

Pourquoi Mathematica?

Pour construire leurs modèles et analyser les données, les scientifiques disposent de trois logiciels généraux : R, MATLAB et Mathematica. MATLAB offre l'avantage d'être assez proche des langages classiques de programmation. Ce qui facilite son apprentissage, mais impose d'écrire de longs programmes. Il est très complet quand on lui adjoint de nombreuses boîtes à outils. Ce qui revient très cher. R est un langage open source, gratuit et qui bénéficie de mises à jour rapides. Moins stable que les logiciels payants, il nécessite aussi d'écrire de nombreuses lignes de code, mais il dispose d'une multitude de boîtes à outils complémentaires. Mathematica est plus original. Contrairement à ce que laisse croire son nom, Mathematica est un langage complet de formalisation, y compris graphique et cartographique, et une gigantesque base de connaissances. Il présente plusieurs avantages.

D'abord, c'est un langage très complet. Il dispose de plus de six mille fonctions qui permettent une approche de tous les domaines de modélisation que fréquente un géographe : statistiques, probabilités, traitements des chroniques et processus stochastiques, macro-simulation par des équations différentielles ordinaires ou partielles, micro-simulation par des automates cellulaires et des systèmes multi-agents, théorie des graphes, traitements d'images, morphologie mathématique, ondelettes, représentations graphiques et cartographiques complètes. En fait, le géographe peut puiser dans un puits sans fond pour construire ses modèles. Cette richesse est immense, car la plupart des fonctions proposent de nombreuses options. De fait, chacune des fonctions est un véritable petit programme. Par exemple, de nombreuses fonctions, que le géographe va mobiliser, nécessitent le calcul de la distance. Or, ce langage offre une vaste palette de distances. Et le géographe peut même introduire sa propre fonction distance. Le géographe dispose d'un véritable outil de cartographie automatique et de ce fait il peut réaliser un SIG ultra-performant en quelques lignes car toutes les fonctions sont applicables à ces objets cartes. Et surtout, tout est intégré dans un seul programme. Pas besoin de recourir sans cesse à des boîtes à outils externes, même si quelques unes sont disponibles.

Par ailleurs, les diverses aides sont d'une richesse pratiquement infinie. Chaque fonction porte un nom anglais intuitif, ce qui facilite l'apprentissage. **Mean[]** permet le calcul de moyenne tandis que **GeoGraphics[]** est un outil de cartographie. Chaque fonction est détaillée à l'aide de multiples exemples. Il n'est pas rare que la définition d'une seule fonction s'étende sur plusieurs pages, avec de nombreuses études de cas qui peuvent faire l'objet d'un simple copier-coller. Par exemple, la fonction **GeoGraphics[]**, qui est à la base de la cartographie, est exposée en 47 pages dans lesquelles plusieurs options et exemples sont longuement illustrés. De plus, il existe des communautés sur Internet qui permettent de poser des questions et d'obtenir très rapidement des réponses. Citons la communauté StackExchange, avec pas moins de 70 000 questions-réponses, et la communauté Wolfram. Pour des problèmes plus compliqués, les Wolfram **Demonstration Projects** sont de véritables programmes prêts à l'emploi, où il suffit souvent d'introduire ses propres données pour obtenir immédiatement des résultats. Il en existe plus de 10 000, qui couvrent tous les champs de la modélisation.

Autre richesse, l'ouverture vers l'extérieur. Il y a un demi siècle, le géographe disposait de rares données, et il devait souvent compléter son information par une pratique assidue du terrain. Cinquante ans plus tard, il est débordé par la profusion des données. Nous sommes passés d'une société sous informée à une société mondiale sur informée. Cette sur information peut même cacher les informations véritablement significatives. Pour se saisir de ces données, le langage Mathematica donne un accès commode aux mégadonnées. Des options de la fonction **Import[]** assurent un lien direct avec d'innombrables fichiers. Dans un autre domaine, la fonction **SocialMediaData[]** favorise le traitement de données de réseaux sociaux

(Facebook, Twitter, LinkedIn,...). De plus, Mathematica possède ses propres bases de données, dont certaines intéressent directement le géographe (**CityData[]**, **CountryData[]**, **WeatherData[]**, **FinancialSata[]**...). Ces bases de données font partie d'une immense base de connaissances, sans cesse remise à jour, qui couvre des milliers de domaine. En fait, le géographe peut importer n'importe quelle donnée sans gros effort. Et, cette ouverture ne se limite pas aux données. Mathematica peut converser et échanger avec d'autres outils de programmation, par exemple R ou NetLogo et les langages classiques, comme Java, C ou Python.

