

Simo K. Kivelä, 14.7.2004

Partikkelit pallon pinnalla

Tehtävänä on sijoittaa annettu määrä keskenään identtisiä partikkeleita mahdollisimman tasaisesti pallon pinnalle ja piirtää kuvio syntyvästä partikkelikonfiguraatiosta.

Tehtävä ei sellaisenaan ole hyvin määritelty: on täsmennettävä, mitä tarkoitetaan tasaisella sijoittamisella. Yksi — mutta ei ainoa — mahdollisuus on ajatella partikkeleiden olevan identtisiä sähkövarauksia, jotka kitkatta liikkuvat pallon pinnalla. Koska identtiset sähkövaraukset hylkivät toisiaan, ne pyrkivät sijoittumaan toisistaan niin kauas kuin mahdollista.

Sijoittuminen mahdollisimman kauas toisistaan voidaan luonnehtia fysikaalisesti siten, että systeemin potentiaalienergia pyrkii hakeutumaan minimiarvoon. Kahden partikkelin tapauksessa potentiaali on oleellisesti $1/r$, missä r on partikkeleiden välinen etäisyys. Useamman partikkelin tapauksessa kokonaispotentiaali on summa kaikista partikkeliparien potentiaaleista:

$$V = \sum_{i \neq j} \frac{1}{r_{ij}},$$

missä r_{ij} on i :n ja j :n partikkelin välinen etäisyys.

Jos partikkeleita on kaksi, ne sijoittuvat ilmeisestikin pallon vastakkaisille puolille.

Jos partikkeleita on kolme, ne joka tapauksessa määräävät erään tason, ja partikkelit siis sijaitsevat tämän tason ja pallon leikkausympyrällä. Jotta partikkelit pääsisivät mahdollisimman kauas toisistaan, tason tulee ilmeisestikin kulkea pallon keskipisteen kautta ja partikkeleiden sijaita ympyrällä 120 asteen välein.

Jos partikkeleita on enemmän, tulos ei ole yhtä selvä. Neljän partikkelin voisi arvata sijoittuvan tetraedrin kärkipisteisiin, mikä onkin oikea arvaus. Tämän jälkeen voisi arvella, että jos partikkeleita on yhtä paljon kuin jossakin Platonin kappaleessa (tetraedri, oktaedri, kuutio, dodekaedri, ikosaedri) on kärkiä, niin partikkelit sijoittuisivat juuri kyseisen Platonin kappaleen kärkiin. Tämä on kuitenkin väärä arvaus.

Seuraavassa esityksessä ongelma ratkaistaan minimoimalla partikkelisysteemin kokonaispotentiaali.

Ratkaisu

Partikkeleiden lukumäärä olkoon

$$In[1]:= n = 5$$

$$Out[1]:= 5$$

Partikkelit sidotaan yksikkösäteisen pallon pinnalle ilmoittamalla niiden sijainti pallokoordinaattien avulla. Pallokoordinaattien avulla lasketaan vastaavat suorakulmaiset koordinaatit ja kukin partikkeli ilmaistaan kolmialkioisena suorakulmaisten koordinaattien muodostamana listana. Partikkelin sijainnin ilmoittavat pallokoordinaatit ovat probleeman tuntemattomat.

Aluksi muodostetaan — hieman teknisesti — pallokoordinaattien nimet, so. probleeman muuttujat:

```
In[2]:= theta = Table[Symbol["theta" <> ToString[k]], {k, 1, n}]
```

```
Out[2]= {theta1, theta2, theta3, theta4, theta5}
```

```
In[3]:= fi = Table[Symbol["fi" <> ToString[k]], {k, 1, n}]
```

```
Out[3]= {fi1, fi2, fi3, fi4, fi5}
```

```
In[4]:= muuttujat = {Table[theta[[k]], {k, 1, n}], Table[fi[[k]], {k, 1, n}]}
```

```
Out[4]= {{theta1, theta2, theta3, theta4, theta5}, {fi1, fi2, fi3, fi4, fi5}}
```

