The purpose of this Maple worksheet is to show how to use *OreModules* in order to check some well-known identities of differential operators appearing in mathematical physics.

For instance, see pages 163-164 of *A Mathematical Introduction to Fluid Mechanics*, A. J. Chorin, J. E. Marsden, 3rd edition, Texts in Applied Mathematics 4, Springer, 1993.

In order to do that, we first define the Weyl algebra A_3 of differential operators with polynomial coefficients.

```
> Alg := DefineOreAlgebra(diff=[D1,x1], diff=[D2,x2], diff=[D3,x3],
> polynom=[x1,x2,x3]):
```

Let us define the gradient, the curl and the divergence operators in $\mathbb{R}^3$.

```
> grad := evalm([[D1], [D2], [D3]]);
```

$$grad := \begin{bmatrix} \mathrm{D1} \\ \mathrm{D2} \\ \mathrm{D3} \end{bmatrix}$$

```
> curl := evalm([[0, -D3, D2], [D3, 0, -D1], [-D2, D1, 0]]);
```

$$curl := \begin{bmatrix} 0 & -\mathrm{D3} & \mathrm{D2} \\ \mathrm{D3} & 0 & -\mathrm{D1} \\ -\mathrm{D2} & \mathrm{D1} & 0 \end{bmatrix}$$

```
> div := evalm([[D1,D2,D3]]);
```

$$div := \begin{bmatrix} \mathrm{D1} & \mathrm{D2} & \mathrm{D3} \end{bmatrix}$$

In order to deal simultaneously with these three differential operators, we put them into a list.

```
> DiffOps := [grad, curl, div];
```

$$DiffOps := [grad,\ curl,\ div]$$

We compute the rank of these differential operators by using *OreRank*:

```
> map(OreRank, DiffOps, Alg);
```

$$[0,\ 1,\ 2]$$

This means that the smooth solutions of these differential operators in $\mathbb{R}^3$ depend respectively on 0, 1 and 2 arbitrary functions in x_1, x_2, x_3.

Now, we can check whether or not these differential operators are parametrizable, i.e., whether or not we can parametrize all their smooth functions in an open convex subset of $\mathbb{R}^3$ by means of arbitrary functions in x_1, x_2, x_3. As it is explained in the literature, we first need to compute the formal adjoints of these differential operators.

```
> Adj := map(Involution, DiffOps, Alg);
```

$$Adj := \left[\begin{bmatrix} -\mathrm{D1} & -\mathrm{D2} & -\mathrm{D3} \end{bmatrix},\ \begin{bmatrix} 0 & -\mathrm{D3} & \mathrm{D2} \\ \mathrm{D3} & 0 & -\mathrm{D1} \\ -\mathrm{D2} & \mathrm{D1} & 0 \end{bmatrix},\ \begin{bmatrix} -\mathrm{D1} \\ -\mathrm{D2} \\ -\mathrm{D3} \end{bmatrix}\right]$$

Then, we need to compute the first extension module with values in *Alg* of their formal adjoints:

```
> Ext1 := map(Exti, Adj, Alg, 1);
```

$$Ext1 := \left[\left[\left[\begin{array}{c} D3 \\ D2 \\ D1 \end{array}\right], \left[\begin{array}{c} 1 \end{array}\right], \mathrm{SURJ}(1)\right], \left[\left[\begin{array}{ccc} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{array}\right], \left[\begin{array}{ccc} -D3 & 0 & D1 \\ -D2 & D1 & 0 \\ 0 & -D3 & D2 \end{array}\right], \left[\begin{array}{c} -D1 \\ -D2 \\ -D3 \end{array}\right]\right],\right.$$
$$\left.\left[\left[\begin{array}{c} 1 \end{array}\right], \left[\begin{array}{ccc} D1 & D2 & D3 \end{array}\right], \left[\begin{array}{ccc} D3 & D2 & 0 \\ 0 & -D1 & D3 \\ -D1 & 0 & -D2 \end{array}\right]\right]\right]$$

```
> Ext1[1][1];
```

$$\left[\begin{array}{c} D3 \\ D2 \\ D1 \end{array}\right]$$

As the first matrix *Ext1*[1][1] of *Ext1*[1] is not the identity matrix, we obtain that the gradient is not parametrizable. Equivalently, we know that the gradient operator defines a torsion A_3-module. The torsion elements of the system $grad\ \theta_1 = 0$ can be directly computed by using *TorsionElements*:

```
> TorsionElements(DiffOps[1], [xi(x1,x2,x3)], Alg);
```

