TireGround

Davide Stocco

March 2020

Generated by Doxygen 1.8.13

Contents

Chapter 1

TireGround

A repository for the code developed by Davide Stocco for his thesis.

Department of Industrial Engineering
Master Degree in Mechatronics Engineering

EN: **Real-Time Computation of Tire/Road Contact using Tailored Algorithms**
IT: **Valutazione Real-Time del Contatto Pneumatico/Strada con Algoritmi Dedicati**

Academic Year 2019 · 2020

Author: **Davide Stocco**
Supervisor & Co-supervisor: **Prof. Enrico Bertolazzi** & **Dr.Eng. Matteo Ragni**

MagicFormula tire model usage

1. Load .rdf file.

```
TireGround::RDF::MeshSurface Road(
  "./file.rdf" // Path to the *.rdf file
);
```

2. Initialize the MagicFormula tire model.

```
TireGround::Tire* TireSD = new TireGround::MagicFormula(
  SectionWidth, // [m]
  AspectRatio,  // [%]
  RimDiameter,  // [in]
  SwitchNumber  // Maximum RoadTriangles in the Tire Shadow (switch to sampling)
);
```

3. Contact evaluation.

```
bool Out = TireSD->setup( Road,    // Road mesh
                          TransfMat // 4x4 total transformation matrix
                        );
```

4. Data extraction.

```
// Variable initialization (for real numbers)
TireGround::vec3 N;
TireGround::vec3 P;
TireGround::real_type Friction;
TireGround::real_type Rho;
TireGround::real_type RhoDot;
TireGround::real_type RelativeCamber;
TireGround::real_type Area;
TireGround::real_type Volume;

// Data extraction (for real numbers)
TireSD->getNormal(N);
```

```
TireSD->getMFpoint(P);
TireSD->getFriction(Friction);
TireSD->getRho(Rho);
TireSD->getRhoDot(PreviousRho,TimeStep,RhoDot);
TireSD->getRelativeCamber(RelativeCamber);
TireSD->getArea(Area);
TireSD->getVolume(Volume);

// Extract data stucture size
TireGround::int_type size = TireSD->getDisksNumber();

// Variable initialization (for vectors)
TireGround::row_vec3 NVec(size);
TireGround::row_vec3 PVec(size);
TireGround::row_vecN FrictionVec(size);
TireGround::row_vecN RhoVec(size);
TireGround::row_vecN RhoDotVec(size);
TireGround::row_vecN RelativeCamberVec(size);
TireGround::row_vecN AreaVec(size);
TireGround::row_vecN VolumeVec(size);

// Data extraction (for vectors)
TireSD->getNormal(NVec);
TireSD->getMFpoint(PVec);
TireSD->getFriction(FrictionVec);
TireSD->getRho(RhoVec);
TireSD->getRhoDot(PreviousRho,TimeStep,RhoDotVec);
TireSD->getRelativeCamber(RelativeCamberVec);
TireSD->getArea(AreaVec);
TireSD->getVolume(VolumeVec);
```

MultiDisk tire model usage

1. Load .rdf file.

```
TireGround::RDF::MeshSurface Road(
  "./file.rdf" // Path to the *.rdf file
);
```

2. Initialize the MultiDisk tire model:

   (a) MultiDisk tire without sidewall radius (uniform cylinder).

```
TireGround::Tire* TireMD = new TireGround::MultiDisk(
  SectionWidth, // [m]
  AspectRatio,  // [%]
  RimDiameter,  // [in]
  PointsNumber, // Sampling points for each disk
  DisksNumber,  // Disks number
  SwitchNumber  // Maximum RoadTriangles in the Tire Shadow (switch to sampling)
);
```

   (b) MultiDisk tire with sidewall radius (uniform cylinder with filleted sidewall edge).

```
TireGround::Tire* TireMD = new TireGround::MultiDisk(
  SectionWidth, // [m]
  AspectRatio,  // [%]
  RimDiameter,  // [in]
  SideRadius,   // Sidewall radius [m]
  PointsNumber, // Sampling points for each disk
  DisksNumber,  // Disks number
  SwitchNumber  // Maximum RoadTriangles in the Tire Shadow (switch to sampling)
);
```

   (c) MultiDisk tire with custom disks radius.

```
TireGround::Tire* TireMD = new TireGround::MultiDisk(
  SectionWidth, // [m]
  AspectRatio,  // [%]
  RimDiameter,  // [in]
  RadiusVec,    // Disks radius vector [m]
  PointsNumber, // Sampling points for each disk
  SwitchNumber  // Maximum RoadTriangles in the Tire Shadow (switch to sampling)
);
```

3. Contact evaluation.

```
bool Out = TireMD->setup( Road,    // Road mesh
                          TransfMat // 4x4 total transformation matrix
                        );
```

4. Data extraction for contact point(s).

```
// Variable initialization (for real numbers)
TireGround::vec3 N;
TireGround::vec3 P;
TireGround::real_type Friction;
TireGround::real_type Rho;
TireGround::real_type RhoDot;
TireGround::real_type RelativeCamber;
TireGround::real_type Area;
TireGround::real_type Volume;

// Data extraction (for real numbers)
TireMD->getNormal(N);
TireMD->getMFpoint(P);
TireMD->getFriction(Friction);
TireMD->getRho(Rho);
TireMD->getRhoDot(PreviousRho,TimeStep,RhoDot);
TireMD->getRelativeCamber(RelativeCamber);
TireMD->getArea(Area);
TireMD->getVolume(Volume);

// Extract data stucture size
TireGround::int_type size = TireSD->getDisksNumber();

// Variable initialization (for vectors)
TireGround::row_vec3 NVec(size);
TireGround::row_vec3 PVec(size);
TireGround::row_vecN FrictionVec(size);
TireGround::row_vecN RhoVec(size);
TireGround::row_vecN RhoDotVec(size);
TireGround::row_vecN RelativeCamberVec(size);
TireGround::row_vecN AreaVec(size);
TireGround::row_vecN VolumeVec(size);

// Data extraction (for vectors)
TireMD->getNormal(NVec);
TireMD->getMFpoint(PVec);
TireMD->getFriction(FrictionVec);
TireMD->getRho(RhoVec);
TireMD->getRhoDot(PreviousRho,TimeStep,RhoDotVec);
TireMD->getRelativeCamber(RelativeCamberVec);
TireMD->getArea(AreaVec);
TireMD->getVolume(VolumeVec);
```

Chapter 2

Namespace Index

2.1 Namespace List

Here is a list of all documented namespaces with brief descriptions:

Chapter 3

Hierarchical Index

3.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Chapter 4

Class Index

4.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 5

Namespace Documentation

5.1 TireGround Namespace Reference

Tire computations routines.

Namespaces

- algorithms

  Algorithms for tire computations routine.
- RDF

  RDF mesh computations routines.

Classes

- class Disk

  Tire disk.
- class ETRTO

  Tire ETRTO denomination.
- class MagicFormula

  Pacejka MagicFormula contact model.
- class MultiDisk

  Multi-disk tire contact model.
- class ReferenceFrame

  Reference frame.
- class SamplingGrid

  Patch evaluation precision.
- class Shadow

  2D shadow (2D bounding box enhacement)
- class Tire

  Base class for Tire models.

Typedefs

- typedef double real_type

  Real number type.
- typedef int int_type

  Integer number type.
- typedef Eigen::Vector2i vec2_int

  2D vector type of real integer type
- typedef Eigen::Vector2d vec2

  2D vector type of real number type
- typedef Eigen::Vector3d vec3

  3D vector type of real number type
- typedef Eigen::Vector4d vec4

  4D vector type of real number type
- typedef Eigen::Matrix3d mat3

  3x3 matrix type of real number type
- typedef Eigen::Matrix4d mat4

  4x4 matrix type of real number type
- typedef Eigen::Matrix< real_type, 1, Eigen::Dynamic > row_vecN

  Row vector type real number type.
- typedef Eigen::Matrix< real_type, Eigen::Dynamic, 1 > col_vecN

  Column vector type real number type.
- typedef Eigen::Matrix< real_type, Eigen::Dynamic, Eigen::Dynamic > matN

  Matrix type of real number type.
- typedef Eigen::Matrix< vec2, 1, Eigen::Dynamic > row_vec2

  Row vector type of 2D vector.
- typedef Eigen::Matrix< vec2, Eigen::Dynamic, 1 > col_vec2

  Column vector type of 2D vector.
- typedef Eigen::Matrix< vec2, Eigen::Dynamic, Eigen::Dynamic > mat_vec2

  Matrix type of 2D vector.
- typedef Eigen::Matrix< vec3, 1, Eigen::Dynamic > row_vec3

  Row vector type of 3D vector.
- typedef Eigen::Matrix< vec3, Eigen::Dynamic, 1 > col_vec3

  Column vector type of 3D vector.
- typedef Eigen::Matrix< vec3, Eigen::Dynamic, Eigen::Dynamic > matN_vec3

  Matrix type of 3D vector.
- typedef Eigen::Matrix< mat4, 1, Eigen::Dynamic > row_mat4

  Matrix type of 4x4 matrix.
- typedef std::basic_ostream< char > ostream_type

  Output stream type.

Variables

- real_type const epsilon = std::numeric_limits<real_type>::epsilon()

  Epsilon type.

5.1.1 Detailed Description

Tire computations routines.

Typedefs for tire computations routine.

file: PatchTire.hh

file: TireGround.hh

5.2 TireGround::algorithms Namespace Reference

Algorithms for tire computations routine.

Functions

- vec3 mean (row_vec3 const &Values)

  Calculate arithmetic weighted mean for 3D vectors.
- real_type weightedMean (row_vecN const &Values, row_vecN const &Weights)

  Calculate arithmetic weighted mean for real numbers.
- vec3 weightedMean (row_vec3 const &Values, row_vecN const &Weights)

  Calculate arithmetic weighted mean for 3D vectors.
- bool intersectPointSegment (vec2 const &Point1, vec2 const &Point2, vec2 const &PointQ)
- bool intersectRayPlane (vec3 const &planeN, vec3 const &planeP, vec3 const &RayPoint, vec3 const &RayDirection, vec3 &IntersectionPt)

  Check if a segment hits a plane and find the intersection point.
- void minmax_XY (row_vec3 const &Points, vec2 &XYmin, vec2 &XYmax)

  Calculate minumum and maximum in XY plane for 3D vectors.
- void minmax_XY (row_vec2 const &Points, vec2 &XYmin, vec2 &XYmax)

  Calculate minumum and maximum in XY plane for 2D vectors.
- real_type trapezoidArea (real_type const Base2, real_type const Base1, real_type const Height)

  Calculate area of a trapeziod [m^2].

5.2.1 Detailed Description

Algorithms for tire computations routine.

5.2.2 Function Documentation

5.2.2.1 intersectPointSegment()

```
bool TireGround::algorithms::intersectPointSegment (
            vec2 const & Point1,
            vec2 const & Point2,
            vec2 const & PointQ )
```

Check if a point lays inside or outside a line segment
Warning: The point query point must be on the same rect of the line segment!

Parameters

Point1	Line segment point 1
Point2	Line segment point 2
PointQ	Query point

5.2.2.2 intersectRayPlane()

```
bool TireGround::algorithms::intersectRayPlane (
            vec3 const & planeN,
            vec3 const & planeP,
            vec3 const & RayPoint,
            vec3 const & RayDirection,
            vec3 & IntersectionPt )
```

Check if a segment hits a plane and find the intersection point.

Parameters

planeN	Plane normal vector
planeP	Plane known point
RayPoint	Ray point
RayDirection	Ray direction
IntersectionPt	Intersection point

5.2.2.3 mean()

```
vec3 TireGround::algorithms::mean (
            row_vec3 const & Values )
```

Calculate arithmetic weighted mean for 3D vectors.

Parameters

Values	Values (3D vectors)

5.2.2.4 minmax_XY() [1/2]

```
void TireGround::algorithms::minmax_XY (
            row_vec3 const & Points,
            vec2 & XYmin,
            vec2 & XYmax )
```

Calculate minumum and maximum in XY plane for 3D vectors.

Parameters

Points	3D points vector
XYmin	Minimum (X, Y) values
XYmax	Maximum (X, Y) values

5.2.2.5 minmax_XY() [2/2]

```
void TireGround::algorithms::minmax_XY (
            row_vec2 const & Points,
            vec2 & XYmin,
            vec2 & XYmax )
```

Calculate minumum and maximum in XY plane for 2D vectors.

Parameters

Points	2D points vector
XYmin	Minimum (X, Y) values
XYmax	Maximum (X, Y) values

5.2.2.6 trapezoidArea()

```
real_type TireGround::algorithms::trapezoidArea (
            real_type const Base2,
            real_type const Base1,
            real_type const Height )  [inline]
```

Calculate area of a trapeziod [m^2].

