TinyLA Reference

Types

VariableMatrix

The `VariableMatrix` class template represents a matrix with variable elements. The elements can be of any scalar type, and the matrix dimensions are specified as template parameters. The data is stored in a contiguous column-major array.

E.g.

C++ Syntax	Mathematical Notation
`auto A = tinyla::VariableMatrix<float, 2, 2, '0'>{};`	$0_{2\times 2} = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$

`auto A = tinyla::VariableMatrix<double, 2, 3, 'A'>{`
`  {1.0, 2.0, 3.0},`
`  {4.0, 5.0, 6.0}`
`};`
	$A_{2\times 3} = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix}$

`auto I = tinyla::VariableMatrix<int, 3, 3, 'I'>::identity();`	$I_{3\times 3} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$

Operators

Given A, B, v and w are `VariableMatrix` objects of compatible dimensions.

```
auto r = 2;   // Number of rows
auto c = 2;   // Number of columns
auto A = tinyla::VariableMatrix<double, r, c, 'A'>{};
auto B = tinyla::VariableMatrix<double, r, c, 'B'>{};
auto C = tinyla::VariableMatrix<std::complex<float>, r, c, 'C'>{};
auto v = tinyla::VariableMatrix<double, r, 1, 'v'>{};
auto w = tinyla::VariableMatrix<double, r, 1, 'w'>{};
```

Operation	C++ Syntax	Mathematical Notation
Assignment	`A = B;`	$A := B$

Indexing	`auto a1 = A.at(i, j);`
	`auto a2 = A[i][j];`
		$A_{i,j}$

Transposition	`auto A_trans1 = transpose(A);`
	`auto A_trans2 = T(A);`
		A^T

Conjugation	`auto C_conj1 = conjugate(C);`
	`auto C_conj2 = conj(C);`
		$\overline{C}$

Adjoint (conjugation and transposition)	`auto C_adj1 = adjoint(C);`
	`auto C_adj2 = adj(C);`
		$C^\dagger$

Operation	**C++ Syntax**	**Mathematical Notation**
Addition	`auto S = A + B;`	$A + B$
Subtraction	`auto D = A - B;`	$A - B$
Matrix-vector multiplication	`auto p = A * v;`	Av
Matrix-matrix multiplication	`auto M = A * B;`	AB
Dot product	`auto d = dot(v, w);`	$v \cdot w$