Strain and stress tensors in Cartesian coordinates

This worksheet demonstrates a few capabilities of SageManifolds (version 1.0, as included in SageMath 7.5) in computations regarding elasticity theory in Cartesian coordinates.

Click here to download the worksheet file (ipynb format). To run it, you must start SageMath with the Jupyter notebook, via the command `sage -n jupyter`

NB: a version of SageMath at least equal to 7.5 is required to run this worksheet:

In [1]:
```
version()
```

Out[1]:
```
'SageMath version 7.5.1, Release Date: 2017-01-15'
```

First we set up the notebook to display mathematical objects using LaTeX rendering:

In [2]:
```
%display latex
```

Euclidean 3-space and Cartesian coordinates

We introduce the Euclidean space as a 3-dimensional differentiable manifold:

In [3]:
```
M = Manifold(3, 'M', start_index=1)
print(M)
```

```
3-dimensional differentiable manifold M
```

We then introduce the Cartesian coordinates (x, y, z) as a chart X on M:

In [4]:
```
X.<x,y,z> = M.chart()
print(X)
X
```

```
Chart (M, (x, y, z))
```

Out[4]: $(M, (x, y, z))$

The associated vector frame is

In [5]:
```
X.frame()
```

Out[5]: $\left(M, \left(\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z}\right)\right)$

We shall expand vector and tensor fields not on this frame, which is the default one on M:

In [6]:
```
M.default_frame()
```

Out[6]: $\left(M, \left(\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z}\right)\right)$

Displacement vector and strain tensor

Let us define the **displacement vector** U in terms of its components w.r.t. the orthonormal Cartesian frame:

In [7]:
```
U = M.vector_field(name='U')
U[:] = [function('U_x')(x,y,z), function('U_y')(x,y,z),
        function('U_z')(x,y,z)]
U.display()
```

Out[7]:
$$U = U_x(x, y, z)\frac{\partial}{\partial x} + U_y(x, y, z)\frac{\partial}{\partial y} + U_z(x, y, z)\frac{\partial}{\partial z}$$

The following computations will involve the metric g of the Euclidean space. At the current stage of SageManifolds, we need to introduce it explicitly, as a Riemannian metric on the manifold M (in a future version of SageManifolds, one shall to declare M as an Euclidean space, and not merely as a manifold, so that it will come equipped with g):

In [8]:
```
g = M.riemannian_metric('g')
print(g)
```

```
Riemannian metric g on the 3-dimensional differentiable manifold M
```

We initialize g by declaring that its components with respect to the frame of Cartesian coordinates are $\mathrm{diag}(1, 1, 1)$:

In [9]:
```
g[1,1], g[2,2], g[3,3] = 1, 1, 1
g.display()
```

Out[9]:
$$g = \mathrm{d}x \otimes \mathrm{d}x + \mathrm{d}y \otimes \mathrm{d}y + \mathrm{d}z \otimes \mathrm{d}z$$

The covariant derivative operator ∇ is introduced as the (Levi-Civita) connection associated with g:

In [10]:
```
nabla = g.connection()
print(nabla)
nabla
```

```
Levi-Civita connection nabla_g associated with the Riemannian metric g
on the 3-dimensional differentiable manifold M
```

Out[10]:
$$\nabla_g$$

The covariant derivative of the displacement vector U is

In [11]:
```
nabU = nabla(U)
print(nabU)
```

```
Tensor field nabla_g(U) of type (1,1) on the 3-dimensional differentiab
le manifold M
```

In [12]:
```
nabU.display()
```

Out[12]:
$$\nabla_g U = \frac{\partial U_x}{\partial x}\frac{\partial}{\partial x}\otimes \mathrm{d}x + \frac{\partial U_x}{\partial y}\frac{\partial}{\partial x}\otimes \mathrm{d}y + \frac{\partial U_x}{\partial z}\frac{\partial}{\partial x}\otimes \mathrm{d}z + \frac{\partial U_y}{\partial x}\frac{\partial}{\partial y}\otimes \mathrm{d}x + \frac{\partial U_y}{\partial y}\frac{\partial}{\partial y}\otimes \mathrm{d}y + \frac{\partial U_y}{\partial z}\frac{\partial}{\partial y}\otimes \mathrm{d}z + \frac{\partial U_z}{\partial x}\frac{\partial}{\partial z}\otimes \mathrm{d}x + \frac{\partial U_z}{\partial y}\frac{\partial}{\partial z}\otimes \mathrm{d}y + \frac{\partial U_z}{\partial z}\frac{\partial}{\partial z}\otimes \mathrm{d}z$$

