# Numerical Optimization in System Dynamics Models: Calibration and Optimal-Policy Search in World3

Ian Costa-Alves* Nicolas Gourdain†
François Gallard‡

June 30, 2026

System Dynamics has long been used to compare policy scenarios by hand, but less so for searching them for the policy that is best under a stated objective, even though optimization is routine for dynamical systems in other fields. This paper brings numerical optimization to the World3 model, which couples population, capital, agriculture, resources, and pollution to simulate the global socio-environmental system. Choosing a policy is treated here as a wicked problem: the dynamics of growth, depletion, and pollution are considered as quantifiable, but the choice of objective is contested, so instead of assuming a single one, the trade-off between maximizing human welfare and minimizing ecological footprint is traced. The model is re-implemented in JAX, first to make it differentiable for gradient-based optimization but also to gain the vectorization and just-in-time compilation that make its repeated evaluation fast, and it is coupled to the GEMSEO optimization framework. Three applications are demonstrated: calibration against historical data, single-objective policy optimization, and the Pareto front between the two objectives. First, the calibration exposes shortcomings of some modeling assumptions, as well as on the suitability of some data proxies used. Secondly, the trade-off between objectives is traced and the most policy-relevant solutions are between the single-objective optima, all of which find solutions that under-invest in industrial capacity relative to World3 Business-As-Usual scenario. The use of gradient-based optimization with automatic differentiation embeded in a multidisciplinary optimization framework enables the resolution of these nonlinear, high-dimensional problems in minutes on a single CPU core while reducing implementation efforts.



**Keywords:** Numerical Methods; Calibration; Optimization; World3

* Design and Operation of Aerospace Vehicles Department, ISAE-SUPAERO
† Aerodynamics, Energetics, and Propulsion Department, ISAE-SUPAERO
‡ Center of Competence on Multidisciplinary Design Optimization, IRT Saint Exupéry

# 1 Introduction

How far can the world's population and economy keep growing on a finite planet, and what would it cost to steer them onto a safer path? *The Limits to Growth* [1] was the first study to pose this question with a global computational model that couples human activity to its environmental limits. This very question which is still relevant half a century later and presently drives the development of Integrated Assessment Models that inform international climate and energy policy [2].

The model behind the study, World3 [3], uses mathematical models and computer simulation to trace long-term trajectories of economic and population growth, and their associated agricultural production, resource consumption and pollution generation. Its scenarios, revisited in later updates [4], explore eleven possible futures set apart by different policy assumptions, most of which end up in overshoot and sharp decline of human welfare within the twenty-first century, unless population and economic growth are stabilized, and empirical comparisons find that its Business-as-Usual (BAU) path has tracked the observed record reasonably well through 2020 [5, 6]. World3 is an example of a System Dynamics model, a modeling style created at MIT in the 1950s by Jay Forrester to study how feedback loops between the parts of a system shapes the behaviour of the whole [7], the mathematical formulation of such models, and the structure of World3, are introduced in Section 2.

On one hand, in the scenario-building exercise the analyst sets the policy levers, runs the simulation, and reads off the outcome, meaning the choice of what to pursue is made by hand and most of the policy space is never examined. Numerical optimization, on the other hand, reverses this by formulating the policy problem as the maximization or minimization of a functional objective, where an optimizer searches the policy space for the solutions, meaning solutions follows from the chosen objective rather than from the policy levers. The approach is well established in Integrated Assessment Modeling, where DICE, among the first climate-economy IAMs [8–10], maximizes discounted per-capita consumption and derives its optimal carbon pricing policy directly from that single objective, even though the choice of the objective function and the treatment of inter-generational justice, remains contested [11–13].

Optimization applied to urban and social policy has drawn criticism since the 1970s, on the grounds that these problem are wicked [14], i.e., problems where stakeholders disagree about the very nature of the issues, about possible solutions, and about the values that should guide implementations [15, 16]. Sustainability sits only partly within this difficulty as these as the issues are identifiable, the population and economic growth that increase human welfare are also responsible for increasing environmental footprint and difficult future welfare generation, and can be quantified, using simplified models such as World3. What remains contested is the choice of objective that should guide policies: what should

[1]: Meadows (1974), *The limits to growth: a report for the club of rome's project on the predicament of mankind*

[2]: Intergovernmental Panel On Climate Change (Ipcc) (2023), *Mitigation Pathways Compatible with Long-term Goals*

[3]: Meadows et al. (1974), *Dynamics of growth in a finite world*

[4]: Meadows et al. (2009), *The limits to growth: the 30-year update*

[5]: Turner (2008), *A comparison of The Limits to Growth with 30 years of reality*

[6]: Herrington (2020), *Update to limits to growth: Comparing the World3 model with empirical data*

[7]: Sterman (2009), *Business dynamics: systems thinking and modeling for a complex world*

[8]: Nordhaus (1992), *An Optimal Transition Path for Controlling Greenhouse Gases*

[9]: Nordhaus (2017), *Revisiting the social cost of carbon*

[10]: Barrage et al. (2023), *Policies, Projections, and the Social Cost of Carbon: Results from the DICE-2023 Model*

[11]: Stern (2007), *The Economics of Climate Change: The Stern Review*

[12]: Goulder et al. (2012), *The Choice of Discount Rate for Climate Change Policy Evaluation*

[13]: Schoenmaker et al. (2024), *Which discount rate for sustainability?*

[14]: Rittel et al. (1973), *Dilemmas in a general theory of planning*

[15]: Alford et al. (2017), *Wicked and less wicked problems: a typology and a contingency framework*

[16]: Head (2022), *Wicked Problems in Public Policy: Understanding and Responding to Complex Challenges*

society optimize for? Multi-objective optimization does not settle that question, but it makes the disagreement explicit by finding the trade-off between diverging objectives [17]. This paper takes human welfare and ecological footprint as two such objectives and traces the trade-off between them, so that the cost of favoring one over the other can be read from a curve rather than assumed.

[17]: Zechman et al. (2013), *An evolutionary algorithm approach to generate distinct sets of non-dominated solutions for wicked problems*

The paper makes three contributions:

- The first is a *differentiable*[1] re-implementation of World3 which is the property that lets efficient gradient-based optimizers work. It is built on JAX [18], a numerical-computing library that can differentiate an entire program automatically, and on the GEMSEO framework [19], which organizes the data exchange between the coupled sectors;
- The second is the calibration of all five sectors at once against an open-access historical dataset, adjusting the model constants so that its outputs match the observed record while keeping the cross-sector feedbacks intact, which also reveals which constants the data can and cannot pin down;
- The third, and the focus of the results, is the reformulation of the model as a policy-optimization problem and the tracing of the *Pareto front*[2] between human welfare and ecological footprint as objectives.

1: A differentiable model is one whose system outputs respond smoothly to small changes in its inputs, and where this sensitivity or derivative is computationally inexpensive to calculate.

[18]: Bradbury et al. (2018), *JAX: composable transformations of Python+NumPy programs*

[19]: Gallard et al. (2018), *GEMS: A Python Library for Automation of Multidisciplinary Design Optimization Process Generation*

2: The set of solutions for which neither goal can be improved without the other getting worse.

The paper is organized as follows. Section 2 reviews the background in three parts: the structure of System Dynamics models and of World3, the numerical optimization concepts used here, and the computing techniques that make gradient-based optimization of such models practical. Section 3 presents the calibration of the model against historical data, together with its results. Section 4 presents the policy optimization and the trade-off between the two objectives. Section 5 discusses what the residuals and the front reveal about the model, and Section 6 draw on the conclusions that can be drawn from the analysis.

# 2 Background

## 2.1 System Dynamics

A System Dynamics model is a system of coupled nonlinear ordinary differential equations built from four elements: stocks, flows, table functions, and delays [20].

[20]: Meadows et al. (2009), *Thinking in systems: a primer*

A *stock* is a measurable quantity that accumulates or depletes over time; stacking the model's $n$ stocks into one column vector gives the *state vector* $\mathbf{s} \in \mathbb{R}^n$, the quantity the differential equation integrates. A *flow* adds to or drains a stock and is computed from the current state, from constant parameters $\theta$, and from exogenous inputs; assembling the incoming and outgoing flows gives the right-hand side $\mathbf{f_{ODE}}$, and the Initial Value Problem is solved given that the states at the initial time $\mathbf{s_0}$ are known.

$$\mathbf{s} = \mathbf{s}(t) = [s_1(t), \ldots, s_n(t)]^\intercal \quad (1)$$

$$\begin{aligned} \frac{d}{dt}\mathbf{s} = \dot{\mathbf{s}} = \mathbf{f_{ODE}}(t, \mathbf{s}, \theta) \\ \mathbf{s}(t_0) = \mathbf{s_0} \end{aligned} \quad (2)$$

Beyond the stocks, a model exposes *instantaneous outputs* $\mathbf{y}$, algebraic functions of the current state alone, such as life expectancy

$$\mathbf{y} = \mathbf{f_{output}}(t, \mathbf{s}, \theta) \quad (3)$$

$$\begin{aligned} \mathbf{s}_{(k+1)} &= T(\mathbf{s}_{(k)}, \mathbf{f}_{\mathbf{ODE}(k)}, \Delta t) \\ &= \mathbf{s}_{(k)} + \Delta t\, \mathbf{f}_{\mathbf{ODE}(k)} \end{aligned} \quad (4)$$

or output per capita, that carry no memory and are recomputed at each instant. Simulating the model produces the *state trajectory*, the states at the discretized steps $t_{(k)} = t_0 + k\,\Delta t$. An explicit one-step scheme advances the state through a time-stepping operator $T$, and for simplicity the forward Euler method is used here.

Some relationships are not algebraic but *table functions* that interpolate between tabulated points. The choice of interpolation matters: linear interpolation between breakpoints is continuous but not differentiable there, an obstacle for gradient-based methods that Section 2.4 removes.

*Delays* introduce inertia. A first-order information delay obeys $\dot{d} = (x - d)/\tau$, with $x$ the input, $d$ the delayed output, and $\tau$ the time constant. A delay of order $N$ is a chain of $N$ such equations, each with time constant $\tau/N$, whose last output $d_N$ is the delayed signal. Figure 1 shows how a unit step propagates through delay of orders one to three. The $N$ stage variables are themselves states: standard implementations recompute them from the outputs at each step, but here they are appended to the global state vector $\mathbf{s}$ and integrated alongside the physical stocks, so their temporal evolution is tracked directly rather than reconstructed.

$$\begin{aligned} \frac{d}{dt} d_1 &= \frac{x - d_0}{\tau/N} \\ \frac{d}{dt} d_i &= \frac{d_{i-1} - d_i}{\tau/N}, \; i = 2, \ldots, N \end{aligned} \tag{5}$$

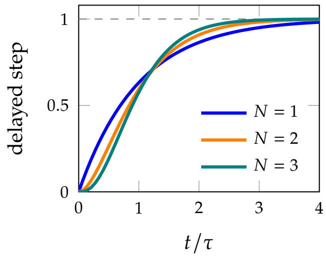


**Figure 1:** Response of an information delay of order $N = 1, 2, 3$ to a unit step, against time normalized by the delay constant $\tau$. Higher order sharpens the lag from a pure exponential toward a sigmoid; each order adds $N$ state components to $\mathbf{s}$.

The behaviour of the whole model follows from its feedback loops: a reinforcing loop, in which a larger stock raises its own inflow, produces exponential growth, while a balancing loop, in which a larger stock raises its own outflow, produces stabilization or decay.

