Computational Photonics

Solving inverse problems

Mostly based on material from

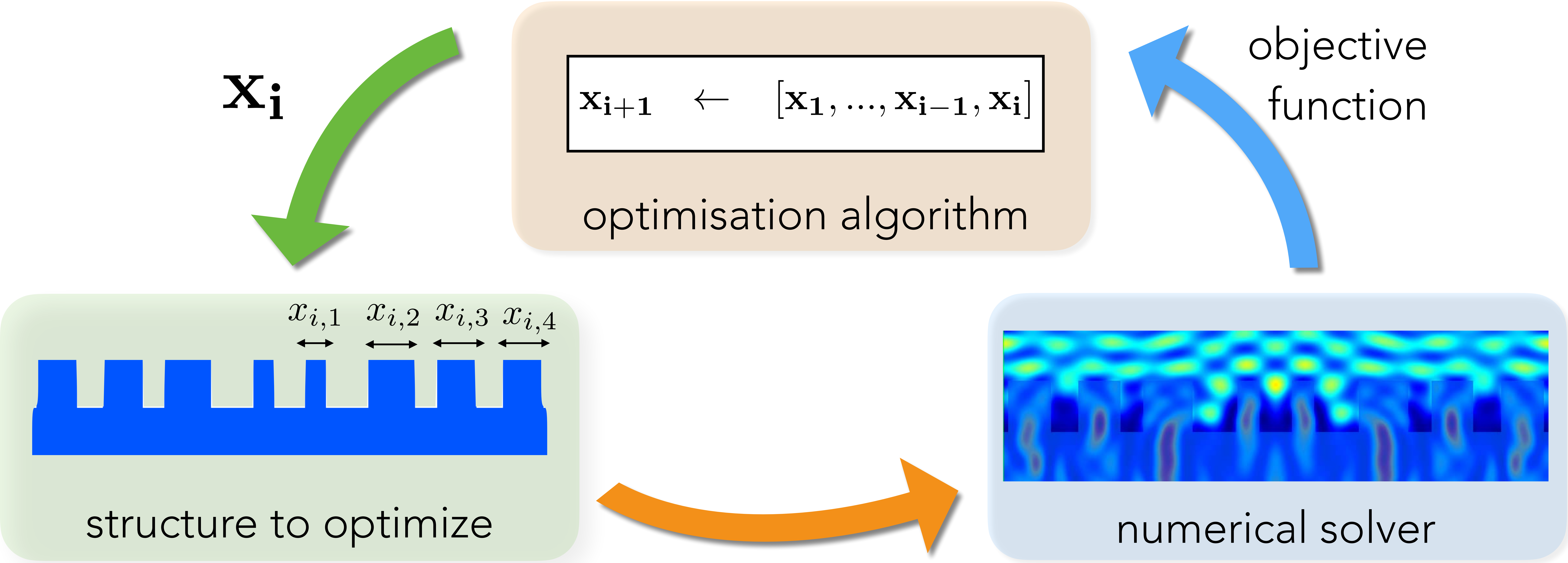
Yannick Augenstein

Topology optimisation

Xavi Garcia-Santiago

Bayesian optimisation

Solving inverse problems



constraints:

- limited time available to find an optimal design
- adapted to fabrication methodologies
- ideally automatic without human intervention

Classification of approaches

number of degrees of freedoms in your system

low dimensional	medium dimensional	high dimensional
(1-3 parameters)	(4-15 parameters)	(15 up to 10^X parameters)
• reach global optimum • scan entire parameter space • use preferential forward solver	• some hope for global optimum • sample parameter space in an effective way	• only go for local optimum • exploit adjoint/backpropagation methods to adjust many DoF
curse of dimensionality	***local / stochastic / model-based***	***topology / neural networks***

overarching concerns:

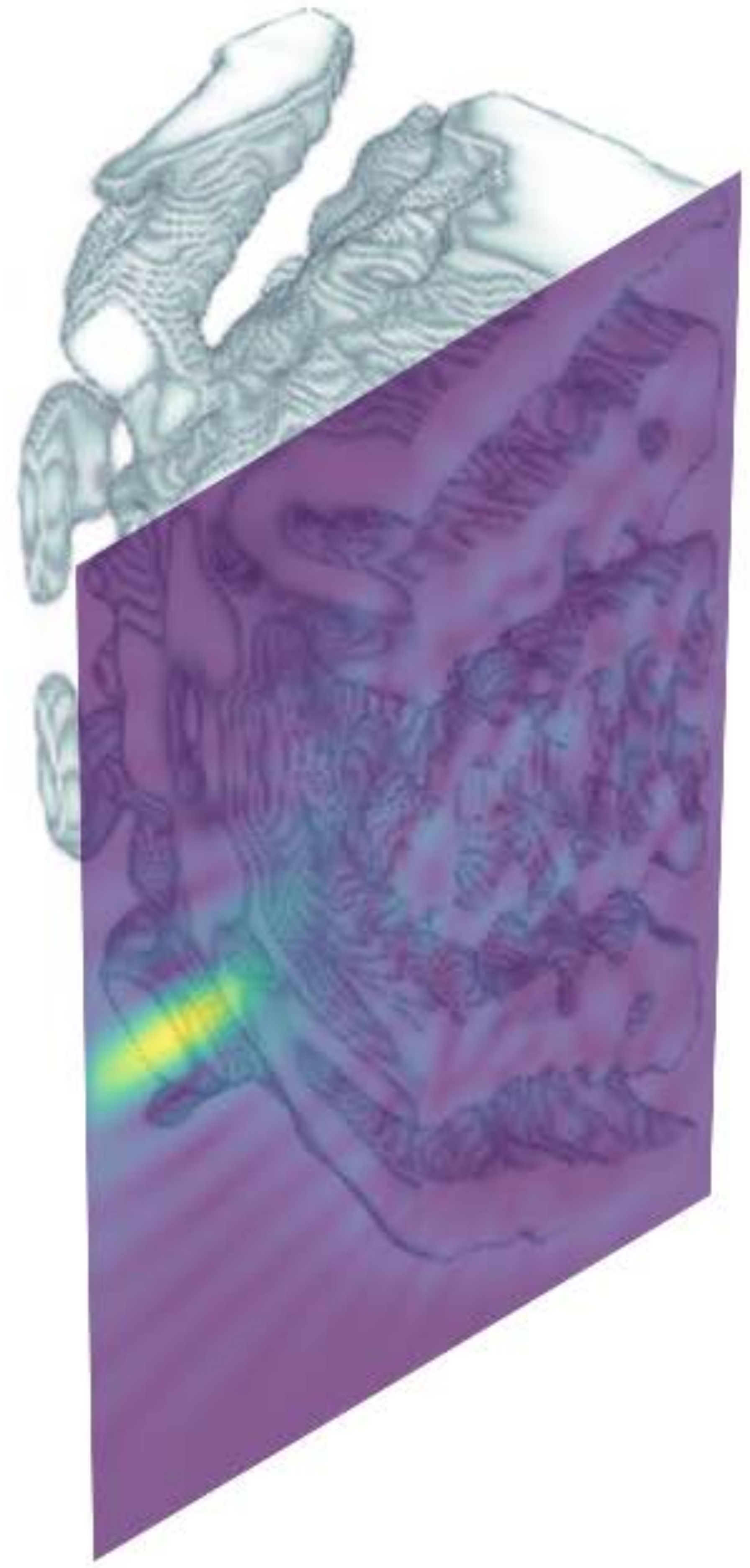
- exploit derivative informations efficiently
- consider time for an individual evaluation
- accurate data is a precious value

Basic idea topology optimisation

Basic idea topology optimisation

Basic idea topology optimisation

Basic idea topology optimisation



Karlsruhe Institute of Technology

Aspects that we discuss in the following

1) **Adjoint formulation**

How to determine with two full wave simulations the gradients of an objective function with respect to all degrees of freedom?

2) **Parametrisation**

How to parametrise the spatial distribution of the permittivity that will be optimised in the next step?

3) **Imposing constraints**

How to express the fact that the permittivity should be a binary function, it should respect minimal feature size, a predefined volume or any other constraint?

4) **Running an optimisation**

How to perform an actual gradient descent?

Basics of adjoint formalism

- generic forward problem

  (PDE discretized in coupled linear equations)

$$\boldsymbol{Ax} = \boldsymbol{b}$$

$\boldsymbol{A} \in \mathbb{R}^{n \times n}$ (system matrix)

$\boldsymbol{x} \in \mathbb{R}^{n}$ (solution vector)

$\boldsymbol{b} \in \mathbb{R}^{n}$ (source vector)

- Maxwell's equations:

$$\left[\nabla \times \nabla \times - \omega^2 \mu_0 \epsilon_0 \epsilon(\mathbf{r}, \omega)\right] \tilde{\mathbf{E}}(\mathbf{r}, \omega) = i\omega\mu_0 \tilde{\mathbf{j}}(\mathbf{r}, \omega)$$

  (material properties appear in an independent term)

- all quantities depend on the design parameters $\boldsymbol{g} \in \mathbb{R}^{m}$

- solution of the forward problem

  [direct (e.g., LU or Cholesky decomposition) or iterative (e.g., Richardson or Jacobi methods)]

$$\boldsymbol{x}(\boldsymbol{g}) = \boldsymbol{A}^{-1}(\boldsymbol{g})\, \boldsymbol{b}(\boldsymbol{g})$$

expensive

Basics of adjoint formalism

- design goodness quantified by objective function $F(\boldsymbol{x})$

- optimisation problem: $\min_{\boldsymbol{g}} F(\boldsymbol{x}(\boldsymbol{g}))$ subject to $\boldsymbol{A}(\boldsymbol{g})\,\boldsymbol{x}(\boldsymbol{g}) = \boldsymbol{b}(\boldsymbol{g})$

- Maxwell's equations: $\min_{\epsilon(\mathbf{r},\omega)} F_{\mathrm{EM}}\left(\tilde{\mathbf{E}}\left(\mathbf{r},\omega\right)\right)$ subject to $\mathbf{M}(\epsilon)\tilde{\mathbf{E}}(\epsilon) = \tilde{\mathbf{s}}(\epsilon)$

material in each pixel is a d.o.f. with respect to figure of merit $F_{\mathrm{EM}}\left(\tilde{\mathbf{E}}(\mathbf{r},\omega)\right)$

- gradient based optimisation requires us to know

$$\frac{\mathrm{d}F}{\mathrm{d}\boldsymbol{g}} = \frac{\mathrm{d}F}{\mathrm{d}\boldsymbol{x}}\frac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}\boldsymbol{g}}$$

1. first term easy: objective function is a scalar function of $\boldsymbol{x}$
2. second term hard: Jacobian is a dense matrix of size $n \times m$

Basics of adjoint formalism

- derivative with respect to **a single** design variable:

$$\frac{\partial \boldsymbol{x}}{\partial g_i} = \frac{\partial \boldsymbol{A}^{-1}}{\partial g_i}\boldsymbol{b} + \boldsymbol{A}^{-1}\frac{\partial \boldsymbol{b}}{\partial g_i} = -\boldsymbol{A}^{-1}\frac{\partial \boldsymbol{A}}{\partial g_i}\boldsymbol{A}^{-1}\boldsymbol{b} + \boldsymbol{A}^{-1}\frac{\partial \boldsymbol{b}}{\partial g_i} = \boldsymbol{A}^{-1}\left(\frac{\partial \boldsymbol{b}}{\partial g_i} - \frac{\partial \boldsymbol{A}}{\partial g_i}\boldsymbol{x}\right)$$

- derivative with respect to **all** design variables:

$$\frac{\mathrm{d}F}{\mathrm{d}\boldsymbol{g}} = \frac{\mathrm{d}F}{\mathrm{d}\boldsymbol{x}}\boldsymbol{A}^{-1}\left(\left[\frac{\partial \boldsymbol{b}}{\partial g_1}, \frac{\partial \boldsymbol{b}}{\partial g_2}, \cdots, \frac{\partial \boldsymbol{b}}{\partial g_m}\right] - \left[\frac{\partial \boldsymbol{A}}{\partial g_1}\boldsymbol{x}, \frac{\partial \boldsymbol{A}}{\partial g_2}\boldsymbol{x}, \cdots, \frac{\partial \boldsymbol{A}}{\partial g_m}\boldsymbol{x}\right]\right)$$

extremely expensive calculation and not necessary

not interested in $\frac{\mathrm{dx}}{\mathrm{dg}}$ **but only in** $\frac{\mathrm{d}F}{\mathrm{dg}}$

KEY DETAIL

Basics of adjoint formalism

- starting from the left of the previous expression:

$$\frac{\mathrm{d}F}{\mathrm{d}\boldsymbol{x}}\boldsymbol{A}^{-1} = \underbrace{\left(\boldsymbol{A}^{-\top}\left(\frac{\mathrm{d}F}{\mathrm{d}\boldsymbol{x}}\right)^{\top}\right)^{\top}}_{\boldsymbol{x}_{\mathrm{aj}}^{\top}} \xrightarrow{\text{adjoint solution}} \boxed{\boldsymbol{A}^{\top}\boldsymbol{x}_{\mathrm{aj}} = \frac{\mathrm{d}F}{\mathrm{d}\boldsymbol{x}^{\top}}}$$

- final gradients

$$\frac{\mathrm{d}F}{\mathrm{d}\boldsymbol{g}} = \boldsymbol{x}_{\mathrm{aj}}^{\top}\left(\left[\frac{\partial \boldsymbol{b}}{\partial g_1}, \frac{\partial \boldsymbol{b}}{\partial g_2}, \cdots, \frac{\partial \boldsymbol{b}}{\partial g_m}\right] - \left[\frac{\partial A}{\partial g_1}\boldsymbol{x}, \frac{\partial A}{\partial g_2}\boldsymbol{x}, \cdots, \frac{\partial A}{\partial g_m}\boldsymbol{x}\right]\right)$$

$$\boxed{\frac{\mathrm{d}F}{\mathrm{d}\boldsymbol{g}} = \boldsymbol{x}_{\mathrm{aj}}^{\top}\left(\frac{\mathrm{d}\boldsymbol{b}}{\mathrm{d}\boldsymbol{g}} - \frac{\mathrm{d}\boldsymbol{A}}{\mathrm{d}\boldsymbol{g}}\boldsymbol{x}\right)}$$

requires two full solutions: $\underbrace{\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}}_{\text{direct}}$ $\underbrace{\boldsymbol{A}^{\top}\boldsymbol{x}_{\mathrm{aj}} = \frac{\mathrm{d}F}{\mathrm{d}\boldsymbol{x}^{\top}}}_{\text{adjoint}}$

Basics of adjoint formalism

- for complex quantities

  (no special derivation here)

$$\frac{\mathrm{d}F}{\mathrm{d}\boldsymbol{g}} = 2\,\mathrm{Re}\left\{\boldsymbol{z}_{\mathrm{aj}}^{\dagger}\left(\frac{\mathrm{d}\boldsymbol{b}}{\mathrm{d}\boldsymbol{g}} - \frac{\mathrm{d}\boldsymbol{A}}{\mathrm{d}\boldsymbol{g}}\boldsymbol{z}\right)\right\}$$

requires two full solutions

$$\underbrace{\boldsymbol{A}\boldsymbol{z} = \boldsymbol{b}}_{\text{direct}} \qquad \underbrace{\boldsymbol{A}^{\dagger}\boldsymbol{z}_{\mathrm{aj}} = \frac{\mathrm{d}F}{\mathrm{d}\boldsymbol{z}^{\dagger}}}_{\text{adjoint}}$$

- Maxwell's equations:

  (source does not depend on d.o.f.)

$$\frac{\mathrm{d}\boldsymbol{A}}{\mathrm{d}\boldsymbol{g}}\boldsymbol{z} = -\omega^2\mu_0\epsilon_0\,\mathrm{diag}(\boldsymbol{z})$$

$$\mathbf{M}^{\mathrm{T}}\tilde{\mathbf{E}}_{\mathrm{adj}} = \frac{\mathrm{d}F_{\mathrm{EM}}}{\mathrm{d}\tilde{\mathbf{E}}^{\mathrm{T}}}$$

$$\frac{\mathrm{d}F_{\mathrm{EM}}}{\mathrm{d}\epsilon} = 2\omega^2\epsilon_0\mu_0\Re\left(\tilde{\mathbf{E}}_{\mathrm{adj}} \odot \tilde{\mathbf{E}}\right)^{\mathrm{T}}$$

Example

imaging a point source (1.55 μm) to another point

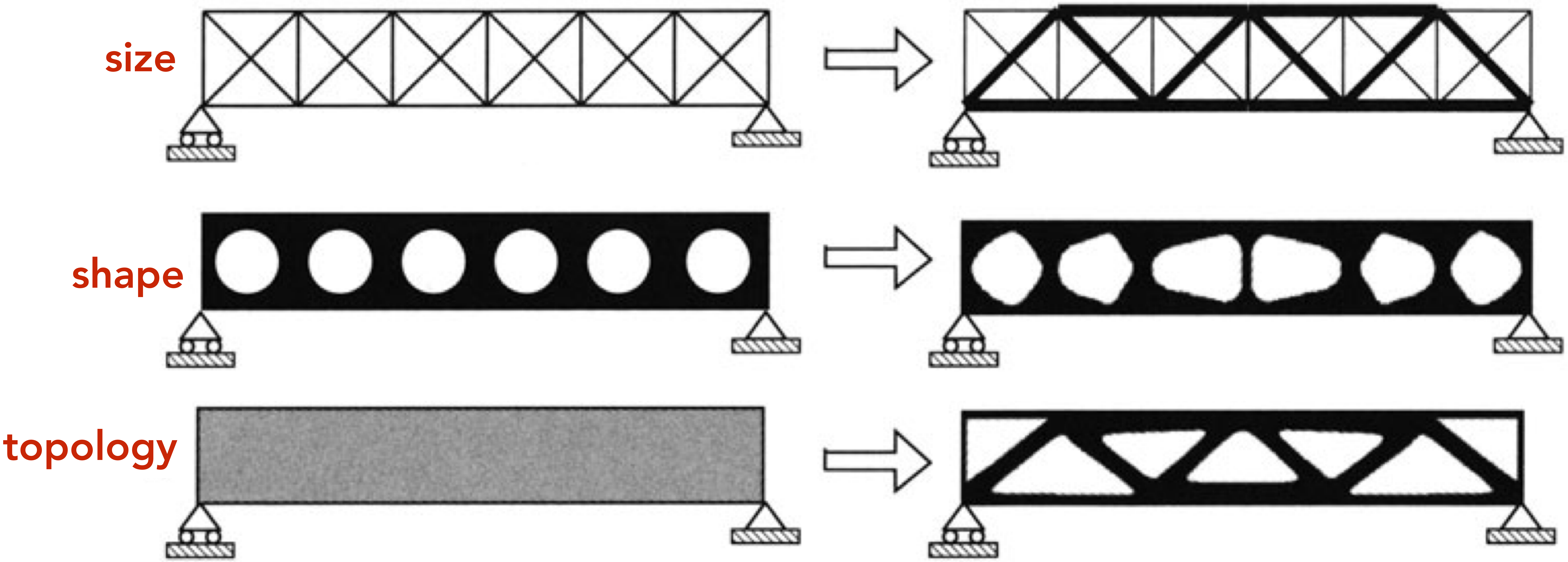
$\arg(\boldsymbol{E}_z)$

$\arg(\boldsymbol{E}_{z,\mathrm{aj}})$

$\mathrm{d}F/\mathrm{d}\boldsymbol{\epsilon}$

$$F(E_z) = \left|E_z(x_{\mathrm{obj}}, y_{\mathrm{obj}})\right|^2$$

Basics of topology optimisation



Martin P. Bendsøe and Ole Sigmund. Topology optimization: theory, methods, and applications. 2nd ed. Engineering online library. Berlin: Springer, 2004.

Basics of topology optimisation

parametrisation through density representation

- design space $$g_i = 1_{\Omega^{\text{mat}}}\, g_i^0$$ $$1_{\Omega^{\text{mat}}} = \begin{cases} 1 & \text{if} \quad x \in \Omega^{\text{mat}} \\ 0 & \text{if} \quad x \in \Omega \setminus \Omega^{\text{mat}} \end{cases}$$

  permittivity value in each pixel (g_i^0)

- discrete values represented by continuous variable

  density: $0 \leq \rho \leq 1$

  nanophotonics: $\epsilon_r = \epsilon_1 + \rho\,(\epsilon_2 - \epsilon_1)$

- permits gradient based optimisation $$\rho^{n+1} = \rho^n - \eta^n \frac{\mathrm{d}F}{\mathrm{d}\rho^n}$$

binarisation necessary → implicit constraints

Basics of topology optimisation

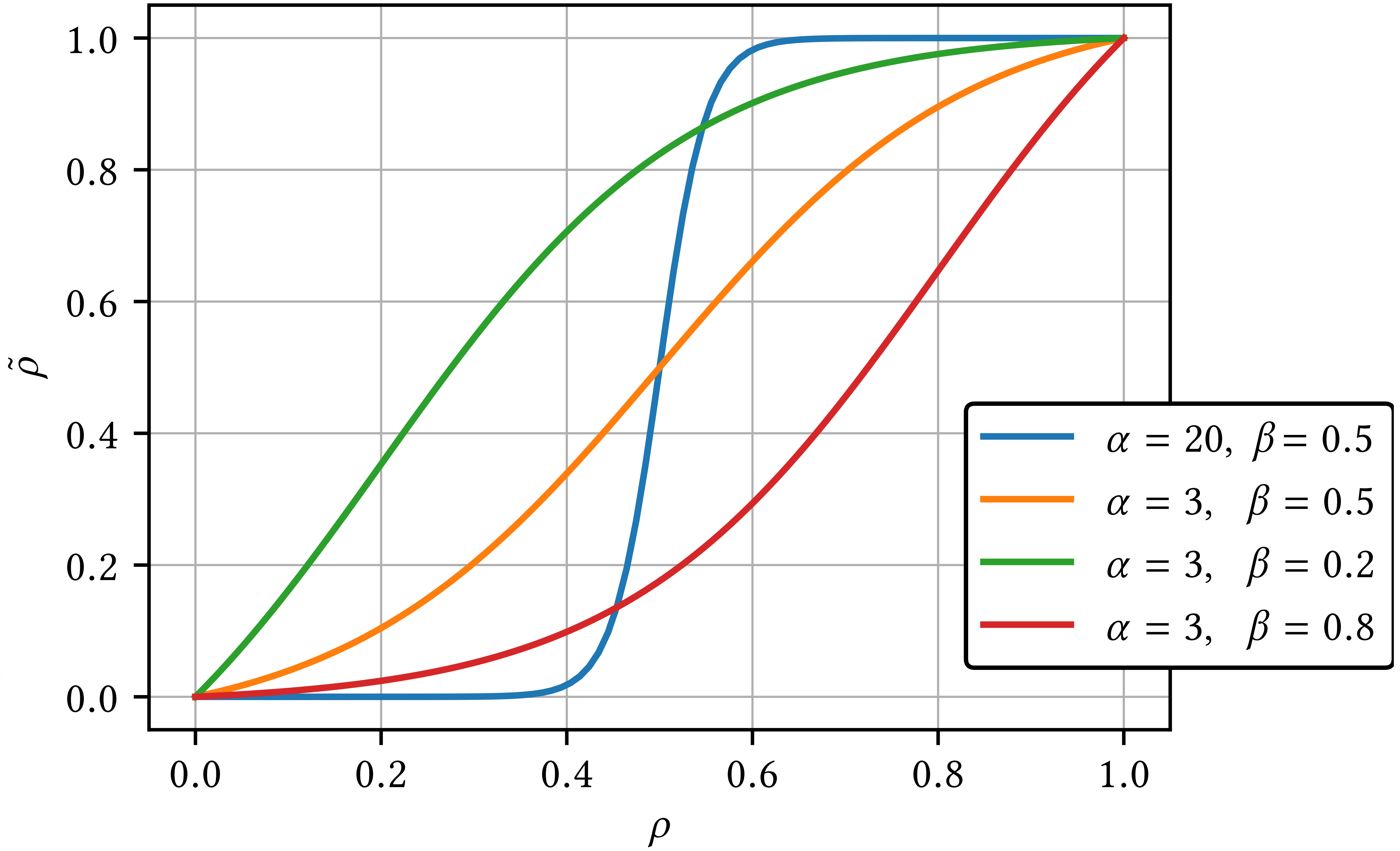
Simplified Isotropic Material with Penalization (SIMP) scheme

