```
> restart;
> eqns:=[x[1]^3+x[1]^2*x[2]-x[1]*x[3]+6,
        exp(x[1])+exp(x[2])-x[3],
        x[2]^2-2*x[1]*x[3]-4];
```

$$eqns := \left[x_1^3 + x_1^2\,x_2 - x_1\,x_3 + 6,\ \mathrm{e}^{x_1} + \mathrm{e}^{x_2} - x_3,\ -2\,x_1\,x_3 + x_2^2 - 4\right] \tag{1}$$

```
> with(linalg):
> F:=evalf(vector(eqns));
```

$$F := \left[\begin{array}{ccc} x_1^3 + x_1^2\,x_2 - 1.\,x_1\,x_3 + 6. & \mathrm{e}^{x_1} + \mathrm{e}^{x_2} - 1.\,x_3 & -2.\,x_1\,x_3 + x_2^2 - 4. \end{array}\right] \tag{2}$$

```
> X:=[seq(x[i],i=1..3)];
```

$$X := \left[x_1,\ x_2,\ x_3\right] \tag{3}$$

```
> J:=jacobian(F,X);
```

$$J := \left[\begin{array}{ccc} 3\,x_1^2 + 2\,x_1\,x_2 - 1.\,x_3 & x_1^2 & -1.\,x_1 \\ \mathrm{e}^{x_1} & \mathrm{e}^{x_2} & -1. \\ -2.\,x_3 & 2\,x_2 & -2.\,x_1 \end{array}\right] \tag{4}$$

```
> with(CodeGeneration);
```

$$[C,\ CSharp,\ Fortran,\ IntermediateCode,\ Java,\ LanguageDefinition,\ Matlab,\ Names,\ Perl,\ Python,\ Save,\ Translate,\ VisualBasic] \tag{5}$$

```
> C(F);
F[0] = pow(x[0], 0.3e1) + pow(x[0], 0.2e1) * x[1] - 0.1e1 * x
[0] * x[2] + 0.6e1;
F[1] = exp(x[0]) + exp(x[1]) - 0.1e1 * x[2];
F[2] = -0.2e1 * x[0] * x[2] + pow(x[1], 0.2e1) - 0.4e1;
> C(J);
J[0][0] = 0.3e1 * pow(x[0], 0.2e1) + 0.2e1 * x[0] * x[1] -
0.1e1 * x[2];
J[0][1] = pow(x[0], 0.2e1);
J[0][2] = -0.1e1 * x[0];
J[1][0] = exp(x[0]);
J[1][1] = exp(x[1]);
J[1][2] = -0.1e1;
J[2][0] = -0.2e1 * x[2];
J[2][1] = 0.2e1 * x[1];
J[2][2] = -0.2e1 * x[0];
```