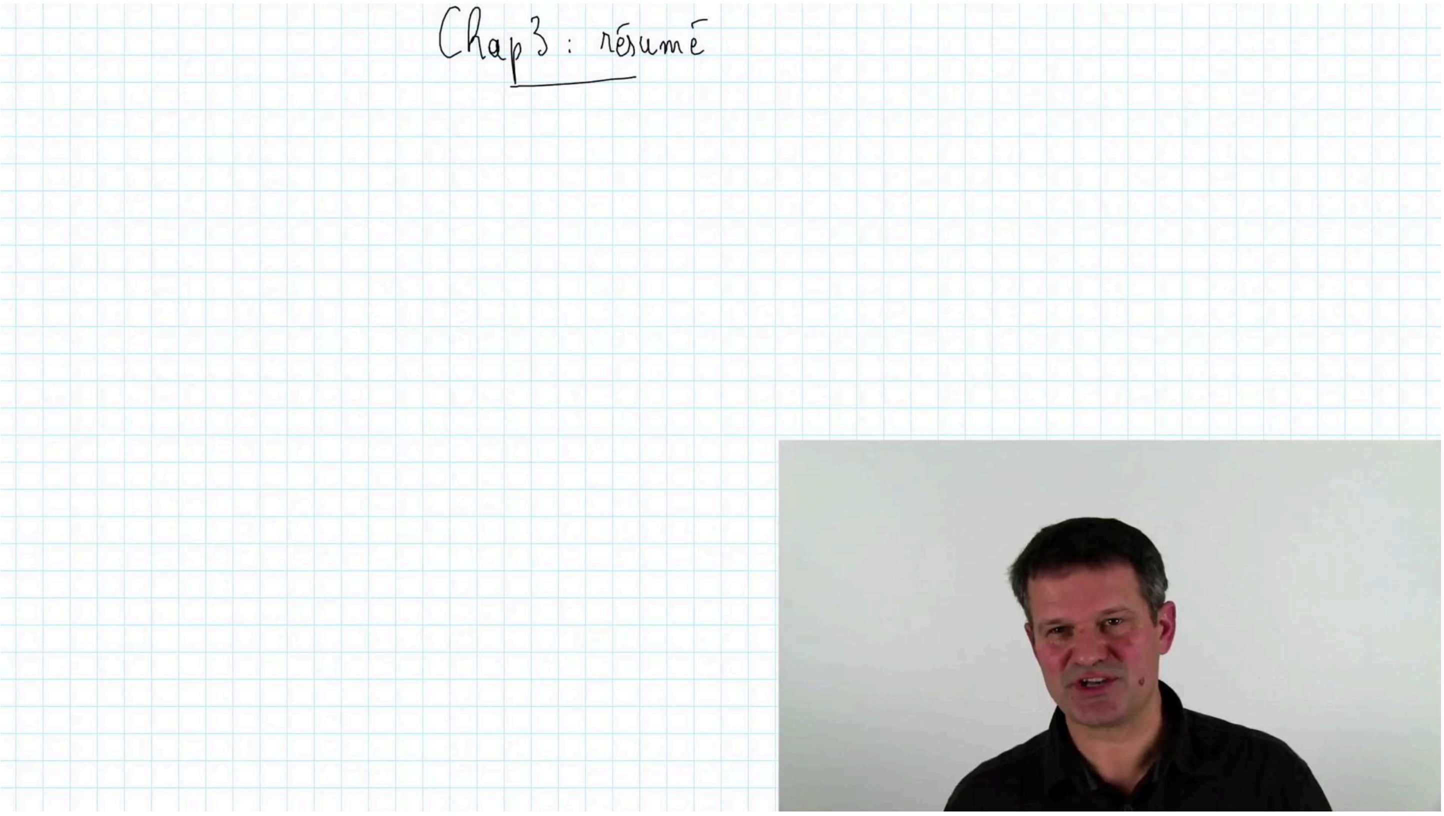
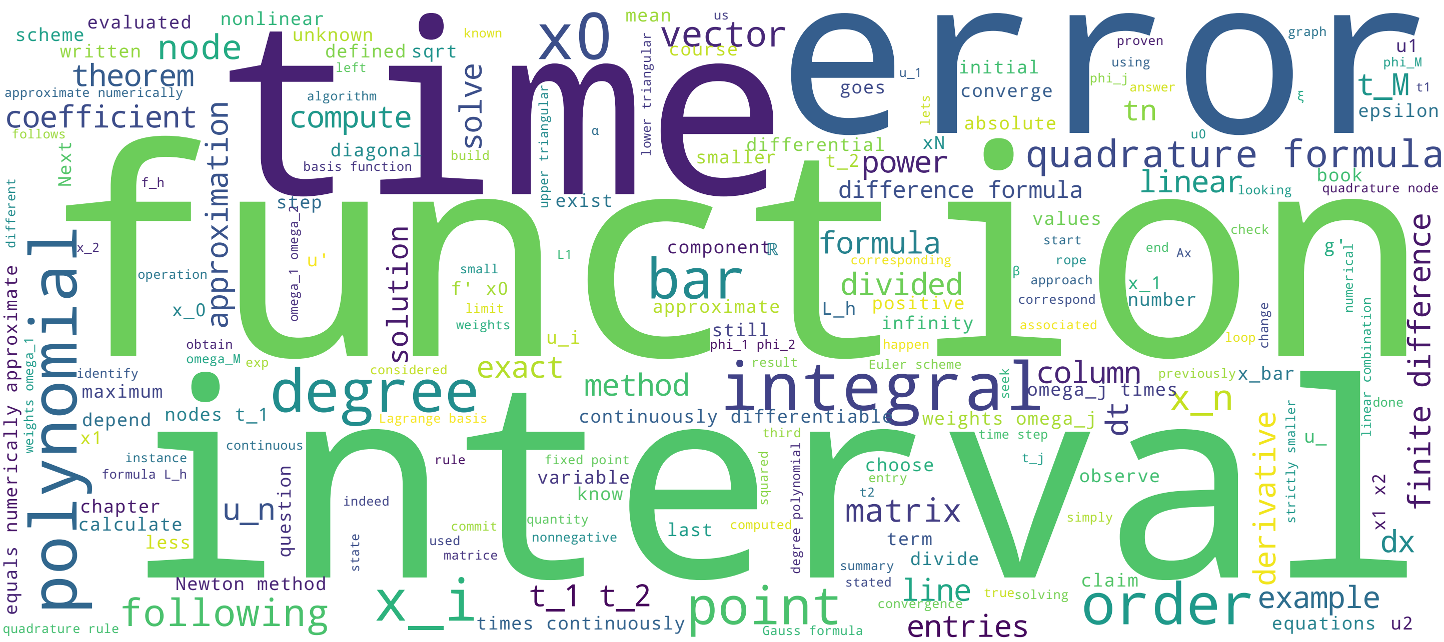

