OPTIMIZATION ALGORITHMS · MASTER'S COURSE

Surrogate Models, Multi-Fidelity and Ensemble-Variational Optimization

Lecture 5 — Optimizing expensive flow simulations when gradients are hard to get

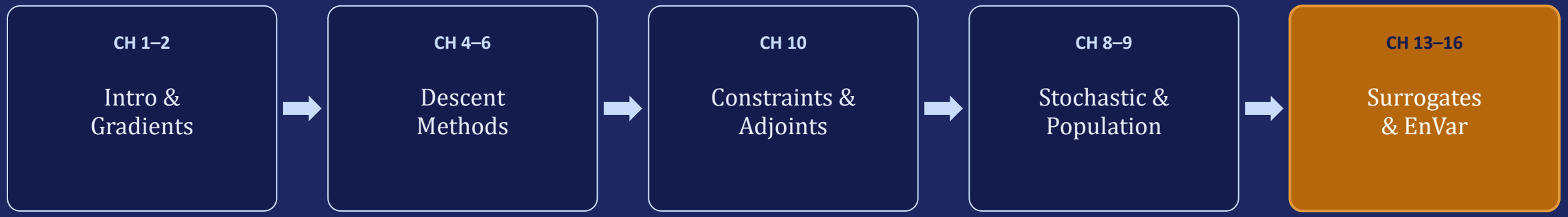
- Why surrogates? Multi-fidelity
- Response surfaces: RBF, Kriging, Bayesian optimization
- Ensemble-variational methods

Based on Kochenderfer & Wheeler, Algorithms for Optimization (MIT Press), Chapters 13–16, and on research lecture material on ensemble-variational optimization

WHERE WE ARE IN THE COURSE

Course Roadmap

Four lectures gave us the building blocks. Today we put them to work on expensive problems: model the objective cheaply, or approximate its gradient from samples.



Today's lecture

Part B draws on Chapters 13–16 of the textbook (sampling plans, surrogate models, probabilistic surrogates, surrogate optimization); Part C is research material.

PART A · MOTIVATION

Why Surrogates?

Gradient-based versus derivative-free — and what multi-fidelity adds

The Problem and the Two Families of Methods

$$\text{minimize}_g \ \mathcal{J}(q,g) \qquad \text{subject to } \frac{dq}{dt} = f(q,g,t), \ \ 0 \le t \le T$$

q: state variable (the flow). g: control or design variable. Every evaluation of $\mathcal{J}$ requires solving the flow equations — one simulation.

Descent-direction methods

Use gradients (first order) or Hessians (second order): conjugate gradient, quasi-Newton, … Lectures 2–3.

Few function evaluations — but need derivatives.

Derivative-free methods

Stochastic and population methods: annealing, CMA-ES, genetic algorithms, PSO, … Lecture 4.

No derivatives — but many function evaluations.

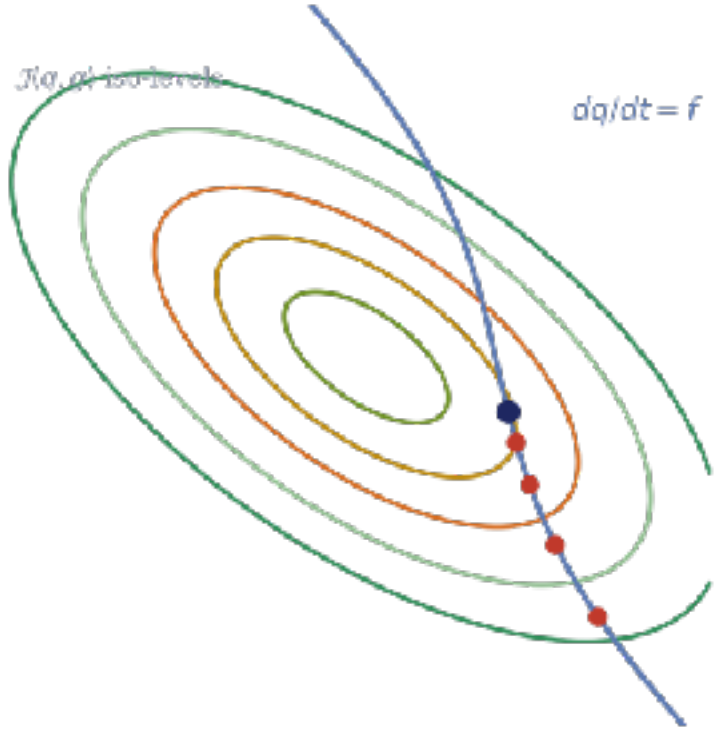
Lagrangian, Adjoint and the Gradient

Enforce the flow equations with a Lagrange multiplier (the adjoint variable). At the optimum, the state manifold is tangent to an iso-level of the objective.

$$\mathcal{L}(q, g, \lambda) = \mathcal{J}(q, g) - \int_0^T \lambda^* \cdot \left(\frac{dq}{dt} - f(q, g, t) \right) dt$$

Setting every variation to zero gives three coupled problems:

$\frac{\partial \mathcal{L}}{\partial \lambda} = 0$	1 Forward problem	the equations governing the flow
$\frac{\partial \mathcal{L}}{\partial q} = 0$	2 Adjoint problem	solved backward in time
$\frac{\partial \mathcal{L}}{\partial g} = 0$	3 Optimality condition	the desired gradient



One adjoint solve gives the gradient with respect to all parameters: highly efficient, with an affordable number of function evaluations.

Where the Adjoint Approach Struggles

Algorithmic challenges

Unsteady problems

The adjoint runs backward in time and needs the whole forward trajectory → checkpointing algorithms.

Time to solution

Long unsteady simulations are sequential → parallel-in-time approaches.

Systematic challenges

Turbulence and chaos

Adjoint gradients grow without bound over long horizons → shadowing approaches.

Differentiability

Non-smooth models and solvers → incomplete differentiation.

Multiple local optima

A gradient method converges to a nearby optimum only.

Extracting the adjoint

Deriving and coding the adjoint equations is intrusive and laborious.

Net result: gradient-based methods are efficient and robust, but local and intrusive.

Robust, Global — and Expensive

1 Start at an initial point → 2 Stochastically explore the domain → 3 Select a new promising point → 4 Repeat until at an optimum

The disadvantage

Too many function evaluations. Each evaluation is a full CFD simulation, so the cost of the optimization is (number of evaluations) × (cost of one simulation).

Illustrative arithmetic

1,000 evaluations × 20 CPU-hours per unsteady simulation = 20,000 CPU-hours for one optimization run — before any repeat for robustness. (Round numbers chosen to illustrate the scaling, not measurements.)

The way out: meta-models — replace the expensive function evaluation with a low-cost surrogate.

METHOD COMPARISON

Gradient-Based versus Derivative-Free

	Gradient-based (adjoint)	Derivative-free (stochastic, population)
Scope	Local: converges to a nearby optimum	Global: can escape local optima and find the global minimum
Cost	Efficient: affordable number of function evaluations	Large number of function evaluations
Implementation	Intrusive: adjoint equations must be derived and coded	Non-intrusive: the simulation is a black box
Turbulent / chaotic flows	Gradients may be inaccurate or unbounded	Applicable to complex PDEs and non-deterministic problems
Robustness	Efficiency and robustness when it applies	Robust, even for non-differentiable objectives

Can we combine the two, to keep the advantages of both? — Part B reduces the cost of derivative-free search; Part C approximates the gradient from samples.

Replace the Expensive Evaluation

A surrogate (meta-model, response surface) is a cheap statistical or numerical model of the objective, built from a limited number of expensive evaluations and used in their place for the search.

Kriging / co-Kriging

Gaussian-process models that also quantify their own uncertainty.

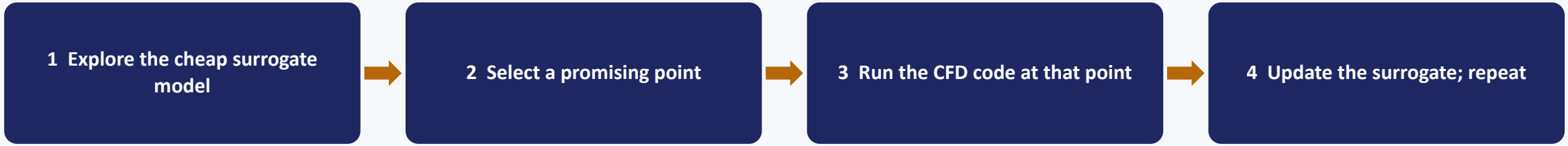
RBF / gradient-enhanced RBF

Interpolation with translates of a radial kernel; gradients can be included.

Neural-network regression

Flexible regression for large data sets and many parameters.

The multi-fidelity optimization loop



What Does “Fidelity” Mean?

Fidelity is how faithfully a model represents the quantity we care about. Most simulation chains offer several levels at very different costs.

Lever	Low fidelity (cheap, less accurate)	High fidelity (expensive, accurate)
Physical model	RANS, empirical correlations	LES, DNS, experiments
Mesh and time step	Coarse grid, large time step	Fine grid, small time step
Dimensionality	2D, axisymmetric, periodic	Full 3D geometry
Simulation length	Short averaging window	Long statistically converged run
Model order	Reduced-order or surrogate model	Full-order solver

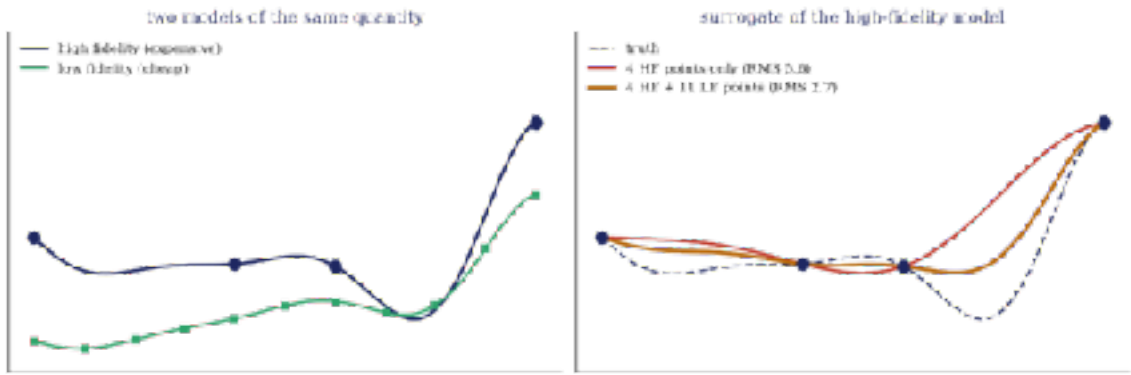
Multi-fidelity optimization uses many cheap runs to learn the landscape and few expensive runs to anchor it.

