

## 0.1 Hydrodynamika srážek asteroidů

Při srážkách pevných těles se potýkáme s následujícím problémem: tělesa na začátku zabírají jen omezený prostor a okolo je vakuum. Po srážce se však rozpadnou na úlomky, které odlétají vysokou rychlostí. Výpočetní doména by tak musela být zbytečně velká a kdyby se ukázalo, že úlomky letí „do Prčic“, byli bychom tamtéž. Proto je mnohem výhodnější pro srážky použít popis *lagrangeovský*, při kterém se pozorovatel pohybuje s prouděním kamkoliv.

### 0.1.1 Lagrangeův popis

Sepišme nejprve potřebné rovnice. Říkali jsme, že sledujeme pohyb, čili na levé straně používáme zásadně totální časové derivace. Jde o obdobu pohybových rovnic, jak je známe pro hmotné body,  $\mathbf{a} = \frac{\mathbf{F}}{m}$ . Rovnice kontinuity je tedy (v jednotkách  $\text{kg m}^{-3} \text{s}^{-1}$ ):

$$\frac{d\rho}{dt} = -\rho \nabla \cdot \mathbf{v}, \quad (1)$$

Navierova–Stokesova rovnice ( $\text{m s}^{-2}$ ):

$$\frac{d\mathbf{v}}{dt} = -\frac{1}{\rho} \nabla P - \nabla \Phi + \frac{1}{\rho} \nabla \cdot \mathbf{S}, \quad (2)$$

1. věta termodynamická ( $\text{J kg}^{-1} \text{s}^{-1}$ ):<sup>1</sup>

$$\frac{dU}{dt} = -\frac{1}{\rho} P \nabla \cdot \mathbf{v} + \frac{1}{\rho} \mathbf{S} : \overbrace{\left[ \nabla \mathbf{v} + (\nabla \mathbf{v})^T \right]}^{= \varepsilon_{ij}}, \quad (3)$$

Poissonova rovnice ( $\text{J kg}^{-1} \text{m}^{-2}$ ):

$$\nabla^2 \Phi = 4\pi G \rho, \quad (4)$$

stavová rovnice pro pevnou látku (Tillotson 1962; Pa)<sup>2</sup>:

$$P = \begin{cases} A\left(\frac{\rho}{\rho_0} - 1\right) + B\left(\frac{\rho}{\rho_0} - 1\right)^2 + a\rho U + \frac{b\rho U}{\frac{U}{U_0} \frac{\rho_0^2}{\rho^2} + 1} & \text{pro } U < U_{iv}, \\ a\rho U + \left[ \frac{b\rho U}{\frac{U}{U_0} \frac{\rho_0^2}{\rho^2} + 1} + A\left(\frac{\rho}{\rho_0} - 1\right) e^{-\beta\left(\frac{\rho_0}{\rho} - 1\right)} \right] e^{-\alpha\left(\frac{\rho_0}{\rho} - 1\right)} & \text{pro } U > U_{cv}, \end{cases} \quad (5)$$

<sup>1</sup> Dvoutečkový součin tenzorů  $\mathbf{S} : \varepsilon_{ij}$  je skalár  $\sum_i \sum_j S_{ij} \varepsilon_{ij}$ , kde přes opakující se indexy  $i$  a  $j$  sčítáme, i kdybychom sumy zapomněli; zde zavedené  $\varepsilon_{ij}$  se jinak nazývá tenzor malých deformací.

<sup>2</sup> Přibližně zohledňuje i fázové přechody, ostatně člen  $\rho U$  odpovídá ideálnímu plynu; dokonalejší stavovou rovnici by byla ANEOS (Melosh 2000) nebo SESAME (Plesko aj. 2014), ale tyto nejsou volně dostupné.

konstituční rovnice pro pevnou látku (Hookeův zákon; Pas<sup>-1</sup>):

$$\frac{d\mathbf{S}}{dt} = 2\mu_1 \frac{1}{2} [\nabla\mathbf{v} + (\nabla\mathbf{v})^T] + (\mu_2 - \frac{2}{3}\mu_1) \nabla \cdot \mathbf{v} \mathbf{I}. \quad (6)$$

Značení jest následující:  $\rho$  hustota,  $\mathbf{v}$  rychlost,  $P$  tlak (izotropní část),  $\mathbf{S}$  tenzor napětí (deviatorická část),  $U$  měrná vnitřní tepelná energie (tentokrát na kg),  $\rho_0$  hustota při nulovém tlaku,  $U_0$  vnitřní energie při nulovém tlaku,  $U_{iv}$  při počínajícím vypařování,  $U_{cv}$  při úplném vypaření,  $A, B, a, b, \alpha, \beta$  další parametry stavové rovnice,  $\mu_1$  dynamická (první) viskozita,  $\mu_2$  objemová viskozita,  $\mathbf{I}$  jednotkový tenzor.<sup>3,4</sup>

Kdybychom chtěli konstituční rovnici rozepsat maticově, viděli bychom, že:

$$\begin{aligned} \frac{d}{dt} \begin{pmatrix} S_{11} & S_{21} & S_{31} \\ S_{12} & S_{22} & S_{32} \\ S_{13} & S_{23} & S_{33} \end{pmatrix} = 2\mu_1 \begin{pmatrix} \frac{\partial v_1}{\partial x_1} & \frac{1}{2} \left( \frac{\partial v_2}{\partial x_1} + \frac{\partial v_1}{\partial x_2} \right) & \frac{1}{2} \left( \frac{\partial v_3}{\partial x_1} + \frac{\partial v_1}{\partial x_3} \right) \\ \frac{1}{2} \left( \frac{\partial v_2}{\partial x_1} + \frac{\partial v_1}{\partial x_2} \right) & \frac{\partial v_2}{\partial x_2} & \frac{1}{2} \left( \frac{\partial v_3}{\partial x_2} + \frac{\partial v_2}{\partial x_3} \right) \\ \frac{1}{2} \left( \frac{\partial v_3}{\partial x_1} + \frac{\partial v_1}{\partial x_3} \right) & \frac{1}{2} \left( \frac{\partial v_3}{\partial x_2} + \frac{\partial v_2}{\partial x_3} \right) & \frac{\partial v_3}{\partial x_3} \end{pmatrix} + \\ + (\mu_2 - \frac{2}{3}\mu_1) \begin{pmatrix} \frac{\partial v_1}{\partial x_1} + \frac{\partial v_2}{\partial x_2} + \frac{\partial v_3}{\partial x_3} & 0 & 0 \\ 0 & \frac{\partial v_1}{\partial x_1} + \frac{\partial v_2}{\partial x_2} + \frac{\partial v_3}{\partial x_3} & 0 \\ 0 & 0 & \frac{\partial v_1}{\partial x_1} + \frac{\partial v_2}{\partial x_2} + \frac{\partial v_3}{\partial x_3} \end{pmatrix}. \end{aligned}$$

Všimněme si, že se zde tenzor napětí  $\mathbf{S}$  velmi liší od tekutin! Napětí je v pevné látce přítomno, i když pohyb ustane. V tekutinách by napětí existovalo pouze při vzájemných pohybech; konstituční rovnice by proto byla bez časové derivace vlevo, tzn.  $\mathbf{S} = \mu_1 [\nabla\mathbf{v} + (\nabla\mathbf{v})^T] + (\mu_2 - \frac{2}{3}\mu_1) \nabla \cdot \mathbf{v} \mathbf{I}$ .

Kolik je zde neznámých funkcí? Jde o  $\rho(t)$ ,  $\mathbf{v}(t)$ ,  $U(t)$ ,  $\Phi(t)$ ,  $P(t)$ ,  $\mathbf{S}(t)$ , tzn. 13, počítáme-li 1 vektor jako 3 skaláry a symetrický tenzor jako 6 skalárů. Uvědomme si, že se jedná výhradně o funkce času, nikoli souřadnic! Někdy se pouze setkáváme s tím, že se body kontinua „indexují“ pomocí *počátečních* souřadnic ( $\mathbf{r}_0$ ), ale to jsou pouhé značky, nikoli nezávislé proměnné.

