



Differentiable CoCiP: exact sensitivities of contrail energy forcing with DIFFCOCIP v1.0

Ramon Dalmau¹, Philippe Very¹, Gabriel Jarry¹, and Franck Ballerini¹

¹Aviation Sustainability Unit, EUROCONTROL, Brétigny-sur-Orge, France

Correspondence: Ramon Dalmau (ramon.dalmau-codina@eurocontrol.int)

Abstract. Reducing contrail warming requires knowing how a flight’s climate effect responds to changes in its trajectory, the weather and the contrail model’s assumptions. The Contrail Cirrus Prediction model (CoCiP) computes this effect as energy forcing, EF , but its reference implementation does not provide these sensitivities. Finite differences could estimate them by perturbing one input at a time, but this is impractical for weather fields holding millions of values. We present DIFFCOCIP v1.0, a reimplementa-
 5 a reimplementa-
 tion that applies reverse-mode automatic differentiation to obtain all these sensitivities in a single backward pass. It reproduces the reference model’s EF to the accumulated floating-point round-off and returns derivatives with respect to waypoint altitudes, gridded humidity and model parameters. Three applications assess the use and limits of these derivatives. In a four-season fleet campaign, altitude gradients guide the placement of candidate manoeuvre windows, and every candidate is then evaluated with the unmodified model. For about one third of reprofiled flights, the selected manoeuvre reduces EF despite
 10 a gradient prediction of an increase, showing why finite manoeuvres must still be checked against the full model. Humidity gradients identify the small part of a four-dimensional weather grid that influences one flight’s EF , including positions reached by its drifting contrail after the aircraft has passed. Among six model parameters, EF proves most sensitive to the humidity-correction coefficient, whose effect is about 7 times larger than that of the next most influential parameter. The derivatives are exact within each smooth region of the implemented model, but do not capture discontinuous changes across the formation,
 15 persistence and termination thresholds. DIFFCOCIP also includes an optional smoothed formulation of these criteria to support gradient-based optimisation; the results reported here use the unmodified model throughout.

1 Introduction

Contrail cirrus forms when aircraft exhaust mixes with sufficiently cold air and produces ice crystals. The crystals can persist and spread when the surrounding air is supersaturated with respect to ice. These clouds trap outgoing infrared radiation and
 20 reflect incoming sunlight. Their net effect, summed over aviation, is warming (Lee et al., 2021; Burkhardt and Kärcher, 2011; Kärcher, 2018). For 2018, Lee et al. (2021) estimated an effective radiative forcing of 57.4 mW m^{-2} from contrail cirrus, compared with 34.3 mW m^{-2} from aviation’s accumulated CO_2 . The same assessment attributed about eight times more uncertainty in aviation’s net forcing to non- CO_2 effects than to CO_2 .

Contrail forcing is particularly sensitive to upper-tropospheric humidity: numerical weather models represent it imper-
 25 fectly (Gierens et al., 2020; Teoh et al., 2022). It also depends on soot emissions and on the microphysical processes gov-



erning ice formation and survival, though comparisons with observations cover only a limited set of conditions (Schumann et al., 2017). Understanding which inputs most strongly affect a prediction is useful for assessing uncertainty and for designing avoidance measures.

Avoidance is possible because contrail warming is concentrated in a small fraction of flights and along particular parts of their routes. In the campaign presented here, for example, the top 1 % of flights by energy forcing accounts for a median of 26 % of daily gross warming: a vertical deviation can sometimes move such a flight outside an ice-supersaturated layer and prevent persistent contrails over that part of the route (Teoh et al., 2020a, 2022). Predicting the benefit, though, requires evaluating how the contrails change when the flight changes altitude.

The Contrail Cirrus Prediction model (CoCiP) (Schumann, 2012) provides a physical model for this calculation. It follows the exhaust deposited along successive segments of a flight as it moves with the wind, evolves and dissipates. Integrating each segment’s radiative effect over its area and lifetime (Schumann et al., 2012), then summing over the flight, gives its energy forcing EF in joules. Positive EF denotes a net warming contribution; negative EF denotes cooling.

This paper asks three questions about that output. How does EF respond to a small change in each waypoint altitude? Which humidity values in the weather grid influence it? Which empirical model constants produce the largest response to a small relative change? Each question concerns a derivative: the local rate of change of EF with respect to an input. Collecting these derivatives gives a gradient, with one entry per input.

The reference implementation, PYCONTRAILS (Shapiro et al., 2026), returns EF but does not return its gradient. A derivative can be approximated by finite differences: increase and decrease an input slightly, rerun the model, and divide the difference in outputs by the difference in inputs. Repeating this for every input requires two runs per input. This is feasible for a few constants, costly for hundreds of waypoint altitudes, and impractical for the roughly four million humidity values in the weather extracts used here. A complete humidity gradient would require about eight million runs per flight by this method.

Reverse-mode automatic differentiation provides another route. It first evaluates the model, then applies the chain rule backwards through its operations to calculate how each input contributed to the output. This backward calculation is the discrete *adjoint* of the simulation. For a scalar output such as EF , it obtains all input derivatives in one backward pass. Its work scales with the underlying computation, rather than requiring a separate simulation for each input. Adjoint methods have long supported variational data assimilation (Le Dimet and Talagrand, 1986; Talagrand and Courtier, 1987; Courtier et al., 1994; Rabier et al., 2000; Errico, 1997). Automatic differentiation generates the derivative from the model code, reducing the need to maintain a separate, hand-derived adjoint (Griewank and Walther, 2008; Baydin et al., 2018; Hascoët and Pascual, 2013; Utke et al., 2008; Heimbach et al., 2005; Forget et al., 2015). Related methods are increasingly used in Earth system modelling (Gelbrecht et al., 2023; Kochkov et al., 2024; Shen et al., 2023; Sapienza et al., 2024).

Existing contrail-avoidance methods provide useful context for this contribution. Several trajectory-optimisation studies use gradients of a precomputed forcing field or a fitted cost function (Sridhar et al., 2010; Simorgh and Soler, 2025; Dietmüller et al., 2023; Yin et al., 2023). These gradients describe variation in that representation of forcing. Differentiating the flight-specific CoCiP simulation instead follows how changes to its inputs propagate through contrail formation, evolution and radiative forcing. Smoothing a cost field can help an optimiser (Sun and Olive, 2025), but changes how sharp transitions are



represented. Other approaches evaluate a set of candidate flight levels directly (Teoh et al., 2020b), use gridded CoCiP for fleet forecasting (Engberg et al., 2025), or screen flights using predicted forcing (Sonabend-W et al., 2024). Direct evaluation is well suited to comparing a small set of manoeuvres. The adjoint adds a complementary capability: detailed sensitivities to trajectories, weather and model parameters.

65 CoCiP presents a specific difficulty for differentiation. It uses yes-or-no tests to decide whether a contrail forms, survives its early development and remains alive at each later time step. Changing an input enough to change one of these decisions can produce a jump in EF . A derivative does not exist at such a jump. Between thresholds, however, the same decisions remain in force and the model has a local derivative. Similar issues arise when differentiating atmospheric models with convection and precipitation switches (Zou, 1997; Xu, 1996; Zou et al., 1993; Ehrendorfer et al., 1999). Section 2.2 explains this distinction
 70 for CoCiP and shows why an exact local derivative is not a prediction of every finite change.

We present DIFFCOCiP v1.0, a reimplementation of CoCiP in JAX (Bradbury et al., 2018). JAX is a Python library that compiles numerical array calculations and automatically differentiates them. Here it differentiates the CoCiP equations and their implemented time-stepping scheme; it does not train a surrogate model. We verify the forward calculation against PYCON-
 TRAILS v0.63.3 and check the derivatives against finite differences away from switching boundaries. Throughout this paper,
 75 *exact* refers to the derivative of the implemented model within a smooth region, up to numerical precision. It does not imply that the physical model or its inputs are free of uncertainty.

The three use cases distinguish the inputs being differentiated:

1. *Waypoint altitudes*, $\partial EF / \partial z$ (Sect. 4). A four-season campaign uses the gradient to place a candidate altitude-change window. It then compares the gradient's linear prediction with a full evaluation of each proposed manoeuvre. This is an
 80 application to control variables, a standard adjoint target (Talagrand and Courtier, 1987; Cacuci, 1981; Navon, 2009).
2. *Gridded humidity*, $\partial EF / \partial q$ (Sect. 5). For one flight, we locate the weather values that influence its forcing. Because the model reads meteorological fields by interpolation rather than by a closed-form function of position, this gradient requires differentiating through that interpolation as well as through the simulation (Sect. 2.3). The resulting map is the input required for first-order propagation of humidity uncertainty, analogous to forecast-sensitivity diagnostics (Langland and Baker, 2004; Baker and Daley, 2000; Cardinali, 2009). This paper demonstrates the sensitivity map; validation of
 85 uncertainty propagation is reserved for a companion study.
3. *Empirical constants*, $\partial EF / \partial \theta$ (Sect. 6). We compare the local response to six model parameters. Such derivatives are also used in inverse problems (Tarantola, 2005), although parameter calibration is outside the scope of this study.

Section 2 explains the model, the meaning and computation of its derivatives, and their verification. Section 3 describes the
 90 common data and computational costs. The three use cases follow in Sections 4–6, before the discussion and conclusions. The appendices give the campaign protocol and the sensitivity to the integration step and maximum contrail age.



2 The differentiable model

This section follows one flight through the forward simulation, then explains how the backward pass obtains its sensitivities. It distinguishes smooth responses from changes in the model's yes-or-no decisions before presenting the adjoint equations, implementation choices and verification.

2.1 The contrail model and its energy forcing

Because everything that follows differentiates this model, what matters here is how it computes and not only what it computes. Figure 1 sketches the sequence.

CoCiP (Schumann, 2012) treats a flight as a chain of *segments*, one between each consecutive pair of waypoints, the successive points along the flight path at which the plan fixes a position, a time and an altitude. Each segment is the piece of exhaust laid down over that stretch of flight, and each is followed separately for its whole life. There are typically a few hundred segments per flight, and a flight's forcing is the sum of what each of them does.

The life of one segment runs through the stages drawn in Fig. 1.

Formation. The Schmidt–Appleman criterion (SAC) (Schumann, 1996; Appleman, 1953) decides whether a visible contrail forms behind that segment at all. As hot, moist exhaust mixes with cold ambient air, the mixture's temperature and vapour content trace a straight line in the temperature–vapour plane, and a contrail forms if and only if that mixing line reaches liquid saturation, so that water droplets condense and then freeze. Whether it does depends on the ambient temperature and pressure and on how much water and heat the engine puts out per unit of air, which is why engine efficiency enters the model here. The test returns yes or no; there is no partial contrail.

The wake-vortex phase. For a contrail that does form, the first few minutes are dominated by the aircraft's wake. The two counter-rotating vortices trailing from the wings entrain the exhaust and sink several hundred metres, compressing and warming the air they carry, so that some fraction of the newly formed ice crystals sublimates and is lost. What survives sets the initial depth, width and ice content of the contrail, and this is where the aircraft's mass and wingspan enter, the vortices' strength following from the weight they support and their spacing from the span. A second yes-or-no test then decides persistence: unless the surrounding air is supersaturated with respect to ice, the surviving crystals evaporate and the contrail is gone within minutes.

Evolution. A segment that persists is followed forward in discrete time steps, 30 min by default. At each step it is advected by the wind, so contrails drift, often hundreds of kilometres from where the aircraft flew; it is deformed by vertical wind shear, which tilts and spreads it; it grows by turbulent diffusion; it loses ice by sedimentation as crystals fall out; and it exchanges water with the surrounding air according to the local humidity, which is read from the weather field at the segment's current position by interpolation. Each step therefore both updates the segment's state and depends on the state the previous step left.

Radiative effect and death. At every step the segment's current ice content, dimensions, particle size and the local radiation determine its instantaneous radiative forcing (Schumann et al., 2012): how much outgoing infrared it traps, minus how much sunlight it reflects away. That forcing is per unit area, so it is multiplied by the area of the contrail the segment has spread



125 into, and by the duration of the time step, and accumulated, so that a wider or longer-lived contrail carries more energy than a narrow or short-lived one at the same forcing per square metre. Accumulation stops when the segment is declared dead, on any of the six thresholds Table 1 lists.

Summing the accumulated forcing over all the segments of a flight gives the flight's *energy forcing* EF , in joules: the time-integrated net radiative energy change attributed to that flight's contrails in the model. EF is signed. A segment that traps more 130 infrared than it reflects sunlight contributes positively and warms; one that reflects more than it traps contributes negatively and cools, as can occur for daytime contrails over dark surfaces. A flight with $EF > 0$ warms on balance in the model, one with $EF < 0$ cools.

When results are aggregated over a day or a week, the warming and cooling sides are kept separate. The *gross warming* of a day is the sum of EF over only those flights with $EF > 0$, and it is the denominator of every fleet-level share reported in 135 this paper; the per-flight removals of Section 4 are instead shares of each flight's own forcing. The reason is arithmetic: the net forcing over a day is warming minus cooling, and on days when the two nearly cancel it passes close to zero, so a percentage of it would be unstable or undefined.

2.2 The discrete structure and the meaning of the derivative

We want the derivative of EF with respect to the quantities a user can change or is uncertain about: the altitude of each 140 waypoint, the humidity at each grid point of the weather field, and each of the model's empirical constants. Writing that derivative down by hand is impracticable, because, as Section 2.1 showed, EF is not a formula but the output of a program with several hundred segments, each integrated over its whole lifetime in 30 min steps.

