Adaptive Localization: Proposals for a high-resolution multivariate system
Version 3.1

1. The implicit Schur product
2. The Bishop method for adaptive localization (ECO-RAP)
3. Element-by-element evaluation of (10)
4. Calculation of the localized covariances
5. Notes for evaluating (22) for a structure function
6. Limiting cases
7. Suggested algorithm with adaptive localization
8. Suggested algorithm with static localization only
9. Adaptive localization with a major simplification

References

1. The implicit Schur product

The Schur product is used in ensemble-based data assimilation to remove long-range correlations

\[ P^f_L = P^f_R \odot \Omega, \]  

where \( P^f_L \) and \( P^f_R \) are the localized and raw forecast error covariance matrices respectively and \( \Omega \) is the localization matrix. These matrices are of size \( 5n \times 5n \) (where \( n \) is the total number of grid points for each of the five parameters, \( \psi, \chi, p, \theta \) and \( q \) [what about \( w \)?]) and so we don't have the ability to store them explicitly.

In the ensemble Kalman filter, \( P^f_R \) is represented by its square-root (ie the \( 5n \times K \) matrix of ensemble perturbations, each divided by \( \sqrt{K-1} \)). Assuming that \( \Omega \) is also in its square-root \((5n \times L)\) form then

\[ P^f_R = P^{1/2}_R P^{T/2}_R = \frac{1}{K-1} \sum_{k=1}^{K} x_k x_k^T, \]  

\[ \Omega = \Omega^{1/2} \Omega^{T/2} = \frac{1}{L-1} \sum_{l=1}^{L} \omega_l \omega_l^T. \]  

In the last line, the square-root of \( \Omega \) is also considered to be comprised of new effective ensemble members, \( \omega_l \), each divided by \( \sqrt{L-1} \). For \( \Omega \) to be a correlation matrix, each component of the \( \omega_l \) must have a variance of unity. Substituting (2) and (3) into (1) and then writing for the \( i, j \) th element of \( P^f_L \) gives

\[ P^f_L = (P^{1/2}_R P^{T/2}_R) \odot (\Omega^{1/2} \Omega^{T/2}) \]  

\[ P^f_{Lij} = (P^{1/2}_R P^{T/2}_R)_{ij} (\Omega^{1/2} \Omega^{T/2})_{ij}. \]
Equation (7) shows that the localized forecast error covariance matrix is effectively made up of approximately \( KL \) ensemble members instead of just \( K \). The effective ensemble members that give rise to the localized covariances can be written as

\[
\tilde{x}_{pq} = x_p \odot \omega_q, \tag{8}
\]

where \( \tilde{x}_{pq} \) is the effective ensemble member comprising the vector Schur product of raw ensemble member \( x_p \) (\( \sqrt{K} - 1 \) times the \( p \)th column of \( \mathbf{P}_R^{1/2} \)) with \( \omega_q \) (\( \sqrt{L} - 1 \) times the \( q \)th column of \( \Omega^{1/2} \)).

2. The Bishop method for adaptive localization (ECO-RAP)

Bishop and Hodyss [1] proposed the following form for \( \Omega^{1/2} \)

\[
\Omega^{1/2} = C_K^{\odot Q} E \Lambda^{1/2}. \tag{9}
\]

Here \( C_K \) is a \( 5n \times 5n \) correlation matrix calculated from the \( K \) ensemble members (see below), and \( E \Lambda^{1/2} \) is a \( 5n \times L \) matrix. \( E \) performs an inverse Fourier transform per parameter, and \( \Lambda^{1/2} \) performs scale-dependent filtering. The \( \odot Q \) superscript in (9) indicates an element-by-element raising of power (a Schur power), where \( Q \) is even. The overbar denotes a normalization so that the \( \Omega \) becomes a correlation matrix. This involves setting the sum of squares of each row of \( \Omega^{1/2} \) to unity.

The localization gains its adaptive property through the \( C_K \) matrix. If it were not for \( C_K \), (9) would be the square-root of a static and homogeneous correlation matrix. The issues of this problem are the following.

1. Determination of (i) the set spectral modes in the horizontal, (ii) the set of vertical modes in the vertical for each model quantity, (iii) an appropriate spectrum, \( \Lambda \), and (iv) a choice of \( L \), the number of modes to truncate.

