

Příloha B – Ukázka zdrojového kódu jednoho programového systému včetně všech využívajících funkcí.

Volací skript - *volaciSoubor.m*

```
1  clc; close all; clear all; %Smaze dispej, promenne a zevre vsechna okna
2  resolution = 55; %Nastaveni velikosti jednoho binu (obr. px)
3  ## Vytvoreni zasob retezcu typu string, ze kterych se skladaji nazvy souboru
4  ## Nazvy vzorku
5  kontakTitle{1} = "PCB";
6  kontakTitle{2} = "MrizCtvrcc";
7  kontakTitle{3} = "Mosaiq";
8  ## Energie
9  kontakE{1} = "13";
10 kontakE{2} = "22";
11 kontakE{3} = "31";
12 ## Uhel rotace
13 kontakRot{1} = "10";
14 kontakRot{2} = "20";
15 kontakRot{3} = "30";
16 kontakRot{4} = "00";
17 ## Vnorene for-cykly kombinujici jednotlivé string retezce
18 for r = 4
19     for i = 1
20         for j = 3
21             ## Podminka osetrujici neexistující kombinace
22                 if (i==1 && j==1)
23                     elseif (i==3 && j==1)
24                         else
25                             data = load([kontakTitle{i},kontakE{j}, "MeVR", kontakRot{r}, ".elist"]);
26                             %Nacte data
27                             [dataX, dataY, dataE, dataA, dataH, dataR] = UpravData (data, resolution);
28                             %Vlastní funkce na upravu dat
29                             %Vytvoreni spekter
30                             [minA, maxE] = getHist2(dataA, dataE, kontakE{j}, [kontakTitle{i},
31 "Rot",kontakRot{r}]); %Vlastni funkce na nastaveni histogramu
32                             minH = 700; % Nastaveni minimalni hodnoty vysky klastru
33                             volejSpektrum(dataE, dataH, 0, kontakE{j}, [kontakTitle{i}, "Rot",
34 kontakRot{r}]); %Vlastni funkce na nastaveni spektra s prislusnou sirkou kanalu
35                             volejSpektrum(dataE, dataH, 2, kontakE{j}, [kontakTitle{i}, "Rot",
36 kontakRot{r}]);
37                             volejSpektrum(dataE, dataH, 4, kontakE{j}, [kontakTitle{i}, "Rot",
38 kontakRot{r}]);
39                             volejSpektrum(dataE, dataH, 10, kontakE{j}, [kontakTitle{i}, "Rot",
40 kontakRot{r}]);
41                             %%Filtrace dat
42                             AccSize =
43                             find((dataA>minA).*(dataE<maxE).*(dataH>minH).*(dataA<90).*(dataE>1000).*(dataH<1000));
44                             %Najde pouze clustery maji vetsi plochu nez minA px a s kulatosti od 0.8 do
45                             1.2
46                             ## preformatovani vyberu dat
47                             dataA = dataA(AccSize);
48                             dataE = dataE(AccSize);
49                             dataH = dataH(AccSize);
50                             dataX = dataX(AccSize);
51                             dataY = dataY(AccSize);
52                             dataR = dataR(AccSize);
53                             %Vytvoreni radiogramu
54                             ## getRadiograph(dataX, dataY, dataE, dataA, dataH, kontakE{j},
55 kontakTitle{i}, resolution(r), minA);
56                             ## Vytvori extrapolovane radiogramy pomoci vlastni funkce
57                             getRadiographI(dataX, dataY, dataE, dataA, dataH, kontakE{j}, [kontakTitle{i},
58 "Rot", kontakRot{r}], resolution, minA, maxE, minH);
59                         endif
60                     endfor
61                 endfor
62             endfor
63         endfor
64     endfor
65 endfor
```

Příloha B – Ukázka zdrojového kódu jednoho programového systému včetně všech využívaných funkcí.

