

# Revisiting PSF models: Unifying framework and high-performance implementation

Yan Liu<sup>1</sup> | Vasiliki Stergiopoulou<sup>2,3</sup> | Jonathan Chuah<sup>1</sup> | Eric Bezzam<sup>2</sup> |  
Gert-Jan Both<sup>4</sup> | Michael Unser<sup>1</sup> | Daniel Sage<sup>1</sup> | Jonathan Dong<sup>1</sup>

<sup>1</sup>Biomedical Imaging Group, École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland

<sup>2</sup>Audiovisual Communications Laboratory, École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland

<sup>3</sup>Galatea Laboratory, École Polytechnique Fédérale de Lausanne, Neuchâtel, Switzerland

<sup>4</sup>HHMI Janelia Research Campus, Ashburn, Virginia, USA

## Correspondence

Jonathan Dong, Biomedical Imaging Group, École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland.  
Email: [jonathan.dong@epfl.ch](mailto:jonathan.dong@epfl.ch)

## Funding information

Schweizerischer Nationalfonds zur Förderung der Wissenschaftlichen Forschung, Grant/Award Numbers: CRSII5\_213521, PZ00P2\_216211

## Abstract

Localisation microscopy often relies on detailed models of point-spread functions. For applications such as deconvolution or PSF engineering, accurate models for light propagation in imaging systems with a high numerical aperture are required. Different models have been proposed based on 2D Fourier transforms or 1D Bessel integrals. The most precise ones combine a vectorial description of the electric field and accurate aberration models. However, it may be unclear which model to choose as there is no comprehensive comparison between the Fourier and Bessel approaches yet. Moreover, many existing libraries are written in Java (e.g., our previous PSF generator software) or MATLAB, which hinders their integration into deep learning algorithms. In this work, we start from the original Richards–Wolf integral and revisit both approaches in a systematic way. We present a unifying framework in which we prove the equivalence between the Fourier and Bessel strategies and detail a variety of correction factors applicable to both of them. Then, we provide a high-performance implementation of our theoretical framework in the form of an open-source library that is built on top of PyTorch, a popular library for deep learning. It enables us to benchmark the accuracy and computational speed of different models and allows for an in-depth comparison of the existing models for the first time. We show that the Bessel strategy is optimal for axisymmetric beams, while the Fourier approach can be applied to more general scenarios. Our work enables the efficient computation of a point-spread function on CPU or GPU, which can then be included in simulation and optimisation pipelines.

## KEYWORDS

localisation microscopy, open-source library, point-spread function, vectorial field propagation

This is an open access article under the terms of the [Creative Commons Attribution](https://creativecommons.org/licenses/by/4.0/) License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

© 2025 The Author(s). *Journal of Microscopy* published by John Wiley & Sons Ltd on behalf of Royal Microscopical Society.

# 1 | INTRODUCTION

The point-spread function (PSF), also referred to as the impulse response, is a fundamental concept that encapsulates key features of an optical microscope. Its monitoring and assessment have been longstanding routines in microscopy, supported by the development of specific tools<sup>1–3</sup> and the creation of a consortium to standardise best practices.<sup>4</sup> A detailed characterisation of the PSF is essential to the design of computational imaging processes such as single-molecule localisation microscopy<sup>5</sup> and super-resolution microscopy, including 3D deconvolution microscopy,<sup>6,7</sup> structured illumination microscopy (SR-SIM),<sup>8,9</sup> fluctuation microscopy,<sup>10,11</sup> stimulated emission-depletion microscopy (STED),<sup>12</sup> and MINIFLUX.<sup>13</sup> The PSF is at the heart of these localisation techniques and helps one to achieve a spatial resolution beyond the optical diffraction limit.

The PSF can be measured experimentally from a z-stack of tiny fluorescent beads.<sup>6,14</sup> However, a theoretical PSF is often required. Here, the challenge lies in the accurate model of light propagation for imaging systems with high numerical aperture (NA). Detailed models have been proposed<sup>15–18</sup> to (a) go beyond simple paraxial approximations, (b) take into account the vectorial nature of the electric field, and (c) include various aberration factors such as the Gibson–Lanni aberrations due to refraction at planar interfaces.<sup>19</sup> Several software packages have been developed to generate precise theoretical 3D PSFs that include various features and aberrations. Among these, the Huygens PSF software<sup>1</sup> is well-known in the microscopy community, and our Java-based ImageJ/FIJI plugin, PSF Generator,<sup>20</sup> has also been widely used in the past. Other open-source alternatives include MATLAB-based solutions<sup>3,21,22</sup> as well as Python-based tools.<sup>23,24</sup>

Computational PSF models have often been applied in fluorescence microscopy. For example, the fitting of a theoretical PSF to a measured one can improve accuracy.<sup>20,25</sup> Detailed models enable the quantification of the 3D uncertainty,<sup>26</sup> while the engineering of PSFs may take full advantage of parametric models.<sup>27–29</sup> They are a key component of high-quality single-molecule localisation microscopy (SMLM) images,<sup>30,31</sup> including single-particle tracking<sup>32,33</sup> and virtual SMLM microscopes.<sup>34,35</sup>

Existing approaches to the computation of the PSF can be categorised into two classes of models: those based on 2D Fourier transforms<sup>3,16</sup> or 1D integrals of Bessel functions.<sup>15,17,18,36</sup> These two classes rely on different assumptions (with Bessel models being more restrictive) and bring different computational trade-offs. Most works focus on one of the two approaches, and the relationship

between them remains unclear. For example, the Gibson–Lanni aberrations<sup>19</sup> or the apodisation factor<sup>15</sup> are only applied in the Bessel case, although they could be generalised to both. To the best of our knowledge, no systematic benchmark of the two strategies has been published, either in terms of accuracy or computational speed.

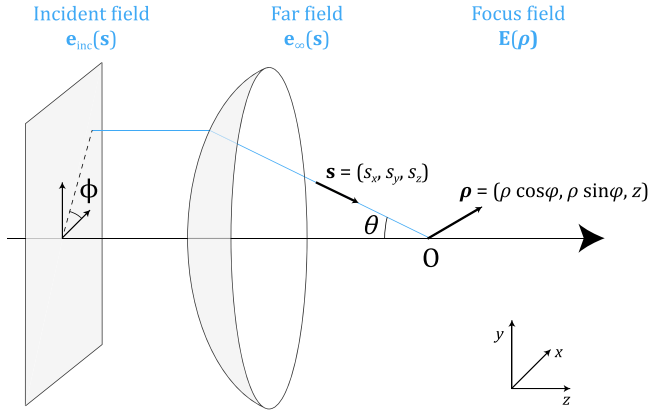
In this work, we propose a unifying framework for PSF models in which we show the equivalence between the Fourier and the Bessel approaches as being two separate parameterisations of the same propagation integral. This enables us to generalise diverse correction factors and apply them to both models. Eventually, the choice is simplified to scalar versus vectorial models, optionally with additional correction factors. We distribute an open-source PyTorch-based library called *psf-generator*. It inherits all the functionalities of PyTorch and allows for a seamless integration with modern learning-based algorithms. We benchmark the two classes of models on CPU and GPU and show that the Bessel model is advantageous for axisymmetric PSFs while the Fourier one is more general and thus more suitable for applications such as PSF engineering. To support the open-source community and facilitate the adoption of high-NA PSF models, we also provide a graphical interface in the napari ecosystem<sup>37</sup> and integrate our work in the chromatix optical simulation framework.<sup>38</sup>

# 2 | BACKGROUND

Electromagnetic waves in optical systems are fundamentally described by the Helmholtz wave equations, yet solving these equations in full generality is computationally intractable. Various approximations and integral formulations have thus been developed. The cornerstone for precise PSF models in high-NA systems is the Richards–Wolf integral,<sup>15</sup> which can be viewed as a vectorial extension of the Debye integral.<sup>39</sup> The conditions of validity for the Richards–Wolf integral have been thoroughly discussed by Wolf and Li.<sup>40</sup> While alternative light propagation models exist, the Huygens–Fresnel approach has been shown to be equivalent, to some extent, for PSF calculations.<sup>41</sup>

This has been the basis of a line of work for PSF models based on Bessel functions, which we will later refer to as the spherical parameterisation of the Richards–Wolf integral. This approach includes simpler formulations like the Kirchhoff model and more sophisticated vectorial representations.<sup>17,18</sup> These models have progressively been refined by incorporating various correction factors to account for additional physical processes: the Gibson–Lanni model for spherical aberrations due to refractive index mismatch<sup>19</sup> (later generalised in Refs. 36, 42), apodisation factors for energy conservation,<sup>15</sup> and Fresnel

<sup>1</sup> <https://svi.nl/>



**FIGURE 1** Geometry of the focusing of optical fields. An incident field  $\mathbf{e}_{\text{inc}}$  is transformed by a focusing element into a converging spherical wave  $\mathbf{e}_{\infty}$ . These fields are parameterised by a unit vector  $\mathbf{s}$ , either in Cartesian coordinates  $(s_x, s_y, s_z)$  or spherical coordinates  $(\theta, \phi)$ . The focus field  $\mathbf{E}$  is parameterised by  $\boldsymbol{\rho} = (x, y, z)$ , rewritten here to introduce cylindrical coordinates  $\rho$  and  $\phi$ .

transmission coefficients for accurate interface modelling.<sup>17</sup> Models using this spherical parameterisation have been implemented in various software libraries in Java<sup>20</sup> and Python,<sup>23</sup> which makes it widely accessible to researchers, albeit with some limitations in computational efficiency and integration with deep-learning frameworks.