Combining Fidelities: A Discrepancy Model

Autoregressive co-Kriging (Kennedy & O'Hagan) models the expensive function as a scaled cheap one plus a correction:

$$f_{\text{hi}}(\mathbf{x}) = \rho\, f_{\text{lo}}(\mathbf{x}) + \delta(\mathbf{x})$$

The low-fidelity model is fitted with many cheap points; the discrepancy δ and the scale ρ are fitted with the few high-fidelity points.



Same 4 expensive evaluations: RMS error 5.6 → 2.7. The dip near x = 0.75 is still missed because no expensive sample lies near it — hence adaptive infill (Part B).

RECAP

Part A — Key Takeaways

1. Gradient-based (adjoint) methods are efficient and robust, but local, intrusive, and fragile for turbulent or chaotic flows.
2. Derivative-free methods are global and non-intrusive, but need very many expensive simulations.
3. A surrogate model replaces most expensive evaluations; the high-fidelity code is called only at promising points.
4. Multi-fidelity methods combine many cheap models with few expensive ones through a discrepancy correction.
5. Next: how to build good surrogates — sampling, regression, RBF, Gaussian processes — and how to optimize with them.

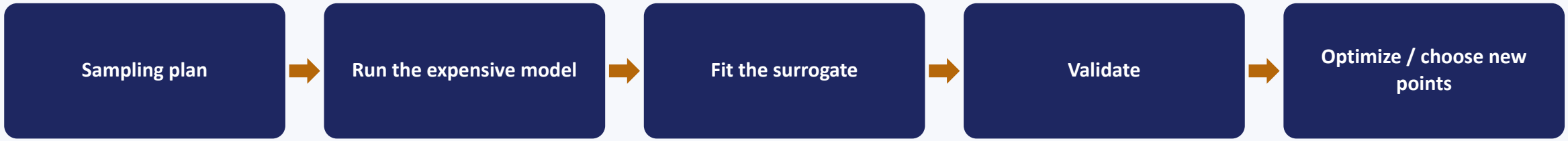
PART B · CHAPTERS 13–16

Surrogate Models and Response Surfaces

Sampling, regression, radial basis functions, Gaussian processes, Bayesian optimization

The Workflow

When evaluating a design is expensive, we build a model of the objective, optimize the model, and use further true evaluations to improve it. Fitting needs an initial set of points — ideally space-filling.



Where to sample?

Chapter 13 — sampling plans that cover the space with a limited budget.

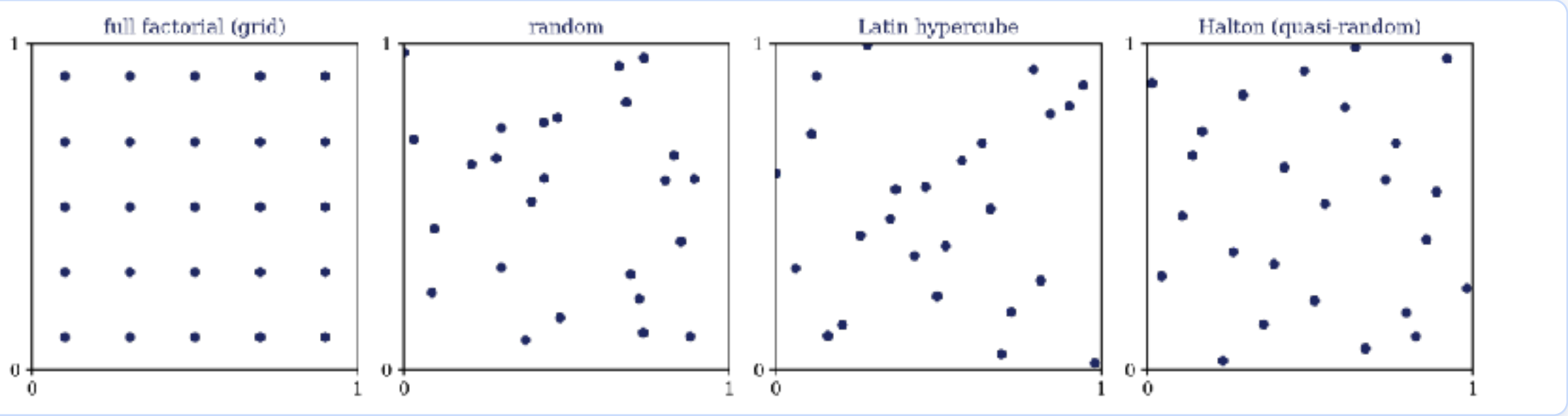
Which model?

Chapters 14–15 — regression with basis functions (incl. RBF); Gaussian processes that also quantify uncertainty.

Where next?

Chapter 16 — infill criteria that balance predicted value against uncertainty.

Where Should We Run the Expensive Model?



Full factorial

A grid of m points per dimension needs m^n samples — exponential in the number of dimensions.

Uniform projection

m samples on an m × m grid, one per row and column (Latin-hypercube style): uniform in every single coordinate.

Space-filling metrics

Discrepancy and pairwise distances measure coverage; a diagonal line of points is a uniform projection plan but covers a narrow strip.

Quasi-random sequences

Low-discrepancy sequences such as Halton and Sobol fill space evenly and can be extended point by point.

Fitting a Surrogate by Regression

Choose model parameters θ so that the surrogate matches the samples in an L_p norm — typically least squares:

$$\text{minimize}_{\boldsymbol{\theta}} \ \|\mathbf{y} - \hat{\mathbf{y}}\|_p \qquad \hat{y}^{(i)} = \hat{f}_{\boldsymbol{\theta}}(\mathbf{x}^{(i)})$$

A linear combination of basis functions $b_i(x)$ can be fitted in closed form with the Moore–Penrose pseudoinverse:

$$\hat{f}(\mathbf{x}) = \sum_{i=1}^{q} \theta_i b_i(\mathbf{x}) = \boldsymbol{\theta}^\top \mathbf{b}(\mathbf{x})$$

$$\boldsymbol{\theta} = \mathbf{B}^{+}\mathbf{y} \qquad \mathbf{B}_{ij} = b_j(\mathbf{x}^{(i)})$$

A purely linear model has n+1 parameters, so it needs at least n+1 samples; the pseudoinverse still returns a unique answer for any sample set.

Polynomial

Products of design-vector components raised to powers; any smooth function is close to a polynomial of sufficient degree (Taylor).

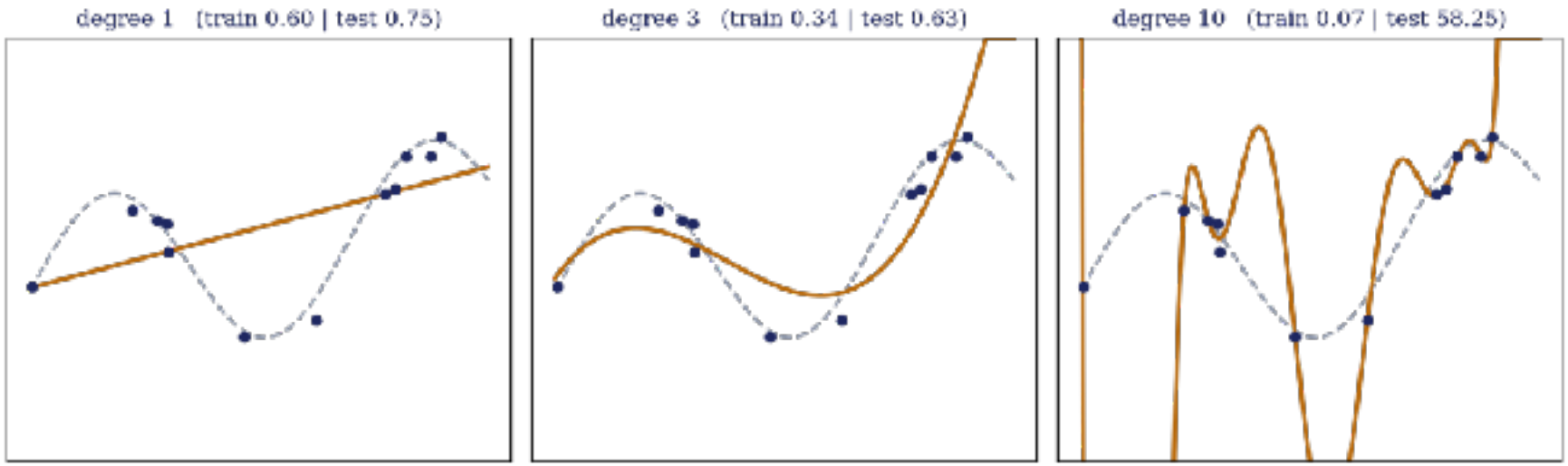
Sinusoidal

sin and cos of 2πix/T up to a chosen degree — natural for periodic behavior.

Radial

$b_i(x) = \psi(\|x - x^{(i)}\|)$: one hill or valley per data point. Gaussian, multiquadric, thin plate spline, cubic, … → next slides.

Overfitting, Regularization and Validation



Polynomial fits to 12 noisy samples (train / test RMS error in each title): the flexible model reproduces the samples but predicts badly between them.

Generalization error

$$\epsilon_{\text{gen}} = \mathbb{E}_{\mathbf{x}\sim\mathcal{X}}\left[\left(f(\mathbf{x}) - \hat{f}(\mathbf{x})\right)^2\right]$$

We care about error on the whole design space. Mean squared error on the training samples alone can be misleadingly small.

Regularization (noisy data)

$$\boldsymbol{\theta} = (\mathbf{B}^\top\mathbf{B} + \lambda\mathbf{I})^{-1}\mathbf{B}^\top\mathbf{y}$$

Penalize large weights with $\lambda\|\theta\|^2$: smoother fits, better predictors when evaluations are noisy.

