

# Interfaçage entre Maple et du code externe

Nicolas Gachadoit  
Ingénieur d'applications



# Qu'est-ce que Maple ?

## Maple est un des logiciels leaders du calcul mathématique et symbolique

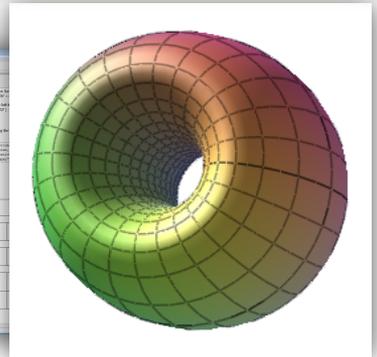
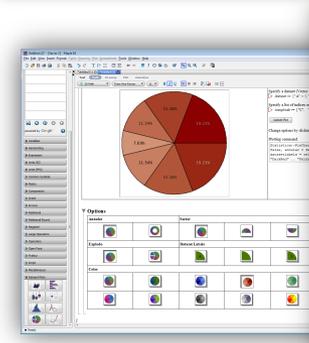
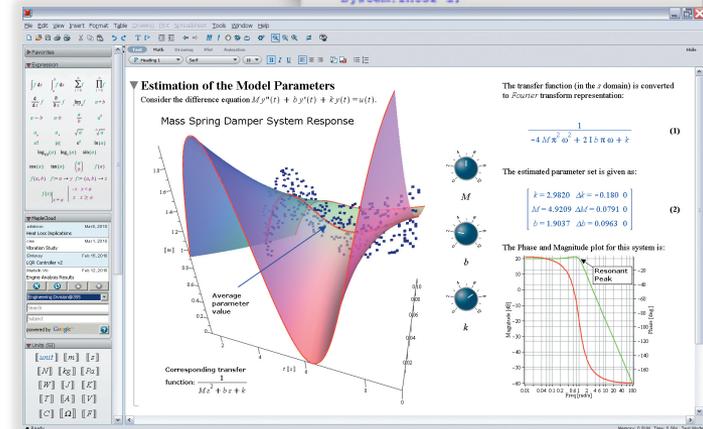
- Calcul symbolique et numérique très performants
- Analyse, exploration, visualisation et résolution de problèmes mathématiques dans un seul environnement
- Différents types d'interaction, y compris via un langage de programmation très complet
- Capture de la connaissance et des calculs dans un format électronique combinant calculs, texte, tracés graphiques, images, sons, vidéos,...

```
> f := proc(n)
local x, i;
x := 0.0;
for i to n do
x := x + i;
end do;
end proc;

> CSharp(f)

public class CodeGenerationClass {
public static System.Double f (System.Int32 n)
{
System.Double x;
System.Int32 i;

```



# Deux possibilités d'interfaçage

## Appel de code externe depuis Maple

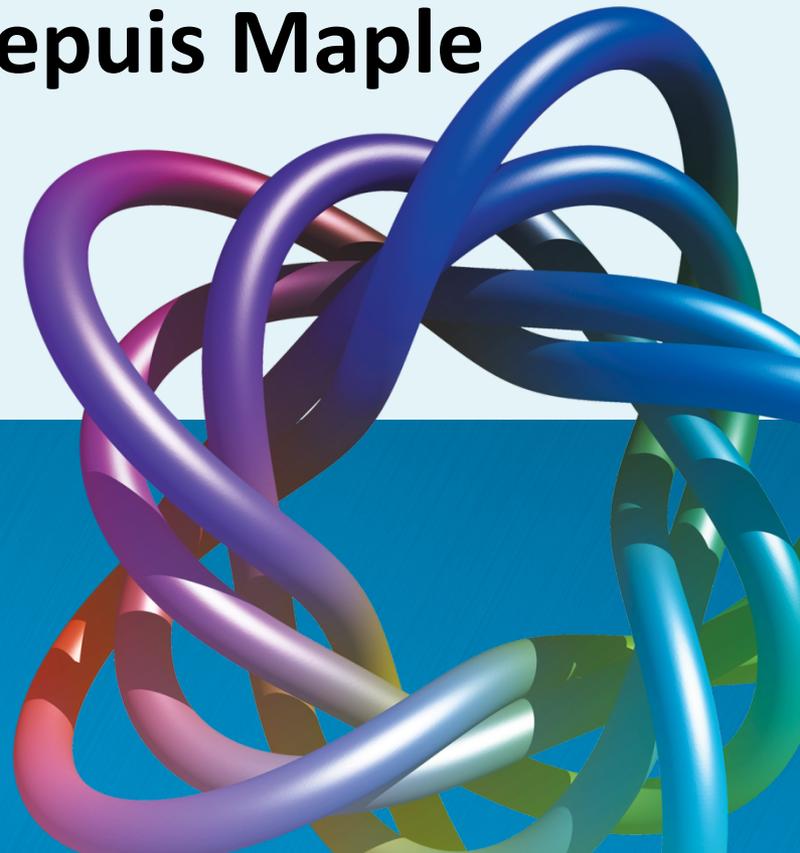
- Disponible en C/C++, Java, FORTRAN
- Cas pratiques:
  - Extension des capacités de Maple grâce à des bibliothèques de calcul existantes
  - Accélération des calculs par rapport à des fonctions écrites en langage Maple