Size alone, however, does not obstruct differentiation, and neither do the features of the model that look at first like obstacles. Reverse-mode automatic differentiation needs no equation: it records the elementary arithmetic operations the program per- 145 forms while it runs and applies the chain rule to that recording, which for a single mid-haul flight holds some 25,000 elementary array operations, large for a human and unremarkable for a compiler. What is written as an integral is, by the time anything is differentiated, already the finite sum $\sum_k g(x_k) \Delta t$ over the model's own time steps, which differentiates term by term. The dependence of each step on the one before it, $x_{k+1} = F_k(x_k, u)$, is genuine, but it is precisely the structure the adjoint method was designed to handle, and Section 2.3 sets out how.

150 Weather interpolation is also part of the differentiated calculation. The model does not have a formula for humidity as a function of position; it has a table of values at grid points and reads between them by interpolation, so a derivative with respect to position must pass through that lookup. The lookup, however, is an interpolation, linear in each of its four coordinates (longitude, latitude, pressure level and time): an explicit arithmetic formula in both the query position and the tabulated values, and hence differentiable in both. Differentiating with respect to the query position gives the sensitivity to where the aircraft is, 155 while differentiating with respect to the tabulated values themselves gives $\partial EF / \partial q$, one number per grid point, which is the object Section 5 uses.

The main difficulty is the model's discrete decisions. Table 1 distinguishes three types. A jump changes EF discontinuously. A kink leaves EF continuous but changes its slope abruptly. A table lookup selects a constant from a finite set; the selected



Table 1. Discrete operations and their effects on differentiation. A jump changes EF discontinuously; a kink changes its slope while leaving its value continuous. A gather is a table lookup whose selected value is constant between boundaries. The first three rows can change which contrail segments contribute, or for how long. RH_i is relative humidity over ice, with RH_i = 1 denoting saturation. A habit is an ice-crystal shape class; SAC is the Schmidt–Appleman criterion.

Decision	Test	Effect on the derivative
Formation (SAC)	$T < T_{\text{crit}}$, the Schmidt–Appleman threshold temperature	jump : a contrail exists or does not
Initial persistence	ice water content and RH _i thresholds	jump : the new contrail lives or dies
End of life	six conditions that must all hold for the contrail to live on: age, optical depth, ice number density, altitude, segment length and latitude within their limits	jump : accumulation stops
Humidity correction, boost step (Sect. 2.4)	RH _i /rhi_adj ≥ 1, the test applied after the division	kink : EF unbroken (a power of 1 is 1), slope not
Habit-class selection	integer table lookup	gather : derivative exactly zero between boundaries, EF stepping at one

constant has zero derivative between selection boundaries and can jump at a boundary. Derivatives within each smooth region remain meaningful, but they cannot be extended through a discontinuity. Weather interpolation also changes slope at grid-cell boundaries. Derivative checks therefore compare evaluations within the same smooth region.

We can now say precisely what the gradient of this model is. Write the contribution of segment i as a product: an indicator that is 1 if the segment forms a contrail and 0 otherwise, a second indicator that is 1 if that contrail persists, and the forcing $e_i(z)$ it accumulates over its life. Summing over segments,

$$EF(z) = \sum_i \mathbb{1}[\text{SAC}_i(z)] \mathbb{1}[\text{persist}_i(z)] e_i(z), \quad (1)$$

where z is the flight’s altitude profile. The indicators are the first two jumps of Table 1, written explicitly. The third, end of life, is not written out: it sits inside $e_i(z)$, deciding how many steps that segment accumulates before it stops. It is a jump of the same kind. On the test flight examined below the formation test never changes its answer, so every jump there comes from the two later tests.

Differentiating Eq. (1) with respect to z shows what the derivative can and cannot see. An indicator is constant almost everywhere: nudge the altitude slightly and a segment that was forming a contrail is still forming one, so the indicator contributes



nothing to the derivative. What survives is

$$\frac{dEF}{dz} = \sum_i \mathbb{1}[SAC_i] \mathbb{1}[persist_i] \frac{de_i}{dz}, \quad (2)$$

a sum of the sensitivities of the contributing contrail segments. Here “currently” refers to the baseline simulation being differentiated, including its entire contrail history. It does not mean the contrails present at a single instant, or contrails left by other flights. The derivative includes changes in ice properties, geometry, radiative forcing and weather sampling, while the discrete formation and survival decisions stay unchanged. Figure 2 shows exactly this measured on a test flight (a synthetic 20-waypoint trajectory whose forcing is far below a real flight’s, hence the gigajoule rather than terajoule scale of the figure): EF against a uniform shift of the whole profile is a staircase of smooth pieces, and the reverse-mode slope is tangent to whichever piece the flight is on. This is the exact derivative within each smooth region. It describes changes to the baseline contrail history while the discrete decisions remain fixed; it does not describe the appearance or disappearance of a contrail at a threshold.

An example illustrates this distinction. Suppose a flight is currently laying persistent contrails along the middle third of its cruise, and consider raising it slightly. If the air above is drier and the discrete decisions remain unchanged, the derivative describes the resulting local change in their forcing. Now suppose that instead of a slight rise the aircraft climbs 600 m, and that at the new altitude a stretch of the flight that had been forming nothing crosses the formation threshold and begins to lay contrail. EF jumps upward. Nothing in the derivative measured at the original altitude anticipated that jump, and no derivative could have, because EF is genuinely discontinuous there. This limitation comes from the discrete decisions in CoCiP. Automatic differentiation faithfully follows those decisions; it does not replace them with smooth transitions.

The campaign in Section 4 considers altitude changes of 1000 ft and 2000 ft: finite changes, large enough to cross model thresholds, so the gradient serves as a local guide for placing a candidate manoeuvre window. Whether that manoeuvre reduces forcing is then determined by running the full model at the proposed profile. This use of the gradient is a placement heuristic, not a claim that it identifies the best possible window.

Returning from operations to the code, one implementation detail explains why Eq. (2) emerges from it automatically. A conditional written as an ordinary Python `if` cannot be used here, because the code is compiled once and run over whole batches of flights, so the values being compared are not known when the program is recorded. Every branch is therefore written as a `where`: an operation that evaluates both alternatives and then selects between them element by element. The derivative of `where(c, a, b)` is `where(c, da, db)`, that is, the same selection applied to the two branch derivatives, with the condition c contributing nothing. Applied at each of the 124 such branches in the reimplementa- tion, that rule is what produces Eq. (2).

2.3 The adjoint of the simulation

The forward pass answers “what is the forcing?” The backward pass answers “how would a small change to each input change that forcing?” For example, a change in ice content at an early time step affects both the radiation at that step and the later evolution of the contrail. The backward pass accounts for both paths by applying the chain rule from the final output back to the inputs. JAX constructs this calculation from the forward code.



2.3.1 The chain rule over a simulation

205 Write the flight's altitude profile as the control vector u (the profile z of Eqs. (1)–(2), renamed to follow the control-variable convention of data assimilation) and the plume state at step k as x_k , so that one evolution step of Section 2.1 is $x_{k+1} = F_k(x_k, u)$ for the K integration steps $k = 0, \dots, K-1$, and the accumulated forcing is $EF = \sum_{k=0}^{K-1} g_k(x_k, u)$, the sum running over the steps actually integrated, so that the state after the last step adds nothing. Now let λ_k denote the sensitivity of the final EF to the plume state at step k : in words, if the ice content or depth at step k were changed by a small amount, λ_k says how
 210 much the flight's total forcing would change. That sensitivity can be computed backwards. The last state influences only the forcing still to be accumulated, while an early state influences everything downstream of it, so λ_k is obtained from λ_{k+1} by accounting for how step k feeds step $k+1$, plus step k 's own direct contribution to the sum: with $\lambda_K = 0$,

$$\lambda_k = \left(\frac{\partial F_k}{\partial x_k} \right)^\top \lambda_{k+1} + \frac{\partial g_k}{\partial x_k},$$

$$\frac{\partial EF}{\partial u} = \left(\frac{\partial x_0}{\partial u} \right)^\top \lambda_0 + \sum_k \left[\left(\frac{\partial F_k}{\partial u} \right)^\top \lambda_{k+1} + \frac{\partial g_k}{\partial u} \right]. \quad (3)$$

The initialisation term $(\partial x_0 / \partial u)^\top \lambda_0$ accounts for any dependence of the initial plume state on the inputs; it is zero if that state
 215 is fixed. The same chain rule applies when u is replaced by the humidity field q or the parameter vector θ . Equation (3) is the adjoint recurrence (Fig. 3), standard in variational data assimilation (Le Dimet and Talagrand, 1986; Talagrand and Courtier, 1987; Errico, 1997). It requires one backward sweep through the evaluated computation. There is no additional model run for each component of u . A forward derivative, whether finite-difference or tangent-linear (the derivative carried forward through the model alongside the values, reverse mode's forward-mode counterpart), perturbs one input at a time, so it must rerun once
 220 per input. Reverse mode starts from the single output and pushes its sensitivity backwards through each operation exactly once, every input collecting its gradient during that one traversal. For a fixed forward calculation and one scalar output, the backward work is a small multiple of the forward work. The input dimension still affects storage and the cost of producing the gradient array, but it does not multiply the number of simulations. This makes gradients over millions of humidity values practical. The price of reverse mode is memory rather than arithmetic: the backward pass consumes the stored intermediate values of
 225 the forward pass, a cost that grows with the length of the recorded computation. At the sizes used here, whole days of flights evaluated in batches, that storage fits on a single GPU without having to recompute parts of the forward pass to save memory, and the measured costs of Section 3 include it.

2.3.2 Keeping inactive branches numerically safe

This rule guards against a failure mode that silently corrupts gradients and that any differentiation of a thresholded model will
 230 encounter. A `where` evaluates *both* of its branches and then selects, which running forward is harmless: the branch not taken is computed, discarded and never seen. In reverse it is not. Take `where( $x > 0$ ,  $\sqrt{x}$ , 0)` at a negative x , which happens routinely, the whole point of the branch being to avoid that case. The discarded value is NaN, not-a-number, and selecting between the two branch derivatives does not dispose of it, because the derivative of the discarded branch is formed before anything is selected and a floating-point product $0 \times \text{NaN}$ is NaN rather than zero. One NaN entering the backward pass then propagates



235 into every gradient downstream, destroying derivatives that are mathematically perfectly well defined. The cure is to make the
 branch not taken evaluate something safe, where $(x > 0, \sqrt{\text{where}(x > 0, x, 1)}, 0)$: the value is unchanged, since for $x \leq 0$ the
 outer `where` still returns 0, while the discarded computation is now $\sqrt{1}$, whose derivative is finite and is then multiplied by
 zero as intended. The released reimplementation applies this guard wherever a discarded branch can go non-finite, and the test
 suite asserts finite gradients on the scenarios it covers (Sect. 2.5); three paths reached only at fleet scale were guarded after the
 240 campaign had run, and Appendix A2 records what that cost.

2.3.3 Optional smoothing for optimisation

The release also includes an optional smoothed model for optimisation. It replaces on/off indicators with sigmoid transitions,
 providing slopes near thresholds that the original indicators do not provide. This changes the model used during the search. A
 candidate found with smoothing must therefore be evaluated again with the original switches. We call this original version the
 245 *hard model*; “certified” in the figures means evaluated by that model with smoothing disabled. It denotes a model evaluation,
 not an operational safety certification.

All numerical results in this paper use the hard model. The altitude campaign evaluates a fixed set of manoeuvres and does
 not use the smoothed model to optimise a continuous trajectory. Continuous optimisation is the subject of a companion study.

2.4 Scope of the released model

250 The release reproduces the per-flight trajectory configuration of `Cocip` in `PYCONTRAILS v0.63.3`. It does not include grid-
 ded `CoCiP`. Unsupported optional microphysics treatments raise an explicit error. Input conventions, including longitude
 range, vertical coordinate and radiation units, are documented in `docs/model.md`, `docs/porting-rules.md` and
`DEVIATIONS.md`. Two aspects of this scope are particularly relevant to the use cases below.

The first boundary is where the derivatives stop. Aircraft performance and emissions are consumed by the model rather than
 255 computed by it, and only two emitted species reach its physics: the water vapour emission index, which enters the Schmidt–
 Appleman mixing-line slope and the exhaust water loading that sets the initial ice water content, and the non-volatile particulate
 matter (nvPM) number emission index, the number of soot particles emitted per kilogram of fuel, which enters through the
 activated fraction, the fraction of those particles on which ice forms, and so sets the initial ice-particle number. The derivatives
 reach these two inputs and stop there; a sensitivity of EF to fuel composition, engine technology or aircraft type would be this
 260 adjoint composed with the adjoint of a performance and emissions model, and that second factor is not part of this release. The
 ambient cirrus optical depth is likewise consumed as a prepared input field, so no derivative flows through it, and the humidity
 sensitivities of Section 5 hold the ambient cirrus fixed. The two emission inputs the derivative does reach are not equally well
 checked: the nvPM number derivative carries the finite-difference check of Section 2.5 and the water vapour index does not, so
 a sensitivity to the latter is differentiable on the same footing but has not been verified against an independent computation.

265 The second boundary is the humidity correction, because two of the use cases differentiate through it. Specific humidity
 is corrected wherever it is read, at the waypoint, at contrail initialisation, and at every integration step, because reanalyses
 underestimate ice supersaturation (Sect. 1). The correction is the exponential-boost scheme of Teoh et al. (2022): the relative



humidity over ice (RH_i ; $RH_i = 1$ is ice saturation) is first divided by `rhi_adj`; where the result is at or above saturation it is then raised to the power `rhi_boost_exponent` and clipped at `clip_upper`; and specific humidity is recovered from the corrected RH_i . The defaults are `rhi_adj` = 0.97, `rhi_boost_exponent` = 1.7 and `clip_upper` = 1.7 in RH_i . Where the paper speaks of the humidity correction as one of the model's uncertain constants, it means the first of those coefficients, `rhi_adj`, and not the scheme as a whole.

