# Investigation of the 48.5 kt Intensity Ceiling in Oracle V6.1 "THETA" Simulation

## 1 Overview of the Issue

All simulation runs of the Oracle V6.1 THETA model plateau at ∼48.5 knots maximum wind speed (≈25 m/s) regardless of "governor" settings. This suggests a systemic limitation in the model's physics or numerics. We performed a code review of key components—stratification and buoyancy calculations, $\theta'$ tendency updates, surface fluxes, and legacy limiters—to diagnose why intensification stalls. Below we detail potential causes and relevant code findings, then recommend improvements.

## 2 Explicit Intensity Limiters (Governors) in Code

Several hard-coded limiters (governors) exist to prevent unrealistic intensification. However, none should trigger at 48.5 kt (25 m/s), indicating they are likely not the cause of the early ceiling:

**Surface Flux Throttle:** The "Governor Protocol" in `apply_surface_fluxes` reduces fluxes at high wind speeds. Wind speeds below 35 m/s get full flux (factor 1.0), then taper to 50% at 70 m/s and 20% at 90 m/s. At 25 m/s (48.5 kt), this throttle remains at 1.0 (no flux cut). The global WISDOM brake (cuts flux beyond 80 m/s) is even higher. In short, no flux governor is active at TS intensity. (If anything, these governors would only engage at far higher wind speeds, beyond Category 5 levels.)

**Buoyancy and Updraft Clamps:** Legacy "thermo firewalls" cap extreme buoyancy and updrafts. By default, buoyancy is limited to `buoyancy_cap` = 0.5 m/s$^2$ via a tanh() clamp, and vertical velocity is limited to `max_updraft` = 50 m/s (also via tanh) if firewalls are enabled. These clamps are only relevant for very large values (e.g. updraft >50 m/s or buoyancy >0.5 m/s$^2$) and were not reached in these runs. Indeed, even with firewalls on, a TS-strength storm would have much smaller updrafts (a few m/s) and buoyancy accelerations. When running fully unconstrained (flags `no_thermo_firewalls`, etc.), these limits are disabled entirely. Thus, no artificial clamp is actively capping intensity at 48.5 kt—the plateau arises despite the governors being non-binding or turned off.

**Conclusion:** The usual "governors" (flux throttling, WISDOM brake, buoyancy/updraft caps) are not the culprit for the 48.5 kt ceiling. The cause likely lies in the model's physical parameterization or energy budget rather than an explicit hard cutoff. We next examine those aspects.

## 3 Moist-Adiabatic Stratification Implementation (`moist_factor`)

Oracle V6.1 introduced a moisture-aware stratification to reduce stabilizing cooling in saturated air. The code computes an "effective" static stability by blending between dry and moist lapse rates. Key details:

**Baseline Dry Profile:** The reference potential temperature $\theta_0$ increases with height at $\gamma_\theta = 4$ K/km (dry stability). Thus, a rising parcel experiences a cooling term $-w\, d\theta_0/dz$ (analogous to adiabatic cooling) that opposes convection.

**Moist Lapse Reduction:** In saturated regions, the model multiplies $d\theta_0/dz$ by a factor *effective_factor* $< 1$ to mimic latent heat release. This factor is computed as:

$$\textit{effective\_factor} = 1 - \textit{saturation\_blend} \times (1 - \textit{moist\_factor})$$

where *moist_factor* is the ratio of moist to dry lapse rate (temperature-dependent) and *saturation_blend* goes from 0 (dry air) to 1 (fully saturated). For example, at warm surface temperatures the code sets *moist_factor* $\approx 0.5$, meaning a fully saturated updraft feels about 50% of the normal stratification. In cold upper levels this ratio is higher ($\sim$0.6), but the code floors it at 0.3 minimum.

