Sur la résolution numérique d'équations…

Préambule

Quel plaisir de reprendre en main, ici, pour l'occasion, mon manuel d'élèves de

MATHÉMATIQUES
Terminales C et E
ANALYSE ET STATISTIQUES
IREM • STRASBOURG
Librairie Istra, 1983

J'en profite pour dire tout le bien que je pense de cette collection et que, d'aussi loin que je me souvienne, c'est peut-être dans ces livres que j'ai vu pour la première fois un algorithme écrit sous forme d'**organigramme**.
Quoi de plus normal alors de retrouver le formidable travail fait sur la résolution numérique d'équations, et de le mouliner avec LARP !

Prérequis

On peut démontrer en Tle S le résultat suivant, conséquence directe du théorème des valeurs intermédiaires : « Si *f* est une fonction continue sur $[a;b]$ et si $f(a)f(b)<0$, alors l'équation $f(x)=0$ a au moins une solution dans $]a;b[$; si de plus, *f* est strictement monotone sur $[a;b]$, l'équation $f(x)=0$ a une unique solution dans $]a;b[$ ».
On suppose dans la suite que l'étude de la fonction *f* a permis de localiser et de séparer de la sorte les racines.
Les méthodes décrites ci-après permettent de déterminer une valeur approchée de ces racines avec la précision souhaitée.

Sur la plan théorique, on trouve bien sûr d'innombrables choses dans la littérature papier ou écran…
Mon objectif ici consiste simplement à (re)mettre en lumière ces méthodes algorithmiques, au travers de mon logiciel de prédilection en la matière : LARP.

A. Méthode de dichotomie[1]

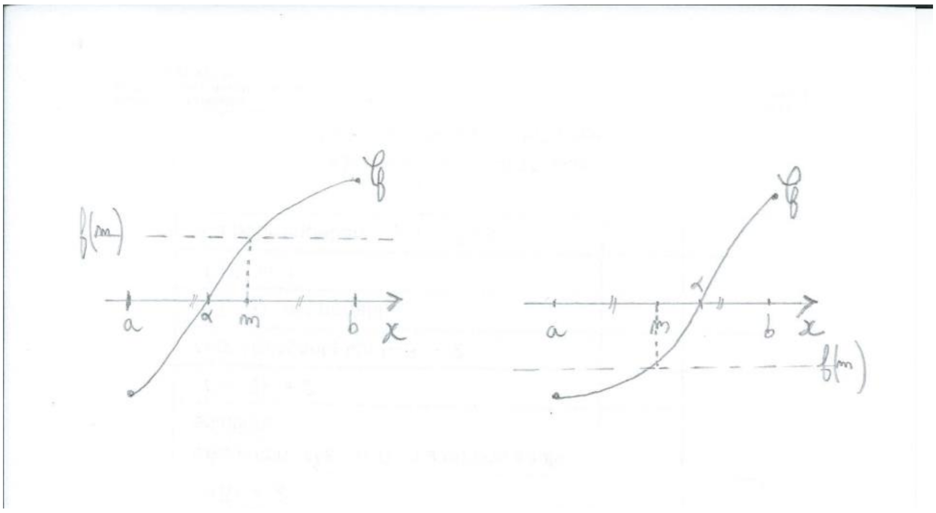
A.1 Description de la méthode

On démarre donc en ayant localisé la racine α entre *a* et *b* ($a<\alpha<b$) ; et on sait par exemple que sur cet intervalle, la fonction *f* (continue) est strictement croissante.

On calcule alors le milieu *m* de *a* et *b* ($m=\frac{a+b}{2}$), puis l'image par *f* de celui-ci $f(m)$, et on la compare à 0. Les deux cas décrits ci-dessous peuvent alors se produire :

[1] Dite aussi « de la bissection »… puisqu'on coupe en deux à chaque itération…

MANGANELLI Stéphan
LEGTA « Louis GIRAUD »
84 200 CARPENTRAS-SERRES



Ici, $f(m)>0$
et on sait alors, du coup, que :
$a<\alpha<m$.
Puis l'on recommence la bissection comme précédemment avec
pour nouvelle borne gauche $a=a$
pour nouvelle borne droite $b=m$.
Et ainsi de suite…

Ici, $f(m)<0$
et on sait alors, du coup, que :
$m<\alpha<b$.
Puis l'on recommence la bissection comme précédemment avec
pour nouvelle borne gauche $a=m$
pour nouvelle borne droite $b=b$.
Et ainsi de suite…

A.2 Résultats, commentaires

On construit, de la sorte, deux suites (a_n) et (b_n) telles que

$a_0=a$, $b_0=b$ puis

SI $f\left(\frac{a_n+b_n}{2}\right)\geq 0$

$$\begin{cases} a_{n+1}=a_n \\ b_{n+1}=\frac{a_n+b_n}{2} \end{cases}$$

SI $f\left(\frac{a_n+b_n}{2}\right)<0$

$$\begin{cases} a_{n+1}=\frac{a_n+b_n}{2} \\ b_{n+1}=b_n \end{cases}$$

On montre alors que ces deux suites sont adjacentes ((a_n) croissante et (b_n) décroissante), de limite commune α, la racine "visée" de l'équation $f(x)=0$.

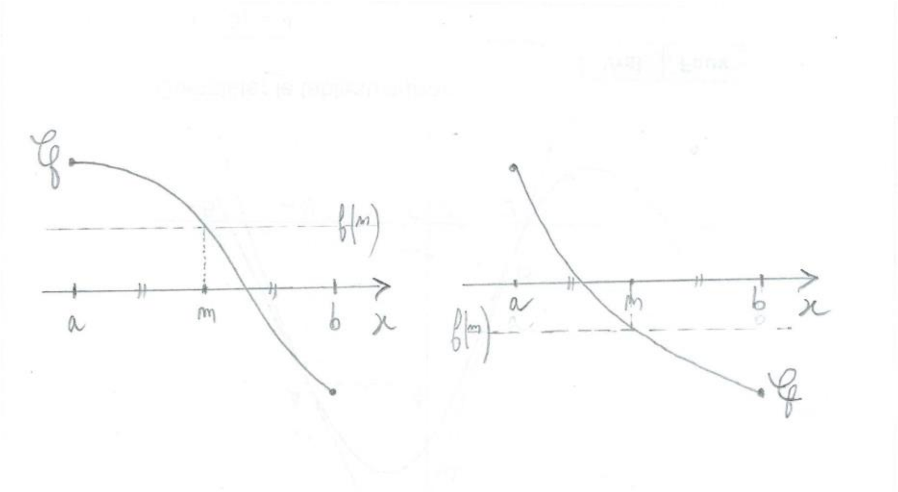
Remarques

(**R1**) Il faut bien prévoir le cas où l'on tombe exactement sur la racine ($f(m)=0$) ; nous le prévoyons dans le test (ici côté gauche avec le cas positif), de façon complètement arbitraire. À partir de là, la suite (b_n) devient stationnaire, égale à la racine α visée ; l'algorithme ne passe plus qu'alors côté droit, avec la suite (a_n) qui, seule, continue à croître et se rapprocher de α.

(**R2**) Par construction, après *n* itérations, on obtient $a_n<\alpha\leq b_n$, avec $b_n-a_n=\frac{b-a}{2^n}$.

L'encadrement de la racine α est donc d'amplitude $\frac{b-a}{2^n}$. On comprend alors que la convergence dont on vient de parler plus haut est d'ordre *géométrique*. On peut donc faire tourner cet algorithme de construction, **TANT QUE** l'on obtient pas la **précision souhaitée**.

