Geoscience Data Journal

WILEY

Geoscience Data Journal | RMetS

DATA ARTICLE OPEN ACCESS

`Falwa`: Python Package to Implement Finite-Amplitude Local Wave Activity Diagnostics on Climate Data

Clare S. Y. Huang[1,2] | Christopher Polster[3] | Noboru Nakamura[2]

[1]Epsilon Data Management LLC, Irving, Texas, USA | [2]The University of Chicago, Chicago, Illinois, USA | [3]Johannes Gutenberg-University Mainz, Mainz, Germany

Correspondence: Noboru Nakamura (nnn@uchicago.edu)

Received: 6 October 2024 | **Revised:** 16 March 2025 | **Accepted:** 28 March 2025

Funding: C.P. was supported by German Science Foundation (DFG). N.N. is funded by NSF Award 2154523 and 2448990.

Keywords: finite-amplitude wave activity | jet stream | local wave activity | Rossby waves | wave-mean flow interactions

ABSTRACT

Weather at the mid-latitudes is governed by cyclones and anticyclones mostly migrating eastward. These weather systems cause the jet streams to undulate; the meandering patterns are known as the Rossby waves. Occasionally, Rossby waves bring forth localised extreme weather phenomena. An example of a finite-amplitude wave phenomenon is atmospheric blocking, which is often associated with heat waves and droughts. Recent development of a finite-amplitude local wave activity (FALWA) theory by Nakamura and collaborators enables comprehensive analysis of the dynamics of finite-amplitude Rossby waves observed in climate data, which helps to understand the drivers of their life cycles. Despite the simplicity of interpretation it brings about, to apply the FALWA diagnostic to climate data requires more involved calculations than the traditional Eulerian framework. This article introduces the open-source Python package `falwa`, which encapsulates the FALWA diagnostics implemented on gridded climate data presented in the authors' previous publications. It reviews the essence of the FALWA theory, the corresponding components in the package that implement the calculations, and where users can find sample notebooks to start with. It aims to serve as a road map for new users to easily navigate through this package. The latter half of this article documents the practices of the developers, which include the documentation tools, continuous integration practice, and repository maintenance using automated GitHub functionalities. The authors also discuss existing numerical issues and future improvement plans. This open-source project aims to promote the broader application of FALWA diagnostics on climate data and model outputs by streamlining complex numerical computations.

The research presented in this manuscript is independent of Epsilon Data Management LLC.

Open-Source Software Details

GitHub Repository: https://github.com/csyhuang/hn2016_falwa/.

Python package: falwa (Finite-Amplitude Local Wave Activity).

Documentation page: https://hn2016-falwa.readthedocs.io/. GitHub Repository

DOI: 10.5281/zenodo.6366562.

1 | Introduction

Recent generalisation of the Rossby wave activity theory by Nakamura and collaborators to the finite-amplitude wave regime (Nakamura and Zhu 2010; Nakamura and Solomon 2010) and subsequently to the longitudinally localised regime (Huang and Nakamura 2016, 2017) has enabled a mechanistic study of localised nonlinear Rossby wave phenomena. The finite-amplitude local wave activity (FALWA) (Huang and Nakamura 2016) diagnostic theory is applicable to gridded meteorological data to quantify the contribution of various physical processes to the growth and decay of finite-amplitude waves. Nakamura and Huang (2018) observed in ERA-Interim data that the wave activity budget of atmospheric blocking, a nonlinear wave phenomenon often associated with heat waves and droughts, is analogous to traffic flow density in a traffic jam (Lighthill and Whitham 1955; Nakamura and Huang 2017). Subsequent studies evaluate the hypothesis by applying FALWA to blocking events observed in model ensembles (Polster and Wirth 2023) and reanalysis data (Barpanda and Nakamura 2024), and also study the contributing physical processes to various stages of the blocking life cycle (Wang et al. 2021; Neal et al. 2022; Lubis et al. 2024). The diagnostic has also been used successfully to evaluate the effects of diabatic heating on the climatological waviness of the jet stream (Lee and Nakamura 2024).

Despite the demonstrated utility of FALWA to elucidate nonlinear wave dynamics, this theory has not been widely adopted in the scientific community presumably due to the barriers in (i) understanding the theory and (ii) translating theory into data analysis. Traditional Eulerian theories of wave-mean flow interactions decompose the flow into zonal mean and eddy component. In contrast, the finite-amplitude wave theory discussed herein defines eddy as deviation from a reference state, which is an invariant state under conservative dynamics (Nakamura and Zhu 2010; Nakamura and Solomon 2010, 2011). The latitude coordinate system, namely, the equivalent latitude (Allen and Nakamura 2003), is computed based on the conservative arrangement of potential vorticity (PV) which involves surface integral of PV. The zonal wind reference state u_{REF} requires a solver of an elliptic equation with appropriate surface boundary conditions (Nakamura and Solomon 2010; Nakamura and Huang 2018; Neal et al. 2022). All these computations involve non-trivial numerical methods.

An overview of theoretical development and examples of applications are given by Nakamura (2024). In this article, we mainly address the second barrier, namely, we wish to provide a portable solution for climate scientists to readily analyse their observation data and climate model output with this diagnostic with minimal technical glitches. Since the first publication of the FALWA series (Huang and Nakamura 2016), we have open-sourced the code that can reproduce the published numerical results via our GitHub repository https://github.com/csyhuang/hn2016_falwa/. This repository has been constantly updated to include new calculations in subsequent publications (Nakamura and Huang 2018; Neal et al. 2022; Lubis et al. (under review)) and serves to fulfil the open-science requirement of the academic journals. We refactored the reusable utilities that can be used to analyse other climate datasets into the Python package `falwa`. This paper will give a formal introduction to this package, `falwa`, which can perform complex numerical computations reviewed in Sections 3 and 4.

The focus of this software paper will be on the usage of the Python package instead of the physics involved. We will introduce the minimal concepts, equations and notations for describing the functionality of `falwa` in Section 3. Readers interested in the formula derivation and scientific discussions shall refer to the cited references there. Section 4 provides a road map for users familiar with the FALWA to navigate through the package structure and choose the suitable components they need to answer particular scientific questions. Section 6 discusses the necessary treatment of input climate model data. Section 7 reviews the maintenance practices currently in use for this open-source GitHub repository. Section 8 summarises the known limitations and plan for future development of the package.

