SPH kernels in SWIFT

Matthieu Schaller

May 18, 2013

1 General Definitions

The smoothing kernels used in SPH are almost always isotropic and can hence be written in 3D as

$$W(\vec{x}, h) = \frac{1}{h^3} f\left(\frac{|\vec{x}|}{h}\right), \tag{1}$$

where $f(q)$ is a dimensionless function, usually a low-order polynomial, normalized to unity. For computational reasons, this kernel usually has a finite support of radius H. In other words,

$$W(\vec{x}, h) = 0 \quad \forall \quad |\vec{x}| > H. \tag{2}$$

One can then define the weighted number of neighbours within H as

$$N_{ngb} = \frac{4}{3}\pi H^3 \sum_j W(\vec{x}_i - \vec{x}_j, h). \tag{3}$$

The value of N_{ngb} is often used in the codes to find the smoothing length of each particle via Newton iterations or a bissection algorithm. H is defined as *the smoothing length* in the GADGET code. This definition is useful for implementation reasons but does not really correspond to a true physical quantity.
The main question is the definition of the smoothing length. The function $W(\vec{x}, h)$ is invariant under the rescaling $h \rightarrow \alpha h$, $f(q) \rightarrow \alpha^{-3} f(\alpha q)$, which makes the definition of h difficult. This ambiguity is present in the litterature with authors using different definition of the *physical* smoothing length, $h = \frac{1}{2}H$ or $h = H$ for instance.
A more physically motivated estimate is the standard deviation of the kernel:

$$\sigma^2 = \frac{1}{3} \int \vec{x}^2 \; W(\vec{x}, h) \; d^3\vec{x} \tag{4}$$

which then allows us to set $h = 2\sigma$. This definition of the smoothing length is more physical as one can demonstrate that the reconstruction of any smooth field $A(\vec{x})$ using interpolation of particles at the point $\vec{x}_i$ can be expanded as

$$A_i \approx A(\vec{x}_i) + \frac{1}{2}\sigma^2 \nabla^2 A(\vec{x}_i) + \mathcal{O}\left(\sigma^4\right). \tag{5}$$

The quantity H/σ is independant of the choice of h made and is purely a functional of $f(q)$. The number of neighbours (used in the code to construct the

neighborhood of a given particle) can then be expressed as a function of this *physical* h (or σ). Or to relate it even more to the particle distribution, we can write $h = \eta\Delta x$, with Δx the mean inter-particle separation:

$$N_{ngb} = \frac{4}{3}\pi\left(\frac{1}{2}\eta\frac{H}{\sigma}\right)^3 = \frac{4}{3}\pi\left(\eta\zeta\right)^3 \tag{6}$$

This definition of the number of neighbours only depends on $f(q)$ (via ζ) and on the mean inter-particle separation. The problem is then fully specified by specifying a form for $f(q)$ and η.
Experiments suggest that $\eta \approx 1.2-1.3$ is a reasonnable choice. The bigger η, the better the smoothing and hence the better the reconstruction of the field. This, however, comes at a higher computational cost as more interactions between neighbours will have to be computed. Also, spline kernels become instable when $\eta > 1.5$.

2 Kernels available in SWIFT

The different kernels available are listed below.

Cubic Spline Kernel

$$f(q) = \frac{1}{\pi}\begin{cases} \frac{3}{4}q^3 - 15q^2 + 1 & \text{if} \quad 0 \leq q < 1 \\ -\frac{1}{4}q^3 + \frac{3}{2}q^2 - 3q + 2 & \text{if} \quad 1 \leq q < 2 \\ 0 & \text{if} \quad q \geq 2 \end{cases}$$

with $\zeta = \frac{1}{2}\sqrt{\frac{40}{3}} \approx 1.825742$. Thus, for a resolution of $\eta = 1.235$, this kernel uses $N_{ngb} \approx 48$. The code uses $h = \frac{1}{2}H = \zeta\sigma$ internally.

