Astronomy & Astrophysics manuscript no. notes
August 20, 2019

Digital filtering

J. Nättilä[1]

Nordita, KTH Royal Institute of Technology and Stockholm University, Roslagstullsbacken 23, SE-10691 Stockholm, Sweden. e-mail: joonas.nattila@su.se

Received XXX / Accepted XXX

ABSTRACT

Notes about digital (and frequency space) current filtering on particle-in-cell simulations.

Key words. Plasmas – Turbulence – Methods: numerical

1. Current filtering

See **?** for a general discussion about filters. Here we will mainly focus on linear filters.

1.1. Fourier-space filtering

Convolution of an image f with kernel w (also known as filter mask) to produce a new image g can be expressed as

$$g = f * w = \int_{\infty}^{\infty} f(\tau)(t-\tau) d\tau. \quad (1)$$

Discrete direct convolution, $g = f \circledast w$, valid for images that are defined on a grid can be expressed as

$$g_{ij} = \sum_{k=-\frac{n}{2}}^{\frac{n}{2}-1} \sum_{l=-\frac{n}{2}}^{\frac{n}{2}-1} w_{kl} f_{i+k,j+l}, \quad (2)$$

for $i, j = 1, \ldots, n$ where $(i+k)$ is replaced by $(i+k \pm n)$, if it falls outside the range from 1 to n. Similar transformation needs to be applied to $(j+l)$. This is the so-called cyclic (wrapped) direct convolution. In this case, the design of filters reduces to finding expressions for the kernel w.

Fourier transform of f, g, and w satisfy

$$g^*_{kl} = w^*_{kl} f^*_{kl}, \quad (3)$$

for $k, l = 1, \ldots, n$, provided that w and f are arrays of the same size. Therefore, filtering in the spatial domain can be expressed with a simple point-by-point multiplication in the frequency domain. Typically different sources claim that it is beneficial to use Fourier domain for filtering if the kernel is bigger than 7×7. This allows us to design filters in the frequency domains, finding expressions for the w^*.

Ideal low-pass filter is given by setting w^*_{kl} to zero beyond certain distance R from the origin, and otherwise to unity as

$$w^*_{kl} = \begin{cases} 1 & \text{if } k^2 + l^2 < R^2, \\ 0 & \text{otherwise,} \end{cases} \quad (4)$$

for $k, l = -\frac{n}{2}, \ldots, \frac{n}{2} - 1$. This will, however, lead to ringing, as is evident from the spatial-domain weights of w that have both negative and positive values.

A filter that minimizes ringing is a Gaussian

$$w_{ij} = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{ \frac{-i^2 + j^2}{2\sigma^2} \right\}, \quad (5)$$

where σ is the standard deviation or a width of the filter in units of pixels. Fourier transform of the Gaussian weights can be expressed by

$$w^*_{kl} = \exp\left\{ \frac{-k^2 + l^2}{2[n/(2\pi\sigma)]^2} \right\}, \quad (6)$$

that is a scaled version of the spatial domain Gaussian, but with a different spread.

In conclusion, a good filter should be resembling a Gaussian as close as possible.

1.2. Digital filtering

Digital filtering describes the process of filtering a quantity on the x-space (withouth any Fourier transforms to k-space). Hamming1977 Collatz1966 (Birdsall & Langdon 1985)

Given a grid quantity $X_i \equiv X(i\Delta x)$ a simple digital filtering is performed by replacing

$$X_i \leftarrow \frac{W X_{i-1} + X_i + W X_{j+1}}{1 + 2W}, \quad (7)$$

where W is the filter weight. Caution is needed to make sure that points on the right-hand side are the original values.

