+

g :

Th

a

t p

fi

F

M

i

|

en

=

A

( △

h

f

)( a

i

) +

B

( △

h

g

)( a

i

).

=

A

[

f

(

a

i

+

1

)

−

f

(

a

i

)

] +

B

[

g

(

a

i

+

1

)

−

g

(

a

i

)

B

g

)

)

(

a

i

) =

(

A

f

+

B

g

)

(

a

i

+

1

)

−

(

A

f

+

B

g

)

(

a

i

)

M →

R

;

A

, B

∈

R

a

n

d

i <

m

,

w

e h

a

v

e

e o

p

e

r

a

t

o

r

△ h

i

s a

l

i

n

e

a

r o

p

e

r

a

t

o

r w

i

t

h

r

e

s

p

e

c

t t

o

in

t

a

i

.

n

it

e d

iff

e

re

n

c

e o

f th

e fi

rs

t o

rd

er

of th

e fu

n

ct

(

△

h

f

)

(

a

i

) =

f

(

a

i

+

1

)

−

f

(

a

i

)

,

i <

m

o

r

f

∈

F,

→

R

}.

a

i

=

a

+

i

h

,

w

i

t

h

i

=

0

, .

.

, m

.

;

a

, h

∈

R

∗

, m

∈

N

∗

c

es

1

.

2. F

in

it

e a

n

d d

iv

id

e

d d

iff

e

re

n

c

es

]

(1

)

o

f.

io

n

f,

}

a

n

d

(△

ph

(△

kh

f

0

I

f

0

≤

i.

k

-

t

h o

r

(

△

h

k

−

1

f

(△

h

(△

L

e

t

0

≤

))(

i < m

de

)(

a

h

k−

1

i <

a

i

) =

△

kh

(

;

k, p

∈

r fi

n

it

e d

iff

i

+

1

) −

(△

f

))(

a

i

)

m

, k ∈

N

△

ph

f

N a

er

h

k−

1

an

)

(

a

i

) =

(

△

h

p

+

k

f

)

(

a

i

)

.

(

3

)

n

d

1

≤

p

+

k

≤ m

−

i

,

t

h

e

n

e

n

c

e

o

f th

e fu

n

c

t

io

n

f,

w

it

h s

t

e

p

f

)

(

a

i

)

,

w

i

t

h

△

0h =

I

¸

s

i

△

1h =

△

h

(2

)

d

1

≤

k

≤ m

−

i

2

=

6

in

g

△

1

△

△

△

△

△

△

en

c

, f

h =

h

k−

1

h

fm

h

fm

h

fm

h

f1

h

f0

h

f

es t

3

=

1

4

, f

0

.

2

5, a

f

i

+

1

−

△

−

1

−

2

△

2h

−

3

△

△

△

△

2h

ab

le

:

4 =

=

1

h

k−

1

fm

−

2h

f

m

−

2h

f

1

2h

f

0

f

(

fi

d

1

7 fo

r

m

, a

i

=

f

i

, k

=

1

, .

2

3

.

.

.

△

.

.

.

△

.

.

.

△

e

n

o

t

e

s

f

t

a

+

.

, m.

h

m−

h

m−

h

m−

(a

i

)

he fi

n

it

e d

iff

e

r

en

c

es ta

b

le

.

i

h

, i

=

0

,

4

,

a

n

d

f0

=

0

,

;

i

=

0

,

1

, .

.

, m .

−

k

.

1

f1

1

f

0

△

mh

f

0

1

f

△

mh

f

)

=

e

x

fi

n

d

(△

kh

f

)(

a

i

),

w

i

t

h

a

i

=

a

+

i

h

, i

∈

N

.

3

8

−

5

4 4

−

9

2 2 2

−

11

△

h

f

△

2hf

△

3hf

△

4hf

g

e

t:

=

α

(D

f

)(

x

r

) +

β

(D

x

r

+1

β

g

)

)

(

x

r

) =

(

α

f

+

β

g

)

(

x

r

+

1

)

2

T

h

e o

p

e

r

a

t

o

r

D

i

s l

i

n

e

a

r w

i

r

a

n

d

x

r

+

1

.

fi

r

st o

rd

er d

iv

id

e

d d

iff

e

re

n

ce

(

D

f

)

(

x

r

) :

=

[

x

, x

r

r

+

1

;

f

] =

f

1

F

o

r

r

∈

N, r < m

,

|

x

i

∈

R

, i

=

0

,

1

, .

.

, m

.

, m

∈

N

∗

D

iv

id

e

d d

iff

e

re

n

g

)(

x

r

),

f

o

r

α

, β ∈

R

.

−

x

r

−

(

α

f

+

β

g

)

(

x

r

)

t

h r

e

s

p

e

c

t t

o

f.

o

f th

e fu

n

c

t

io

n

f,

r

e

g

a

r

d

x

r

+

1

−

x

r

(x

r

+

1

) −

f

(x

r

)

}

a

n

d

f

:

X

→

R

.

c

es

(4

)

in

g

i=

0

(

D

m

f

)

(

x

0

) =

X m

d

k

= m

w

e h

a

v

e

T

h

e o

p

e

r

a

t

o

r

D

k

i

s l

.

