Modélisation, simulation & méthodes numériques

IUT SGM

Y. Morel

https://xymaths.fr/IUT/

Qu'est-ce qu'un modèle ?

Qu'est-ce qu'un modèle ?

4 classes, au moins, de signification et d'usage :
modèle *comme*

- référent, ou prototype, à reproduire (pour un peintre par ex.)
- maquette d'un dispositif réel (pour un architecte par ex.)
- idéal (modèle de beauté, de réussite, de sainteté, ...)
- formalisme logico-physico-mathématique

Qu'est-ce qu'un modèle ?

4 classes, au moins, de signification et d'usage :
modèle *comme*

- référent, ou prototype, à reproduire (pour un peintre par ex.)
- maquette d'un dispositif réel (pour un architecte par ex.)
- idéal (modèle de beauté, de réussite, de sainteté, ...)
- formalisme logico-physico-mathématique

Modèle, donc, comme original, copie, archétype, ou réalisation simple.

Une **modélisation** d'un phénomène est sa représentation dans un système différent abstrait, souvent (pour nous) physico-mathématique.

Un modèle permet de dépasser la "simple" description du phénomène et de fournir un outil d'analyse, généralement par **réduction** de la complexité

Un modèle (scientifique) est une représentation simplifiée, et idéalisée, qui permet de mieux appréhender un phénomène, ses différents paramètres et d'en prévoir le comportement.

Une **modélisation** d'un phénomène est sa représentation dans un système différent abstrait, souvent (pour nous) physico-mathématique.

Un modèle permet de dépasser la "simple" description du phénomène et de fournir un outil d'analyse, généralement par **réduction** de la complexité

Un modèle (scientifique) est une représentation simplifiée, et idéalisée, qui permet de mieux appréhender un phénomène, ses différents paramètres et d'en prévoir le comportement.

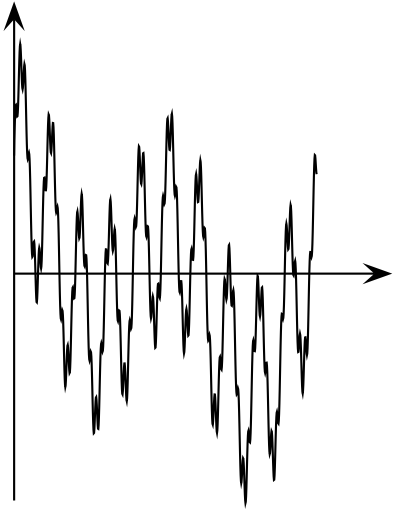
Une **simulation numérique** (ou informatique) est l'exécution d'un programme informatique, traduisant un modèle mathématique, sur un ordinateur

On traîte, et calcule, des **informations numériques** dès lors qu'on utilise un système informatisé,

- signal : son, image, vidéo, ... dans un ordinateur
- données issues d'appareils de mesure électroniques et "numériques"
- résolution approchée de problèmes mathématique / physique

La qualification de **numérique** s'oppose à celles de formelle, littérale, algébrique en mathématiques, et à **analogique** en sciences expérimentales.

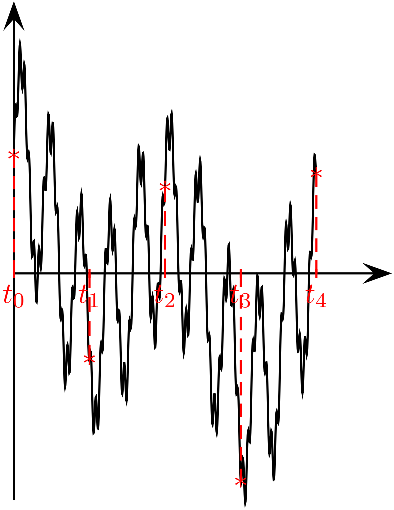
L'esprit humain, et les systèmes analogiques, peuvent traîter **l'infini**, pas les systèmes numériques.