Funkce - *UpravData.m*

```
24 ## Funkce upravena pro cteni dat z EvListu
25 ## promenna dpx odpovida velikosti jednoho binu v um
26 function [dataX, dataY, dataE, dataA, dataH, dataR] = UpravData (data0, dpx)
27     ## data0 = data;
28     ## osetreni podminky jesti existuje promena "delky pixelu"
29     if exist("dpx")
30     else
31         dpx = 55;
32     endif
33     z1 = dpx/1000; %Vytvori prevracenou hodnotu zvetseni
34     dataX = data0(:,3)/z1; %ulozi souradnice X
35     dataY = data0(:,4)/z1; %ulozi souradnice Y
36     dataE = data0(:,5); %ulozi hodnoty E [keV]
37     dataA = data0(:,8); %ulozi hodnoty A [px]
38     dataH = data0(:,9); %ulozi hodnoty H [keV]
39     dataR = data0(:,11); %ulozi kulatost tj. pomer sirky a delky klastru
40 endfunction
```

Funkce - *GetHist2.m*

```
25 function [minA, maxE] = getHist2 (dataA, dataE, title1, title0)%Vytvori a ulozi
histogram
26 ## Zobrazeni histogramu
27     hf = figure;
28     hist(dataA,min(dataA):max(dataA)); %spektrum velikosti
29     xlabel('Area size [px]');
30     ylabel('Counts [-]');
31     title(['Area size histogram: ',title0, " ", title1, " MeV"]);
32     set(gca, "yscale", "log");
33     print(hf, ["histA", title0, title1, ".png"], "-dpng"); %ulozi histogram jako obrazek
34     close(hf);
35 ## Hledani prvni rostouci hodnoty v histogramu
36     [n, x] = hist(dataA,min(dataA):max(dataA));
37     ## difN = n(2:length(n))-n(1:(length(n)-1));
38     ## rostN = find(difN>0);
39     rostN = find(n==min(n(3:find(n==max(n(3:length(n)))))));
40     minA = rostN(1);
41     xE = floor(min(dataE)):ceil(max(dataE));
42     [nnE, xxE] = hist(dataE,xE);
43     iNE = find(nnE>0.5*(max(nnE(300:length(nnE)))));%Najde indexy, kde je cetnost
energie vetsi nez 1/2 maximalni hodnoty
44     maxE = iNE(length(iNE));
45 endfunction
```

Příloha B – Ukázka zdrojového kódu jednoho programového systému včetně všech využívaných funkcí.

Funkce - *volejSpektrum.m*

```
25 function volejSpektrum (dataE, dataH, par, title1, title0)
26     if par>0
27         [xE nE] = spectrumvector(max(dataE),par); %Parametry pro vytvoreni spektra s
relacni sirkou kanalu
28         [xH nH] = spectrumvector(max(dataH),par);
29         titlePar = [num2str(par), "% relative"];
30     else
31         xE = floor(min(dataE)):ceil(max(dataE)); %Parametry pro vytvoreni spektra s
pevnou sirkou kanalu
32         xH = floor(min(dataH)):ceil(max(dataH));
33         titlePar = "fixed";
34     endif
35
36     %Energeticke spektrum
37     hfE = figure;
38     [nnE, xxE] = hist(dataE,xE);
39     plot(xxE, nnE); %spektrum energie
40     xlabel("Energy [keV]");
41     ylabel("Counts [-]");
42     title(["Energy spectrum with ", titlePar, " channel width: ",title0, " ", title1, "MeV"]);
43     set(gca, "yscale", "log");
44     print(hfE, ["EnergySpectrumKeV", title0, title1, num2str(par), ".png"], "-dpng");
45     close(hfE);
46
47     %% Vyskove spektrum
48     %spektrum s pevnou šířkou kanálu
49     hfH = figure;
50     [nnH, xxH] = hist(dataH,xH);
51     plot(xxH, nnH); %spektrum vysek
52     xlabel("Height [keV]");
53     ylabel("Counts [-]");
54     title(["Height spectrum with ", titlePar, " channel width: ",title0, " ", title1, "MeV"]);
55     set(gca, "yscale", "log");
56     print(hfH, ["HeightSpectrumKeV", title0, title1, num2str(par), ".png"], "-dpng");
57     close(hfH);
58
59 endfunction
```

Příloha B – Ukázka zdrojového kódu jednoho programového systému včetně všech využívajících funkcí.

