DTU Image Viewer & Analyzer

DIVA

version 0.9

http://www.imm.dtu.dk/<tilde>diva/
diva@imm.dtu.dk

Section for Image Analysis
Informatics & Mathematical Modelling
The Technical University of Denmark

IMM

March 16, 2001

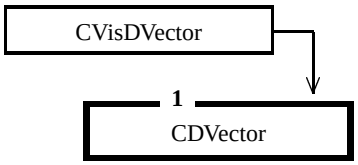
Contents

1

class **CDVector** : public virtual CVisDVector

Vector class with double precision.

Inheritance



Public Members

Vector class with double precision based upon the VisSDK vector class CVisDVector.

See Also: CDMatrix
Author: Rune Fisker, Mikkel B. Stegmann, Henrik Aans, Lars Pedersen et al.
Version: 2-5-2001

1.1

DIVAMatrixExport void **CrossProduct** (const CDVector &v1, const CDVector &v2)

Forms the cross product of two vectors.

Forms the cross product of two vectors and store the result in this. If the vector does have the correct size, it will be resized.

Return Value: `Nothing.`
Parameters: `v1` Input vector.
`v2` Input vector.
Author: Mikkel B. Stegmann
Version: 3-16-2001

1.2

DIVAMatrixExport void **ElementDivide** (const CDVector& vector)

Divide two vectors element-wise.

Divide two vectors element-wise. Corresponding MatLab operator "./".

Return Value: `Nothing.`
Parameters: `vector` The input vector to divide this vector with.

Author: Rune Fisker
Version: 1-5-1999

1.3

DIVAMatrixExport void **ElementMultiply** (const CDVector& vector)

Multiplies two vectors element-wise.

Multiplies two vectors element-wise. Corresponding MatLab operator ".*".

Return Value: `Nothing.`
Parameters: `vector` The input vector to multiply this vector with.
Author: Rune Fisker
Version: 1-5-1999

1.4

DIVAMatrixExport void **Eq** (const double B, CDVector& C) const

Equal - compares a vector and a double.

Compares of a vector and a double and returns the result as a binary vector i.e.:

C(i) = 1 if A(i) == B C(i) = 0 else

Corresponding MatLab function EQ()

Return Value: `Nothing.`
Parameters: `B` Input double.
`C` Output result vector.
Author: Lars Pedersen
Version: 7-12-2000

1.5

DIVAMatrixExport void **Eq** (const CDVector& B, CDVector& C) const

Equal - compares a vector and a vector.

Compares of a vector and a vector and returns the result as a binary matrix i.e.:

C(i) = 1 if A(i) == B C(i) = 0 else

Corresponding MatLab function EQ()

Return Value: `Nothing.`
Parameters: `B` Input vector.
`C` Output result vector.
Author: Lars Pedersen
Version: 7-12-2000

1.6

DIVAMatrixExport void **FromFile** (const CDString& sFilename)

Reads a vector from disk in binary format.

Reads a vector from disk in binary format.

Return Value: `Nothing.`
Parameters: `sFilename` Input file name.
Author: Mikkel B. Stegmann
Version: 2-5-2001

1.7

DIVAMatrixExport void **FromFile** (FILE *fh)

Reads a vector from disk in binary format.

Reads a vector from disk in binary format.

Return Value: `Nothing.`
Parameters: `fh` Open file handle (opened using fopen())
Author: Mikkel B. Stegmann
Version: 2-5-2001

1.8

DIVAMatrixExport void **FromMatlab** (const CDString& sFilename, const CDString& sName)

Reads a vector from disk in MatLab (.m) format.

Reads a vector from disk in MatLab (.m) format into 'this'.

Notice that this should be used for storage a (really) large vectors, due to the computational and i/o overhead induced by the simple MatLab text format.

Also, remember that MatLab can't read (.m) files with lines longer than 4096 bytes.

If no communication with MatLab is needed, but merely to/from disk functionality within a DIVA program, it is suggested to use the fast binary i/o methods ToFile() and FromFile().

Return Value:	`Nothing.`	
Parameters:	`sFilename`	Input file name.
	`sName`	The name to search for (and load) inside the matlab file.
Author:	Henrik Aans	
Version:	2-5-2001	

1.9

DIVAMatrixExport void **Ge** (const double B, CDVector& C) const

Greater Than or Equal - compares a double and a vector.

Compares of a double and a vector and returns the result as a binary matrix i.e.:

C(i) = 1 if A(i) >= B C(i) = 0 else

Corresponding MatLab function GE()

Return Value:	`Nothing.`	
Parameters:	`B`	Input double.
	`C`	Output result vector.
Author:	Lars Pedersen	
Version:	7-12-2000	

1.10

DIVAMatrixExport void **Ge** (const CDVector& B, CDVector& C) const

Greater Than or Equal - compares a vector and a vector.