Validation

Holdout split; k-fold cross-validation (leave-one-out when k = m); bootstrap. Always estimate error on points not used for fitting.

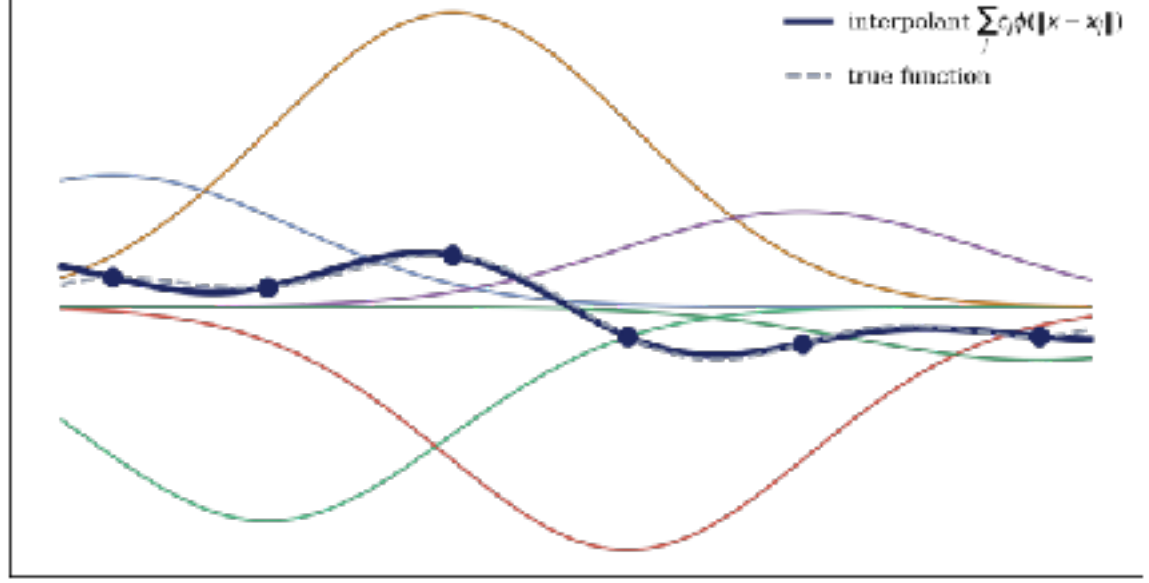
Linear Combinations of Translates of One Kernel

Key idea: place one copy of a single radial kernel ϕ(r) at every node and add them up. Only distances enter, so scattered nodes in any dimension — a cube, a sphere, a torus — are fine.

$$I_X f(\mathbf{x}) = \sum_{j=1}^{N} c_j \, \phi(\|\mathbf{x} - \mathbf{x}_j\|)$$

$$A_X \, \mathbf{c} = \mathbf{f}, \qquad (A_X)_{ij} = \phi(\|\mathbf{x}_i - \mathbf{x}_j\|)$$

The coefficients come from requiring the interpolant to reproduce the data at the nodes. If ϕ is positive definite, A_x is positive definite and the coefficients are uniquely determined by X and f.



Thin colored curves: the weighted kernels $c_j\varphi(\|x - x_j\|)$. Thick navy: their sum, which passes through every data point.

Positive Definite Radial Kernels

A kernel is positive definite if the interpolation matrix is positive definite for every set of distinct nodes:

$$\sum_{i=1}^{N}\sum_{j=1}^{N} b_i\,\phi(\|\mathbf{x}_i-\mathbf{x}_j\|)b_j > 0 \quad \text{for all } \mathbf{b}\neq 0$$

Theorem (Schoenberg, 1938)

Let Φ(r) = ϕ(√r). Then ϕ is positive definite on $\mathbb{R}^d$ for all d if and only if Φ is completely monotone on [0, ∞) and not constant.

Hence the Gaussian, inverse multiquadric and inverse quadratic kernels (right) all work in any dimension.

Gaussian

$$\phi(r) = e^{-(\varepsilon r)^2}$$

Inverse multiquadric

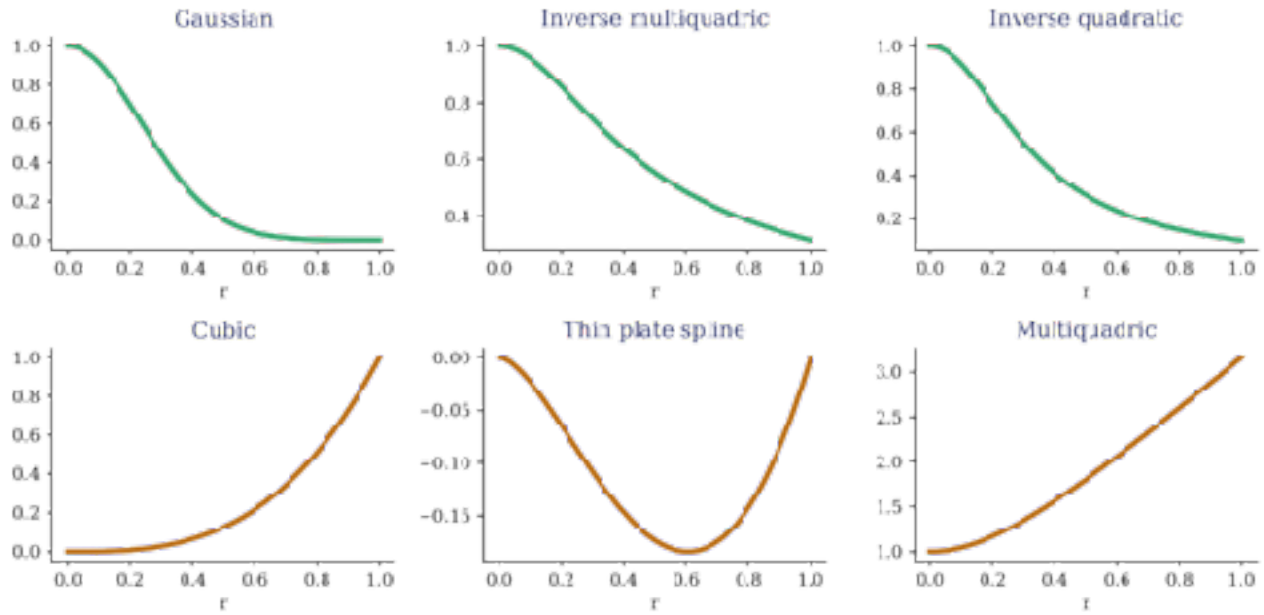
$$\phi(r) = \frac{1}{\sqrt{1+(\varepsilon r)^2}}$$

Inverse quadratic

$$\phi(r) = \frac{1}{1+(\varepsilon r)^2}$$

ε is the shape parameter — more on this shortly.

Important Kernels That Are Not Positive Definite



Cubic

$$\phi(r) = r^3$$

Cubic spline in 1-D

Thin plate spline

$$\phi(r) = r^2 \log r$$

Generalizes the energy-minimizing spline to 2-D

Multiquadric

$$\phi(r) = \sqrt{1 + (\varepsilon r)^2}$$

Popular; first used in any RBF application (Hardy, 1971)

These fall under conditionally positive definite (CPD) kernels of order k: the quadratic form is positive for every nonzero b that satisfies $\Sigma_j b_j p(x_j) = 0$ for all polynomials p of degree < k. The theory uses generalized Fourier transforms or generalized completely monotone functions (see Wendland, 2005, Ch. 8).

Polynomial Augmentation and Unisolvency

For a CPD kernel of order k, add a polynomial of degree < k (basis $p_1,\dots,p_n$) and impose side conditions on the coefficients:

$$I_X f(\mathbf{x}) = \sum_{j=1}^{N} c_j\, \phi(\|\mathbf{x}-\mathbf{x}_j\|) + \sum_{\ell=1}^{n} d_\ell\, p_\ell(\mathbf{x}), \qquad \sum_{j=1}^{N} c_j\, p_\ell(\mathbf{x}_j) = 0$$

$$\begin{bmatrix} A & P \\ P^{\top} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{c} \\ \mathbf{d} \end{bmatrix} = \begin{bmatrix} \mathbf{f} \\ 0 \end{bmatrix}$$

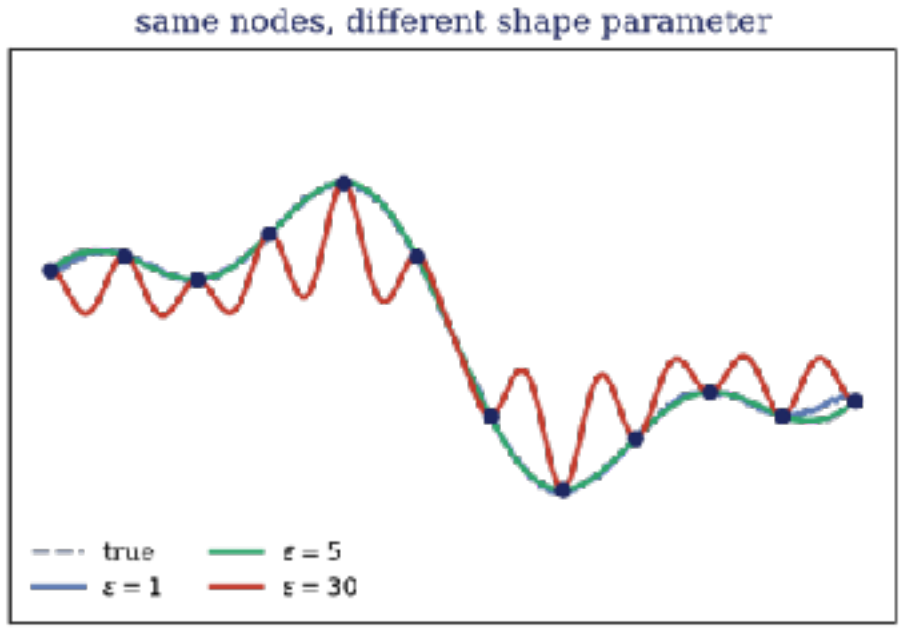
with $a_{ij} = \varphi(\|x_i - x_j\|)$ and $p_i\ell = p\ell(x_i)$.