**Effect on Stability:** In completely saturated air, the smallest *effective_factor* is $\sim$0.3 (if *moist_factor* hits the 0.3 floor). This means the weakest stratification the model will allow is 30% of the dry value. In other words, even an eyewall at 100% RH still experiences some stabilizing cooldown (just reduced). The model never truly approaches neutral stability ($N^2 \to 0$)—contrary to the intention that "effective $N^2 \to 0$ in eyewall" noted in the code docs. Real hurricanes can achieve near moist-neutral interiors, so this conservative reduction may be capping convection intensity. The persistent $\sim$48.5 kt ceiling suggests the storm may be finding an equilibrium where the residual stratification (30–50% of dry) balances its buoyant forcing. Essentially, the eyewall might not be "allowed" to become buoyantly unstable enough to accelerate beyond tropical-storm strength.

**Recommendation:** Consider loosening the moist stratification constraint. Lower the minimum *moist_factor* or adjust the RH blend so that a saturated, warm-core column sees almost zero effective stratification (approaching true moist-adiabatic lapse). This could allow stronger updrafts and deeper convection, potentially breaking the intensity ceiling. It may be useful to diagnose the average *effective_factor* in the eyewall during runs (the code already tracks `mean_effective_factor` and `mean_saturation_blend` for logging)—if it's, say, 0.5, then the model is still damping half of the instability even in core updrafts. Reducing that further (closer to 0) might enable continued intensification. Care is needed to avoid instabilities, but a small test (e.g. allowing *moist_factor* down to 0.2 or even 0.1 for very warm saturated air) could reveal if the intensity limit moves upward.

## 4 $\theta'$ Tendency Calculation and Upper-Level Amplification

Another suspect is a miscalibration in how potential-temperature perturbations ($\theta'$) are updated, especially with respect to temperature at high altitudes. In V6, all heating/cooling is ultimately applied to $\theta'$ (with diagnostic $T$ derived afterward). We found that the model consistently multiplies temperature changes by a factor $\theta/T$ when converting to $\theta'$ changes. Examples:

**Latent Heat Release:** Upon condensation, an excess moisture $dq$ yields a temperature rise $dT = (L_v/C_p)\, dq$. This is converted to a $\theta'$ increase via $d\theta = (\theta/T)\, dT$. Here $\theta$ is the total potential temperature at that grid cell and $T$ is the absolute temperature.

**Surface Sensible Heat Flux:** Similarly, when adding heat at the surface, the model computes the temperature change at the surface layer and then scales it to a $\theta'$ change by $(\theta/T)$. This is referred to as "Five's correction" in comments. Essentially, if the surface air is cooler than its $\theta$, a given heat input raises $\theta'$ slightly less, maintaining consistency with thermodynamics.

This $\theta/T$ factor is physically based ($\theta$ is conserved; $T$ varies with pressure), but it can amplify thermal perturbations aloft: At high altitudes, $T$ is low (e.g. 200–230 K), while $\theta$ of a lifted parcel can be quite high (350–380 K). The ratio $\theta/T$ might be on the order of 1.5–2.0 in the upper troposphere, meaning a 1 K local warming translates to a ∼1.5–2 K increase in $\theta$. The code indeed allows such amplification: no cap is imposed on $\theta/T$, so upper-level heating yields proportionally larger $\theta'$.

Why could this limit intensity? If the model's latent heating predominantly warms the upper layers (e.g. 10–15 km) with a large $\theta'$ effect, the warm-core anomaly might be distributed too high. A very elevated warm core raises upper pressure levels but has less impact on surface pressure deficit (and thus gradient wind). In essence, the storm might be "wasting" energy warming the stratosphere or upper outflow layer rather than the mid-levels where it most aids intensification. This could produce a self-limiting behavior: further condensation just amplifies $\theta'$ aloft (which the stable stratification resists), instead of lowering surface pressure substantially. It's notable that 48.5 kt is a modest intensity—perhaps the model's warm core is capped in the mid-levels and any additional latent heat goes into an area that yields diminishing returns for wind speed. This aligns with point (1) above: stratification still active → buoyancy rises but mostly felt aloft → limited surface wind increase.