- power law $\epsilon_r = \epsilon_1 + \hat{\rho}\,(\epsilon_2 - \epsilon_1) \quad \text{with} \quad \hat{\rho} = \rho^p$

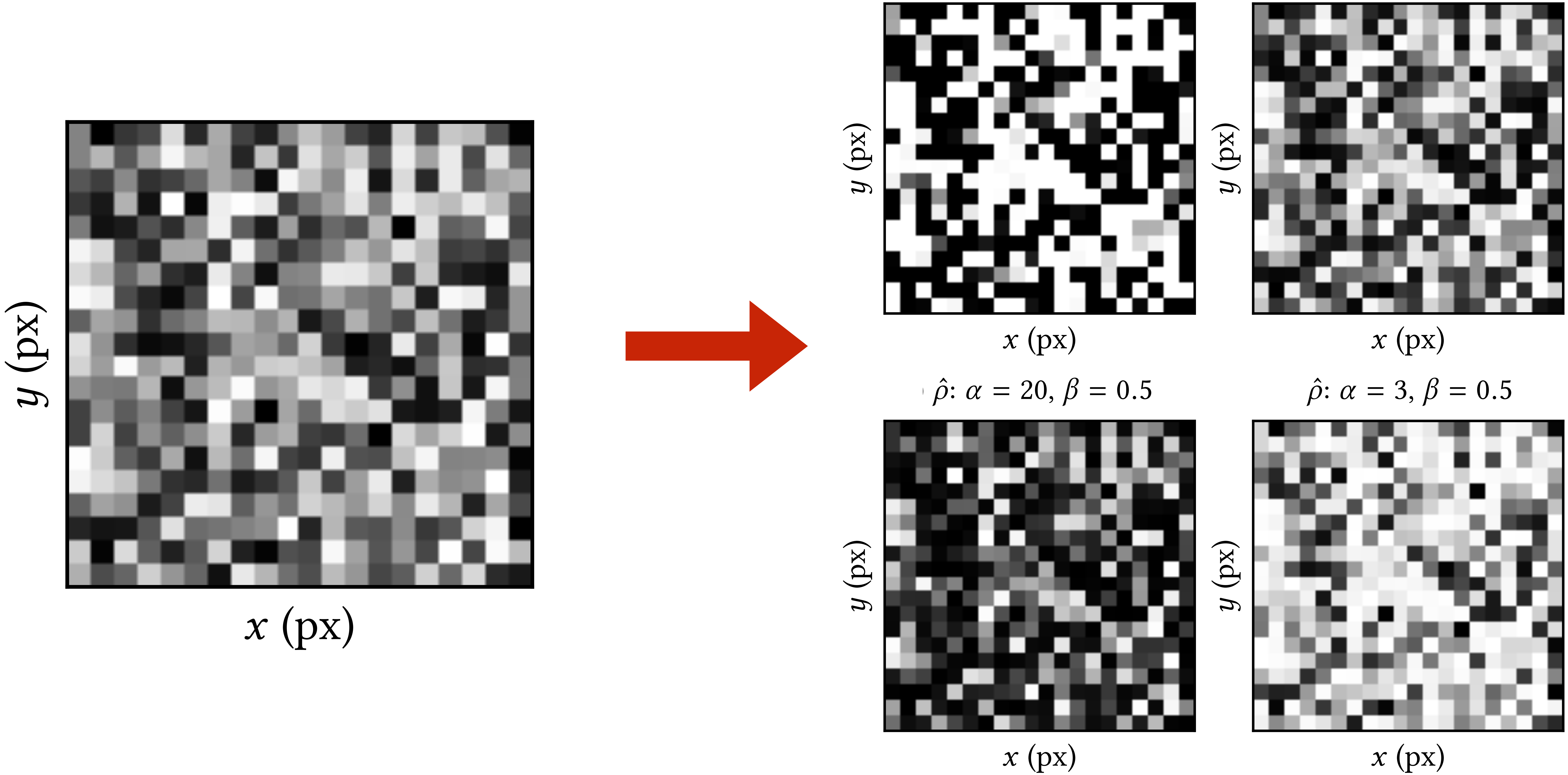
heuristically: $p = 3$

- projection scheme
  (smoothened Heaviside)

$$\hat{\rho} = \frac{\tanh(\alpha\beta) + \tanh(\alpha(\rho - \beta))}{\tanh(\alpha\beta) + \tanh(\alpha(1 - \beta))}$$



Basics of topology optimisation



$\hat{\rho}$: $\alpha = 20$, $\beta = 0.5$

$\hat{\rho}$: $\alpha = 3$, $\beta = 0.5$

$\hat{\rho}$: $\alpha = 3$, $\beta = 0.2$

$\hat{\rho}$: $\alpha = 3$, $\beta = 0.8$

Basics of topology optimisation

accommodating minimal feature sizes

- linear filtering (suffers from banding artefacts)

$$\tilde{\rho}_i = \frac{\sum_{j\in\mathcal{D}_i} w_{ij}\rho_j}{\sum_{j\in\mathcal{D}_i} w_{ij}} \quad \text{with} \quad w_{ij} = \begin{cases} r_{\min} - \left|\boldsymbol{r}_i - \boldsymbol{r}_j\right| & \forall \boldsymbol{r}_j \in \mathcal{D}_i \\ 0 & \text{otherwise} \end{cases}$$

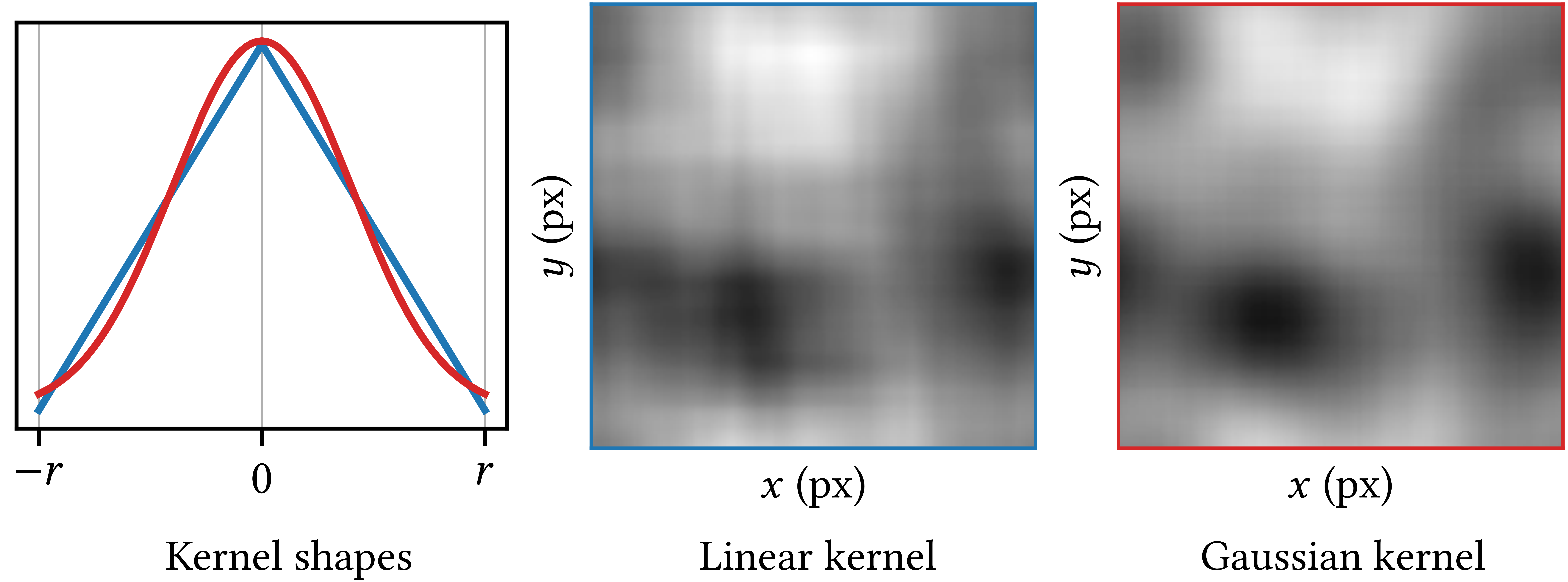
- Gaussian filtering

$$\tilde{\rho}_i = \frac{\sum_{j\in\mathcal{D}_i} w_{ij}\rho_j}{\sum_{j\in\mathcal{D}_i} w_{ij}} \quad \text{with} \quad w_{ij} = \begin{cases} r_{\min} \exp\left(-\frac{\left|\boldsymbol{r}_i - \boldsymbol{r}_j\right|^2}{2\sigma^2}\right) & \forall \boldsymbol{r}_j \in \mathcal{D}_i \\ 0 & \text{otherwise} \end{cases}$$

$$\sigma = r_{\min}/\sqrt{3}$$

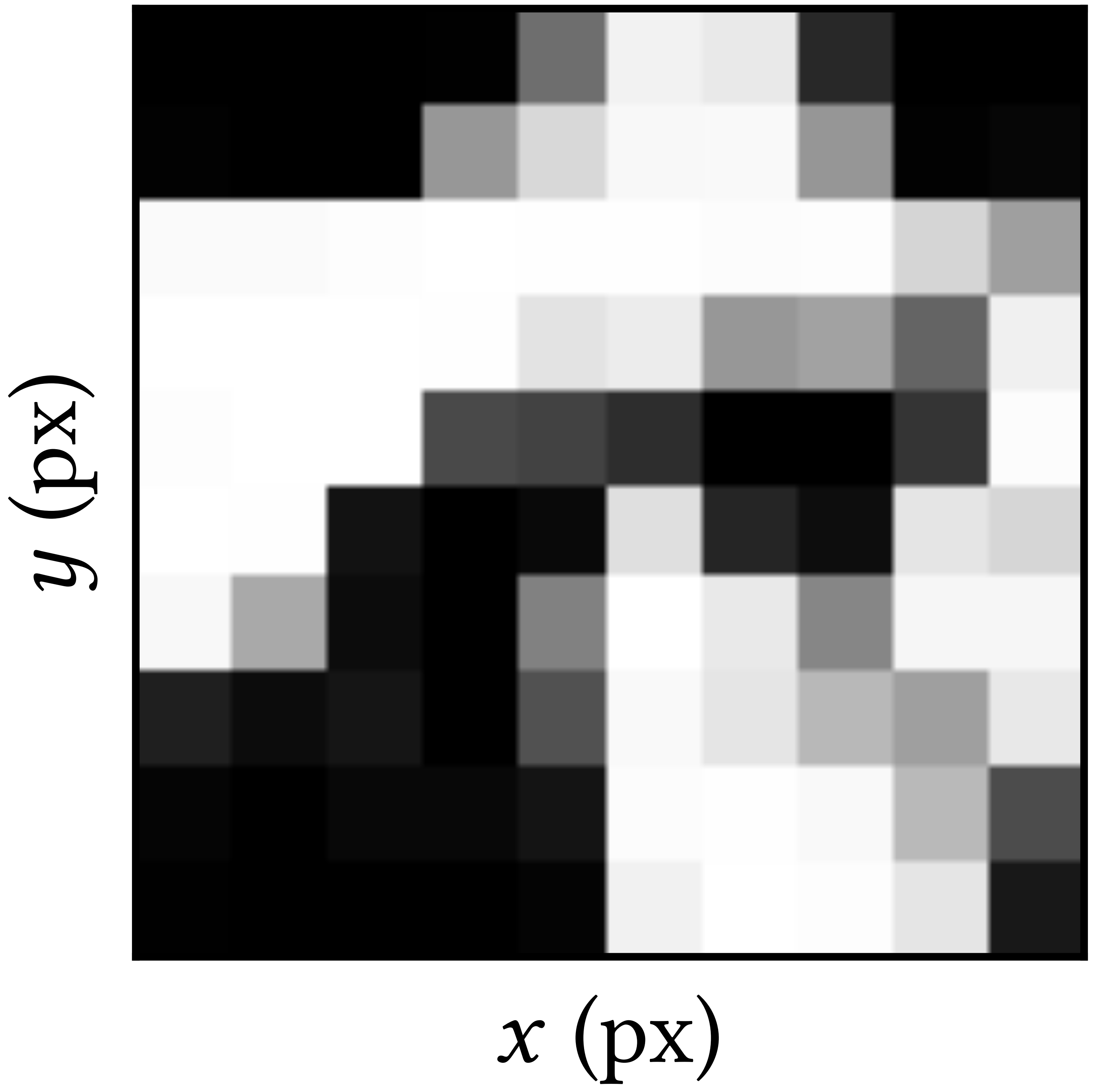
- implemented as convolution

$$\tilde{\rho}_n = (k * \rho)_n = \sum_{\substack{i=1 \\ 0<n-m+i\leq n}}^{m} k_{m-i+1}\, \rho_{n-m+i}\,,$$

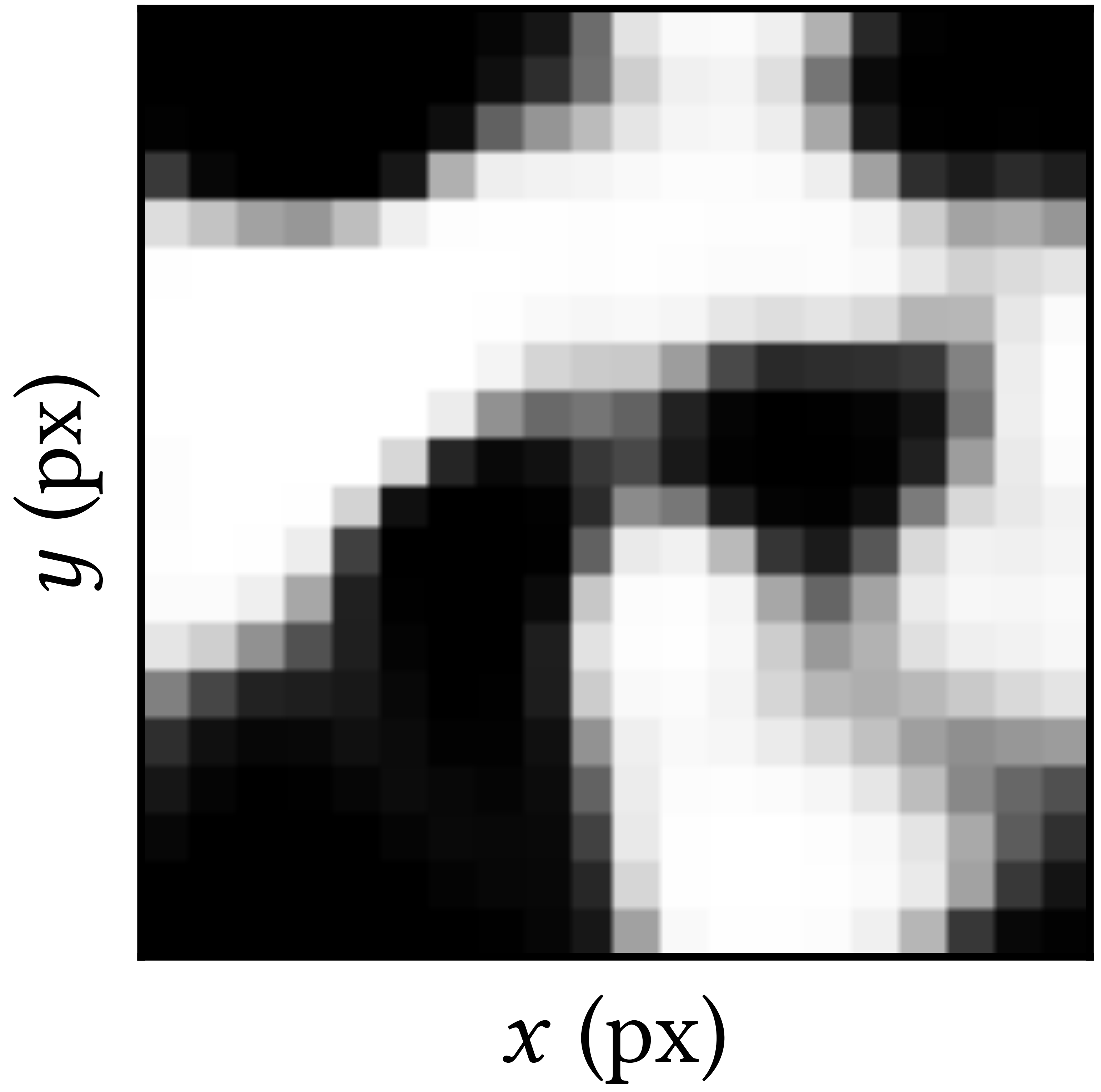

−r
0
r
Kernel shapes
y (px)
x (px)
Linear kernel
y (px)
x (px)
Gaussian kernel

mesh independency

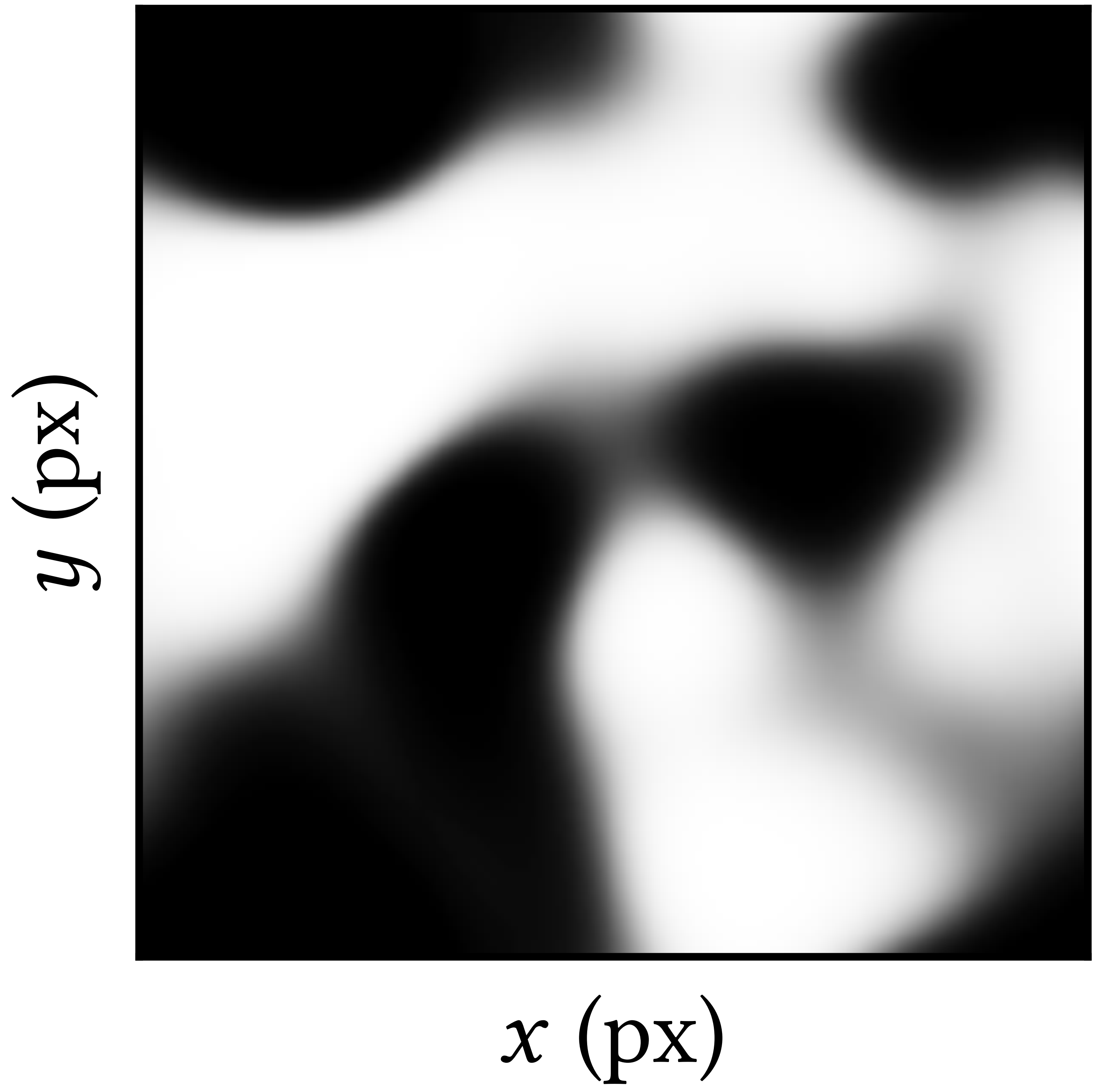
(results shall not depend on mesh size chosen in simulations)



10×10



20×20



1000×1000

Basics of topology optimisation

explicit constraints → add penalty terms to objective function

- volume constraint

(placing a fixed amount of material)

$$F_V(\rho) = -\frac{|V_{\text{target}} - V(\rho)|}{V_{\max}} \quad \text{with} \quad V(\rho) = \sum_{i\in\Omega} \rho_i$$

- binarization constraint

$$F_B(\rho) = |\langle \rho_\geq \rangle - \langle \rho_\leq \rangle| - 1$$

$$\rho_\geq = \sum_{\substack{i\in\Omega \\ \forall \rho \geq 0.5}} \rho_i \qquad \rho_\leq = \sum_{\substack{i\in\Omega \\ \forall \rho \leq 0.5}} \rho_i$$

all types of constraints are permissible

Now actual topology optimisation

starting from the definition of adjoint gradients and applying chain rule

$$\frac{\mathrm{d}F}{\mathrm{d}\rho} = -2\,\mathrm{Re}\left\{\boldsymbol{x}_{\mathrm{aj}}^{\dagger}\,\frac{\mathrm{d}\boldsymbol{A}}{\mathrm{d}\rho}\boldsymbol{x}\right\} = -2\,\mathrm{Re}\left\{\boldsymbol{x}_{\mathrm{aj}}^{\dagger}\,\frac{\mathrm{d}\boldsymbol{A}}{\mathrm{d}\epsilon_r}\frac{\mathrm{d}\epsilon_r}{\mathrm{d}\hat{\rho}}\frac{\mathrm{d}\hat{\rho}}{\mathrm{d}\tilde{\rho}}\frac{\mathrm{d}\tilde{\rho}}{\mathrm{d}\rho}\boldsymbol{x}\right\}$$

- derivative of density parametrisation

$$\epsilon_r(\hat{\rho}) = \epsilon_1 + \hat{\rho}\,(\epsilon_2 - \epsilon_1) \quad \Rightarrow \quad \frac{\mathrm{d}\epsilon_r}{\mathrm{d}\hat{\rho}} = (\epsilon_2 - \epsilon_1)$$

- derivative of binarisation

$$\hat{\rho}(\tilde{\rho}) = \frac{\tanh(\alpha\beta) + \tanh(\alpha(\tilde{\rho} - \beta))}{\tanh(\alpha\beta) + \tanh(\alpha(1 - \beta))} \quad \Rightarrow \quad \frac{\mathrm{d}\hat{\rho}}{\mathrm{d}\tilde{\rho}} = \frac{\alpha - \alpha\tanh^2(\alpha(\tilde{\rho} - \beta))}{\tanh(\alpha\beta) + \tanh(\alpha(1 - \beta))}$$