.

, x

.

r

+

k

;

f

i =

h

x

r

+

1

, .

.

a

l

s

o

d

e

n

o

t

e

d

b

y

hx

, .

r

k

-

t

h o

rd

er d

iv

id

e

d d

x

r

+

k

−

x

r

(D

k

−

1

f

)(

x

r

+

1

) −

(D

3

L

e

t

, k r

∈

N

,

0

≤ r <

(x

i

−

x

0

)

.

.

.

|

.

.

.

(x

f

(x

i

)

in

e

ar w

it

h r

es

x

r

+

k

−

x

, x

.

r

+

k

;

f

i −

hx

, .

r

.

, x

.

r

+

k

;

f

i .

R

e

iff

e

re

n

ce

of

k

−

1

f

)

(

x

r

)

,

m

a

n

d

1

≤ k

i

−

x

m

)

.

(

p

e

c

t t

o

f.

r

.

.

, x

.

r

+

k

−

1

;

f

i

l

a

t

i

o

n

(5

) ca

n

b

e w

t

he fu

n

c

t

io

n

f,

a

t

(

w

i

t

h

D

0

=

1

,

D

1

=

D

≤ m

−

r

, m

∈

N

∗

.

T

7

)

(6

)

r

it

x

r

.

5

)

,

he

i

)

,

i

=

0

,

1

, .

.

, m

.

.

1

D

fm

−

1

D

fm

−

2

D

2

fm

−

2

.

.

.

D

f2

D

2

f2

D

f1

D

2

f1

D

m

−

1

f1

D

f0

D

2

f0

.

.

.

D

m

−

1

f0

D

m

f0

D

f

D

2

f

.

.

.

D

m

−

1

f

D

m

f

id

e

d d

iff

e

r

en

ce

s:

p

r

o

o

f fo

l

lo

w

s by co

m

p

le

te in

du

c

t

io

n

w

it

h re

s

p

e

c

t to

m

.

k=

0

[

x

0

, .

.

, x

.

m

;

f g

] =

X m

[

x

0

, .

.

, x

.

k

;

f

]

[

x

, .

k

.

, x

.

m

;

g

]

.

5

I

f

f

, g

:

X

→

R

t

h

e

n

5

, x

4

=

7

a

n

d

f0

=

3, f1

=

5

, f2

=

9, f3

=

1

1

, f4

=

1

5

.

F

o

r

m

t

h

e

d

iv

id

e

d d

iff

e

r

e

n

c

e

s t

a

b

l

e

fo

r

x

0

=

1

, x

1

=

2

,

8

a

n

d

f0

=

4

, f1

=

8, f2

=

2

0, f3

=

4

8.

1

F

o

r

m

t

h

e

d

i

v

i

d

e

d

d

i

ff

e

r

e

n

c

e

s t

a

b

l

e

f

o

r

x

0

=

2

, x

1

=

4

,

1 0

D

2

f D

3

f

19

fo

r

m

th

e d

iv

id

e

d d

iff

e

r

en

c

es ta

b

le

.

F

o

r

x

0

=

0, x

1

=

1

, x

2

=

2

, x

3

=

4

a

n

d

f0

=

3, f1

=

4

,

o

n

a

r

o

u

n

d t

h

e

s

p

e

c

ifi

c

p

o

in

t

x

0

, b

u

t

n

o

t

o

n

t

h

e

e

n

t

ir

e

T

a

y

lo

r

p

o

ly

n

o

m

ia

l

s

a

g

r

e

e

a

s

c

lo

s

e

l

y a

s

p

o

s

s

ib

l

e

w

it

h a

R

n

(

x

) =

(

x

(

−

n

x

+

0

1

)

n

)

+

!

1

f

(

n

+

1

)

(

ξ

)

.

]

t

h

e

r

e e

x

i

s

t

s a

n

u

m

b

e

r

ξ

b

e

t

w

e

e

n

x

0

a

n

d

x

s

u

c

h

t

h

a

t

t

h

e r

e

m

a

i

n

d

e

r (

t

h

e e

r

r

o

r

).

f

(

x

) =

T

n

(

x

) +

R

n

(x

),

th

e a

p

p

r

o

x

im

a

t

io

n

fo

r

m

u

la

k=

0

T

n

(

x

) =

X n

(

x

−

k

x

!

0

)

k

f

(

k

)

(

x

0

)

(

8

)

, b]

a

n

d

c

o

n

s

i

d

e

r

x

0

∈

[

a

, b

].

T

h

e T

a

y

l

o

r p

o

l

y

n

o

m

i

a

l i

s

1

(T

a

y

l

o

r t

h

e

o

r

e

m

) L

e

t

f ∈

C

n

[

a

, b

], s

u

c

h t

h

a

t t

h

e

r

e e

x

i

s

t

s

2

.