Another line of work on PSF modelling is based on Fourier transforms, both in scalar<sup>43</sup> and vectorial<sup>16</sup> formulations. These models are based on a Cartesian parameterisation of the underlying Richards–Wolf integral and they represent a more general counterpart of the spherical parameterisation. Recently, these high-NA Fourier models have been implemented in MATLAB<sup>3</sup> and Tensorflow as part of a PSF fitting library.<sup>44</sup> Adequate sampling of the Fourier transform is crucial for obtaining high-resolution PSFs and avoiding aliasing. A common trick based on the chirp Z transform is usually implemented to achieve it.<sup>3,16,29</sup>

### 3 | THEORY

#### 3.1 | The Richards–Wolf model

As depicted in Figure 1, one obtains the PSF by computing the propagation of light after going through a focusing element, typically a microscope objective or lens. The incident field  $\mathbf{e}_{\text{inc}}(\mathbf{s})$ , also called the pupil function, is represented by a disk with a maximal cutoff angle defined by the NA of the imaging system. The focusing element transforms the planar incident field into a spherical wave,  $\mathbf{e}_{\infty}(\mathbf{s})$ , evaluated on the Gaussian reference sphere. This corresponds to an ensemble of far fields that propagate with direction  $\mathbf{s}$

and converge to the focal point  $O$ . Our goal is to compute the focused electric field  $\mathbf{E}(\boldsymbol{\rho})$  near  $O$ . Due to the reciprocity of light propagation, this model can also be extended to the emission of a point source to the back focal plane of a microscope objective.

The model introduced by Ref. (37) is the starting point that allows us to derive all the precise PSF models described in previous works. The focal field is given by a sum of plane waves with direction  $\mathbf{s} = (s_x, s_y, s_z)$ , as expressed by

$$\mathbf{E}(\boldsymbol{\rho}) = -\frac{ifk}{2\pi} \iint_{\Omega} \mathbf{e}_{\infty}(\mathbf{s}) \exp(ik\mathbf{s} \cdot \boldsymbol{\rho}) d\Omega, \quad (1)$$

where  $\boldsymbol{\rho} = (x, y, z) = (\rho \cos \phi, \rho \sin \phi, z)$  is the position vector in the focal region of the lens,  $\mathbf{s} = (s_x, s_y, s_z) = (\sin \theta \cos \phi, \sin \theta \sin \phi, \cos \theta)$  is a unit vector that describes the direction of an incoming ray,  $f$  is the focal length of the lens,  $k = \frac{2\pi n}{\lambda}$  is the wavenumber,  $\lambda$  is the wavelength,  $n$  is the refractive index of the propagation medium, and  $\mathbf{e}_{\infty}(\mathbf{s})$  describes the field distribution on the Gaussian reference sphere. We integrate over the set  $\Omega$  of solid angles defined on a region  $s_x^2 + s_y^2 \leq s_{\text{max}}^2$ , where  $s_{\text{max}} = \frac{\text{NA}}{n_i}$  is the cut-off determined by the NA and  $n_i$  is the refractive index of the immersion medium. The angle  $\theta$  is therefore defined within the immersion medium. We first describe the different classes of models based on this simplified concise equation and postpone to Section 3.4 the introduction of correction factors.

#### 3.2 | Scalar models

As a first step, it is common to rely on a scalar approximation to simplify calculations, especially in low-NA scenarios. In this case, the far field is equal to the incident field, so that  $\mathbf{e}_{\infty}(\mathbf{s}) = \mathbf{e}_{\text{inc}}(\mathbf{s})$ . The focal field is then given by

$$E(\boldsymbol{\rho}) = -\frac{ifk}{2\pi} \iint_{\Omega} e_{\infty}(s_x, s_y) \exp(iks \cdot \boldsymbol{\rho}) d\Omega. \quad (2)$$

This expression involves a two-dimensional integral over the pupil disk. Two specific parameterisations yield the two classes of models described previously.

The Cartesian parameterisation utilises both  $s_x$  and  $s_y$  coordinates with  $d\Omega = ds_x ds_y / s_z$ , which results in

$$E(\boldsymbol{\rho}) = -\frac{ifk}{2\pi} \iint_{s_x^2 + s_y^2 \leq s_{\text{max}}^2} \frac{e_{\infty}(s_x, s_y)}{s_z} \exp(iks_z z) \exp \times (ik(s_x x + s_y y)) ds_x ds_y. \quad (3)$$

In this form, the focused field at a  $z$  transverse plane is given by the 2D inverse Fourier transform of  $(e_\infty(s_x, s_y) \exp(iks_z z)/s_z)$ , where  $s_z = \sqrt{1 - s_x^2 - s_y^2}$ . Thus, the Cartesian parameterisation of the Richards–Wolf integral leverages the speed and efficiency of the fast Fourier transform (FFT) algorithm.

Alternatively, the spherical approach parameterises the problem with two angles  $\theta \in [0, \theta_{\max}]$  (the maximum angle  $\theta_{\max}$  is determined by the NA) and  $\phi \in [0, 2\pi]$ , as depicted in Figure 1. With  $\rho = (x, y, z) = (\rho \cos \phi, \rho \sin \phi, z)$  and  $d\Omega = \sin \theta d\theta d\phi$ , the field in the focal region can be rewritten as

$$E(\rho) = -\frac{ifk}{2\pi} \int_0^{\theta_{\max}} d\theta \int_0^{2\pi} d\phi e_\infty(\theta, \phi) \exp \times (ik\rho \sin \theta \cos(\phi - \varphi)) \exp(ikz \cos \theta) \sin \theta. \quad (4)$$

Equation (4) can be further simplified if one assumes that the pupil function is axisymmetric (rotational invariant), in the sense that  $e_\infty(\theta, \phi) = e_\infty(\theta)$ . In this case, the integral over  $\phi$  in (4) can be computed explicitly using the Bessel function  $J_0$ <sup>2</sup>, which leads to:

$$E(\rho) = -ifk \int_0^{\theta_{\max}} d\theta e_\infty(\theta) J_0(k\rho \sin \theta) \exp \times (ikz \cos \theta) \sin \theta. \quad (5)$$

Defocus is included in these models via the defocus phase factor  $\exp\{iks_z z\} = \exp(ikz \cos \theta)$  where  $z$  is the defocus distance. This expression, also known as angular spectrum propagation,<sup>43</sup> accurately models the propagation of an electric field in a homogeneous medium.

### 3.3 | Vectorial models

The electric field is a vectorial quantity. Consequently, vectorial propagation models are necessary to accurately account for the propagation and crosstalk between the components of the vector field. Precise vectorial models are crucial for high-NA systems, where the need to consider high angles arises.

In the vectorial model, the dependence on the incident field  $\mathbf{e}_{\text{inc}}(\mathbf{s})$  of the far field  $\mathbf{e}_\infty(\mathbf{s})$  now requires us to perform the basis change from a cylindrical to a spherical coordinate system, as in

$$\mathbf{e}_\infty(\theta, \phi) = \begin{bmatrix} q_s(1 - \cos 2\phi) + q_p(1 + \cos 2\phi) \cos \theta \\ (-q_s + q_p \cos \theta) \sin 2\phi \\ -2q_p \cos \phi \sin \theta \end{bmatrix} \frac{e_{\text{inc}}^x(\theta, \phi)}{2} + \begin{bmatrix} (-q_s + q_p \cos \theta) \sin 2\phi \\ q_s(1 + \cos 2\phi) + q_p(1 - \cos 2\phi) \cos \theta \\ -2q_p \sin \phi \sin \theta \end{bmatrix} \frac{e_{\text{inc}}^y(\theta, \phi)}{2}, \quad (6)$$

where  $\mathbf{e}_{\text{inc}} = [e_{\text{inc}}^x, e_{\text{inc}}^y, 0]$ . The so-called Fresnel transmission coefficients  $q_s$  and  $q_p$  have been introduced to account for the partial reflections at interfaces, which depend on the polarisation state and incidence angle. For each polarisation, the coefficients correspond to the product of all transmission coefficients for each interface from medium  $m$  to  $m + 1$ , with

$$q_s^m = \frac{2n_m \cos \theta_m}{n_m \cos \theta_m + n_{m+1} \cos \theta_{m+1}}, \quad q_p^m = \frac{2n_m \cos \theta_m}{n_{m+1} \cos \theta_m + n_m \cos \theta_{m+1}}. \quad (7)$$

It is worth noting that in the low-NA limit ( $\theta = 0$ ) with  $q_s = q_p = 1$ , (6) simplifies to  $\mathbf{e}_\infty(\theta, \phi) = \mathbf{e}_{\text{inc}}(\theta, \phi)$ . This explains why the scalar model did not account for this geometric change of variables.