**Recommendation:** Verify the vertical profile of the warm-core. We suggest adding diagnostics for the height of maximum $\theta'$ or temperature perturbation over time. If it shows that the warm anomaly is skewing too high, consider adjustments: for instance, limit the $\theta'$ amplification at very low pressures (perhaps by capping $\theta/T$ to a reasonable value) or ensure the bulk of latent heating occurs where it is most effective (maybe via an improved microphysics triggering more warming at 5–8 km heights). Another approach is introducing a modest radiative cooling or enhanced mixing at the top to bleed off excessive upper-level heat, preventing it from trapping energy aloft. This would force more of the storm's heat engine to operate in the middle troposphere. In summary, make sure the $\theta'$ tendency calculation isn't inadvertently biasing energy to the top of the domain in a way that stifles intensification.

(On the plus side, the $\theta'$-based formulation avoids unphysical wrap-around of high $\theta$ air from the top to surface in the periodic vertical grid. The issues identified are about magnitude, not the conceptual approach.)

## 5 Surface Fluxes vs. Moisture Consumption Balance

Tropical cyclones intensify by taking heat and moisture from the ocean, so an imbalance in surface flux input versus moisture loss can create a hard ceiling. We inspected `boundary_conditions.py` and the flux application in the simulation:

**Flux Magnitudes:** The bulk formulas for evaporation and sensible heat are standard. Latent moisture flux $F_q$ is given by:

$$F_q = C_d\,\rho_{\rm air}\,V_{\rm 10m}\,(q_{\rm sat}(T_o) - q_{\rm air})$$

and sensible heat flux $F_h$ by:

$$F_h = C_h\,\rho_{\rm air}\,c_p\,V_{\rm 10m}\,(T_{\rm ocean} - T_{\rm air})$$

(with $C_d$ and $C_h$ on the order of $10^{-3}$). At 25 m/s winds, these coefficients are $\sim 2 \times 10^{-3}$, so fluxes are quite high—likely several hundred W/m$^2$ latent and a bit lower sensible, which should favor intensification. Importantly, no "flux governor" cuts these fluxes at 25 m/s (as noted earlier). The code only blends fluxes to zero over land (`land_fraction`) and applies the high-wind brakes at much higher speeds. So, on the face of it, the ocean input should be ample for growth.

**Moisture Replenishment vs. Saturation:** A potential problem arises if the near-surface air becomes too moist. The evaporation term ($q_{\rm sat} - q_{\rm air}$) will shrink as $q_{\rm air}$ approaches saturation.

In the model, the core was pre-moistened to 95% RH initially, and as the storm spins up, the surface layer in the eyewall likely stays near saturation. Thus evaporation in the eyewall could effectively stall, since you can't evaporate much into already humid air. Real hurricanes circumvent this by importing drier inflow from outside the eyewall, but in our simulation the domain might be saturating over time. If the entire surface layer (at least within a large radius of the storm) becomes close to 100% RH, the net moisture flux into the storm engine could plateau or even drop—capping further intensification. In code, we see flux is applied to the whole grid's lowest layer. If large areas reach $q_{\text{air}} \approx q_{\text{sat}}$, the mean moisture flux will decline. This is worth checking via diagnostics (e.g. monitor the mean or max surface RH during the run).

**Condensation and Precipitation Sink:** Every time condensation occurs, the excess moisture is removed (set to saturation) and latent heat added. The condensed water is not tracked as liquid; effectively, it rains out of the system instantly (no re-evaporation of rain). This means there is a permanent sink of water substance from the air. Over time, if surface flux does not keep up, the system's total water may deplete or stop increasing, limiting convection. The model does count `total_condensed` mass, but it does not explicitly recycle it anywhere. In nature, some rain could re-evaporate in outer regions, moistening them, but here any supersaturation just converts to heat and the water is gone. To sustain intensification, the surface must continuously supply new moisture. If that supply stagnates (as per the point above, due to high RH or limited area of flux), intensification will halt when moisture input = moisture output. We suspect the storm hit that equilibrium around 48.5 kt.

