

## Thème : Équations différentielles

Cet énoncé est celui de la deuxième épreuve orale (épreuve sur dossier) du Capes Externe de mathématiques, proposé aux candidat(e)s le 21 Juillet 2006.

Pour consulter les archives de cette épreuve orale, depuis la session 2005, on se reportera au site officiel du jury, à l'adresse <http://capes-math.org/>

### 1. L'exercice proposé au candidat

On se propose de déterminer les fonctions  $f$  définies et dérivables sur l'ensemble  $\mathbb{R}$  et vérifiant pour tout réel  $x$  l'équation  $(E) : f'(x) = f(-x)$

1. Démontrer que la fonction nulle est solution de cette équation.
2. Dans cette question et les suivantes, la fonction  $f$  est supposée non identiquement nulle.  
Après avoir prouvé que  $f$  est deux fois dérivable, trouver une équation linéaire du second ordre  $(E')$  admettant  $f$  comme solution.
3. (a) Résoudre  $(E')$ .  
(b) En déduire les fonctions  $f$  solutions de  $(E)$ .

### 2. Le travail demandé au candidat

En aucun cas, le candidat ne doit rédiger sur sa fiche sa solution de l'exercice. Celle-ci pourra néanmoins lui être demandée partiellement ou en totalité lors de l'entretien avec le jury

***Pendant sa préparation, le candidat traitera les questions suivantes :***

- Q.1) Dégager les méthodes et les théorèmes utilisés dans cet exercice.
- Q.2) En admettant que l'équation  $(E)$  possède une unique solution vérifiant  $f(0) = 1$ , proposer un algorithme qui permette d'obtenir une représentation graphique approchée de cette solution sur l'intervalle  $[-3; 3]$

***Sur ses fiches, le candidat rédigera et présentera :***

- À l'aide de la calculatrice, l'algorithme permettant d'obtenir la construction de l'approximation évoqué à la question Q. 2).
- Deux exercices sur le thème : « Équations différentielles »

## Proposition de corrigé avec le Classpad 300

### I. L'exercice proposé au candidat

Il est évident que la fonction nulle est solution de  $(E)$ .

Si  $f$  est solution de  $(E)$ , elle est dérivable sur  $\mathbb{R}$  et sa dérivée  $f'$ , qui vérifie  $f'(x) = f(-x)$  pour tout  $x$  de  $\mathbb{R}$ , est elle-même dérivable sur  $\mathbb{R}$  (car composée des applications dérivables  $x \mapsto -x$  et  $x \mapsto f(x)$ ). L'application  $f$  est donc deux fois dérivable sur  $\mathbb{R}$ .

Si on dérive  $(E)$  membre à membre, on obtient :  $\forall x \in \mathbb{R}, f''(x) = -f'(-x) = -f(x)$ .

Ainsi  $f$  est solution de l'équation différentielle  $(E')$  :  $y'' + y = 0$  sur  $\mathbb{R}$ .

La solution générale de  $(E')$  s'écrit  $\forall x \in \mathbb{R}, y(x) = a \cos x + b \sin x$  ( $a$  et  $b$  deux réels donnés).

Si  $f$  est solution de  $(E)$ , elle est donc de la forme précédente.

Réciproquement, supposons qu'il existe  $a, b$  dans  $\mathbb{R}$  tels que :  $\forall x \in \mathbb{R}, f(x) = a \cos x + b \sin x$ .

Alors pour tout  $x$  de  $\mathbb{R}$ , on a 
$$\begin{cases} f'(x) = -a \sin x + b \cos x \\ f(-x) = a \cos x - b \sin x \end{cases}$$

Si  $f$  solution de  $(E)$ , alors  $b = a$  (considérer la valeur  $x = 0$ ), la réciproque étant évidente.

Conclusion : Les solutions de  $(E)$  sont les applications  $f$  pour lesquelles il existe un réel  $a$  tel que, pour tout  $x$  de  $\mathbb{R}$ ,  $f(x) = a(\cos x + \sin x)$ .