Compares of a vector and a vector and returns the result as a binary matrix i.e.:

C(i) = 1 if A(i) >= B C(i) = 0 else

Corresponding MatLab function GE()

Return Value: `Nothing.`
Parameters:
`B` Input vector.
`C` Output result vector.
Author: Lars Pedersen
Version: 7-12-2000

1.11

DIVAMatrixExport void **Gt** (const double B, CDVector& C) const

Greater Than - compares a double and a vector.

Compares of a double and a vector and returns the result as a binary matrix i.e.:

C(i) = 1 if A(i) > B C(i) = 0 else

Corresponding MatLab function GT()

Return Value: `Nothing.`
Parameters:
`B` Input vector.
`C` Output result vector.
Author: Lars Pedersen
Version: 7-12-2000

1.12

DIVAMatrixExport void **Gt** (const CDVector& B, CDVector& C) const

Greater Than - compares a vector and a vector.

Compares of a vector and a vector and returns the result as a binary matrix i.e.:

C(i) = 1 if A(i) > B C(i) = 0 else

Corresponding MatLab function GT()

Return Value: `Nothing.`
Parameters: `B` Input vector.
`C` Output result vector.
Author: Lars Pedersen
Version: 7-12-2000

1.13

DIVAMatrixExport void **Le** (const double B, CDVector& C) const

Less Than or Equal - compares a double and a vector.

Compares of a double and a vector and returns the result as a binary matrix i.e.:

C(i) = 1 if A(i) <= B C(i) = 0 else

Corresponding MatLab function LE()

Return Value: `Nothing.`
Parameters: `B` Input double.
`C` Output result vector.
Author: Lars Pedersen
Version: 7-12-2000

1.14

DIVAMatrixExport void **Le** (const CDVector& B, CDVector& C) const

Less Than or Equal - compares a vector and a vector.

Compares of a vector and a vector and returns the result as a binary matrix i.e.:

C(i) = 1 if A(i) <= B C(i) = 0 else

Corresponding MatLab function LE()

Return Value: `Nothing.`
Parameters: `B` Input vector.
`C` Output result vector.
Author: Lars Pedersen

Version: 7-12-2000

1.15

DIVAMatrixExport void **Log** ()

Takes the natural logarithm of each element.

Takes the natural logarithm of each element.

Return Value: `Nothing.`
Author: Lars Pedersen
Version: 7-12-2000

1.16

DIVAMatrixExport void **Lt** (const double B, CDVector& C) const

Less Than - compares a double and a vector.

Compares of a double and a vector and returns the result as a binary matrix i.e.:

C(i) = 1 if A(i) < B C(i) = 0 else

Corresponding MatLab function LT()

Return Value: `Nothing.`
Parameters: `B` Input double.
`C` Output result vector.
Author: Lars Pedersen
Version: 7-12-2000

1.17

DIVAMatrixExport void **Lt** (const CDVector& B, CDVector& C) const

Less Than - compares a vector and a vector.

Compares of a vector and a vector and returns the result as a binary matrix i.e.:

C(i) = 1 if A(i) < B C(i) = 0 else

Corresponding MatLab function LT()

Return Value: `Nothing.`
Parameters: `B` Input vector.
`C` Output result vector.
Author: Lars Pedersen
Version: 7-12-2000

1.18

DIVAMatrixExport double **Max** () const

Finds the maximum element in the vector.

Finds the maximum element in the vector.

Return Value: `The` maximum value of the elements.
Author: Rune Fisker
Version: 3-5-1999

1.19

DIVAMatrixExport double **Mean** () const

Calculates the mean value of the vector.

Calculates the mean value of the vector.

Return Value: `The` mean value.
Author: Rune Fisker
Version: 3-5-1999

1.20

DIVAMatrixExport double **Min** () const

Finds the minimum element in the vector.

Finds the minimum element in the vector.

Return Value: `The` minimum value of the elements.
Author: Rune Fisker
Version: 3-5-1999

1.21

DIVAMatrixExport double **Min** (int& iPos) const

Finds the minimum element in the vector and its position.

Finds the minimum element in the vector and its position.

Return Value: `The` minimum value of the elements and its position.
Author: Rune Fisker
Version: 1-9-1999

1.22

DIVAMatrixExport void **Ne** (const double B, CDVector& C) const

Not Equal - compares a double and a vector.

