Approximate Bayesian Computation (ABC) - Lecture Notes

Handbook of Approximate Bayesian Computation

Lecture Notes

2026-01-20

Chapter 1: Overview of Approximate Bayesian Computation (ABC)

Why ABC Exists (Motivation)

Bayesian inference lives and dies by the **likelihood**. If you can evaluate $p(y_{\text{obs}} \mid \theta)$, then Bayes' theorem gives you the posterior:

$$\pi(\theta \mid y_{\text{obs}}) \propto p(y_{\text{obs}} \mid \theta)\pi(\theta)$$

Standard tools—MCMC, importance sampling, SMC—*all assume* that likelihood evaluations are cheap enough to compute thousands or millions of times.

But many modern models violate this assumption:

- Likelihoods exist **in principle**, but are **computationally intractable**
- Normalizing constants depend on parameters (e.g. Markov random fields)
- Models defined **only via simulation** (agent-based models, climate simulators)
- Likelihoods implicit via quantiles or characteristic functions
- Massive datasets without low-dimensional sufficient statistics

The choice becomes grim:

- Simplify the model → lose realism
- Abandon Bayesian inference → lose coherence

ABC offers a third path.

The Core Idea (Likelihood-Free Intuition)

ABC replaces **likelihood evaluation** with **likelihood simulation**.

Key philosophical pivot:

> Instead of asking *"What is the probability of the observed data under ?"*
> ABC asks *"Could plausibly have generated data that look like what we observed?"*

If you can:

1. Sample $\theta \sim \pi(\theta)$
2. Simulate $y \sim p(y \mid \theta)$

Then inference is still possible—*even if the likelihood is never evaluated numerically.*

Likelihood-Free Rejection Sampling

Start with classical rejection sampling for the posterior. Replace the acceptance probability (which involves the likelihood) with a **simulation-based test**:

1. Draw $\theta \sim \pi(\theta)$
2. Simulate $y \sim p(y \mid \theta)$
3. Accept θ **if simulated data match observed data**

If data are **discrete**, exact matching is possible: $y = y_{\text{obs}}$

This produces **exact posterior samples**, with *no approximation.* This is likelihood-free — but **not yet ABC**.

The Necessity of Approximation (Why ABC Is Approximate)

Exact matching is fragile:

- Acceptance rates are astronomically low
- Impossible for continuous data
- Becomes unusable in high dimensions

The fix is conceptual but profound:

> Replace "exactly equal" with "sufficiently close"

Introduce:

- A distance $|y - y_{\text{obs}}|$

- A tolerance $h > 0$

Accept if: $|y - y_{\text{obs}}| \leq h$

Now the algorithm is feasible—but approximate. This is where **Approximate Bayesian Computation** truly begins.

What Distribution Is ABC Sampling From?

ABC does **not** sample from the true posterior.

Instead, it samples from:

$$\pi_{\text{ABC}}(\theta \mid y_{\text{obs}}) \propto \int K_h(|y - y_{\text{obs}}|) p(y \mid \theta) \pi(\theta) dy$$

Interpretation:

- The likelihood is **smoothed**
- ABC replaces $p(y_{\text{obs}} \mid \theta)$ with a **kernel-averaged likelihood**
- As $h \to 0$, the approximation converges to the true posterior

This makes ABC mathematically equivalent to **Bayesian inference with an approximate likelihood**.

Kernel Viewpoint (A Crucial Insight)

Instead of a hard cutoff, ABC often uses a kernel:

$$K_h(u) = \frac{1}{h} K\left(\frac{u}{h}\right)$$

Common kernels:

- Uniform
- Epanechnikov
- Gaussian

This smooths acceptance:

- Perfect matches get high weight
- Near matches get partial credit
- Poor matches fade to zero

The **choice of tolerance (h)** dominates the error. Kernel choice matters far less.

Summary Statistics: The Make-or-Break Decision

High-dimensional data doom ABC unless we compress intelligently.

Replace data comparison:

$$|y - y_{\text{obs}}| \quad \rightarrow \quad |S(y) - S(y_{\text{obs}})|$$

Goals of summary statistics:

- Low dimension
- Highly informative about θ
- Ideally sufficient (rare in practice)

This is not optional—it's the central design problem of ABC.

Poor summaries → biased posteriors
Good summaries → surprisingly accurate inference

What ABC Is—and Is Not

Important distinctions:

- **Likelihood-free ABC**
  - Exact matching → likelihood-free but not approximate
- **ABC = likelihood-free + approximation**
- ABC approximates the posterior, not Bayes' theorem
- Approximation error is controlled by:
  - Tolerance (h)
  - Choice of summary statistics
  - Distance metric

ABC trades **bias for feasibility**, consciously and transparently.