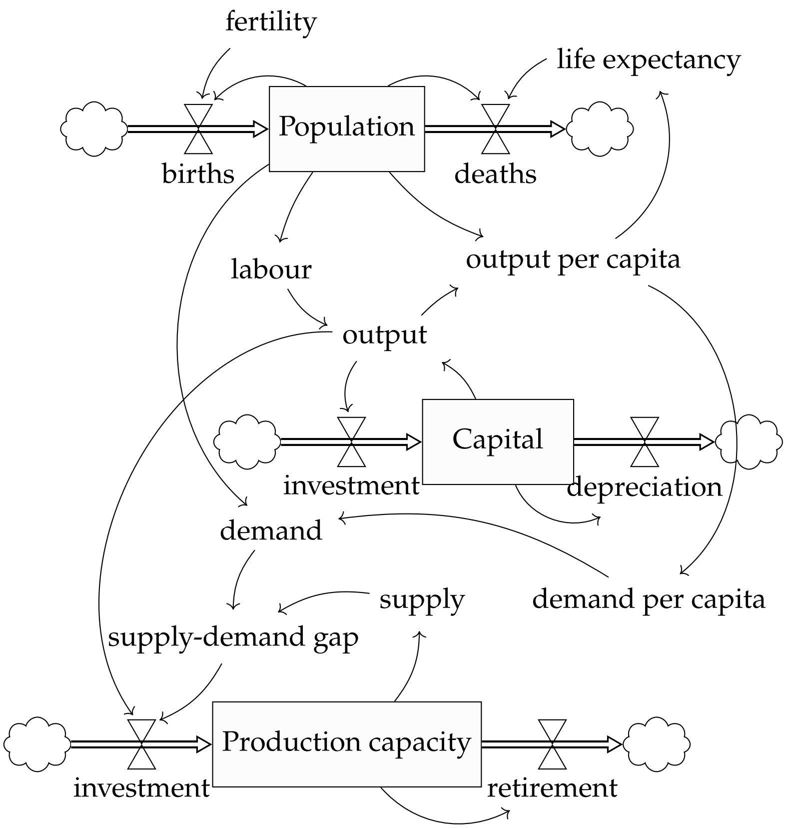


**Figure 2:** Stock-flow diagram of population and capital growth with production of one good in disequilibrium.

For example, Figure 2 shows the stock-flow diagram for a system with population and capital growth, as well as the production of a generic good in disequilibrium (when there is a mismatch between supply and demand). Population growth has a reinforcing loop that dominates over the balancing loop, leading to an exponential growth. The similar can be said for capital accumulation, where the behavior of capital investment dominates over depreciation. The behavior of production capacity, however, is dominated by two balancing loops, meaning its temporal evolution will depend on the strength of population and capital growth (both increasing capacity) relative to its balancing loops (stabilizing capacity growth).

Increasing population increases:

- overall births, leading to a **stronger** population increase
- overall deaths, leading to a **lower** population increase

Increasing capital leads to increased:

- output, investment, and a **stronger** capital increase
- stock subject to depreciation, and a **lower** capital increase

Rising capacity increases:

- retirement rates, and a **lower** capacity increase
- supply, lowers supply-demand gap, and **lower** investments in capacity increase

## 2.2 The World3 model

The World3 model uses similar elements to model the future evolution of society, instead of one single good, the model account for food and services as goods in disequilibrium, and resource as a good in equilibrium. The model couples five sectors of society [4], shown together in Figure 3.

Two reinforcing feedback loops are the at the core of the model, ruling the population and capital growth, which are kept in check by five balancing feedbacks, namely mortality, depreciation, land erosion, pollution damage, and resource depletion. In most scenarios the balancing feedbacks eventually dominate, producing the overshoot-and-collapse trajectories, and which feedback activates first depends on the capital allocation decisions [1]. In total the model contains five sub-sectors:

- Population: the dynamics of demographics is computed using birth and death rates, as well as maturation rates between different population groups. The sector is divided into four sub-populations according to age group (0 to 14 years, 15 to 44, 45 to 64 and more than 65 years), allowing for group-specific birth and maturation rates, as well as life expectancy, which falls due to scarce food, weak health services, and pollution;
- Capital: sector for measuring the industrial and service capital stocks of the society. Labour, industrial capital, and resource usage are combined to generate industrial output (US$ per year), which is then allocated into investments into agriculture, services, industry, or consumption. An increase in industrial output per capita will yield an increased consumption of food, services, resources and consequently higher pollution rates, while an increase in service output per capita yields a decrease in the mortality;
- Agriculture: responsible to estimating food production from investments and its consequences to the land (remaining arable land, erosion, fertility deterioration and regeneration). The cost for food production increases as available arable land reduces, low values of *food per capita* will increase the

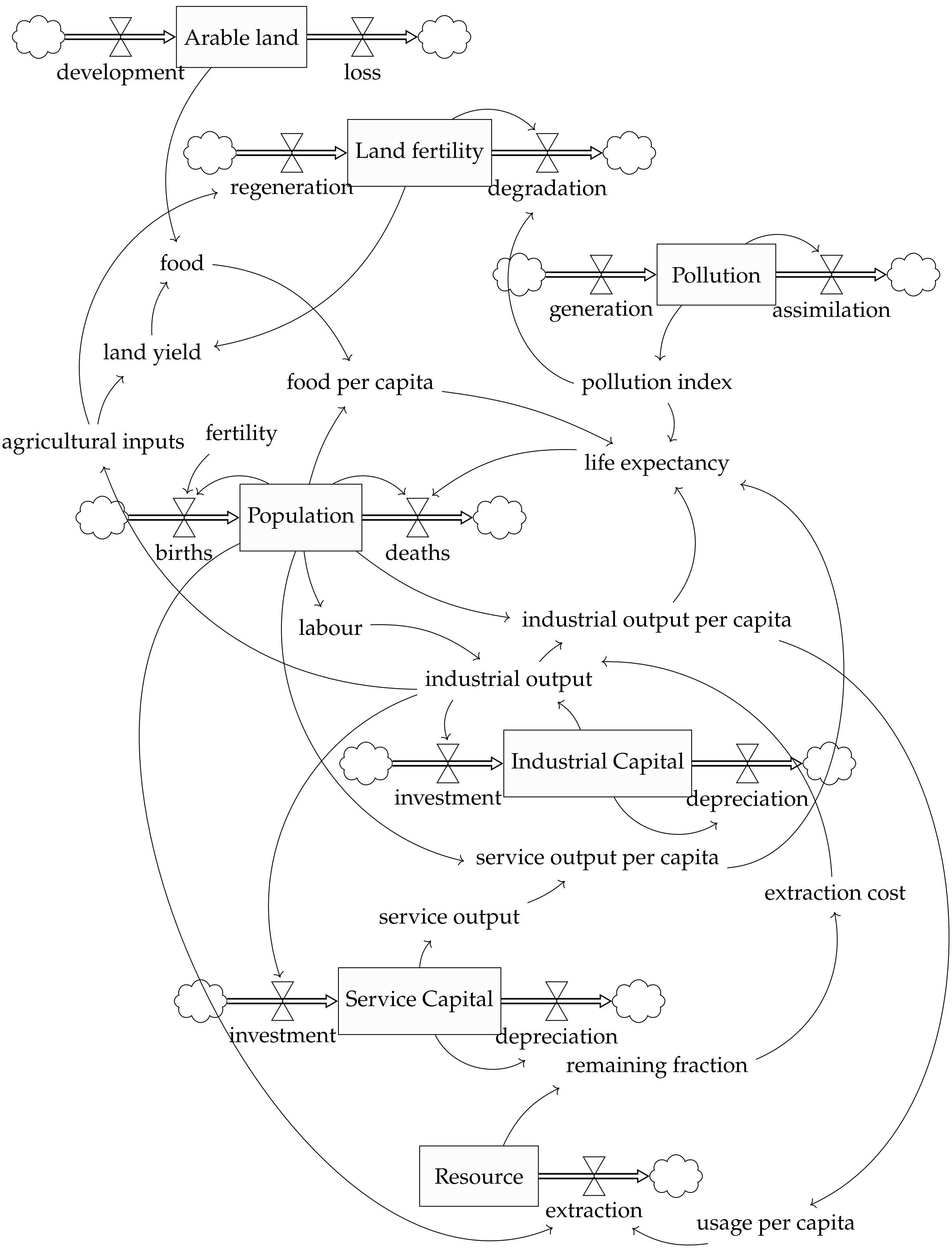


**Figure 3:** Simplified stock-flow diagram of the World3 model.

mortality and the fraction of capital allocated to agriculture (in order to feed the population according to its demands). It is expressed in kg vegetable equivalent (1 kg of animal meat corresponds to 7 kg vegetable equivalent);

- Resources: sector that computes stock, extraction rate and extraction cost of non-renewable resources. Industrial output is the main driver for resource use, as the remaining stock diminishes extraction costs increase. Because there are several different types of resources in reality, it uses a metric normalized by the generalized resource usage rate in 1970;
- Pollution: sector for modeling persistent pollution stock, and pollution assimilation using a half-life that increase as the stock rises. The main activities that generate pollution are industrial and agricultural productions, high pollution yields increases in mortality, in land erosion and fertility degradation. It also uses a metric normalized by the persistent pollution in 1970.

## 2.3 Numerical optimization

Optimization searches in a space of variables for the values that minimize an objective $f_{obj}$. In its simplest case (unconstrained and smooth), the problem can be reduced into a root-finding problem of the gradient of the objective $\nabla f_{obj} = \mathbf{0}$. For problems carrying only bound constraints, the limited-memory quasi-Newton method L-BFGS-B [21] is a standard choice, approximating the Hessian from past gradients to take well-scaled steps.

[21]: Liu et al. (1989), *On the limited memory BFGS method for large scale optimization*

In many practical problems, however, the decision variables may be subject to a set of constraints, and solving these requires formulating a nonlinear programming (NLP) problem (Equation 6), where $f_{obj}$ is the objective, $\mathbf{g}$ and $\mathbf{h}$ are equality and inequality constraints, and the decision variables are bounded. At a constrained optimum the objective gradient no longer vanishes due to the presence of active constraints, and may satisfy some optimality conditions, such as the Karush-Kuhn-Tucker (KKT) system (Equation 7).

$$\begin{aligned} &\min_{\mathbf{x}\in\mathbb{R}^n} f_{obj}(\mathbf{x}) \\ &\text{subject to:} \\ &\quad \mathbf{g}(\mathbf{x}) = \mathbf{0} \\ &\quad \mathbf{h}(\mathbf{x}) \le \mathbf{0} \\ &\quad \mathbf{x_{low}} \le \mathbf{x} \le \mathbf{x_{up}} \end{aligned} \tag{6}$$

The Lagrangian $\mathscr{L}$ adjoins each constraint to the objective through a multiplier, with $\boldsymbol{\lambda}$ free for the equalities and $\boldsymbol{\mu} \ge \mathbf{0}$ for the inequalities. Each multiplier measures how much the optimum would improve if its constraint were relaxed, and gradient-based solvers such as sequential quadratic programming and interior-point methods work by driving the iterate toward a point that satisfies this system.

$$\begin{aligned} \mathscr{L}(\mathbf{x},\boldsymbol{\lambda},\boldsymbol{\mu}) &= f_{obj}(\mathbf{x}) + \boldsymbol{\lambda}^\top \mathbf{g}(\mathbf{x}) + \boldsymbol{\mu}^\top \mathbf{h}(\mathbf{x}) \\ \nabla f_{obj} + \nabla \mathbf{g}\,\boldsymbol{\lambda} + \nabla \mathbf{h}\,\boldsymbol{\mu} &= \mathbf{0} \\ \boldsymbol{\mu} \ge \mathbf{0}, \quad \mu_i h_i &= 0 \end{aligned} \tag{7}$$

When the objective and constraints are linear in $\mathbf{x}$ the problem is a linear program, which is the form most Integrated Assessment Models adopt because their equations are linear by construction. Nonlinear constrained problems must use a different category of algorithms such as gradient-free or gradient-based algorithms.

Gradient-free methods sample the objective at many points and need no derivative information, so they apply to any model and can explore the whole search space, but they offer no guarantee that a solution is optimal and they scale poorly, because the number of model runs needed to converge grows faster than the number of variables [22].

[22]: Martins et al. (2022), *Engineering Design Optimization*

Gradient-based methods instead follow the local slope of the objective downhill and converge in far fewer iterations, but on a non-convex landscape they settle on a point that is best only within its local neighborhood, and they require the objective and constraints to vary smoothly with the variables. There are several approaches within these type of algorithms, sequential quadratic programming, used here in its SLSQP form [23], which solves a sequence of local quadratic approximations subject to linearized constraints. Other approaches for nonlinear problems, such as interior-point methods, first enforce constraints through a barrier term then proceeds onto minimizing the objective.

[23]: Kraft (1988), *A Software Package for Sequential Quadratic Programming*

Because a gradient-based search only returns the optimum of the basin it starts in, a *multi-start* strategy can be used to probe a non-convex landscape: several starting points are drawn, here by an optimized Latin-hypercube design that spreads them through the variable bounds.

The definitions presented so far suffise to formulate the *calibration* as an inverse problem (Figure 4): the model constants $\theta$ are unknown and are chosen by the optimizer, which evaluates the trajectory of state and outputs $\mathbf{y}$, and the objective function is evaluated as the error between data $\mathbf{y}*$ and the simulated outputs in the same periods $\mathbf{y}(\mathbf{t}*)$.