### II. Un algorithme d'approximation d'une solution

On va s'inspirer de la méthode d'Euler.

On pose  $x_k = \frac{3k}{n}$ , avec par exemple  $n = 10$  ou  $n = 30$ . Quand l'entier  $k$  décrit  $\{-n, \dots, n\}$ , les  $x_k$  décrivent une subdivision régulière du segment  $[-3, 3]$ , de pas  $h = \frac{3}{n}$ .

On va chercher une valeur approchée  $y_k$  de  $f(x_k)$ , par récurrence finie sur  $|k|$ .

Tout d'abord on nous donne la valeur exacte  $y_0 = f(x_0) = f(0) = 1$ .

Ensuite, on suppose connues les valeurs approchées respectives  $y_{-k}$  et  $y_k$  de  $f(x_{-k})$  et  $f(x_k)$ , pour une valeur de  $k$  dans  $\{0, \dots, n-1\}$ .

On a alors  $f(x_{k+1}) = f(x_k + h) \approx f(x_k) + hf'(x_k)$ , avec  $f'(x_k) = f(-x_k) = f(x_{-k})$ .

Nous connaissons par hypothèse les valeurs approchées  $y_k$  pour  $f(x_k)$  et  $y_{-k}$  pour  $f(x_{-k})$ .

On en déduit qu'une valeur approchée pour  $f(x_{k+1})$  est  $y_{k+1} = y_k + hy_{-k}$ .

De même  $f(x_{-k-1}) = f(x_{-k} - h) \approx f(x_{-k}) - hf'(x_{-k})$  avec  $f'(x_{-k}) = f'(-x_k) = f(x_k)$ .

On en déduit qu'une valeur approchée pour  $f(x_{-k-1})$  est  $y_{-k-1} = y_{-k} - hy_k$ .

Conclusion : les égalités  $y_0 = 1$  et 
$$\begin{cases} y_{k+1} = y_k + hy_{-k} \\ y_{-k-1} = y_{-k} - hy_k \end{cases}$$
 permettent de proche en proche de

construire les points  $(x_k, y_k)$ , avec  $-n \leq k \leq n$ , le tracé du nuage de ces points donnant une idée approchée de la courbe représentative de la solution  $f$  de  $(E)$  vérifiant  $f(0) = 1$ , c'est-à-dire de l'application  $x \mapsto \cos x + \sin x$ .

On écrit un programme `eqdif` (voir fig1) calculant la liste des abscisses  $x_k$ , la liste des ordonnées  $y_k$ , et les plaçant respectivement dans les variables `list1` et `list2`.

Le programme `eqdif` prend en argument la valeur  $n$  du nombre de pas, puis il forme la liste des ordonnées  $y_k$  à partir d'une liste initiale  $\{y_0 = 1\}$  qu'il complète progressivement en  $\{y_{-1}, y_0, y_1\}$ , puis  $\{y_{-2}, y_{-1}, y_0, y_1, y_2\}$ , etc, jusqu'à la liste finale  $\{y_n, \dots, y_{-2}, y_{-1}, y_0, y_1, y_2, \dots, y_n\}$ .

On voit (fig2) un exemple d'appel du programme `eqdif`, avec l'argument  $n = 10$  (donc les abscisses sont séparées de 0.3) puis avec  $n = 30$  (les abscisses sont alors séparées de 0.1). Évidemment le temps de calcul est d'autant plus long qu'on augmente la valeur de  $n$ .