Different compact filters can be obtained by selecting different weightings. A filter with $W = \frac{1}{2}$ is called a binomial filter and has a positive smoothing function in k-space. Selecting $W < 0$ produces a so-called compensation filter as it enhances the original signal. $W = -\frac{1}{6}$ gives a compensation that cancels second-order attenuation of a $W = \frac{1}{2}$ filter, i.e., applying a $W = \frac{1}{2}$ filter followed by a $W = -\frac{1}{6}$ produces a fourth-order attenuated filter corresponding to a broader stencil with weights of $(1/16)[-1 \quad 4 \quad 10 \quad 4 \quad -1]$.

All of the filtering operations above need a temporary scratch array of the quantity being processed. A scratch-memory-free alternatives can be derived by assuming multiple passes with different filters. Simplest one can be obtained by replacing X_i with a forward-pass of $X_i + U X_{i+1}$

and a backward-pass of $X_i + UX_{i-1}$, for a multi-pass filter parameter U. This is equal to the one-pass filter for

$$W = \frac{U}{1+U^2} \tag{8}$$

or

$$U = \frac{1}{2W} \pm \sqrt{\frac{1}{(2W)^2} - 1}. \tag{9}$$

For U to be real valued, we require $-\frac{1}{2} \leq W \leq \frac{1}{2}$. Binomial two-pass filter (equal to $\frac{1}{4}[1 \quad 2 \quad 1]$; $W = \frac{1}{2}$) corresponds to $U = 1$ with a forward-pass of $\frac{1}{2}[0 \quad 1 \quad 1]$ and a return-pass with $\frac{1}{2}[1 \quad 1 \quad 0]$. The corresponding two-pass compensator ($W = -\frac{1}{6}$) has $U = -3 + \sqrt{8} \approx -0.171573$ to be applied after the smoothing.

Two-dimensional filters can be generated by using two one-dimensional filters (shown here for the second-order Binomial filter B_2) as

$$B_2^{(2D)} \equiv B_2 \otimes B_2^T = \frac{1}{4}[1 \quad 2 \quad 1] \otimes \frac{1}{4}\begin{bmatrix}1\\2\\1\end{bmatrix} = \frac{1}{16}\begin{bmatrix}1 & 2 & 1\\2 & 4 & 1\\1 & 2 & 1\end{bmatrix}. \tag{10}$$

A scratch-memory-free version also exists in a form of a four-pass filter (consisting of $C_\rightarrow \equiv \frac{1}{2}[0 \quad 1 \quad 1]$ and $C_\leftarrow \equiv \frac{1}{2}[1 \quad 1 \quad 0]$) as

$$B_2^{(2D)} = C_\rightarrow \otimes C_\leftarrow \otimes C_\rightarrow^T \otimes C_\leftarrow^T \tag{11}$$

$$= \frac{1}{2}\begin{bmatrix}0 & 0 & 0\\0 & 1 & 1\\0 & 0 & 0\end{bmatrix} \frac{1}{2}\begin{bmatrix}0 & 0 & 0\\1 & 1 & 0\\0 & 0 & 0\end{bmatrix} \frac{1}{2}\begin{bmatrix}0 & 1 & 0\\0 & 1 & 0\\0 & 0 & 0\end{bmatrix} \frac{1}{2}\begin{bmatrix}0 & 0 & 0\\0 & 1 & 0\\0 & 1 & 0\end{bmatrix}. \tag{12}$$

A possible choice for the compensator can be $(20, -1, -1)$ or $(36, -6, 1)$ check Birdsall filter notation assumed to be (center, middle, corner)

$$\begin{bmatrix}-1 & -1 & -1\\-1 & 20 & -1\\-1 & -1 & -1\end{bmatrix} \quad \text{or} \quad \begin{bmatrix}1 & -6 & 1\\-6 & 36 & -6\\1 & -6 & 1\end{bmatrix}. \tag{13}$$

Here, one should, however, note that scratch-free versions can not be vectorized, which means that for modern processing units are probably not the optimal choice.

