 1000 )
     PARAMETER ( INMAX       = 5000 )
     PARAMETER ( CHARMAX     = 5   )
     PARAMETER ( STRINGMAX   = 30  )
     PARAMETER ( TIMESTEPMAX  = 6000)
     PARAMETER ( MAXFIRES     = 100)
*
* MODELMAX = maximal number of models used to accident modelling
* OUTMODELMAX = maximal number of outputs of each model
*
     PARAMETER ( MODELMAX     = 10  )
     PARAMETER ( INPMODELMAX  = 20  )
     PARAMETER ( PARMODELMAX  = 20  )
     PARAMETER ( OUTMODELMAX  = 20  )
* STATMAX = maximal number of descriptive statistics
     PARAMETER ( STATMAX      = 6   )
* CATEGORMAX = maximal number of categories
     PARAMETER ( CATEGORMAX   = 50  )
* CATTOTALMAX = the maximal total number of categories =
CATEGORMAX^CHARMAX 125000
     PARAMETER ( CATTOTALMAX  = 125000 )
* NLEVELMAX = the maximal number of levels for sorting samples of
frequncies
      PARAMETER ( NLEVELMAX   = 6   )
*
* UNIT specifiers for output of simulated values of actions characteristics
     PARAMETER (  INUNIT1     =  5  )
     PARAMETER (  INUNIT2     =  6  )
     PARAMETER ( OUTUNIT0     = 50  )
     PARAMETER ( OUTUNIT1     = 51  )
```

```
      PARAMETER ( OUTUNIT2     = 52  )
      PARAMETER ( OUTUNIT3     = 53  )
      PARAMETER ( OUTUNIT4     = 54  )
      PARAMETER ( OUTUNIT5     = 55  )
* UNIT specifier for trasferring descriptive statisitics of simulated values
* of characteristics
      PARAMETER ( OUTUNIT6     = 56  )
* UNIT specifier for transferring  the results of categorisation of the samples
* of simulated  values of characteristics created during the simulation
      PARAMETER ( OUTUNIT7     = 57  )
* UNIT specifier for transferring results of clustering of samples of
* frequencies of simulated values of characteristics
      PARAMETER ( OUTUNIT8     = 58  )
* UNIT specifier for output files containing clusters of samples of frequencies
      PARAMETER ( OUTUNIT9     = 59  )
      PARAMETER ( OUTUNIT10    = 60  )
      PARAMETER ( OUTUNIT11    = 61  )
      PARAMETER ( OUTUNIT12    = 62  )
      PARAMETER ( OUTUNIT13    = 63  )
      PARAMETER ( OUTUNIT14    = 64  )
      PARAMETER ( OUTUNIT15    = 65  )
*
      PARAMETER ( OUTUNIT16    = 66  )
      PARAMETER ( OUTUNIT17    = 67  )
      PARAMETER ( OUTUNIT18    = 68  )
      PARAMETER ( OUTUNIT19    = 69  )
      PARAMETER ( OUTUNIT20    = 70  )
      PARAMETER ( OUTUNIT21    = 71  )
      PARAMETER ( OUTUNIT22    = 72  )
* UNIT specifier for output file containing estimates of annual probability
* of occurence of accidental action
      PARAMETER ( OUTUNIT23    = 73  )
* statistica files
*     PARAMETER ( OUTUNIT20    = 73  )
*     PARAMETER ( OUTUNIT21    = 74  )
*
      REAL*8 FUNCTION GAMA1(seed,aphpa,beta)
*
*  ATTENTION ! The parameter 'alpha' must be an integer
*  Algorithm G-1 from R.Y.Rubinstein, Sec. 3.6.2
*
```

```
      INTEGER seed,alpha
      REAL*8 beta,x,v,URAND
*
      x=.0D0
      DO i=1,alpha
       v=-DLOG(URAND(seed))
       x=x+v
      END DO
      GAMA1=x/beta
*
      RETURN
      END

       SUBROUTINE PARAMETERS(seed,istoch,pi)
*
      INCLUDE 'constants.for'
* The "seed"
      INTEGER seed
* A variable to decide whether the stochastisation of
* model parameters should be carried out
      INTEGER istoch
* Vectors of parameters of MODELMAX models (OUTMODELMAX
parameters for each model)
      REAL*8 pi(MODELMAX,OUTMODELMAX)
* functions used in the subroutine
      REAL*8 GRAND,URAND,LGRAND1,GAMMA1,BETA2
*
      IF(istoch.EQ.1) THEN
* parameter related to the annual probability of the initiating event
        p0=0.d0
* parameters of the first model
* tau-02 parameter (minutes) in the oxygen concentration formula
        pi(1,1)=4.d0
* delta-ksi parameter in the oxygen concentration formula
        pi(1,2)=GRAND(seed, 0.2255d0,5.5d0)
* ksi parameter
        pi(1,3)=0.50d0+0.15d0*URAND(seed)
*
* parameters of the second model (the function psi of t)
* psi zero
        pi(2,1)=
```

```
   &      1.35d0+1.15d0*URAND(seed)+
   &      1.35d0+1.15d0*URAND(seed)
* tau
      pi(2,2)=
   &      45.d0+10.d0*URAND(seed)+
   &      45.d0+10.d0*URAND(seed)
* parameters used for calculaiton of heat flux
* heat transfer coefficient
      pi(3,1)=
   &      5.5d0+2.d0*URAND(seed)+
   &      5.5d0+2.d0*URAND(seed)
* remaining parameters
* parameter ro
      pi(4,1)=GRAND(seed,93.d0,465.d0)
* parameter A
      pi(4,2)=505.d0+590.d0*URAND(seed)
* parameter B
      pi(4,3)=2430.d0+120.d0*URAND(seed)
* parameter teta
      pi(4,4)=
   &      0.6013d0+0.0922d0*URAND(seed)+
   &      0.6013d0+0.0922d0*URAND(seed)
* the power of the heat flux
      pi(4,5)=GRAND(seed,0.04d0,0.5d0)
*
    ELSE IF(istoch.EQ.0) THEN
* parameter related to the annual probability of the initiating event
      p0=0.d0
      pi(1,1)=4.d0
      pi(1,2)=5.5d0
      pi(1,3)=0.575d0
      pi(1,4)=21.d0
      pi(2,1)=3.85d0
      pi(2,2)=100.d0
      pi(3,1)=13.0d0
      pi(4,1)=465.d0
      pi(4,2)=800.d0
      pi(4,3)=2490.d0
      pi(4,4)=1.2065d0
      pi(4,5)=0.5d0
    ELSE
```

```
        WRITE(*,'(A70,I2,A1)') ' The program terminated due to '//
     &          'incorrectly specified  "istoch" value:',istoch
        STOP
       END IF
*
* ********** THE END OF THE CASE STUDY SPECIFIC PART OF THE
SUBROUTINE **********
*
      RETURN
      END

      SUBROUTINE READDATA(seed,n0,nr,treq,ntau,nfire)
*
     INCLUDE 'constants.for'
* the "seed"
     INTEGER seed
* number of repetitions of outer loop
* number of repetitions of inner loop
     INTEGER n0, nr
* The required time to failure
     REAL*8  treq
* The number of time steps in the fire history
      INTEGER ntau
* The number of fire under consideration
      INTEGER nfire
*
* read the flag value controlling the deepness of computations
* read the seed value, the number of repetitoins of the outer and inner loop,
* the flag value for a stochastisation of the seed value
     READ(1,'(I20)') seed
     READ(1,'(I10)') nr
     READ(1,'(I10)') n0
     READ(1,'(D10.3)') treq
     READ(1,'(I10)') ntau
     READ(1,'(I10)') nfire
*
      RETURN
      END

     REAL FUNCTION URAND(IY)
     INTEGER IY
```

```
*
*  URAND IS A UNIFORM RANDOM NUMBER GENERATOR BASED ON
THEORY
*  AND SUGGESTIONS GIVEN BY KNUTH (1969). THE INTEGER
*  IY SHOULD BE INITIALIZED TO AN ARBITRARY INTEGER RPIOR
TO THE
*  FIRST CALL TO URAND.  THE CALLING PROGRAM SHOULD NOT
ALTER THE
*  VALUE OF IY BETWEEN SUBSEQUENT CALLS TO URAND.
VALUES OF URAND
*  WILL BE RETURNED IN THE INTERVAL (0,1).
*
   INTEGER IA, IC, ITWO, M2, M, MIC
   real*8  HALFM
   real*8 S
   DATA M2/ 0/, ITWO/ 2/
*
*  IN FIRST ENTRY, COMPUTE MACHINE INTEGER WORD LENGTH
   IF( M2 .EQ. 0 )THEN
    M = 1
10      IF( M .GT. M2 )THEN
     M2 = M
     M = ITWO * M2
     GOTO 10
    END IF
    HALFM = M2
*
*  COMPUTE MULTIPLIER AND INCREMENT FOR LINEAR
CONGRUENTIAL METHOD
    IA = 8 * INT(HALFM * ATAN(1.D0) / 8.D0) +5
    IC = 2 * INT(HALFM * (0.5D0 - SQRT(3.D0) / 6.D0)) + 1
    MIC = (M2 - IC) + M2
*
*  S IS THE SCALE FACTOR FOR CONVERTING TO FLOATING POINT
    S = 0.5 / HALFM
   END IF
*
*  COMPUTE NEXT RANDOM NUMBER
   IY = IY * IA
*
```

