

ODEM v1.0 User Manual

Metin Baykara

March 2026

Contents

1	Introduction	3
2	Requirements	3
2.1	Software	3
2.2	Hardware	3
3	Input data	3
3.1	Meteorological forcing	3
3.1.1	ERA5	3
3.1.2	MERRA-2	3
3.2	SoilGrids v2.0	4
3.3	Aerodynamic roughness	4
3.4	MODIS LAI	4
3.5	Land cover erodibility mask	4
4	Directory structure	4
5	Running the model	5
5.1	Command-line interface	5
5.2	Single file vs. directory input	5
5.3	Multi-month parallel runs	5
6	Output files	6
6.1	Monthly emission files	6
6.2	Annual mean file	6
6.3	Diagnostic fields	6
7	Post-processing	7
7.1	odem_analysis.py	7
7.1.1	Summary mode	7
7.1.2	Comparison mode	7
7.1.3	Seasonal cycle	7
7.1.4	Output directory	7
7.2	odem_verify.py	7
8	Physics parameters	8
9	Interpreting the output	8
9.1	Units and size range	8
9.2	Global budget	8
9.3	Regional budgets	8
10	Troubleshooting	9
11	Example workflow	9

12 Test case **10**
12.1 Unit tests (no data required) 10
12.2 Single-month test case 10
12.3 Input data sources 10

13 Code availability **10**

1 Introduction

ODEM (Offline Dust Emission Model) is a standalone Python model that computes global gridded dust emission fluxes from reanalysis meteorological forcing. It implements the brittle fragmentation emission scheme of Kok et al. (2014) with the parameter choices and implementation of Leung et al. (2023). ODEM accepts either ERA5 or MERRA-2 as meteorological forcing and produces emission fields at the native resolution of each reanalysis.

This manual describes how to install, configure, and run ODEM v1.0, and how to interpret its output. The emission physics are described in the companion paper; this manual focuses on practical usage.

2 Requirements

2.1 Software

- Python 3.10 or later
- NumPy
- SciPy
- xarray
- netCDF4
- pandas
- matplotlib (for post-processing only)

Install all dependencies:

```
pip install -r requirements.txt
```

2.2 Hardware

- Minimum: 1 CPU core, 4 GB RAM (MERRA-2 at 0.5°)
- Recommended: 1 CPU core, 16 GB RAM (ERA5 at 0.25°)
- Disk: ~20 GB for input data; ~3.5 GB output per year (ERA5 1-hourly) or ~250 MB (MERRA-2)

Each calendar month runs independently. For full-year simulations, 12 monthly jobs can be submitted in parallel (e.g. via SLURM).

3 Input data

ODEM requires five categories of input data: meteorological forcing, soil properties, aerodynamic roughness, vegetation index, and land cover. All static datasets are freely available; meteorological forcing requires registration with the respective data provider.

3.1 Meteorological forcing

Exactly one forcing source must be specified. ERA5 and MERRA-2 are mutually exclusive.

3.1.1 ERA5

Download from the Copernicus Climate Data Store (<https://cds.climate.copernicus.eu>).

- Resolution: 0.25°, 1-hourly (or 3-hourly)
- File naming: era5_dust_YYYYMM.nc
- Required variables: *zust* (friction velocity), *swvl1* (volumetric soil moisture, layer 1), *t2m* (2 m temperature), *d2m* (2 m dewpoint), *sp* (surface pressure), *sd* (snow depth), *lsm* (land-sea mask)
- Optional: *blh* (boundary layer height, for intermittency correction)

Registration and a CDS API key (~/.cdsapirc) are required.

3.1.2 MERRA-2

Download from the NASA GES DISC (<https://disc.gsfc.nasa.gov>).