**Recommendation:** Implement diagnostic tracking of the moisture budget to confirm the above. For example, record time series of: (a) domain-integrated water vapor, (b) cumulative surface moisture added, and (c) cumulative water condensed/removed. This will show if the model is moisture-limited. If it is, potential fixes include: increasing the effective surface moisture flux (e.g. a slight boost factor or higher $C_d$), or introducing some representation of rain fallback/secondary evaporation outside the core to keep the outer environment moist (though that's complex to add). Even without new physics, simply logging the mean surface humidity and flux over time is helpful. If we see the moisture flux dropping as the storm intensifies (due to RH $\to$ 1), then the model is essentially starving itself of fuel. In that case, allowing some dry air entrainment (perhaps by not pre-moistening 95% to the very surface or by mixing in slightly drier air from above in the inflow layer) might paradoxically allow more flux and further intensification. It's a delicate balance; the model may currently err on the side of too quickly saturating the inflow layer.

Additionally, confirm that the flux calculations are correctly scaled. The code accounts for the time step and grid cell thickness when adding moisture/heat. As long as the `domain_scaler.dimensionless_to_ph` conversion is correct for $dz$ (20 km domain height over `nz` levels), the flux magnitude should be physically consistent. Any error there (e.g. using wrong units for $dz$) could under-supply moisture. However, given the values used ($\rho = 1.225$, etc.), this seems properly handled. The likely issue is not a bug in flux code, but the equilibrium between flux and condensation being reached too soon.

# 6 Other Possible Hidden Constraints or Loss Mechanisms

We also considered whether any untracked energy sinks or numerical artifacts from the legacy code might be limiting intensity:

**Numerical Diffusion (Turbulence):** The model uses a Smagorinsky subgrid turbulence scheme with an aggressive "resolution boost" factor (1500×) to achieve effective eddy viscosity in the 0.1–0.5 range. This is necessary for stability on a coarse grid, but it also means substantial turbulent diffusion of momentum and heat. Excessive diffusion can sap energy from the vortex—

smoothing out the winds and warm core. If the intensity stalls, part of the reason could be that any further strengthening generates turbulence that dissipates the added energy as fast as it's gained. We should verify the eddy viscosity or enstrophy levels in the eyewall as it intensifies. It might be worth tuning the Smagorinsky $C_s$ or the boost factor downward to see if the ceiling shifts (acknowledging the risk of more small-scale noise).

**Surface Drag Dissipation:** With winds of ∼25 m/s, the surface drag (stress) is not trivial. The drag formula ($\tau = C_d \rho |V| V$) will remove kinetic energy from the flow and convert it to heat. The model computes drag stress in `calculate_surface_drag`, and presumably applies it in the momentum solver each step (though that code wasn't included in the snippet, it likely subtracts a term from $u, v$ at $k = 0$ proportional to $\tau \cdot dt/(\rho \Delta z)$). If the drag is strong, it might reach a balance with the energy input (this is essentially Emanuel's potential intensity theory: equilibrium when input = dissipation). A quick estimate: at 25 m/s, $C_d \approx 2 \times 10^{-3}$, so frictional dissipation $\sim 2 \times 10^{-3} \times 1.225 \times (25)^3 \approx 3800$ W/m$^2$ per unit area in the eyewall—comparable to or exceeding typical flux input (∼1000 W/m$^2$ latent + sensible). This rough comparison suggests the model's dissipation might indeed equal input around 25–30 m/s, explaining the stalling. In reality, $C_k/C_d$ (ratio of heat to drag coefficient) and ocean thermodynamics determine the max intensity. If our $C_h$ (heat exchange coeff) or flux is a bit low (e.g. $C_h = 1.8 \times 10^{-3}$ base) and $C_d$ is $\sim 2 \times 10^{-3}$, the ratio might be <1, which is known to limit hurricane intensity. The model's default values produce $C_k \approx C_h \approx 0.0018$ and $C_d$ up to 0.002—a ratio ∼0.9 at high wind, slightly less favorable than often assumed in theory (∼1.0). Inadequate $C_k/C_d$ could be throttling the achievable intensity.