```
*   THE FOLLOWING STATEMENT IS FOR COMPUTERS WHICH DO
NOT ALLOW
*  INTEGER OVERFLOW ON ADDITION
     IF( IY .GT. MIC ) IY = (IY - M2) - M2
*
     IY = IY + IC
*
*  THE FOLLOWING STATEMENT IS FOR COMPUTERS WHERE THE
*   WORD LENGTH FOR ADDITION IS GREATER THAN FOR
MULTIPLICATION
     IF( IY / 2 .GT. M2 ) IY = (IY - M2) - M2
*
*   THE FOLLOWING STATEMENT IS FOR COMPUTERS WHERE
INTEGER OVERFLOW
*  AFFECTS THE SIGN BIT
     IF( IY .LT. 0 ) IY = (IY + M2) + M2
     URAND = FLOAT(IY) * S
     RETURN
     END
```

Annex D. Computer code “Barrier”

```
      PROGRAM Barrier main
*
* Flags for writing
     INTEGER iflag,iflag_write
* Variable related to the resistance and aciton effect
      REAL*8  Rm,p_max,t_d,xt,p_a,alpha,p_r
      REAL*8  safety_margin_1, safety_margin_2, safety_margin_3
* Variables related to steel strength, stength enhancement, elasticity miodulus of
steel
     REAL*8  sig_y,sig_y_dyn, gamma, e_s, period
* Variables related to geometry of profiled steel
     REAL*8  x_span,x_span_mean,xe,l,z,t,h,l3,s
* Ductility ratio
     REAL*8  mu
* Variables of corecting factors
     REAL*8  ff,fc,fk,fl
* Deflections: elastic capacity, dynamic defl., plastic capacity
     REAL*8 yc_el,ye_dyn,yc_pl
* Variables related to MC simulation
```

```
      INTEGER NMC, NEPST, seed, ibinar
      INTEGER ibinar_1,ibinar_2,ibinar_3,sumibinar_1,sumibinar_2,
     &       sumibinar_3
      REAL*8  Pf_1,Pf_2,Pf_3
* Declarations of functions
      REAL*8  GRAND, LGRAND1
*
      OPEN(5,FILE='Barrier_data.txt',STATUS='OLD')
      OPEN(6,FILE='Barrier_results.txt',STATUS='OLD')
      OPEN(7,FILE='Barrier_MC.txt',STATUS='OLD')
      OPEN(8,FILE='Barrier_epistemic.txt',STATUS='OLD')
*
      READ(5,'(I20)') NMC
      READ(5,'(I20)') NEPST
      READ(5,'(I20)') seed
* Read the value of the incident overpressure in kPa
      READ(5,'(E20.5)') p_max
* Read the value of the incident posotive in ms
      READ(5,'(E20.5)') t_d
      READ(5,'(E20.5)') alpha
      READ(5,'(E20.5)') p_a
* Read the mean value of the profiled sheet span
      READ(5,'(E20.5)') x_span_mean
      READ(5,'(I20)')   iflag
      READ(5,'(I20)')   iflag_write
* Switch the incident overpressure from kPa to MPa
     p_max=p_max*1.d-3
* Switch the incident psotive duration from ms to s
     t_d=t_d*1.d-3
*
      IF(iflag.EQ.1) WRITE(6,'(15A)') 'RESULTS'
*
* Write information on MC simulation
      WRITE(7,'(/)')
      WRITE(7,'(25A)') 'Results of MC simulation'
      WRITE(7,'(/)')
      WRITE(7,'(25A\)') 'Number of MC trials: '
      WRITE(7,'(I10)')   NMC
      WRITE(7,'(25A)') 'The seed: '
      WRITE(7,'(I10)') seed
      WRITE(7,'(/)')
```

```
*
*****************************************************************
***
* THE OUTER LOOP OF MC SIMULATION                              *
*****************************************************************
***
*
*     WRITE(8,'(10A15)') 'gamma','ff','fc','fk'
      DO iepst=1,NEPST
*
        sumibinar_1=0
        sumibinar_2=0
* Generate values of parameters which are uncertain in the epistemic sense
* Steel enhancement factor
        gamma=1.d0+0.12d0*BETA2(seed,1,9)
* Corecting factors for pressure capacity
       fc=BETA2(seed,70,1)
       ff=1.d0-0.2d0*BETA2(seed,2,4)
        fk=1.d0-0.3d0*BETA2(seed,6,6)
*       WRITE(8,'(10E15.7)') gamma,ff,fc,fk
*****************************************************************
***
* THE NESTED LOOP OF MC SIMULATION                              *
*****************************************************************
***
*      GOTO 1000
        DO imc=1,NMC
* Generate values of basic variables which are uncertain in the aleatory sense
* Sample the value of static steel strength
         sig_y=LGRAND1(seed, 51.d0, 460.d0)
* Sample the value of elastic modulus of steel
         e_s=GRAND(seed,12000.d0,200000.d0)
* Sample the value of the span
         x_span=GRAND(seed,x_span_mean*0.005d0,x_span_mean)
*
            CALL ACTION(iflag,p_max,t_d,alpha,p_a,p_r)
            CALL RESISTANCE(iflag,p_max,t_d,p_r,x_span,sig_y,gamma,
   &                 sig_y_dyn,e_s,ff,fc,fk,mu,Rm,yc_el,yc_pl,
   &                 ye_dyn,ye_dyn_el,period)
*
* Calculation of safety margin
```

```
          safety_margin_1=Rm/p_r
          safety_margin_2=yc_el-ye_dyn_el
          safety_margin_3=yc_pl-ye_dyn
          ibinar_1=0
          ibinar_2=0
          ibinar_3=0
          IF(safety_margin_1.LE.1.d0) ibinar_1=1
          IF(safety_margin_2.LE.0.d0) ibinar_2=1
            IF(safety_margin_3.LE.0.d0) ibinar_3=1
           sumibinar_1=sumibinar_1+ibinar_1
           sumibinar_2=sumibinar_2+ibinar_2
           sumibinar_3=sumibinar_3+ibinar_3
*         IF((NMC .GT.1) .AND. (NMC.LE. 1000)) THEN
*           WRITE(7,'(I10,4E15.7)') imc, Rm*1000, yc_pl, p_max
*         END IF

        END DO
*
        Pf_1=sumibinar_1*1.d0/NMC
        Pf_2=sumibinar_2*1.d0/NMC
       Pf_3=sumibinar_3*1.d0/NMC

       IF(iepst.EQ.1) WRITE(7,'(5A15)') 'p_max','t_d','Pf1','Pf2','Pf3'
        IF(NMC.GT.1) THEN
*         WRITE(7,'(/)')
         WRITE(7,'(5E15.7)') p_max,t_d,Pf_1,Pf_2,Pf_3
*         WRITE(7,'(/)')
        END IF
        IF(iflag_write.EQ.1) THEN
* Write values of pressure capacity and plastic dynamic deflection
*         WRITE(7,'(A42,2E12.4)') ' ', Rm*1000
*        WRITE(6,'(3E15.7)') Mr,bendingmoment/1000,safety_margin
        END IF
* THE END OF THE NESTED MC LOOP
*
1000  CONTINUE
      END DO
*
* THE END OF THE OUTER MC LOOP
*
      STOP
```

```
      END

       SUBROUTINE ACTION(iflag,p_max,t_d,alpha,p_a,p_r)
*
* Flag for writing
      INTEGER iflag
* Variables related to action effect
       REAL*8  p_max,t_d,i_i,alpha,p_a,mach,mach_r,p_r
       REAL*8  cr
*      Mach number:
      mach=DSQRT((25.d0/7)*(p_max**2/((p_max+p_a)*(p_max+7.d0*p_a))))
*      Reflected Mach number:
       mach_r=mach*DSIN(alpha*3.14159d0/180)
*      Reflected overpressure p_r[MPA]:
      cr=((7.d0*mach_r**2-1)*(7*mach**2-1)-36)/(42*(mach**2-1))
      p_r=cr*(p_max-p_a)+p_a
       ratio=p_r/p_max
*
       RETURN
       END

      REAL*8 FUNCTION BETA2(seed,alpha,beta)
*
*  ATTENTION ! The parameters 'alpha' and 'beta' must be integers
*  The function 'GAMA1' is applied
*  Algorithm Be-1 from R.Y.Rubinstein, Sec. 3.6.3
*
      INTEGER seed,alpha,beta
      REAL*8  y1,y2
      REAL*8 GAMA1
*
      y1=GAMA1(seed,alpha,1.D0)
      y2=GAMA1(seed,beta,1.D0)
*
      BETA2=y1/(y1+y2)
*
      RETURN
      END

      REAL*8 FUNCTION GAMA1(seed,alpha,beta)
*
```

```
*  ATTENTION ! The parameter 'alpha' must be an integer
*  Algorithm G-1 from R.Y.Rubinstein, Sec. 3.6.2
*
   INTEGER seed,alpha
   REAL*8 beta,x,v,URAND
*
   x=.0D0
   DO i=1,alpha
    v=-DLOG(URAND(seed))
    x=x+v
   END DO
   GAMA1=x/beta
*
   RETURN
   END