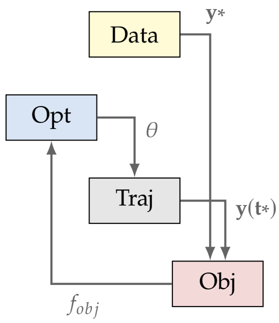


**Figure 4:** Calibration as an inverse problem, shown as the data flow between components. The optimizer (Opt) proposes constants $\theta$; the simulator integrates the trajectory (Traj) and returns the outputs $\mathbf{y}(t^*)$ at the data times; the objective (Obj) scores them against the data $\mathbf{y}^*$ and returns $f_{obj}$ to the optimizer.

Writing $\mathbf{y}(t_{(k)};\theta)$ for the simulated outputs at a given time period $t_{(k)} \in \mathbf{t}*$ and $\mathbf{y}^*(t_{(k)})$ for the data, calibration minimizes a weighted least-squares error over the calibration window.

$$\min_{\theta\in\Theta} \; f_{obj}(\theta) = \sum_k \sum_j w_j \left(y_j(t_{(k)};\theta) - y_j^*(t_{(k)})\right)^2 \tag{8}$$

Minimizing $f_{obj}$ with a gradient-based method calls for its gradient with respect to every constant in $\theta$, which is not straightforward as their influence the residual is indirect: it enters the state integration at each step, and every state feeds the next, so its influence on the accumulated error must be propagated along the whole trajectory (Figure 5). Recovering that sensitivity means either differentiating every model equation with respect to the states and the constants, or estimating it numerically by finite differences or automatic differentiation (Section 2.4).

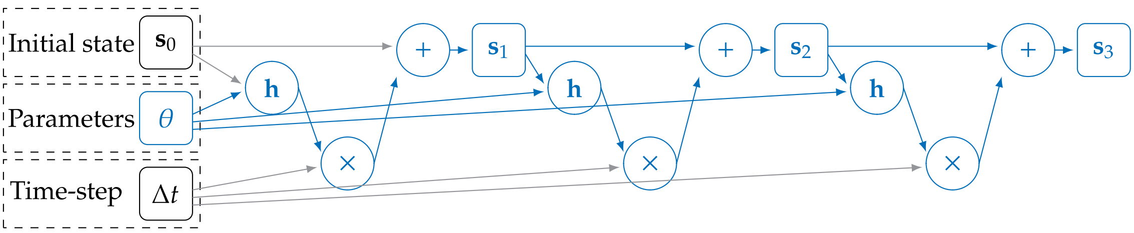


**Figure 5:** Computational graph of each elementary operation of solving an initial value problem using the forward Euler method over three time-steps. The effect of the parameter set

### 2.3.1 Multi-objective optimization

When multiple objectives are considered instead of a single scalar one, the goal of the optimization is not to find a single optimal solution, but a set of optimal compromises between each mono-objective optima.

Some optimization algorithms are made specifically to handle multi-objective problems. Evolutionary algorithms, for instance, are particularly easy to use in multi-objective settings due to the fact that they are population-based and therefore are able to keep a set of solutions instead of a single one [24, 25], but are subject to the common pitfalls of heuristic gradiend-free optimization.

[24]: Deb et al. (2002), *A fast and elitist multiobjective genetic algorithm: NSGA-II*
[25]: Igel et al. (2007), *Covariance Matrix Adaptation for Multi-objective Optimization*

Another approach is to keep single-objective optimization algorithms, and to employ them with methods (Figure 6) that can extend to multi-objective problems, such as:

- Weighted sum: assign weights to each component and use their weighted sum as the objective to maximize. This allows for a linear trade-off between objectives and the Pareto front may be explored by varying the weigths, but only the convex portion of the Pareto front can be obtained [22, Chapter 9];
- Epsilon constraint: keep a single component as the objective, while leaving all others as additional constraints. The Pareto front is then explored by solving multiple sub-optimizations with varying constraint values, but may lead to a non-uniform spacing between the solutions in the objective space [26];
- Normal boundary intersection: addresses the uniform spacing of the Pareto front, by constructing a plane in the objective space containing each of the single component optima, and then to perform sub-optimizations that maximize the normal distance between the solution and the plane [27, 28].

[26]: (1971), *On a Bicriterion Formulation of the Problems of Integrated System Identification and System Optimization*

[27]: Das et al. (1998), *Normal-Boundary Intersection: A New Method for Generating the Pareto Surface in Nonlinear Multicriteria Optimization Problems*
[28]: Shukla (2007), *On the Normal Boundary Intersection Method for Generation of Efficient Front*

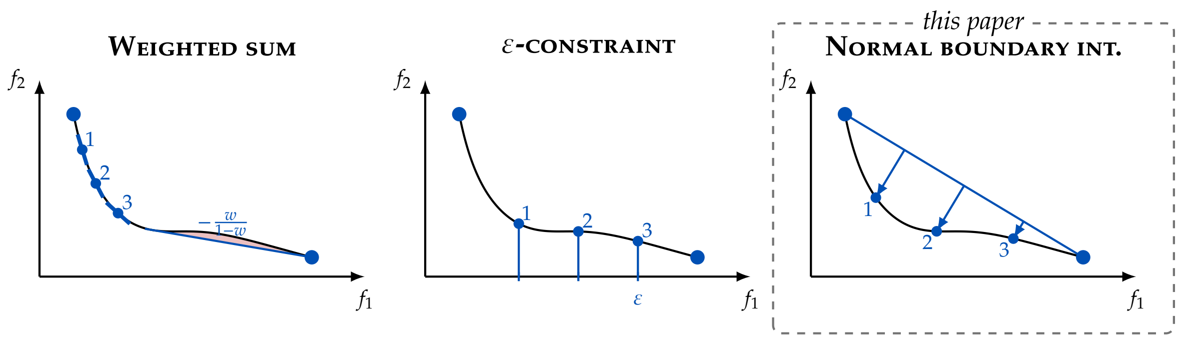


**Figure 6:** Three scalarization strategies for tracing the Pareto front of a bi-objective problem, with the same front shown in each panel. Each method returns the two single-objective optima at the extremities together with $N = 3$ points between them, seven in total. The front is convex except for one bulge in the middle. *Weighted sum* returns the minimizer of $w f_1 + (1 - w) f_2$ for each weight, so it follows the convex hull and skips the non-convex bulge (shaded). *$\varepsilon$-constraint* minimizes $f_2$ subject to equally spaced upper bounds on $f_1$, which gives unevenly spaced points. *Normal boundary intersection* places equally spaced points on the line joining the optima and solves, along the normal at each one, for the boundary point (numbered), tracing the front evenly. All point coordinates are computed inside the figure from the front function. Adapted from [22].

### 2.3.2 Optimal control

When the optimization problem is embedded in a continuous dynamical system, such as when choosing time-dependent policies or calibrating a simulation model, the problem is considered an optimal control problem. Given a system with initial state, the goal is to find the control trajectory $\mathbf{u}(\mathbf{t})$ that minimizes a cost functional with end-point and a path-dependent terms. A notational convention is used throughout: $\mathbf{u}(\mathbf{t})$, with the time argument set in bold, denotes the *whole* control trajectory over $[t_0, t_f]$ that the optimizer chooses, whereas $\mathbf{u}(t)$ denotes its *value* at a single instant $t$; the same applies to the state $\mathbf{s}(t)$ along an integrated trajectory.

$$
\begin{aligned}
&\min_{\mathbf{u}(\mathbf{t})\in\mathcal{U}} f_{obj}(\mathbf{u}(\mathbf{t})) = \phi(\mathbf{s}(t_f)) + \int_{t_0}^{t_f} \ell(\mathbf{s}(t), \mathbf{u}(t), t)\, dt \\
&\quad\text{where:} \\
&\qquad \dot{\mathbf{s}}(t) = \mathbf{f_{ODE}}(t, \mathbf{s}(t), \mathbf{u}(t), \theta) \\
&\qquad \mathbf{s}(t_0) = \mathbf{s_0} \\
&\text{subject to:} \\
&\qquad \mathbf{g}(\mathbf{s}, \mathbf{u}) = \mathbf{0} \\
&\qquad \mathbf{h}(\mathbf{s}, \mathbf{u}) \leq \mathbf{0} \\
&\qquad \mathbf{u_{low}} \leq \mathbf{u} \leq \mathbf{u_{up}}
\end{aligned}
\tag{9}
$$

Two families of methods solve this problem, and they differ in the order in which they optimize decisions and discretize the trajectory over time-steps.

*Indirect methods* operate under the philosophy of "optimize then discretize". They apply the Calculus of Variations and Pontryagin's Minimum Principle [29] to derive first-order necessary conditions analytically, without evaluating the objective directly. The dynamics $\dot{\mathbf{s}} = \mathbf{f_{ODE}}$ are adjoined to the cost as an equality constraint holding at every instant, exactly the role the constraints $\mathbf{g}$ play in the Lagrangian of Equation 7.

*Direct* methods instead discretize the trajectory first and directly solve the resulting finite-dimensional NLP to a general-purpose optimizer ("discretize then optimize") [30]. They have gained in popularity due to their flexibility, and these methods may differ in how much of the trajectory is promoted to decision variables (Figure 9):

- *single shooting* parameterizes only the control and recovers the state by one forward integration from $\mathbf{s_0}$, so the optimizer sees the control alone;
- *multiple shooting* splits the horizon into segments integrated independently, adding the inter-segment states as variables and *defect* constraints that stitch the segments together, which improves conditioning and parallelization;
- *direct collocation* promotes every state and control node to a

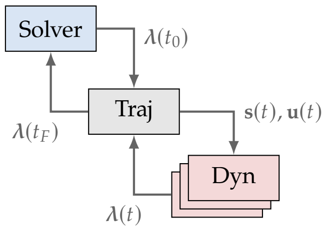


**Figure 7:** Indirect single shooting. The solver iterates on the initial costates $\lambda(t_0)$; the coupled state-costate system is integrated (Traj) against the dynamics (Dyn), and the terminal costate $\lambda(t_F)$ must satisfy the transversality condition. The boundary data split between the two ends make this a two-point boundary value problem.

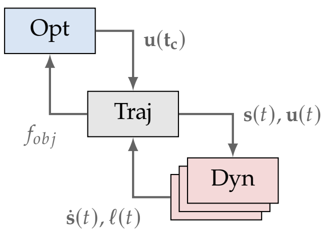


**Figure 8:** Direct single shooting. The optimizer (Opt) sets the whole control trajectory $\mathbf{u}(\mathbf{t_c})$ at the control points; the trajectory is integrated forward (Traj) through the dynamics (Dyn) and the discounted cost $f_{obj}$ is returned. Compare the indirect scheme of Figure 7.

[29]: Pontrjagin (1964), *The mathematical theory of optimal processes*

[30]: Kelly (2017), *An Introduction to Trajectory Optimization: How to Do Your Own Direct Collocation*

variable and enforces the dynamics by algebraic constraints at interior collocation points, with no explicit time integration [31, 32].

[31]: Hargraves et al. (1987), *Direct trajectory optimization using nonlinear programming and collocation*
[32]: Betts (2010), *Practical methods for optimal control and estimation using nonlinear programming*

*Policy optimization*, the second application introduced above, is an example of an optimal control problem and this work uses single shooting as a proof of concept: the World3 ODE is integrated forward for each candidate policy, which keeps the formulation simple and reuses the existing simulator, at the cost of long sensitivity chains that automatic differentiation (Section 2.4) is well suited to handle.