2. Efficient determination and action of \( C_K^{\odot Q} \).

For reference, (9) has the following multivariate form (followed by a specification of the dimensions of each matrix).

\[
\begin{bmatrix}
E_p \Lambda_p^{1/2} \\
E_x \Lambda_x^{1/2} \\
E_p \Lambda_p^{1/2} \\
E_q \Lambda_q^{1/2} \\
E_q \Lambda_q^{1/2}
\end{bmatrix} =
\begin{bmatrix}
C_K^{\odot Q} \\
E \Lambda^{1/2}
\end{bmatrix}, \tag{10}
\]

\([5n \times L] = [5n \times 5n] [5n \times L].\)

The \([5n \times L]\) part of the right hand side of (10) is the static localization. It imposes no
multivariate localization modulation; it limits univariate lengthscales of each variable. Localization associated with the multivariate part of the problem is handled by the adaptive matrix, $C_K$.

3. **Element-by-element evaluation of (10)**
Equation (10) has a high operation count. In the HRTM there are $n = 360 \times 288 \times 70 = 7.26 \times 10^8$ grid points. The five variables means that there are $3.6 \times 10^7$ variables. Clearly special attention must be paid towards efficiency of the problem and any approximations that can be made should be made.

Let $(C_K^\circ \mathbf{E}^{1/2})_{ip,k}$ be column $k$ and field position $i$ for parameter $p$ of $C_K^\circ \mathbf{E}^{1/2}$.

$$
(C_K^\circ \mathbf{E}^{1/2})_{ip,k} = \sum_{j \neq p'} (C_K^\circ)_{ip,j} \mathbf{E}_{j',j} (\Lambda_{p'}^{1/2})_{k,k},
$$

where $i, j$ go from 1 to $n$ and $p, p'$ run over each parameter ($\psi$, $\chi$, $p$, $\theta$, $q$). The matrix $C_K$ (a correlation matrix found from the ensemble members) has the following form

$$
C_K = \Sigma^{-1} P_k \Sigma^{-1},
$$

$$
= \frac{1}{K - 1} \sum_{k = 1}^K \Sigma^{-1} x_k x_k^T \Sigma^{-1},
$$

where $\Sigma$ is the diagonal standard deviation matrix. Element $i, j$ between parameters $p, p'$ is

$$
(C_K)_{ip,jp'} = \frac{1}{K - 1} \sum_{k = 1}^K \frac{(x_i)_k^T (x_j)_k}{\sigma^2_p \sigma^2_p'},
$$

and

$$
(C_K^\circ)_{ip,jp'} = (C_K)_{ip,jp'}.\]

The normalization in (9) and (10) (ie the overbar) means that the localization matrix has to be calculated row-wise. Normalization gives the matrix $\Omega^{1/2}$.

4. **Calculation of the localized covariances**
The localized covariance element $ip, i'p'$ is, from (7)

$$
(P^{\circ}_{i}^T)_{ip,i'p'} = \frac{1}{(K - 1)(L - 1)} \sum_{k = 1}^K \sum_{l = 1}^L (x_k \odot \omega_i)_{ip} (x_k \odot \omega_i)_{l'p'},
$$

$$
= \frac{1}{(K - 1)(L - 1)} \sum_{k = 1}^K \sum_{l = 1}^L (x_k)_{ip} (\omega_i)_{ip} (x_k)_{l'p'} (\omega_i)_{l'p'}. \]

$(x_k)_{ip}$ and $(x_k)_{l'p'}$ are readily available, $(\omega_i)_{ip}$ and $(\omega_i)_{l'p'}$ are not. The relationship between the columns of $\Omega^{1/2}$ and $\omega_i$ is

$$
(\omega_i)_{ip} = \sqrt{L - 1} (\Omega^{1/2})_{ip,l},
$$

where $\Omega^{1/2}$ is to be written in terms of its components (9). The overbar on (9) can be dealt with by a factor $\mu_{ip}$, which normalizes

$$
(\Omega^{1/2})_{ip,l} = \mu_{ip} (C_K^\circ \mathbf{E}^{1/2})_{ip,l},
$$

3
Adaptive Localization: Proposals for a high-resolution multivariate system

where \( \mu_{ip} = \frac{1}{\sum_{l=1}^{L} (C_{K}^{\omega} E\Lambda_{l}^{1/2})_{ip,l}^2} \).

