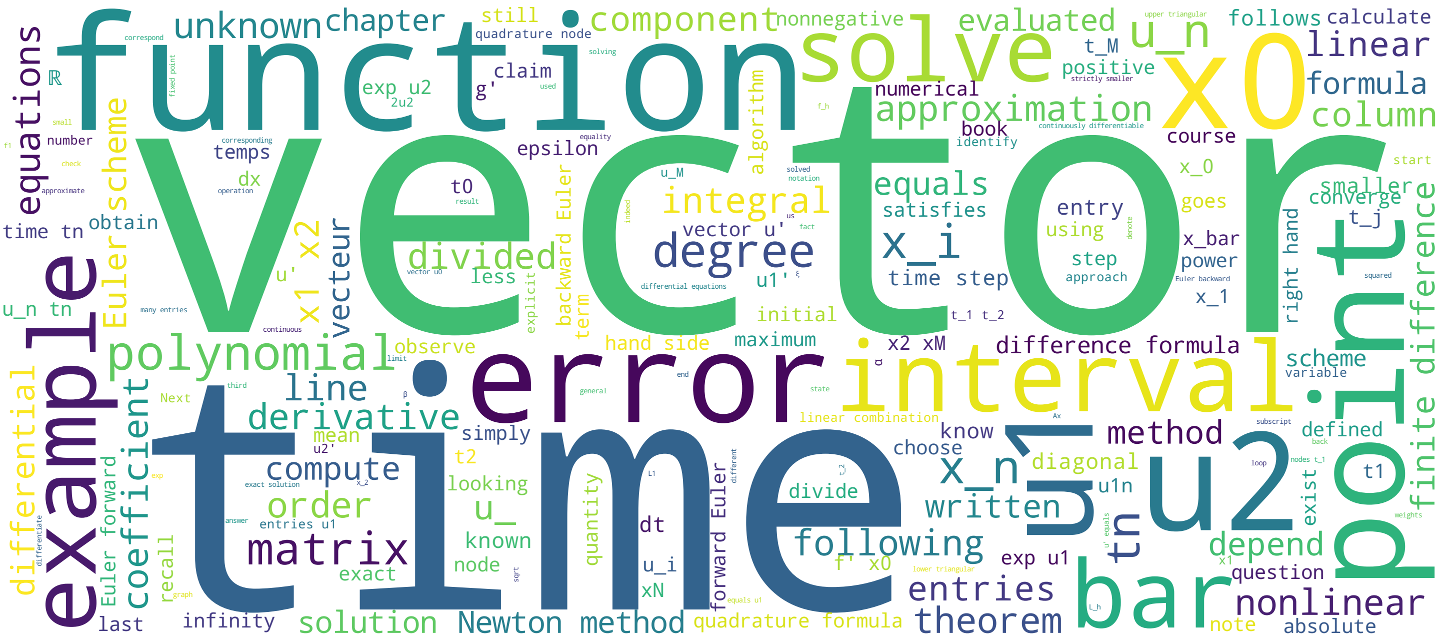
Chapitre 9 : Systèmes d'équations différentielles du 1er ordre

Introduction à l'analyse numérique

Prof. Marco Picasso



Video

EPFL

Chap 9. Systèmes d'eq. différentielles du 1^{er} ordre.

Cherche $\vec{u}(t) = \begin{pmatrix} u_1(t) \\ u_2(t) \\ \vdots \\ u_n(t) \end{pmatrix}$ tq $\begin{cases} \dot{\vec{u}}(t) = \vec{f}(\vec{u}(t), t) & t>0 \\ \vec{u}(0) = \vec{u}_0 \end{cases}$ où $\vec{f} : \mathbb{R}^n \times \mathbb{R}_+ \to \mathbb{R}^n$, $(\vec{x}, t) \to \vec{f}(\vec{x}, t)$

In the last part of the course, we'll solve first order systems of differential equations. So, the problem I want to solve is as follows. I am looking for some vector u(t), the vector u(t) is the vector with entries u1(t), u2(t), until uM(t), which satisfies the differential equation qui s'écrit toujours : u'(t), but this time with a vector on the u' equals to f(u(t),t), with a vector on f, et de *t*, pour *t* positif, with an initial value which is the vector u evaluated at the time 0 equal to some given vector u0. So each of the components of this vector u0 is given. The notations are as follows: here is the function f, a vector function, so here f is a given function. so it maps a vector x, and a time t to f(vector)(x (vector), t). so x is a vector with M components, autant de composantes que le vecteur *u*. x is in ℝ^M, t is nonnegative, in ℝ+, then, f(x,t), f with as many entries as u, so here, u'(t), point vector, is simply the vector u that we differentiate at each entry, so u1'(t), u2'(t), ... uM'(t). So f has as many entries as the vector u, or the vector u'.