Now actual topology optimisation

- derivative of convolution

$$\tilde{\rho}(\rho) = k * \rho = \begin{pmatrix} k_1\rho_1 \\ k_2\rho_1 + k_1\rho_2 \\ k_3\rho_1 + k_2\rho_2 + k_1\rho_3 \\ \vdots \\ \sum_{i=1}^{m} k_{m-i+1}\rho_{n-m+i} \end{pmatrix}$$

with $\quad 0 < n - m + i \leq n$

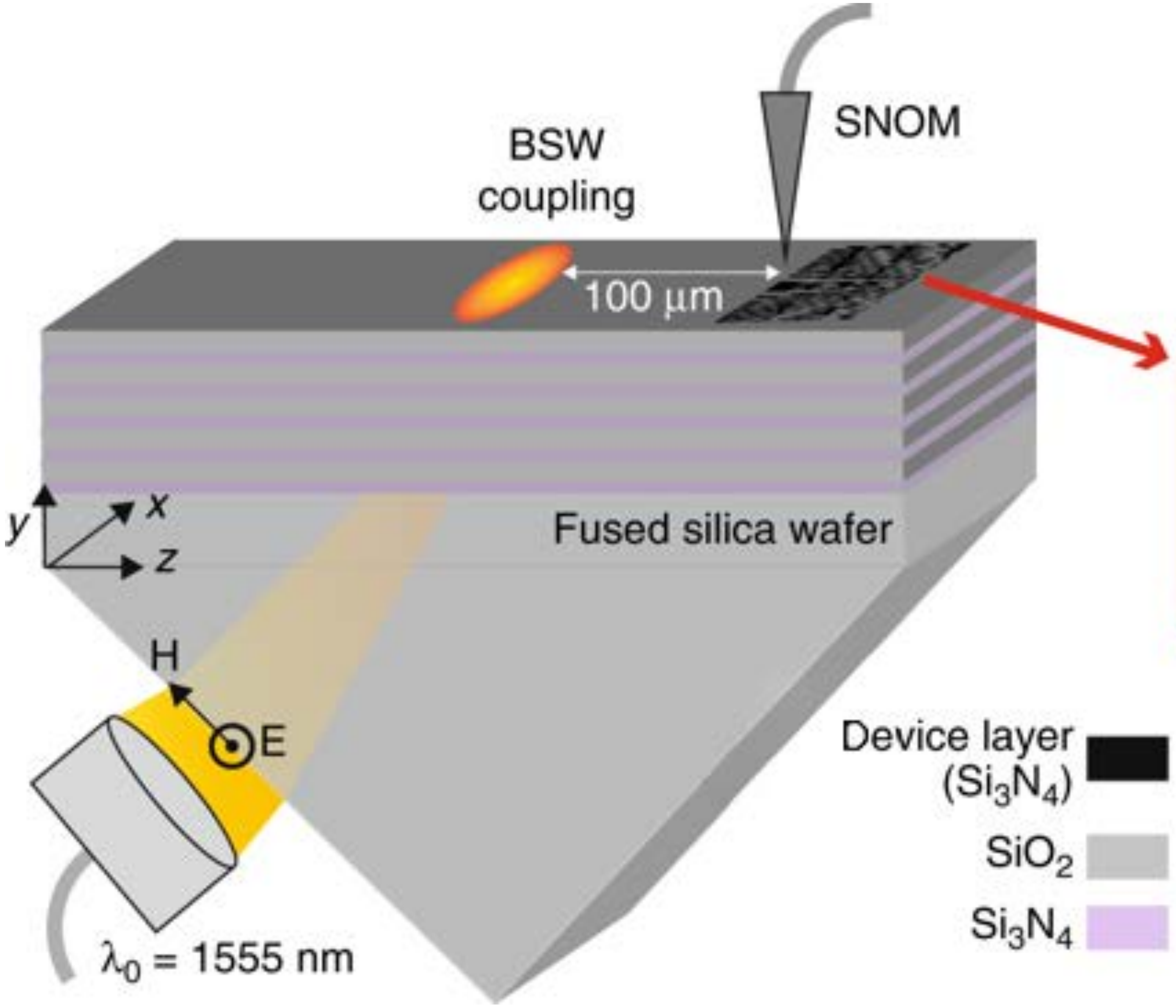
$$\tilde{\rho}(\rho) = K\rho = \begin{pmatrix} k_1 & 0 & \cdots & 0 & 0 \\ k_2 & k_1 & \cdots & 0 & 0 \\ k_3 & k_2 & \cdots & 0 & 0 \\ \vdots & k_3 & \cdots & k_1 & 0 \\ k_{m-2} & \vdots & \cdots & k_2 & k_1 \\ k_{m-1} & k_{m-2} & \vdots & k_3 & k_2 \\ k_m & k_{m-1} & \cdots & \vdots & k_3 \\ 0 & k_m & \cdots & k_{m-2} & \vdots \\ 0 & 0 & \cdots & k_{m-1} & k_{m-2} \\ 0 & 0 & \cdots & k_m & k_{m-1} \\ 0 & 0 & \cdots & 0 & k_m \end{pmatrix} \begin{pmatrix} \rho_1 \\ \rho_2 \\ \rho_3 \\ \vdots \\ \rho_{n-2} \\ \rho_{n-1} \\ \rho_n \end{pmatrix}$$

$\rightarrow$ $$\frac{\mathrm{d}\tilde{\rho}}{\mathrm{d}\rho} = K$$

finally

$$\frac{\mathrm{d}F}{\mathrm{d}\rho} = -2\,\mathrm{Re}\left\{ \boldsymbol{x}_{\mathrm{aj}}^{\dagger} \frac{\mathrm{d}\boldsymbol{A}}{\mathrm{d}\epsilon_r} \frac{\mathrm{d}\epsilon_r}{\mathrm{d}\hat{\rho}} \frac{\mathrm{d}\hat{\rho}}{\mathrm{d}\tilde{\rho}} \frac{\mathrm{d}\tilde{\rho}}{\mathrm{d}\rho} \boldsymbol{x} \right\} = 2\omega^2 \mu_0 \epsilon_0 \left(\epsilon_2 - \epsilon_1\right) \frac{\mathrm{d}\hat{\rho}}{\mathrm{d}\tilde{\rho}} K\, \mathrm{Re}\left\{ \boldsymbol{x}_{\mathrm{aj}}^{\dagger}\, \boldsymbol{x} \right\}$$

Example I



Focusing of a Bloch Surface Wave

objective function

$$F(\boldsymbol{E}) = \sum_{i,j\in\chi} \left|\boldsymbol{E}_{ij}\right|^2$$

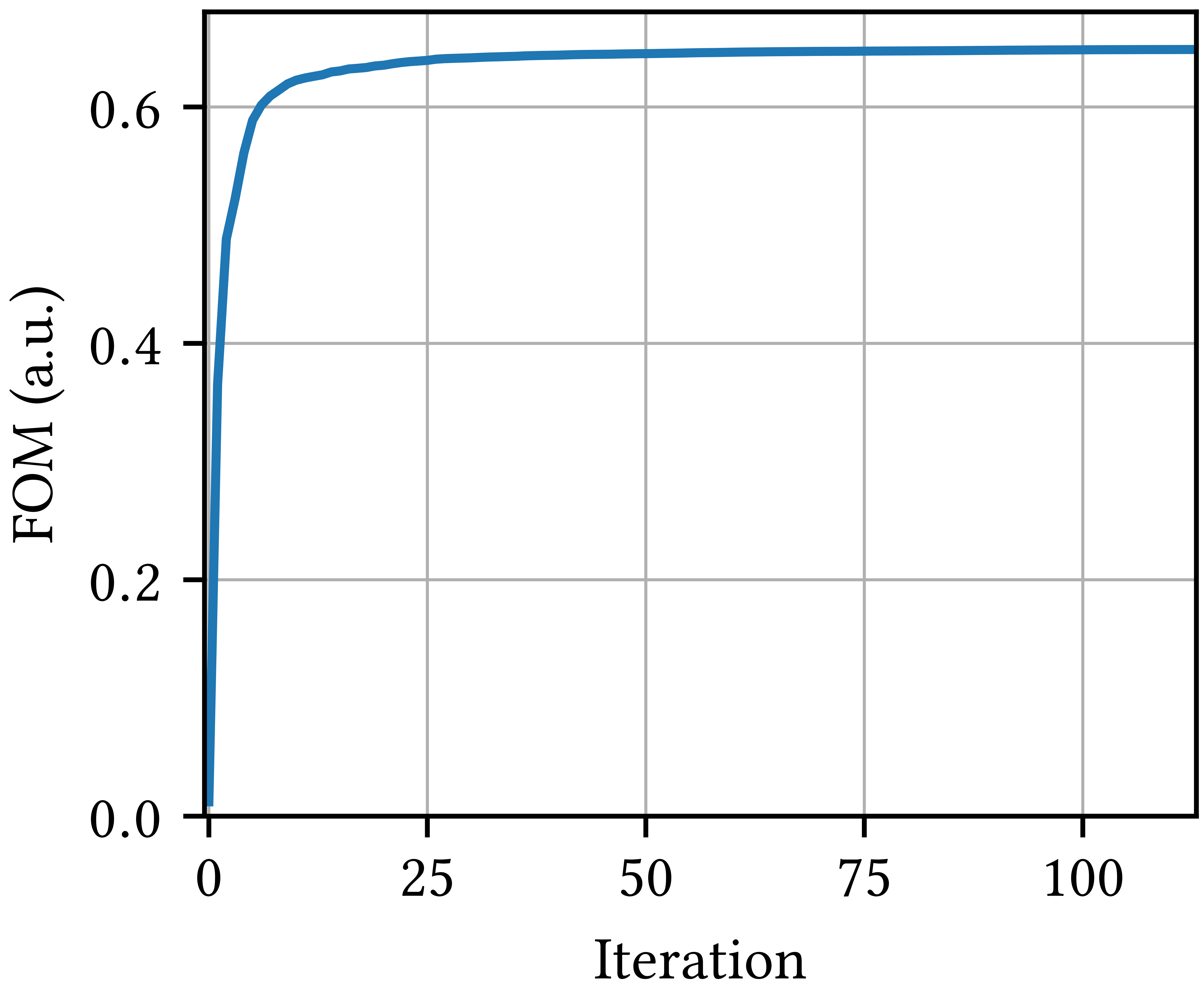
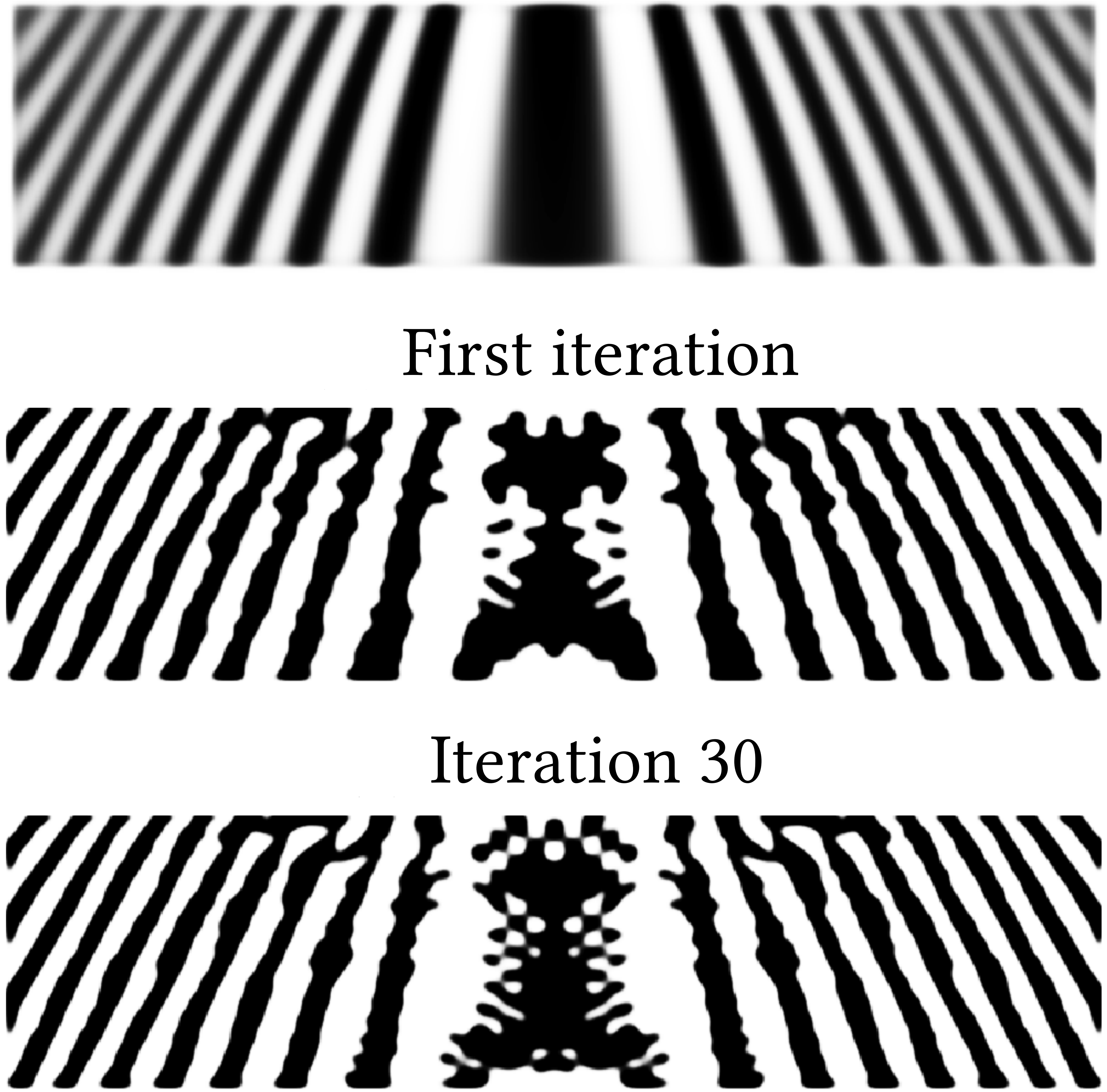