Combining (17), (18) and (11) gives

\[
(\omega_{i})_{ip} = \sqrt{L - 1} \mu_{ip} \sum_{j,l'} (C_{K}^{\omega})_{ip,j,l'} E_{p',i,j,l} (\Lambda_{l'_{2}}^{1/2})_{i,l}.
\]

Substituting (20) into (16) gives an expression for the localized covariances in terms of quantities that are known

\[
(P_{l})_{ip,i'p'} = \frac{1}{(K - 1)} \times \\
\sum_{k=1}^{K} \sum_{i=1}^{L} (x_{i})_{ip} (x_{i})_{i'p'} \left( \mu_{ip} \sum_{f' \neq \cdot} (C_{K}^{\omega})_{ip,f'p} E_{f',i',j,l} (\Lambda_{l'_{2}}^{1/2})_{i,l} \right) \left( \mu_{i'p'} \sum_{f' \neq \cdot} (C_{K}^{\omega})_{i'p',f'p} E_{f',i',j,l} (\Lambda_{l'_{2}}^{1/2})_{i,l} \right).
\]

This summation has to be arranged so that it can be evaluated in the most efficient way, allowing for evaluation of the \( \mu_{ip} \) coefficients

\[
(P_{l})_{ip,i'p'} = \frac{1}{(K - 1)} \sum_{k=1}^{K} (x_{i})_{ip} (x_{i})_{i'p'} \times \\
\sum_{i=1}^{L} \left( \mu_{ip} \sum_{f'} \sum_{p'} (C_{K}^{\omega})_{ip,f'p} E_{f',i',j,l} (\Lambda_{l'_{2}}^{1/2})_{i,l} \right) \left( \mu_{i'p'} \sum_{f'} \sum_{p'} (C_{K}^{\omega})_{i'p',f'p} E_{f',i',j,l} (\Lambda_{l'_{2}}^{1/2})_{i,l} \right).
\]

From (19) and (11)

\[
\mu_{ip} = \frac{1}{\sqrt{\sum_{l=1}^{L} (\sum_{r',r''} (C_{K}^{\omega})_{ip,r',r''} E_{r',i',r''} (\Lambda_{r''_{2}}^{1/2})_{i,l})^2}}.
\]

In Sec. 9, we consider a major simplification of these equations that is considered to make way for their efficient evaluation for large systems. For now though we consider the exact form of the equations.

5. Notes for evaluating (22) for a structure function

- For a structure function, \( i' \) and \( p' \) will both be fixed.
- The \( f'' \) summations that appear in the above may be evaluated on a reduced resolution grid (e.g. every 10-points).
- The summations \( \sum_{r',r''} (C_{K}^{\omega})_{ip,r',r''} E_{r',i',r''} (\Lambda_{r''_{2}}^{1/2})_{i,l} \) appear in (22) and in (23). Store these for all \( l \) for each \( i, p \) to allow (23) to be evaluated.

6. Limiting cases

- Choosing \( Q \to \infty \) leads to \( C_{K}^{\omega} = I \) (i.e. only elements that have matrix element identically unity will survive the Schur power). This is equivalent to the case with no adaptive localization.
- Choosing \( Q = 0 \) is non-physical. It will set each non-zero matrix element in \( C_{K}^{\omega} \)
Note a fundamental difference between the conventional and the Schur matrix products. For the conventional matrix product

$$AB = C,$$  \hfill (24)

setting B to the identity matrix will leave A = C. For the Schur matrix product

$$A \otimes B = C,$$  \hfill (25)

setting all elements of B to 1 will leave A = C.

Exploring the case when there is no adaptive localization, \(Q \to \infty\), then (10) becomes

$$\Omega^{1/2} = E\Lambda^{1/2} = \begin{bmatrix} E_q\Lambda^{1/2}_q \\ E_x\Lambda^{1/2}_x \\ E_p\Lambda^{1/2}_p \\ E_q\Lambda^{1/2}_q \end{bmatrix},$$  \hfill (26)

which is block diagonal in parameter. Considering only one parameter, then (26) gives the following square root

$$(\Omega^{1/2})_{iq} = \mu_i E_{iq}\Lambda^{1/2}_{qq},$$  \hfill (27)

where

$$\mu_i = \sqrt{\frac{1}{\sum_{l=1}^{L} (E_{il}\Lambda_{ll})^2}}.$$  \hfill (28)