### 0.1.2 Elasticita, plasticita a praskliny

U pevných látek musíme být obezřetní, neboť platnost Hookeova zákona je omezena. Platí pouze v *elastické oblasti* (viz obr. 1), ve které je vztah mezi deformací (tj.

<sup>3</sup> Symboly  $\nabla\mathbf{v}$ ,  $(\nabla\mathbf{v})^T$  jsou tenzory druhého řádu se složkami  $\nabla_i v_j$ , resp.  $\nabla_j v_i$ .

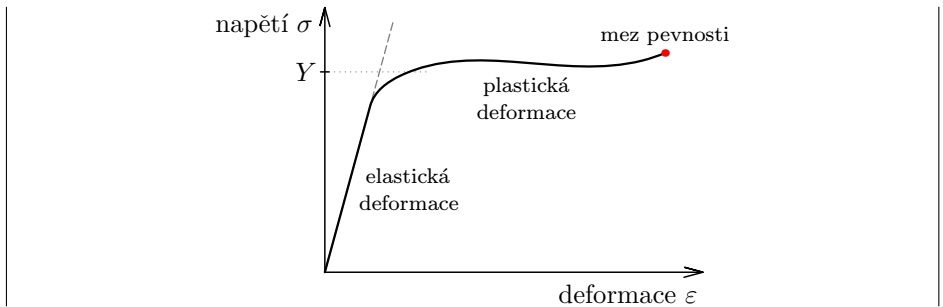
<sup>4</sup> Faktor  $-\frac{2}{3}\mu_1$  je před  $\nabla \cdot \mathbf{v}$  proto, že izotropní působení již bylo vyčleněno jako tlak, resp.  $\nabla P$ , čili zde ho musíme odečíst! V Navierově–Stokesově rovnici, tzn. po provedení  $\nabla \cdot \mathbf{S}$ , by odpovídal faktoru  $+\frac{1}{3}\mu_1$ .

relativní změnou délky) a napětím lineární, schematicky  $\varepsilon = E\sigma$ ; kde  $E$  označuje Youngův modul pružnosti.

Pak ovšem následuje *plastická oblast*, kde deformace rostou i při takřka konstantním (velkém) napětí. Toto popisují poddajné vztahy (angl. yielding), například von Misesovo kritérium. Jde o horní mez pro tenzor napětí, resp. jeho neizotropickou část:

$$\mathbf{s} = f\mathbf{S}, \quad f = \min \left[ \frac{Y^2}{3J_2}, 1 \right], \quad J_2 = \frac{1}{2}S^{\alpha\beta}S^{\alpha\beta}, \quad (7)$$

kde  $Y$  označuje mez pružnosti.



**Obr. 1** — Typická závislost mezi napětím  $\sigma$  a deformací  $\varepsilon$  pro pevné látky. První oblast, elastická, je přibližně lineární,  $\sigma = E\varepsilon$ . Druhá oblast je plastická, ve které  $\varepsilon$  roste i při stálém (velkém)  $\sigma \simeq Y$ . Nakonec je při určitém  $\varepsilon$  překročena mez pevnosti. Podle Maindl (2013).

Nakonec musíme uvážit *mez pevnosti*, po jejímž překročení se nepatrné poruchy v materiálu rozvíjejí jako praskliny, rostoucí řádově poloviční rychlostí zvuku. Grady a Kipp (1980) pro popis těchto jevů zavedli parametr *poškození*  $D$  (angl. damage),  $0 \leq D \leq 1$ , a provádějí logickou úpravu celkového tenzoru napětí pro případ stlačování nebo roztahování:

$$\sigma_{\alpha\beta} = \begin{cases} -P\delta_{\alpha\beta} + (1-D)S_{\alpha\beta} & \text{pro } P \geq 0, \\ -(1-D)P\delta_{\alpha\beta} + (1-D)S_{\alpha\beta} & \text{pro } P < 0. \end{cases} \quad (8)$$

Zcela zničený materiál (rozuměj rozdrcený na prach) s  $D = 1$  nevykazuje žádná nediagonální napětí ani se nijak nebrání roztahování.

Související a neméně důležitou vlastností materiálu je koncentrace poruch, respektive její rozdělení (Weibull 1939):

$$n(\epsilon) = k\epsilon^m \quad (9)$$

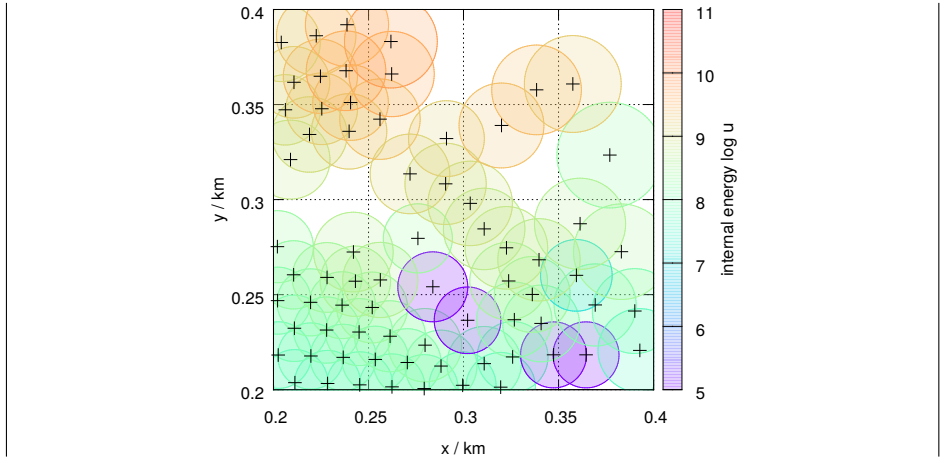
dle aktivační meze  $\epsilon$ , nad kterou se začnou poruchy rozvíjet v praskliny;  $k$ ,  $m$  jsou materiálové parametry. Rovnice pro vývoj poškození je:

$$\frac{dD^{\frac{1}{3}}}{dt} = \left[ \left( \frac{c_g}{R_s} \right)^3 + \left( \frac{m+3}{3} \alpha^{\frac{1}{3}} \epsilon^{\frac{m}{3}} \right)^3 \right]^{\frac{1}{3}}, \quad (10)$$

kde  $c_g$  je rychlost růstu poruch (řádově polovina rychlosti zvuku  $c_s$ ),  $R_s$  poloměr příslušný objemu, v němž poruchy studujeme, a parametr  $\alpha \equiv 8\pi c_g^3 k / [(m+1)(m+2)(m+3)]$ . Podrobný popis této složité problematiky je uveden v Benz a Asphaug (1994).

### 0.1.3 Metoda hlazená částicová (SPH)

Pro numerické výpočty se často používá metoda SPH (z angl. smoothed particle hydrodynamics), čili metoda hlazená částicová. Namísto kontinua uvažujeme množinu vzájemně interagujících částic (obr. 2). Hustota je prostě daná jejich počtem v daném objemu. Pohybují se jako by šlo hmotné body (dle lagrangeovských rovnic), ale mají samozřejmě přisouzeno vícero vlastností ( $\mathbf{v}$ ,  $U$ ,  $P$ ,  $\mathbf{S}$ ). Možná bychom jim proto neměli říkat částice, ale raději „numerická vozidla“.



**Obr. 2** — Grafické znázornění rozlehlých částic v metodě SPH. Poloměr zde odpovídá hladící délce  $h$ , interakce (gradienty) se počítají pro částice vzdálené méně než  $2h$ .