- Resolution: 0.5°×0.625°, 1-hourly
- File naming: merra2_dust_YYYYMM.nc
- Required variables: *USTAR* (friction velocity), *SFMC* (surface soil moisture), *SNODP* (snow depth)

- Optional: PBLH (boundary layer height), RHOA (air density — used directly if present; otherwise computed from temperature and pressure)
 - Land mask: derived from `DISPH > 0` or SFMC validity
 - Collections: M2T1NXFLX (surface flux), M2T1NXLND (land surface)
- A NASA Earthdata login is required.

3.2 SoilGrids v2.0

Download from <https://soilgrids.org>.

Required files in the `-soilgrids` directory:

- `SG_clay.nc4` — topsoil clay fraction (%)
- `SG_silt.nc4` — topsoil silt fraction (%)
- `SG_bdod.nc4` — bulk density (kg m^{-3} ; optional)

If `SG_bdod.nc4` is absent, bulk density is estimated from porosity: $\rho_b = (1 - \phi) \times \rho_p$, where $\phi = 0.45 + 0.05 \times (f_{clay}/100)$.

Fallback filenames `T_CLAY.nc4` and `T_SILT.nc4` (HWSD format) are also accepted.

3.3 Aerodynamic roughness

The aeolian roughness length z_{0a} from Prigent et al. (2005) is a single NetCDF file at 0.25° resolution. The variable `Z0a` is stored in cm and converted to m internally.

Pass via `-prigent`:

```
--prigent prigent_data/Prigent_2005_roughness_0.25x0.25.nc
```

3.4 MODIS LAI

8-day MODIS LAI composites (MOD15A2H), regridded to 0.5° . The files must be in a directory searchable recursively and follow the naming convention containing a YYYYMMDD date substring:

```
MOD15A2H_UHAM-ICDC__20060101__fv0.03.nc
MOD15A2H_UHAM-ICDC__20060109__fv0.03.nc
...
```

The nearest composite in time is used for each model timestep. Fill values (255) are treated as $\text{LAI}=0$. Values are clipped to $[0, 10]$.

3.5 Land cover erodibility mask

MODIS MCD12C1 land cover (NASA LP DAAC), preprocessed into an erodibility fraction $f_{erod} \in [0, 1]$. Barren, open shrubland, and sparse vegetation classes receive positive erodibility; forests, water, snow, and urban areas are masked to zero. If `-landcover` is not specified, $f_{erod} = 1$ everywhere (all land cells can emit).

4 Directory structure

A recommended directory layout for a full-year ERA5 simulation:

```
project/
  odem.py
  odem_analysis.py
  odem_verify.py
  requirements.txt
  era5_data/
    era5_dust_200601.nc
    era5_dust_200602.nc
    ...
    era5_dust_200612.nc
  soilgrids/
    SG_clay.nc4
    SG_silt.nc4
```

```

SG_bdod.nc4
prigent_data/
  Prigent_2005_roughness_0.25x0.25.nc
modis_lai/2006/
  MOD15A2H_UHAM-ICDC__20060101__fv0.03.nc
...
modis_landcover/
  modis_landcover_erodibility_2006.nc
output_2006/      <-- created by odem.py
  odem_200601.nc
...
  odem_200612.nc
  odem_annual_mean.nc
  odem_diag_mean.nc  <-- with --save-diags

```

5 Running the model

5.1 Command-line interface

ODEM is invoked from the command line. Exactly one of `-era5` or `-merra2` must be specified.

```

python odem.py \
  --era5 era5_data/ \
  --soilgrids soilgrids/ \
  --prigent prigent_data/Prigent_2005_roughness_0.25x0.25.nc \
  --modis-lai modis_lai/2006 \
  --landcover modis_landcover/modis_landcover_erodibility_2006.nc \
  --output-dir output_2006

```

Table 1 lists all command-line arguments.