(**R3**) Dans le cas où *f* est strictement décroissante sur $[a;b]$, il suffit juste d'inverser dans le test, l'avancement des deux suites… ou bien, ce qui revient au même, d'inverser la condition du test…



SI $f(\frac{a_n+b_n}{2})\geq 0$

$$\begin{cases} a_{n+1}=\frac{a_n+b_n}{2} \\ b_{n+1}=b_n \end{cases}$$

SI $f(\frac{a_n+b_n}{2})<0$

$$\begin{cases} a_{n+1}=a_n \\ b_{n+1}=\frac{a_n+b_n}{2} \end{cases}$$

A.3 Algorithme et programme avec LARP

A.3.1 Un premier exemple

Comme exemple, nous prendrons l'exercice 4, Bac ES, Nouvelle-Calédonie, Mars 2014.

> Soit f la fonction définie sur l'intervalle $[2\,;5]$ par
>
> $$f(x)=(3-x)\mathrm{e}^x+1,$$
>
> soit f' sa fonction dérivée et soit f'' sa fonction dérivée seconde.
>
> **1.** Montrer que, pour tout nombre réel x appartenant à l'intervalle $[2\,;5]$, $f'(x)=(2-x)\mathrm{e}^x$ et $f''(x)=(1-x)\mathrm{e}^x$.
>
> **2.** Étudier les variations de la fonction f sur l'intervalle $[2\,;5]$.
>
> **3.** Justifier que l'équation $f(x)=0$ admet une unique solution α dans l'intervalle $[2\,;5]$.
> Montrer que : $3<\alpha<4$.

On montre que *f* est strictement décroissante sur son intervalle $[2;5]$ de définition.
Le théorème de la bijection permet alors de répondre à la question 3.

La question 5 (ci-dessous) ensuite propose un algorithme de dichotomie pour approcher α …

5. On considère l'algorithme suivant :

Variables :	a, b, m et r sont des nombres réels
Initialisation :	Affecter à a la valeur 3
	Affecter à b la valeur 3,05
Entrée :	Saisir r
Traitement :	TANT QUE $b - a > r$
	Affecter à m la valeur $\dfrac{a+b}{2}$
	SI $f(m) > 0$
	ALORS Affecter à a la valeur m
	SINON Affecter à b la valeur m
	FIN SI
	FIN TANT QUE
Sortie :	Afficher a.
	Afficher b

a. Faire fonctionner l'algorithme précédent avec $r = 0{,}01$ en recopiant et complétant le tableau ci-dessous. On arrondira au millième les valeurs de $f(m)$.

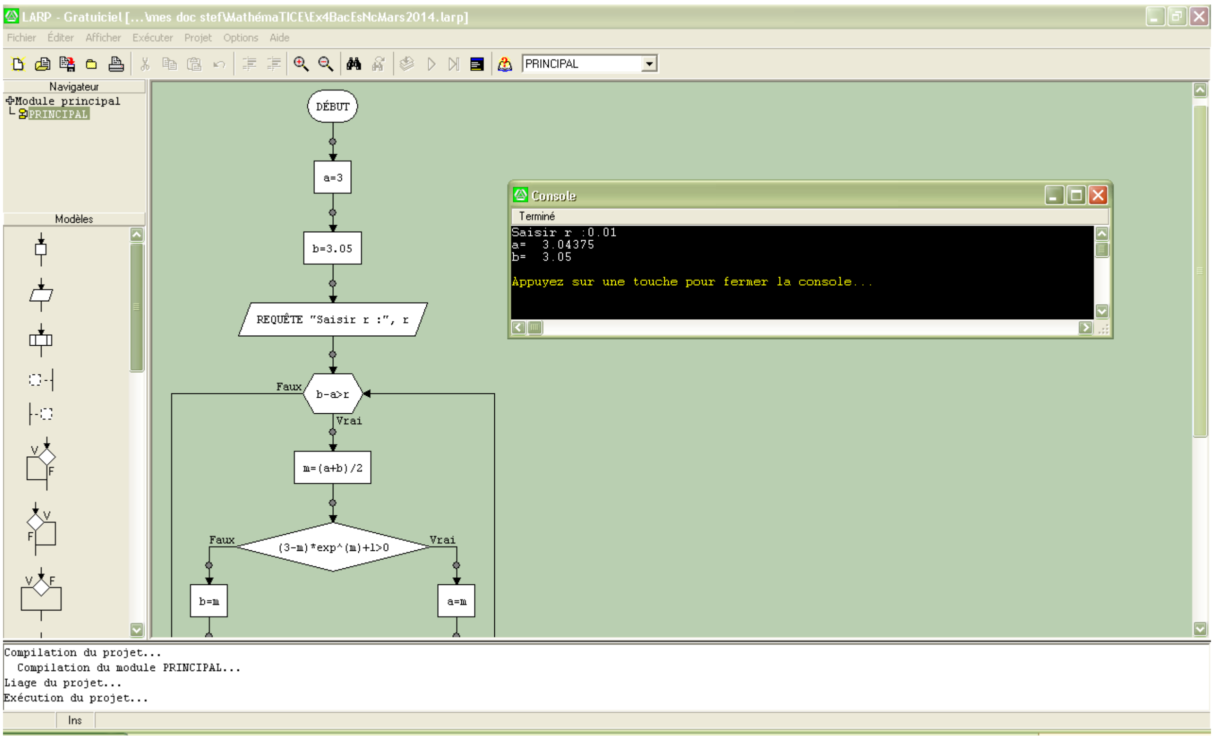
	$b-a$	$b-a>r$	m	$f(m)$	$f(m)>0$	a	b
Initialisation						3	3,05
étape 1	0,05	oui	3,025	0,485	oui	3,025	3,05
étape 2							
étape 3							

b. Interpréter les résultats trouvés pour a et b à la fin de l'étape 3.

Remarques

(**R1**) Écrit comme cela, l'algorithme ne fonctionnera pas dans le sens où, pour la condition du test, il faudra rentrer la formule de *f*, ou prévoir de la définir dans un module (sous-programme) auxiliaire…

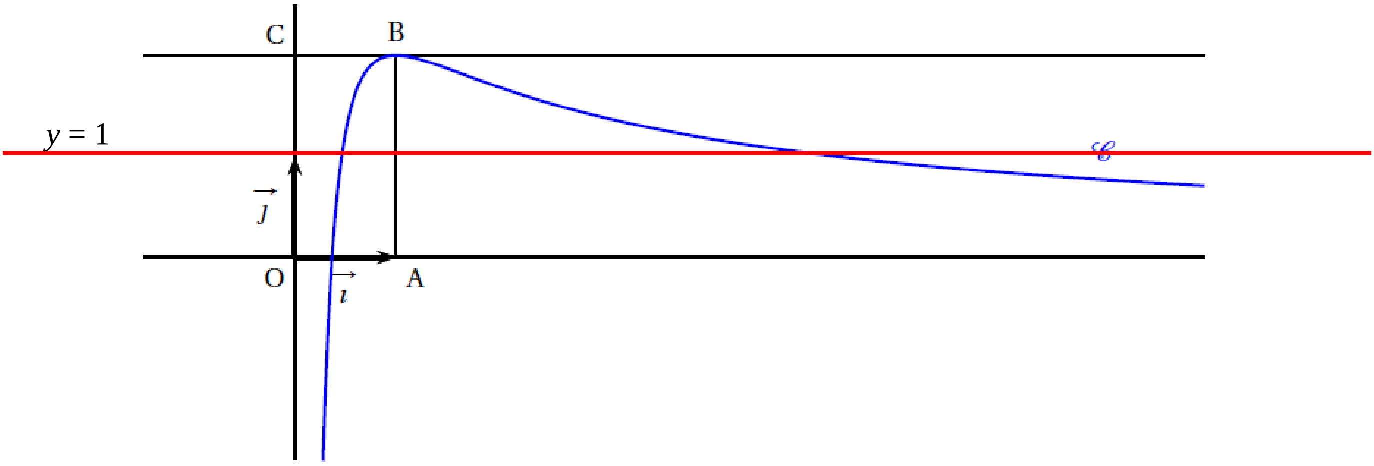
(**R2**) On peut vérifier que l'on est bien dans le cas de traitement d'une fonction décroissante… Le cas où $f(m)=0$ est ici prévu avec le cas négatif.