2.5 Verification

The test suite runs the reference implementation, held at PYCONTRAILS v0.63.3, side by side with the reimplementation and compares their outputs at three levels, each tolerance fixed before the first comparison was run: every ported subroutine to a relative 10^{-12} , the step-by-step contrail histories to 10^{-6} , and the end-to-end EF to 10^{-4} , each level looser than the last because it accumulates the arithmetic of the ones before it. On the campaign's busiest day (17,406 flights), every one of the 6,418 contrail-forming flights meets the end-to-end tolerance against a serial reference run, with a median per-flight difference of 2.0×10^{-8} , and no flight forms a contrail in one implementation but not the other.

The reverse-mode derivatives are checked the same way, against central finite differences, over every input the use cases differentiate and over two of the six constants of Section 6 that the test flight exposes directly; the largest disagreement the suite records is a relative 1.5×10^{-8} . All six constants are checked again, the same way, on the campaign flights of Section 6. Because the model has jumps and kinks, a comparison is skipped whenever its two nudged evaluations fall on opposite sides of one, decided from which waypoints form and survive rather than from how far the answers differ, and a comparison whose EF is identically zero is not counted either. Two safeguards sit behind the suite: a recorded run per module in which the code was deliberately broken and the tests rejected it, and a dated ledger of every known deviation from the reference implementation.

One defect survived into the campaign record rather than into the released model, and Appendix A sets it out: for 38 of the 398,467 campaign flights ($<0.01\%$), the backward pass as run returned non-finite gradients, from three code paths since guarded. What these checks verify is narrow. It is that the reimplementation computes the reference model's forcing, and that its derivatives are the derivatives of what it computes; whether the reference model is right about the atmosphere is a separate question that this paper does not address.

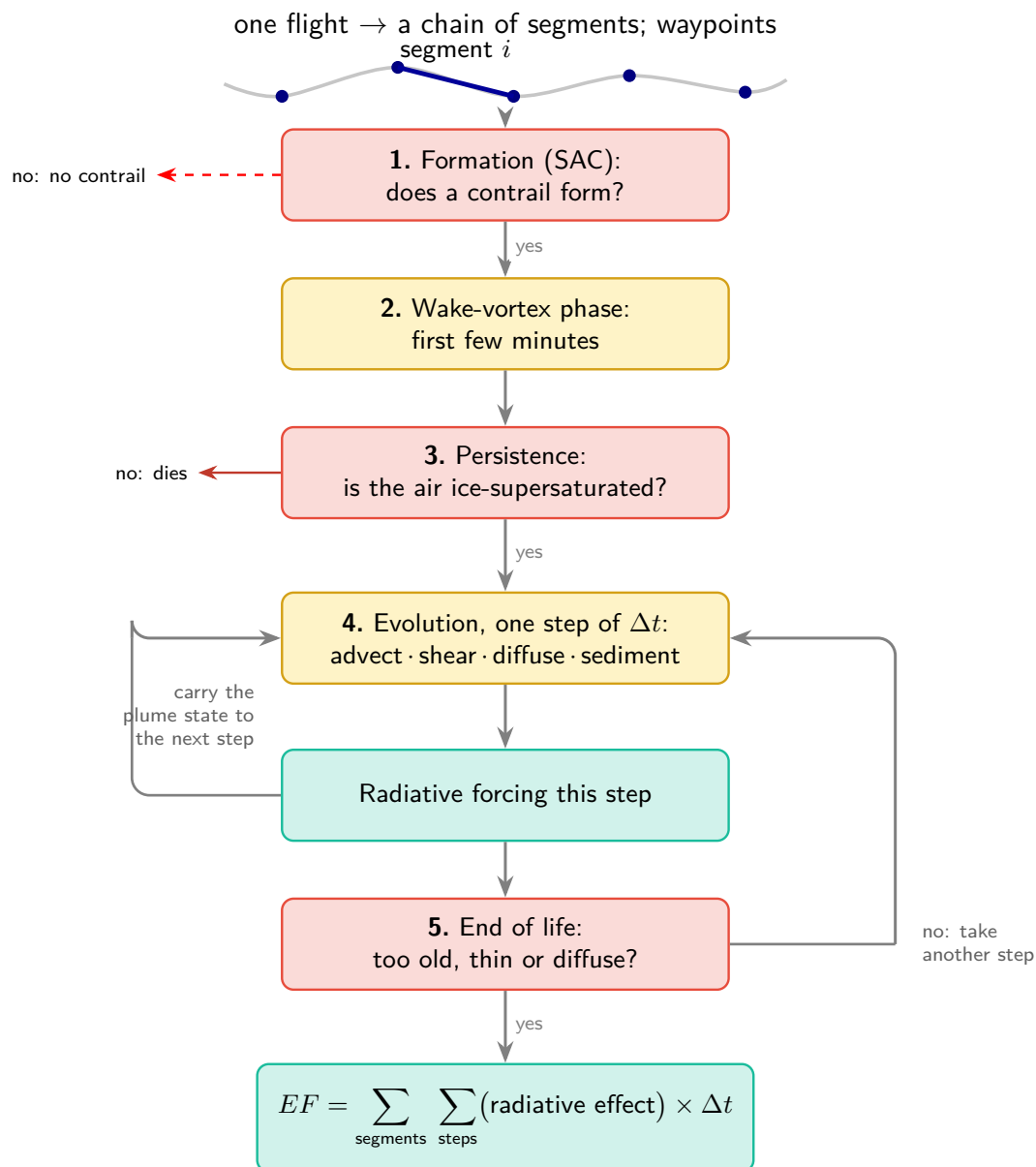


Figure 1. How CoCiP computes one flight’s energy forcing. Each segment joins two consecutive flight waypoints. Formation and initial-persistence tests determine whether its exhaust enters the evolution loop. Surviving contrails move with the wind and change their ice content and dimensions; their radiative contribution accumulates until an end-of-life test stops the calculation. Summing over time and flight segments gives EF in joules. The red outlines mark the three groups of yes-or-no tests that can introduce jumps in EF . Table 1 lists all six end-of-life conditions; the drawing shows three examples.

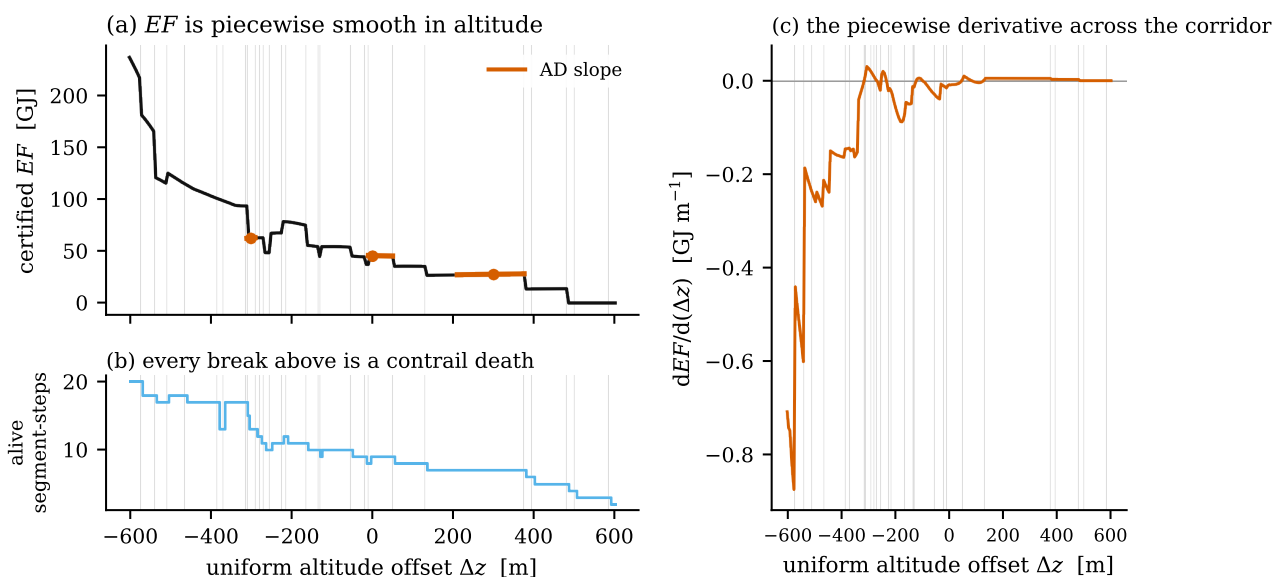


Figure 2. Energy forcing and its derivative as a synthetic 20-waypoint flight is shifted vertically. All panels use the same uniform offset Δz , scanned in 5 m steps over ± 600 m. (a) EF from the unmodified model (labelled “Certified”), with automatic-differentiation slopes at -300 , 0 and $+300$ m. Each slope is tangent to its local smooth piece. (b) Number of contributing survival states: one segment-step is counted for each segment alive at each 30 min step. Grey lines mark changes in this count. All nineteen segments pass the formation test throughout the scan, so the changes arise from persistence and lifetime. (c) The local derivative $dEF/d(\Delta z)$. Its sign and magnitude need not describe the trend across several jumps. The synthetic flight’s forcing is shown in gigajoules, whereas the real-flight examples below use terajoules.

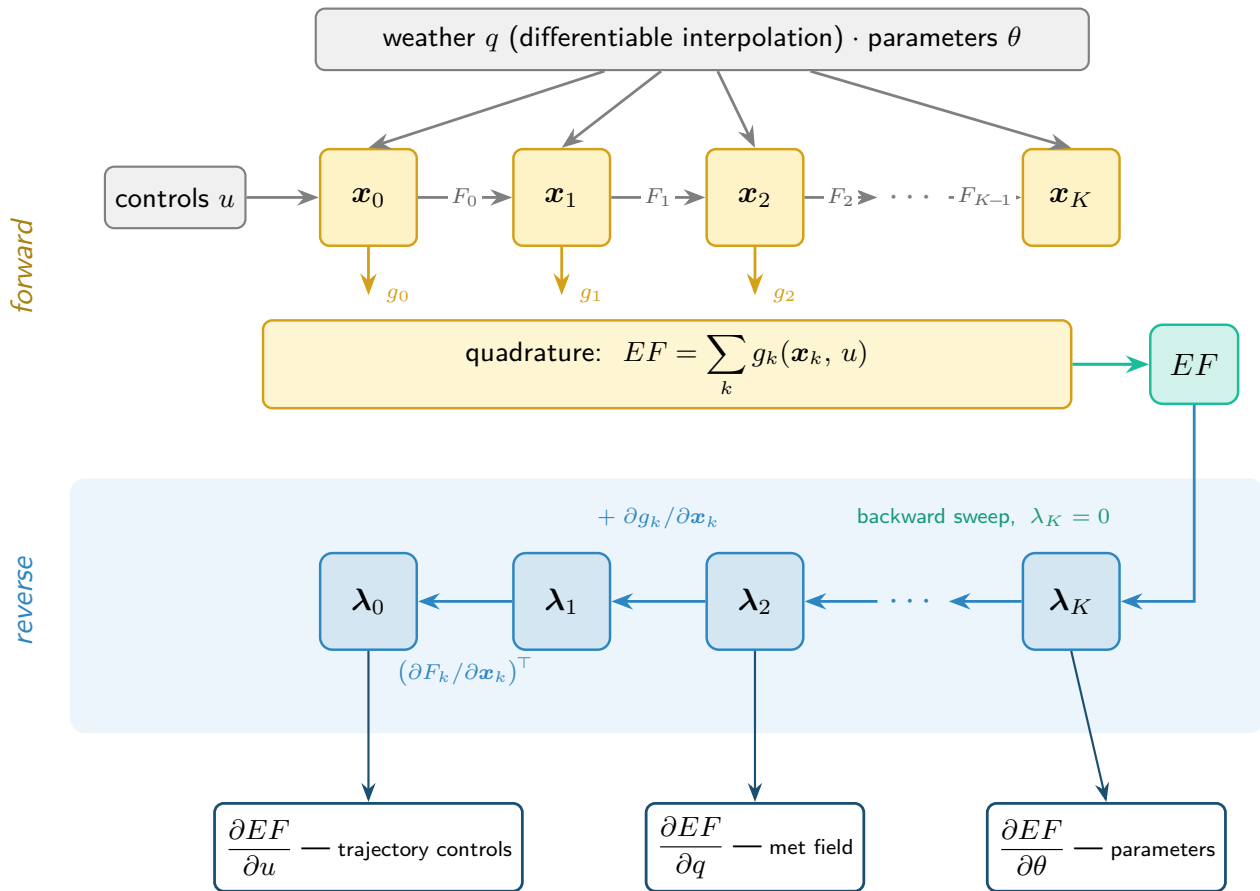


Figure 3. Forward simulation and backward sensitivity calculation. Top: each step updates the plume state x_k and contributes to the energy-forcing sum. Trajectory controls u , weather q and parameters θ affect this calculation. The controls are drawn entering once for readability, but may affect initialisation and subsequent steps. Bottom: the adjoint propagates sensitivities backwards and accumulates derivatives for all three input groups. The terminal sensitivity is $\lambda_K = 0$ because the final state adds no separate forcing term. The derivative of each step's forcing contribution supplies the source term in Eq. (3); the arrow from EF marks the start of the reverse calculation.