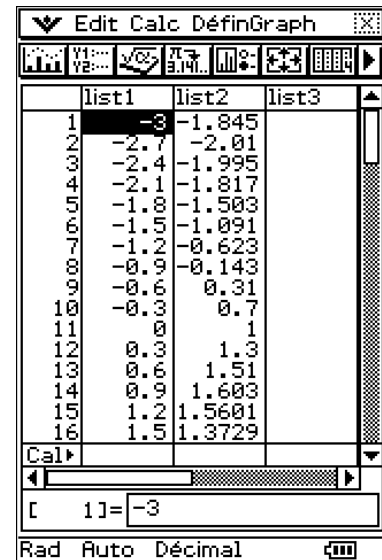
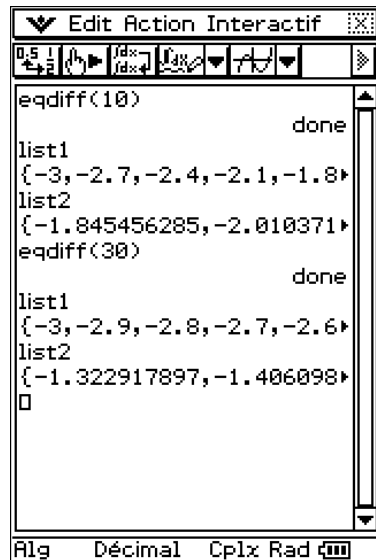
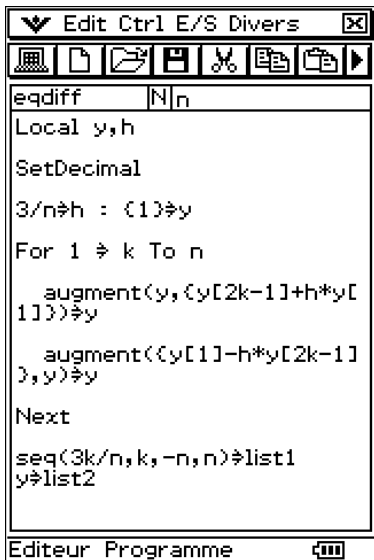


fig1 : le programme `eqdif`    fig2 : avec  $n = 10$  ou  $n = 30$     fig3 : dans l'application Statistiques

On peut alors passer dans Statistiques, et placer `list1` et `list2` dans les deux premières colonnes (cf fig3, où on a utilisé les listes résultant de l'appel de `eqdif` avec le paramètre  $n = 10$ ).

On utilise pour régler l'affichage (fig4) et pour lancer le tracé (fig5).

On utilise pour saisir la véritable solution  $x \mapsto y_1(x) = \cos x + \sin x$ , et pour la tracer en superposition du nuage de points. On se rend compte de la qualité de l'approximation (très bonne autour de 0, mais plus imprécise aux extrémités de l'intervalle).