A.3.2 Un deuxième exemple

<u>Remarque</u> : Bien que l'on peut toujours se ramener à résoudre $f(x)=0$, il arrive aussi que l'algorithme proposé s'opère sur une équation du type $f(x)=k$ (avec un k par forcément nul). C'était le cas lors du dernier sujet de Bac S proposant une dichotomie :

Métropole, Juin 2013, avec $f(x)=\dfrac{2+2\ln x}{x}$ sur $]0;+\infty[$…



Le sens de variation de la fonction et le théorème de la bijection permettait de répondre à la question 3. ci-dessous.

3. **a.** Démontrer que l'équation $f(x) = 1$ admet une unique solution α sur l'intervalle $]0, 1]$.

b. Par un raisonnement analogue, on démontre qu'il existe un unique réel β de l'intervalle $]1, +\infty]$ tel que $f(\beta) = 1$.

Déterminer l'entier n tel que $n < \beta < n+1$.

4. On donne l'algorithme ci-dessous.

```
Variables :        a, b et m sont des nombres réels.
Initialisation :   Affecter à a la valeur 0.
                   Affecter à b la valeur 1.
Traitement :       Tant que b − a > 0,1
                      | Affecter à m la valeur (1/2)(a + b).
                      | Si f(m) < 1 alors Affecter à a la valeur m.
                      | Sinon Affecter à b la valeur m.
                      | Fin de Si.
                   Fin de Tant que.
Sortie :           Afficher a.
                   Afficher b.
```

a. Faire tourner cet algorithme en complétant le tableau ci-dessous que l'on recopiera sur la copie.

	étape 1	étape 2	étape 3	étape 4	étape 5
a	0				
b	1				
$b-a$					
m					

b. Que représentent les valeurs affichées par cet algorithme ?

c. Modifier l'algorithme ci-dessus pour qu'il affiche les deux bornes d'un encadrement de β d'amplitude 10^{-1}.

<u>Remarques</u>

(**R1**) Écrit comme cela, l'algorithme ne fonctionnera pas dans le sens où, pour la condition du test, il faudra rentrer la formule de *f*, ou prévoir de la définir dans un module (sous-programme) auxiliaire…

(**R2**) La saisie de la précision n'étant pas demandée mais directement implantée dans le programme (ici 0,1), l'utilisateur doit aller la modifier à chaque fois… Une requête comme dans l'exercice précédent semble plus pertinente.

(**R3**) Pour répondre à la question c., on doit :

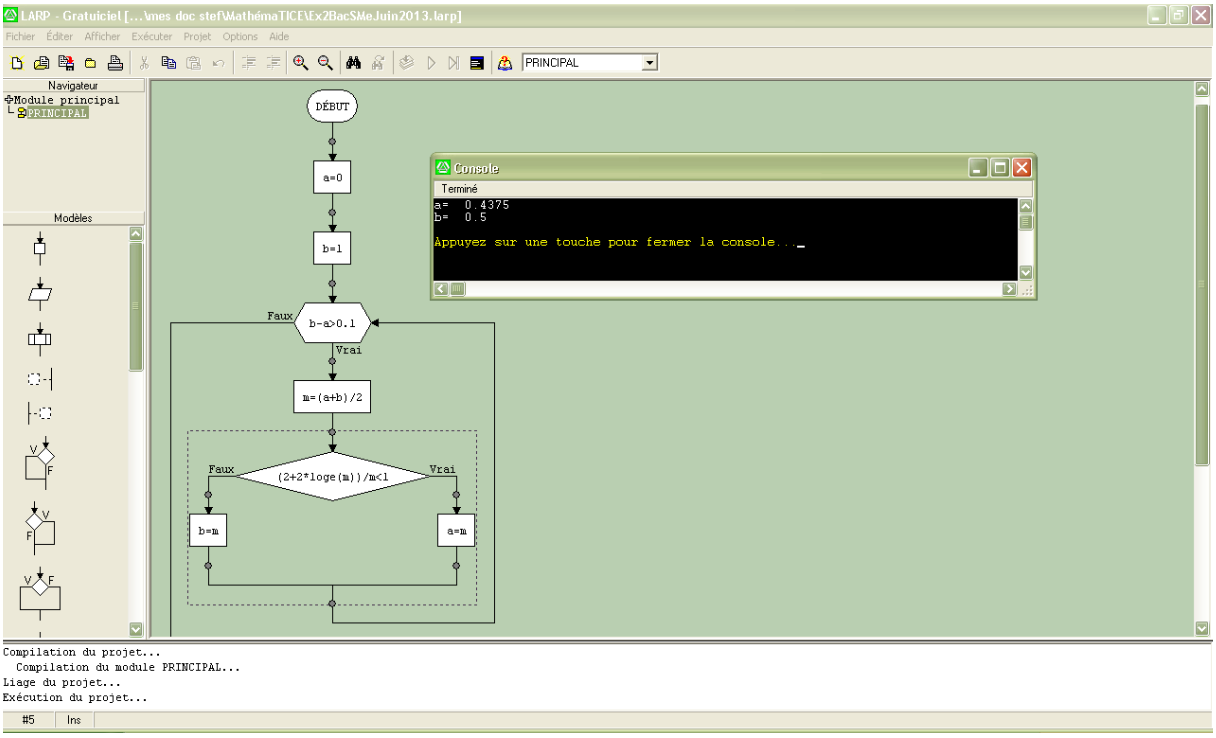
1. Modifier les bornes initiales en prenant les entiers *n* et *n*+1 trouvés précédemment (Q 3.b.) et qui encadrent la racine β, soit 5 et 6.
2. Faire l'inversion dans le Test conditionnel pour l'adapter à la décroissance de la fonction sur l'intervalle $]1;+\infty[$