3 The model at fleet scale

The three use cases need a common setting: a fleet, a day, a weather extract and a model configuration. This section fixes it, and measures what evaluating and differentiating a whole day of traffic costs.

295 Traffic comprises all EUROCONTROL flight plans filed for the European core area (35° to 60° N, 10° W– 30° E) above 15 000 ft on the 28 campaign days (the weeks of 1–7 January, 8–14 April, 8–14 July, and 7–13 October 2024), resampled to 60 s waypoints. We use filed plans rather than flown trajectories, the traffic record available to us, and call each flight’s unmodified altitude profile its *filed profile*; per-day evaluated counts range from 7,182 to 17,406 after filters that drop flights outside the spatial window or operated by aircraft types the performance model does not support. Since a filed plan gives a
 300 route and a level but no fuel burn, aircraft performance (true airspeed, fuel flow, engine efficiency, mass) is computed with the Poll–Schumann model (Poll and Schumann, 2021a, b) as implemented in PYCONTRAILS, and soot is not resolved per engine: every flight is given the same nvPM number emission index of $1.5 \times 10^{15} \text{ kg}^{-1}$ of fuel and fixed fuel constants representative of Jet A, so the shortlists do not represent engine-specific differences in soot emissions. Aircraft performance and geometry still differ between aircraft types. Flight-plan data are proprietary to EUROCONTROL; all statistics reported here are aggregates,
 305 and no flight identifiers are published, the case-study flights of Sections 4.2 and 6 being reported by aircraft type and date alone.

The air those flights fly through is the ERA5 reanalysis (Hersbach et al., 2020), on its native 0.25° latitude–longitude grid, hourly, on the eight pressure levels from 150 hPa to 400 hPa that span the cruise band. Each day is extracted over a 30 h
 310 window so that late departures and their contrails remain inside it, and each extract is $31 \times 8 \times 101 \times 161$ points in time, pressure level, latitude and longitude, so the specific-humidity field alone carries 4,032,728 values. That count is the input dimension of $\partial EF / \partial q$, and the reason a finite-difference version of it is out of reach. Specific humidity is corrected with the scheme of Section 2.4 at its default coefficients ($\text{rhi_adj} = 0.97$, boost exponent 1.7), a correction of ERA5’s known ice-supersaturation biases anchored to measurements from instrumented in-service airliners (the IAGOS programme) (Teoh et al., 2022).

The campaign uses the default 30 min integration step and a maximum contrail age of 3 h. The latter is shorter than the library
 315 default of 20 h and reduces the number of plume-evolution steps. A longer simulation also needs weather from consecutive daily extracts. Contrails leaving the regional domain are terminated at its boundary, irrespective of the age cap.

These numerical settings affect the results. Appendix B shows that a finer integration step changes absolute forcing but retains most of the warming captured by the shortlist on the day tested. Extending the age cap changes both forcing totals and shortlist membership. All comparisons below use the same configuration on both sides, making them internally consistent.
 320 This does not establish that their magnitudes or rankings would be unchanged with a longer age cap.

We measured computational cost on eight benchmark days from 2024, with two days per season. The following medians exclude a one-off compilation of 2 to 14 minutes. On one NVIDIA RTX PRO 6000 Blackwell GPU, evaluating EF for a day of traffic takes about 18 s; evaluating EF and the full altitude gradient takes about 19 s. On a 24-core Intel Xeon w7-3545, the corresponding times are 24 s and 99 s. The additional backward calculation therefore costs about 1 s on the GPU and 75 s



325 on the CPU: figures specific to this hardware, batching and model configuration rather than universal overheads, but ones that show the altitude gradient can be computed alongside forcing at fleet scale.

4 Use case 1: the altitude gradient

For each waypoint altitude z_i , the gradient entry $\partial EF / \partial z_i$ describes the local change in total forcing when that altitude changes and the other CoCiP inputs are held fixed. A negative entry favours a small climb at that waypoint, a positive one a small descent; a large magnitude, meanwhile, identifies a location where the baseline prediction is locally sensitive to altitude.

We use this information to place an altitude-change window within a flight, then evaluate full manoeuvres of 1000 ft and 2000 ft. The campaign serves two purposes: it demonstrates this placement method at fleet scale and measures how well a local linear prediction describes the evaluated finite changes. It does not seek an optimal continuous trajectory. Appendix A gives the full protocol.

335 4.1 Fleet evaluation across four seasons

We evaluated every flight of the 28 days that survives the filters of Section 3, 398,467 filed flight plans in all. Daily totals vary by more than an order of magnitude (gross warming between roughly 19 PJ and 576 PJ), and warming and cooling can nearly cancel (on one July day the net forcing is a tenth of the gross), which is why fleet-level shares in this paper are quoted against gross warming (Section 2.1). The campaign rests on one premise, that a day's warming concentrates in a small minority of its flights, and it holds on every one of the 28 days: the top 1 % of flights by evaluated EF accounts for a median of 26 % of daily gross warming and at least 15 % on every day; summed over the whole campaign it is 23.4 %.

The campaign has two stages. First, the screening stage computes every flight's EF and selects the highest-forcing 1 % of flights on each day. Second, the evaluation stage compares eight candidates for each selected flight:

- four whole-cruise shifts: -2000 , -1000 , $+1000$ and $+2000$ ft, retaining the shape of the filed cruise profile;
- 345 – four windowed shifts of the same sizes, which leave the filed level over part of the cruise and return to it afterwards.

For the windowed candidates, we accumulate $|\partial EF / \partial z_i|$ along the route. The window runs between the points at which this cumulative sum reaches 10 % and 90 % of its total, with entry and exit ramps added as described in Appendix A2. Thus the gradient determines the window, while both climb and descent candidates are evaluated. Its sign does not exclude either direction.

350 Aircraft performance and fuel are recomputed for each candidate with the Poll–Schumann model. The reference CoCiP implementation then evaluates the candidate's forcing. We retain the candidate with the lowest EF if it improves on the filed profile. Fuel change is reported separately; it is not the selection objective. Credited avoided warming is capped at the filed flight's positive forcing, so cooling beyond zero receives no additional credit.

Across the four weeks, 3,973 flights were shortlisted and 3,958 had a candidate that reduced forcing. Of these 3,958 flights, 355 3,402 (86 %) selected a whole-cruise shift and 556 (14 %) a windowed shift. The median reduction relative to each flight's



own filed forcing was 95 % for the former group and 70 % for the latter. These two medians come from different selected groups, so comparing them does not estimate the benefit of window placement specifically. The comparison below assesses linear predictions using all evaluated windowed candidates, including those not selected.

4.2 Accuracy of first-order predictions for altitude changes

360 For a candidate with altitude changes Δz_i , the gradient gives the first-order prediction

$$\Delta EF_{\text{pred}} = \sum_i \frac{\partial EF}{\partial z_i} \Delta z_i. \quad (4)$$

The evaluated change is $\Delta EF = EF_{\text{candidate}} - EF_{\text{filed}}$. Both quantities use the same sign convention: a negative value means a reduction. Equation (4) extends the local slope at the filed profile over the full proposed altitude change; the model evaluation follows the candidate’s actual contrail history.

365 We compare these quantities for the windowed candidates at each size, including both directions for every shortlisted flight. None of the 38 flights with non-finite campaign gradients reached the shortlist (Appendix A2), so each size has two candidates for each of the 3,973 shortlisted flights: 7,946 candidates in all. The pooled fitted slope of evaluated versus predicted change is 0.13 at 2000 ft and 0.45 at 1000 ft. The latter is 3.5 times larger (Appendix A3). A slope of 0.13 means that the fitted response is about one eighth of the linear prediction. It is not a per-flight ratio or a correction factor that makes every prediction accurate.

370 This discrepancy is consistent with the model’s threshold structure: an altitude change can stop an existing contrail, create one elsewhere or change its lifetime. Smooth nonlinear responses can also depart from a straight-line extrapolation. In addition, candidate evaluations recompute aircraft performance, whereas the CoCiP altitude derivative does not differentiate through that performance model (Sect. 2.4). The comparison therefore measures the predictive value of the gradient in the full candidate workflow; it does not attribute every discrepancy solely to a threshold crossing.

375 The sign can differ as well. Among all 3,958 flights with a selected improvement, 1,299 (33 %) selected a candidate for which the gradient predicted an increase in forcing. This fraction ranges from 20–48 % across the four weeks. Unlike the fitted slopes above, this statistic includes selected whole-cruise as well as windowed candidates.

380 Figure 4 dissects one intervention of each kind end to end, on a common three-row layout: the vertical cross-section (“curtain”) of RHi along the track, with the filed and reprofiled profiles; the evaluated per-waypoint forcing before and after the manoeuvre; and the altitude gradient that placed the manoeuvre window. The first example shows agreement between the direction favoured by the gradient and the evaluated outcome; the second shows a reversal.

Figure 4a shows a case with agreement: a Boeing 747-8 freighter (ICAO type B748) crossing the Iberian peninsula on 1 January 2024 through moist air with drier air above it; the reprofiled stretch at flight level 350 (FL350, i.e. 35 000 ft pressure altitude) sits in drier air than the filed FL330 over most of its length. The gradient favours climbing, and the evaluation agrees: every climb candidate beats the filed profile, and the windowed 2000 ft climb shown removes 784 TJ of the flight’s 1152 TJ while *saving* 243 kg of fuel, the higher level sitting nearer the aircraft’s optimum. The agreement is not uniform along the track: one waypoint near 1400 km carries a large positive spike, a place where a climb would thicken the contrail rather than thin it,

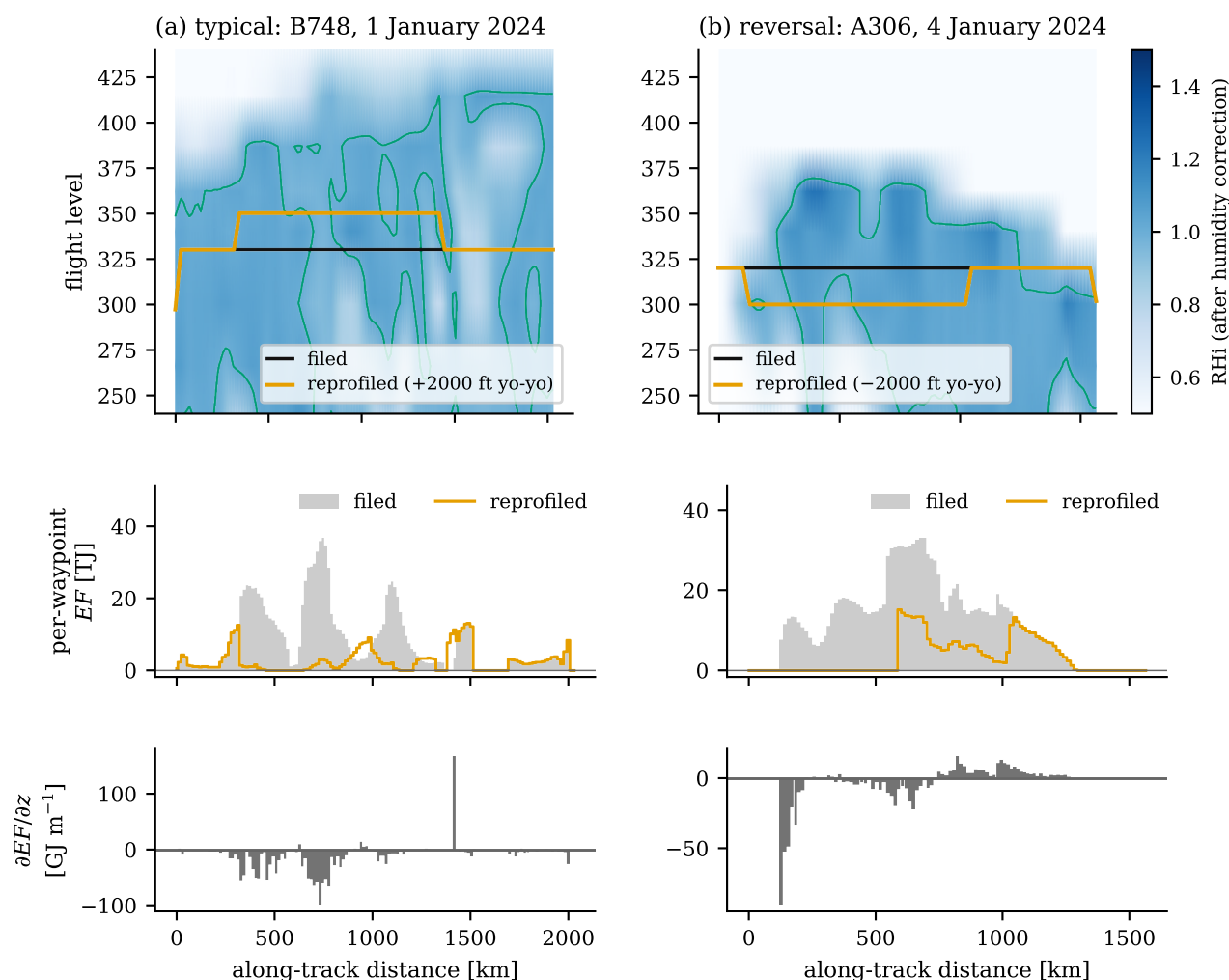


Figure 4. Two interventions in detail: (a) agreement, a B748 on 1 January 2024, in which the gradient’s prediction and the evaluated outcome agree; (b) a reversal, an A306 on 4 January 2024, in which they disagree. Top: relative humidity over ice (RH_i; 1 is ice saturation, after the humidity correction of Sect. 2.4) along the track, as a vertical cross-section in flight level (hundreds of feet of pressure altitude), with the filed and reprofiled profiles; the filed profile is hidden where the two coincide, and the green contour marks ice saturation. Middle: evaluated per-waypoint forcing before (grey fill) and after (orange) the manoeuvre. Bottom: the altitude gradient of the filed profile, the quantity that placed the manoeuvre window (negative values favour climbing); it is not recomputed for the reprofiled trajectory. In (a) a 2000 ft windowed climb moves the cruise into markedly drier air, as the gradient predicted; in (b) a 2000 ft windowed descent drops the flight out of the supersaturated layer at its lower edge, against the gradient’s prediction. Both panels plot the windowed candidate, which isolates the manoeuvre window; on the flight in (b) the candidate the workflow keeps is the whole-cruise descent, which removes slightly more for more fuel.



but it lies at the edge of the window and the gradient summed over the window still favours climbing. This example illustrates agreement in direction; the full evaluation supplies the magnitude of the reduction.

