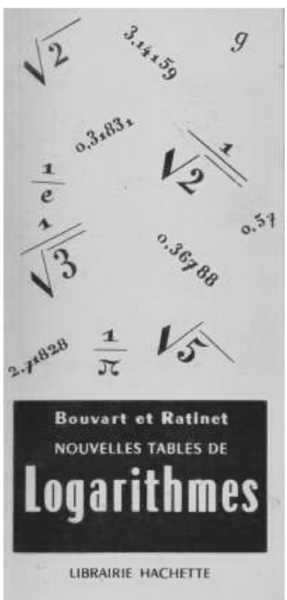
Calculer un logarithme avec une calculette « 4 opérations »

Bernard Langer

Le but de cet article est de montrer qu'*il est possible de calculer* un logarithme à l'aide d'une calculette « 4 opérations ». Il s'agit donc de construire un *algorithme* en respectant un certain nombre de *contraintes*. Ce type de situation est courant en mathématiques (constructions géométriques à la règle et au compas, par exemple).

Pour commencer…



Nous allons illustrer la démarche algorithmique sur un exemple qui peut être mis en œuvre sur une calculette « 4 opérations » basique. Ces calculatrices présentent une particularité que l'on ne retrouve pas, en général, sur les modèles plus sophistiqués :

Lorsqu'on tape la séquence de touches : $\boxed{\times}\boxed{=}$ le nombre affiché est mis *en facteur constant multiplicatif*. Cette remarque permet d'obtenir très rapidement les puissances successives d'un décimal. Par exemple :

La séquence $\boxed{2}\boxed{\times}\boxed{=}$ affiche 4.
Un nouvel appui sur $\boxed{=}$ affiche 8.
Un nouvel appui sur $\boxed{=}$ affiche 16.
etc.
Ainsi la séquence $\boxed{2}\boxed{\times}\boxed{=}\boxed{=}\boxed{=}\boxed{=}\boxed{=}\boxed{=}\boxed{=}\boxed{=}\boxed{=}\boxed{=}$ (9 fois la touche $\boxed{=}$) affiche $2^{10} = 1\,024$.
C'est cette fonctionnalité que nous allons exploiter dans ce qui suit pour déterminer des valeurs approchées des logarithmes décimaux.

En route vers le logarithme décimal…

Nous noterons le logarithme décimal log : pour tout réel strictement positif α, $\log\alpha = \dfrac{\ln\alpha}{\ln 10}$.

Rappelons que tout décimal strictement positif a s'écrit de manière unique :

(*) bernard.langer@laposte.net

$$\begin{cases} a = x \times 10^n \\ \text{avec } 1 \le x < 10 \text{ et } n \in \mathbb{Z} \end{cases}.$$

Il s'agit là de l'écriture scientifique de a.
Exemples : $2 = 2 \times 10^0$; $2\,000 = 2 \times 10^3$; $0{,}02 = 2 \times 10^{-2}$;
Par conséquent : $\log a = \log(x \times 10^n) = \log x + n$.
n est appelé *caractéristique* de log a et le développement décimal de log x est la *mantisse* de log a.
Les anciennes tables de logarithme ne donnaient en général que les mantisses.
L'entier n se calcule de tête, la difficulté du calcul de log a réside uniquement dans celle du calcul de sa mantisse.
Remarquons enfin que, puisque $1 \le x < 10$, nous aurons $0 \le \log x < 1$. La mantisse s'écrit donc sous la forme : $\log x = \overline{0,d_1d_2d_3\cdots}$.
Nous nous proposons de calculer les décimales d_1, d_2, d_3, …

Quelques remarques :

<u>Remarque 1</u> : Pour $a > 0$, $\log a = -\log\left(\dfrac{1}{a}\right)$. Ceci nous permet de limiter l'étude, au prix d'un éventuel changement de signe, au cas $a > 1$.