Compares of a double and a vector and returns the result as a binary matrix i.e.:

C(i) = 1 if A(i) != B C(i) = 0 else

Corresponding MatLab function NE()

Return Value: `Nothing.`
Parameters: `B` Input double.
`C` Output result vector.
Author: Lars Pedersen
Version: 7-12-2000

1.23

DIVAMatrixExport void **Ne** (const CDVector& B, CDVector& C) const

Not Equal - compares a vector and a vector.

Compares of a vector and a vector and returns the result as a binary matrix i.e.:

C(i) = 1 if A(i) != B C(i) = 0 else

Corresponding MatLab function NE()

Return Value: `Nothing.`
Parameters: `B` Input vector.
`C` Output result vector.
Author: Lars Pedersen
Version: 7-12-2000

1.24

DIVAMatrixExport double **Norm1** () const

Calculates the one-norm (L1) of the vector.

Calculates the one-norm (L1) of the vector. Also known as the city block metric.

L1(v) = —x_1— + —x_2— .. + —x_n—

Return Value: `The` L1 norm.
Author: Mikkel B. Stegmann
Version: 2-5-2001

1.25

DIVAMatrixExport double **Norm2** () const

Calculates the two-norm (L2) of the vector.

Calculates the two-norm (L2) of the vector. Also known as the Euclidean length.

L2(v) = sqrt(x_1^2 + x_2^2 .. + x_n^2)

Return Value: `The` L2 norm.
Author: Mikkel B. Stegmann
Version: 2-5-2001

1.26

DIVAMatrixExport double **NormInf** () const

Calculates the infinity-norm (Linf) of the vector.

Calculates the infinity-norm (Linf) of the vector. Also known as the Chebyshev Norm.

Linf(v) = max(—x_1—, —x_2— ... , —x_n—)

Return Value: `The` L2 norm.
Author: Mikkel B. Stegmann
Version: 2-5-2001

1.27

DIVAMatrixExport void **Normalize2** ()

Normalizes the vector to unit length, using the 2-norm.

Normalizes the vector to unit length, using the 2-norm.

Return Value: `Nothing.`
Author: Mikkel B. Stegmann
Version: 2-5-2001

1.28

DIVAMatrixExport void **Pow** (double dP)

Takes the power 'dP' of each element.

Takes the power 'dP' of each element.

Return Value: `Nothing.`
Parameters: `dP` The exponent.
Author: Rune Fisker
Version: 1-5-1999

1.29

DIVAMatrixExport void **Rand** ()

Uniformly distributed random numbers.

Inserts uniformly distributed random numbers in the range [0;1].

Return Value: `Nothing.`
Author: Mikkel B. Stegmann
Version: 3-9-2001

1.30

DIVAMatrixExport void **Reverse** ()

Reverses the vector.

Reverses the vector.

Return Value: `Nothing.`
Author: Mikkel B. Stegmann
Version: 3-5-2000

1.31

DIVAMatrixExport double **Skewness** () const

Calculates the skewness of the vector.

Calculates the skewness the vector.

Return Value: `The` skewness.
Author: Rune Fisker
Version: 3-5-1999

1.32

DIVAMatrixExport void **Sort** (bool ascending)

Sorts the vector.

Sorts the vector in either ascending (default) or descending order. Implemented using the standard quicksort, qsort().

Return Value: `Nothing.`
Parameters: `ascending` If true (default) the vector is sorted in ascendingorder. If false its sorted in descending order.
Author: Mikkel B. Stegmann
Version: 2-2-2001

1.33

DIVAMatrixExport void **Sqr** ()

Squares each element.

Squares each element.

Return Value: `Nothing.`
Author: Rune Fisker
Version: 1-5-1999

1.34

DIVAMatrixExport void **Sqrt** ()

Takes the square root of each element.

Takes the square root of each element.

Return Value: `Nothing.`
Author: Rune Fisker
Version: 1-5-1999

1.35

DIVAMatrixExport double **Std** () const

Calculates the standard deviation of the vector.

Calculates the standard deviation the vector.

Return Value: `The` skewness.
Author: Rune Fisker
Version: 3-5-1999

1.36

DIVAMatrixExport double **Sum** () const

Calculates the sum of the vector.

Calculates the sum of the vector.

Return Value: `The` sum.
Author: Rune Fisker
Version: 3-5-1999

1.37

DIVAMatrixExport void **ToFile** (const CDString& sFilename) const

Writes the vector to disk in binary format.

Writes the vector to disk in binary format. Use 'readbin.m' to load such a file into MatLab (placed in the diva/matlab dir).