☞ en retournant l'inégalité dans la condition

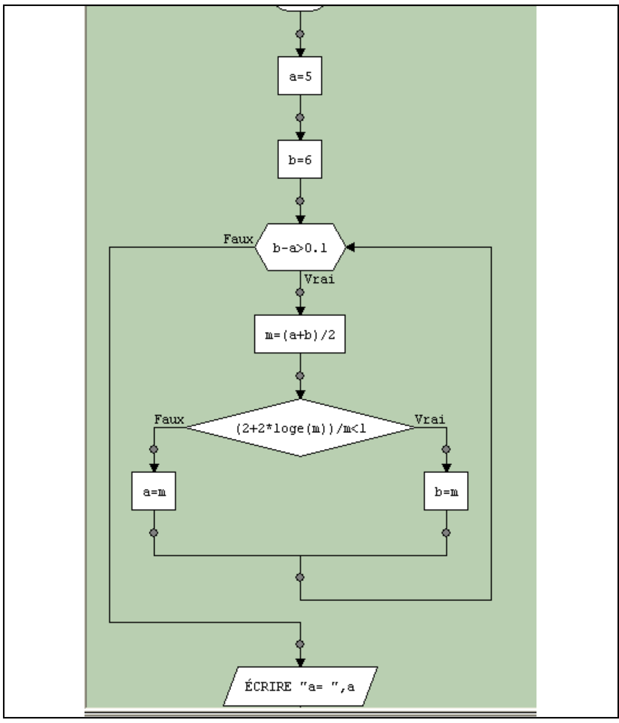
OU

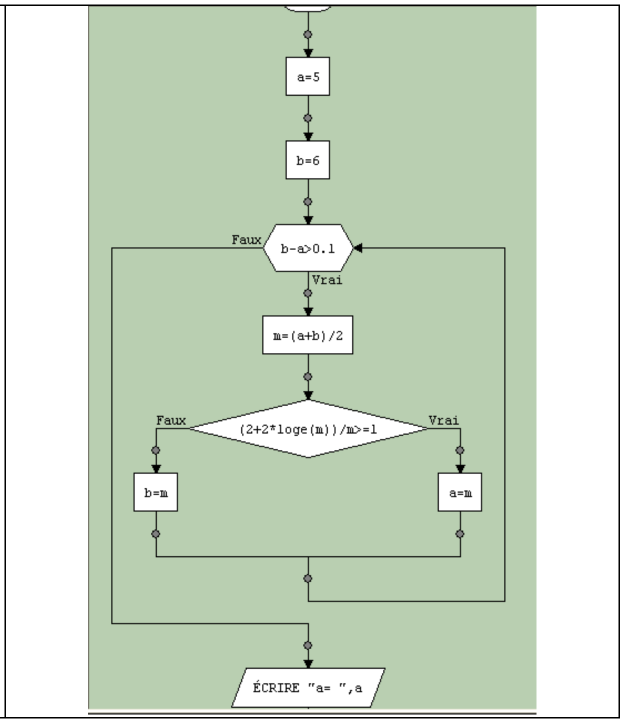
☞ en échangeant l'ordre d'avancement des deux suites

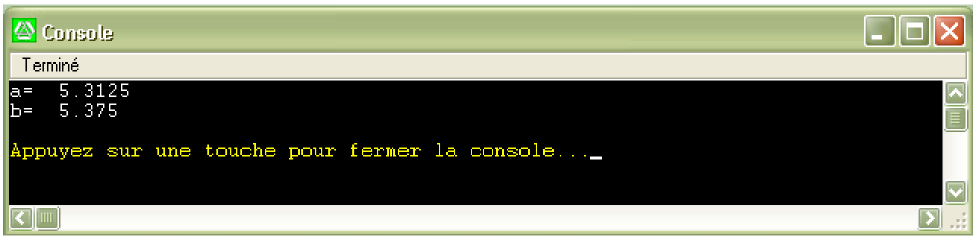
L’algorithme de l’énoncé (pour réponse aux questions 4.a. et 4.b.)



Les deux possibilités pour réponse à la question 4.c.