Notes

Summary

0m 03s

Chap 9. Systèmes d'éq. différentielles du 1er ordre.

Cherche $\vec{u}(t) = \begin{pmatrix} u_1(t) \\ u_2(t) \\ \vdots \\ u_n(t) \end{pmatrix}$ tq $\begin{cases} \dot{\vec{u}}(t) = \vec{f}(\vec{u}(t), t) & t>0 \\ \vec{u}(0) = \vec{u}_0 \end{cases}$ où $\vec{f} : \mathbb{R}^M \times \mathbb{R}_+ \to \mathbb{R}^n$, $(\vec{x}, t) \to \vec{f}(\vec{x}, t)$

$$\vec{f}(\vec{x}, t) = \begin{pmatrix} f_1(x_1, x_2 \cdots x_n, t) \\ f_2(x_1, x_2 \cdots x_M, t) \\ \vdots \\ f_n(x_1, x_2, \dots x_n, t) \end{pmatrix}$$

Ex : $M = 2$ $\quad \vec{u}(t) = \begin{pmatrix} u_1(t) \\ u_2(t) \end{pmatrix}$ tq $\begin{cases} \dot{u}_1(t) = -2u_1(t) + u_2(t) - e^{-u_1(t)} \\ \dot{u}_2(t) = u_1(t) - 2u_2(t) - e^{-u_2(t)} \end{cases}$ $\quad \begin{cases} u_1(0) = 1 \\ u_2(0) = 1 \end{cases}$

Schéma d'Euler progr : $\dfrac{u^{n+1} - u^n}{h} = f(u_n, t_n$

So here, for example, I note : x, the vector of entries x1,x2,..., xM, so f(x,t), would be the vector function, its first component is f1, which depends on the entries of x, x1, x2,...,xM, and of time, f2 depends of x1, x2,..., xM and time, until fM that also depends o x1, x2,... until xM, and time. Voilà. Let's give an example in the case M=2. We look for u(t) the vector of entries u1(t), u2(t), such that u1'(t), the first component of the vector u', u2'(t), the second entry of the vector u', equals -so here let's give an example of a function f-. for example, take -u1(t)+u2(t)-exp(-u1(t)), plus *u2(t)* moins *exponentielle moins u1(t)*. this example corresponds in fact to the discretization of a nonlinear heat problem, chapter 12 in the book, and u2'(t) is u1(t)-2u2(t) -exp(-u2(t)), with 2 initial values u1 at the time 0 is known, and u2 at the time 0 is also known, for example 1. Alors, écrivons dans le cas général : *u point* égal *f* avec un vecteur sur le *u point* et un vecteur sur le *f*, Let's write the progressive Euler's scheme, and the backward Euler's scheme. The forward Euler's scheme, was (u_(n+1)-u_n)/h, equals f(u_n,tn).

Notes

Summary

1m 49s

Chap 9 - Systèmes d'éq. différentielles du 1er ordre.

Cherche $\vec{u}(t) = \begin{pmatrix} u_1(t) \\ u_2(t) \\ \vdots \\ u_n(t) \end{pmatrix}$ tq $\begin{cases} \dot{\vec{u}}(t) = \vec{f}(\vec{u}(t), t) & t>0 \\ \vec{u}(0) = \vec{u}_0 \end{cases}$ où $\vec{f} : \mathbb{R}^n \times \mathbb{R}_+ \to \mathbb{R}^n$, $(\vec{x}, t) \to \vec{f}(\vec{x}, t)$ $\quad \vec{f}(\vec{x}, t) = \begin{pmatrix} f_1(x_1, x_2 \dots x_n, t) \\ f_2(x_1, x_2 \dots x_n, t) \\ \vdots \\ f_n(x_1, x_2, \dots x_n, t) \end{pmatrix}$

Ex : $n = 2$ $\quad \vec{u}(t) = \begin{pmatrix} u_1(t) \\ u_2(t) \end{pmatrix}$ tq $\begin{cases} \dot{u}_1(t) = -2u_1(t) + u_2(t) - e^{-u_1(t)} \\ \dot{u}_2(t) = u_1(t) - 2u_2(t) - e^{-u_2(t)} \end{cases}$ $\quad \begin{cases} u_1(0) = 1 \\ u_2(0) = 1 \end{cases}$

Schéma d'Euler progr : $\dfrac{\vec{u}^{n+1} - \vec{u}^n}{h} = \vec{f}(\vec{u}^n, t_n)$ $\quad \vec{u}^n = \begin{pmatrix} u_1^n \\ u_2^n \\ \vdots \\ u_n^n \end{pmatrix}$ approximation de $\vec{u}(t_n) = \begin{pmatrix} u_1(t_n) \\ u_2(t_n) \\ \vdots \\ u_n(t_n) \end{pmatrix}$