For detailed requirements of input parameters, most up-to-date usage instructions and sample jupyter notebook demonstration, please refer to the package documentation. As this open source project is still being actively developed, this paper will only cover package functionalities up to release `v2.0.0` (August 17, 2024).

2 | Related Work

There exist Python packages that implement a subset of the computation procedures that are available in the `falwa` package. For instance, `xinvert` (Qian 2023) is a Python package that provides numerical solvers to balanced equations in geophysical fluid dynamics problems, that is, problems in atmospheric sciences and physical oceanography, formulated as second-order partial differential equations. This package computes the quasigeostrophic potential vorticity (QGPV) of the geophysical fluid of interest, while `falwa` computes QGPV particularly for atmospheric air. Another python package by the same author, `xcontour` (Qian 2022), computes the adiabatic reference states proposed in Nakamura and Solomon (2010) and the LWA proposed in Huang and Nakamura (2016), but not the local wave activity fluxes (Huang and Nakamura 2016, 2017; Nakamura and Huang 2018; Nakamura 2024) in the evolution equation of local wave activity. To our knowledge, `falwa` is the first and only existing open-source software package that implements the complete set of finite-amplitude local wave activity diagnostics, including the computation of reference states, local wave activity, and its fluxes (i.e., Equations 3a and 3b).

Regarding utilisation of `falwa`, there are two examples of software packages that the authors know of which import `falwa` as dependencies. The first one is `barotropic` (Polster 2024) by one of the authors, which simulates barotropic fluid on a sphere and uses `falwa` as a flow field diagnostic. Another one is `climopy` (Davis 2021), a succinct toolset under development by a climate researcher for analysing climate data.

3 | Finite-Amplitude Local Wave Activity (FALWA)

Definitions of symbols used are listed in Table 1. For a detailed review of the finite-amplitude wave theory, please refer to Nakamura (2024). A discussion on the limit of validity of this theory ('Breakdown of Quasigeostrophic Assumption') is included in S1.

TABLE 1 | Definition of symbols. *Note that the eddy shall be defined in terms of angular momentum $\omega = u/a\cos\phi$ such that $u_e(\lambda,\phi,\phi',z,t) \equiv u(\lambda,\phi+\phi',z,t)\cos\phi/\cos(\phi+\phi') - u_{\mathrm{REF}}(\phi,z,t)$. See corrigendum of Nakamura and Huang (2018) for details.

List of symbols	
$a = 6378$ km	Radius of earth
$H = 7$ km	Reference scale height
$p_0 = 1000$ hPa	Reference surface pressure
$R = 287$ J K^{-1} kg^{-1}	Gas constant for dry air
$c_p = 1004$ J K^{-1} kg^{-1}	Specific heat at constant pressure (dry air)
$\kappa = R/c_p = 0.286$	
$\Omega = 7.29\times10^{-5}$ s^{-1}	Rotation rate of the Earth
$\lambda \in [0, 2\pi]$ [radian]	Longitude (0 at the IRM)
$\phi \in [-0.5\pi, 0.5\pi]$ [radian]	Latitude (0 at the equator)
$z = -H\ln(p/p_0)$ [m]	Pseudoheight (p [hPa] is hydrostatic pressure)
$f = 2\Omega\sin\phi$ [s^{-1}]	Coriolis parameter (planetary vorticity)
t [s]	Time
$u(\lambda,\phi,z,t) = a\cos\phi(D\lambda/Dt)$ [ms^{-1}]	Zonal wind speed (positive eastward)
$v(\lambda,\phi,z,t) = a(D\phi/Dt)$ [ms^{-1}]	Meridional wind speed (positive northward)
$\theta(\lambda,\phi,z,t) = T(p/p_0)^{-\kappa}$ [K]	Potential temperature (T [K] is air temperature)
$\widetilde{\theta}(z,t)$ [K]	Area-mean θ over the hemisphere
$q(\lambda,\phi,z,t)$ [s^{-1}]	Potential vorticity
$(u_{\mathrm{REF}}, \theta_{\mathrm{REF}}, q_{\mathrm{REF}})(\phi,z,t)$	(u,θ,q) in a wave-free reference state ($v_{\mathrm{REF}} = 0$)
$\Delta\phi(\lambda,\phi,z,t)$ [radian]	Meridional displacement of contour $q = q_{\mathrm{REF}}$ from ϕ
$\phi' \in [0, \Delta\phi]$ [radian]	Displacement latitude relative to ϕ
$((u_e, v_e, \theta_e, q_e))(\lambda,\phi,\phi',z,t)$	Deviation* of (u,v,θ,q) from $(u_{\mathrm{REF}}, 0, \theta_{\mathrm{REF}}, q_{\mathrm{REF}})$ at ϕ
$\widetilde{A}^*(\lambda,\phi,z,t)$ [ms^{-1}]	Local wave activity [see (1)]
$\widetilde{\dot{A}}^*(\lambda,\phi,z,t)$ [ms^{-2}]	Nonconservative source and sink of $\widetilde{A}^*(\lambda,\phi,z,t)$
$\langle\cdots\rangle(\lambda,\phi,t) = \int_0^\infty(\cdots)e^{-z/H}dz / \int_0^\infty e^{-z/H}dz$	Density-weighted column average operator

Compared to the widely used linear Rossby wave theory, the advancement brought about by the finite-amplitude wave theory lies partly in how the reference states are redefined to be a (quasi-)conservative eddy-free state, which is loosely related to Kelvin's circulation around a wavy contour of PV. Traditionally, the zonal mean is taken as the reference states to define eddies, but the zonal mean can be modified by finite-amplitude eddies even in the absence of non-conservative forces; thus, it is not an adiabatic invariant.

Given that the quasi-geostrophic (QG) PV (q) is materially conserved, under geostrophic advection, an eddy-free reference state of QGPV $q_{\mathrm{REF}}(\phi,z,t)$ can be defined by the area enclosed by the contour $q = q_{\mathrm{REF}}(\phi,z,t)$ (Allen and Nakamura 2003; Nakamura and Zhu 2010). Via the invertibility of PV (Hoskins et al. 1985), one could obtain reference-states for zonal wind u_{REF} and potential temperature θ_{REF} by numerically solving an elliptic equation in global/hemispheric domain with appropriate boundary conditions (see Table 4 for details). We can interpret u_{REF} as a hypothetical 'initial state' of zonal wind devoid of eddies, from which the current (wavy) state could evolve adiabatically.