Quartic Spline Kernel

$$f(q) = \frac{1}{20\pi}\begin{cases} 6q^4 - 15q^2 + \frac{115}{8} & \text{if} \quad 0 \leq q < \frac{1}{2} \\ -4q^4 + 20q^3 - 30q^2 + 5q + \frac{55}{4} & \text{if} \quad \frac{1}{2} \leq q < \frac{3}{2} \\ q^4 - 10q^3 + \frac{75}{2}q^2 - \frac{125}{2}q + \frac{625}{16} & \text{if} \quad \frac{3}{2} \leq q < \frac{5}{2} \\ 0 & \text{if} \quad q \geq \frac{5}{2} \end{cases}$$

with $\zeta = \frac{1}{2}\sqrt{\frac{375}{23}} \approx 2.018932$. Thus, for a resolution of $\eta = 1.235$, this kernel uses $N_{ngb} \approx 64.9$. The code uses $h = \frac{2}{5}H = \frac{4}{5}\zeta\sigma$ internally.

Quintic Spline Kernel

$$f(q) = \frac{1}{120\pi}\begin{cases} -10q^5 + 30q^4 - 60q^2 + 66 & \text{if} \quad 0 \leq q < 1 \\ 5q^5 - 45q^4 + 150q^3 - 210q^2 + 75q + 51 & \text{if} \quad 1 \leq q < 2 \\ -q^5 + 15q^4 - 90q^3 + 270q^2 - 405q + 243 & \text{if} \quad 2 \leq q < 3 \\ 0 & \text{if} \quad q \geq 3 \end{cases}$$

with $\zeta = \frac{1}{2}\sqrt{\frac{135}{7}} \approx 2.195775$. Thus, for a resolution of $\eta = 1.235$, this kernel uses $N_{ngb} \approx 83.5$. The code uses $h = \frac{1}{3}H = \frac{2}{3}\zeta\sigma$ internally.

Wendland $C2$ Kernel

$$f(q) = \frac{21}{2\pi} \begin{cases} 4q^5 - 15q^4 + 20q^3 - 10q^2 + 1 & \text{if} \quad 0 \le q < 1 \\ 0 & \text{if} \quad q \ge 1 \end{cases}$$

with $\zeta = \frac{1}{2}\sqrt{15} \approx 1.93649$. Thus, for a resolution of $\eta = 1.235$, this kernel uses $N_{ngb} \approx 57.3$. The code uses $h = H = 2\zeta\sigma$ internally.

Wendland $C4$ Kernel

$$f(q) = \frac{495}{32\pi} \begin{cases} \frac{35}{3}q^8 - 64q^7 + 140q^6 - \frac{448}{3}q^5 + 70q^4 - \frac{28}{3}q^2 + 1 & \text{if} \quad 0 \le q < 1 \\ 0 & \text{if} \quad q \ge 1 \end{cases}$$

with $\zeta = \frac{1}{2}\sqrt{\frac{39}{2}} \approx 2.20794$. Thus, for a resolution of $\eta = 1.235$, this kernel uses $N_{ngb} \approx 84.9$. The code uses $h = H = 2\zeta\sigma$ internally.

Wendland $C6$ Kernel

$$f(q) = \frac{1365}{64\pi} \begin{cases} 32q^{11} - 231q^{10} + 704q^9 - 1155q^8 + 1056q^7 - 462q^6 + 66q^4 - 11q^2 + 1 & \text{if} \quad 0 \le q < 1 \\ 0 & \text{if} \quad q \ge 1 \end{cases}$$

with $\zeta = \frac{1}{2}\sqrt{24} \approx 2.44949$. Thus, for a resolution of $\eta = 1.235$, this kernel uses $N_{ngb} \approx 116$. The code uses $h = H = 2\zeta\sigma$ internally.

3 Kernel Derivatives

The derivatives of the kernel function have relatively simple expressions:

$$\begin{aligned} \vec{\nabla}_x W(\vec{x}, h) &= \frac{1}{h^4} f'\left(\frac{|\vec{x}|}{h}\right) \frac{\vec{x}}{|\vec{x}|} \\ \frac{\partial W(\vec{x}, h)}{\partial h} &= -\frac{1}{h^4}\left[3f\left(\frac{|\vec{x}|}{h}\right) + \frac{|\vec{x}|}{h} f'\left(\frac{|\vec{x}|}{h}\right)\right] \end{aligned}$$

Note that for all the kernels listed above, $f'(0) = 0$.