Conceptual Interpretations of ABC

ABC can be understood as:

- Bayesian inference with noisy observations
- Kernel-smoothed likelihood inference
- Prior predictive filtering
- A Monte Carlo test of model plausibility

All interpretations agree on one thing:

> ABC is coherent Bayesian inference under computational constraints.

Big Picture Takeaways

1. ABC exists because realistic models break classical Bayesian computation
2. Simulation can replace likelihood evaluation
3. Approximation enters through tolerance and summaries
4. ABC approximates the posterior, not Bayesian reasoning itself
5. The art of ABC lies in:
    - Summary statistics
    - Calibration of tolerance
    - Understanding approximation error

Chapter 2: On the History of Approximate Bayesian Computation

Why Study the History of ABC?

This chapter is not about algorithms. It is about **how statisticians reasoned their way into ABC before they had a name for it**.

ABC did not arrive as a fully formed method. It emerged gradually, driven by:

- Models becoming more realistic than their likelihoods
- Scientists insisting on Bayesian coherence anyway
- Computational pragmatism outrunning analytic elegance

Understanding this history prevents a common mistake: thinking ABC is a modern computational trick rather than a **deeply Bayesian idea with old roots**.

The Prehistory: Bayesian Inference Before Likelihoods Were Sacred

Early Bayesian analysis was often **simulation-based by necessity**, not by choice.

Before fast computers:

- Likelihoods were sometimes unknown or unusable
- Posterior reasoning relied on conditional arguments, predictive reasoning, and approximation

Crucially, Bayesians always understood that:

> If you can simulate from a model, you understand it more deeply than if you can merely write its likelihood.

This attitude sets the stage for ABC.

Population Genetics: Where ABC Was Forced Into Existence

ABC's true origin lies in **population genetics**, not statistics departments.

Population genetic models had three properties:

1. Extremely realistic stochastic mechanisms (coalescent models)
2. Likelihoods involving combinatorial explosions
3. Easy forward simulation of genetic data

Scientists could simulate millions of datasets—but could not evaluate a single likelihood.

The core inferential question became:

> "Which parameter values generate data resembling what we observe in nature?"

That question is ABC in embryo form.

Early Likelihood-Free Ideas (Before the Name "ABC")

Long before the term ABC was coined, researchers were already doing the following:

- Propose parameters from a prior
- Simulate data from the model
- Keep parameters that produce "reasonable" data

These procedures were:

- Ad hoc

- Domain-specific
- Justified heuristically, not formally

Yet they were Bayesian in spirit:

- Prior → simulation → posterior-like filtering

Chapter 2 emphasizes that **ABC formalized existing scientific practice**, rather than inventing something alien.

The Key Conceptual Leap: Conditioning on Similarity, Not Equality

Early methods implicitly conditioned on events like:

> "The simulated data look similar to the observed data"

This is subtle but profound.

In classical Bayesian inference: $\pi(\theta \mid y_{\text{obs}})$

In proto-ABC: $\pi(\theta \mid y \approx y_{\text{obs}})$

This shift replaces exact conditioning with **neighborhood conditioning**. The chapter stresses that this is not a violation of Bayesian logic—it is a **change in what we condition on**.

The Naming of ABC (Early 2000s)

The term **Approximate Bayesian Computation** crystallized in the early 2000s, when researchers began to:

- Explicitly acknowledge approximation
- Study the effect of tolerances
- Analyze the resulting posterior mathematically

Once named, ABC became:

- A research object
- A class of algorithms
- A bridge between statistics and scientific simulation

Naming mattered because it allowed:

- Theoretical analysis
- General-purpose algorithms
- Cross-disciplinary transfer

Retrospective Insight: ABC as Predictive Conditioning

A key historical reinterpretation presented in this chapter is:

> ABC is Bayesian inference using the **prior predictive distribution** as the engine of learning.

Rather than conditioning on a likelihood value, ABC conditions on:

- Predictive agreement between simulated and observed data

This reframes ABC as:

- Model criticism
- Predictive adequacy filtering
- Bayesian learning via simulation

Seen this way, ABC is less radical than it first appears.

Why ABC Was Inevitable

The chapter makes a strong implicit argument: ABC was not optional.

Once science embraced:

- Agent-based models
- Stochastic simulators
- Complex mechanistic systems

Likelihood-based inference alone could not survive.