Return Value: `Nothing.`
Parameters: `sFilename` Output file name.
Author: Mikkel B. Stegmann
Version: 2-5-2001

1.38

DIVAMatrixExport void **ToFile** (FILE *fh) const

Writes the vector to disk in binary format.

Writes the vector to disk in binary format. Use 'readbin.m' to load such a file into MatLab (placed in the diva/matlab dir).

Return Value: `Nothing.`
Parameters: `fh` Open file handle (opened using fopen())
Author: Mikkel B. Stegmann
Version: 2-5-2001

1.39

DIVAMatrixExport void **ToMatlab** (const CDString& sFilename, const CDString& sName, const CDString& sComment, bool fAppend) const

Writes the vector to disk in MatLab (.m) format.

Writes the vector to disk in MatLab (.m) format. To read the vector into MatLab write e.g. 'my_vector.m' at the MatLab prompt.

Notice that this should be used for storage a (really) large vectors, due to the computational and i/o overhead induced by the simple MatLab text format.

Also, remember that MatLab can't read (.m) files with lines longer than 4096 bytes.

If no communication with MatLab is needed, but merely to/from disk functionality within a DIVA program, it is suggested to use the fast binary i/o methods ToFile() and FromFile().

Return Value: `Nothing.`
Parameters:
`sFilename` Output file name. Should have the extension '.m'.
`sName` Name of destination matlab variable.
`sComment` Optional comment inside the file.
`fAppend` If true, the vector is appended to the file 'sFilename'.
Author: Henrik Aans

Version: 2-5-2001

1.40

DIVAMatrixExport CDString **ToString** (const bool fNewline) const

Returns a string representing the vector.

Returns a string representing the vector.

Return Value: `The` output string.
Parameters: `fNewline` If true, the string is terminated with a new line.
Author: Rune Fisker
Version: 1-5-1999

1.41

DIVAMatrixExport double **Var** () const

Calculates the variance of the vector.

Calculates the variance the vector.

Return Value: `The` variance.
Author: Rune Fisker
Version: 3-5-1999

1.42

DIVAMatrixExport CDVector& **operator=** (const CVisDVector &vIn)

Assignment operator.

Assignment operator.

Return Value: `This.`
Parameters: `vIn` Input vector.
Author: Rune Fisker
Version: 2-5-2001

1.43

DIVAMatrixExport CDVector& **operator=** (const CDVector &vIn)

Assignment operator.

Assignment operator.

Return Value: `This.`
Parameters: `vIn` Input vector.
Author: Rune Fisker
Version: 2-5-2001

1.44

DIVAMatrixExport CDVector& **operator=** (double value)

Assignment operator.

Assignment operator. Sets all element to the input value.

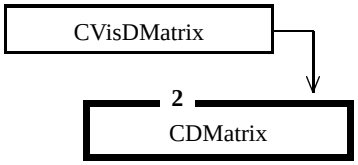
Return Value: `This.`
Parameters: `vIn` Input double.
Author: Rune Fisker
Version: 2-5-2001

2

class **CDMatrix** : public virtual CVisDMatrix

Matrix class with double precision.

Inheritance



Public Members

Matrix class with double precision based upon the VisSDK matrix class CVisDMatrix.

See Also: CDVector
Author: Rune Fisker, Mikkel B. Stegmann, Henrik Aans, Lars Pedersen et al.

Version: 2-5-2001

2.1

DIVAMatrixExport void **Col** (int i, CDVector& vCol) const

Return the i-th column of the matrix.

Return the i-th column of the matrix. The vector will be resized if it doesn't have the right length.

Alternatively, one could use the more costly CVisDVector CVisDMatrix::Row(int r) method.

To set the i-th col use: void CVisDMatrix::SetColumn(int c, const CVisDVector &v)

Return Value: `Nothing.`
Parameters:
`i` The column number.
`vCol` Output vector;
Author: Rune Fisker
Version: 3-14-2001

2.2

DIVAMatrixExport void **CombVert** (CVisDMatrix &Top, CVisDMatrix &Bottom)

Composes the matrix of Top and Bottom (on top of each other).

Composes the matrix of Top and Bottom (on top of each other).

Return Value: `Nothing.`
Parameters:
`Top` Matrix to be placed in the top of this matrix.
`Bottom` Matrix to be placed in the bottom of this matrix.
Author: Rune Fisker
Version: 3-14-2001

2.3

DIVAMatrixExport void **Diag** (const CDVector &vec)

Make the matrix a diagonal matrix containing 'vec'.

This function transforms the matrix into a diagonal matrix, with the values of 'vec' in the diagonal.