En théorie : 2 infinis

- en "temps", $t \in [0; T]$
- valeurs du signal : $f(t) \in \mathbb{R}$

Numérisation d'un signal :

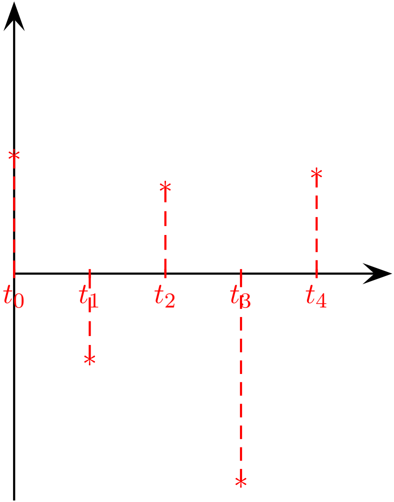


En théorie : 2 infinis

- en "temps", $t \in [0;T]$
- valeurs du signal : $f(t) \in \mathbb{R}$

Numérisation d'un signal :

- échantillonnage (temporel)

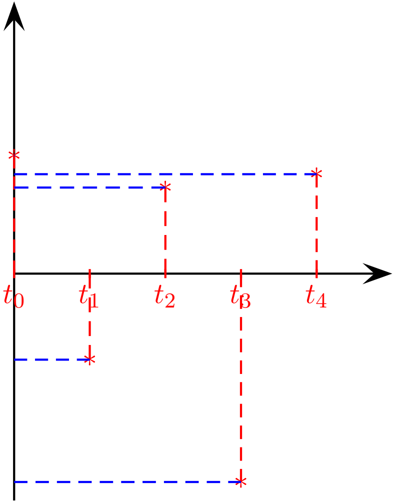


En théorie : 2 infinis

- en "temps", $t \in [0; T]$
- valeurs du signal : $f(t) \in \mathbb{R}$

Numérisation d'un signal :

- échantillonnage (temporel)



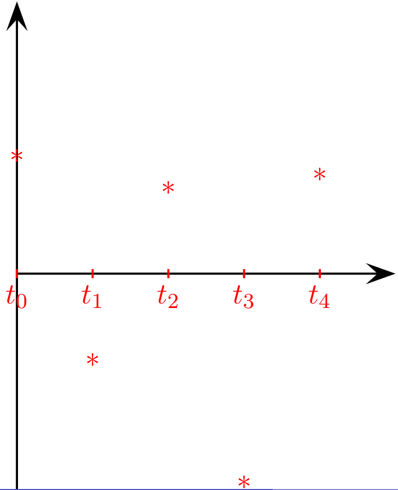
En théorie : 2 infinis

- en "temps", $t \in [0; T]$
- valeurs du signal : $f(t) \in \mathbb{R}$

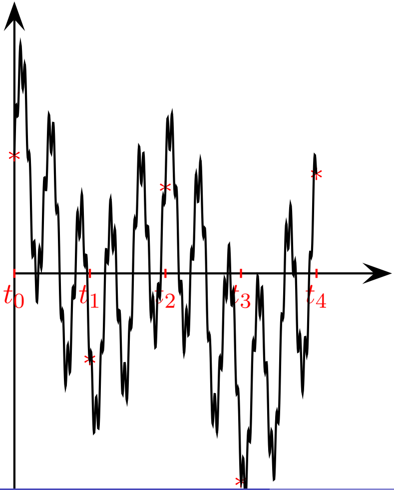
Numérisation d'un signal :

- échantillonnage (temporel)
- quantification des valeurs du signal

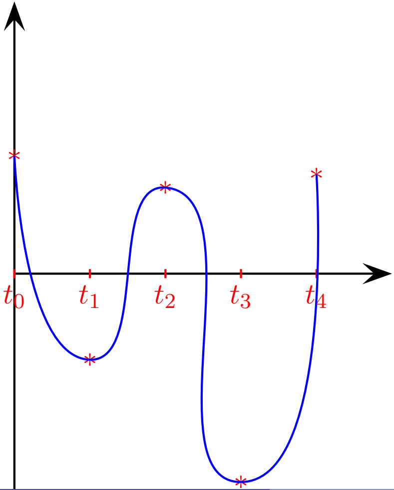
Problème inverse : reconstruction du signal analogique d'origine ?



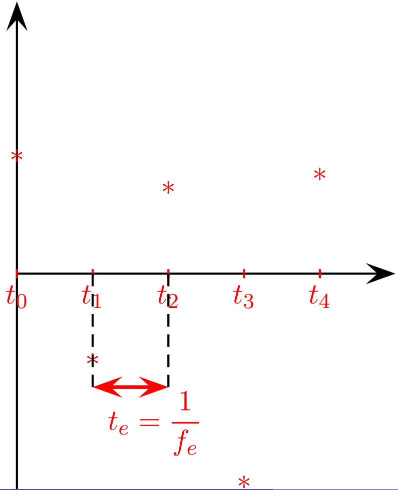
Problème inverse : reconstruction du signal analogique d'origine ?



