

4 Algorithm analysis

4.1 Memory requirement

In this section, we elaborate on the memory consumption of the proposed method in comparison with SEAGLE. First, let us state that gradient based methods, such as NAGD or CG, have similar memory requirements. It corresponds roughly to three times the size of the optimization variable which is the part that is common to both algorithms. The additional memory requirement that is specific to SEAGLE relies only on the storage of the NAGD iterates during the forward computation. Suppose that $K_{\text{NAGD}} \in \mathbb{N}$ iterations are necessary to compute the forward model with (9) and that the region Ω is sampled over $N \in \mathbb{N}$ pixels (voxels, in 3D). Since the total field $\mathbf{u}_p(\mathbf{f})$ computed by NAGD is complex-valued, each pixel is represented with 16 bytes (double precision for accurate computations). Hence, the difference of memory consumption between SEAGLE and our method is

$$\Delta_{\text{Mem}} = N \times K_{\text{NAGD}} \times 16 \text{ [bytes]}, \quad (26)$$

which corresponds to the storage of the K_{NAGD} intermediate iterates of NAGD. Here, we assumed that $\nabla \mathcal{D}$ was computed by sequentially adding the partial gradients $\nabla \mathcal{D}_p$ associated to the P incident fields. Hence, once the partial gradient associated to one incident angle is computed by successively applying the forward model (NAGD) and the error-backpropagation procedure, the memory used to store the intermediate iterates can be recycled to compute the partial gradient associated to the next incident angle. However, when the parallelization strategy detailed in Section is used, the memory requirement is multiplied by the number $N_{\text{Threads}} \in \mathbb{N}$ of threads, so that

$$\Delta_{\text{Mem}} = N \times K_{\text{NAGD}} \times N_{\text{Threads}} \times 16 \text{ [bytes]}. \quad (27)$$

Indeed, since the threads of a single computer share memory, computing N_{Threads} partial gradients in parallel requires N_{Threads} times more memory.

For illustration, we give in Fig. 2 the evolution of Δ_{Mem} as a function of N for different values of K_{NAGD} and N_{Threads} . One can see with the vertical dashed lines that, for 3D volumes, the memory used by SEAGLE quickly reaches several tens of Megabytes, even for small volumes (*e.g.*, $128 \times 128 \times 128$), to hundreds of Gigabytes for the larger volumes that are typical of microscopy (*e.g.*, $512 \times 512 \times 256$). This shows the limitation of SEAGLE for 3D reconstruction in the presence of a shortage of memory resources and reinforces the interest of the proposed alternative.

4.2 Conjugate gradient *vs.* Nesterov accelerated gradient descent for (9)

Due to Proposition 3.1, we can compute both (9) and $\mathbf{J}_{h_p}^H(\mathbf{f})$ using any state-of-the-art quadratic optimization algorithm. This contrasts with SEAGLE, where one must derive the error-backpropagation rule from the forward algorithm, which may limit its choice. We now provide numerical evidence that GC is more efficient than NAGD for solving (9). To this end, we consider a circular object (bead) of radius r_{bead}