ABC emerged because:

- Simulation scaled faster than algebra
- Realism outran analytic tractability
- Scientists refused to abandon Bayesian reasoning

What History Teaches Us About Using ABC Today

From the historical arc, several lessons emerge:

1. ABC is a **last resort**, but often the *only honest resort*
2. Approximation should be acknowledged, not hidden
3. Summary statistics are a scientific judgment, not a technical nuisance
4. ABC lives at the intersection of statistics and domain knowledge

This explains why ABC thrives in:

- Genetics
- Ecology
- Climate science
- Epidemiology
- Financial market simulation

Big Picture Synthesis

Chapter 2 reframes ABC as:

- A formalization of long-standing scientific inference
- A return to predictive reasoning
- A response to computational reality, not a rejection of theory

ABC is not "Bayes without likelihoods." It is **Bayes with humility about what can be computed**.

Chapter 3: Regression Approaches for ABC

The Problem That Forces Regression into ABC

By the end of Chapter 1, ABC is already in trouble.

You learned that:

- Small tolerance (h) → accurate posterior, terrible acceptance rate
- Large tolerance (h) → usable acceptance rate, biased posterior
- High-dimensional summaries → both problems at once

Chapter 3 asks a sharp question:

> If we are *already* accepting imperfect simulations, can we **correct** for their imperfection statistically?

Regression-adjusted ABC is the first serious answer.

The Key Insight: Accepted Samples Still Contain Information

In basic ABC, once a simulated dataset is "close enough," we:

- Keep the parameter draw
- Throw away how close it actually was

This is wasteful.

Accepted samples satisfy: $s^{(i)} \approx s_{\text{obs}}$ but not exactly.

Regression ABC treats this as **measurement error**:

- The parameter $\theta^{(i)}$ is associated with a summary statistic error
- If we understand how θ varies with s, we can adjust back to s_{obs}

This is the conceptual leap of the chapter.

The Basic Regression Model

Suppose we have accepted ABC samples:

$$(\theta^{(i)}, s^{(i)}), \quad i = 1, \dots, N$$

Model the relationship locally as:

$$\theta^{(i)} = a + B(s^{(i)} - s_{\text{obs}}) + \varepsilon^{(i)}$$

This is not a global model of the posterior. It is a **local linear approximation near the observed summaries**.

The corrected parameter is:

$$\theta^{*(i)} = \theta^{(i)} - B(s^{(i)} - s_{\text{obs}})$$

This adjustment answers the counterfactual:

> "What would this parameter draw have been if its simulated summary had exactly matched the observed one?"

Why This Works (Intuition)

Think geometrically.

- ABC samples form a cloud in (s, θ)-space
- The observed summary s_{obs} is a vertical slice
- Regression estimates the slope of the cloud
- Adjustment projects points onto the slice

The tighter the local linear approximation, the better the correction.

This is **bias correction**, not variance reduction.

Weighted Local Regression (Kernel Matters Again)

Not all accepted samples are equally informative.

Samples closer to s_{obs} should matter more.

So regression is performed with kernel weights:

$$w^{(i)} = K_h(|s^{(i)} - s_{\text{obs}}|)$$

This ensures:

- Locality
- Stability
- Compatibility with ABC's kernel interpretation

The regression is therefore *conditional on closeness.*

Nonlinear and Heteroskedastic Extensions

Linear regression is often not enough.

Chapter 3 introduces:

- Polynomial regression
- Transformation of parameters (e.g., log-scale)
- Models where variance depends on s

General form:

$$\theta = m(s) + \sigma(s)\varepsilon$$

Regression ABC can:

- Adjust the mean
- Adjust the spread
- Preserve posterior shape better than raw ABC

This is crucial when posteriors are skewed or constrained.

What Regression ABC Is Actually Approximating

Regression ABC implicitly estimates:

$$\mathbb{E}[\theta \mid s = s_{\text{obs}}]$$

But remember:

- ABC conditions on $s \approx s_{\text{obs}}$
- Regression moves that conditioning *closer* to equality

It does **not** recover the true posterior automatically. It refines an already approximate object.

When Regression ABC Helps — and When It Fails

Helps when:

- Summary statistics are informative but imperfect
- Relationship between θ and s is smooth
- Tolerance must be moderately large for feasibility

Fails when:

- Summaries are weak or insufficient
- Relationship is highly nonlinear locally
- Dimensionality is high relative to accepted samples

Regression cannot create information that summaries do not contain.

Regression ABC and the Bias–Variance Tradeoff

This chapter reframes ABC error as:

- **Tolerance bias** (large h)
- **Regression bias** (model misspecification)
- **Monte Carlo variance**

Regression ABC:

- Reduces tolerance bias
- Slightly increases variance
- Risks model-induced bias

This is not a free lunch—it is a **controlled trade**.