Meaning that \(\Omega\) from (27) with (3) give

$$\Omega_{ij} = \sum_{q=1}^{L} (\Omega^{1/2})_{iq}(\Omega^{1/2})_{qj},$$  \hfill (29)

$$= \mu_i\mu_j \sum_{q=1}^{L} E_{iq}E_{jq}^\dagger\Lambda_{qq},$$  \hfill (30)

where \(\dagger\) means complex conjugate (we add this here because the illustration below makes use of a complex Fourier transform). In 1-D (30) becomes

$$\Omega_{ij} = \mu_i\mu_j \sum_{q=1}^{L} \exp(ik_q(r_i - r_j))\Lambda_{qq}$$  \hfill (31)

where \(k_q\) is the qth wavenumber and \(r_i\) is the position of the \(i\)th grid point. If \(L\) covers the complete spectrum and \(\Lambda_{qq}\) is constant (broad localization in spectral space) then orthogonality gives

$$\Omega_{ij} = \mu_i^2 \delta_{ij},$$  \hfill (32)

meaning that this Schur product will be diagonal and will completely localize in real space. A narrower localization in spectral space, ie \(\Lambda_{qq} \to 0\) with increasing \(q\) (qualitatively similar to smaller \(L\)) then the localization in real space will be broader.
7. Suggested algorithm with adaptive localization

Costs for each loop are specified at the end of each loop in red for the case when no efficiencies are used, and in blue when efficiencies are used. Numerical terms in blue brackets are for the specific model domain ($n = 360 \times 288 \times 70$ and 5 parameters). Assume for now that $L = 50$, except for the numerical values in green which are for the efficiency costs, but for $L = 1$.

1. ===== Calculation of right-hand bracketed term in (22) =====
2. Loop round $j'', p''$
3. Evaluate $\chi(j'', p'') = (C_{K}^{O})_{ij,p''} K (24) K (24) (24)$
4. End loop ($j'', p''$) $5Kn (871 \times 10^{6}) 5Kn / 100 (9 \times 10^{6}) (9 \times 10^{6})$
5. $\alpha(\cdot) = 0$
6. $\beta = 0$
7. Loop round $l$
8. Loop round $j'', p''$
9. $\alpha (l) \pm \chi(j'', p'') E_{ij,p''} (\Lambda_{ij}^{(2)})_{ij}$
10. End loop ($j'', p''$) $5n (36 \times 10^{4}) 5n / 100 (36 \times 10^{4}) (36 \times 10^{4})$
11. $\beta \pm \alpha^{2} (l)$
12. End loop ($l$) $5Ln (1.8 \times 10^{9}) 5Ln / 100 (18 \times 10^{6}) (36 \times 10^{4})$
13. $\mu_{ip} = \sqrt{\beta}$
14. ===== Loop around destination points in the structure function ======
15. Loop round $i, p$
16. ===== Calculation of left-hand bracketed term in (22) ======
17. $\gamma = 0$
18. Loop round $k$
19. $\gamma \dagger (\lambda_{i})_{ip} (\lambda_{k})_{ip}$
21. ===== Calculation of middle bracketed term in (22) ======
22. Loop round $j'', p''$
23. Evaluate $\chi(j'', p'') = (C_{K}^{O})_{ij,p''} K (24) K (24) (24)$
24. End loop ($j'', p''$) $5Kn (871 \times 10^{6}) 5Kn / 100 (9 \times 10^{6}) (9 \times 10^{6})$
25. $\epsilon(\cdot) = 0$
26. $\beta = 0$
27. Loop round $l$
28. Loop round $j'', p''$
29. $\epsilon (l) \pm \chi(j'', p'') E_{ij,p''} (\Lambda_{ij}^{(2)})_{ij}$
30. End loop ($j'', p''$) $5n (36 \times 10^{4}) 5n / 100 (36 \times 10^{4}) (36 \times 10^{4})$
31. $\beta \pm \epsilon^{2} (l)$
32. End loop ($l$) $5Ln (1.8 \times 10^{9}) 5Ln / 100 (18 \times 10^{6}) (363 \times 10^{3})$
33. $\mu_{ip} = \sqrt{1/\beta}$
Adaptive Localization: Proposals for a high-resolution multivariate system