Figure of merit

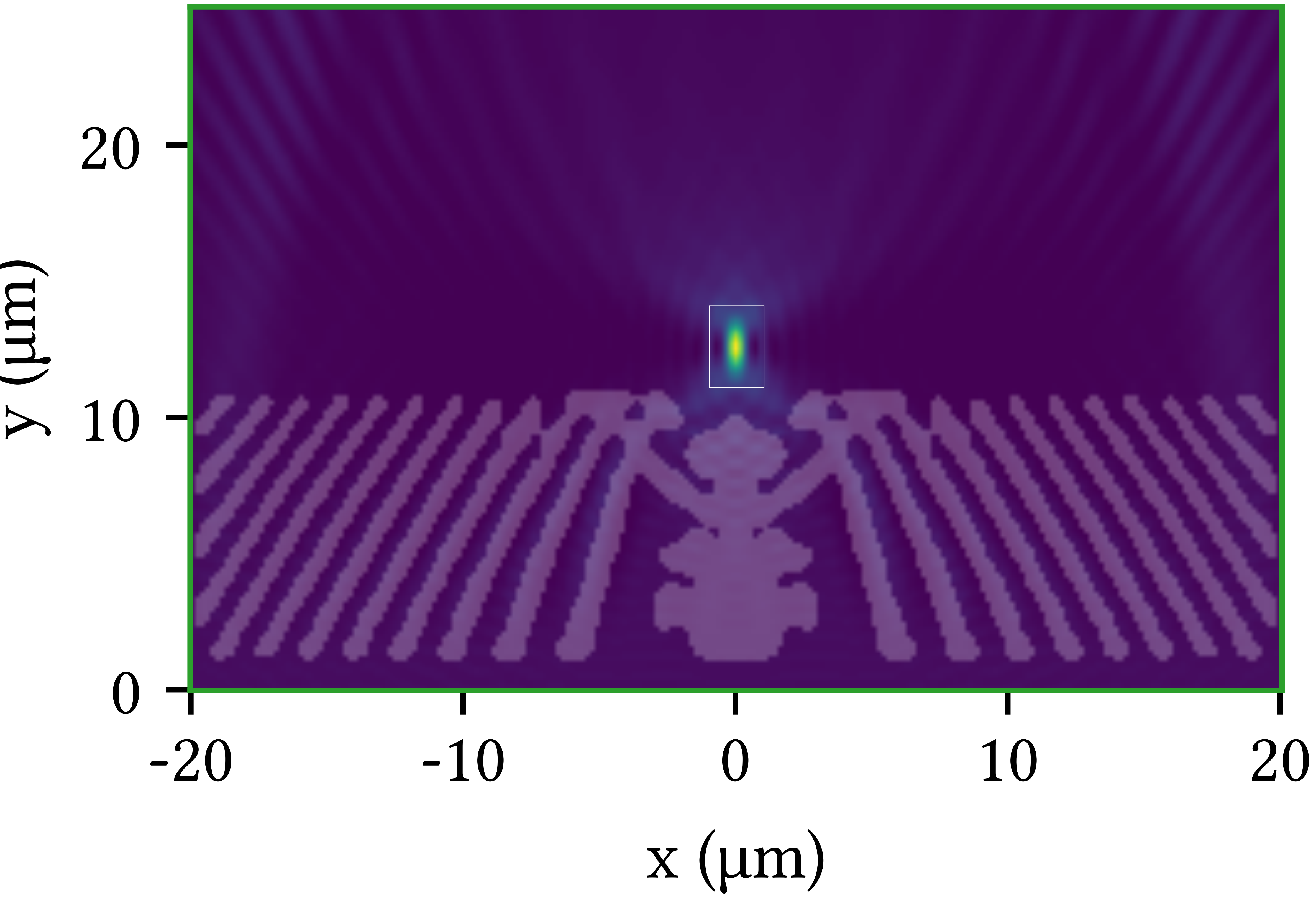


First iteration

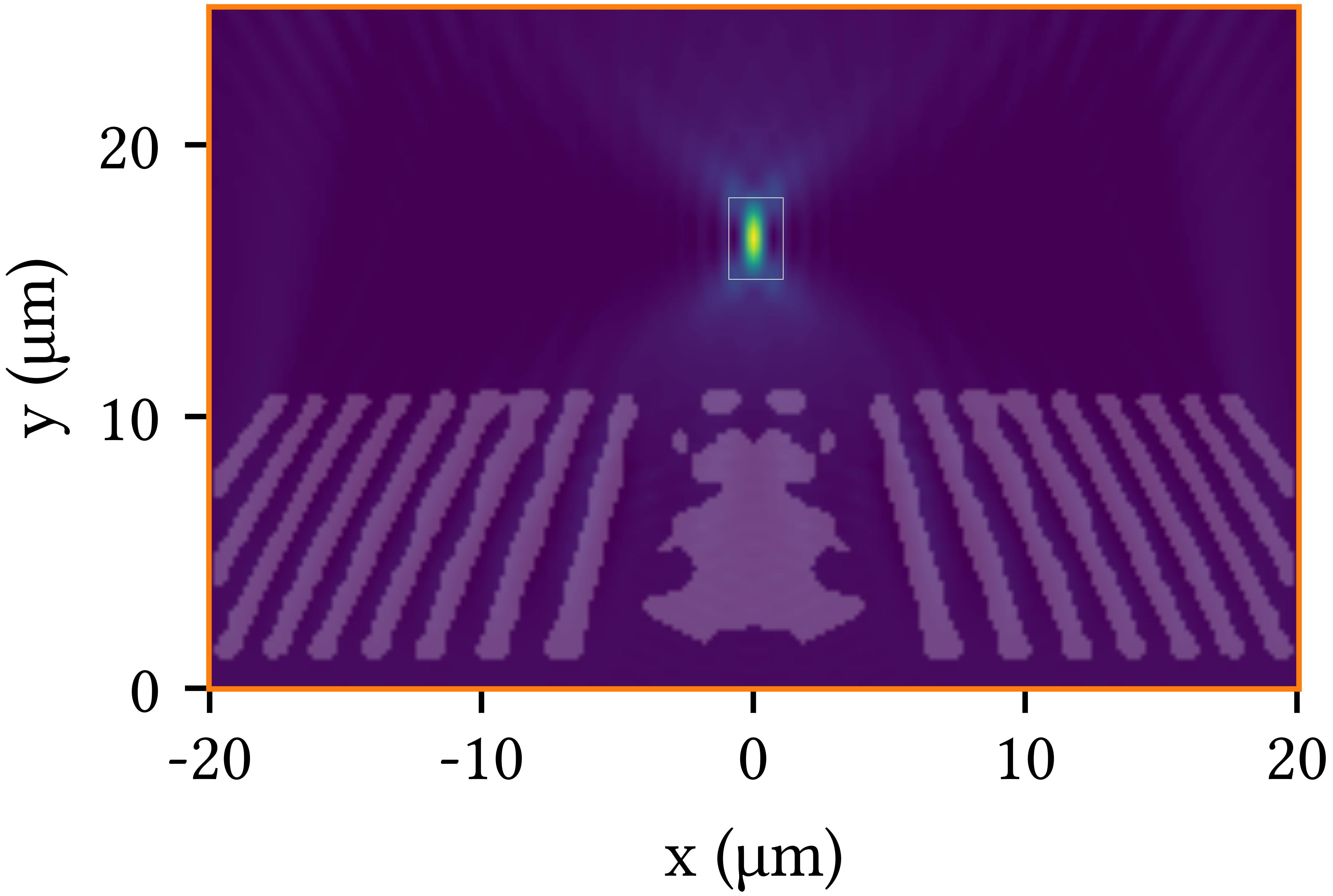
Iteration 30

Final iteration

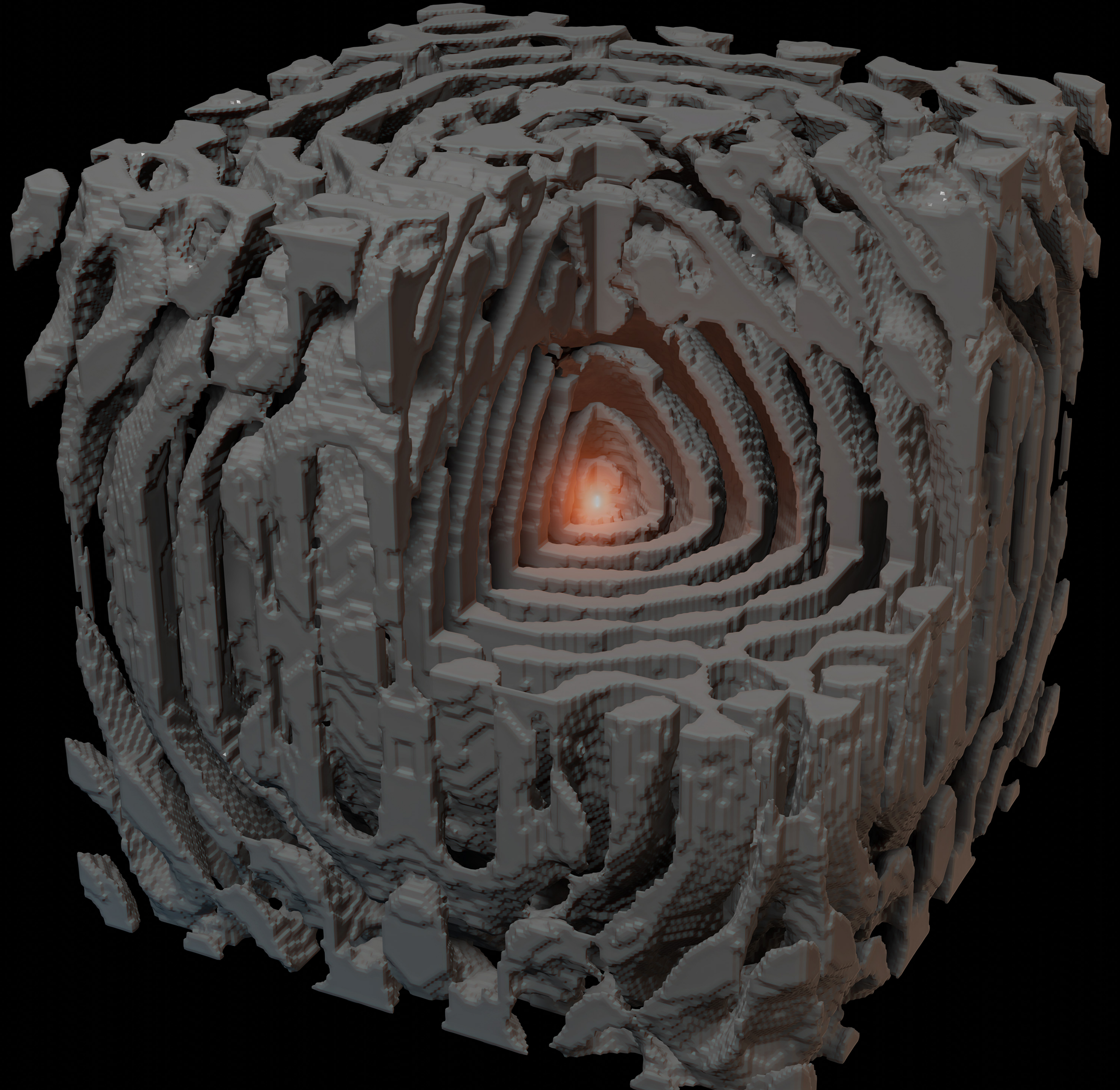
Example I



$d = 1.0\,\mu\text{m}$, $l = 1\,\mu\text{m}$

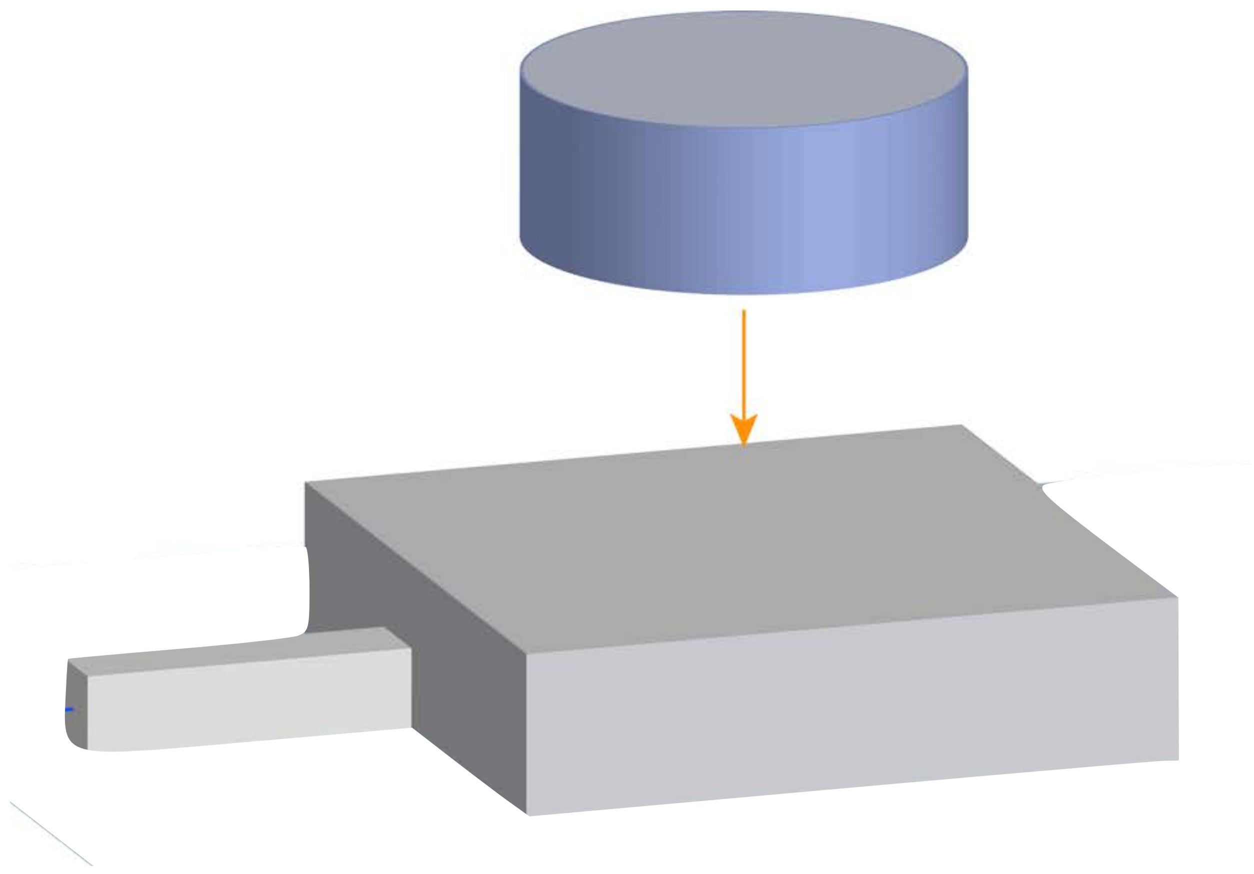


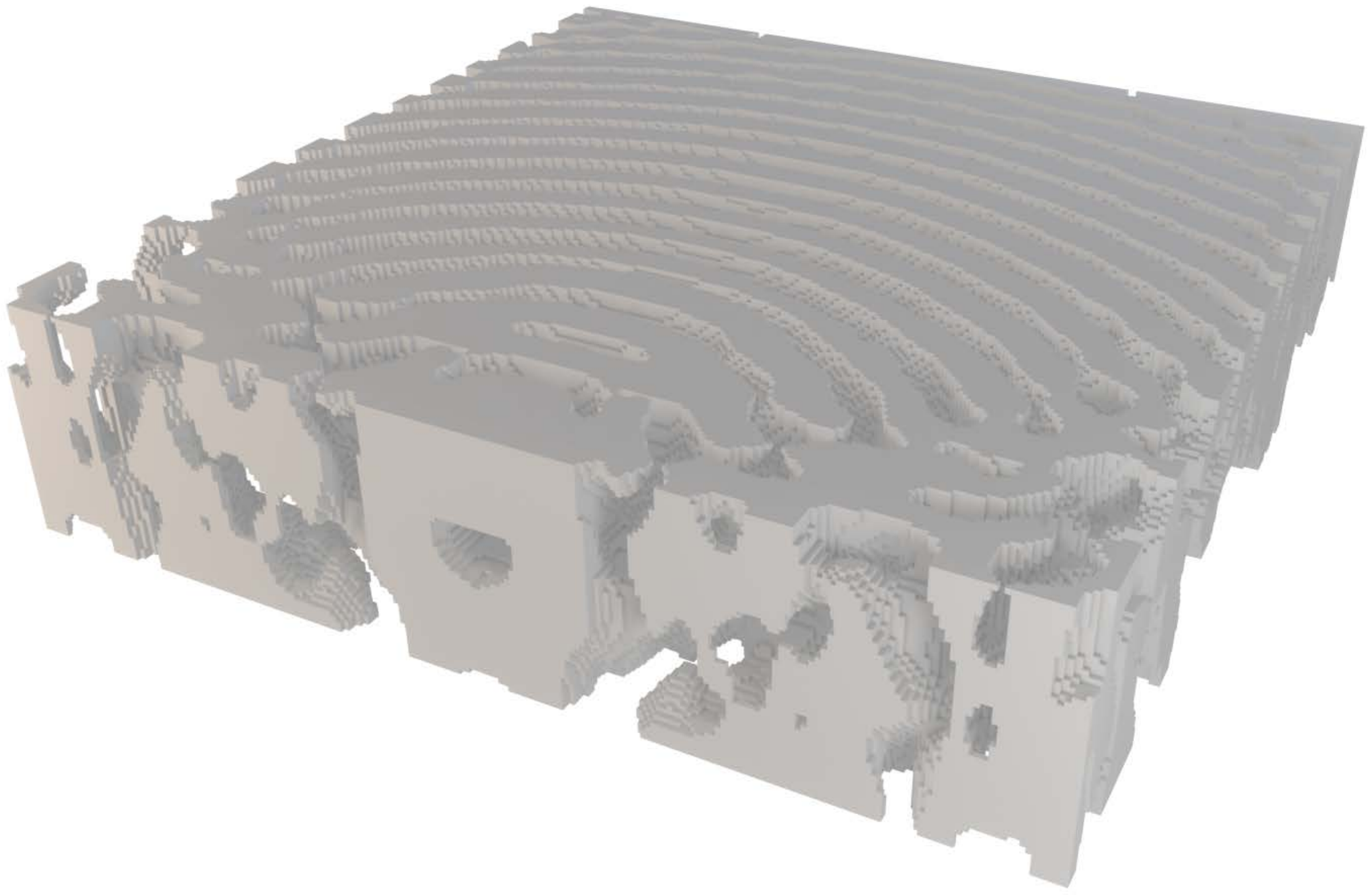
$d = 1.0\,\mu\text{m}$, $l = 5\,\mu\text{m}$

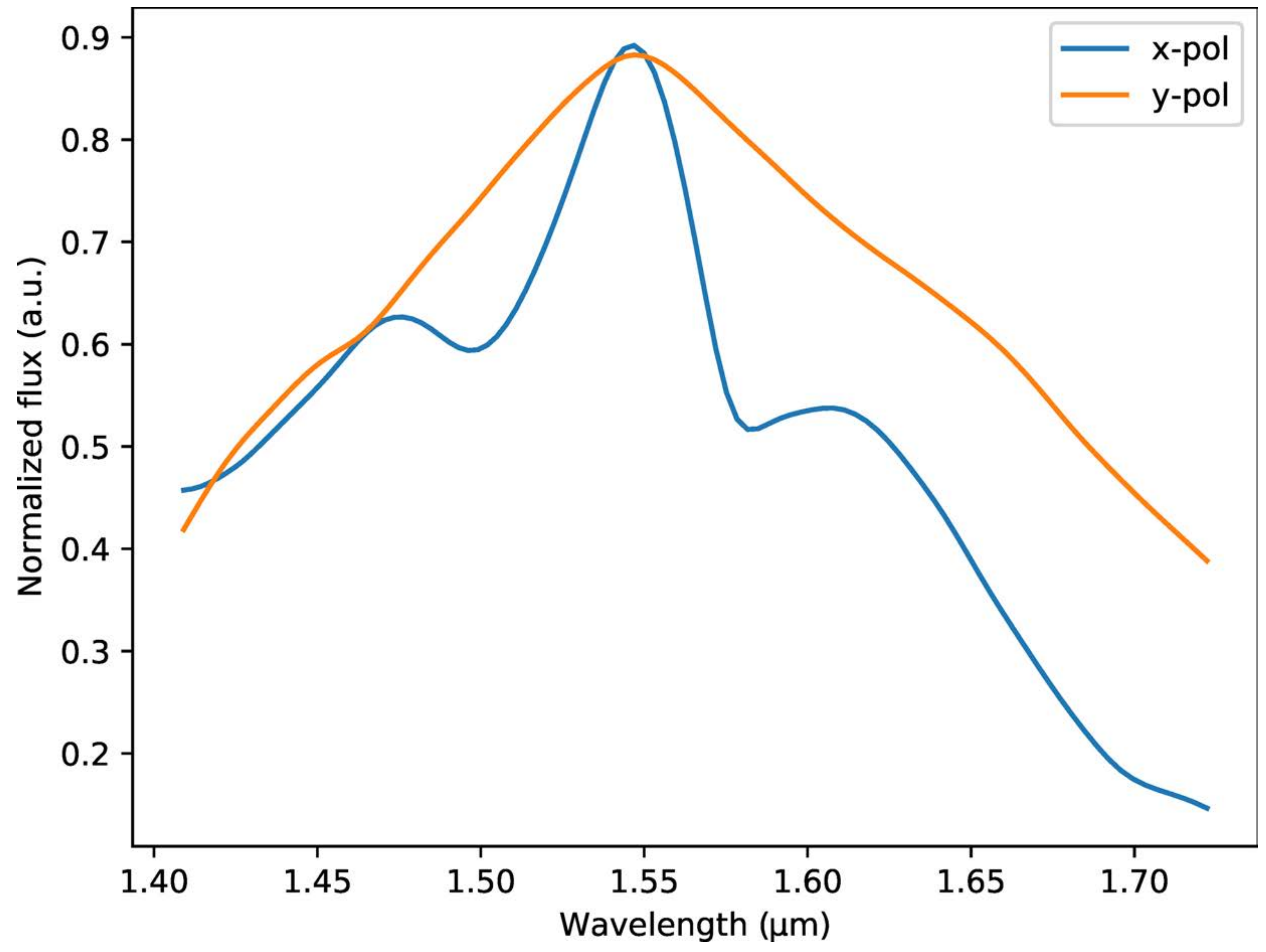


Example III

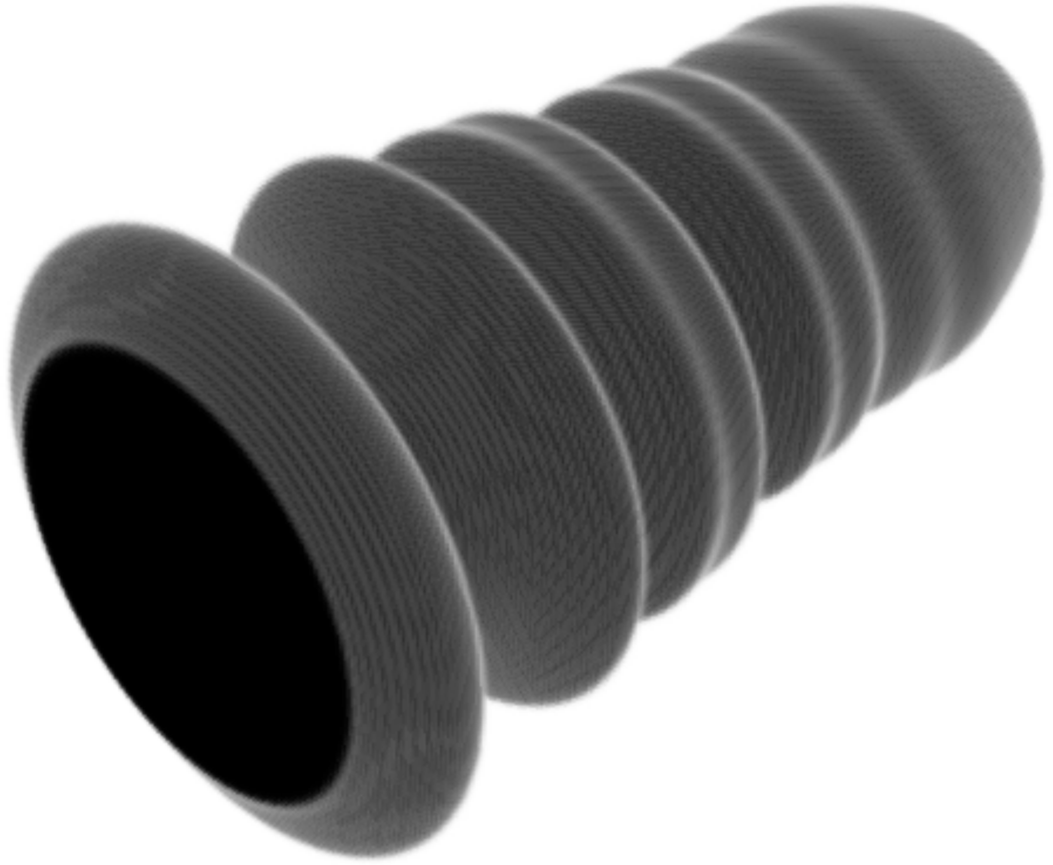
Top-coupler

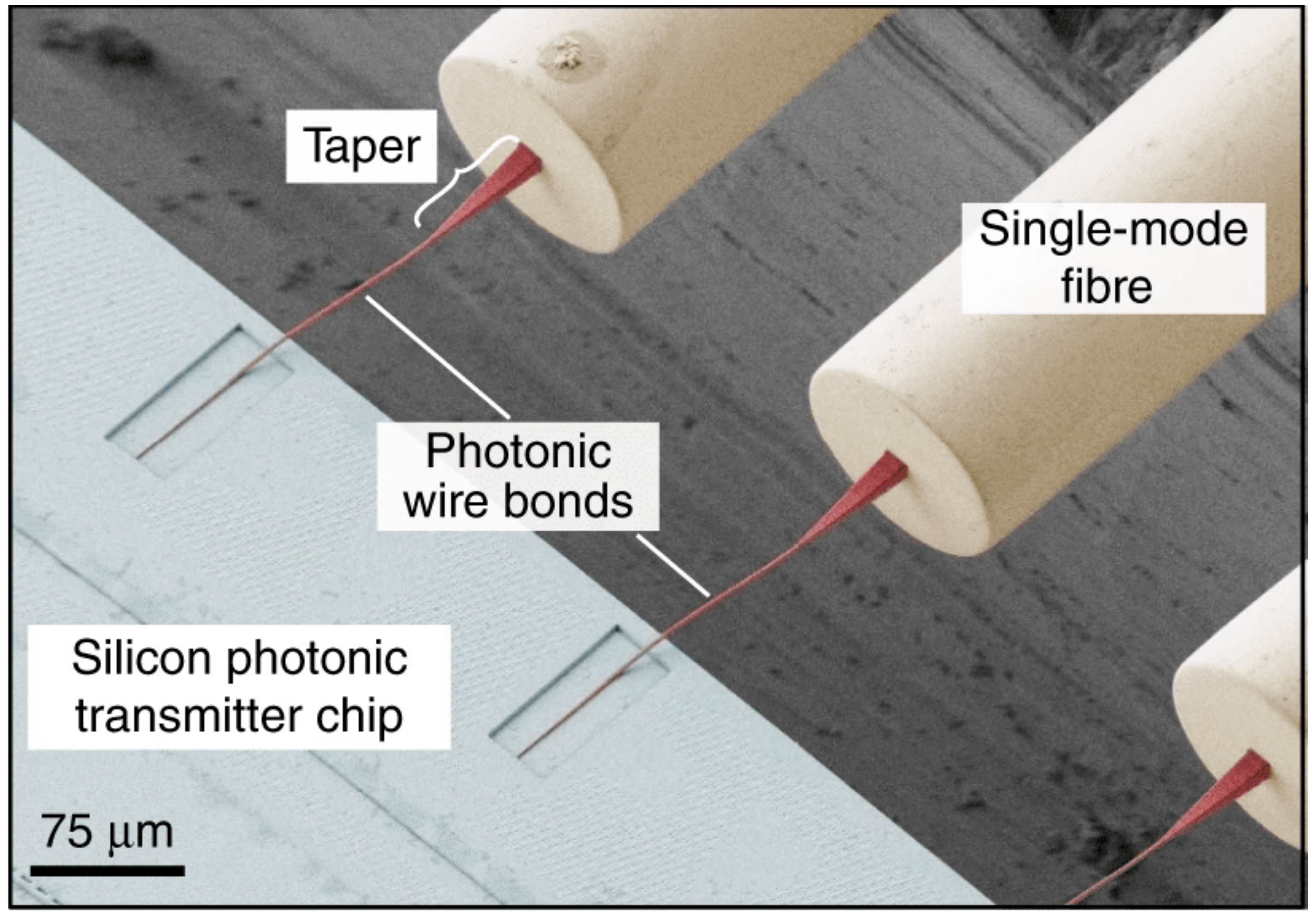


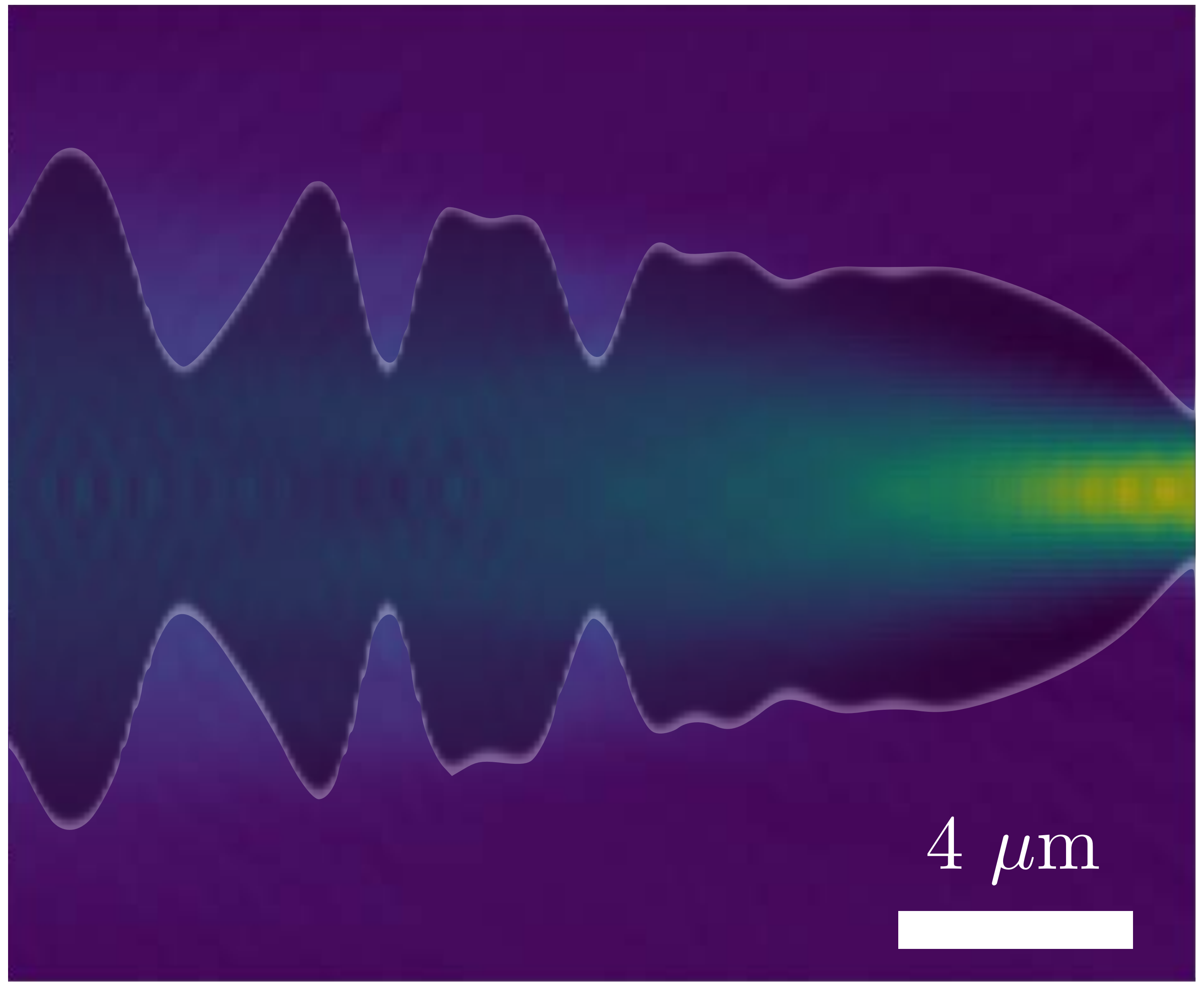




Butt-Coupler







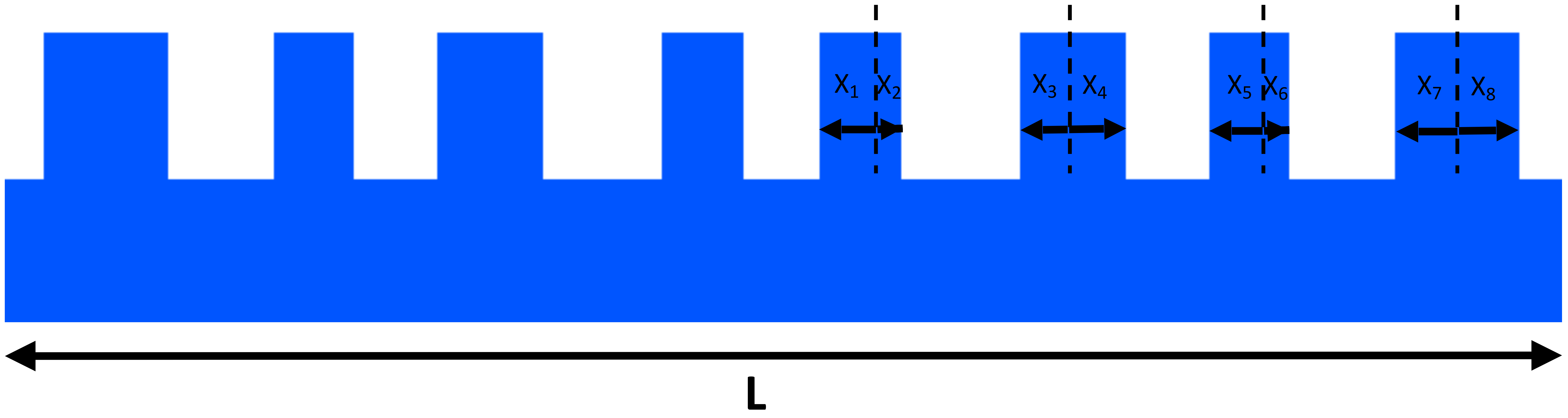
Bayesian optimization

strive to find a global optimum $f_{\text{ob}}(\mathbf{x}_{\text{opt}}) \leq f_{\text{ob}}(\mathbf{x}) \, \forall \, \mathbf{x} \in \mathcal{D}$