Conceptual Takeaway

Regression ABC changes the role of simulation:

Simulation is no longer just a filter. It becomes **training data** for a local statistical correction.

This is the beginning of a broader theme in ABC:

> Use simulation to *learn* the likelihood structure indirectly.

Later chapters will push this further.

How Chapter 3 Fits into the ABC Arc

- Chapter 1: ABC as kernel-smoothed likelihood
- Chapter 2: ABC as predictive conditioning
- **Chapter 3: ABC as regression-corrected inference**
- Chapter 4: Embedding ABC inside MCMC and SMC
- Chapter 5: Designing summaries that make regression meaningful

Regression ABC is the first step toward **modern, scalable likelihood-free inference**.

Final Thought

Regression ABC is not cosmetic. It is a recognition that once approximation is unavoidable, **statistical correction is obligatory**.

ABC stopped being “accept/reject” here. It started becoming **statistical learning via simulation**.

Chapter 4: ABC Samplers

Why Samplers Matter: Rejection ABC Is a Dead End

By now, one fact is unavoidable:

Plain rejection ABC does not scale.

Even with:

- Relaxed tolerances
- Regression adjustment
- Carefully chosen summaries

Rejection sampling still:

- Wastes simulations
- Dies in moderate dimensions
- Becomes unusable for realistic models

Chapter 4 addresses the survival question:

> How do we embed ABC inside *efficient Bayesian samplers* without reintroducing the likelihood?

The answer: **ABC–MCMC and ABC–SMC**.

The Core Difficulty: ABC Breaks Standard MCMC

Standard MCMC relies on ratios like:

$$\frac{p(y_{\text{obs}} \mid \theta')\pi(\theta')}{p(y_{\text{obs}} \mid \theta)\pi(\theta)}$$

But ABC replaces $p(y_{\text{obs}} \mid \theta)$ with:

$$p_{\text{ABC}}(y_{\text{obs}} \mid \theta) = \int K_h(|s - s_{\text{obs}}|)p(s \mid \theta)ds$$

This quantity:

- Is unknown
- Is noisy
- Depends on simulation

So how can a Markov chain be constructed?

The Key Trick: Augment the State Space

ABC–MCMC avoids likelihood ratios by **expanding the target distribution**.

Instead of sampling just θ, sample:

$$(\theta, y)$$

Target joint density:

$$\pi_{\text{ABC}}(\theta, y \mid y_{\text{obs}}) \propto K_h(|y - y_{\text{obs}}|)p(y \mid \theta)\pi(\theta)$$

This restores something crucial:

- The acceptance ratio can be computed
- Likelihood simulation replaces likelihood evaluation
- Detailed balance holds

This move is conceptually elegant: **the simulator becomes part of the Markov state**.

ABC–MCMC Algorithm (Conceptual Flow)

Each iteration:

1. Propose $\theta' \sim q(\theta' \mid \theta)$
2. Simulate $y' \sim p(y \mid \theta')$
3. Accept with probability:

$$\alpha = \min\left(1, \frac{K_h(|y' - y_{\text{obs}}|)\pi(\theta')q(\theta \mid \theta')}{K_h(|y - y_{\text{obs}}|)\pi(\theta)q(\theta' \mid \theta)}\right)$$

No likelihood appears. Only simulation and kernel weights.

Why ABC–MCMC Is Both Powerful and Dangerous

Strengths

- Uses local proposals
- Explores parameter space efficiently
- Avoids massive rejection waste

Weaknesses

- Sticky chains when tolerance is small
- Poor mixing in high dimensions
- Kernel weights can collapse acceptance rates

This is the same pathology as standard MCMC—now amplified by simulation noise.

ABC–MCMC is efficient *only if the proposal is excellent.*

Sequential Monte Carlo (SMC): A Different Philosophy

SMC attacks the problem differently.

Instead of one chain, we evolve a **population of particles** through a sequence of distributions:

$$\pi_{\text{ABC},t}(\theta \mid y_{\text{obs}})$$

Each stage uses:

- A tighter tolerance (h_t)
- Reweighting instead of rejection

- Resampling to focus computation

This is **annealing in data space**, not parameter space.

ABC–SMC Algorithm (Conceptual View)

Initialize:

- Sample $\theta^{(i)} \sim \pi(\theta)$
- Simulate $y^{(i)}$
- Keep particles with $|s^{(i)} - s_{\text{obs}}| \leq h_1$

Iterate:

1. Perturb particles via a kernel
2. Simulate new datasets
3. Accept if within tolerance (h_t)
4. Reweight and resample
5. Decrease tolerance

Particles gradually migrate toward regions that better reproduce the observed data.