## Appel de Maple depuis du code externe (API OpenMaple)

- Disponible en C/C++, Java, Visual Basic
- Cas pratique: utilisation de Maple dans des applications nécessitant des calculs mathématiques évolués

*Rappel: on peut également étendre les possibilités de Maple grâce à l'ajout de fonctions écrites en langage Maple*

# Appel de code externe depuis Maple



# Appel de code externe depuis Maple

## Etapas à suivre:

1. Ecrire dans le langage “externe” le code de la fonction à ajouter.  
Exemple: factorielle (écrite en C)

```
int __stdcall factorialC (int n) {  
    unsigned i;  
    unsigned long long factorial = 1;  
    for (i=1; i<=n; ++i) {  
        factorial *= i;  
    }  
    return factorial;  
}
```

*Pour info: la fonction de calcul de factorielle existe déjà dans Maple...*

# Appel de code externe depuis Maple

## Etapas à suivre:

2. Compiler cette fonction sous la forme d'une librairie partagée.  
Exemple sur Windows avec le compilateur Visual Studio:

```
cl -Gz factorialC.c -link -dll -export:factorialC -out:factorialC.dll
```

Les commandes de compilation sont disponibles dans l'aide pour différents OS

3. Dans Maple, lier ce dernier avec cette librairie partagée:

```
factorialC := define_external( 'factorialC',  
                              'C',  
                              'n'::integer[4],  
                              'RETURN'::integer[4],  
                              'LIB'="factorialC.dll");
```

# Appel de code externe depuis Maple

## Etapas à suivre:

4. Appel de cette nouvelle fonction comme n'importe quelle fonction Maple:

**factorialC(10)**

*Les fonctionnalités de calcul parallèle de Maple supportent l'appel de code externe.*

*Il est possible de passer des tableaux en paramètre.*

# Appel de code externe depuis Maple

## Création du lien dans les autres langages:

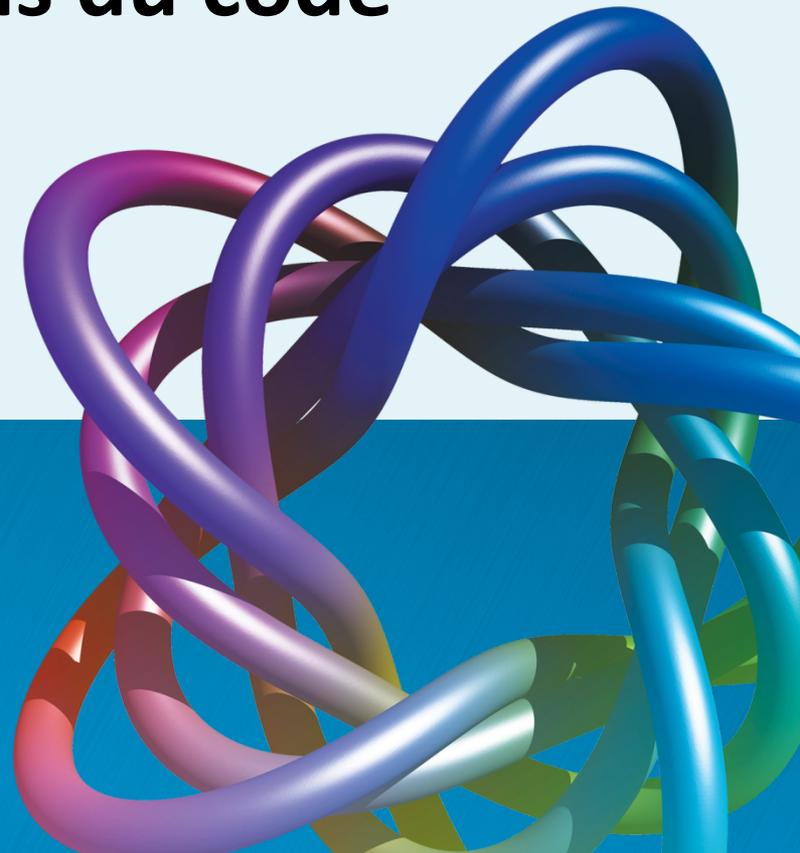
- Java:

```
factorialJava := define_external( 'factorialJava',  
                                'Java',  
                                'n'::integer[4],  
                                'RETURN'::integer[4],  
                                'CLASS'="factorialJava",  
                                'CLASSPATH'=".")
```