Argument	Default	Description
<code>-era5</code> PATH	—	ERA5 NetCDF file or directory
<code>-merra2</code> PATH	—	MERRA-2 NetCDF file or directory
<code>-soilgrids</code> PATH	(required)	Directory with SoilGrids files
<code>-prigent</code> PATH	(required)	Prigent z_{0a} NetCDF
<code>-modis-lai</code> PATH	(required)	MODIS LAI directory
<code>-landcover</code> PATH	None	Erodibility mask NetCDF; if omitted, $f_{erod} = 1$ everywhere
<code>-output-dir</code> PATH	output_v2	Output directory
<code>-C-tune</code> FLOAT	0.05	Global tuning constant C_{tune}
<code>-save-diags</code>	False	Save time-mean diagnostic fields

Table 1: ODEM command-line arguments.

5.2 Single file vs. directory input

When `-era5` or `-merra2` points to a single file, ODEM processes that file (typically one month). When pointing to a directory, ODEM reads all matching files (`era5_dust_*.nc` or `merra2_dust_*.nc`), sorted chronologically, and processes them sequentially. Monthly output files are written as each month completes. The annual mean file (`odem_annual_mean.nc`) is written only when two or more months are processed in a single invocation.

5.3 Multi-month parallel runs

For production runs, it is efficient to run each month as a separate job:

```

# Example: 12 parallel jobs (bash loop)
for m in 01 02 03 04 05 06 07 08 09 10 11 12; do

```

```
python odem.py \
  --era5 era5_data/era5_dust_2006${m}.nc \
  --soilgrids soilgrids/ \
  --prigent prigent_data/Prigent_2005_roughness_0.25x0.25.nc \
  --modis-lai modis_lai/2006 \
  --landcover modis_landcover/modis_landcover_erodibility_2006.nc \
  --output-dir output_2006 &
done
wait
```

Each job writes its own monthly output file (`odem_YYYYMM.nc`). The annual mean must be computed afterwards from the 12 monthly files, either by running ODEM with the full directory or by using `odem_analysis.py`.

On HPC systems, each monthly job can be submitted as a separate SLURM task. A single ERA5 month at 1-hourly resolution requires approximately 4 hours on one CPU core with 16 GB memory.

6 Output files

6.1 Monthly emission files

Filename: `odem_YYYYMM.nc`

- Dimensions: (time, lat, lon)
- Variable: `F_d` — dust emission flux [$\text{kg m}^{-2} \text{s}^{-1}$]
- Coordinates: lat, lon, time
- Compression: zlib level 4

Each timestep is stored independently. For ERA5 at 1-hourly resolution, a single month contains 720–744 timesteps.

6.2 Annual mean file

Filename: `odem_annual_mean.nc`

Written only for multi-month runs (two or more months in a single invocation). Contains:

- `F_d_mean` — time-averaged emission flux [$\text{kg m}^{-2} \text{s}^{-1}$]
- `F_d_max` — maximum instantaneous flux [$\text{kg m}^{-2} \text{s}^{-1}$]
- Attributes: `period` (start–end dates), `n_timesteps` (total timesteps averaged)

6.3 Diagnostic fields

Filename: `odem_diag_mean.nc` (requires `-save-diags`)

Time-mean values of all intermediate variables, enabling component-level verification. Key variables:

Variable	Unit	Description
<code>rhoa</code>	kg m^{-3}	Air density
<code>ustar_ft0</code>	m s^{-1}	Dry fluid threshold u_{*ft}^0
<code>ustar_ft_wet</code>	m s^{-1}	Moisture-corrected threshold
<code>ustar_it</code>	m s^{-1}	Impact threshold ($0.82 \times u_{*ft}^0$)
<code>wnd_frc_slt</code>	m s^{-1}	Saltation friction velocity u_{*s}
<code>f_bare</code>	1	Bare erodible soil fraction
<code>f_eff_r</code>	1	Rock drag partition (Marticorena & Bergametti)
<code>f_eff_v</code>	1	Vegetation drag partition (Okin)
<code>F_eff</code>	1	Combined drag partition ($f_{eff,r} \times f_{eff,v}$)
<code>kappa</code>	1	Fragmentation exponent (capped at 3.0)
<code>C_d</code>	1	Erodibility coefficient
<code>eta</code>	1	Intermittency factor (0–1)
<code>f_moisture</code>	1	Soil moisture correction factor

Table 2: Diagnostic variables.

