1 Etude de suites récurrentes

1.1 Etude de suites numériques

Il s'agit ici d'étudier le comportement des suites définies par $u_0 = 0.1$
et la relation de récurrence $u_{n+1} = f(u_n)$
où $f(x) = 4\,a\,x(1-x)$.

1.2 Calcul de u_n

Ecrire une procédure de paramètres a et n qui calcule u_n et l'essayer pour $a = 0.4$ et $n = 10$.
Comme on l'a déjà fait, à chaque étape de la boucle, on calcule le terme qui suit `u` et on le réaffecte ensuite à `u`.
Cette réaffectation s'écrit **avant** le calcul...

```
> suite:=proc(a,n) local u; u:=.1; from 1 to n do u:=4*a*u*(1-u) od; u
end;
```

$$\begin{array}{l} suite := \mathbf{proc}(a,n) \\ \mathbf{local}\; u; \\ \quad u := .1\,; \\ \quad \mathbf{to}\; n\; \mathbf{do}\; u := 4*a*u*(1-u)\; \mathbf{end\ do}\,; \\ \quad u \\ \mathbf{end\ proc} \end{array}$$

```
> suite(.4,10);
```

$$.3745086274$$

1.3 Etude de la convergence de la suite

Ecrire une procédure qui calcule le premier n tel que $|u_{n+1} - u_n| \leqslant d$, de paramètres a et d, limitée à 500 itérations en cas de divergence ou de convergence lente...
En sortie, on donnera n,u_n,u_{n+1}.

```
> rang:=proc(a,d) local i,u,v; u:=.1;v:=4*a*u*(1-u); for i from 1 to
500 while abs(u-v)>d do u:=v;v:=4*a*u*(1-u) od; i,u,v end;
```

$$\begin{array}{l} rang := \mathbf{proc}(a,d) \\ \mathbf{local}\; i,u,v; \\ \quad u := .1\,; \\ \quad v := 4*a*u*(1-u)\,; \\ \quad \mathbf{for}\; i\; \mathbf{to}\; 500\;\; \mathbf{while}\; d < \mathrm{abs}(u-v)\; \mathbf{do}\; u := v\,;\, v := 4*a*u*(1-u)\; \mathbf{end\ do}\,; \\ \quad i,u,v \\ \mathbf{end\ proc} \end{array}$$

On va suivre, sur le tableau ci-dessous, l'exécution de ce programme en regardant les valeurs des différentes variables à **l'entrée** de la boucle, juste aprés le `do`.

	Valeur à l'entrée de la boucle :							
Variable Maple	N° 1	N° 2	N° 3	$\cdots$	N° k	N° k+1	$\cdots$	N° n
`i`	1	2	3		k	k+1		n
`u`	u_0	u_1	u_2		u_{k-1}	u_k		u_{n-1}
`v`	u_1	u_2	u_3		u_k	u_{k+1}		u_n

Essayons cette procédure avec différentes valeurs de a et d.

```
> rang(.5,.00001);
```

$$7, .4999996860, .5000000000$$

```
> rang(.7,.00001);
```

$$46, .6428618160, .6428534044$$

```
> rang(.74,.00001);
```

$$227, .6621570944, .6621670272$$

```
> rang(.75,.00001);
```

$$501, .6561213688, .6768783548$$

```
> rang(.8,.00001);
```

$$501, .5130445096, .7994554904$$

On voit bien que, plus a augmente, la convergence est de plus en plus lente mais semble toujours exister jusque 0,75.
Au delà, il ne semble plus y avoir convergence, ou alors, elle est extrèmement lente.

1.4 Etude d'une suite de polynômes

On considère la suite de polynômes définie par :

- $P_0(x) = 1$
- $P_1(x) = x$
- $P_{n+2}(x) = x\,P_{n+1}(x) + P_n(x)$ pour n entier naturel

Ecrire une procédure de paramètre n, qui calcule de façon réduite et ordonnée P_n.

```
> poly:=proc(n) local u,v,w; u:=1;v:=x; from 2 to n do
w:=expand(x*v+u,x);u:=v;v:=w od; v end;
```

poly := **proc**(n)
local u, v, w;
$u := 1$; $v := x$; **from** 2 **to** n **do** $w :=$ expand$(x * v + u, x)$; $u := v$; $v := w$ **end do** ; v
end proc

Essayons cette procédure :

```
> poly(2);
```

$$x^2 + 1$$

```
> poly(3);
```

$$x^3 + 2\,x$$

```
> poly(12);
```

$$x^{12} + 11\,x^{10} + 45\,x^8 + 84\,x^6 + 70\,x^4 + 21\,x^2 + 1$$

Remarquons qu'une suite de polynômes se programme pratiquement comme une suite numérique...

2 A retenir

2.1 Variable Maple et sens mathématique

Il convient de bien séparer :

- les variables Maple, appelées par leur nom,
- et leurs valeurs successives, leur sens mathématique, en rapport direct avec l'énoncé.

Ainsi, quand on programme une suite, les termes successifs de la suite sont, le plus souvent, les valeurs successives **d'une seule variable** Maple.

2.2 Corps d'une boucle

- Quand on effectue un calcul itératif, au moyen d'une boucle, on a une ou plusieurs variables Maple qui représentent les valeurs de certaines variables mathématiques à un moment donné du calcul.
- Le corps de la boucle, entre le `do` et le `od`, est ainsi le passage d'une étape du calcul à l'étape suivante. Il consiste donc à **modifier** les valeurs des variables Maple concernées.
- Avant la boucle, il y a donc nécessairement une **première** affectation des variables Maple à leur valeur initiale, ce qu'on appelle l'initialisation.