The Cartesian parameterisation of the vectorial model consists of the integral

$$\mathbf{E}(\rho) = -\frac{ifk}{2\pi} \iint_{s_x^2 + s_y^2 \leq s_{\max}^2} \frac{\mathbf{e}_\infty(s_x, s_y)}{s_z} \exp(iks_z z) \exp \times (ik(s_x x + s_y y)) ds_x ds_y, \quad (8)$$

which essentially boils down to the computation of the inverse Fourier transform of  $(\mathbf{e}_\infty(s_x, s_y) \exp(iks_z z)/s_z)$ , similar to the scalar case.

With the help of coordinate transformations similar to the scalar case, we derive the spherical parameterisation of the field in the focal region as

$$\mathbf{E}(\rho) = -\frac{ifk}{2\pi} \int_0^{\theta_{\max}} d\theta \int_0^{2\pi} d\phi \mathbf{e}_\infty(\theta, \phi) \exp(ik\rho \sin \theta \cos(\phi - \varphi)) \exp(ikz \cos \theta) \sin \theta. \quad (9)$$

<sup>2</sup> The spherical parameterisation often uses the following identities, where  $J_n$  is the Bessel function of  $n$ th order of the first kind:

$$\int_0^{2\pi} \cos(n\phi) \exp(ix \cos(\phi - \varphi)) d\phi = 2\pi(i)^n J_n(x) \cos(n\varphi) \\ \int_0^{2\pi} \sin(n\phi) \exp(ix \cos(\phi - \varphi)) d\phi = 2\pi(i)^n J_n(x) \sin(n\varphi)$$

Inserting (6) into (9) and using the axisymmetric assumption of the incident field, we obtain a simplified expression of the focal field as

$$\mathbf{E}(\boldsymbol{\rho}) = -\frac{ifk}{2} \begin{bmatrix} [I_0^x - I_2^x \cos 2\varphi] - I_2^y \sin 2\varphi \\ -I_2^x \sin 2\varphi + [I_0^y + I_2^y \cos 2\varphi] \\ -2iI_1^x \cos \varphi - 2iI_1^y \sin \varphi \end{bmatrix}, \quad (10)$$

where

$$\begin{aligned} I_0^a(\rho, z) &= \int_0^{\theta_{\max}} e_{\text{inc}}^a(\theta) \sin \theta (\cos \theta + 1) J_0(k\rho \sin \theta) \\ &\quad \times \exp(ikz \cos \theta) d\theta \\ I_1^a(\rho, z) &= \int_0^{\theta_{\max}} e_{\text{inc}}^a(\theta) \sin^2 \theta J_1(k\rho \sin \theta) \exp \\ &\quad \times (ikz \cos \theta) d\theta \\ I_2^a(\rho, z) &= \int_0^{\theta_{\max}} e_{\text{inc}}^a(\theta) \sin \theta (\cos \theta - 1) J_2(k\rho \sin \theta) \\ &\quad \times \exp(ikz \cos \theta) d\theta \end{aligned} \quad (11)$$

with  $a \in \{x, y\}$ .

### 3.4 | Correction factors

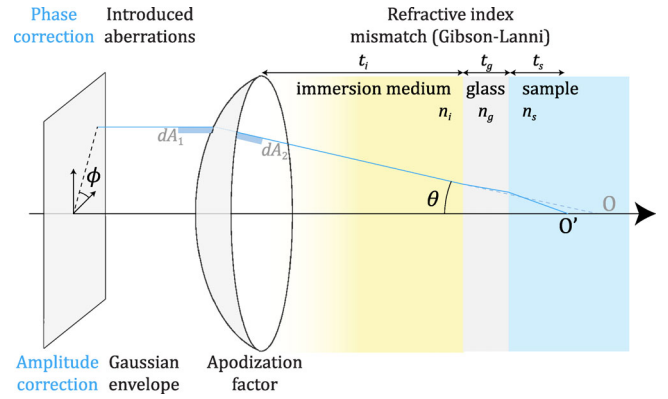
Precise PSF models commonly take into account several physical effects which may affect the true PSF. They are added as amplitude factors  $a(\mathbf{s})$  and phase factors  $W(\mathbf{s})$  in the original integral over solid angles in (1), leading to

$$\mathbf{E}(\boldsymbol{\rho}) = -\frac{ifk}{2\pi} \iint_{\Omega} a(\mathbf{s}) \exp(iW(\mathbf{s})) \mathbf{e}_{\infty}(\mathbf{s}) \exp(ik\mathbf{s} \cdot \boldsymbol{\rho}) d\Omega. \quad (12)$$

Equation (12) enables us to express these correction factors in full generality for both the vectorial and scalar models and for both the Cartesian and spherical parameterisations. We present a detailed list of these correction factors in Section 3.4.1–3.4.4 and their graphical description in Figure 2.

#### 3.4.1 | Gibson–Lanni aberrations

Microscopes typically have stratified layers of different refractive indices. The biological sample is usually aqueous, on top of which we place a coverslip made of glass, and the whole sample is then put in a water or oil immersion medium to increase the numerical aperture. The micro-



**FIGURE 2** Correction factors and their associated physical origins. Top part: phase correction factors, either introduced on the incident field or due to refraction. Bottom part: amplitude correction factors that model the incident beam envelope or the apodisation factor for energy conservation.

scope objectives are designed to provide aberration-free images in a specific setting with design values for refractive indices and thicknesses of each layer. Any mismatch introduces spherical aberrations due to refraction at the interface between the layers. These aberrations can be computed using the formula<sup>3</sup>

$$\begin{aligned} W(\mathbf{s}) &= \frac{2\pi}{\lambda} \left( t_s \sqrt{n_s^2 - n_i^2 \sin^2 \theta} + t_i \sqrt{n_i^2 - n_g^2 \sin^2 \theta} \right. \\ &\quad - t_i^* \sqrt{n_i^{*2} - n_g^2 \sin^2 \theta} + t_g \sqrt{n_g^2 - n_s^2 \sin^2 \theta} \\ &\quad \left. - t_g^* \sqrt{n_g^{*2} - n_s^2 \sin^2 \theta} \right), \end{aligned} \quad (13)$$

where  $n_s$ ,  $n_i$ ,  $n_g$  are the refractive indices of the sample, immersion medium, and glass, respectively,  $t_s$ ,  $t_i$ ,  $t_g$  are the thicknesses of the sample, immersion medium, and glass respectively, and their counterparts with stars are the design conditions.

In practice, it is challenging to assess the thickness of the immersion medium. Since this distance is manually tuned to obtain an optimal focus of a point emitter at depth  $t_s$  on the camera, this focusing condition gives the relation

$$t_i = t_s + n_i \left( -\frac{t_s}{n_s} - \frac{t_g}{n_g} + \frac{t_g^*}{n_g^*} + \frac{t_i^*}{n_i^*} \right), \quad (14)$$

which can be inserted in 13. This particular expression has first been derived for the spherical scalar case in Ref. (13) and extended to the spherical vectorial case in Refs. (51, 52).

<sup>3</sup> Correction factors will be expressed in spherical coordinates, with  $\sin \theta$  being computed in the Cartesian case as  $\sin \theta = \frac{NA}{n_i} \sqrt{s_x^2 + s_y^2}$ .



### 3.4.2 | Arbitrary phase aberrations

Refined aberration models can be introduced to describe imperfections in the optical system, for the benefit of PSF engineering or wavefront shaping. Designed aberrations can be introduced purposefully via a phase mask or a spatial light modulator at the pupil plane. The aberrations are often parameterised by Zernike polynomials (a set of orthonormal polynomials defined on the pupil disk) or by a direct fixed phase mask to obtain desired PSFs. We write it in the most general case as

$$W(\mathbf{s}) = \sum_{k=0}^{K-1} c_k Z_k(\mathbf{s}) + W_0(\mathbf{s}). \quad (15)$$

Equation (15) is composed of the sum of two terms. The first term is an inner product of the first  $K$  Zernike polynomials and their corresponding coefficients  $c_k$ . The second term can be used to include special cases not covered by the Zernike polynomials (e.g., a vortex phase ramp that leads to a donut PSF as is typically used in STED).