Problème inverse : reconstruction du signal analogique d'origine ?



Problème inverse : reconstruction du signal analogique d'origine ?



Le problème est résolu par Schannon dans son célèbre théorème :

Les échantillons $s_1, s_2, s_3, \ldots$ *aux instants* $t_i = it_e$ *interpolent un unique signal* s *de fréquence maximale* f_m *dès lors que*

$$f_e > 2f_m$$

Chaque personne infectée transmet, en moyenne, le virus à p autres personnes.

C'est un modèle géométrique : le nombre de personnes infectées suit une progression géométrique de raison p.
Si le 1$^{\text{er}}$ jour une (seule) personne est infectée, alors p^n personnes le sont à leur tour au bout de n jours.

- si $p < 1$, le nombre de malades tend vers 0
- si $p > 1$, le virus se répand

En désignant par $S(n)$ le nombre de personnes infectées le n-ième jour, on a la relation géométrique

$$S(n+1) = pS(n)$$

Pour $p = 2$

- au bout de 10 jours, il y donc $2^{10} = 1024$ malades,
- puis $2^{20} \simeq 10^6$ au bout de 20 jours,
- puis $2^{30} \simeq 10^9$ au bout de 30 jours...

Pour $p = 3$

- au bout de 10 jours, il y donc $3^{10} \simeq 60\,000$ malades,
- puis $3^{20} \simeq 3,5.10^9$ au bout de 20 jours,
- puis $3^{30} \simeq 2.10^{14}$ au bout de 30 jours ! !

Le modèle précédent est (très, trop) simple :

- permet d'appréhender le phénomène (très difficile à imaginer, naïvement, sans calcul).
- ne prend pas en compte de nombreux phénomènes : guérisons, morts, réinfection, dynamique des populations, vaccin, ...

On peut améliorer le modèle précédent en différenciant 3 catégories de personnes

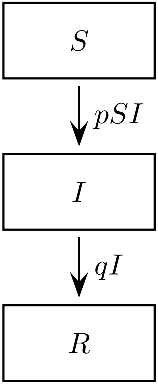
- Susceptible
- Infected
- Recovered

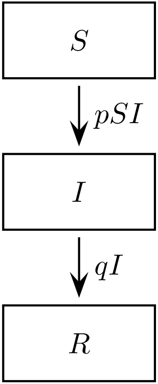
Le modèle précédent est (très, trop) simple :

- permet d'appréhender le phénomène (très difficile à imaginer, naïvement, sans calcul).
- ne prend pas en compte de nombreux phénomènes : guérisons, morts, réinfection, dynamique des populations, vaccin, ...

On peut améliorer le modèle précédent en différenciant 3 catégories de personnes

- **S**usceptible
- **I**nfected
- **R**ecovered


S
pSI
I
qI
R

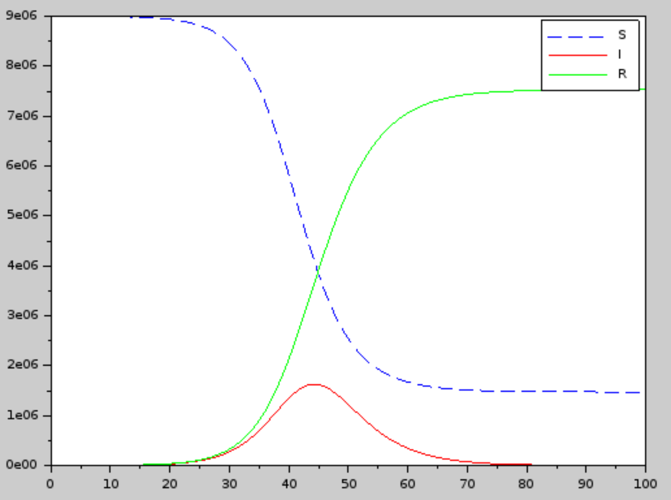


$$S(n+1) = S(n) - pS(n)I(n)$$

$$I(n+1) = I(n) + pS(n)I(n) - qI$$

$$R(n+1) = R(n) + qI$$

Simulation numérique : résultats



Vers un modèle plus fin ?

