Comprehensive Holographic Parallel Beam Modulation inside Material based on Automatic Differentiation: Supplementary Document

1. Derivation of Eq. 1 & Eq. 2 in article

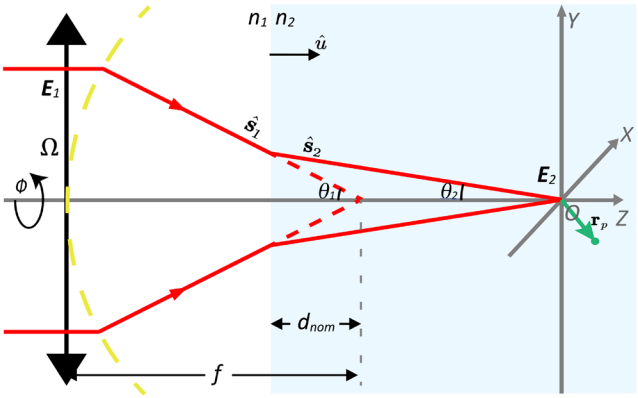

Fig. S1 Schematic diagram of electric field inside the material.

In the framework of tight focus, at any point $P(x,y,z)$ in the focal region, the electric vectors $\mathbf{E}$ can be expressed in the form as a summation of the plane waves that are leaving the aperture:

$$\mathbf{E}(x,y,z) = -\frac{ik}{2\pi}\iint_{\Omega}\frac{\mathbf{a}(s_x,s_y)}{s_z}\exp(ik\hat{\mathbf{s}}^{T}\cdot\mathbf{r}_p)ds_x ds_y, \tag{S1}$$

where k is the wave number, Ω is the exit aperture of the lens, and $\mathbf{a}(s_x,s_y)$ is the electric complex amplitude in the exit aperture. $\hat{\mathbf{s}}$ is a unit vector pointing from a point in the exit aperture to the focus, $\hat{\mathbf{s}} = [s_x,s_y,s_z]^T$, $\mathbf{r}_p$ is a vector pointing from a focus to P in the focal region, $\mathbf{r}_p = [x,y,z]^T$.

Now the focal region is moved into the material as Fig. S1. n_1 and n_2 are the refractive index (RI) of the atmosphere and the material. In view of the RI mismatch, we need to explore the behavior of the electric field in the interface to obtain a comprehensive solution of electric field inside the material. In the interface inside atmosphere ($z = -d_{nom}$), the electric field is given by

$$\mathbf{E}_1(x,y,-d) = -\frac{ik_1}{2\pi}\iint_{\Omega_1}\mathbf{a}(s_{1x},s_{1y})/s_{1z}\times\exp[ik_1(s_{1x}x+s_{1y}y-s_{1z}d_{nom})]ds_{1x}ds_{1y}. \tag{S2}$$

Eq. 2 is in a form of the sum of plane wave components. We assuming that the field in the material is also a sum of refracted plane wave components. The transformation of incident plane wave to refracted plane wave obeys the Fresnel law, so the refracted plane wave can be described as a linear function of $\mathbf{a}(s_{1x},s_{1y})/s_{1z}$, i.e. $\mathbf{T}\cdot\mathbf{a}(s_{1x},s_{1y})/s_{1z}$, where $\mathbf{T}$ is a refraction operator that is a function of angle of incidence and n_1, n_2. Therefore, the electric field on the interface of material is derived as

$$\mathbf{E}_1(x,y,-d) = -\frac{ik_1}{2\pi}\iint_{\Omega_1}\mathbf{T}\cdot\mathbf{a}(s_{1x},s_{1y})/s_{1z}\cdot\exp[ik_1(s_{1x}x+s_{1y}y-s_{1z}d_{nom})]ds_{1x}ds_{1y}, \tag{S3}$$

On the other hand, as Török et al. [1] suggested, we can also represent the field in the second material as a sum of plane waves

$$\mathbf{E}_2(\mathbf{r}_p) = -\frac{ik_2}{2\pi}\iint_{\Omega_2}\mathbf{F}(\hat{\mathbf{s}}_2)\exp(ik_2\hat{\mathbf{s}}_2{}^{T}\cdot\mathbf{r}_p)ds_{2x}ds_{2y}. \tag{S4}$$

in which $\mathbf{F}(\hat{\mathbf{s}}_2)$ is to be determined. Next, we will calculate $\mathbf{F}(\hat{\mathbf{s}}_2)$ using Eq. S3 as a boundary condition of Eq. S5. Firstly, we will establish the relationship between $\hat{\mathbf{s}}_1$ and $\hat{\mathbf{s}}_2$.