Démarrer avec Mathematica

Après avoir lancé le logiciel, aller dans le menu File, le sous-menu New et choisir Notebook. Une page vierge s'ouvre, un notebook ou cahier est créé. Dans ce cahier, le géographe tape des instructions en entrée dans une ou plusieurs cellules visualisées par des crochets à droite du notebook. En sélectionnant la cellule et en tapant en même temps les touches "Shift + entrée" du clavier, le programme procède à l'exécution des instructions de cette cellule. Une nouvelle cellule *Out* fait apparaître le résultat. En retournant dans le menu File et le sous-menu Save, le notebook est sauvegardé et, comme un fichier Word, il peut être re-ouvert lors d'une prochaine session.

Toutes les fonctions Mathematica, à l'exception de quelques fonctions relatives au système, commencent par une majuscule et sont généralement directement compréhensibles ayant quelques connaissances de la langue anglaise. Toutes les fonctions écrites par le géographe qui programme doivent obligatoirement commencer par une minuscule. Toutes utilisent des crochets [] à l'intérieur desquels figurent les données et les options.

```
In[*]:= data = Range[5]
           |plage
data1 = Range[8];
           |plage
data3 = Range[1, 11, 3]
           |plage
```

```
Out[*]:= {1, 2, 3, 4, 5}
```

```
Out[*]:= {1, 4, 7, 10}
```

Soit trois formes de la fonction **Range[]** qui crée une suite de nombres. La première ligne crée une plage de chiffres, 1, 2, 3, 4, 5, qui sont transférés dans le fichier data. Les virgules séparent les éléments d'une liste. Cette liste de nombres, séparés par des virgules, est à l'intérieur d'accolades {} qu'il ne faut pas confondre avec les crochets. Les accolades délimitent une liste tandis que les crochets encadrent les paramètres d'une fonction. La deuxième ligne se termine par un point-virgule, ce qui empêche l'affichage des résultats sur l'écran. Mais les 8 nombres sont bien à l'intérieur du fichier data1. La troisième ligne permet de créer une liste de nombres de 1 à 11, mais avec un pas de 3. Et le résultat apparaît à l'écran.

Les doubles crochets permettent de prendre un objet dans une liste. Ainsi, l'instruction suivante:

```
In[*]:= data3[[3]]
```

```
Out[*]:= 7
```

donne le troisième nombre de la liste data3, soit 7. Voici un exercice un peu plus compliqué avec la fonction **Table[]** qui est très utilisée pour créer des tableaux.

```
In[*]:=
data6 = Table[(i + j^2), {i, 1, 5, 1}, {j, 1, 3, 1}]
      |table
TableForm[data6]
      |forme de table
```

```
Out[*]= {{2, 5, 10}, {3, 6, 11}, {4, 7, 12}, {5, 8, 13}, {6, 9, 14}}
```

```
Out[*]/TableForm=
  2   5   10
  3   6   11
  4   7   12
  5   8   13
  6   9   14
```

La première instruction crée un tableau qui se présente sous la forme d'une liste de liste. L'instruction suivante présente le résultat sous la forme d'un tableau.