Les paramètres p et q sont à chercher / adapter. Le sociologue Hugues Lagrange propose[1] le coefficient

$$p' = p\Big(1 - D(n)\Big)^k$$

où $D(n)$ est le nombre de morts le n-ième jour et k un paramètre à ajuster.

Ce modèle incorpore ici une dimension psychologique : les personnes susceptibles d'être contaminées adaptent leur comportement en fonction du contexte, en particulier face au danger qui s'évalue en fonction du nombre de morts.

[1] Hugues Lagrange, *Coronavirus et comportement individuel*

Les modèles précédents décrivent le même phénomène, de plus en plus finement, de plus en plus justement ?

"Essentially, all models are wrong, but some are useful"

George E.P. Box

Quels sont les objectifs d'une modélisation ?

Les modèles précédents décrivent le même phénomène, de plus en plus finement, de plus en plus justement ?

"Essentially, all models are wrong, but some are useful"

George E.P. Box

Quels sont les objectifs d'une modélisation ?

- comprendre
- prédire
- agir

Les modèles précédents décrivent le même phénomène, de plus en plus finement, de plus en plus justement ?

"Essentially, all models are wrong, but some are useful"

George E.P. Box

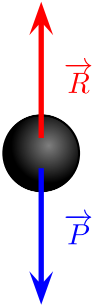
Quels sont les objectifs d'une modélisation ?

- comprendre
- prédire
- agir

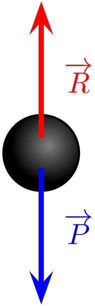
et plus généralement (cognitivement) :

- parce qu'on n'a pas le choix !

  nous essayons, avec notre cerveau et nos compétences limités de comprendre un monde et une réalité (très, trop) complexe.
  Il faut simplifier, élaguer dans les trop nombreux paramètres, ...



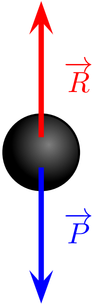
- Modèle physique : $\begin{cases} m\overrightarrow{a} = \overrightarrow{P} - \overrightarrow{R} \\ P = \|\overrightarrow{P}\| = mg \\ R = \|\overrightarrow{R}\| = kv \end{cases}$



- Modèle physique : $\left\{\begin{array}{l} m\overrightarrow{a} = \overrightarrow{P} - \overrightarrow{R} \\ P = \|\overrightarrow{P}\| = mg \\ R = \|\overrightarrow{R}\| = kv \end{array}\right.$

- Résolution mathématique : en projetant, avec $\overrightarrow{a} = \overrightarrow{v'}$,

$$mv' = mg - kv \iff v' + \frac{k}{m}v = g$$
$$\iff v(t) = \frac{mg}{k} + \left(v_0 - \frac{mg}{k}\right) e^{-\frac{k}{m}t}$$

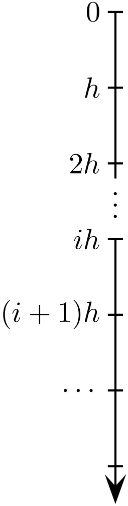


- Modèle physique 2 : $\begin{cases} m\overrightarrow{a} = \overrightarrow{P} - \overrightarrow{R} \\ P = \|\overrightarrow{P}\| = mg \\ R = \|\overrightarrow{R}\| = kv^2 \end{cases}$
- Résolution mathématique

$$v' + \frac{k}{m}v^2 = g$$

  - Équation non linéaire
  - résolution mathématique exacte ? ?

On cherche à déterminer une solution approchée au problème.



- On discrétise (maillage), ici en 1D, et on recherche alors une approximation de la solution aux nœuds $t_i = ih$:

$$v_i := v(ih)$$

- On approxime "l'équation".
  Plus précisément on approche la dérivée avec les formules de Taylor :

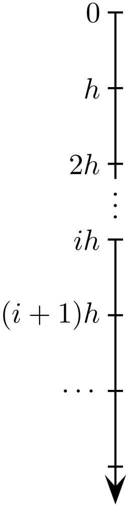
$$v\left(t + dt\right) = v(t) + dt\, v'(t) + \frac{dt^2}{2} v''(t) + \dots$$

soit, avec $t = t_i = ih$ et $dt = h$,

$$v_{i+1} = v_i + hv_i' + O\left(h^2\right)$$

$$\iff \quad v_i' \simeq \frac{v_{i+1} - v_i}{h}$$

On cherche à déterminer une solution approchée au problème.