Funkce - *getRadiograph1.m*

```
25 function getRadiograph1 (dataX, dataY, dataE, dataA, dataH, title1, title0, dpx, minA, maxE, minH)
26     dataX = round(dataX);
27     dataY = round(dataY);
28     dimenze = ceil(256*55/dpx);
29     obr1 = zeros(dimenze); %prazdna matice pro obr dle prumernych velikosti
30     obr2 = zeros(dimenze); %-|- dle poctu
31     obr3 = zeros(dimenze); %-|- dle poctu maximalni velikosti
32     obr4 = zeros(dimenze); %-|- dle prumerne energie
33     obr5 = zeros(dimenze); %-|- dle maximalni energie
34     obr6 = zeros(dimenze); %-|- dle maximalni hodnoty
35     obr7 = zeros(dimenze); %-|- dle prumeru maximalnich hodnot
36     data = [dataY, dataX, dataA, dataE, dataH];
37     [sD iD] = sortrows(data);%Srovná hodnoty
38     sX = sD(:,1);
39     sY = sD(:,2);
40     sX0 = zeros(length(sX)+1,1);
41     sY0 = zeros(length(sY)+1,1);
42     sX0(2:length(sX0)) = sX;
43     sY0(2:length(sY0)) = sY;
44     ##[counts, centers] = hist3 ([sX, sY],[dimenze, dimenze]);
45     ##imagesc (centers{1}, centers{2}, counts);
46     sS = sY+(sX-1)*dimenze;
47     rozdil = sX-sX0(1:length(sX))+sY-sY0(1:length(sY)); %najde shodné souřadnice
48     idxR = double(find(rozdil==0)); %vytvorí index shodných souřadnic, kde idxR(i) =>
p(i) = p(i-1) {p je bod}
49     idx1 = double(find(rozdil~=0)); %vytvorí index souřadnic, které jsou jiné než předchozí, kde
idx(i) => p(i) ~= p(i-1) {p je bod}
50     idx10 = zeros(length(idx1)+1,1);
51     idx10(2:length(idx10)) = idx1;
52     jdx = idx1-idx10(1:length(idx1)); %vypočte počet shodných pixelů, kde jdx(i) se rovná počtu
shodných pixelů před p(idx1(i)) tj. p(idx1(i)) = p(idx1(i)-jdx(-)+1):p(idx1(i))
53     obr2(sS(idx1))=jdx; %vytvorí obrazek countů
54     ## imagesc(obr2) %zobrazí radiogram podle countů
55     ## colormap(jet)
56     ## colorbar
57     ## xlabel('x - position [px]');
58     ## ylabel('y - position [px]');
59     MaticeJ = zeros(max(jdx),length(jdx),3); %vytvorí matici do které se uloží všechny hodnoty,
kde řádek odpovídá jedné souřadnici a sloupce jednotlivým clusterům
60     for i=1:length(jdx)
61         MaticeJ(1:jdx(i),i,1) = sD((idx1(i)-jdx(i)+1):idx1(i),3); %uloží hodnotu A
62         MaticeJ(1:jdx(i),i,2) = sD((idx1(i)-jdx(i)+1):idx1(i),4); %uloží hodnotu E
63         MaticeJ(1:jdx(i),i,3) = sD((idx1(i)-jdx(i)+1):idx1(i),5); %uloží hodnotu H
64     endfor
65     maxJ = zeros(size(MaticeJ,2),size(MaticeJ,3));
66     meanJ = zeros(size(MaticeJ,2),size(MaticeJ,3));
67     maxJ(:,1:3) = max(MaticeJ);
68     sumJ(:,1:3) = sum(MaticeJ);
69     idxJ = zeros(length(jdx),3);
70     for jm = 1:3
71         idxJ(:,jm) = (jdx);
72     endfor
73     meanJ = sumJ./jdx;
74
75
76     obr1(sS(idx1)) = meanJ(:,1); %uloží průměrné hodnoty velikosti clusteru na daných
souřadnicích
77     %obr2(i,j) = length(index); %uloží počet událostí v daném bodě
```