Parameters

Base2	Base 1
Base1	Base 2
Height	Heigth

5.2.2.7 weightedMean() [1/2]

```
real_type TireGround::algorithms::weightedMean (
            row_vecN const & Values,
            row_vecN const & Weights )
```

Calculate arithmetic weighted mean for real numbers.

Parameters

Values	Values (real numbers)
Weights	Weights (real numbers)

5.2.2.8 weightedMean() [2/2]

```
vec3 TireGround::algorithms::weightedMean (
            row_vec3 const & Values,
            row_vecN const & Weights )
```

Calculate arithmetic weighted mean for 3D vectors.

Parameters

Values	Values (3D vectors)
Weights	Weights (real numbers)

5.3 TireGround::RDF Namespace Reference

RDF mesh computations routines.

Namespaces

- algorithms

  Algorithms for RDF mesh computations routine.

Classes

- class BBox2D

  2D Bounding Box class
- class MeshSurface

  Mesh surface.
- class Triangle3D

  3D triangle (pure geometrical description)
- class TriangleRoad

  3D triangles for road representation

Typedefs

- typedef std::shared_ptr< TriangleRoad > TriangleRoad_ptr

  Shared pointer to TriangleRoad object.
- typedef std::vector< TriangleRoad_ptr > TriangleRoad_list

  Vector of shared pointers to TriangleRoad objects.

5.3.1 Detailed Description

RDF mesh computations routines.

5.4 TireGround::RDF::algorithms Namespace Reference

Algorithms for RDF mesh computations routine.

Functions

- void split (std::string const &in, std::vector< std::string > &out, std::string const &token)

  Split a string into a string array at a given token.
- std::string tail (std::string const &in)

  Get tail of string after first token and possibly following spaces.
- std::string firstToken (std::string const &in)

  Get first token of string.
- template<typename T >
  T const & getElement (std::vector< T > const &elements, std::string const &index)

  Get element at given index position.

5.4.1 Detailed Description

Algorithms for RDF mesh computations routine.

5.4.2 Function Documentation

5.4.2.1 firstToken()

```
std::string TireGround::RDF::algorithms::firstToken (
            std::string const & in )
```

Get first token of string.

Parameters

in	Input string

5.4.2.2 getElement()

```
template<typename T >
T const& TireGround::RDF::algorithms::getElement (
            std::vector< T > const & elements,
            std::string const & index )
```

Get element at given index position.

Parameters

elements	Elements vector
index	Index position

5.4.2.3 split()

```
void TireGround::RDF::algorithms::split (
            std::string const & in,
            std::vector< std::string > & out,
            std::string const & token )
```

Split a string into a string array at a given token.

Parameters

in	Input string
out	Output string vector
token	Token

5.4.2.4 tail()

```
std::string TireGround::RDF::algorithms::tail (
            std::string const & in )
```

Get tail of string after first token and possibly following spaces.

Parameters

in	Input string

Chapter 6

Class Documentation

6.1 TireGround::RDF::BBox2D Class Reference

2D Bounding Box class

```
#include <RoadRDF.hh>
```

Public Member Functions

- BBox2D ()

  Default constructor.
- BBox2D (vec3 const Vertices[3])

  Variable set constructor.
- void setXmin (real_type const _Xmin)

  Set X_{min} shadow domain.
- void setYmin (real_type const _Ymin)

  Set Y_{min} shadow domain.
- void setXmax (real_type const _Xmax)

  Set X_{max} shadow domain.
- void setYmax (real_type const _Ymax)

  Set Y_{max} shadow domain.
- real_type getXmin (void) const

  Get X_{min} shadow domain.
- real_type getYmin (void) const

  Get Y_{min} shadow domain.
- real_type getXmax (void) const

  Get X_{max} shadow domain.
- real_type getYmax (void) const

  Get Y_{max} shadow domain.
- void clear (void)

  Clear the bounding box domain.
- void print (ostream_type &stream) const

  Print bounding box domain.
- void updateBBox2D (vec3 const Vertices[3])

  Update the bounding box domain with three input vertices.

6.1.1 Detailed Description

2D Bounding Box class

6.1.2 Constructor & Destructor Documentation

6.1.2.1 BBox2D()

```
TireGround::RDF::BBox2D::BBox2D (
            vec3 const Vertices[3] )  [inline]
```

Variable set constructor.

Parameters

Vertices	Vertices reference vector

6.1.3 Member Function Documentation

6.1.3.1 print()

```
void TireGround::RDF::BBox2D::print (
            ostream_type & stream ) const  [inline]
```

Print bounding box domain.

Parameters

stream	Output stream type

6.1.3.2 updateBBox2D()

```
void TireGround::RDF::BBox2D::updateBBox2D (
            vec3 const Vertices[3] )
```

Update the bounding box domain with three input vertices.

Parameters

Vertices	Vertices reference vector

The documentation for this class was generated from the following file:

- include/RoadRDF.hh

6.2 TireGround::Disk Class Reference

Tire disk.

```
#include <PatchTire.hh>
```

Public Member Functions

- Disk (Disk &&)=default

  Enable && operator.
- Disk ()

  Default constructor.
- Disk (vec2 const &_OriginXZ, real_type _OffsetY, real_type _Radius)

  Variable set constructor.
- void set (Disk const &in)

  Copy the Disk object.
- void setOriginXZ (vec2 const &_OriginXZ)

  Set origin on XZ plane.
- vec2 const & getOriginXZ (void) const

  Get origin vector XZ-axes coordinates.
- vec3 getOriginXYZ (void) const

  Get origin vector XYZ-axes coordinates.
- real_type getOffsetY (void) const

  Get origin Y-axis coordinate.
- real_type getRadius (void) const

  Get Disk radius.
- void contactTriangles (RDF::TriangleRoad_list const &TriList, ReferenceFrame const &RF, vec3 &Normal, real_type &Friction, real_type &Area) const
- void contactPlane (vec3 const &Normal, vec3 const &Point, ReferenceFrame const &RF, real_type &Area) const
- void pointOnDisk (vec3 const &Normal, ReferenceFrame const &RF, vec3 &DiskPoint, vec3 &NormalOnDisk) const

  Get the points on Disk the circumference and on a given plane.
- real_type segmentArea (real_type const Length) const
- bool isPointInside (vec2 const &Point) const

  Check if a point in Disk reference frame is inside or outside the Disk.
- real_type y (real_type const x) const

  Evaluate Y at a query X value on the lower side Disk circumfererence.
- real_type segmentLength (vec2 const Point1, vec2 const Point2) const

  Evaluate a generic segment length given 2 points on the Disk circumfererence.
- int_type intersectSegment (vec2 const &Point1, vec2 const &Point2, vec2 &Intersect1, vec2 &Intersect2) const
- bool intersectPlane (vec3 const &Plane_Normal, vec3 const &Plane_Point, vec3 &Line_↩ Direction, vec3 &Line_Point) const
- real_type getLineArea (vec2 const &Point1_XZ, vec2 const &Point2_XZ) const

  Get a two points line segment area [m^2] (as ouput) inside the Disk.

6.2.1 Detailed Description

Tire disk.

6.2.2 Constructor & Destructor Documentation

6.2.2.1 Disk()

```
TireGround::Disk::Disk (
             vec2 const & _OriginXZ,
             real_type _OffsetY,
             real_type _Radius )  [inline]
```

Variable set constructor.

Parameters

_OriginXZ	(X_0, Z_0) origin coordinate
_OffsetY	Y_0 origin coordinate (offset from center)
_Radius	Radius

6.2.3 Member Function Documentation

6.2.3.1 contactPlane()

```
void TireGround::Disk::contactPlane (
             vec3 const & Normal,
             vec3 const & Point,
             ReferenceFrame const & RF,
             real_type & Area ) const
```

Get the contact area [m^2] inside the single Disk given a plane in absolute reference frame

Parameters

Normal	Plane normal in absolute reference frame
Point	Plane point in absolute reference frame
RF	Tire ReferenceFrame
Area	Contact area [m^2]

6.2.3.2 contactTriangles()

```
void TireGround::Disk::contactTriangles (
             RDF::TriangleRoad_list const & TriList,
             ReferenceFrame const & RF,
             vec3 & Normal,
             real_type & Friction,
             real_type & Area ) const
```

Get area weighted mean road normal versor, area weighted mean friction and contact area [m^2] inside the single Disk of segments described by the intersection of triangles on XZ-plane

Parameters

TriList	Shadow / MeshSurface intersected triangles
RF	Tire ReferenceFrame
Normal	Area weighted mean road normal versor

Parameters

Friction	Area weighted mean contact friction
Area	Contact area [m^2]

6.2.3.3 getLineArea()

```
real_type TireGround::Disk::getLineArea (
            vec2 const & Point1_XZ,
            vec2 const & Point2_XZ ) const
```

Get a two points line segment area [m^2] (as ouput) inside the Disk.

Parameters

Point1_XZ	Point 1 in Disk reference frame
Point2_XZ	Point 2 in Disk reference frame

6.2.3.4 intersectPlane()

```
bool TireGround::Disk::intersectPlane (
            vec3 const & Plane_Normal,
            vec3 const & Plane_Point,
            vec3 & Line_Direction,
            vec3 & Line_Point ) const
```

Check if two plane intersects and find the intersecting rect given two points in Disk reference frame

Parameters

Plane_Normal	Plane normal vector in Disk reference frame
Plane_Point	Plane known point in Disk reference frame
Line_Direction	Rect direction vector in Disk reference frame
Line_Point	Plane known point in Disk reference frame

6.2.3.5 intersectSegment()

```
int_type TireGround::Disk::intersectSegment (
            vec2 const & Point1,
            vec2 const & Point2,
            vec2 & Intersect1,
            vec2 & Intersect2 ) const
```

Find the intersection points between the Disk and a two points line segment in Disk reference frame (output integer gives number of intersection points)

Parameters

Point1	Line segment point 1 in Disk reference frame

Parameters

Point2	Line segment point 2 in Disk reference frame
Intersect1	Intersection point 1 in Disk reference frame
Intersect2	Intersection point 2 in Disk reference frame

6.2.3.6 isPointInside()

```
bool TireGround::Disk::isPointInside (
            vec2 const & Point ) const
```

Check if a point in Disk reference frame is inside or outside the Disk.

Parameters

Point	Query point in Disk reference frame

6.2.3.7 segmentArea()

```
real_type TireGround::Disk::segmentArea (
            real_type const Length ) const  [inline]
```

Get the contact patch area under the intersection plane in absolute reference frame [m^2]

Parameters

Length	Chord length

6.2.3.8 segmentLength()

```
real_type TireGround::Disk::segmentLength (
            vec2 const Point1,
            vec2 const Point2 ) const  [inline]
```

Evaluate a generic segment length given 2 points on the Disk circumfererence.

Parameters

Point1	Point 1
Point2	Point 2

6.2.3.9 set()

```
void TireGround::Disk::set (
            Disk const & in )  [inline]
```

Copy the Disk object.

Parameters

in	Disk object to be copied

6.2.3.10 setOriginXZ()

```
void TireGround::Disk::setOriginXZ (
            vec2 const & _OriginXZ )   [inline]
```

Set origin on XZ plane.

Parameters

_OriginXZ	New origin on XZ plane

6.2.3.11 y()

```
real_type TireGround::Disk::y (
            real_type const x ) const   [inline]
```

Evaluate Y at a query X value on the lower side Disk circumference.

Parameters

x	Query X value

The documentation for this class was generated from the following file:

- include/PatchTire.hh

6.3 TireGround::ETRTO Class Reference

Tire ETRTO denomination.

`#include <PatchTire.hh>`

Public Member Functions

- ETRTO ()

  Default constructor.
- ETRTO (real_type _SectionWidth, real_type _AspectRatio, real_type _RimDiameter)

  Variable set constructor.
- real_type getSidewallHeight (void) const

  Get sidewall height [m].
- real_type getTireDiameter (void) const

  Get external tire diameter [m].
- real_type getTireRadius (void) const

  Get external tire radius [m].
- real_type getSectionWidth (void) const

Get section width [m].

- void print (ostream_type &stream) const

  Display tire data.

6.3.1 Detailed Description

Tire ETRTO denomination.

6.3.2 Constructor & Destructor Documentation

6.3.2.1 ETRTO()

```
TireGround::ETRTO::ETRTO (
            real_type _SectionWidth,
            real_type _AspectRatio,
            real_type _RimDiameter )   [inline]
```

Variable set constructor.

Parameters

_SectionWidth	Tire section width [m]
_AspectRatio	Tire aspect ratio [%]
_RimDiameter	Rim diameter [in]

6.3.3 Member Function Documentation

6.3.3.1 print()

```
void TireGround::ETRTO::print (
            ostream_type & stream ) const   [inline]
```

Display tire data.