1. T

a

y

lo

r in

te

r

p

o

la

t

io

n

COURSE 2

2.2. Lagrange interpolation

Let [*a, b*] *⊂* R*, xi ∈* [*a, b*]*, i* = 0*,* 1*, ..., m* such that *xi 6*= *xj*for *i 6*= *j* and consider *f* : [*a, b*] *→* R*.*

The Lagrange interpolation problem (LIP) consists in determining the polynomial *P* of the smallest degree for which

*P*(*xi*) = *f*(*xi*)*, i* = 0*,* 1*, ..., m* (1)

i.e., the polynomial of the smallest degree which passes through the distinct points (*xi, f*(*xi*))*, i* = 0*,* 1*, ..., m.*

Definition 1 *A solution of (LIP) is called* Lagrange interpolation polynomial*, denoted by Lmf.*

Remark 2 *We have (Lmf*)(*xi*) = *f*(*xi*)*, i* = 0*,* 1*, ..., m.*

*Lmf ∈* P*m (*P*m is the space of polynomials of at most m-th degree*)*.*

The Lagrange interpolation polynomial is given by

*m*

(*Lmf*)(*x*) = X *i*=0

*`i*(*x*)*f*(*xi*)*,* (2)

where by *`i*(*x*) denote the Lagrange fundamental interpolation polynomials.

We have

*u*(*x*) = Q*m j*=0

(*x − xj*)*,*

*ui*(*x*) = *u*(*x*)

*x − xi*= (*x − x*0)*...*(*x − xi−*1)(*x − xi*+1)*...*(*x − xm*) = Q*m*

*j*=0

*j6*=*i*

and

*ui*(*xi*)=(*x − x*0)*...*(*x − xi−*1)(*x − xi*+1)*...*(*x − xm*)

(*x − xj*) *x − xj*

*`i*(*x*) = *ui*(*x*)

(*xi − x*0)*...*(*xi − xi−*1)(*xi − xi*+1)*...*(*xi − xm*)=Q*m*

*j*=0

*j6*=*i*

*xi − xj,* (3)

for *i* = 0*,* 1*, ..., m.*

How do we know that the interpolation polynomial expanded in powers of *x* (Course 1) and the polynomial constructed as in (2) represent the same polynomial?

Assume we have computed two interpolating polynomials *Q*(*x*) and *P*(*x*) each of degree *m* such that

*Q*(*xj*) = *f*(*xj*) = *P*(*xj*)*, j* = 0*, ..., m.*

Then we can form the difference

*d*(*x*) = *Q*(*x*) *− P*(*x*)*,*

that is a polynomial of degree less or equal to *m.*

Because of the interpolation property of *P* and *Q,* we have *d*(*xj*) = *Q*(*xj*) *− P*(*xj*) = 0*, j* = 0*, ..., m.*

A non-zero polynomial of degree less than or equal to *m* cannot have more than *m* zeros. But *d* has *m* + 1 distinct zeros, hence it must be identically zero*,* so *Q*(*x*) = *P*(*x*)*.*

Proposition 3 *We also have*

*`i*(*x*) = *u*(*x*)

(*x − xi*)*u0*(*xi*)*, i* = 0*,* 1*, ..., m.* (4)

Proof. We have *ui*(*x*) = *u*(*x*)

*~~x−x~~i,* so *u*(*x*) = *ui*(*x*)(*x − xi*)*.* We get *u0*(*x*) =

*ui*(*x*) + (*x − xi*)*u0i*(*x*)*,* whence it follows *u0*(*xi*) = *ui*(*xi*)*.* So, as *`i*(*x*) = *ui*(*x*)

*ui*(*xi*)

we get

*`i*(*x*) = *ui*(*x*)

*u0*(*xi*)=*u*(*x*)

(*x − xi*)*u0*(*xi*)*, i* = 0*,* 1*, ..., m.* (5)

Theorem 4 *The operator Lm is linear.*

Proof.

*m*

*m*

*Lm*(*αf* + *βg*)(*x*) = X *i*=0

*`i*(*x*)(*αf* + *βg*)(*xi*) = X *i*=0

[*`i*(*x*)*αf*(*xi*) + *`i*(*x*)*βg*(*xi*)]

= *α*(*Lmf*)(*x*) + *β*(*Lmg*)(*x*)*,*

so

*Lm*(*αf* + *βg*) = *αLmf* + *βLmg, ∀f, g* : [*a, b*] *→* R and *α, β ∈* R*.*

Example 5 *a) Consider the nodes x*0*, x*1 *and a function f to be inter polated.*

*b) Find the Lagrange polynomial that interpolates the data in the following table and find the approximative value of f*(*−*0*.*5)*.*

*x −*1 0 3

*f*(*x*) 8 *−*2 4

*Sol.*

a) We have *m* = 1*,*

*u*(*x*) = (*x − x*0)(*x − x*1)

*u*0(*x*) = *x − x*1

*u*1(*x*) = *x − x*0

(*L*1*f*)(*x*) = *l*0(*x*)*f*(*x*0) + *l*1(*x*)*f*(*x*1)

