Appendix A

CORRELATION TRACKING ALGORITHM DETAILS

A.1 SETTING UP THE GRIDPOINT ARRAY

In §2.3, it is stated that the correlation tracking algorithm is applied to two equally sized images $I_1(x, y)$ and $I_2(x, y)$. Our notation is such that integer values of the pixel coordinates x and y index the pixels in the two images, though we will also need to refer to fractional pixel values in what follows. An array of N gridpoints at which local displacements are to be detected is denoted by the coordinates (x_n, y_n), where $n = 1, 2, \cdots, N$. By the following method, local displacements are calculated so that the pixels in the neighborhood of (x_n, y_n) in image I_1 optimally coincide with the pixels in the neighborhood of (x_n, y_n) in image I_2 when shifted by the relative amount $(\delta x, \delta y)$.

Before calculating the optimal displacement, we formally define the neighborhood of each gridpoint as those pixels contained in a square subimage centered on the gridpoint in question. For example, if the subimages have $2p + 1$ pixels on a side (that is, p is the subimage half-width), the subimage centered on the gridpoint (x_n, y_n) extracted from image I_1 is thus comprised of all pixels $I_1(x_i, y_j)$ with

$$x_i = x_n - p + i \qquad \text{and} \qquad y_j = y_n - p + j, \tag{A.1}$$

where the indices i and j run from $i, j = 0, 1, \cdots, 2p$. Therefore, this subimage is bounded horizontally by $x = x_n - p$ and $x = x_n + p$ and bounded vertically by $y = y_n - p$

and $y = y_n + p$ inclusive. As shorthand, we define S_1^n as the subimage centered on the measurement gridpoint (x_n, y_n) extracted from image I_1. The conventions defined above are equally applicable to image I_2.

A.2 THE MERIT FUNCTION

Given the two corresponding subimages S_1^n and S_2^n centered on the same gridpoint (x_n, y_n) extracted from I_1 and I_2, we now wish to determine the relative shift between the two subimages such that the topology of the two subimages maximally coincides. To calculate this optimal displacement, we determine the displacement vector $(\delta x, \delta y)$ such that the following merit function is minimized:

$$\begin{aligned} m(\delta x, \delta y) = \sum_{i=0}^{2p-1} \sum_{j=0}^{2p-1} \Bigg\{ & W\left(x_i - x_n + \frac{1}{2}, y_j - y_n + \frac{1}{2}\right) \\ & \times \left[I_1\left(x_i + \frac{1}{2} + \frac{\delta x}{2}, y_j + \frac{1}{2} + \frac{\delta y}{2}\right) \right. \\ & \left. - I_2\left(x_i + \frac{1}{2} - \frac{\delta x}{2}, y_j + \frac{1}{2} - \frac{\delta y}{2}\right)\right]^2 \Bigg\}. \end{aligned} \tag{A.2}$$

In its essence, the merit function $m(\delta x, \delta y)$ in equation (A.2) above is a weighted sum of the squares of the differences between the pixel values of the two corresponding subimages after shifting the two subimages with respect to each other by the relative offset $(\delta x, \delta y)$.

The merit function is arranged so that the fractional pixel shifts are measured with respect to a position a half-pixel above and a half-pixel to the right of the pixels in the subimages; that is, the fractional pixel shifts are measured with respect to the points $\left(x_i + \frac{1}{2}, y_j + \frac{1}{2}\right)$. This formulation explains the presence of the various $\frac{1}{2}$'s added to the pixel indices, as well as the upper limit of $2p - 1$ for each of the sums.

Additionally, the merit function is set up so that the subimages are shifted in a symmetric fashion. For given horizontal and vertical displacements δx and δy, S_1^n and S_2^n are each shifted by the same amounts $\frac{\delta x}{2}$ and $\frac{\delta y}{2}$ but in opposite directions. This symmetric shifting stabilizes the behavior of $m(\delta x, \delta y)$ near the point $(\delta x, \delta y) = (0, 0)$.