Why ABC–SMC Is Often Superior

ABC–SMC:

- Adapts automatically
- Avoids stuck chains
- Uses simulations efficiently
- Handles multimodality better

It also exposes a deep insight:

> ABC inference is easier when we **learn the posterior gradually**, not all at once.

This mirrors:

- Tempered posteriors
- Data assimilation
- Sequential learning systems

Tolerance Schedules Are the Real Algorithm

Chapter 4 emphasizes something subtle:

The sampler is not the hard part. **Choosing the tolerance schedule is.**

Too fast:

- Particle collapse
- Weight degeneracy

Too slow:

- Wasted computation
- Weak posterior concentration

Tolerance selection becomes a **design decision**, not a tuning nuisance.

Relationship to Regression ABC

Regression adjustment can be applied:

- After ABC–MCMC
- After ABC–SMC
- Within SMC populations

This produces hybrid methods:

- Sampling + correction
- Exploration + bias reduction

ABC becomes a *pipeline*, not a single algorithm.

Big Conceptual Shift Introduced by Chapter 4

Before this chapter:

- ABC was a filter

After this chapter:

- ABC becomes a **Monte Carlo inference engine**

Simulation is no longer just a test—it is a **driver of posterior exploration**.

How Chapter 4 Fits the Bigger Arc

- Chapter 1: ABC as kernel-smoothed likelihood
- Chapter 2: ABC as predictive conditioning
- Chapter 3: ABC as regression-corrected inference
- **Chapter 4: ABC as scalable Monte Carlo**
- Chapter 5: Making summaries worthy of all this machinery

This chapter is where ABC becomes *practical*.

Final Thought

ABC samplers teach a profound lesson:

> When likelihoods are unavailable, **the geometry of simulation replaces the algebra of probability**.

Markov chains and particle systems don't need likelihoods. They need **comparisons**, **weights**, and **movement**.

That realization is what makes likelihood-free Bayesian inference possible at scale.

ABC in Evolution and Ecology

The Core Problem ABC Solves

In a lot of evolution + ecology models, you can **simulate** data easily, but you can't **evaluate the likelihood** $p(x \mid \theta)$ without summing over tons of hidden states (genealogies, latent population trajectories, agent-based microstates, etc.).

Beaumont calls out that asymmetry explicitly: *forward simulation is easy; inverse inference is hard.*

ABC is "likelihood-free" Bayesian inference: you replace explicit likelihood evaluation with **simulation + comparison**.

The Bayes Backbone (What We'd Do If the Likelihood Were Nice)

Standard Bayes:

$$p(\theta \mid x) \propto p(x \mid \theta)\pi(\theta)$$

And the pain point is that normalizing constant (the marginal likelihood) is a nasty integral in realistic models.

ABC's move: **treat the simulator as the likelihood engine**, and approximate the conditional distribution by keeping simulations "close enough" to observed data.

The Prototype ABC Algorithm (Rejection ABC)

When data are high-dimensional, ABC usually compares **summary statistics** $S(x)$ rather than raw data x. The basic rejection ABC idea is:

1. Sample $\theta_i \sim \pi(\theta)$
2. Simulate $x_i \sim p(x \mid \theta_i)$ (via your model simulator)
3. Compute distance $\rho(S(x_i), S(y))$
4. Accept θ_i if distance $\leq \varepsilon$

That's the "stone axe" version. It works, but it gets wrecked by the **curse of dimensionality**: as the dimension of $S(\cdot)$ grows, almost nothing lands near the target.

So the modern ABC story becomes: **how do we (a) sample more efficiently and/or (b) extract more information from simulations we already paid for?**

The "Big Three" ABC Families (Beaumont's Taxonomy)

Beaumont organizes ABC into three main approaches:

(1) Regression / conditional density estimation, **(2) MCMC-ABC**, **(3) SMC-ABC**.

Regression-Adjusted ABC (Conditional Density Estimation)

Key idea: rejection ABC is secretly doing **conditional density estimation**: you sample from the joint $(\theta, S(x))$ and try to estimate $p(\theta \mid S(y))$.

Beaumont et al. (2002) style regression-ABC:

- Simulate lots of pairs (θ_i, S_i)
- Weight points closer to $S(y)$ more heavily (kernel weighting)
- Fit a local regression $\theta \approx a + b^\top S$
- **Adjust** θ_i as if you had simulated exactly at $S(y)$

Beaumont describes a specific kernel-weighted local linear setup and the adjustment step:

$$\theta_i^* = \theta_i - \widehat{E}[\theta \mid S(x_i)] + \widehat{E}[\theta \mid S(y)]$$

so you "slide" simulated parameters toward what they would be if their summaries matched the observed summaries.