The finite-amplitude local wave activity $\widetilde{A}^*(\lambda,\phi,z,t)$, is computed as the line integral of QGPV deviation from q_{REF} along its contour excursion (Huang and Nakamura (2016), equation 1; Huang and Nakamura (2017), Equation 1):

$$\widetilde{A}^*(\lambda,\phi,z,t) = -\frac{a}{\cos\phi_e}\int_0^{\Delta\phi} q_e(\lambda,\phi,\phi',z,t)\cos(\phi+\phi')d\phi'. \tag{1}$$

where $q_e(\lambda,\phi,\phi',z,t) \equiv q(\lambda,\phi+\phi',z,t) - q_{\mathrm{REF}}(\phi,z,t)$. Taken literally, it measures the meridional displacement of QGPV from the reference state and hence the waviness of the flow. Physically, it quantifies the angular pseudomomentum density carried by the Rossby wave (Nakamura 2024). The zonal mean of (1) recovers the finite-amplitude wave activity (Nakamura and Zhu 2010; Nakamura and Solomon 2010) $A^*(z,\phi,t)$. FALWA is

a nonnegative quantity by construction and tends to maximise where the meandering of the jet stream is great. However, its true utility lies in the fact that it satisfies a relatively simple budget that is easy to interpret and evaluable with meteorological data.

Denoting the density-weighted column average by

$$\langle \dots \rangle \equiv \frac{\int_0^\infty \rho_0(\dots)dz}{\int_0^\infty \rho_0 dz} \equiv \frac{\int_0^\infty \rho_0(\dots)dz}{H}, \tag{2}$$

the governing equations of the column budget of LWA are (Nakamura 2024):

$$\frac{\partial \langle \tilde{A}^* \rangle}{\partial t}(\lambda,\phi,t)\cos\phi = \underbrace{-\frac{1}{a\cos\phi}\frac{\partial}{\partial\lambda}\langle F_\lambda\rangle}_{\text{conv.ofzonaladvectiveflux}} \underbrace{+\frac{1}{a\cos\phi}\frac{\partial}{\partial\phi}\left(\langle u_e v_e\rangle \cos^2\phi\right)}_{\text{div.ofeddymomentumflux}} \underbrace{+\frac{f}{H}\frac{v_e\theta_e\cos\phi}{\partial\tilde{\theta}/\partial z}\Big|_{z=0}}_{\text{low-levelmeridionalheatflux}} \underbrace{+\langle \dot{\tilde{A}}^* \rangle\cos\phi}_{\text{residual}}, \tag{3a}$$

$$\langle F_\lambda \rangle \equiv \underbrace{\langle u_{\mathrm{REF}}\tilde{A}^*\rangle\cos\phi}_{\text{adv. flux f1}} \underbrace{-a\langle \int_0^{\Delta\phi} u_e q_e \cos(\phi+\phi')d\phi' \rangle}_{\text{adv. flux f2}} + \underbrace{\frac{1}{2}\langle v_e^2 - u_e^2 - \frac{R}{H}\frac{e^{-\kappa z/H}\theta_e^2}{\partial\tilde{\theta}/\partial z}\rangle\cos\phi}_{\text{adv. flux f3}}. \tag{3b}$$

Under the quasigeostrophic approximation, the last term of (3a), $\left\langle \dot{\tilde{A}}^* \right\rangle \cos\phi$, reflects non-conservative sources and sinks of FALWA. Integrating (3a) and compare the LHS with the RHS within an episode of weather event, one can quantify the contributions of different physical processes, expressed in terms of different flux terms of RHS, to the growth and decay of FALWA (LHS) throughout the episode.

To summarise, the evaluation of the wave activity budget involves 3 steps:

I. Interpolate input data onto regular pseudo-height grid, compute static stability and QGPV.

II. Compute reference states $q_{\mathrm{REF}}(\phi,z,t)$, $u_{\mathrm{REF}}(\phi,z,t)$ and $\theta_{\mathrm{REF}}(\phi,z,t)$.

III. Compute (column-budget) wave activity and its fluxes (3a, 3b).

These computation procedures are encapsulated by the class `QGFieldBase` in the `falwa` package, which will be introduced in detail in Section 5.

4 | GitHub Repository Structure

The package `falwa` is located in the `src/`directory of the GitHub repository. It consists of the following directories:

- `src/falwa/`: the package directory which contains the Python modules and Fortran source code (`.f90` files in subdirectory `src/falwa/f90_modules/`) to be compiled to F2PY modules upon installation by the Meson Build system (Pakkanen and the Meson Development Team 2024)
- `notebooks/`: collection of jupyter notebooks (`.ipynb`) which contain sample codes calling `falwa` and visualise analysis results. New users are recommended to start with these examples.
- `tests/`: unit tests for functionalities in `falwa`. They are expected to pass after a user has properly compiled the F2PY modules and installed the package.
- `scripts/`: sample code (`*.py`) for specific analyses that utilise `falwa`
- `docs/`: detailed documentation of `falwa` in Sphinx

Section 5 that follows will introduce the modules that contain the public interfaces of `falwa` which encapsulate the computations outlined in Section 3.

5 | Package Structure and Public Interface

5.1 | `Falwa.Oopinterface`—Object-Oriented Interface to Computations

`QGFieldBase` is the abstract class which defines the interface to access functionalities computing QGPV, reference states, local wave activity and barotropic components of wave activity (and) fluxes for stratified 3D-quasigeostrophic flow from *a snapshot in time* of 3D-weather field variables (u, v, T) (λ, ϕ, z). The physical fields required, coordinates and physical constants required for the `QGFieldBase` constructor are partially listed in Table 2. The flow of the 3-step computational procedures are described in Figure 1. The detailed descriptions of the three corresponding public methods mentioned in Section 3 are listed in Table 3.

The details of the three steps will be described in the following subsections.

5.1.1 | Step (I): Field Interpolation and QGPV Computation on Evenly-Spaced Pseudoheight Grid

This step is invoked by `QGFieldBase.interpolate_fields()`.