Both the one-half pixel offset and the symmetric shift simplify the computational task of evaluating I_1 and I_2 at interpixel values. This scheme guarantees that the interpolating point always falls within the square framed by the four pixels (x_i, y_j), $(x_i, y_j + 1)$, $(x_i + 1, y_j)$, and $(x_i + 1, y_j + 1)$, as long as only shifts of less than one pixel are considered. For this condition to be true, we require

$$|\delta x| \leq 1 \qquad \text{and} \qquad |\delta y| \leq 1. \tag{A.3}$$

For the magnitudes of the velocities we hope to measure in this work, this condition is always met.

The function W in equation (A.2) above is a weighting function multiplying the squared pixel differences so that the terms in the double sum arising from points closer to the subimage center (x_n, y_n) are weighted more heavily in the sum than those farther away. We use a Gaussian function for W,

$$W(x, y) = \exp\left(-\frac{x^2 + y^2}{\sigma^2}\right), \tag{A.4}$$

although any tapered function will suffice. The adjustable parameter σ is the e-folding distance of the Gaussian, which in turn determines the spatial resolution of the resulting velocity field. The parameter p introduced above, which characterizes the size of the subimages, should be chosen such that the subimages extend far enough out to encompass the peak and falloff of W. Choosing p too large will only increase the computational workload while the extra outlying points contribute little to the weighted sum in equation (A.2). Conversely, choosing p too small may not encompass all of the points surrounding the measurement gridpoint which have any significant weighting in the double sum. In this work, p is chosen to be slightly larger than σ. Typically, $p = 11$ pixels for $\sigma = 8$ pixels.

As seen by inspecting equation (A.2), minimizing the merit function requires a method for evaluating I_1 and I_2 at fractional pixel coordinates. In general, if we know the value of a function f at the four bounding points (x_1, y_1), (x_2, y_1), (x_1, y_2), and (x_2, y_2), we can bilinearly interpolate between the abscissae to estimate the value of f at any point interior to the bounding points:

$$\begin{aligned} f(x_1 + \Delta x, y_1 + \Delta y) = (1 - A)(1 - B)\, f(x_1, y_1) + A(1 - B)\, f(x_2, y_1) \\ + (1 - A)B\, f(x_1, y_2) + AB\, f(x_2, y_2), \end{aligned} \tag{A.5}$$

where the scaled coordinates A and B are defined

$$A \equiv \frac{\Delta x}{x_2 - x_1} \qquad \text{and} \qquad B \equiv \frac{\Delta y}{y_2 - y_1}. \tag{A.6}$$

For the point $(x_1 + \Delta x, y_1 + \Delta y)$ to be interior to the bounding points, we must have

$$0 \le A \le 1 \qquad \text{and} \qquad 0 \le B \le 1, \tag{A.7}$$

otherwise we would be *extrapolating* rather than *interpolating*.

The bilinear interpolation formula (A.5) is used to evaluate I_1 and I_2 at fractional pixel coordinates. Identifying the indices x_1, x_2, y_1, and y_2 in equation (A.5) with the pixel indices x_i, x_{i+1}, y_j, and y_{j+1}, respectively, and setting $\Delta x = \frac{1}{2} + \frac{\delta x}{2}$ and $\Delta y = \frac{1}{2} + \frac{\delta y}{2}$, we obtain the following expression for the I_1 term in equation (A.2) above:

$$\begin{aligned} I_1\left(x_i + \frac{1}{2} + \frac{\delta x}{2}, y_j + \frac{1}{2} + \frac{\delta y}{2}\right) = \left(\frac{1}{2} - \frac{\delta x}{2}\right)\left(\frac{1}{2} - \frac{\delta y}{2}\right) I_1(x_i, y_j) \\ + \left(\frac{1}{2} + \frac{\delta x}{2}\right)\left(\frac{1}{2} - \frac{\delta y}{2}\right) I_1(x_{i+1}, y_j) \\ + \left(\frac{1}{2} - \frac{\delta x}{2}\right)\left(\frac{1}{2} + \frac{\delta y}{2}\right) I_1(x_i, y_{j+1}) \\ + \left(\frac{1}{2} + \frac{\delta x}{2}\right)\left(\frac{1}{2} + \frac{\delta y}{2}\right) I_1(x_{i+1}, y_{j+1}). \end{aligned} \tag{A.8}$$