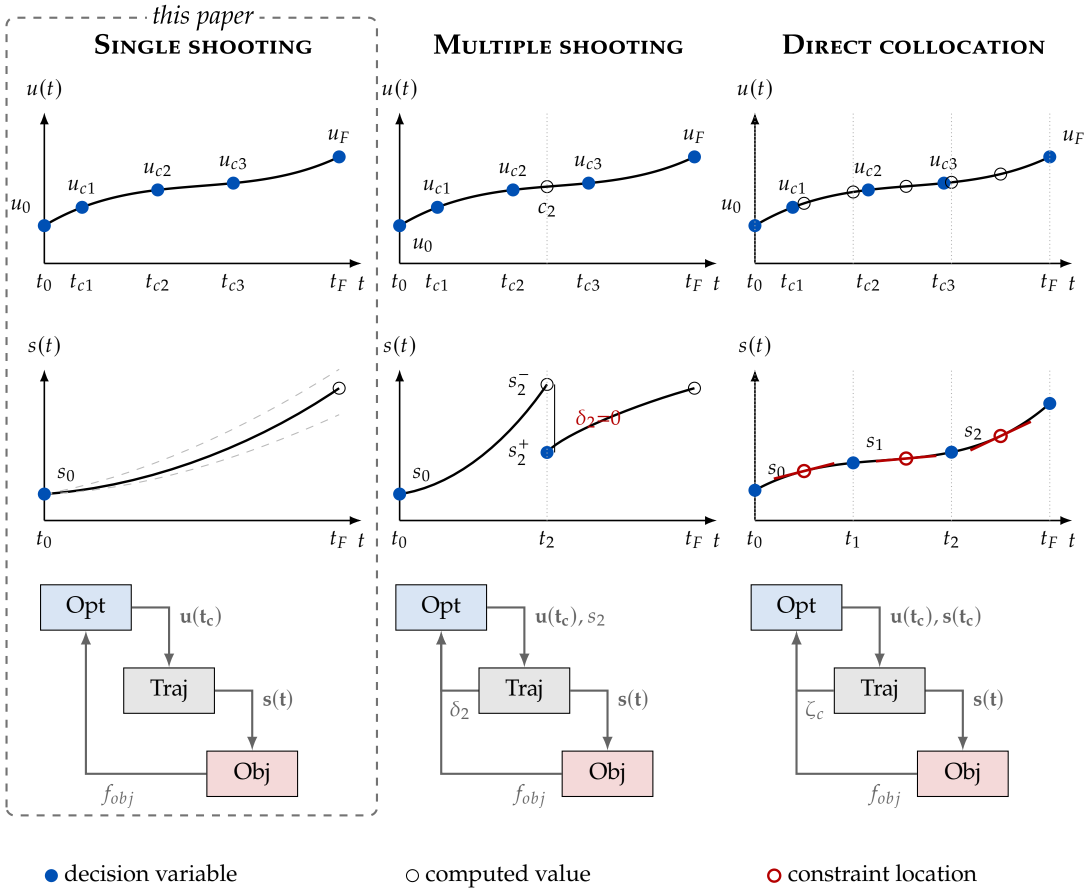


**Figure 9:** Three transcription strategies for an optimal-control problem, shown from the optimizer's point of view. The top row shows the control discretization $u(t)$, the middle row shows the state $s(t)$. Moving from left to right the optimization problem grows in dimension but becomes more sparse, better conditioned, and parallelisable. The N2 diagrams below each method show the interaction between the Optimizer (**Opt**), Trajectory time-integration (**Traj**), and Objective function (**Obj**). Adapted from [32].

Time-dependent controls $\mathbf{u}(t)$ are parameterized by a set of values at control points and interpolated to every step. The state is obtained with forward integration through the time-stepping operator $T$. The state recursion is evaluated inside the objective rather than handed to the optimizer, which produces long differentiation chain that Section 2.4 addresses.

$$\min_{\mathbf{u}\in\mathcal{U}} J(\mathbf{u}) = \sum_{k=0}^{N} \delta_{(k)}\, \ell\left(\mathbf{s}_{(k)}\right)$$

where:

$$\mathbf{s}_{(k+1)} = T\left(\mathbf{s}_{(k)},\ \mathbf{h}(t_{(k)}, \mathbf{s}_{(k)}, \mathbf{u}_{(k)}),\ \Delta t\right) \tag{10}$$

## 2.4 Numerical techniques

In order to exploit the scalability of gradient-based solvers, objective and constraints must be at least once continuously differentiable ($C^1$), so that $\nabla f_{obj}$ and the constraint gradients give a reliable search direction. Furthermore, these gradients must be available with respect to *every* input, the 55 constants for calibration or the full set of control points for the policy problem.

A gradient estimated numerically by finite differences (Equation 11) perturbs each input in turn, so a full gradient costs about $n+1$ model evaluations for $n$ inputs. When each evaluation integrates the model over a century, this becomes the dominant cost of the optimization; deriving the derivatives analytically would remove the runtime cost but is manpower-intensive and error-prone for a model with hundreds of coupled relationships [33].

$$\frac{\partial f_{obj}}{\partial x_i} \approx \frac{f_{obj}(\mathbf{x} + h\,\mathbf{e}_i) - f_{obj}(\mathbf{x})}{h} \tag{11}$$

[33]: Vanwynsberghe (2021), *PyWorld3 - The World3 model revisited in Python*

Differentiable programming resolves this by treating the whole simulation as a composition of elementary operations and assembles exact derivatives by automatic differentiation (AD), at a cost of roughly one extra model pass and with no derivative code written by hand. This approach is now standard in machine learning for calibrating models against data [34]. AD has two modes. Reverse mode propagates derivatives backward from the output and suits a single scalar against many inputs, as in the calibration loss and the policy objective, at the cost of storing intermediate values; forward mode propagates them alongside the computation and becomes preferable only when a formulation has many time-resolved constraints.

[34]: Blondel et al. (2024), *The Elements of Differentiable Programming*

Exact gradients still require the model itself to be differentiable, which the standard formulation is not. Many World3 relationships are table functions read by linear interpolation, continuous but with "jumps" it its slope near interpolation points (Figure 10). The re-implementation therefore replaces the piecewise-linear tables with smooth splines and the hard clips with smooth saturation, so

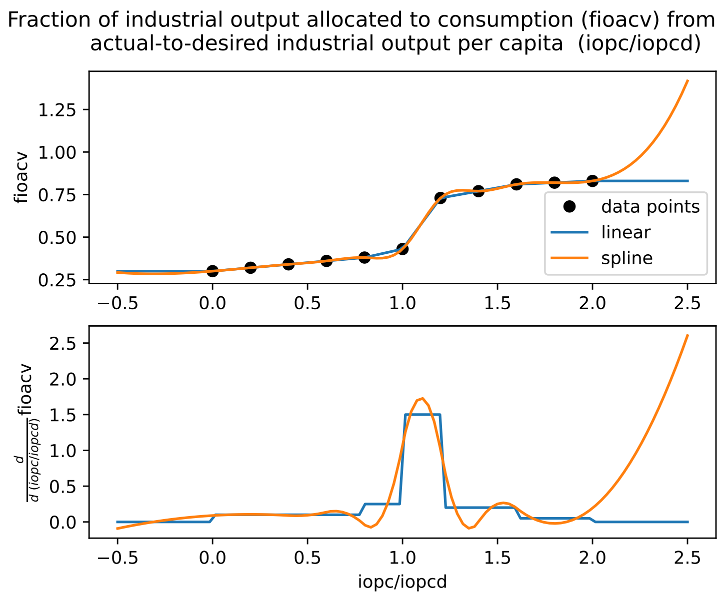


**Figure 10:** A World3 table function, the consumption-allocation fraction $f_{\text{ioacv}}$ against the output ratio $iopc/iopcd$, read by linear interpolation: continuous, but with a kink at every breakpoint where the derivative jumps.

the model becomes differentiable throughout.

The delays are folded into the global state vector and advanced with the physical stocks. A single coupled solve at the initial time sets the delay states, and every later step is then a feed-forward update (Figure 5) carried by the forward Euler scheme, with no per-step solver.

Two further techniques from high-performance computing keep the repeated evaluations fast. Just-in-time (JIT) compilation turns the simulation into optimized machine code on first execution, amortizing a one-time cost over the many runs an optimizer requires, while vectorization evaluates the same operation across many points at once. Both are provided by JAX [18], coupled to the model through the GEMSEO optimization framework [19] by the GEMSEO-JAX plug-in [35].

[35]: Costa-Alves et al. (n.d.), *gemseo-jax*

## 3 Calibration

A historical dataset covering all five World3 sectors was assembled from open-access sources as part of a supervised student project. The sources are: UN Population Division [36] and UN World Urbanization Prospects [37] for population; World Bank API [38] for capital; FAOSTAT [39] for agriculture; Our World in Data [40] for resources; and EDGAR v6 [41], the Global Carbon Budget [42], and NOAA Mauna Loa records [43] for pollution. Table 1 lists the variables compared against history and their source.

[36]: United Nations, Department of Economic and Social Affairs, Population Division (2022), *World Population Prospects*

[37]: United Nations, Department of Economic and Social Affairs, Population Division (2018), *World Urbanization Prospects: The 2018 Revision*

[38]: World Bank Group (2024), *World Bank Open Data API*

[39]: Food and Agriculture Organization of the United Nations (2024), *FAOSTAT Statistical Database*

[40]: Ritchie et al. (2023), *Energy*

[41]: European Commission, Joint Research Centre (JRC)/PBL Netherlands Environmental Assessment Agency (2021), *Emissions Database for Global Atmospheric Research (EDGAR), release version 6.0*

[42]: Friedlingstein et al. (2019), *Global Carbon Budget 2019*

[43]: National Oceanic and Atmospheric Administration (2023), *Trends in Atmospheric Carbon Dioxide: Mauna Loa $CO_2$ monthly mean data*

The pollution dataset carries a known mismatch: World3's pollution sector represents persistent toxic chemicals, not greenhouse gases, and lacks multi-pool atmospheric dynamics as needed for simulating climate pollutants [44].

[44]: Leach et al. (2021), *FaIRv2.0.0: a generalized impulse response model for climate uncertainty and future scenario exploration*

Calibration instantiates the inverse problem of Equation 8, with one mean-squared-error term per fitted variable over the 1960–2020 period, although some data are only available for a reduced temporal window (such as for the Capital and Pollution sectors). By default the weight $w_j$ counts every variable equally, so the population sector, which supplies 14 of the 22 series, carries the most of the residual in the objective. Section 5 considers down-weighting the series that only loosely match their model counterpart.

Variables that share units with the data (population, life expectancy, and arable land) are compared directly. Capital (as industrial output per capita) and pollution (as the persistent-pollution index) are recorded in different units, so each model and data series is normalized to its year-2000 value. The resource sector is the single exception, for which a different calibration method was employed: proved reserves grew over the period while the model's lone stock can only deplete, so its initial stock of reserves is fixed from the last entry of the proved-reserve data.

The 54 constants in $\theta$ are estimated by the multi-start search of Section 2.3: fifteen Latin-hypercube starts, each iterated until finding the best parameter fit with L-BFGS-B [21], with the lowest-residual solution retained. For comparison, the same objective is also minimized from a single start at the World3 Business-As-Usual (BAU) scenario constants (Table 2). The gradient $\nabla_\theta f_{obj}$ is obtained with one reverse-mode AD pass through the full simulation.

**Table 1:** The model variables compared against historical data in each World3 sector, with the open-access sources from which that data was assembled. Population, capital, agriculture, and pollution contribute 22 fitted variables to the calibration objective; the resource sector is shown for context but excluded, its initial stock being fixed from proved-reserve data instead (see text).

| Sector | Model variables compared (count) | Historical data source |
|---|---|---|
| Population | total population, life expectancy, the four age cohorts (0–14, 15–44, 45–64, 65+) and their death rates, crude birth rate, crude death rate, total fertility, annual births (14) | UN World Population Prospects [36] |
| Capital | industrial capital, service capital, industrial output per capita, labour force (4) | World Bank [38] |
| Agriculture | arable land, potential arable land, land fertility (3) | FAOSTAT [39] |
| Pollution | persistent-pollution index (1) | EDGAR [41], Global Carbon Budget [42], NOAA Mauna Loa [43] |
| Resources | nonrenewable resource fraction (excluded) | Our World in Data [40] |