Chap3 : résumé


function
interval
time
error
polynomial
integral
degree
quadrature formula
x_i
point
order
bar
matrix
difference formula
finite difference
derivative
approximation
solution
solve
x0
vector
node
theorem
coefficient
compute
exact
method
following
divided
formula
column
x_n
linear
tn
t_M
power
example
entries
t_1 t_2
u_n
dx
dt
line

EPFL

Video

Chap 3 : résumé

- $f : [a,b] \to \mathbb{R}$ cont $\int_a^b f(x)dx$ $\qquad h = (b-a)/N$
- $\int_a^b f(x)dx = \frac{h}{2} \sum_{i=0}^{N-1} \int_{-1}^{1} f(x_i + h\frac{t+1}{2})dt$
- $J(g) = \omega_1 g(t_1) + \omega_2 g(t_2) + \dots + \omega_M g(t_M)$ appr. $\int_{-1}^{1} g(t)$.

Now lets give a summary of chapter 3, probably the hardest one in this course. So, we want to approximate numerically, for a function f defined on the interval [a,b] in R and continuous on that interval, we seek to approximate numerically the integral between a and b of f(x) dx. So how do we do this? We divide the interval [a,b] into sub-intervals x_i to x_i+1 and we have proven that the integral between a and b of f(x) dx can be written as a sum: h / 2 with h equal to b minus a over N; it is a sum over each sub-interval, for i from 0 to N-1; with the integral over -1 to 1. We have done a change of variable to have the integral over -1 to 1. The function f evaluated in x_i plus h times (t+1) dt. Next we give ourselves the quadrature formula J(g) to numerically approximate the integral over -1 and 1 of a function g(t) dt. Then J(g) is omega_1 times g(t_1) plus omega_2 times g(t_2) up to omega_M times g(t_M). When I give myself a quadrature rule, this means that the nodes t_1, t_2 up to t_M and the corresponding weights omega_1, omega_2 up to omega_M are prescribed. This quadrature formula is to numerically approximate the integral between -1 and 1 of g(t) dt.

Notes

Summary

0m 00s

Chap 3 : résumé

- $f:[a,b] \to \mathbb{R}$ cont $\int_a^b f(x)\,dx$ $\quad h=(b-a)/N$
- $\int_a^b f(x)\,dx = \frac{h}{2} \sum_{i=0}^{N-1} \int_{-1}^{1} f\left(x_i + h\,\frac{t+1}{2}\right) dt$
- $J(g) = \omega_1 g(t_1) + \omega_2 g(t_2) + \dots + \omega_M g(t_M)$ appr. $\int_{-1}^{1} g(t)\,dt$
- $L_h(f) = \frac{h}{2} \sum_{i=0}^{N-1} \sum_{j=1}^{M} \omega_j f\left(x_i + h\,\frac{t_j+1}{2}\right)$
- thm 3.1 : $\int_{-1}^{1} p(t)\,dt = J(p)$ $\forall p \in \mathbb{P}_r$ $\quad \left| \int_a^b f(x)\,dx - L_h(f) \right| = O(h^{r+1})$ $\quad f \in C^{r+1}[a,b]$
- thm 3.2. $t_1 t_2 \dots t_M$ donnés $\varphi_1 \varphi_2 \dots \varphi_M$ base de Lagr de $\mathbb{P}_{M-1}$ ass. $t_1 t_2 \dots t_M$