=*x − x*1

*x*0 *− x*1*f*(*x*0) + *x − x*0

*x*1 *− x*0*f*(*x*1)*,*

which is the line passing through the given points (*x*0*, f*(*x*0)) and (*x*1*, f*(*x*1))*.*

b) We have *m* = 2*.* The Lagrange polynomial is

(*L*2*f*)(*x*) = *l*0(*x*)*f*(*x*0) + *l*1(*x*)*f*(*x*1) + *l*2(*x*)*f*(*x*2)*.*

*u*(*x*) = (*x* + 1)(*x −* 0)(*x −* 3) and it follows

(*−*1 *−* 0)(*−*1 *−* 3)=14*x*(*x −* 3)

*l*0(*x*) = (*x −* 0)(*x −* 3)

*l*1(*x*) = (*x* + 1)(*x −* 3)

(0 + 1)(0 *−* 3)= *−*13(*x* + 1)(*x −* 3)

*l*2(*x*) = (*x* + 1)(*x −* 0)

(3 + 1)(3 *−* 0)=112*x*(*x* + 1)*,*

The polynomial is

(*L*2*f*)(*x*) = 2*x*(*x −* 3) + 23(*x* + 1)(*x −* 3) + 13*x*(*x* + 1)*.*

and (*L*2*f*)(*−*0*.*5) = 2*.*25*.*

Remark 6 *Disadvantages of the form (2) of Lagrange polynomial: requires many computations and if we add or substract a point we have to start with a complete new set of computations.*

Remark 7 *Formula (6) needs half of the number of arithmetic oper ations needed for (2) and it is easier to add or substract a point.*

The Lagrange polynomial generates the Lagrange interpolation for mula

*f* = *Lmf* + *Rmf,*

where *Rmf* denotes the remainder (the error).

Theorem 8 *Let α* = min*{x, x*0*, ..., xm} and β* = max*{x, x*0*, ..., xm}. If f ∈ Cm*[*α, β*] *and f*(*m*)*is derivable on* (*α, β*) *then ∀x ∈* (*α, β*)*, there exists ξ ∈* (*α, β*) *such that*

(*R mf*)(*x*) = *u*(*x*)

(*m* + 1)!*f*(*m*+1)(*ξ*)*.* (7)

Proof. Consider

*F*(*z*) =

*u*(*z*) (*Rmf*)(*z*) *u*(*x*) (*Rmf*)(*x*)

*.*

whence (*R mf*)(*x*) = *u*(*x*)

(*m*+1)!*f*(*m*+1)(*ξ*).

Corrolary 9 *If f ∈ Cm*+1[*a, b*] *then*

*|*(*Rmf*)(*x*)*| ≤ |u*(*x*)*|*

(*m* + 1)!

*f*(*m*+1)*∞, x ∈* [*a, b*]

*where k·k∞ denotes the uniform norm, and kfk∞* = max *x∈*[*a,b*]*|f*(*x*)*|.*

Example 10 *If we know that* lg 2 = 0*.*301*,* lg 3 = 0*.*477*,* lg 5 = 0*.*699*, find* lg 76*. Study the approximation error.*

Example 11 *Which is the limit of the error for computing √*115 *using Lagrange interpolation formula for the nodes x*0 = 100*, x*1 = 121 *and x*2 = 144? *Find the approximative value of √*115*.*

COURSE 3

The Aitken’s algorithm

Let [*a, b*] *⊂* R*, xi ∈* [*a, b*]*, i* = 0*,* 1*, ..., m* such that *xi ̸*= *xj*for *i ̸*= *j* and consider *f* : [*a, b*] *→* R*.*

Usually, for a practical approximation problem, for a given function *f* : [*a, b*] *→* R we have to find the approximation of *f*(*α*)*, α ∈* [*a, b*] with an error not greater than a given *ε >* 0*.*

If we have enough information about *f* and its derivatives, we use the inequality *|*(*Rmf*)(*x*)*| ≤ ε* to find *m* such that (*Lmf*)(*α*) approximates *f*(*α*) with the given precision.

We may use the condition *|u*(*x*)*|*

*f*(*m*+1)*∞≤ ε,* but it should be

known

(*m*+1)!

*f*(*m*+1)*∞*or a majorant of it.

A practical method for computing the Lagrange polynomial is the Aitken’s algorithm. This consists in generating the table:

*x*0 *f*00

*x*1 *f*10 *f*11

*x*2 *f*20 *f*21 *f*22

*x*3 *f*30 *f*31 *f*32 *f*33

...............