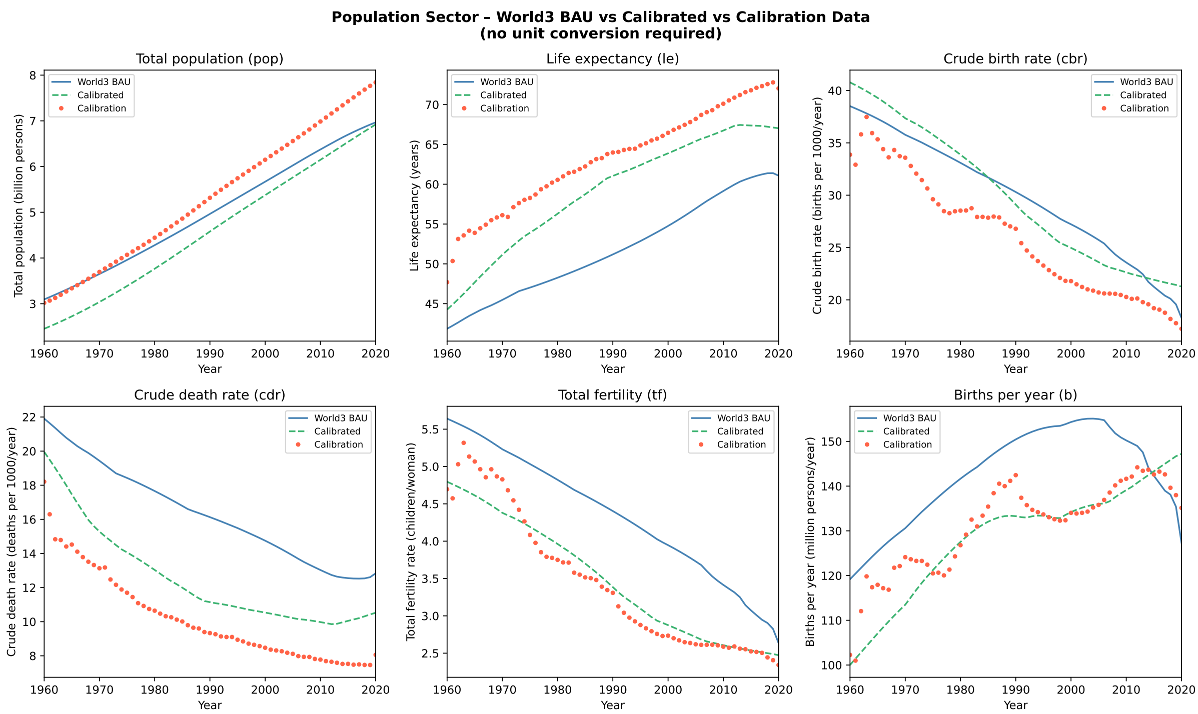


**Figure 11:** Calibration: Population sector. Calibrated (green dashed) versus BAU (blue solid) and historical data (red dots). Data: UN WPP [36].

## 3.1 Results

Figures 11–15 compare the calibrated scenario with the standard BAU scenario for each sector.

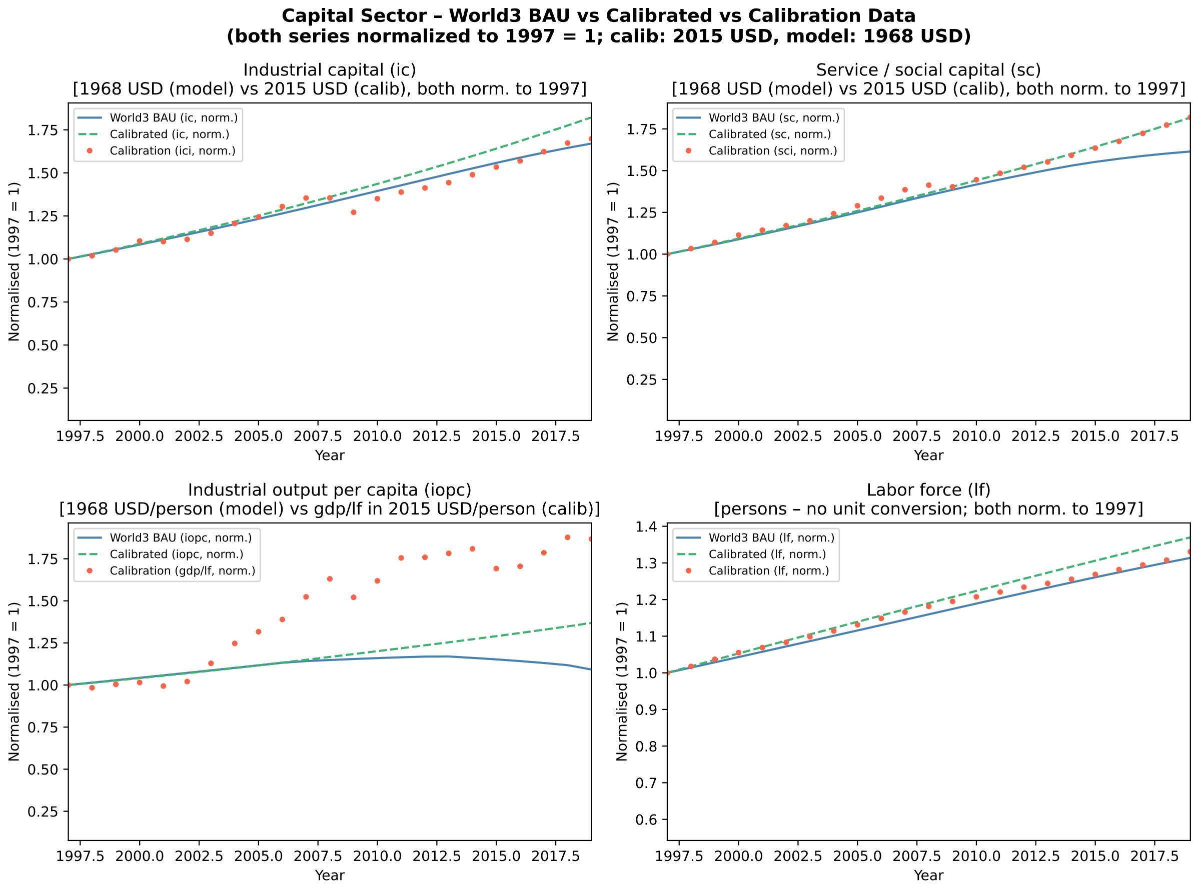


**Figure 12:** Calibration: Capital sector. Data: World Bank [38].

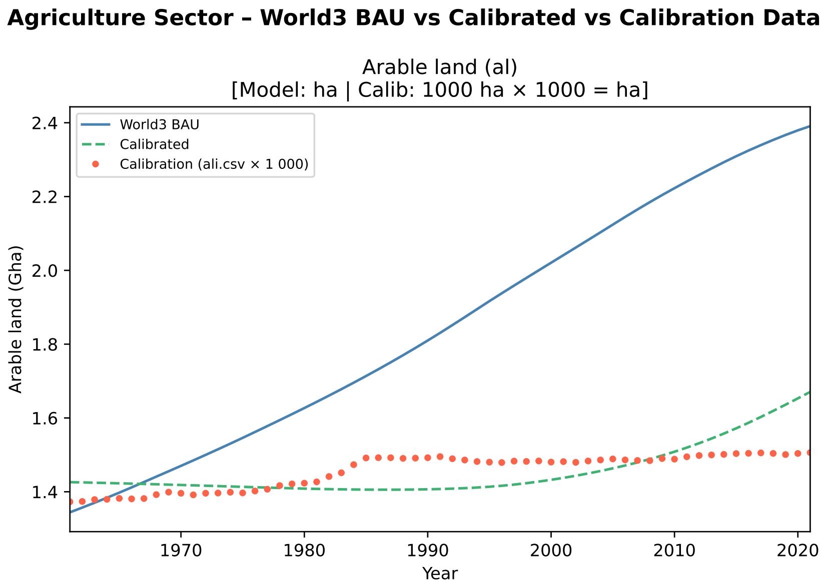


**Figure 13:** Calibration: Agriculture sector. Data: FAOSTAT [39].

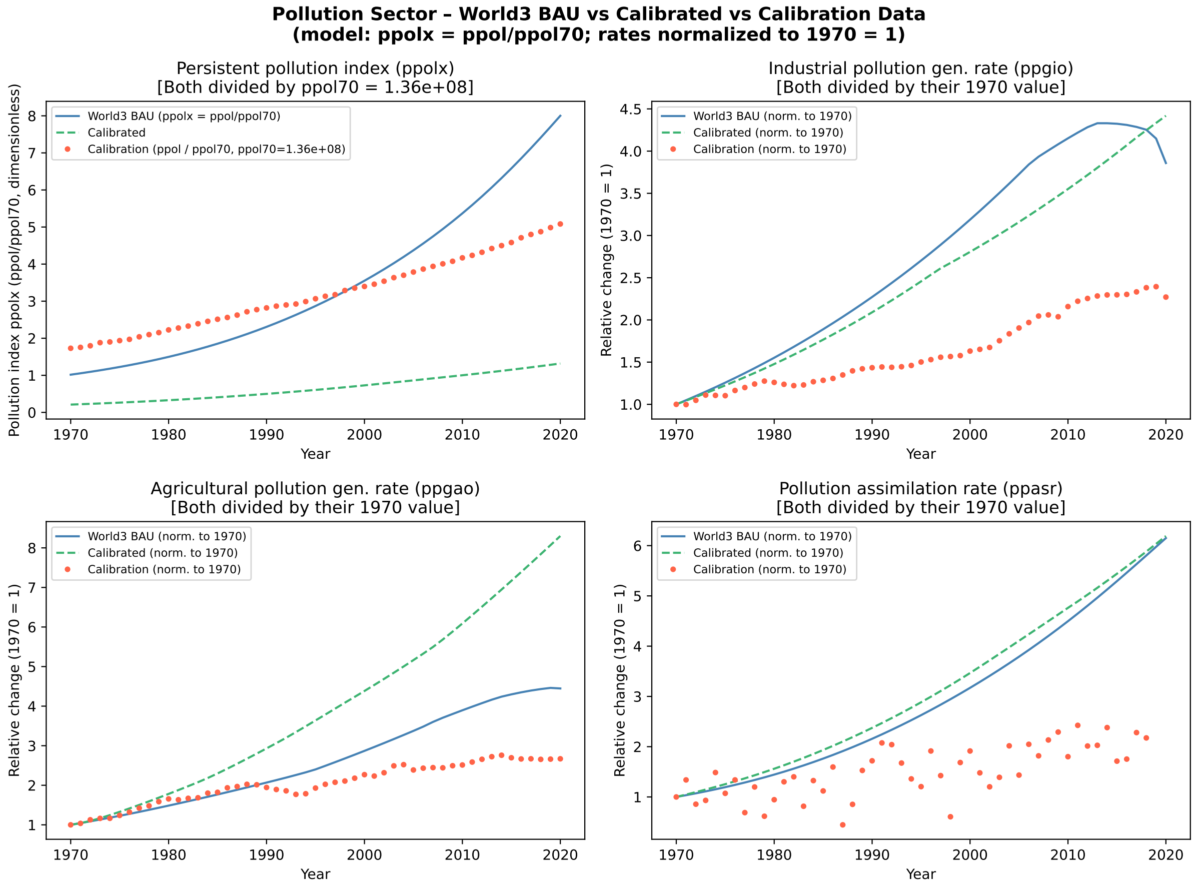


**Figure 14:** Calibration: Pollution sector. Data: EDGAR [41], Global Carbon Budget [42], NOAA [43].

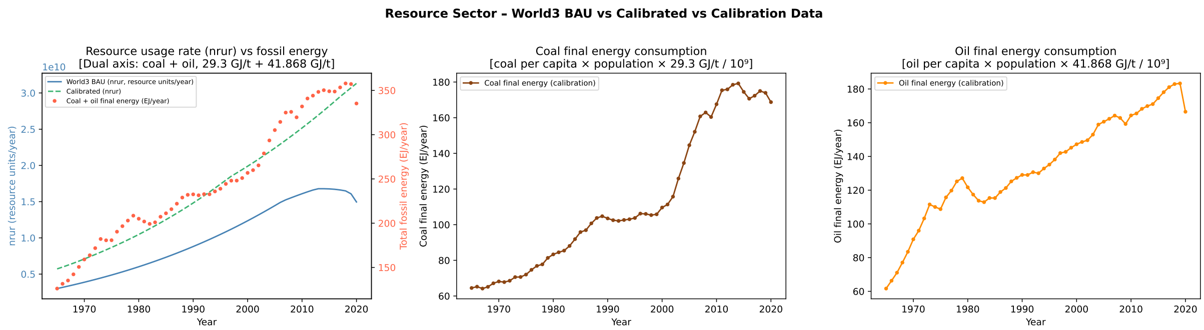


**Figure 15:** Calibration: Resource sector. Data: Our World in Data [40].

The population sector is the most fully constrained, contributing fourteen of the twenty-two series across the age structure (Figure 11): the four age cohorts, their cohort death rates, the crude birth and death rates, total fertility, and annual births are fitted separately rather than in aggregate. The calibrated model reproduces the demographic transition of the period, the death rate falling first and the birth rate following with a lag, total population and life expectancy are each matched to within about five to six percent.