Il est possible de placer une fonction à l'intérieur d'une autre fonction. La fonction intérieure est exécutée en premier. Puis la fonction extérieure agit sur les résultats livrés par la fonction intérieure. Les deux lignes précédentes peuvent donc être regroupées, et seul le tableau final apparaît :

```
In[*]:= TableForm[Table[(i + j^2), {i, 1, 5, 1}, {j, 1, 3, 1}]]
      |forme de table |table
```

```
Out[*]/TableForm=
  2   5   10
  3   6   11
  4   7   12
  5   8   13
  6   9   14
```

Les listes peuvent contenir tout objet, des nombres, des graphiques, des images, des cartes, des équations, du texte, et même plusieurs objets de nature différente, par exemple du texte et une image. Il est facile d'appliquer une très grand nombre de fonctions à ces listes. Voici un exemple qui crée la même liste de nombre que précédemment, et calcule les moyennes sur les colonnes, puis sur les lignes en transposant d'abord les données :

```
In[*]:= data = Table[(i + j^2), {i, 1, 5, 1}, {j, 1, 3, 1}]
      |table
Mean[data] // N
      |valeur moyenne |valeur numérique
Mean[Transpose[data]] // N
      |valeur· |transposée |valeur numérique
Out[*]= {{2, 5, 10}, {3, 6, 11}, {4, 7, 12}, {5, 8, 13}, {6, 9, 14}}
Out[*]= {4., 7., 12.}
Out[*]= {5.6666667, 6.6666667, 7.6666667, 8.6666667, 9.6666667}
```

Voici un exemple d'une liste de pays cartographiés :

```
In[ ]:= GeolistPlot[{{ Bulgaria COUNTRY  ✓ }, Greece COUNTRY  ✓ },
  { Albania COUNTRY  ✓ }, Romania COUNTRY  ✓ }, { Turkey COUNTRY  ✓ }]]
```



La carte représente la liste de cinq pays.

Des aides à profusion

Mathematica offre d'abord des aides classiques, sous la forme de plusieurs centaines de livres, dont la liste est accessible en consultant le site : <https://www.wolfram.com/books/> et des centaines d'articles. Parmi les ouvrages essentiels, citons quatre initiations très pédagogiques :

Le cours photocopié de Thierry Verdel disponible sur son site : <https://thierryverdel.wixsite.com/home/-math>.

Cliff Hastings, Kelvin Mischo, Michael Morrison (2020), *Hands-on start to Wolfram Mathematica*, Wolfram Media.

Stephen Wolfram, 2015, *An Elementary Introduction To The Wolfram Language*, Wolfram Media.

Une introduction rapide pour un futur programmeur est disponible sur le site web en langue française : <https://www.wolfram.com/language/fast-introduction-for-programmers/fr/interactive-usage/>

Deux revues libres d'accès sont centrées sur ce logiciel : *The journal of Mathematica et Complex System*. Leurs articles sont récupérables sur Internet aux adresses :

<https://www.mathematica-journal.com>

<https://www.complex-systems.com/issues/10-6/>

Par ailleurs, chaque année, se tiennent des symposium et des conférences techniques qui traitent des dernières nouveautés incluses dans le logiciel. Pour 2020, on se reportera à l'adresse web :

<https://www.wolfram.com/events/technology-conference/2020/>

On y trouve la plupart des communications faites lors du symposium 2019.

Outre ces documents, le lecteur peut voir des centaines de vidéo consacrées à ce logiciel. Le débutant peut visionner un ouvrage d'introduction écrit par Stephen Wolfram à l'adresse :

<https://www.wolfram.com/language/elementary-introduction/2nd-ed/?source=footer> vidéo

Il peut aussi avoir des cours gratuits et parfois payants à l'adresse :

<https://www.wolfram.com/wolfram-u/?source=footer>

et suivre les avancées de cet ensemble de logiciels à l'adresse :

<https://www.youtube.com/user/WolframResearch/videos>

Enfin, sans prétendre être exhaustif, il est possible de poser des questions pour résoudre un problème, d'échanger à l'intérieur de deux communautés, dont les adresses sont :

<https://mathematica.stackexchange.com/questions?sort=newest>

<https://community.wolfram.com>

Beaucoup de questions et de réponses concernent le débutant.