Based on the vectorial law of refraction, there is

$$k_2\hat{\mathbf{s}}_2 - k_1\hat{\mathbf{s}}_1 = (k_2\cos\theta_2 - k_1\cos\theta_1)\hat{\mathbf{u}}, \tag{S5}$$

where $\hat{\mathbf{u}}$ represents the normal of the interface, so we can get

$$k_2 s_{2x} = k_1 s_{1x}, k_2 s_{2y} = k_1 s_{1y}. \tag{S6}$$

Now Eq. S4 is rewritten as

$$\mathbf{e}_2(\mathbf{r}_p) = -\frac{ik_2}{2\pi}\iint_{\Omega_1}\mathbf{F}(\hat{\mathbf{s}}_2)\exp\left(ik_2\hat{\mathbf{s}}_2{}^{\mathrm{T}}\cdot\mathbf{r}_p\right)\cdot\det\left(\mathbf{J}_0\left(s_{1x},s_{1y};s_{2x},s_{2y}\right)\right)\mathrm{d}s_{1x}\mathrm{d}s_{1y}, \tag{S7}$$

where $\mathbf{J}_0\left(s_{1x},s_{1y};s_{2x},s_{2y}\right)$ is the Jacobian matrix introduced by coordinate substitution, and $\det\left(\mathbf{J}_0\left(s_{1x},s_{1y};s_{2x},s_{2y}\right)\right)$ equals to $(k_1/k_2)^2$ according to Eq. S6. It is easily to find that Eq. S7 satisfies the boundary condition represented by Eq. S3 when

$$\mathbf{F}(\hat{\mathbf{s}}_2) = \frac{k_2}{k_1}\mathbf{T}\frac{\mathbf{a}\left(s_{1x},s_{1y}\right)}{s_{1z}}\exp[-id_{nom}(k_1 s_{1z} - k_2 s_{2z})]. \tag{S8}$$

Now we can combine Eq. S8 and Eq. S7, and obtain

$$\mathbf{e}_2(\mathbf{r}_p) = -\frac{ik_1}{2\pi}\cdot$$
$$\iint_{\Omega_1}\mathbf{T}\cdot\frac{\mathbf{a}\left(s_{1x},s_{1y}\right)}{s_{1z}}\exp[-id_{nom}(k_1 s_{1z} - k_2 s_{2z})]\exp(ik_2 s_{2z} z)\exp\left[ik_1\left(s_{1x}x + s_{1y}y\right)\right]\mathrm{d}s_{1x}\mathrm{d}s_{1y}. \tag{S9}$$

Eq. S9 is important in this derivation, where the first exponential term stands for the aberration and focus shifting introducing by RI mismatch, the second exponential term stands for the phase accumulation along depth, and the third exponential term stands for phase difference of the wave front at off-axis points.

To systematize Eq. S9 into a form of Fourier transform, some coordinate substitution is needed. We have

$$\boldsymbol{k}_1 = \begin{pmatrix} k_{1x} \\ k_{1y} \\ k_{1z} \end{pmatrix} = k_1\begin{pmatrix} -s_{1x} \\ -s_{1y} \\ s_{1z} \end{pmatrix} = k_1\begin{pmatrix} -\sin\theta_1\cos\phi \\ -\sin\theta_1\sin\phi \\ \cos\theta_1 \end{pmatrix}, \tag{S10}$$

$$\boldsymbol{k}_2 = \begin{pmatrix} k_{2x} \\ k_{2y} \\ k_{2z} \end{pmatrix} = k_2\begin{pmatrix} -s_{2x} \\ -s_{2y} \\ s_{2z} \end{pmatrix} = k_2\begin{pmatrix} -\sin\theta_2\cos\phi \\ -\sin\theta_2\sin\phi \\ \cos\theta_2 \end{pmatrix}, \tag{S11}$$