<u>Remarque 2</u> : Avec les notations précédentes : $10\log x = \overline{d_1,d_2d_3\cdots}$. En notant $[\alpha]$ la partie entière de α, il est clair que $d_1 = [10 \log x]$.
<u>Remarque 3</u> : Comme tout réel, on peut encadrer x^{10} entre deux puissances entières consécutives de 10 à savoir : $10^m \le x^{10} < 10^{m+1}$ (m est donc le nombre de chiffres, diminué de 1, de l'écriture décimale de la partie entière de x^{10}).
Et en passant aux logarithmes : $m \le 10 \log x < m + 1$.
En combinant cette inégalité avec le résultat de la remarque 2 on constate que $d_1 = m$.

<u>Remarque 4</u> : $\log\left(\dfrac{x^{10}}{10^{d_1}}\right) = 10\log x - d_1 = \overline{0,d_2d_3d_4\cdots}$. Cette remarque permettra d'itérer le processus.

À présent nous sommes en mesure de construire un algorithme permettant de calculer log a, a étant un décimal strictement positif donné. Nous nous limiterons à l'obtention de 10 décimales. Comme nous destinons cet algorithme à être mis en œuvre sur une calculette « 4 opérations » nous nous limiterons aux seules opérations $\boxed{+}$, $\boxed{-}$, $\boxed{\times}$, $\boxed{\div}$.

Un premier algorithme

L'algorithme ci-dessous a été obtenu à l'aide de la méthode déductive Médée.
Dans ce tableau « @» est utilisé pour désigner *l'ancienne valeur* d'une variable. Ainsi $n = @n + 1$ signifie que la variable n est augmentée de 1.

Lexique		Définitions
	7	*résultat* = *écrire* s, c, « , », d
	6	s : si $a < 1$ alors s = « - » sinon s = « + »
	3	$c = nbChiff\,(t)$
	4	$x = t/10^c$
	5	d : *répéter 10 fois*
		2 — *Ajouter* u *en queue de liste* d
		1 — $u = nbChiff\,(x^{10})$
		3 — $x = @\,x^{10}/10^u$
	2	t: si $a<1$ alors $t = \frac{1}{a}$ sinon $t = a$
	1	a = *donnée*

Lexique :

s : car	Signe du résultat + ou -
c : entier	La caractéristique de log a
d : liste entiers	Les 10 premières décimales de la mantisse.
t : décimal	Nombre dont on calcule effectivement le logarithme (cf remarque 1)
nbChiff : fonction	Retourne le nombre de chiffres, diminué de 1, de la partie entière d'un décimal positif écrit dans le système décimal.
u : entier	Décimale courante
x : décimal	Voir les explications ci-dessus
a : décimal	Nombre dont on calcule le logarithme

		Fonction *nbChiff(x)*
	4	*résultat* = n
	3	n : *répéter tant que* $v>10$
		1 — $n = @\,n+1$
		2 — $v = @\,v/10$
	2	$n_i = 0$
	1	$v_i = x$

n : entier	Valeur courante du nombre de chiffres de x
v : décimal	Résultat des divisions successives de x par 10

- Nous pouvons optimiser cet algorithme en remarquant que puisque le test $a < 1$ intervient à deux endroits différents nous pouvons regrouper les instructions correspondantes.
- x^{10} est calculé deux fois ; il est donc judicieux d'introduire une variable supplémentaire (que nous noterons p) pour stocker la valeur de x^{10}.

Voici une seconde version de l'algorithme. Nous en avons profité pour ordonner les instructions ce qui rend la colonne de numérotation inutile.

Définitions

a = *donnée*

si $a > 1$ alors $\begin{cases} s = "+" \\ t = a \end{cases}$ sinon $\begin{cases} s = "-" \\ t = 1/a \end{cases}$

$c = nbChiff\,(t)$

$x = t/10^c$

d : *répéter 10 fois*

- $p = x^{10}$
- $u = nbChiff\,(p)$
- *Ajouter* u *en queue de liste* d
- $x = p/10^u$

résultat = *écrire* s, c, « , », d

Fonction *nbChiff(v)*

$n = 0$

répéter tant que $v>10$

- $n = @\,n+1$
- $v = @\,v/10$

résultat = n

Cet algorithme peut être traduit directement en *Python*, *Scratch* ou *AlgoBox*.
Nous donnons en Annexe 1 sa traduction sous forme d'un tableau de calcul Excel.