Likewise for I_2, where now $\Delta x = \frac{1}{2} - \frac{\delta x}{2}$ and $\Delta y = \frac{1}{2} - \frac{\delta y}{2}$, we have

$$\begin{aligned} I_2\left(x_i + \frac{1}{2} - \frac{\delta x}{2}, y_j + \frac{1}{2} - \frac{\delta y}{2}\right) = {} & \left(\frac{1}{2} + \frac{\delta x}{2}\right)\left(\frac{1}{2} + \frac{\delta y}{2}\right) I_2(x_i, y_j) \\ & + \left(\frac{1}{2} - \frac{\delta x}{2}\right)\left(\frac{1}{2} + \frac{\delta y}{2}\right) I_2(x_{i+1}, y_j) \\ & + \left(\frac{1}{2} + \frac{\delta x}{2}\right)\left(\frac{1}{2} - \frac{\delta y}{2}\right) I_2(x_i, y_{j+1}) \\ & + \left(\frac{1}{2} - \frac{\delta x}{2}\right)\left(\frac{1}{2} - \frac{\delta y}{2}\right) I_2(x_{i+1}, y_{j+1}). \end{aligned} \tag{A.9}$$

Note that the restrictions on A and B given above in equation (A.7) imply that $|\delta x| \leq 1$ and $|\delta y| \leq 1$, as already assumed by equation (A.3). We now substitute equations (A.8) and (A.9) into equation (A.2) to obtain

$$\begin{aligned} m(\delta x, \delta y) = \sum_{i=0}^{2p-1} \sum_{j=0}^{2p-1} \Bigg\{ & W\left(x_i - x_n + \frac{1}{2}, y_j - y_n + \frac{1}{2}\right) \\ & \times \Bigg[\left(\frac{1}{2} - \frac{\delta x}{2}\right)\left(\frac{1}{2} - \frac{\delta y}{2}\right) I_1(x_i, y_j) \\ & + \left(\frac{1}{2} + \frac{\delta x}{2}\right)\left(\frac{1}{2} - \frac{\delta y}{2}\right) I_1(x_{i+1}, y_j) \\ & + \left(\frac{1}{2} - \frac{\delta x}{2}\right)\left(\frac{1}{2} + \frac{\delta y}{2}\right) I_1(x_i, y_{j+1}) \\ & + \left(\frac{1}{2} + \frac{\delta x}{2}\right)\left(\frac{1}{2} + \frac{\delta y}{2}\right) I_1(x_{i+1}, y_{j+1}) \\ & - \left(\frac{1}{2} + \frac{\delta x}{2}\right)\left(\frac{1}{2} + \frac{\delta y}{2}\right) I_2(x_i, y_j) \\ & - \left(\frac{1}{2} - \frac{\delta x}{2}\right) - \left(\frac{1}{2} + \frac{\delta y}{2}\right) I_2(x_{i+1}, y_j) \\ & - \left(\frac{1}{2} + \frac{\delta x}{2}\right)\left(\frac{1}{2} - \frac{\delta y}{2}\right) I_2(x_i, y_{j+1}) \\ & - \left(\frac{1}{2} - \frac{\delta x}{2}\right)\left(\frac{1}{2} - \frac{\delta y}{2}\right) I_2(x_{i+1}, y_{j+1})\Bigg]^2\Bigg\}. \end{aligned} \tag{A.10}$$

This expression for $m(\delta x, \delta y)$ depends only upon the unknown displacement $(\delta x, \delta y)$ and the data values of the pixels in the two subimages extracted from I_1 and I_2.

A.3 MINIMIZING THE MERIT FUNCTION

To find the values of δx and δy which minimize the merit function (A.10), we will need to evaluate the derivatives $\dfrac{\partial m}{\partial(\delta x)}$ and $\dfrac{\partial m}{\partial(\delta y)}$. This task is simplified by first expanding the squared term in equation (A.10). To proceed, first define the quantities

$$d_1 = I_1(x_j, y_k) - I_2(x_{j+1}, y_{k+1}),$$

$$d_2 = I_1(x_j, y_{k+1}) - I_2(x_{j+1}, y_k),$$

$$d_3 = I_1(x_{j+1}, y_k) - I_2(x_j, y_{k+1}),$$

$$d_4 = I_1(x_{j+1}, y_{k+1}) - I_2(x_j, y_k),$$

and then substitute these differences into equation (A.10):