Parameters

stream	Output stream type

The documentation for this class was generated from the following file:

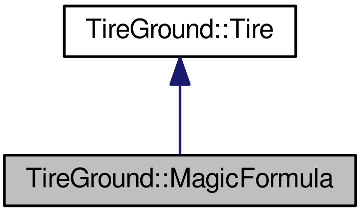
- include/PatchTire.hh

6.4 TireGround::MagicFormula Class Reference

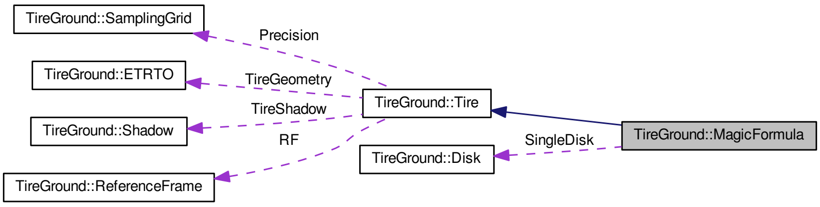
Pacejka MagicFormula contact model.

```
#include <PatchTire.hh>
```

Inheritance diagram for TireGround::MagicFormula:



Collaboration diagram for TireGround::MagicFormula:



Public Member Functions

- ~MagicFormula ()

  Default destructor.
- MagicFormula (real_type const SectionWidth, real_type const AspectRatio, real_type const RimDiameter, int_type const SwitchN)

  Variable set constructor.
- void getNormal (vec3 &_Normal) const override

  Get contact normal versor.
- void getNormal (row_vec3 &_Normal) const override

  Get contact normal versors vector.
- void getMFpoint (vec3 &_DiskPoint) const override

  Get Magic Formula contact point.
- void getMFpoint (row_vec3 &_DiskPoint) const override

  Get Magic Formula contact point vector.
- void getFriction (real_type &_Friction) const override

  Get contact point friction.
- void getFriction (row_vecN &_Friction) const override

  Get contact point friction vector.
- void getMFpointRF (mat4 &PointRF) const override

  Get Magic Formula contact point reference frame with 4x4 transformation matrix.

- void getMFpointRF (row_mat4 &_MFpointRF) const override

  Get Magic Formula contact point reference frame vector with 4x4 transformation matrix.
- void getRho (real_type &Rho, real_type &RhoDot, real_type const RhoOld, real_type const Time) const override
- void getRho (row_vecN &Rho, row_vecN &RhoDot, row_vecN const RhoOld, real_type const Time) const override
- void getArea (real_type &_Area) const override

  Get approximated contact area on Disk plane [m^2].
- void getArea (row_vecN &_Area) const override

  Get approximated contact area vector on Disk plane [m^2].
- void getVolume (real_type &_Volume) const override

  Get approximated contact volume [m^3].
- void getVolume (row_vecN &Volume) const override

  Get approximated contact volume vector [m^3].
- bool setup (RDF::MeshSurface &Mesh, mat4 const &TM) override

  Update current tire position and find contact parameters.
- void setup (vec3 const &Plane_Normal, vec3 const &Plane_Point, real_type const Plane_↩ Friction, mat4 const &TM) override
- void print (ostream_type &stream) const override

  Print contact parameters.
- void printETRTOGeometry (ostream_type &stream) const

  Display Tire ETRTO geometry data.
- G2lib::AABBtree::PtrAABB const getAABBtree (void) const

  Get total Tire Shadow G2Lib::AABBtree (3D projection on ground)
- G2lib::AABBtree::PtrAABB const getUpperSideAABBtree (void) const

  Get upper side Tire Shadow G2Lib:AABBtree (3D projection on ground)
- G2lib::AABBtree::PtrAABB const getLowerSideAABBtree (void) const

  Get lower side Tire Shadow G2Lib:AABBtree (3D projection on ground)
- void setReferenceFrame (ReferenceFrame const &_RF)
- ReferenceFrame const & getReferenceFrame (void) const

  Get tire ReferenceFrame object.
- void setOrigin (vec3 const &Origin)

  Set a new tire origin.
- void setRotationMatrix (mat3 const &RotationMatrix)
- void setTotalTransformationMatrix (mat4 const &TM)
- real_type getEulerAngleX (void) const
- real_type getEulerAngleY (void) const
- real_type getEulerAngleZ (void) const
- void getRelativeCamber (real_type &RelativeCamber) const

  Get relative camber angle [rad].
- int_type getDisksNumber (void) const

  Dimension of the contact points data structure (disks number)

Protected Member Functions

- MagicFormula (MagicFormula const &)=delete

  Deleted copy constructor.
- MagicFormula const & operator= (MagicFormula const &)=delete

  Deleted copy operator.
- void evaluateContact (RDF::TriangleRoad_list const &TriList) override

Evaluate contact with RoadTriangles.
- void fourPointsSampling (RDF::TriangleRoad_list const &TriList, vec3 &P_star)

  Perform triangles sampling on 4 points at $\pm 0.1*R$ along X and $\pm 0.3*W$ along Y.
- bool pointSampling (RDF::TriangleRoad_list const &TriList, vec3 const &RayOrigin, vec3 const &RayDirection, vec3 &SampledPt, real_type &TriFriction=quietNaN, vec3 &Tri←Normal=vec3_NaN) const

  Perform one point sampling (ray-triangle intersection)

Protected Attributes

- Disk SingleDisk

  Single Disk.
- vec3 Normal

  Contact normal versor.
- vec3 MeshPoint

  Contact point on Mesh (for Magic Formula)
- vec3 DiskPoint

  Contact point on undeformed Disk circumference (not for Magic Formula)
- real_type Friction

  Contact friction.
- real_type Area

  Contact area [m^2].
- SamplingGrid Precision

  Contacth patch evaluating precision.
- ETRTO TireGeometry

  Tire ETRTO denomination.
- ReferenceFrame RF

  ReferenceFrame.
- Shadow TireShadow

  Tire shadow.

6.4.1 Detailed Description

Pacejka MagicFormula contact model.

6.4.2 Constructor & Destructor Documentation

6.4.2.1 MagicFormula()

```
TireGround::MagicFormula::MagicFormula (
            real_type const SectionWidth,
            real_type const AspectRatio,
            real_type const RimDiameter,
            int_type const SwitchN )  [inline]
```

Variable set constructor.

Parameters

SectionWidth	Tire section width [m]
AspectRatio	Tire aspect ratio [%]
RimDiameter	Rim diameter [in]
SwitchN	Maximum RoadTriangles in the Tire Shadow (switch to sampling)

6.4.3 Member Function Documentation

6.4.3.1 evaluateContact()

```
void TireGround::MagicFormula::evaluateContact (
            RDF::TriangleRoad_list const & TriList )  [override], [protected], [virtual]
```

Evaluate contact with RoadTriangles.

Parameters

TriList	Shadow/MeshSurface intersected triangles

Implements TireGround::Tire.

6.4.3.2 fourPointsSampling()

```
void TireGround::MagicFormula::fourPointsSampling (
            RDF::TriangleRoad_list const & TriList,
            vec3 & P_star )  [protected]
```

Perform triangles sampling on 4 points at ±0.1∗R along X and ±0.3∗W along Y.

Parameters

TriList	Shadow/MeshSurface intersected triangles

6.4.3.3 getArea() [1/2]

```
void TireGround::MagicFormula::getArea (
            real_type & _Area ) const  [inline], [override], [virtual]
```

Get approximated contact area on Disk plane [m^2].

Parameters

_Area	Contact area [m^2]

Implements TireGround::Tire.

6.4.3.4 getArea() [2/2]

```
void TireGround::MagicFormula::getArea (
            row_vecN & _Area ) const  [inline], [override], [virtual]
```

Get approximated contact area vector on Disk plane [m^2].

Parameters

_Area	Contact area vector [m^2]

Implements TireGround::Tire.

6.4.3.5 getEulerAngleX()

```
real_type TireGround::Tire::getEulerAngleX (
            void  ) const  [inline], [inherited]
```

Get current Euler angles [rad] for X-axis
Warning: Factor as $[R_z][R_x][R_y]$!

6.4.3.6 getEulerAngleY()

```
real_type TireGround::Tire::getEulerAngleY (
            void  ) const  [inline], [inherited]
```

Get current Euler angles [rad] for Y-axis
Warning: Factor as $[R_z][R_x][R_y]$!

6.4.3.7 getEulerAngleZ()

```
real_type TireGround::Tire::getEulerAngleZ (
            void  ) const  [inline], [inherited]
```

Get current Euler angles [rad] for Z-axis
Warning: Factor as $[R_z][R_x][R_y]$!

6.4.3.8 getFriction() [1/2]

```
void TireGround::MagicFormula::getFriction (
            real_type & _Friction ) const  [inline], [override], [virtual]
```

Get contact point friction.

Parameters

_Friction	Contact point friction

Implements TireGround::Tire.

6.4.3.9 getFriction() [2/2]

```
void TireGround::MagicFormula::getFriction (
            row_vecN & _Friction ) const  [inline], [override], [virtual]
```

Get contact point friction vector.

Parameters

_Friction	Contact point friction vector

Implements TireGround::Tire.

6.4.3.10 getMFpoint() [1/2]

```
void TireGround::MagicFormula::getMFpoint (
            vec3 & _DiskPoint ) const  [inline], [override], [virtual]
```

Get Magic Formula contact point.

Parameters

_DiskPoint	Magic Formula contact point

Implements TireGround::Tire.

6.4.3.11 getMFpoint() [2/2]

```
void TireGround::MagicFormula::getMFpoint (
            row_vec3 & _DiskPoint ) const  [inline], [override], [virtual]
```

Get Magic Formula contact point vector.

Parameters

_DiskPoint	Contact point vector on Disk

Implements TireGround::Tire.

6.4.3.12 getMFpointRF() [1/2]

```
void TireGround::MagicFormula::getMFpointRF (
            mat4 & PointRF ) const  [override], [virtual]
```

Get Magic Formula contact point reference frame with 4x4 transformation matrix.

Parameters

PointRF	Magic Formula contact point reference frame

Implements TireGround::Tire.

6.4.3.13 getMFpointRF() [2/2]

```
void TireGround::MagicFormula::getMFpointRF (
            row_mat4 & _MFpointRF ) const  [inline], [override], [virtual]
```

Get Magic Formula contact point reference frame vector with 4x4 transformation matrix.

Parameters

_MFpointRF	Magic Formula ontact point reference frames vector

Implements TireGround::Tire.

6.4.3.14 getNormal() [1/2]

```
void TireGround::MagicFormula::getNormal (
            vec3 & _Normal ) const  [inline], [override], [virtual]
```

Get contact normal versor.

Parameters

_Normal	Contact point normal versor

Implements TireGround::Tire.

6.4.3.15 getNormal() [2/2]

```
void TireGround::MagicFormula::getNormal (
            row_vec3 & _Normal ) const  [inline], [override], [virtual]
```

Get contact normal versors vector.

Parameters

_Normal	Contact point normal direction vector

Implements TireGround::Tire.

6.4.3.16 getRelativeCamber()

```
void TireGround::Tire::getRelativeCamber (
            real_type & RelativeCamber ) const  [inherited]
```

Get relative camber angle [rad].

Parameters

RelativeCamber	Relative camber angle

6.4.3.17 getRho() [1/2]

```
void TireGround::MagicFormula::getRho (
            real_type & Rho,
            real_type & RhoDot,
            real_type const RhoOld,
            real_type const Time ) const  [override], [virtual]
```

Get contact depth at center point [m] and it time derivative [m/s]
Warning: (if negative the tire does not touch the ground)!

Parameters

Rho	Depth at center point [m/s]
RhoDot	Contact depth derivative [m/s]
RhoOld	Previous time step Rho [m]
Time	Time step [s]

Implements TireGround::Tire.

6.4.3.18 getRho() [2/2]

```
void TireGround::MagicFormula::getRho (
             row_vecN & Rho,
             row_vecN & RhoDot,
             row_vecN const RhoOld,
             real_type const Time ) const  [inline], [override], [virtual]
```

Get contact depth matrix [m] and it time derivatives [m/s]
Warning: (if negative the tire does not touch the ground)!

Parameters

Rho	Depth matrix [m/s]
RhoDot	Contact depth derivative matrix [m/s]
RhoOld	Previous time step Rho matrix [m]
Time	Time step [s]

Implements TireGround::Tire.

6.4.3.19 getVolume() [1/2]

```
void TireGround::MagicFormula::getVolume (
             real_type & _Volume ) const  [inline], [override], [virtual]
```

Get approximated contact volume [m^3].

Parameters

_Volume	Contact volume [m^3]

Implements TireGround::Tire.

6.4.3.20 getVolume() [2/2]

```
void TireGround::MagicFormula::getVolume (
             row_vecN & Volume ) const  [inline], [override], [virtual]
```

Get approximated contact volume vector [m^3].