Vezmeme nejprve rovnici kontinuity (1). Využijeme následující integrální reprezentaci funkce pomocí konvoluce s Diracovou distribucí  $\delta$ :

$$\mathbf{v}(\mathbf{r}) = \int_{\Omega} \mathbf{v}(\mathbf{r}') \delta(|\mathbf{r} - \mathbf{r}'|, h) d\Omega.$$

K čemu nám to vůbec je, vyjádřit hodnotu v bodě  $\mathbf{r}$  pomocí hodnot „někde jinde“?! Genialita postupu (Gingold a Monaghan 1977, Lucy 1977) spočívá v tom, že  $\delta$  nahradíme vhodnou diferencovatelnou funkcí  $W$ , zvanou *kernel*, a následně se elegantně zbavíme všech prostorových derivací, konkrétně  $\nabla \cdot \mathbf{v}$ . Čili:

$$\mathbf{v}(\mathbf{r}) \doteq \int_{\Omega} \mathbf{v}(\mathbf{r}') W(|\mathbf{r} - \mathbf{r}'|, h) d\Omega, \quad (11)$$

kde  $\Omega$  označuje oblast prostoru,  $h$  hladící délku; jde o aproximaci s přesností  $\mathcal{O}(h^2)$ . Pak vyjádříme divergenci:

$$\begin{aligned} \nabla \cdot \mathbf{v}(\mathbf{r}) &= \int_{\Omega} [\nabla_{\mathbf{r}'} \cdot \mathbf{v}(\mathbf{r}')] W(|\mathbf{r} - \mathbf{r}'|, h) d\Omega = \\ &= \int_{\Omega} \nabla_{\mathbf{r}'} \cdot [\mathbf{v}(\mathbf{r}') W(|\mathbf{r} - \mathbf{r}'|, h)] d\Omega - \int_{\Omega} \mathbf{v}(\mathbf{r}') \cdot \nabla_{\mathbf{r}'} W(|\mathbf{r} - \mathbf{r}'|, h) d\Omega = \\ &= \int_{\partial\Omega} \mathbf{v}(\mathbf{r}') \overbrace{W(|\mathbf{r} - \mathbf{r}'|, h)}^{= 0 \text{ na hranici}} d\vec{\Gamma} - \dots = \\ &= - \int_{\Omega} \mathbf{v}(\mathbf{r}') \cdot \nabla_{\mathbf{r}'} W(|\mathbf{r} - \mathbf{r}'|, h) d\Omega = \int_{\Omega} \mathbf{v}(\mathbf{r}') \cdot \nabla_{\mathbf{r}} W(|\mathbf{r} - \mathbf{r}'|, h) d\Omega, \quad (12) \end{aligned}$$

kde jsme využili derivaci součinu, Gaussovu větu a skutečnost, že na hranici je kernel  $W$  roven nule. Poznamenejme, že operátor  $\nabla$  nejprve působí na  $\mathbf{r}'$ , ale na konci jsme uplatnili antisymetrii  $\nabla W$ . Derivace samozřejmě nezmizely úplně; zůstal nám zde ale jakýsi vážený průměr gradientu  $W$ , který lze při zvoleném  $W$  předpočítat.

Diskretizace v prostoru spočívá v nahrazení integrálu sumou:

$$\nabla \cdot \mathbf{v}_i \doteq \sum_{j=1}^{N_{\text{okolo}}} \mathbf{v}_j \cdot \nabla W(|\mathbf{r}_i - \mathbf{r}_j|, h) \frac{m_j}{\rho_j}, \quad (13)$$

přičemž sčítáme pouze přes částice nacházející se v okolí  $|\mathbf{r}_i - \mathbf{r}_j| < h$ .

Nejjednodušší diskretizace v čase by zřejmě byla:

$$\frac{d\rho}{dt} \doteq \frac{\rho_i^{n+1} - \rho_i^n}{\Delta t}; \quad (14)$$

výsledné explicitní integrační schéma je pak:

$$\rho_i^{n+1} = \rho_i^n - \Delta t \rho_i^n \sum_j \mathbf{v}_j^n \cdot \nabla W(|\mathbf{r}_i - \mathbf{r}_j|, h) \frac{m_j}{\rho_j^n}.$$

Pro Navierovu–Stokesovu rovnici (2) bychom potřebovali  $\nabla P$ :

$$\nabla P(\mathbf{r}) = \int_{\Omega} [\nabla_{\mathbf{r}'} P(\mathbf{r}')] W(|\mathbf{r} - \mathbf{r}'|, h) d\Omega = \int_{\Omega} P(\mathbf{r}') \nabla_{\mathbf{r}} W(|\mathbf{r} - \mathbf{r}'|, h) d\Omega,$$

což po diskretizaci vede na:

$$\nabla P_i \doteq \sum_j P_j \nabla W(|\mathbf{r}_i - \mathbf{r}_j|, h) \frac{m_j}{\rho_j}. \quad (15)$$

Tento výraz však není symetrický vzhledem k  $i, j$ , což se ukazuje jako nebezpečné z hlediska numerického. Obdobně musíme vyjádřit  $\nabla \cdot \mathbf{S}$ :

$$\nabla \cdot \mathbf{S}(\mathbf{r}) = \int_{\Omega} [\nabla_{\mathbf{r}'} \cdot \mathbf{S}(\mathbf{r}')] W(|\mathbf{r} - \mathbf{r}'|, h) d\Omega = \int_{\Omega} \mathbf{S}(\mathbf{r}') \cdot \nabla_{\mathbf{r}} W(|\mathbf{r} - \mathbf{r}'|, h) d\Omega,$$

čili:

$$\nabla \cdot \mathbf{S}_i \doteq \sum_j \mathbf{S}_j \cdot \nabla W(|\mathbf{r}_i - \mathbf{r}_j|, h) \frac{m_j}{\rho_j}, \quad (16)$$

s touž poznámkou.

Pro energetickou rovnici (3) bychom potřebovali ještě znát diskretizaci tenzoru (psáno raději ve složkách):

$$\nabla_{\alpha} v_{\beta} = \int_{\Omega} [\nabla_{\alpha} v_{\beta}] W(|\mathbf{r} - \mathbf{r}'|, h) d\Omega = \sum_j v_{\beta j} \nabla_{\alpha} W(|\mathbf{r}_i - \mathbf{r}_j|, h) \frac{m_j}{\rho_j}.$$

Celá metoda SPH by tedy mohla vypadat takto (píšeme ovšem stručněji  $W_{ij}(h)$  namísto  $W(|\mathbf{r}_i - \mathbf{r}_j|, h)$ ):

$$\rho_i^{n+1} = \rho_i^n - \Delta t \rho_i^n \sum_j \mathbf{v}_j^n \cdot \nabla W_{ij}(h) \frac{m_j}{\rho_j^n}, \quad (17)$$

$$\mathbf{v}_i^{n+1} = \mathbf{v}_i^n - \frac{\Delta t}{\rho_i^n} \sum_j P_j^n \nabla W_{ij}(h) \frac{m_j}{\rho_j^n} + \frac{\Delta t}{\rho_i^n} \sum_j \mathbf{S}_j^n \cdot \nabla W_{ij}(h) \frac{m_j}{\rho_j^n}, \quad (18)$$

$$U_i^{n+1} = U_i^n - \Delta t P_i^n \sum_j \mathbf{v}_j^n \cdot \nabla W_{ij}(h) \frac{m_j}{\rho_j^n} + \\ + \sum_{\alpha=1}^3 \sum_{\beta=1}^3 S_{\alpha\beta}^n \frac{1}{2} \sum_j \left[ v_{\beta j}^n \frac{\partial}{\partial x_{\alpha}} W_{ij}(h) + v_{\alpha j}^n \frac{\partial}{\partial x_{\beta}} W_{ij}(h) \frac{m_j}{\rho_j^n} \right]. \quad (19)$$

Pro jednoduchost jsme si dovolili vynechat gravitační zrychlení ( $-\nabla\Phi$ ), abychom nemuseli řešit Poissonovu rovnici, nicméně viz níže.