- On discrétise (maillage), ici en 1D, et on recherche alors une approximation de la solution aux nœuds $t_i = ih$:

$$v_i := v(ih)$$

- On approxime "l'équation".
  On aboutit alors à :

$$v' + \frac{k}{m}v^2 = g$$

$$\Longrightarrow \quad \frac{v_{i+1} - v_i}{h} + \frac{k}{m}v_i^2 = g$$

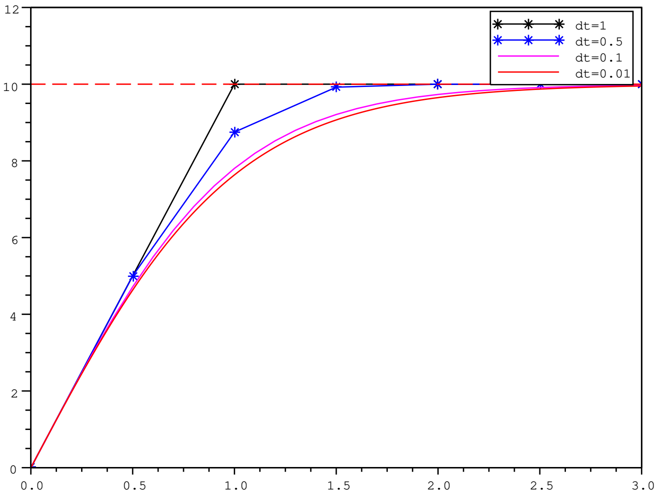
$$\Longleftrightarrow \quad v_{i+1} = v_i + h\left(g - \frac{k}{m}v_i^2\right)$$

On implémente (ou programme) alors notre relation discrète

$$v_{i+1} = v_i + h\left(g - \frac{k}{m}v_i^2\right)$$

En pseudo-code (pseudo-Scilab) :

Initialisation : k, m, g, N
v(0)=0
Pour i de 0 à N :
$$v(i+1) = v(i) + h\left(g - \frac{k}{m}v(i)^2\right)$$
Fin
plot(v), *//ou tout autre post-traitement...*


dt=1
dt=0.5
dt=0.1
dt=0.01
0
2
4
6
8
10
12
0.0
0.5
1.0
1.5
2.0
2.5
3.0

- Exactitude de la solution approchée trouvée ?
- Précision ?
- Stabilité ? ou domaine de stabilité ?
- Précision vs. temps de calcul ?

- Exactitude de la solution approchée trouvée ?
  - *Analyse mathématique & numérique : a-t'on convergence vers la solution φ du problème :*

$$\lim_{h \to 0} \max_{0 \leqslant i \leqslant N} |v(t_i) - \varphi(t_i)| = 0 \ ?$$

  - *Question informatique fondamentale : comment savoir qu'un programme informatique est "juste" ? (s'exécute, garanti sans bug et retourne un bon résultat)*

- Précision ?
- Stabilité ? ou domaine de stabilité ?
- Précision vs. temps de calcul ?

- Exactitude de la solution approchée trouvée ?
  - *Analyse mathématique & numérique : a-t'on convergence vers la solution φ du problème :*

$$\lim_{h \to 0} \max_{0 \leqslant i \leqslant N} \left| v\left(t_i\right) - \varphi\left(t_i\right) \right| = 0 \ ?$$

  - *Question informatique fondamentale : comment savoir qu'un programme informatique est "juste" ? (s'exécute, garanti sans bug et retourne un bon résultat)*
    $\rightarrow$ *cas connus théoriquement*
    $\rightarrow$ *cas d'inter-comparaison*
    $\rightarrow$ *comparaison avec résultats expérimentaux*
- Précision ?
- Stabilité ? ou domaine de stabilité ?
- Précision vs. temps de calcul ?

- Exactitude de la solution approchée trouvée ?
- Précision ?

  Estimation a priori de l'erreur entre la vraie solution et la solution approchée calculée

- Stabilité ? ou domaine de stabilité ?
- Précision vs. temps de calcul ?

- Exactitude de la solution approchée trouvée ?
- Précision ?
- Stabilité ? ou domaine de stabilité ?