Příloha B – Ukázka zdrojového kódu jednoho programového systému včetně všech využívajících funkcí.

```
78 obr3(sS(idx1)) = maxJ(:,1); %ulozi maximalni hodnoty velikosti clusteru na danych
souradnicich
79 obr4(sS(idx1)) = meanJ(:,2); %ulozi prumerne hodnoty energie clusteru na danych
souradnicich
80 obr5(sS(idx1)) = maxJ(:,2); %ulozi maximalni hodnoty energie clusteru na danych
souradnicich
81 obr6(sS(idx1)) = maxJ(:,3); %ulozi maximalni hodnoty v danych bodech
82 obr7(sS(idx1)) = meanJ(:,3); %vypocte prumer maximalnich hodnot v ddanych bodech
83 ## Dopocet chybejicich hodnot
84 [m, n] = find(obr2==0); %najde, kde nejsou v obrazku zadne hodnoty
85 filtM1 = zeros(3); %vytvori interpolacni masky
86 filtM2 = zeros(3);
87 filtM3 = zeros(3);
88 filtM4 = zeros(3);
89 filtM5 = zeros(3);
90 filtM6 = zeros(3);
91 filtM7 = zeros(3);
92 Maska1 = zeros(dimenze+2); %vytvori masky obrazku s okrajema
93 Maska2 = zeros(dimenze+2);
94 Maska3 = zeros(dimenze+2);
95 Maska4 = zeros(dimenze+2);
96 Maska5 = zeros(dimenze+2);
97 Maska6 = zeros(dimenze+2);
98 Maska7 = zeros(dimenze+2);
99 Maska1(2:(dimenze+1),2:(dimenze+1)) = obr1; % Ulozi obraze do masky a vytvori se okraje
100 Maska2(2:(dimenze+1),2:(dimenze+1)) = obr2;
101 Maska3(2:(dimenze+1),2:(dimenze+1)) = obr3;
102 Maska4(2:(dimenze+1),2:(dimenze+1)) = obr4;
103 Maska5(2:(dimenze+1),2:(dimenze+1)) = obr5;
104 Maska6(2:(dimenze+1),2:(dimenze+1)) = obr6;
105 Maska7(2:(dimenze+1),2:(dimenze+1)) = obr7;
106 for k = 1:length(m)
107     filtM1 = Maska1(m(k):(m(k)+2),n(k):(n(k)+2)); %Ulozi hodnoty do interpolacni masky
108     filtM2 = Maska2(m(k):(m(k)+2),n(k):(n(k)+2));
109     filtM3 = Maska3(m(k):(m(k)+2),n(k):(n(k)+2));
110     filtM4 = Maska4(m(k):(m(k)+2),n(k):(n(k)+2));
111     filtM5 = Maska5(m(k):(m(k)+2),n(k):(n(k)+2));
112     filtM6 = Maska6(m(k):(m(k)+2),n(k):(n(k)+2));
113     filtM7 = Maska7(m(k):(m(k)+2),n(k):(n(k)+2));
114     pocetS = sum(sum(filtM2>0)); %vypocte pocet hodnot v interpolacnich maskach
115     if pocetS>0
116         obr1(m(k),n(k)) = sum(filtM1(:))/pocetS; %Ulozi prumer z interpolacni masky
do prazne hodnoty
117         obr2(m(k),n(k)) = sum(filtM2(:))/pocetS;
118         obr3(m(k),n(k)) = sum(filtM3(:))/pocetS;
119         obr4(m(k),n(k)) = sum(filtM4(:))/pocetS;
120         obr5(m(k),n(k)) = sum(filtM5(:))/pocetS;
121         obr6(m(k),n(k)) = sum(filtM6(:))/pocetS;
122         obr7(m(k),n(k)) = sum(filtM7(:))/pocetS;
123     endif
124 endfor
125
126
127
128 save("-mat", ["A+", num2str(minA), "E-", num2str(maxE), "H+", num2str(minH)] title0, title1,
"MeVfiguresInt.mat", "obr1", "obr2", "obr3", "obr4", "obr5", "obr6", "obr7");
129
130 ## Hledani parametru zobrazeni
131 sortA = sort(dataA);
132 sortE = sort(dataE);
```