390 Figure 4b shows the exception: an Airbus A300-600 freighter (A306) on 4 January 2024, the winter week’s largest evaluated reduction. Over the first two thirds of the track the aircraft flies at its filed FL320, within ice-supersaturated air; the gradient’s magnitude concentrates there, the window is placed there, and the gradient’s sign, summed over the window, points to a climb. The evaluated best manoeuvres are nevertheless the *descents*: the windowed 2000 ft descent shown removes 915 TJ of the flight’s 1289 TJ for 108 kg of additional fuel, the whole-cruise descent, the candidate the workflow selects here, removes
 395 slightly more at 184 kg, and both beat every climb. The middle panel shows where that reduction comes from: over the first part of the track the descent removes the flight’s forcing outright, and it adds a little back late, where the reprofiled trace runs above the filed one. The local slope and the evaluated descent answer different questions. At the filed profile, the slope favours a small climb. The full descent to FL300 instead reaches the lower edge of the supersaturated layer and removes persistent contrails over part of the track. That threshold crossing is not described by the local slope. The humidity cross-section helps
 400 explain the descent’s benefit, but does not by itself establish why the total local derivative favours climbing. Both the original profile and the proposed profile must be evaluated to quantify the finite change.

These examples explain why the workflow evaluates both directions. The gradient supplies a candidate window and a local prediction; the full evaluation determines the result of each finite manoeuvre. The principle resembles model-management approaches in which an approximation proposes candidates and a full calculation evaluates them (Alexandrov et al., 1998;
 405 Peherstorfer et al., 2018). Here the approximation is the local linear response, rather than a separate reduced model. The experiment supports this propose-and-evaluate use of the gradient; it does not establish that its window is globally optimal.

5 Use case 2: forecast sensitivity

The second use case asks which humidity values influence a flight’s forcing. The input q is the entire four-dimensional humidity array: longitude, latitude, pressure level and time. Each component of $\partial EF/\partial q$ describes a small change to one stored grid
 410 value, with the other inputs fixed. Computing the complete gradient by central finite differences would require two model runs for each of the 4,032,728 values. Reverse mode obtains it in one backward pass.

We compute that pass for the flight this paper already follows, the A306 of Section 4.2, on its own day and on the full campaign grid. The extract is a reanalysis, so its fields are reconstructions of past weather rather than forecasts; the same pass on a forecast field would give that forecast’s own sensitivity, with the plume positions following the forecast winds. It returns
 415 $\partial EF/\partial q$: for every one of the 4,032,728 humidity values in the extract, the change in this flight’s forcing per unit change in that value, all obtained together. Specific humidity is a mass fraction, so each is a number of joules per kilogram of water per kilogram of air, and the largest is 5.4×10^{17} in those units. The result is extremely sparse. Only 2,196 values (0.054 %) are non-zero, spanning 685 of the 16,261 horizontal grid columns and confined to the 250 hPa and 300 hPa pressure levels, two of the extract’s eight (Fig. 5). This sparsity follows from the local weather interpolation. Each lookup uses the neighbouring grid
 420 points in four dimensions, giving sixteen corners per interpolation stencil. Only values connected to the contributing contrail

history can have non-zero sensitivity. Even a value that is read can have zero derivative if its effect is blocked by an inactive branch or a clipped correction. Thus, a zero entry means no first-order effect in this baseline calculation; it does not prove that a large humidity change there could never matter. The non-zero entries in this example lie within the two stated pressure levels and along the plume's drift over the 3 h maximum life. A longer contrail history can involve more locations and times.

425 The sensitivities are spot-checked against finite differences on this grid: ten randomly chosen nonzero entries agree to better than one part in a million, and perturbing zero-entry cells leaves EF exactly unchanged.

The sensitivity map extends beyond the flight track, following the sequence of positions the contrails occupy as the wind carries them, several hundred kilometres east of where the aircraft flew, one arc per hour of the extract, since each hour's field is read only where the plume sits at that hour. The dependence therefore runs forward in time as well as outward in space: the
 430 forcing depends on the humidity where the contrail *will be* over its hours of life, not where the aircraft is, and the adjoint is the object that carries those future positions, having been computed backwards through the advection.

Two details fix what this map means. First, the quantity being differentiated: the stored reanalysis humidity values, not an independently varied corrected field, since the humidity correction sits inside the computation and the chain rule already carries its effect, leaving the gradient with respect to the uncorrected input array. Second, the map is local, valid for the filed flight and
 435 its baseline contrail history. Humidity changes large enough to alter formation or survival can produce effects that this map does not predict.

The map also has a quantitative use. Forecast humidity error is not small (Gierens et al., 2020; Teoh et al., 2022), and its statistics are, in data assimilation practice, summarised by an error covariance $\mathbf{B}$, the symbol data assimilation uses: a matrix stating how wrong each humidity value tends to be, and which values tend to err together. Given $\mathbf{B}$ and the gradient
 440 $\mathbf{g} = \partial EF / \partial q$, first-order propagation gives the resulting uncertainty on the forcing as $\text{Var}(EF) \approx \mathbf{g}^\top \mathbf{B} \mathbf{g}$: one number per flight, the variance of its predicted forcing induced by forecast humidity error, obtained without additional contrail-model runs once the gradient and covariance are available. Because the gradient is sparse, only the few thousand cells that carry it ever enter the product. We do not test the product against a forecast ensemble here, a set of forecasts whose spread is used to estimate forecast uncertainty; the Discussion sets out how that would be done.

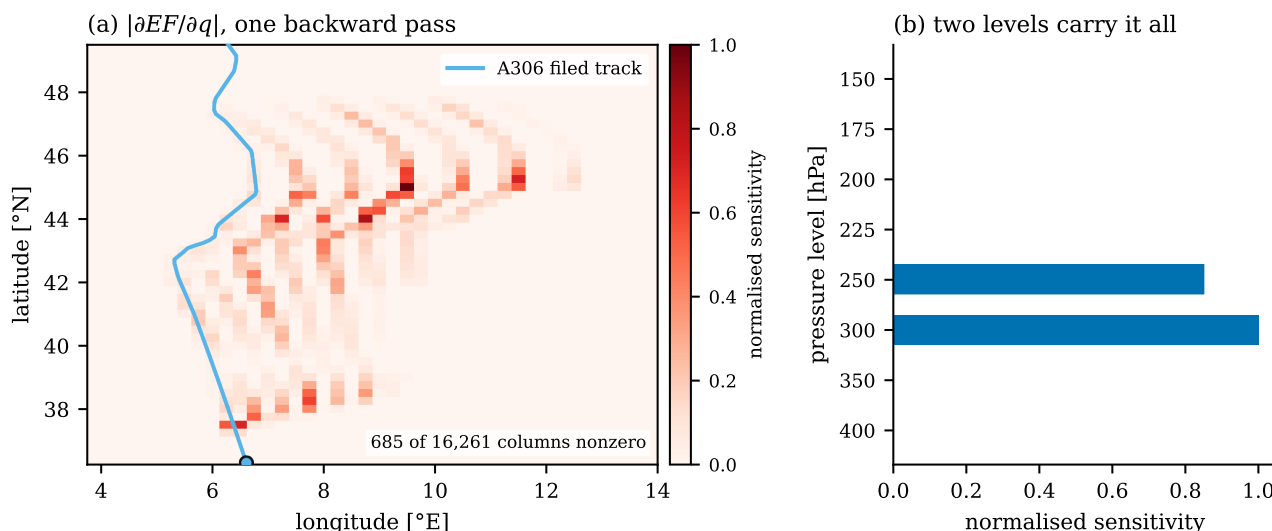


Figure 5. The A306’s forcing sensitivity on the real campaign grid, from one backward pass along the filed profile. (a) $|\partial EF/\partial q|$, normalised to the field maximum, reduced by taking the maximum magnitude over pressure levels and hours; the flight track in blue, with a circle at its start. The repeated arcs east of the track are the plume at successive hours of the extract: the forcing depends on humidity where the contrail *will be*, not where the aircraft is. 2,196 of 4,032,728 values are non-zero; the panel annotation counts columns instead, those values falling in 685 of the 16,261 horizontal grid columns. The view is zoomed to the active region: the extract spans 10° W–30° E and 35°–60° N, and the field is zero everywhere outside the view. (b) The same field’s maximum magnitude at each pressure level, taken over all cells and hours and normalised to the largest level: the entire sensitivity lies on the 250 hPa and 300 hPa surfaces.

445 6 Use case 3: parameter sensitivity

The final use case compares the local influence of six empirical constants (Table 2). These constants describe processes that Co-CiP does not resolve explicitly, such as turbulence and humidity correction. One backward pass per flight returns the derivative of EF with respect to all six.

This is a selected parameter set, rather than an exhaustive sensitivity analysis. We keep the humidity boost exponent and upper clip fixed; these act only in particular humidity regimes, whereas `rhi_adj` rescales humidity on each read before
 450 boosting and clipping. The ice-number floor is also excluded because it is inactive at the soot levels used here. A parameter can have zero local sensitivity where the relevant process or branch is inactive; such a zero does not make its sensitivity mathematically invalid.

The ranking measures the response to a relative parameter change. Determining which parameter contributes most uncer-
 455 tainty would need more than that: estimates of each parameter’s own uncertainty, and of how the parameters depend on one another. Deciding which measurement would be most useful runs into the same limit, needing information beyond the derivative alone.



Table 2. The six uncertain scalar constants of CoCiP whose elasticities Section 6 ranks, with the names the reference implementation gives them. Each stands in for physics the model does not resolve. The last column records where the constant’s derivative was compared with central finite differences: “suite” is the frozen gate of Section 2.5 and “flights” the check of Section 6, which covers all six, so the two marked both are covered twice. All six meet the reported numerical checks.

Constant	Stands in for	FD check
Humidity correction (<code>rhi_adj</code>)	the rescaling of the reanalysis humidity each time the model reads it (Section 2.4)	flights
Sedimentation impact factor	how strongly falling ice crystals deepen the plume	suite, flights
Initial wake-vortex depth	the depth of the young contrail when the wake-vortex phase of Section 2.1 ends	suite, flights
Shear-enhancement exponent	the amplification of the gridded wind shear that stands in for fluctuations the grid cannot resolve	flights
Turbulence scale	the assumed magnitude of turbulent vertical air motion	flights
Effective vertical resolution	the depth over which the weather data are treated as resolving vertical structure, which sets how sharply humidity and shear may vary between levels	flights

We checked all six derivatives on the five winter case-study flights against central finite differences, using nine step sizes spanning four decades. This gives 30 parameter–flight comparisons and 270 step-level checks. As in Sect. 2.5, comparisons that cross a switching boundary are excluded. Every comparison agrees to within a relative 10^{-6} over at least four consecutive step sizes. At the closest agreement for each comparison, the relative difference is 1.0×10^{-8} or smaller.

The natural unit for the answer is the *elasticity* $\theta_j/EF \cdot \partial EF/\partial \theta_j$: the percentage change in the flight’s forcing per one percent change in constant θ_j . Elasticities are dimensionless, so constants with different units and magnitudes can be compared on one axis: an elasticity of -6 means that a value one percent too high would have lowered the predicted forcing by about six percent, and one of $+0.1$ that the same error moves the answer by a tenth of a percent.

We compute all six elasticities for the five most-warming flights of 4 January 2024, the winter case-study day of Section 4.2, at the campaign configuration (Fig. 6). Which constant leads is unambiguous and the same on all five flights, though the three smallest overlap between flights and do not keep a fixed order among themselves. The humidity correction dominates at an EF -weighted elasticity of -6.6 (per-flight magnitudes 5.1 to 9.7): the local linear response predicts a five-to-ten-percent change in these flights’ forcing for a one-percent change in that constant. The sign is negative because the correction of Section 2.4 divides RHi by this constant, so a larger value means drier air and less forcing; the magnitude is large, we suggest, because ice growth and persistence depend on the excess of RHi over one, of which a one-percent rescaling of RHi is a far larger fraction. Every other constant sits far below it. The next largest, the turbulence scale at $+0.92$, is a factor of seven smaller, followed by the shear-enhancement exponent at $+0.62$. Nor is the ranking a property of one winter day: repeated on the campaign’s summer day with the largest gross warming, on five summer flights, the same ordering of the six forcing-weighted elasticities

is recovered, with the humidity correction still dominant at -7.1 and only the small wake-vortex entry flipping sign. These elasticities measure the local smooth response. Finite changes that cross thresholds need separate evaluation (Sect. 2.2).

The two humidity use cases describe complementary sensitivities. Section 5 varies individual weather-grid values and shows where the prediction is sensitive, whereas `rhi_adj` changes the correction applied wherever humidity is read: its large elasticity shows that this shared rescaling strongly affects the selected flights at the baseline configuration.