so $\exp[-id_{nom}(k_1 s_{1z} - k_2 s_{2z})]$ can be written as $\exp[id_{nom}(k_2\cos\theta_2 - k_1\cos\theta_1)] = \exp\left(i\psi(\theta_1,\theta_2,d_{nom})\right)$, $\exp(ik_2 s_{2z} z)$ can be written as $\exp(ik_{2z} z)$, and $\exp\left[ik_1\left(s_{1x}x + s_{1y}y\right)\right]$ can be written as $\exp\left[-i\left(k_{1x}x + k_{1y}y\right)\right]$. Variable of integration now will be changed to $\mathrm{d}k_{1x}\mathrm{d}k_{1y}$. Therefore, we obtain

$$\begin{aligned}\mathbf{E}_2(x,y,z) &= -\frac{i}{2\pi k_1}\iint_{\Omega_1}\mathbf{T}\cdot\frac{\mathbf{a}\left(s_{1x},s_{1y}\right)}{\cos\theta_1}\exp(i\psi)\exp(ik_{2z}z)\exp\left[-i\left(k_{1x}x + k_{1y}y\right)\right]\mathrm{d}k_{1x}\mathrm{d}k_{1y} \\ &= -\frac{i}{2\pi k_1}\mathcal{F}\left\{\mathbf{T}\cdot\frac{\mathbf{a}\left(s_{1x},s_{1y}\right)}{\cos\theta_1}\cdot\exp(i\psi)\exp(ik_{2z}z)\right\}.\end{aligned} \tag{S12}$$

In the scope of the article, the incident beam is x-polarized light. As a result, in Eq. S12, $\mathbf{T}\cdot\mathbf{a}\left(s_{1x},s_{1y}\right)$ now can be written as (some constants are omitted)

$$\mathbf{T}\cdot\mathbf{a}\left(s_{1x},s_{1y}\right) = \mathbf{T}\cdot E_1 = f\sqrt{\cos(\theta_1)}\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{12} & a_{22} & a_{23} \\ -a_{13} & a_{23} & a_{33} \end{bmatrix}\begin{bmatrix} E_1 \\ 0 \\ 0 \end{bmatrix}, \tag{S13}$$

where f is the focal length, E_1 is the electric complex amplitude of x-polarized component. Values in the matrix is listed as

$$\begin{aligned} a_{11} &= \tau_p\cos^2\phi\cos\theta_2 + \tau_s\sin^2\phi, \\ a_{12} &= \left(\tau_p\cos\theta_2 - \tau_s\right)\cos\phi\sin\phi, \\ a_{13} &= \tau_p\cos\phi\sin\theta_2, \\ a_{22} &= \tau_s\cos\phi^2 + \tau_p\cos\theta_2\sin^2\phi, \\ a_{23} &= \tau_p\sin\phi\sin\theta_2, \\ a_{33} &= \tau_p\cos\theta_2, \end{aligned} \tag{S14}$$

Thus, using the 2-dimensional Fourier transform in Eq. S12, we can realize internal light field computation with whole polarization components fast.

2. Automatic differentiation

We review reverse-mode automatic differentiation for better understand of error back propagation in Fig. 2. For a detailed overview, see [2]. Automatic differentiation is based on the chain rule for partial derivatives.

First, if we have a composite function as

$$E = f_3(\boldsymbol{v_2}), \boldsymbol{v_2} = f_2(\boldsymbol{v_1}), \boldsymbol{v_1} = f_1(\boldsymbol{x})$$

Where the bolded variables are vectors, like $\boldsymbol{v_2}$ contains $v_{2,1}, \cdots, v_{2,k}, \cdots, v_{2,K}$. E is the final single-variable real-valued function, like the cost function we have used. $\boldsymbol{x}$ is the initial independent variable, e.g., the light path difference on each pixel we use.