Partikkelit voidaan nyt ilmaista listana, jossa kukin partikkeli on esitetty sen suorakulmaisten koordinaattien avulla:

```
In[5]:= partikkelit = Table[{Cos[theta[[k]]] * Cos[fi[[k]]],
    Cos[theta[[k]]] * Sin[fi[[k]]], Sin[theta[[k]]]}, {k, 1, n}]
```

```
Out[5]= {{Cos[fi1] Cos[theta1], Cos[theta1] Sin[fi1], Sin[theta1]},
    {Cos[fi2] Cos[theta2], Cos[theta2] Sin[fi2], Sin[theta2]},
    {Cos[fi3] Cos[theta3], Cos[theta3] Sin[fi3], Sin[theta3]},
    {Cos[fi4] Cos[theta4], Cos[theta4] Sin[fi4], Sin[theta4]},
    {Cos[fi5] Cos[theta5], Cos[theta5] Sin[fi5], Sin[theta5]}}
```

Systeemin potentiaalifunktio voidaan muodostaa tämän jälkeen. Osapotentiaali on partikkeliparin aiheuttama potentiaali ja kokonaispotentiaali partikkeliparien potentiaaleista muodostettu summa. Summeerauksessa käytetyt indeksiarvot $1 \leq j \leq n-1$, $j+1 \leq k \leq n$ merkitsevät, että jokainen partikkelipari tulee otetuksi huomioon.

```
In[6]:= osaPotentiaali[p_, q_] := 1 / Sqrt[Sum[(p[[k]] - q[[k]])^2, {k, 1, 3}]]
```

```
In[7]:= potentiaali = Sum[osaPotentiaali[partikkelit[[j]], partikkelit[[k]],
  {j, 1, n - 1}, {k, j + 1, n}]
```

```
Out[7]= 1 / (sqrt((Cos[fi1] Cos[theta1] - Cos[fi2] Cos[theta2])^2 +
  (Cos[theta1] Sin[fi1] - Cos[theta2] Sin[fi2])^2 +
  (Sin[theta1] - Sin[theta2])^2)) +
  1 / (sqrt((Cos[fi1] Cos[theta1] - Cos[fi3] Cos[theta3])^2 +
  (Cos[theta1] Sin[fi1] - Cos[theta3] Sin[fi3])^2 +
  (Sin[theta1] - Sin[theta3])^2)) +
  1 / (sqrt((Cos[fi2] Cos[theta2] - Cos[fi3] Cos[theta3])^2 +
  (Cos[theta2] Sin[fi2] - Cos[theta3] Sin[fi3])^2 +
  (Sin[theta2] - Sin[theta3])^2)) +
  1 / (sqrt((Cos[fi1] Cos[theta1] - Cos[fi4] Cos[theta4])^2 +
  (Cos[theta1] Sin[fi1] - Cos[theta4] Sin[fi4])^2 +
  (Sin[theta1] - Sin[theta4])^2)) +
  1 / (sqrt((Cos[fi2] Cos[theta2] - Cos[fi4] Cos[theta4])^2 +
  (Cos[theta2] Sin[fi2] - Cos[theta4] Sin[fi4])^2 +
  (Sin[theta2] - Sin[theta4])^2)) +
  1 / (sqrt((Cos[fi3] Cos[theta3] - Cos[fi4] Cos[theta4])^2 +
  (Cos[theta3] Sin[fi3] - Cos[theta4] Sin[fi4])^2 +
  (Sin[theta3] - Sin[theta4])^2)) +
  1 / (sqrt((Cos[fi1] Cos[theta1] - Cos[fi5] Cos[theta5])^2 +
  (Cos[theta1] Sin[fi1] - Cos[theta5] Sin[fi5])^2 +
  (Sin[theta1] - Sin[theta5])^2)) +
  1 / (sqrt((Cos[fi2] Cos[theta2] - Cos[fi5] Cos[theta5])^2 +
  (Cos[theta2] Sin[fi2] - Cos[theta5] Sin[fi5])^2 +
  (Sin[theta2] - Sin[theta5])^2)) +
  1 / (sqrt((Cos[fi3] Cos[theta3] - Cos[fi5] Cos[theta5])^2 +
  (Cos[theta3] Sin[fi3] - Cos[theta5] Sin[fi5])^2 +
  (Sin[theta3] - Sin[theta5])^2)) +
  1 / (sqrt((Cos[fi4] Cos[theta4] - Cos[fi5] Cos[theta5])^2 +
  (Cos[theta4] Sin[fi4] - Cos[theta5] Sin[fi5])^2 +
  (Sin[theta4] - Sin[theta5])^2))
```