Et si les valeurs des paramètres, k, m, n, ou v_0 varient un peu ?

ou, issues de données expérimentales, ne sont pas très précises ?

ou encore, n'ont, de toute façon, qu'une précision limitée à la précision machine. . .

$\hookrightarrow$ La stabilité d'une méthode numérique est sa capacité a ne pas amplifier les erreurs

- Précision vs. temps de calcul ?

- Exactitude de la solution approchée trouvée ?
- Précision ?
- Stabilité ? ou domaine de stabilité ?
- Précision vs. temps de calcul ?

  Évaluation de la complexité algorithmique
  - *Estimation du temps de calcul nécessaire*
  - *Précision vs. temps de calcul ?*

    $\hookrightarrow$ *Combien "coûte" un résultat 2 fois plus précis ?*

On note φ la solution exacte de $\varphi' + K\varphi^2 = g$ (avec $K = \dfrac{k}{m}$)

et (v_n) la suite définie par $v_{i+1} = v_i + hg - hKv_i^2$

On pose $\varphi_i = \varphi(ih)$ et on cherche à estimer l'erreur commise $e_i = |v_i - \varphi_i|$.

Tout repose ici sur l'approximation de la dérivée (Taylor) :

$$\varphi_{i+1} = \varphi_i + h\varphi_i' + O\left(h^2\right)$$

et donc, dans l'équation différentielle

$$\varphi_{i+1} = \varphi_i + hg - hK\varphi_i^2 + O\left(h^2\right)$$

On note φ la solution exacte de $\varphi' + K\varphi^2 = g$ (avec $K = \dfrac{k}{m}$)

et (v_n) la suite définie par $\boxed{v_{i+1} = v_i + hg - hKv_i^2}$

On pose $\varphi_i = \varphi(ih)$ et on cherche à estimer l'erreur commise $e_i = |v_i - \varphi_i|$.

Tout repose ici sur l'approximation de la dérivée (Taylor) :

$$\varphi_{i+1} = \varphi_i + h\varphi_i' + O\left(h^2\right)$$

et donc, dans l'équation différentielle

$$\boxed{\varphi_{i+1} = \varphi_i + hg - hK\varphi_i^2 + O\left(h^2\right)}$$

et ainsi,

$$\begin{aligned} \Longrightarrow e_{i+1} = |v_{i+1} - \varphi_{i+1}| &\leqslant |v_i - \varphi_i| + hK\left|v_i^2 - \varphi_i^2\right| + O(h^2) \\ &\leqslant (1 + 2MhK)\, e_i + O\left(h^2\right) \end{aligned}$$

Par récurrence, on arrive alors à : $e_i \leqslant (1+2MhK)^i e_0 + iO\left(h^2\right)$

et, à l'extrémité de l'intervalle $[0;T]$: $e_N \leqslant (1+2MhK)^N e_0 + NO\left(h^2\right)$

or $T = Nh \iff N = O\left(\dfrac{1}{h}\right)$, d'où

$$e_N \leqslant (1+2MhK)^N e_0 + O(h)$$

avec $e_0 = |v_0 - \varphi_0|$.

Par récurrence, on arrive alors à : $e_i \leqslant (1+2MhK)^i e_0 + iO\left(h^2\right)$

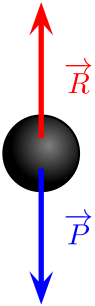
et, à l'extrémité de l'intervalle $[0;T]$: $e_N \leqslant (1+2MhK)^N e_0 + NO\left(h^2\right)$

or $T = Nh \iff N = O\left(\dfrac{1}{h}\right)$, d'où

$$e_N \leqslant (1+2MhK)^N e_0 + O(h)$$

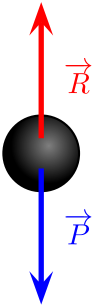
avec $e_0 = |v_0 - \varphi_0|$.
Ainsi, on peut répondre à quelques questions :

- si $e_0 = 0$, pour tout i, $\lim\limits_{h\to 0} e_i = 0$ et les valeurs approchées calculées v_i convergent vers les valeurs exactes $\varphi_i = \varphi(t_i)$
- la précision est de l'ordre de h
- le schéma numérique est stable, si $e_0 \neq 0$, le facteur $(1+2MhK)^N$ est borné (par e^{2MKT}), et les erreurs ne s'amplifient pas
- si on divise par exemple par 2 le pas h, donc mutliplie par 2 le nombre de calculs, la précision est aussi divisée par 2
  (et par 4 pour les termes en $O\left(h^2\right)$)