Here, we define

$$\overline{\boldsymbol{v_2}} = \frac{\partial E}{\partial \boldsymbol{v_2}}, \overline{\boldsymbol{v_1}} = \frac{\partial E}{\partial \boldsymbol{v_1}}, \overline{\boldsymbol{x}} = \frac{\partial E}{\partial \boldsymbol{x}}, \tag{S15}$$

and it can be derived that

$$(\overline{\boldsymbol{v_2}})_k = \frac{\partial E}{\partial v_{2,k}}, \tag{S16}$$

$$(\overline{\boldsymbol{v_1}})_w = \sum_k \frac{\partial E}{\partial v_{2,k}} \frac{\partial v_{2,k}}{\partial v_{1,w}} = \sum_k (\overline{\boldsymbol{v_2}})_k \frac{\partial v_{2,k}}{\partial v_{1,w}}, \tag{S17}$$

$$(\overline{\boldsymbol{x}})_r = \sum_w \frac{\partial E}{\partial v_{1,w}} \frac{\partial v_{1,w}}{\partial x_r} = \sum_w (\overline{\boldsymbol{v_1}})_w \frac{\partial v_{1,w}}{\partial x_r}. \tag{S18}$$

It can be noticed that the chain rule has been written in a form of matrix multiplication, where the first multiplier vector can be obtained from the previously derived partial differentiation and passed on continuously, and the latter multiplier matrix can be obtained from the Jacobian matrix in this step.

In practice, there are some noteworthy details, such as the operation for complex functions, the operation for linear operators (e.g., FFT), and so on. At the same time, it is not necessary to calculate complicated matrix multiplication every time because some of the analytic expressions have been given in the reference [3]. Using some established open-source libraries may also be convenient [4,5].

3. CZT & inverse CZT

The chirp z-transform (CZT) is defined as follows:

$$\mathrm{X}_k = \sum_{t=0}^{T-1} \mathrm{x}_t A^{-t} W^{tk}, k = 0,1, \cdots, M-1 \tag{S19}$$

when $A = 1$ and $W = \exp(-2\pi i/M)$, it becomes to conventional Discrete Fourier Transform (DFT). By manipulating A and W, some special spectrum (the output array of the transform X_k) can be obtained. CZT gradually becomes popular in computing optics because it can realize nondestructive zoom-in of light field.

Similar to the relationship between FFT and iFFT, we naturally want to find the inverse procedure of CZT for using it in iteration or error backpropagation. However, inverse CZT for arbitrary parameters and input is not always valid. Considering that in phase retrieval problems, we only need the inverse CZT algorithm that evenly sampled on the unit circle, we do not expect a universal inverse CZT. Instead, we give a parameter-restricted but usable inverse CZT algorithm procedure in algorithm S1 based on the function "CZT()" that already available in Matlab[6].

Algorithm S1. Inverse 2D CZT

Inputs : F_{in} a $m_1 \times m_1$ matrix at which inverse CZT will be performed
$Ratio$ a parameter about the zoom-in ratio
m a parameter depending the matrix size after inverse CZT
Outputs : F_{out} a $m \times m$ matrix at which inverse CZT has been performed
procedure INVERSE_CZT(F_{in}, $Ratio$, m)
$m_1 \leftarrow$ length(F_{in}) ▷obtaining the size of input matrix F_{in}
$W \leftarrow \exp(-2\pi i Ratio/m_1)$ ▷defining the point interval on the unit circle W, same as CZT
$O_1 \leftarrow$ czt(F_{in}, m, $conj(W)$, 1) ▷performing CZT with complex conjugate of W
$X \leftarrow$ linspace(1, m, m)^T ▷obtaining a arithmetic progression from 1 to m
$X \leftarrow$ repmat(X, [1, m_1]) ▷copy X m_1 times
$O_1 \leftarrow O_1/\exp(i\pi Ratio X)\cdot Ratio/m_1$ ▷performing phase and amplitude compensations
$X \leftarrow$linspace(1, m, m)^T ▷ obtaining a arithmetic progression from 1 to m
$X \leftarrow$ repmat(X, [1, m]) ▷ copy X m times
$O_2 \leftarrow$ czt($O_1{}^T$, m, $conj(W)$, 1) ▷ performing CZT alone the other dimension
$O_2 \leftarrow O_2/\exp(i\pi Ratio X)\cdot Ratio/m$ ▷ performing phase and amplitude compensations
$F_{out} \leftarrow O_2{}^T\cdot\exp(i2\pi Ratio)$ ▷ transposing the matrix to correct pose
return F_{out}

In the algorithm S1, some functions are official built-in functions of Matlab. repmat(X, [1, m]) returns an matrix that contains m copies of input in its column dimension. czt(input, m, W, A) returns the m-length chirp Z-transform (CZT) of input along the spiral contour defined by W and A.

