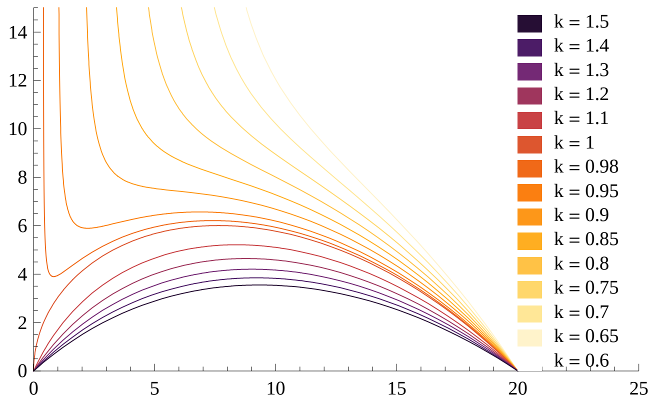
▪ Ujuja trajektoor 90° külghoovuse puhul, kui suunda hoitakse teatud punktile.

▪ Eelmise mudeli vigadest

Eelmisel mudelil, mis ei kasutanud diferentsiaalvõrrandite lahendamisvõtteid, ilmnesid olulised puudused.
Nagu märkis Rando, ei saa siis, kui ujuja omakiirus v_u ja jõe voolu kiirus v_j on võrdsed, mitte mingil juhul vastuvoolu liikuda (isegi mingil väikesel osal trajektoorist!); ka on vastuvoolu liikumine võimatu loomulikult juhul, kui ujuja kiirus on väiksem jõe voolu kiirusest. Olgu $\frac{vu}{vj} = k$, Kui k >1, siis on ujuja kiirus seisvas vees suurem jõe voolamise kiirusest; kui k = 1, siis on need kiirused võrdsed; kui k<1, siis voolab jõgi ujujast kiiremini.

Alljärgnevatel joonistel on kujutatud trajektoore erinevate k väärtuste korral. Paraku on arvuti poolt välja joonistatud trajektooride hulgas ka selliseid (k = 1, k = 0, 98, k = 0.95, mille puhul toimus ülalmainitud kiiruste puhul vastuvoolu liikumine.



Kuna öeldakse, et loodus ütleb "jah" sosistades, aga "ei" möirates, siis tuleks eelmine mudel kõrvale jätta.
Paraku ei olnud viga mudelis endas. Mudeli aluseks olevad liikumisvõrrandid olid siiski õiged. Probleemid tekkisid sellest, et arvutialgebra pakett *Mathematica* 6.0 andis mudelis tekkivat mittelineaarset võrrandisüsteemi lahendades ühe veateate, aga pakkus siiski välja tegelikele trajektooridele sarnaseid trajektoore pakkuva positiivse lahendi:

$$\mathbf{Solve}\left[-vj\,x + s\left(vj - \frac{vu\,y}{\sqrt{x^2+y^2}}\right) == 0\ \&\&\ x^2 + y^2 \neq 0,\ \{y, x\}\right]$$

Solve::svars : Equations may not give solutions for all "solve" variables. ≫

$$\left\{\left\{y \to -\frac{vj\,x\sqrt{-s^2+2\,s\,x-x^2}}{\sqrt{s^2\,vj^2 - s^2\,vu^2 - 2\,s\,vj^2\,x + vj^2\,x^2}}\right\},\ \left\{y \to \frac{vj\,x\sqrt{-s^2+2\,s\,x-x^2}}{\sqrt{s^2\,vj^2 - s^2\,vu^2 - 2\,s\,vj^2\,x + vj^2\,x^2}}\right\}\right\}$$

Esimese mudeli puhul sai välditud diferentsiaalvõrrandite kasutamist. Teise mudeli puhul said matemaatiliseks aparatuuriks aga just nimelt diferentsiaalvõrrandid.

Teise mudeli algallikas

Teine mudel leiab trajektooride võrrandid diferentsiaalvõrrandite analüütilise lahendamise teel ja selle algallikas paikneb aadressil:

https://answers.yahoo.com/question/index?qid=20081023044325AAVQb0z

Tõlkisin selle lahenduse eesti keelde, lisasin veidike kommentaare, muutsin kohati tähistusviisi mõistlikumaks ja kõrvaldasin ühe pisivea. Minu tegevuse tulemuse võib leida aadressilt

http://www.tlu.ee/~tonu/modesimu/difvorrandid/ujujatrajektoor2.pdf

Ülesande Püstitus

Ujuja ujub üle jõe laiusega s, ta ujumise kiirus seisvas vees on v_u jõe voolu kiirus on v_j. Ujuja stardib punktist A (*s*; 0) ja tema sihtpunktiks on *B* (0; 0). Ujuja liigub alati sihtpunkti suunas. Suvalisel ajahetkel on ujuja asukoha koordinaadid (*x*; *y*).