$t_n = nh \quad n = 0, 1, 2 \dots$

explicite $\quad \vec{u}^{n+1} = \vec{u}^n + h\vec{f}(\vec{u}^n, t_n)$

Schéma d'Euler rétrograde : $\dfrac{\vec{u}^{n+1} - \vec{u}^n}{h} = \vec{f}(\vec{u}^{n+1}, t_{n+1})$

implicite $\quad \vec{u}^{n+1} - \vec{u}^n - h\vec{f}(\vec{u}^{n+1}, t_{n+1}) = \vec{0}$

So here I must simply add a vector on u_(n+1), a vector on u_n, a vector on f, and here a vector on u_n. So u^n denotes the vector with components u_1^nn, u_2^n,... until u_M^n, which is an approximation of the exact solution u evaluated at tn. So recall that tn is n*h, n=0,1,2,.... Donc on approche la solution aux temps *t0*, *t1*, *t2*, etc. Donc *u* au temps *tn*, c'est simplement le vecteur *u* évalué au temps *tn*. I write u_1(tn), u2(tn)... until uM evaluated at the time tn. Here we have 2 indices the component index, a subscript and the index corresponding to time, a superscript. That's Euler's forward method. This scheme is explicit in the sense where we can write u^(n+1) as a function of u^n. Donc c'est un schéma explicite, comme tout à l'heure. So the vector u^(n+1), this is an equality between two vectors, that is all the components of the vector on the left hand side are equal to the components of the vector on the right hand side, equals u^n+h times f(u_n,tn). This is the backward Euler's scheme: qui est que, (u^(n+1)-u^n)/h equals f(u^(n+1),t(n+1)). So this is an implicit scheme, which means we have an implicit relation for u^(n+1), u^(n+1)-u^n -h f(u^(n+1),t(n+1)) equals 0, the vector 0.

Notes

Summary

4m 14s

Chap 9 - Systèmes d'éq. différentielles du 1er ordre.

Cherche $\vec{u}(t) = \begin{pmatrix} u_1(t) \\ u_2(t) \\ \vdots \\ u_M(t) \end{pmatrix}$ tq $\begin{cases} \dot{\vec{u}}(t) = \vec{f}(\vec{u}(t), t) & t > 0 \\ \vec{u}(0) = \vec{u}_0 \end{cases}$ où $\vec{f} : \mathbb{R}^M \times \mathbb{R}_+ \to \mathbb{R}^n$, $(\vec{x}, t) \to \vec{f}(\vec{x}, t)$ $\quad \vec{f}(\vec{x}, t) = \begin{pmatrix} f_1(x_1, x_2 \cdots x_M, t) \\ f_2(x_1, x_2 \cdots x_M, t) \\ \vdots \\ f_M(x_1, x_2 \cdots x_M, t) \end{pmatrix}$

Ex : $M = 2$ $\quad \vec{u}(t) = \begin{pmatrix} u_1(t) \\ u_2(t) \end{pmatrix}$ tq $\begin{cases} \dot{u}_1(t) = -2u_1(t) + u_2(t) - e^{-u_1(t)} \\ \dot{u}_2(t) = u_1(t) - 2u_2(t) - e^{-u_2(t)} \end{cases}$ $\quad \begin{cases} u_1(0) = 1 \\ u_2(0) = 1 \end{cases}$

Schéma d'Euler progr : $\dfrac{\vec{u}^{n+1} - \vec{u}^n}{h} = \vec{f}(\vec{u}^n, t_n)$ $\quad \vec{u}^n = \begin{pmatrix} u_1^n \\ u_2^n \\ \vdots \\ u_M^n \end{pmatrix}$ approximation de $\vec{u}(t_n) = \begin{pmatrix} u_1(t_n) \\ u_2(t_n) \\ \vdots \\ u_M(t_n) \end{pmatrix}$

$t_n = nh \quad n = 0, 1, 2 \dots$

explicite $\quad \vec{u}^{n+1} = \vec{u}^n + h\vec{f}(\vec{u}^n, t_n)$

Schéma d'Euler rétrograde : $\dfrac{\vec{u}^{n+1} - \vec{u}^n}{h} = \vec{f}(\vec{u}^{n+1}, t_{n+1})$

implicite $\quad \underbrace{\vec{u}^{n+1} - \vec{u}^n - h\vec{f}(\vec{u}^{n+1}, t_{n+1})}_{\vec{g}(\vec{u}^{n+1}) = \vec{0}} = \vec{0}$ $\quad$ à chaque pas de temps, système non lin. M éq. et M inconnues $\to$ Newton.