$$\left[\left[\begin{array}{c} \frac{\partial}{\partial x3}\,\theta_1(x1,\,x2,\,x3) = 0 \\ \frac{\partial}{\partial x2}\,\theta_1(x1,\,x2,\,x3) = 0 \\ \frac{\partial}{\partial x1}\,\theta_1(x1,\,x2,\,x3) = 0 \end{array}\right], \left[\begin{array}{c} \theta_1(x1,\,x2,\,x3) = \xi(x1,\,x2,\,x3) \end{array}\right]\right]$$

Therefore, we obtain that θ_1 is killed by the differential operators D1, D2 and D3, which is clear from the definition of the gradient operator. Therefore, the solutions of the gradient operator are locally constant.

As the first matrix *Ext1*[2][1] (resp., *Ext1*[3][1]) is the identity matrix, we see that the curl and the divergence operators are parametrizable and they define torsion-free A_3-modules. This last point can be easily checked by computing the torsion elements of their corresponding modules.

```
> TorsionElements(DiffOps[2], [seq(eta[i](x1,x2,x3), i=1..3)], Alg);
> TorsionElements(DiffOps[3], [seq(eta[i](x1,x2,x3), i=1..3)], Alg);
```

$$[]$$
$$[]$$

Using the fact that the curl and the divergence operators define two torsion-free A_3-modules, we obtain that *Ext1*[2][3] (resp., *Ext1*[3][3]) is a parametrization of the curl (resp., divergence) operator. This result can be checked directly by using *Parametrization*:

```
> map(Parametrization, [curl, div], Alg);
```

$$\left[\left[\begin{array}{c} -(\frac{\partial}{\partial x1}\,\xi_1(x1,\,x2,\,x3)) \\ -(\frac{\partial}{\partial x2}\,\xi_1(x1,\,x2,\,x3)) \\ -(\frac{\partial}{\partial x3}\,\xi_1(x1,\,x2,\,x3)) \end{array}\right], \left[\begin{array}{c} (\frac{\partial}{\partial x3}\,\xi_1(x1,\,x2,\,x3)) + (\frac{\partial}{\partial x2}\,\xi_2(x1,\,x2,\,x3)) \\ -(\frac{\partial}{\partial x1}\,\xi_2(x1,\,x2,\,x3)) + (\frac{\partial}{\partial x3}\,\xi_3(x1,\,x2,\,x3)) \\ -(\frac{\partial}{\partial x1}\,\xi_1(x1,\,x2,\,x3)) - (\frac{\partial}{\partial x2}\,\xi_3(x1,\,x2,\,x3)) \end{array}\right]\right]$$

Up to a sign, we obtain that the curl (resp., divergence) operator is parametrized by the gradient (resp., curl) operator. We check that the divergence operator is parametrized by means of 3 arbitrary functions ξ_1, ξ_2 and ξ_3 whereas it is defined by 1 equation in 3 unknowns. Therefore, we can wonder if there exist some minimal parametrizations of the divergence operator, i.e., parametrizations depending only on 2 arbitrary functions. We know that it is possible as the divergence defines a torsion-free module (see J.-F. Pommaret, A. Quadrat, *Localization and parametrization of linear multidimensional systems*, Systems & Control Letters, 37 (1999), pp. 247-260). Minimal parametrizations of the divergence operator can be obtained using *MimimalParametrizations*.

```
> Ps := MinimalParametrizations(div, Alg);
```

$$Ps := \left[\begin{bmatrix} \mathrm{D3} & \mathrm{D2} \\ 0 & -\mathrm{D1} \\ -\mathrm{D1} & 0 \end{bmatrix}, \begin{bmatrix} \mathrm{D3} & 0 \\ 0 & \mathrm{D3} \\ -\mathrm{D1} & -\mathrm{D2} \end{bmatrix}, \begin{bmatrix} \mathrm{D2} & 0 \\ -\mathrm{D1} & \mathrm{D3} \\ 0 & -\mathrm{D2} \end{bmatrix} \right]$$

Therefore, we obtain

$$\begin{array}{lcl} div\, z = 0 & \Longleftrightarrow & z = Ps[1]\,(\xi_1, \xi_2)^T, \\ div\, z = 0 & \Longleftrightarrow & z = Ps[2]\,(\xi_1, \xi_2)^T, \\ div\, z = 0 & \Longleftrightarrow & z = Ps[3]\,(\xi_1, \xi_2)^T, \end{array}$$

where z and $(\xi_1, \xi_2)^T$ are two smooth vectors in an open convex subset of $\mathbb{R}^3$.