Kasutame failis www.tlu.ee/~tonu/modesimu/difvorrandid/ujujatrajektoor2.pdf tuletatud valemit :

$$y = \frac{s^{\frac{v_j}{v_u}} x^{1-\frac{v_j}{v_u}} - s^{-\frac{v_j}{v_u}} x^{1+\frac{v_j}{v_u}}}{2}$$

```
Remove["Global`*"]
```

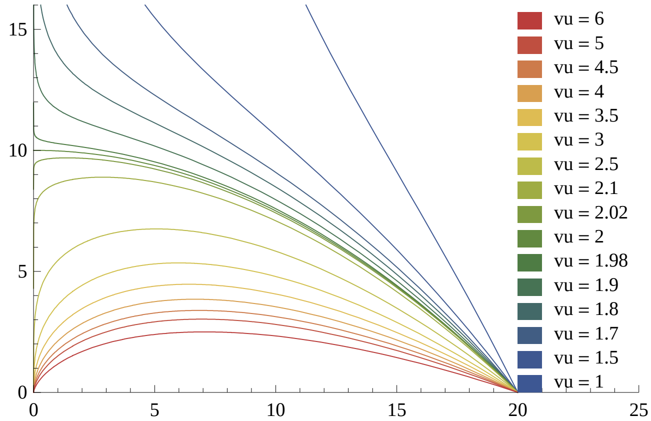
$$G[s\_, vj\_, vu\_, x\_] := \frac{s^{\frac{vj}{vu}} x^{1-\frac{vj}{vu}} - s^{-\frac{vj}{vu}} x^{1+\frac{vj}{vu}}}{2};$$

Niiviisi parameetreid muutes saab joonistada välja terveid selliste joonte parvesid. Jätame parameetrid jõe laius *s* ja voolukiirus *vj* iga katse puhul muutumatuteks, ning varieerime parameetrit ujuja kiirus *vu*. Kui niiviisi teha, (näiteks *s =20* ja *vj*=2 puhul nagu alljärgnevas), siis saame terveid graafikute parvesid, mis kirjeldavad ujuja trajektoori sõltuvust tema ujumise kiirusest:

```
s = 20;
```

```
vuList = {1, 1.5, 1.7, 1.8, 1.9, 1.98, 2, 2.02, 2.1, 2.5, 3, 3.5, 4, 4.5, 5, 6};
jooned = G[20, 2, #, x] & /@ vuList;
cols = Table[ColorData["DarkRainbow"][i], {i, 0, 1, 1 / Length@vuList}];
legend = Table[{cols[[i]], Rectangle[{s, i - 1}, {s + 1, i - 0.3}],
    Black, Text["vu = " <> ToString[vuList[[i]]], {s + 1.5, i - 1}, {-1, -1}]},
   {i, 1, Length@vuList}];
```

```
Plot[jooned, {x, 0, 20}, AspectRatio → Automatic, PlotRange → {{0, 25}, {0, 16}},
 PlotStyle → cols, Epilog → legend]
```



Tähistades $r = \frac{v_j}{v_u}$ jõe voolu kiiruse ja ujuja seisvas vees ujumise kiiruse suhet, saame esialgse valemi esitada kujul :

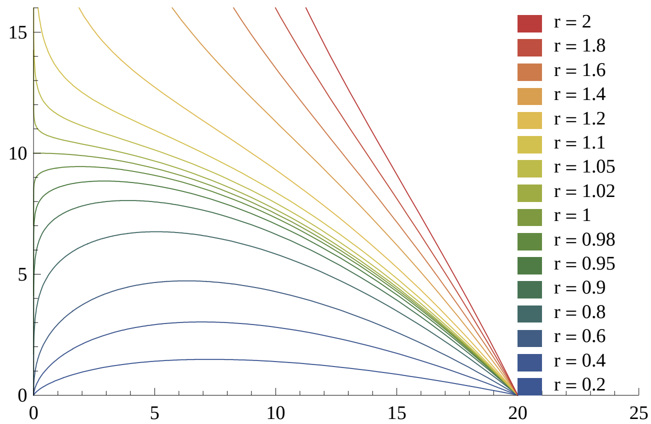
$$y = \frac{s^r x^{1-r} - s^{-r} x^{1+r}}{2}$$

Defineerime kolme muutujaga funktsiooni, milles suurus *y* sõltub suurusest *x*, aga saab muuta ka parameetrite *s*, ja *r* väärtusi.

```
H[s_, r_, x_] := (s^r x^(1-r) - s^-r x^(1+r))/2;

rlist = {0.2, 0.4, 0.6, 0.8, 0.9, 0.95, 0.98, 1, 1.02, 1.05, 1.1, 1.2, 1.4, 1.6, 1.8, 2};
jooned = H[20, #, x] & /@ rlist;
cols = Table[ColorData["DarkRainbow"][i], {i, 0, 1, 1 / Length@rlist}];
legend = Table[{cols[[i]], Rectangle[{s, i - 1}, {s + 1, i - 0.3}],
    Black, Text["r = " <> ToString[rlist[[i]]], {s + 1.5, i - 1}, {-1, -1}]},
   {i, 1, Length@rlist}];

Plot[jooned, {x, 0, s}, AspectRatio → Automatic, PlotRange → {{0, 25}, {0, 16}},
 PlotStyle → cols, Epilog → legend]
```