As downstream tasks involve the computation of spatial derivatives and surface/line integrals, to simplify the discretization scheme, the pressure coordinate is first converted onto a pseudoheight grid: $z = -H\log p/p_o$. The input physical fields are then interpolated onto an evenly-spaced (Δz) pseudoheight grid.

TABLE 2 | Input arguments for `QGField` constructor. Refer to the package documentation page for an exhaustive list of user-defined parameters and physical constants.

Variable type	Variable name	Symbol	Unit	Description
Coordinates	xlon	λ	degree	Longitude
	ylat	ϕ	degree	Latitude
	plev	p	hPa	Pressure
Climate data (physical fields)	u_field	u	ms^{-1}	Zonal wind
	v_field	v	ms^{-1}	Meridional wind
	t_field	T	K	Temperature
Physical constants	scale_height	H	meter	Scale height of the atmosphere
	cp	c_p	$JK^{-1}kg^{-1}$	Specific heat at constant pressure (dry air)
	omega	Ω	s^{-1}	Rotation rate of the earth
	...	...		...
User-defined parameters	kmax	k_{max}	—	Number of pseudoheight levels
	dz	Δz	m	Incremental pseudoheight between levels
	eq_boundary_index	n/a	—	Equatorward boundary of latitudinal domain
	...	...	...	...

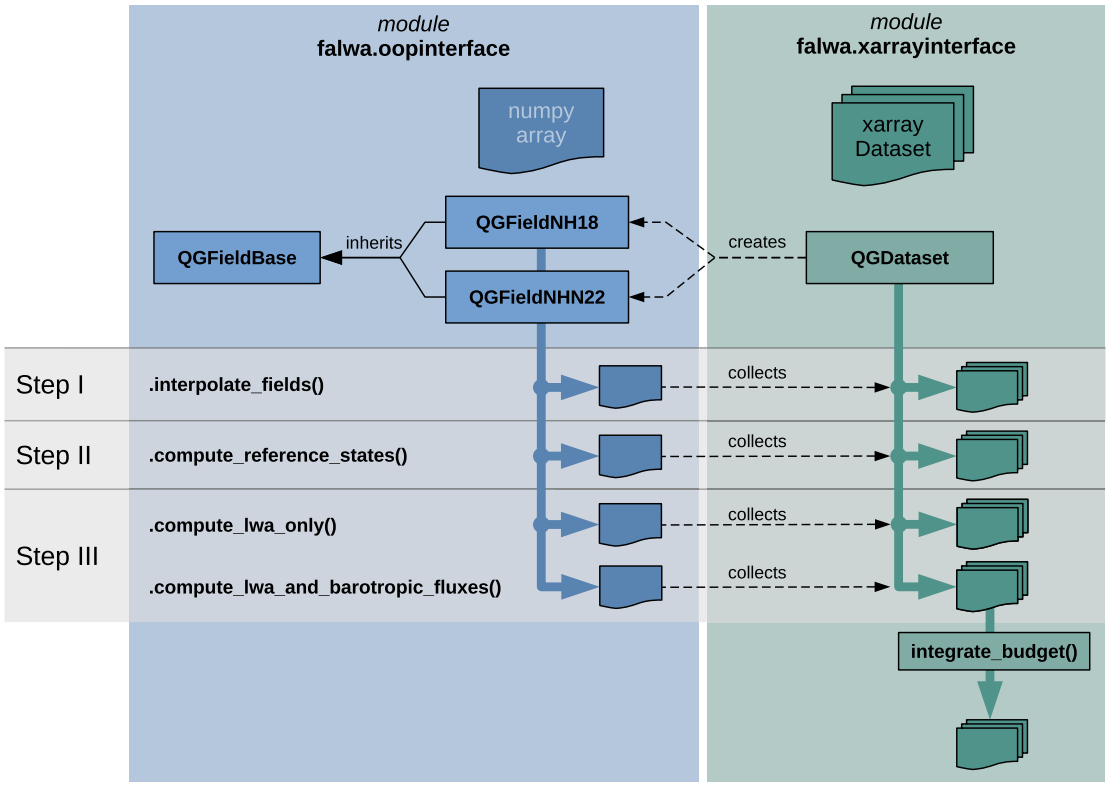

FIGURE 1 | A schematic diagram showing the structure of the public interface of `falwa`. The parent class `QGFieldBase` (Section 5.1) defines the computational steps listed in Table 3, and the corresponding child classes encapsulate the actual implementations. The xarray interface `QGDataset` (Section 5.2) manages sequential `QGFieldBase` objects that contain data at different time steps, and implement time-series analyses such as time-integrated budget analysis. Refer to Section 5 and Table 3 for detailed descriptions of each method/component.

TABLE 3 | List of public methods of `QGFieldBase` and their descriptions, prerequisites to implement them, and their respective outputs.

Method	Operations	Prerequisite	Output
`interpolate_fields`	**Step I** Interpolate u, v, theta onto regular pseudoheight grid. Compute QGPV correspondingly.	$u(\lambda,\phi,z)$, $v(\lambda,\phi,z)$, $T(\lambda,\phi,z)$, have non-null values on all grid points	• interpolated_u • interpolated_v • interpolated_t • static_stability • qgpv
`compute_reference_states`	**Step II** Compute q_{REF} by box counting. Then solve elliptic equation involving q_{REF} with proper boundary conditions to obtain u_{REF} and θ_{REF} . See Table 4 for detailed implementation of the subclasses of QGFieldBase.	`interpolate_fields` has been called.	• q_{REF} qref • u_{REF} uref* • θ_{REF} ptref* *Note that the returned values of these variables are not physical when the elliptic equation solver does not converge. In that case, an error/warning will be raised (depends on user's choice).
`compute_lwa_only`	**Step III** Compute $\widetilde{A}^*$ and $\left\langle \widetilde{A}^* \right\rangle$ only	`compute_reference_states` has been called such that q_{REF} has been computed.	• lwa_baro $\left\langle \widetilde{A}^* \right\rangle \cos\phi$ • u_baro $\langle u \rangle \cos\phi$ • lwa A^*
`compute_lwa_and_barotropic_fluxes`	**Step III** Compute $\widetilde{A}^*$, $\left\langle \widetilde{A}^* \right\rangle$ and all the terms in the RHS of equations 1	`compute_reference_states` has been called and elliptic equation solver for u_{REF} converges.	• adv_flux_f1 • adv_flux_f2 • adv_flux_f3 • convergence_zonal_advective_flux • divergence_eddy_momentum_flux • meridional_heat_flux lwa_baro $\left\langle \widetilde{A}^* \right\rangle \cos\phi$ • u_baro $\langle u \rangle \cos\phi$ • lwa A^*

The computation of QGPV $q(\lambda,\phi,z,t)$ involves the calculation of (1) relative vorticity by taking the curl of the horizontal velocity field and (2) static stability which involves a derivative along the dimension of pseudoheight. Derivatives are computed numerically using central-differencing scheme.