Mais puisque nous le destinons à être mis en oeuvre à l'aide d'une calculette, il faut quelques ultimes adaptations :

- Nous avons déjà expliqué comment calculer x^{10}.
- Le nombre de chiffres d'un résultat sera « compté » en repérant la position de la virgule sur l'afficheur.
- Pour diviser p par 10^u on utilisera la séquence : $\boxed{\div}\boxed{1}\boxed{0}$ … $\boxed{0}$ (u fois la touche $\boxed{0}$). Cette façon de procéder permet de déplacer la virgule sans avoir à retaper le nombre affiché.

Calcul de log *a*

log *a* =			,									

1	Taper le décimal *a*	
2	**Si a > 1** écrire « **+** » dans la première case libre du tableau résultat	**Sinon** écrire « **-** » dans la première case libre du tableau résultat $\boxed{\div}\boxed{=}$
3	Compter le nombre *n* de chiffres, diminué de 1, de la partie entière de *a* Ecrire *n* dans la première case libre du tableau résultat	
4	$\boxed{\div}\boxed{1}\boxed{0}$…$\boxed{0}$ (*n* zéros) $\boxed{=}$	
5	$\boxed{\times}\boxed{=}\boxed{=}\boxed{=}\boxed{=}\boxed{=}\boxed{=}\boxed{=}\boxed{=}\boxed{=}$	
6	Retourner à l'étape 3	

Voici une illustration de mise en œuvre :

Calcul de log *2 718*

log *2718* =	+	3	,	4	3	4	2	4					

Etapes	Affichage	n
1	2718	
2	On écrit + dans le tableau résultat	
3	On écrit 3 dans le tableau résultat	3
4	2.718	
5	22003.636	
3	On écrit 4 dans le tableau résultat	4
4	2.2003636	
5	2660.3845	
3	On écrit 3 dans le tableau résultat	3
4	2.6603845	
5	17760.043	

3	On écrit 4 dans le tableau résultat	**4**
4	1.7760043	
5	312.20516	
3	On écrit 2 dans le tableau résultat	**2**
4	3.1220516	
5	87983.394	
3	On écrit 4 dans le tableau résultat	**4**
…	**etc.**	

Tout semble pour le mieux dans le meilleur des mondes…
Lorsque l’afficheur de la calculatrice a une capacité de 10 chiffres, l’algorithme précédent fonctionne parfaitement abstraction faite du cumul éventuel des erreurs d’arrondi qui ne sont pas étudiées ici.
Lorsque l’afficheur de la calculatrice ne comporte que 8 chiffres (c’est actuellement le cas le plus courant), on peut se heurter à un problème de dépassement de capacité dans le calcul de x^{10}. La calculette se bloque alors et une remise à zéro est nécessaire (touche [CA] en général).
Voici une parade (imparfaite car elle occasionne la perte d’un chiffre significatif dans les calculs) : x^{10} est supérieur à 10^8, et donc ne peut plus être affiché sur 8 positions, dès que $x > 10^{0,8} \simeq 6{,}309\,573\,5$. Dans ce cas la décimale d_i cherchée est nécessairement « 9 ». Pour obtenir le chiffre suivant du développement décimal, il faut donc calculer $\dfrac{x^{10}}{10^9}$. Or $\dfrac{x^{10}}{10^9} = \left(\dfrac{x^{10}}{10^{10}}\right) \times 10 = \left(\dfrac{x}{10}\right)^{10} \times 10$ et puisque $\dfrac{x}{10}$ est inférieur à $10^{0,8}$, on peut à nouveau calculer sa puissance dixième à l’aide de la calculette… Il suffira de rétablir la situation en multipliant le résultat par 10.
L’algorithme ci-dessous tient compte de cette remarque :

1	Taper le décimal *a*	
2	**Si a > 1 alors** écrire « **+** » dans la première case libre du tableau résultat	**Sinon** **a)** écrire « **-** » dans la première case libre du tableau résultat **b)** [+] [=]
3	Compter le nombre *n* de chiffres, diminué de 1, de la partie entière de *a*. Ecrire *n* dans la première case libre du tableau résultat	
4	[÷] [1] [0] **…** [0] **(*n* zéros)** [=]	
5	**<u>Répéter</u> <u>*tant que*</u> affichage > 6,3095735** **a)** Ecrire 9 dans la première case libre du tableau résultat **b)** [÷] [1] [0] [=] **c)** [×] [=] [=] [=] [=] [=] [=] [=] [=] [=] **d)** [×] [1] [0] [=]	
6	[×] [=] [=] [=] [=] [=] [=] [=] [=] [=]	
7	**Aller en 3**	