7 Post-processing

7.1 odem_analysis.py

The companion analysis script produces summary statistics, maps, and comparison plots.

7.1.1 Summary mode

Analyses a single output file (annual mean or single month):

```
# Annual mean
python odem_analysis.py --summary output/odem_annual_mean.nc

# Single month (annualises the total)
python odem_analysis.py --summary output/odem_200601.nc --monthly
```

Produces: emission map, zonal mean profile, regional budget bar chart, and printed global/regional statistics.

7.1.2 Comparison mode

Compares two annual mean files side-by-side:

```
python odem_analysis.py --compare \
    output_era5/odem_annual_mean.nc \
    output_m2/odem_annual_mean.nc
```

Produces: side-by-side maps on the same colour scale, difference map, spatial correlation, and per-region comparison table.

7.1.3 Seasonal cycle

Reads all monthly files from a directory and plots the seasonal cycle:

```
python odem_analysis.py --seasonal output/ --outdir plots/
```

7.1.4 Output directory

All plots are saved to the directory specified by `-outdir` (default: `odem_plots/` in the current working directory).

7.2 odem_verify.py

The verification script tests ODEM output at five levels:

1. **Physics unit tests** — formula-level checks at known inputs (no data files required)
2. **Diagnostic ranges** — intermediate fields within physical bounds
3. **Spatial sanity** — emission patterns match dust source climatology
4. **Global and regional budget** — comparison with Kok et al. (2021) observational constraints
5. **Input data checks** — ERA5 grid and variable consistency

```
# Physics unit tests only (no data files needed)
python odem_verify.py --unit-tests

# Full verification
python odem_verify.py \
    --unit-tests \
    --emission output/odem_annual_mean.nc \
    --diags output/odem_diag_mean.nc \
    --era5 era5_data/era5_dust_200601.nc \
    --outdir verify_results/

# Single-month run (annualises budget check)
python odem_verify.py \
    --emission output/odem_200601.nc \
    --monthly --outdir verify_results/
```

8 Physics parameters

All tunable constants are defined in the `CONST` dictionary at the top of `odem.py`. Table 3 lists the parameters that control the emission physics.

Parameter	Symbol	Default	Description
<code>C_tune</code>	C_{tune}	0.05	Global tuning constant (Leung et al. 2023)
<code>C_d0</code>	C_{d0}	4.4×10^{-5}	Baseline erodibility coefficient
<code>C_e</code>	C_e	2.0	Erodibility exponent
<code>C_alpha</code>	C_α	2.7	Fragmentation exponent coefficient
<code>kappa_cap</code>	—	3.0	Maximum fragmentation exponent
<code>ustar_st0</code>	u_{*st}^0	0.16 m s^{-1}	Reference threshold
<code>A_SL</code>	A_{SL}	0.0123	Shao & Lu aerodynamic coefficient
<code>gamma_SL</code>	γ	$1.65 \times 10^{-4} \text{ kg s}^{-2}$	Interparticle cohesion
<code>B_it</code>	B_{it}	0.82	Impact/fluid threshold ratio
<code>D_p_arid</code>	$D_{p,arid}$	$127 \mu\text{m}$	Median diameter (arid cells)
<code>Psi_0</code>	Ψ_0	$7.8 \times 10^{-6} \text{ m}$	Non-arid intercept
<code>Psi_1</code>	Ψ_1	$1.24 \times 10^{-4} \text{ m}$	Non-arid slope
<code>LAI_thr</code>	—	$1.0 \text{ m}^2 \text{ m}^{-2}$	LAI threshold for arid classification
<code>f0</code>	f_0	0.32	Okin friction ratio
<code>c_okin</code>	c	4.8	Okin e-folding distance
<code>X</code>	X	10.0 m	Inter-obstacle distance (MB95)

Table 3: ODEM physics parameters with default values.