   REAL FUNCTION GRAND*8(IY, STDEV, AMEAN)
*  A ROUTINE TO GENERATE PSEUDO RANDOM NUMBERS FROM A
GAUSSIAN
*  DISTRIBUTION WITH A MEAN OF AMEAN AND A STANDARD
DEVIATION OF
*  STDEV AS SPECIFIED BY USER.
*  THE ROUTINE MAKES USE OF THE FACT THAT THE DISTRIBUTION
OF
*  SAMPLE MEANS IS GAUSSIAN. SAMPLE MEANS OF 12 UNIFORMLY
*  DISTRIBUTED PSEUDO RANDOM NUMBERS ARE TAKEN AND
TRANSFORMED
*  TO THE DESIRED DISTRIBUTION.
*  THE REPRODUCABILITY OF THE VALUES RETURNED WILL BE
AFFECTED BY
*  ANY USE THAT IS MADE OF URAND IN THE INTERIM.
*  THIS ROUTINE REFERENCES URAND TO GENERATE THE
UNIFORMLY
*  DISTRIBUTED RANDOM NUMBERS
*
   real*8 STDEV, AMEAN
   GRAND = 0.0D0
   DO 10 I = 1,12
    GRAND = GRAND + URAND(IY)
10  CONTINUE
   GRAND = (GRAND - 6.0D0 ) * STDEV + AMEAN
```

```
      RETURN
      END

      REAL FUNCTION LGRAND1*8(IY, STDEV, AMEAN)
*  A ROUTINE TO GENERATE PSEUDO RANDOM NUMBERS FROM A LOG-NORMAL
*  DISTRIBUTION WITH A MEAN OF AMEAN AND A STANDARD DEVIATION OF
*  STDEV AS SPECIFIED BY USER.
*  DISTRIBUTED PSEUDO RANDOM NUMBERS OF STANDARD NORMAL DISTRIBUTION
*  ARE TAKEN AND TRANSFORMED  TO THE DESIRED DISTRIBUTION.
*  THE REPRODUCABILITY OF THE VALUES RETURNED WILL BE AFFECTED BY
*  ANY USE THAT IS MADE OF URAND IN THE INTERIM.
*  THIS ROUTINE REFERENCES GRAND TO GENERATE THE NORMAL DISTRIBUTED
*  RANDOM NUMBERS AND URAND TO GENERATE THE UNIFORMLY DISTRIBUTED
*  RANDOM NUMBERS
*
      real*8 STDEV, AMEAN
      real*8 M, SIGMA, Z
*
      SIGMA = DSQRT( DLOG( 1.D0 + (STDEV**2)/(AMEAN**2) ) )
      M = DLOG(AMEAN / DSQRT( 1.D0 + (STDEV**2)/(AMEAN**2) ))
      Z = DEXP( M + SIGMA * GRAND(IY,1.D0,0.D0) )
      LGRAND1 = Z
      RETURN
      END

       SUBROUTINE MU_FACTOR(Rm,p_r,t_d,period,mu)
*
* This subroutine computes approximate values of the ductility factor mu from Biggs (1964)
* using the graphical representation of this factor in Fig 2.24 (page 74)
* The pressure signal for which the values of mu are valid is triangular
*
* Variable related to the resistance and aciton effect
       REAL*8  Rm,p_r,t_d
```

```
* Variables related to steel strength, stength enhancement, elasticity miodulus of
steel
     REAL*8  period
* Ductility ratio
     REAL*8  mu
* Auxiliary variables
     REAL*8  ratio_t,ratio_r
*
      ratio_t=t_d/period
      ratio_r=Rm/p_r
*
*     CALL BEEPQQ(1000,1000)
*     READ (*,'(G12.4)') ratio_t
* Ratio 0.9
      IF(ratio_r.LT.1.0d0) THEN
        mu=2.7541d0*DEXP(0.2069d0*ratio_t)
      END IF
* Ratio 1.0
      IF(ratio_r.GT.1.0d0.AND.ratio_r.LE.1.1d0) THEN
       mu=2.1167+0.6189*ratio_t-0.0083*ratio_t**2
      END IF
* Ratio 1.2
      IF(ratio_r.GT.1.1d0.AND.ratio_r.LE.1.35d0) THEN
        mu=1.16+0.5442*ratio_t-0.0288*ratio_t**2
     END IF
* Ratio 1.5
      IF(ratio_r.GT.1.35d0.AND.ratio_r.LE.1.75d0) THEN
        mu=0.8077+0.2144*ratio_t-0.0111*ratio_t**2
      END IF
* Ratio 2.0
     IF(ratio_r.GT.1.75d0) THEN
        mu=1.0
        IF(ratio_t.LE.7.d0)
    &          mu=0.7253d0+0.0705d0*ratio_t-0.0044d0*ratio_t**2
     END IF
*     CALL BEEPQQ(100,100)
*     WRITE(*,'(A20,F12.4)') ' Rm/pr = ', ratio_r
*     WRITE(*,'(A20,D12.4)') ' mu factor = ',mu
*     STOP
*
      RETURN
```

```
      END

      SUBROUTINE
RESISTANCE(iflag,p_max,t_d,p_r,x,sig_y,gamma,sig_y_dyn,
     &                      e_s,ff,fc,fk,mu,Rm,yc_el,yc_pl,ye_dyn,
     &                      ye_dyn_el,period)
*
* Flag for writing
     INTEGER iflag
* The resistance of steel barrier and action effect
      REAL*8  Rm,p_max,t_d,p_r
* Variables related to material properties of steel
     REAL*8  sig_y,sig_y_dyn,e_s
* Enhancement factor for steel strength
     REAL*8  gamma
* Variables related to geometry of profiled steel
     REAL*8  x,xe,l,t,h,l1,l3,s
* Ductility ratio
     REAL*8  mu
* Variables of corecting factors
     REAL*8  ff,fc,fk,fl
* Ssection modulus, 2nd moment of inertia
     REAL*8 w,w_el,iner,iner_el
* Spring stiffness, weight, natural period, natural circular frequency
     REAL*8 k,weight,period,omega
* Dynamic load factor and time where DLF reaches maximum
     REAL*8 dlf_max,t_dlf
* Deflections
     REAL*8 yc_el,yc_pl,ye_dyn,ye_dyn_el
*
      l=0.25d0
      t=0.0025d0
      h=0.045d0
      l1=0.0625d0
      l3=0.045d0
      s=0.0602d0
      fl=(1.6*fk)/(1+0.6*fk)
* Effective span:
     xe=x*fl
* Section modulus:
      w=t*h*(2.d0*l3+s)/2
```

```
      w_el=(t/(3.d0*h))*(l3*(t**2.d0)+s**3.d0+3.d0*(h**2.d0)*l3)
* Dynamic yield strength of steel
      sig_y_dyn=sig_y*gamma
* Second moment of inertia
     iner=w*(h/2.d0)
      iner_el=w_el*(h/2.d0)
* Resitance:
      Rm=(8.d0*sig_y_dyn*w)/(xe**2*l)*ff*fc
* Dynamic elastic deflection capacity:
     yc_el=(5.d0*sig_y_dyn*w*xe**2*ff*fc)/(48.d0*e_s*iner_el*fk)
* Spring stifness:
*      k=(48.d0*e_s*iner)/(5.d0*w*xe*ff*fc)
* Cross-section area of profiled sheet:
     weight=x*(2*l1+l3+2*s)*t*7850.d0
* Natural period and natural circular frequency
*      period=2.d0*3.14159d0*DSQRT(weight/k)
* Assuned value for S3 profiled sheets
      period=(-0.5d0*x+4.55d0)/1.d+3
     omega=2d0*3.14159d0/period
* Time to reach DLF maximum, and DLF(max):
      t_dlf=3.14159d0/omega
      dlf_max=2+DSIN(omega*t_dlf)/(omega*t_d)-
    &t_dlf/t_d
* Compute the dynamic deflection
     ye_dyn=(p_r*(xe**4.d0)/(384.d0*iner*e_s))*dlf_max
      ye_dyn_el=(p_r*(xe**4.d0)/(384.d0*iner_el*e_s))*dlf_max
* Compute the ductility factor
      CALL MU_FACTOR(Rm,p_r,t_d,period,mu)
*
      IF(iflag.EQ.1) THEN
        WRITE(6,'(/A30,D15.4)') 'Overpressure, p_max=', p_max
        WRITE(6,'(/A30,D15.4)') 'Positive phase duration, t_d=', t_d
        WRITE(6,'(/A30,D15.4)') 'Reflected overpressure, p_r=', p_r
        WRITE(6,'(/A30,D15.4)') 'Rm/p_r=', Rm/p_r
        WRITE(6,'(/A30,D15.4)') 't_d/T=', t_d/period
        WRITE(6,'(/A30,D15.4)') 'mu=', mu
     END IF
* Compute the dynamic plastic deflection capacity with the ductility factor "mu"
      yc_pl=yc_el*mu
*
      IF(iflag.EQ.1) THEN
```