- FORTRAN:

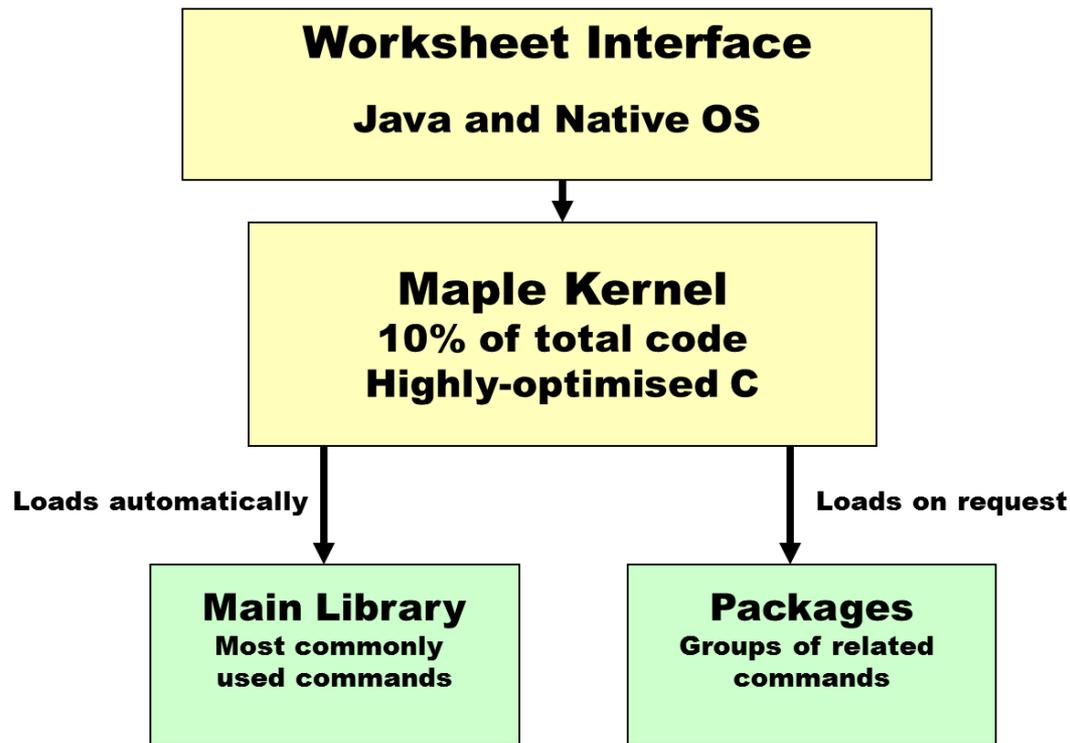
```
factorialFortran := define_external( 'factorialFortran',  
                                     'FORTRAN',  
                                     'n'::integer[4],  
                                     'RETURN'::integer[4],  
                                     'LIB'="factorialFortran.dll")
```

# Appel de Maple depuis du code externe



# Appel de Maple depuis du code externe

## Architecture de Maple:



L'API OpenMaple permet d'accéder directement au noyau

# Appel de Maple depuis du code externe

## Etapas à suivre:

1. Ecrire dans le langage “externe” un programme faisant appel aux fonctions de l’API OpenMaple
2. Compiler ce programme.

Exemple sur Windows avec le compilateur Visual Studio:

```
cl main.c
```

```
-I"C:/Program Files/Maple 2016/extern/include"
```

```
"C:/Program Files/Maple 2016/bin.X86_64_WINDOWS/maplec.lib"
```

Les commandes de compilation sont disponibles dans l’aide pour différents OS

# Appel de Maple depuis du code externe

## Etapes à suivre:

3. Exécuter le programme (à l'extérieur de Maple)

## Attention:

- Maple doit être installé sur la machine sur laquelle s'exécute le programme compilé
- Il n'est pas possible d'appeler simultanément 2 noyaux Maple (ou plus) via OpenMaple

**Questions ?**