The only parameter intended for user modification is C_{tune} , which can be set via the `-C-tune` command-line flag. All other parameters are fixed at the values used by Leung et al. (2023). Changing them requires editing the `CONST` dictionary in `odem.py`.

9 Interpreting the output

9.1 Units and size range

ODEM output is in $\text{kg m}^{-2} \text{ s}^{-1}$. This is the total dust emission flux across all particle sizes, as determined by the Kok (2011) brittle fragmentation size distribution. To compare with observational estimates that report PM20 (particles with diameter $d < 20 \mu\text{m}$), divide the ODEM total by approximately 2.5 (i.e. $\text{PM20} \approx 40\%$ of total emission).

9.2 Global budget

For the year 2006, the expected global totals are:

- ODEM-ERA5 (1-hourly): $\sim 15\,500 \text{ Tg yr}^{-1}$
- ODEM-M2 (1-hourly): $\sim 12\,750 \text{ Tg yr}^{-1}$
- Kok et al. (2021) PM20 constraint: $5000 \pm 1600 \text{ Tg yr}^{-1}$

To convert from the output units to Tg yr^{-1} , integrate the emission flux over the grid cell area and time:

$$E_{global} = \sum_i F_{d,i} \times A_i \times \Delta t$$

where A_i is the grid cell area (m^2) and Δt is the timestep duration (s). This integration is performed automatically by `odem_analysis.py -summary`.

9.3 Regional budgets

ODEM uses the 11 dust source regions defined by Kok et al. (2021). The regional breakdown is printed by `odem_analysis.py -summary` and can be compared with the observational constraints in Table 2 of Kok et al. (2021).

10 Troubleshooting

No MODIS LAI files found

The `-modis-lai` directory must contain `.nc` files with an 8-digit date substring (YYYYMMDD) in the filename. Files are searched recursively.

No annual mean file produced

The annual mean is only written when two or more months are processed in a single invocation. For parallel monthly runs, compute the annual mean from the 12 monthly files using `odem_analysis.py`.

High memory usage with ERA5

ERA5 at 0.25° has >1 million grid cells. Peak memory is approximately 16 GB for a single-month 1-hourly run. If memory is limited, use MERRA-2 (~4 GB) or reduce the temporal resolution of the ERA5 input.

Different global totals from monthly files vs. annual mean

If the annual mean was written by a single-month run (before the v1.0 fix), it contains only that month's data. Verify using `odem_analysis.py -summary`: the `n_timesteps` attribute should be 8760 (1-hourly) or 2920 (3-hourly) for a full year.

11 Example workflow

A complete workflow for a one-year ERA5 simulation:

1. **Obtain input data** from the sources listed in Sect. 3. Prepare monthly ERA5 files as `era5_dust_YYYYMM.nc` and place static datasets in the directory structure shown in Sect. 4.
2. **Run ODEM** (12 months in parallel):

```
for m in 01 02 03 04 05 06 07 08 09 10 11 12; do
  python odem.py \
    --era5 era5_data/era5_dust_2006${m}.nc \
    --soilgrids soilgrids/ \
    --prigent prigent_data/Prigent_2005_roughness_0.25x0.25.nc \
    --modis-lai modis_lai/2006 \
    --landcover modis_landcover/modis_landcover_erodibility_2006.nc \
    --output-dir output_2006 \
    --save-diags &
done
wait
```

3. **Compute annual mean** (from 12 monthly files):

```
python odem.py \
  --era5 era5_data/ \
  --soilgrids soilgrids/ \
  --prigent prigent_data/Prigent_2005_roughness_0.25x0.25.nc \
  --modis-lai modis_lai/2006 \
  --landcover modis_landcover/modis_landcover_erodibility_2006.nc \
  --output-dir output_2006
```

Alternatively, compute the annual mean from existing monthly output using `odem_analysis.py -summary`.