### 0.1.4 Alternativní vyjádření prostorových derivací

Existují ovšem i alternativní lepší formulace (ekvivalentní až na  $\mathcal{O}(h^2)$ ), které jsou numericky stabilnější (Cossins 2010, Price 2008). Vycházejí z lagrangiánu nebo z následující vektorové identity, resp. gradientu součinnu:

$$\nabla P \rho^a = \rho^a \nabla P + P a \rho^{a-1} \nabla \rho,$$

odkud:

$$\nabla P = \frac{1}{\rho^a} [\nabla P \rho^a - P a \rho^{a-1} \nabla \rho].$$

Speciálně pro  $a = 1$ :

$$\nabla P = \frac{1}{\rho} [\nabla P \rho - P \rho^{-1} \nabla \rho],$$

což by mimochodem přesně řešilo konstantní  $P$ . Pro  $a = -1$  bychom obdrželi:

$$\nabla P = \rho \nabla P \rho^{-1} + P \rho^{-1} \nabla \rho,$$

což by naopak zachovávalo celkovou hybnost. Tlakový člen by měl po provedení příslušných diskretizací podobu buď:

$$\nabla P_i = \frac{1}{\rho_i} \sum_j (P_i - P_j) \nabla W_{ij}(h) m_j,$$

nebo:

$$\nabla P_i = \rho_i \sum_j \left( \frac{P_i}{\rho_i^2} + \frac{P_j}{\rho_j^2} \right) \nabla W_{ij}(h) m_j;$$

obzvláště poslední je krásně symetrický.

### 0.1.5 Kernel čili hladící funkce

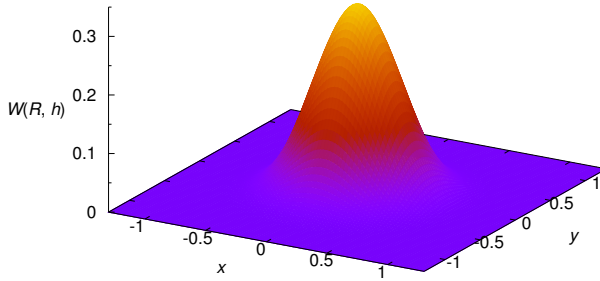
Ideálně by měl kernel  $W$  splňovat tyto požadavky: i) normalita, ii) kompaktnost, iii) limita pro  $h \rightarrow 0$  je  $\delta$  funkce, iv) pozitivita, v) být ryze klesající, vi) symetrie, vii) hladkost. Splnit všech sedm najednou je takřka nemožné, například Gaussova funkce postrádá ii). Nicméně nejpoužívanější kubický spline na tom není špatně, lze mu vytýkat jen vii) (obr. 3,  $R \equiv r/h$ ):

$$W(R, h) = \frac{3}{2\pi h^3} \begin{cases} \frac{2}{3} - 4R^2 + 4R^3 & \text{pro } 0 \leq R < \frac{1}{2}, \\ \frac{4}{3} - 4R + 4R^2 - \frac{4}{3}R^3 & \text{pro } \frac{1}{2} \leq R < 1, \\ 0 & \text{pro } R \geq 1. \end{cases} \quad (20)$$

Jeho první derivace je:

$$W'(R, h) = \frac{3}{2\pi h^3} \begin{cases} -8R + 12R^2 & \text{pro } 0 \leq R < \frac{1}{2}, \\ -4 + 8R - 4R^2 & \text{pro } \frac{1}{2} \leq R < 1, \\ 0 & \text{pro } R \geq 1, \end{cases} \quad (21)$$

druhá je lomená a třetí nespojitá. Volba hladící funkce může ovlivnit přesnost řešení a je nutné jí věnovat zvýšenou pozornost, obzvláště v případech derivací vyšších řádů.



Obr. 3 — Kubický spline  $W(R, h)$  dle rovnice (20).

### 0.1.6 Umělá viskozita

V případě nadzvukových srážek se setkáváme s rázovými vlnami, což při omezeném rozlišení znamená skokovou změnu, čili nekonečné gradienty. Rovnice (17) až (19) pak bohužel divergují. Řeší se to „rozmazáním“ průběhu veličin pomocí uměle zvýšené viskozity<sup>5</sup> (Monaghan 1989):

$$\Pi_{ij} = \begin{cases} \frac{\alpha_{av} c_{s,ij} \mu_{ij} + \beta_{av} \mu_{ij}^2}{\rho_{ij}} & \text{pro } \mathbf{v}_{ij} \cdot \mathbf{r}_{ij} < 0 \\ 0 & \text{jinak,} \end{cases}$$

kde

$$\mu_{ij} = \frac{h \mathbf{v}_{ij} \cdot \mathbf{r}_{ij}}{|\mathbf{r}_{ij}|^2 + \epsilon h^2}$$

a parametry jsou obvykle  $\alpha_{av} = 1,5$ ,  $\beta_{av} = 3$ ;  $\epsilon = 0,01$  je malé číslo. Získáme tak konvergenci, ovšem za cenu ztráty rozlišení. Obvyklým testem je porovnání s analytickým řešením rázové trubice (Sod 1978).

<sup>5</sup> Alternativně bychom museli použít riemannovský algoritmus, přesně nebo přibližně řešící Rankinovy–Hugoniotovy rovnice (Godunov 1959, Roe 1981).



Členy, které musíme doplnit, jsou jednak  $-\sum_j m_j \Pi_{ij} \nabla_j W_{ij}$  do rovnice (18) a jednak  $+\sum_j m_j \Pi_{ij} \mathbf{v}_{ij} \cdot \nabla_j W_{ij}$  do (19), protože to, co se viskózně zbrzdí, se v zájmu zachování celkové energie trochu zahřeje.

### 0.1.7 Metoda *k*-d stromu

Nyní je třeba přiznat, že v metodě SPH je skryt zádrhel, vlastně dva: i) musíme hledat, které částice jsou v okolí všech částic, ii) gravitace je dalekodosahová síla, tudíž musíme beztak počítat přes všechny pro všechny. Obojí by vedlo k „brutálnímu“ algoritmu o složitosti  $\mathcal{O}(N^2)$ , což je zásadní omezení pro  $N \gg 10^3$ .

Naštěstí existuje algoritmus se složitostí  $\mathcal{O}(N \log_2 N)$ , nazývaný *k*-d strom, neboli *k*-rozměrný strom. V našem případě bude třírozměrný, ale kvůli názornosti je zde dvourozměrný. Namísto zdlouhavého slovního popisu jej rovnou implementujeme v objektovém jazyce Python. Nejprve si vytvoříme tři třídy — pro částici, uzel a strom, dle nichž se budou vytvářet jednotlivé objekty:

```
class Particle(object):
    def __init__(self, r):
        self.r = r

class Node(object):
    def __init__(self, particle, left, right):
        self.particle = particle
        self.left = left
        self.right = right

class Kdtree(object):
    def __init__(self, k, particles):
        self.k = k

    def build_tree(particles, depth=0):
        if len(particles) == 0:
            return None

        axis = depth % self.k
        particles.sort(key=lambda particle: particle.r[axis])
        i = len(particles) // 2 # median index

        return Node( \
            particles[i], \
            build_tree(particles[:i], depth+1), \
            build_tree(particles[i+1:], depth+1), \
        )

    self.root = build_tree(particles)

def main():
    r = [(2, 3), (5, 4), (9, 6), (4, 7), (8, 1), (7, 2)]
    particles = []
    for i in xrange(0, len(r)):
        particles.append(Particle(r[i]))
```

