SUBMITTED BY: BREITENBERGER, NICODEMUS

1 Exercise: Calculation of the work

Point moving in force field:

$$\vec{F} = (3x + y)\vec{i} + (x + 2y)\vec{j}$$

Two paths:

$$\gamma_1(t) = \begin{cases} t\vec{i} & t \in [0, 3) \\ 3\vec{i} + (t-3)\vec{j} & t \in [3, 6] \end{cases}$$

$$\gamma_2(t) = \left\{ t\vec{i} + t\vec{j} \quad t \in [0, 3) \right.$$

Calculate work :

$$W_{ab} = \int_{\vec{r}_a}^{\vec{r}_b} \vec{F}(\vec{r}) d\vec{r} = \int_{t_a}^{t_b} \vec{F}(\vec{r}) \frac{d\vec{r}}{dt} dt$$

With $\vec{r}(t) = \gamma_1(t)$:

$$W_{ab} = \int_0^3 ((3t+0)\vec{i}+(t+0)\vec{j}) \frac{d(t\vec{i})}{dt} dt + \int_3^6 ((9+(t-3))\vec{i}+(3+2(t-3))\vec{j}) \frac{d(3\vec{i} + (t-3)\vec{j})}{dt} dt$$

$$W_{ab} = \int_0^3 (9 + (t-3))dt + \int_3^6 (3 + 2(t-3))dt$$

$$= \int_0^3 (6 + t)dt + \int_3^6 (2t - 3)dt = \left[6t + \frac{1}{2}t^2\right]_0^3 + \left[t^2 - 3t\right]_3^6$$

$$= 18 + 4.5 + 36 - 18 - 9 + 9 = 40.5$$

With $\vec{r}(t) = \gamma_2(t)$:

$$W_{ab} = \int_0^3 ((3t + t)\vec{i} + (t + 2t)\vec{j}) \frac{d(t\vec{i} + t\vec{j})}{dt} dt$$

$$= \int_0^3 ((3t + t)\vec{i} + (t + 2t)\vec{j})(\vec{i} + \vec{j})dt$$

$$= \int_0^3 (3t + t + t + 2t)dt = \int_0^3 (3t + t + t + 2t)dt = \int_0^3 7t dt$$

$$= \left[\frac{7}{2}t^2\right]_0^3 = \frac{7}{2}9 = 31.5$$

The work depends on the path the point goes through the force field.