34. Structure function for $i, p$ can be evaluated - see (33)
35. End loop $(i, p)$

$$5n [K + 5n (1 + K + L)] \ (10^{15})$$
$$5n [K + 5n (1 + K + L)/100] \ (10^{15}) \ (3.3 \times 10^{14})$$

$$\langle \mathbf{P}_{ij}^{l} \rangle_{i',p',l'} = \frac{\gamma \mu_{i,p} \mu_{i',p'}}{(K-1)} \sum_{l=1}^{L} \varepsilon (l) \alpha (l). \quad (33)$$

8. Suggested algorithm with static localization only

Without the adaptive localization the problem becomes considerably simpler. In this case (22) and (23) become

$$\langle \mathbf{P}_{ij}^{l} \rangle_{i',p',l'} = \frac{1}{(K-1)} \left( \sum_{k=1}^{K} (x_k)_i (x_k')_{i'} \right) \sum_{l=1}^{L} \left( \mu_{i,p} \mathbf{E}_{p,i,l} (\Lambda_{p}^{1/2})_{i,l} \right) \left( \mu_{i',p'} \mathbf{E}_{p',i',l} (\Lambda_{p'}^{1/2})_{i',l} \right), \quad (34)$$

$$\mu_{i,p} = \frac{1}{\sqrt{\sum_{l=1}^{L} (\mathbf{E}_{p,i,l} (\Lambda_{p}^{1/2})_{i,l})^{2}}} \quad (35)$$

1. ===== Calculation of right-hand bracketed term in (34) =====
2. $\beta = 0$
3. Loop round $l$
4. $\alpha (l) = \mathbf{E}_{p',i',l} (\Lambda_{p'}^{1/2})_{i',l}$
5. $\beta = \alpha^{2} (l)$
6. End loop $(l)$ $L \ (50) \ (1)$
7. $\mu_{i',p'} = \sqrt{1/\beta}$
8. ===== Loop around destination points in the structure function =====
9. Loop round $i, p$
10. ===== Calculation of left-hand bracketed term in (34) =====
11. $\gamma = 0$
12. Loop round $k$
13. $\gamma = \ast (x_k)_i (x_k')_{i'}$
14. End loop $(k)$ $K \ (24) \ (24)$
15. ===== Calculation of middle bracketed term in (34) =====
16. $\beta = 0$
17. Loop round $l$
18. $\epsilon (l) = \mathbf{E}_{p,i,l} (\Lambda_{p}^{1/2})_{i,l}$
19. $\beta = \epsilon^{2} (l)$
20. End loop $(l)$ $L \ (50) \ (1)$
21. $\mu_{i,p} = \sqrt{1/\beta}$
22. Structure function for $i, p$ can be evaluated - see (33)
23. End loop $(i, p)$ $5n [K + L] \ (2.7 \times 10^{9}) \ (0.9 \times 10^{9})$

9. Adaptive localization with a major simplification

In Sec. 7 we considered an algorithm for the brute-force evaluation of (22) and (23) for the
evaluation of localized covariances, and in Sec. 8 we considered the limiting case where the localization was static. Neither of these approaches are useful for large systems (the algorithm in Sec. 7 is prohibitive and the algorithm in Sec. 8 is inadequate for many purposes). Here we consider a simplification to the covariance formulae that may be useable and useful.

First, recap the equations that are to be evaluated. The localized covariance matrix elements from (16)

$$\left( P^L \right)_{i p, i' p'} = \frac{1}{(K - 1)(L - 1)} \sum_{k=1}^{K} \sum_{j=1}^{L} (x_i)_{i p} (\omega_i)_{i p} (x_{i'})_{i' p'} (\omega_{i'})_{i' p'},$$  \hspace{1cm} (16)$$

the localization members from (17)

$$\left( \omega_i \right)_{i p} = \sqrt{L - 1} \left( \Omega^{1/2} \right)_{i p, i l},$$  \hspace{1cm} (17)$$

and elements of the localization matrix from (18)

$$\left( \Omega^{1/2} \right)_{i p, i l} = \mu_{i p} \left( C^0 K E \Lambda^{1/2} \right)_{i p, i l},$$  \hspace{1cm} (18)$$

where, from (19)

$$\mu_{i p} = \sqrt{\frac{1}{\sum_{l=1}^{L} \left( C^0 K E \Lambda^{1/2} \right)_{i p, i l}}}.$$

These are straight copies of equations previously given in this document. In [2] it is suggested that considerable efficiency savings can be made in the evaluation of (18) (and hence in (16)) in the case of adaptive localization if the matrix $C^0 K$ is approximated by one that has separable structure functions. This is now explored.