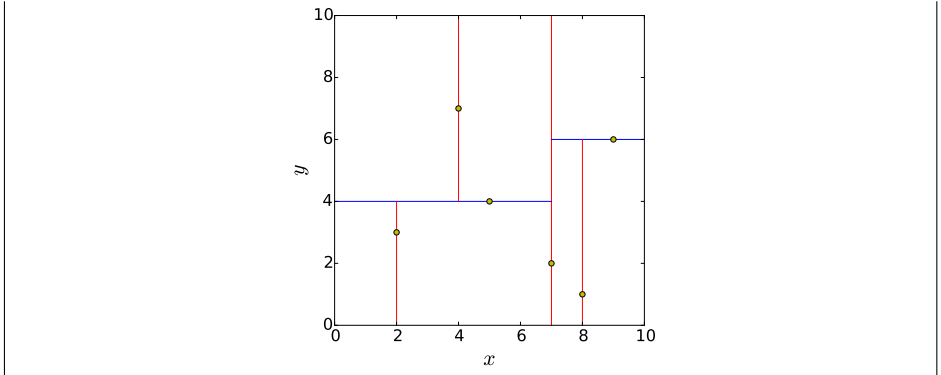
```

kdtree = Kdtree(2, particles)

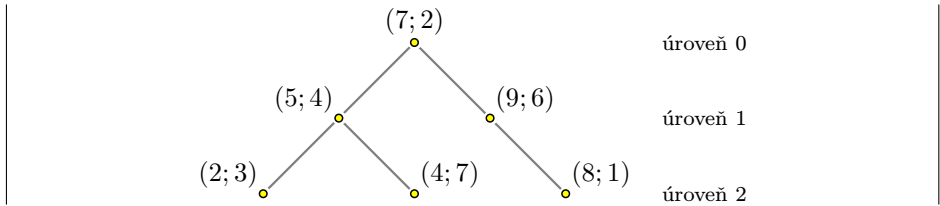
if __name__ == "__main__":
    main()

```

Co vidíme: na konci v hlavním programu se prostě zavolá hlavní funkce. V ní si připravíme seznam částic. Voláním `Particle` se vždy vytváří objekt částice. Objekty zde používáme proto, abychom později částicím mohli snadno připsat další vlastnosti, např. hmotnost, že. Voláním `Kdtree` vzniká objekt strom, přičemž se spouští příslušná metoda `__init__`, kde se definuje rekurzivní funkce `build_tree`. Ta třídí částice při prvním volání podle jejich souřadnice  $x$ , nalezne index mediánu  $i$  a voláním `Node` tvoří jeden uzel, sestávající z částice, levé větve (s částicemi majícími  $x < x_i$ ) a pravé větve ( $x > x_i$ ), kterýžto obratem vrátí. Při druhém volání třídí podle  $y$ , pak znovu dle  $x$ ,  $y$ , atd. až nezbude nic. Výsledný strom je nakreslen na obr. 4 a 5.



**Obr. 4** — Dvourozměrný strom vytvořený ze šesti (žlutých) částic. Svislé a vodorovné linie vyznačují, kde se strom rozvětvil podle souřadnice  $x$  (červeně) nebo  $y$  (modře). Jako první proběhlo větvení v bodě se souřadnicí  $x_i = 7$ .



**Obr. 5** — Jiné znázornění téhož stromu, na kterém je patrné větvení. Uzel v úrovni 0 se nazývá kořen, kdežto uzly bez větví v nejnižších úrovních (zde 2) se nazývají listy.

Pro hledání nejbližšího souseda bychom do třídy `Kdtree` doplnili vhodnou metodu, opět s jistou rekurzivní funkcí:

```

def nearest_neighbor(self, destination):
    best = [None, float('inf')] # particle & squared distance

    def recursive_search(node, depth=0):
        if node is None:
            return

        particle, left, right = node.particle, node.left, node.right
        node_sqdist = square_distance(particle.r, destination)
        if node_sqdist < best[1]:
            best[:] = particle, node_sqdist # in-place!

        axis = depth % self.k
        diff = destination[axis] - particle.r[axis]
        close, away = (left, right) if diff <= 0.0 else (right, left)

        recursive_search(close, depth+1)
        if diff**2 < best[1]:
            recursive_search(away, depth+1)

    recursive_search(self.root)
    return best[0], math.sqrt(best[1])
  
```

a volali ji jako `particle, dist = kdtree.nearest_neighbor((8, 5))`. Vrátila by pochopitelně `(9, 6)` a `1.4242...`<sup>6</sup>. Vidíme, že doslova na pár řádcích lze dosáhnout kýžené složitosti, respektive jednoduchosti. Úprava pro hledání  $N_{\text{okolo}}$  nejbližších částic jest triviální.

S výhodou můžeme strom použít i pro rychlý, ale jen *přibližný* výpočet gravitačního zrychlení. Prerekvizitou je, že si pro všechny uzly (rekurzivně) spočteme jejich celkovou hmotnost `node.tm`, polohu těžiště `node.cm` a kvadrát celkového rozměru `node.sqsize`. Pak si zvolíme *otevřací úhel*  $\varphi_{\text{open}}$  v radiánech a zrychlení spočteme (rekurzivně):

<sup>6</sup> Alternativně se místo stromu používá lineární seznam (angl. linked list).

---

```

def compute_gravity(self, destination, phi=0.5):
    phi2 = phi*phi
    eps = 1.e-16

    def recursive_gravity(node):
        dist2 = square_distance(node.cm, destination)
        ag = [0.0] * self.k

        if node.sqsize/(dist2+eps) < phi2: # do NOT open this node
            if dist2 > eps:
                tmp = node.tm/(dist2*math.sqrt(dist2))
                for i in xrange(0, self.k):
                    ag[i] = tmp*(node.cm[i]-destination[i])
            else:
                dist2 = square_distance(node.particle.r, destination)
                if dist2 > eps:
                    tmp = node.particle.m/(dist2*math.sqrt(dist2))
                    for i in xrange(0, self.k):
                        ag[i] = tmp*(node.particle.r[i]-destination[i])

                for branch in node.left, node.right:
                    if branch is not None:
                        accel = recursive_gravity(branch)
                        for i in xrange(0, self.k):
                            ag[i] += accel[i]

        return ag

    return recursive_gravity(self.root)

```

Když je uzel (se svými větvemi) malý nebo daleko, takže vytnutý úhel  $\varphi < \varphi_{\text{open}}$ , počítáme pouze gravitaci těžiště. V opačném případě uzel otevřeme a sčítáme gravitaci částice a obou větví. Čím menší  $\varphi_{\text{open}}$ , tím přesnější výsledek, ale při  $\varphi_{\text{open}} \rightarrow 0$  skončíme samozřejmě s  $\mathcal{O}(N^2)$ .

### 0.1.8 Multipólový rozvoj

Existuje ovšem možnost, jak výpočet gravitace zpřesnit i při zachování složitosti  $\mathcal{O}(N \log_2 N)$ . Namísto pouhého těžiště spočteme pro soubor  $N$  hmotných bodů vícero gravitačních momentů neboli *multipólů* (Stadel 2001, str. 8). Využijeme přitom *řádovou* notaci, kdy se podtržením označuje nikoli index nýbrž *počet* indexů:

$$M^n \equiv \sum_i^N m_i x_i^n. \quad (22)$$

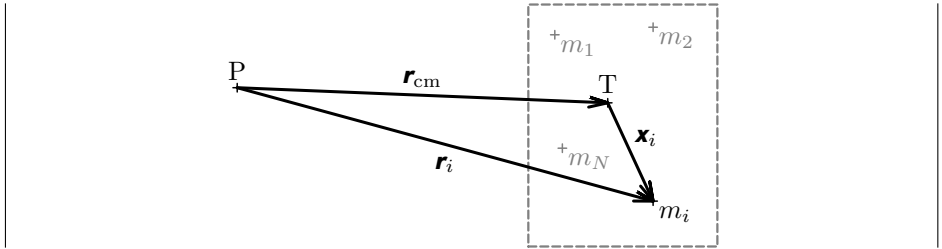
Pak  $M^0 \equiv M = \sum_i m_i$  je monopól, tj. skalár známý jako celková hmotnost,  $M^1 \equiv M^j = \sum_i m_i x_i^j$  dipól, vektor odpovídající těžišti (který bude nulový, budeme-li  $\mathbf{x}_i$  k těžišti  $\mathbf{r}_{\text{cm}}$  vztahovat, ale spočteme si jej pro kontrolu),  $M^2 \equiv M^{jk} = \sum_i m_i x_i^j x_i^k$

kvadrupól,  $M^{\underline{3}} \equiv M^{jkl}$  oktapól,  $M^{\underline{4}} \equiv M^{jklm}$  hexadekupól, příp. tenzory řádů ještě vyšších.

