▪ Introduction to *Mathematica.* Volume III - XtraStuff

download this document from `www.hakenberg.de`

⚠ Please **deactivate NUM-Lock**.
⚠ Proceed by copying the expressions into *Mathematica*. Press SHIFT RET together to **evaluate** your input. Vary the input.
⚠ When you recieve weird error messages upon evaluation, although your input seems correct, use the menu item **Kernel** → **Quit-Kernel** to reset the memory of *Mathematica* and start evaluation from the beginning.

✍ Understand the function `γInit`

```
γInit[n_Int][X0_] := MapIndexed[x^(n)_#2[[1]][0] == #1 &, X0];
```

✍ Evaluate f at 1/2 and plot the function within an appropriate interval, where

```
f = Interpolation[{{0, 2}, {1, -1}, {3, 1}, {5, 6}}]
```

✍ Describe the difference between `NestList` and `FixedPointList`.

✍ Think about how to setup a general matrix in arbitrary dimension $n \times n$ such as

$$\begin{pmatrix} a_1 & a_2 & a_3 & a_4 \\ a_2 & a_5 & a_6 & a_7 \\ a_3 & a_6 & a_8 & a_9 \\ a_4 & a_7 & a_9 & a_{10} \end{pmatrix}$$

✍ Our presentation of legal syntax in *Mathematica* is far from complete. That does not automatically mean, you can't live without them. But if brevity and efficiency appeals to you, we point you to **Programming** → **Assignments** in the help browser. You find things such as

```
Unprotect[Sin];
Sin /: Sin[x_] Cos[y_] := Sin[x + y] / 2 + Sin[x - y] / 2;
```

✍ Imagine a graph with vertices $\{v_1, v_2, v_3\}$ and directed edges $v_1 \to v_2$ and $v_2 \to v_3$. We encode the connectivity by the matrix A below. The entry of the matrix product $(A.A)_{i,j}$ corresponds to the number of paths of length 2 going from $v_j \to v_i$. In our example $(A.A)_{3,1} = 1$, because the only path connecting v_1 and v_3 is $v_1 \to v_2 \to v_3$. Design code, that outputs the connectivity relation table $B_{i,j} \in \{0, 1\}^{n \times n}$ with $B_{i,j} = 1 \Leftrightarrow \exists$ a directed path from $v_j \to \ \ldots \to v_i$.

$$A = \begin{pmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}$$

▪ Solutions to some of the problems

`f[1/2]` tells you that mathematica uses algebraic (polynomial) interpolation, when fed with non-numerical input.

Concerning the general symmetric bilinear form, a `Module` allows you to assign a single entry of a list (⚠ `With` does not replace `Module` here). We are not aware of a more elegant solution than:

```
Sym[n_] := Module[{A = Array[0 &, {n, n}], cnt = 0},
   Array[(A[[#1, #2]] = If[#1 ≤ #2, a_++cnt, A[[#2, #1]]]) &, {n, n}]];
```

The connectivity relation table is indeed the result of a fixed point routine. Try the below code with more exciting matrices *A* and use the command `FixedPoint` instead of `FixedPointList`.

```
myMax[a_] := Min[a, 1];
SetAttributes[myMax, Listable]
FixedPointList[myMax[#1 + #1.#1] &, A]
```