**Boundary and Domain Effects:** The model uses a triply-periodic domain with a limited top (20 km). A sponge layer is applied near the lateral boundaries (and possibly top) to prevent reflections. If the storm's circulation or outflow reaches those sponge regions, energy can be artificially dissipated. For example, strong outflow at 15–18 km might hit a sponge at the top boundary, damping the upper-level high-$\theta$ air and effectively cooling the storm's outflow. This would act as a heat sink on the system. The code sets `self.sponge_strength` from config (default 0.003) and builds a `sponge_mask` near edges. Although details of its application weren't in the snippet, it likely damps momentum and $\theta'$ near the domain boundaries. If the storm grows large or its circulation touches those edges, the sponge could bleed kinetic energy or heat, impeding further growth. Checking the storm's size relative to domain (2000 km wide) is worthwhile—if it's expanding too much, it might encounter the sponge. However, at TS intensity the vortex is probably not huge, so this is a minor factor. It's more relevant if trying to reach major hurricane size in a limited box.

**Model Formulation Limits:** Finally, note that Oracle V6.1 remains a Boussinesq-based model with added $\theta'$ field. Large pressure drops (50–100 mb) and density changes in intense hurricanes violate Boussinesq assumptions to some extent. The model may inherently struggle to go beyond a weak hurricane because of this. The designers partially accounted for compressibility via the $\theta'$ formulation and reference state, but certain nonlinear effects (like pressure feedback on surface flux or atmospheric compressibility at extreme winds) are not fully captured. If all else is correct, this could be an underlying ceiling. Addressing that would require moving to a compressible or anelastic formulation, which is non-trivial. Still, since 48.5 kt is well below even Cat1, it's likely not the Boussinesq limit per se but rather the other factors listed above that are dominant.

# 7 Conclusions and Recommendations

The persistent ∼48.5 kt intensity ceiling appears to be a physical/numerical equilibrium rather than a single hard cap. The review points to a combination of factors creating this equilibrium:

1. **Residual Stratification Constraint:** The moist-aware stratification scheme, while improved,

might be overly stabilizing. The eyewall never becomes truly neutral, limiting further buoyant acceleration. Relaxing the moist lapse assumption (lower *moist_factor* minimum or expanding RH threshold to treat $> \sim 90\%$ RH as effectively moist-adiabatic) could allow stronger convection. This should be tested carefully, as it affects model stability—but it aligns with physical expectations for a mature hurricane eye. Monitor the impact on intensity if adjusted.

2. **Energy Distribution with Height:** The $\theta'$-based thermodynamics may be funneling too much heat into upper levels due to the $\theta/T$ scaling. This can be addressed by examining the vertical structure of heating. If confirmed, one might introduce a slight dampening of $\theta'$ increases above a certain altitude or incorporate processes to remove heat aloft (radiative cooling or stronger overshoot-induced mixing). Another straightforward diagnostic: compare the model's warm-core magnitude and height against known hurricanes. If the core is weaker and higher than expected for a given intensity, that's a sign of the issue.