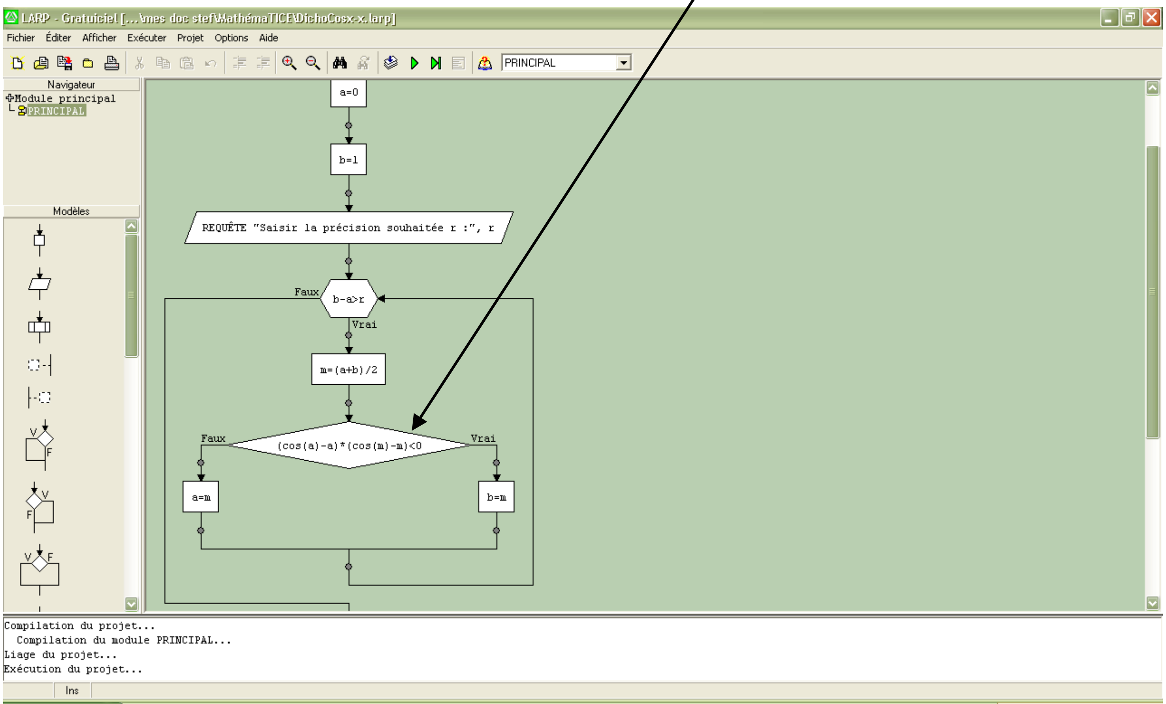


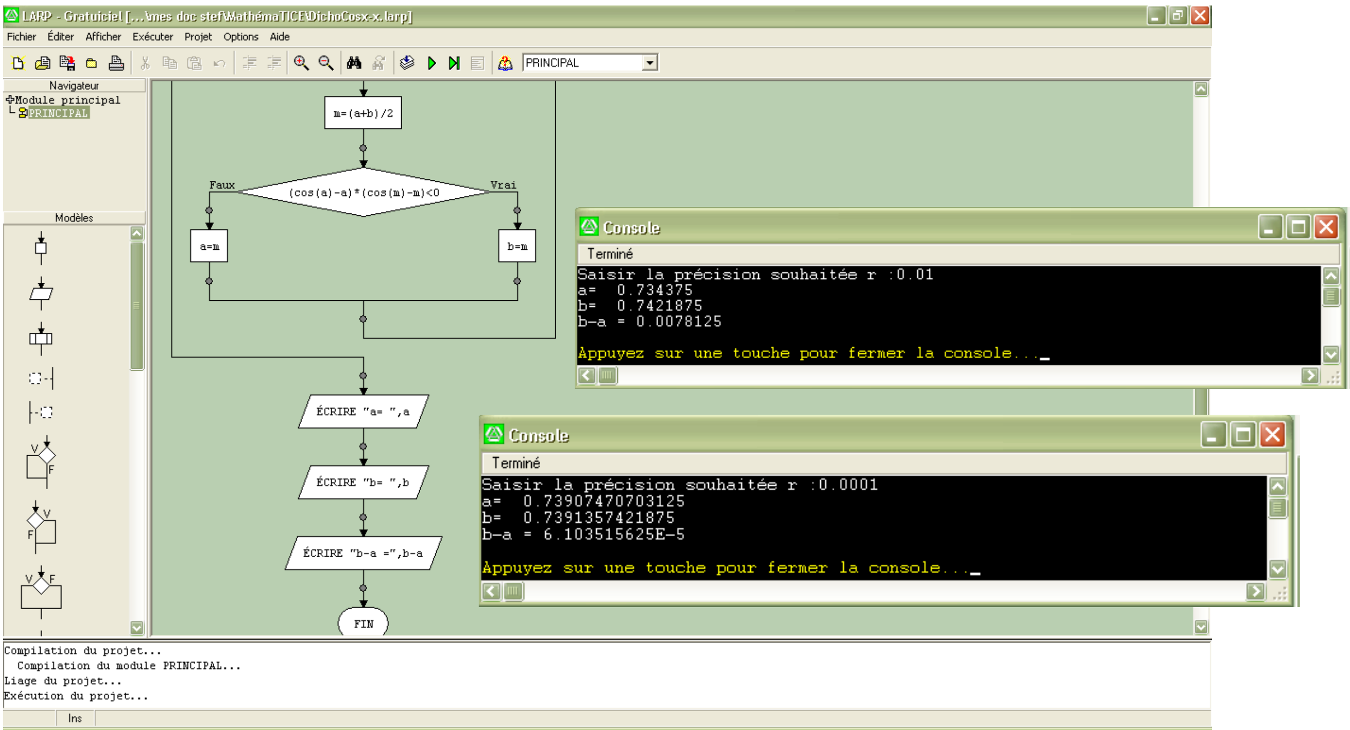


A.3.3 Un troisième exemple

Pour comparer les trois méthodes proposées ici, nous suivrons l'exemple suivant dans chaque partie (A, B et C) : Approximation de l'unique solution de l'équation $\cos x = x$.
On peut mettre l'équation sous la forme $\cos x - x = 0$ et chercher donc l'unique racine de la fonction $f : x \longmapsto \cos x - x$, strictement décroissante sur IR et s'annulant entre 0 et 1.

<u>Remarque préalable</u> : on teste ici le signe du produit $f(a) \times f(m)$ qui permet de traiter de la même façon les deux sens de variation…
En effet, quel que soit ce sens, si ce produit est négatif, cela signifie que la racine se situe entre *a* et *m* ; et c'est donc *m* qui devient la nouvelle borne *b* supérieure !



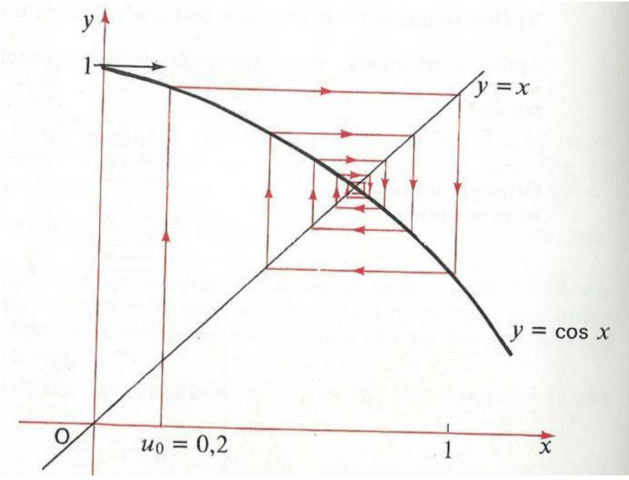


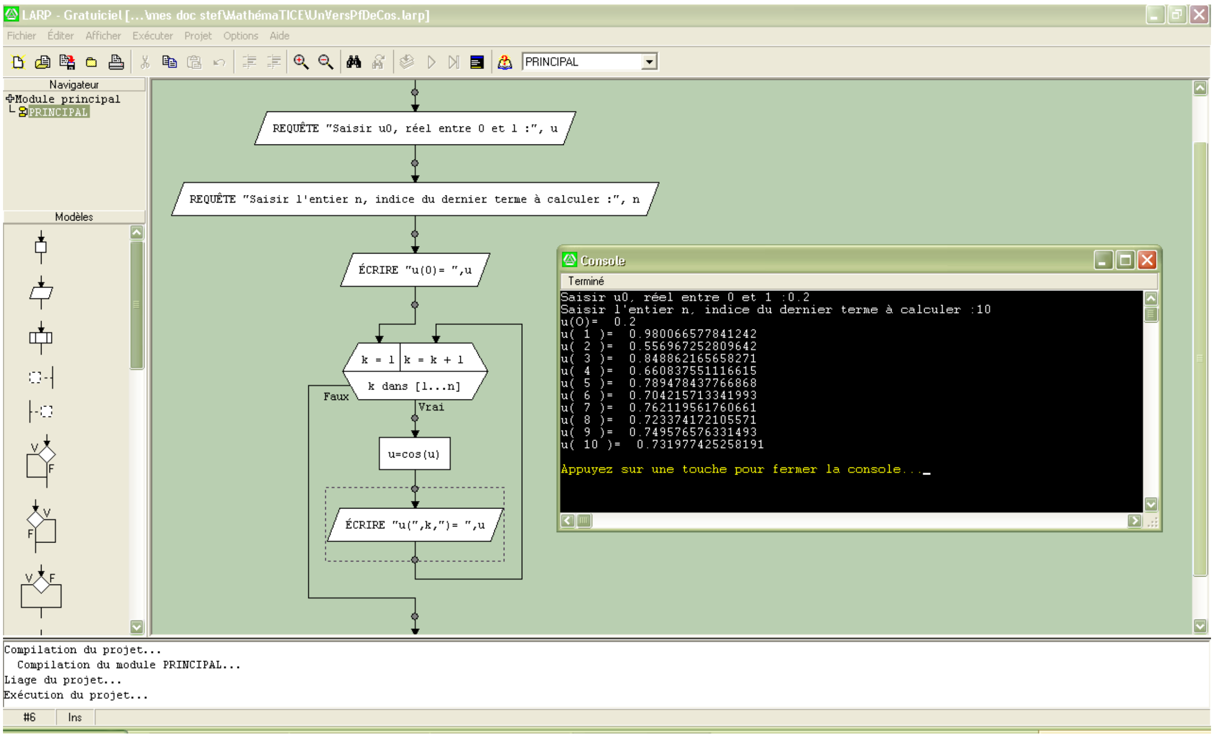
B. Méthode de la tangente (ou de Newton-Raphson, 1669)

Nous travaillerons sur l'exemple précédent de l'équation $\cos x = x$.

B.1 Approche par une suite récurrente…

On démontre que la suite (u_n) définie par $\begin{cases} u_0 \in [0;1] \\ u_{n+1} = \cos u_n \end{cases}$ est convergente ; sa limite est donc l'unique point fixe de la fonction cos !