5.1.2 | Step (II): Compute Reference State by Solving Elliptic Equation

This step is invoked by `QGFieldBase.compute_reference_states()`.

`QGFieldBase` has two subclasses: `QGFieldNH18` and `QGFieldNHN22`, which respectively implement the reference state solvers (i.e., Step (II)) with numerical procedures and boundary conditions documented in Nakamura and Huang (2018) and Neal et al. (2022). The computation is invoked by `QGFieldBase.compute_reference_states()`.

Equivalent latitude, $Q(\phi,z,t)$, is first computed based on the hemispheric distribution of $q(\lambda,\phi,z,t)$. $q_{\mathrm{REF}}(\phi,z,t)$ is then served as an input to solve for $u_{\mathrm{REF}}(\phi,z,t)$ (See Table 4 for the equations involved). The potential reference state temperature $\theta_{\mathrm{REF}}(\phi,z,t)$ is obtained by inverting the thermal wind relation with the upper boundary conditions specified. The two implementations of elliptic equation solver for $u_{\mathrm{REF}}(\phi,z,t)$ based on the same no-slip lower boundary conditions, but different latitudinal boundary conditions, are shown below in Table 4.

The reason for switching from the boundary condition used in Nakamura and Huang (2018) (`QGFieldNH18`) to that in Neal et al. (2022) (`QGFieldNHN22`) is that, the solver `QGFieldNH18` sometimes does not converge when there is a sign change in zonal mean of $\left(u+\widetilde{A}^*\right)$ at the equator, which is kind of arbitrary (as QG assumption actually breaks down there and the sign change does not have direct implication to midlatitude dynamics). The boundary conditions of `QGFieldNHN22`, which is a few degrees away from the equator, avoids introducing spurious eddy forcing, and the solution is not sensitive to the precise latitude chosen as the boundary.

5.1.3 | Step (III): Compute Wave Activity and Corresponding Barotropic Flux Components

This step is invoked by `QGFieldBase.compute_lwa_and_barotropic_fluxes()`.

Taking $q_{\mathrm{REF}}(\phi,z,t)$, $u_{\mathrm{REF}}(\phi,z,t)$ and $\theta_{\mathrm{REF}}(\phi,z,t)$ as input, this method computes Equation (1) and barotropic components of all terms except the residual on the RHS of Equations (3a) and (3b).

TABLE 4 | Comparison between the solvers of $u_{\mathrm{REF}}(\phi,z,t)$ among the `QGFieldBase` subclasses.

QGFieldBase subclass	QGFieldNH18	QGFieldNHN22
Equations to solve	Nakamura and Huang (2018) corrigendum, equations S10–S11	Neal et al. (2022), equations S13–S16
Equatorward latitudinal boundary condition	$u_{\mathrm{REF}}(\phi,z,t) = \left(\overline{u}+\overline{A}\right)(\phi,z,t)$ at $\phi = 0^{\circ}$	$u_{\mathrm{REF}}(\phi,z,t)\cos\phi = \frac{K(\phi,z,t)-2\pi\Omega a^2\cos^2\phi}{2\pi a}$ at $\phi = 5^{\circ}$ N where $K(\phi,z,t)$ is the Kelvin's circulation at 5° N
Solver for linear system of equations	Successive over-relaxation (SOR) iterative solver written in Fortran (`falwa/f90_modules/compute_reference_states.f90`)	Matrix inversion using LU factorization from LAPACK in SciPy (`from scipy.linalg.lapack import dgetri, dgetrf`

The implementations of `QGFieldNH18` and `QGFieldNHN22` differ only in the definition of latitudinal domain for computing the line integral $\int_0^{\Delta\phi}(\ldots)d\phi'$.

If reference states were not solved properly (See Section 8.2 for possible reasons), users have the option to call `QGFieldBase.compute_lwa_only` to skip the computation of wave activity fluxes (See Table 3).

5.2 | `Falwa.Xarrayinterface`—Metadata-Aware Wrapper for Computations

`QGDataset` is a wrapper based on `xarray` that manages multiple `QGFieldBase` objects in a *sequence* (See Figure 1). Users can input snapshots of climate data, for example, a time series of climate data, or outputs from an ensemble of models, into `QGDataset` and invoke the three public methods of `QGFieldBase` on the snapshots simultaneously. The collection of computed variables will be returned as an `xarray.Dataset` object with coordinate information and metadata attached.

`QGDataset` provides a convenient interface to time-series budget analysis (e.g., Polster and Wirth (2023)). The methods for operating on the output of `QGDataset` are located in `falwa.xarrayinterface`. By applying the method `integrate_budget` on the output `xarray.Dataset` object from `QGDataset.compute_lwa_and_barotropic_fluxes` with the range of time steps specified (denoted by t_{start} and t_{end}), the terms in Equation (5) below, that is, the time-integrated version of Equations (3a) and (3b), are computed.