$$\begin{aligned} m(\delta x, \delta y) = \sum_{i=0}^{2p-1} \sum_{j=0}^{2p-1} \Bigg\{ \frac{W_{ij}}{16} \Big[&(1-\delta x)(1-\delta y)\, d_1 + (1-\delta x)(1+\delta y)\, d_2 \\ &+ (1+\delta x)(1-\delta y)\, d_3 + (1+\delta x)(1+\delta y)\, d_4 \Big]^2 \Bigg\}, \end{aligned} \tag{A.11}$$

where $W_{ij} = W\left(x_i - x_n + \dfrac{1}{2}, y_j - y_n + \dfrac{1}{2}\right)$. Then, after expanding the square and regrouping we have

$$\begin{aligned} m(\delta x, \delta y) = \sum_{i=0}^{2p-1} \sum_{j=0}^{2p-1} \Bigg\{ \frac{W_{ij}}{16} \Big[&a_1 + a_2\, \delta x^2 + a_3\, \delta y^2 + a_4\, \delta x^2 \delta y^2 + a_5\, \delta x \\ &+ a_6\, \delta y + a_7\, \delta x\, \delta y + a_8\, \delta x^2 \delta y + a_9\, \delta x\, \delta y^2 \Big] \Bigg\}, \end{aligned} \tag{A.12}$$

where, after some algebra, the a-coefficients are found to be

$$
\begin{aligned}
a_1 &= (d_1 + d_2 + d_3 + d_4)^2, \\
a_2 &= (d_1 + d_2 - d_3 - d_4)^2, \\
a_3 &= (d_1 - d_2 + d_3 - d_4)^2, \\
a_4 &= (d_1 - d_2 - d_3 + d_4)^2, \\
a_5 &= -2(d_1 + d_2 + d_3 + d_4)(d_1 + d_2 - d_3 - d_4), \\
a_6 &= -2(d_1 + d_2 + d_3 + d_4)(d_1 - d_2 + d_3 - d_4), \\
a_7 &= 4(d_1^2 - d_2^2 - d_3^2 + d_4^2), \\
a_8 &= -2(d_1 + d_2 - d_3 - d_4)(d_1 - d_2 - d_3 + d_4), \\
a_9 &= -2(d_1 - d_2 + d_3 - d_4)(d_1 - d_2 - d_3 + d_4).
\end{aligned}
$$

Since the offsets δx and δy are independent of the quantities being summed over, we can pull them out of the sum and define the coefficients

$$A_m \equiv \sum_{i=0}^{2p-1} \sum_{j=0}^{2p-1} \frac{W_{ij}}{16} a_m, \tag{A.13}$$

where $m = 1, 2, \cdots, 9$. Substituting equation (A.13) into equation (A.12), we have

$$
\begin{aligned}
m(\delta x, \delta y) =& A_1 + A_2\, \delta x^2 + A_3\, \delta y^2 + A_4\, \delta x^2 \delta y^2 + A_5\, \delta x \\
&+ A_6\, \delta y + A_7\, \delta x\, \delta y + A_8\, \delta x^2 \delta y + A_9\, \delta x\, \delta y^2.
\end{aligned} \tag{A.14}
$$

Note that the A_m-coefficients can be calculated as soon as the subimages have been extracted, as they depend only upon W and the data values contained in the two subimages S_1^n and S_2^n.

To find the point $(\delta x, \delta y)$ which minimizes $m(\delta x, \delta y)$, we follow the usual procedure of setting the first partial derivatives of m equal to zero and solving the resulting system of equations:

$$\frac{\partial m}{\partial(\delta x)} = 0 \quad \Longrightarrow \quad \left(A_5 + A_7 \delta y + A_9 \delta y^2\right) + 2\delta x \left(A_2 + A_8 \delta y + A_4 \delta y^2\right) = 0, \tag{A.15}$$

$$\frac{\partial m}{\partial(\delta y)} = 0 \quad \Longrightarrow \quad \left(A_6 + A_7 \delta x + A_8 \delta x^2\right) + 2\delta y \left(A_3 + A_9 \delta x + A_4 \delta x^2\right) = 0. \tag{A.16}$$

At this point, we note that solving for either δx or δy using equation (A.15) or (A.16) and substituting the resulting expression into the other yields a quintic equation. Rather than look for zeroes in such a quintic equation, the correlation tracking algorithm uses an iterative scheme to approximate the extremal value of $(\delta x, \delta y)$. Starting with an estimate for δy, the algorithm obtains an estimate for δx by solving equation (A.15). Using this estimate for δx, a new estimate for δy is calculated by solving equation (A.16). This stepping scheme is iterated until either δx and δy differ from its previous estimate by less than 10^{-4}.