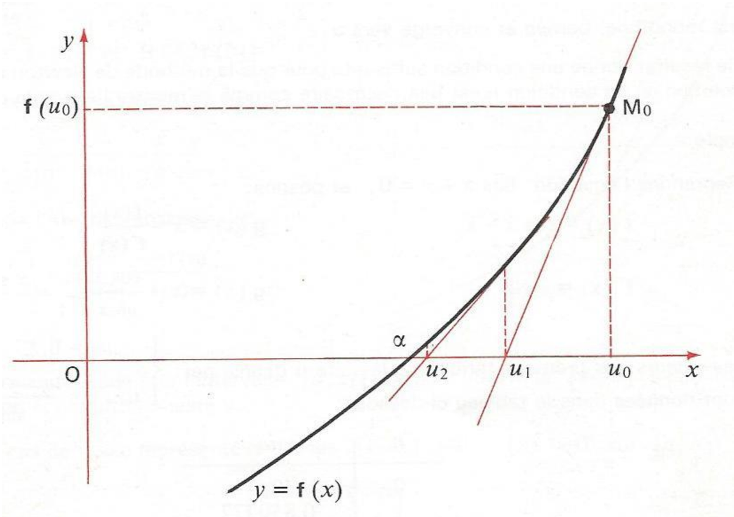
<u>Remarque</u> : On observe, aussi bien graphiquement que numériquement, que la convergence n'est pas très rapide. On peut montrer que (u_n) converge un peu plus vite qu'une suite géométrique de raison 0,85 puisque $|u_n - \alpha| < |u_0 - \alpha| \times (0,85)^n$.

L'idée consiste alors, pour accélérer la convergence, de transformer l'équation de départ de sorte que l'on puisse majorer par une suite géométrique de raison la plus proche de 0…

B.2 Une amélioration : la méthode de Newton

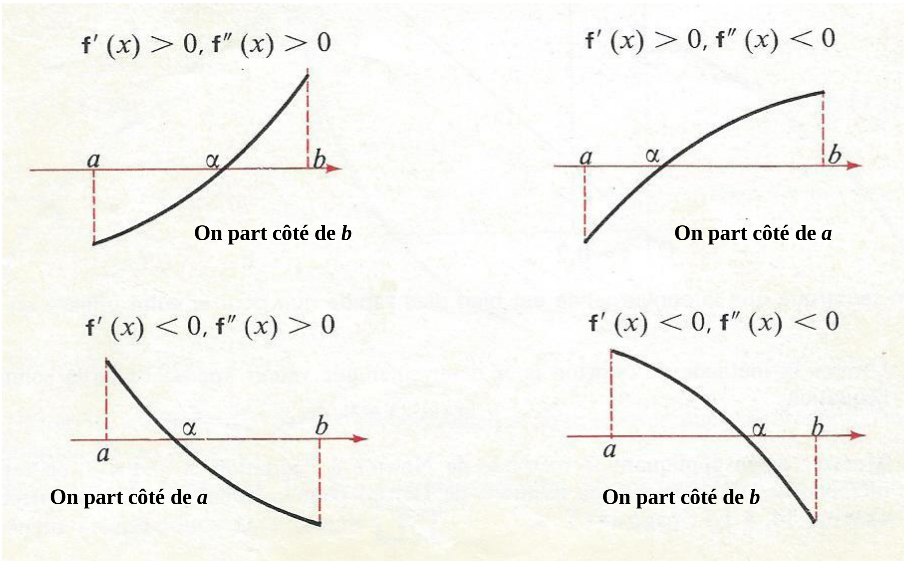
On met alors l'équation $f(x)=0$ sous la forme $x-\dfrac{f(x)}{f'(x)}=0$, et on construit la suite récurrente vérifiant $u_{n+1}=u_n-\dfrac{f(u_n)}{f'(u_n)}$.

Par construction, u_{n+1} est l'abscisse du point d'intersection avec l'axe des abscisses, de la tangente à $\mathscr{C}_f$ au point M_n d'abscisse u_n : d'où le nom donnée à cette méthode !



Supposons que l'on a localisé la racine dans un intervalle sur lequel f' et f'' gardent un signe constant…

On montre alors que, dans chacun des quatre cas possibles (ci-dessous), la suite (u_n) définie par $\begin{cases} u_0\in[a;b] \\ u_{n+1}=u_n-\dfrac{f(u_n)}{f'(u_n)} \end{cases}$, avec $f(u_0)\times f'(u_0)>0$ (pour partir du bon côté ! [2]), est monotone, bornée et convergente vers α.



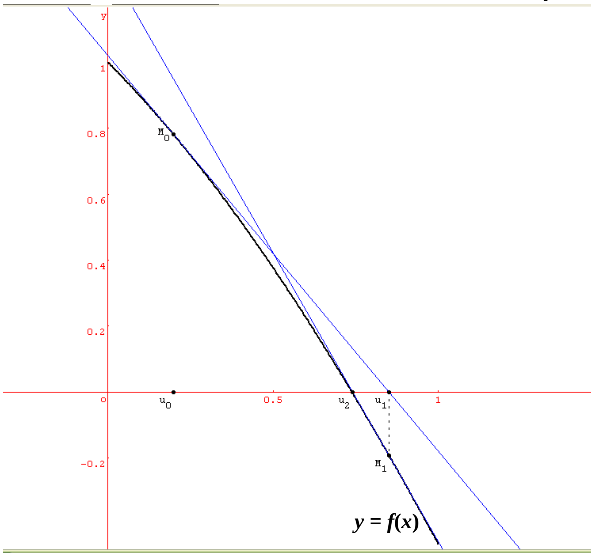
[2] Cette condition est suffisante mais non nécessaire ; voir le traitement de l'exemple (démarrage à $u_0=0{,}2$)

Pour notre équation $\cos x = x$, que l'on écrit ici $\cos x - x = 0$, on a :

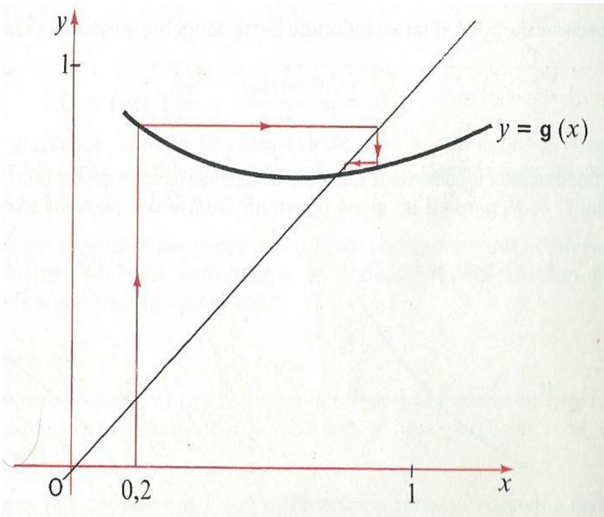
$$f(x) = \cos x - x \qquad g(x) = x - \frac{f(x)}{f'(x)}$$