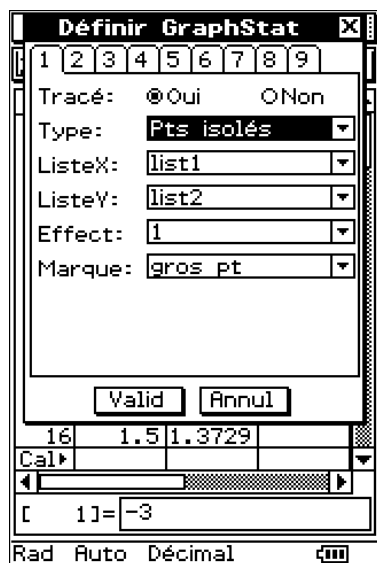


fig4 : définir le tracé

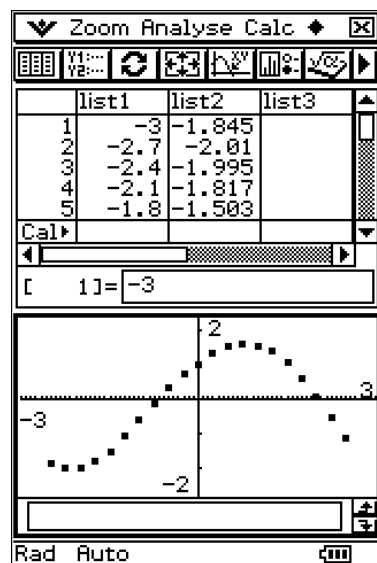


fig5 : le nuage de points

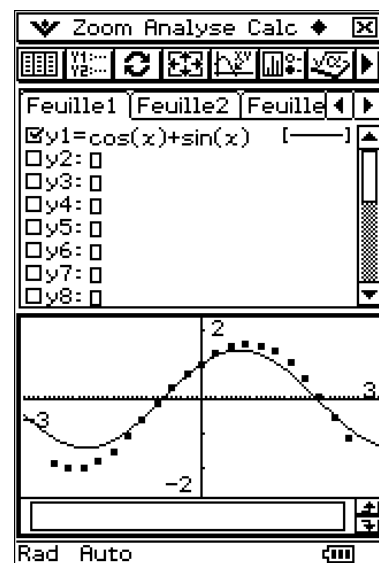


fig6 : comparer avec  $y(x)$

On recommence avec les listes obtenues avec  $n = 30$  (donc avec des abscisses séparées par 0.1 plutôt que 0.3) et on constate que l'approximation est meilleure (fig7 à fig9).

Pour la représentation conjointe du nuage de points et de la courbe  $y_1(x)$ , on a ici effectué un zoom boîte pour mieux visualiser la qualité (améliorée) de l'approximation.

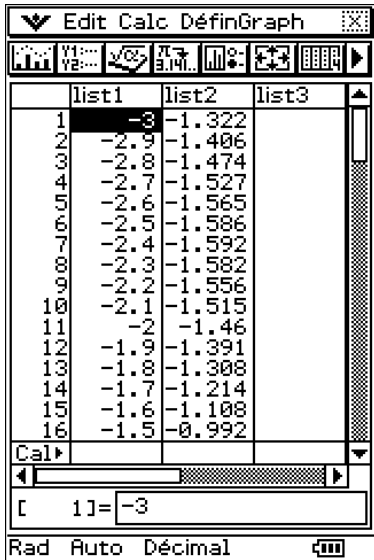


fig7 : les listes avec  $n = 30$

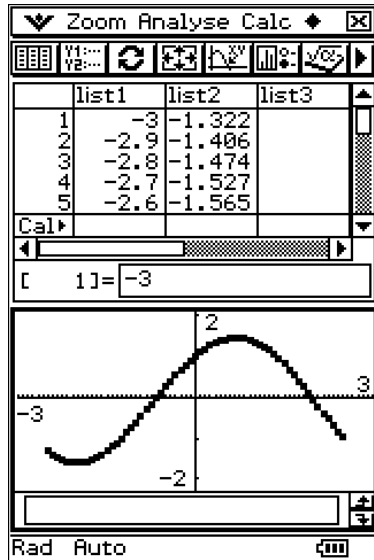


fig8 : nouveau nuage de points

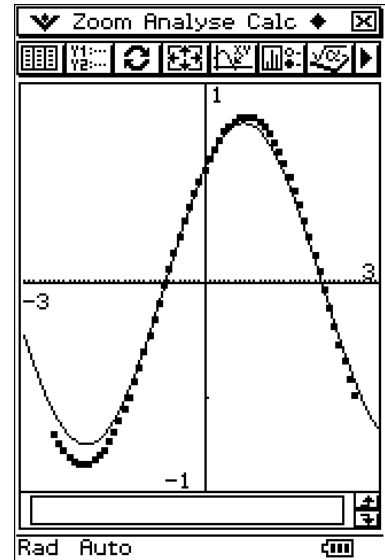


fig9 : meilleure approximation

### III. Équations différentielles d'ordre 2 avec le Classpad

L'instruction pour résoudre des équations différentielles avec le Classpad s'appelle dSolve.

Pour l'utiliser dans l'application  il est préférable d'utiliser le menu « Interactif » (fig10).

Le Classpad ouvre alors une fenêtre permettant de préciser quelle équation différentielle on souhaite résoudre. On a ici indiqué l'équation  $y'' + y = 0$  (où  $y$  est une fonction de la variable indépendante  $x$ ), sans préciser de condition initiale particulière (fig11).

On voit également (fig12) comment préciser une condition initiale (ici  $y(0) = 1$  et  $y'(0) = 2$ ).