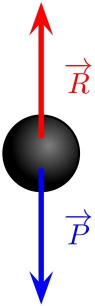
- Modèle physique : $\begin{cases} m\overrightarrow{a} = \overrightarrow{P} - \overrightarrow{R} \\ P = \|\overrightarrow{P}\| = mg \\ R = \|\overrightarrow{R}\| = kv \end{cases}$

- Valeur du coefficient k ?
  Valeur issue de l'expérimentation : forme du corps & viscosité du fluide
  $\hookrightarrow$ À partir de résultats expérimentaux, on détermine une valeur de k

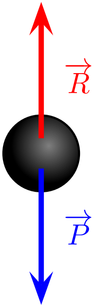


- Modèle physique : $\begin{cases} m\overrightarrow{a} = \overrightarrow{P} - \overrightarrow{R} \\ P = \|\overrightarrow{P}\| = mg \\ R = \|\overrightarrow{R}\| = kv \end{cases}$

- Valeur du coefficient k ?
  Valeur issue de l'expérimentation : forme du corps & viscosité du fluide
  $\hookrightarrow$ À partir de résultat**s** expériment**aux**, on détermine **une** valeur de k



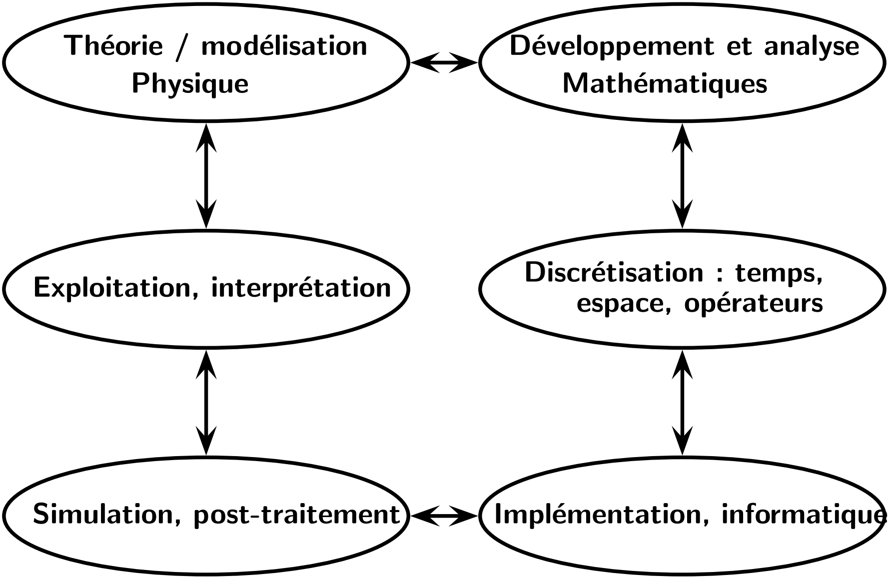
- Modèle physique 2 : $\begin{cases} m\overrightarrow{a} = \overrightarrow{P} - \overrightarrow{R} \\ P = \|\overrightarrow{P}\| = mg \\ R = \|\overrightarrow{R}\| = kv^2 \end{cases}$
- Valeur du coefficient k ?
  Valeur issue de l'expérimentation : forme du corps & viscosité du fluide
  $\hookrightarrow$ À partir de résultats expérimentaux, on détermine une valeur de k



- Modèle physique $\color{red}{3}$: $\begin{cases} m\overrightarrow{a} = \overrightarrow{P} - \overrightarrow{R} \\ P = \|\overrightarrow{P}\| = mg \\ R = \|\overrightarrow{R}\| = \color{red}{k_1 v + k_2 v^2 + \dots} \end{cases}$

$\hookrightarrow$ détermination de n coefficients k_1, k_2, …, k_n
à partir de $N \geqslant n$ mesures.

- Résolution d'un système surdéterminé ?
- "Solution" $(k_1; k_2; \dots)$ la "meilleure" ?


Théorie / modélisation
Physique
Développement et analyse
Mathématiques
Exploitation, interprétation
Discrétisation : temps,
espace, opérateurs
Simulation, post-traitement
Implémentation, informatique

La **modélisation** d'un système est la recherche / utilisation de lois afin de décrire son état, son évolution dans l'espace et le temps.