By applying the quadrature formula to this function, we have an approximation, denoted as L_h(f), which was equal to h over 2 times the sum over all the intervals, for i from 0 to N-1, of the sum over all the nodes times weights, for j from 1 to M, the weights omega_j times the function f evaluated in x_i + h times (t_j + 1) divided by 2. Next we stated a theorem, theorem 3.1, which is the following: if the quadrature formula is exact for polynomials of degree r, thus suppose that the quadrature formula is exact for polynomials of degree r, that is the integral over -1 to 1 of p(t) dt equal to J(p), equals the sum of the omega_j times p(t_j) for all polynomial p of degree r, then the error I commit when I approximate the integral between a and b of f(x) dx by the formula L_h(f), then the error is of order h to the power r+1 assuming that the function f is sufficiently smooth, that is r+1 continuously differentiable in the interval [a,b]. Next, theorem 3.2 states the following: that is if the nodes are given, say t_1, t_2 up to t_M on the interval [-1,1] are given, we can then build the functions phi_1, phi_2 up to phi_M which are the Lagrange basis functions of the polynomial p of degree M-1 associated to the nodes t_1, t_2 up to t_M.

Notes

Summary

1m 55s

Chap 3 : résumé

- $f : [a,b] \to \mathbb{R}$ cont $\int_a^b f(x)dx$ $h = (b-a)/N$
- $\int_a^b f(x)dx = \frac{h}{2} \sum_{i=0}^{N-1} \int_{-1}^{1} f(x_i + h \frac{t+1}{2}) dt$
- $J(g) = \omega_1 g(t_1) + \omega_2 g(t_2) + \dots + \omega_M g(t_M)$ appr. $\int_{-1}^{1} g(t)dt$
- $L_h(f) = \frac{h}{2} \sum_{i=0}^{N-1} \sum_{j=1}^{M} \omega_j f(x_i + h \frac{t_j+1}{2})$
- thm 3.1 : $\int_{-1}^{1} p(t)dt = J(p)$ $\forall p \in \mathbb{P}_r$ $\left|\int_a^b f(x)dx - L_h(f)\right| = O(h^{r+1})$ $f \in C^{r+1}[a,b]$
- thm 3.2 . $t_1 t_2 \dots t_M$ donnés $\varphi_1 \varphi_2 \dots \varphi_M$ base de Lagr. de $\mathbb{P}_{M-1}$ ass. $t_1 t_2 \dots t_M$ $\omega_j = \int_{-1}^{1} \varphi_j(t)dt$ $j = 1, \dots, M$
  $\int_{-1}^{1} p(t)dt = J(p)$ $\forall p \in \mathbb{P}_{M-1}$ $\left|\int_a^b f(x) - L_h(f)\right| = O(h^M)$ $f \in C^M[a,b]$
- $t_1 t_2 \dots t_M$ zéros polyn Gauss-leg.
  $\int_{-1}^{1} p(t)dt = J(p)$ $\forall p \in \mathbb{P}_{2M-1}$ $\left|\int_a^b f(x)dx - L_h(f)\right| = O(h^{2M})$ $f \in C^{2M}[a,b]$
- Ex: trap. rect $O(h^2)$ Simpson $O(h^4)$ Gauss 2 $O(h^4)$

The weights omega_j are then computed from the following formula omega_j equal to the integral over -1 to 1 of phi_j(t) dt, for all j from 1 to M. In this case I state that the quadrature formula is exact for all polynomials of degree M-1. If the function f is smooth enough, the error I commit with this formula here is an error of order h to the power M, this is true if f is M times continuously differentiable on the interval [a,b]. Now it happens that there exists a smart choice of the nodes t_1, t_2 up to t_M. If t_1, t_2 up to t_M are the zeros of the Gauss-Legendre polynomial, this can be considered as a good choice, since the quadrature formula is exact for polynomials of degree 2M-1, instead of M-1. We shift from M-1 to 2M-1 just by choosing adequately the nodes. Then the error, the integral from a to b of f(x) dx minus L_h(f), L_h(f) which is defined here, the error is of order h to the power 2M, provided f is 2M times continuously differentiable on the interval [a,b]. For instance, the trapezoidal formula and the midpoint formula are quadrature formula of order h to the power 2. Simpson's formula, a rule with 3 nodes, is of order h to the power 4. The Gauss formula with 2 nodes, the 2 nodes being -1 / sqrt(3) and 1 / sqrt(3), has convergence of order h to the power 4.

Notes

Summary

4m 03s