We convert it to a tensor field of type (0,2) (i.e. a bilinear form) by lowering the upper index with g:

In [13]:
```
nabU_form = nabU.down(g)
print(nabU_form)
```

```
Tensor field of type (0,2) on the 3-dimensional differentiable manifold
 M
```

In [14]:
```
nabU_form.display()
```

Out[14]:

$$\frac{\partial U_x}{\partial x} \mathrm{d}x \otimes \mathrm{d}x + \frac{\partial U_x}{\partial y} \mathrm{d}x \otimes \mathrm{d}y + \frac{\partial U_x}{\partial z} \mathrm{d}x \otimes \mathrm{d}z + \frac{\partial U_y}{\partial x} \mathrm{d}y \otimes \mathrm{d}x + \frac{\partial U_y}{\partial y} \mathrm{d}y \otimes \mathrm{d}y + \frac{\partial U_y}{\partial z} \mathrm{d}y \otimes \mathrm{d}z + \frac{\partial U_z}{\partial x} \mathrm{d}z \otimes \mathrm{d}x + \frac{\partial U_z}{\partial y} \mathrm{d}z \otimes \mathrm{d}y + \frac{\partial U_z}{\partial z} \mathrm{d}z \otimes \mathrm{d}z$$

The **strain tensor** ε is defined as the symmetrized part of this tensor:

In [15]:
```
E = nabU_form.symmetrize()
print(E)
```

```
Field of symmetric bilinear forms on the 3-dimensional differentiable m
anifold M
```

In [16]:
```
E.set_name('E', latex_name=r'\varepsilon')
E.display()
```

Out[16]:

$$\varepsilon = \frac{\partial U_x}{\partial x} \mathrm{d}x \otimes \mathrm{d}x + \left(\frac{1}{2} \frac{\partial U_x}{\partial y} + \frac{1}{2} \frac{\partial U_y}{\partial x} \right) \mathrm{d}x \otimes \mathrm{d}y + \left(\frac{1}{2} \frac{\partial U_x}{\partial z} + \frac{1}{2} \frac{\partial U_z}{\partial x} \right) \mathrm{d}x \otimes \mathrm{d}z + \left(\frac{1}{2} \frac{\partial U_x}{\partial y} + \frac{1}{2} \frac{\partial U_y}{\partial x} \right) \mathrm{d}y \otimes \mathrm{d}x + \frac{\partial U_y}{\partial y} \mathrm{d}y \otimes \mathrm{d}y + \left(\frac{1}{2} \frac{\partial U_y}{\partial z} + \frac{1}{2} \frac{\partial U_z}{\partial y} \right) \mathrm{d}y \otimes \mathrm{d}z + \left(\frac{1}{2} \frac{\partial U_x}{\partial z} + \frac{1}{2} \frac{\partial U_z}{\partial x} \right) \mathrm{d}z \otimes \mathrm{d}x + \left(\frac{1}{2} \frac{\partial U_y}{\partial z} + \frac{1}{2} \frac{\partial U_z}{\partial y} \right) \mathrm{d}z \otimes \mathrm{d}y + \frac{\partial U_z}{\partial z} \mathrm{d}z \otimes \mathrm{d}z$$

Let us display the components of ε, skipping those that can be deduced by symmetry:

In [17]:
```
E.display_comp(only_nonredundant=True)
```

Out[17]:

$$\begin{array}{lcl} \varepsilon_{xx} & = & \frac{\partial U_x}{\partial x} \\ \varepsilon_{xy} & = & \frac{1}{2} \frac{\partial U_x}{\partial y} + \frac{1}{2} \frac{\partial U_y}{\partial x} \\ \varepsilon_{xz} & = & \frac{1}{2} \frac{\partial U_x}{\partial z} + \frac{1}{2} \frac{\partial U_z}{\partial x} \\ \varepsilon_{yy} & = & \frac{\partial U_y}{\partial y} \\ \varepsilon_{yz} & = & \frac{1}{2} \frac{\partial U_y}{\partial z} + \frac{1}{2} \frac{\partial U_z}{\partial y} \\ \varepsilon_{zz} & = & \frac{\partial U_z}{\partial z} \end{array}$$

Stress tensor and Hooke's law

To form the stress tensor according to Hooke's law, we introduce first the Lamé constants:

In [18]:
```
var('ll', latex_name=r'\lambda')
```

Out[18]: λ

In [19]:
```
var('mu', latex_name=r'\mu')
```

Out[19]: μ

The trace (with respect to g) of the bilinear form ε is obtained by (i) raising the first index (`pos=0`) by means of g and (ii) by taking the trace of the resulting endomorphism:

In [20]:
```
trE = E.up(g, pos=0).trace()
print(trE)
```

```
Scalar field on the 3-dimensional differentiable manifold M
```

In [21]:
```
trE.display()
```

Out[21]:
$$\begin{array}{llcl} & M & \longrightarrow & \mathbb{R} \\ & (x, y, z) & \longmapsto & \frac{\partial U_x}{\partial x} + \frac{\partial U_y}{\partial y} + \frac{\partial U_z}{\partial z} \end{array}$$

The **stress tensor** S is obtained via Hooke's law for isotropic material:

$$S = \lambda \, \mathrm{tr}\varepsilon \; g + 2\mu \, \varepsilon$$

In [22]:
```
S = ll*trE*g + 2*mu*E
print(S)
```

```
Field of symmetric bilinear forms on the 3-dimensional differentiable m
anifold M
```

In [23]:
```
S.set_name('S')
S.display()
```

Out[23]:
$$S = \left((\lambda + 2\,\mu)\frac{\partial U_x}{\partial x} + \lambda\frac{\partial U_y}{\partial y} + \lambda\frac{\partial U_z}{\partial z} \right) \mathrm{d}x \otimes \mathrm{d}x + \left(\mu\frac{\partial U_x}{\partial y} + \mu\frac{\partial U_y}{\partial x} \right) \mathrm{d}x \otimes \mathrm{d}y$$
$$+ \left(\mu\frac{\partial U_x}{\partial z} + \mu\frac{\partial U_z}{\partial x} \right) \mathrm{d}x \otimes \mathrm{d}z + \left(\mu\frac{\partial U_x}{\partial y} + \mu\frac{\partial U_y}{\partial x} \right) \mathrm{d}y \otimes \mathrm{d}x$$
$$+ \left(\lambda\frac{\partial U_x}{\partial x} + (\lambda + 2\,\mu)\frac{\partial U_y}{\partial y} + \lambda\frac{\partial U_z}{\partial z} \right) \mathrm{d}y \otimes \mathrm{d}y + \left(\mu\frac{\partial U_y}{\partial z} + \mu\frac{\partial U_z}{\partial y} \right) \mathrm{d}y \otimes \mathrm{d}z$$
$$+ \left(\mu\frac{\partial U_x}{\partial z} + \mu\frac{\partial U_z}{\partial x} \right) \mathrm{d}z \otimes \mathrm{d}x + \left(\mu\frac{\partial U_y}{\partial z} + \mu\frac{\partial U_z}{\partial y} \right) \mathrm{d}z \otimes \mathrm{d}y$$
$$+ \left(\lambda\frac{\partial U_x}{\partial x} + \lambda\frac{\partial U_y}{\partial y} + (\lambda + 2\,\mu)\frac{\partial U_z}{\partial z} \right) \mathrm{d}z \otimes \mathrm{d}z$$

In [24]:
```
S.display_comp(only_nonredundant=True)
```

Out[24]:

$$
\begin{array}{lcl}
S_{xx} & = & (\lambda + 2\mu)\frac{\partial U_x}{\partial x} + \lambda\frac{\partial U_y}{\partial y} + \lambda\frac{\partial U_z}{\partial z} \\
S_{xy} & = & \mu\frac{\partial U_x}{\partial y} + \mu\frac{\partial U_y}{\partial x} \\
S_{xz} & = & \mu\frac{\partial U_x}{\partial z} + \mu\frac{\partial U_z}{\partial x} \\
S_{yy} & = & \lambda\frac{\partial U_x}{\partial x} + (\lambda + 2\mu)\frac{\partial U_y}{\partial y} + \lambda\frac{\partial U_z}{\partial z} \\
S_{yz} & = & \mu\frac{\partial U_y}{\partial z} + \mu\frac{\partial U_z}{\partial y} \\
S_{zz} & = & \lambda\frac{\partial U_x}{\partial x} + \lambda\frac{\partial U_y}{\partial y} + (\lambda + 2\mu)\frac{\partial U_z}{\partial z}
\end{array}
$$

Each component can be accessed individually:

In [25]:
```
S[1,2]
```

Out[25]:

$$\mu\frac{\partial U_x}{\partial y} + \mu\frac{\partial U_y}{\partial x}$$

Divergence of the stress tensor

The divergence of the stress tensor is the 1-form:

$$f_i = \nabla_j S^j{}_i$$

In a next version of SageManifolds, there will be a function `divergence()`. For the moment, to evaluate f, we first form the tensor $S^j{}_i$ by raising the first index (`pos=0`) of S with g:

In [26]:
```
SU = S.up(g, pos=0)
print(SU)
```

```
Tensor field of type (1,1) on the 3-dimensional differentiable manifold
 M
```