```
      WRITE(6,'(/A30,D15.4)') 'The span x=',x
*     WRITE(6,'(/A30,D15.4)')'The static steel strength sigma y=',sig_y
*     WRITE(6,'(/A30,D15.4)')'The modulus of elasticity E=',e_s
*     WRITE(6,'(/A30,D15.4)') 'Section modulus w=', w
      WRITE(6,'(/A30,D15.4)') 'The resistance Rm=', Rm
      WRITE(6,'(/A30,D15.4)') 'Elastic defl. capacity yc_el=', yc_el
*      WRITE(6,'(/A30,D15.4)') 'Weight m=', weight
*     WRITE(6,'(/A30,D15.4)') 'Natural period T=', period
*     WRITE(6,'(/A30,D15.4)') 'Natural circular freq. omega=',omega
      WRITE(6,'(/A30,D15.4)') 'Dynamic load factor DLF(max)=', dlf_max
      WRITE(6,'(/A30,D15.4)') 'Dynamic defl. ye_dyn=', ye_dyn
      WRITE(6,'(/A30,D15.4)') 'Plastic defl. capacity yc_pl=', yc_pl
      END IF
*
* Parameters used for calculation of resisting moment
*
*
* Calculation of cross-sectional geometry (random and deterministic quantities)
*
* Calculation of material characteristics (random and deterministic quantities)
*
*
* DETERMINATION OF RESISTANCE MOMENT
*
      RETURN
      END

      REAL FUNCTION URAND(IY)
      INTEGER IY
*
*  URAND IS A UNIFORM RANDOM NUMBER GENERATOR BASED ON
THEORY
*  AND SUGGESTIONS GIVEN BY KNUTH (1969). THE INTEGER
*   IY SHOULD BE INITIALIZED TO AN ARBITRARY INTEGER RPIOR
TO THE
*   FIRST CALL TO URAND.  THE CALLING PROGRAM SHOULD NOT
ALTER THE
*    VALUE OF IY BETWEEN SUBSEQUENT CALLS TO URAND.
VALUES OF URAND
*  WILL BE RETURNED IN THE INTERVAL (0,1).
*
```

```
      INTEGER IA, IC, ITWO, M2, M, MIC
      real*8  HALFM
      real*8 S
      DATA M2/ 0/, ITWO/ 2/
*
*  IN FIRST ENTRY, COMPUTE MACHINE INTEGER WORD LENGTH
      IF( M2 .EQ. 0 )THEN
        M = 1
10        IF( M .GT. M2 )THEN
          M2 = M
          M = ITWO * M2
          GOTO 10
        END IF
        HALFM = M2
*
*     COMPUTE MULTIPLIER AND INCREMENT FOR LINEAR
CONGRUENTIAL METHOD
        IA = 8 * INT(HALFM * ATAN(1.D0) / 8.D0) +5
        IC = 2 * INT(HALFM * (0.5D0 - SQRT(3.D0) / 6.D0)) + 1
        MIC = (M2 - IC) + M2
*
*  S IS THE SCALE FACTOR FOR CONVERTING TO FLOATING POINT
        S = 0.5 / HALFM
      END IF
*
*  COMPUTE NEXT RANDOM NUMBER
      IY = IY * IA
*
*   THE FOLLOWING STATEMENT IS FOR COMPUTERS WHICH DO
NOT ALLOW
*  INTEGER OVERFLOW ON ADDITION
      IF( IY .GT. MIC ) IY = (IY - M2) - M2
*
      IY = IY + IC
*
*  THE FOLLOWING STATEMENT IS FOR COMPUTERS WHERE THE
*    WORD LENGTH FOR ADDITION IS GREATER THAN FOR
MULTIPLICATION
      IF( IY / 2 .GT. M2 ) IY = (IY - M2) - M2
*
```

```
*   THE FOLLOWING STATEMENT IS FOR COMPUTERS WHERE
INTEGER OVERFLOW
*  AFFECTS THE SIGN BIT
     IF( IY .LT. 0 ) IY = (IY + M2) + M2
     URAND = FLOAT(IY) * S
     RETURN
     END
```

Annex E. Computer code "Chimney"

```
      PROGRAM Chimney main
     USE DFPORT
* Arrays of basic variables
      REAL*8  x(100)
     REAL*8  x1(100),x2(100),x3(100),x4(100),x5(100)
     REAL*8  x6(100),x7(100),x8(100),x9(100),x10(100)
* Arrays with characteristics of basic variables
     REAL*8  mean1(100),mean2(100),mean3(100),mean4(100),mean5(100)
     REAL*8  mean6(100),mean7(100),mean8(100),mean9(100),mean10(100)
     REAL*8  stdv1(100),stdv2(100),stdv3(100),stdv4(100),stdv5(100)
     REAL*8  stdv6(100),stdv7(100),stdv8(100),stdv9(100),stdv10(100)
* Array for variables which are considered to be uncertain in epistemic sense
     REAL*8  psi(100)
* Parameters of the exponential model used to generate correlations between
* random variables related to different segments of chimney (Bottenbruch,
Pradlwarter, Schueller 1989)
     REAL*8  lambda(100)
*
     REAL*8  ro1(100,100),ro2(100,100),ro3(100,100)
     REAL*8  ro4(100,100),ro5(100,100)
     REAL*8  ro6(100,100),ro7(100,100),ro8(100,100)
     REAL*8  ro9(100,100),ro10(100,100)
     REAL*8  cov1(100,100),cov2(100,100),cov3(100,100)
     REAL*8  cov4(100,100),cov5(100,100)
     REAL*8  cov6(100,100),cov7(100,100),cov8(100,100)
     REAL*8  cov9(100,100),cov10(100,100)
     REAL*8  c1(100,100),c2(100,100),c3(100,100)
      REAL*8  c4(100,100),c5(100,100)
     REAL*8  c6(100,100),c7(100,100),c8(100,100)
     REAL*8  c9(100,100),c10(100,100)
*
```

```
      REAL*8  height,tair
* Parameters of the main reinforcement on the compression side ("1") and
tension side ("2")
* (diameters and spacing)
     INTEGER dscir1,dscir2,spscir1,spscir2
* The number and diameter of bars used for the additional reinforcement on one
side of the chimney section
     INTEGER n_bars, dsadd
* Reinforcement distributed along the wall of chimney
* Reinforcement added at the edges of opening
      REAL*8  Ascir,Asadd
* Information related to discretisation of chimney
      REAL*8  alt(100),malt(100), ralt(100)
      INTEGER nsegm
      REAL*8  lamda
      REAL*8  Vrc(100),Vf(100),Wrc(100), Wf(100), W(100)
* Wind characteristics for different exposure categories
      INTEGER exposure_category
      REAL*8  z0, D0, walpha, wdelta
      REAL*8  wind10
* Chimney section for which the bending moment and resistance is computes
      INTEGER n_segm_current
* Initial data used for accounting for the dynamic effect of the wind action
      REAL*8 k_correct(14,2), m_correct(6,2)
      REAL*8 k_coef(100),m_coef(100)
      REAL*8 beta(100)
* Variable for the computation time retrieval
     REAL*8 elapsed_time
* Flag for writing
     INTEGER iflag
*
      REAL*8  totalweight
* The resistance and aciton effect of the chimney section with holes
      REAL*8  Mr,bendingmoment,safety_margin
* Parameters used for determination of additional moment
      REAL*8  moment_add

* Variables related to MC simulation
      INTEGER NMC, NEPST, seed, ibinar
      REAL*8 Pf,sumibinar
* Declarations of functions
```

```
      REAL*8 GRAND, LGRAND
*
      OPEN(5,FILE='Chimney_data.txt',STATUS='OLD',FORM='FORMATTE
D')
      OPEN(6,FILE='Chimney_results.txt',STATUS='OLD',FORM='FORMATT
ED')
      OPEN(7,FILE='Chimney_MC.txt',STATUS='OLD',FORM='FORMATTE
D')
*
* Files containing information on generation of values of external diameter, X1
      OPEN(10,FILE='X1_Means_and_stds.txt',STATUS='OLD')
      OPEN(11,FILE='X1_Corrs.txt',STATUS='OLD',FORM='FORMATTED')
      OPEN(12,FILE='X1_Covars.txt',STATUS='OLD',FORM='FORMATTED')
      OPEN(13,FILE='X1_Matrix_c.txt',STATUS='OLD',FORM='FORMATTE
D')
      OPEN(14,FILE='X1_Ext_diameter.txt',STATUS='OLD',FORM='FORMA
TTED')
* Files containing information on generation of values of wall thickness, X2
      OPEN(15,FILE='X2_Means_and_stds.txt',STATUS='OLD')
      OPEN(16,FILE='X2_Corrs.txt',STATUS='OLD',FORM='FORMATTED')
      OPEN(17,FILE='X2_Covars.txt',STATUS='OLD',FORM='FORMATTED')
      OPEN(18,FILE='X2_Matrix_c.txt',STATUS='OLD',FORM='FORMATTE
D')
      OPEN(19,FILE='X2_Int_diameter.txt',STATUS='OLD',FORM='FORMAT
TED')
* Files containing information on generation of values of concrete density, X3
      OPEN(20,FILE='X3_Means_and_stds.txt',STATUS='OLD')
      OPEN(21,FILE='X3_Corrs.txt',STATUS='OLD',FORM='FORMATTED')
      OPEN(22,FILE='X3_Covars.txt',STATUS='OLD',FORM='FORMATTED')
      OPEN(23,FILE='X3_Matrix_c.txt',STATUS='OLD',FORM='FORMATTE
D')
      OPEN(24,FILE='X3_Concr_density.txt',STATUS='OLD',FORM='FORMA
TTED')
* Files containing information on generation of values of brick lining density,
X4
      OPEN(25,FILE='X4_Means_and_stds.txt',STATUS='OLD')
      OPEN(26,FILE='X4_Corrs.txt',STATUS='OLD',FORM='FORMATTED')
      OPEN(27,FILE='X4_Covars.txt',STATUS='OLD',FORM='FORMATTED')
      OPEN(28,FILE='X4_Matrix_c.txt',STATUS='OLD',FORM='FORMATTE
D')
```