4. **Verify:**

```
python odem_verify.py --unit-tests
python odem_verify.py \
  --emission output_2006/odem_annual_mean.nc \
  --diags output_2006/odem_diag_mean.nc \
  --outdir verify/
```

5. **Analyse:**

```
python odem_analysis.py --summary output_2006/odem_annual_mean.nc
python odem_analysis.py --seasonal output_2006/ --outdir plots/
```

12 Test case

This section describes how to verify that ODEM is installed correctly and produces expected results.

12.1 Unit tests (no data required)

The physics verification suite tests all emission formulas at known inputs without requiring any external data files:

```
python odem_verify.py --unit-tests
```

All 38 tests should pass within floating-point tolerance. If any test fails, the installation is incorrect.

12.2 Single-month test case

To verify a full model run, process January 2006 with ERA5 forcing and compare against the reference output archived on Zenodo (10.5281/zenodo.19390605).

1. Download ERA5 data for January 2006 from the Copernicus Climate Data Store (<https://cds.climate.copernicus.eu>).
2. Download static datasets from their original providers (Sect. 3).
3. Run:

```
python odem.py \  
  --era5 era5_data/era5_dust_200601.nc \  
  --soilgrids soilgrids/ \  
  --prigent prigent_data/Prigent_2005_roughness_0.25x0.25.nc \  
  --modis-lai modis_lai/2006 \  
  --landcover modis_landcover/modis_landcover_erodibility_2006.nc \  
  --output-dir test_output \  
  --save-diags
```

4. Verify against reference values:

```
python odem_analysis.py --summary test_output/odem_200601.nc --monthly
```

Expected output for January 2006 (ERA5, 1-hourly, $C_{tune} = 0.05$):

- Global total (annualised): $\sim 15\,500\text{ Tg yr}^{-1}$ ($\pm 5\%$ depending on ERA5 version)
- North Africa dominates ($>60\%$ of global emission)
- Oceans, forests, and ice-covered regions produce zero emission
- All diagnostic fields in `odem_diag_mean.nc` should have physically reasonable ranges (e.g. $0 < u_{*s} < 2\text{ m s}^{-1}$, $0 < \eta \leq 1$, $0 \leq f_{bare} \leq 1$)

A reference output file for this test case is included in the Zenodo archive for direct comparison.

12.3 Input data sources

All input datasets are freely available from their original providers. ODEM does not redistribute these datasets.

- **ERA5:** Copernicus Climate Data Store (<https://cds.climate.copernicus.eu>; free registration required)
- **MERRA-2:** NASA GES DISC (<https://disc.gsfc.nasa.gov>; NASA Earthdata login required)
- **SoilGrids v2.0:** <https://soilgrids.org>
- **MODIS land cover (MCD12C1):** NASA EOSDIS LP DAAC (<https://lpdaac.usgs.gov>)
- **MODIS LAI (MOD15A2H):** NASA EOSDIS LP DAAC
- **Prigent roughness:** Prigent, C., Tegen, I., Aires, F., Marticorena, B., and Zribi, M.: Estimation of the aerodynamic roughness length in arid and semi-arid regions over the globe with the ERS scatterometer, *J. Geophys. Res.*, 110, D09205, <https://doi.org/10.1029/2004JD005370>, 2005; available from the dataset authors

The required variables, file naming conventions, and resolutions for each dataset are documented in Sect. 3.

13 Code availability

ODEM v1.0 source code is available at <https://github.com/Metn/odem> and archived on Zenodo (10.5281/zenodo.19390605) under the MIT licence. The Zenodo archive includes the source code, user manual, and reference test case output for January 2006.