Vectors and Matrices

Vectors and matrices are omnipresent in computational structural analysis. However, vectors and matrices are not a core part of Python and C++. The closest is lists in Python and arrays in C++ but they do not deliver basic functionality needed in vector and matrix operations. For that reason, we use the NumPy library in Python and the vector class from the standard C++ library.

Vectors and Matrices in Python

Here is how the matrix triple product $\mathbf{K}_g=\mathbf{T}_{lg}^{\mathrm{T}}\mathbf{K}_l\mathbf{T}_{lg}$ can be evaluated in Python:

```
import numpy as np

Tlg = np.array([(1.0,  1.0),
                (1.0,  1.0)])

Kl = np.array([(5.0,  2.0),
               (2.0,  9.0)])

Kg = (np.transpose(Tlg).dot(Kl)).dot(Tlg)
```

Arrays in C++

An array of integers is declared like this:

```
     int a[3]
```

It can be initialized in the same line as it is declared:

```
     int a[3] = {1,2,3};
```

When utilizing this vector it is important to note that in C++ the indexing starts at zero:

```
     b=a[0]+a[1]+a[2];
```

A matrix of double-precision real numbers is declared as follows:

```
     double c[3][3];
```

The std::vector in C++

To use vectors, the following include statement is necessary:

```
     #include <vector>
```

It is also possible to say “include namespace std” immediately thereafter, but this approach is not adopted here to avoid the possibility of conflict between constructs in the standard library and the developer’s convention. Instead, “std::” is used, so that a vector (either a pointer or not) is declared like this:

```
     std::vector<double> *vp = new std::vector<double>(10);

     std::vector<double> v(10);
```

Similarly, a pointer to a matrix is declared as follows (notice the space between > and >):

```
std::vector< std::vector<int> > *mp = new . . .
   std::vector< std::vector<int> >(10, std::vector<int>(12));
```

```
std::vector< std::vector<int> > m(10, std::vector<int>(12));
```

The elements of the vectors and matrices are automatically initialized to zero. Other values can be inserted like this:

```
     (*vp)[3] = 19.0;
     v[3] = 18.0;
     (*mp)[2][3] = 19.0;
     m[2][3] = 18.0;
```

The elements are read like this:

```
double a = (*vp)[3];
double b = v[3];
double c = (*mp)[2][3];
double d = m[2][3];
```