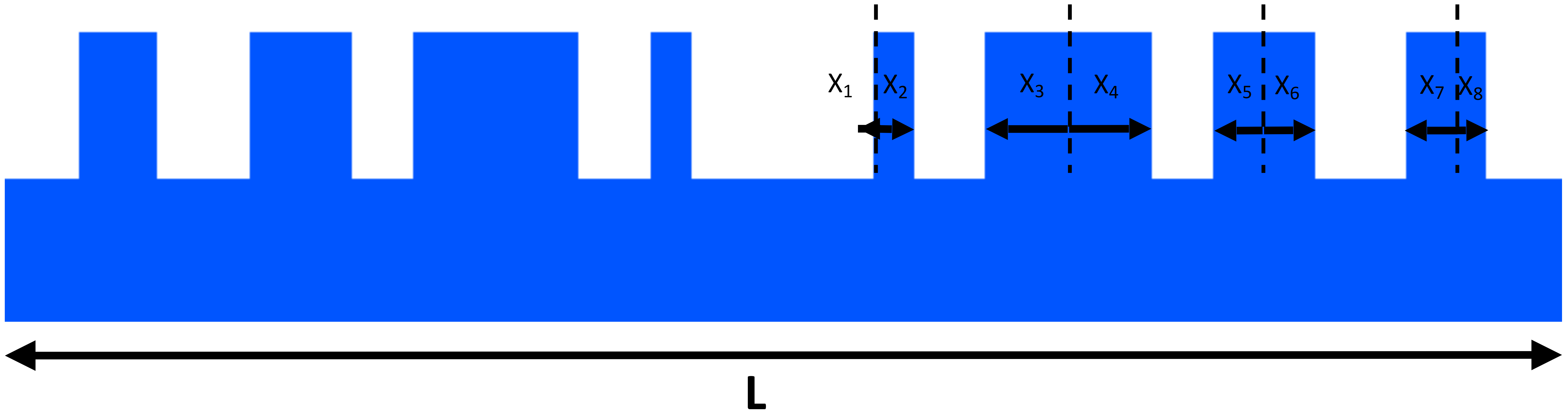
- particular design parametrised in vector $\mathbf{x} = [x_1, x_2, ..., x_d]^T$
- one element in the possible design space $\mathcal{D}$
- Gaussian processes: all possible functions that possibly could be the objective function, mean and standard deviation are well defined
- derive from that a next data point to be evaluated

Curse of dimensionality

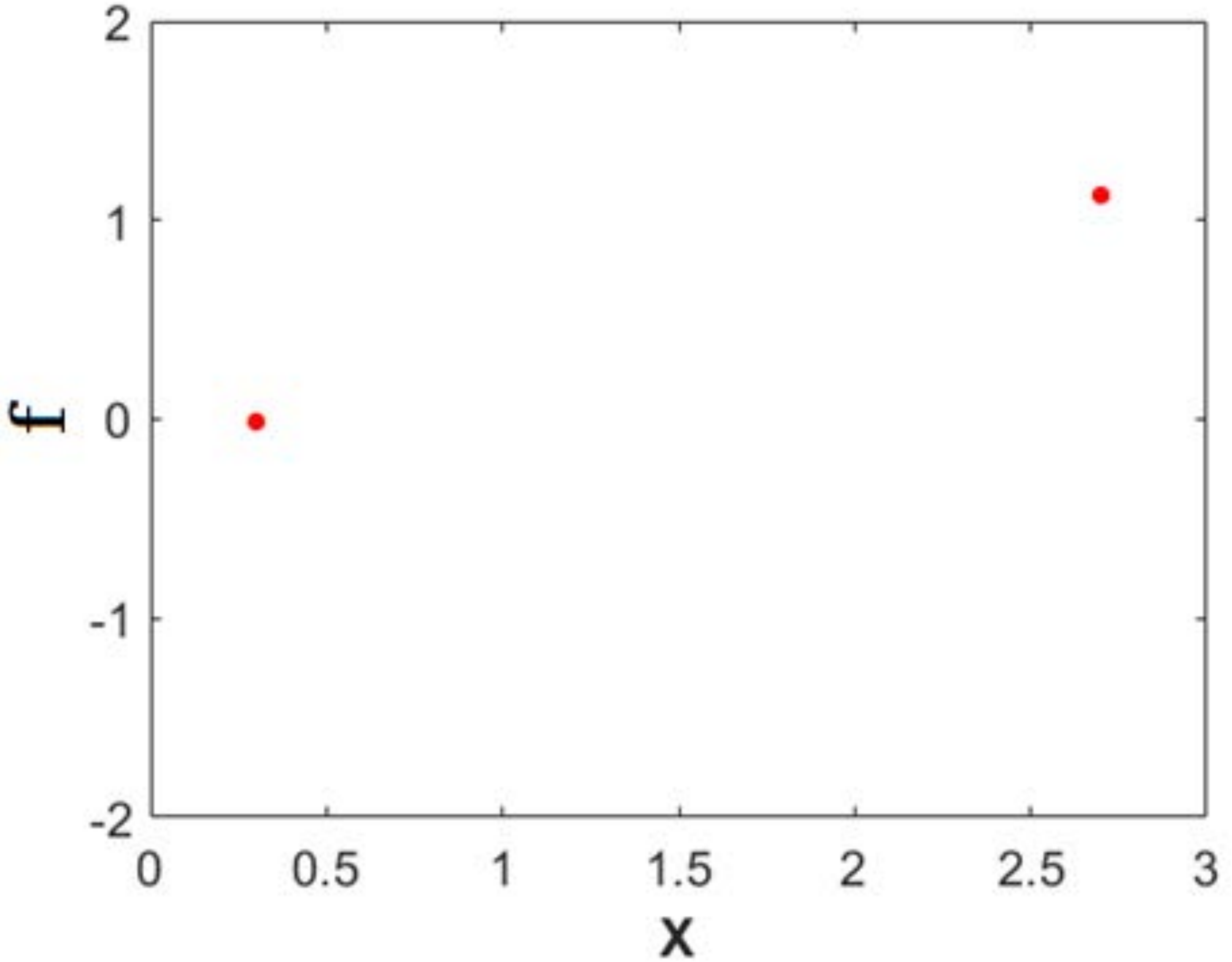
sample 1



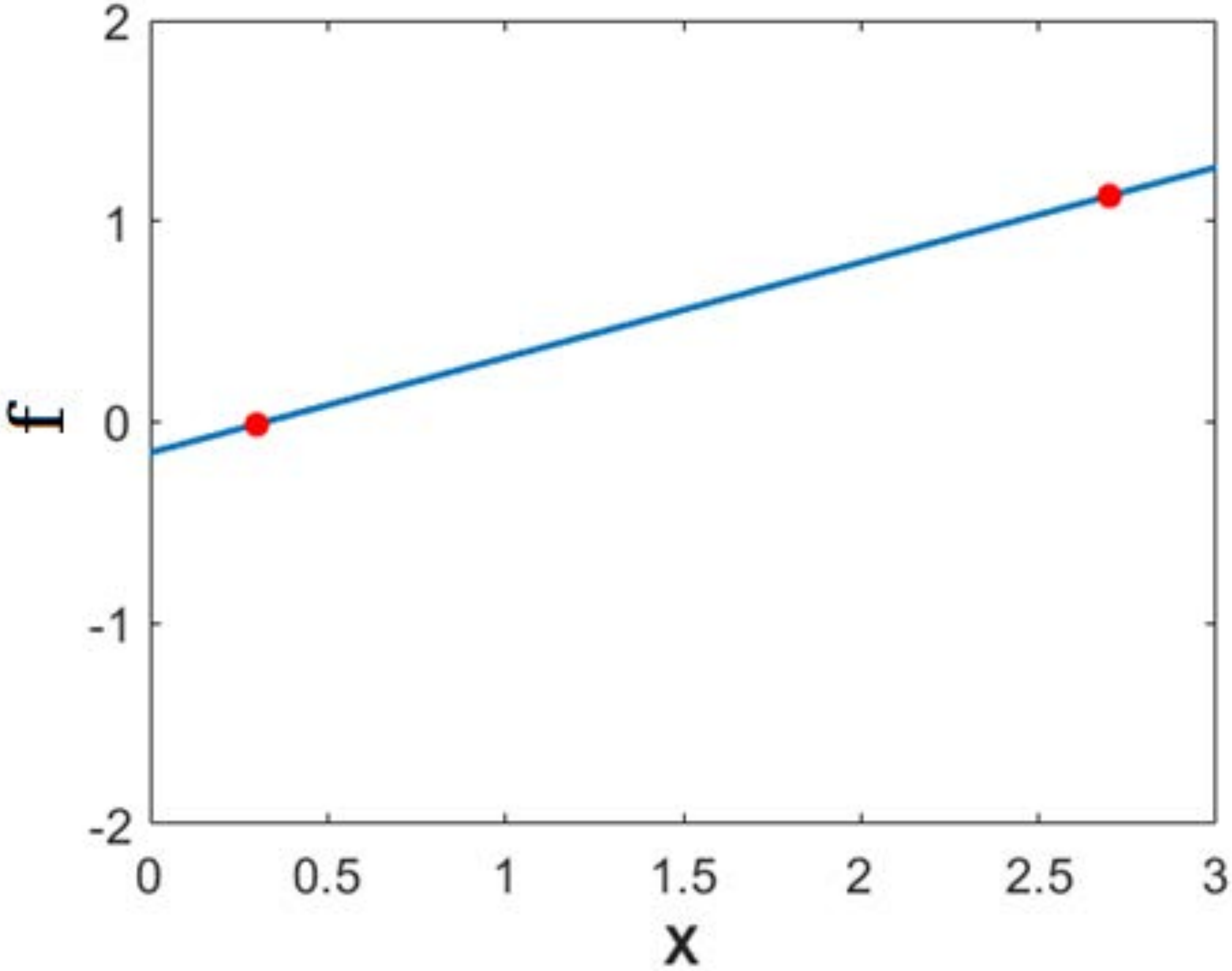
sample 2



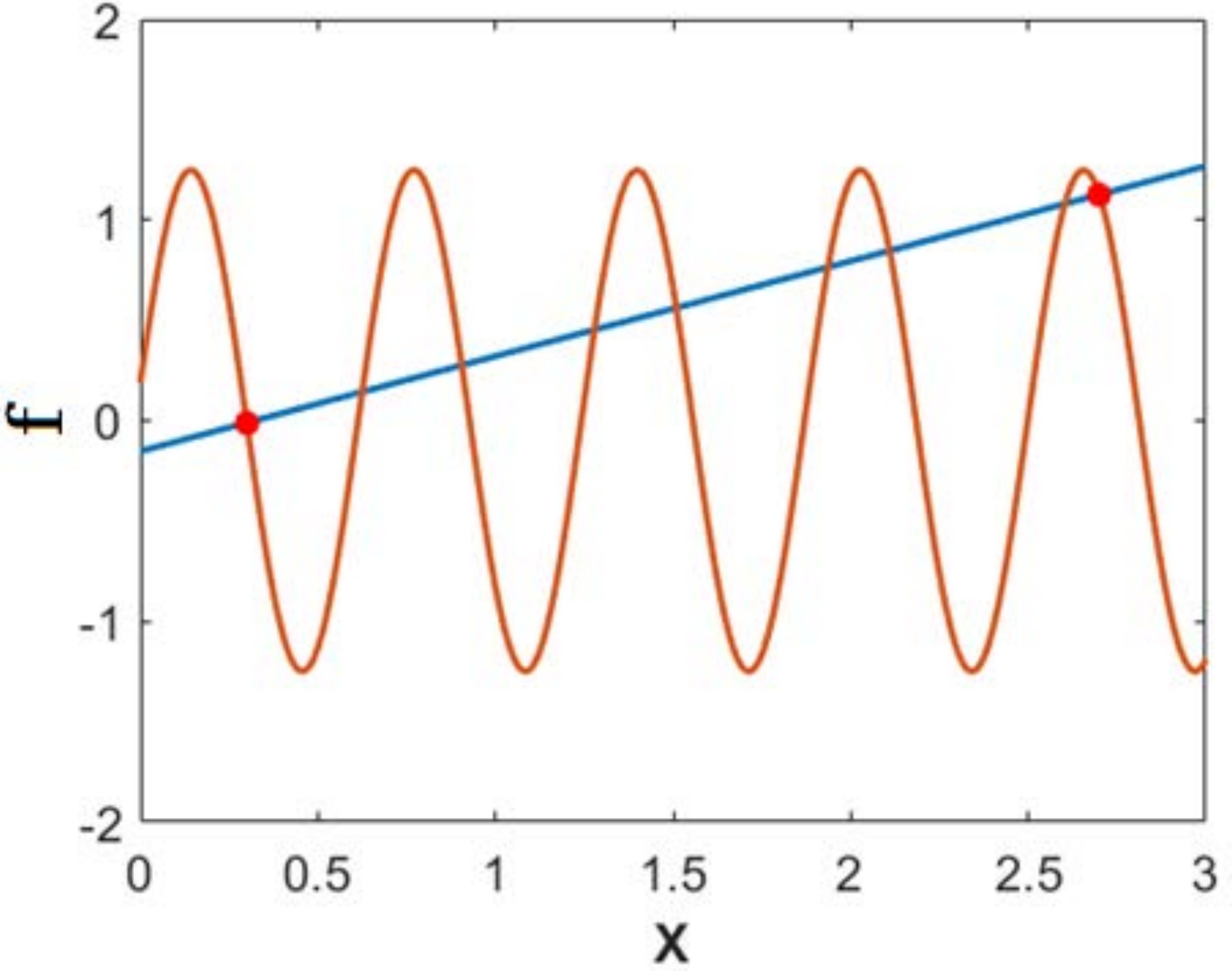
Curse of dimensionality



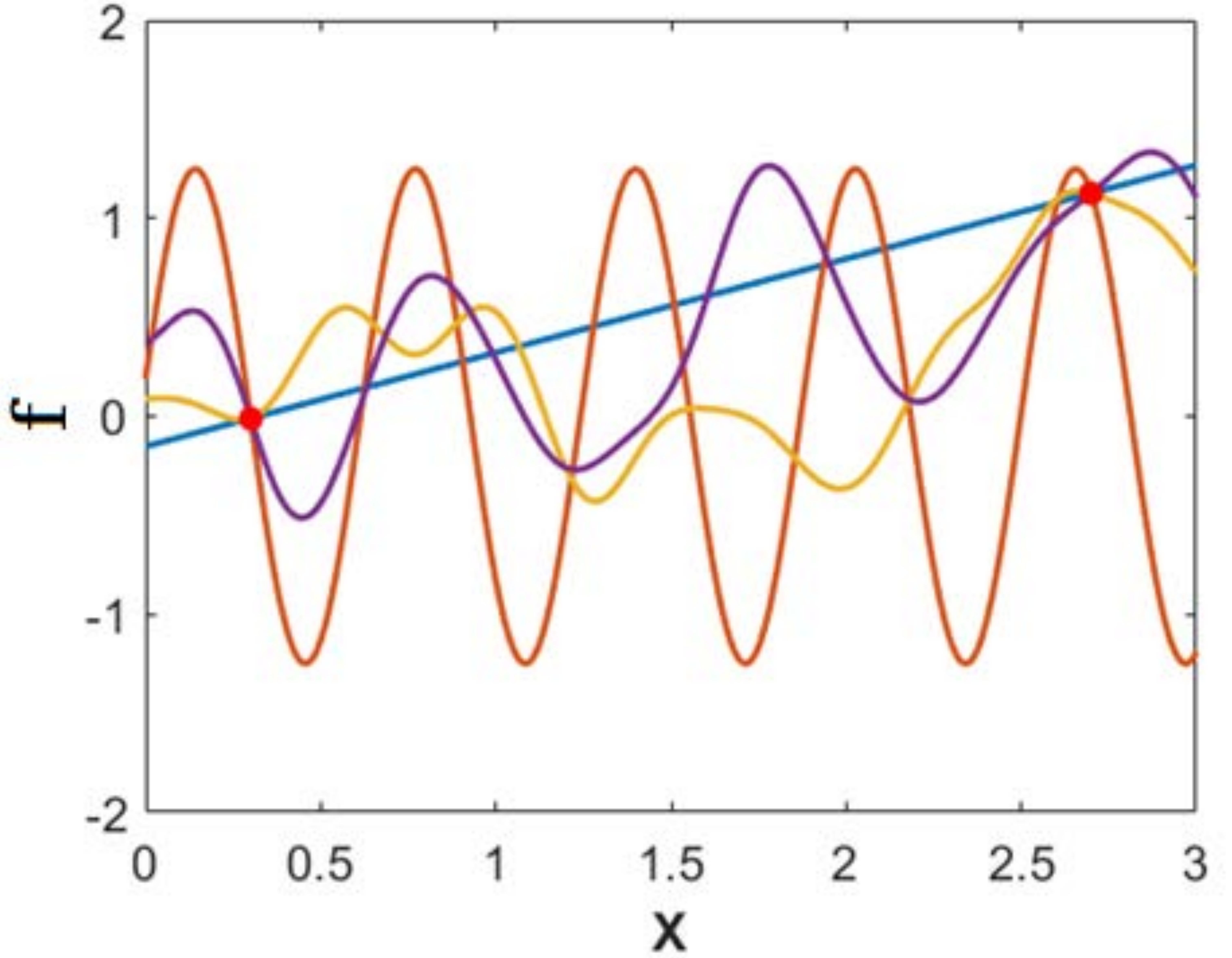
Curse of dimensionality



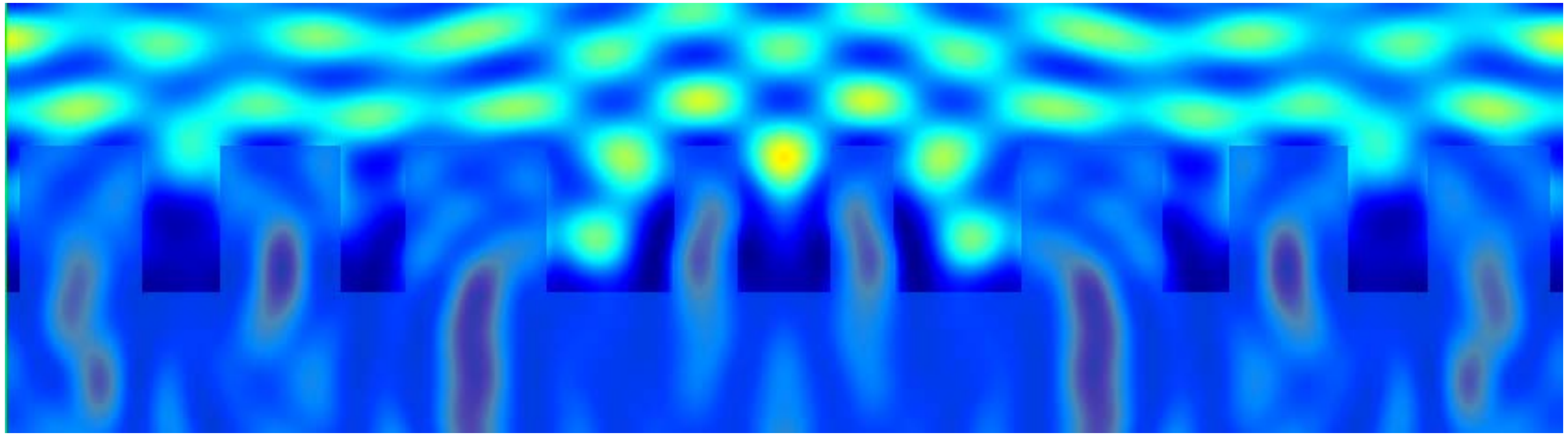
Curse of dimensionality



Curse of dimensionality

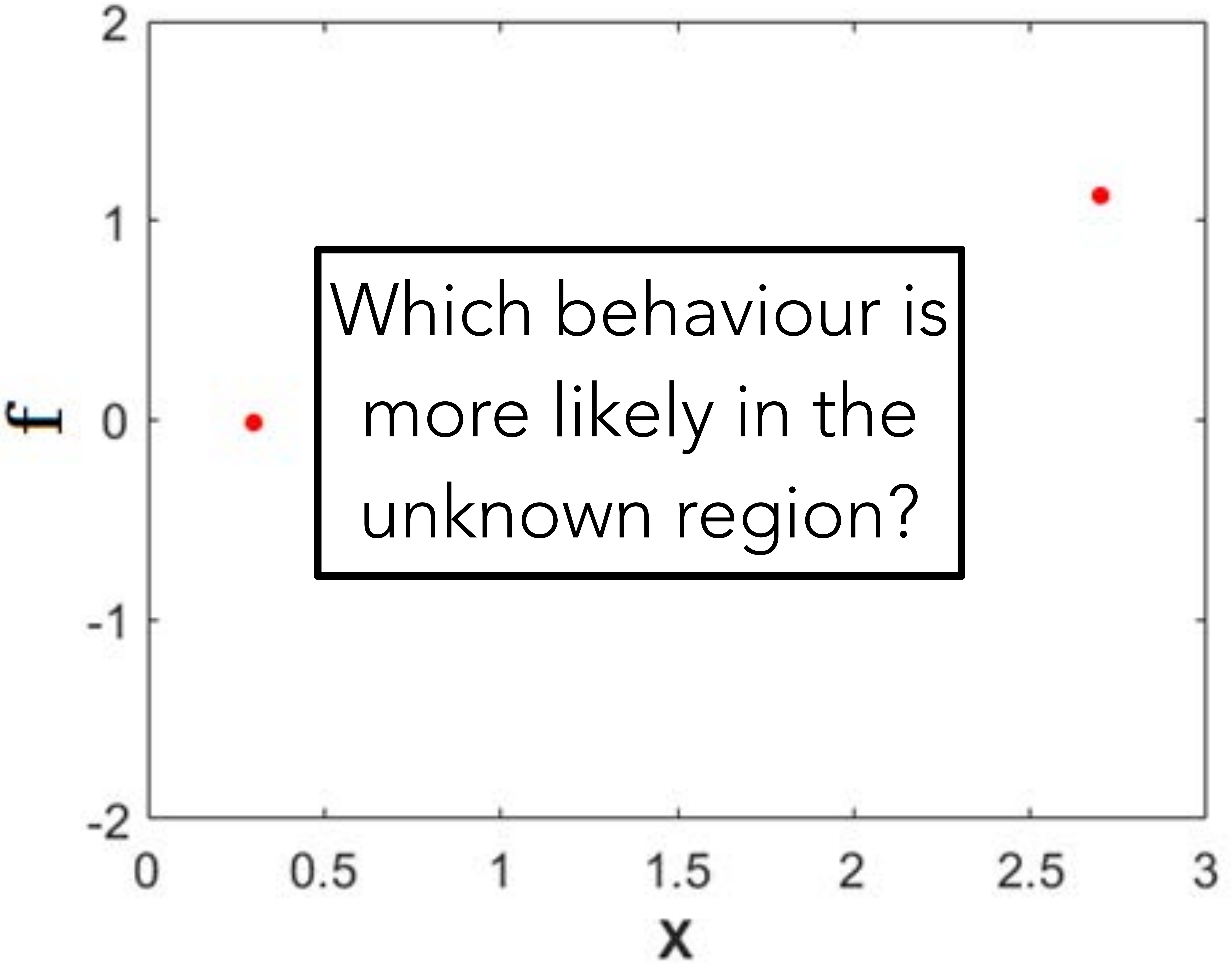


Curse of dimensionality



dimensions	Number of evaluations	
	Coarse sampling (2s/d)	Finer sampling (5s/d)
2	4	25
6	64	15,625
12	4,096	244,140,625
20	1,048,576	9.5367 e13

Curse of dimensionality



Which behaviour is more likely in the unknown region?