Since the arbitrary phase aberrations described in Equation (15) may not necessarily be axisymmetric, they can only be applied to the most general, Cartesian parameterisation.

### 3.4.3 | Apodisation factor

The apodisation factor is an amplitude-correction factor that ensures energy conservation during the change of basis from cylindrical coordinates (incident field  $\mathbf{e}_{\text{inc}}$ ) to spherical coordinates (far field  $\mathbf{e}_{\infty}$ ), which takes prominence especially for high-NA objectives. Since areas of cross-sections are modified, the field is also rescaled accordingly. Such rescaling ensures that the differential areas  $dA_1$  on the plane and  $dA_2$  on the sphere, as shown in Figure 2, remain consistent under the change of coordinates. The correcting factor is

$$A(\mathbf{s}) = \sqrt{\cos \theta} \quad (16)$$

when going from cylindrical to spherical in the focusing configuration of Figure 1.

### 3.4.4 | Gaussian envelope

The incident illumination can also depart from a perfect uniform plane wave. In particular, we often assume a Gaussian envelope and expressed it as

$$A(\mathbf{s}) = \exp\left(-\frac{\sin^2 \theta}{s_{\text{env}}^2}\right), \quad (17)$$

where the constant  $s_{\text{env}}$  determines the size of the envelope.

## 4 | IMPLEMENTATION

### 4.1 | PyTorch library

We provide a high-performance open-source PyTorch library `psf-generator`<sup>4</sup> to generate 2D and 3D PSFs. The library implements the four PSF models described in Section 3:

- `ScalarCartesianPropagator` (3);
- `ScalarSphericalPropagator` (5);
- `VectorialCartesianPropagator` (8);
- `VectorialSphericalPropagator` (10).

Users can choose between these propagator types, as well as configure physical (e.g., numerical aperture, wavelength, and field of view), and numerical parameters (e.g., image dimensions and number of z-planes). Our library also allows the users to freely apply any kind of correction factors tailored to their microscope on any chosen propagator as described in Section 3.4.2. The propagators use these parameters to first define the far field (pupil) and propagate it to obtain the focus field (PSF). Finally, the user can visualise, save, and export the generated PSF using our built-in utility functions. Written in PyTorch, the library easily integrates into deep-learning workflows, leveraging native features of PyTorch, such as automatic differentiation.

In our unifying framework, these models encompass all previously proposed PSF models based on the Richards–Wolf integral. As discussed in Section 3.2, the spherical propagators require the axisymmetric assumption. When it is valid, the Cartesian and spherical propagators perform equivalent integral computations and result in the same PSF. Both are proposed in our library as they differ in terms of computational efficiency and applicability.

Any tensor corresponding to a field has a shape of the form `(z, channel, x, y)` to benefit from PyTorch performance optimisation, both for data loading and computation parallelisation. Computation are performed plane-by-plane on the axial  $z$  axis, while  $x$  and  $y$  correspond to the transverse sizes of 2D images. Akin to greyscale versus RGB images, the `channel` dimension is equal to 1

<sup>4</sup> [https://github.com/Biomedical-Imaging-Group/psf\\_generator](https://github.com/Biomedical-Imaging-Group/psf_generator)

for scalar models and 3 in the vectorial case. We use the ZernikePy library<sup>5</sup> to generate Zernike polynomials.

The Cartesian parameterisation relies on multiple calls of 2D Fourier transforms with a desired field of view and pixel size. To account for the very small physical dimensions associated to pixels in localisation microscopy, we have implemented a custom variant of the 2D FFT in PyTorch.<sup>3,16,29</sup> It enables arbitrary pixel sizes and is based on the chirp Z-transform, at the cost of a convolution and three FFTs. The computational complexity of a single plane is still  $O(n \log n)$ , with  $n$  the size of the transverse plane. The size of the Fourier transform is doubled as we only keep the valid convolution range.

The spherical parameterisation relies on the fast and accurate evaluation of multiple one-dimensional integrals that involve the Bessel function of first order  $J_0$  over the interval  $[0, \theta_{\max}]$ , as defined in Equation (5) for the scalar case. To maximise speed, we vectorise the computation of the batch of 1D integrals at different defocus distances via *torch.vmap*. Then, for each 1D integral, we adopt the composite Simpson's rule<sup>45</sup> to benefit from its high accuracy, good numerical stability, and minimal computational overhead. The computational complexity in this case is  $O(n)$ , with  $n$  the number of integration steps in the interval  $[0, \theta_{\max}]$ . As automatic differentiation of Bessel functions is not natively supported by PyTorch as of version 2.3, we also provide a differentiable version of the Bessel functions in our library.

## 4.2 | Open-source interoperability

Motivated by FAIR principles for data and code management,<sup>46</sup> we not only open-source our implementations but make them interoperable with other tools. Our goal is to further support the dissemination and adoption of accurate physical models for high-NA light focusing. We present two complementary examples: an intuitive and rapid visualisation tool aimed at end users and an integration with a comprehensive optical simulation library intended for researchers.

napari is an open-source, Python-based image viewer designed for interactive visualisation and analysis of multidimensional biological and scientific image data. Built for researchers and developers working with large imaging data from optical microscopy and other imaging modalities, it provides a high-performance interface for exploring, annotating, and processing images. napari is also extensible via plugins, benefiting from the scientific Python ecosystem.

We provide an interface to generate and visualise 3D PSFs using our PyTorch-based framework, as depicted in Figure 3. Available as a plugin on the napari hub,<sup>6</sup> users can select between the four different propagation models and configure a set of physical and numerical parameters through a user-friendly graphical interface. This facilitates the introduction of high-NA correction factors to all four propagators types. PSFs are computed in real-time and rendered instantly, with support for both CPU and GPU backends. Our plugin allows immediate visual feedback and options to save the resulting volumes. It is designed to provide a code-free solution for the generation, visualisation, and export of high-NA PSFs for downstream applications.

Chromatix is an open-source wave optics library capable of simulating a wide range of optical systems such as light field imaging, holography and quantitative phase imaging.<sup>38</sup> Built on JAX, it supports GPU acceleration and full end-to-end differentiability, making it particularly suitable for large-scale inverse problems and integration with deep learning applications. Compared to PyTorch, JAX offers more seamless support for automatic differentiation and just-in-time (JIT) compilation, which can be advantageous for optimising physics-based simulations based on wave optics.

To ensure maximal performance and seamless integration with the rest of the library, the high-NA focusing models have been reimplemented in JAX. We focused on the vectorial Cartesian parameterisation, which offers greater generality than the spherical formulation and supports arbitrary pupil fields and aberrations. This functionality is introduced via the `high_na_ff_lens()` function, which complements the existing `ff_lens()` implementation used to model the Fourier transform of low-NA lenses. This addition enables Chromatix to accurately model high-NA microscope systems and facilitates integration into existing simulation pipelines.

These two examples highlight the importance of actively promoting interoperability within the open-source ecosystem. Nevertheless, all figures and benchmarks presented in this work were produced based on our comprehensive PyTorch-based library, which serves as the foundation of our study.

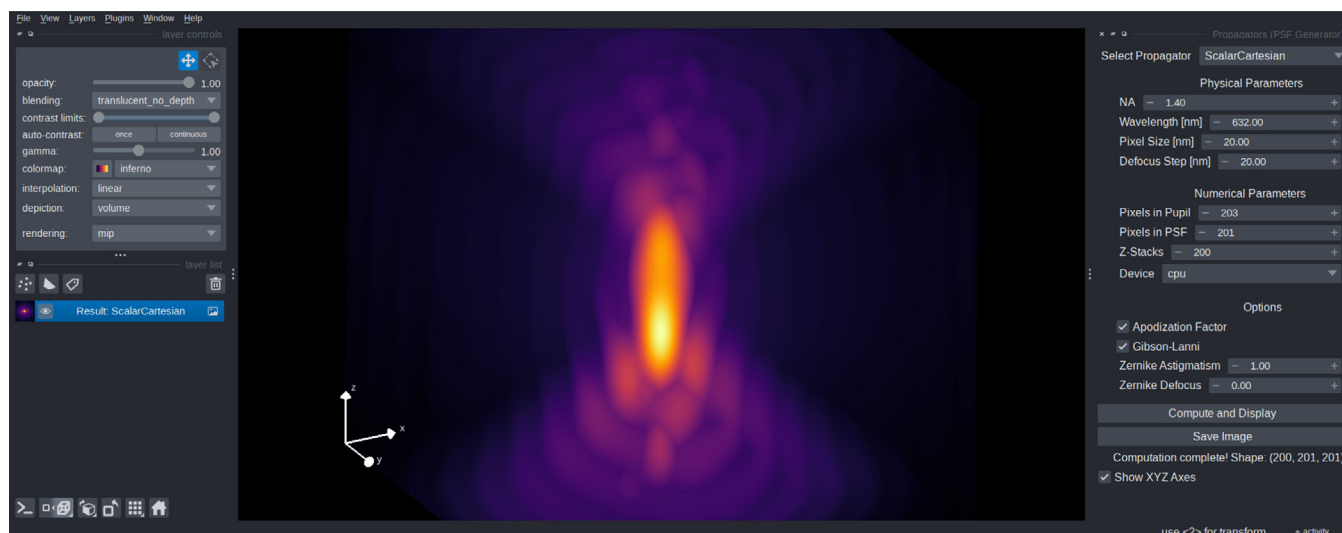
## 5 | RESULTS

### 5.1 | PSF gallery

We provide in Figures 4 and 5a gallery of PSFs to showcase our PyTorch implementation. We used a wavelength of 632