The analysis is centred on the evaluation of $\left( C^0 K E \Lambda^{1/2} \right)_{i p, i l}$, which is one of the most expensive parts of the calculation

$$\left( C^0 K E \Lambda^{1/2} \right)_{i p, i l} = \sum_{l'} \left( C_K \right)_{i p, i l'} \left( E_{l' p'} \Lambda^{1/2} \right)_{l' j}. \hspace{1cm} (36)$$

Now consider the case when rows of $C_K$ are approximated by separable functions. Since index $i'$ represents all three dimensions in space, this step requires a change of notation. Let a given $i$ represent a unique combination of $x$, $y$, $z$ and let $i'$ represent $x'$, $y'$, $z'$. Then $\left( C_K \right)_{i p, i' p'}$ may be written as

$$\left( C_K \right)_{i p, i' p'} = C_K (x, y, z, p; x', y', z', p'). \hspace{1cm} (37)$$

Assuming separable functions means that $\left( C_K \right)_{i p, i' p'}$ is approximated by

$$\left( C_K \right)_{i p, i' p'} \approx C_K (x, y, z, p; x', y', z', p') \times C_K (x, y, z, p; x, y', z, p') \times C_K (x, y, z, p; x, y', z', p'), \hspace{1cm} (38)$$

ie the row associated with $x$, $y$, $z$, $p$ is a function of $x'$, $y'$, $z'$, $p'$ and is written as the product of three functions, one a function of $x'$, $p'$, another a function of $y'$, $p'$ and another a function of $z'$, $p'$. This is separable in $x'$, $y'$, $z'$-space. Note that unfortunately, $C_K$ written in this way is not guaranteed to be symmetric (as is required for a correlation matrix), but it is assumed that this is not a vital for localization, as is presumably the case in [2].
This is useful if columns of \( E \) are also separable, which they are under the planned formulation. In the same notation as used above, and noting that \( j' \) (not \( j \)) is a wavevector index representing \( k_j', k_j'' \), then \( E_{j',j} \) may be written

\[
E_{j',j} = f_{j'}(k_j', x') \times f_{j'}(k_j', y') \times f_{j'}(k_j', z'),
\]

where \( f_{j'}(k_j', x'), f_{j'}(k_j', y') \) and \( f_{j'}(k_j', z') \) are orthogonal functions (trigonometric in the horizontal and EOF in the vertical).

Separability is useful because it makes evaluation of (36) cost effective as follows

\[
\left( C_k^Q E \Lambda^{1/2} \right)_{ip,j} = \sum_{i',p'} (C_k^Q (x, y, z, p; x', y', z, p') C_k^Q (x, y, z, p; x, y, z', p')) \times
\]

\[
f_{j'}(k_j', x') f_{j'}(k_j', y') f_{j'}(k_j', z') \Lambda^{1/2}(k_j', k_j', k_j'),
\]

\[
= \Lambda^{1/2}(k_j', k_j', k_j') \sum_{i',p'} \left[ \sum_x C_k^Q (x, y, z, p; x', y, z, p') f_{j'}(k_j', x') \times \right.
\]

\[
\left. \sum_y C_k^Q (x, y, z, p; x, y', z, p') f_{j'}(k_j', x') \times \right]
\]

\[
\sum_z C_k^Q (x, y, z, p; x, y, z', p') f_{j'}(k_j', x')
\]

where notational changes have been made for compatibility with recent discussion. Remember that \( i \) is shorthand for \( x, y, z \) and \( j \) is shorthand for \( k_j', k_j'', k_j' \). The 3-D integral in (36) has been replaced by three integrals over each dimension (plus parameters) in (40). At the resolution of 360 \( \times \) 288 \( \times \) 70 and with five parameters, this reduces the operation count from 36 288 000 to just 3590. This is 10 000 times more efficient.