$$f'(x) = -\sin x - 1 \qquad g(x) = \frac{x \sin x + \cos x}{\sin x + 1}$$

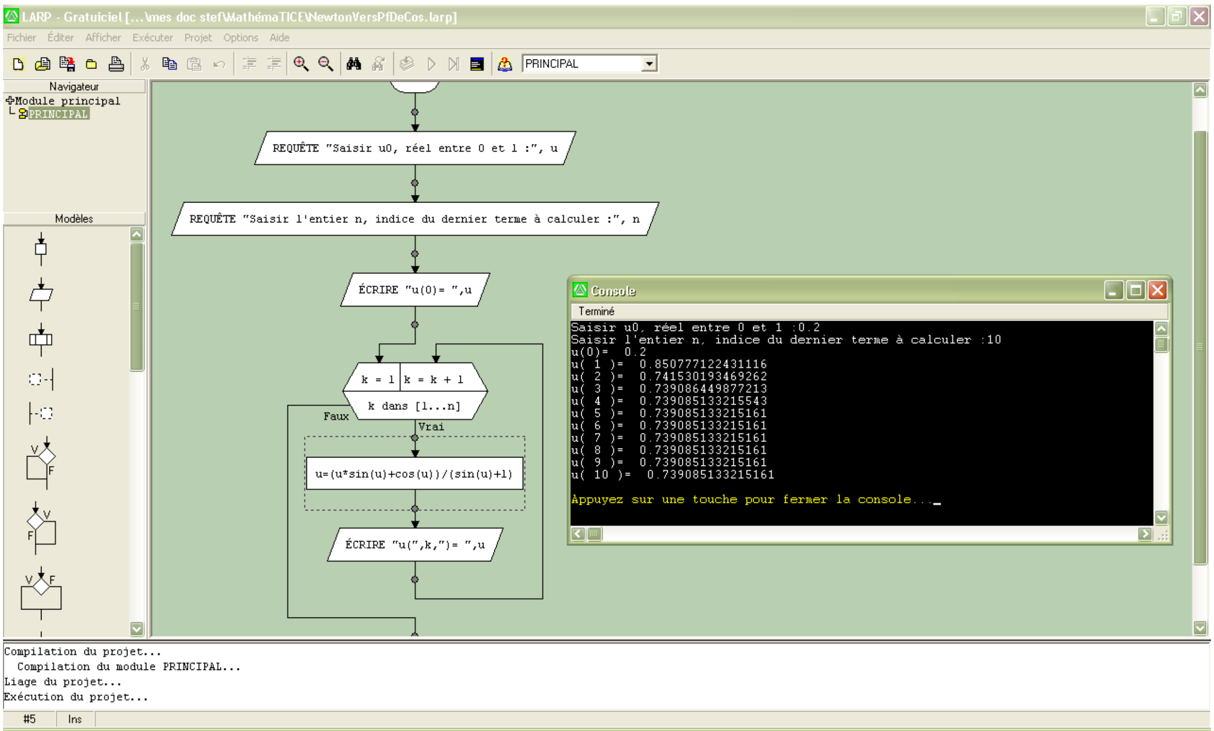
$$\begin{cases} u_0 \in [0;1] \\ u_{n+1} = \dfrac{u_n \sin u_n + \cos u_n}{\sin u_n + 1} \end{cases}$$; on peut continuer à prendre $u_0 = 0{,}2$ même si on ne part pas du coup du bon côté… car on s'y retrouve (du bon côté) dès u_1 !



Dès la 2e tangente, on est sur α !



La 2e marche de l'escalier ne peut se tracer !



Remarque : À tous les niveaux, on observe que la convergence est bien plus rapide !

B.3 Un test d'arrêt

Se pose alors la question suivante : « Comment choisir *n* pour atteindre une précision souhaitée dans l'approximation de α ? »
Le résultat suivant[3] permet de conclure dans certains cas :

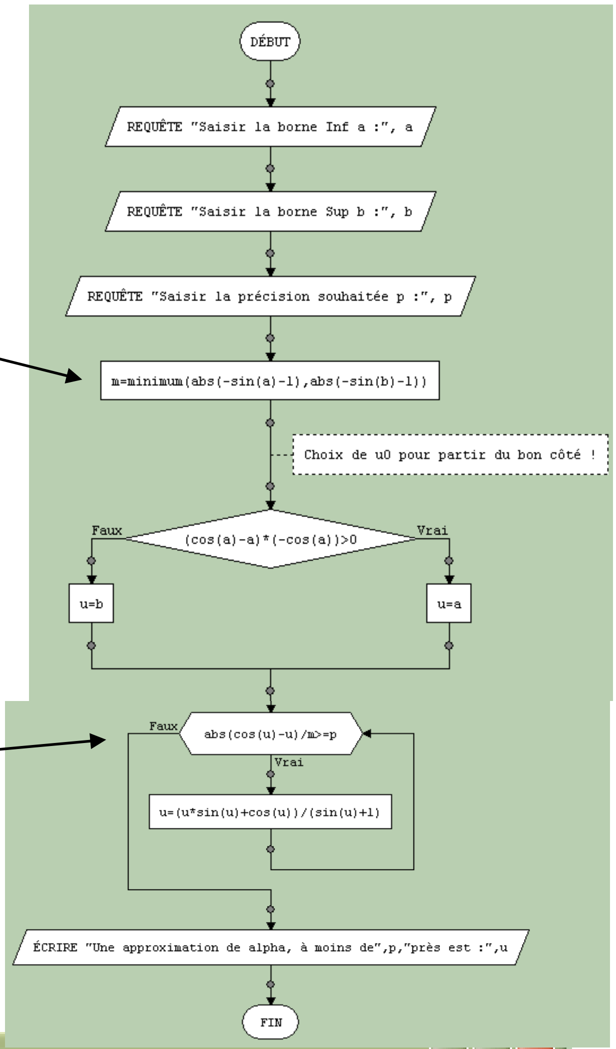
Soit α une racine de l'équation $f(x)=0$ et β une valeur approchée de α.   Si sur $[a;b]$ contenant α et β on $\lvert f'(x)\rvert \geq m > 0$, alors $\lvert \beta-\alpha\rvert \leq \frac{\lvert f(\beta)\rvert}{m}$.

Dans le cas de la méthode de Newton, on a supposé que f'' garde un signe constant sur $[a;b]$.
Du coup, $\lvert f'(x)\rvert$ est minimal soit en *a*, soit en *b* !

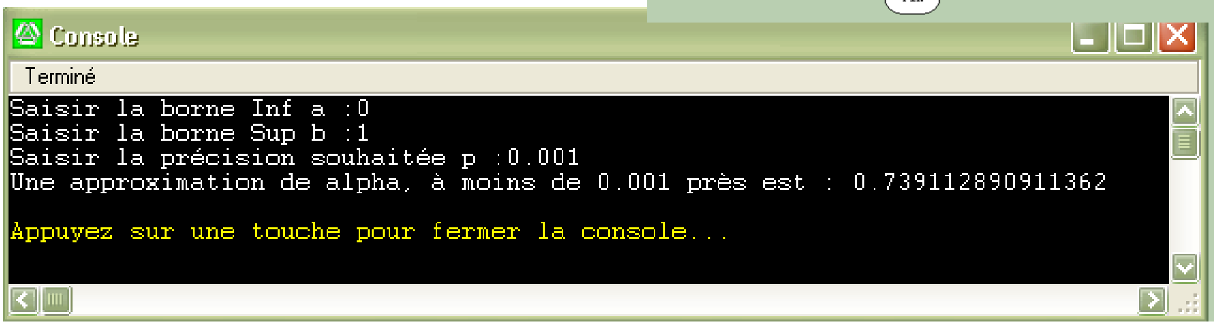
On peut donc prendre $m = Min(\lvert f'(a)\rvert, \lvert f'(b)\rvert)$;

et si p est la précision souhaitée,

on peut s'arrêter dès que $\frac{\lvert f(\beta)\rvert}{m} < p$.