*xm fm*0 *fm*1 *fm*2 *fm*3 *... fmm*

where

*fi*0 = *f*(*xi*)*, i* = 0*,* 1*, ..., m,*

and

*fi,j*+1 =1

*xi − xj*

*fjj xj − x fij xi − x*

*, i* = 0*,* 1*, ..., m*; *j* = 0*, ..., i −* 1*.*

For example,

*f*11 =1

*x*1 *− x*0

=1

*f*00 *x*0 *− x*

*f*10 *x*1 *− x*

*x*1 *− x*0[*f*00(*x*1 *− x*) *− f*10(*x*0 *− x*)] *x*0 *− x*1*f*(*x*0) + *x − x*0

=*x − x*1

*x*1 *− x*0*f*(*x*1) = (*L*1*f*)(*x*)*,*

so *f*11 is the value in *x* of Lagrange polynomial for the nodes *x*0*, x*1*.* We have

*fii* = (*Lif*)(*x*)*,*

*Lif* being Lagrange polynomial for the nodes *x*0*, x*1*, ..., xi.* So *f*11*, f*22*, ..., fii, ..., fmm* is a sequence of approximations of *f*(*x*)*.*

If the interpolation procedure is convergent then the sequence is also convergent, i.e., lim*m→∞fmm* = *f*(*x*)*.* By Cauchy convergence criterion it follows

*i→∞|fii − fi−*1*,i−*1*|* = 0*.*

lim

This could be used as a stopping criterion, i.e.,

*fii − fi−*1*,i−*1 *≤ ε,* for a given precision *ε >* 0*.*

Recommendation is to sort the nodes *x*0*, x*1*, ..., xm* with respect to the distance to *x,* such that

*|xi − x| ≤*

*xj − x* if *i < j, i, j* = 1*, ..., m.*

Example 1 *Approximate √*115 *with precision ε* = 10*−*3*, using Aitken’s algorithm.*

Newton interpolation polynomial

A useful representation for Lagrange interpolation polynomial is *m*

(*Lmf*)(*x*) := (*Nmf*)(*x*) = *f*(*x*0) + X *i*=1

*m*

(*x − x*0)*...*(*x − xi−*1)(*Dif*)(*x*0) (1)

= *f*(*x*0) + X *i*=1

(*x − x*0)*...*(*x − xi−*1)[*x*0*, ..., xi*; *f*]*,*

which is called Newton interpolation polynomial; where (*Dif*)(*x*0)(or denoted [*x*0*, ..., xi*; *f*]) is the *i*-th order divided difference of the function *f* at *x*0*,* given by the table

*f Df D*2*f ... D*m*−*1*f Dmf*

*x*0 *f*0 *Df*0 *D*~~2~~*f*0 *... D~~m−~~*~~1~~*f*0 *D~~m~~f*0 *x*1 *f*1 *Df*1 *D*2*f*1 *Dm−*1*f*1

*x*2 *f*2 *Df*2 *D*2*f*2

*... ... ...*

*xm−*2 *fm−*2 *Dfm−*2 *D*2*fm−*2

*xm−*1 *fm−*1 *Dfm−*1

*xm fm*

Newton interpolation formula is

*f* = *Nmf* + *Rmf,*

where *Rmf* denotes the remainder.

Assume that we add the point (*x, f*(*x*)) at the top of the table of divided differences:

*f Df ... Dm*+1*f*

*x f*(*x*) (*Df*)(*x*) = [*x, x*0; *f*] [*x, x*0*, ..., xm*; *f*] *x*0 *f*(*x*0) (*Df*)(*x*0) = [*x*0*, x*1; *f*] *...*

*x*1 *f*(*x*1) (*Df*)(*x*1) = [*x*1*, x*2; *f*]

*... ... ...*

*xm−*1 *f*(*xm−*1) (*Df*)(*xm−*1) = [*xm−*1*, xm*; *f*]

*xm f*(*xm*)

For obtaining the interpolation polynomial we consider [*x, x*0; *f*] = *f*(*x*0) *− f*(*x*)

*x*0 *− x*=*⇒ f*(*x*) = *f*(*x*0) + (*x − x*0)[*x, x*0; *f*] (2)

[*x, x*0*, x*1; *f*] = [*x*0*, x*1; *f*] *−* [*x, x*0; *f*]

*x*1 *− x*(3)

=*⇒* [*x, x*0; *f*] = [*x*0*, x*1; *f*] + (*x − x*1)[*x, x*0*, x*1; *f*]*.*

Inserting (3) in (2) we get

*f*(*x*) = *f*(*x*0) + (*x − x*0)[*x*0*, x*1; *f*] + (*x − x*0)(*x − x*1)[*x, x*0*, x*1; *f*]*.* If we continue eliminating the divided differences involving *x* in the same way, we get

*f*(*x*) = (*Nmf*)(*x*) + (*Rmf*)(*x*)

with

*m*

(*Nmf*)(*x*) = *f*(*x*0) + X

*i*=1

(*x − x*0)*...*(*x − xi−*1)[*x*0*, ..., xi*; *f*]

and the remainder (the error) given by

(*Rmf*)(*x*) = (*x − x*0)*...*(*x − xm*)[*x, x*0*, ..., xm*; *f*]*.* (4)

We notice that

(*Nif*)(*x*) = (*Ni−*1*f*)(*x*) + (*x − x*0)*...*(*x − xi−*1)[*x*0*, ..., xi*; *f*] so the Newton polynomials of degree 2*,* 3*, ...,* can be iteratively gener ated, similarly to Aitken’s algorithm.