Finally, let us define the Laplacian Δ.

```
> Delta := Mult(div, grad, Alg);
```

$$\Delta := \left[\ \mathrm{D1}^2 + \mathrm{D2}^2 + \mathrm{D3}^2\ \right]$$

Now, let us check the classical identity *curl curl* = *grad div* - Δ I_3. We first compute the composition *curl curl*:

```
> curlcurl := Mult(curl, curl, Alg);
```

$$curlcurl := \begin{bmatrix} -\mathrm{D3}^2 - \mathrm{D2}^2 & \mathrm{D2}\,\mathrm{D1} & \mathrm{D3}\,\mathrm{D1} \\ \mathrm{D2}\,\mathrm{D1} & -\mathrm{D3}^2 - \mathrm{D1}^2 & \mathrm{D3}\,\mathrm{D2} \\ \mathrm{D3}\,\mathrm{D1} & \mathrm{D3}\,\mathrm{D2} & -\mathrm{D2}^2 - \mathrm{D1}^2 \end{bmatrix}$$

Then, we compute the composition *grad div*:

```
> graddiv := Mult(grad, div, Alg);
```

$$graddiv := \begin{bmatrix} \mathrm{D1}^2 & \mathrm{D2}\,\mathrm{D1} & \mathrm{D3}\,\mathrm{D1} \\ \mathrm{D2}\,\mathrm{D1} & \mathrm{D2}^2 & \mathrm{D3}\,\mathrm{D2} \\ \mathrm{D3}\,\mathrm{D1} & \mathrm{D3}\,\mathrm{D2} & \mathrm{D3}^2 \end{bmatrix}$$

Finally, let us define $\Delta\, I_3$:

```
> nablaI3 := linalg[diag](Delta$3);
```

$$nablaI3 := \begin{bmatrix} \mathrm{D1}^2 + \mathrm{D2}^2 + \mathrm{D3}^2 & 0 & 0 \\ 0 & \mathrm{D1}^2 + \mathrm{D2}^2 + \mathrm{D3}^2 & 0 \\ 0 & 0 & \mathrm{D1}^2 + \mathrm{D2}^2 + \mathrm{D3}^2 \end{bmatrix}$$

Then, *curl curl* $-$ *grad div* $+ \Delta\, I_3$ gives

```
> evalm(curlcurl-graddiv+nablaI3);
```

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

which proves the identity.

Let us prove the identity $\Delta\ (f\,g) = \mathrm{f}\ \Delta\, g\ +\ g\,\Delta\, f\ +\ 2\ (grad\, f)\ .\ (grad\, g)$, where f and g are two functions of x_1, x_2 and x_3 and "." denotes the standard inner product in $\mathbb{R}^3$.

We first define *grad* f and *grad* g.

```
> gradf := ApplyMatrix(grad, [f(x1,x2,x3)], Alg);
```

$$gradf := \begin{bmatrix} \frac{\partial}{\partial x1}\,\mathrm{f}(x1,\, x2,\, x3) \\ \frac{\partial}{\partial x2}\,\mathrm{f}(x1,\, x2,\, x3) \\ \frac{\partial}{\partial x3}\,\mathrm{f}(x1,\, x2,\, x3) \end{bmatrix}$$

```
> gradg := ApplyMatrix(grad, [g(x1,x2,x3)], Alg);
```

$$gradg := \begin{bmatrix} \frac{\partial}{\partial x1}\,\mathrm{g}(x1,\, x2,\, x3) \\ \frac{\partial}{\partial x2}\,\mathrm{g}(x1,\, x2,\, x3) \\ \frac{\partial}{\partial x3}\,\mathrm{g}(x1,\, x2,\, x3) \end{bmatrix}$$