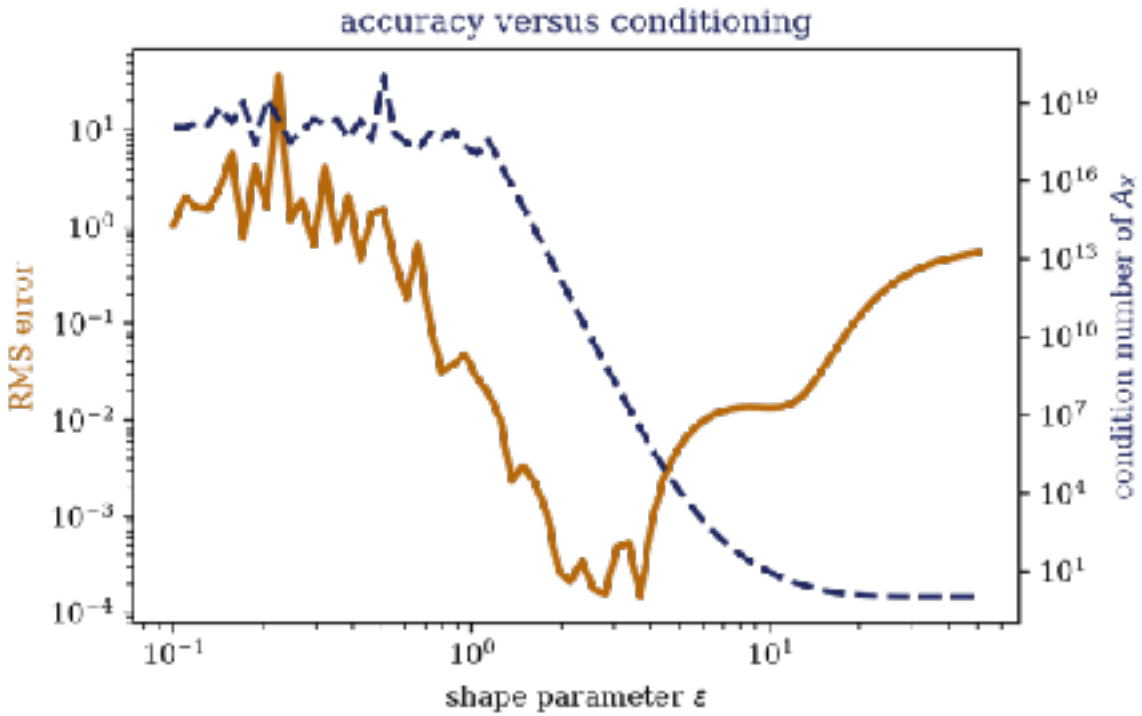
Theorem (Micchelli, 1986)

The system is invertible for any distinct nodes X provided that
- rank(P) = n (X is unisolvent on polynomials of degree < k),
- $\Phi = \phi(\sqrt{\cdot})$ is completely monotone of order k on $(0, \infty)$,
- $\Phi^{(k)}$ is not constant.

Example — thin plate spline in $\mathbb{R}^2$: $\varphi(r) = r^2 \log r$ with $p_1 = 1$, $p_2 = x$, $p_3 = y$. The system has a unique solution provided the nodes are not collinear.

The Shape Parameter ε

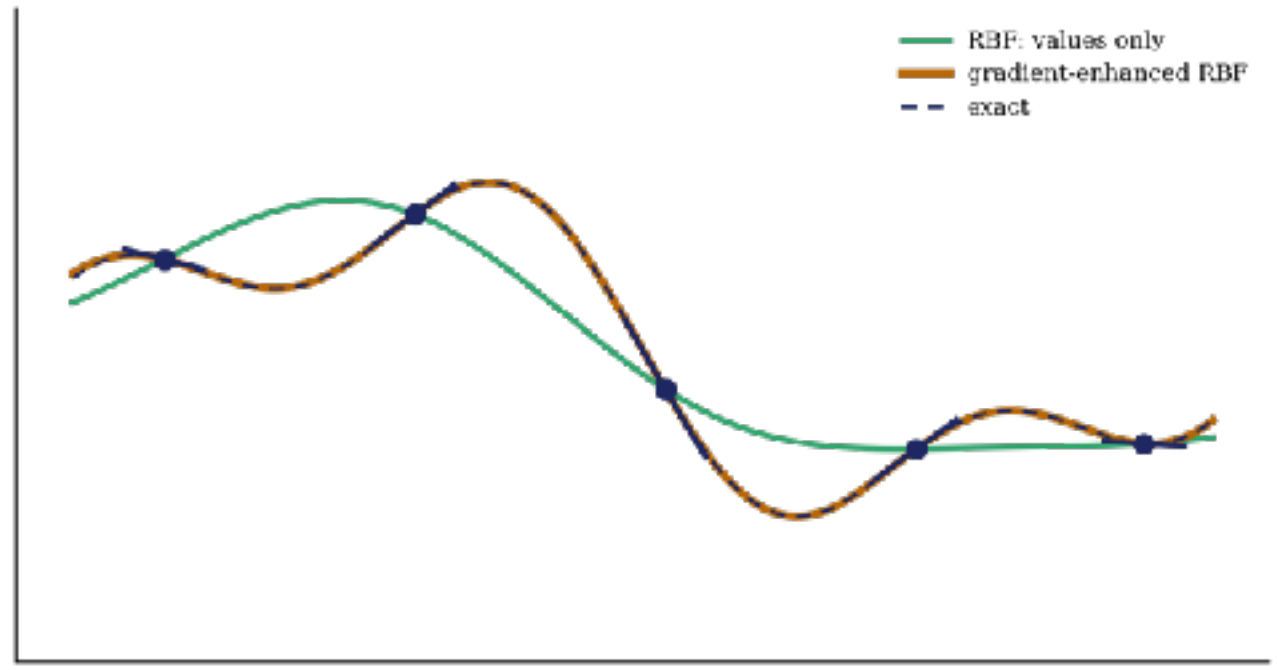




Gaussian RBF, 12 equispaced nodes. Small ε: nearly flat kernels — accurate, but A_x is severely ill-conditioned (numerical noise at ε = 1). Large ε: spiky kernels — well conditioned, poor between nodes.

Choose ε by cross-validation or likelihood — the sweet spot is far from the extremes. Surrogate accuracy in the case study below also depends strongly on such internal kernel parameters.

Gradient-Enhanced RBF



Five nodes. Short bars mark the prescribed slope at each node.

Also interpolate the gradient at each node (Hermite-type interpolation):

$$I_X f(\mathbf{x}_i) = f_i, \qquad \nabla I_X f(\mathbf{x}_i) = \nabla f_i \qquad i = 1, \ldots, N$$

More data per sample

Each node supplies 1 + d conditions instead of 1.

Large accuracy gain

In this 1-D test: RMS error ≈ 0.005 versus 0.34 for values-only RBF on the same 5 nodes.

The price

Gradients must be available — e.g. from an adjoint or from an ensemble (Part C).

Gaussian processes use the same idea: treat values and gradients as jointly Gaussian (textbook §15.4) — see next slides.

Gaussian Processes and Kriging

A Gaussian process is a distribution over functions: any finite set of function values is jointly Gaussian, with a mean function m(x) and a covariance function (kernel) k(x, x′).

$$f(\mathbf{x}) \sim \mathcal{GP}(m(\mathbf{x}),\, k(\mathbf{x}, \mathbf{x}'))$$

$$k(x, x') = \exp\left(-\frac{(x - x')^2}{2\ell^2}\right)$$

The length-scale ℓ is the distance over which the function changes significantly; larger ℓ gives smoother functions.

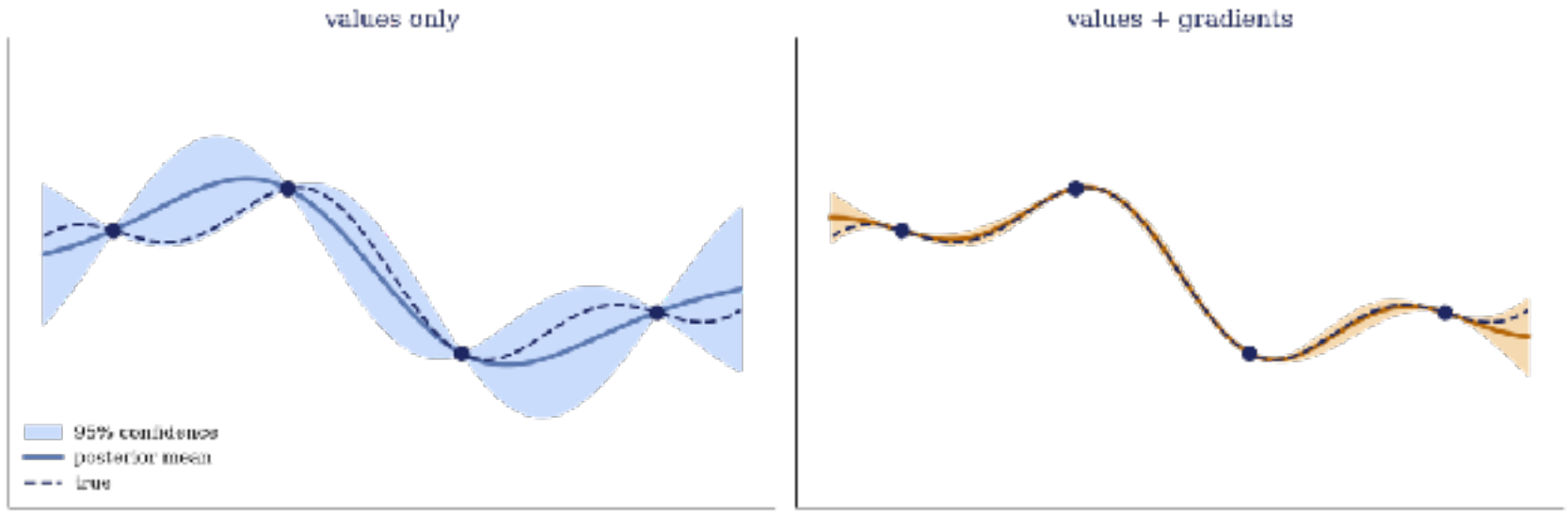
Conditioning on the data (X, y) gives the prediction and its uncertainty in closed form:

$$\hat{\mu}(\mathbf{x}) = m(\mathbf{x}) + \mathbf{K}(\mathbf{x}, X)\,\mathbf{K}(X, X)^{-1}(\mathbf{y} - \mathbf{m}(X))$$

$$\hat{v}(\mathbf{x}) = \mathbf{K}(\mathbf{x}, \mathbf{x}) - \mathbf{K}(\mathbf{x}, X)\,\mathbf{K}(X, X)^{-1}\,\mathbf{K}(X, \mathbf{x})$$

The mean is $m(x) + \vartheta^T K(x, X)$ with $\vartheta = K(X,X)^{-1}(y - m(X))$ — the same structure as RBF interpolation. What the GP adds is the variance.