<sup>5</sup> <https://pypi.org/project/ZernikePy/>

<sup>6</sup> <https://www.napari-hub.org/plugins/napari-psfgenerator>



**FIGURE 3** Graphical user interface of our PSF simulation plugin integrated with napari (v0.4.0, 2025-06-17). Left: napari viewer panel enabling interactive 3D navigation and rendering options. Middle: interactive visualisation area, displaying here a slice of a high-NA 3D PSF. Right: plugin interface for selecting the propagator model, specifying physical and numerical parameters (e.g., NA, wavelength, Zernike aberrations, CPU or GPU backend), and exporting results.

nm, a circular polarisation for the vectorial models, and we display the amplitude (in arbitrary units) of the beams for better contrast. The 2D slices of the same 3D beam share the same dynamic range. In Figure 4, we present the focal electric field computed using both the scalar and vectorial models for objectives with different NAs. In the low-NA case (top two rows), the resulting PSFs from both propagators are similar. The beams differ more from each other, as expected, in the high-NA case (bottom two rows): the rings are blurred out as the energy is spread into different components of the focus field.

In Figure 5, we present additional PSFs computed with the vectorial propagator in the high-NA setting. The impact of the Gibson–Lanni correction factor on a beam is shown in Figure 5a–c. The refractive indices and thicknesses of the sample, immersion medium, and glass coverslip are  $n_s = 1.3$ ,  $n_i = 1.5$ ,  $n_g = 1.5$ ,  $t_s = 1 \mu\text{m}$ ,  $t_g = 170 \mu\text{m}$ , while  $t_i$  is computed using Equation (14). We observe that the spherical aberration it introduces degrades the quality of the focus. We also show the donut PSF, which has a vortex phase mask in the pupil plane (Figure 5d–f) and the half-moon PSF, which has a  $\pi$ -phase jump in its pupil plane (Figure 5g–i). Finally, we demonstrate an example with arbitrary phase aberrations using the Zernike polynomials in Figure 5j–l. Here, some amount of astigmatism is introduced, as is often done to encode defocus information,<sup>47</sup> and we show the evolution of the beam shape along the z-axis.

## 5.2 | Computational performance

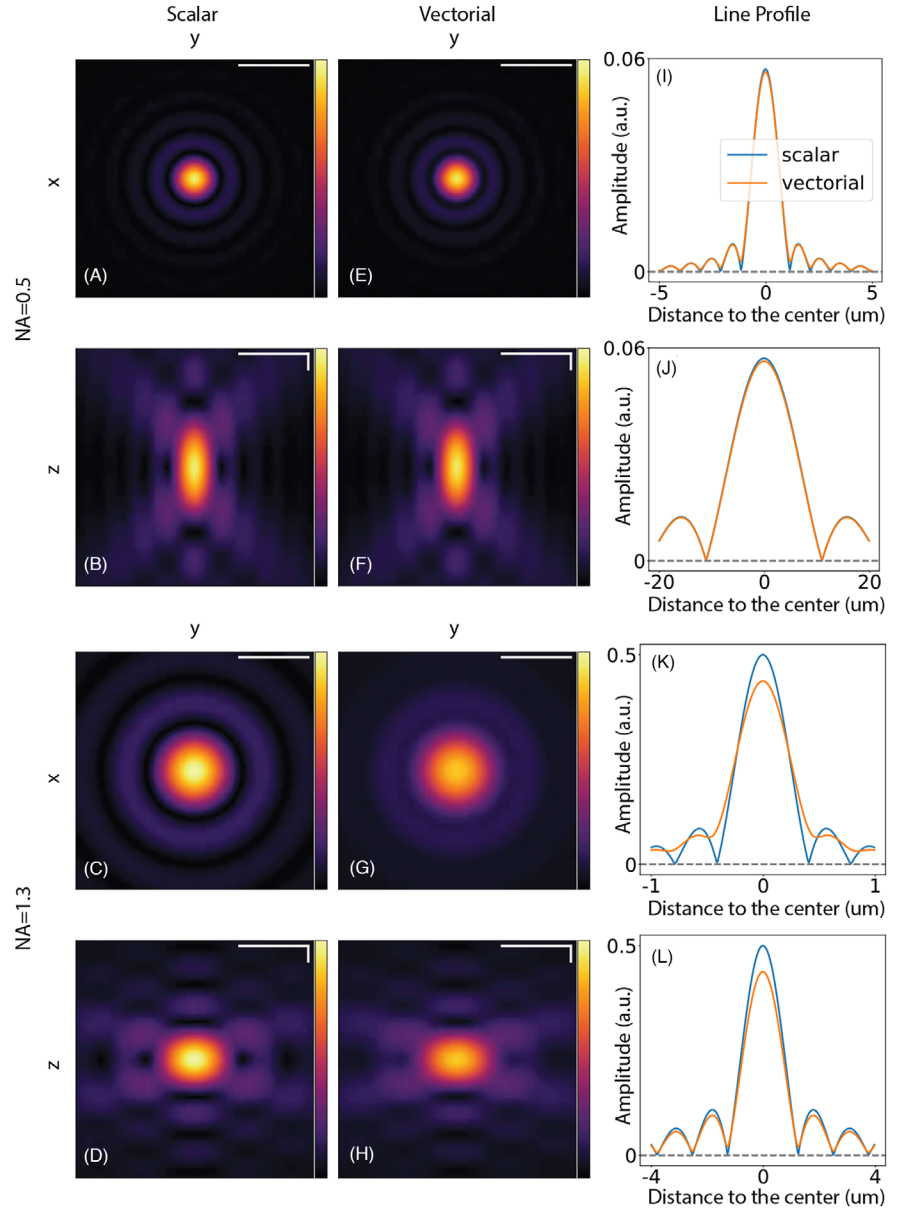
### 5.2.1 | Speed benchmark

We benchmark the runtime to compute a single  $201 \times 201$  2D PSF image against the size of the pupil. The image is captured at focus and all input parameters take default values in our library. We compare the runtime of all four propagators on CPU and GPU and show the results in Figure 6. The benchmarking was performed on a machine with an Intel i9-10900X CPU and an NVIDIA GeForce RTX 3090 GPU.

We observe that the runtime of all propagators increases with the size of the pupil on CPU. The Cartesian propagators (red) are faster than the spherical ones (blue) on small sizes ( $<512$  pixels) but slower on larger sizes. This illustrates the computational complexity of each method. Scalar propagators (dotted) are faster than their vectorial counterparts (solid), by roughly 1.5 times for the Cartesian and 3 times for the spherical cases. On GPU, Cartesian propagators are faster than on CPU at the same grid size and the curves (red) behave similarly. Spherical propagators, however, exhibit a flat curve (blue), which indicates that they benefit heavily from the GPU parallelisation. The speed improvement between scalar and vectorial propagators is small, especially for the Cartesian case. Hence, vectorial propagators should be preferred over the scalar ones as the accuracy gain does not come at a high



**FIGURE 4** Unaberrated PSFs generated by the scalar (a–d) and vectorial (e–h) models, in the case of low-NA (0.5) (first two rows) and high-NA (1.3) (last two rows). (a, e, c, g) Slice of the  $x$ - $y$  plane. (b, f, d, h) Slice of the  $z$ - $y$  plane. (i–l) Intensity profiles along a vertical line through the centre of the PSFs. The  $x$ -axis shows the relative distance to the centre of the image in micrometres. Scale bars represent  $3\ \mu\text{m}$  and  $0.6\ \mu\text{m}$  in the low- and high- NA cases, respectively. For images of  $z$ - $y$  planes (b, f, d, h), scale bars for the  $y$  and  $z$  axes are indicated by the horizontal and vertical bars, respectively.



computational cost. Moreover, Cartesian propagators are recommended if one works with images of small size for both scalar and vectorial cases. For larger sizes, however, spherical propagators should be preferred, especially when a GPU is available.