2 Exercise: Generating a grid for Katchalski-Katzir

```
#include <BALL/FORMAT/PDBFile.h>
#include <BALL/KERNEL/system.h>
#include <BALL/KERNEL/chain.h>
#include <BALL/KERNEL/residue.h>
#include <BALL/KERNEL/atom.h>
#include <BALL/KERNEL/protein.h>
#include <BALL/KERNEL/PTE.h>
#include <BALL/MATHS/vector3.h>
#include <array>
#include <iostream>
#include <fstream>

using namespace std;
using namespace BALL;

int alpha1 = 1;
int beta1 = 0;
int gamma1 = 2;

int main(int argc, char* argv[]){

  //   BALL::Vector3 translationVector(1,1,1);
  System kkSystem;
  PDBFile sourceFile;
  // BALL::PDBFile translateFile;

  vector<Vector3> atompos;
  Vector3 pos;
  vector<float> vwradii;
  float vwr;

  //Sanity checks for command-line arguments
  if(argc == 2){

    sourceFile.open(argv[1], ios::in);
  }else{

    cout << "Wrong amount of Parameters\n\n Useage: prog inFile \n
    ";
    return 1;
```

```
40
41  }
42
43  if(sourceFile.is_open()){
44
45    //the read action for a pdbfile reads the data into a system
      which can be manipulated afterwards - and saved
46    sourceFile.read(kkSystem);
47    sourceFile.close();
48
49    // get only first protein
50    Protein* protein = kkSystem.getProtein(0);
51
52    if(protein->countChains() > 0){
53
54      for(ChainIterator ch_it = protein->beginChain(); +ch_it; ++
      ch_it){
55
56        for(ResidueIterator r_it = ch_it->beginResidue(); +r_it;
      ++r_it){
57
58          for(AtomIterator a_it = r_it->beginAtom(); +a_it; ++a_it
      ){
59            Element element = a_it->getElement();
60            // name
61            cout << element.getName() << " ";
62            // position
63            pos = a_it->getPosition();
64            // save position in vector
65            atompos.push_back(pos);
66            cout << pos  << " ";
67            // van der waals radius
68            vwr = element.getVanDerWaalsRadius();
69            cout << vwr << endl;
70            vwradii.push_back(vwr);
71          }
72        }
73      }
74    }
75    for(int i(0); i < atompos.size(); i++){
76      cout << atompos[i] << vwradii[i] << endl;
77    }
78  }else{
79    cout << "Could not open PDB file." << '\n';
80  }
81
82  // dimension of katchalski katzir grid n*m*l
83  int n = 50;
84  int m = 50;
```

```
85    int l = 50;
86    float px;
87    float py;
88    float pz;
89
90    float kkgrid[n][m][l];
91    // initialize kkgrid
92    for (int i = 0; i < n; i++){
93
94      for (int j = 0; j < m; j++){
95
96        for (int k = 0; k < l; k++){
97          int c = 0;
98          while (c < atompos.size()){
99
100           px = 0.1*(i-n/2);
101           py = 0.1*(j-m/2);
102           pz = 0.1*(k-l/2);
103
104           // inside
105           if (px > (atompos[c][0] - 0.1*vwradii[c]) && px < (
      atompos[c][0] + 0.1*vwradii[c]) && py > (atompos[c][1] - 0.1*
      vwradii[c]) && py < (atompos[c][1] + 0.1*vwradii[c]) && pz > (
      atompos[c][2] - 0.1*vwradii[c]) && pz < (atompos[c][2] + 0.1*
      vwradii[c]) ){
106             kkgrid[i][j][k] = beta1;
107           }
108           // outside
109           else if  (px < (atompos[c][0] - 0.1*vwradii[c]) ||  px >
       (atompos[c][0] + 0.1*vwradii[c]) || py > (atompos[c][1] - 0.1
      *vwradii[c]) || py < (atompos[c][1] + 0.1*vwradii[c]) || pz >
      (atompos[c][2] - 0.1*vwradii[c]) || pz < (atompos[c][2] + 0.1*
      vwradii[c]) ){
110             kkgrid[i][j][k] = gamma1;
111           }
112           // boundary
113           else{
114             kkgrid[i][j][k] = alpha1;
115           }
116           c++;
117         }
118       }
119     }
120   }
121
122   // safe kkgrid to file
123   ofstream kkfile;
124   kkfile.open("kkgrid.txt");
125   kkfile << "x y z value" << endl;
```

```
126
127   for (int x(0); x < n; x++){
128
129     for (int y(0); y < m; y++){
130
131       for (int z(0); z < l; z++){
132
133         kkfile << 0.1*(x-n/2) << " " << 0.1*(y-m/2) << " " << 0.1*
      (z-l/2) << " " << kkgrid[x][y][z] << endl;
134       }
135     }
136   }
137
138   kkfile.close();
139
140   // countour plots
141   system("gnuplot plotkkgrid.gnu");
142
143   return 0;
144
145 }
```

Gnuplot script "plotkkgrid.gnu":

```
1  # gnuplot script to plot katchalski katzir output
2
3  set parametric
4  set contour base
5  set view 0,0,1
6  unset surface
7  set cntrparam levels 5
8  set dgrid3d
9
10 set title "Katchalski-Katzir Grid entries for different planes"
11
12 set size 2,2
13 set origin 0,0
14 set multiplot layout 3,1 columnsfirst scale 1,1
15
16 set xr [-10.0:10.0]
17 set yr [-10.0:10.0]
18 set zr [0:2]
19 set xlabel "x"
20 set ylabel "y"
21 splot "kkgrid.txt" using 1:2:4 title "x-y plane" with line
22
23 set xr [-10.0:10.0]
24 set yr [-10.0:10.0]
```

```
25 set zr [0:2]
26 set xlabel "y"
27 set ylabel "z"
28 plot "kkgrid.txt" using 2:3:4 title "y-z plane" with line
29
30 set xr [-10.0:10.0]
31 set yr [-10.0:10.0]
32 set zr [0:2]
33 set xlabel "x"
34 set ylabel "z"
35 plot "kkgrid.txt" using 1:3:4 title "x-z plane" with line
36
37 unset multiplot
```