Par exemple, la loi de Newton (PFD), les équations de Maxwell, de Navier-Stockes ...

L'**analyse mathématique** d'un modèle est nécessaire pour

- s'assurer que le problème est bien posé mathématiquement, c'est-à-dire
  - qu'il admet une solution, et une seule,
  - que cette solution est "stable" par rapport aux paramètres et données
- fournir un cadre mathématique adapté : espace de fonctions, signification des opérateurs, ...
- éventuellement simplifier et/ou reformuler le problème

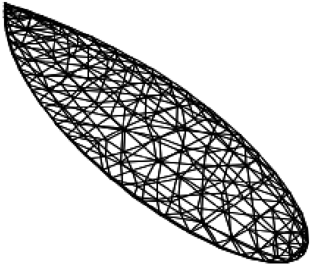
La plupart des équations modélisant des systèmes physiques réels n'admettent pas de solution analytique explicite.

On se lance alors dans une résolution numérique approchée :

- discrétisation, temps et espace (maillage)
- approximation des équations

  $\hookrightarrow$ construction d'un schéma numérique
- étude du schéma

  $\hookrightarrow$ stabilité, convergence, performance



Toute cette étape est le champ disciplinaire de l'**analyse numérique**.

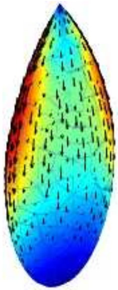
Une fois le schéma numérique écrit et validé, il reste à s'en servir, en l'**implémentant** sur machine.

Il existe un grand nombre de langages de programmation : python, c, c++, fortran, Matlab/Scilab, java, php, javascript, ...

Si un des objectifs de l'analyse numérique est de fournir des méthodes et schémas numériques efficaces, des dévéloppements purement informatiques partagent aussi ce même objectif :

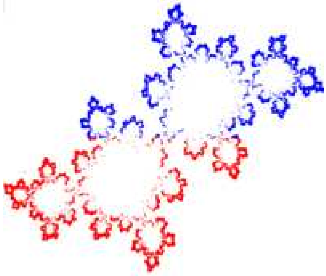
- logiciels *(évolution des langages de programmation)*
- matériels *(parallélisme massif, processeurs multi-cœurs, ...)*
- ordinateurs quantiques ...

Tout ce qui précède permet de faire des simulations, c'est-à-dire d'obtenir des résultats d'expérience.



En retour, ces simulations peuvent permettre de découvrir certains phénomènes, numériques certes, mais aussi mathématiques, physiques, ...
par l'observation par exemple d'un phénomène non prévu (physique et/ou mathématique et/ou numérique)

Par exemple, la **théorie du chaos** est née de phénomènes numériques surprenants *(découverts par Lorenz dans les années 60, antithèse d'une certaine façon de la stabilité numérique)*
puis les fractales, IFS, attracteur, ...



- se familiariser avec la chaîne de résolution d'un problème physique
- savoir implémenter quelques méthodes numériques de résolution de problèmes simples
  - discrétiser les équations algébriques et différentielles
  - calculer numériquement des dérivées et intégrales
  - Interpoler / extrapoler des données
  - programmer (Scilab, Excel), et simuler

I - Éléments de calcul numérique

- a) Erreurs de calcul numérique courantes
  Notion de conditionnement
- b) Méthodes itératives et propagation des erreurs

II - Résolution des équations $f(x) = 0$

- a) Principes généraux, TVI, points fixes
- b) Méthodes par dichotomie, de la sécante, de Newton

III - Approximation des dérivées

- a) Formule de Taylor
- b) Approximation des dérivée 1ère et 2nde
- c) Précision des approximations

IV - Calcul d'intégrales

- a) Principe général
- b) Méthodes des rectangles, des trapèzes
- c) Précision des méthodes

IV - Résolution d'équations différentielles

a) Principe général - Discrétisation
b) Méthode d'Euler
c) Méthodes de Runge-Kutta

V - Interpolation, extrapolation de données - Moindres carrés

a) Principes généraux
b) Méthode des moindres carrés

VI - Résolution de système linéaires

Méthodes de Gauss, Gauss-Jordan et Cholesky

VII - Introduction à la méthodes des éléments finis

a) Formulation variationnelle ou intégrale
b) Méthode de Galerkin et éléments finis
c) Maillage et éléments finis 2D, 3D
d) Problèmes d'évolution