Ex : Euler progr $\quad \dfrac{u_1^{n+1} - u_1^n}{h} = -2u_1^n + u_2^n$

So we are looking for u^(n+1) such that g(u^(n+1))=0. So at each time step, we must solve a nonlinear system with M equations and M unkowns. The M unknowns are written here, in vector form, and the M unknowns, are u1^(n+1), u2^(n+1), until u_M^(n+1). We'll use Newton's method to solve this linear system You'll do that in exercise class, and write the corresponding program. For example, let's go back to the example from before. Euler's forward scheme, in the case where this differential equation here, would be written as follows : So, forward Euler, would be written here : (u1^(n+1)-u1^n)/h, so it's the approximation of u1' at the time tn. And on the right hand side we would have -2*u1^n+u2^n minus exp(-u1^n). The second equation would become here : (u2^(n+1)-u2^n)/h equals u1^n-2*u2^n minus exp (-u2^n). Therefore we can write u1^(n+1) as a function previously computed values, so as u2^(n+1). de toutes ces grandeurs qui ont déjà été calculées. But in the case of Euler's backward scheme, it's written the following way : you still have (u1_(n+1)-u1_n)/h, donc l'approximation, ici, de *u1 point* au temps, cette fois-ci, *tn+1*, equals -2u1^n+1 +u2^(n+1), -exp(u1^(n+1)).

Notes

Summary

6m 49s

Chap 9. Systèmes d'éq. différentielles du 1^{er} ordre.

Cherche $\vec{u}(t) = \begin{pmatrix} u_1(t) \\ u_2(t) \\ \vdots \\ u_M(t) \end{pmatrix}$ tq $\begin{cases} \dot{\vec{u}}(t) = \vec{f}(\vec{u}(t), t) & t>0 \\ \vec{u}(0) = \vec{u}_0 \end{cases}$ où $\vec{f}: \mathbb{R}^M \times \mathbb{R}_+ \to \mathbb{R}^M$, $(\vec{x}, t) \to \vec{f}(\vec{x}, t)$

$$\vec{f}(\vec{x},t) = \begin{pmatrix} f_1(x_1, x_2 \cdots x_M, t) \\ f_2(x_1, x_2 \cdots x_M, t) \\ \vdots \\ f_M(x_1, x_2, \dots x_M, t) \end{pmatrix}$$

Ex: $M=2$ $\vec{u}(t) = \begin{pmatrix} u_1(t) \\ u_2(t) \end{pmatrix}$ tq $\begin{cases} \dot{u}_1(t) = -2u_1(t) + u_2(t) - e^{-u_1(t)} \\ \dot{u}_2(t) = u_1(t) - 2u_2(t) - e^{-u_2(t)} \end{cases}$ $\begin{cases} u_1(0) = 1 \\ u_2(0) = 1 \end{cases}$

Schéma d'Euler progr : $\dfrac{\vec{u}^{n+1} - \vec{u}^n}{h} = \vec{f}(\vec{u}^n, t_n)$ $\quad \vec{u}^n = \begin{pmatrix} u_1^n \\ u_2^n \\ \vdots \\ u_M^n \end{pmatrix}$ approximation de $\vec{u}(t_n) = \begin{pmatrix} u_1(t_n) \\ u_2(t_n) \\ \vdots \\ u_M(t_n) \end{pmatrix}$

$t_n = nh \quad n = 0, 1, 2 \dots$

explicite $\quad \vec{u}^{n+1} = \vec{u}^n + h\vec{f}(\vec{u}^n, t_n)$

Schéma d'Euler rétrograde : $\dfrac{\vec{u}^{n+1} - \vec{u}^n}{h} = \vec{f}(\vec{u}^{n+1}, t_{n+1})$

implicite $\quad \underbrace{\vec{u}^{n+1} - \vec{u}^n - h\vec{f}(\vec{u}^{n+1}, t_{n+1})}_{\vec{g}(\vec{u}^{n+1})} = \vec{0}$ $\quad$ à chaque pas de temps, système non lin. M éq. et M inconnues → Newton.

Ex: Euler progr $\quad \dfrac{u_1^{n+1} - u_1^n}{h} = -2u_1^n + u_2^n$

And for the second equation you would have: (u2^(n+1)-u2^n)/h equals u1^(n+1) -2u2^(n+1) -exp(u2^(n+1)). So, we obtain a nonlinear system with unknowns u1^(n+1) and u2^(n+1) that satisfies those 2 nonlinear equations. Donc vous avez ici : *u1n+1*, *u1n+1*, *u2n+1*, *u1n+1*. So at each time step, a nonlinear system of two equations has to be solved. The unknowns are u1^(n+1), u2^(n+1), and here are both equations. So, we use Newton's method to solve this nonlinear system of 2 equations with 2 unknowns. In the general case, in the case where we have a differential system of M equations you need to solve, at each time step, a nonlinear system of M equations and M unknowns.

Notes

Summary

9m 47s