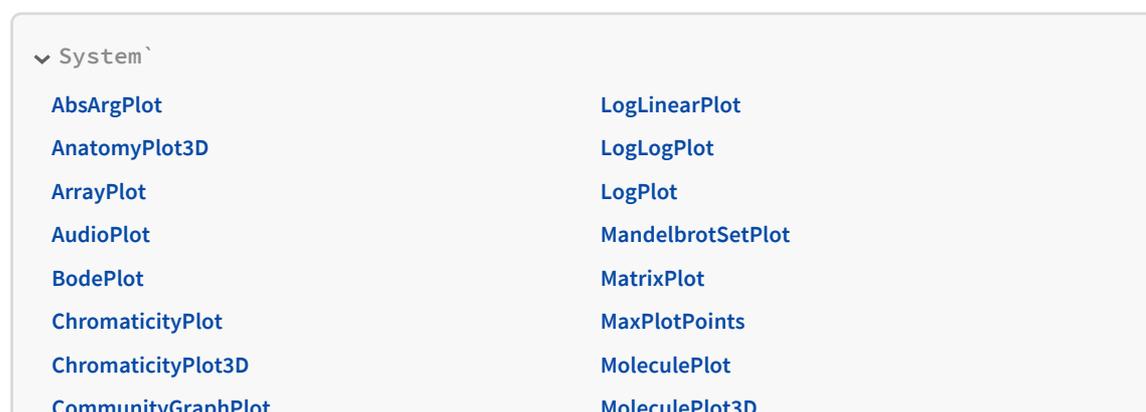
Quand le géographe travaille dans un notebook, différentes aides sont directement accessibles à partir du menu **Help**. Par exemple, pour avoir une idée des fonctions spécifiques à la géographie, il suffit de choisir dans le menu **Help** le sous-menu **Wolfram Documentation**. Un tableau apparaît sur l'écran avec la case **Geographic Data & Computation**. En appuyant sur cette case et en choisissant le sous menu **Maps & Cartography** le chercheur obtient une longue liste de fonctions utiles pour réaliser des modèles cartographiques. Ces fonctions sont classées par groupe. Le premier groupe comprend les fonctions **GeoListPlot[]**, **GeoBubbleChart[]**, **GeoRegionValuePlot[]**,...

En cliquant sur une fonction, par exemple la première de la liste, donc **GeoListPlot[]**, le chercheur accède à la fiche d'aide pour cette fonction. Pour chaque fonction on dispose d'un notebook, toujours organisé selon les mêmes rubriques : *details and options*, *basic exemples*, *scope*, *options*, *applications*,... Au bas de la page, sont indiquées une liste de *Related functions*, qui fournit une petite liste de fonctions similaires. Chacun de ces notebook est plus ou moins volumineux suivant la fonction retenue. Surtout, il est possible d'intégrer, par un simple copier-coller, les exemples exposés dans cette aide dans le notebook que le chercheur construit. Au total, les aides de toutes les fonctions représentent plus de 150 000 pages.

Pour accéder plus directement à une fonction que l'on connaît, il suffit de taper le nom de cette fonction dans le notebook, de la sélectionner comme dans un traitement de texte, puis de choisir le sous menu **Find Selected Function** dans le menu **Help**, et le notebook d'aide de la fonction apparaît immédiatement.

Parfois on a une idée de la fonction que l'on souhaite utiliser, mais son nom est imparfaitement retenu. Il est alors possible de poser une question dans le notebook. Par exemple, je souhaite connaître une fonction pour réaliser un graphique, un plot en anglais. Il suffit de taper :