The capital sector still carries residuals between simulated outputs and data (Figure 12). Industrial capital, service capital, and the labour force grow broadly in step with the record once normalized, but industrial output per capita rises more slower than the GDP-per-worker series after the 2000s. With a fixed capital-output ratio and no endogenous productivity growth, the model cannot reproduce the late acceleration of output, which can be identified as a missing mechanism in the model.

Agriculture is matched to a similar margin (Figure 13). But the model is still unable to reproduce the plateau in arable land as incomes develop and demand for food rises.

Pollution is the sector that follows historical data the least (Figure 14). The single-stock representation can be tuned to the observed rise in concentration of $CO_2$, but the lacking representation of carbon-cycle dynamics and variability in assimilation rates of the carbon sinks.

The resource sector is shown for context but kept out of the objective (Figure 15), as the initial reserve stock is fixed from data instead of handled by the optimizer. The model reproduces the historical rise in resource use driven by industrial output, yet its single depletable stock can only decline, whereas measured proved reserves grew over the period as discovery outpaced extraction.

The single BAU start already reaches a close fit, with objective 0.826, and the multi-start search improves it marginally, to 0.803: the nominal World3 constants sit in a good basin, but the multi-start confirms the presence of multiple local optima (Table 2).

## 4 Policy optimization

Once the model is calibrated, it can be reformulated as an optimal control problem for the 2000–2100 period. In the World3 scenarios there are three variables that control the capital allocation among sectors: the fraction of industrial output allocated to agriculture ($f_{\text{ioaa}}$), to consumption ($f_{\text{ioac}}$), and to services ($f_{\text{ioas}}$). In the standard model these variables are directly computed, where assumptions are made on how these high-level decisions are made depending on the current state of the system. Here, however, they are disconnected from that structure and handed to the optimizer as time-dependent decision variables, and they satisfy $f_{\text{ioaa}} + f_{\text{ioac}} + f_{\text{ioas}} \leq 1$ at every step.

This problem is solved using the single-shooting NLP of Equation 10, instantiated on the calibrated model. Each control is a piecewise-linear trajectory of 11 decadal points over 2000–2100, the first of which is considered to be fixed at the BAU starting value. This results in a 30-variable decision vector, which are then smoothed by a first-order delay ($\tau$ = 25 years) before entering the World3 ODE. The 55 calibrated constants are held fixed, the state is integrated forward from the calibrated year-2000 state, and the running cost is discounted at annual rate $\rho = 0.02$, which decreases the net present value of gains/losses that occur in the far future, this effect is discussed in Section 5. Two running costs are considered:

- *Human Welfare Index*: a per-capita welfare index that combines: a life-expectancy term, an education term (assumed to be a direct function of industrial output per capita), and an income term. Each term saturates at high values so that none can be traded without limit against the others, also, as this is a per-capita index, it is weighted by population before being discounted and summed;
- *Ecological Footprint*: aggregates the land footprint that human activity requires: arable land, urban-industrial land, and the land that would be needed to absorb persistent pollution, all expressed relative to the available biocapacity. Its cumulative discounted value is minimized, which favours policies that keep land conversion and pollution low.

The two single-objective problems are minimized with SLSQP [23], which respects the allocation constraint at every control point. Each is solved both from the BAU scenario as a starting point and with the fifteen-point multi-start search, in order test if the objectives have multiple local minima. Yet, as shown in Table 2, the single-start and multi-start result in the same single-objective optimum found, so the single-start strategy from BAU start is kept for the multi-objective exploration. The Pareto front is then traced with the modified Normal Boundary Intersection (mNBI) method [28] spread over 18 total sub-optimizations with SLSQP: the two anchors together with 16 evenly spaced interior sub-problems, spread uniformly along the front.

### 4.1 Results

Figure 16 shows the Pareto Front obtained. Discounted welfare varies by about a factor of 1.7 and discounted footprint by about a fifth. The front is mostly convex: starting from the welfare-optimal policy (point 1), the first reductions in footprint cost relatively little welfare, whereas reaching the footprint-optimal policy (point 18) costs about two-fifths of the discounted welfare. The practical consequence is that the policies of interest lie between the anchors: a representative intermediate policy achieves about half of the total footprint reduction while keeping more than nine-tenths of the welfare.

The calibrated business-as-usual policy, plotted in the same objective space, lies above the front and is dominated by it: at the welfare level of the baseline the optimized policies reach a markedly lower footprint, simiarly at the same footprint level as BAU the Pareto policies achieve much higher welfare.

The eight output panels trace how the allocation choices propagate through the model outputs (Figure 17). Industrial output per capita diverges after about 2030, separating the welfare-oriented policies, which sustain higher output, life expectancy, and welfare across the century, from the footprint-oriented ones, which hold the persistent-pollution index and the footprint low but shrink the economy. Food per capita follows output with a lag, through the capital allocated to agriculture, so the policies that diminish industry activity also decrease food production, which, in turn, lowers life-expectancy and the population trajectory. The pollution peak is both lower and earlier under the footprint-oriented policies, so policy acts on the amplitude of the overshoot and not only on its timing. The nonrenewable resource fraction declines in every policy, fastest under the welfare-oriented ones that sustain the highest output and slowest under the footprint-optimal one, though no policy halts the decline as most of the model structure remains unaffected by the reformulation.

The allocation trajectories behind optimized policies reveal a consistent under-investment in industry relative to BAU, this is present in high-welfare and is even stronger in low-footprint points (Figure 18). Moving along the front from the welfare-optimal policy to the footprint-optimal one, the share of industrial output reinvested in industry falls, from about half to almost nothing, and is redirected mainly into consumption, which is used up rather than reinvested in new productive capital. Slowing industrial investment slows capital growth, and with it the pollution and land conversion that drive the footprint, at the cost of lower output and welfare. The right-hand panels express the same allocations in absolute terms, confirming that these relative differences also impact the absolute output flow directed to each sector.

Table 2 summarizes the wall-clock times for both the BAU single start and the fifteen-start multi-start. A single-start of any task takes well under a minute, and the full BAU workflow, calibration followed by both single-objective optimizations and the Pareto exploration, completes in about four minutes on a single CPU core. The multi-start verification is heavier, the fifteen-start calibration alone taking about eight minutes, but as it returns the same optima for policy optimization, they were not kept for the Pareto front exploration. In the calibration case this is not the case, as the multi-start improves the calibration objective, indicating that this problem may have multiple local minima. For comparison, sector-by-sector calibration experiments using the original pyworld3 with finite-difference gradients required over one hour for the population sector alone. The JAX re-implementation achieves speedups of two to three orders of magnitude for both simulation and linearization.

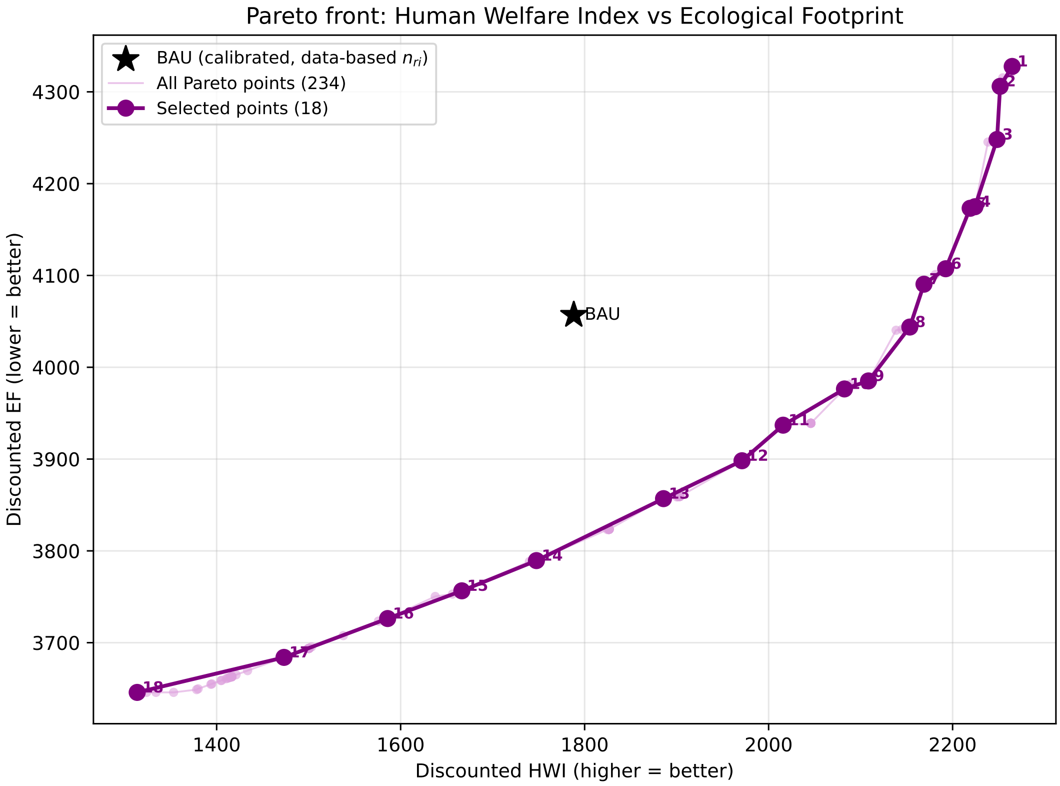


**Figure 16:** Pareto front between discounted Human Welfare Index (horizontal, higher is better) and discounted Ecological Footprint (vertical, lower is better), each summed over 2000–2100. The light curve shows all 192 Pareto-efficient policies returned by mNBI in its intermediate steps, and the dark markers number eighteen sub-optimization optima: from the welfare-optimal policy (point 1) to the footprint-optimal policy (point 18). The black star marks the calibrated business-as-usual baseline, which is dominated by all points in the Pareto front.

**Table 2:** Computational performance, comparing the BAU single start with the fifteen-start MultiStart for the three single-objective tasks. All runs on a single Intel Core i7-10850H at 2.70 GHz, each in its own process so the JIT compilation overhead (≈ 12 s) is included in every wall time. Iterations are unique points in the GEMSEO database (the mNBI total spans all sub-problems). The Pareto front is traced single-start only, since mNBI admits no MultiStart sub-optimizer (see text). The calibration objective column gives BAU / MultiStart; the single-objective values are identical to the digits shown.

| | | | BAU start | | MultiStart | | |
|---|---|---|---|---|---|---|---|
| Task | Algorithm | Variables | Iters | Wall (s) | Iters | Wall (s) | Final objective |
| Calibration | L-BFGS-B | 54 | 602 | 54.6 | 11891 | 502.6 | 0.826 / 0.803 |
| Max. HWI | SLSQP | 30 | 38 | 29.0 | 9413 | 167.7 | 2265 |
| Min. EF | SLSQP | 30 | 9 | 28.2 | 1723 | 61.4 | 3646 |
| Pareto (mNBI) | NLOPT_SLSQP | 30 | 1736 | 159.5 | — | | — |