```
      OPEN(29,FILE='X4_Lining_density.txt',STATUS='OLD')
* Files containing information on generation of values of brick lining thickness,
X5
      OPEN(30,FILE='X5_Means_and_stds.txt',STATUS='OLD')
      OPEN(31,FILE='X5_Corrs.txt',STATUS='OLD',FORM='FORMATTED')
      OPEN(32,FILE='X5_Covars.txt',STATUS='OLD',FORM='FORMATTED')
      OPEN(33,FILE='X5_Matrix_c.txt',STATUS='OLD',FORM='FORMATTE
D')
      OPEN(34,FILE='X5_Lining_thickness.txt',STATUS='OLD')
* Files containing information on generation of values of concrete
secant_modulus, X6
      OPEN(35,FILE='X6_Means_and_stds.txt',STATUS='OLD')
      OPEN(36,FILE='X6_Corrs.txt',STATUS='OLD',FORM='FORMATTED')
      OPEN(37,FILE='X6_Covars.txt',STATUS='OLD',FORM='FORMATTED')
      OPEN(38,FILE='X6_Matrix_c.txt',STATUS='OLD',FORM='FORMATTE
D')
      OPEN(39,FILE='X6_Secant_modulus.txt',STATUS='OLD')
* File containing information values of segment weight, W
     OPEN(40,FILE='W_Segment weight.txt',STATUS='OLD')
* File containing information on values of bending moment
     OPEN(41,FILE='M_bending moment.txt',STATUS='OLD')
* File containing information on values of additional bending moment
     OPEN(42,FILE='M_additional bending moment.txt',STATUS='OLD')

*
      READ(5,'(I20)') NMC
      READ(5,'(I20)') NEPST
      READ(5,*)
      READ(5,'(I20)') seed
      READ(5,'(E20.5)') height
      READ(5,'(I20)') nsegm
      READ(5,'(I20)') exposure_category
      READ(5,'(E20.5)') wind10
      READ(5,'(I20)') n_segm_current
      READ(5,*)
     READ(5,'(I20)') dscir1
      READ(5,'(I20)') dscir2
      READ(5,'(I20)') spscir1
      READ(5,'(I20)') spscir2
     READ(5,'(I20)') n_bars
     READ(5,'(I20)') dsadd
```

```
      READ(5,*)
      READ(5,'(I20)')  iflag
      READ(5,'(I20)')  iflag_write

* Read correcting coefficients
      DO i=1,14
        READ(5,'(2F6.2)') k_correct(i,1),k_correct(i,2)
      END DO
      DO i=1,6
        READ(5,'(2F6.2)') m_correct(i,1),m_correct(i,2)
      END DO
*
*
      WRITE(6,'(15A)') 'RESULTS'
      WRITE(6,'(/30A)') 'Height of chimney:'
      WRITE(6,'(G12.5)') height
      WRITE(6,'(/30A)') 'Number of segments:'
      WRITE(6,'(I5)')    nsegm
      WRITE(6,'(/)')
* Write information on MC simulation
      WRITE(7,'(/)')
      WRITE(7,'(25A)') 'Results of MC simulation'
      WRITE(7,'(/)')
      WRITE(7,'(25A\)') 'Number of MC trials: '
      WRITE(7,'(I10)') NMC
      WRITE(7,'(25A)') 'The seed: '
      WRITE(7,'(I10)') seed
      WRITE(7,'(/)')
*
      IF     (exposure_category.EQ.1) THEN
        psi(2)=80.d0
        psi(3)=0.0251d0
        walpha=0.33333333d0
        wdelta=457.d0
      ELSE IF(exposure_category.EQ.2) THEN
        psi(2)=20.d0
        psi(3)=0.0105d0
        walpha=0.22222222d0
        wdelta=366.d0
      ELSE IF(exposure_category.EQ.3) THEN
        psi(2)=3.5d0
```

```
         psi(3)=0.0050d0
         walpha=0.14285714d0
         wdelta=274.d0
        ELSE IF(exposure_category.EQ.4) THEN
         psi(2)=0.7d0
         psi(3)=0.0030d0
         walpha=0.1d0
         wdelta=213.d0
        END IF
*
* Assign air gap thickness between concrete and ceramics surfaces
       tair=0.05d0
*
        WRITE(6,'(A20,I5/)') 'Exposure categoty', exposure_category
*
*************************************************************
***
* THE OUTER LOOP OF MC SIMULATION                            *
*************************************************************
        DO iepst=1,NEPST
* Generate values of parameters which are uncertain in the epistemic sense
         lambda(1)=GRAND(seed,0.01d0,0.2d0)
         lambda(2)=GRAND(seed,0.01d0,0.2d0)
         lambda(3)=GRAND(seed,0.01d0,0.2d0)
         lambda(4)=GRAND(seed,0.01d0,0.2d0)
         lambda(5)=GRAND(seed,0.01d0,0.2d0)
         lambda(6)=GRAND(seed,0.01d0,0.2d0)
* Sample parameters used for determination of horizontal wind forces
*   The dynamic coefficient
         psi(1)=1.2d0+1.2d0*BETA2(seed,10,1)
*   Aerodynamic coefficient
         psi(6)=GRAND(seed,0.063d0,0.7d0)
*    Variable used to express epistemic uncertainty in the coefficient of wind
pressure fluctution
         psi(7)=GRAND(seed,0.05d0,1.0d0)
* Sample values of the uncertain paramaters of the logarithmic law used to
compute wind speed profile
*   Roughness length
*        psi(2)=(0.01d0+0.04d0*BETA2(seed,50,30))*10
         psi(2)=GRAND(seed,0.015d0,0.3d0)
*   Surface drag coefficient
```

```
*      psi(3)=(0.003d0+0.003*BETA2(seed,50,25))*2.5
       psi(3)=GRAND(seed,0.0006875d0,0.01375d0)
*   Karman's coefficient
       psi(4)=GRAND(seed,0.01d0,0.4d0)
*   Zero-plane displacemen
       psi(5)=0.0d0
*   Compute chimney geometry according to the discretisation scheme
       CALL discretisation(iflag,height,lamda,nsegm,alt,
    &       malt,k_correct,m_correct,14,6,2,k_coef,m_coef)
* Include the code lines containing preparatory computations for generating
values
* from a multinormal distribution of the vectors of basic variables X1 to X6
       INCLUDE 'Multinormal_prepare.for'
*
*******************************************************************
***
* THE NESTED LOOP OF MC SIMULATION                               *
*******************************************************************
***
*
       elapsed_time = TIMEF( )
       sumibinar=0.d0
       DO imc=1,NMC
* Generate values of the external diameter in individual segments
        CALL MULTIGRAND(seed,mean1,c1,nsegm,x1)
        IF(iflag_write.EQ.1)
    &       WRITE(14,'(100D15.7)') (x1(i), i=1,nsegm)
* Generate values of the wall thickness in individual segments
        CALL MULTIGRAND(seed,mean2,c2,nsegm,x2)
        IF(iflag_write.EQ.1)
    &       WRITE(19,'(100D15.7)') (x2(i), i=1,nsegm)
* Generate values of concrete density in individual segments
        CALL MULTIGRAND(seed,mean3,c3,nsegm,x3)
        IF(iflag_write.EQ.1)
    &       WRITE(24,'(100D15.7)') (x3(i), i=1,nsegm)
* Generate values of brick lining density in individual segments
        CALL MULTIGRAND(seed,mean4,c4,nsegm,x4)
        IF(iflag_write.EQ.1)
    &       WRITE(29,'(100D15.7)') (x4(i), i=1,nsegm)
* Generate values of brick lining density in individual segments
        CALL MULTIGRAND(seed,mean5,c5,nsegm,x5)
```

```
      IF(iflag_write.EQ.1)
   &        WRITE(34,'(100D15.7)') (x5(i), i=1,nsegm)
* Generate values of secant modulus of concrete
      CALL MULTIGRAND(seed,mean6,c6,nsegm,x6)
      IF(iflag_write.EQ.1)
   &        WRITE(39,'(100D15.7)') (x6(i), i=1,nsegm)
*
* Sample the strength of concrete
      x(2)=GRAND(seed,5.d0,33.d0)
* Sample the values of strength of main and additional reinforcement
      x(3)=LGRAND(seed,54.d0,490.d0)
      x(4)=LGRAND(seed,54.d0,490.d0)
* Sample the values of external and internal concrete covers
      x(5)=GRAND(seed,0.00125d0,0.05d0)
      x(6)=GRAND(seed,0.00125d0,0.05d0)
*