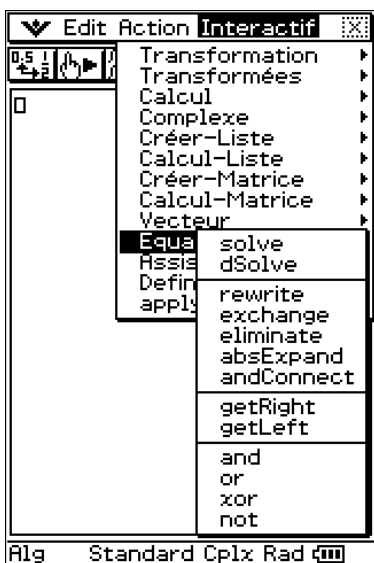


fig10 : la fonction dSolve

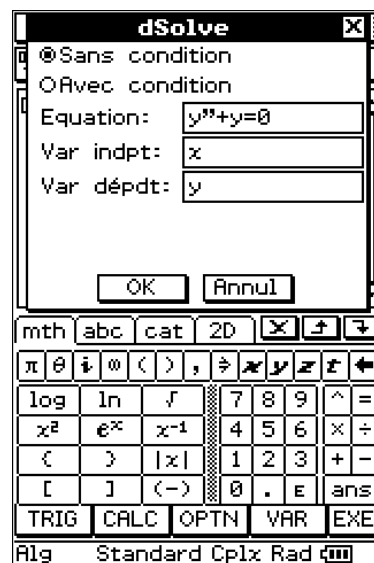


fig11 : l'équation  $y'' + y = 0$

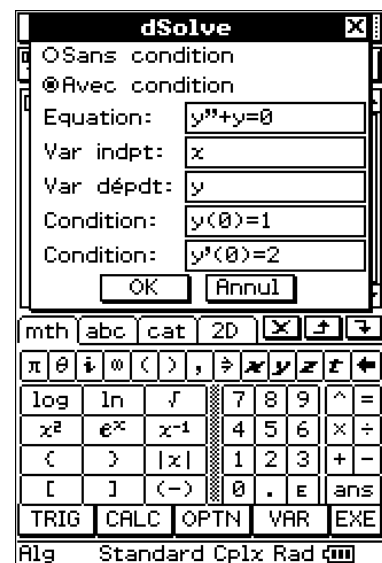

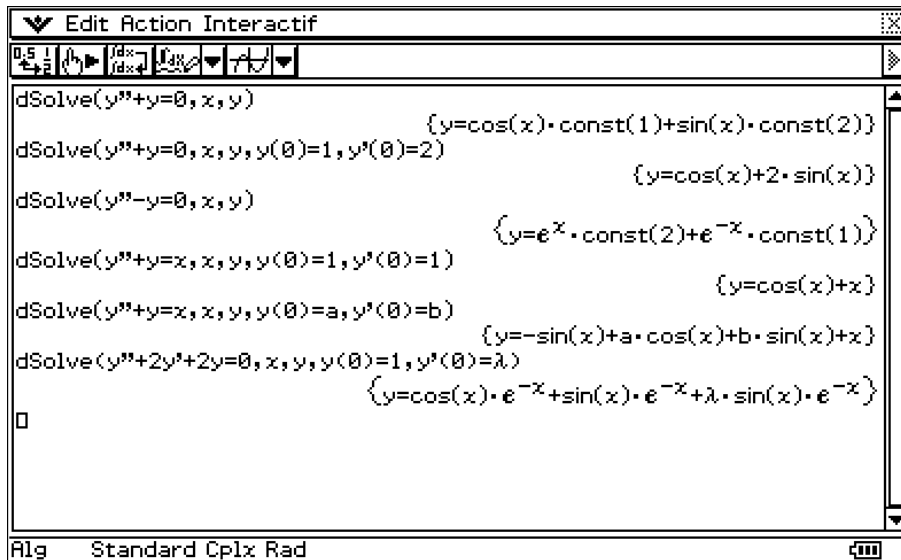


fig12 : avec condition initiale

L'expression résultant du choix effectué par l'utilisateur est alors renvoyée dans l'application  où elle est évaluée.