Parameters

Volume	Contact volume vector [m^3]

Implements TireGround::Tire.

6.4.3.21 pointSampling()

```
bool TireGround::Tire::pointSampling (
            RDF::TriangleRoad_list const & TriList,
            vec3 const & RayOrigin,
            vec3 const & RayDirection,
            vec3 & SampledPt,
            real_type & TriFriction = quietNaN,
            vec3 & TriNormal = vec3_NaN ) const  [protected], [inherited]
```

Perform one point sampling (ray-triangle intersection)

Parameters

TriList	Shadow/MeshSurface intersected triangles
RayOrigin	Ray origin
RayDirection	Ray direction
SampledPt	Intersection point
TriFriction	Intersected triangle friction
TriNormal	Intersected triangle normal

6.4.3.22 print()

```
void TireGround::MagicFormula::print (
            ostream_type & stream ) const  [override], [virtual]
```

Print contact parameters.

Parameters

stream	Output stream type

Implements TireGround::Tire.

6.4.3.23 printETRTOGeometry()

```
void TireGround::Tire::printETRTOGeometry (
            ostream_type & stream ) const  [inline], [inherited]
```

Display Tire ETRTO geometry data.

Parameters

stream	Output stream type

6.4.3.24 setOrigin()

```
void TireGround::Tire::setOrigin (
            vec3 const & Origin )  [inline], [inherited]
```

Set a new tire origin.

Parameters

Origin	Tire origin

6.4.3.25 setReferenceFrame()

```
void TireGround::Tire::setReferenceFrame (
            ReferenceFrame const & _RF )  [inline], [inherited]
```

Copy the tire ReferenceFrame object
Warning: Rotation matrix must be orthonormal!

Parameters

_RF	ReferenceFrame object to be copied

6.4.3.26 setRotationMatrix()

```
void TireGround::Tire::setRotationMatrix (
            mat3 const & RotationMatrix )  [inline], [inherited]
```

Set a new 3x3 rotation matrix
Warning: Rotation matrix must be orthonormal!

Parameters

RotationMatrix	Rotation matrix

6.4.3.27 setTotalTransformationMatrix()

```
void TireGround::Tire::setTotalTransformationMatrix (
            mat4 const & TM )  [inline], [inherited]
```

Set 4x4 total transformation matrix
Warning: Rotation matrix must be orthonormal!

Parameters

TM	4x4 total transformation matrix

6.4.3.28 setup() [1/2]

```
bool TireGround::MagicFormula::setup (
            RDF::MeshSurface & Mesh,
            mat4 const & TM )  [override], [virtual]
```

Update current tire position and find contact parameters.

Parameters

Mesh	MeshSurface object (road)
TM	4x4 total transformation matrix

Implements TireGround::Tire.

6.4.3.29 setup() [2/2]

```
void TireGround::MagicFormula::setup (
            vec3 const & Plane_Normal,
            vec3 const & Plane_Point,
            real_type const Plane_Friction,
            mat4 const & TM )  [override], [virtual]
```

Update current tire position and find contact parameters with external plane

Parameters

Plane_Normal	Plane normal vector
Plane_Point	Plane known point
Plane_Friction	Friction on plane
TM	4x4 total transformation matrix

Implements TireGround::Tire.

The documentation for this class was generated from the following file:

- include/PatchTire.hh

6.5 TireGround::RDF::MeshSurface Class Reference

Mesh surface.

`#include <RoadRDF.hh>`

Public Member Functions

- MeshSurface ()

  Default set constructor.
- MeshSurface (TriangleRoad_list const &_PtrTriangleVec)

  Variable set constructor.
- MeshSurface (std::string const &Path)

  Variable set constructor.
- TriangleRoad_list const & getTrianglesList (void) const

  Get all triangles inside the mesh as a vector.
- TriangleRoad_ptr const getTriangle (unsigned i) const

  Get i-th TriangleRoad.
- G2lib::AABBtree::PtrAABB const getAABBPtr (void) const

  Get AABBtree object.
- void printData (std::string const &FileName) const

Print data in file.
- std::vector< G2lib::BBox::PtrBBox > const & getPtrBBoxList () const

  Get the mesh G2lib bounding boxes pointers vector.
- void set (MeshSurface const &in)

  Copy the MeshSurface object.
- bool LoadFile (std::string const &Path)

  Load the RDF model and print information on a file.
- bool intersectAABBtree (G2lib::AABBtree::PtrAABB const &AABBTreePtr, RDF::TriangleRoad_list &TrianglesList) const

  Intersect the mesh AABB tree with an external AABB tree.
- bool intersectBBox (std::vector< G2lib::BBox::PtrBBox > const &BBoxPtr, RDF::TriangleRoad_list &TrianglesList) const

  Update the mesh AABBtree with an external G2lib::BBox object pointer vector.

6.5.1 Detailed Description

Mesh surface.

6.5.2 Constructor & Destructor Documentation

6.5.2.1 MeshSurface() [1/2]

```
TireGround::RDF::MeshSurface::MeshSurface (
            TriangleRoad_list const & _PtrTriangleVec )  [inline]
```

Variable set constructor.

Parameters

_PtrTriangleVec	Road triangles pointer vector list

6.5.2.2 MeshSurface() [2/2]

```
TireGround::RDF::MeshSurface::MeshSurface (
            std::string const & Path )  [inline]
```

Variable set constructor.

Parameters

Path	Path to the RDF file

6.5.3 Member Function Documentation

6.5.3.1 intersectAABBtree()

```
bool TireGround::RDF::MeshSurface::intersectAABBtree (
            G2lib::AABBtree::PtrAABB const & AABBTreePtr,
            RDF::TriangleRoad_list & TrianglesList ) const
```

Intersect the mesh AABB tree with an external AABB tree.

Parameters

AABBTreePtr	External AABBtree object pointer
TrianglesList	Intersected TriangleRoad vector list

6.5.3.2 intersectBBox()

```
bool TireGround::RDF::MeshSurface::intersectBBox (
             std::vector< G2lib::BBox::PtrBBox > const & BBoxPtr,
             RDF::TriangleRoad_list & TrianglesList ) const
```

Update the mesh AABBtree with an external G2lib::BBox object pointer vector.

Parameters

BBoxPtr	External G2lib::BBox object pointer vector
TrianglesList	Intersected TriangleRoad vector list

6.5.3.3 LoadFile()

```
bool TireGround::RDF::MeshSurface::LoadFile (
             std::string const & Path )
```

Load the RDF model and print information on a file.

Parameters

Path	Path to the RDF file

6.5.3.4 printData()

```
void TireGround::RDF::MeshSurface::printData (
             std::string const & FileName ) const
```

Print data in file.

Parameters

FileName	File name in which print data

6.5.3.5 set()

```
void TireGround::RDF::MeshSurface::set (
             MeshSurface const & in )  [inline]
```

Copy the MeshSurface object.

Parameters

in	MeshSurface object to be copied

The documentation for this class was generated from the following file:

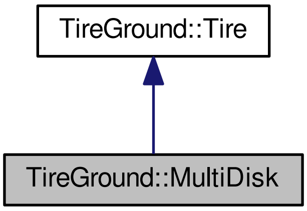
- include/RoadRDF.hh

6.6 TireGround::MultiDisk Class Reference

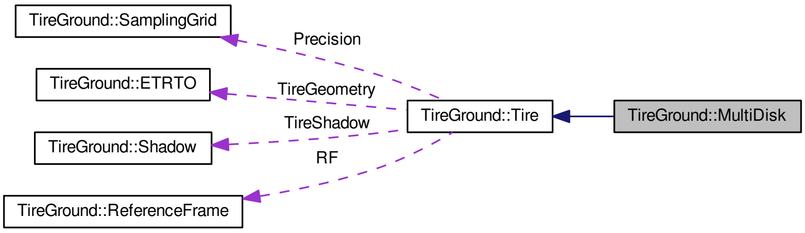
Multi-disk tire contact model.

```
#include <PatchTire.hh>
```

Inheritance diagram for TireGround::MultiDisk:



Collaboration diagram for TireGround::MultiDisk:



Public Member Functions

- ~MultiDisk ()

  Default destructor.
- MultiDisk (real_type const SectionWidth, real_type const AspectRatio, real_type const RimDiameter, int_type const PointsN, int_type const DisksN, int_type const SwitchN)

  Variable set constructor.

- MultiDisk (real_type const SectionWidth, real_type const AspectRatio, real_type const RimDiameter, real_type const SideRadius, int_type const PointsN, int_type const DisksN, int_type const SwitchN)

  Variable set constructor.
- MultiDisk (real_type const SectionWidth, real_type const AspectRatio, real_type const RimDiameter, row_vecN const DisksRadius, int_type const PointsN, int_type const SwitchN)

  Variable set constructor.
- real_type getPointstep (void) const

  Get grid step on X-axis between sampling points [m].
- real_type getDiskStep (void) const

  Get step on Y-axis between disks [m].
- void getNormal (vec3 &_Normal) const override

  Get contact normal mean versor.
- void getDiskOriginXYZ (row_vec3 &Origin) const

  Get disks origin (X, Y, Z).
- void getDiskOriginXYZ (int_type const i, vec3 &Origin) const

  Get i-th Disk origin (X, Y, Z).
- void setDiskOriginXZ (row_vec2 &Origin)

  Set disks origin (X, Y, Z).
- void setDiskOriginXZ (int_type const i, vec2 &Origin)

  Set i-th Disk origin (X, Y, Z).
- void getNormal (row_vec3 &_NormalVec) const override

  Get contact normal versors vector.
- void getDiskNormal (int_type const i, vec3 &_Normal) const

  Get i-th Disk contact normal versor.
- void getMFpoint (vec3 &_DiskPoint) const override

  Get Magic Formula contact point.
- void getMFpoint (row_vec3 &_DiskPointVec) const override

  Get Magic Formula contact points vector.
- void getDiskMFpoint (int_type const i, vec3 &_DiskPoint) const

  Get i-th Disk Magic Formula contact point.
- void getFriction (real_type &_Friction) const override

  Get area weighted mean contact friction.
- void getFriction (row_vecN &_Friction) const override

  Get contact frictions vector.
- void getDiskFriction (int_type const i, real_type &_Friction) const

  Get i-th Disk contact friction.
- void getMFeffectiveRF (mat4 &PointRF) const

  Get effective contact point reference frame with 4x4 transformation matrix.
- void getMFpointRF (mat4 &PointRF) const override

  Get Magic Formula contact point reference frame with 4x4 transformation matrix.
- void getMFpointRF (row_mat4 &PointRF) const override

  Get Magic Formula contact point reference frames vector with 4x4 transformation matrix.
- void getDiskMFpointRF (int_type const i, mat4 &PointRF) const

  Get Disk Magic Formula contact point reference frame with 4x4 transformation matrix.
- void getRho (real_type &Rho, real_type &RhoDot, real_type const RhoOld, real_type const Time) const override
- void getRho (row_vecN &Rho, row_vecN &RhoDot, row_vecN const RhoOld, real_type const Time) const override

- void getDiskRho (int_type const i, real_type &Rho, real_type &RhoDot, real_type const RhoOld, real_type const Time) const
- void getArea (real_type &_Area) const override

  Get approximated mean contact area on Disk plane [m^2].
- void getArea (row_vecN &_AreaVec) const override

  Get approximated contact areas vector on Disk plane [m^2].
- void getVolume (real_type &Volume) const override

  Get approximated contact volume [m^3].
- void getVolume (row_vecN &Volume) const override

  Get approximated contact volumes vector [m^3].
- void getMFeffectiveY (real_type &effectiveY) const

  Get effective Y-axis coordinate of contact point [m].
- void getMFeffectiveR (real_type &Radius) const

  Get effective radius of contact point [m].
- bool setup (RDF::MeshSurface &Mesh, mat4 const &TM) override

  Update current tire position and find contact parameters.
- void setup (vec3 const &Plane_Normal, vec3 const &Plane_Point, real_type const Plane_↩ Friction, mat4 const &TM) override
- void print (ostream_type &stream) const override

  Print contact parameters.
- void printETRTOGeometry (ostream_type &stream) const

  Display Tire ETRTO geometry data.
- G2lib::AABBtree::PtrAABB const getAABBtree (void) const

  Get total Tire Shadow G2Lib::AABBtree (3D projection on ground)
- G2lib::AABBtree::PtrAABB const getUpperSideAABBtree (void) const

  Get upper side Tire Shadow G2Lib:AABBtree (3D projection on ground)
- G2lib::AABBtree::PtrAABB const getLowerSideAABBtree (void) const

  Get lower side Tire Shadow G2Lib:AABBtree (3D projection on ground)
- void setReferenceFrame (ReferenceFrame const &_RF)
- ReferenceFrame const & getReferenceFrame (void) const

  Get tire ReferenceFrame object.
- void setOrigin (vec3 const &Origin)

  Set a new tire origin.
- void setRotationMatrix (mat3 const &RotationMatrix)
- void setTotalTransformationMatrix (mat4 const &TM)
- real_type getEulerAngleX (void) const
- real_type getEulerAngleY (void) const
- real_type getEulerAngleZ (void) const
- void getRelativeCamber (real_type &RelativeCamber) const

  Get relative camber angle [rad].
- int_type getDisksNumber (void) const

  Dimension of the contact points data structure (disks number)

Protected Member Functions

- bool pointSampling (RDF::TriangleRoad_list const &TriList, vec3 const &RayOrigin, vec3 const &RayDirection, vec3 &SampledPt, real_type &TriFriction=quietNaN, vec3 &Tri↩ Normal=vec3_NaN) const

  Perform one point sampling (ray-triangle intersection)

Protected Attributes

- SamplingGrid Precision

  Contacth patch evaluating precision.
- ETRTO TireGeometry

  Tire ETRTO denomination.
- ReferenceFrame RF

  ReferenceFrame.
- Shadow TireShadow

  Tire shadow.