There are limitations for the proposed algorithm S1. The first one is the contour must along the unit circle and be evenly sampled. m should be an even number. $Ratio$ should be a non-negative integer power of $1/2$, like $1/4$ and $1/32$. Besides, considering in the vast majority of cases, the region of interest is located at the center of the entire image plane, algorithm S1 is set to only zoom into the center. Algorithm S1 is also limited to the square matrix input. Admittedly, Algorithm S1 is with many limitations, but in most practical optics problems it is enough and works well.

An example is given to verify the performance of algorithm S1. We generated a random complex matrix to be the input (as shown in Fig. S1 (a) and (b)), and performed CZT with different $Ratio$s. Subsequently, the output of CZT is used as the input of algorithm S1, and corresponding inverse CZT is performed to recover the original matrix. The results are shown in Fig. S1 to illustrate the validity in different situations. It can be seen that with the decrease of the $Ratio$, the outputs of CZT, i.e. the spectrums, gradually zoom in to the center at higher magnifications, and are more detailed. After inverse CZT, the amplitude and phase are in full accord with original matrix in Fig. S1 (a) and (b) when $Ratio$ is relatively large, but in the case that $Ratio$ equals to $1/64$, Fig. S1 (e7) and (f7) is significantly different with original matrix. The reason is the scale of the spectrum (Fig. S1 (d7)) is so small that large proportion of the spectrum is no longer contained. Fig. S1 (c) illustrates the numerical error in the 7 cases of different $Ratio$s corresponding to the lower images.

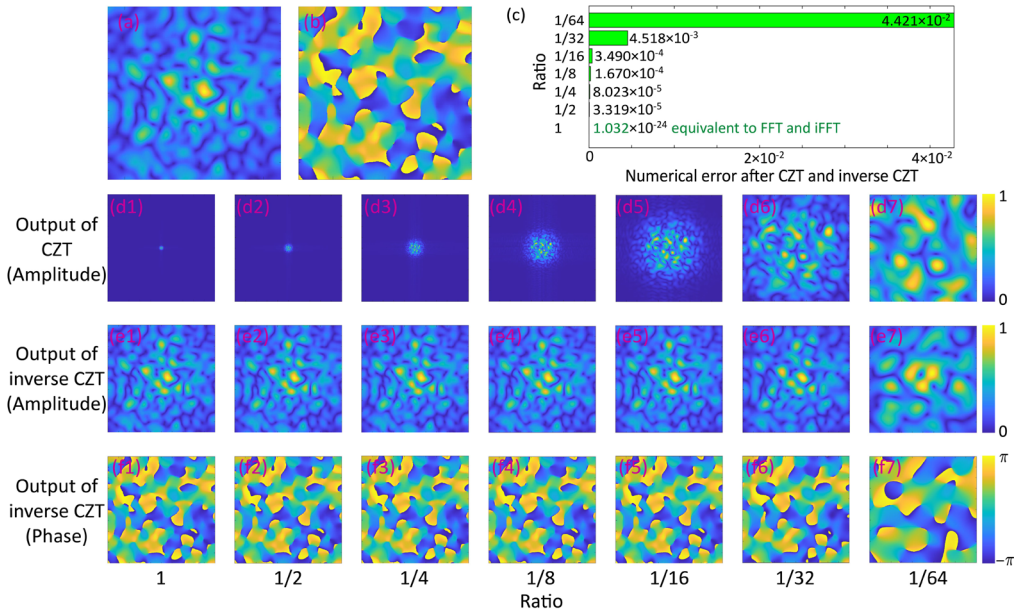

Fig. S2. The test result of inverse CZT algorithm presented in algorithm S1. (a), (b) The amplitude and phase distributions of the original matrix as the input of the algorithm. (c) The numerical error of the 7 tests corresponding to different $Ratio$s from 1 to $1/64$. The values of error are labeled in the figure. (d1) - (d7) The spectrum obtained with different $Ratio$s. (e1) - (e7) The amplitude of the output of inverse CZT. (e1) - (e7) The phase of the output of inverse CZT.