$$\underbrace{\int_{t_{\mathrm{end}}}^{t_{\mathrm{start}}} \langle \dot{\tilde{A}}^{*} \rangle \cos\phi dt}_{\text{residual}} = \underbrace{\left(\langle \tilde{A}^{*} \rangle(\lambda,\phi,t_{\mathrm{end}}) - \langle \tilde{A}^{*} \rangle(\lambda,\phi,t_{\mathrm{start}})\right)\cos\phi}_{\text{delta\_lwa}} + \underbrace{\int_{t_{\mathrm{end}}}^{t_{\mathrm{start}}} \left(\frac{1}{a\cos\phi}\frac{\partial}{\partial\lambda}\langle F_{\lambda}\rangle\right)dt}_{\text{integrated\_convergence\_zonal\_advective\_flux}} - \underbrace{\int_{t_{\mathrm{end}}}^{t_{\mathrm{start}}} \frac{1}{a\cos\phi}\frac{\partial}{\partial\phi}\left(\langle u_e v_e\rangle\cos^2\phi\right)dt}_{\text{integrated\_divergence\_eddy\_momentum\_flux}} - \underbrace{\int_{t_{\mathrm{end}}}^{t_{\mathrm{start}}} \frac{f}{H}\frac{v_e\theta_e\cos\phi}{\partial\tilde{\theta}/\partial z}|_{z=0}dt}_{\text{integrated\_meridional\_heat\_flux}} \tag{4}$$

Below is an example code block to compute the integrated budget for 5 time slices:

```
>>> qgds = QGDataset(data).
>>> ...
>>> terms = qgds.compute_lwa_and_barotropic_
fluxes().
>>> integrated_budget_terms = integrate_bud-
get(terms.isel({"time": slice(5, 10)})).
```

The integrated budget terms, for example the residual, are conveniently accessible from the output `Dataset` object:

```
>>> residual = integrated_budget_terms.
residual
```

5.3 | `Falwa.Basis`—Functional Interface to Fundamental Numerical Routines

The python module `falwa.basis` consists of stand-alone functions implemented in pure `numpy` to compute equivalent

TABLE 5 | Input and output arguments of function `falwa.basis.eqvlat_fawa`, which computes the equivalent latitude `qref`, and finite-amplitude (zonal-mean) wave activity `fawa` assuming input `vort` is potential vorticity if `output_fawa=True`. Refer to the package documentation page for detailed usage.

Type	Name	Description
Input	`ylat`	1-d numpy array of latitude (in degree)
	`vort`	2-d numpy array of vorticity or tracer field values
	`area`	2-d numpy array specifying differential areal element of each grid point
	`n_points`	Analysis resolution (number of grid points) used to calculate equivalent latitude.
	`planet_radius`	Radius of spherical planet of interest consistent with input `area`.
	`output_fawa`	Boolean indicating whether to output FAWA assuming `vort` is vorticity.
Output	`qref`	1-d numpy array of value $q_{\mathrm{REF}}(y)$ on y-grid `ylat`
	`fawa`	if `output_fawa=True`, returns 1-d FAWA computed from vorticity field `vort`, otherwise `None`.

latitude and finite-amplitude (local) wave activity based on an input field on a spherical surface using the box-counting method. The functionality, input and output of the two main functions in this module, `falwa.basis.eqvlat_fawa` and `falwa.basis.lwa`, are respectively listed in Tables 5 and 6.

Providing PV as the input `vort` to `falwa.basis.eqvlat_fawa`, one could calculate equivalent latitude and (zonal mean) wave activity in a spherical barotropic model as in Solomon and Nakamura (2012) and Huang and Nakamura (2016). To obtain equivalent latitude for passive tracers as in Allen and Nakamura (2003), one could use the tracer field as input `vort` (the optional argument `output_fawa` shall be set to `False`).

If researchers are interested in applying the local wave activity formula to detect anomalies in terms of zonal asymmetry, as in Huang and Nakamura (2016) figures 8 and 9, on other physical fields such as geopotential height (Martineau et al. 2017) or water vapour (Lu et al. 2018), they can do so by first computing the 'equivalent latitude' `qref` of the field of interest `var` with `falwa.eqvlat_fawa` by substituting `vort=var`, followed by inputting `var` and `qref` into the function `basis.lwa` to compute the local anomaly measures. The partition of this measure into cyclonic and anti-cyclonic contributions can be done with slight modification, which has been made available in `falwa v2.1.0` released on March 15, 2025.

TABLE 6 | Input and output arguments of function `falwa.basis.lwa`, which computes local wave activity based on the input field `vort` and corresponding equivalent latitude `q_part`. Refer to the package documentation page for detailed usage.

Type	Name	Description
Input	`nlon`	Longitudinal dimension of `vort`
	`nlat`	Latitudinal dimension of `vort`
	`vort`	2-d numpy array of vorticity (or tracer) values
	`q_part`	1-d numpy array of equivalent latitude, i.e., output `qref` from `falwa.basis.eqvlat_fawa`.
	`dmu`	1-d numpy array of latitudinal differential length element.
Output	`lwact`	2-d numpy array of local wave activity calculated

6 | Treatment of Input Model Data

Regardless of the interface used, since the finite-amplitude wave theory is defined on the equivalent latitude grid, which is constructed based on *globally conserved* quantities, the input gridded data have to be *well defined on all grid points*. Previous publications by the authors process reanalysis data (e.g., ERA-Interim [Dee et al. 2011]; ERA-5 [Hersbach et al. 2020]) or model data with 'underground' grid points assigned reasonable physical field values, so using `QGFieldBase` in those studies does not require further pre-processing of data.

However, if one wants to analyse output from General Circulation Models (GCMs) with topography incorporated such that 'underground' grid points are masked out (e.g., in the form of masked `numpy` arrays), one has to fill in those missing values before passing them as arguments into `QGFieldBase`. A Poisson solver can, for instance, perform a grid-fill such that 'underground' grid points are assigned non-extremum values that have minimal impact on downstream computations (Lubis et al. 2018a, 2018b; and personal communication). The artificially filled-in grid points shall be masked out when presenting the final results.

7 | Maintenance of the Code Repository

7.1 | Continuous Integration

The GitHub repo hosting `falwa` has been using GitHub workflow to automatically test the compilation of all components in `falwa`, run through *unit tests* and *integration tests* in the test suite (modules in `tests/`constructed with `pytest`), and deliver a test coverage report with warnings upon a detected significant drop in coverage via the free service `codecov.io`. This workflow is triggered by pushing or merging new commits onto the `master` branch of the repo.

Because not all possible runtime environments (operating system, Python interpreter, package versions, etc.) can be covered

by the GitHub workflow, users are advised via the `readme.md` of the repo to run through all tests locally after installation of `falwa` to ensure all components are working as expected.

7.2 | Documentation

With the structure and configuration files stored in the repo directory `docs/`, upon push to remote repository, documentation is compiled automatically via the online platform `ReadtheDocs`: https://hn2016-falwa.readthedocs.io/.