Prediction, Uncertainty, Gradients and Noise



Same four samples, same kernel. Right: adding gradient observations shrinks the 95% confidence band (mean width 0.24 → 0.04 in this 1-D example).

$$\hat{\mu}(\mathbf{x}) = m(\mathbf{x}) + \mathbf{K}(\mathbf{x}, X)(\mathbf{K}(X, X) + \nu\mathbf{I})^{-1}(\mathbf{y} - \mathbf{m}(X))$$

$$\log p(\mathbf{y} \mid X, \nu, \boldsymbol{\theta}) = -\frac{1}{2}\mathbf{y}^{\top}(\mathbf{K}_{\boldsymbol{\theta}} + \nu\mathbf{I})^{-1}\mathbf{y} - \frac{1}{2}\log|\mathbf{K}_{\boldsymbol{\theta}} + \nu\mathbf{I}| - \frac{n}{2}\log 2\pi$$

Surrogate Families at a Glance

Surrogate	Data regime	Uncertainty estimate	Gradient information	Worth remembering
RBF	Tens to hundreds of scattered points	Not built in (use cross-validation)	Not used	Simple; shape parameter and ill-conditioning matter
Gradient-enhanced RBF	Few points with gradients	Not built in	Interpolated at nodes	Large accuracy gain per sample; needs gradients
Kriging (Gaussian process)	Tens to thousands of points	Yes — posterior variance	Optional (§15.4)	Natural for Bayesian optimization; fitting cost grows with data
Co-Kriging	Many cheap + few expensive points	Yes	Optional	Needs correlated fidelity levels
Neural network	Large data sets, many inputs or outputs (e.g. fields)	Not built in (ensembles, dropout, …)	Via automatic differentiation	Flexible but data-hungry; training and tuning effort

General guidance, not strict rules. For a few hundred samples of a smooth objective, RBF and Kriging are usually the first choices.

Choosing the Next Evaluation

With a probabilistic surrogate, each candidate point has a predicted value $\hat{\mu}$ and an uncertainty $\hat{\sigma}$. Three greedy rules combine them differently:

Prediction-based

$$\mathbf{x}^{(m+1)} = \arg\min_{\mathbf{x}\in\mathcal{X}} \hat{\mu}(\mathbf{x})$$

Exploit: take the minimizer of the mean.

Ignores uncertainty — new samples can land right next to existing ones, wasting evaluations.

Error-based

$$\mathbf{x}^{(m+1)} = \arg\max_{\mathbf{x}\in\mathcal{X}} \hat{\sigma}(\mathbf{x})$$

Explore: take the most uncertain point.

Reduces uncertainty overall, but samples regions unlikely to hold the global minimum.

Lower confidence bound

$$LB(\mathbf{x}) = \hat{\mu}(\mathbf{x}) - \alpha\,\hat{\sigma}(\mathbf{x}), \qquad \alpha \geq 0$$

Trade off with α ≥ 0.

α = 0 recovers prediction-based search; α → ∞ error-based exploration.

The tension between exploitation (low predicted value) and exploration (high uncertainty) is the heart of surrogate optimization.

Probability and Expected Improvement

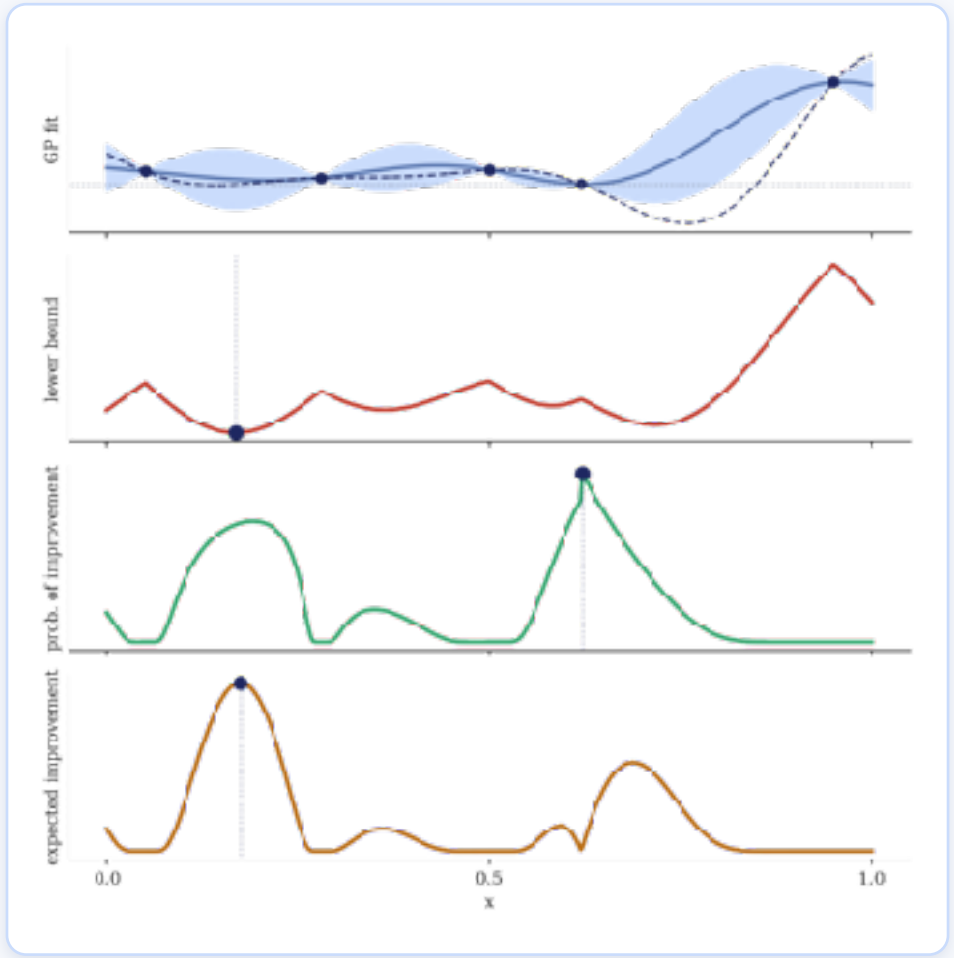
Let y_min be the best value sampled so far. The improvement of a new sample y is I(y) = y_min − y if y < y_min, and 0 otherwise.

$$P(y < y_{\min}) = \Phi\left(\frac{y_{\min} - \hat{\mu}}{\hat{\sigma}}\right)$$

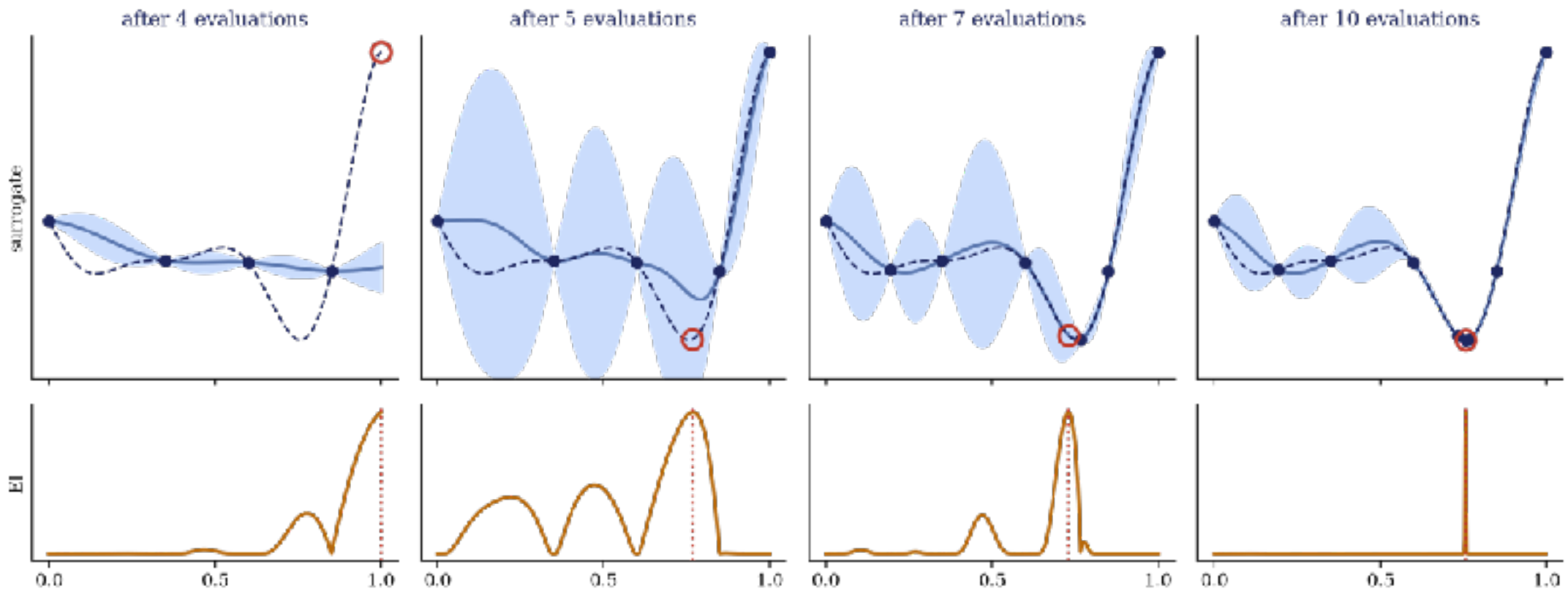
PI ignores how large the improvement could be — it prefers safe, tiny gains.

$$\mathbb{E}[I(y)] = (y_{\min} - \hat{\mu})\,\Phi\left(\frac{y_{\min} - \hat{\mu}}{\hat{\sigma}}\right) + \hat{\sigma}\,\varphi\left(\frac{y_{\min} - \hat{\mu}}{\hat{\sigma}}\right)$$

Φ and ϕ are the standard normal cumulative distribution and density. EI weighs the size of the gain, so it balances exploration and exploitation automatically. Both vanish where $\hat{\sigma} = 0$ (already-sampled, noise-free points).