Elasticity is a local measure. Global sensitivity analysis (Saltelli et al., 2008; Pianosi et al., 2016) addresses a different question by sampling a specified range or distribution of parameter values. The adjoint makes the local calculation inexpensive for each flight; it does not replace an analysis of responses across thresholds or across broad parameter ranges.

The parameter sensitivities demonstrated here provide a basis for calibrating CoCiP against observations of contrail evolution. A natural extension is to apply the differentiable formulation to observable properties, such as contrail width and optical depth, and to objectives measuring discrepancies between simulations and observations. Gradients of these objectives could support the estimation of parameters controlling contrail spreading, turbulent mixing and sedimentation across multiple observed contrails. This application, which is not evaluated here, would require accounting for observational and meteorological uncertainties, relating model outputs to the measured quantities, and assessing whether the observations can distinguish the effects of individual parameters. The optional smoothed formulation may ease optimisation when parameter changes cross formation, persistence or termination thresholds. Such calibration objectives still need to be developed and evaluated.

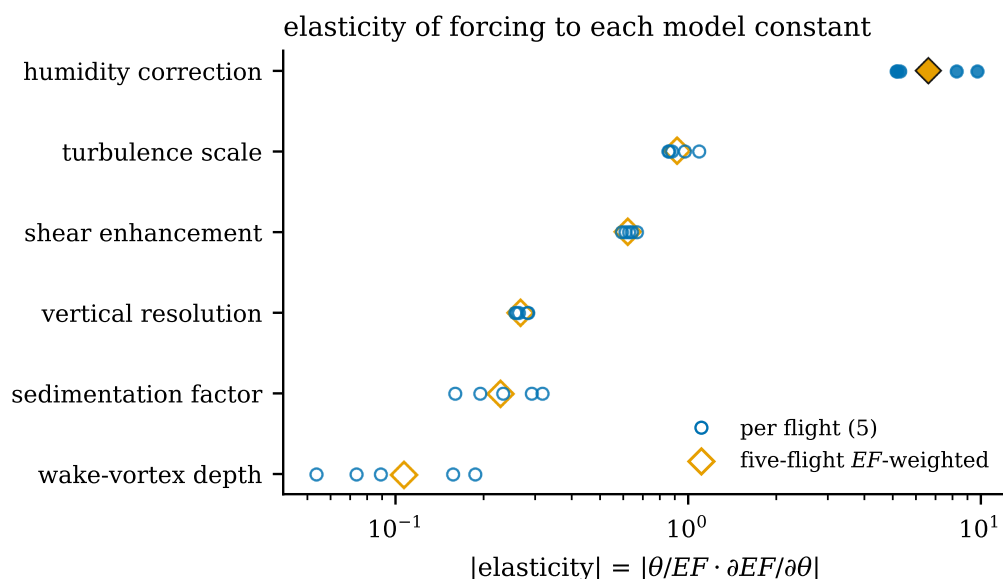


Figure 6. Elasticity of each flight’s forcing to CoCiP’s six uncertain scalar constants, $|\theta/EF \cdot \partial EF/\partial \theta|$, for the five most-warming flights of 4 January 2024 (Section 4.2; circles) and their aggregate weighted by each flight’s forcing (diamonds), log scale. Three of the five flights sit within 0.2 of one another in the top row and overlap there. Filled markers denote negative elasticities, open markers positive; only the humidity correction is negative, because raising that constant dries the air the model sees and so lowers the forcing. All six derivatives of one flight come from a single backward pass.

7 Discussion

DIFFCOCIP adds local sensitivities to the forward CoCiP calculation. Its value differs across the three applications. For altitude changes, the gradient identifies where a baseline flight is sensitive and can guide the placement of candidate manoeuvres. For a humidity field with millions of values, reverse mode makes the complete sensitivity map practical. For a small set of empirical constants, finite differences remain feasible, while the adjoint returns the local ranking in one pass. These uses share a common interpretation: the derivative describes the smooth response around the inputs of the evaluated simulation.

The altitude campaign illustrates why this interpretation matters. The fitted slope between evaluated and predicted changes is 0.13 for 2000 ft windowed manoeuvres, and about one third of the selected improvements have a predicted change of the wrong sign. The latter statistic includes both windowed and whole-cruise candidates. Neither result contradicts the derivative checks (Sect. 2.2): formation and survival thresholds, smooth nonlinearities and the recomputation of aircraft performance all distinguish a finite candidate evaluation from the local CoCiP slope.

The campaign also separates flight screening from manoeuvre placement. Flights are selected by evaluated forcing. The gradient is used only to place the windowed candidates; whole-cruise candidates require no gradient and are selected for 86 % of the flights whose profiles change. The results therefore demonstrate a practical way to use altitude sensitivities, rather than



proving that gradient placement is necessary for avoidance or that it finds the optimal window. A continuous trajectory search would pose a separate problem. The optional smoothed model supplies gradients near thresholds, and each resulting candidate would still need evaluation with the hard model. A companion study examines that optimisation problem.

The humidity map opens a further application: first-order uncertainty propagation. With a humidity-error covariance $\mathbf{B}$ and gradient $\mathbf{g}$, the approximation $\text{Var}(EF) \approx \mathbf{g}^T \mathbf{B} \mathbf{g}$ can provide a forcing-variance estimate without evaluating CoCiP for every weather-ensemble member. This extends a standard forecast-sensitivity diagnostic (Langland and Baker, 2004) to contrail forcing. It requires validation against the spread obtained by running the model across an ensemble, particularly where members change formation or survival outcomes. The present paper establishes the required gradient, not the accuracy of that variance approximation. A companion study addresses humidity uncertainty and this validation.

Several features of the experiment define the scope of its results. The weather is a reanalysis, and candidates are selected and scored against that same weather: the campaign does not measure performance under forecast error, a constraint that operational trials identify as important (Dean et al., 2025; Sausen et al., 2024; Sonabend-W et al., 2024). Traffic, too, is bounded in scope, comprising filed plans in one European region over four weeks of 2024, so the results are not observations of flown avoidance manoeuvres, nor a global climatology.

The candidate set changes altitude while holding track and waypoint times fixed: it does not include lateral rerouting, nor interactions between flights such as traffic conflicts and shared airspace constraints. Aircraft climb capability and the speed envelope are likewise unchecked beyond the performance model's validity range, and the fixed fleet-wide soot number emission index leaves out engine-specific emissions differences. Extending the workflow to operational planning would need to address each of these.

The derivative inherits the forward model's assumptions and uncertainties, especially those associated with humidity and contrail microphysics (Gierens et al., 2020; Teoh et al., 2022; Schumann et al., 2017). Numerical settings matter as well. Appendix B shows that absolute forcing changes with the integration step, and that the 3 h age cap is shorter than needed for convergence in the samples examined. A common configuration supports comparisons within this study, but does not guarantee that those comparisons remain unchanged under a different configuration. We report energy forcing directly rather than converting it to a single CO₂ equivalent, which would require additional assumptions about climate response and contrail-cirrus efficacy (Bickel et al., 2020).

Operational trials have demonstrated that small vertical deviations can be flown (Sausen et al., 2024; Sonabend-W et al., 2024), that contrail avoidance can be incorporated into commercial flight planning (Martin Frias et al., 2024), and that dispatcher-led avoidance can work at network scale (Sankar et al., 2026). Efficient sensitivities could complement these workflows by pointing to the weather inputs that most influence forcing and by feeding the uncertainty estimates and candidate generation discussed above. Derivatives with respect to soot emissions could also quantify local responses to changes in the emission inputs, subject to the performance and emissions boundaries described in Sect. 2.4. Evaluating these uses under forecast error and operational constraints is the next step.



8 Conclusions

540 DIFFCOCIP v1.0 provides a verified differentiable implementation of the per-flight CoCiP model. Reverse-mode automatic differentiation returns the sensitivity of a flight’s energy forcing to its waypoint altitudes, gridded humidity inputs and empirical parameters. The forward calculation reproduces PYCONTRAILS v0.63.3 within the stated tolerances, and independent finite-difference checks verify the local derivatives. In the reported GPU benchmark, computing a day’s forcing and altitude gradients takes about 19 s, compared with 18 s for forcing alone.

545 The three applications give concrete uses for this capability. Altitude gradients provide a rule for placing candidate manoeuvre windows, while full model evaluations determine whether the manoeuvres help. The campaign shows that finite flight-level changes can differ substantially from local linear predictions. The humidity gradient locates a case-study flight’s sensitivity within a small part of a four-million-value weather grid, including the later positions of its drifting contrails. Parameter elasticities identify humidity rescaling as the strongest local influence among the six constants examined on the selected winter and
 550 summer flights.

These derivatives are exact within the model’s smooth regions, not across its jumps (Sect. 2.2), and so establish local responses, input attribution and candidate generation rather than predictions across thresholds. The released model and verification suite provide a basis for further work on trajectory optimisation, forecast uncertainty and parameter estimation, with finite changes checked against the full contrail simulation.

555 Appendix A: Campaign protocol and linearisation statistics

This appendix specifies the campaign that Section 4 summarises: its two workflow stages, with the sign conventions and counting rules they apply, and the statistics behind the linearisation error of Section 4.2.

A1 The screening stage

The screening stage answers one question per flight, cheaply enough to ask it of the whole fleet every day: how much does
 560 this flight warm? One forward pass evaluates every flight’s EF (Sect. 3); each day’s flights are ranked by that evaluated EF , and the top 1 % advance to the evaluation stage of Appendix A2. Ranking on the evaluated objective itself rather than on a proxy is affordable because that pass takes seconds. Every flight in the resulting campaign shortlists has positive EF ; the few shortlisted flights that end the week unprofiled are those for which no candidate improves on the filed plan.

A2 The evaluation stage

565 Section 4.1 describes the eight candidates; two details of their construction belong here. Whole-cruise shifts preserve the filed step-climb shape, and the manoeuvre window of a windowed candidate is the contiguous waypoint band containing the central 80 % of the cumulative magnitude of $|\partial EF / \partial z_i|$, with entry and exit ramps limited to 1000 ft min^{-1} at the 60 s waypoint spacing.



One residual affects the gradient this placement needs. For 38 of 398,467 flights ($<0.01\%$), the backward pass returned non-
 570 finite values in the campaign as run. The test suite of Section 2.5 checks that gradients are finite, but only on the test flights and
 weather it ships with; these 38 flights reached, at fleet scale, three code paths those tests do not exercise (in formation, humidity
 scaling and interpolation), where an unguarded discarded branch produced NaN and, as Section 2.3 explains, contaminated the
 whole gradient. All three were guarded after the campaign record was frozen, and re-running the 38 flights against the released
 version returns finite gradients for every one of them, at forward EF unchanged. The campaign was not regenerated, so these
 575 flights carry no gradient in the campaign record; they keep their forward evaluation and remain in all denominators, and their
 forward values agree with the reference implementation.

For every candidate, ground track and waypoint times are held fixed, so groundspeed is unchanged; true airspeed follows
 from the winds at the new levels, and fuel flow, engine efficiency, and mass are recomputed by the Poll–Schumann model for
 every candidate before evaluation by the reference implementation. The filed profile itself is re-evaluated through the same
 580 pipeline and must reproduce the flight’s screening-stage EF within tolerance, tying the two stages together.

For each candidate we report the evaluated change $\Delta EF = EF_{\text{candidate}} - EF_{\text{filed}}$, so a negative ΔEF is a reduction of the
 flight’s forcing, and the corresponding fuel change from the performance model. The gradient’s first-order prediction of the
 same quantity is $\Delta EF_{\text{pred}} = \sum_i (\partial EF / \partial z_i) \Delta z_i$ with Δz_i the candidate’s actual altitude changes; comparing the two on the
 same candidates is how the linearisation’s error is measured (Section 4.2). Because European semicircular cruising rules tie
 585 the permissible flight levels to the compass direction flown (the magnetic track) (European Commission, 2012, Appendix 3),
 a 1000 ft offset lands on levels reserved for the opposite direction while a 2000 ft offset preserves parity, so for each flight the
 two ± 2000 ft yo-yos are its rule-compliant windowed candidates.

Per flight, the selected candidate is the one with the most negative evaluated ΔEF ; fuel is reported, not optimised. An
 intervention is counted only when the flight’s filed profile warms ($EF > 0$) and the selected candidate lowers EF ; the credited
 590 reduction is capped at the flight’s own forcing, so that pushing a flight past zero, or deepening a cooling contrail, is never
 credited as avoided warming.

A3 The linearisation error

The campaign’s evaluated outcomes are what measures the linearisation’s error. Regressing evaluated on gradient-predicted
 ΔEF over each week’s windowed yo-yos, of both sizes and both directions, gives a pooled slope of 0.13 at 2000 ft (95 %
 595 interval 0.114 to 0.142) and 0.45 at 1000 ft (0.425 to 0.464), a factor of 3.5, over 7,946 candidates at each size (3,973 flights,
 two candidates each); the per-week slopes at 2000 ft run from 0.03 in autumn to 0.24 in winter. The interval is a block bootstrap
 over flights rather than over candidates, since the two candidates from one flight are not independent draws. On the same
 evaluated outcomes, the selected candidate is one whose predicted ΔEF was positive for 33 % of reprofiled flights (1,299 of
 3,958; 20–48 % across the four weeks).



600 **Appendix B: Discretisation and age-cap sensitivity**

Two configuration choices of Section 3 truncate the physics for cost: the 30 min integration step and the 3 h contrail age cap. This appendix measures what each of the two decides.

