Fast CSD

Problem is to minimise per iteration:

$$F(f_n) = ||Mf_n - s||^2 + \lambda||H_{n-1}f_n||^2$$

Where M maps current FOD SH coefficients f_n to DW signals s, and H_{n-1} maps FOD SH coefficients f_n to amplitudes along set of negative directions identified in previous iteration (i.e. the matrix formed by the rows of H for which $Hf_{n-1} < 0$, where H maps FOD to amplitude along a dense set of (300) directions.

Solve by differentiating and setting to zero:

$$\Rightarrow \frac{\partial F}{\partial f_n} = 2M^T(Mf_n - s) + 2\lambda H_{n-1}{}^T H_{n-1} f_n = 0$$

$$(M^T M + \lambda H_{n-1}{}^T H_{n-1}) f_n = M^T s$$

Define $Q = M^T M + \lambda H_{n-1}{}^T H_{n-1}$, which by construction is a square positive-definite symmetric matrix of size $n_{SH} \times n_{SH}$. If needed, positive definiteness can be enforced with a small minimum norm regulariser (helps a lot with poorly conditioned direction sets and/or super-resolution):

$$Q = M^T M + \lambda H_{n-1}{}^T H_{n-1} + \mu I$$

Solve using Cholesky decomposition:

$$Q = LL^T$$

where L is lower triangular. Then problem can be solved by back-substitution:

$$Ly = M^T s$$

$$L^T f_n = y$$

To speeds things up further, form $P = M^T M + \mu I$, and update to form Q by rank-n update with H_{n-1}. Form $M^T s$ initially, and re-use over iterations since it doesn't change. Algorithm then looks like:

- form $P = M^T M + \mu I$
- for each voxel:
  - form $z = M^T s$
  - estimate f_0 by solving $Pf_0 = z$ (note in practice I use a simplified $l_{max} = 4$ solution here, but I'm not sure it makes a great deal of difference what the initial FOD actually is).
  - then iterate until no change in rows of H used in H_n:
    - form H_{n-1} given f_{n-1}
    - form $Q = P + \lambda H_{n-1}{}^T H_{n-1}$ by rank-n update
    - solve $Qf_n = z$ using Cholesky decomposition