Bayesian Optimization in Action



Gaussian-process surrogate + expected improvement on a standard 1-D benchmark with several local minima (Forrester function). Four initial samples; red circle = next evaluation. The global minimum (x ≈ 0.758) is found within eight evaluations in total, and EI then collapses.

RBF Surface + Stochastic Global Search

DYCORS

A stochastic global algorithm that uses an RBF response surface to screen candidate points; efficient for large numbers of parameters (Regis & Shoemaker).

Candidate generation

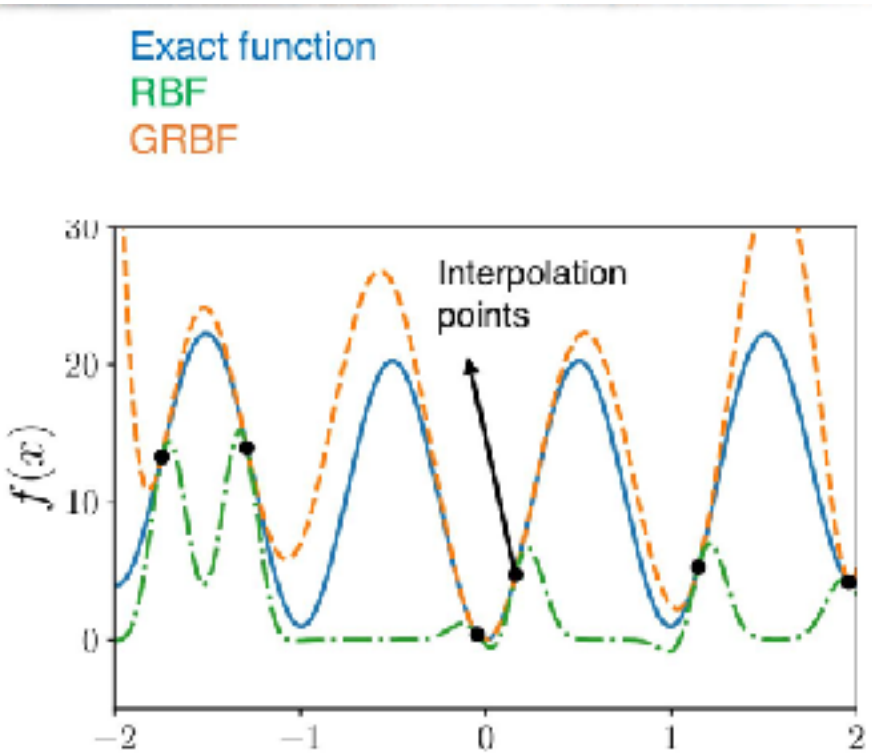
Candidates are produced by perturbing a subset of the coordinates of the current best point; the subset shrinks as iterations proceed, which is what lets it scale to many parameters.

Gradient enhancement

Gradient-enhanced RBF (G-DYCORS) injects gradient information (e.g. adjoint) into the surface.

The multi-fidelity loop

Explore the cheap surface; run the CFD code only at the selected point; update.

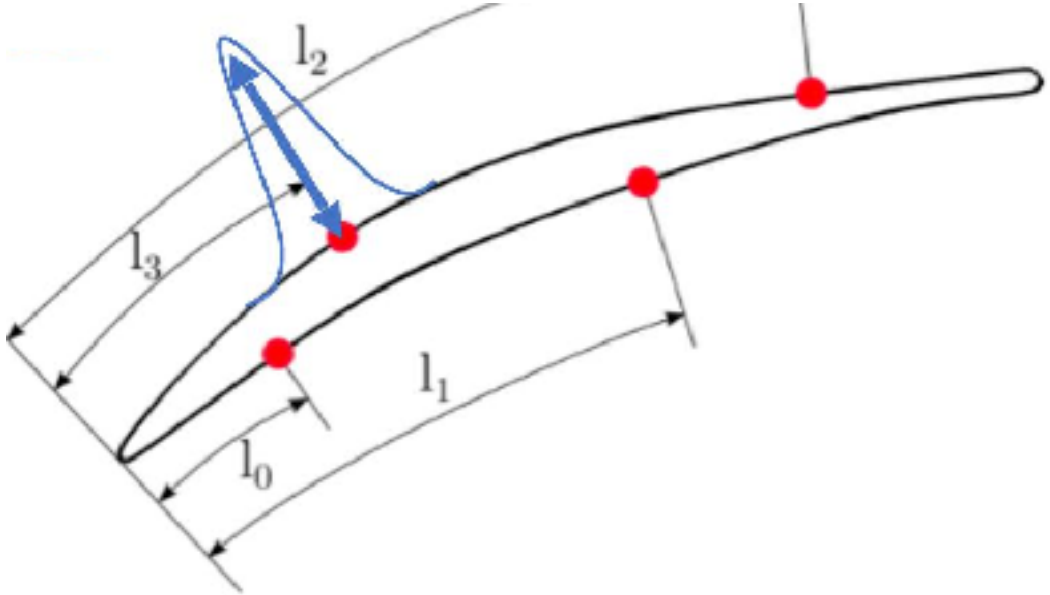


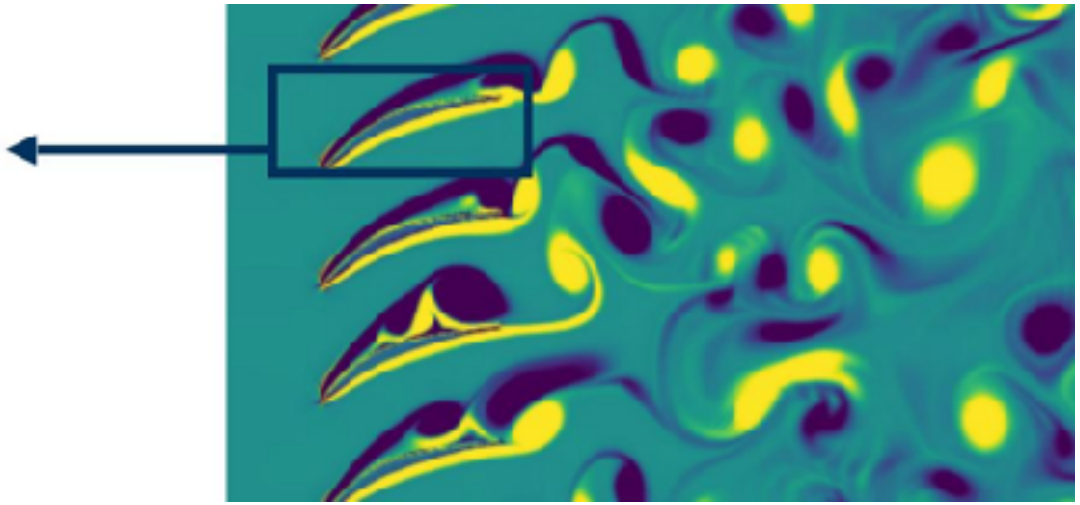
1-D test: exact function, RBF and GRBF built from the marked interpolation points. GRBF accuracy depends strongly on an internal kernel parameter.

Source: Quirós Rodríguez, Fosas de Pando & Sayadi (2024), Comput. Phys. Commun. 298, 109122.

CASE STUDY — UNSTEADY FLOW CONTROL

Actuating a Turbine-Blade Cascade





Objective

Minimize the pressure loss

Actuator

Tangential velocity imposed on the blade surface

Per actuator

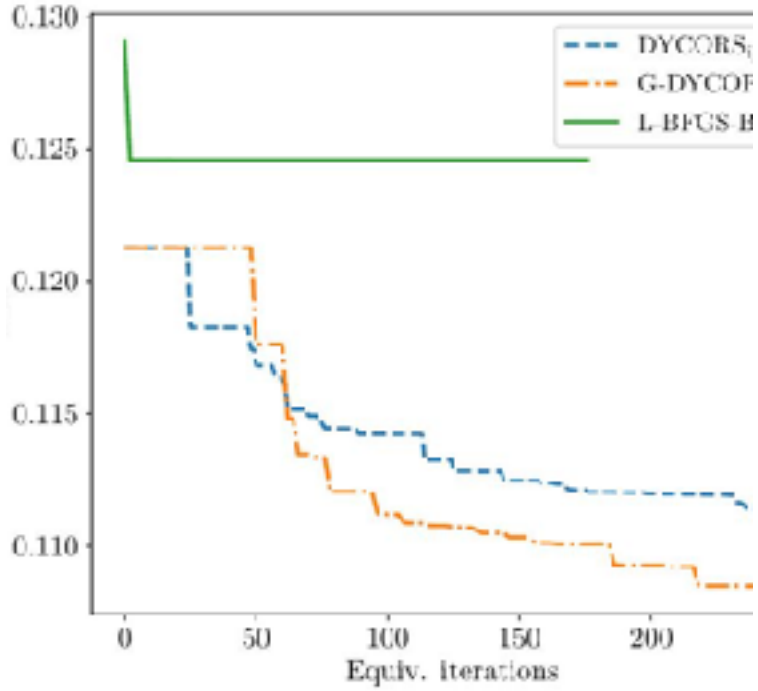
Amplitude, width, location, frequency, phase

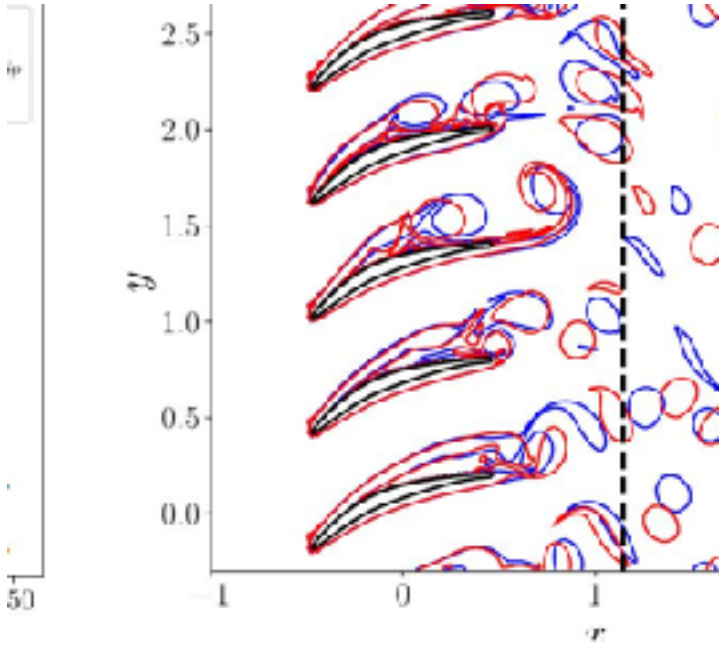
Dimension

24 control parameters; Re = 800 and 4000

Unsteady, chaotic flow with 24 parameters: exactly the setting where adjoint gradients are unreliable and plain stochastic search is costly.