In[]:= ? *Plo*

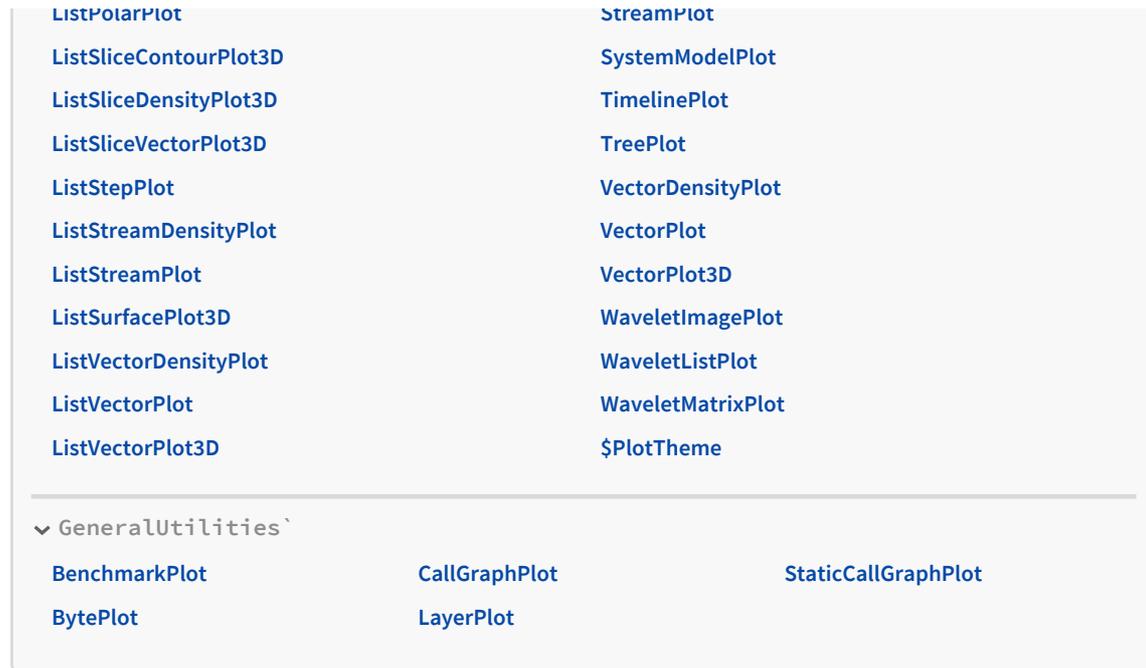


Out[]:=

```

ComplexContourPlot
ComplexListPlot
ComplexPlot
ComplexPlot3D
ComplexRegionPlot
ComplexStreamPlot
ComplexVectorPlot
ContourPlot
ContourPlot3D
DateListLogPlot
DateListPlot
DateListStepPlot
DensityPlot
DensityPlot3D
DiscretePlot
DiscretePlot3D
FeatureSpacePlot
FeatureSpacePlot3D
GeoContourPlot
GeoDensityPlot
GeoListPlot
GeoRegionValuePlot
GeoStreamPlot
GeoVectorPlot
GraphPlot
GraphPlot3D
JuliaSetPlot
LayeredGraphPlot
LineIntegralConvolutionPlot
ListContourPlot
ListContourPlot3D
ListCurvePathPlot
ListDensityPlot
ListDensityPlot3D
ListLineIntegralConvolutionPlot
ListLinePlot
ListLogLinearPlot
ListLogLogPlot
ListLogPlot
ListPlot
ListPlot3D
ListPointPlot3D
NicholsPlot
NumberLinePlot
NyquistPlot
ParametricPlot
ParametricPlot3D
Plot
Plot3D
Plot3Matrix
PlotDivision
PlotJoined
PlotLabel
PlotLabels
PlotLayout
PlotLegends
PlotMarkers
PlotPoints
PlotRange
PlotRangeClipping
PlotRangeClipPlanesStyle
PlotRangePadding
PlotRegion
PlotStyle
PlotTheme
PolarPlot
ProbabilityPlot
ProbabilityScalePlot
QuantilePlot
RegionPlot
RegionPlot3D
ReImPlot
ReliefPlot
RevolutionPlot3D
RootLocusPlot
RulePlot
SingularValuePlot
SliceContourPlot3D
SliceDensityPlot3D
SliceVectorPlot3D
SphericalPlot3D
StackedDateListPlot
StackedListPlot
StreamDensityPlot

```

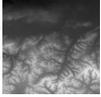


La longue liste de fonctions de type Plot suffit à montrer la richesse d'une partie des représentations graphiques que le chercheur peut employer avec ce logiciel. Il manque d'ailleurs dans cette liste des fonctions graphiques qui ne comporte pas le terme Plo. Citons par exemple la fonction **Dendrogram[]** pour dessiner un arbre de classification.