Gaussian process

- probability density function for a set of points

$$P(\boldsymbol{F}) = \frac{1}{(2\pi)^{\frac{N}{2}} \left|\Sigma\right|^{\frac{1}{2}}} e^{-\frac{1}{2}(\boldsymbol{F}-\boldsymbol{\mu})^T \Sigma^{-1} (\boldsymbol{F}-\boldsymbol{\mu})}$$

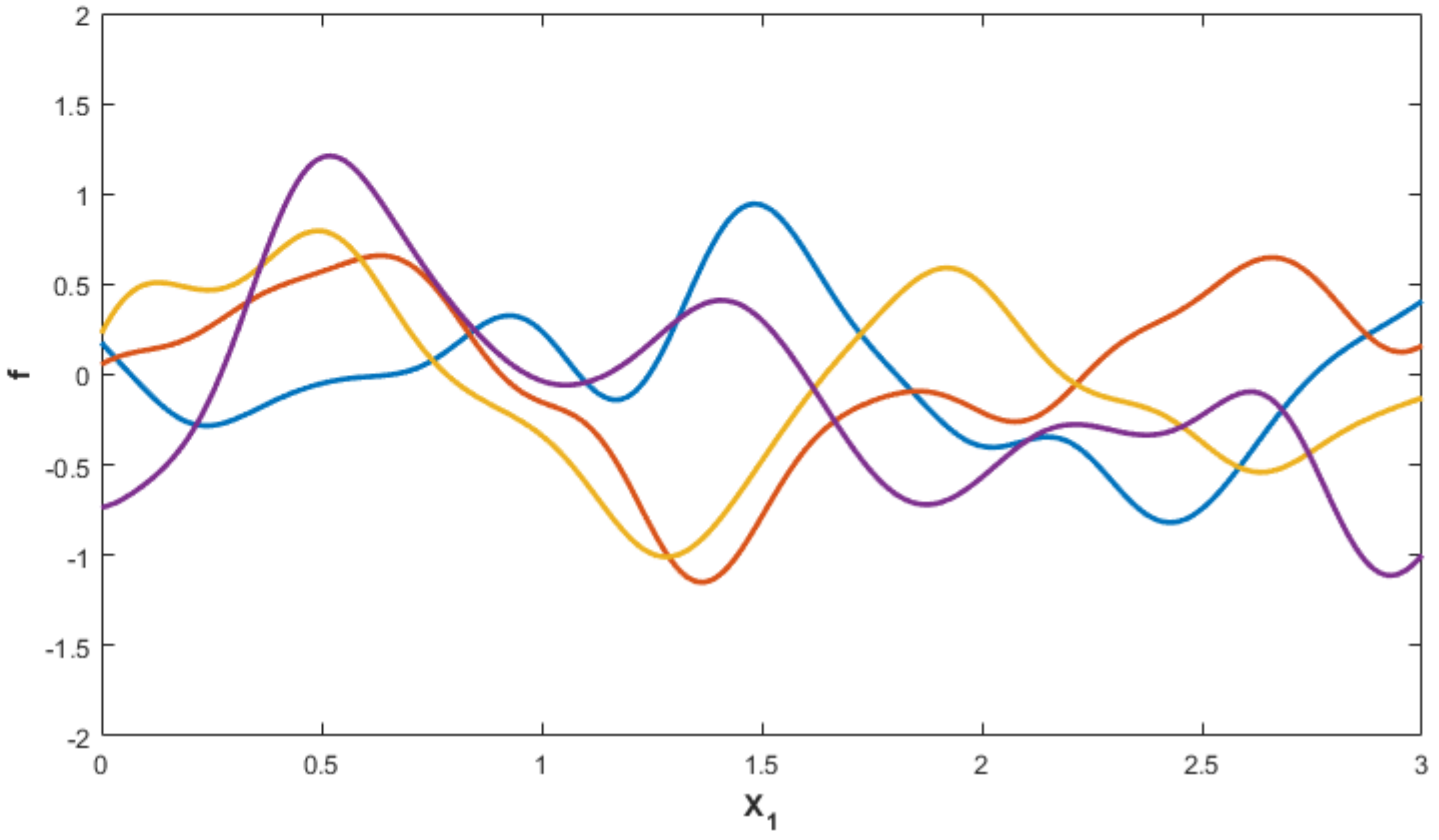
covariance function: $\mathrm{k}(\mathrm{x}, \mathrm{x}') = \sigma^2 \, e^{-\frac{(x-x')^2}{l^2}}$ $\quad \Sigma = \left[k\left(\boldsymbol{x_i}, \boldsymbol{x_j}\right)\right]_{i,j}$

vector of mean values: $\boldsymbol{\mu} = [\mu(\boldsymbol{x_1}), \ldots, \mu(\boldsymbol{x_N})]$

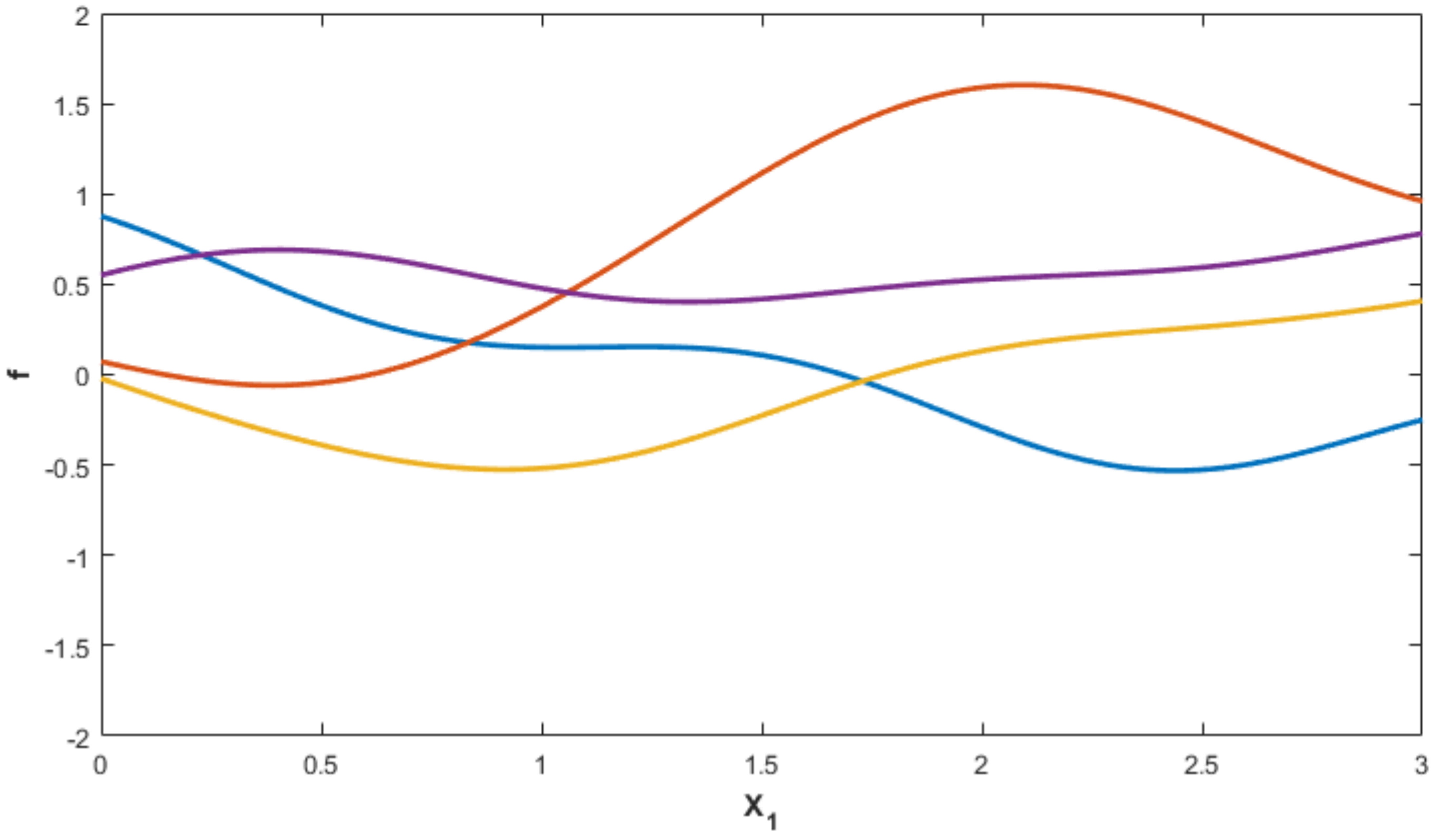
probability distribution of function $\mathbf{F} = [f_1, \ldots, f_N]$

Gaussian process

$l = 0.2$

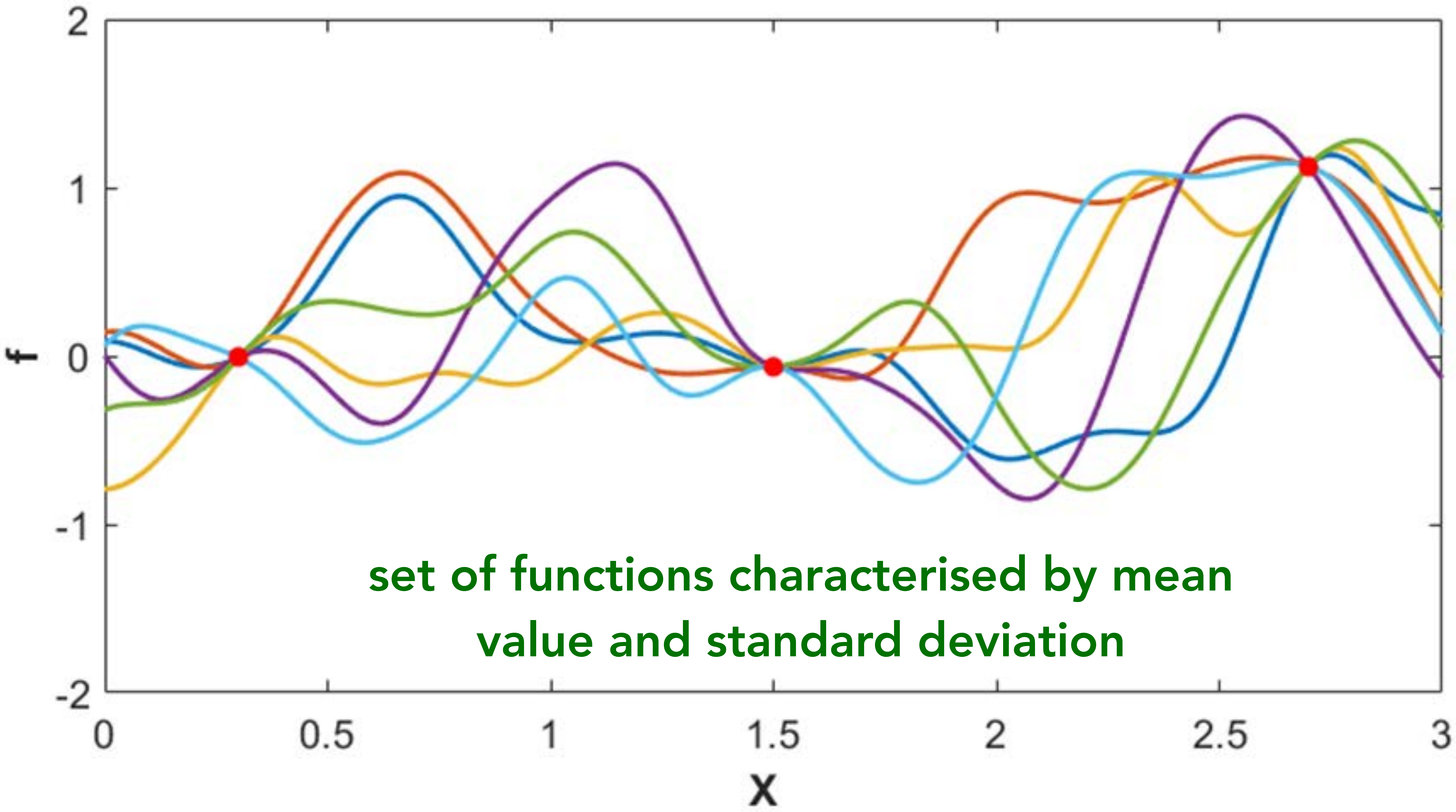


$l = 0.8$



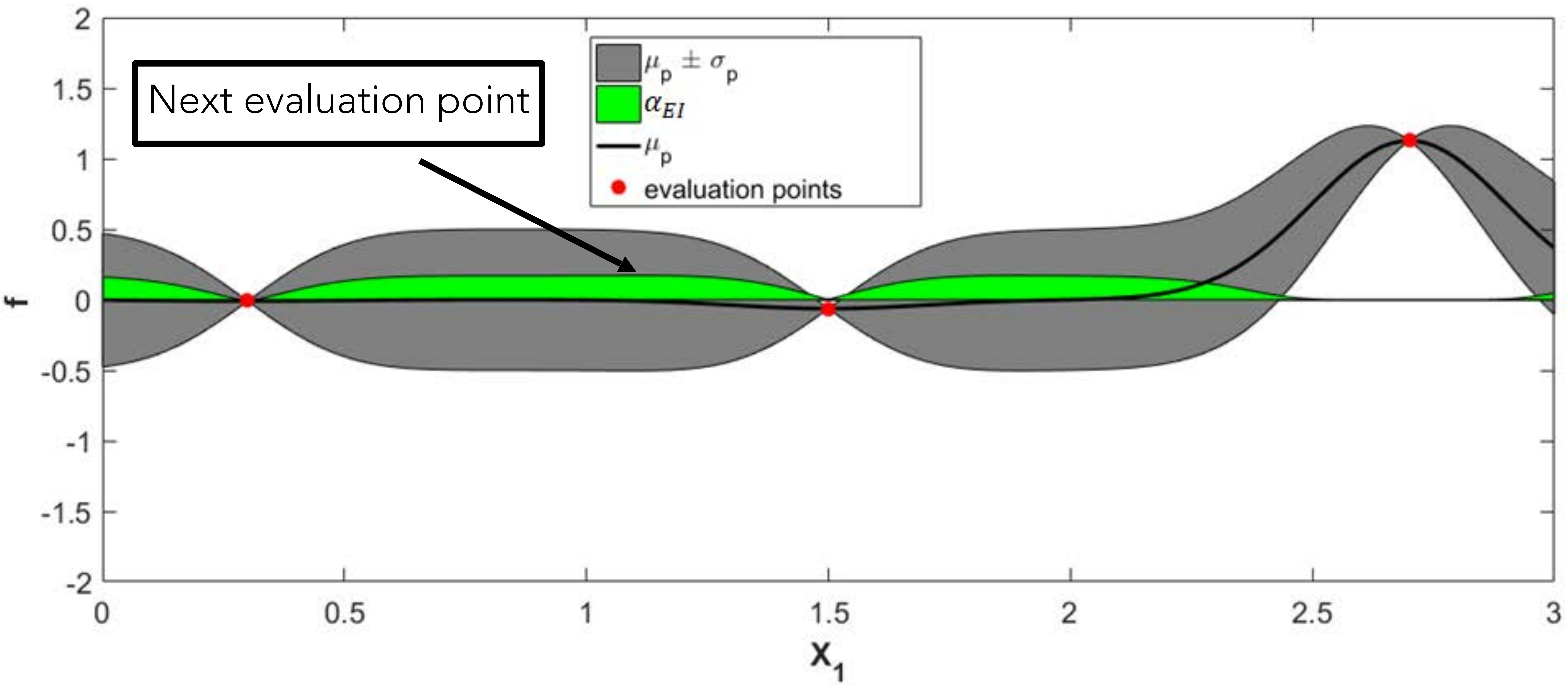
different functions samples from a GP modelled with a Gaussian covariance function

Gaussian process



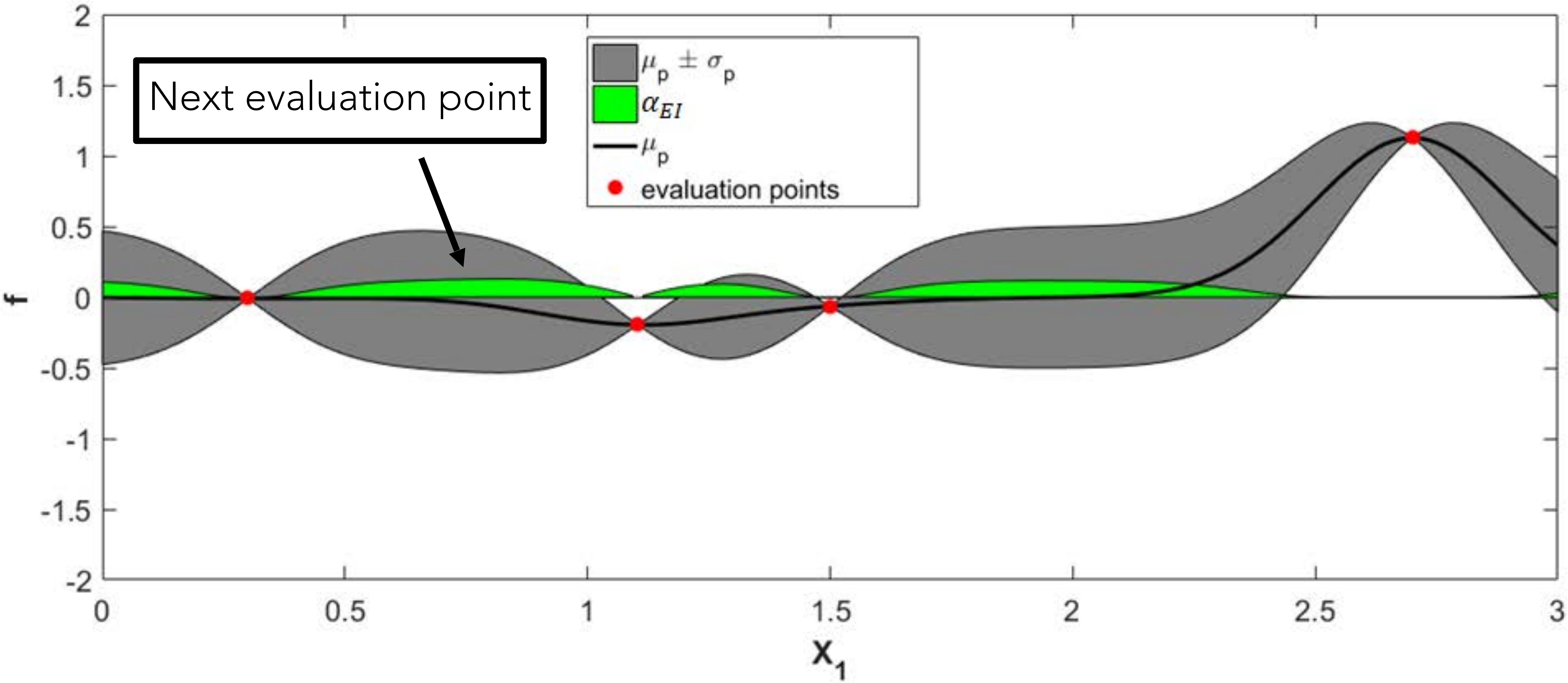
Bayesian optimisation using Gaussian process

acquisition function (expected improvement) $\alpha_{EI} = \mathbb{E}\left[\max\left(0, f_{min} - f(x)\right)\right]$



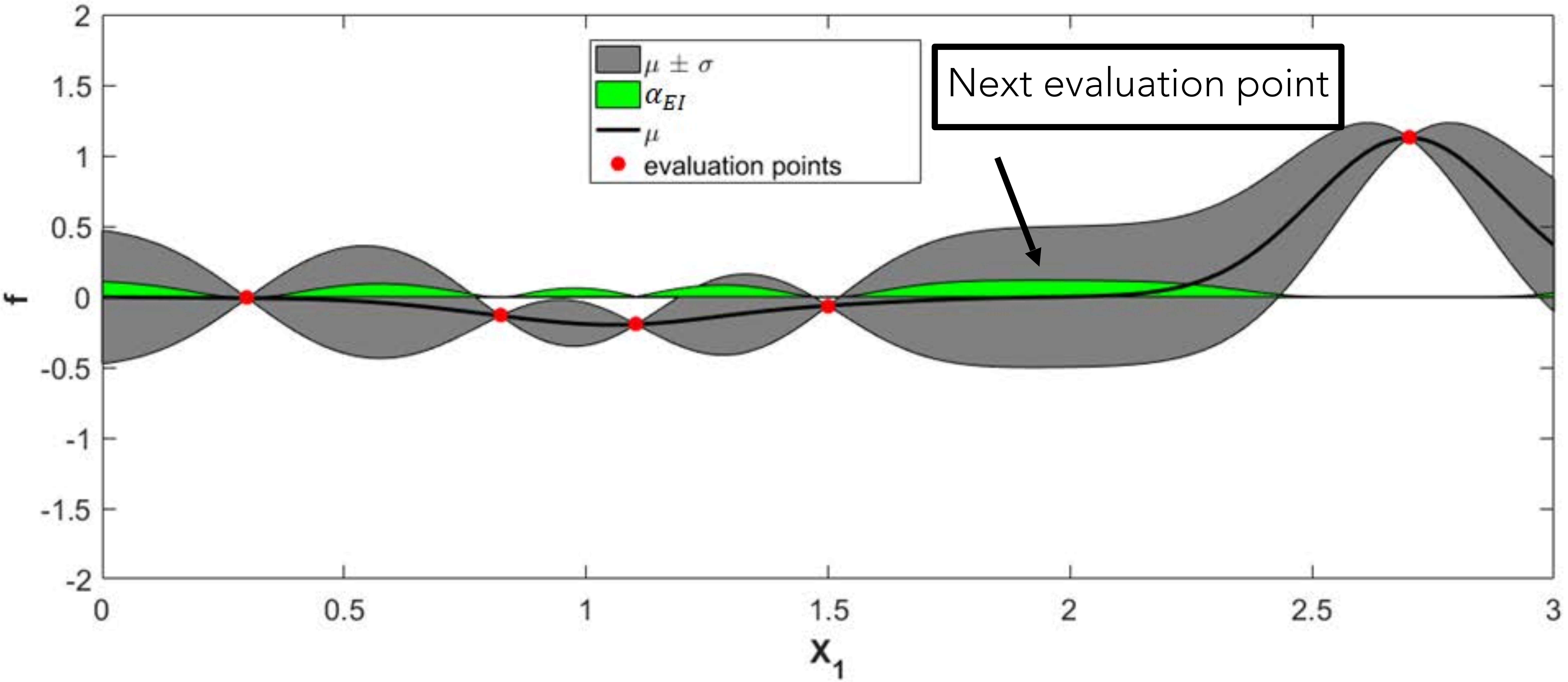
Bayesian optimisation using Gaussian process

acquisition function (expected improvement) $\alpha_{EI} = \mathbb{E}\left[\max\left(0, f_{min} - f(x)\right)\right]$



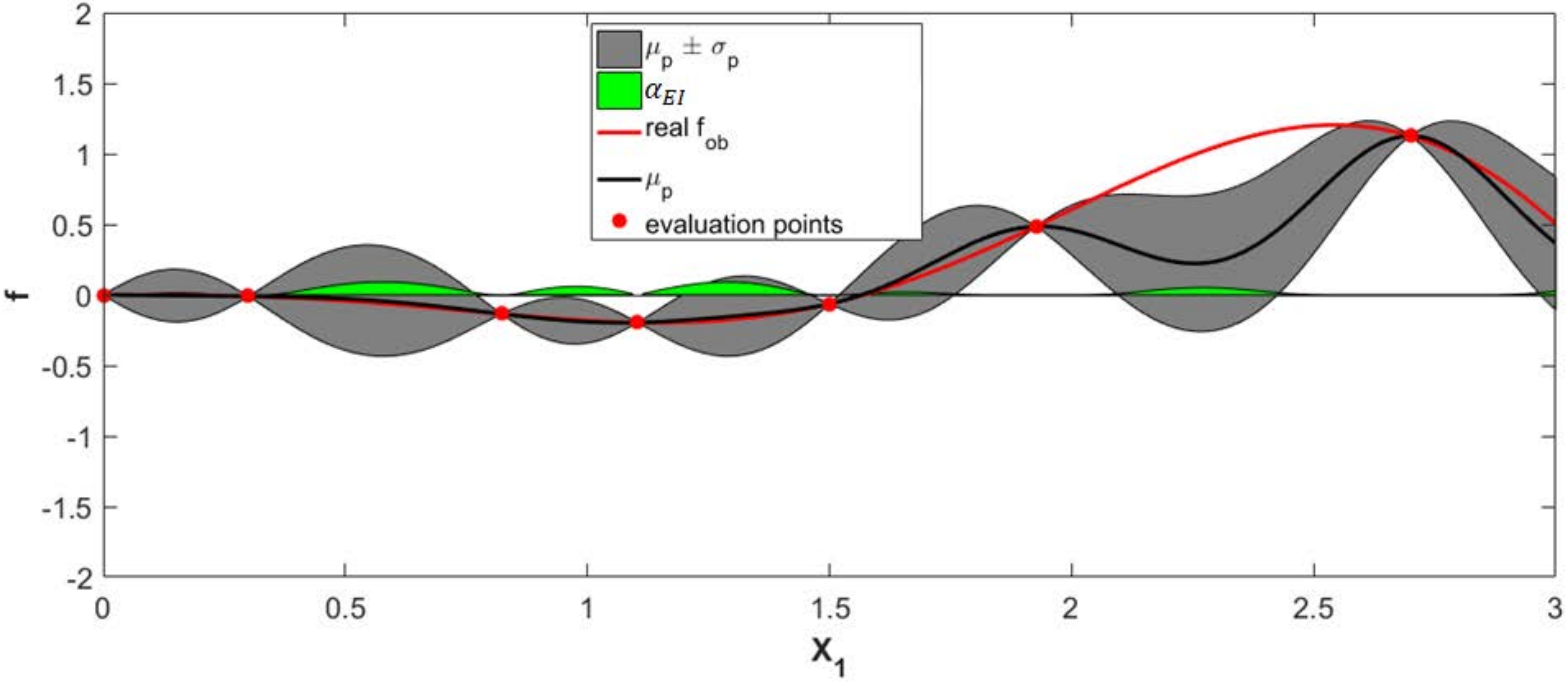
Bayesian optimisation using Gaussian process

acquisition function (expected improvement) $\alpha_{EI} = \mathbb{E}\left[\max\left(0, f_{min} - f(x)\right)\right]$