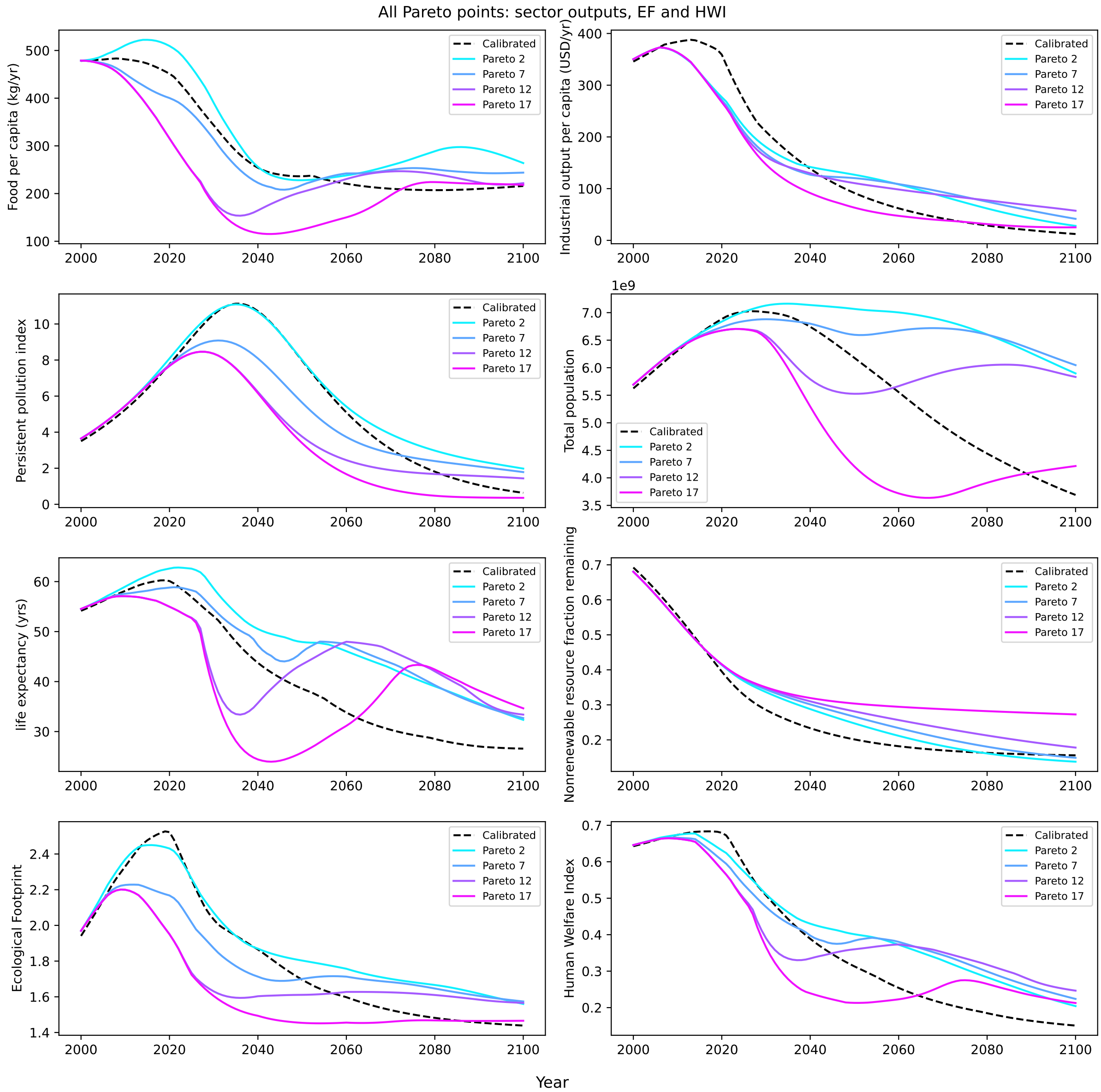


**Figure 17:** Annual trajectories over 2000–2100 for the calibrated baseline and four representative Pareto policies (points 2, 7, 12, and 17, spanning the welfare-optimal toward the footprint-optimal end). The eight panels show food per capita, industrial output per capita, persistent pollution index, total population, life expectancy, nonrenewable resource fraction remaining, Ecological Footprint, and Human Welfare Index. Welfare-optimal policies sustain higher output, life expectancy, and welfare, while footprint-optimal policies hold pollution and footprint low but shrink the economy. The resource fraction declines in every policy, fastest for the welfare-oriented ones.

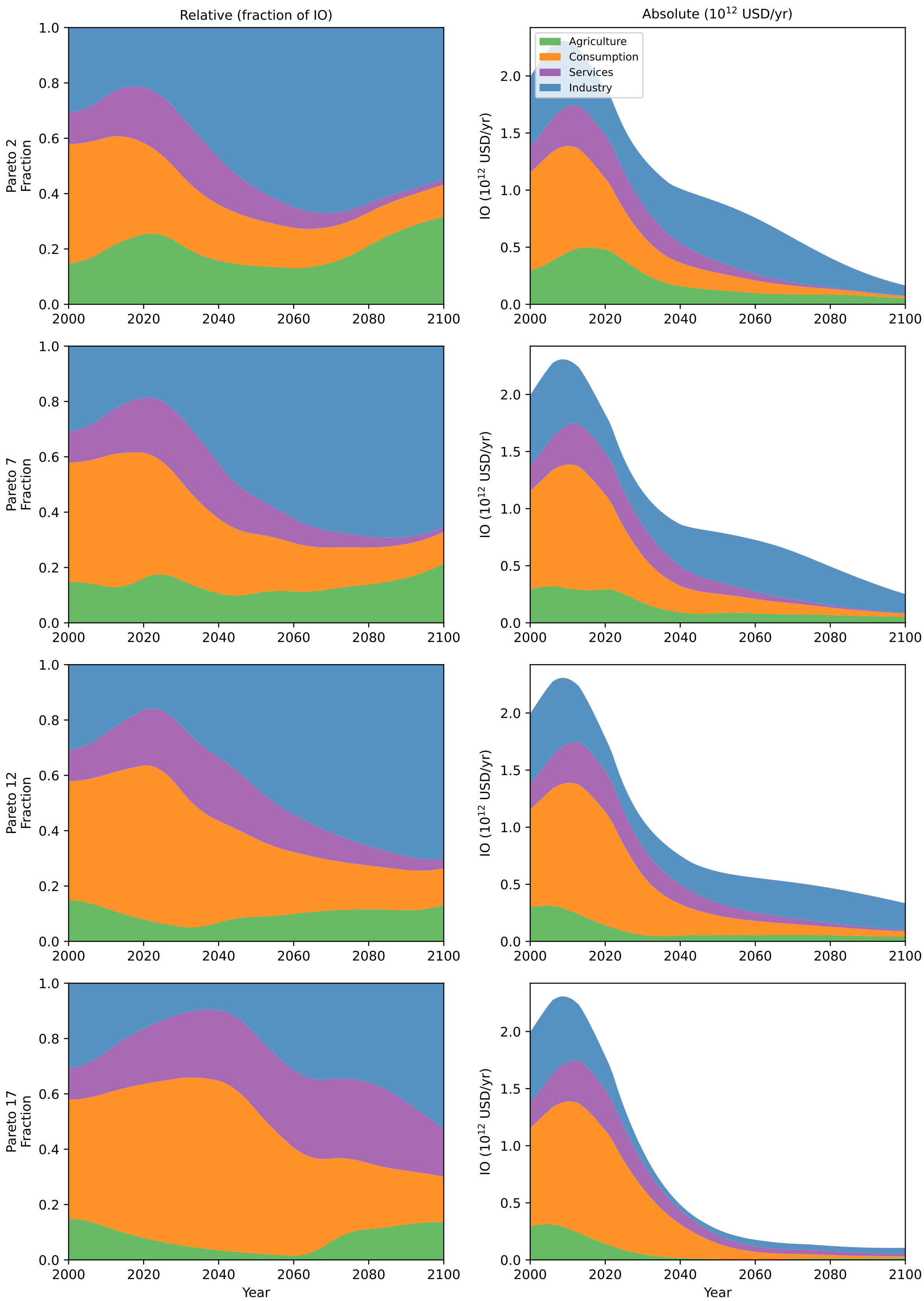


**Figure 18:** Industrial-output allocation over 2000–2100 for four representative Pareto policies (points 2, 7, 12, and 17). Each row is one policy. The left panel gives the allocation as a fraction of industrial output among agriculture, consumption, services, and reinvestment in industry, and the right panel gives the same allocation in absolute terms ($10^{12}$ USD/yr) together with total industrial output. From the welfare-leaning policy (point 2) to the footprint-leaning one (point 17), the share reinvested in industry falls and the consumption share rises.

## 5 Discussion

The approach presented for calibration and optimization of policy-relevant models starts with the assumption that the problem is sufficiently understood and that the mathematical model is valid to simulate the system's response to a wide set of different directions. Yet, the formulation of such models as well as ensuring that policy-makers have sufficient trust on them are not addressed here. The World3 model considered, while relevant for demonstrating numerical capabilities, is not detailed enough to provide insights to drive practical environmental policies due to: oversimplified aggregations (regions, resources, pollution types, produced goods), lack of technological detail concerning energy production, and lack of accounting for the specificities of climate pollutants (with lifetimes dependant on concentration, temperature, and sinks).

The calibration application demonstrated is shown to be capable of tuning model parameters to well fit historical behavior, and can serve as a diagnostic for the model, where the largest residuals can point to modeling limitations or inconsistencies between the model outputs and the data proxies used to calibrate them. The largest residual is in the capital sector, where industrial output per capita grows more slowly than the historical record, consistent with the absence of endogenous productivity growth. The pollution trend can be tuned to the historical rise, but the single-stock representation still cannot reproduce multi-pool carbon-cycle dynamics, a known structural limitation of the model. In contrast, population, life expectancy, and the arable land plateau are all well reproduced, indicating adequate model detail and data proxies.

The fact that the entire model is calibrated simultaneously also exposes structures invisible to sector-by-sector approaches: pollution-driven mortality multiplier is unidentifiable from 1970–2020 data because pollution remains below its activation threshold, the link between capital allocation to agriculture and food-per-capita mortality is visible only when Capital and Population parameters are estimated jointly. Yet the approach could be further improved:

- Pollution sector: instead of fitting a single persistent-pollution stock to a $CO_2$-derived proxy, one could either supply a proxy closer to the model's intent, an index of persistent toxic chemicals rather than greenhouse gases. Another improvement would be to replace the sector with a small climate emulator that resolves atmospheric pools and radiative forcing, such as FAIR [44], at the cost of departing from World3's original structure;
- Resource sector: as the single-stock issue present in the pollution sector, a wide number of other nonrenewable resource proxies could be used for the calibration, such as metal consumption and stocks, as well as a modification of the model to account for recycling based on the in-use metal stock. Another approach is to consider energy flow as the main resource input to the economy, which would need to account for both renewable and nonrenewable technologies;
- Calibration weights: the current approach is to give an equal weight for every variable for which data is available, even though some sectors have far more calibration data (the case of the population sector) and some variables are known to not match the data proxy used for it (pollution and resource sectors). Revising these weights downwards could let better-matched sectors to drive the fit while still using the weaker data;
- Table functions: while the calibration step focuses on the constant parameters of the model, another approach would be to also include the tabulated points used for interpolation. Currently, this would significantly increase the number of variables to be calibrated (as there are 53 table functions in World3) and can risk overfitting the model, as this would further modify the original structure of the model and there are relatively few variables with verified data for the calibration.

Regarding policy optimization, results depend on modeling choices that deserve scrutiny. Both objectives discount the future at a 2 % discount rate, which is considered to be at the lower end of the range used for climate-economy discounting [9, 11, 13], but which is judged to be appropriate due to the century-long prospection period [12]. Even if this factor is considered to be low, outcomes at 50 years away are at about 36 % of an equivalent one today, lower rates would tilt the whole front toward long-term sustainability and welfare, so the position of the trade-off curve is itself a normative choice.

The control parameterization adds a second assumption: decadal control points smoothed by a 25-year delay impose gradual change and rule out abrupt switches. This mirrors the inertia of real socioeconomic systems but also prevents fast-switching strategies, and the sensitivity of the front to the lag and the number of control points remains unexplored.

The chosen methods are deliberate but not unique. The multi-start versus single-start comparison (Table 2) shows the BAU World3 scenario is already a strong initial guess: multi-start confirms the optima and improves the calibration only marginally. The single-shooting transcription is the simplest of several options, other approaches such as multiple shooting or direct collocation were not experimented with, but could improve conditioning over the century-long horizon.

## 6 Conclusion

The goal of the current work is to demonstrate numerical optimization methods applied to the World3 model.

At first, the relevant litterature background highligths that the scalability of gradient-based methods to high-dimensional problems, such as most System Dynamics models are, due to the amount of parameters and variables these models have, but also due to the time-dependent nature of such problems. Using these methods on such problems is not straightfoward due to poor differentiability and gradient propagation, which are overcome by tailored numerical techniques, such as: replacing piecewise-linear tables with splines, folding delays into the state vector, and re-implementing World3 in frameworks with automatic differentiation (AD).

On the full five-sector model this makes the simultaneous calibration of 54 observables, optimal control, and Pareto front exploration feasible within minutes on a single CPU core, against more than an hour for a single sector calibration with gradient-free optimizers. The numerical recipe is general: AD removes the need to write derivatives by hand, just-in-time compilation amortizes the cost of repeated evaluation, and gradient-based solvers avoid increasing number of iterations as decision space increases, which together turn the poor scaling of standard System Dynamics optimization into a tractable problem.