Then, let us compute $(grad\, f)\ .\ (grad\, g)$:

```
> gradfgradg := linalg[innerprod](
> convert(ApplyMatrix(grad, [f(x1,x2,x3)], Alg), vector),
> convert(ApplyMatrix(grad, [g(x1,x2,x3)], Alg), vector));
```

$$gradfgradg := (\tfrac{\partial}{\partial x1}\,\mathrm{f}(x1,\, x2,\, x3))\,(\tfrac{\partial}{\partial x1}\,\mathrm{g}(x1,\, x2,\, x3)) \\ +\,(\tfrac{\partial}{\partial x2}\,\mathrm{f}(x1,\, x2,\, x3))\,(\tfrac{\partial}{\partial x2}\,\mathrm{g}(x1,\, x2,\, x3)) + (\tfrac{\partial}{\partial x3}\,\mathrm{f}(x1,\, x2,\, x3))\,(\tfrac{\partial}{\partial x3}\,\mathrm{g}(x1,\, x2,\, x3))$$

We compute $\Delta(fg)$:

```
> Deltafg := ApplyMatrix(Delta, [f(x1,x2,x3)*g(x1,x2,x3)], Alg);
```

$$Deltafg := \\ \Big[(\tfrac{\partial^2}{\partial x1^2}\,\mathrm{f}(x1,\, x2,\, x3))\,\mathrm{g}(x1,\, x2,\, x3) + 2\,(\tfrac{\partial}{\partial x1}\,\mathrm{f}(x1,\, x2,\, x3))\,(\tfrac{\partial}{\partial x1}\,\mathrm{g}(x1,\, x2,\, x3)) \\ +\,\mathrm{f}(x1,\, x2,\, x3)\,(\tfrac{\partial^2}{\partial x1^2}\,\mathrm{g}(x1,\, x2,\, x3)) + (\tfrac{\partial^2}{\partial x2^2}\,\mathrm{f}(x1,\, x2,\, x3))\,\mathrm{g}(x1,\, x2,\, x3) \\ +\,2\,(\tfrac{\partial}{\partial x2}\,\mathrm{f}(x1,\, x2,\, x3))\,(\tfrac{\partial}{\partial x2}\,\mathrm{g}(x1,\, x2,\, x3)) + \mathrm{f}(x1,\, x2,\, x3)\,(\tfrac{\partial^2}{\partial x2^2}\,\mathrm{g}(x1,\, x2,\, x3)) \\ +\,(\tfrac{\partial^2}{\partial x3^2}\,\mathrm{f}(x1,\, x2,\, x3))\,\mathrm{g}(x1,\, x2,\, x3) + 2\,(\tfrac{\partial}{\partial x3}\,\mathrm{f}(x1,\, x2,\, x3))\,(\tfrac{\partial}{\partial x3}\,\mathrm{g}(x1,\, x2,\, x3)) \\ +\,\mathrm{f}(x1,\, x2,\, x3)\,(\tfrac{\partial^2}{\partial x3^2}\,\mathrm{g}(x1,\, x2,\, x3))\Big]$$

Finally, we compute $f\,\Delta\,g$ and $g\,\Delta\,f$:

```
> fDeltag := f(x1,x2,x3)*ApplyMatrix(Delta, [g(x1,x2,x3)], Alg);
```

$$fDeltag := \mathrm{f}(x1,\, x2,\, x3)\,\begin{bmatrix} (\tfrac{\partial^2}{\partial x1^2}\,\mathrm{g}(x1,\, x2,\, x3)) + (\tfrac{\partial^2}{\partial x2^2}\,\mathrm{g}(x1,\, x2,\, x3)) + (\tfrac{\partial^2}{\partial x3^2}\,\mathrm{g}(x1,\, x2,\, x3)) \end{bmatrix}$$

```
> gDeltaf := g(x1,x2,x3)*ApplyMatrix(Delta, [f(x1,x2,x3)], Alg);
```

$$gDeltaf := \mathrm{g}(x1,\, x2,\, x3)\,\begin{bmatrix} (\tfrac{\partial^2}{\partial x1^2}\,\mathrm{f}(x1,\, x2,\, x3)) + (\tfrac{\partial^2}{\partial x2^2}\,\mathrm{f}(x1,\, x2,\, x3)) + (\tfrac{\partial^2}{\partial x3^2}\,\mathrm{f}(x1,\, x2,\, x3)) \end{bmatrix}$$

Then, the identity directly follows from:

```
> simplify(evalm(Deltafg-fDeltag-gDeltaf-2*gradfgradg));
```

$$\begin{bmatrix} 0 \end{bmatrix}$$

Let f be a function of $x = (x1,\, x2,\, x3)$ and $F(x) = (F_1(x), F_2(x), F_3(x))^T$ a vector of functions. Then, let us prove the identity $curl(f\,F) = f\,curlF + grad\,f \wedge F$.