The documentation of the package's interface is compiled with `Sphinx` (https://www.sphinx-doc.org/) based on the docstring of each function or class. The repo directory `notebooks/`contains jupyter notebook files which demonstrate the usage of `falwa`. The jupyter notebooks are included in the documentation pages with `nbsphinx` ('Jupyter Notebook Tools for Sphinx').

8 | Outlook and Future Work

This open-source package `falwa` is still under active development. The authors have been actively addressing issues raised on the GitHub Issue page: https://github.com/csyhuang/hn2016_falwa/issues. It does not only serve as a channel to report bugs/issues in the repo, but also a platform for scientific discussion regarding the use of finite-amplitude wave theory. The authors welcome proposals of new functionalities and collaboration in incorporating them into `falwa` for the use of the community.

There are three aspects in which improvement can be made to the current version of `falwa`:

8.1 | Scientific Calculations

8.1.1 | Vertical Profile of Wave Activity Fluxes

The package `falwa` so far only contains functionalities to implement climate data analyses published by the authors (Nakamura and Huang 2018; Neal et al. 2022; Polster and Wirth 2023) which studied the dynamics of atmospheric blocking. Given the mostly barotropic structure of atmospheric blocking, analysing the vertically averaged wave activity budget suffices to answer the most pertinent scientific questions. However, the finite-amplitude Rossby wave theory itself can address a wider scope of questions. To study longitudinally localised phenomena that involve vertical propagation of waves, one shall compute the 3-dimensional wave activity budget (and corresponding flux vectors, if desired). The calculation of the 3-dimensional wave activity budget will be available in `falwa v2.2.0` tentatively released in April 2025.

8.1.2 | Contribution of Diabatic Heating to Wave Activity Budget

Previous studies take the residual term $\langle \dot{A} \rangle \cos\phi$ as a measure of overall non-conservative forcing, which consists of (positive) diabatic forcing, (negative) diffusion and frictional damping (Huang and Nakamura 2017; Neal et al. 2022; Lee and Nakamura 2024). To back out the diabatic source on wave activity, if the diabatic heating term is available in the reanalysis data/model output, one could theoretically estimate the contribution of diabatic heating to the wave activity budget (i.e., longitudinally local version of $\Delta\Sigma$ in Nakamura and Zhu (2010) equation (23) or Lubis et al. (2018a) equation (6)). This calculation of $\Delta\Sigma$ will be included in `QGFieldBase` class in release `v2.1.0` along with the calculation results published in Lubis et al. (under review).

8.2 | Numerical Issues

8.2.1 | Convergence of the Elliptic Equation Solver

It has been reported by users multiple times that the elliptic equation solver to compute reference states (Step (II)) does not converge for datasets with a latitudinal grid resolution finer than 1°. The current way to resolve this issue is to subsample the input fields on a coarser grid and interpolate the outputs back to the original finer grid. As the reference state is typically smooth, this should not affect the results adversely. The authors are still researching into how to solve this purely numerical issue.

8.2.2 | Dependence of Results on Discretisation Scheme and Resolution

Given the numerical differentiation and integration required to compute local wave activity and fluxes, the analysis results depend on the discretization scheme and resolution used. The discretization schemes used in `falwa` are simple ones of low order: central differencing for differentiation and the rectangular rule for integration. Higher order discretization schemes are desired to reduce analysis error. On top of the dependence on discretization scheme, since LWA takes into account the fine-scale structure of QGPV, the budget does depend quantitatively on the analysis resolution. For users to optimise the computation cost, it would be useful to carry out sensitivity analysis on discretization resolution as well.

8.3 | Performance Optimization

8.3.1 | Computation Speeds of `Numpy` vs. F2PY Modules

A large portion of numerical procedures is implemented by `F2PY` modules for the sake of computing speed compared to using `numpy`, which in turn compromises readability and poses challenges in code maintenance and portability, as `F2PY` modules have to be compiled by Fortran compilers. The authors are still in search of a solution that could improve portability without compromising computing speed. Users are advised to use the `numpy` version with MKL released by Intel,[1] which significantly speeds up the LU factorization procedures in `QGFieldNHN22`.

8.3.2 | Memory-Efficient and Dask-Aware QGDataset

When a `QGDatset` object is created, all the `QGFieldBase` objects managed by the Dataset are immediately instantiated

and kept in memory for later use. For larger datasets, this eager evaluation requires substantial amounts of main memory to be available throughout the lifetime of the `QGDatset` object. This is acceptable when analysing a small number of climate data snapshots but quickly leads to memory issues when analysing data that covers an extensive period. The authors are still exploring solutions to avoid this memory-intensive behaviour, for example, by instantiating and evaluating the `QGFieldBase` objects lazily.

Another improvement for `QGDataset` would be to make it dask-aware, that is, to allow for distributing the computations onto multiple processors, enabling parallelism and manipulation of even larger datasets.

Acknowledgements

This work is an extension of C.H.'s PhD dissertation work at the University of Chicago. C.P.'s work on the package was partially carried out within the Transregional Collaborative Research Center 165 'Waves to Weather' funded by the German Science Foundation (DFG). N.N. is funded by NSF Award 2154523 and 2448990. The authors would like to thank Sandro Lubis for helpful discussions on the preprocessing procedures of model data mentioned in Section 6. This project is hosted on GitHub with continuous integration supported by GitHub Workflow and Codecov.[2] The documentation is compiled by and hosted on the community platform `ReadTheDocs`.[3] The authors thank the anonymous referee for valuable comments on our manuscript.

Conflicts of Interest

The authors declare no conflicts of interest.

Data Availability Statement

The data that support the findings of this study are openly available in hn2016_falwa at https://github.com/csyhuang/hn2016_falwa.

Endnotes

[1] Intel Distribution for Python: https://www.intel.com/content/www/us/en/developer/tools/oneapi/distribution-for-python.html. Retrieved on September 30, 2024.

[2] Codecov: Code Coverage Testing & Insights Solution: https://codecov.io.

[3] https://readthedocs.com/.

References

Allen, D. R., and N. Nakamura. 2003. "Tracer Equivalent Latitude: A Diagnostic Tool for Isentropic Transport Studies." *Journal of the Atmospheric Sciences* 60, no. 2: 287–304.