A.4 ITERATING WITH BICUBIC SHIFTS

To attain a more accurate estimate of the optimal displacement $(\delta x, \delta y)$, we could use a higher-ordered interpolation scheme (such as bicubic interpolation) to evaluate I_1 and I_2 at interpixel values in the merit function (A.2) above. However, we find it simpler to use the optimal shifts $(\delta x, \delta y)$ calculated by minimizing m given by equation (A.10), which uses bilinear shifts to estimate interpixel data values, as a guess for what the optimal displacement would be had bicubic interpolation been used in minimizing the merit function. Then one of the subimages is bicubically shifted by the known amount $(\delta x, \delta y)$ and the procedure is repeated. This iteration scheme is chosen because it is computationally faster to bicubically shift images by a known amount, as opposed to calculating an unknown displacement assuming bicubic interpolation was used to evaluate the functions I_1 and I_2 at interpixel values.

The correlation tracking algorithm therefore proceeds as follows. At each measurement gridpoint (x_n, y_n), an optimal displacement is calculated using bilinear shifts in the merit function as described in §A.2. Define this initial displacement as $(\delta x, \delta y)_{\mathrm{agg}}$. The subimage S_2^n is then shifted by this offset $(\delta x, \delta y)_{\mathrm{agg}}$ using *bicubic* interpolation. An additional optimal displacement vector $(\delta x', \delta y')$ is then calculated using the merit function containing bilinear shifts, where S_1^n and the (bicubically) shifted version of S_2^n

are now used as input images. The total displacement becomes the sum of the initial displacement and this additional displacement, so that the aggregate displacement $(\delta x, \delta y)_{\text{agg}} = (\delta x, \delta y)^{\text{old}}_{\text{agg}} + (\delta x', \delta y')$.

We now continue to add to the aggregate displacement by iterating until convergence. That is, subimage S_2^n is now bicubically shifted by the aggregate displacement $(\delta x, \delta y)_{\text{agg}}$, and another additional displacement vector is calculated using S_1^n and the newly shifted version of S_2^n. This new displacement vector is then added to the aggregate. Convergence occurs when the displacement vector produced using S_1^n and a shifted version of S_2^n has a $|\delta x|$ or $|\delta y|$ less than 10^{-4}. This aggregate shift $(\delta x, \delta y)_{\text{agg}}$ is then returned as the optimal displacement for the gridpoint in question. This bicubic iteration process provides an estimate for what the optimal displacement would have been had bicubic shifts (rather than bilinear shifts) been used to evaluate I_1 and I_2 at interpixel values in equation (A.2).

A.5 CORRELATION TRACKING PITFALLS

In determining the optimal displacement for each measurement gridpoint, several problems may befall the correlation algorithm described above. As stipulated by equation (A.3), if either aggregate offset δx_{agg} or δy_{agg} is more than one pixel in either direction, the algorithm sets the offsets to zero and the gridpoint is flagged. As stated earlier, we do not expect the motions in the time series analyzed in this these to possess displacements of more than one pixel.

Two problems may occur when iterating using the bicubic shifts. First, if more than twenty iterations occur prior to convergence, the offsets are set to zero and the gridpoint is flagged. This situation usually occurs when the merit function contains no prominent minimum causing the algorithm to wander around in $(\delta x, \delta y)$-space, as is caused by when the subimages contain no prominent topological features.

The second problem which may occur during the bicubic iteration process is a degradation of the merit function. Ideally, the value of the merit function continues to decrease as the minimum value of $(\delta x, \delta y)$ is approached. If the value of the merit function is found to increase between iterations, suggesting that the next aggregate offset is actually farther away from the minimum in $m(\delta x, \delta y)$, the current aggregate displacements are returned and the gridpoint is flagged. This problem generally occurs when the topology of the merit function near the minimum value of $(\delta x, \delta y)$ is somewhat lumpy, containing several local extrema in the neighborhood of the absolute minimum.