4. Superposition algorithm, GSW algorithm and 3DIFTA

Here we give the pseudocodes of superposition algorithm [7], GSW algorithm [8] and 3DIFTA [9]. Both superposition algorithm and GSW algorithm require the defocusing phase of each target focus, so an assistant function is presented first.

Algorithm S2. Generating defocusing phase

Inputs : k wavenumber in vacuo
f focus distance of objective lens
n_2 refractive index of material
$\triangle x$ an array containing all the x-direction defocusing amounts for all the foci
$\triangle y$ an array containing all the y-direction defocusing amounts for all the foci
$\triangle z$ an array containing all the z-direction defocusing amounts for all the foci
Outputs : Φ_d a 3D matrix containing all defocusing phase pattern for all the foci

procedure GENERATING_DEFOCUSING_PHASE (k, f, n_2, $\triangle x$, $\triangle y$, $\triangle z$)
  $N \leftarrow$ length($\triangle x$) ▷obtaining the number of all foci
  for $n = 1$ **to** N **do**
    $PS_X \leftarrow k\triangle x \cdot x/f$ ▷obtaining the x-direction phase shift component
    $PS_Y \leftarrow k\triangle y \cdot y/f$
    $PS_Z \leftarrow -k\triangle z \cdot \mathrm{sqrt}(n_2^2-(x^2+y^2)/f^2)$
    $\Phi_d[n] \leftarrow PS_X+PS_Y+PS_Z$ ▷obtaining the overall phase shift component
  return Φ_d

the output of Algorithm S2, Φ_d will be an input of Algorithm S3 and Algorithm S4.

Algorithm S3. Superposition algorithm

Inputs : Φ_d a 3D matrix containing all defocusing phase pattern for all the foci
Outputs : $\Phi_{superposition}$ phase pattern generated
procedure SUPERPOSITION_ALGORITHM (Φ_d)
    $N \leftarrow$ length(Φ_d, 3) ▷obtaining the number of all foci
    for $n = 1$ **to** N **do**
        $U[n] \leftarrow \exp(i\Phi_d[n] + i2\pi \cdot rand[1])$ ▷introducing a random phase offset
        $\Phi_{superposition} \leftarrow \arg(\sum_n U[n])$ ▷overlaying all the defocusing phase
    return $\Phi_{superposition}$

Algorithm S4. GSW algorithm

Inputs : *maxiter* max iteration number
        Φ_d a 3D matrix containing all defocusing phase pattern for all the foci
Outputs : Φ_{GSW} phase pattern generated
procedure GSW_ALGORITHM (*maxiter*, Φ_d)
    $N \leftarrow$ length(Φ_d, 3) ▷obtaining the number of all foci
    $M \leftarrow$ length(Φ_d, 1) ▷obtaining the resolution of phase pattern
    $\Phi_{GSW} \leftarrow$ ones(M, M)
    $\Omega \leftarrow$ ones(N, 1)
    while t<*maxiter* **do**
        $t \leftarrow t+1$ ▷obtaining the number of all foci
        for $n = 1$ **to** N **do**
            $U_1 \leftarrow \exp(i\Phi_{GSW}) / \exp(i\Phi_d[n])$ ▷obtaining the complex
            $V[n] \leftarrow \text{sum}(U_1)/L$ ▷obtaining a value that represents the light field on focus n
        $\Omega \leftarrow \Omega \cdot \text{mean}(\text{abs}(V)) / \text{abs}(V)$ ▷updating Ω with V
        for $n = 1$ **to** N **do**
            $U_2[n] \leftarrow \exp(i\Phi_d[n]) \cdot \Omega[n] \cdot V[n] / \text{abs}(V)$ ▷updating U_2 with V and the weight Ω
        $\Phi_{GSW} \leftarrow \arg(\sum_n U_2[n])$ ▷updating Φ_{GSW} by overlaying all $U_2[n]$
    return Φ_{GSW}