Surrogate-Assisted Search versus a Gradient Method





- Applicable to complex flows
- Applicable to chaotic configurations
- Finds the global minima
- **Large number of function evaluations**
- Robustness

Re = 4000. L-BFGS-B stalls almost immediately near f ≈ 0.1245; DYCORS and G-DYCORS keep improving, reaching roughly 0.111 and 0.1085 (read from the plot) within about 250 equivalent iterations. Right: red = control off, blue = control on.

Amber: the remaining drawback.

RECAP

Part B — Key Takeaways

1. Surrogate-based optimization: sample the space, run the expensive model, fit a surrogate, validate, then choose new points.
2. Regression with basis functions fits a surrogate; validate on held-out data to guard against overfitting.
3. RBF interpolation uses translates of one radial kernel; (conditionally) positive definite kernels guarantee a unique solution; the shape parameter trades accuracy for conditioning.
4. Gaussian processes (Kriging) add uncertainty to the same kernel structure and enable Bayesian optimization through criteria such as expected improvement.
5. Gradients, when available, make surrogates dramatically more accurate; surrogate-assisted stochastic search handled a 24-parameter unsteady flow where L-BFGS-B stalled.

PART C · ENSEMBLE-VARIATIONAL METHODS

Gradient-Based Optimization Without an Adjoint

Approximate the gradient from an ensemble of simulations

MOTIVATION

Can We Combine the Two Families?

Gradient-based methods

- Efficiency and robustness
- **Local method**
- **Differentiation is complex; adjoint extraction is intrusive**

Multi-fidelity derivative-free methods

- Global method
- Easy to implement (non-intrusive)
- Applicable to complex PDEs and non-deterministic problems
- **Large number of function evaluations**

Green: advantage. Amber: drawback.

Idea: keep the gradient-based update, but build the gradient from an ensemble of simulations instead of an adjoint solve.

Why Adjoint-Free?

Interfacial flows

The equations are very complicated to differentiate: deriving and coding the adjoint is a major effort.

Turbulent / chaotic flows

Gradients from adjoint methods are not accurate over long horizons.

The gradient is an approximation anyway

In practice optimizers only need a good enough descent direction — which an ensemble can supply.

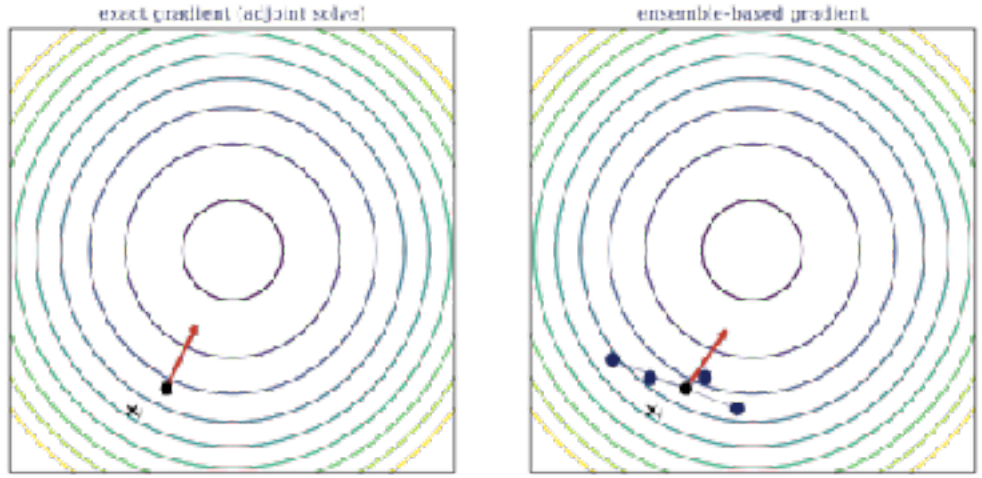


Illustration on a simple quadratic: the ensemble-based descent direction (right) closely follows the exact gradient (left) — here within about 9°.

Approximating the Gradient from Samples

Finite differences

$$\frac{\partial f}{\partial x_k} \approx \frac{f(\mathbf{x} + h\,\mathbf{e}_k) - f(\mathbf{x})}{h}$$

One sample per direction: d + 1 simulations for d parameters. Accurate for small h, but the cost grows with d.

SPSA

$$\hat{g}_k = \frac{f(\mathbf{x} + c\Delta) - f(\mathbf{x} - c\Delta)}{2c\,\Delta_k}$$

Simultaneous perturbation: all parameters perturbed at once along a random Δ (e.g. ±1 entries). Two simulations whatever d — an under-sampled finite difference, hence noisy.

Ensemble-variational

$$g \approx P\,w$$

Use N_ens perturbed simulations together and optimize how they are combined: “optimize the sample space”. The best descent direction inside the ensemble's span is found by a small least-squares problem.

Same ingredients (perturbed simulations), increasing cleverness in how they are used.

The Formulation

1 Ensemble

$$P = \left[\mathbf{x}^{(1)} - \mathbf{x}_i,\ \mathbf{x}^{(2)} - \mathbf{x}_i,\ \ldots,\ \mathbf{x}^{(N_{\text{ens}})} - \mathbf{x}_i\right]$$

N_ens members $x^{(j)}$ perturb the current design x_i; the columns of P are the perturbation directions.

2 Linear model

$$f(\mathbf{x}_i + P\mathbf{w}) \approx f(\mathbf{x}_i) + Y\mathbf{w}, \qquad Y = \left[f(\mathbf{x}^{(j)}) - f(\mathbf{x}_i)\right]_{j=1}^{N_{\text{ens}}}$$

The ensemble gives the response of f along those directions: a linear model of f in the span of P.

3 Quadratic cost

$$\mathcal{J}(\mathbf{w}) = \frac{1}{2}\|\mathbf{y}\|^2_{\Sigma_H^{-1}} + \frac{1}{2}(N_{\text{ens}} - 1)\|\mathbf{p}\|^2_I$$

A cost in the weights w: misfit of the linear model (weighted by Σ_H) plus a term limiting the step. Gradient and Hessian in w are straightforward.

4 Step and update

$$-\gamma \frac{\partial \mathcal{J}}{\partial \mathbf{x}} \simeq \mathbf{p} = P\mathbf{w}$$

The step p = Pw is a linear combination of the ensemble directions and approximates the descent direction, then $x_{i+1} = x_i + p$ (equivalently $x_i - \gamma \partial \mathcal{J}/\partial x$).

Here $y = f(x_i) + Yw$ is the linearized objective, and the minimization over w is solved with Newton, conjugate gradient or L-BFGS — the dimension is only N_ens.

One EnVar Iteration

Perturb x_i to form N_ens members → **Run N_ens simulations (in parallel)** → **Build P and the response matrix** → **Minimize $\mathcal{J}(w)$: a small quadratic problem** → **Step p = Pw; update x**

Cost per iteration

N_ens independent simulations — no adjoint, no code changes, and they run in parallel.

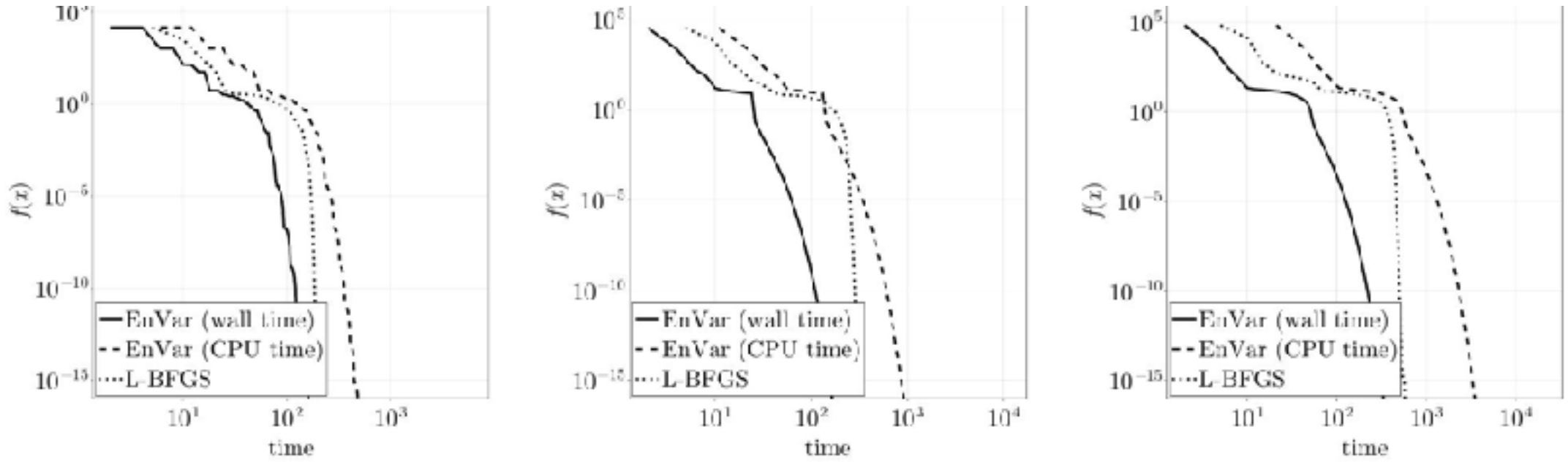
The inner problem

Quadratic in w, with only N_ens unknowns: negligible compared with a CFD run. Newton, conjugate gradient or L-BFGS apply (Lectures 2–3).

