Package: odin.dust (via r-universe)

November 11, 2026

Title Compile Odin to Dust

Version 0.3.13

Description Less painful than it sounds, this package compiles an odin model to use dust, our new stochastic model system. Supports only a subset of odin models (discrete time stochastic models with no interpolation and no delays).

License MIT + file LICENSE

Encoding UTF-8

Language en-GB

URL https://github.com/mrc-ide/odin.dust

BugReports https://github.com/mrc-ide/odin.dust/issues

Imports R6, brio, cpp11, decor, dust (>= 0.15.1), odin (>= 1.5.0), tibble, vctrs

Suggests dde, knitr, mockery, pkgload, rmarkdown, socialmixr, testthat

RoxygenNote 7.2.2

Roxygen list(markdown = TRUE)

VignetteBuilder knitr

Remotes mrc-ide/dust, mrc-ide/odin

Config/pak/sysreqs cmake make libuv1-dev

Repository https://ncov-ic.r-universe.dev

Date/Publication 2024-10-02 15:02:46 UTC

RemoteUrl https://github.com/mrc-ide/odin.dust

RemoteRef master

RemoteSha bb34b58b785b7a1724f76d236d847a31626288a5

Contents

odin_dust 2
odin_dust_options 3
odin_dust_package 4

Index **5**

`odin_dust` *Create a dust odin model*

Description

Compile an odin model to work with dust.

Usage

```
odin_dust(x, ..., options = NULL)

odin_dust_(x, ..., options = NULL)
```

Arguments

`x`	Either the name of a file to read, a text string (if length is greater than 1 elements will be joined with newlines) or an expression.
`...`	Arguments passed to odin_dust_options, including `real_type`, `gpu`, `verbose`, `workdir`, `no_check_unused_equations` and `rewrite_dims`.
`options`	An odin::odin_options or odin_dust_options object. If given it overrides arguments; if it is already a `odin_dust_options` object it is returned unmodified. Otherwise it is passed through to odin::odin_options where it will override arguments in `...` but respond to the `odin_dust` specific options (`real_type`, etc)

Details

Note that this does not (yet) support the full odin output object, instead creating the more limited dust interface. However, for many uses this should be considerably faster than the interface that odin normally uses (built on dde).

Including custom code

When including custom C++ code you may want to set additional options in order to enable compilation. You can do this by including pseudo-attributes

- `// [[odin.dust::cpp_std(C++17)]]` - use this to change the C++ standard used in compilation; this is passed to `dust::dust()` as the `cpp_std` option. It is only necessary to pass in values greater than C++11 at present as that is dust's default.
- `// [[odin.dust::linking_to(pkg)]]` - use this to make include files present in an R package (e.g., BH) available. You can use as many of these attributes as you need.

`odin_dust_options` *Options for odin_dust*

Description

Options for controlling odin_dust; this is a superset of odin::odin_options

Usage

```
odin_dust_options(
  ...,
  real_type = NULL,
  rng_state_type = NULL,
  gpu = NULL,
  gpu_generate = NULL,
  compiler_options = NULL,
  optimisation_level = NULL,
  options = NULL
)
```

Arguments

`...`	Arguments passed to odin::odin_options, including `verbose`, `workdir`, `no_check_unused_equations` and `rewrite_dims`.
`real_type`	C++ type to use for real (floating point) numbers. Defaults to `double`.
`rng_state_type`	C++ type to use for the random number generator. Defaults to `dust::random::generator<real_type>` which selects a generator based on your real type. The default prior to dust 0.10.0 was `dust::random::xoshiro256starstar_state`.
`gpu`	**Experimental!** Generate support for running models on a GPU. This implies `gpu_generate` but *does* require a gpu and nvcc toolchain. Currently not supported within package code. This argument will be passed through to `dust::dust()` and so to enable compilation for a gpu, pass either `TRUE` or the results of dust::dust_cuda_options)
`gpu_generate`	**Experimental** Generate gpu support code. This does not require a gpu or nvcc toolchain, but creates code that could be compiled for a gpu. This is primarily intended for testing and development as the generated code will be slower than the normal cpu version, and the compilation time will be considerably slower. Currently not supported within package code.
`compiler_options`	Additional options for the C++ compiler, passed to dust::dust
`optimisation_level`	Shorthand control over optimisation level, passed to dust::dust. If provided should be one of `none` (slow running, fast building), `standard` or `max` (enables fast math)
`options`	An odin::odin_options or odin_dust_options object. If given it overrides arguments; if it is already a `odin_dust_options` object it is returned unmodified. Otherwise it is passed through to odin::odin_options where it will override arguments in `...` but respond to the `odin_dust` specific options (`real_type`, etc)

Value

A list of options (class `odin_options`) to pass to `odin.dust::odin_dust`

Examples

```
odin.dust::odin_dust_options()
```

`odin_dust_package` *Update package code*

Description

Update generated code in a package that uses odin and dust to provide a model. This will generate new dust code in `inst/dust` and from that generate a full model in `src`, and an R interface in `R/dust.R`, along with the cpp11 attributes that are needed to use the model.

Usage

```
odin_dust_package(path, options = NULL)
```

Arguments

`path`	Path to the package root (the directory that contains `DESCRIPTION`)
`options`	An odin::odin_options or odin_dust_options object. If given it overrides arguments; if it is already a `odin_dust_options` object it is returned unmodified. Otherwise it is passed through to odin::odin_options where it will override arguments in `...` but respond to the `odin_dust` specific options (`real_type`, etc)

Value

The path to the package

Index

dust::dust, *3*
dust::dust(), *2*, *3*
dust::dust_cuda_options, *3*

odin::odin_options, *2–4*
odin_dust, 2, *3*
odin_dust_ *(*odin_dust*)*, 2
odin_dust_options, *2*, 3, *3*, *4*
odin_dust_package, 4