Let us first define the vector F and $grad\,f$:

```
> vecF := evalm([seq([F[i](x1,x2,x3)], i=1..3)]);
```

$$vecF := \begin{bmatrix} F_1(x1, x2, x3) \\ F_2(x1, x2, x3) \\ F_3(x1, x2, x3) \end{bmatrix}$$

```
> gradf := ApplyMatrix(grad, evalm([[f(x1,x2,x3)]]), Alg);
```

$$gradf := \begin{bmatrix} \frac{\partial}{\partial x1} \mathrm{f}(x1, x2, x3) \\ \frac{\partial}{\partial x2} \mathrm{f}(x1, x2, x3) \\ \frac{\partial}{\partial x3} \mathrm{f}(x1, x2, x3) \end{bmatrix}$$

Then, we compute $curl(f\,F)$ and $f\,curl\,F$:

```
> curlfF := ApplyMatrix(curl, evalm(f(x1,x2,x3)*vecF), Alg);
```

$$curlfF :=$$

$$\Big[-(\tfrac{\partial}{\partial x3} \mathrm{f}(x1, x2, x3))\, F_2(x1, x2, x3) - \mathrm{f}(x1, x2, x3)\, (\tfrac{\partial}{\partial x3} F_2(x1, x2, x3))$$
$$+ (\tfrac{\partial}{\partial x2} \mathrm{f}(x1, x2, x3))\, F_3(x1, x2, x3) + \mathrm{f}(x1, x2, x3)\, (\tfrac{\partial}{\partial x2} F_3(x1, x2, x3))\Big]$$

$$\Big[(\tfrac{\partial}{\partial x3} \mathrm{f}(x1, x2, x3))\, F_1(x1, x2, x3) + \mathrm{f}(x1, x2, x3)\, (\tfrac{\partial}{\partial x3} F_1(x1, x2, x3))$$
$$- (\tfrac{\partial}{\partial x1} \mathrm{f}(x1, x2, x3))\, F_3(x1, x2, x3) - \mathrm{f}(x1, x2, x3)\, (\tfrac{\partial}{\partial x1} F_3(x1, x2, x3))\Big]$$

$$\Big[-(\tfrac{\partial}{\partial x2} \mathrm{f}(x1, x2, x3))\, F_1(x1, x2, x3) - \mathrm{f}(x1, x2, x3)\, (\tfrac{\partial}{\partial x2} F_1(x1, x2, x3))$$
$$+ (\tfrac{\partial}{\partial x1} \mathrm{f}(x1, x2, x3))\, F_2(x1, x2, x3) + \mathrm{f}(x1, x2, x3)\, (\tfrac{\partial}{\partial x1} F_2(x1, x2, x3))\Big]$$

```
> fcurlF := f(x1,x2,x3)*ApplyMatrix(curl, vecF, Alg);
```

$$fcurlF := \mathrm{f}(x1, x2, x3) \begin{bmatrix} -(\frac{\partial}{\partial x3} F_2(x1, x2, x3)) + (\frac{\partial}{\partial x2} F_3(x1, x2, x3)) \\ (\frac{\partial}{\partial x3} F_1(x1, x2, x3)) - (\frac{\partial}{\partial x1} F_3(x1, x2, x3)) \\ -(\frac{\partial}{\partial x2} F_1(x1, x2, x3)) + (\frac{\partial}{\partial x1} F_2(x1, x2, x3)) \end{bmatrix}$$

