This documentation is mostly generated by ChatGPT.

Symars Documentation

1. Enum DType

- **Description:** The numeric type (`f32` or `f64`) used for computations in the generated Rust code.

2. class GenScalar

- **Description:** Generates Rust functions for scalar SymPy expressions.
- **Constructor:**

```
def __init__(self, dtype: DType, tol: float = 1e-9, debug: bool = False):
```

- `dtype`: A `DType` instance specifying the numeric type.
- `tol`: (Optional) Tolerance for float comparisons. Default: `1e-9`.
- `debug`: (Optional) If `True`, enables debug output. Default: `False`.
- **Public Methods:**
  - `generate_func(func_name, expr)`: Generates a Rust function for a scalar expression.
  - `generate_func_given_params(func_name, expr, params)`: Generates a Rust function for a scalar expression with specified parameter names.

3. class GenNalgebra

- **Description:** Generates Rust functions for SymPy matrices using the `nalgebra` crate.
- **Constructor:**
  - Same as `GenScalar`.
- **Public Methods:**
  - `generate(mat, func_name)`: Generates a Rust function for the matrix compatible with `nalgebra::SMatrix`.

4. class GenArrayVec

- **Description:** Generates Rust functions for array-based vector representations.

- **Constructor:**
  - Same as `GenScalar`.
- **Public Methods:**
  - `generate(mat, func_name)`: Generates Rust code to store the matrix as a flattened vector.

5. class GenFaer

- **Description:** Generates Rust functions for SymPy matrices using the `faer` crate.
- **Constructor:**
  - Same as `GenScalar`.
- **Public Methods:**
  - `generate(mat, func_name)`: Generates a Rust function for the matrix compatible with `faer::MatMut`.

6. class GenFaerVec

- **Description:** Generates Rust functions for SymPy vector with the `faer` crate.
  - Note that `faer::Col`, `faer::Row` and `faer::Mat` are different types.
- **Constructor:**
  - Same as `GenScalar`.
- **Public Methods:**
  - `generate(mat, func_name)`: Generates Rust code for SymPy vector representations.

7. class GenSparse

- **Description:** Generates Rust functions for triplet representations for sparse matrices.
- **Constructor:**
  - Same as `GenScalar`.
- **Public Methods:**
  - `params(mat)`: Returns the parameters (symbols) required by the matrix.
  - `generate(mat, func_name)`: Generates Rust functions for sparse representations.

8. class GenDense

This is not a user-faced class. Inspect to it only if you believe bug exists.

- **Description:** Generates Rust functions for dense matrices.
- **Constructor:**
  - Same as `GenScalar`.
- **Public Methods:**
  - `generate(mat, func_name)`: Generates Rust functions to represent the entries of a dense matrix.