Putting together (16), (17), (18), (19) and (40) gives

\[
(P_{l})_{ip,j} = \frac{1}{(K-1)} \sum_{k=1}^{K} \sum_{l=1}^{L} (\chi_k)_{ip} (\Omega_l^{1/2})_{ip,ld} (\chi_k)_{l'l'} (\Omega_l^{1/2})_{l'l',ld},
\]

\[
= \frac{\mu_{ij} \mu_{l'l'}}{(K-1)} \sum_{k=1}^{K} \sum_{l=1}^{L} (\chi_k)_{ip} (C_k^Q E \Lambda^{1/2})_{ip,ld} (\chi_k)_{l'l'} (C_k^Q E \Lambda^{1/2})_{l'l',ld},
\]

\[
= \frac{\mu_{ij} \mu_{l'l'}}{(K-1)} \sum_{k=1}^{K} (\chi_k)_{ip} \times
\]

\[
\sum_{l=1}^{L} \Lambda^{1/2}(k_j', k_j', k_j') \left[ \sum_{i',p'} \sum_x C_k^Q (x, y, z, p; x', y, z, p') f_{j'}(k_j', x') \times \right.
\]

\[
\left. \sum_y C_k^Q (x, y, z, p; x, y', z, p') f_{j'}(k_j', x') \sum_z C_k^Q (x, y, z, p; x, y, z', p') f_{j'}(k_j', x') \right],
\]

where notational changes have been made for compatibility with recent discussion. Remember that \( i \) is shorthand for \( x, y, z \) and \( j \) is shorthand for \( k_j', k_j'', k_j' \). The 3-D integral in (36) has been replaced by three integrals over each dimension (plus parameters) in (40). At the resolution of 360 \( \times \) 288 \( \times \) 70 and with five parameters, this reduces the operation count from 36 288 000 to just 3590. This is 10 000 times more efficient.
Adaptive Localization: Proposals for a high-resolution multivariate system

\[
\sum_{y''} C_k^L(x, y, z, p; x, y'', z, p'') f_{\mu_p}^y (k_i, y'') \times \\
\sum_{z''} C_k^L(x, y, z, p; x, y, z'', p'') f_{\mu_p}^{z''} (k_i, z'') \times \\
A^{1/2} (k^x_i, k^y_i, k^z_i) \sum_{p'} \left[ \sum_{x''} C_k^L(x', y', z', p'; x', y', z', p'') f_{\mu_p}^{x''} (k_i, x'') \times \\
\sum_{y''} C_k^L(x', y', z', p'; x', y'', z', p'') f_{\mu_p}^{y''} (k_i, y'') \times \\
\sum_{z''} C_k^L(x', y', z', p'; x', y', z', p'') f_{\mu_p}^{z''} (k_i, z'') \right],
\]

where \( \mu_{\mu_p} = \left( \sum_{l=1}^{L} \left( \sum_{p'} \sum_{x'} C_k^L(x, y, z, p; x', y, z, p'') f_{\mu_p}^{x'} (k_i, x') \times \\
\sum_{y''} C_k^L(x, y, z, p; x, y'', z, p'') f_{\mu_p}^{y''} (k_i, y'') \times \\
\sum_{z''} C_k^L(x, y, z, p; x, y, z'', p'') f_{\mu_p}^{z''} (k_i, z'') \right) \right)^{-1/2} \).

The suggested algorithm is now given for this case. Costs for each loop are specified at the end of each loop in blue. Numerical terms in blue brackets are for the specific model domain (\(n = 360 \times 288 \times 70\) and 5 parameters). Assume for now that \(L = 50\).

1. ===== Calculation of term in (41) for (fixed) \(i', p'\) (\(i'\) denotes a particular \(x', y', z'\)) =====
2. Loop round \(p''\)
3. Loop round \(x''\)
4. Evaluate \(\chi_{i'}^{p'} (x'') = C_k(x', y', z', p'; x'', y', z', p'') K\) (24)
5. End loop (\(x''\)) \(360K\) (8 640)
6. Loop round \(y''\)
7. Evaluate \(\chi_{i'}^{p'} (y'') = C_k(x', y', z', p'; x', y'', z', p'') K\) (24)
8. End loop (\(y''\)) \(288K\) (6 912)
9. Loop round \(z''\)
10. Evaluate \(\chi_{i'}^{p'} (z'') = C_k(x', y', z', p'; x', y', z'', p'') K\) (24)
11. End loop (\(z''\)) \(70K\) (1 680)
12. End loop (\(p''\)) \(3 590K\) (86 160)
13. \(\alpha (\cdot) = 0\)
14. \(\beta = 0\)
15. Loop round \(l\)
16. Loop round \(p''\)
Adaptive Localization: Proposals for a high-resolution multivariate system