3. **Moisture and Flux Budget:** It's critical to ensure the storm isn't starved of flux. We recommend instrumenting the code to log cumulative surface moisture flux vs. cumulative condensation each hour of simulation. If those lines converge early, that's the intensity limit right there (input = output). To push it further, one could increase input (e.g. artificially boost flux by 10–20% via `q_flux_boost_factor`) or decrease output (e.g. allow some condensed moisture to remain as cloud water that might later re-evaporate). In the short term, a simpler tweak is raising `C_h_base` or the humidity difference—e.g. raising ocean temperature slightly or lowering initial base humidity outside the core to maintain a larger $q_{\text{sat}} - q$ gradient. This would feed the storm more. Also, double-check that the `no_flux_governor` flag truly disables the throttle if desired—currently the code does not explicitly skip the throttle when `no_flux_governor=True` (that flag may need to bypass the interpolation that reduces flux at high winds). Although the throttle wasn't active at 25 m/s, if one is testing intensification beyond that, ensuring `no_wisdom` and `no_flux_governor` indeed remove all brakes is important for diagnosis.

4. **Diagnostics and Testing:** Enable additional logging: surface RH field (min, max, mean), effective stratification factor in core, $\theta'$ max height, etc., as discussed. With these, run a fully-unconstrained simulation and see if/when fluxes or buoyancy taper off. If the model still plateaus, experiment with one change at a time: e.g. Case A: halve the stratification lapse (try $\gamma_\theta = 2$ K/km or allow *moist_factor* 0.2) and see if intensity improves; Case B: boost flux (say `q_flux_boost_factor=1.2`) and see if that breaks the ceiling; Case C: reduce artificial diffusion (lower Smagorinsky $C_s$ from 0.17 to 0.12 or similar) and see if that permits stronger winds (watch for noise). Each test will illuminate which factor is key.

5. **Longer-term:** If the above fails, consider the possibility that the model's design inherently limits intensity (Boussinesq approximation, lack of ocean cooling feedback, etc.). The environment SST is fixed (no cooling of ocean despite intense flux). Real storms cool their surface, which can limit intensity, but here that feedback is off—meaning if anything the model should overshoot intensity, not undershoot. So ocean coupling is not the issue here (the code even has OHC fields prepared, but no SST cooling implemented in V6.1). This reinforces that the cap is from atmospheric processes. Another factor not yet mentioned is core size: if the vortex remains too broad or too small, it might not achieve higher winds. The initial condition uses a Rankine vortex with $v_{\max}$ set to 25 m/s (which is exactly the `U_CHAR`, i.e. 1.0 in non-dim units). That means we started the simulation with a 25 m/s vortex! Indeed, `v_max` = 25.0 / `U_CHAR` = 1.0 dimensionless and `U_CHAR` = 25 m/s, so initial max winds were already $\sim$25 m/s. The model basically spun up a tropical storm initially and (in these cases) it never grew beyond that. Per-

haps the initial condition or spin-up procedure needs adjustment to allow strengthening—e.g. a smaller initial radius or a slightly weaker initial vortex that can grow organically. With the current setup, maybe the storm immediately mixes out and settles near the imposed 25 m/s. (The log confirms initial max wind ∼25 m/s.) We might try initializing a weaker vortex (say 15 m/s) and see if it can surpass 25 m/s after some time, or if it still plateaus there. This could differentiate a code-imposed ceiling from an artifact of the initial vortex design.

In summary, the evidence points to an equilibrium between positive feedbacks (flux, latent heating) and negative feedbacks (stratification, moisture depletion, friction) reached at a relatively low intensity. To overcome it, we should weaken the negative feedbacks: allow the eyewall to become more unstable (reduce effective stratification), ensure continued moisture influx (prevent premature saturation or increase $C_k/C_d$), and verify that energy isn't being lost aloft or numerically dissipated excessively. Implementing the above recommendations and additional diagnostics will help identify the dominant factor and guide adjustments so that the model can realistically simulate stronger hurricanes when physically warranted.

**Sources:** The analysis is supported by the Oracle V6.1 code: moist-aware stratification logic, $\theta'$ tendency and buoyancy calculations, surface flux and governor protocols, and condensation routines, as cited above.