On voit (fig13) ce que donnent les deux appels précédents à la fonction `dSolve` (pour une meilleure lisibilité, on a affiché ici les résultats tels qu'ils apparaissent dans la fenêtre du « Classpad Manager » en mode « Resizable »).



```

Edit Action Interactif
dSolve(y''+y=0, x, y)
{y=cos(x).const(1)+sin(x).const(2)}
dSolve(y''+y=0, x, y, y(0)=1, y'(0)=2)
{y=cos(x)+2.sin(x)}
dSolve(y''-y=0, x, y)
{y=e^x.const(2)+e^-x.const(1)}
dSolve(y''+y=x, x, y, y(0)=1, y'(0)=1)
{y=cos(x)+x}
dSolve(y''+y=x, x, y, y(0)=a, y'(0)=b)
{y=-sin(x)+a.cos(x)+b.sin(x)+x}
dSolve(y''+2y'+2y=0, x, y, y(0)=1, y'(0)=lambda)
{y=cos(x).e^-x+sin(x).e^-x+lambda.sin(x).e^-x}

```

fig13 : quelques équations différentielles linéaires d'ordre 2

- ◇ L'expression `dSolve(y''+y=0, x, y)` s'évalue en  $y = \cos(x) \cdot \text{const}(1) + \sin(x) \cdot \text{const}(2)$ . Dans ce résultat `const(1)` et `const(2)` désignent bien sûr des constantes arbitraires.
- ◇ L'expression `dSolve(y''+y=0, x, y, y(0)=1, y'(0)=2)` s'évalue en  $y = \cos(x) + 2 \cdot \sin(x)$ . C'est en effet la seule solution qui satisfasse aux conditions initiales indiquées ici.


On peut bien sûr chercher à résoudre d'autres équations linéaires du second ordre (fig13) :

- ◇ `dSolve(y''-y=0, x, y)` s'évalue en  $y = e^x \cdot \text{const}(2) + e^{-x} \cdot \text{const}(1)$ .
- ◇ On voit également que la seule solution de  $y'' + y = x$  qui vérifie  $y(0) = 1$  et  $y'(0) = 1$  est donnée par  $y(x) = \cos(x) + x$ .
- ◇ Il est possible d'obtenir la solution générale d'une équation en indiquant des conditions initiales en fonction de deux paramètres formels, ce qui évite l'utilisation des noms prédéfinis `const(k)` par le Classpad.

On voit par exemple que l'expression `dSolve(y''+y=x, x, y, y(0)=a, y'(0)=b)` renvoie le résultat  $y = -\sin(x) + a \cdot \cos(x) + b \cdot \sin(x) + x$ .

- ◇ Dernier de nos exemples, on voit que les solutions de  $y'' + 2y' + 2y = 0$  qui vérifient la condition  $y(0) = 1$  et  $y'(0) = \lambda$  sont les applications  $x \mapsto y(x) = (\cos(x) + \sin(x))e^{-x} + \lambda \sin(x)e^{-x}$ .

On peut bien sûr utiliser l'application  pour étudier graphiquement les solutions d'une équation différentielle d'ordre 2.

Dans cet environnement, on choisit l'icône .

On va par exemple étudier l'équation différentielle (E) :  $y'' + 2y' + 2y = x^2$ .



On se ramène à un système différentiel d'ordre 1 en posant  $y_1 = y$  et  $y_2 = y'$ .

En effet l'équation (E) équivaut alors à  $\begin{cases} y_1' = y_2 \\ y_2' = y'' = x^2 - 2y - 2y' = x^2 - 2y_1 - 2y_2 \end{cases}$

On entre donc ces informations sous la forme indiquée (fig14).

Dans l'onglet « C Init » on peut préciser un (ou plusieurs) jeu(x) de conditions initiales.

Ici on pose  $x_i=0$ ,  $y_{1i}=1$ ,  $y_{2i}=-1$  ce qui revient à chercher la solution  $x \mapsto y(x)$  qui vérifie les conditions initiales  $y(0) = 1$  et  $y'(0) = -1$ .

