
PartialDerivatives: public TriangularMatrix<Cartesian3>
+ PartialDerivatives(maximum_order_of_derivatives: GLint = 0)
+ loadNullVectors(): GLvoid
+ <<const>> clone(): PartialDerivatives*
DirectionalDerivatives: public ColumnMatrix<Cartesian3>
+ DirectionalDerivatives(maximum_order_of_derivatives: GLint = 0)
+ loadNullVectors(): GLvoid
+ <<const>> clone(): DirectionalDerivatives*
TensorProductSurface3
{abstract}
+ ImageColorScheme: {DEFAULT_NULL_FRAGMENT,
X_VARIATION_FRAGMENT, Y_VARIATION_FRAGMENT, Z_VARIATION_FRAGMENT,
NORMAL_LENGTH_FRAGMENT,
GAUSSIAN_CURVATURE_FRAGMENT, MEAN_CURVATURE_FRAGMENT,
WILLMORE_ENERGY_FRAGMENT, LOG_WILLMORE_ENERGY_FRAGMENT,
UMBILIC_DEVIATION_ENERGY_FRAGMENT, LOG_UMBILIC_DEVIATION_ENERGY_FRAGMENT,
TOTAL_CURVATURE_ENERGY_FRAGMENT, LOG_TOTAL_CURVATURE_ENERGY_FRAGMENT}
_min_value: std::vector<GLdouble>
_max_value: std::vector<GLdouble>
_closed: std::vector<GLboolean>
_vbo_data: GLuint
_data: Matrix<Cartesian3>
+ TensorProductSurface3(u_min: GLdouble, u_max: GLdouble, v_min: GLdouble, v_max: GLdouble,
row_count: GLint, column_count: GLint,
u_closed: GLboolean = GL_FALSE, v_closed: GLboolean = GL_FALSE)
+ TensorProductSurface3(surface: const TensorProductSurface3&)
+ operator =(rhs: const TensorProductSurface3&): TensorProductSurface3&
+ setInterval(type: variable::Type, min_value: GLdouble, max_value: GLdouble): GLboolean
+ <<const>> interval(type: variable::Type, min_value: GLdouble&, max_value: GLdouble&): GLvoid
+ <<const>> operator ()(row: GLint, column: GLint): const Cartesian3&
+ operator ()(row: GLint, column: GLint): Cartesian3&
+ <<const>> blendingFunctionValues(type: variable::Type, parameter_value: GLdouble,
values: RowMatrix<GLdouble>&): GLboolean
+ <<const>> calculateAllPartialDerivatives(maximum_order_of_derivatives: GLint,
u: GLdouble, v: GLdouble,
pd: PartialDerivatives&): GLboolean
+ <<const>> calculateDirectionalDerivatives(direction: variable::Type,
maximum_order_of_derivatives: GLint,
u: GLdouble, v: GLdouble,
d: DirectionalDerivatives &): GLboolean
+ <<const>> generateImage(u_div_point_count: GLint, v_div_point_count: GLint,
color_scheme: ImageColorScheme = DEFAULT_NULL_FRAGMENT,
usage_flag: GLenum = GL_STATIC_DRAW): TriangleMesh3*
+ <<const>> generateIsoparametricLines(type: variable::Type, iso_line_count: GLint,
maximum_order_of_derivatives: GLint, div_point_count: GLint,
usage_flag: GLenum = GL_STATIC_DRAW):
RowMatrix<SP<GenericCurve3>::Default>*
+ updateDataForInterpolation(u_knot_vector: const RowMatrix<GLdouble>&,
v_knot_vector: const ColumnMatrix<GLdouble>&,
data_points_to_interpolate: const Matrix<Cartesian3>&): GLboolean
+ deleteVertexBufferObjectsOfData(): GLvoid
+ <<const>> renderData(program: const ShaderProgram&,
u_render_mode: GLenum = GL_LINE_STRIP,
v_render_mode: GLenum = GL_LINE_STRIP): GLboolean
+ <<const>> renderData(u_render_mode: GLenum = GL_LINE_STRIP,
v_render_mode: GLenum = GL_LINE_STRIP,
vec3_position_location: GLint = 0): GLboolean
+ updateVertexBufferObjectsOfData(usage_flag: GLenum = GL_STATIC_DRAW): GLboolean
+ <<const>> rowCount(): GLint
+ <<const>> columnCount(): GLint
+ ~TensorProductSurface3()