Algorithm S4. 3DIFTA

Inputs : *maxiter* max iteration number
M_X resolution along x axis
M_Y resolution along y axis. $M_Y = M_X$ is required.
M_Z resolution along z axis
Img3d_target target light intensity distribution as a 3D matrix
Outputs : Φ_{3DIFTA} phase pattern generated
procedure 3DIFTA ($maxiter, M_X, M_X, M_X, \Phi_d$)
$K3d \leftarrow$ zeros(Mx,My,Mz)
$X \leftarrow$ [$-M_X/2+1:1:-M_X/2$]
$Y \leftarrow$ [$-M_Y/2+1:1:-M_Y/2$]
$Z \leftarrow$ [$-M_X+1:1:-M_X+M_Z$]
[X,Y,Z] $\leftarrow$ meshgrid(X,Y,Z) ▷obtaining coordinates along x, y, and z axis.
$k\_r \leftarrow$ sqrt($X^2+Y^2+Z^2$) ▷obtaining radius of the Ewald cap in k-space
$K3d \leftarrow$ zeros(Mx,My,Mz)
$K3d(k\_r<Mx \ \& \ k\_r>Mx\text{-}3) \leftarrow 1$ ▷obtaining the Ewald cap, with a thickness of 3
for $n1 = 1$ **to** M_X **do** ▷only remaining the outermost cap
for $n2 = 1$ **to** M_Y **do**
for $n3 = 1$ **to** M_Z **do**
if $K3d(n1, n2, n3) = 1$ **then**
$K3d(n1, n2, n3: M_Z) \leftarrow 0$
$k3d\_target \leftarrow K3d$
while $t<maxiter$ **do**
$t \leftarrow t+1$
$Img3d \leftarrow$ fftshift(fft3($k3d$)) ▷simulating image space
$Img3d \leftarrow Img3d\_target \cdot \exp(j \cdot \text{angle}(Img3d))$ ▷implementing target constraint
$K3d \leftarrow$ ifft3(ifftshift($Img3d$)) ▷simulating k-space
$K3d \leftarrow k3d\_target \cdot \exp(j \cdot \text{angle}(K3d))$ ▷implementing physical constraint
$\Phi_{3DIFTA} \leftarrow$ sum($K3d$, 3) ▷squeeze 3D Ewald cap to 2D phase pattern
return Φ_{3DIFTA}

References

1. P. Török, P. Varga, Z. Laczik, and G. Booker, “Electromagnetic diffraction of light focused through a planar interface between materials of mismatched refractive indices: an integral representation,” J. Opt. Soc. Am. A 12, 325–332 (1995).
2. A. H. Gebremedhin and A. Walther, "An introduction to algorithmic differentiation," WIREs Data Mining Knowl Discov **10**(1), (2020).
3. A. S. Jurling and J. R. Fienup, "Applications of algorithmic differentiation to phase retrieval algorithms," J. Opt. Soc. Am. A 31(7), 1348 (2014).
4. B. Bell, “CppAD: a package for C++ algorithmic differentiation,” in Computational Infrastructure for Operations Research (2012), https://github.com/coin-or/CppAD.
5. “Tools for Automatic Differentiation,” https://www.autodiff.org/?module=Tools
6. “czt,” documentation of the chirp Z-transform, https://ww2.mathworks.cn/help/signal/ref/czt.html
7. J. Zhang, N. Pégard, J. Zhong, H. Adesnik, and L. Waller, "3D computer-generated holography by non-convex optimization," Optica 4(10), 1306 (2017).
8. R. Di Leonardo, F. Ianni, and G. Ruocco, "Computer generation of optimal holograms for optical trap arrays," Opt. Express 15(4), 1913 (2007).
9. H. Zhang, J. Xu, H. Li, G. Xu, Y. Xiao, W. Cheng, X. Tang, and Y. Qin, "Modulation of high-quality internal multifoci based on modified three-dimensional Fourier transform," Opt. Lett. 48(4), 900 (2023).