6.6.1 Detailed Description

Multi-disk tire contact model.

6.6.2 Constructor & Destructor Documentation

6.6.2.1 MultiDisk() [1/3]

```
TireGround::MultiDisk::MultiDisk (
            real_type const SectionWidth,
            real_type const AspectRatio,
            real_type const RimDiameter,
            int_type const PointsN,
            int_type const DisksN,
            int_type const SwitchN )  [inline]
```

Variable set constructor.

Parameters

SectionWidth	Tire section width [m]
AspectRatio	Tire aspect ratio [$\%$]
RimDiameter	Rim diameter [in]
PointsN	Sampling points for each Disk (divisions on X-axis)
DisksN	Number of Disks (divisions on Y-axis -1)
SwitchN	Maximum RoadTriangles in the Tire Shadow (switch to sampling)

6.6.2.2 MultiDisk() [2/3]

```
TireGround::MultiDisk::MultiDisk (
            real_type const SectionWidth,
            real_type const AspectRatio,
            real_type const RimDiameter,
            real_type const SideRadius,
            int_type const PointsN,
            int_type const DisksN,
            int_type const SwitchN )  [inline]
```

Variable set constructor.

Parameters

SectionWidth	Tire section width [m]
AspectRatio	Tire aspect ratio [$\%$]
RimDiameter	Rim diameter [in]
SideRadius	Sidewall radius [m]
PointsN	Sampling points for each Disk (divisions on X-axis)
DisksN	Number of Disks (divisions on Y-axis -1)
SwitchN	Maximum RoadTriangles in the Tire Shadow (switch to sampling)

6.6.2.3 MultiDisk() [3/3]

```
TireGround::MultiDisk::MultiDisk (
             real_type const SectionWidth,
             real_type const AspectRatio,
             real_type const RimDiameter,
             row_vecN const DisksRadius,
             int_type const PointsN,
             int_type const SwitchN )  [inline]
```

Variable set constructor.

Parameters

SectionWidth	Tire section width [m]
AspectRatio	Tire aspect ratio [$\%$]
RimDiameter	Rim diameter [in]
DisksRadius	Disks radius vector [m]
PointsN	Sampling points for each Disk (divisions on X-axis)
SwitchN	Maximum RoadTriangles in the Tire Shadow (switch to sampling)

6.6.3 Member Function Documentation

6.6.3.1 getArea() [1/2]

```
void TireGround::MultiDisk::getArea (
             real_type & _Area ) const  [inline], [override], [virtual]
```

Get approximated mean contact area on Disk plane [m^2].

Parameters

_Area	Contact area [m^2]

Implements TireGround::Tire.

6.6.3.2 getArea() [2/2]

```
void TireGround::MultiDisk::getArea (
            row_vecN & _AreaVec ) const  [inline], [override], [virtual]
```

Get approximated contact areas vector on Disk plane [m^2].

Parameters

_AreaVec	Contact areas vector [m^2]

Implements TireGround::Tire.

6.6.3.3 getDiskFriction()

```
void TireGround::MultiDisk::getDiskFriction (
            int_type const i,
            real_type & _Friction ) const  [inline]
```

Get i-th Disk contact friction.

Parameters

i	i-th Disk
_Friction	Disk contact friction

6.6.3.4 getDiskMFpoint()

```
void TireGround::MultiDisk::getDiskMFpoint (
            int_type const i,
            vec3 & _DiskPoint ) const  [inline]
```

Get i-th Disk Magic Formula contact point.

Parameters

i	i-th Disk
_DiskPoint	Disk Magic Formula contact point

6.6.3.5 getDiskMFpointRF()

```
void TireGround::MultiDisk::getDiskMFpointRF (
            int_type const i,
            mat4 & PointRF ) const
```

Get Disk Magic Formula contact point reference frame with 4x4 transformation matrix.

Parameters

i	i-th Disk
PointRF	Magic Formula contact point reference frame

6.6.3.6 getDiskNormal()

```
void TireGround::MultiDisk::getDiskNormal (
             int_type const i,
             vec3 & _Normal ) const  [inline]
```

Get i-th Disk contact normal versor.

Parameters

i	i-th Disk
_Normal	Contact normal versor

6.6.3.7 getDiskOriginXYZ() [1/2]

```
void TireGround::MultiDisk::getDiskOriginXYZ (
             row_vec3 & Origin ) const  [inline]
```

Get disks origin (X, Y, Z).

Parameters

Origin	Disks origin

6.6.3.8 getDiskOriginXYZ() [2/2]

```
void TireGround::MultiDisk::getDiskOriginXYZ (
             int_type const i,
             vec3 & Origin ) const  [inline]
```

Get i-th Disk origin (X, Y, Z).

Parameters

i	i-th Disk
Origin	Disks origin

6.6.3.9 getDiskRho()

```
void TireGround::MultiDisk::getDiskRho (
             int_type const i,
             real_type & Rho,
             real_type & RhoDot,
             real_type const RhoOld,
             real_type const Time ) const
```

Get i-th Disk contact depth [m] and it time derivative [m/s]
Warning: (if negative the tire does not touch the ground)!

Parameters

i	i-th Disk
Rho	Disk contact depth
RhoDot	Contact depth derivative [m/s]
RhoOld	Previous time step Rho [m]
Time	Time step [s]

6.6.3.10 getEulerAngleX()

```
real_type TireGround::Tire::getEulerAngleX (
             void  ) const  [inline], [inherited]
```

Get current Euler angles [rad] for X-axis
Warning: Factor as $[R_z][R_x][R_y]$!

6.6.3.11 getEulerAngleY()

```
real_type TireGround::Tire::getEulerAngleY (
             void  ) const  [inline], [inherited]
```

Get current Euler angles [rad] for Y-axis
Warning: Factor as $[R_z][R_x][R_y]$!

6.6.3.12 getEulerAngleZ()

```
real_type TireGround::Tire::getEulerAngleZ (
             void  ) const  [inline], [inherited]
```

Get current Euler angles [rad] for Z-axis
Warning: Factor as $[R_z][R_x][R_y]$!

6.6.3.13 getFriction() [1/2]

```
void TireGround::MultiDisk::getFriction (
             real_type & _Friction ) const  [override], [virtual]
```

Get area weighted mean contact friction.

Parameters

_Friction	Area weighted mean contact friction

Implements TireGround::Tire.

6.6.3.14 getFriction() [2/2]

```
void TireGround::MultiDisk::getFriction (
             row_vecN & _Friction ) const  [inline], [override], [virtual]
```

Get contact frictions vector.

Parameters

_Friction	Contact frictions vector

Implements TireGround::Tire.

6.6.3.15 getMFeffectiveR()

```
void TireGround::MultiDisk::getMFeffectiveR (
              real_type & Radius ) const
```

Get effective radius of contact point [m].

Parameters

Radius	Effective radius of contact point [m]

6.6.3.16 getMFeffectiveRF()

```
void TireGround::MultiDisk::getMFeffectiveRF (
              mat4 & PointRF ) const
```

Get effective contact point reference frame with 4x4 transformation matrix.

Parameters

PointRF	Magic Formula contact point reference frame

6.6.3.17 getMFeffectiveY()

```
void TireGround::MultiDisk::getMFeffectiveY (
              real_type & effectiveY ) const
```

Get effective Y-axis coordinate of contact point [m].

Parameters

effectiveY	Effective Y-axis coordinate of contact point [m]

6.6.3.18 getMFpoint() [1/2]

```
void TireGround::MultiDisk::getMFpoint (
              vec3 & _DiskPoint ) const  [inline], [override], [virtual]
```

Get Magic Formula contact point.

Parameters

_DiskPoint	Magic Formula contact point

Implements TireGround::Tire.

6.6.3.19 getMFpoint() [2/2]

```
void TireGround::MultiDisk::getMFpoint (
            row_vec3 & _DiskPointVec ) const  [inline], [override], [virtual]
```

Get Magic Formula contact points vector.

Parameters

_DiskPointVec	Magic Formula contact points vector

Implements TireGround::Tire.

6.6.3.20 getMFpointRF() [1/2]

```
void TireGround::MultiDisk::getMFpointRF (
            mat4 & PointRF ) const  [override], [virtual]
```

Get Magic Formula contact point reference frame with 4x4 transformation matrix.

Parameters

PointRF	Magic Formula contact point reference frame

Implements TireGround::Tire.

6.6.3.21 getMFpointRF() [2/2]

```
void TireGround::MultiDisk::getMFpointRF (
            row_mat4 & PointRF ) const  [inline], [override], [virtual]
```

Get Magic Formula contact point reference frames vector with 4x4 transformation matrix.

Parameters

PointRF	Magic Formula contact point reference frames vector

Implements TireGround::Tire.

6.6.3.22 getNormal() [1/2]

```
void TireGround::MultiDisk::getNormal (
            vec3 & _Normal ) const  [inline], [override], [virtual]
```

Get contact normal mean versor.

Parameters

_Normal	Contact normal mean versor

Implements TireGround::Tire.

6.6.3.23 getNormal() [2/2]

```
void TireGround::MultiDisk::getNormal (
            row_vec3 & _NormalVec ) const  [inline], [override], [virtual]
```

Get contact normal versors vector.

Parameters

_NormalVec	Contact normal versors vector

Implements TireGround::Tire.

6.6.3.24 getRelativeCamber()

```
void TireGround::Tire::getRelativeCamber (
            real_type & RelativeCamber ) const  [inherited]
```

Get relative camber angle [rad].

Parameters

RelativeCamber	Relative camber angle

6.6.3.25 getRho() [1/2]

```
void TireGround::MultiDisk::getRho (
            real_type & Rho,
            real_type & RhoDot,
            real_type const RhoOld,
            real_type const Time ) const  [override], [virtual]
```

Get contact depth at center point [m] and it time derivative [m/s]
Warning: (if negative the tire does not touch the ground)!

Parameters

Rho	Depth at center point [m/s]
RhoDot	Contact depth derivative [m/s]
RhoOld	Previous time step Rho [m]
Time	Time step [s]

Implements TireGround::Tire.

6.6.3.26 getRho() [2/2]

```
void TireGround::MultiDisk::getRho (
            row_vecN & Rho,
            row_vecN & RhoDot,
            row_vecN const RhoOld,
            real_type const Time ) const  [override], [virtual]
```

Get contact depths vector [m] and it time derivatives [m/s]
Warning: (if negative the tire does not touch the ground)!

Parameters

Rho	Depth matrix [m/s]
RhoDot	Contact depth derivative matrix [m/s]
RhoOld	Previous time step Rho matrix [m]
Time	Time step [s]

Implements TireGround::Tire.

6.6.3.27 getVolume() [1/2]

```
void TireGround::MultiDisk::getVolume (
            real_type & Volume ) const  [inline], [override], [virtual]
```

Get approximated contact volume [m^3].

Parameters

Volume	Contact volume [m^3]

Implements TireGround::Tire.

6.6.3.28 getVolume() [2/2]

```
void TireGround::MultiDisk::getVolume (
            row_vecN & Volume ) const  [inline], [override], [virtual]
```

Get approximated contact volumes vector [m^3].

Parameters

Volume	Contact volumes vector [m^3]

Implements TireGround::Tire.

6.6.3.29 pointSampling()

```
bool TireGround::Tire::pointSampling (
            RDF::TriangleRoad_list const & TriList,
            vec3 const & RayOrigin,
            vec3 const & RayDirection,
            vec3 & SampledPt,
            real_type & TriFriction = quietNaN,
            vec3 & TriNormal = vec3_NaN ) const  [protected], [inherited]
```

Perform one point sampling (ray-triangle intersection)

Parameters

TriList	Shadow/MeshSurface intersected triangles
RayOrigin	Ray origin
RayDirection	Ray direction
SampledPt	Intersection point
TriFriction	Intersected triangle friction
TriNormal	Intersected triangle normal

6.6.3.30 print()

```
void TireGround::MultiDisk::print (
            ostream_type & stream ) const  [override], [virtual]
```

Print contact parameters.