Gravitační potenciál souboru hmotných bodů posléze můžeme vyjádřit pomocí Taylorova rozvoje okolo těžiště (viz obr. 6):

$$\Phi = \sum_i \overbrace{-\frac{G}{|\mathbf{r}_i|}}^{\gamma(r_i)} m_i = \sum_i m_i \gamma(|\mathbf{r}_{\text{cm}} + \mathbf{x}_i|) = \sum_{n=0}^{\infty} \frac{1}{n!} [\partial_{\underline{n}} \gamma(r_{\text{cm}})] M^{\underline{n}}, \quad (23)$$

kde jsme promptně využili definice (22) a zkráceného zápisu  $\partial_{\underline{n}} \equiv \partial/\partial r^{\underline{n}}$ . Aproximace pochopitelně spočívá ve sčítání jen do  $n < \infty$ , např. 4.



**Obr. 6** — K výpočtu gravitačního potenciálu  $\Phi$  v bodě P, způsobovaného soustavou  $N$  hmotných bodů. Bod T označuje polohu těžiště této soustavy. Podle Stadel (2001).

Abychom nepočítali spočtené, měli bychom multipóly spočtené pro větve využít pro výpočet multipólu uzlu. Nejedná se ovšem o triviální součet, protože se posunuje poloha těžiště! Musíme proto použít Steinerovu větu, respektive její zobecnění v „šílené“ závorkové notaci, kdy závorky uzavřené přes podtržené indexy značí sčítání přes všechny unikátní permutace (Stadel 2001, str. 17):

$$M_{\text{posunuto}}^{\underline{n}} = \sum_{m=0}^n M^{(n-m)d^{\underline{m}}}, \quad (24)$$

kde vektor  $d^j$  je onen posun. Při implementaci je však vše nutné rozepisovat...

### 0.1.9 Počáteční a okrajové podmínky

Vytvoření počátečních podmínek pro simulace srážek je jednoduché — spočívá generování požadovaného počtu částic SPH, jejich rozmístění uvnitř koule, resp. simulovaného tělesa, přiřazení počátečních  $\rho_0$ ,  $\mathbf{v}_0$ ,  $U_0$ . Často se používá rovnoměrné rozmístění částic; nelze použít zcela náhodné, protože by vznikaly nereálné velké gradienty. Možná je optimální něco mezi tím (viz Diehl aj. 2012).

---

Okrajové podmínky jsou ještě jednodušší, neboť okolo je vakuum, čili nemusíme dělat nic. Ostatně metoda SPH je pro tento případ zvláště vhodná.

V opačném (horším) případě bychom museli vytvářet *virtuální částice* s vhodnými vlastnostmi, které by vymezovaly hranici. Připisuje se jim odpudivá síla, která brání unikání normálních částic z daného prostoru. Druhou možností je použití *duchařských částic*, které aktuální stav částic uvnitř prostoru zrcadlí vně, pouze rychlosti jsou opačné.

### 0.1.10 Fragmentační fáze

Po kontaktu projektilu a terče začíná první fáze srážky, a to hydrodynamická fragmentace terče. Pro terče o velikosti planetek ( $D \lesssim 100$  km) lze simulaci provádět metodou SPH (Benz a Asphaug 1994) bez vlivu gravitace, tzn. bez řešení Poissonovy rovnice.<sup>7</sup> Požadujeme ale, aby její trvání odpovídalo přinejmenším době průchodu rázové vlny terčem tam a zpět:

$$\Delta t_{\min} = \frac{2D}{v_{\text{imp}}} \simeq 40 \text{ s}.$$

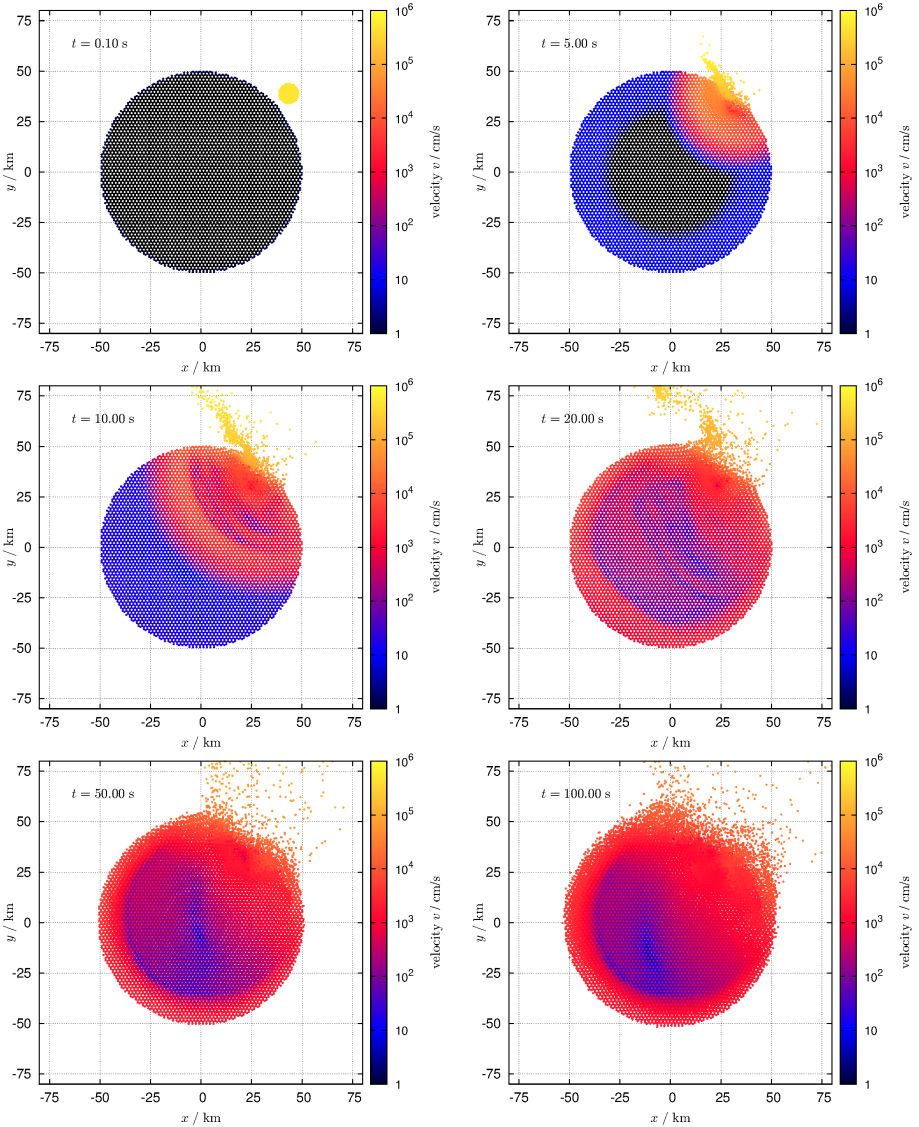
Naopak nejdelší přípustná doba je dána poměrem typické rychlosti výhozu  $v_{\text{ej}}$ , což přirozeně bývá  $v_{\text{esc}} = \sqrt{2GM/R}$ , a gravitačního zrychlení  $a_{\text{g}} = GM/R^2$ , čili:

$$\Delta t_{\max} = \frac{v_{\text{esc}}}{a_{\text{g}}} = \sqrt{\frac{3}{2\pi G\rho}} \simeq 10^3 \text{ s}.$$

Průběh jedné takové simulace zachycuje obr. 7.

---

<sup>7</sup> Pro impakty planetárních rozměrů si to ovšem dovolit nemůžeme (Canup 2004).