Remark 2 *The remainder for Lagrange interpolation formula is also given by*

(*R mf*)(*x*) = (*x − x*0)*...*(*x − xm*)

(*m* + 1)! *f*(*m*+1)(*ξ*)*,*

*with ξ between x, x*0*, ..., xm, so, by (4), it follows that* the divided differences are approximations of the derivatives

[*x, x*0*, ..., xm*; *f*] = *f*(*m*+1)(*ξ*)

(*m* + 1)! *.*

Example 3 *Find L*2*f for f*(*x*) = sin *πx, and x*0 = 0*, x*1 = 1~~6~~*, x*2 = 1~~2~~*, in both forms.*

Sol. a) We have *u*(*x*) = *x*(*x −* 1~~6~~)(*x −* 1~~2~~); *u*0(*x*) = (*x −* 1~~6~~)(*x −* 1~~2~~); *u*1(*x*) = *x*(*x −* 1~~2~~); *u*2(*x*) = *x*(*x −* 1~~6~~)

(*L*2*f*)(*x*) = X2 *i*=0

*li*(*x*)*f*(*xi*) = X2 *i*=0

*ui*(*x*)

*ui*(*xi*)*f*(*xi*)

=(*x −* 1~~6~~)(*x −* 1~~2~~)

(*−*1~~6~~)(*−*1~~2~~)0 +*x*(*x −* 1~~2~~)

2+*x*(*x −* 1~~6~~)

1~~2~~*·*1~~3~~1

= *−*3*x*2 +72*x.*

1

1~~6~~(*−*1~~3~~)

b)

(*N*2*f*)(*x*) = *f*(0) + X2 *i*=1

(*x − x*0)*...*(*x − xi−*1)(*Dif*)(*x*0)

= *f*(0) + (*x − x*0)(*Df*)(*x*0) + (*x − x*0)(*x − x*1)(*D*2*f*)(*x*0) = *x*(*Df*)(*x*0) + *x*(*x −*16)(*D*2*f*)(*x*0)

The table of divided differences:

*f Df D*2*f*

*x*

0

0 3 *−*3

1~~6~~1~~2~~

1~~2~~3~~2~~

1

so

(*N*2*f*)(*x*) = 3*x −* 3*x*(*x −*16) = *−*3*x*2 +72*x.*

*•* Useful if you have the Lagrange polynomial based on some set of data points (*xi, f*(*xi*))*, i* = 0*,* 1*, . . . , n*, and you get a new data point, (*xn*+1*, f*(*xn*+1)).

Definition 4 *Let f be a function defined at x*0*, x*1*, . . . , xn and suppose that m*1*, . . . , mk are k distinct integers with* 0 *≤ mi ≤ n, for every i. The Lagrange polynomial that interpolates f*(*x*) *at the k points xm*1*, . . . , xmkis denoted by Pm*1*,...,mk*(*x*)*.*

Example 5 *Consider x*0 = 1*, x*1 = 2*, x*2 = 3*, x*3 = 4*, x*4 = 6 *and f*(*x*) = *ex. Determine P*1*,*2*,*4(*x*) *and use it to approximate f*(5)*.* (*x*1 *− x*2)(*x*1 *− x*4)*f*(*x*1) + (*x − x*1)(*x − x*4)

*P*1*,*2*,*4(*x*) = (*x − x*2)(*x − x*4) =(*x −* 3)(*x −* 6)

(*x*2 *− x*1)(*x*2 *− x*4)*f*(*x*2) + (*x − x*1)(*x − x*2) (*x*4 *− x*1)(*x*4 *− x*1)*f*(*x*4)

(2 *−* 3)(2 *−* 6)*f*(2) + (*x −* 2)(*x −* 6)

(3 *−* 2)(3 *−* 6)*f*(3) + (*x −* 2)(*x −* 3)

(6 *−* 2)(6 *−* 3)*f*(6)

*So, f*(5) *≈ P*1*,*2*,*4(5)*, more specifically*

*P*1*,*2*,*4(5) = (5 *−* 3)(5 *−* 6)

(2 *−* 3)(2 *−* 6)*e*2 +(5 *−* 2)(5 *−* 6)

(3 *−* 2)(3 *−* 6)*e*3 +(5 *−* 2)(5 *−* 3)

(6 *−* 2)(6 *−* 3)*e*6

= *−*0*.*5*e*2 + *e*3 + 0*.*5*e*6 *≈* 218*.*105*.*

The following results present a method for recursively generating La grange’s polynomial approximation.

Theorem 6 *Let f be a function defined at the points x*0*, x*1*, . . . , xk and let xi and xj be two distinct points in this set. Then*

*P*(*x*) =(*x − xj*)*P*0*,*1*,...,j−*1*,j*+1*,...,k*(*x*) *−* (*x − xi*)*P*0*,*1*,...,i−*1*,i*+1*,...,k*(*x*) *xi − xj*

*is the kth Lagrange polynomial that approximates f at the k* + 1 *nodes x*0*, x*1*, . . . , xk.*

Denote by *Pj*(*x*) = *f*(*xj*).