Un appui sur  affiche alors le tracé de la solution spécifiée (fig15). Bien sûr il est possible de préciser les limites de la fenêtre graphique avant ou après le tracé (icône .

On peut préciser plusieurs jeux de conditions initiales et obtenir les courbes représentatives sur un même graphique. Ici, on a choisi les conditions initiales  $\begin{cases} y(0) = k \\ y'(0) = -1 \end{cases}$  pour  $-3 \leq k \leq 3$  (fig16, après agrandissement de la fenêtre de tracé).

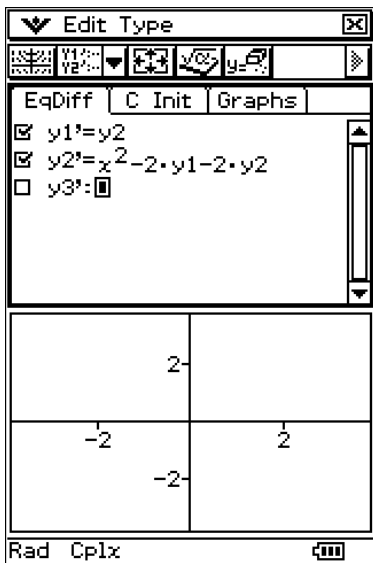


fig14 : l'équation différentielle

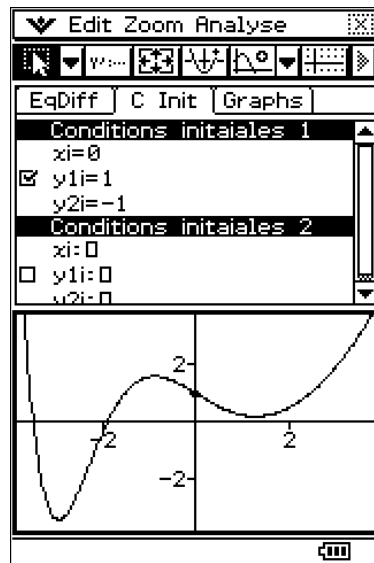


fig15 : tracé avec une CI

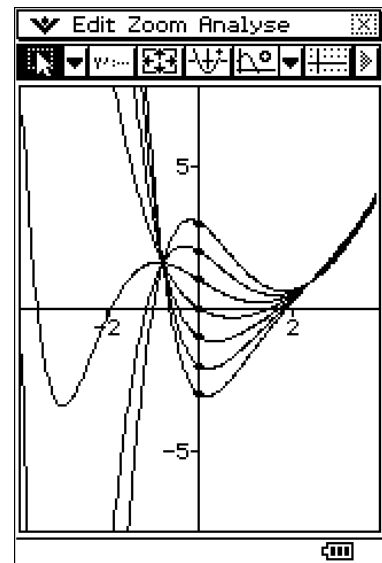


fig16 : et avec plusieurs CI

Sur l'exemple précédent, il semble bien que toutes les courbes représentatives soient asymptotes à une même courbe. Cela est confirmé par la solution formelle donnée par le Classpad.

L'asymptote commune est en effet la parabole  $x \mapsto y(x) = \frac{x^2}{2} - x + \frac{1}{2}$ .

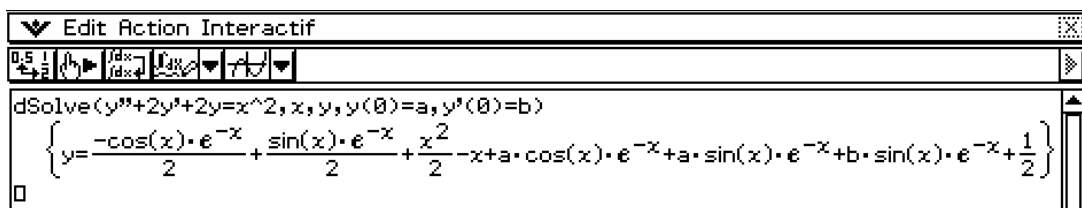


fig 17 : solution générale de l'équation différentielle  $y'' + 2y' + 2y = x^2$