Package: amatrix.arrayfire (via r-universe)

July 2, 2026

Type Package

Title ArrayFire Backend Scaffold for amatrix

Version 0.0.0.9000

Description ArrayFire backend package for amatrix. Registers a backend contract implementation and uses the ArrayFire C API for dense matmul when ArrayFire is available at build time.

License MIT + file LICENSE

Encoding UTF-8

Depends R (>= 4.3)

Imports amatrix, methods, jsonlite

Suggests testthat (>= 3.0.0)

SystemRequirements ArrayFire (>= 3.8; optional, see ARRAYFIRE_PREFIX)

NeedsCompilation yes

Config/testthat/edition 3

Repository https://bbuchsbaum.r-universe.dev

Date/Publication 2026-07-02 13:18:49 UTC

RemoteUrl https://github.com/bbuchsbaum/amatrix

RemoteRef HEAD

RemoteSha aa1a39a22def2841186cc195dd1b4c7e0d5ff8e2

RemoteSubdir backends/amatrix.arrayfire

Contents

amatrix_arrayfire_register *ArrayFire Backend for amatrix*

Description

Functions for registering and inspecting the ArrayFire GPU backend. The backend is optional: if ArrayFire is not installed the package loads but all GPU operations fall back to CPU.

Usage

```
amatrix_arrayfire_register(overwrite = TRUE)
amatrix_arrayfire_enable_probe(register = TRUE)
amatrix_arrayfire_is_available()
amatrix_arrayfire_native_available()
amatrix_arrayfire_capabilities()
amatrix_arrayfire_diagnostics()
amatrix_arrayfire_bridge_info()
amatrix_arrayfire_backend()
```

Arguments

overwrite	logical; whether to overwrite a previously registered backend of the same name (default TRUE).
register	logical; whether to register the backend with **amatrix** after enabling the runtime probe (default TRUE).

Details

Call amatrix_arrayfire_register() once per session (or place it in '.Rprofile') to make the ArrayFire backend available for dispatch. ArrayFire registration is opt-in; loading **amatrix.arrayfire** no longer auto-registers the backend unless options(amatrix.enable_arrayfire = TRUE) is set before package load.

amatrix_arrayfire_enable_probe() activates the runtime probe for the current session by setting AMATRIX_ARRAYFIRE_PROBE_GPU=1, enables optional registration, and returns the probed availability invisibly.

amatrix_arrayfire_is_available() returns TRUE when ArrayFire was found at build time and initialises successfully at runtime. Availability checks are probe-gated so ordinary package load and backend listing stay safe.

amatrix_arrayfire_diagnostics() returns details about the active compute backend, device count, and LAPACK availability.

The ARRAYFIRE_PREFIX environment variable can be set before installation to point to a non-standard ArrayFire location.

Value

`amatrix_arrayfire_register` returns the backend name invisibly.

`amatrix_arrayfire_enable_probe` returns the probed availability invisibly.

`amatrix_arrayfire_is_available` and `amatrix_arrayfire_native_available` return a scalar logical.

`amatrix_arrayfire_capabilities` returns a character vector of supported operation names.

`amatrix_arrayfire_diagnostics` and `amatrix_arrayfire_bridge_info` return named lists with diagnostic information.

`amatrix_arrayfire_backend` returns the backend contract list used internally by **amatrix**.

See Also

`amatrix::amatrix_backend_names()`, `amatrix::amatrix_backend_status()`

Index