      CALL chimney_weight(iflag,x,x1,x2,x3,x4,x5,
   &       100,nsegm,lamda,height,tair,alt,
   &        Vrc,Vf,Wrc,Wf,W,totalweight)
      IF(iflag_write.EQ.1)
   &        WRITE(40,'(100E15.7)') (W(i), i=1,nsegm)
*
      CALL actioneffect(iflag,height,wind10,walpha,
   &                              k_coef,m_coef,beta,
   &                  nsegm,lamda,alt,malt,
   &                     x1,x2,x6,
   &                  x,psi,100,
   &                              Wrc,Wf,totalweight,
   &                  n_segm_current,moment_add,bendingmoment)
      IF(iflag_write.EQ.1)
   &        WRITE(41,'(E15.7)') (bendingmoment)
      IF(iflag_write.EQ.1)
   &        WRITE(42,'(D15.7)') (moment_add)
*
      CALL resistance(iflag,nsegm,lamda,alt,malt,
   &                 x,x1,x2,x3,x4,x5,x6,x7,x8,x9,x10,100,
   &   dscir1*1.d-3,dscir2*1.d-3,spscir1*1.d-3,spscir2*1.d-3,
   &                 n_bars, dsadd,Ascir,Asadd,
   &                 W,n_segm_current,totalweight,Mr)
*
```

```
*          WRITE(6,'(/A20,D15.7)') ' Bendingmoment, kNm=', bendingmoment
*
* Calculation of safety margin
        safety_margin=Mr-bendingmoment/1000
        ibinar=0
        IF(safety_margin.LE.0.d0) ibinar=1
         sumibinar=sumibinar+ibinar
        IF((NMC .GT.1) .AND. (NMC.LE. 1000)) THEN
         WRITE(7,'(I10,4E15.7)') imc, totalweight,
   &             Mr,bendingmoment/1000,Mr-bendingmoment/1000
        END IF

       END DO
* THE END OF THE NESTED MC LOOP
*
       elapsed_time = TIMEF( )
       Pf=sumibinar/NMC
       IF(NMC .GT.1) THEN
*        WRITE(7,'(/)')
*         WRITE(7,'(A3,E15.7,A7,G9.2)') 'Pf=',Pf,'Time=',elapsed_time/60
        WRITE(7,'(2E15.7)') Pf,elapsed_time/60
*        WRITE(7,'(/)')
       END IF
       WRITE(*,'(A20,G9.2,A5)') 'Computation time =',elapsed_time/60,
   &                       ' min'
       IF(iflag.EQ.1) THEN
       WRITE(6,'(/A42/)') 'Resistance (MNm) and bending moment (MNm)'
        WRITE(6,'(3E15.7)') Mr,bendingmoment/1000,safety_margin
       END IF
*
      END DO
*
* THE END OF THE OUTER MC LOOP
*
*
*
      STOP
     END

      SUBROUTINE actioneffect(iflag,height,wind10,walpha,
   &                                   k_coef,m_coef,beta,
```

```
     &                  nsegm,lamda,alt,malt,
     &                     x1,x2,x6,
     &                  x,psi,nx,
     &                                 Wrc,Wf,totalweight,n_segm_current,
     &                  moment_add,bendingmoment)
* Arrays of basic variables
      REAL*8  x1(nsegm),x2(nsegm),x6(nsegm)
* Flag for writing
      INTEGER iflag
       REAL*8  wind10
* Parameters for the laws of gradient wind speed
       REAL*8  z0, D0, karman, zh
       INTEGER nx,nsegm
      REAL*8  x(nx)
* Array for variables which are considered to be uncertain in epistemic sense
      REAL*8  psi(nx)
* Arrays for altitudes of segments
       REAL*8  alt(nsegm)
       REAL*8  malt(nsegm),lamda
* Initial data used for accounting for the dynamic effect of the wind action
       REAL*8  k_coef(nsegm),m_coef(nsegm),beta(nsegm),ksi
       REAL*8  sum, sumf(nsegm), fmax
       REAL*8  kcoef, mcoef
       REAL*8  windspeed_pl(nsegm)
       REAL*8  windspeed_ln(nsegm)
       REAL*8  pressure(nsegm),force(nsegm)
* Chimney segment with the lower section for which the bending moment and
resistance are computed
       INTEGER n_segm_current
       REAL*8  bendingmoment
       REAL*8  upper_alt, lower_alt
*
* Parameters used for determination of additional moment
      REAL*8 totalweight,height,
     &       inertia(nsegm),EI(nsegm),factor1(nsegm),
     &       factor2(nsegm),factor3(nsegm), defl(nsegm),
     &       Wrc(100), Wf(100), moment_add
*
       IF(iflag.EQ.1)
     &      WRITE(6,'(/20A)') 'ACTION EFFECT IS:'
* Assign values of the epistemic variables
```

```
      ksi=psi(1)
      z0=psi(2)
      D0=psi(3)
      karman=psi(4)
      zh=psi(5)
      Cd=psi(6)
*
      IF(iflag.EQ.1)
     &      WRITE(6,'(/A11,11A15)') 'Segment#','Altitude','Mid.altitude',
     &      'Segment radius','Wind speed_pl','Wind speed_ln',
     &      'k coef.','m coef.','Pressure','Force','beta'
*
      sum=0.d0
***************************
**    DO i=1,nsegm
      DO i=1,n_segm_current
* Windspeed according to the logarithmic law
*       z0=0.05d0
*       D0=0.006d0
        windspeed_ln(i)=DSQRT(D0)*DLOG((malt(i)-zh)/z0)*wind10/karman
*       WRITE(*,'(I3, 3F10.4)') i, z0,D0,windspeed_ln(i)
* Calculate by interpolation the correcting factors
        beta(i)=1.d0+ksi*m_coef(i)*psi(7)
        pressure(i)=windspeed_ln(i)**2/1628
       force(i)=lamda*Cd*pressure(i)*x1(i)*beta(i)*k_coef(i)
* Contribution of segment to total moment due to the wind pressure (kNm)
        sum=sum+force(i)*malt(i)
      IF(iflag.EQ.1)
     &  WRITE(6,'(I11,10D15.5)') i, alt(i), malt(i),
     &       windspeed_pl(i),windspeed_ln(i),
     &     k_coef(i), m_coef(i), pressure(i),force(i),beta(i)
      END DO
      bendingmoment=sum

* CALCULATION OF ADDITIONAL MOMENT

      DO i=1,nsegm
* Moment of inertia of the segment (i):
     inertia(i)=3.1416d0*(x1(i)**4.d0-
     &         (x1(i)-x2(i))**4.d0)/64.d0
* Stiffness of the segment (i):
```

```
      EI(i)=x6(i)*inertia(i)
      END DO
* Calculation of chimney top deflection fmax
      fmax=0.d0
       DO i=1,nsegm
      upper_alt=malt(i)
        IF(i.EQ.nsegm) THEN
          lower_alt=0.d0
       ELSE
          lower_alt=malt(i+1)
        END IF
       factor1(i)=(upper_alt**3-lower_alt**3)/3
      factor2(i)=(upper_alt**2-lower_alt**2)/2
       factor3(i)= upper_alt   -lower_alt
       sumf(i)=(force(i)/1000*factor1(i)-force(i)/1000*(height+
     &       malt(i))*factor2(i)+force(i)/1000*malt(i)*height*
     &       factor3(i))/EI(i)
       fmax=fmax+sumf(i)*1000
      END DO
*
* Calculation of horizontal deflections of the segments and additional moment
      moment_add=0.d0
**      DO i=1,nsegm
      DO i=1,n_segm_current
       defl(i)=fmax*(malt(i)/height)**2
        moment_add=moment_add+(Wrc(i)+Wf(i))*defl(i)
      END DO
* Calculation of total bending moment
      bendingmoment=bendingmoment+moment_add
       RETURN
       END

       SUBROUTINE chimney_weight(iflag,x,x1,x2,x3,x4,x5,nx,
     &                  nsegm,lamda,height,tair,alt,
     &                   Vrc,Vf,Wrc,Wf,W,totalweight)
* Flag for writing
      INTEGER iflag
       REAL*8  x(nx)
* Arrays of basic variables
      REAL*8  x1(nsegm),x2(nsegm),x3(nsegm),x4(nsegm),x5(nsegm)
       INTEGER nx,nsegm
```

```
      REAL*8  lamda,height,tair,alt(nsegm)
* Radii and volumes of chimney segments
      REAL*8  rad(nsegm),rad1(nsegm),rad2(nsegm),rad3(nsegm)
      REAL*8  Vext(nsegm),Vint(nsegm)
      REAL*8  Vextf(nsegm),Vintf(nsegm)
      REAL*8  Vrc(nsegm), Vf(nsegm)
      REAL*8  Wrc(nsegm), Wf(nsegm), W(nsegm)
*
      REAL*8  ratio1,ratio2,rtop,rtop1,rtop2,rtop3
      REAL*8  totalweight
***
      IF(iflag.EQ.1) THEN
      WRITE(6,'(/40A)') 'FEATURES OF THE SEGMENTS ARE:'
      WRITE(6,'(A8,11A15)') 'Segment#','Altitude','Radexttop',
     &    'Radextbot','Radinttop','Radintbot','Vext','Vint',
     &     'Vrc','Vextf','Vintf','Vf'
      END IF
* Calculate weight of individual segments and total weight of chimney in kN
      totalweight=0.d0
      DO i=1,nsegm
* Computation of RC volume of each segment of the chimney
       Vrc(i)=3.1459d0*lamda*(x1(i)**2-(x1(i)-2*x2(i))**2)/4
* Computation of RC weight of each segment of the chimney
       Wrc(i)=Vrc(i)*x3(i)
* Computation of lining volume of each segment of the chimney
       Vf(i)=3.1459d0*lamda*((x1(i)-x2(i)-tair)**2-
     &  (x1(i)-x2(i)-tair-x5(i))**2)/4
* Computation of lining weight of each segment of the chimney
       Wf(i)=Vf(i)*x4(i)
* Computation of total weight of each segment
     IF ((nsegm-i)*lamda .GE. 12.5d0) THEN
        W(i)=Wrc(i)+Wf(i)
      ELSE
       W(i)=Wrc(i)
      END IF
*Calculation of axial force
        totalweight=totalweight+W(i)
      END DO