Finally, let us compute $grad \wedge F$:

```
> DeltafxF := convert(linalg[crossprod](convert(gradf, vector),
> convert(vecF, vector)), matrix);
```

$$DeltafxF := \begin{bmatrix} (\frac{\partial}{\partial x2} \mathrm{f}(x1, x2, x3))\, F_3(x1, x2, x3) - (\frac{\partial}{\partial x3} \mathrm{f}(x1, x2, x3))\, F_2(x1, x2, x3) \\ (\frac{\partial}{\partial x3} \mathrm{f}(x1, x2, x3))\, F_1(x1, x2, x3) - (\frac{\partial}{\partial x1} \mathrm{f}(x1, x2, x3))\, F_3(x1, x2, x3) \\ (\frac{\partial}{\partial x1} \mathrm{f}(x1, x2, x3))\, F_2(x1, x2, x3) - (\frac{\partial}{\partial x2} \mathrm{f}(x1, x2, x3))\, F_1(x1, x2, x3) \end{bmatrix}$$

Then, we easily check the previous identity by computing:

```
> map(simplify, evalm(curlfF-fcurlF-DeltafxF));
```

$$\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

Finally, let $x = (x1, x2, x3)$ and $F(x) = (F_1(x), F_2(x), F_3(x))^T$ and $G(x) = (G_1(x), G_2(x), G_3(x))^T$ be two vectors of $\mathbb{R}^3$. Let us prove the identity $curl(F \times G) = F\, div\, G - G\, div\, F$+($G$. $grad$) F - (F . $grad$) G, where "." denotes the standard inner product in $\mathbb{R}^3$.

```
> Alg2 := DefineOreAlgebra(diff=[D1,x1], diff=[D2,x2], diff=[D3,x3],
> func=[F[1],F[2],F[3],G[1],G[2],G[3]]):
```

Let us first define the two vectors F and G.

```
> vecF := evalm([seq([F[i](x1,x2,x3)], i=1..3)]);
```

$$vecF := \begin{bmatrix} F_1(x1,\, x2,\, x3) \\ F_2(x1,\, x2,\, x3) \\ F_3(x1,\, x2,\, x3) \end{bmatrix}$$

```
> vecG := evalm([seq([G[i](x1,x2,x3)], i=1..3)]);
```

$$vecG := \begin{bmatrix} G_1(x1,\, x2,\, x3) \\ G_2(x1,\, x2,\, x3) \\ G_3(x1,\, x2,\, x3) \end{bmatrix}$$

Then, let us compute the wedge product $F \wedge G$ and $curl(F \wedge G)$:

```
> vecFxvecG := convert(linalg[crossprod](convert(vecF, vector),
> convert(vecG, vector)), matrix);
```

$$vecFxvecG := \begin{bmatrix} F_2(x1,\, x2,\, x3)\, G_3(x1,\, x2,\, x3) - F_3(x1,\, x2,\, x3)\, G_2(x1,\, x2,\, x3) \\ F_3(x1,\, x2,\, x3)\, G_1(x1,\, x2,\, x3) - F_1(x1,\, x2,\, x3)\, G_3(x1,\, x2,\, x3) \\ F_1(x1,\, x2,\, x3)\, G_2(x1,\, x2,\, x3) - F_2(x1,\, x2,\, x3)\, G_1(x1,\, x2,\, x3) \end{bmatrix}$$

```
> curlFxG := ApplyMatrix(curl, vecFxvecG, Alg2);
```

$$curlFxG :=$$

$$\Big[- (\tfrac{\partial}{\partial x3}\, F_3(x1,\, x2,\, x3))\, G_1(x1,\, x2,\, x3) - F_3(x1,\, x2,\, x3)\, (\tfrac{\partial}{\partial x3}\, G_1(x1,\, x2,\, x3))$$
$$+ (\tfrac{\partial}{\partial x3}\, F_1(x1,\, x2,\, x3))\, G_3(x1,\, x2,\, x3) + F_1(x1,\, x2,\, x3)\, (\tfrac{\partial}{\partial x3}\, G_3(x1,\, x2,\, x3))$$
$$+ (\tfrac{\partial}{\partial x2}\, F_1(x1,\, x2,\, x3))\, G_2(x1,\, x2,\, x3) + F_1(x1,\, x2,\, x3)\, (\tfrac{\partial}{\partial x2}\, G_2(x1,\, x2,\, x3))$$
$$- (\tfrac{\partial}{\partial x2}\, F_2(x1,\, x2,\, x3))\, G_1(x1,\, x2,\, x3) - F_2(x1,\, x2,\, x3)\, (\tfrac{\partial}{\partial x2}\, G_1(x1,\, x2,\, x3))\Big]$$