The previous Theorem implies that the interpolating polynomials can

be generated recursively. For example

*P*0*,*1 =1

*x*1 *− x*0[(*x − x*0)*P*1 *−* (*x − x*1)*P*0]

*P*1*,*2 =1

*x*2 *− x*1[(*x − x*1)*P*2 *−* (*x − x*2)*P*1]

*P*0*,*1*,*2 =1 *x*2 *− x*0

h(*x − x*0)*P*1*,*2 *−* (*x − x*2)*P*0*,*1i

and so on. In the table below it can be seen how they are generated, taking into account the fact that each row is completed before the succeeding rows are begun.

*x*0 *P*0

*x*1 *P*1 *P*0*,*1

*x*2 *P*2 *P*1*,*2 *P*0*,*1*,*2

*x*3 *P*3 *P*2*,*3 *P*1*,*2*,*3 *P*0*,*1*,*2*,*3

*x*4 *P*4 *P*3*,*4 *P*2*,*3*,*4 *P*1*,*2*,*3*,*4 *P*0*,*1*,*2*,*3*,*4

This procedure is called the Neville’s method.

*•* Going down in the table *→* using consecutive points *xi*, with larger *i*

*•* Going to the right *→* increasing the degree of the interpolating polynomial.

*•* The points appear consecutively in each entry =*⇒* we need to give only a starting point and the number of additional points used.

*•* Let *Qi,j*(*x*) = *Pi−j,i−j*+1*,...,i−*1*,i*(*x*), with 0 *≤ j ≤ i*.

*•* The table becomes

*x*0 *P*0 = *Q*0*,*0

*x*1 *P*1 = *Q*1*,*0 *P*0*,*1 = *Q*1*,*1

*x*2 *P*2 = *Q*2*,*0 *P*1*,*2 = *Q*2*,*1 *P*0*,*1*,*2 = *Q*2*,*2

*x*3 *P*3 = *Q*3*,*0 *P*2*,*3 = *Q*3*,*1 *P*1*,*2*,*3 = *Q*3*,*2 *P*0*,*1*,*2*,*3 = *Q*3*,*3

*x*4 *P*4 = *Q*4*,*0 *P*3*,*4 = *Q*4*,*1 *P*2*,*3*,*4 = *Q*4*,*2 *P*1*,*2*,*3*,*4 = *Q*4*,*3 *P*0*,*1*,*2*,*3*,*4 = *Q*4*,*4 with *Qi,j* =(*xi−x*)*Qi−*1*,j−*1+(*x−xi−j*)*Qi,j−*1

*~~x~~i~~−x~~i−j,* for *i* = 1*,* 2*, . . . , n*; *j* = 1*,* 2*, . . . , i*.

Neville’s Iterated Interpolation Algorithm. Evaluate the polynomial *P* on *n* + 1 given nodes, *x*0*, . . . , xn*, at a given point *x*, for a function *f.*

Input : The nodes *x, x*0*, x*1*, . . . , xn*; the values of the function *f*(*x*0)*, . . . , f*(*xn*as the first column of *Q* (*Q*0*,*0*, Q*1*,*0*, . . . , Qn,*0).

Output : the table *Q* with *P*(*x*) = *Qn,n*.

Step 1 for *i* = 1*,* 2*, . . . , n*

for *j* = 1*,* 2*, . . . , i*

*Qi,j* =(*xi−x*)*Qi−*1*,j−*1+(*x−xi−j*)*Qi,j−*1

*~~x~~i~~−x~~i−j.*

Step 2 Output(Q); Stop.

It can be additionally added a stopping criterion

*|Qi,j − Qi−*1*,j−*1*| < ε,*

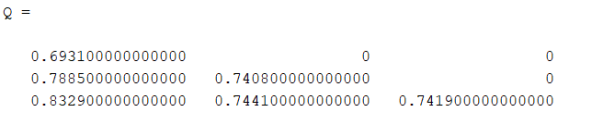
with *ε* a given error tolerance. If the inequality is not fulfilled, a new interpolation node *xi*+1 is added.

Example 7 *Approximate* ln(2*.*1) *using the function f*(*x*) = ln *x and the data*

*x 2 2.2 2.3*

ln *x 0.6931 0.7885 0.8329*

ln(2*.*1) = *f*(2*.*1) *and we have obtained the table*

**

So, ln(2*.*1) *≈* 0*.*7419, with an error of 3*.*7344 *·* 10*−*5.