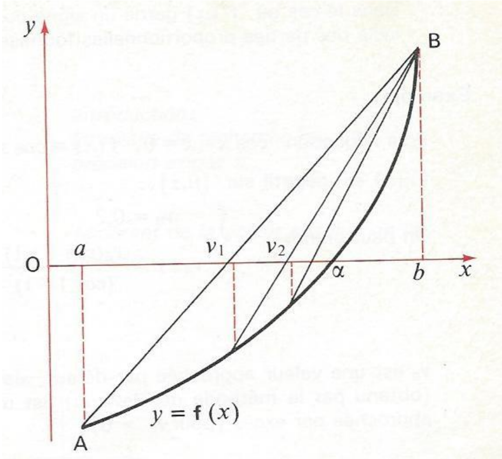
Et voici le résultat…



[3] Se démontre avec le théorème des accroissements finis

C. Méthode des parties proportionnelles[4] (Lagrange)

C.1 Description de la méthode



Sur la figure ci-contre ($f'(x)>0$, $f''(x)>0$) : $v_0=a$.

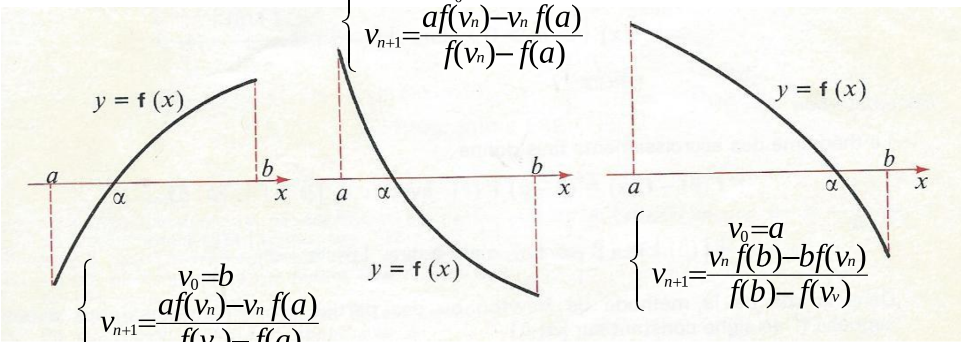
On trace la sécante (AB) qui coupe (Ox) en la valeur v_1.
Et on réitère le procédé sur l'intervalle $[v_1;b]$,
et ainsi de suite…

On crée ainsi la suite (v_n) définie par $\begin{cases} v_0=a \\ v_{n+1}=\dfrac{v_n f(b)-bf(v_n)}{f(b)-f(v_v)} \end{cases}$,

croissante, majorée, qui converge vers α.

<u>Remarque</u> : on peut définir une suite (v_n) de manière analogue pour les trois autres cas…

$\begin{cases} v_0=b \\ v_{n+1}=\dfrac{af(v_n)-v_n f(a)}{f(v_n)-f(a)} \end{cases}$



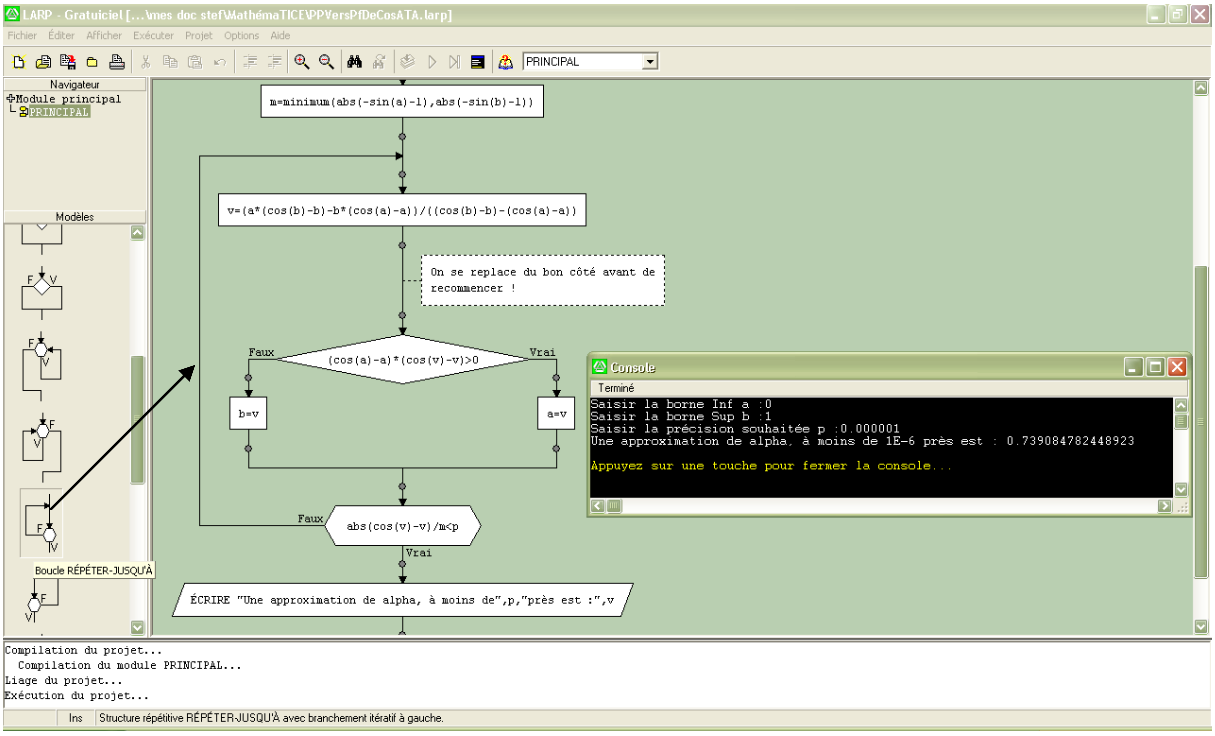
$\begin{cases} v_0=b \\ v_{n+1}=\dfrac{af(v_n)-v_n f(a)}{f(v_n)-f(a)} \end{cases}$

$\begin{cases} v_0=a \\ v_{n+1}=\dfrac{v_n f(b)-bf(v_n)}{f(b)-f(v_v)} \end{cases}$

C.2 Organigramme avec LARP : Boucle RÉPÉTER-JUSQU'À

Ici c'est une boucle RÉPÉTER-JUSQU'À qui est de mise…
Par ailleurs, après le premier calcul (celui de v_1), on prévoit de replacer du bon côté par un test conditionnel qui n'a finalement son utilité qu'au premier passage…

[4] On rencontre d'autres noms, pour la même méthode ou des méthodes proches : de *la fausse position*, de *la corde*, des *sécantes*, d'*interpolation linéaire*. Peu importe le nom, il faut y regarder de près et en comprendre le processus d'itération précis…


LARP - Gratuiciel [...\mes doc stef\MathémaTICE\PPVersPfDeCosATA.larp]
m=minimum(abs(-sin(a)-1),abs(-sin(b)-1))
v=(a*(cos(b)-b)-b*(cos(a)-a))/((cos(b)-b)-(cos(a)-a))
On se replace du bon côté avant de recommencer !
(cos(a)-a)*(cos(v)-v)>0
Faux
Vrai
b=v
a=v
abs(cos(v)-v)/m<p
ÉCRIRE "Une approximation de alpha, à moins de",p,"près est :",v
Boucle RÉPÉTER-JUSQU'À
Console
Saisir la borne Inf a :0
Saisir la borne Sup b :1
Saisir la précision souhaitée p :0.000001
Une approximation de alpha, à moins de 1E-6 près est : 0.739084782448923
Appuyez sur une touche pour fermer la console...
Compilation du projet...
Compilation du module PRINCIPAL...
Liage du projet...
Exécution du projet...
Structure répétitive RÉPÉTER-JUSQU'À avec branchement itératif à gauche.