Return Value: `Nothing.`
Parameters: `vec` Vector to place in the diagonal.
Author: Rune Fisker
Version: 3-14-2001

2.4

DIVAMatrixExport void **ElementDivide** (const CDMatrix& matrix)

Element-wise matrix division.

Divide each element in the matrix with the corresponding element of the input matrix.

Return Value: `Nothing.`
Author: Rune Fisker
Version: 5-1-1999

2.5

DIVAMatrixExport void **ElementMultiply** (const CDMatrix& matrix)

Element-wise matrix multiply.

Multiply each element in the matrix with the corresponding element of the input matrix.

Return Value: `Nothing.`
Author: Rune Fisker
Version: 5-1-1999

2.6

DIVAMatrixExport void **Eq** (const double B, CDMatrix& C) const

Element wise equal to. Comparison of a matrix and a double.

Compares each element in this matrix (A) with double B. Result matrix C has elements with values 1 or 0.

C(i,j) = 1 if A(i,j) == B C(i,j) = 0 else

Return Value: `Nothing.`
Author: Lars Pedersen
Version: 6-12-2000

2.7

DIVAMatrixExport void **Eq** (const CDMatrix& B, CDMatrix& C) const

Element wise equal to. Comparison of two matrices.

Compares each element in this matrix (A) with corresponding element in B. Result matrix C has elements with values 1 or 0.

C(i,j) = 1 if A(i,j) == B(i,j) C(i,j) = 0 else

Return Value: `Nothing.`
Author: Lars Pedersen
Version: 7-12-2000

2.8

DIVAMatrixExport void **Eye** ()

Converts the matrix in the identity matrix.

Converts the matrix in the identity matrix - i.e. zeros all-over except the ones in the diagonal.

Return Value: `Nothing.`
Author: Rune Fisker
Version: 5-1-1999

2.9

DIVAMatrixExport void **FlipLR** ()

Flip matrix in left/right direction.

FlipLR modifies the matrix with row preserved and columns flipped in the left/right direction.

Return Value: `Nothing.`
See Also: FlipUD
Author: Mikkel B. Stegmann
Version: 3-9-2001

2.10

DIVAMatrixExport void **FlipUD** ()

Flip matrix in up/down direction.

FlipUD(X) modifies the matrix with columns preserved and rows flipped in the up/down direction.

Return Value: `Nothing.`
See Also: FlipLR
Author: Mikkel B. Stegmann
Version: 3-9-2001

2.11

DIVAMatrixExport void **FromFile** (const CDString & sFilename)

Reads a matrix from disk in binary format.

Reads a matrix from disk in binary format, as written from CDMatrix::ToFile().

The matrix is resizew if it doesn't fit the disk matrix.

Return Value: `Nothing.`
Parameters: `sFilename` Input file name.
See Also: ToFile
Author: Mikkel B. Stegmann
Version: 3-14-2000

2.12

DIVAMatrixExport void **FromFile** (FILE *fh)

Reads a matrix from disk in binary format.

Reads a matrix from disk in binary format (from the current position of the file pointer), as written from CDMatrix::ToFile().

The matrix is resizew if it doesn't fit the disk matrix.

Return Value: `Nothing.`
Parameters: `fh` Open file handle.
See Also: ToFile
Author: Mikkel B. Stegmann
Version: 3-14-2000

2.13

DIVAMatrixExport void **FromMatlab** (const CDString& sFilename, const CDString& sName)

Reads a matrix from disk in MatLab (.m) format.

Reads a matrix from disk in MatLab (.m) format into 'this'.

Notice that this should be used for storage a (really) large matrices, due to the computational and i/o overhead induced by the simple MatLab text format.

Also, remember that MatLab can't read (.m) files with lines longer than 4096 bytes.

If no communication with MatLab is needed, but merely to/from disk functionality within a DIVA program, it is suggested to use the fast binary i/o methods ToFile() and FromFile().

Return Value: `Nothing.`
Parameters: `sFilename` Input file name.
`sName` The name to search for (and load) inside the matlab file.
Author: Henrik Aans
Version: 14-7-1999

2.14

DIVAMatrixExport void **Ge** (const double B, CDMatrix& C) const

Element wise greater than or equal. Comparison of a matrix and a double.

Compares each element in this matrix (A) with double B. Result matrix C has elements with values 1 or 0.

C(i,j) = 1 if A(i,j) >= B C(i,j) = 0 else

Return Value: `Nothing.`
Author: Lars Pedersen
Version: 7-12-2000

2.15

DIVAMatrixExport void **Ge** (const CDMatrix& B, CDMatrix& C) const

Element wise greater than or equal. Comparison of two matrices.