Pros

- You can use a larger tolerance (ε) and still get decent posteriors.
- Often dramatically improves efficiency vs pure rejection.

Cons / Gotchas

- If $S(y)$ lies near/outside the prior predictive cloud, regression becomes extrapolation (bad news).
- Linear adjustment can misbehave near parameter bounds (can push adjusted draws outside the prior support).

Modern upgrades: nonlinear regression (e.g., neural nets) and heteroscedastic adjustment that also scales residual variance—not just the mean.

MCMC-ABC (Sampling Closer to Posterior Than the Prior)

Problem with rejection/regression: you draw θ from the prior even when the posterior is tiny.

MCMC-ABC (Marjoram et al. 2003 style):

- Propose θ' from a kernel around current θ
- Simulate data under θ'
- Accept move only if simulated summaries are within tolerance ε, then do the usual Metropolis-Hastings accept/reject based on priors and proposal symmetry (since there's no explicit likelihood).

Pros

- You spend simulation budget exploring where the posterior actually lives.

Cons

- Mixing can be terrible, especially in tails: acceptance depends on the event "simulation lands within ε of observed summaries," which can be rare. Beaumont notes practical fixes like annealing ε during burn-in or letting ε vary as part of the chain.

SMC-ABC (Sequential Monte Carlo / Populations of Particles)

SMC-ABC is the "swarm intelligence" version:

- Start with a big tolerance ε_1 (easy acceptance)
- Then gradually tighten ε_t
- At each stage, **resample + perturb** particles (parameter samples) and reweight by importance weights to correct for not sampling from the prior anymore

Beaumont emphasizes SMC-ABC's practical appeal: it can dramatically reduce wasted simulations when the posterior is narrow relative to the prior.

But there's a "brick wall" effect: once you get very tight tolerances, acceptance rates can collapse without much gain. A common move is to add regression adjustment at the final stage.

ABC Model Choice: Choosing Between Stories, Not Just Parameters

ABC can do Bayesian model choice too: compare models $\mu_1, \mu_2, \ldots$ via posterior model probabilities or Bayes factors (ratios of marginal likelihoods).

A classic ABC trick Beaumont highlights:

- If you run ABC under each model using the **same tolerance** and summaries, then **acceptance rates** are proportional to each model's marginal likelihood.
- So the ratio of acceptance rates estimates the Bayes factor.

Regression-based model choice:

- Treat model index as categorical and regress model indicator on summaries (logistic/multinomial regression) to estimate $P(M = \mu_i \mid S(y))$.

Warning label: model choice is often *more fragile* than parameter inference because insufficient summaries can bias Bayes factor estimates. Beaumont stresses proper priors matter and that small tolerance rejection can be needed for accuracy.

Hierarchical ABC (HABC): “Many Similar Units” Problems

Evolution/ecology loves hierarchical structure: many loci, many populations, many sites.

Hierarchical Bayesian model idea:

- Unit-level parameters θ_i share hyperparameters ϕ
- You infer $\phi + \theta_i$ jointly, enabling “borrowing strength.”

ABC trouble: naïvely, you’d need tons of unit-specific summary stats → dimensionality explosion.

Beaumont reviews work that separates:

- **Symmetric summaries** (order-invariant, like means/variances across units) to infer hyperparameters,
- Plus unit-specific summaries only when inferring a particular unit parameter.

The meta-lesson: in hierarchical ABC you often need a **two-stage design** to avoid drowning in summaries.

Summary Statistics: The “Secret Boss Fight” of ABC

ABC quality lives or dies on $S(\cdot)$.

Ideal target: **Bayes sufficiency**:

$$p(\theta \mid x) = p(\theta \mid S(x))$$

but in real life you approximate.

Main tensions Beaumont highlights:

- **More summaries** capture more information… but **destroy acceptance** (curse of dimensionality).
- Some summaries may be irrelevant, causing you to reject good simulations because of noise in uninformative directions. Weighted distances can help (hyper-ellipse idea).
- Dimension reduction: PLS (partial least squares) uses correlation with parameters; PCA uses variance structure.
- Selecting summaries: Joyce & Marjoram style “leave-one-out summary importance” by checking posterior changes when removing a statistic.

A practical way to think about it:

- Summaries are your **data contract** between reality and simulator.

- Distance ρ is your **loss function**.
- ε is your **approximation dial**.