Nous donnons un dernier exemple de mise en œuvre :

Calcul de log *314*

log *314* =	+	2	,	4	9	6	9	2	9	6			

Etapes	Affichage	n
1	314	
2	On écrit + dans le tableau résultat	
3	On écrit 2 dans le tableau résultat	2
4	3.14	
6	93174.369	
3	On écrit 4 dans le tableau résultat	4
4	9.3174369	
5a	On écrit 9 dans le tableau résultat	
5b	0.9317436	
5c	0.4931328	
5d	4.931328	
6	8504304.4	
3	On écrit 6 dans le tableau résultat	6
4	8.5043044	
5a	On écrit 9 dans le tableau résultat	
5b	0.8504304	
5c	0.1978733	
5d	1.978733	
6	920.17806	
3	On écrit 2 dans le tableau résultat	2
4	9.2017806	
5a	On écrit 9 dans le tableau résultat	
5b	0.920178	
5c	0.4352293	
5d	4.352293	
6	2438828	
3	On écrit 6 dans le tableau résultat	6
…	etc.	

Une calculatrice scientifique donne le même résultat avec une dépense énergétique bien moindre…

Annexe 1

Log =			,										

Etapes	Affichage	n

Annexe 2

(Mise en oeuvre du premier algorithme à l'aide d'une feuille Excel)

	A	B	C	D	E	F	G	H	I	
1	3,14									
2		*boucle1*	*boucle2*	*boucle3*	*boucle4*	*boucle5*	*boucle6*	*boucle7*	*boucle8*	*bo*
3	1	3,14	9,317437339	4,931338417	8,504486052	1,979159212	9,221622111	4,447062313	3,025047159	6,4
4	2	9,8596	86,81463856	24,31809858	72,32628301	3,917071187	85,03831437	19,77636322	9,150910313	41,
5	3	30,959144	808,8899549	119,9207737	615,0978651	7,752507525	784,1912001	87,94671957	27,68193524	264
6	4	97,21171216	7536,781468	591,3699185	5231,091214	15,34344669	7231,51491	391,1045422	83,73915956	16
7	5	305,2447762	70223,48907	2916,245198	44487,74227	30,36712385	66686,29779	1739,26627	253,3149067	108
8	6	958,4685972	654302,9591	14380,99197	378345,3836	60,10137292	614955,8383	7734,625483	766,2895388	69
9	7	3009,591395	6096426,822	70917,53819	3217633,038	118,9501859	5670890,356	34396,3615	2318,061992	44
10	8	9450,116981	56803074,9	349718,3805	27364315,29	235,4213562	52294807,9	152962,7629	7012,246843	287
11	9	29673,36732	529259091	1724579,685	232719437,7	465,9363458	482242956,8	680234,9384	21212,37739	184
12	10	93174,37339	4931338417	8504486,052	1979159212	922,1622111	4447062313	3025047,159	64168,44196	118
13										
14		93 174	4 931 338 416	8 504 486	1 979 159 212	922	4 447 062 313	3 025 047	64 168	118
15	0,	4	9	6	9	2	9	6	4	

- La donnée *a* est placée en A1.
- Pour ne pas alourdir la complexité des formules, les calculs ne sont corrects que pour $1 < a < 9$.
- Il est fait usage de la fonction ENT (partie entière).
- La formule placée en B15 compte le nombre de caractères (moins un) de la partie entière placée en B14
- Pour obtenir davantage de décimales il suffit de copier la colonne C dans les colonnes suivantes…