Compares each element in this matrix (A) with corresponding element in B. Result matrix C has elements with values 1 or 0.

C(i,j) = 1 if A(i,j) <= B(i,j) C(i,j) = 0 else

Return Value: `Nothing.`
Author: Lars Pedersen
Version: 7-12-2000

2.16

DIVAMatrixExport void **Gt** (const double B, CDMatrix& C) const

Element wise greater than. Comparison of a matrix and a double.

Compares each element in this matrix (A) with double B. Result matrix C has elements with values 1 or 0.

C(i,j) = 1 if A(i,j) > B C(i,j) = 0 else

Return Value: `Nothing.`
Author: Lars Pedersen
Version: 7-12-2000

2.17

DIVAMatrixExport void **Gt** (const CDMatrix& B, CDMatrix& C) const

Element wise greater than. Comparison of two matrices.

Compares each element in this matrix (A) with corresponding element in B. Result matrix C has elements with values 1 or 0.

C(i,j) = 1 if A(i,j) > B(i,j) C(i,j) = 0 else

Return Value: `Nothing.`
Author: Lars Pedersen
Version: 7-12-2000

2.18

DIVAMatrixExport void **Kron** (CDMatrix &mX, CDMatrix &mY)

Forms the Kronecker tensor product of two matrices.

Forms the Kronecker tensor product of two matrices. The result is placed in this.

Return Value: `Nothing.`
Author: Rune Fisker
Version: 5-1-1999

2.19

DIVAMatrixExport void **Le** (const double B, CDMatrix& C) const

Element wise less than or equal. Comparison of a matrix and a double.

Compares each element in this matrix (A) with double B. Result matrix C has elements with values 1 or 0.

C(i,j) = 1 if A(i,j) <= B C(i,j) = 0 else

Return Value: `Nothing.`
Author: Lars Pedersen
Version: 7-12-2000

2.20

DIVAMatrixExport void **Le** (const CDMatrix& B, CDMatrix& C) const

Element wise less than or equal. Comparison of two matrices.

Compares each element in this matrix (A) with corresponding element in B. Result matrix C has elements with values 1 or 0.

C(i,j) = 1 if A(i,j) <= B(i,j) C(i,j) = 0 else

Return Value: `Nothing.`
Author: Lars Pedersen
Version: 7-12-2000

2.21

DIVAMatrixExport void **Log** ()

Takes the natural logarithm (base e=2.71..) of each matrix element.

Takes the natural logarithm (base e=2.71..) of each matrix element.

Return Value: `Nothing.`
Author: Lars Pedersen
Version: 7-12-2000

2.22

DIVAMatrixExport void **Lt** (const double B, CDMatrix& C) const

Element wise less than. Comparison of a matrix and a double.

Compares each element in this matrix (A) with double B. Result matrix C has elements with values 1 or 0.

C(i,j) = 1 if A(i,j) < B C(i,j) = 0 else

Return Value: `Nothing.`
Author: Lars Pedersen
Version: 7-12-2000

2.23

DIVAMatrixExport void **Lt** (const CDMatrix& B, CDMatrix& C) const

Element wise less than. Comparison of two matrices.

Compares each element in this matrix (A) with corresponding element in B. Result matrix C has elements with values 1 or 0.

C(i,j) = 1 if A(i,j) < B(i,j) C(i,j) = 0 else

Return Value: `Nothing.`
Author: Lars Pedersen
Version: 7-12-2000

2.24

DIVAMatrixExport double **Mean** () const

Calcs the mean value of the matrix.

Calcs the mean value of the matrix.

Return Value: `The` mean value.
Author: Rune Fisker
Version: 3-14-2001

2.25

DIVAMatrixExport void **MeanCol** (CDVector& vMean) const

Calc the mean of each column into a vector.

Calc the mean of each column into a vector. The vector will be resized if it doesn't have the right length.

Return Value: `Nothing.`
Parameters: `vMean` Output vector.
Author: Rune Fisker
Version: 3-14-2001

2.26

DIVAMatrixExport void **Ne** (const double B, CDMatrix& C) const

Element wise not equal to. Comparison of a matrix and a double.

Compares each element in this matrix (A) with double B. Result matrix C has elements with values 1 or 0.

C(i,j) = 1 if A(i,j) != B C(i,j) = 0 else

Return Value: `Nothing.`
Author: Lars Pedersen
Version: 7-12-2000

2.27

DIVAMatrixExport void **Ne** (const CDMatrix& B, CDMatrix& C) const

Element wise not equal to. Comparison of two matrices.

