Interfaces of transformation matrix

Transformation matrix is used to describe lows of changes for multidimensional distributions. Each cell of matrix describes how much of material will be transferred from one class of multidimensional distribution to another.

Constructors
CTransformMatrix
Functions to work with dimensions
SetDimensions GetDimensions GetClasses GetDimensionsNumber
Functions to get data
GetValue GetVectorValue
Functions to set data
SetValue SetVectorValue
Other functions
Normalize ClearData Clear

Constructors

CTransformMatrix *(void)*

Basic constructor. Creates an empty matrix.

CTransformMatrix *(unsigned _nType, unsigned _nClasses)*

Creates matrix to transform one-dimensional distribution with type *_nType* and *_nClasses* classes. *_nType* is one of the predefined types of distributions (DISTR_SIZE, DISTR_FORM_FACTOR, etc. Refer to the file 'Defines.pdf'). All values in matrix will be set to 0.

CTransformMatrix *(unsigned _nType1, unsigned _nClasses1, unsigned _nType2, unsigned _nClasses2)*

Creates matrix to transform two-dimensional distribution with types *_nType1* and *_nType2* and classes *_nClasses1* and *_nClasses2*. *_nType1* and *_nType2* are from predefined types of distributions (DISTR_SIZE, DISTR_FORM_FACTOR, etc. Refer to the file 'Defines.pdf'). All values in matrix will be set to 0.

CTransformMatrix *(const std::vector<unsigned> &_vTypes, const std::vector<unsigned> &_vClasses)*

Creates transformation matrix for distribution with specified types and classes. *_vTypes* and *_vClasses* must have the same length. *_vTypes* is the vector of predefined types of distributions (DISTR_SIZE, DISTR_FORM_FACTOR, etc. Refer to the file 'Defines.pdf'). All values in matrix will be set to 0.

Functions to work with dimensions

bool SetDimensions *(unsigned _nType, unsigned _nClasses)*

Sets new dimensions set to the matrix in order to transform one-dimensional distribution with type *_nType* and *_nClasses* classes. *_nType* is one of the predefined types of distributions (DISTR_SIZE, DISTR_FORM_FACTOR, etc. Refer to the file 'Defines.pdf'). Old data will be erased and matrix will be initialized with zeroes. Returns *false* on error.

bool SetDimensions *(unsigned _nType1, unsigned _nClasses1, unsigned _nType2, unsigned _nClasses2)*

Sets new dimensions set to the matrix in order to transform two-dimensional distribution. *_nType1* and *_nType2* are one of the predefined types of distributions (DISTR_SIZE, DISTR_FORM_FACTOR, etc. Refer to the file 'Defines.pdf'). Types must be unique. *_nClasses1* and *_nClasses2* are number of classes in corresponding distributions. Old data will be erased and matrix will be initialized with zeroes. Returns *false* on error.

bool SetDimensions *(unsigned _nType1, unsigned _nClasses1, unsigned _nType2, unsigned _nClasses2, unsigned _nType3, unsigned _nClasses3)*

Sets new dimensions set to the matrix in order to transform three-dimensional distribution. *_nType1*, *_nType2* and *_nType3* are one of the predefined types of distributions (DISTR_SIZE, DISTR_FORM_FACTOR, etc. Refer to the file 'Defines.pdf'). Types must be unique. *_nClasses1*, *_nClasses2* and *_nClasses3* are number of classes in corresponding distributions. Old data will be erased and matrix will be initialized with zeroes. Returns *false* on error.

bool SetDimensions *(const std::vector<unsigned> &_vTypes, const std::vector<unsigned> &_vClasses)*

Sets new dimensions set with types *_vTypes* and numbers of classes *_vClasses*. *_vTypes* is the vector of predefined types of distributions (DISTR_SIZE, DISTR_FORM_FACTOR, etc. Refer to the file 'Defines.pdf'). All old data will be erased and matrix will be initialized with zeroes. Sizes of vectors *_vTypes* and *_vClasses* must be equal. Returns *false* on error.

std::vector<unsigned> GetDimensions *()*

Returns vector with all current defined dimensions types.

std::vector<unsigned> GetClasses *()*

Returns vector with current numbers of classes.

unsigned GetDimensionsNumber *()*

Returns current number of dimensions.

Functions to get data

double GetValue *(unsigned _nCoordSrc, unsigned _nCoordDst)*

Returns value by specified coordinates according to all defined dimensions in transformation matrix for one-dimensional distribution. *_nCoordSrc* is coordinate of a source class, *_nCoordDst* is coordinate of a destination class. Returning value is a mass fraction, which will be transferred from the source class to the destination class. Works with one-dimensional distribution only. Returns *-1* on error.

double GetValue *(unsigned _nCoordSrc1, unsigned _nCoordSrc2, unsigned _nCoordDst1, unsigned _nCoordDst2)*

Returns value by specified coordinates according to all defined dimensions in transformation matrix for two-dimensional distribution. *_nCoordSrc1* and *_nCoordSrc2* are coordinates of a source class, *_nCoordDst1* and *_nCoordDst2* are coordinate of a destination class. Returning value is a mass

fraction, which will be transferred from the source class to the destination class. Works with two-dimensional distribution only. Returns *-1* on error.