      IF(iflag.EQ.1)
     & WRITE(6,'(/A30,D15.7/)')
```

```
   &      'Total weight of chimney (kN):', totalweight
*
      RETURN
      END

      SUBROUTINE discretisation(iflag,height,lamda,nsegm,
   &      alt,malt,k_correct,m_correct,n14,n5,n2,k_coef,m_coef)
* Flag for writing
     INTEGER iflag
* Information related to discretisation of chimney
      REAL*8  height, alt(nsegm),malt(nsegm)
      REAL*8  windspeed(nsegm)
      INTEGER nx,nsegm
      REAL*8  lamda, rtop
* Initial data used for accounting for the dynamic effect of the wind action
      REAL*8 k_correct(n14,n2), m_correct(n5,n2)
      REAL*8 k_coef(nsegm),m_coef(nsegm)
      INTEGER i1,i2

*
      IF(iflag.EQ.1)
   &      WRITE(6,'(/40A)') 'ALTITUDES & RADII OF THE SEGMENTS
ARE:'
      IF(iflag.EQ.1)
   &      WRITE(6,'(A9,13A15)') 'Segment#','Altitude','Mid.altitude',
   &      'k coef.','m coef.'
* Calculate altitudes and radii of segments
      lamda=height/nsegm
      DO i=1,nsegm
* Altitude of the top and middle of each segment
        alt(i)=height-lamda*(i-1)
        malt(i)=alt(i)-0.5d0*lamda
      END DO
*
      DO i=1,nsegm
        j1=1
        DO WHILE(k_correct(j1,2).LT.malt(i))
         j1=j1+1
        END DO
        IF(j1.NE.1) THEN
         k_coef(i)=((k_correct(j1,1)-k_correct(j1-1,1))/
```

```
     &            (k_correct(j1,2)-k_correct(j1-1,2)))*
     &          (malt(i)-k_correct(j1-1,2)) + k_correct(j1-1,1)
        ELSE
         k_coef(i)=((k_correct(j1,1)-0.0d0)/
     &            (k_correct(j1,2)-0.0d0))*
     &          (malt(i)-0.0d0) + 0.0d0
        END IF
        j2=1
        DO WHILE(m_correct(j2,2).LT.malt(i))
         j2=j2+1
        END DO
        IF(j2.NE.1) THEN
         m_coef(i)=((m_correct(j2,1)-m_correct(j2-1,1))/
     &            (m_correct(j2,2)-m_correct(j2-1,2)))*
     &          (malt(i)-m_correct(j2-1,2)) + m_correct(j2-1,1)
        ELSE
         m_coef(i)=0.35d0
        END IF
       END DO
*
      DO i=1,nsegm
      IF(iflag.EQ.1)
     &       WRITE(6,'(I8,7D15.5)') i, alt(i), malt(i),
     &                k_coef(i),m_coef(i)
      END DO

*
      RETURN
      END

* Prepare values for the external diameter generation, X1
        CALL CHIMNEY_GEOMETRY(malt,nsegm,height,
     &                 23.46d0,9.6d0,0.005d0,0.005d0,mean1,stdv1)
        IF(iflag_write.EQ.1)
     &       WRITE(10,'(I4,F9.3,2D15.7)')
     &        (i, malt(i),mean1(i),stdv1(i), i=1,nsegm)
*  Create matrices of correlation coefficients and covariances
        CALL CHIMNEY_COVS(malt,lambda(1),stdv1,ro1,cov1,nsegm)
        IF(iflag_write.EQ.1) THEN
          DO i=1,nsegm
            WRITE(11,'(100F7.4)')  (ro1(i,j), j=1,nsegm)
```

```
          WRITE(12,'(100E12.5)') (cov1(i,j), j=1,nsegm)
        END DO
      END IF
*   Compute the matrix "c"
      CALL MATRIX_C(stdv1,ro1,cov1,c1,nsegm)
      IF(iflag_write.EQ.1) THEN
        DO i=1,nsegm
          WRITE(13,'(100E12.5)') (c1(i,j), j=1,nsegm)
        END DO
      END IF
*
* Prepare values for the wall thickness generation, X2
      CALL CHIMNEY_GEOMETRY(malt,nsegm,height,
   &                  0.7d0,0.2d0,0.03d0,0.02d0,mean2,stdv2)
      IF(iflag_write.EQ.1)
   &       WRITE(15,'(I4,F9.3,2D15.7)')
   &        (i, malt(i),mean2(i),stdv2(i), i=1,nsegm)
*   Create matrices of correlation coefficients and covariances
      CALL CHIMNEY_COVS(malt,lambda(2),stdv2,ro2,cov2,nsegm)
      IF(iflag_write.EQ.1) THEN
        DO i=1,nsegm
          WRITE(16,'(100F7.4)')  (ro2(i,j), j=1,nsegm)
          WRITE(17,'(100E12.5)') (cov2(i,j), j=1,nsegm)
        END DO
      END IF
*   Compute the matrix "c"
     CALL MATRIX_C(stdv2,ro2,cov2,c2,nsegm)
     IF(iflag_write.EQ.1) THEN
       DO i=1,nsegm
         WRITE(18,'(100E12.5)') (c2(i,j), j=1,nsegm)
       END DO
     END IF
*
* Prepare values for the concrete density generation, X3
     CALL CHIMNEY_GEOMETRY(malt,nsegm,height,
   &                  25.d0,25.d0,0.06d0,0.05d0,mean3,stdv3)
     IF(iflag_write.EQ.1)
   &      WRITE(20,'(I4,F9.3,2D15.7)')
   &       (i, malt(i),mean3(i),stdv3(i), i=1,nsegm)

*   Create matrices of correlation coefficients and covariances
```

```
      CALL CHIMNEY_COVS(malt,lambda(3),stdv3,ro3,cov3,nsegm)
      IF(iflag_write.EQ.1) THEN
        DO i=1,nsegm
          WRITE(21,'(100F7.4)')  (ro3(i,j), j=1,nsegm)
          WRITE(22,'(100E12.5)') (cov3(i,j), j=1,nsegm)
        END DO
      END IF
*   Compute the matrix "c"
      CALL MATRIX_C(stdv3,ro3,cov3,c3,nsegm)
      IF(iflag_write.EQ.1) THEN
        DO i=1,nsegm
          WRITE(23,'(100E12.5)') (c3(i,j), j=1,nsegm)
        END DO
      END IF
*
* Prepare values for the brick lining density generation, X4
      CALL CHIMNEY_GEOMETRY(malt,nsegm,height,
     &                 20.d0,20.d0,0.05d0,0.05d0,mean4,stdv4)
      IF(iflag_write.EQ.1)
     &      WRITE(25,'(I4,F9.3,2D15.7)')
     &       (i, malt(i),mean4(i),stdv4(i), i=1,nsegm)

*   Create matrices of correlation coefficients and covariances
      CALL CHIMNEY_COVS(malt,lambda(4),stdv4,ro4,cov4,nsegm)
      IF(iflag_write.EQ.1) THEN
        DO i=1,nsegm
          WRITE(26,'(100F7.4)')  (ro4(i,j), j=1,nsegm)
          WRITE(27,'(100E12.5)') (cov4(i,j), j=1,nsegm)
        END DO
      END IF
*   Compute the matrix "c"
      CALL MATRIX_C(stdv4,ro4,cov4,c4,nsegm)
      IF(iflag_write.EQ.1) THEN
        DO i=1,nsegm
          WRITE(28,'(100E12.5)') (c4(i,j), j=1,nsegm)
        END DO
      END IF
*
* Prepare values for the brick lining thickness generation, X5
      CALL CHIMNEY_GEOMETRY(malt,nsegm,height,
     &                 0.12d0,0.12d0,0.05d0,0.05d0,mean5,stdv5)
```

```
      IF(iflag_write.EQ.1)
     &      WRITE(30,'(I4,F9.3,2D15.7)')
     &        (i, malt(i),mean5(i),stdv5(i), i=1,nsegm)

*   Create matrices of correlation coefficients and covariances
      CALL CHIMNEY_COVS(malt,lambda(5),stdv5,ro5,cov5,nsegm)
      IF(iflag_write.EQ.1) THEN
        DO i=1,nsegm
          WRITE(31,'(100F7.4)')  (ro5(i,j), j=1,nsegm)
          WRITE(32,'(100E12.5)') (cov5(i,j), j=1,nsegm)
        END DO
      END IF
*   Compute the matrix "c"
      CALL MATRIX_C(stdv5,ro5,cov5,c5,nsegm)
      IF(iflag_write.EQ.1) THEN
        DO i=1,nsegm
          WRITE(33,'(100E12.5)') (c5(i,j), j=1,nsegm)
        END DO
      END IF
*
* Prepare values for the concrete secant_modulus, X6
      CALL CHIMNEY_GEOMETRY(malt,nsegm,height,
     &                 31.d3,31.d3,0.124d0,0.124d0,mean6,stdv6)
      IF(iflag_write.EQ.1)
     &      WRITE(35,'(I4,F9.3,2D15.7)')
     &        (i, malt(i),mean6(i),stdv6(i), i=1,nsegm)
*   Create matrices of correlation coefficients and covariances
      CALL CHIMNEY_COVS(malt,lambda(6),stdv6,ro6,cov6,nsegm)
      IF(iflag_write.EQ.1) THEN
        DO i=1,nsegm
          WRITE(36,'(100F7.4)')  (ro6(i,j), j=1,nsegm)
          WRITE(37,'(100E12.5)') (cov6(i,j), j=1,nsegm)
        END DO
      END IF
*   Compute the matrix "c"
      CALL MATRIX_C(stdv6,ro6,cov6,c6,nsegm)
      IF(iflag_write.EQ.1) THEN
        DO i=1,nsegm
          WRITE(38,'(100E12.5)') (c6(i,j), j=1,nsegm)
        END DO
      END IF
```