Parameters

stream	Output stream type

Implements TireGround::Tire.

6.6.3.31 printETRTOGeometry()

```
void TireGround::Tire::printETRTOGeometry (
            ostream_type & stream ) const  [inline], [inherited]
```

Display Tire ETRTO geometry data.

Parameters

stream	Output stream type

6.6.3.32 setDiskOriginXZ() [1/2]

```
void TireGround::MultiDisk::setDiskOriginXZ (
            row_vec2 & Origin )  [inline]
```

Set disks origin (X, Y, Z).

Parameters

Origin	New Disks origin vector

6.6.3.33 setDiskOriginXZ() [2/2]

```
void TireGround::MultiDisk::setDiskOriginXZ (
             int_type const i,
             vec2 & Origin )  [inline]
```

Set i-th Disk origin (X, Y, Z).

Parameters

i	i-th Disk
Origin	New Disks origin vector

6.6.3.34 setOrigin()

```
void TireGround::Tire::setOrigin (
             vec3 const & Origin )  [inline], [inherited]
```

Set a new tire origin.

Parameters

Origin	Tire origin

6.6.3.35 setReferenceFrame()

```
void TireGround::Tire::setReferenceFrame (
             ReferenceFrame const & _RF )  [inline], [inherited]
```

Copy the tire ReferenceFrame object
Warning: Rotation matrix must be orthonormal!

Parameters

_RF	ReferenceFrame object to be copied

6.6.3.36 setRotationMatrix()

```
void TireGround::Tire::setRotationMatrix (
             mat3 const & RotationMatrix )  [inline], [inherited]
```

Set a new 3x3 rotation matrix
Warning: Rotation matrix must be orthonormal!

Parameters

RotationMatrix	Rotation matrix

6.6.3.37 setTotalTransformationMatrix()

```
void TireGround::Tire::setTotalTransformationMatrix (
             mat4 const & TM )  [inline], [inherited]
```

Set 4x4 total transformation matrix
Warning: Rotation matrix must be orthonormal!

Parameters

TM	4x4 total transformation matrix

6.6.3.38 setup() [1/2]

```
bool TireGround::MultiDisk::setup (
            RDF::MeshSurface & Mesh,
            mat4 const & TM )  [override], [virtual]
```

Update current tire position and find contact parameters.

Parameters

Mesh	MeshSurface object (road)
TM	4x4 total transformation matrix

Implements TireGround::Tire.

6.6.3.39 setup() [2/2]

```
void TireGround::MultiDisk::setup (
            vec3 const & Plane_Normal,
            vec3 const & Plane_Point,
            real_type const Plane_Friction,
            mat4 const & TM )  [override], [virtual]
```

Update current tire position and find contact parameters with external plane

Parameters

Plane_Normal	Plane normal vector
Plane_Point	Plane known point
Plane_Friction	Friction on plane
TM	4x4 total transformation matrix

Implements TireGround::Tire.

The documentation for this class was generated from the following file:

- include/PatchTire.hh

6.7 TireGround::ReferenceFrame Class Reference

Reference frame.

```
#include <PatchTire.hh>
```

Public Member Functions

- ReferenceFrame ()

  Default constructor.
- ReferenceFrame (vec3 const &_Origin, mat3 const &_RotationMatrix)

  Variable set constructor.
- bool isEmpty (void)

  Check if ReferenceFrame object is empty.
- mat3 const & getRotationMatrix (void) const

  Get current 3x3 rotation matrix.
- mat3 getRotationMatrixInverse (void) const

  Get current 3x3 rotation matrix inverse.
- vec3 getX (void) const

  Get current X-axis versor.
- vec3 getY (void) const

  Get current Y-axis versor.
- vec3 getZ (void) const

  Get current Z-axis versor.
- vec3 const & getOrigin (void) const

  Get origin position.
- void setOrigin (vec3 const &_Origin)

  Set origin position.
- void setRotationMatrix (mat3 const &_RotationMatrix)

  Set 3x3 rotation matrix.
- void setTotalTransformationMatrix (mat4 const &TM)

  Set 4x4 total transformation matrix.
- mat4 getTotalTransformationMatrix (void)

  Get 4x4 total transformation matrix.
- void set (ReferenceFrame const &in)
- real_type getEulerAngleX (void) const
- real_type getEulerAngleY (void) const
- real_type getEulerAngleZ (void) const

6.7.1 Detailed Description

Reference frame.

6.7.2 Constructor & Destructor Documentation

6.7.2.1 ReferenceFrame()

```
TireGround::ReferenceFrame::ReferenceFrame (
            vec3 const & _Origin,
            mat3 const & _RotationMatrix )  [inline]
```

Variable set constructor.

Parameters

_Origin	Origin position
_RotationMatrix	3x3 rotation matrix

6.7.3 Member Function Documentation

6.7.3.1 getEulerAngleX()

```
real_type TireGround::ReferenceFrame::getEulerAngleX (
            void  ) const
```

Get current Euler angles [rad] for X-axis
Warning: Factor as $[R_z][R_x][R_y]$!

6.7.3.2 getEulerAngleY()

```
real_type TireGround::ReferenceFrame::getEulerAngleY (
            void  ) const
```

Get current Euler angles [rad] for Y-axis
Warning: Factor as $[R_z][R_x][R_y]$!

6.7.3.3 getEulerAngleZ()

```
real_type TireGround::ReferenceFrame::getEulerAngleZ (
            void  ) const
```

Get current Euler angles [rad] for Z-axis
Warning: Factor as $[R_z][R_x][R_y]$!

6.7.3.4 set()

```
void TireGround::ReferenceFrame::set (
            ReferenceFrame const & in )  [inline]
```

Copy the tire ReferenceFrame object
Warning: Rotation matrix must be orthonormal!

Parameters

in	ReferenceFrame object to be copied

6.7.3.5 setOrigin()

```
void TireGround::ReferenceFrame::setOrigin (
            vec3 const & _Origin )  [inline]
```

Set origin position.

Parameters

_Origin	Origin position

6.7.3.6 setRotationMatrix()

```
void TireGround::ReferenceFrame::setRotationMatrix (
            mat3 const & _RotationMatrix )  [inline]
```

Set 3x3 rotation matrix.

Parameters

_RotationMatrix	3x3 rotation matrix

6.7.3.7 setTotalTransformationMatrix()

```
void TireGround::ReferenceFrame::setTotalTransformationMatrix (
            mat4 const & TM )  [inline]
```

Set 4x4 total transformation matrix.

Parameters

TM	4x4 total transformation matrix

The documentation for this class was generated from the following file:

- include/PatchTire.hh

6.8 TireGround::SamplingGrid Class Reference

Patch evaluation precision.

`#include <PatchTire.hh>`

Public Member Functions

- SamplingGrid ()

  Default constructor.
- SamplingGrid (int_type _PointsN, int_type _DisksN)

  Variable set constructor.
- SamplingGrid (int_type _PointsN, int_type _DisksN, int_type _Switch)

  Variable set constructor.
- int_type getPointsNumber (void) const

  Get number of sampling points for each Disk (divisions on X-axis)
- int_type getDisksNumber (void) const

  Get number of Disks (divisions on Y-axis -1)
- unsigned getSwitchNumber (void) const

  Get number of maximum RoadTriangles in the Tire Shadow (switch to sampling)
- void setSwitchNumber (int_type const _Switch)

  Set number of maximum RoadTriangles in the Tire Shadow (switch to sampling)
- void set (int_type _PointsN, int_type _DisksN, int_type _Switch)

  Set number of divisions.
- void set (SamplingGrid const &in)

  Copy the SamplingGrid object.

6.8.1 Detailed Description

Patch evaluation precision.

6.8.2 Constructor & Destructor Documentation

6.8.2.1 SamplingGrid() [1/2]

```
TireGround::SamplingGrid::SamplingGrid (
            int_type _PointsN,
            int_type _DisksN )  [inline]
```

Variable set constructor.

Parameters

_PointsN	Sampling points for each Disk (divisions on X-axis)
_DisksN	Number of Disks (divisions on Y-axis -1)

6.8.2.2 SamplingGrid() [2/2]

```
TireGround::SamplingGrid::SamplingGrid (
            int_type _PointsN,
            int_type _DisksN,
            int_type _Switch )  [inline]
```

Variable set constructor.

Parameters

_PointsN	Sampling points for each Disk (divisions on X-axis)
_DisksN	Number of Disks (divisions on Y-axis -1)
_Switch	Maximum RoadTriangles in the Tire Shadow (switch to sampling)

6.8.3 Member Function Documentation

6.8.3.1 set() [1/2]

```
void TireGround::SamplingGrid::set (
            int_type _PointsN,
            int_type _DisksN,
            int_type _Switch )  [inline]
```

Set number of divisions.

Parameters

_PointsN	Sampling points for each Disk (divisions on X-axis)
_DisksN	Number of Disks (divisions on Y-axis -1)
_Switch	Maximum RoadTriangles in the Tire Shadow (switch to sampling)

6.8.3.2 set() [2/2]

```
void TireGround::SamplingGrid::set (
            SamplingGrid const & in )  [inline]
```

Copy the SamplingGrid object.

Parameters

in	SamplingGrid object to be copied

6.8.3.3 setSwitchNumber()

```
void TireGround::SamplingGrid::setSwitchNumber (
            int_type const _Switch )  [inline]
```

Set number of maximum RoadTriangles in the Tire Shadow (switch to sampling)

Parameters

_Switch	New switch number

The documentation for this class was generated from the following file:

- include/PatchTire.hh

6.9 TireGround::Shadow Class Reference

2D shadow (2D bounding box enhacement)

`#include <PatchTire.hh>`

Public Member Functions

- Shadow ()

  Default constructor.
- Shadow (ETRTO const &TireGeometry, ReferenceFrame const &RF)
- void update (ETRTO const &TireGeometry, ReferenceFrame const &RF)
- G2lib::AABBtree::PtrAABB const getAABBtree (void) const

  Get total Tire G2Lib::AABBtree (3D projection on ground)
- G2lib::AABBtree::PtrAABB const getUpperSideAABBtree (void) const

  Get upper side Tire G2Lib:AABBtree (3D projection on ground)
- G2lib::AABBtree::PtrAABB const getLowerSideAABBtree (void) const

  Get lower side Tire G2Lib:AABBtree (3D projection on ground)

6.9.1 Detailed Description

2D shadow (2D bounding box enhacement)

6.9.2 Constructor & Destructor Documentation

6.9.2.1 Shadow()

```
TireGround::Shadow::Shadow (
            ETRTO const & TireGeometry,
            ReferenceFrame const & RF )  [inline]
```

Variable set constructor
Warning: Rotation matrix must be orthonormal!

Parameters

TireGeometry	Tire ETRTO denomination
RF	Tire ReferenceFrame

6.9.3 Member Function Documentation

6.9.3.1 update()

```
void TireGround::Shadow::update (
            ETRTO const & TireGeometry,
            ReferenceFrame const & RF )
```

Update the 2D tire shadow domain
Warning: Rotation matrix must be orthonormal!

Parameters

TireGeometry	Tire ETRTO denomination
RF	Tire ReferenceFrame

The documentation for this class was generated from the following file:

- include/PatchTire.hh

6.10 TicToc Class Reference

Public Member Functions

- void **tic** ()
- void **toc** ()
- real_type **elapsed_s** () const
- real_type **elapsed_ms** () const

The documentation for this class was generated from the following file:

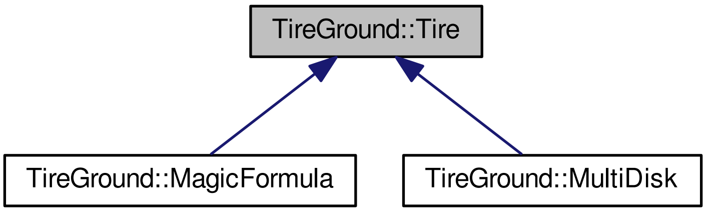
- include/TicToc.hh

6.11 TireGround::Tire Class Reference

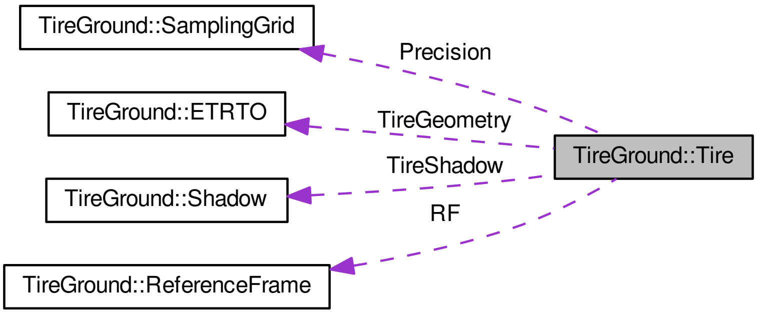
Base class for Tire models.

