

# Package: PINNProgCens (via r-universe)

October 9, 2026

**Type** Package

**Title** Physics-Informed Neural Networks for Progressive Censoring

**Version** 0.1.0

**Description** Implementation of Physics-Informed Neural Networks ('PINN') for lifetime estimation under progressive Type-II censoring schemes. Combines parametric baseline hazards with physical differential degradation models.

**License** MIT + file LICENSE

**Encoding** UTF-8

**Imports** deSolve, stats

**Config/roxygen2/version** 8.0.0

**NeedsCompilation** no

**Author** Okechukwu J. Obulezi [aut, cre]

**Maintainer** Okechukwu J. Obulezi <oj.obulezi@unizik.edu.ng>

**Repository** <https://obulezi-oj.r-universe.dev>

**Date/Publication** 2026-08-09 08:10:15 UTC

**RemoteUrl** <https://github.com/cran/PINNProgCens>

**RemoteRef** HEAD

**RemoteSha** bbd5d7cf4e31409259c8e1547381fbe123817271

## Contents

|                                 |   |
|---------------------------------|---|
| compute_ode_residuals . . . . . | 2 |
| default_ode_system . . . . .    | 2 |
| fit_pinn_progcens_mle . . . . . | 3 |
| interpolate_state . . . . .     | 4 |
| pinn_hazard . . . . .           | 4 |
| pinn_progcens_loss . . . . .    | 5 |
| softplus . . . . .              | 5 |
| solve_ode_trajectory . . . . .  | 6 |

|              |          |
|--------------|----------|
| <b>Index</b> | <b>7</b> |
|--------------|----------|

---

|                       |                                                |
|-----------------------|------------------------------------------------|
| compute_ode_residuals | <i>Physics Residuals at Collocation Points</i> |
|-----------------------|------------------------------------------------|

---

**Description**

Physics Residuals at Collocation Points

**Usage**

compute\_ode\_residuals(sol\_colloc, z\_colloc, phi, ode\_func)

**Arguments**

- sol\_colloc      Solved ODE states across collocation grid.
- z\_colloc        Collocation evaluation grid points.
- phi             Physics differential parameters.
- ode\_func        Differential system function.

**Value**

Vector of physical model residuals.

---

|                    |                                                          |
|--------------------|----------------------------------------------------------|
| default_ode_system | <i>Default Physical ODE System (Degradation Process)</i> |
|--------------------|----------------------------------------------------------|

---

**Description**

Default Physical ODE System (Degradation Process)

**Usage**

default\_ode\_system(t, h, phi)

**Arguments**

- t                Time variable.
- h                Current degradation state.
- phi             Parameter vector governing the differential equation.

**Value**

List containing derivative dh/dt.

---

`fit_pinn_progcens_mle` *Maximum Likelihood Estimation via PINN Optimization*

---

## Description

Maximum Likelihood Estimation via PINN Optimization

## Usage

```
fit_pinn_progcens_mle(  
  times,  
  R,  
  z_colloc,  
  ode_func = default_ode_system,  
  max_epochs = 50  
)
```

## Arguments

|                         |                                                    |
|-------------------------|----------------------------------------------------|
| <code>times</code>      | Observed progressive censored failure times.       |
| <code>R</code>          | Progressive removal counts.                        |
| <code>z_colloc</code>   | Collocation points for differential equation loss. |
| <code>ode_func</code>   | Differential system function.                      |
| <code>max_epochs</code> | Maximum optimization iterations.                   |

## Value

List containing parameter estimates and standard errors.

## Examples

```
times_obs <- c(0.85, 1.42, 2.10)  
R_vec     <- c(1, 0, 1)  
colloc    <- seq(0.1, 3.0, length.out = 5)  
fit_pinn_progcens_mle(times_obs, R_vec, colloc, max_epochs = 5)
```

---

|                   |                                     |
|-------------------|-------------------------------------|
| interpolate_state | <i>Interpolate Trajectory State</i> |
|-------------------|-------------------------------------|

---

**Description**

Interpolate Trajectory State

**Usage**

```
interpolate_state(tau, sol_obs)
```

**Arguments**

|         |                                |
|---------|--------------------------------|
| tau     | Intermediary integration time. |
| sol_obs | Trajectory solution matrix.    |

**Value**

Scalar estimated state value.

---

|             |                                                |
|-------------|------------------------------------------------|
| pinn_hazard | <i>Physics-Regularized Hazard Rate Mapping</i> |
|-------------|------------------------------------------------|

---

**Description**

Physics-Regularized Hazard Rate Mapping

**Usage**

```
pinn_hazard(t, alpha, beta, h_t, psi)
```

**Arguments**

|       |                                                 |
|-------|-------------------------------------------------|
| t     | Evaluation time point.                          |
| alpha | Weibull scale parameter.                        |
| beta  | Weibull shape parameter.                        |
| h_t   | Degradation state at time t.                    |
| psi   | List containing network weights W2 and bias b2. |

**Value**

Hazard rate value.

**Examples**

```
pinn_hazard(1.0, 1.0, 1.0, 0.5, list(W2 = matrix(1,1,1), b2 = 0))
```

---

|                    |                                                                          |
|--------------------|--------------------------------------------------------------------------|
| pinn_progcens_loss | <i>Combined Structural Loss (Censored Likelihood + Physics Residual)</i> |
|--------------------|--------------------------------------------------------------------------|

---

**Description**

Combined Structural Loss (Censored Likelihood + Physics Residual)

**Usage**

```
pinn_progcens_loss(params, times, R, z_colloc, ode_func, mu = 0.1)
```

**Arguments**

|          |                                                               |
|----------|---------------------------------------------------------------|
| params   | List of current candidate parameters (alpha, beta, phi, psi). |
| times    | Vector of observed failure times.                             |
| R        | Vector of progressive censorship removals.                    |
| z_colloc | Vector of physics collocation evaluation points.              |
| ode_func | Physics ODE differential system function.                     |
| mu       | Physics loss regularization hyperparameter penalty weight.    |

**Value**

Total scalar PINN structural loss value.

---

|          |                                     |
|----------|-------------------------------------|
| softplus | <i>Softplus Activation Function</i> |
|----------|-------------------------------------|

---

**Description**

Softplus Activation Function

**Usage**

```
softplus(x)
```

**Arguments**

|   |                                 |
|---|---------------------------------|
| x | Numeric vector or matrix input. |
|---|---------------------------------|

**Value**

Smooth approximation to ReLU.

**Examples**

```
softplus(c(-1, 0, 1))
```

---

`solve_ode_trajectory`    *Forward State Trajectory Integration*

---

**Description**

Forward State Trajectory Integration

**Usage**

```
solve_ode_trajectory(times, phi, ode_func)
```

**Arguments**

|                       |                                                 |
|-----------------------|-------------------------------------------------|
| <code>times</code>    | Vector of evaluation time points.               |
| <code>phi</code>      | Differential parameters.                        |
| <code>ode_func</code> | Right-hand-side differential equation function. |

**Value**

Matrix of interpolated trajectories at evaluated times.

# Index

`compute_ode_residuals`, [2](#)

`default_ode_system`, [2](#)

`fit_pinn_progcens_mle`, [3](#)

`interpolate_state`, [4](#)

`pinn_hazard`, [4](#)

`pinn_progcens_loss`, [5](#)

`softplus`, [5](#)

`solve_ode_trajectory`, [6](#)