**Matrix Object Design**

Update : 13/08/2021

Toomas Vooglaid

François Jouen

Qingtian Xie

matrix is an object where matrix properties are stored in integer! fields and data is stored as vector!

matrix constructor takes following arguments:

mType: matrix type as integer [1: Char, 2: Integer, 3: Float]

mBits: bit-size as integer [8|16|32 for char! and integer!, 32|64 for float!]

mSize: matrix size as pair with COLSxROWS (e.g 3x3)

mData: matrix values as block transformed into vector for fast computation

When address is given as pair MxN it follows Red semantics of COLxROW

When address is given as two integers M an N, the order is ROW COL as conventional in math literature

TBD

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Function Name | Help | New Matrix | Modify Matrix | Routine? |
| *Internal Functions* | *Non-documented* |  |  |  |
| \_getIdx | row x col, return index |  |  |  |
| \_getRowIdx | given index of element, return number of row |  |  |  |
| \_getColIdx | given index of element, return number of column |  |  |  |
| \_getAt | get value at given row and col |  |  |  |
| \_setAt | set value at given row and col |  | yes |  |
| \_product | block product |  |  |  |
| \_swapDim | swap mat dimensions |  | yes |  |
| \_matSizeEQ? | have matrices equivalent size? |  |  |  |
| \_matTypeEQ? | have matrices equivalent type? |  |  |  |
| \_matDepthEQ? | have matrices equivalent bit-size? |  |  |  |
| \_matSimilar? | are matrices similar? |  |  |  |
| \_matOp | basic math operator for two matrices |  | yes |  |
| \_matScalarOp | scalar operator on matrix |  | yes |  |
| \_rowOp | operations on row |  |  |  |
| \_colOp | Operation on column |  |  |  |
| \_matRREF | reduced Row Eschelon Form |  | yes |  |
| \_changeData | change matrix data |  | yes |  |
| \_copy | copy matrix | yes |  |  |
| *Matrix Creation* | *Documented* |  |  |  |
| create | creates rows x columns matrix | yes |  |  |
| init | initializes and creates rows x columns matrix | yes |  |  |
| scalar | creates a scalar matrix | yes |  |  |
| identity | creates identity matrix (I) | yes |  |  |
| zero | creates zero (null) matrix | yes |  |  |
| header | return matrix properties (block) |  |  |  |
| order | return matrix size (pair) |  |  |  |
| *Matrix Properties* | *Documented* |  |  |  |
| trace | get trace of square matrix |  |  |  |
| diagonal | get matrix main diagonal |  |  |  |
| determinant | get matrix determinant |  |  |  |
| eigens | matrix eigen values |  |  |  |
| square? | square matrix (logic) |  |  |  |
| null? | is null? Any matrix |  |  |  |
| singular? | is singular? Any matrix |  |  |  |
| nonSingular? | non-singular? Any matrix |  |  |  |
| degenerate? | Is degenerate? Any matrix |  |  |  |
| nonDegenerate? | non-degenerate? Any matrix |  |  |  |
| Invertible? | is invertible? Any matrix |  |  |  |
| diagonal? | is diagonal? Square matrix |  |  |  |
| symmetric? | is symmetric? Square matrix |  |  |  |
| upper? | is upper triangular matrix? |  |  |  |
| lower? | is lower triangular matrix? |  |  |  |
| *Matrix elements access* | *Documented* |  |  |  |
| getAt | get value at COLxROW coordinate |  |  |  |
| setAt | set value at COLxROW coordinate |  | yes |  |
| *Matrix rows & columns* | *Documented* |  |  |  |
| getCol | return a new matrix column n (vector) | yes |  |  |
| getRow | return a new matrix row n (vector) | yes |  |  |
| removeRow | remove row in matrix |  | yes |  |
| removeCol | remove column in matrix |  | yes |  |
| insertRow | insert row in matrix |  | yes |  |
| appendRow | append row in matrix |  | yes |  |
| insertCol | insert column in matrix |  | yes |  |
| appendCol | append column to matrix |  | yes |  |
| switchCols | Switch cols in matrix |  | yes |  |
| switchRows | switch two rows in matrix |  | yes |  |
| augment | augment matrix with another matrix (same number of rows) |  | yes |  |
| split | split matrix |  | yes |  |
| slice | Slice matrix |  | yes |  |
| rowScalarProduct | scalar multiplication of a matrix row |  | yes |  |
| rowAdd | vector addition to a matrix row |  | yes |  |
| rowSub | vector subtraction to a matrix row |  | yes |  |
| rowProduct | vector product of a matrix row |  | yes |  |
| rowDivision | vector division of a matrix row |  | yes |  |
| rowRemainder | vector remainder of a matrix row |  | yes |  |
| rowAnd | vector AND of a matrix row |  | yes |  |
| rowOr | vector OR of a matrix row |  | yes |  |
| rowXor | vector XOR of a matrix row |  | yes |  |
| colScalarProduct |  |  |  |  |
| colAdd |  |  |  |  |
| colSub |  |  |  |  |
| colProduct |  |  |  |  |
| colDivision |  |  |  |  |
| colRemainder |  |  |  |  |
| colAnd |  |  |  |  |
| colOr |  |  |  |  |
| colXor |  |  |  |  |
| *Matrix Transform* | *Documented* |  |  |  |
| transpose | transpose matrix |  | yes |  |
| negative | negate integer of float matrices |  | yes |  |
| rotate | rotate matrix |  | yes |  |
| rotateRow | row rotation |  | yes |  |
| rotateCol | column(s) rotation |  | yes |  |
| invert | matrice inversion | yes |  |  |
| *Matrix Compute* | *Documented* |  |  |  |
| product | matrix product as float value |  |  |  |
| sigma | matrix sum as float value |  |  |  |
| mean | matrix mean as float value |  |  |  |
| mini | min value of the matrix as scalar |  |  |  |
| maxi | max value of the matrix as scalar |  |  |  |
| addition | addition of two matrices | yes |  |  |
| subtraction | subtraction of two matrices | yes |  |  |
| standardProduct | standard product of 2 compatible matrices | yes |  |  |
| HadamardProduct | Hadamard product of compatible matrices | yes |  |  |
| KroneckerProduct | Kronecker product of two matrices | yes |  |  |
| division | division of two matrices | yes |  |  |
| scalarAddition | matrix + value | yes |  |  |
| scalarSubtraction | matrix - value | yes |  |  |
| scalarProduct | product of scalar multiplication of the matrix | yes |  |  |
| scalarDivision | matrix / value | yes |  |  |
| scalarRemainder | matrix % value | yes |  |  |
| scalarAnd | matrix AND value | yes |  |  |
| scalarOr | matrix OR value | yes |  |  |
| scalarXor | matrix XOR value | yes |  |  |
| scalarRightShift | matrix right shift (>>) | yes |  |  |
| scalarRightShiftUnsigned | matrix right shift (unsigned >>>) | yes |  |  |
| scalarLeftShift | matrix left shiht (<<) | yes |  |  |
| *Matrix Decomposition* | *Documented* |  |  |  |
| getIdentity | get (left or right) identity matrix for given matrix | yes |  |  |
| LU | lower–upper (LU) matrix decomposition | yes |  |  |
| *Matrix Form* | *Documented* |  |  |  |
| show | form matrix |  |  |  |