### 5.2.2 | Accuracy benchmark

We benchmark the accuracy of the Cartesian and spherical scalar propagators with the analytic Airy disk for asymptotic limit (the Fourier transform of a perfect circular aperture) given by

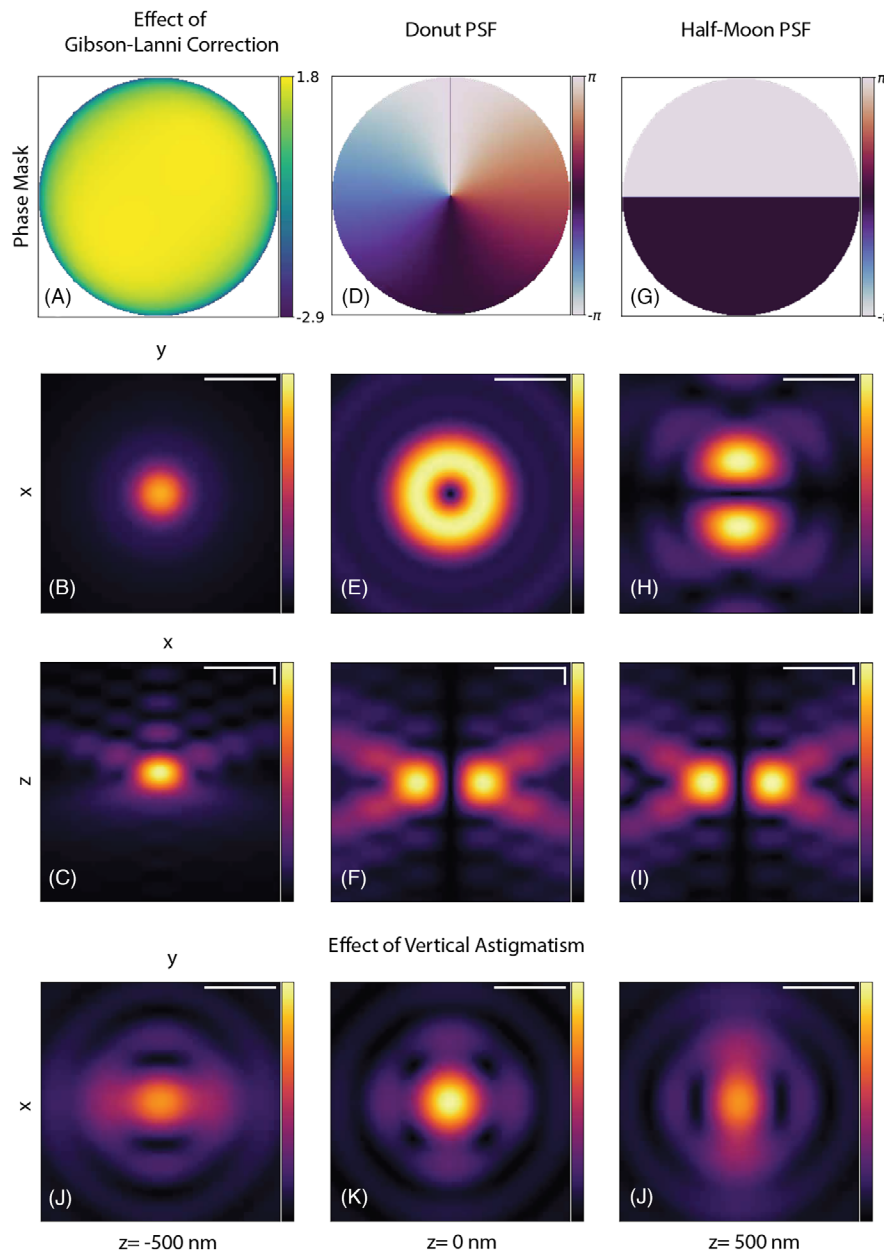
$$F_{\text{AD}}(\rho) = \frac{2J_1(\rho)}{\rho}, \quad (18)$$

where  $J_1$  is the Bessel function of the first order of the first kind. As the Airy disk is the Fourier transform of a perfect unit modulus circular aperture, an additional factor  $s_z = \cos \theta$  is introduced in Equations (3) and (5). This corresponds to a paraxial approximation.

The spherical integral is computed using two numerical integration rules: Riemann rule (a first-order method) and Simpson rule (a fourth-order method). The error  $\delta$  is the  $L_2$ -norm of the difference between the output electric field  $E$  of the propagator and  $F_{\text{AD}}$ , as in

$$\delta = \|E - F_{\text{AD}}\|_2. \quad (19)$$

We observe in Figure 7 that the error decreases as the number of points in the integration domain increases in all cases. The spherical propagator has a linear convergence



**FIGURE 5** PSFs with phase aberrations generated by the vectorial Cartesian propagator at high-NA (1.3). (a, d, g) Introduced phase masks at the pupil plane. (b, e, h) Slice of the  $x$ - $y$  plane at  $z = 0$ . (c, f, i) Slice of the  $z$ - $x$  plane at  $y = 0$ . (j-l) PSF with vertical astigmatism at three  $z$ -planes (in focus (k), 500 nm above (j), and 500 nm below focus (l)). All PSFs were generated with circular polarisation, except the Half-Moon PSF where  $x$ -axis linear polarisation is being used. The scale bars in all images represent 0.6  $\mu\text{m}$ .

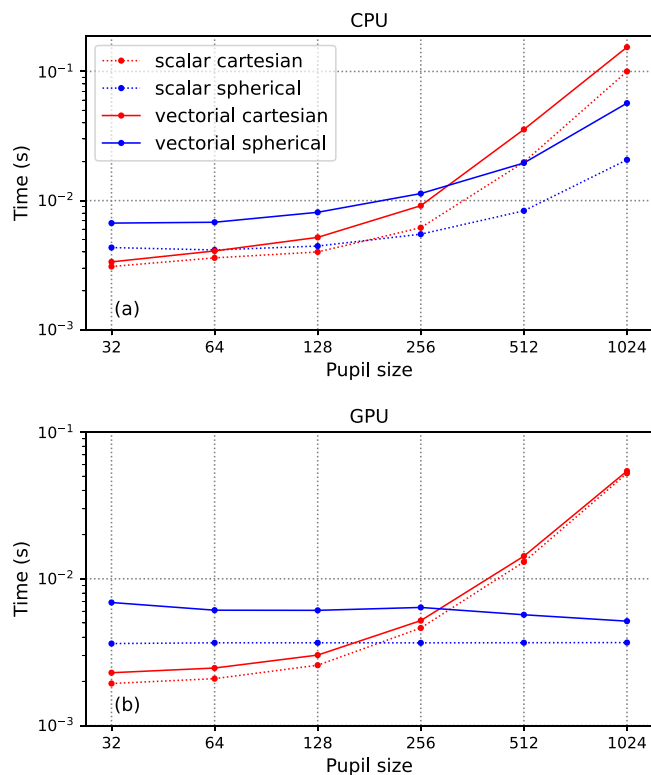
rate using Riemann rule (green) and fourth-order convergence rate using Simpson rule (blue), which correlates well with their expected accuracy. The Cartesian propagator, based on our custom FFT, shows a convergence rate between first and second order.

## 6 | CONCLUSION

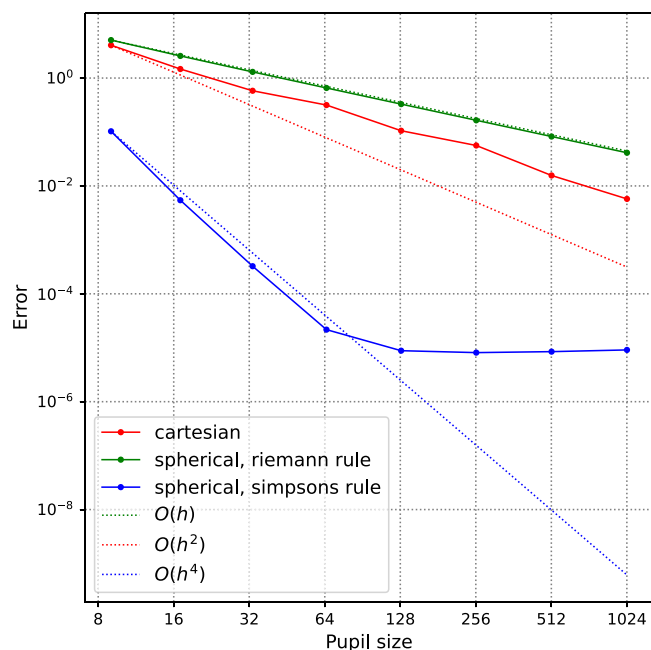
In this work, we have introduced a unifying theory for accurate PSF models, revisiting previous approaches and

generalising correction factors. This framework demonstrates the equivalence of Cartesian and spherical methods as different parameterisations of the same propagation integral, providing a simplified understanding of light propagation in high-NA systems.