From the numerical point-of-view the filtering is always a balance between number of passes and size of the filter stencil. It also matters in what order we are asking the variables from the array; a modern processor has a prefetcher that can try to guess our memory access patterns. A suitable selection for the $2D$ seems to be a two-pass filtering (up and down) with the compact 3-point binomial (requiring 3 values, X_{i-1}, X_i, and X_{i+1}). A modern prefetcher might even be able to cope with a one-pass compact Binomial (requiring 9 values in $2D$).

Note also that thanks to the usage of tiles (with a lenght of 3 buffer zones for the current arrays), it can make sense to combine multiple compact Binomial passes into a fewer passes with an extended stencil (supporting a 6 order stencil with 7 [3+1+3] points with one MPI communication round).

1.3. Binomial filters

Binomial filters form a compact approximation for the (discretized) Gaussian.[1] Hamming 1977 Binomial coefficients are given by

$$\binom{N}{n} = \frac{N!}{(N-n)!N!}, \tag{14}$$

where $n = 0, \ldots, N$ and N is the order of the binomial filter B_N. It forms a rapid approximation for the Gaussian distribution as $\sigma \approx \sqrt{N}/2$. [2] First few are given as

$$B_0 = [1] \tag{15}$$

$$B_1 = \frac{1}{2}[1 \quad 1] \tag{16}$$

$$B_2 = \frac{1}{4}[1 \quad 2 \quad 1] \tag{17}$$

$$B_3 = \frac{1}{8}[1 \quad 3 \quad 3 \quad 1] \tag{18}$$

$$B_4 = \frac{1}{16}[1 \quad 4 \quad 6 \quad 4 \quad 1] \tag{19}$$

$$B_5 = \frac{1}{32}[1 \quad 5 \quad 10 \quad 10 \quad 5 \quad 1] \tag{20}$$

$$B_6 = \frac{1}{64}[1 \quad 6 \quad 15 \quad 20 \quad 15 \quad 6 \quad 1] \tag{21}$$

$$\vdots \tag{22}$$

The binomial filter coefficients are applied for a quantity X (defined on a grid as $X_i \equiv X(i\Delta x)$ as

$$X_i^{f_2} = B_2\{X_i\} = \frac{1}{4}X_{i-1} + \frac{1}{2}X_i + \frac{1}{4}X_{i+1}, \tag{23}$$

where $X_i^{f_2}$ the filtered quantity and $B_2\{X\}$ denotes an application of a B_2 filter on the original quantity. Applying higher-order binomial filters generalizes naturally from here.

Discrete-time Fourier transform of the binomial filter $B_2 = \frac{1}{4}[1 \quad 2 \quad 1]$ is proportional to a single period of a cosine as

$$B_2^*[k] = \sum_n B_2[n] \cos[kn] = \frac{1}{2} + \frac{1}{2}\cos[k]. \tag{24}$$

For higher-order filters of rank N we similarly have

$$B_N^*[k] = \left(\frac{1}{2} + \frac{1}{2}\cos(k)\right)^N. \tag{25}$$

1.4. Temporal filters

Another possibility, besides usage of spatial filters, is to filter electric field in time. This is known as the Friedman filter, defined as

$$\boldsymbol{E}^{(n)} = \left(1 + \frac{\theta}{2}\right)\boldsymbol{E}^{(n)} - \left(1 - \frac{\theta}{2}\right)\boldsymbol{E}^{(n-1)} + \frac{1}{2}(1-\theta)^2\bar{\boldsymbol{E}}^{(n-2)}, \tag{26}$$

where $\bar{\boldsymbol{E}}^{(n-1)} = \boldsymbol{E}^{(n-2)} + \theta\boldsymbol{E}^{(n-3)}$, and the filtering parameter $\theta \in [0, 1]$.

References

Birdsall, C. & Langdon, A. 1985, Plasma physics via computer simulation, The Adam Hilger series on plasma physics (McGraw-Hill, New York)

[1] `http://www.cse.yorku.ca/~kosta/CompVis_Notes/binomial_filters.pdf`

[2] Alternatively, binomial filters correspond to the rows of the Pascal's triangle.