Příloha B – Ukázka zdrojového kódu jednoho programového systému včetně všech využívaných funkcí.

```
133 sortH = sort(dataH);
134 c_minA = sortA(floor(length(sortA)/4));
135 c_maxA = sortA(ceil(3*length(sortA)/4));
136 c_minE = sortE(floor(length(sortE)/4));
137 c_maxE = sortE(ceil(3*length(sortE)/4));
138 c_minH = sortH(floor(length(sortH)/4));
139 c_maxH = sortH(ceil(3*length(sortH)/4));
140
141 h1 = figure;
142 imagesc(obr1)
143 colormap(jet)
144 colorbar
145 xlabel('x - position [px]');
146 ylabel('y - position [px]');
147 title(['Areas (mean value of bins): ',title0, ' ',title1, ' MeV']);
148 my_colormap=colormap;
149 %my_colormap = my_colormap(65-(1:64),:);
150 my_colormap(1,:) = [0 0 0];
151 %my_colormap(256,1) = 0;
152 colormap(my_colormap);
153 caxis ([c_minA c_maxA]);%
154 %caxis("auto");%[0 2100]);
155 hc1 = colorbar;
156 set(get(hc1,"label"),"string","Area size [px]","fontsize",22);
157 print(h1, ["A_mean_Int", title0, title1, "MeVRes", "A+", num2str(minA), "E-", num2str(maxE),
"H+", num2str(minH), "R", num2str(dpx), "um.png"], "-dpng");
158 %close(h1);
159
160 h2 = figure;
161 imagesc(obr2)
162 colormap(jet)
163 colorbar
164 xlabel('x - position [px]');
165 ylabel('y - position [px]');
166 title(['Counts: ',title0, ' ',title1, ' MeV']);
167 my_colormap=colormap;
168 %my_colormap = my_colormap(65-(1:64),:);
169 my_colormap(1,:) = [0 0 0];
170 %my_colormap(256,1) = 0;
171 colormap(my_colormap);
172 %caxis ([15 32]);%
173 %caxis("auto");%[0 2100]);
174 hc2 = colorbar;
175 set(get(hc2,"label"),"string","Counts [-]","fontsize",22);
176 print(h2, ["C_Int", title0, title1, "MeVRes", "A+", num2str(minA), "E-", num2str(maxE), "H+",
num2str(minH), "R", num2str(dpx), "um.png"], "-dpng");
177 %close(h2);
178
179 h3 = figure;
180 imagesc(obr3)
181 colormap(jet)
182 colorbar
183 xlabel('x - position [px]');
184 ylabel('y - position [px]');
185 title(['Areas (max. value of bins): ',title0, ' ',title1, ' MeV']);
186 my_colormap=colormap;
187 %my_colormap = my_colormap(65-(1:64),:);
188 my_colormap(1,:) = [0 0 0];
189 %my_colormap(256,1) = 0;
190 colormap(my_colormap);
```