Selleks, et arvutada välja trajektoori pikkus valemi $S = \int_{x1}^{x2} \sqrt{1 + (f'(x))^2}\, dx$ abil,

leiame funktsiooni H tuletise ja nimetame selle HTul :

```
s =.; r =.; HTul[s_, r_, x_] = D[H[s, r, x], x]
```

$$\frac{1}{2}\left((1-r)\, s^r x^{-r} - (1+r)\, s^{-r} x^r\right)$$

Leame nüüd trajektoori pikkusi ühe ja sama *s*=20, aga erinevate *r* väärtuste puhul:
Kõigepealt piirjuhtum, kus *r* =1 (siis ei jõua ujuja punkti B, aga jõuab vastaskaldal mingisse teise punkti);
juhtum, kus *r*= 0,5 (hoovuse kiirus on ujuja liikumiskiirusest seisva vee suhtes on 2 korda väiksem)
juhtum kus r =0,1 (hoovuse kiirus on ujuja liikumiskiirusest seisva vee suhtes on 10 korda väiksem)
juhtum, kus *r* = 0,99 (hoovuse kiirus on ujuja kiirusest 1% võrra väiksem)
juhtum, kus *r* = 1,01 (hoovuse kiirus on ujuja kiirusest 1% võrra suurem)

```
s = 20; r = 1; NIntegrate[Sqrt[1 + (HTul[s, r, x])^2], {x, 0, s}]
```

22.9559

```
s = 20; r = 0.5; NIntegrate[Sqrt[1 + (HTul[s, r, x])^2], {x, 0, s}]
```

22.4687

```
s = 20; r = 0.99; NIntegrate[Sqrt[1 + (HTul[s, r, x])^2], {x, 0, s}]
```

32.3115

```
s = 20; r = 1.01; NIntegrate[Sqrt[1 + (HTul[s, r, x])^2], {x, 0, s}]
```

NIntegrate::slwcon :
Numerical integration converging too slowly; suspect one of the following: singularity, value of the integration is 0, highly oscillatory integrand, or WorkingPrecision too small. ≫

NIntegrate::ncvb :
NIntegrate failed to converge to prescribed accuracy after 9 recursive bisections in x near {x} = $\{2.44826\times10^{-224}\}$.
NIntegrate obtained 4.818475278732521`*^283 and 4.818475278732521`*^283
for the integral and error estimates. ≫

4.81848×10^{283}

Kokkuvõte - Et antud mudeli puhul on valemite tuletamine tehtud ära pliiatsi ja paberiga, siis on see mudel täpsem ja füüsikaliselt usaldusväärsem kui eelmine mudel, mis ei kasutanud diferentsiaalvõrrandite süsteemi analüütilist lahendamist, vaid töötas heas usus, et *Mathematica* 6.0 lahendab mittelineaarseid võrrandisüsteeme õigesti... Paraku on ka sellel mudelil teatavaid puudujääke. See mudel on koostatud spetsiaalselt lihtsaima erijuhtumi jaoks, kus hoovuse suund on risti lõiguga *AB*. Kui aga koostada liikumise diferentsiaalvõrrandid juhtumi jaoks, kus ujujale või kajakimatkajale mõjub mingi muu nurga all mõjuv hoovus, siis neid võrrandeid ei saa niisama lihtsalt analüütiliselt lahendada kui antud lihtsaima erijuhtumi puhul. Kuigi trajektooride kujud ja pikkused on lihtsalt arvutatavad, võib tekkida raskusi trajektooride läbimise aegade arvutamisega.

Samas, tegin ära esimesed katsed selliste liikumise diferentsiaalvõrranditega, mille puhul ujujale või kajakimatkajale mõjub mõjub hoovus 90 kraadisest nurgast erineva nurga all. Kuna analüütiliselt ma hetkel neid võrrandeid lollikindlal meetodil lahendada ei mõista, siis proovisin numbrilist lahendamist ja üllatav küll, sain juba väga häid tulemusi. Tulemuseks tekkis hulk huvitavaid trajektoore, mis 90 kraadise erijuhtumi puhul langevad kokku antud mudeli omadega. Lisaks muule paistab, et numbriliste meetodite puhul ei ole raske määrata ka trajektoori läbimiseks kuluvat aega.

01.05.2014. Tõnu Tõnso