Model Checking: ABC Forces You to Look at Predictive Distributions

Beaumont stresses validation is not a side quest—ABC relies on it.

Two big diagnostics:

Prior Predictive Checks

Simulate from the prior, compute $S(x)$, and see whether observed $S(y)$ sits inside that cloud. If it's an outlier, your model is probably misspecified—and regression adjustment becomes especially risky.

Posterior Predictive Checks

Sample θ from the (approximate) posterior, simulate new datasets, and compare summaries to observed.

Also: Beaumont notes that "coverage tests" should often be done under repeated sampling from the **prior**, because Bayesian credible intervals are calibrated that way; testing at a single fixed parameter value can mislead you about ABC approximation error.

Applications Beyond Population Genetics: Why Ecologists Should Care

Beaumont's review is explicit: ABC started in population genetics, but its shape fits lots of evolution/ecology problems.

Systems Biology (ODE/SDE Time Series, Networks, ABMs)

Many models are dynamical systems (ODEs / SDEs) with time-series data; SMC-ABC is popular here, using distances between observed and simulated trajectories.

He also mentions:

- Model choice for competing transmission models (via SMC-ABC)
- Early ABC work on agent-based models (ABMs) in cancer cell differentiation using rejection ABC
- Protein interaction networks where you simulate graphs and compare topology summaries, using MCMC-ABC to discriminate network evolution mechanisms

Ecology & Paleontology

Examples include:

- Lotka–Volterra abundance time series (via SMC-ABC examples)
- Species abundance models (e.g., sequential broken stick), using summaries like number of species and frequencies of top species
- Neutral ecological model inference for speciation/immigration
- Extinction rates from phylogenies using MCMC-ABC
- Fossil count data + branching process models using ABC variants

The common structure: **simulators are easy; likelihoods are brutal**—so ABC becomes a practical "first serious inference step."

The Big Philosophical Punchline (Beaumont's "Creative Tension")

Beaumont argues ABC is especially useful in the **exploratory phase**—when the model is still evolving, ad hoc, and you want a Bayesian workflow without spending months deriving a tractable likelihood. Agent-based modeling is singled out as a particularly compelling domain because likelihoods are often hopeless there.

A Practical ABC Workflow

If you were doing ABC on, say, an ecological dynamical model:

1. **Define simulator**: given θ, generate synthetic dataset x.
2. **Pick summaries** $S(x)$: start with domain-motivated statistics; keep them small but informative.
3. **Pick distance** ρ: scale summaries; consider weights if some are noisy/uninformative.
4. **Choose algorithm**:

    - Quick prototype: rejection + regression adjustment
    - Hard posterior / narrow region: SMC-ABC
    - Tricky posterior geometry: MCMC-ABC (careful with mixing)

5. **Do prior predictive check** (is your observed summary even plausible?)
6. **Run inference**, then **posterior predictive check** for adequacy.
7. If model choice: be extra paranoid; ensure same summaries across models, and validate.

Closing Thought

ABC is a bargain with the universe:

- You trade **exactness** (explicit likelihood) for **generality** (any simulator).
- You pay with **design responsibility**: summaries, distance, tolerance, diagnostics.
- When done carefully, it turns "I can simulate this but can't analyze it" into "I can do Bayesian inference anyway."

ABC-SMC and Data Assimilation

Here's the clean mapping: **ABC–SMC is a particle filter / SMC sampler where the "likelihood" is replaced by a kernel measuring discrepancy between simulated and observed summaries.** Once you see that, it snaps into place as a *data assimilation filter with an implicit/approximate observation operator*.

The Shared Skeleton: Mutation → Correction → Selection

Data assimilation particle filters and SMC samplers share the same three-step loop:

1. **Mutation / forecast**: push particles forward with a proposal/transition kernel
2. **Correction / analysis**: reweight particles by how well they explain observations
3. **Selection / resampling**: resample to fight weight degeneracy

The ABC–SMC chapter in the handbook literally describes SMC samplers in these terms—mutation, correction, selection—and notes these are "particles" and "particle filters / interacting particle systems."

So the assimilation template is already sitting there.