`#include <PatchTire.hh>`

Inheritance diagram for TireGround::Tire:



Collaboration diagram for TireGround::Tire:



Public Member Functions

- ∼Tire ()

  Default destructor.
- Tire (real_type const SectionWidth, real_type const AspectRatio, real_type const Rim←Diameter, int_type const PointsN, int_type const DisksN)

  Variable set constructor.
- void printETRTOGeometry (ostream_type &stream) const

  Display Tire ETRTO geometry data.
- G2lib::AABBtree::PtrAABB const getAABBtree (void) const

  Get total Tire Shadow G2Lib::AABBtree (3D projection on ground)
- G2lib::AABBtree::PtrAABB const getUpperSideAABBtree (void) const

  Get upper side Tire Shadow G2Lib:AABBtree (3D projection on ground)

- G2lib::AABBtree::PtrAABB const getLowerSideAABBtree (void) const

  Get lower side Tire Shadow G2Lib:AABBtree (3D projection on ground)
- void setReferenceFrame (ReferenceFrame const &_RF)
- ReferenceFrame const & getReferenceFrame (void) const

  Get tire ReferenceFrame object.
- void setOrigin (vec3 const &Origin)

  Set a new tire origin.
- void setRotationMatrix (mat3 const &RotationMatrix)
- void setTotalTransformationMatrix (mat4 const &TM)
- real_type getEulerAngleX (void) const
- real_type getEulerAngleY (void) const
- real_type getEulerAngleZ (void) const
- void getRelativeCamber (real_type &RelativeCamber) const

  Get relative camber angle [rad].
- int_type getDisksNumber (void) const

  Dimension of the contact points data structure (disks number)
- virtual void getRho (real_type &Rho, real_type &RhoDot, real_type const RhoOld, real_↩ type const Time) const =0
- virtual void getRho (row_vecN &Rho, row_vecN &RhoDot, row_vecN const RhoOld, real↩ _type const Time) const =0
- virtual void getNormal (vec3 &Normal) const =0

  Get contact normal versor.
- virtual void getNormal (row_vec3 &Normal) const =0

  Get contact normal versors vector.
- virtual void getMFpoint (vec3 &Point) const =0

  Get Magic Formula contact point.
- virtual void getMFpoint (row_vec3 &Point) const =0

  Get Magic Formula contact point vector.
- virtual void getFriction (real_type &Friction) const =0

  Get contact point friction.
- virtual void getFriction (row_vecN &Friction) const =0

  Get contact frictions vector.
- virtual void getMFpointRF (mat4 &PointRF) const =0

  Get Magic Formula contact point reference frame with 4x4 transformation matrix.
- virtual void getMFpointRF (row_mat4 &PointRF) const =0

  Get Magic Formula contact point reference frame vector with 4x4 transformation matrix.
- virtual void getArea (real_type &_Area) const =0

  Get approximated contact area on Disk plane [m^2].
- virtual void getArea (row_vecN &Area) const =0

  Get approximated contact areas vector on Disk plane [m^2].
- virtual void getVolume (real_type &Volume) const =0

  Get approximated contact volume [m^3].
- virtual void getVolume (row_vecN &_Volume) const =0

  Get approximated contact volume [m^3].
- virtual void evaluateContact (RDF::TriangleRoad_list const &TriList)=0

  Evaluate contact with RoadTriangles.
- virtual bool setup (RDF::MeshSurface &Mesh, mat4 const &TM)=0

  Update current tire position and find contact parameters.
- virtual void setup (vec3 const &Plane_Normal, vec3 const &Plane_Point, real_type const Plane_Friction, mat4 const &TM)=0
- virtual void print (ostream_type &stream) const =0

  Print contact parameters.

Protected Member Functions

- Tire (Tire const &)=delete

  Deleted copy constructor.
- Tire const & operator= (Tire const &)=delete

  Deleted copy operator.
- bool pointSampling (RDF::TriangleRoad_list const &TriList, vec3 const &RayOrigin, vec3 const &RayDirection, vec3 &SampledPt, real_type &TriFriction=quietNaN, vec3 &Tri↩ Normal=vec3_NaN) const

  Perform one point sampling (ray-triangle intersection)

Protected Attributes

- SamplingGrid Precision

  Contacth patch evaluating precision.
- ETRTO TireGeometry

  Tire ETRTO denomination.
- ReferenceFrame RF

  ReferenceFrame.
- Shadow TireShadow

  Tire shadow.

6.11.1 Detailed Description

Base class for Tire models.

6.11.2 Constructor & Destructor Documentation

6.11.2.1 Tire()

```
TireGround::Tire::Tire (
            real_type const SectionWidth,
            real_type const AspectRatio,
            real_type const RimDiameter,
            int_type const PointsN,
            int_type const DisksN )  [inline]
```

Variable set constructor.

Parameters

SectionWidth	Tire section width [m]
AspectRatio	Tire aspect ratio [$\%$]
RimDiameter	Rim diameter [in]
PointsN	Sampling points for each Disk (divisions on X-axis)
DisksN	Number of Disks (divisions on Y-axis -1)

6.11.3 Member Function Documentation

6.11.3.1 evaluateContact()

```
virtual void TireGround::Tire::evaluateContact (
            RDF::TriangleRoad_list const & TriList )  [pure virtual]
```

Evaluate contact with RoadTriangles.

Parameters

TriList	Shadow/MeshSurface intersected triangles

Implemented in TireGround::MagicFormula.

6.11.3.2 getArea() [1/2]

```
virtual void TireGround::Tire::getArea (
            real_type & _Area ) const  [pure virtual]
```

Get approximated contact area on Disk plane [m^2].

Parameters

_Area	Contact area [m^2]

Implemented in TireGround::MultiDisk, and TireGround::MagicFormula.

6.11.3.3 getArea() [2/2]

```
virtual void TireGround::Tire::getArea (
            row_vecN & Area ) const  [pure virtual]
```

Get approximated contact areas vector on Disk plane [m^2].

Parameters

Area	Contact areas vector [m^2]

Implemented in TireGround::MultiDisk, and TireGround::MagicFormula.

6.11.3.4 getEulerAngleX()

```
real_type TireGround::Tire::getEulerAngleX (
            void  ) const  [inline]
```

Get current Euler angles [rad] for X-axis
Warning: Factor as $[R_z][R_x][R_y]$!

6.11.3.5 getEulerAngleY()

```
real_type TireGround::Tire::getEulerAngleY (
            void  ) const  [inline]
```

Get current Euler angles [rad] for Y-axis
Warning: Factor as $[R_z][R_x][R_y]$!

6.11.3.6 getEulerAngleZ()

```
real_type TireGround::Tire::getEulerAngleZ (
             void ) const  [inline]
```

Get current Euler angles [rad] for Z-axis
Warning: Factor as $[R_z][R_x][R_y]$!

6.11.3.7 getFriction() [1/2]

```
virtual void TireGround::Tire::getFriction (
             real_type & Friction ) const  [pure virtual]
```

Get contact point friction.

Parameters

Friction	Contact point friction

Implemented in TireGround::MultiDisk, and TireGround::MagicFormula.

6.11.3.8 getFriction() [2/2]

```
virtual void TireGround::Tire::getFriction (
             row_vecN & Friction ) const  [pure virtual]
```

Get contact frictions vector.

Parameters

Friction	Contact frictions vector

Implemented in TireGround::MultiDisk, and TireGround::MagicFormula.

6.11.3.9 getMFpoint() [1/2]

```
virtual void TireGround::Tire::getMFpoint (
             vec3 & Point ) const  [pure virtual]
```

Get Magic Formula contact point.

Parameters

Point	Magic Formula contact point

Implemented in TireGround::MultiDisk, and TireGround::MagicFormula.

6.11.3.10 getMFpoint() [2/2]

```
virtual void TireGround::Tire::getMFpoint (
             row_vec3 & Point ) const  [pure virtual]
```

Get Magic Formula contact point vector.

Parameters

Point	Magic Formula Contact point vector

Implemented in TireGround::MultiDisk, and TireGround::MagicFormula.

6.11.3.11 getMFpointRF() [1/2]

```
virtual void TireGround::Tire::getMFpointRF (
             mat4 & PointRF ) const  [pure virtual]
```

Get Magic Formula contact point reference frame with 4x4 transformation matrix.

Parameters

PointRF	Magic Formula contact point reference frame

Implemented in TireGround::MultiDisk, and TireGround::MagicFormula.

6.11.3.12 getMFpointRF() [2/2]

```
virtual void TireGround::Tire::getMFpointRF (
             row_mat4 & PointRF ) const  [pure virtual]
```

Get Magic Formula contact point reference frame vector with 4x4 transformation matrix.

Parameters

PointRF	Magic Formula ontact point reference frames vector

Implemented in TireGround::MultiDisk, and TireGround::MagicFormula.

6.11.3.13 getNormal() [1/2]

```
virtual void TireGround::Tire::getNormal (
             vec3 & Normal ) const  [pure virtual]
```

Get contact normal versor.

Parameters

Normal	Contact point normal direction

Implemented in TireGround::MultiDisk, and TireGround::MagicFormula.

6.11.3.14 getNormal() [2/2]

```
virtual void TireGround::Tire::getNormal (
             row_vec3 & Normal ) const  [pure virtual]
```

Get contact normal versors vector.

Parameters

Normal	Contact point normal direction vector

Implemented in TireGround::MultiDisk, and TireGround::MagicFormula.

6.11.3.15 getRelativeCamber()

```
void TireGround::Tire::getRelativeCamber (
             real_type & RelativeCamber ) const
```

Get relative camber angle [rad].

Parameters

RelativeCamber	Relative camber angle

6.11.3.16 getRho() [1/2]

```
virtual void TireGround::Tire::getRho (
             real_type & Rho,
             real_type & RhoDot,
             real_type const RhoOld,
             real_type const Time ) const  [pure virtual]
```

Get contact depth at center point [m]
Warning: (if negative the tire does not touch the ground)!

Parameters

Rho	Depth at center point [m/s]
RhoDot	Contact depth derivative [m/s]
RhoOld	Previous time step Rho [m]
Time	Time step [s]

Implemented in TireGround::MultiDisk, and TireGround::MagicFormula.

6.11.3.17 getRho() [2/2]

```
virtual void TireGround::Tire::getRho (
             row_vecN & Rho,
             row_vecN & RhoDot,
             row_vecN const RhoOld,
             real_type const Time ) const  [pure virtual]
```

Get contact depth vector [m] and it time derivatives [m/s]
Warning: (if negative the tire does not touch the ground)!

Parameters

Rho	Depth matrix [m/s]

Parameters

RhoDot	Contact depth derivative matrix [m/s]
RhoOld	Previous time step Rho matrix [m]
Time	Time step [s]

Implemented in TireGround::MultiDisk, and TireGround::MagicFormula.

6.11.3.18 getVolume() [1/2]

```
virtual void TireGround::Tire::getVolume (
             real_type & Volume ) const  [pure virtual]
```

Get approximated contact volume [m^3].

Parameters

Volume	Contact volume [m^3]

Implemented in TireGround::MultiDisk, and TireGround::MagicFormula.

6.11.3.19 getVolume() [2/2]

```
virtual void TireGround::Tire::getVolume (
             row_vecN & _Volume ) const  [pure virtual]
```

Get approximated contact volume [m^3].

Parameters

_Volume	Contact volume vector [m^3]

Implemented in TireGround::MultiDisk, and TireGround::MagicFormula.

6.11.3.20 pointSampling()

```
bool TireGround::Tire::pointSampling (
             RDF::TriangleRoad_list const & TriList,
             vec3 const & RayOrigin,
             vec3 const & RayDirection,
             vec3 & SampledPt,
             real_type & TriFriction = quietNaN,
             vec3 & TriNormal = vec3_NaN ) const  [protected]
```

Perform one point sampling (ray-triangle intersection)

Parameters

TriList	Shadow/MeshSurface intersected triangles
RayOrigin	Ray origin
RayDirection	Ray direction

Parameters

SampledPt	Intersection point
TriFriction	Intersected triangle friction
TriNormal	Intersected triangle normal

6.11.3.21 print()

```
virtual void TireGround::Tire::print (
            ostream_type & stream ) const  [pure virtual]
```

Print contact parameters.

Parameters

stream	Output stream type

Implemented in TireGround::MultiDisk, and TireGround::MagicFormula.

6.11.3.22 printETRTOGeometry()

```
void TireGround::Tire::printETRTOGeometry (
            ostream_type & stream ) const  [inline]
```

Display Tire ETRTO geometry data.