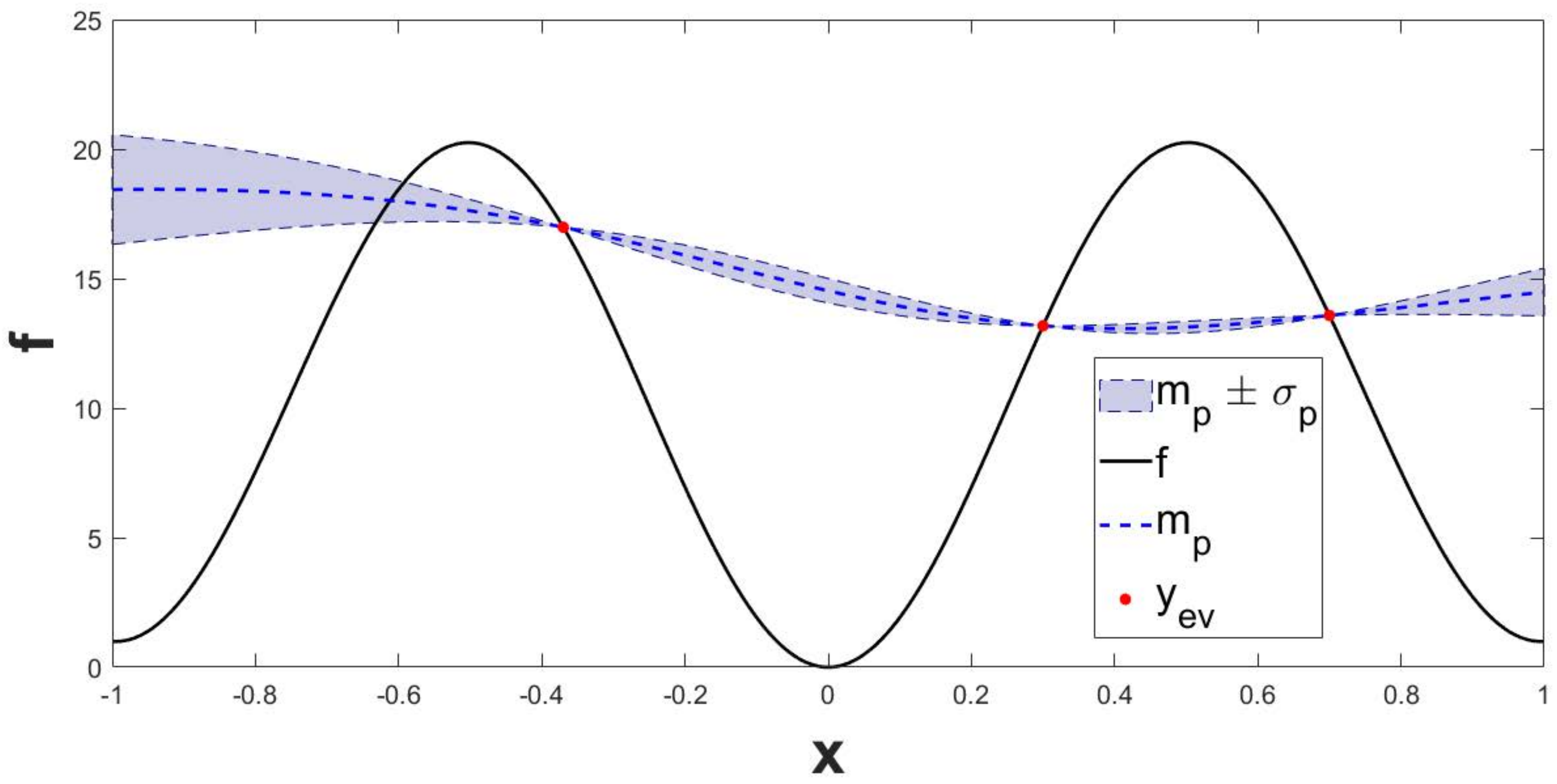


Bayesian optimisation using Gaussian process

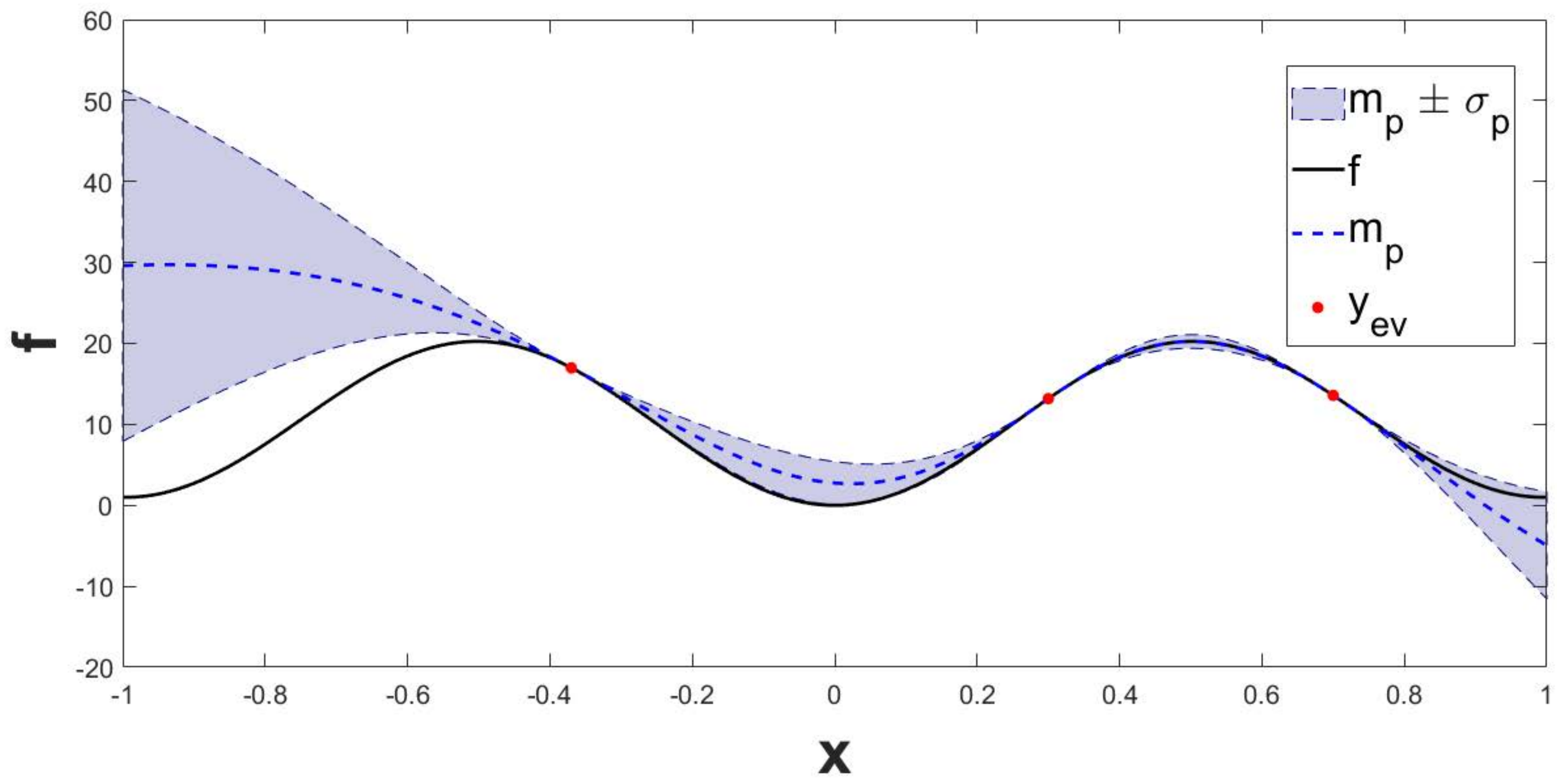
acquisition function (expected improvement) $\alpha_{EI} = \mathbb{E}\left[\max\left(0, f_{min} - f(x)\right)\right]$



Adding gradient information helps!