What Changes in ABC–SMC: The "Observation Likelihood" Becomes a Discrepancy Kernel

In a standard particle filter for a state-space model:

- State evolution: $x_t \sim p(x_t \mid x_{t-1})$
- Observation model: $y_t \sim p(y_t \mid x_t)$
- Weights: $w_t^{(i)} \propto w_{t-1}^{(i)} p(y_t \mid x_t^{(i)})$

In **ABC–SMC**, we often aren't filtering latent states (x_t); we're sampling **parameters** θ (or augmented ($\theta, \text{latent stuff}$)). But the correction step looks the same if you treat the discrepancy as an approximate likelihood:

$$w^{(i)} \propto w^{(i)} K_h(\rho(S(x^{(i)}), S(y_{\text{obs}})))$$

That is exactly the ABC "kernel likelihood" view: ABC defines an approximate likelihood by integrating a kernel against the simulator output. Chapter 4's ABC–SMC discussion is explicitly built around **a sequence of ABC targets defined by decreasing tolerance/bandwidth** (an "annealing"/reverse-annealing schedule).

Assimilation translation:

- Observation operator: $H(\cdot)$ is "compute summaries" ($S(\cdot)$)
- Observation noise: "tolerance" (h) (or ε)
- Likelihood: replaced by kernel $K_h(\rho(\cdot, \cdot))$

So ABC–SMC is a filter where you can't write $p(y \mid x)$, but you can still score particles by how close their simulated observations are.

The "Time Index" in ABC–SMC Is Not Physical Time (Usually) — It's an Annealing Path

This is the most important conceptual mismatch to resolve.

- In data assimilation, index t is **real time** (new data arrives).
- In ABC–SMC, index m is typically an **algorithmic time**: you move through a sequence of target distributions by tightening tolerance h_m.

The handbook's ABC–SMC algorithm uses a sequence $h_0 \geq h_1 \geq ... \geq h_M$ and explicitly calls this annealing-like, with automated schedules to avoid particle collapse.

Assimilation analogy: ABC–SMC's "analysis steps" are like repeatedly assimilating the *same observation* but with **shrinking observation noise**. Early stages = very noisy observation (loose match); late stages = precise observation (tight match).

That's why it feels like tempering / continuation methods in filtering.

Mapping the Equations: Particle Filter ABC–SMC

Here's a literal dictionary.

Data assimilation particle filter

- Forecast: $x_t^{(i)} \sim p(x_t \mid x_{t-1}^{(i)})$
- Weight: $w_t^{(i)} \propto w_{t-1}^{(i)} p(y_t \mid x_t^{(i)})$
- Resample if ESS low

ABC–SMC sampler (parameter filtering)

- Mutation: $\theta_m^{(i)} \sim g_m(\theta \mid \theta_{m-1}^{(a(i))})$ (perturbation kernel)
- Simulate pseudo-data: $x_m^{(i)} \sim p(x \mid \theta_m^{(i)})$
- Weight: $w_m^{(i)} \propto w_{m-1}^{(a(i))} K_{h_m}(\rho(S(x_m^{(i)}), S(y_{\text{obs}})))\times$ (importance correction terms)

So: same structure; different "likelihood."

Where EnKF Fits (and Where It Doesn't)

An **Ensemble Kalman Filter (EnKF)** is also mutation → correction → resampling-ish, but correction is *Gaussian-linear* and updates via covariances.

ABC–SMC differs:

- EnKF assumes an explicit observation ($y = H(x) + \eta$) with Gaussian-ish errors and uses linear updates.
- ABC–SMC does not assume linear/Gaussian; it uses **nonparametric weighting** via the discrepancy kernel.

However, conceptually you can think of ABC–SMC as the "fully nonlinear, non-Gaussian" cousin, closest to a particle filter. If your summaries are roughly linear and noise behaves nicely, EnKF is cheaper; if not, ABC–SMC/particle filters are the robust hammer.

Practical "Data Assimilation" Interpretations That Are Actually Useful

Interpretation A: ABC–SMC = repeated assimilation with decreasing observation noise

Loose tolerance → broad posterior; tight tolerance → concentrated posterior. This is exactly what the adaptive tolerance schedules are trying to manage: keep ESS from collapsing while tightening.

Interpretation B: summary statistics are your observation operator

Bad summaries = bad observation operator → filter locks onto the wrong manifold.

Interpretation C: discrepancy metric is your likelihood model choice

Your $\rho(\cdot,\cdot)$ and kernel K_h are effectively specifying the measurement model. This is why distance scaling/whitening matters as much as prior choice.

If You're Thinking "Data Assimilation Model Builder": What to Steal from Each Side

From data assimilation → improve ABC–SMC:

- Adaptive resampling thresholds via ESS (already central in Algorithm 4.8)
- Better mutation kernels (analogous to better proposal models / localization ideas)
- Tempering schedules (ABC tolerance schedule is basically tempering)

From ABC–SMC → improve data assimilation:

- Likelihood-free correction when observation likelihood is unknown but simulation is possible
- Use of summary statistics when raw observation comparison is impossible/high-dim