Potentiaalifunktion lauseke on pitkäkö. Kun partikkeleita on n kappaletta, kyseessä on $2n$ muuttujan funktio. Tavoitteena on funktion minimiarvon löytäminen. Minimiarvoa vastaavat muuttujien arvot, so. pallokoordinaattien arvot antavat etsityn partikkelikonfiguraation.

Minimiarvo ja vastaava minimikohta voidaan laskea Mathematican funktiolla FindMinimum. Tämä käyttää iteratiivista algoritmia, jolle on annettava jotkin alkuarvot ja joka antaa tulokseksi alkuarvoista riippuen jonkin paikallisen minimin. Absoluuttista minimiä ei siten välttämättä löydetä. Kokeilemalla useita kertoja erilaisilla alkuarvoilla tilanne saadaan kuitenkin melko luotettavasti kartoitetuksi.

Alkuarvot voidaan syöttää käsin, mutta koska niitä on suhteellisen paljon eikä ole aivan helppoa mieltää, millaiset olisivat sopivia, voidaan alkuarvot generoida satunnaisesti pallokoordinaattien vaihteluväleiltä ($-\pi/2 \leq \vartheta \leq \pi/2$, $-\pi \leq \varphi \leq \pi$).

```
In[8]:= alkuarvot = {Table[Random[Real, {-Pi / 2, Pi / 2}], {k, 1, n}],
  Table[Random[Real, {-Pi, Pi}], {k, 1, n}]}
Out[8]= {{1.41782, -0.322664, 0.137728, -1.35344, 1.45411},
  {2.83955, -0.510143, -2.47585, 2.69836, 1.23015}}
```

Funktiossa FindMinimum tarvittava alkuasetus on tällöin muotoa

```
In[9]:= alkuasetus = Transpose[{Flatten[muuttujat], Flatten[alkuarvot]}]
Out[9]= {{theta1, 1.41782}, {theta2, -0.322664}, {theta3, 0.137728},
  {theta4, -1.35344}, {theta5, 1.45411}, {fi1, 2.83955},
  {fi2, -0.510143}, {fi3, -2.47585}, {fi4, 2.69836}, {fi5, 1.23015}}
```

Funktion kutsu tapahtuu teknisesti hieman mutkikkaalla tavalla, jotta voidaan helposti hyödyntää aiemmin muodostetut muuttujat. Optiolla MaxIterations säädellään iteratiivisen laskennan pituutta. Jos ratkaisua ei löydy, ts. iteraatio ei suppene, arvoa voi suurentaa. Vaihtoehtona on vaihtaa alkuarvoja.

```
In[10]:= rtk = Apply[FindMinimum,
  Flatten[{potentiaali, alkuasetus, MaxIterations -> 15}, 1]]
Out[10]= {6.47469, {theta1 -> 0.335276, theta2 -> -0.333363,
  theta3 -> 0.433159, theta4 -> -1.23048, theta5 -> 0.548588, fi1 -> 2.68934,
  fi2 -> -0.451619, fi3 -> -1.86178, fi4 -> 2.50175, fi5 -> 0.904378}}
```

Jos iteraatio suppenee, tuloksena saadaan jokin paikallinen minimiarvo ja sitä vastaavat muuttujien, ts. pallokoordinaattien arvot. Eri alkuarvoista lähettäessä saatetaan saada erilaisia paikallisia minimejä. Kyseessä ei siten välttämättä ole absoluuttinen minimi, mutta toistamalla laskenta ja vertailemalla saatuja minimiarvoja saadaan jonkinlainen käsitys siitä, onko löydetty absoluuttinen minimi.

```
In[11]:= minimiarvo = rtk[[1]]
Out[11]= 6.47469
```