Compares each element in this matrix (A) with corresponding element in B. Result matrix C has elements with values 1 or 0.

C(i,j) = 1 if A(i,j) != B(i,j) C(i,j) = 0 else

Return Value: `Nothing.`
Author: Lars Pedersen

Version: 7-12-2000

2.28

DIVAMatrixExport void **OneWayANOVA** (double& dZ, int& nDFModel, int& nDFError) const

One-way analysis of variance (ANOVA).

Determination of the fluctuations observed in a sample, and their dependencies in the form of a one-way analysis of variance (ANOVA).

Return Value: `Nothing.`
Parameters: `dZ`
`nDFModel`
`nDFError`
Author: Rune Fisker
Version: 3-14-2001

2.29

DIVAMatrixExport void **OuterProduct** (const CDVector &v1, const CDVector &v2)

Forms the outer product of two vectors.

Forms the outer product of two vectors and store the result in this. If the matrix does have the correct size, it will be resized.

Return Value: `Nothing.`
Parameters: `v1` Input vector.
`v2` Input vector.
Author: Mikkel B. Stegmann
Version: 3-16-2001

2.30

DIVAMatrixExport void **Rand** ()

Uniformly distributed random numbers.

Inserts uniformly distributed random numbers in the range [0;1].

Return Value: `Nothing.`
Author: Mikkel B. Stegmann
Version: 3-9-2001

2.31

DIVAMatrixExport void **Row** (int i, CDVector& vRow) const

Return the i-th row of the matrix.

Return the i-th row of the matrix. The vector will be resized if it doesn't have the right length.

Notice that due to the row major nature of matrices, row read/writes are *much* faster than col read/writes.

Alternatively, one could use the more costly CVisDVector CVisDMatrix::Row(int r) method.

To set the i-th row use: void CVisDMatrix::SetRow(int r, const CVisDVector &v)

Return Value: `Nothing.`
Parameters:
`i` The row number.
`vRow` Output vector;
Author: Mikkel B. Stegmann
Version: 3-14-2001

2.32

DIVAMatrixExport void **Sqr** ()

Takes the power of two of each element.

Takes the power of two of each element.

Return Value: `Nothing.`
Author: Rune Fisker
Version: 5-1-1999

2.33

DIVAMatrixExport void **Sqrt** ()

Takes the square root of each element.

Takes the square root of each element.

Return Value: `Nothing.`
Author: Rune Fisker
Version: 5-1-1999

2.34

DIVAMatrixExport double **Std** () const

Calc the standard diviation of the matrix.

Calc the standard diviation of the matrix.

Return Value: `The` standard diviation value.
Author: Rune Fisker
Version: 3-14-2001

2.35

DIVAMatrixExport void **StdCol** (CDVector& vStd) const

Calc the standard deviation of each column into a vector.

Calc the standard deviation of each column into a vector. The vector will be resized if it doesn't have the right length.

Return Value: `Nothing.`
Parameters: `vStd` Output vector.
Author: Rune Fisker
Version: 3-14-2001

2.36

DIVAMatrixExport double **Sum** () const

Calc the total sum of the matrix.

Calc the total sum of the matrix.

Return Value: `The` sum.
Author: Rune Fisker
Version: 3-14-2001

2.37

DIVAMatrixExport void **SumCol** (CDVector& vSum) const

Calc the sum of each column into a vector.

Calc the sum of each column into a vector. The vector will be resized if it doesn't have the right length.

Return Value: `Nothing.`
Parameters: `vSum` Output vector.
Author: Rune Fisker
Version: 3-14-2001

2.38

DIVAMatrixExport void **TTest** (const int iCol1, const int iCol2, double& dZ, int& nDF) const

Student's T-test.

Student's T-test.

Return Value: `Nothing.`
Parameters: `iCol1`
`iCol2`
`dZ`
`nDF`
Author: Rune Fisker
Version: 3-14-2001

2.39

DIVAMatrixExport void **ToFile** (const CDString & sFilename) const

Writes the matrix to disk in binary format.

Writes the matrix to disk in binary format. The dimensions are saved as two doubles (!!?) (rows,cols) in the start.

Modified: Mikkel B. Stegmann 14/3-00

Return Value: `Nothing.`
Parameters: `sFilename` Input file name.
See Also: FromFile
Author: Henrik Aans
Version: 7-22-1999

2.40

DIVAMatrixExport void **ToFile** (FILE *fh) const

Writes the matrix to disk in binary format.

Writes the matrix to disk in binary format. The dimensions are saved as two doubles (!!?) (rows,cols) in the start. The matrix is written to the binary file 'fh' at the current position of the file pointer.