$$\Big[(\tfrac{\partial}{\partial x3}\, F_2(x1,\, x2,\, x3))\, G_3(x1,\, x2,\, x3) + F_2(x1,\, x2,\, x3)\, (\tfrac{\partial}{\partial x3}\, G_3(x1,\, x2,\, x3))$$
$$- (\tfrac{\partial}{\partial x3}\, F_3(x1,\, x2,\, x3))\, G_2(x1,\, x2,\, x3) - F_3(x1,\, x2,\, x3)\, (\tfrac{\partial}{\partial x3}\, G_2(x1,\, x2,\, x3))$$
$$- (\tfrac{\partial}{\partial x1}\, F_1(x1,\, x2,\, x3))\, G_2(x1,\, x2,\, x3) - F_1(x1,\, x2,\, x3)\, (\tfrac{\partial}{\partial x1}\, G_2(x1,\, x2,\, x3))$$
$$+ (\tfrac{\partial}{\partial x1}\, F_2(x1,\, x2,\, x3))\, G_1(x1,\, x2,\, x3) + F_2(x1,\, x2,\, x3)\, (\tfrac{\partial}{\partial x1}\, G_1(x1,\, x2,\, x3))\Big]$$

$$\Big[- (\tfrac{\partial}{\partial x2}\, F_2(x1,\, x2,\, x3))\, G_3(x1,\, x2,\, x3) - F_2(x1,\, x2,\, x3)\, (\tfrac{\partial}{\partial x2}\, G_3(x1,\, x2,\, x3))$$
$$+ (\tfrac{\partial}{\partial x2}\, F_3(x1,\, x2,\, x3))\, G_2(x1,\, x2,\, x3) + F_3(x1,\, x2,\, x3)\, (\tfrac{\partial}{\partial x2}\, G_2(x1,\, x2,\, x3))$$
$$+ (\tfrac{\partial}{\partial x1}\, F_3(x1,\, x2,\, x3))\, G_1(x1,\, x2,\, x3) + F_3(x1,\, x2,\, x3)\, (\tfrac{\partial}{\partial x1}\, G_1(x1,\, x2,\, x3))$$
$$- (\tfrac{\partial}{\partial x1}\, F_1(x1,\, x2,\, x3))\, G_3(x1,\, x2,\, x3) - F_1(x1,\, x2,\, x3)\, (\tfrac{\partial}{\partial x1}\, G_3(x1,\, x2,\, x3))\Big]$$

Let us compute $F\, div\, G$ and $G\, div\, F$:

```
> FdivG := evalm(vecF*ApplyMatrix(div, vecG, Alg2)[1,1]);
```

$$FdivG := \begin{bmatrix} \%1\, F_1(x1,\, x2,\, x3) \\ \%1\, F_2(x1,\, x2,\, x3) \\ \%1\, F_3(x1,\, x2,\, x3) \end{bmatrix}$$

$$\%1 := (\tfrac{\partial}{\partial x1}\, G_1(x1,\, x2,\, x3)) + (\tfrac{\partial}{\partial x2}\, G_2(x1,\, x2,\, x3)) + (\tfrac{\partial}{\partial x3}\, G_3(x1,\, x2,\, x3))$$

```
> GdivF := evalm(vecG*ApplyMatrix(div, vecF, Alg2)[1,1]);
```

$$GdivF := \left[\begin{array}{c} \%1\, G_1(x1,\, x2,\, x3) \\ \%1\, G_2(x1,\, x2,\, x3) \\ \%1\, G_3(x1,\, x2,\, x3) \end{array}\right]$$

$$\%1 := (\tfrac{\partial}{\partial x1}\, F_1(x1,\, x2,\, x3)) + (\tfrac{\partial}{\partial x2}\, F_2(x1,\, x2,\, x3)) + (\tfrac{\partial}{\partial x3}\, F_3(x1,\, x2,\, x3))$$