Parameters

stream	Output stream type

6.11.3.23 setOrigin()

```
void TireGround::Tire::setOrigin (
            vec3 const & Origin )  [inline]
```

Set a new tire origin.

Parameters

Origin	Tire origin

6.11.3.24 setReferenceFrame()

```
void TireGround::Tire::setReferenceFrame (
            ReferenceFrame const & _RF )  [inline]
```

Copy the tire ReferenceFrame object
Warning: Rotation matrix must be orthonormal!

Parameters

_RF	ReferenceFrame object to be copied

6.11.3.25 setRotationMatrix()

```
void TireGround::Tire::setRotationMatrix (
            mat3 const & RotationMatrix )  [inline]
```

Set a new 3x3 rotation matrix
Warning: Rotation matrix must be orthonormal!

Parameters

RotationMatrix	Rotation matrix

6.11.3.26 setTotalTransformationMatrix()

```
void TireGround::Tire::setTotalTransformationMatrix (
            mat4 const & TM )  [inline]
```

Set 4x4 total transformation matrix
Warning: Rotation matrix must be orthonormal!

Parameters

TM	4x4 total transformation matrix

6.11.3.27 setup() [1/2]

```
virtual bool TireGround::Tire::setup (
            RDF::MeshSurface & Mesh,
            mat4 const & TM )  [pure virtual]
```

Update current tire position and find contact parameters.

Parameters

Mesh	MeshSurface object (road)
TM	4x4 total transformation matrix

Implemented in TireGround::MultiDisk, and TireGround::MagicFormula.

6.11.3.28 setup() [2/2]

```
virtual void TireGround::Tire::setup (
            vec3 const & Plane_Normal,
            vec3 const & Plane_Point,
            real_type const Plane_Friction,
            mat4 const & TM )  [pure virtual]
```

Update current tire position and find contact parameters with external plane

Parameters

Plane_Normal	Plane normal vector
Plane_Point	Plane known point
Plane_Friction	Friction on plane
TM	4x4 total transformation matrix

Implemented in TireGround::MultiDisk, and TireGround::MagicFormula.

The documentation for this class was generated from the following file:

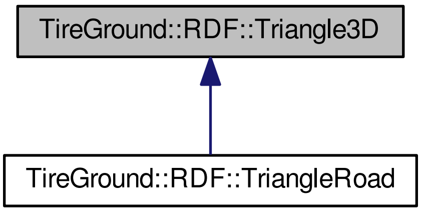
- include/PatchTire.hh

6.12 TireGround::RDF::Triangle3D Class Reference

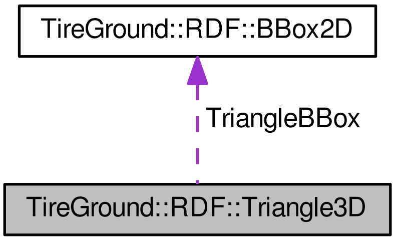
3D triangle (pure geometrical description)

```
#include <RoadRDF.hh>
```

Inheritance diagram for TireGround::RDF::Triangle3D:



Collaboration diagram for TireGround::RDF::Triangle3D:



Public Member Functions

- Triangle3D ()

  Variable set constructor.
- Triangle3D (vec3 const _Vertices[3])

  Variable set constructor.
- void setVertices (vec3 const _Vertices[3])

  Set new vertices and update bounding box domain.
- void setVertices (vec3 const &Vertex0, vec3 const &Vertex1, vec3 const &Vertex2)

  Set new vertices then update bounding box domain and normal versor.
- vec3 const & getNormal (void) const

  Get normal versor.
- vec3 const & getVertex (unsigned i) const

  Get i-th vertex.
- BBox2D const & getBBox (void) const

  Get Triangle3D bonding box BBox2D.
- void print (ostream_type &stream) const

  Print vertices data.
- bool intersectRay (vec3 const &RayOrigin, vec3 const &RayDirection, vec3 &IntPt) const
- int_type intersectEdgePlane (vec3 const &PlaneN, vec3 const &PlaneP, int_type const Edge, vec3 &IntPt1, vec3 &IntPt2) const
- bool intersectPlane (vec3 const &PlaneN, vec3 const &PlaneP, std::vector< vec3 > &IntPts) const

Protected Member Functions

- Triangle3D (Triangle3D const &)=delete

  Deleted copy constructor.
- Triangle3D & operator= (Triangle3D const &)=delete

  Deleted copy operator.

Protected Attributes

- vec3 Vertices [3]

    Vertices reference vector.
- vec3 Normal

    Triangle normal versor.
- BBox2D TriangleBBox

    Triangle 2D bounding box (XY plane)

6.12.1 Detailed Description

3D triangle (pure geometrical description)

6.12.2 Constructor & Destructor Documentation

6.12.2.1 Triangle3D()

```
TireGround::RDF::Triangle3D::Triangle3D (
             vec3 const _Vertices[3] )  [inline]
```

Variable set constructor.

Parameters

_Vertices	Vertices reference vector

6.12.3 Member Function Documentation

6.12.3.1 intersectEdgePlane()

```
int_type TireGround::RDF::Triangle3D::intersectEdgePlane (
             vec3 const & PlaneN,
             vec3 const & PlaneP,
             int_type const Edge,
             vec3 & IntPt1,
             vec3 & IntPt2 ) const
```

Check if an edge of the Triangle3D object hits a and find the intersection point

Parameters

PlaneN	Plane normal vector
PlaneP	Plane known point
Edge	Triangle edge number (0:2)
IntPt1	Intersection point 1
IntPt2	Intersection point 2

6.12.3.2 intersectPlane()

```
bool TireGround::RDF::Triangle3D::intersectPlane (
            vec3 const & PlaneN,
            vec3 const & PlaneP,
            std::vector< vec3 > & IntPts ) const
```

Check if a plane intersects a Triangle3D object and find the intersection points

Parameters

PlaneN	Plane normal vector
PlaneP	Plane known point
IntPts	Intersection points

6.12.3.3 intersectRay()

```
bool TireGround::RDF::Triangle3D::intersectRay (
            vec3 const & RayOrigin,
            vec3 const & RayDirection,
            vec3 & IntPt ) const
```

Check if a ray hits a Triangle3D object through Möller-Trumbore intersection algorithm

Parameters

RayOrigin	Ray origin position
RayDirection	Ray direction vector
IntPt	Intersection point

6.12.3.4 print()

```
void TireGround::RDF::Triangle3D::print (
            ostream_type & stream ) const  [inline]
```

Print vertices data.

Parameters

stream	Output stream type

6.12.3.5 setVertices() [1/2]

```
void TireGround::RDF::Triangle3D::setVertices (
            vec3 const _Vertices[3] )  [inline]
```

Set new vertices and update bounding box domain.

Parameters

_Vertices	Vertices reference vector

6.12.3.6 setVertices() [2/2]

```
void TireGround::RDF::Triangle3D::setVertices (
            vec3 const & Vertex0,
            vec3 const & Vertex1,
            vec3 const & Vertex2 )  [inline]
```

Set new vertices then update bounding box domain and normal versor.

Parameters

Vertex0	Vertex 1
Vertex1	Vertex 2
Vertex2	Vertex 3

The documentation for this class was generated from the following file:

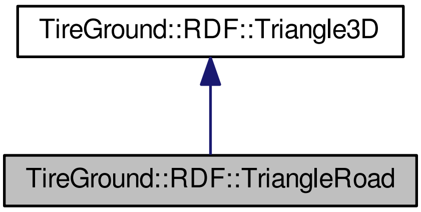
- include/RoadRDF.hh

6.13 TireGround::RDF::TriangleRoad Class Reference

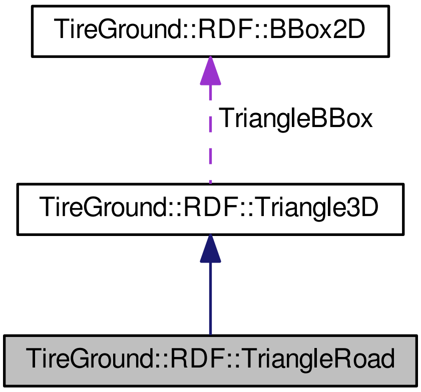
3D triangles for road representation

```
#include <RoadRDF.hh>
```

Inheritance diagram for TireGround::RDF::TriangleRoad:



Collaboration diagram for TireGround::RDF::TriangleRoad:



Public Member Functions

- TriangleRoad ()

  Default set constructor.
- TriangleRoad (vec3 const _Vertices[3], real_type _Friction)

  Variable set constructor.
- void setFriction (real_type _Friction)

  Set friction coefficient.
- real_type getFriction (void) const

  Get friction coefficent on the face.
- void setVertices (vec3 const _Vertices[3])

  Set new vertices and update bounding box domain.
- void setVertices (vec3 const &Vertex0, vec3 const &Vertex1, vec3 const &Vertex2)

  Set new vertices then update bounding box domain and normal versor.
- vec3 const & getNormal (void) const

  Get normal versor.
- vec3 const & getVertex (unsigned i) const

  Get i-th vertex.
- BBox2D const & getBBox (void) const

  Get Triangle3D bonding box BBox2D.
- void print (ostream_type &stream) const

  Print vertices data.
- bool intersectRay (vec3 const &RayOrigin, vec3 const &RayDirection, vec3 &IntPt) const
- int_type intersectEdgePlane (vec3 const &PlaneN, vec3 const &PlaneP, int_type const Edge, vec3 &IntPt1, vec3 &IntPt2) const
- bool intersectPlane (vec3 const &PlaneN, vec3 const &PlaneP, std::vector< vec3 > &IntPts) const

Protected Attributes

- vec3 Vertices [3]

  Vertices reference vector.
- vec3 Normal

  Triangle normal versor.
- BBox2D TriangleBBox

  Triangle 2D bounding box (XY plane)

6.13.1 Detailed Description

3D triangles for road representation

6.13.2 Constructor & Destructor Documentation

6.13.2.1 TriangleRoad()

```
TireGround::RDF::TriangleRoad::TriangleRoad (
             vec3 const _Vertices[3],
             real_type _Friction )  [inline]
```

Variable set constructor.

Parameters

_Vertices	Vertices reference vector
_Friction	Friction coefficient

6.13.3 Member Function Documentation

6.13.3.1 intersectEdgePlane()

```
int_type TireGround::RDF::Triangle3D::intersectEdgePlane (
             vec3 const & PlaneN,
             vec3 const & PlaneP,
             int_type const Edge,
             vec3 & IntPt1,
             vec3 & IntPt2 ) const  [inherited]
```

Check if an edge of the Triangle3D object hits a and find the intersection point

Parameters

PlaneN	Plane normal vector
PlaneP	Plane known point
Edge	Triangle edge number (0:2)
IntPt1	Intersection point 1
IntPt2	Intersection point 2

6.13.3.2 intersectPlane()

```
bool TireGround::RDF::Triangle3D::intersectPlane (
            vec3 const & PlaneN,
            vec3 const & PlaneP,
            std::vector< vec3 > & IntPts ) const  [inherited]
```

Check if a plane intersects a Triangle3D object and find the intersection points

Parameters

PlaneN	Plane normal vector
PlaneP	Plane known point
IntPts	Intersection points

6.13.3.3 intersectRay()

```
bool TireGround::RDF::Triangle3D::intersectRay (
            vec3 const & RayOrigin,
            vec3 const & RayDirection,
            vec3 & IntPt ) const  [inherited]
```

Check if a ray hits a Triangle3D object through Möller-Trumbore intersection algorithm

Parameters

RayOrigin	Ray origin position
RayDirection	Ray direction vector
IntPt	Intersection point

6.13.3.4 print()

```
void TireGround::RDF::Triangle3D::print (
            ostream_type & stream ) const  [inline], [inherited]
```

Print vertices data.

Parameters

stream	Output stream type

6.13.3.5 setFriction()

```
void TireGround::RDF::TriangleRoad::setFriction (
            real_type _Friction )  [inline]
```

Set friction coefficient.

Parameters

_Friction	New friction coefficient

6.13.3.6 setVertices() [1/2]

```
void TireGround::RDF::Triangle3D::setVertices (
            vec3 const _Vertices[3] )  [inline], [inherited]
```

Set new vertices and update bounding box domain.

Parameters

_Vertices	Vertices reference vector

6.13.3.7 setVertices() [2/2]

```
void TireGround::RDF::Triangle3D::setVertices (
            vec3 const & Vertex0,
            vec3 const & Vertex1,
            vec3 const & Vertex2 )  [inline], [inherited]
```

Set new vertices then update bounding box domain and normal versor.

Parameters

Vertex0	Vertex 1
Vertex1	Vertex 2
Vertex2	Vertex 3

The documentation for this class was generated from the following file:

- include/RoadRDF.hh

Index