Příloha B – Ukázka zdrojového kódu jednoho programového systému včetně všech využívaných funkcí.

```
191 caxis ([c_maxA max(sortA)]);%
192 %caxis("auto");%[0 2100]);
193 hc3 = colorbar;
194 set(get(hc3,"label"),"string","Area size [px]","fontsize",22);
195 print(h3, ["A_max_Int", title0, title1, "MeVRes", "A+", num2str(minA), "E-", num2str(maxE),
"H+", num2str(minH), "R", num2str(dpx), "um.png"], "-dpng");
196 %close(h3);
197
198 h4 = figure;
199 imagesc(obr4)
200 colormap(jet)
201 colorbar
202 xlabel('x - position [px]');
203 ylabel('y - position [px]');
204 title (["Energy (mean value of bins): ",title0, " ",title1, " MeV"]);
205 my_colormap=colormap;
206 %my_colormap = my_colormap(65-(1:64),:);
207 my_colormap(1,:) = [0 0 0];
208 %my_colormap(256,1) = 0;
209 colormap(my_colormap);
210 caxis ([c_minE c_maxE]);%
211 %caxis("auto");%[0 2100]);
212 hc4 = colorbar;
213 set(get(hc4,"label"),"string","Energy [keV]","fontsize",22);
214 print(h4, ["E_mean_Int", title0, title1, "MeVRes", "A+", num2str(minA), "E-", num2str(maxE),
"H+", num2str(minH), "R", num2str(dpx), "um.png"], "-dpng");
215 %close(h4);
216
217 h5 = figure;
218 imagesc(obr5)
219 colormap(jet)
220 colorbar
221 xlabel('x - position [px]');
222 ylabel('y - position [px]');
223 title (["Energy (max. value of bins): ",title0, " ",title1, " MeV"]);
224 my_colormap=colormap;
225 %my_colormap = my_colormap(65-(1:64),:);
226 my_colormap(1,:) = [0 0 0];
227 %my_colormap(256,1) = 0;
228 colormap(my_colormap);
229 caxis ([c_maxE max(sortE)]);%
230 %caxis("auto");%[0 2100]);
231 hc5 = colorbar;
232 set(get(hc5,"label"),"string","Energy [keV]","fontsize",22);
233 print(h5, ["E_max_Int", title0, title1, "MeVRes", "A+", num2str(minA), "E-", num2str(maxE),
"H+", num2str(minH), "R", num2str(dpx), "um.png"], "-dpng");
234 %close(h5);
235
236 h6 = figure;
237 imagesc(obr6)
238 colormap(jet)
239 colorbar
240 xlabel('x - position [px]');
241 ylabel('y - position [px]');
242 title (["Height (max. value of bins): ",title0, " ",title1, " MeV"]);
243 my_colormap=colormap;
244 %my_colormap = my_colormap(65-(1:64),:);
245 my_colormap(1,:) = [0 0 0];
246 %my_colormap(256,1) = 0;
247 colormap(my_colormap);
```

Příloha B – Ukázka zdrojového kódu jednoho programového systému včetně všech využívajících funkcí.

```
248   caxis ([c_maxH max(sortH)]);%
249   %caxis("auto");%[0 2100]);
250   hc6 = colorbar;
251   set(get(hc6,"label"),"string","Height [keV]","fontsize",22);
252   print(h6, ["H_max_Int", title0, title1, "MeVRes", "A+", num2str(minA), "E-", num2str(maxE),
"H+", num2str(minH), "R", num2str(dpx), "um.png"], "-dpng");
253   %close(h6);
254
255   h7 = figure;
256   imagesc(obr7)
257   colormap(jet)
258   colorbar
259   xlabel('x - position [px]');
260   ylabel('y - position [px]');
261   title (["Height (mean value of bins): ",title0, " ",title1, " MeV"]);
262   my_colormap=colormap;
263   %my_colormap = my_colormap(65-(1:64),:);
264   my_colormap(1,:) = [0 0 0];
265   %my_colormap(256,1) = 0;
266   colormap(my_colormap);
267   caxis ([c_minH c_maxH]);%
268   %caxis("auto");%[0 2100]);
269   hc7 = colorbar;
270   set(get(hc7,"label"),"string","Height [keV]","fontsize",22);
271   print(h7, ["H_mean_Int", title0, title1, "MeVRes", "A+", num2str(minA), "E-", num2str(maxE),
"H+", num2str(minH), "R", num2str(dpx), "um.png"], "-dpng");
272   %close(h7);
273
274
275 endfunction
```