	A	B	C
1	3,14		
2		*boucle1*	*boucle2*
3	1	=A1	=B12/10^B15
4	=A3+1	=B3*A1	=C3*C$3
5	=A4+1	=B4*A1	=C4*C$3
6	=A5+1	=B5*A1	=C5*C$3
7	=A6+1	=B6*A1	=C6*C$3
8	=A7+1	=B7*A1	=C7*C$3
9	=A8+1	=B8*A1	=C8*C$3
10	=A9+1	=B9*A1	=C9*C$3
11	=A10+1	=B10*A1	=C10*C$3
12	=A11+1	=B11*A1	=C11*C$3
13			
14		=ENT(B12)	=ENT(C12)
15	0,	=NBCAR(TEXTE(B14;"#"))-1	=NBCAR(TEXTE(C14;"#"))-1
16			

Annexe 3

(Extrait d'une table de logarithmes)
Bouvart & Ratinet (Librairie Hachette 1957)

II. Logarithmes des nombres de 1 à 10 000.

15	
1	1,5
2	3,0
3	4,5
4	6,0
5	7,5
6	9,0
7	10,5
8	12,0
9	13,5

14	
1	1,4
2	2,8
3	4,2
4	5,6
5	7,0
6	8,4
7	9,8
8	11,2
9	12,6

13	
1	1,3
2	2,6
3	3,9
4	5,2
5	6,5
6	7,8
7	9,1
8	10,4
9	11,7

12	
1	1,2
2	2,4
3	3,6
4	4,8
5	6,0
6	7,2
7	8,4
8	9,6
9	10,8

N	0	1	2	3	4	5	6	7	8	9
300	47 712	727	741	756	770	784	799	813	828	842
1	857	871	885	900	914	929	943	958	972	986
2	48 001	015	029	044	058	073	087	101	116	130
3	144	159	173	187	202	216	230	244	259	273
4	287	302	316	330	344	359	373	387	401	416
5	430	444	458	473	487	501	515	530	544	558
6	572	586	601	615	629	643	657	671	686	700
7	714	728	742	756	770	785	799	813	827	841
8	855	869	883	897	911	926	940	954	968	982
9	996	*010	*024	*038	*052	*066	*080	*094	*108	*122
310	49 136	150	164	178	192	206	220	234	248	262
1	276	290	304	318	332	346	360	374	388	402
2	415	429	443	457	471	485	499	513	527	541
3	554	568	582	596	610	624	638	651	665	679
4	693	707	721	734	748	762	776	790	803	817
5	831	845	859	872	886	900	914	927	941	955
6	969	982	996	*010	*024	037	*051	*065	*079	*092
7	50 106	120	133	147	161	174	188	202	215	229
8	243	256	270	284	297	311	325	338	352	365
9	379	393	406	420	433	447	461	474	488	501
320	515	529	542	556	569	583	596	610	623	637
1	651	664	678	691	705	718	732	745	759	772
2	786	799	813	826	840	853	866	880	893	907
3	920	934	947	961	974	987	*001	*014	*028	*041
4	51 055	068	081	095	108	121	135	148	162	175
5	188	202	215	228	242	255	268	282	295	308
6	322	335	348	362	375	388	402	415	428	441
7	455	468	481	495	508	521	534	548	561	574
8	587	601	614	627	640	654	667	680	693	706
9	720	733	746	759	772	786	799	812	825	838
330	851	865	878	891	904	917	930	943	957	970
1	983	996	*009	*022	*035	*048	*061	*075	*088	*101
2	52 114	127	140	153	166	179	192	205	218	231
3	244	257	270	284	297	310	323	336	349	362
4	375	388	401	414	427	440	453	466	479	492
5	504	517	530	543	556	569	582	595	608	621
6	634	647	660	673	686	699	711	724	737	750
7	763	776	789	802	815	827	840	853	866	879
8	892	905	917	930	943	956	969	982	994	*007
9	53 020	033	046	058	071	084	097	110	122	135
340	148	161	173	186	199	212	224	237	250	263
1	275	288	301	314	326	339	352	364	377	390
2	403	415	428	441	453	466	479	491	504	517
3	529	542	555	567	580	593	605	618	631	643
4	656	668	681	694	706	719	732	744	757	769
5	782	794	807	820	832	845	857	870	882	895
6	908	920	933	945	958	970	983	995	*008	*020
7	54 033	045	058	070	083	095	108	120	133	145
8	158	170	183	195	208	220	233	245	258	270
9	283	295	307	320	332	345	357	370	382	394
N	0	1	2	3	4	5	6	7	8	9