```
      SUBROUTINE resistance(iflag,nsegm,lamda,alt,malt,
     &                x,x1,x2,x3,x4,x5,x6,x7,x8,x9,x10,nx,
     &                dscir1,dscir2,spscir1, spscir2,
     &                n_bars, dsadd,Ascir,Asadd,
     &                W,n_segm_current,Nr,Mr)
*
* Flag for writing
     INTEGER iflag
* Total number of basic varables
      INTEGER nx
* Total number of segments
* Chimney segment with the lower section for which the bending moment and
resistance are computed
     INTEGER nsegm, n_segm_current
      REAL*8  x(nx)
     REAL*8  x1(nsegm),x2(nsegm),x3(nsegm),x4(nsegm),x5(nsegm)
     REAL*8  x6(nsegm),x7(nsegm),x8(nsegm),x9(nsegm),x10(nsegm)
* Arrays for altitudes of segments
      REAL*8  alt(nsegm)
      REAL*8  malt(nsegm)
* The number and diameter of bars used for the additional reinforcement on one
side of the chimney section
     INTEGER n_bars, dsadd
* Reinforcement distributed along the wall of chimney
* Reinforcement added at the edges of opening (one side (tension or
comopression))
      REAL*8  Ascir,Asadd
* Total weight of chimney in kN calculated in "segments" subroutine and
* segment height
      REAL*8 Nr,W(nsegm),lamda
*
      REAL*8 A,Atot,Aring1,Aring2, r1,r2, t1,t2,dcir,ds1,ds2
* Variables related to relative characteristic of the compression zone
      REAL*8 gamarc,gamaf
      REAL*8 alpha,beta1,beta2,betas
      REAL*8 fc,fy,fyc,fycir,fycirc
      REAL*8 numerator,denominator
* Variables exressing cross sectional dimensions and related coefficients
      REAL*8 rs1,rs2,yscir,zscir
      REAL*8 zc,zs1,zs2,alphas1,alphas2
```

```
*
* The resistance of the chimney seciton with wholes
      REAL*8 Mr
* Height and width of opening, wall thickness
     REAL*8 ho,wo,t
* Concrete density, external and internal diameter
      REAL*8 ro,d_ext,d_int
* Concrete cover and reinforcement parameters
* (diameters, spacing and strength)
     REAL*8 cc_ext,cc_int,dscir1,spscir1,dscir2
      REAL*8 spscir2,rs,rsc,rscir,rsccir
* Compression strength of concrete
     REAL*8 rb
* Parameter used for determination of resistance moment
     REAL*8 pibeta,p_m,pialpha,d_ascir1,d_ascir2,n_ascir1,n_ascir2
    &      ascir,ab,alpha_cir,ys_cir,z_b,zs_cir,zs,kalpha

      IF(iflag.EQ.1)
    &      WRITE(6,'(/20A)') 'RESISTANCE IS:'
*
* Parameters used for calculation of resisting moment
* TENTATIVE APPROXIMATE VALUES
* Height of opening, m
     ho=16.8d0
* Width of opening, m
     wo=8.03d0
* Avarage wall thickness within the technological opening, m
     t=0.7d0
* Concrete density, kN/m3
     ro=25.d0
*
* Determination of chimney weigth (reduced with openings), kN
* Substract the weights of the segments which are lower than the segment under
analysis
      DO i=1,nsegm
        IF(i.GT.n_segm_current)THEN
          IF(i.GE.93.AND.i.LE.99) THEN
               W(i)=W(i)-lamda*wo*t*2
           END IF
          Nr=Nr-W(i)
       ELSE
```

```
        END IF
      END DO
*
* Calculation of cross-sectional geometry (random and deterministic quantities)
*
* External diameter at the cross-section under consideration, m
     d_ext=x1(n_segm_current)
* Inertnal diameter at the cross-section under consideration, m
     d_int=d_ext-2*x2(n_segm_current)
* Angle of opening (pi*beta), rad
     pibeta=ATAN((wo/2)/(d_ext/2))
* Diameter at center line of wall (additional reinforcement), m
     d_m=(d_ext+d_int)/2
* Angle of placement of additional reinforcement (pi*alpha), rad
     pialpha=pibeta*2
* Deterministic concrete cover value used for the caluculation of reinforcement
area
     cc_ext=x(5)
     cc_int=x(6)
* External diameter of placement of main reinforcement (ascir1), m
     d_ascir1=d_ext-2*(cc_ext-dscir1/2)
* Internal diameter of placement of main reinforcement (ascir2), m
     d_ascir2=d_int+2*(cc_int+dscir2/2)
*

* Compute the area of the additional reinforcement on one side of the chimney
section
     Asadd=n_bars*(0.785398d0*(dsadd*1.d-3)**2)
*     WRITE(*,'(E15.7)') Asadd
*     STOP
*
* Calculation of material characteristics (random and deterministic quantities)
*
* Compression strength of concrete, MPa
     rb=x(2)
* Strengths of reinforcement
      rscir=x(3)
      rs=x(4)
* Notional strengths of steel in compression (a deterministic value)
      rsc=400.d0
      rsccir=400.d0
```

```
*
* DETERMINATION OF RESISTANCE MOMENT
* Number of bars of ascir1, pc.
     n_ascir1=3.1416*d_ascir1/spscir1
* Number of bars of ascir2, pc.
     n_ascir2=3.1416*d_ascir2/spscir2
* Total area of main reinforcement ascir, m2
     Ascir=3.1416*dscir1**2/4*n_ascir1+3.1416*dscir2**2/4*n_ascir2
* Area of concrete (reduced with openings), m2
     ab=3.1416*(d_ext**2/4-d_int**2/4)*(1-(pibeta/3.1416*180)*4/360)
* Determination of relative parameter of compression zone alpha_cir
     alpha_cir=((1-pibeta)*rscir*1000*ascir+rs*1000*asadd-rsc*asadd+
    &           pibeta*rb*1000*ab+pibeta*rscir*1000*ascir+
    &         Nr)/(rb*1000*ab+(rscir*1000+rsccir*1000)*ascir)
* Determination of parameters ys_cir, z_b, zs_cir, zs, zs2, k_alpha
     ys_cir=d_m/2*(SIN(1-alpha_cir)-SIN(pibeta))/(1-alpha_cir-pibeta)
      z_b=((d_ext/2+d_int/2)/2)*(SIN(alpha_cir)-SIN(pibeta))/
    &   (alpha_cir-pibeta)+ys_cir
      zs_cir=d_m/2*((SIN(alpha_cir)-SIN(pibeta))/(alpha_cir-
    &      pibeta))+ys_cir
      zs=d_m/2*(SIN(pialpha)-SIN(pibeta))/(pialpha-pibeta)+ys_cir
      kalpha=0.97*alpha_cir**2-1.13*alpha_cir+1.26d0
* Determination of resistance moment, MNm
     Mr=(rb*ab*z_b*(alpha_cir-pibeta)+rsccir*ascir*zs_cir*(alpha_cir-
    &   pibeta)+rs*asadd*zs+rsc*asadd*zs)*kalpha
*
     GOTO 1000
* Expore the section under analysis
       WRITE(6,'(A20,I4)') 'Section #', n_segm_current
       WRITE(6,'(A20,D15.7)') '     Ext diam =', x1(n_segm_current)
       WRITE(6,'(A20,D15.7)') 'Wall thickness=', x2(n_segm_current)
       WRITE(6,'(A20,D15.7)') 'Sgmt top altitude=', alt(n_segm_current)
       WRITE(6,'(A20,D15.7)') 'Sgmt mid altitude=',malt(n_segm_current)
       WRITE(6,'(A20,D15.7)') '     Int diam =', d_int
       WRITE(6,'(A20,D15.7)') 'Ext conc cover=', cc_ext
       WRITE(6,'(A20,D15.7)') 'Int conc cover=', cc_int
       WRITE(6,'(A20,D15.7)') 'Ext r cenntre diam=', d_ascir1
       WRITE(6,'(A20,D15.7)') 'Int r cenntre diam=', d_ascir2
       WRITE(6,'(A20,D15.7)') ' Concrete strength=', rb
       WRITE(6,'(A20,D15.7)') 'Main steel strength=', rscir
       WRITE(6,'(A20,D15.7)') 'Add. steel strength=', rs
```

```
      WRITE(6,'(A20,D15.7)') ' Chimney weight, kN=', Nr
      WRITE(6,'(A20,D15.7)') '    Resistance, kNm=', Mr
*
1000  RETURN
      END
```