We have also developed a high-performance implementation in PyTorch, which is open-source and supported by comprehensive documentation. This library allows for efficient CPU/GPU computation of PSFs, with functionalities such as automatic differentiation that make it particularly suitable for optimisation pipelines.



**FIGURE 6** Time to generate a  $201 \times 201$  PSF in terms of the numerical size of the pupil and the propagator, on CPU (a) and GPU (b). Each data point in the plots is averaged over 10 runs.



**FIGURE 7** Accuracy of the Cartesian (red) and the spherical propagators using Riemann rule (green) and Simpson rule (blue). The step size of integration is  $h$ .

In practice, the choice of propagators depends on the symmetry of the pupil function. Based on our benchmark, for axisymmetric pupil functions, spherical propagators are recommended thanks to their high accuracy and scalability. They are also particularly amenable to GPU parallelisation. For non-axisymmetric cases, such as those involving Zernike aberrations or specialised phase masks, Cartesian propagators are required. In all these cases, the difference in computational time is relatively modest for pupil sizes up to a few hundred pixels. Hence, the vectorial Cartesian propagator could be a solid default choice in most applications.

We hope for our PyTorch library to contribute to the rapidly growing field of applying deep neural networks on physical imaging models.<sup>48</sup> Learning-based methods have demonstrated their effectiveness in various applications, such as deconvolution<sup>49–51</sup> and 3D SMLM,<sup>52</sup> with state-of-the-art deep-learning tools such as DeepSTORM<sup>53</sup> and DECODE.<sup>54</sup> For instance, some use cases of our framework include generating large reference datasets to train networks or adapting images based on physical rules to enable learning in self-supervised inversions<sup>55</sup> and generative adversarial networks.<sup>56</sup>

## ACKNOWLEDGEMENTS

Vasiliki Stergiopoulou acknowledges funding from the Swiss National Science Foundation under Grant CRSII5\_213521, 'DigiLight – Programmable Third-Harmonic Generation (THG) Microscopy Applied to Advanced Manufacturing'. Jonathan Dong acknowledges funding from the Swiss National Science Foundation (Grant PZ00P2\_216211). We would like to thank Eric Sinner for his valuable help in software engineering.

Open access publishing facilitated by Ecole polytechnique federale de Lausanne, as part of the Wiley - Ecole polytechnique federale de Lausanne agreement via the Consortium Of Swiss Academic Libraries.

## REFERENCES

- Matthews, C., & Cordelieres, F. P. (2010). MetroloJ: An ImageJ plugin to help monitor microscopes' health. In *ImageJ User & Developer Conference Proceedings* (pp. 1–6). <https://imagej.net/events/conferences>
- Theer, P., Mongis, C., & Knop, M. (2014). PSFj: Know your fluorescence microscope. *Nature Methods*, 11(10), 981–982.
- Miora, R. H. D., Rohwer, E., Kielhorn, M., Sheppard, C., Bosman, G., & Heintzmann, R. (2024). Calculating point spread functions: Methods, pitfalls, and solutions. *Optics Express*, 32(16), 27278–27302.
- Nelson, G., Alexopoulos, I., Azevedo, M., Barachati, F., Belavaev, Y., Carvalho, M. T., Cesbron, Y., Dauphin, A., Corbett, A. D., & Dobbie, I. M. (2022). Monitoring the point spread function for quality control of confocal microscopes. <https://doi.org/10.17504/protocols.io.bp2l61vw1vqe/v1>

5. Lelek, M., Gyparakis, M. T., Beliu, G., Schueder, F., Griffié, J., Manley, S., Jungmann, R., Sauer, M., Lakadamyali, M., & Zimmer, C. (2021). Single-molecule localization microscopy. *Nature reviews Methods Primers*, 1(1), 39.
6. Sibarita, J.-B. (2005). Deconvolution microscopy. *Microscopy Techniques*, 95, 201–243.
7. Sage, D., Donati, L., Soulez, F., Fortun, D., Schmit, G., Seitz, A., Guiet, R., Vonesch, C., & Unser, M. (2017). Deconvolution-Lab2: An open-source software for deconvolution microscopy. *Methods*, 115, 28–41.
8. Heintzmann, R., & Cremer, C. G. (1999). Laterally modulated excitation microscopy: Improvement of resolution by using a diffraction grating. In *Optical Biopsies and Microscopic Techniques III* (Vol. 3568, pp. 185–196). SPIE.
9. Gustafsson, M. G. (2000). Surpassing the lateral resolution limit by a factor of two using structured illumination microscopy. *Journal of Microscopy*, 198(2), 82–87.
10. Dertinger, T., Colyer, R., Iyer, G., Weiss, S., & Enderlein, J. (2009). Fast, background-free, 3D super-resolution optical fluctuation imaging (SOFI). *Proceedings of the National Academy of Sciences*, 106(52), 22287–22292.
11. Stergiopoulou, V., Calatroni, L., de Morais Goulart, H., Schaub, S., & Blanc-Féraud, L. (2022). COLORME: Super-resolution microscopy based on sparse blinking/fluctuating fluorophore localization and intensity estimation. *Biological Imaging*, 2.
12. Hell, S. W. (2007). Far-field optical nanoscopy. *Science*, 316(5828), 1153–1158.
13. Balzarotti, F., Eilers, Y., Gwosch, K. C., Gynnå, A. H., Westphal, V., Stefani, F. D., Elf, J., & Hell, S. W. (2017). Nanometer resolution imaging and tracking of fluorescent molecules with minimal photon fluxes. *Science*, 355(6325), 606–612.
14. Marin, Z., Graff, M., Barentine, A. E., Soeller, C., Chung, K. K. H., Fuentes, L. A., & Baddeley, D. (2021). PYMEVisualize: An open-source tool for exploring 3D super-resolution data. *Nature Methods*, 18(6), 582–584.
15. Richards, B., & Wolf, E. (1959). Electromagnetic diffraction in optical systems. II. Structure of the image field in an aplanatic system. *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, 253(1274), 358–379.
16. Leutenegger, M., Rao, R., Leitgeb, R. A., & Lasser, T. (2006). Fast focus field calculations. *Optics Express*, 14(23), 11277–11291.
17. Aguet, F. (2009). Super-resolution fluorescence microscopy based on physical models. PhD thesis, EPFL.
18. Novotny, L., & Hecht, B. (2012). *Principles of nano-optics* (2nd ed.). Cambridge University Press, Cambridge.
19. Gibson, S. F., & Lanni, F. (1991). Experimental test of an analytical model of aberration in an oil-immersion objective lens used in three-dimensional light microscopy. *Journal of the Optical Society of America A*, 8(10), 1601–1613.
20. Kirshner, H., François, A., Sage, D., & Unser, M. (2013). 3-D PSF fitting for fluorescence microscopy: Implementation and localization application. *Journal of Microscopy*, 249(1), 13–25.
21. Nasse, M. J., & Woehl, J. C. (2010). Realistic modeling of the illumination point spread function in confocal scanning optical microscopy. *Journal of the Optical Society of America A*, 27(2), 295–302.
22. Schneider, M. C., Hinterer, F., Jesacher, A., & Schütz, G. J. (2024). Interactive simulation and visualization of point spread functions in single molecule imaging. *Optics Communications*, 560, 130463.
23. Caprile, F., Masullo, L. A., & Stefani, F. D. (2022). Pyfocus – A Python package for vectorial calculations of focused optical fields under realistic conditions. Application to toroidal foci. *Computer Physics Communications*, 275, 108315.
24. Prigent, S., Nguyen, H.-N., Leconte, L., Valades-Cruz, C. A., Hajj, B., Salamero, J., & Kervrann, C. (2023). SPITFIR (e): A supermaneuverable algorithm for fast denoising and deconvolution of 3D fluorescence microscopy images and videos. *Scientific Reports*, 13(1), 1489.
25. Li, Y., Mund, M., Hoess, P., Deschamps, J., Matti, U., Nijmeijer, B., Sabinina, V. J., Ellenberg, J., Schoen, I., & Ries, J. (2018). Real-time 3D single-molecule localization using experimental point spread functions. *Nature Methods*, 15(5), 367–369.
26. Dong, J., Maestre, D., Conrad-Billroth, C., & Juffmann, T. (2021). Fundamental bounds on the precision of iSCAT, COBRI and dark-field microscopy for 3D localization and mass photometry. *Journal of Physics D: Applied Physics*, 54(39), 394002.
27. Shechtman, Y., Sahl, S. J., Backer, A. S., & Moerner, W. E. (2014). Optimal point spread function design for 3D imaging. *Physical Review Letters*, 113(13), 133902.
28. Opatovski, N., Nehme, E., Zoref, N., Barzilai, I., Orange Kedem, R., Ferdman, B., Keselman, P., Alalouf, O., & Shechtman, Y. (2024). Depth-enhanced high-throughput microscopy by compact PSF engineering. *Nature Communications*, 15(1), 4861.
29. Liu, Y., Dong, J., Maya, J. A., Balzarotti, F., & Unser, M. (2025). Point-spread-function engineering in MINFLUX: Ptimality of donut and half-moon excitation patterns. *Optics Letters*, 50(1), 37–40.
30. Sinkó, J., Kákonyi, R., Rees, E., Metcalf, D., Knight, A. E., Kaminski, C. F., Szabó, G., & Erdélyi, M. (2014). TestSTORM: Simulator for optimizing sample labeling and image acquisition in localization based super-resolution microscopy. *Biomedical Optics Express*, 5(3), 778–787.
31. Sage, D., Kirshner, H., Pengo, T., Stuurman, N., Min, J., Manley, S., & Unser, M. (2015). Quantitative evaluation of software packages for single-molecule localization microscopy. *Nature Methods*, 12(8), 717–724.
32. Wieser, S., & Schütz, G. J. (2008). Tracking single molecules in the live cell plasma membrane—Do's and Don'ts. *Methods*, 46(2), 131–140.
33. Shen, H., Tauzin, L. J., Baiyasi, R., Wang, W., Moringo, N., Shuang, B., & Landes, C. F. (2017). Single particle tracking: From theory to biophysical applications. *Chemical Reviews*, 117(11), 7331–7376.
34. Griffié, J., Pham, T., Sieben, C., Lang, R., Cevher, V., Holden, S., Unser, M., Manley, S., & Sage, D. (2020). Virtual-SMLM, a virtual environment for real-time interactive SMLM acquisition. *bioRxiv*.
35. Bourgeois, D. (2023). Single molecule imaging simulations with advanced fluorophore photophysics. *Communications Biology*, 6(1), 53.
36. Török, P., & Varga, P. (1997). Electromagnetic diffraction of light focused through a stratified medium. *Applied Optics*, 36(11), 2305–2312.
37. Sofroniew, N., Lambert, T., Evans, K., Nunez-Iglesias, J., Bokota, G., Winston, P., Peña-Castellanos, G., Yamauchi, K., Bussonnier, M., Doncila Pop, D., Can Solak, A., Liu, Z., Wadhwa, P., Burt, A.,