double GetValue *(const std::vector<unsigned>& _vCoordsSrc,*
const std::vector<unsigned>& _vCoordsDst)
Returns value by specified coordinates according to all defined dimensions. *_vCoordsSrc* are coordinates of a source class, *_vCoordsDst* are coordinates of a destination class. Sizes of vectors *_vCoordsSrc* and *_vCoordsDst* must be equal and must correspond to the number of currently defined dimensions. Returning value is a mass fraction, which will be transferred from the source class to the destination class. Returns *-1* on error.

double GetValue *(const std::vector<unsigned>& _vDimsSrc,*
const std::vector<unsigned>& _vCoordsSrc, const std::vector<unsigned>& _vDimsDst,
const std::vector<unsigned>& _vCoordsDst)
Returns value according to specified coordinates and dimensions. Number of dimensions must be the same as defined in the transformation matrix, but their sequence can be different. Sizes of all vectors must be equal. Returning value is a mass fraction, which will be transferred from the source class to the destination class. Returns *-1* on error.

bool GetVectorValue *(const std::vector<unsigned>& _vCoordsSrc,*
const std::vector<unsigned>& _vCoordsDst, std::vector<double>& _vResult)
Returns vector value by specified coordinates according to all defined dimensions. Size of one vector of coordinates must be equal to the number of dimensions in transformation matrix; size of the second one must be one less. Returning value *_vResult* is a vector of mass fractions, which will be transferred from the source to the destination. Returns *false* on error.

bool GetVectorValue *(const std::vector<unsigned>& _vDimsSrc,*
const std::vector<unsigned>& _vCoordsSrc, const std::vector<unsigned>& _vDimsDst,
const std::vector<unsigned>& _vCoordsDst, std::vector<double>& _vResult)
Returns vector of values according to specified coordinates and dimensions sequence. Number of dimensions must be the same as defined in the transformation matrix, but their sequence can be different. Size of one vector of coordinates must be equal to the number of dimensions in transformation matrix; size of the second one must be one less. Returning value *_vResult* is a vector of mass fractions, which will be transferred from the source to the destination. Returns *false* on error.

Functions to set data

bool SetValue *(unsigned _nCoordSrc, unsigned _nCoordDst, double _dValue)*
Sets value by specified coordinates for one-dimensional distribution. *_nCoordSrc* is a coordinate of the source class; *_nCoordDst* is a coordinate of the destination class. *_dValue* is a mass fraction, which will be transferred from the source class to the destination class. Returns *false* on error.

bool SetValue *(unsigned _nCoordSrc1, unsigned _nCoordSrc2, unsigned _nCoordDst1,*
unsigned _nCoordDst2, double _dValue)
Sets value by specified coordinates for two-dimensional distribution. *_nCoordSrc1* and *_nCoordSrc2* are coordinate of the source class; *_nCoordDst1* and *_nCoordDst2* are coordinate of the destination class. *_dValue* is a mass fraction, which will be transferred from the source class to the destination class. Returns *false* on error.

bool SetValue *(const std::vector<unsigned>& _vCoordsSrc,*
const std::vector<unsigned>& _vCoordsDst, double _dValue)
Sets value by specified coordinates and full dimensions set. *_vCoordsSrc* are coordinates of the source class, *_vCoordsDst* are coordinates of the destination class. Sizes of vectors *_vCoordsSrc* and *_vCoordsDst* must be equal. *_dValue* is a mass fraction, which will be transferred from the source class to the destination class. Returns *false* on error.

bool SetValue *(const std::vector<unsigned>& _vDimsSrc,*
const std::vector<unsigned>& _vCoordsSrc, const std::vector<unsigned>& _vDimsDst,
const std::vector<unsigned>& _vCoordsDst, double _dValue)
Sets value according to specified coordinates and dimensions. Number of dimensions must be the same as defined in the transformation matrix, but their sequence can be different. Sizes of all vectors must be equal. *_dValue* is a mass fraction, which will be transferred from the source class to the destination class. Returns *false* on error.

bool SetVectorValue *(const std::vector<unsigned>& _vCoordsSrc,*
const std::vector<unsigned>& _vCoordsDst, const std::vector<double>& _vValue)
Sets vector of values by specified coordinates according to all defined dimensions. Size of one vector of coordinates must be equal to the number of dimensions in transformation matrix; size of the second one must be one less. *_vValue* is a vector of mass fractions, which will be transferred from the source to the destination. Returns *false* on error.

bool SetVectorValue *(const std::vector<unsigned>& _vDimsSrc,*
const std::vector<unsigned>& _vCoordsSrc, const std::vector<unsigned>& _vDimsDst,
const std::vector<unsigned>& _vCoordsDst, const std::vector<double>& _vValue)
Sets vector of values according to specified coordinates and dimensions sequence. Number of dimensions must be the same as defined in the transformation matrix, but their sequence can be different. Size of one vector of coordinates must be equal to the number of dimensions in transformation matrix; size of the second one must be one less. *_vValue* is a vector of mass fractions, which will be transferred from the source to the destination. Returns *false* on error.

Other functions

void Normalize *()*
Normalizes data in matrix: sets sum of material which transfers from each single class to 1.

void ClearData *()*
Sets all data in matrix equal to 0.

void Clear *()*
Removes all data and information about dimensions from the matrix.