Modified: Mikkel B. Stegmann 14/3-00

Return Value: `Nothing.`
Parameters: `fh` Open file handle.
See Also: FromFile
Author: Mikkel B. Stegmann

Version: 4-14-1999

2.41

DIVAMatrixExport void **ToLaTeX** (const CDString& sFilename, const CDString& sVarName, const CDString& sComment, bool fWriteAsTable, const CDString& sNumFmt, const CDString& sBracketType, bool fAppend) const

Writes the matrix in LaTeX format. Throws: CVisError.

Writes the matrix in LaTeX format. Throws: CVisError.

Return Value:	`Nothing.`	
Parameters:	`sFilename`	Destination file name.
	`sVarName`	Name of matrix.
	`sComment`	Comment in the LaTeX file.
	`fWriteAsTable`	If true the matrix is formatted as a table.
	`sNumFmt`	printf c-stype format of numbers, e.g. "%.2f".
	`sBracketType`	Either "[" or "(".
	`fAppend`	If true 'sFilename' is append with this matrix.
Author:	Mikkel B. Stegmann	
Version:	4-5-2000	

2.42

DIVAMatrixExport void **ToMatlab** (const CDString& sFilename, const CDString& sName, const CDString& sComment, bool fAppend) const

Writes the matrix to disk in MatLab (.m) format.

Writes the matrix to disk in MatLab (.m) format. To read the matrix into MatLab write e.g. 'my_matrix.m' at the MatLab prompt.

Notice that this should be used for storage a (really) large matrices, due to the computational and i/o overhead induced by the simple MatLab text format.

Also, remember that MatLab can't read (.m) files with lines longer than 4096 bytes.

If no communication with MatLab is needed, but merely to/from disk functionality within a DIVA program, it is suggested to use the fast binary i/o methods ToFile() and FromFile().

Return Value:	`Nothing.`	
Parameters:	`sFilename`	Output file name. Should have the extension '.m'.
	`sName`	Name of destination matlab variable.
	`sComment`	Optional comment inside the file.
	`fAppend`	If true, the vector is appended to the file 'sFilename'.
Author:	Henrik Aans	
Version:	14-7-1999	

2.43

DIVAMatrixExport CDString **ToString** () const

Returns a string representing the matrix.

Returns a string representing the matrix.

Return Value:	`A` string.
Author:	Rune Fisker
Version:	3-14-2001

2.44

DIVAMatrixExport void **TriL** (CDMatrix &matrix, const int K) const

Extract lower triangular part of matrix.

Extracts the elements on and below the K-th diagonal. K = 0 is the main diagonal, K > 0 is above the main diagonal and K < 0 is below the main diagonal.

Return Value:	`Nothing.`
Author:	Lars Pedersen
Version:	6-12-2000

2.45

DIVAMatrixExport void **TriU** (CDMatrix &matrix, const int K) const

Extract upper triangular part of matrix.

Extracts the elements on and above the K-th diagonal. K = 0 is the main diagonal, K > 0 is above the main diagonal and K < 0 is below the main diagonal.

Return Value: `Nothing.`
Author: Lars Pedersen
Version: 5-12-2000

2.46

DIVAMatrixExport double **Var** () const

Calc the variance of the matrix.

Calc the variance of the matrix.

Return Value: `The` variance value.
Author: Rune Fisker
Version: 3-14-2001

2.47

DIVAMatrixExport void **VarCol** (CDVector& vVar) const

Calc the variance of each column into a vector.

Calc the variance of each column into a vector. The vector will be resized if it doesn't have the right length.

Return Value: `Nothing.`
Parameters: `vVar` Output vector.
Author: Rune Fisker
Version: 3-14-2001

2.48

DIVAMatrix.h DIVAMatrixExport CDMatrix& **operator=** (const CVisDMatrix &mat)

Assignment operator.

Assignment operator (CVisDMatrix).

Return Value: `This.`
Parameters: `mat` Input matrix.
Author: Rune Fisker
Version: 3-14-2001

2.49

DIVAMatrixExport CDMatrix& **operator=** (const CDMatrix &mat)

Assignment operator.

Assignment operator (CDMatrix).

Return Value: `This.`
Parameters: `mat` Input matrix.
Author: Rune Fisker
Version: 3-14-2001

2.50

DIVAMatrixExport CDMatrix& **operator=** (double value)

Assignment operator.

Assignment operator (double). Set all values equal to 'value'.

Return Value: `This.`
Parameters: `value` Input double.
Author: Rune Fisker
Version: 3-14-2001

Class Graph