B1 The integration step

On one benchmark day, stepping at 1 min instead of 30 min raised the day-total EF by 43 % with both implementations still
 605 agreeing with each other; per flight the median $|EF|$ change was 31 % (90th percentile 85 %), 286 flights changed the sign
 of their net forcing, and 153 flights formed persistent contrails at 1 min that formed none at 30 min. A sensitivity of that size
 cannot be assumed to leave a top-percentile shortlist of a few hundred flights intact, so we checked: the shortlist chosen at the
 coarse step captures 96 % of the gross warming that the shortlist chosen at the fine step captures, measured against the finer
 step's own totals. The absolute forcing is therefore sensitive to the discretisation while the *ordering* it induces, which is what
 610 the workflow consumes, is largely not. This is one day and one pair of steps, and what it establishes is that the shortlist survives
 the coarser step, not that every quantity computed on the shortlisted flights would.

B2 The maximum-age cap

EF is likewise far from converged in maximum age at 3 h, and for the flights whose contrails are still alive at that cap the
 gap to the converged answer is larger than the integration-step effect of Appendix B1. We reran 400 flights of a winter day
 615 and 400 of a summer day, chosen to span the range of forcing values rather than to represent the fleet, at a series of increasing
 age caps from 3 h to 30 h, on weather stitched from three consecutive daily extracts so that no contrail is ended by the edge
 of the weather file; summed warming rises by 125 % and 61 % respectively between the two ends (the samples are not fleet-
 representative, so their rise is not the fleet's). Both plateau by 10 h, however: beyond it the totals move by less than 0.02 %,
 and the 20 h and 30 h sums are identical to the last digit on both days, no contrail in either sample surviving that long. That
 620 last observation is as much about this domain as about the physics, because a contrail advected out of the regional extract is
 terminated at the boundary whatever its age (Sect. 3), so a longer-lived contrail may simply have left. Within that extent, the
 library default of 20 h is a converged answer rather than a longer arbitrary truncation, and only contrails still alive near the cap
 are affected, those that die early being unchanged.

The cap does not only move the totals: it moves which flights the screening selects. A shortlist redrawn at 20 h shares only
 625 53 % of its flights with the one drawn at 3 h, which is what separates the cap's effect from the integration step of Appendix B1,
 where the ordering is left largely alone. Every comparison drawn in this paper does apply the same cap on both sides, so those
 comparisons are internally consistent; but, given how far shortlist membership itself shifts, that consistency is no evidence of
 invariance to the age cap. The reported gradient diagnostics would need recomputing to establish their behaviour at the longer
 cap.



630 *Code availability.* DIFFCOCIP and its verification suite are open source at <https://github.com/ramondalmau/diffcocip> (Apache-2.0), and build
on our open aircraft-performance tooling (Dalmau et al., 2026). The version described here, DIFFCOCIP v1.0, is tagged in the repository and
includes the three guards added after the campaign record was frozen (Appendix A2). The supported configuration of Section 2.4, the
porting rules, and every verified departure from the reference implementation are documented in the repository itself, in `docs/model.md`,
635 `docs/porting-rules.md` and a dated deviation ledger (`DEVIATIONS.md`), alongside the negative-control record of the verification
suite. The campaign scripts that drive the screening, evaluation and analysis run against the access-controlled traffic dataset and are not
part of the released repository; a deposit in which those inputs are replaced by their public ADS-B equivalents (Automatic Dependent
Surveillance–Broadcast, the position reports aircraft transmit) is in preparation and will accompany the archived version where redistribution
permits.

Data availability. ERA5 reanalysis (Hersbach et al., 2020) is publicly available from the Copernicus Climate Data Store; this paper contains
640 modified Copernicus Climate Change Service information 2024, and neither the European Commission nor ECMWF is responsible for any
use of that information. The EUROCONTROL flight-plan data are proprietary and cannot be redistributed, and the per-day aggregate result
files derived from them are governed by the same terms and are likewise not released. No flight identifiers (callsigns, airline, origin or
destination) are published: the results are fleet-level aggregates, except for case-study flights on two campaign days, which are reported by
aircraft type and date, with their tracks and altitude profiles shown in rasterised figure panels only.

645 The workflow does not depend on that particular source. The weather side is public and the traffic side enters only as trajectories with
timing, so an equivalent study can be assembled from openly available ADS-B trajectories; absolute totals would differ, ADS-B recording
what was flown where a flight plan records what was intended, but every comparison drawn here is internal to a single consistent source.

Author contributions. R.D. designed the study, implemented and verified the differentiable reimplementations, ran the campaigns and the
analyses, and drafted the manuscript. P.V., G.J. and F.B. contributed to the design of the study and the interpretation of the results, and
650 reviewed and edited the manuscript. All authors read and approved the submitted version.

Competing interests. The authors declare no competing interests.

Acknowledgements. This work rests on PYCONTRAILS (Shapiro et al., 2026), the open-source implementation of CoCiP maintained by the
Contrail Cirrus open-source organisation. It is both the model we reimplemented and the reference every subroutine, stage and gradient here
is verified against, and a reimplementations of this kind is only checkable because it and the model description behind it (Schumann, 2012)
655 are open. We thank its developers.

The experiments, analysis and initial draft of this manuscript were produced by the authors. Large language model tooling (Claude,
Anthropic; ChatGPT, OpenAI) was used, under author direction, for consistency auditing of the manuscript against its campaign artefacts,
for literature retrieval with independent DOI verification, and for punctuation, grammar, spelling and sentence-structure checks to improve



660 readability. The two schematic figures (Figs. 1 and 3) were generated with Google Gemini from author-written prompts and revised under author direction. Every data figure was drawn from the campaign artefacts by the scripts in the repository, and all numerical results were produced by the models and scripts cited above, not by a language model. The authors reviewed and take full responsibility for every claim, interpretation and sentence in the manuscript.

The views expressed in this paper are those of the authors alone and do not necessarily reflect the official policy or position of EURO-CONTROL.

665 *Financial support.* This research has received funding from the SESAR 3 Joint Undertaking under the European Union's Horizon Europe research and innovation programme, under grant agreement No. 101114613 (CICONIA) and No. 999483733 (E-CONTRAIL 2). The Joint Undertaking receives support from the European Union's Horizon Europe research and innovation programme and the SESAR 3 Joint Undertaking members other than the Union.



References

- Alexandrov, N. M., Dennis, J. E., Lewis, R. M., and Torczon, V.: A Trust-Region Framework for Managing the Use of Approximation Models in Optimization, *Structural Optimization*, 15, 16–23, <https://doi.org/10.1007/BF01197433>, 1998.
- Appleman, H.: The Formation of Exhaust Condensation Trails by Jet Aircraft, *Bulletin of the American Meteorological Society*, 34, 14–20, <https://doi.org/10.1175/1520-0477-34.1.14>, 1953.
- Baker, N. L., and Daley, R.: Observation and Background Adjoint Sensitivity in the Adaptive Observation-Targeting Problem, *Quarterly Journal of the Royal Meteorological Society*, 126, 1431–1454, <https://doi.org/10.1002/qj.49712656511>, 2000.
- Baydin, A. G., Pearlmutter, B. A., Radul, A. A., and Siskind, J. M.: Automatic Differentiation in Machine Learning: A Survey, *Journal of Machine Learning Research*, 18, 1–43, <http://jmlr.org/papers/v18/17-468.html>, 2018.
- Bickel, M., Ponater, M., Bock, L., Burkhardt, U., and Reineke, S.: Estimating the Effective Radiative Forcing of Contrail Cirrus, *Journal of Climate*, 33, 1991–2005, <https://doi.org/10.1175/JCLI-D-19-0467.1>, 2020.
- Bradbury, J., Frostig, R., Hawkins, P., Johnson, M. J., Leary, C., Maclaurin, D., Necula, G., Paszke, A., VanderPlas, J., Wanderman-Milne, S., and Zhang, Q.: JAX: Composable Transformations of Python+NumPy Programs, <http://github.com/jax-ml/jax>, software; version 0.10.2 used in this work, 2018.
- Burkhardt, U., and Kärcher, B.: Global Radiative Forcing from Contrail Cirrus, *Nature Climate Change*, 1, 54–58, <https://doi.org/10.1038/nclimate1068>, 2011.
- Cacuci, D. G.: Sensitivity Theory for Nonlinear Systems. I. Nonlinear Functional Analysis Approach, *Journal of Mathematical Physics*, 22, 2794–2802, <https://doi.org/10.1063/1.525186>, 1981.
- Cardinali, C.: Monitoring the Observation Impact on the Short-Range Forecast, *Quarterly Journal of the Royal Meteorological Society*, 135, 239–250, <https://doi.org/10.1002/qj.366>, 2009.
- Courtier, P., Thépaut, J.-N., and Hollingsworth, A.: A Strategy for Operational Implementation of 4D-Var, Using an Incremental Approach, *Quarterly Journal of the Royal Meteorological Society*, 120, 1367–1387, <https://doi.org/10.1002/qj.49712051912>, 1994.
- Dalmau, R., Jarry, G., and Very, P.: openleaf: Open-Source Aircraft Performance Modeling and Trajectory Optimization, <https://github.com/eurocontrol-asu/openleaf>, EUROCONTROL Aviation Sustainability Unit, package `openleaf-performance`; git commit 7244606 used in this work, 2026.
- Dean, T. R., Abbott, T. H., Engberg, Z., Masson, N., Teoh, R., Itcovitz, J. P., Stettler, M. E. J., and Shapiro, M. L.: Impact of Forecast Stability on Navigational Contrail Avoidance, *Environmental Research: Infrastructure and Sustainability*, 5, 045 008, <https://doi.org/10.1088/2634-4505/ae1da5>, 2025.
- Dietmüller, S., Matthes, S., Dahlmann, K., Yamashita, H., Simorgh, A., Soler, M., Linke, F., Lührs, B., Meuser, M. M., Weder, C., Grewe, V., Yin, F., and Castino, F.: A Python Library for Computing Individual and Merged Non-CO₂ Algorithmic Climate Change Functions: CLIMaCCF V1.0, *Geoscientific Model Development*, 16, 4405–4425, <https://doi.org/10.5194/gmd-16-4405-2023>, 2023.
- Ehrendorfer, M., Errico, R. M., and Raeder, K. D.: Singular-Vector Perturbation Growth in a Primitive Equation Model with Moist Physics, *Journal of the Atmospheric Sciences*, 56, 1627–1648, [https://doi.org/10.1175/1520-0469\(1999\)056<1627:SVPGIA>2.0.CO;2](https://doi.org/10.1175/1520-0469(1999)056<1627:SVPGIA>2.0.CO;2), 1999.
- Engberg, Z., Teoh, R., Abbott, T., Dean, T., Stettler, M. E. J., and Shapiro, M. L.: Forecasting Contrail Climate Forcing for Flight Planning and Air Traffic Management Applications: The CocipGrid Model in `pycontrails` 0.51.0, *Geoscientific Model Development*, 18, 253–286, <https://doi.org/10.5194/gmd-18-253-2025>, 2025.



- 705 Errico, R. M.: What Is an Adjoint Model?, *Bulletin of the American Meteorological Society*, 78, 2577–2591, [https://doi.org/10.1175/1520-0477\(1997\)078<2577:WIAAM>2.0.CO;2](https://doi.org/10.1175/1520-0477(1997)078<2577:WIAAM>2.0.CO;2), 1997.
- European Commission: Commission Implementing Regulation (EU) No 923/2012 of 26 September 2012 Laying Down the Common Rules of the Air and Operational Provisions Regarding Services and Procedures in Air Navigation (SERA), Publications Office of the European Union, <https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=celex:32012R0923>, appendix 3, Tables of cruising levels; see SERA.5025, 2012.
- 710 Forget, G., Campin, J.-M., Heimbach, P., Hill, C. N., Ponte, R. M., and Wunsch, C.: ECCO Version 4: An Integrated Framework for Non-Linear Inverse Modeling and Global Ocean State Estimation, *Geoscientific Model Development*, 8, 3071–3104, <https://doi.org/10.5194/gmd-8-3071-2015>, 2015.
- Gelbrecht, M., White, A., Bathiany, S., and Boers, N.: Differentiable Programming for Earth System Modeling, *Geoscientific Model Development*, 16, 3123–3135, <https://doi.org/10.5194/gmd-16-3123-2023>, 2023.
- 715 Gierens, K., Matthes, S., and Rohs, S.: How Well Can Persistent Contrails Be Predicted?, *Aerospace*, 7, 169, <https://doi.org/10.3390/aerospace7120169>, 2020.
- Griewank, A., and Walther, A.: *Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation*, Society for Industrial and Applied Mathematics, Philadelphia, PA, 2nd edn., <https://doi.org/10.1137/1.9780898717761>, 2008.
- 720 Hascoët, L., and Pascual, V.: The Tapenade Automatic Differentiation Tool: Principles, Model, and Specification, *ACM Transactions on Mathematical Software*, 39, 20:1–20:43, <https://doi.org/10.1145/2450153.2450158>, 2013.
- Heimbach, P., Hill, C., and Giering, R.: An Efficient Exact Adjoint of the Parallel MIT General Circulation Model, Generated via Automatic Differentiation, *Future Generation Computer Systems*, 21, 1356–1371, <https://doi.org/10.1016/j.future.2004.11.010>, 2005.
- Hersbach, H., Bell, B., Berrisford, P., Hirahara, S., Horányi, A., Muñoz-Sabater, J., Nicolas, J., Peubey, C., Radu, R., Schepers, D., Simons, A., Soci, C., Abdalla, S., Abellan, X., Balsamo, G., Bechtold, P., Biavati, G., Bidlot, J., Bonavita, M., De Chiara, G., Dahlgren, P., Dee, D., Diamantakis, M., Dragani, R., Flemming, J., Forbes, R., Fuentes, M., Geer, A., Haimberger, L., Healy, S., Hogan, R. J., Hólm, E., Janisková, M., Keeley, S., Laloyaux, P., Lopez, P., Lupu, C., Radnoti, G., de Rosnay, P., Rozum, I., Vamborg, F., Villaume, S., and Thépaut, J.-N.: The ERA5 Global Reanalysis, *Quarterly Journal of the Royal Meteorological Society*, 146, 1999–2049, <https://doi.org/10.1002/qj.3803>, 2020.
- 725 Kärcher, B.: Formation and Radiative Forcing of Contrail Cirrus, *Nature Communications*, 9, 1824, <https://doi.org/10.1038/s41467-018-04068-0>, 2018.
- Kochkov, D., Yuval, J., Langmore, I., Norgaard, P., Smith, J., Mooers, G., Klöwer, M., Lottes, J., Rasp, S., Düben, P., Hatfield, S., Battaglia, P., Sanchez-Gonzalez, A., Willson, M., Brenner, M. P., and Hoyer, S.: Neural General Circulation Models for Weather and Climate, *Nature*, 632, 1060–1066, <https://doi.org/10.1038/s41586-024-07744-y>, 2024.
- 735 Langland, R. H., and Baker, N. L.: Estimation of Observation Impact Using the NRL Atmospheric Variational Data Assimilation Adjoint System, *Tellus A*, 56, 189–201, <https://doi.org/10.3402/tellusa.v56i3.14413>, 2004.
- Le Dimet, F.-X., and Talagrand, O.: Variational Algorithms for Analysis and Assimilation of Meteorological Observations: Theoretical Aspects, *Tellus A: Dynamic Meteorology and Oceanography*, 38, 97–110, <https://doi.org/10.3402/tellusa.v38i2.11706>, 1986.
- Lee, D. S., Fahey, D. W., Skowron, A., Allen, M. R., Burkhardt, U., Chen, Q., Doherty, S. J., Freeman, S., Forster, P. M., Fuglestedt, J., Gettelman, A., De León, R. R., Lim, L. L., Lund, M. T., Millar, R. J., Owen, B., Penner, J. E., Pitari, G., Prather, M. J., Sausen, R., and Wilcox, L. J.: The Contribution of Global Aviation to Anthropogenic Climate Forcing for 2000 to 2018, *Atmospheric Environment*, 244, 117 834, <https://doi.org/10.1016/j.atmosenv.2020.117834>, 2021.
- 740