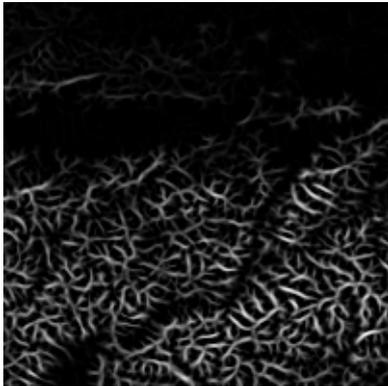
Dans le menu **Help** deux autres sous-menu sont utiles. Les **Demonstration Projects**, plus de 12 000, sont des programmes qui répondent à des interrogations de l'utilisateur. En choisissant ce sous-menu, une fenêtre s'ouvre dans le navigateur Internet et le chercheur fait des choix en fonction de ces centres d'intérêt. Quant au sous-menu **Wolfram Web**, il donne accès au site Web où sont concentrés des milliers d'informations.

Deux exemples pour jouer

Le premier exemple relève du traitement d'images. Il comprend deux fonctions imbriquées. La première **RidgeFilter[]**, à l'intérieur, applique un filtre pour mettre en valeur les lignes directrices d'un modèle numérique de terrain illustré par l'image. Pour rendre cette image plus lisible, la première instruction est incluse dans la deuxième **ImageAdjust[]** :

```
In[8]:= ImageAdjust[RidgeFilter[]]
```

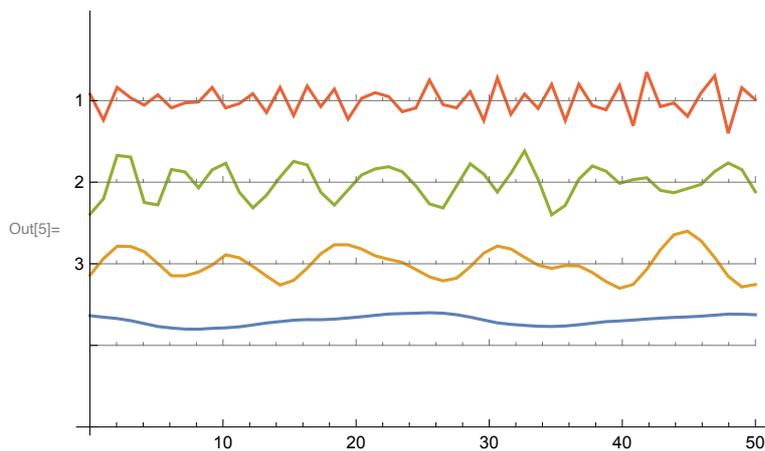
```
Out[8]=
```



Si vous avez une autre image de MNT insérez la entre les crochets à la place de l'image fournie et observez le résultat.

Un second exemple de programme avec 3 instructions imbriquées. La première, à l'intérieur **RandomInteger[]** crée une liste de 50 nombres aléatoires, dont la valeur est comprise entre 0 et 15. La deuxième instruction **StationaryWaveletTransform[]** avec ses deux options *DaubechiesWavelet[4]* et *3 niveaux*, séparées par une virgule, réalise une décomposition en ondelettes sur la liste de nombres créés, avec comme option de procéder à une décomposition de Daubechies, d'ordre 4 et sur 3 niveaux. La troisième instruction, **WaveletListPlot[]**, la plus externe, donne une représentation graphique sur votre écran du résultat de cette décomposition :

```
In[5]:= WaveletListPlot[
  trace liste de coefficients de transforme d'ondelette
  StationaryWaveletTransform[RandomInteger[15, 50], DaubechiesWavelet[4], 3]]
  transformée d'ondelettes stationnaire entier aléatoire ondelette Daubechies
```



Forcez vous à bien percevoir l'imbrication des 3 fonctions. De plus, en relançant ce petit programme, vous voyez apparaître sous le résultat, une bande grise avec diverses possibilités pour modifier ce graphique. De plus, avec un clic droit sur votre souris d'autres possibilités s'offrent à vous pour retravailler ce graphique et le sauvegarder.

ATTENTION : Comme vous avez ouvert un notebook qui était fermé, en lançant cette ligne de 3 instructions, vous relancez la fonction **RandomInteger[]**, et vous pouvez obtenir un résultat visuel différent.

Conclusion

Plutôt que d'apprendre des logiciels spécialisés, le géographe devrait se concentrer sur un des trois langages généralistes qui permettent de répondre à toutes ses questions.