- Buckley, G., Sweet, A., Migas, L., Hilsenstein, V., Gaifas, L., ... Royer, L. (2019). napari: A multi-dimensional image viewer for python. Zenodo. <https://doi.org/10.5281/zenodo.3555620>
38. Deb, D., Both, G.-J., Bezzam, E., Kohli, A., Yang, S., Chaware, A., Allier, C., Cai, C., Anderberg, G., Eybposh, M. H., Schneider, M. C., Heintzmann, R., Rivera-Sanchez, F. A., Simmerer, C., Meng, G., Tormes-Vaquero, J., Han, S., Shanmugavel, S. C., Maruvada, T., & Turaga, S. C. (2025). Chromatix: A differentiable, GPU-accelerated wave-optics library. *bioRxiv*, pages 2025–04.
  39. Debye, P. J. W. (1908). Der lichtdruck auf kugeln von beliebigem material. PhD thesis, Ludwig-Maximilians Universität München.
  40. Wolf, E., & Li, Y. (1981). Conditions for the validity of the Debye integral representation of focused fields. *Optics Communications*, 39(4), 205–210.
  41. Egner, A., & Hell, S. (1999). Equivalence of the Huygens–Fresnel and Debye approach for the calculation of high aperture point-spread functions in the presence of refractive index mismatch. *Journal of Microscopy*, 193(3), 244–249.
  42. Török, P., Varga, P., Laczik, Z., & Booker, G. (1995). Electromagnetic diffraction of light focused through a planar interface between materials of mismatched refractive indices: An integral representation. *Journal of the Optical Society of America A*, 12(2), 325–332.
  43. Goodman, J. (2005). *Introduction to Fourier optics*. McGraw-Hill physical and quantum electronics series. W. H. Freeman.
  44. Liu, S., Chen, J., Hellgoth, J., Müller, L.-R., Ferdman, B., Karras, C., Xiao, D., Lidke, K. A., Heintzmann, R., Shechtman, Y., Li, Y., & Ries, J. (2024). Universal inverse modeling of point spread functions for SMLM localization and microscope characterization. *Nature Methods*, 21(6), 1082–1093.
  45. Stoer, J., & Bulirsch, R. (2002). *Introduction to numerical analysis*. Texts in applied mathematics. Vol. 12. New York, NY: Springer New York.
  46. Barker, M., Chue Hong, N. P., Katz, D. S., Lamprecht, A.-L., Martinez-Ortiz, C., Psomopoulos, F., Harrow, J., Castro, L. J., Gruenpeter, M., Martinez, P. A., & Honeyman, T. (2022). Introducing the fair principles for research software. *Scientific Data*, 9(1), 622.
  47. Kao, H. P., & Verkman, A. (1994). Tracking of single fluorescent particles in three dimensions: Use of cylindrical optics to encode particle position. *Biophysical Journal*, 67(3), 1291–1300.
  48. Belthangady, C., & Royer, L. A. (2019). Applications, promises, and pitfalls of deep learning for fluorescence image reconstruction. *Nature Methods*, 16(12), 1215–1225.
  49. Weigert, M., Schmidt, U., Boothe, T., Müller, A., Dibrov, A., Jain, A., Wilhelm, B., Schmidt, D., Broaddus, C., Culley, S., Rocha-Martins, M., Segovia-Miranda, F., Norden, C., Henriques, R., Zerial, M., Solimena, M., Rink, J., Tomancak, P., Royer, L., & Myers, E. W. (2018). Content-aware image restoration: Pushing the limits of fluorescence microscopy. *Nature Methods*, 15(12), 1090–1097.
  50. Li, Y., Su, Y., Guo, M., Han, X., Liu, J., Vishwasrao, H. D., Li, X., Christensen, R., Sengupta, T., Moyle, M. W., Rey-Suarez, I., Chen, J., Upadhyaya, A., Usdin, T. B., Colón-Ramos, D. A., Liu, H., Wu, Y., & Shroff, H. (2022). Incorporating the image formation process into deep learning improves network performance. *Nature Methods*, 19(11), 1427–1437.
  51. Yanny, K., Monakhova, K., Shuai, R. W., & Waller, L. (2022). Deep learning for fast spatially varying deconvolution. *Optica*, 9(1), 96–99.
  52. Sage, D., Pham, T.-A., Babcock, H., Lukes, T., Pengo, T., Chao, J., Velmurugan, R., Herbert, A., Agrawal, A., Colabrese, S., Wheeler, A., Archetti, A., Rieger, B., Ober, R., Hagen, G. M., Sibarita, J.-B., Ries, J., Henriques, R., Unser, M., & Holden, S. (2019). Super-resolution fight club: Assessment of 2D and 3D single-molecule localization microscopy software. *Nature Methods*, 16(5), 387–395.
  53. Nehme, E., Freedman, D., Gordon, R., Ferdman, B., Weiss, L. E., Alalouf, O., Naor, T., Orange, R., Michaeli, T., & Shechtman, Y. (2020). DeepSTORM3D: Dense 3D localization microscopy and PSF design by deep learning. *Nature Methods*, 17(7), 734–740.
  54. Speiser, A., Müller, L.-R., Hoess, P., Matti, U., Obara, C. J., Legant, W. R., Kreshuk, A., Macke, J. H., Ries, J., & Turaga, S. C. (2021). Deep learning enables fast and dense single-molecule localization with high accuracy. *Nature Methods*, 18(9), 1082–1090.
  55. Kobayashi, H., Solak, A. C., Batson, J., & Royer, L. A. (2020). Image deconvolution via noise-tolerant self-supervised inversion. *arXiv preprint arXiv:2006.06156*.
  56. Cachia, M., Stergiopoulou, V., Calatroni, L., Schaub, S., & Blanc-Féraud, L. (2023). Fluorescence image deconvolution microscopy via generative adversarial learning (FluoGAN). *Inverse Problems*, 39(5), 054006.

**How to cite this article:** Liu, Y., Stergiopoulou, V., Chuah, J., Bezzam, E., Both, G.-J., Unser, M., Sage, D., & Dong, J. (2025). Revisiting PSF models: Unifying framework and high-performance implementation. *Journal of Microscopy*, 1–13. <https://doi.org/10.1111/jmi.70045>