Where randomness enters

The ensemble perturbations are random, so the method does not stall exactly at a saddle point or an unlucky local structure (cf. noisy descent, Lecture 4).

Beale Function: EnVar versus L-BFGS



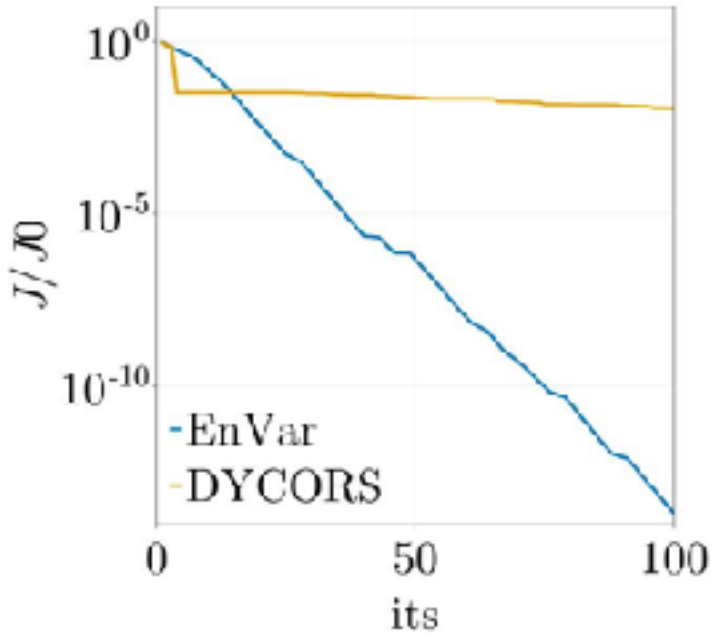
(a) Results for $d = 5$. (b) Results for $d = 10$. (c) Results for $d = 20$.

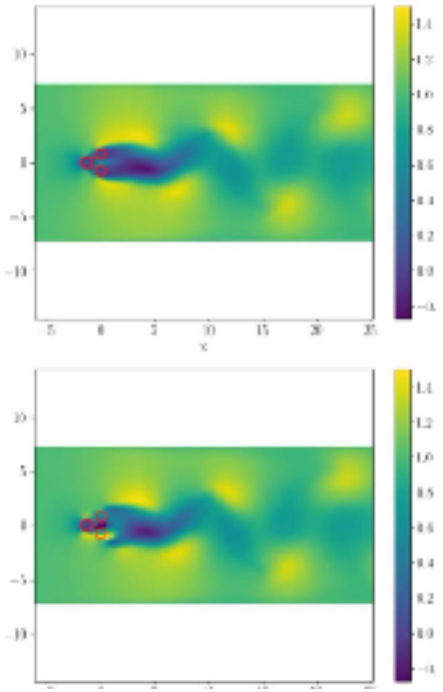
A modified d-dimensional Beale function, d = 5, 10, 20. EnVar with momentum (best configuration shown) against L-BFGS; f(x) versus time.

Reading the plots: in wall-clock time EnVar converges fastest, since its ensemble members run in parallel; in CPU time it costs more than L-BFGS — the price of being adjoint- and derivative-free.

Three Rotating Cylinders at Re = 50

Control objective: the angular velocity of each cylinder — three parameters. Compared with the surrogate-assisted stochastic method (DYCORS) per number of function evaluations.





J/J_0 versus iterations: DYCORS stalls at order 10^{-1}, while EnVar keeps decreasing below 10^{-10} (read from the plot).

Right: velocity field, top without control, bottom with EnVar-optimized rotation.

Properties and Trade-offs

What it offers

- Works for high-dimensional parameter spaces
- Does not get stuck in local minima — thanks to the stochasticity of the algorithm
- Has the robustness of gradient-based methods
- Less costly than purely stochastic methods
- Does not suffer from lack of differentiability or the chaotic nature of the problem

What to keep in mind

- Every iteration needs N_ens simulations: CPU cost exceeds that of an adjoint-based method
- The gradient is an approximation — its quality depends on the ensemble size and perturbation size
- A step size or regularization must be chosen
- Best exploited when simulations can run in parallel

The right-hand list is our reading of the method, not a result from the tests.

EnVar as a Local Surrogate

	Global surrogate (RBF, Kriging)	Ensemble-variational
Model	Fitted to all samples; valid over the whole domain	Linear model in the span of the current ensemble; valid near x_i
Lifetime	Persistent: refined as data accumulate	Rebuilt at every iteration
Built from	Any set of past evaluations (reusable)	N_ens fresh, centered perturbations
Strength	Global exploration; can find the global optimum	Fast local descent without an adjoint
Weakness	Accuracy degrades with dimension	Local; needs N_ens simulations per step
Natural combination	Use global surrogate search to find a good basin	Then polish with EnVar — the hybrid idea of Lecture 4

Both replace an expensive function by a cheap local or global model; they differ in how long the model lives and how far it is trusted.

RECAP

Part C — Key Takeaways

1. Gradient-based methods are efficient but intrusive; derivative-free methods are global but costly — EnVar aims to combine the advantages.

2. The gradient is built from an ensemble of perturbed simulations; no adjoint equations are needed.

3. EnVar finds the best combination of ensemble directions by minimizing a small quadratic cost in the weights w.

4. On analytical and flow-control tests it outperformed stochastic and L-BFGS baselines in the metrics shown — at the price of N_ens simulations per iteration.

5. EnVar can be read as a local surrogate rebuilt each iteration, complementing the global surrogates of Part B.

COURSE VIEW

Choosing a Method in Practice

Approach	Needs derivatives?	Handles turbulence / chaos	Evaluation budget	Best when…
Adjoint-based descent (Lectures 2–3)	Yes (adjoint)	Poorly over long horizons	Very small	Smooth, differentiable problem with an available adjoint
Ensemble-variational	No (ensemble gradient)	Yes	N_ens simulations per iteration	Many parameters, parallel resources, no adjoint
Surrogate + global search (RBF, Kriging, BO)	No (optional)	Yes	Small to moderate	Expensive simulations, moderate dimension
Stochastic / population methods (Lecture 4)	No	Yes	Large	Rugged landscapes; cheap evaluations; global search

Hybrids are common: global surrogate search to find a basin, then a local gradient or ensemble-gradient method to refine. No single method wins on every problem.

COURSE RECAP

The Course in Five Lectures

1

Foundations

Problem formulation, critical points, derivatives: finite differences, complex step, automatic differentiation.

2

Local Descent

Line search, trust regions; gradient, conjugate-gradient, momentum, Adam; Newton and quasi-Newton.

3

Constraints

Lagrange multipliers, KKT, duality, penalty and barrier methods; one-shot and adjoint ideas.

4

Stochastic & Population

Noisy descent, annealing, cross-entropy, CMA-ES; genetic algorithms, DE, PSO.

5

Surrogates & EnVar

Multi-fidelity, RBF, Kriging, Bayesian optimization; ensemble-variational optimization.

Common thread: build a model of the landscape — local (gradient, Hessian, ensemble) or global (surrogate, distribution, population) — and decide how far to trust it.

PROJECT DIRECTIONS

Applying the Methods to Simple Flows

1. **Adjoint-based passive control**
   Optimize a small obstacle or shape with adjoint gradients.

2. **Extended Kalman filter**
   State estimation and feedback control on a reduced model.

3. **Ensemble-variational optimization**
   Adjoint-free gradients from simulation ensembles.

4. **Kriging / neural-net surrogates**
   Surrogate-based or Bayesian optimization of a design.

5. **Genetic algorithms for flow control**
   Evolve controller parameters; compare with CMA-ES.

6. **Reinforcement learning**
   Learn a feedback policy against a reward.

7. **Data assimilation for flows**
   Reconstruct a flow from sparse data (identical-twin test).

Shaded: topics closest to today's lecture. Suggested test beds: 1D Burgers or Kuramoto–Sivashinsky first, then a laminar cylinder wake at Re ≈ 40–100.

DISCUSSION

Questions to Think About

1. A surrogate fitted to 50 CFD runs predicts well at the samples but badly in between. Which diagnostics would you run, and what would you change?

2. Why does expected improvement balance exploration and exploitation automatically, while a plain minimization of the surrogate mean does not?

3. When would you accept the extra CPU cost of EnVar over an adjoint method? Over a surrogate-based global search?

4. You have a cheap 2-D RANS model and an expensive 3-D LES. How would you structure a multi-fidelity optimization?

Selected References

- Kochenderfer & Wheeler (2019). Algorithms for Optimization. MIT Press — Chapters 13–16.
- Kennedy & O'Hagan (2000). Predicting the output from a complex computer code when fast approximations are available. Biometrika 87.
- Forrester, Sobester & Keane (2007). Multi-fidelity optimization via surrogate modelling. Proc. R. Soc. A 463.
- Forrester, Sobester & Keane (2008). Engineering Design via Surrogate Modelling. Wiley.
- Peherstorfer, Willcox & Gunzburger (2018). Survey of multifidelity methods in uncertainty propagation, inference and optimization. SIAM Review 60.
- Jones, Schonlau & Welch (1998). Efficient global optimization of expensive black-box functions. J. Global Optim. 13.
- RBF theory: Buhmann (2003); Wendland (2005); Fasshauer (2007); Fornberg & Flyer, A Primer on Radial Basis Functions with Applications to the Geosciences (SIAM); Schoenberg (1938); Micchelli (1986); Hardy (1971).
- Regis & Shoemaker (2007), INFORMS J. Computing 19(4); Wu Zongmin (1992), Approx. Theory Appl. 8(2) — as cited in the lecture material.
- Quirós Rodríguez, Fosas de Pando & Sayadi (2024). Gradient-enhanced stochastic optimization of high-fidelity simulations. Comput. Phys. Commun. 298, 109122.
- Liu, Xiao & Wang (2008). An ensemble-based four-dimensional variational data assimilation scheme. Mon. Weather Rev. 136 (verify).
- Spall (1992). Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. IEEE Trans. Autom. Control 37 (verify).

References compiled without database access: please check titles, volumes and years before distributing.

Questions?

Reading: Algorithms for Optimization, Chapters 13–16, and the project reference list

Kochenderfer, M. J., & Wheeler, T. A. Algorithms for Optimization. MIT Press, 2019.