**Obr. 7** — Simulace srážky mezi terčem o průměru  $D = 100$  km a projektilem s  $D = 10$  km, s impaktní rychlostí  $v_{\text{imp}} = 5$  km/s a úhlem  $\varphi_{\text{imp}} = 45^\circ$ . Graf ukazuje polohy  $(x, y)$  přibližně  $1,42 \cdot 10^5$  SPH částic; jejich barvy odpovídají logaritmu rychlosti  $v$  (v jednotkách  $\text{cm s}^{-1}$ ). Integrace byla řízena Courantovým číslem  $C = 1,0$ , typický časový krok byl tedy  $\Delta t \simeq 10^{-4}$  s, trvání  $t_{\text{stop}} = 100$  s. Materiálové parametry pro bazalt (Benz and Asphaug 1999): hustota při nulovém tlaku  $\rho_0 = 2,7 \text{ g cm}^{-3}$ , objemový modul  $A = 2,67 \cdot 10^{11} \text{ erg cm}^{-3}$ , nelineární tlakový

člen  $B = 2,67 \cdot 10^{11} \text{ erg cm}^{-3}$ , Tillotsonovy parametry  $U_0 = 4,87 \cdot 10^{12} \text{ erg g}^{-1}$ ,  $a = 0,5$ ,  $b = 1,5$ ,  $\alpha = 5,0$ , a  $\beta = 5,0$ , počínající vypařování  $U_{iv} = 4,72 \cdot 10^{10} \text{ erg g}^{-1}$ , úplné vypaření  $U_{cv} = 1,82 \cdot 10^{11} \text{ erg g}^{-1}$ , modul pružnosti ve smyku  $\mu = 2,27 \cdot 10^{11} \text{ erg cm}^{-3}$ , mez pružnosti  $Y = 3,5 \cdot 10^{10} \text{ erg g}^{-1}$ , měrná energie tavení  $U_{melt} = 3,4 \cdot 10^{10} \text{ erg g}^{-1}$ , parametry Weibullova rozdělení poruch  $k = 4,0 \cdot 10^{29} \text{ cm}^{-3}$ , a  $m = 9,0$ . Počítáno programem SPH5 (Benz and Asphaug 1994).

### 0.1.11 Reakumulační fáze

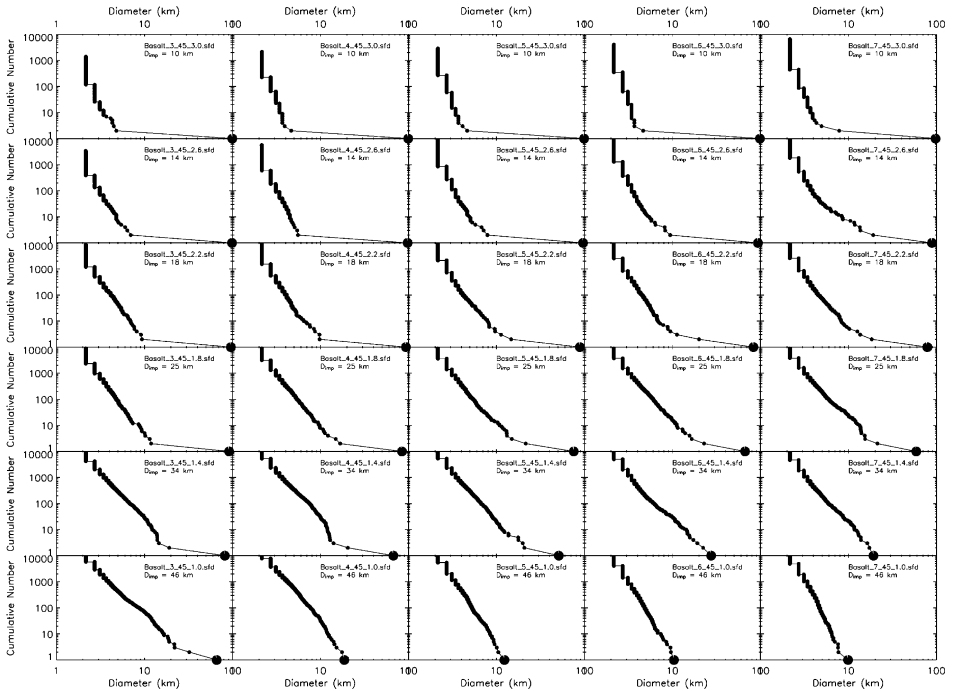
Druhou navazující fází je gravitační reakumulace. Tu je možno v prvním přiblížení počítat naopak *bez* jakékoliv hydrodynamiky, jako čistě gravitační interakci fragmentů počítanou  $N$ -částicově (Richardson et al. 2000). Výsledek hydrodynamického výpočtu se ovšem musí nějak transformovat do podoby počátečních podmínek pro pevné částice. Dalšími obvyklými zjednodušujícími předpoklady jsou: i) všechno jsou koule; ii) při prvním dotyku se koule spojují do větší koule. Ztrácíme tím samozřejmě jakékoliv informace o rotaci nebo tvaru fragmentů. Počáteční poloměry koulí mohou být vypočteny z vlastností částic SPH jako:

$$R_i = \frac{3m_i}{4\pi\rho_i},$$

přičemž částice zplyněné ( $U_i \gtrsim U_{cv}$ ) se zahazují. Navíc, aby byla úloha výpočetně zvládnutelná, používá se výše uvedená přibližná metoda  $k$ -d stromu, čili působení vzdálených částic se nepočítá jednotlivě, ale hromadně.

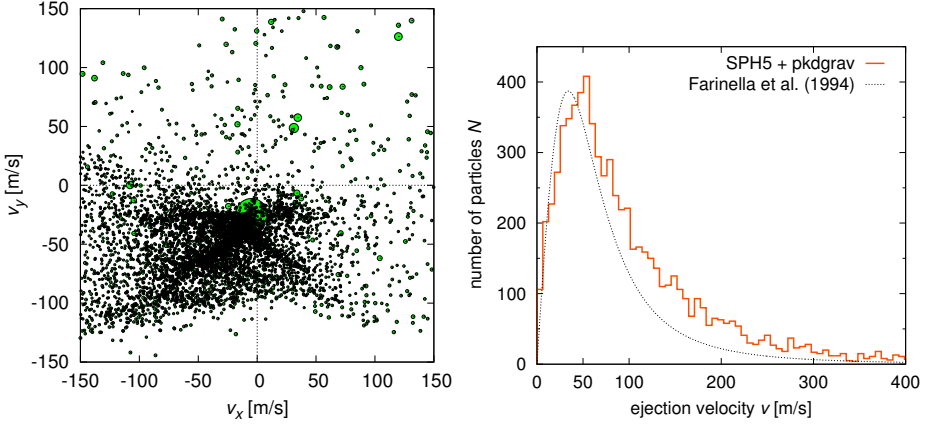
Výsledek celé simulace (obr. 9) lze porovnávat například s pozorovaným rozdělením velikostí členů dané rodiny asteroidů (Durda aj. 2007, Benavidez aj. 2012), případně s pozorovaným rychlostním polem (Nesvorný aj. 2006). Musíme samozřejmě uvážit, že obojí mohlo být změněno při dlouhodobém orbitálním vývoji rodiny a sekundárních srážkách.





**Obr. 8** — Kumulativní rozdělení velikostí  $N(>D)$  pro podmnožinu 30 různých simulací srážek monolitických asteroidů s průměrem terče  $D = 100$  km a impaktním úhlem  $\phi_{\text{imp}} = 45^\circ$ . Jednotlivé simulace se liší rozměrem projektilu  $d = 10, 14, 18, 25, 34$  a  $46$  km (směrem shora dolů) a impaktní rychlostí  $v_{\text{imp}} = 3, 4, 5, 6$  a  $7$  km/s (zleva doprava). To odpovídá rozsahu od pouhého kráterování po katastrofický rozpad, což se projevuje i zřetelně odlišnými histogramy  $N(>D)$ .

Převzato z Durda aj. (2007).