17. \( \alpha^x = 0 \)
18. Loop round \( x'' \)
19. \( \alpha^x = \chi_{x'}^x (x'') f_{x'}^x (k_i^x, x'') \)
20. End loop \( (x'') \) \( 360 \) \( (360) \)
21. \( \alpha^y = 0 \)
22. Loop round \( y'' \)
23. \( \alpha^y = \chi_{y'}^y (y'') f_{y'}^y (k_j^y, y'') \)
24. End loop \( (y'') \) \( 288 \) \( (288) \)
25. \( \alpha^z = 0 \)
26. Loop round \( z'' \)
27. \( \alpha^z = \chi_{z'}^z (z'') f_{z'}^z (k_l^z, z'') \)
28. End loop \( (z'') \) \( 70 \) \( (70) \)
29. \( \alpha (l) = \alpha^x \alpha^y \alpha^z \)
30. End loop \( (p''') \) \( 3 \) \( 590 \) \( (3 \) \( 590) \)
31. \( \alpha (l) = \Lambda^{1/2} (k_i^x, k_j^y, k_l^z) \)
32. \( \beta = \alpha^2 (l) \)
33. End loop \( (l) \) \( 3 \) \( 590L \) \( (179 \) \( 500) \)
34. \( \mu_{i,j'} = \sqrt{\chi(p') } \)
35. ===== Loop around destination points in the structure function =====
36. Loop round \( i, p \) \( (i \) denotes a particular \( x, y, z) \)
37. ===== Calculation of static term in \( (41) \) =====
38. \( \gamma = 0 \)
39. Loop round \( k \)
40. \( \gamma = (x_h)_{i,p} (x_h)_{i,j'} \)
41. End loop \( (k) \) \( K \) \( (24) \)
42. ===== Calculation of term in \( (41) \) for (variable) \( i, p \) =====
43. Loop round \( p'' \)
44. Loop round \( x'' \)
45. Evaluate \( \chi_{x'}^x (x'') = C_K (x, y, z, p; x'', y, z, p'') \) \( K \) \( (24) \)
46. End loop \( (x'') \) \( 360K \) \( (8 \) \( 640) \)
47. Loop round \( y'' \)
48. Evaluate \( \chi_{y'}^y (y'') = C_K (x, y, z, p; x, y'', z, p'') \) \( K \) \( (24) \)
49. End loop \( (y'') \) \( 288K \) \( (6 \) \( 912) \)
50. Loop round \( z'' \)
51. Evaluate \( \chi_{z'}^z (z'') = C_K (x, y, z, p; x, y, z'', p'') \) \( K \) \( (24) \)
52. End loop \( (z'') \) \( 70K \) \( (1 \) \( 680) \)
53. End loop \( (p''') \) \( 3 \) \( 590K \) \( (86 \) \( 160) \)
54. \( \epsilon (:) = 0 \)
55. \( \beta = 0 \)
56. Loop round $l$
57. Loop round $p''$
58. $\epsilon^x = 0$
59. Loop round $x''$
60. $\epsilon^x = \chi_{p''}^x(x'')f_{p''}^x(k_i, x'')$
61. End loop ($x''$) 360 (360)
62. $\epsilon^y = 0$
63. Loop round $y''$
64. $\epsilon^y = \chi_{p''}^y(y'')f_{p''}^y(k_i, y'')$
65. End loop ($y''$) 288 (288)
66. $\epsilon^z = 0$
67. Loop round $z''$
68. $\epsilon^z = \chi_{p''}^z(z'')f_{p''}^z(k_i, z'')$
69. End loop ($z''$) 70 (70)
70. $\epsilon (l) = \epsilon^x\epsilon^y\epsilon^z$
71. End loop $p''$ 3 590 (3 590)
72. $\epsilon (l) = \Lambda^{1/2}(k_i, k_i, k_i)$
73. $\beta = \epsilon^2 (l)$
74. End loop ($l$) 3 590L (179 500)
75. $\mu_{ip} = \sqrt{1/\beta}$
76. Structure function for $i, p$ can be evaluated - see (33)
77. End loop ($i, p$) 5n $[K + 3 590K + 3 590L]$ (10$^{13}$)

This cost can be reduced by looping only round those $i$ that are in the same plane as $i'$. Instead of multiplying by $n$, this multiple in the last line is

$$360 \times 288 + 288 \times 70 + 360 \times 70 = 103 680 + 20 160 + 25 200 = 149 040.$$  This reduced cost is $745 200[265 684] \approx 2 \times 10^{11}$.

References