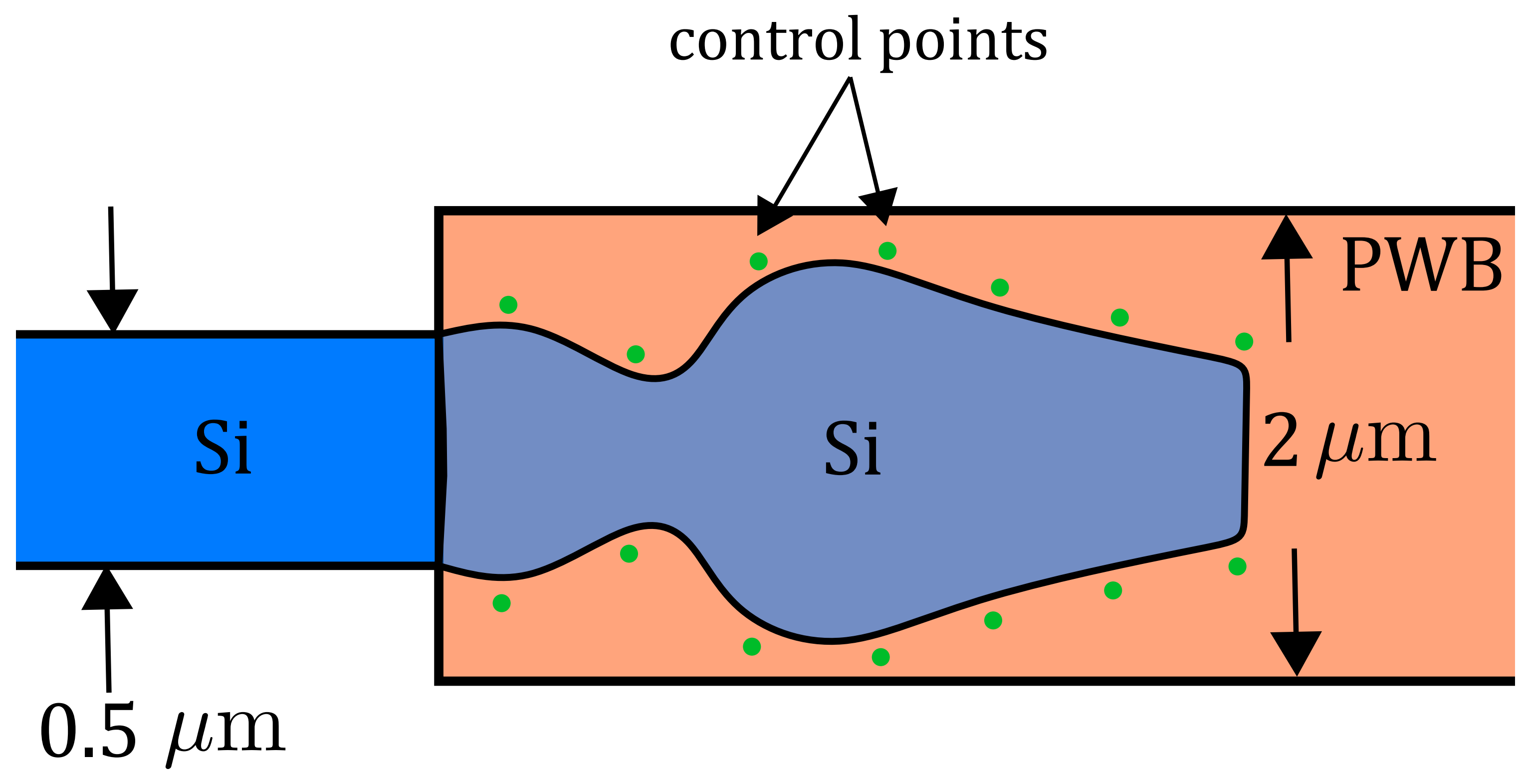
no gradient information

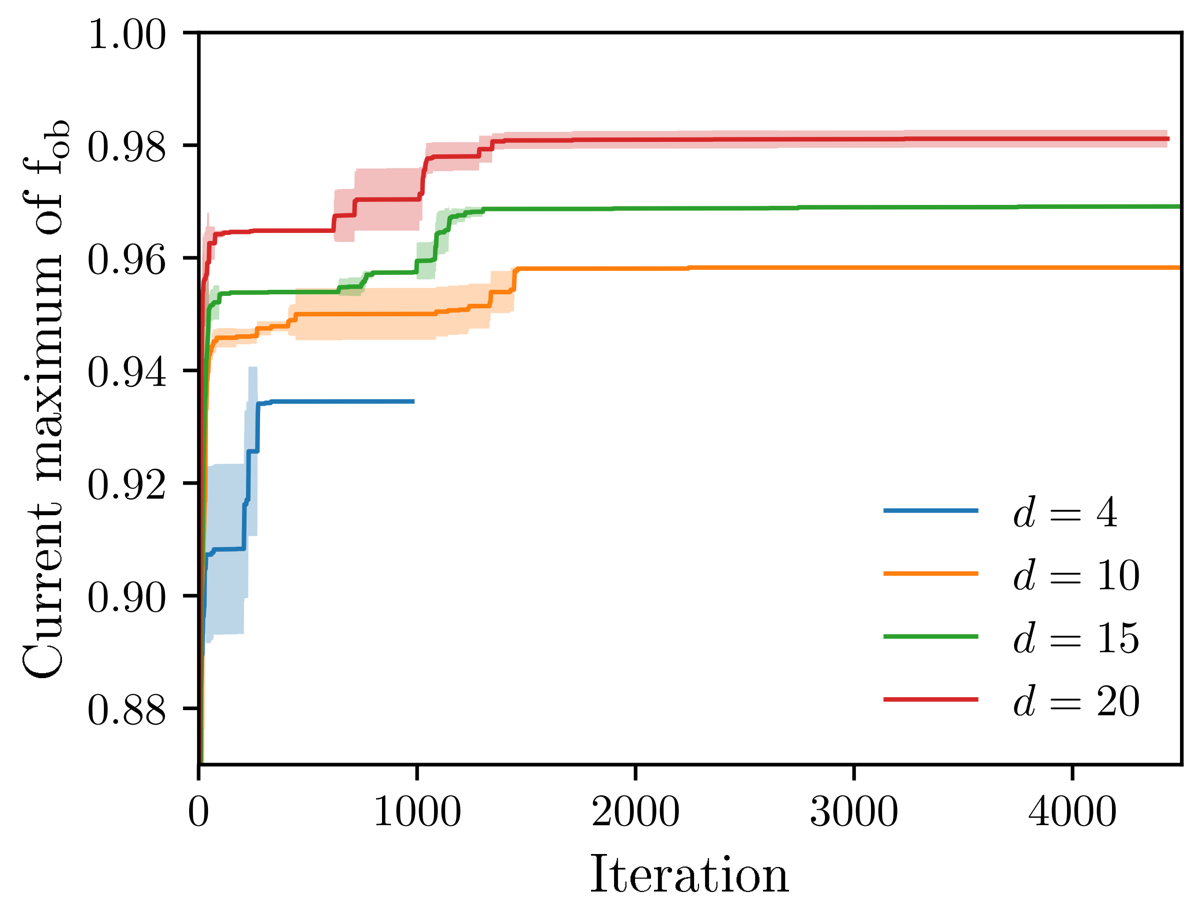


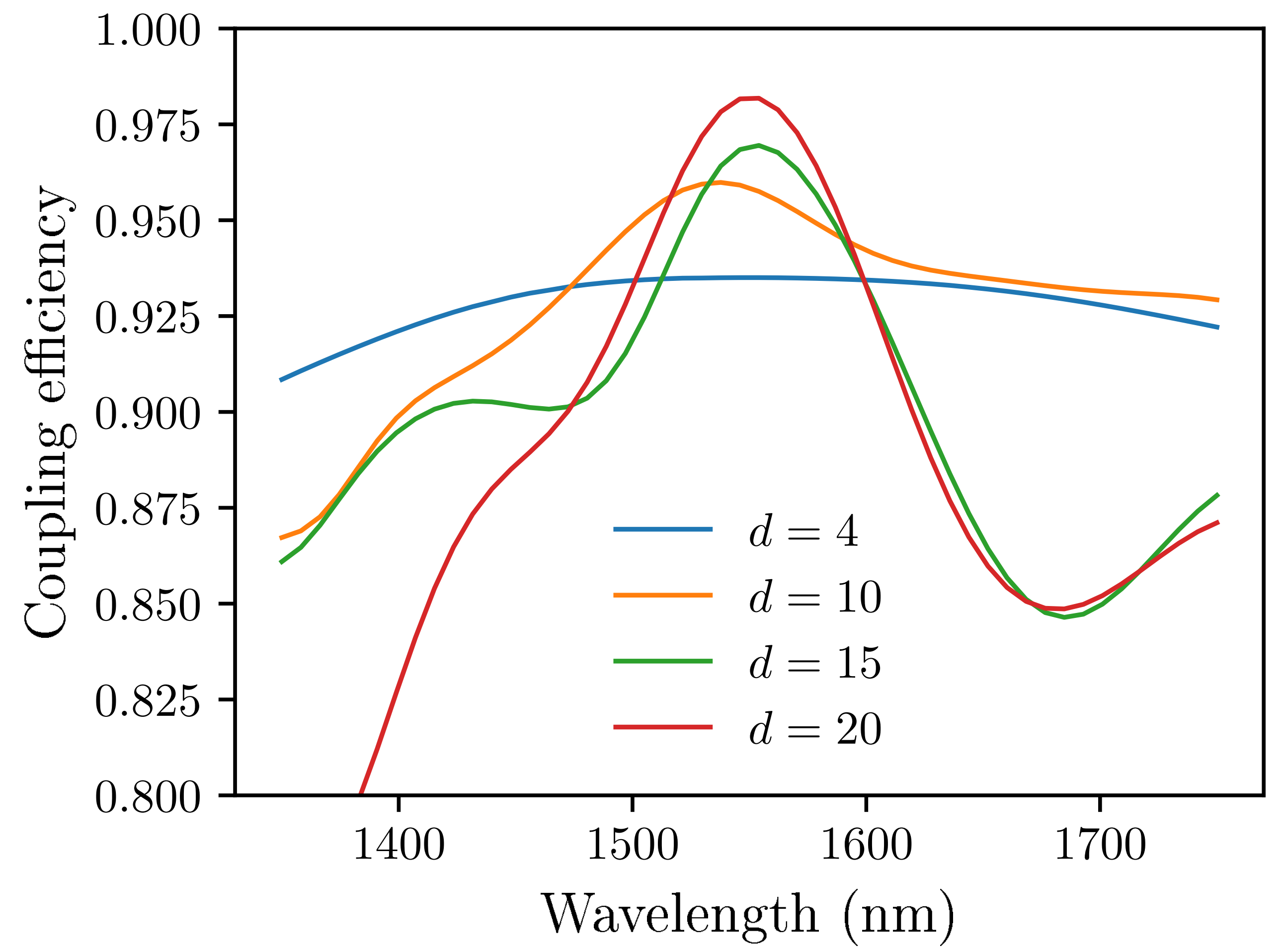
gradient information

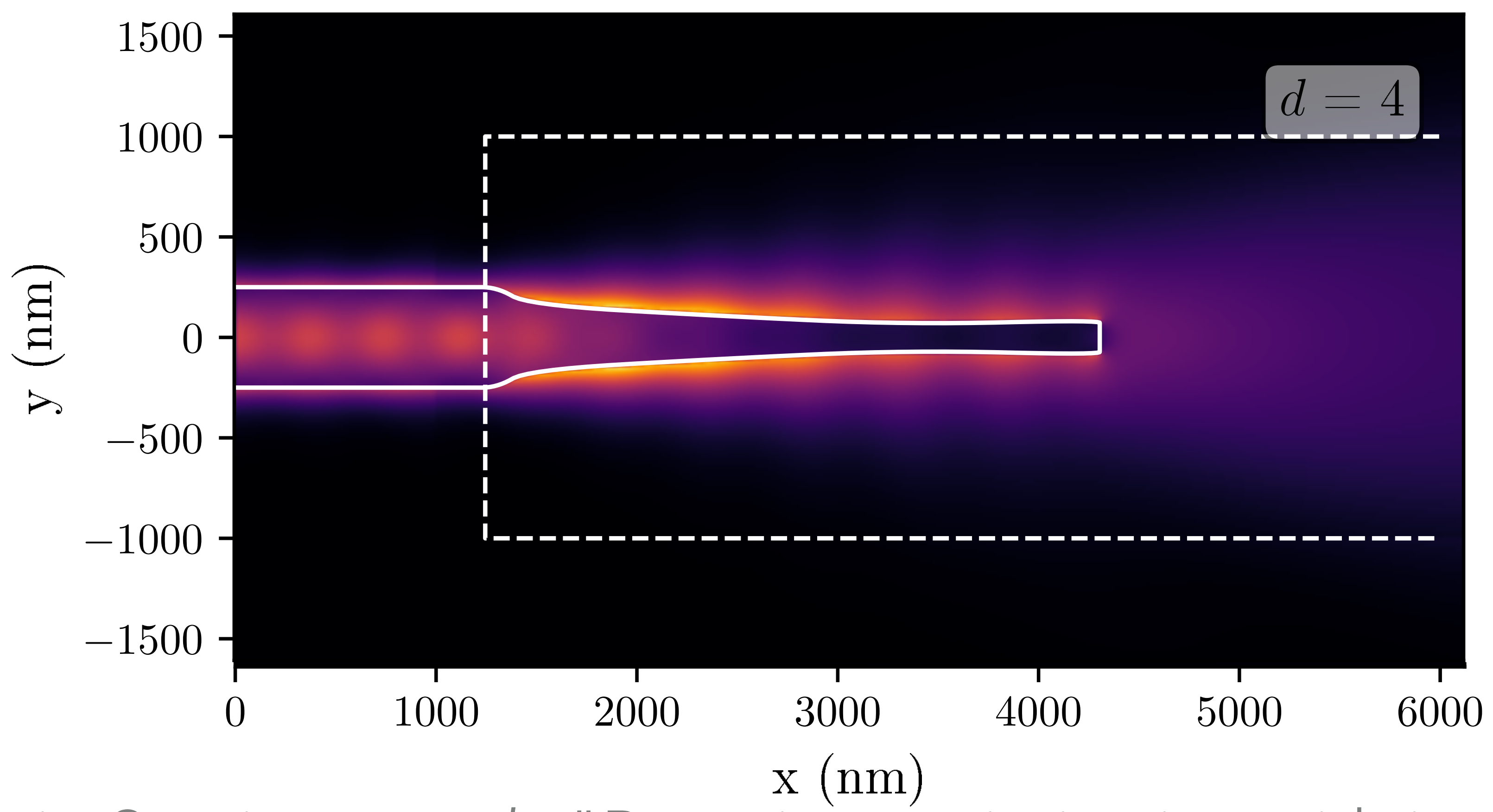
Bayesian optimization

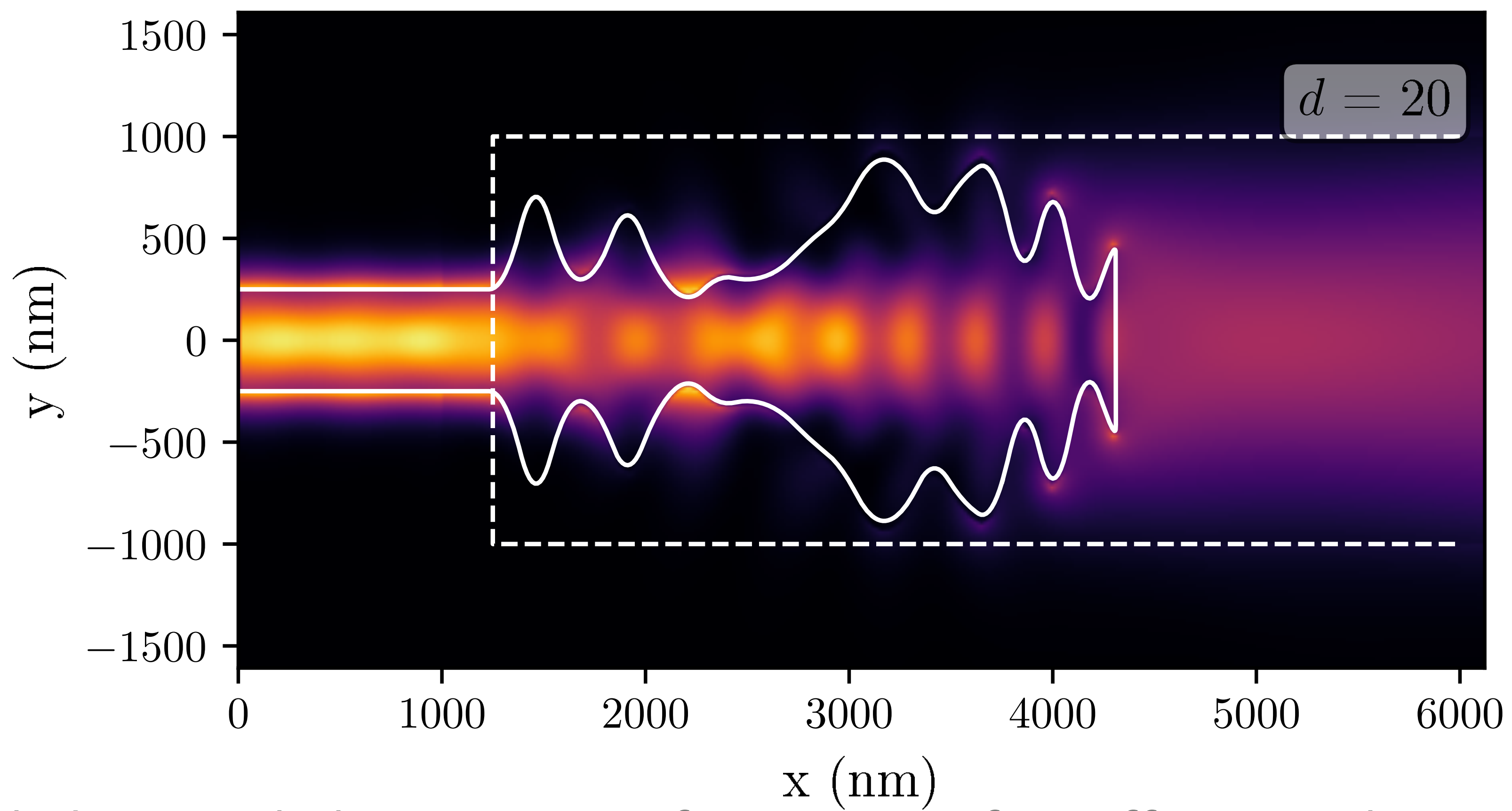
- photonic wire bonds are freeform dielectric waveguides written with 3D direct laser lithography
- part of optical chips that consist of different materials capitalising on strength of different materials
- major of source of losses are incomplete coupling sites to the chips as well as curved trajectory







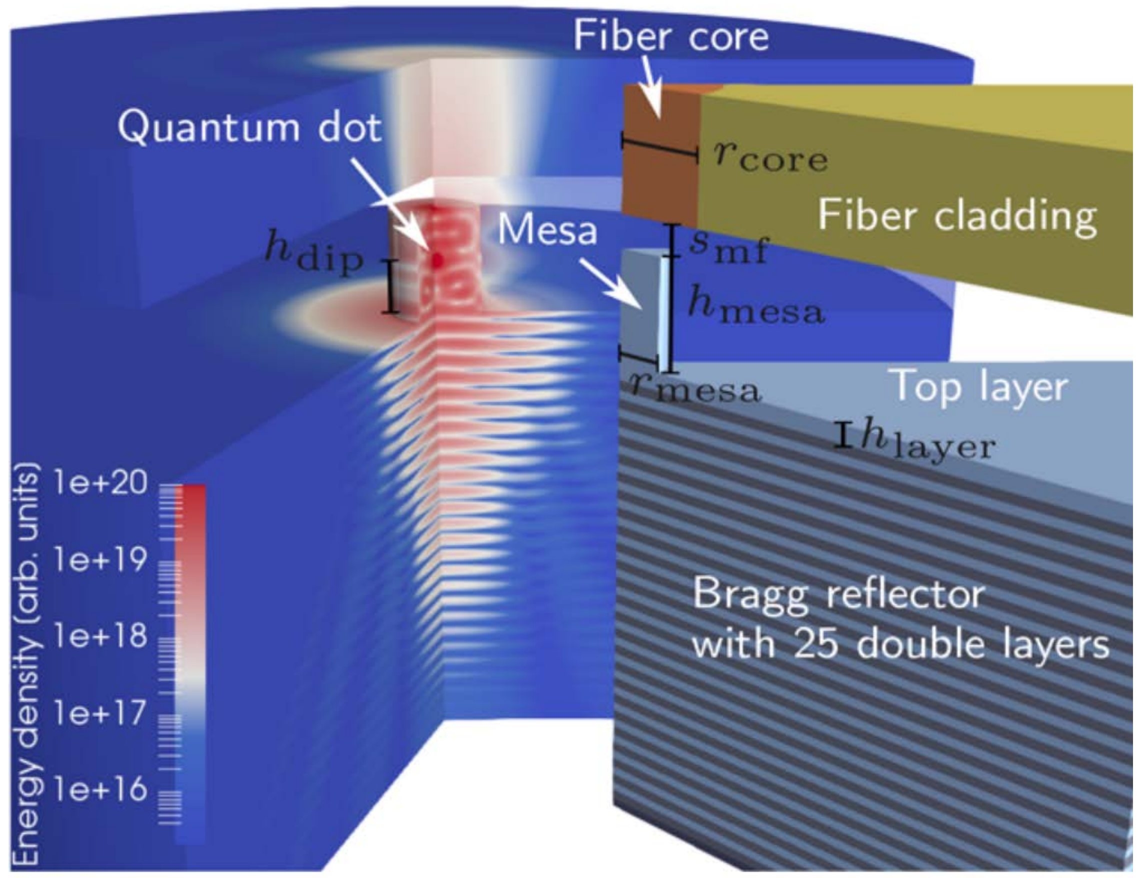


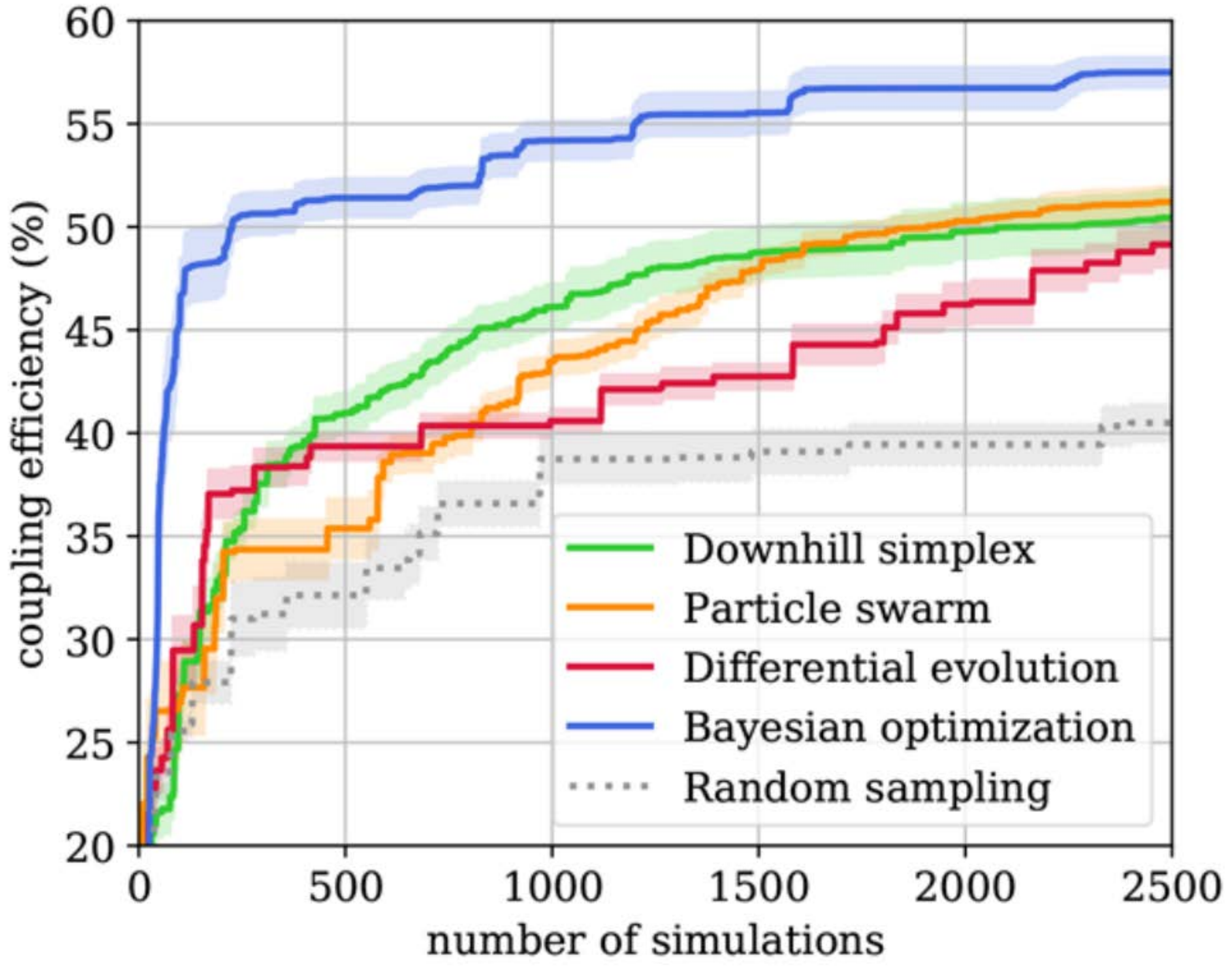


X. Garcia-Santiago *et al.*, "Bayesian optimization with improved scalability and derivative information for efficient design of nanophotonic structures", IEEE Journal of Lightwave Technology **39**, 167 (2021)

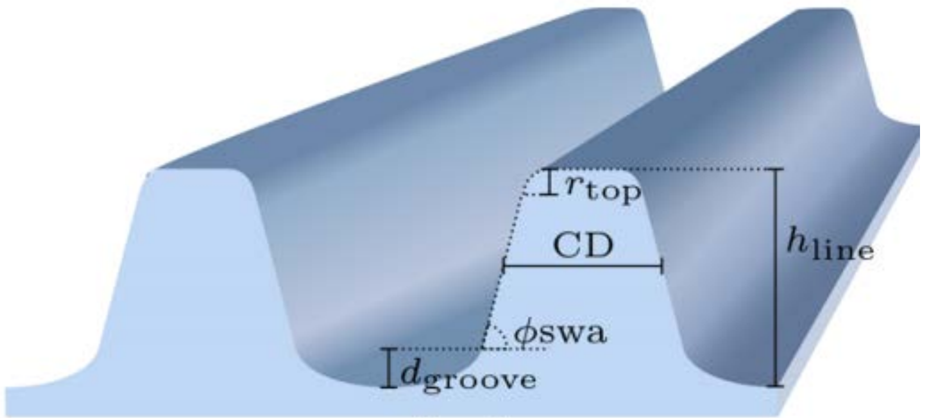
Selected further applications for Bayesian optimization

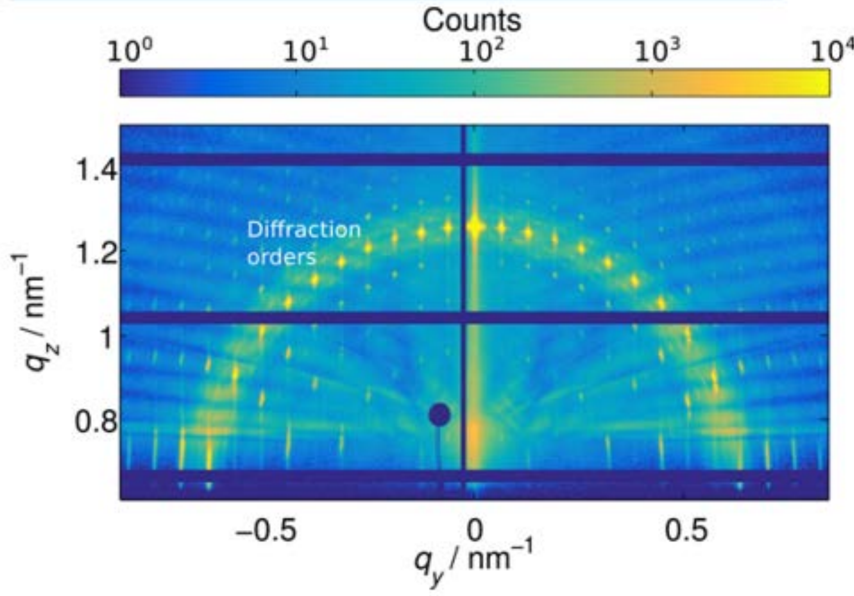
quantum dot coupler

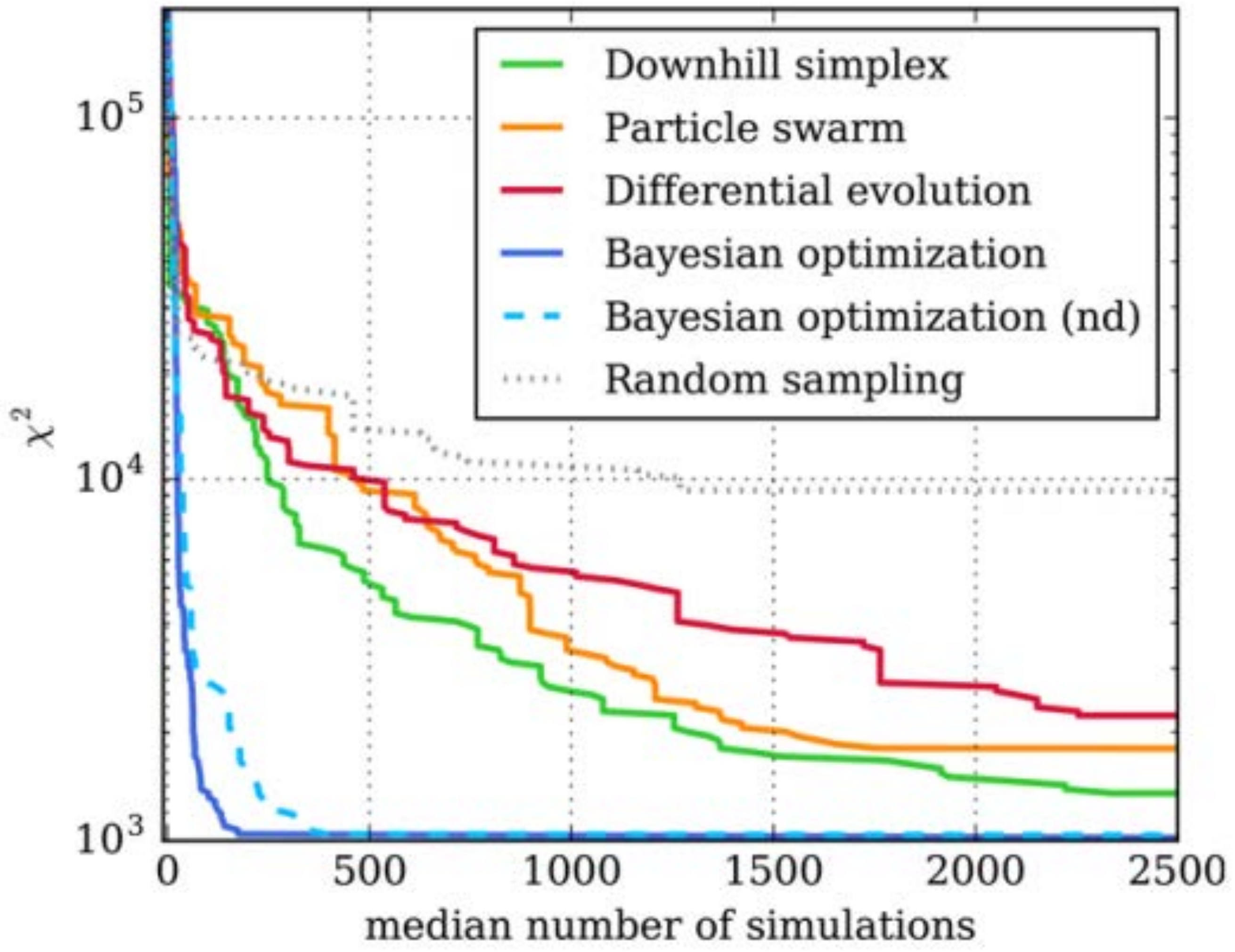




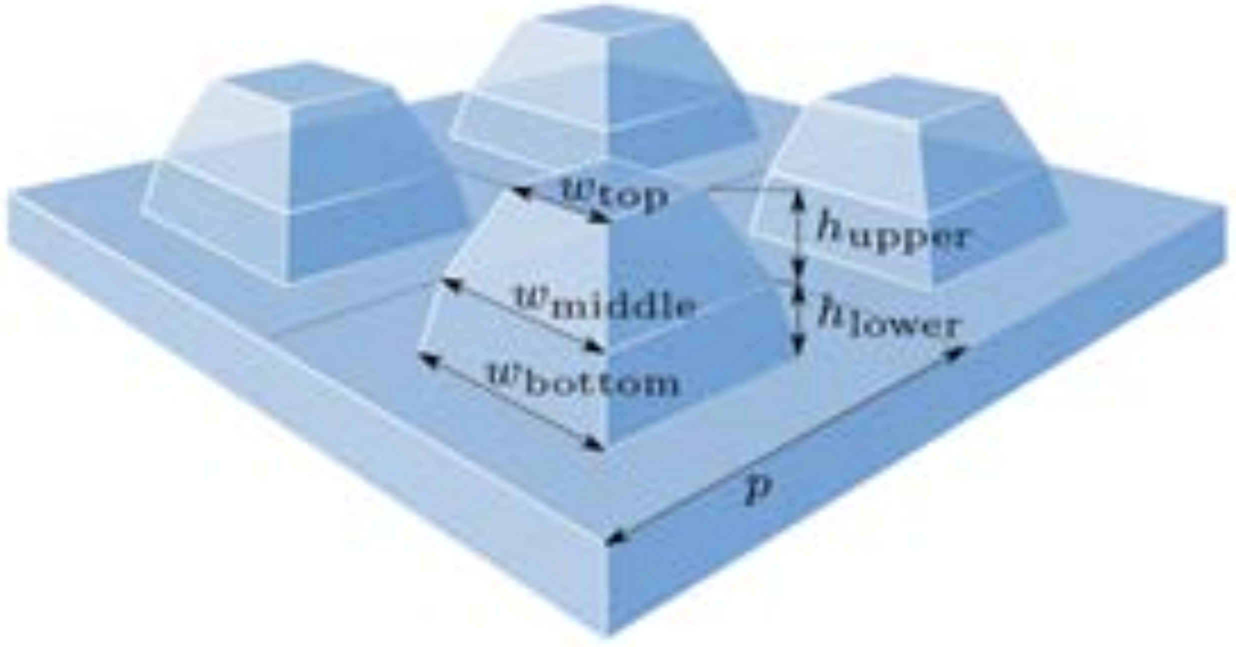
parameter reconstruction

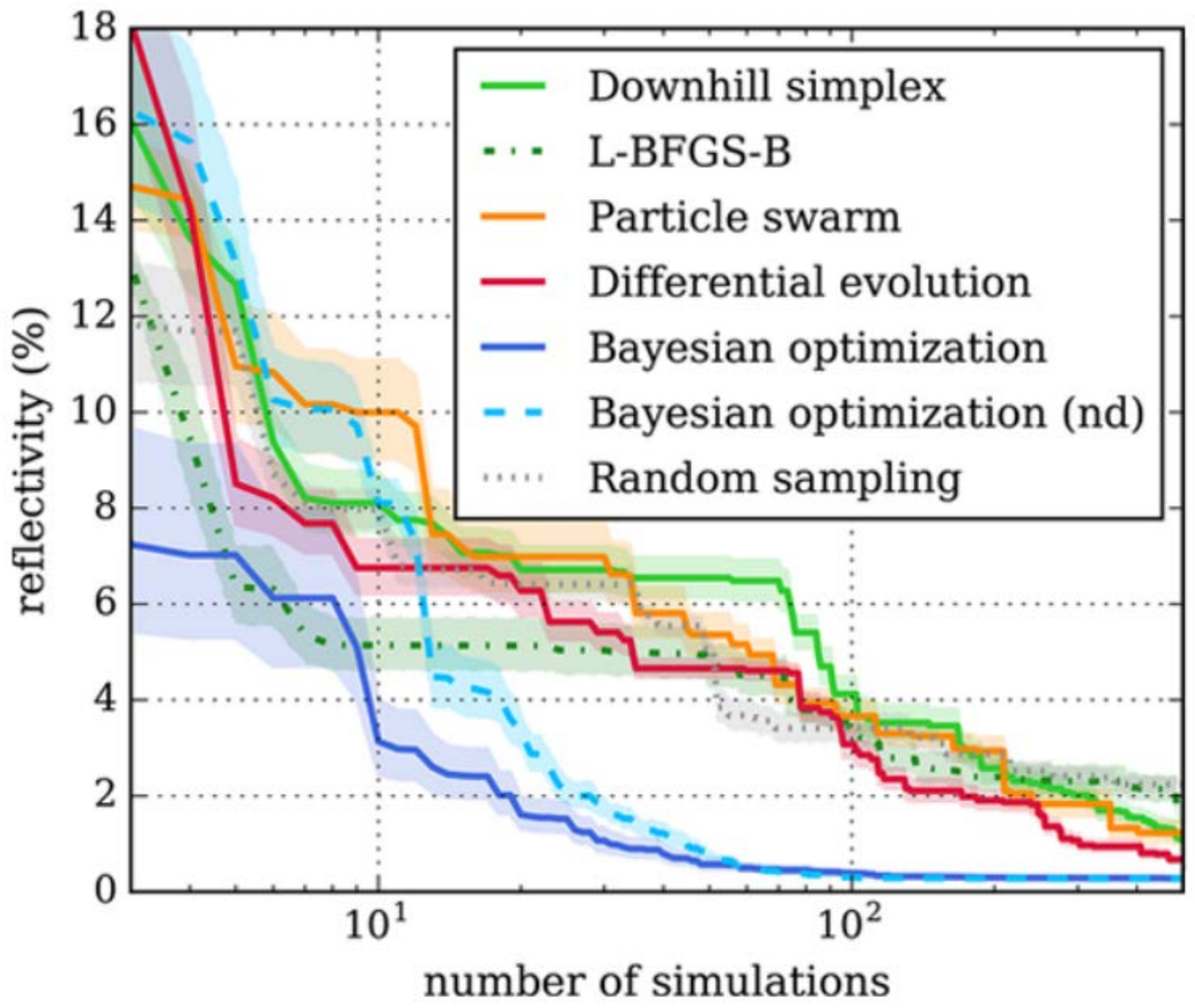






antireflective grating





P.-I. Schneider *et al.*, "Benchmarking five global optimization approaches for nano-optical shape optimization and parameter reconstruction", ACS Photonics, **6**, 2726 (2019)

Computational Photonics

Solving inverse problems