Finally, let us compute $(G\ .\ grad)\ F$ and $(F\ .\ grad)\ G$:

```
> DiagG := linalg[diag](linalg[innerprod](convert(vecG, vector),
> convert(div, vector))$3);
```

$$DiagG := \left[\begin{array}{ccc} \%1 & 0 & 0 \\ 0 & \%1 & 0 \\ 0 & 0 & \%1 \end{array}\right]$$

$$\%1 := G_1(x1,\, x2,\, x3)\, \mathrm{D1} + G_2(x1,\, x2,\, x3)\, \mathrm{D2} + G_3(x1,\, x2,\, x3)\, \mathrm{D3}$$

```
> DiagF := linalg[diag](linalg[innerprod](convert(vecF, vector),
> convert(div, vector))$3);
```

$$DiagF := \left[\begin{array}{ccc} \%1 & 0 & 0 \\ 0 & \%1 & 0 \\ 0 & 0 & \%1 \end{array}\right]$$

$$\%1 := F_1(x1,\, x2,\, x3)\, \mathrm{D1} + F_2(x1,\, x2,\, x3)\, \mathrm{D2} + F_3(x1,\, x2,\, x3)\, \mathrm{D3}$$

```
> GNablaF := ApplyMatrix(DiagG, vecF, Alg2);
```

$$GNablaF :=$$

$$\Big[G_1(x1,\, x2,\, x3)\, (\tfrac{\partial}{\partial x1}\, F_1(x1,\, x2,\, x3)) + (\tfrac{\partial}{\partial x2}\, F_1(x1,\, x2,\, x3))\, G_2(x1,\, x2,\, x3)$$
$$+ (\tfrac{\partial}{\partial x3}\, F_1(x1,\, x2,\, x3))\, G_3(x1,\, x2,\, x3)\Big]$$

$$\Big[(\tfrac{\partial}{\partial x1}\, F_2(x1,\, x2,\, x3))\, G_1(x1,\, x2,\, x3) + G_2(x1,\, x2,\, x3)\, (\tfrac{\partial}{\partial x2}\, F_2(x1,\, x2,\, x3))$$
$$+ (\tfrac{\partial}{\partial x3}\, F_2(x1,\, x2,\, x3))\, G_3(x1,\, x2,\, x3)\Big]$$

$$\Big[(\tfrac{\partial}{\partial x1}\, F_3(x1,\, x2,\, x3))\, G_1(x1,\, x2,\, x3) + (\tfrac{\partial}{\partial x2}\, F_3(x1,\, x2,\, x3))\, G_2(x1,\, x2,\, x3)$$
$$+ G_3(x1,\, x2,\, x3)\, (\tfrac{\partial}{\partial x3}\, F_3(x1,\, x2,\, x3))\Big]$$

```
> FNablaG := ApplyMatrix(DiagF, vecG, Alg2);
```

$$FNablaG :=$$

$$\Big[F_1(x1,\, x2,\, x3)\, (\tfrac{\partial}{\partial x1}\, G_1(x1,\, x2,\, x3)) + F_2(x1,\, x2,\, x3)\, (\tfrac{\partial}{\partial x2}\, G_1(x1,\, x2,\, x3))$$
$$+ F_3(x1,\, x2,\, x3)\, (\tfrac{\partial}{\partial x3}\, G_1(x1,\, x2,\, x3))\Big]$$

$$\Big[F_1(x1,\, x2,\, x3)\, (\tfrac{\partial}{\partial x1}\, G_2(x1,\, x2,\, x3)) + F_2(x1,\, x2,\, x3)\, (\tfrac{\partial}{\partial x2}\, G_2(x1,\, x2,\, x3))$$
$$+ F_3(x1,\, x2,\, x3)\, (\tfrac{\partial}{\partial x3}\, G_2(x1,\, x2,\, x3))\Big]$$

$$\Big[F_1(x1,\, x2,\, x3)\, (\tfrac{\partial}{\partial x1}\, G_3(x1,\, x2,\, x3)) + F_2(x1,\, x2,\, x3)\, (\tfrac{\partial}{\partial x2}\, G_3(x1,\, x2,\, x3))$$
$$+ F_3(x1,\, x2,\, x3)\, (\tfrac{\partial}{\partial x3}\, G_3(x1,\, x2,\, x3))\Big]$$

Finally, we check the previous identity by computing:

```
>  simplify(evalm(curlFxG-map(expand, FdivG)+
>  map(expand, GdivF)-GNablaF+FNablaG));
```

$$\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

More classical identities of differential operators can be checked similarly.