The tools used here originate in the machine learning ecosystem, where they calibrate flexible function approximators against data. In that setting the model carries minimal domain knowledge: a neural network can fit almost any mapping, but its residuals show where the fit is poor, not why. System Dynamics models sit at the opposite end. Their structure embeds causal hypotheses through the model structure (feedback loops, stocks, flows, and delays) and calibration fits parameters within that structure, which makes the residuals interpretable. A persistent pollution residual localizes to a missing atmospheric pathway. An unidentifiable mortality multiplier reveals an inactive feedback loop. Resource depletion that no policy can halt points to the absence of substitution in the resource sector.

That same structure is also a source of bias. The functional forms and aggregation levels chosen by the modeler limit the behaviors the model can produce, which is a source of error, since World3 cannot represent technological substitution or greenhouse gas dynamics, but also a source of knowledge, since overshoot-and-collapse follows from the feedback structure rather than from parameter tuning. Used in this way, methods built for data-driven calibration serve a model-based purpose, delivering both computational efficiency and structural diagnosis, and the welfare-footprint trade-off they make explicit is one that scenario analysis leaves implicit.

## Code availability

All the scripts and data required to reproduce the results from this work are openly available at `https://github.com/iancostalves/gemseo-jax-world3`.

## Acknowledgements

The authors thank Emile Dosso, Clarisse Fenaux, Clément Mabileau, Lou Muraz, Baptiste Ronfard, and Matthieu Mbarga Owona, whose supervised engineering project at ISAE-SUPAERO produced the multi-sector historical dataset used in this work.

## Bibliography

[1] Donella H. Meadows. *The limits to growth: a report for the club of rome's project on the predicament of mankind*. eng. Ed. by Club of Rome and Potomac Associates. 2. ed. A potomac associates book. New York: Universe books, 1974 (cited on pages 2, 5).

[2] 'Mitigation Pathways Compatible with Long-term Goals'. In: *Climate Change 2022 - Mitigation of Climate Change*. Ed. by Intergovernmental Panel On Climate Change (Ipcc). 1st ed. Cambridge University Press, Aug. 2023, pp. 295–408. DOI: `10.1017/9781009157926.005`. (Visited on 05/19/2025) (cited on page 2).

[3] Dennis L. Meadows et al., eds. *Dynamics of growth in a finite world*. eng. Cambridge, Mass: Wright-Allen, 1974 (cited on page 2).

[4] Donella H. Meadows, Jørgen Randers, and Dennis L. Meadows. *The limits to growth: the 30-year update*. en. Reprint. London: Earthscan, 2009 (cited on pages 2, 5).

[5] Graham M. Turner. 'A comparison of *The Limits to Growth* with 30 years of reality'. In: *Global Environmental Change*. Globalisation and Environmental Governance: Is Another World Possible? 18.3 (Aug. 2008), pp. 397–411. DOI: `10.1016/j.gloenvcha.2008.05.001`. (Visited on 09/12/2025) (cited on page 2).

[6] Gaya Herrington. 'Update to limits to growth: Comparing the World3 model with empirical data'. In: *Journal of Industrial Ecology* 25 (2020). DOI: `10.1111/jiec.13084` (cited on page 2).

[7] John D. Sterman. *Business dynamics: systems thinking and modeling for a complex world*. eng. Nachdr. Boston: Irwin/McGraw-Hill, 2009 (cited on page 2).

[8] William D. Nordhaus. 'An Optimal Transition Path for Controlling Greenhouse Gases'. en. In: *Science* 258.5086 (Nov. 1992), pp. 1315–1319. DOI: `10.1126/science.258.5086.1315`. (Visited on 09/13/2023) (cited on page 2).

[9] William D. Nordhaus. 'Revisiting the social cost of carbon'. en. In: *Proceedings of the National Academy of Sciences* 114.7 (Feb. 2017), pp. 1518–1523. DOI: `10.1073/pnas.1609244114`. (Visited on 09/13/2023) (cited on pages 2, 22).

[10] Lint Barrage and William Nordhaus. *Policies, Projections, and the Social Cost of Carbon: Results from the DICE-2023 Model*. en. Tech. rep. w31112. Cambridge, MA: National Bureau of Economic Research, Apr. 2023, w31112. DOI: `10.3386/w31112`. (Visited on 01/07/2025) (cited on page 2).

[11] Nicholas Stern. *The Economics of Climate Change: The Stern Review*. Cambridge University Press, 2007 (cited on pages 2, 22).

[12] Lawrence Goulder and Roberton Williams. *The Choice of Discount Rate for Climate Change Policy Evaluation*. en. Tech. rep. w18301. Cambridge, MA: National Bureau of Economic Research, Aug. 2012, w18301. DOI: `10.3386/w18301`. (Visited on 01/28/2025) (cited on pages 2, 23).

[13] Dirk Schoenmaker and Willem Schramade. 'Which discount rate for sustainability?' en. In: *Journal of Sustainable Finance and Accounting* 3 (Sept. 2024), p. 100010. DOI: 10.1016/j.josfa.2024.100010. (Visited on 01/28/2025) (cited on pages 2, 22).

[14] Horst W. J. Rittel and Melvin M. Webber. 'Dilemmas in a general theory of planning'. en. In: *Policy Sciences* 4.2 (June 1973), pp. 155–169. DOI: 10.1007/BF01405730. (Visited on 03/31/2025) (cited on page 2).

[15] John Alford and Brian W Head. 'Wicked and less wicked problems: a typology and a contingency framework'. en. In: *Policy and Society* 36.3 (July 2017), pp. 397–413. DOI: 10.1080/14494035.2017.1361634. (Visited on 03/31/2025) (cited on page 2).

[16] Brian W. Head. *Wicked Problems in Public Policy: Understanding and Responding to Complex Challenges*. en. Cham: Springer International Publishing, 2022. (Visited on 04/01/2025) (cited on page 2).

[17] Emily M. Zechman, Marcio H. Giacomoni, and M. Ehsan Shafiee. 'An evolutionary algorithm approach to generate distinct sets of non-dominated solutions for wicked problems'. In: *Engineering Applications of Artificial Intelligence* 26.5 (2013), pp. 1442–1457. DOI: https://doi.org/10.1016/j.engappai.2013.03.004 (cited on page 3).

[18] James Bradbury et al. *JAX: composable transformations of Python+NumPy programs*. Version 0.3.13. 2018 (cited on pages 3, 13).

[19] Francois Gallard et al. 'GEMS: A Python Library for Automation of Multidisciplinary Design Optimization Process Generation'. en. In: *2018 AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*. Kissimmee, Florida: American Institute of Aeronautics and Astronautics, Jan. 2018. DOI: 10.2514/6.2018-0657. (Visited on 01/19/2025) (cited on pages 3, 13).

[20] Donella H. Meadows and Diana Wright. *Thinking in systems: a primer*. eng. London: Earthscan, 2009 (cited on page 3).

[21] Dong C. Liu and Jorge Nocedal. 'On the limited memory BFGS method for large scale optimization'. In: *Mathematical Programming* 45.1–3 (1989), pp. 503–528. DOI: 10.1007/BF01589116 (cited on pages 7, 13).

[22] Joaquim R. R. A. Martins and Andrew Ning. *Engineering Design Optimization*. Cambridge, UK: Cambridge University Press, Jan. 2022 (cited on pages 8, 9).

[23] D. Kraft. *A Software Package for Sequential Quadratic Programming*. Deutsche Forschungs- und Versuchsanstalt für Luft- und Raumfahrt Köln: Forschungsbericht. Wiss. Berichtswesen d. DFVLR, 1988 (cited on pages 8, 17).

[24] K. Deb et al. 'A fast and elitist multiobjective genetic algorithm: NSGA-II'. In: *IEEE Transactions on Evolutionary Computation* 6.2 (2002), pp. 182–197. DOI: 10.1109/4235.996017 (cited on page 9).

[25] Christian Igel, Nikolaus Hansen, and Stefan Roth. 'Covariance Matrix Adaptation for Multi-objective Optimization'. In: *Evolutionary Computation* 15.1 (Mar. 2007), pp. 1–28. DOI: 10.1162/evco.2007.15.1.1 (cited on page 9).

[26] 'On a Bicriterion Formulation of the Problems of Integrated System Identification and System Optimization'. In: *IEEE Transactions on Systems, Man, and Cybernetics* SMC-1.3 (1971), pp. 296–297. DOI: 10.1109/TSMC.1971.4308298 (cited on page 9).

[27] Indraneel Das and J. E. Dennis. 'Normal-Boundary Intersection: A New Method for Generating the Pareto Surface in Nonlinear Multicriteria Optimization Problems'. In: *SIAM Journal on Optimization* 8.3 (1998), pp. 631–657. DOI: 10.1137/S1052623496307510 (cited on page 9).

[28] Pradyumn Kumar Shukla. 'On the Normal Boundary Intersection Method for Generation of Efficient Front'. In: May 2007, pp. 310–317. DOI: 10.1007/978-3-540-72584-8_40 (cited on pages 9, 17).

[29] Lev Semonovic Pontrjagin, ed. *The mathematical theory of optimal processes*. Reprint. A Pergamon Press book. New York: MacMillan, 1964 (cited on page 10).

[30] Matthew Kelly. 'An Introduction to Trajectory Optimization: How to Do Your Own Direct Collocation'. en. In: *SIAM Review* 59.4 (Jan. 2017), pp. 849–904. DOI: 10.1137/16M1062569. (Visited on 09/12/2025) (cited on page 10).

[31] C.r. Hargraves and S.w. Paris. 'Direct trajectory optimization using nonlinear programming and collocation'. In: *Journal of Guidance, Control, and Dynamics* 10.4 (July 1987). Publisher: American Institute of Aeronautics and Astronautics, pp. 338–342. DOI: 10.2514/3.20223. (Visited on 09/12/2025) (cited on page 11).

[32] John T. Betts. *Practical methods for optimal control and estimation using nonlinear programming*. 2nd ed. Advances in design and control. Philadelphia: Society for Industrial and Applied Mathematics, 2010 (cited on page 11).

[33] Charles Vanwynsberghe. *PyWorld3 - The World3 model revisited in Python*. Nov. 2021 (cited on page 12).

[34] Mathieu Blondel and Vincent Roulet. *The Elements of Differentiable Programming*. 2024. URL: https://arxiv.org/abs/2403.14606 (cited on page 12).

[35] Ian Costa-Alves et al. *gemseo-jax*. URL: https://gitlab.com/gemseo/dev/gemseo-jax (visited on 09/19/2025) (cited on page 13).

[36] United Nations, Department of Economic and Social Affairs, Population Division. *World Population Prospects*. Report. Accessed: 2025-09-19. 2022. URL: https://population.un.org/wpp/ (cited on pages 13, 14).

[37] United Nations, Department of Economic and Social Affairs, Population Division. *World Urbanization Prospects: The 2018 Revision*. Accessed: 2025-09-19. 2018. URL: https://population.un.org/wup/ (cited on page 13).

[38] World Bank Group. *World Bank Open Data API*. Accessed via `wbgapi` Python package, 2025-09-19. 2024. URL: https://data.worldbank.org/ (cited on pages 13–15).

[39] Food and Agriculture Organization of the United Nations. *FAOSTAT Statistical Database*. Accessed: 2025-09-19. 2024. URL: https://www.fao.org/faostat/en/ (cited on pages 13–15).

[40] Hannah Ritchie, Pablo Rosado, and Max Roser. *Energy*. Accessed: 2025-09-19. 2023. URL: https://ourworldindata.org/energy (cited on pages 13, 14, 16).

[41] European Commission, Joint Research Centre (JRC)/PBL Netherlands Environmental Assessment Agency. *Emissions Database for Global Atmospheric Research (EDGAR), release version 6.0*. Accessed: 2025-09-19. 2021. URL: https://edgar.jrc.ec.europa.eu/ (cited on pages 13, 14, 16).

[42] Pierre Friedlingstein et al. 'Global Carbon Budget 2019'. In: *Earth System Science Data* 11.4 (2019), pp. 1783–1838. DOI: 10.5194/essd-11-1783-2019 (cited on pages 13, 14, 16).

[43] National Oceanic and Atmospheric Administration. *Trends in Atmospheric Carbon Dioxide: Mauna Loa $CO_2$ monthly mean data*. Accessed: 2025-09-19. 2023. URL: https://gml.noaa.gov/ccgg/trends/ (cited on pages 13, 14, 16).

[44] N. J. Leach et al. 'FaIRv2.0.0: a generalized impulse response model for climate uncertainty and future scenario exploration'. In: *Geoscientific Model Development* 14.5 (2021), pp. 3007–3036. DOI: 10.5194/gmd-14-3007-2021 (cited on pages 13, 22).