Partikkelien suorakulmaiset koordinaatit saadaan tämän jälkeen helposti:

```
In[12]:= konfiguraatio = partikkelit /. rtk[[2]]
Out[12]= {{-0.849383, 0.412661, 0.329029},
  {0.850208, -0.412397, -0.327222}, {-0.260401, -0.869488, 0.419741},
  {-0.267762, 0.199295, -0.942648}, {0.527464, 0.670699, 0.521483}}
```

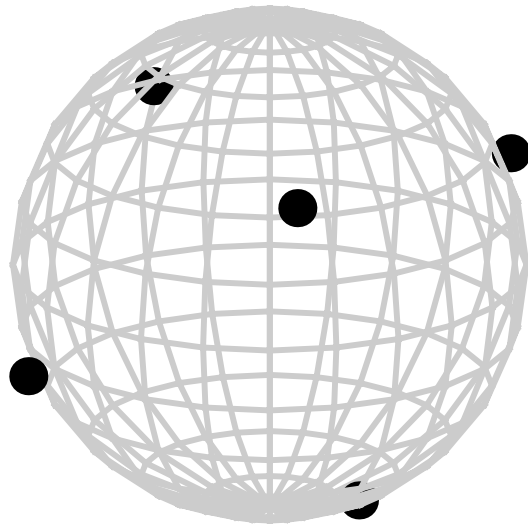
Tulokset graafisessa muodossa

Tieto partikkelien suorakulmaisista koordinaateista ei helposti anna mielikuvaa konfiguraatiosta. Graafinen tulostus on hyödyllisempi. Tällöin on luontevaa esittää myös pallopinta, jolla partikkelit sijaitsevat.

```
In[13]:= pallo = First[ParametricPlot3D[0.95 * {Cos[t] Cos[f], Cos[t] Sin[f], Sin[t]},
  {t, -Pi / 2, Pi / 2}, {f, -Pi, Pi}, PlotPoints -> {13, 25},
  DisplayFunction -> Identity]] /. Polygon -> Line;
```

```
In[14]:= pisteet = Map[Point, konfiguraatio];
```

```
In[15]:= kuvio = Show[Graphics3D[{{GrayLevel[0.8], Thickness[0.008], pallo},  
  {GrayLevel[0], PointSize[0.05], pisteet}}],  
  Shading → False, Boxed → False, ViewPoint → {100, 100, 50}]
```



```
Out[15]= - Graphics3D -
```

Animaatio auttaa vielä paremmin hahmottamaan pisteiden sijainnin:

```
In[16]:= << LiveGraphics3D.m
```

```
In[17]:= WriteLiveForm["partikkelit.m", kuvio]
```

```
- Animaatio -
```

Harjoitustehtäviä

1) Tutki tapauksia, missä pisteitä on 6, 7 tai 8. Vastaako kuutta pistettä oktaedri ja kahdeksaa pistettä kuutio? Kokeile useita erilaisia alkuarvoja.

2) Partikkeleiden tasaisen sijoittamisen kriteeri voidaan valita monella eri tavalla. Yhtenä mahdollisuutena on minimoida lauseketta, jossa osapotentiaali on muotoa $1/r^2$, vaikkei tämä fysikaalinen potentiaali olekaan. Tutki, ovatko konfiguraatiot tällöin erilaisia.

3) Potentiaalin minimoimisen sijasta voidaan maksimoida partikkeliparien etäisyyksien minimiarvoa, ts. vaatia, että lähinnä toisiaan olevien partikkeleiden välinen etäisyys on mahdollisimman suuri. Etäisyys voidaan tällöin mitata joko suoraviivaisesti tai pallon pintaa pitkin. Muunna laskenta tätä tapausa vastaavaksi. (Tarvittava funktio on tällöinkin `FindMinimum`; funktion f maksimin hakeminen on nimittäin sama asia kuin funktion $-f$ minimin hakeminen. Maksimoitava funktio ei ole tässä tapauksessa differentioituva, koska se on etäisyyksien minimiarvo, ts. muotoa `Min[...]`. Tällöin `FindMinimum` tarvitsee kaksi alkuarvoa kutakin muuttujaa kohden; ks. Mathematican dokumentaatiota. Iteraation suppeneminen on usein ongelmallista.)