**Obr. 9** — Vlevo konečné rychlosti těles po simulaci srážky terče  $D = 100$  km a projektilu  $d = 25$  km, s rychlostí  $v_{\text{imp}} = 5 \text{ km s}^{-1}$  a úhlem  $\phi_{\text{imp}} = 45^\circ$ . Velikosti symbolů odpovídají hmotnostem. Vpravo odpovídající diferenciální histogram rychlostí výhozu  $dN(v_{\text{ej}})$ , v porovnání s jednoduchým modelem Farinelly aj. (1994). Vidíme, že vrchol histogramu je na hodnotě srovnatelné s únikovou rychlostí  $v_{\text{esc}}$  původního mateřského tělesa. Počítáno programem Pkdgrav (Richardson aj. 2000).

### 0.1.12 Škálovací zákon pro terče

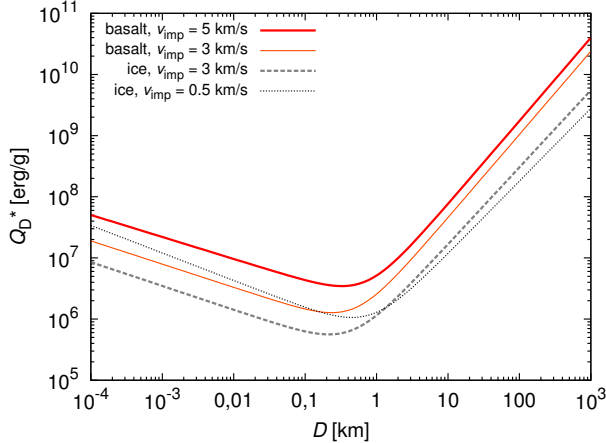
Z celé sady simulací lze odvodit škálovací zákon  $Q_D^*(r)$  (Benz a Asphaug 1999), čili závislost pevnosti (v jednotkách  $\text{J kg}^{-1}$ ) na poloměru  $r$ , která je nejdůležitějším vstupem pro statistické modely srážek (viz kap. ??). Stačí pomocí interpolace zjistit, při jaké kinetické energii  $E_k$  projektilu dochází k rozptýlení poloviny hmotnosti terče. Průběh funkce lze vystihnout polynomem:

$$Q_D^* = \frac{1}{q_{\text{fact}}} (Q_0 r^a + B \rho r^b), \quad (25)$$

kde hodnoty parametrů pro různé materiály shrnuje tab. 1 a funkci obr. 10. Faktor  $q_{\text{fact}}$  se užívá pouze tehdy, chceme-li vyjádřit, že pevnost je  $q$  krát menší než nějaký standardní materiál; jinak nemá smysl. Všimněme si, že  $a$  je záporné a  $b$  kladné. Důvodem prvního je skutečnost, že v malých balvanech je málo poruch (prasklin), a tudíž jsou pevnější. Tato část škálovacího zákona je ostatně ověřena laboratorními experimenty. Důvodem druhého je rostoucí gravitační vazebná energie  $E_v$ , ale závislost nebývá tak strmá jako  $b = -2$ , neboť nerozptylujeme celý terč, nýbrž pouze nějakou polovinu.

	$v_{\text{imp}}$	$Q_0$	$B$	$a$	$b$
	km s	erg g <sup>-1</sup>	erg cm <sup>3</sup> g <sup>-2</sup>		
basalt	5	$9,0 \cdot 10^7$	0,5	-0,36	1,36
led	3	$1,6 \cdot 10^7$	1,2	-0,39	1,26

**Tab. 1** — Parametry škálovacího zákona (25) pro basaltický a ledový materiál s monolitickou strukturou. Převzato z Benz a Asphaug (1999).



**Obr. 10** — Škálovací zákon  $Q_D^*$ , neboli závislost pevnosti na průměru  $D$ , pro monolitický basalt a led a různé hodnoty impaktní rychlosti  $v_{\text{imp}}$ . Podle Benz a Asphaug (1999).

- [1] BENAVIDEZ, P. G., DURDA, D. D., ENKE, B. L., BOTTKÉ, W. F., NESVORNÝ, D., RICHARDSON, D. C., ASPHAUG, E., MERLINE, W. J. *A comparison between rubble-pile and monolithic targets in impact simulations: Application to asteroid satellites and family size distributions.* *Icarus*, **219**, 57, 2012.
- [2] BENZ, W., ASPHAUG, E. *Impact simulations with fracture. I. Method and tests.* *Icarus*, **107**, s. 98–116, 1994.
- [3] BENZ, W., ASPHAUG, E. *Catastrophic disruptions revisited.* *Icarus*, **142**, s. 5–20, 1999.
- [4] CANUP, R. M. *Simulations of a late lunar-forming impact.* *Icarus*, **168**, 433, 2004.
- [5] COSSINS, P. *Smoothed particle hydrodynamics.* *ArXiv*, 1007.1245v2, 2010.
- [6] DIEHL, S., ROCKEFELLER, G., FRYER, C. L., RIETHMILLER, R., STATLER, T.S. *Generating optimal initial conditions for Smooth Particle Hydrodynamics simulations.* *ArXiv*, 1211.0525, 2012.
- [7] DURDA, D. D., BOTTKÉ, W. F., NESVORNÝ, D., ENKE, B. L., MERLINE, W. J., ASPHAUG, E., RICHARDSON, D. C. *Size-frequency distributions of fragments from SPH/N-body simulations of asteroid impacts: Comparison with observed asteroid families.* *Icarus*, **186**, 498, 2007.
- [8] FARINELLA, P., FROESCHLÉ, C., GONCZI, R. *Meteorite delivery and transport.* *IAUS Conf.*, **160**, 205, 1994.

- 
- [9] GINGOLD, R. A., MONAGHAN, J. J. *Smoothed particle hydrodynamics — Theory and application to non-spherical stars*. *Mon. Not. R. Astr. Soc.*, **181**, s. 375–389, 1977.
- [10] GODUNOV, S. K. *A difference scheme for numerical computation of discontinuous solution of hyperbolic equation*. *Math. Sbornik*, **47**, s. 271–306, 1959.
- [11] LUCY, L. B. *A numerical approach to the testing of the fission hypothesis*. *Astron. J.*, **82**, s. 1013–1024, 1977.
- [12] MAINDL, T. I., SCHÄRFER, CH., SPEITH, R. *SPH for simulating impacts and collisions* [online]. [cit. 2014-08-12], (<http://www.univie.ac.at/adg/Teaching/Planetformation2012/NFNSPH2013-01-22.pdf>), 2013.
- [13] MELOSH, H. J. *A new and improved equation of state for impact computations*. *LPI Conf.*, **31**, 1903, 2000.
- [14] MONAGHAN, J. *On the problem of penetration in particle methods*. *J. Comput. Phys.*, **82**, 1, s. 1–15, 1989.
- [15] NESVORNÝ, D., ENKE, B. L., BOTTKE, W. F., DURDA, D. D., ASPHAUG, E., RICHARDSON, D. C. *Karin cluster formation by asteroid impact*. *Icarus*, **183**, s. 296–311, 2006.
- [16] NIEDOBA, P. *Bezsíťové metody ve výpočetní dynamice tekutin*. Diplomová práce, VUT Brno, 2012.
- [17] PRICE, D. J. *Smoothed particle hydrodynamics and magnetohydrodynamics*. *J. Comp. Phys.*, **231**, 759, 2012.
- [18] RICHARDSON, D. C., QUINN, T., STADEL, J., LAKE, G. *Direct large-scale N-Body simulations of planetesimal dynamics*. *Icarus*, **143**, 45, 2000.
- [19] ROE, P. L. *Approximate Riemann solvers, parameter vectors and difference schemes*. *J. Comput. Phys.*, **43**, s. 357–372, 1981.
- [20] SOD, G. A. *A survey of several finite difference methods for systems of nonlinear hyperbolic conservation laws*. *J. Comput. Phys.*, **27**, s 1–31, 1978.
- [21] STADEL, J. G. *Cosmological N-body simulations and their analysis*. Ph.D. Thesis, Univ. of Washington, 2001.
- [22] TILLOTSON, J. H. *Metallic equations of state for hypervelocity impact*. *General Atomic Report*, GA-3216, 1962.

# Literatura

Učebnice

Reference