Barpanda, P., and N. Nakamura. 2024. "Local Wave-Activity Analysis of Atmospheric Blocks in the Northern Hemisphere Winter." *Authorea Preprints*. https://doi.org/10.22541/essoar.171441675.50570950.

Davis, L. 2021. "Climopy." https://github.com/climopy-dev/climopy.

Dee, D. P., S. M. Uppala, A. J. Simmons, et al. 2011. "The Era-Interim Reanalysis: Configuration and Performance of the Data Assimilation System." *Quarterly Journal of the Royal Meteorological Society* 137, no. 656: 553–597.

Hersbach, H., B. Bell, P. Berrisford, et al. 2020. "The Era5 Global Reanalysis." *Quarterly Journal of the Royal Meteorological Society* 146, no. 730: 1999–2049.

Hoskins, B. J., M. E. McIntyre, and A. W. Robertson. 1985. "On the Use and Significance of Isentropic Potential Vorticity Maps." *Quarterly Journal of the Royal Meteorological Society* 111, no. 470: 877–946.

Huang, C. S. Y., and N. Nakamura. 2016. "Local Finite-Amplitude Wave Activity as a Diagnostic of Anomalous Weather Events." *Journal of the Atmospheric Sciences* 73: 211–229.

Huang, C. S. Y., and N. Nakamura. 2017. "Local Wave Activity Budgets of the Wintertime Northern Hemisphere: Implication for the Pacific and Atlantic Storm Tracks." *Geophysical Research Letters* 44: 5673–5682.

Lee, H.-I., and N. Nakamura. 2024. "Imprint of Diabatic Processes in the Waviness of the Jet Stream: An Analysis of Local Wave Activity Budget." *Journal of Climate* 37: 5703–5719.

Lighthill, M. J., and G. B. Whitham. 1955. "On Kinematic Waves II. A Theory of Traffic Flow on Long Crowded Roads." *Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences* 229, no. 1178: 317–345.

Lu, J., D. Xue, Y. Gao, G. Chen, L. Leung, and P. Staten. 2018. "Enhanced Hydrological Extremes in the Western United States Under Global Warming Through the Lens of Water Vapor Wave Activity." *NPJ Climate and Atmospheric Science* 1, no. 1: 7.

Lubis, S. W., B. Harrop, J. Lu, et al. 2024. "Cloud-Radiative Effects Significantly Increase Wintertime Atmospheric Blocking in the Euro-Atlantic Sector." (Under review).

Lubis, S. W., C. S. Y. Huang, and N. Nakamura. 2018a. "Role of Finite-Amplitude Eddies and Mixing in the Life Cycle of Stratospheric Sudden Warmings." *Journal of the Atmospheric Sciences* 75: 3987–4003.

Lubis, S. W., C. S. Y. Huang, N. Nakamura, N. E. Omrani, and M. Jucker. 2018b. "Role of Finite-Amplitude Rossby Waves and Nonconservative Processes in Downward Migration of Extratropical Flow Anomalies." *Journal of the Atmospheric Sciences* 75: 1385–1401.

Martineau, P., G. Chen, and D. A. Burrows. 2017. "Wave Events: Climatology, Trends, and Relationship to Northern Hemisphere Winter Blocking and Weather Extremes." *Journal of Climate* 30, no. 15: 5675–5697.

Nakamura, N. 2024. "Large-Scale Eddy-Mean Flow Interaction in the Earth's Extratropical Atmosphere." *Annual Review of Fluid Mechanics* 56: 349–377.

Nakamura, N., and C. S. Y. Huang. 2017. "Local Wave Activity and the Onset of Blocking Along a Potential Vorticity Front." *Journal of the Atmospheric Sciences* 74: 2341–2362.

Nakamura, N., and C. S. Y. Huang. 2018. "Atmospheric Blocking as a Traffic Jam in the Jet Stream." *Science* 361: 42–47.

Nakamura, N., and A. Solomon. 2010. "Finite-Amplitude Wave Activity and Mean Flow Adjustments in the Atmospheric General Circulation. Part I: Quasigeostrophic Theory and Analysis." *Journal of the Atmospheric Sciences* 67: 3967–3983.

Nakamura, N., and A. Solomon. 2011. "Finite-Amplitude Wave Activity and Mean Flow Adjustments in the Atmospheric General Circulation. Part II: Analysis in the Isentropic Coordinate." *Journal of the Atmospheric Sciences* 68: 2783–2799.

Nakamura, N., and D. Zhu. 2010. "Finite-Amplitude Wave Activity and Diffusive Flux of Potential Vorticity in Eddy–Mean Flow Interaction." *Journal of the Atmospheric Sciences* 67: 2701–2716.

Neal, E., C. S. Y. Huang, and N. Nakamura. 2022. "The 2021 Pacific Northwest Heat Wave and Associated Blocking: Meteorology and the Role of an Upstream Cyclone as a Diabatic Source of Wave Activity." *Geophysical Research Letters* 49: e2021GL097699.

Pakkanen, J., and The Meson Development Team. 2024. "The Meson Build System." https://mesonbuild.com.

Polster, C. 2024. "Barotropic." https://github.com/chpolste/barotropic.

Polster, C., and V. Wirth. 2023. "The Onset of a Blocking Event as a "Traffic Jam": Characterization With Ensemble Sensitivity Analysis." *Journal of the Atmospheric Sciences* 80: 1681–1699.

Qian, Y.-K. 2022. "xcontour." https://github.com/miniufo/xcontour.

Qian, Y.-K. 2023. "Xinvert: A Python Package for Inversion Problems in Geophysical Fluid Dynamics." *Journal of Open Source Software* 8, no. 89: 5510.

Solomon, A., and N. Nakamura. 2012. "An Exact Lagrangian-Mean Wave Activity for Finite-Amplitude Disturbances to Barotropic Flow on a Sphere." *Journal of Fluid Mechanics* 693: 69–92.

Wang, M., Y. Zhang, and J. Lu. 2021. "The Evolution Dynamical Processes of Ural Blocking Through the Lens of Local Finite-Amplitude Wave Activity Budget Analysis." *Geophysical Research Letters* 48, no. 10: e2020GL091727.

Supporting Information

Additional supporting information can be found online in the Supporting Information section.