The divergence is obtained by taking the trace on the first index (`0`) and the third one (`2`) of the tensor $(\nabla S)^j{}_{ik} = \nabla_k S^j{}_i$:

In [27]:
```
divS = nabla(SU).trace(0,2)
print(divS)
```

```
1-form on the 3-dimensional differentiable manifold M
```

In [28]:
```
divS.set_name('f')
divS.display()
```

Out[28]:

$$
\begin{aligned}
f = & \left((\lambda + 2\mu)\frac{\partial^2 U_x}{\partial x^2} + \mu\frac{\partial^2 U_x}{\partial y^2} + \mu\frac{\partial^2 U_x}{\partial z^2} + (\lambda + \mu)\frac{\partial^2 U_y}{\partial x\partial y} + (\lambda + \mu)\frac{\partial^2 U_z}{\partial x\partial z}\right) \mathrm{d}x \\
& + \left((\lambda + \mu)\frac{\partial^2 U_x}{\partial x\partial y} + \mu\frac{\partial^2 U_y}{\partial x^2} + (\lambda + 2\mu)\frac{\partial^2 U_y}{\partial y^2} + \mu\frac{\partial^2 U_y}{\partial z^2} + (\lambda + \mu)\frac{\partial^2 U_z}{\partial y\partial z}\right) \mathrm{d}y \\
& + \left((\lambda + \mu)\frac{\partial^2 U_x}{\partial x\partial z} + (\lambda + \mu)\frac{\partial^2 U_y}{\partial y\partial z} + \mu\frac{\partial^2 U_z}{\partial x^2} + \mu\frac{\partial^2 U_z}{\partial y^2} + (\lambda + 2\mu)\frac{\partial^2 U_z}{\partial z^2}\right) \mathrm{d}z
\end{aligned}
$$

In [29]:
```
divS.display_comp()
```

Out[29]:

$$f_x = (\lambda + 2\mu)\frac{\partial^2 U_x}{\partial x^2} + \mu\frac{\partial^2 U_x}{\partial y^2} + \mu\frac{\partial^2 U_x}{\partial z^2} + (\lambda + \mu)\frac{\partial^2 U_y}{\partial x\partial y} + (\lambda + \mu)\frac{\partial^2 U_z}{\partial x\partial z}$$

$$f_y = (\lambda + \mu)\frac{\partial^2 U_x}{\partial x\partial y} + \mu\frac{\partial^2 U_y}{\partial x^2} + (\lambda + 2\mu)\frac{\partial^2 U_y}{\partial y^2} + \mu\frac{\partial^2 U_y}{\partial z^2} + (\lambda + \mu)\frac{\partial^2 U_z}{\partial y\partial z}$$

$$f_z = (\lambda + \mu)\frac{\partial^2 U_x}{\partial x\partial z} + (\lambda + \mu)\frac{\partial^2 U_y}{\partial y\partial z} + \mu\frac{\partial^2 U_z}{\partial x^2} + \mu\frac{\partial^2 U_z}{\partial y^2} + (\lambda + 2\mu)\frac{\partial^2 U_z}{\partial z^2}$$

Displaying the components one by one:

In [30]:
```
divS[1]
```

Out[30]:

$$(\lambda + 2\mu)\frac{\partial^2 U_x}{\partial x^2} + \mu\frac{\partial^2 U_x}{\partial y^2} + \mu\frac{\partial^2 U_x}{\partial z^2} + (\lambda + \mu)\frac{\partial^2 U_y}{\partial x\partial y} + (\lambda + \mu)\frac{\partial^2 U_z}{\partial x\partial z}$$

In [31]:
```
divS[2]
```

Out[31]:

$$(\lambda + \mu)\frac{\partial^2 U_x}{\partial x\partial y} + \mu\frac{\partial^2 U_y}{\partial x^2} + (\lambda + 2\mu)\frac{\partial^2 U_y}{\partial y^2} + \mu\frac{\partial^2 U_y}{\partial z^2} + (\lambda + \mu)\frac{\partial^2 U_z}{\partial y\partial z}$$

In [32]:
```
divS[3]
```

Out[32]:

$$(\lambda + \mu)\frac{\partial^2 U_x}{\partial x\partial z} + (\lambda + \mu)\frac{\partial^2 U_y}{\partial y\partial z} + \mu\frac{\partial^2 U_z}{\partial x^2} + \mu\frac{\partial^2 U_z}{\partial y^2} + (\lambda + 2\mu)\frac{\partial^2 U_z}{\partial z^2}$$

In [ ]:
```

```