- Martin Frias, A., Shapiro, M. L., Engberg, Z., Zopp, R., Soler, M., and Stettler, M. E. J.: Feasibility of Contrail Avoidance in a Commercial Flight Planning System: An Operational Analysis, *Environmental Research: Infrastructure and Sustainability*, 4, 015013, <https://doi.org/10.1088/2634-4505/ad310c>, 2024.
- Navon, I. M.: Data Assimilation for Numerical Weather Prediction: A Review, in: *Data Assimilation for Atmospheric, Oceanic and Hydrologic Applications*, pp. 21–65, Springer, https://doi.org/10.1007/978-3-540-71056-1_2, 2009.
- Peherstorfer, B., Willcox, K., and Gunzburger, M.: Survey of Multifidelity Methods in Uncertainty Propagation, Inference, and Optimization, *SIAM Review*, 60, 550–591, <https://doi.org/10.1137/16M1082469>, 2018.
- Pianosi, F., Beven, K., Freer, J., Hall, J. W., Rougier, J., Stephenson, D. B., and Wagener, T.: Sensitivity Analysis of Environmental Models: A Systematic Review with Practical Workflow, *Environmental Modelling & Software*, 79, 214–232, <https://doi.org/10.1016/j.envsoft.2016.02.008>, 2016.
- Poll, D. I. A., and Schumann, U.: An Estimation Method for the Fuel Burn and Other Performance Characteristics of Civil Transport Aircraft in the Cruise. Part 1 Fundamental Quantities and Governing Relations for a General Atmosphere, *The Aeronautical Journal*, 125, 257–295, <https://doi.org/10.1017/aer.2020.62>, 2021a.
- Poll, D. I. A., and Schumann, U.: An Estimation Method for the Fuel Burn and Other Performance Characteristics of Civil Transport Aircraft During Cruise: Part 2, Determining the Aircraft’s Characteristic Parameters, *The Aeronautical Journal*, 125, 296–340, <https://doi.org/10.1017/aer.2020.124>, 2021b.
- Rabier, F., Järvinen, H., Klinker, E., Mahfouf, J.-F., and Simmons, A.: The ECMWF Operational Implementation of Four-Dimensional Variational Assimilation. I: Experimental Results with Simplified Physics, *Quarterly Journal of the Royal Meteorological Society*, 126, 1143–1170, <https://doi.org/10.1002/qj.49712656415>, 2000.
- Saltelli, A., Ratto, M., Andres, T., Campolongo, F., Cariboni, J., Gatelli, D., Saisana, M., and Tarantola, S.: *Global Sensitivity Analysis. The Primer*, John Wiley & Sons, Chichester, <https://doi.org/10.1002/9780470725184>, 2008.
- Sankar, T., Dean, T., Abbott, T., Blickstein, J., Martín Frías, A., Galyen, M., Grenham, R., Hodgson, P., McCloskey, K., Pechman, A., Robarge, T., Sanekommu, D., Sarna, A., Sonabend-W, A., Stettler, M., Zopp, R., and Geraedts, S.: Efficacy of Scalable Airline-Led Contrail Avoidance, <https://doi.org/10.48550/arXiv.2603.06909>, 2026.
- Sapienza, F., Bolibar, J., Schäfer, F., Groenke, B., Pal, A., Boussange, V., Heimbach, P., Hooker, G., Pérez, F., Persson, P.-O., and Rackauckas, C.: Differentiable Programming for Differential Equations: A Review, <https://doi.org/10.48550/arXiv.2406.09699>, 2024.
- Sausen, R., Hofer, S., Gierens, K., Bugliaro, L., Ehrmanntraut, R., Sitova, I., Walczak, K., Burridge-Diesing, A., Bowman, M., and Miller, N.: Can We Successfully Avoid Persistent Contrails by Small Altitude Adjustments of Flights in the Real World?, *Meteorologische Zeitschrift*, 33, 83–98, <https://doi.org/10.1127/metz/2023/1157>, 2024.
- Schumann, U.: On Conditions for Contrail Formation from Aircraft Exhausts, *Meteorologische Zeitschrift*, 5, 4–23, <https://doi.org/10.1127/metz/5/1996/4>, 1996.
- Schumann, U.: A Contrail Cirrus Prediction Model, *Geoscientific Model Development*, 5, 543–580, <https://doi.org/10.5194/gmd-5-543-2012>, 2012.
- Schumann, U., Mayer, B., Graf, K., and Mannstein, H.: A Parametric Radiative Forcing Model for Contrail Cirrus, *Journal of Applied Meteorology and Climatology*, 51, 1391–1406, <https://doi.org/10.1175/JAMC-D-11-0242.1>, 2012.
- Schumann, U., Baumann, R., Baumgardner, D., Bedka, S. T., Duda, D. P., Freudenthaler, V., Gayet, J.-F., Heymsfield, A. J., Minnis, P., Quante, M., Raschke, E., Schlager, H., Vazquez-Navarro, M., Voigt, C., and Wang, Z.: Properties of Individual Contrails: A Compilation



- 780 of Observations and Some Comparisons, *Atmospheric Chemistry and Physics*, 17, 403–438, <https://doi.org/10.5194/acp-17-403-2017>, 2017.
- Shapiro, M., Engberg, Z., Teoh, R., Stettler, M. E. J., and Dean, T.: pycontrails: Python Library for Modeling Aviation Climate Impacts, Version 0.63.3, <https://github.com/contrailcirrus/pycontrails/releases/tag/v0.63.3>, version 0.63.3 used in this work, 2026.
- Shen, C., Appling, A. P., Gentine, P., Bandai, T., Gupta, H., Tartakovsky, A., Baity-Jesi, M., Fenicia, F., Kifer, D., Li, L., Liu, X., Ren, W.,
 785 Zheng, Y., Harman, C. J., Clark, M., Farthing, M., Feng, D., Kumar, P., Aboelyazeed, D., Rahmani, F., Song, Y., Beck, H. E., Bindas, T., Dwivedi, D., Fang, K., Höge, M., Rackauckas, C., Mohanty, B., Roy, T., Xu, C., and Lawson, K.: Differentiable Modelling to Unify Machine Learning and Physical Models for Geosciences, *Nature Reviews Earth & Environment*, 4, 552–567, <https://doi.org/10.1038/s43017-023-00450-9>, 2023.
- Simorgh, A., and Soler, M.: Climate-Optimized Flight Planning Can Effectively Reduce the Environmental Footprint of Aviation in Europe
 790 at Low Operational Costs, *Communications Earth & Environment*, 6, 66, <https://doi.org/10.1038/s43247-025-02031-8>, 2025.
- Sonabend-W, A., Elkin, C., Dean, T., Dudley, J., Ali, N., Blickstein, J., Brand, E., Broshears, B., Chen, S., Engberg, Z., Galyen, M., Geraedts, S., Goyal, N., Grenham, R., Hager, U., Hecker, D., Jany, M., McCloskey, K., Ng, J., Norris, B., Opel, F., Rothenberg, J., Sankar, T., Sanekommu, D., Sarna, A., Schütt, O., Shapiro, M., Soh, R., Van Arsdale, C., and Platt, J. C.: Feasibility Test of Per-Flight Contrail Avoidance in Commercial Aviation, *Communications Engineering*, 3, 184, <https://doi.org/10.1038/s44172-024-00329-7>, 2024.
- 795 Sridhar, B., Ng, H. K., and Chen, N. Y.: Aircraft Trajectory Optimization and Contrails Avoidance in the Presence of Winds, in: 10th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference, <https://doi.org/10.2514/6.2010-9139>, 2010.
- Sun, J., and Olive, X.: Contrail, or Not Contrail, That Is the Question: The “Feasibility” of Climate-Optimal Routing, <https://doi.org/10.48550/arXiv.2504.13907>, 2025.
- Talagrand, O., and Courtier, P.: Variational Assimilation of Meteorological Observations with the Adjoint Vorticity Equation. I: Theory,
 800 *Quarterly Journal of the Royal Meteorological Society*, 113, 1311–1328, <https://doi.org/10.1002/qj.49711347812>, 1987.
- Tarantola, A.: Inverse Problem Theory and Methods for Model Parameter Estimation, Society for Industrial and Applied Mathematics, Philadelphia, <https://doi.org/10.1137/1.9780898717921>, 2005.
- Teoh, R., Schumann, U., Majumdar, A., and Stettler, M. E. J.: Mitigating the Climate Forcing of Aircraft Contrails by Small-Scale Diversions and Technology Adoption, *Environmental Science & Technology*, 54, 2941–2950, <https://doi.org/10.1021/acs.est.9b05608>, 2020a.
- 805 Teoh, R., Schumann, U., and Stettler, M. E. J.: Beyond Contrail Avoidance: Efficacy of Flight Altitude Changes to Minimise Contrail Climate Forcing, *Aerospace*, 7, 121, <https://doi.org/10.3390/aerospace7090121>, 2020b.
- Teoh, R., Schumann, U., Gryspeerdt, E., Shapiro, M., Molloy, J., Koudis, G., Voigt, C., and Stettler, M. E. J.: Aviation Contrail Climate Effects in the North Atlantic from 2016 to 2021, *Atmospheric Chemistry and Physics*, 22, 10 919–10 935, <https://doi.org/10.5194/acp-22-10919-2022>, 2022.
- 810 Utke, J., Naumann, U., Fagan, M., Tallent, N., Strout, M., Heimbach, P., Hill, C., and Wunsch, C.: OpenAD/F: A Modular Open-Source Tool for Automatic Differentiation of Fortran Codes, *ACM Transactions on Mathematical Software*, 34, 18:1–18:36, <https://doi.org/10.1145/1377596.1377598>, 2008.
- Xu, Q.: Generalized Adjoint for Physical Processes with Parameterized Discontinuities. Part I: Basic Issues and Heuristic Examples, *Journal of the Atmospheric Sciences*, 53, 1123–1142, [https://doi.org/10.1175/1520-0469\(1996\)053<1123:GAFFPW>2.0.CO;2](https://doi.org/10.1175/1520-0469(1996)053<1123:GAFFPW>2.0.CO;2), 1996.
- 815 Yin, F., Grewe, V., Castino, F., Rao, P., Matthes, S., Dahlmann, K., Dietmüller, S., Frömming, C., Yamashita, H., Peter, P., Klingaman, E., Shine, K. P., Lührs, B., and Linke, F.: Predicting the Climate Impact of Aviation for En-Route Emissions: The Algorithmic Climate



Change Function Submodel ACCF 1.0 of EMAC 2.53, Geoscientific Model Development, 16, 3313–3334, <https://doi.org/10.5194/gmd-16-3313-2023>, 2023.

820 Zou, X.: Tangent Linear and Adjoint of “On-Off” Processes and Their Feasibility for Use in 4-Dimensional Variational Data Assimilation, Tellus A: Dynamic Meteorology and Oceanography, 49, 3–31, <https://doi.org/10.3402/tellusa.v49i1.12209>, 1997.

Zou, X., Navon, I. M., and Sela, J. G.: Variational Data Assimilation with Moist Threshold Processes Using the NMC Spectral Model, Tellus A: Dynamic Meteorology and Oceanography, 45, 370–387, <https://doi.org/10.3402/tellusa.v45i5.14900>, 1993.