Example 8 *Approximate √*115 *using Neville’s algorithm, with 3 given nodes.*

COURSE 4

2.3. Hermite interpolation

Example 1 *In the following table there are some data regarding a moving car. We may estimate the position (and the speed) of the car when the time is t* = 10 *using Hermite interpolation.*

*Time* 0 3 5 8 13

*Distance* 0 225 383 623 993

*Speed* 75 77 80 74 72

Let *xk ∈* [*a, b*]*, k* = 0*,* 1*, ..., m* be such that *xi 6*= *xj,* for *i 6*= *j* and let *rk ∈* N*, k* = 0*,* 1*, ..., m.* Consider *f* : [*a, b*] *→* R such that there exist *f*(*j*)(*xk*)*, k* = 0*,* 1*, ..., m*; *j* = 0*,* 1*, ..., rk* and *n* = *m* + *r*0 + *...* + *rm.*

The Hermite interpolation problem (HIP) consists in determining the polynomial *P* of the smallest degree for which

*P*(*j*)(*xk*) = *f*(*j*)(*xk*)*, k* = 0*, ..., m*; *j* = 0*, ..., rk.*

Definition 2 *A solution of (HIP) is called* Hermite interpolation polynomial*, denoted by Hnf.*

Hermite interpolation polynomial, *Hnf,* satisfies the interpolation conditions:

(*Hnf*)(*j*)(*xk*) = *f*(*j*)(*xk*)*, k* = 0*, ..., m*; *j* = 0*, ..., rk.*

Hermite interpolation polynomial is given by

*m*

(*Hnf*)(*x*) = X *k*=0

*rk*

X

*j*=0

*hkj*(*x*)*f*(*j*)(*xk*) *∈* P*n,* (1)

where *hkj*(*x*) denote the Hermite fundamental interpolation poly nomials. They fulfill the relations:

*h*(*p*)

*kj* (*xν*) = 0*, ν 6*= *k, p* = 0*,* 1*, ..., rν*

*h*(*p*)

*kj* (*xk*) = *δjp, p* = 0*,* 1*, ..., rk,* for *j* = 0*,* 1*, ..., rk* and *ν, k* = 0*,* 1*, ..., m,*

with *δjp* =

(1*, j* = *p* 0*, j 6*= *p.*

We denote by

*m*

(*x − xk*)*rk*+1 and *uk*(*x*) = *u*(*x*)

(*x − xk*)*rk*+1*.*

We have

*u*(*x*) = Y *k*=0

*hkj*(*x*) = (*x − xk*)*j j*!*uk*(*x*)

*r*X *k−j υ*=0

(*x − xk*)*ν ν*!

"1

*uk*(*x*)

#(*ν*)

*x*=*xk*

*.* (2)

Example 3 *Find the Hermite interpolation polynomial for a function f for which we know f*(0) = 1*, f0*(0) = 2 *and f*(1) = *−*3 *(equivalent with x*0 = 0 *multiple node of order 2 or double node, x*1 = 1 *simple node).*

Sol. We have *x*0 = 0*, x*1 = 1*, m* = 1*, r*0 = 1*, r*1 = 0*, n* = *m*+*r*0+*r*1 = 2

(*H*2*f*)(*x*) = X1 *k*=0

*rk*

X

*j*=0

*hkj*(*x*)*f*(*j*)(*xk*)

= *h*00(*x*)*f*(0) + *h*01(*x*)*f0*(0) + *h*10(*x*)*f*(1)*.*

We have *h*00*, h*01*, h*10. These fulfills relations:

*h*(*p*)

*kj* (*xν*) = 0*, ν 6*= *k, p* = 0*,* 1*, ..., rν*

*h*(*p*)

*kj* (*xk*) = *δjp, p* = 0*,* 1*, ..., rk,* for *j* = 0*,* 1*, ..., rk* and *ν, k* = 0*,* 1*, ..., m.* We have *h*00(*x*) = *a*1*x*2 + *b*1*x* + *c*1 *∈* P2*,* with *a*1*, b*1*, c*1 *∈* R*,* and the

system 

*h*00(*x*0) = 1 *h0*00(*x*0) = 0 *h*00(*x*1) = 0

*⇔*

 

*h*00(0) = 1 *h0*00(0) = 0 *h*00(1) = 0

that becomes 

*c*1 = 1

*b*1 = 0

*a*1 + *b*1 + *c*1 = 0*.*

Solution is: *a*1 = *−*1*, b*1 = 0*, c*1 = 1 so *h*00(*x*) = *−x*2 + 1*.* We have *h*01(*x*) = *a*2*x*2 + *b*2*x* + *c*2 *∈* P2*,* with *a*2*, b*2*, c*2 *∈* R*.* The system

is 

*h*01(*x*0) = 0 *h0*01(*x*0) = 1 *h*01(*x*1) = 0

*⇔*

 

*h*01(0) = 0 *h0*01(0) = 1 *h*01(1) = 0

and we get *h*01(*x*) = *−x*2 + *x.*

We have *h*10(*x*) = *a*3*x*2 + *b*3*x* + *c*3 *∈* P2*,* with *a*3*, b*3*, c*3 *∈* R*.* The system

is 

*h*10(*x*0) = 0 *h0*10(*x*0) = 0 *h*10(*x*1) = 1

*⇔*

 

*h*10(0) = 0 *h0*10(0) = 0 *h*10(1) = 1

and we get *h*10(*x*) = *x*2*.*

The Hermite polynomial is

(*H*2*f*)(*x*) = *−x*2 + 1 *−* 2*x*2 + 2*x −* 3*x*2 = *−*6*x*2 + 2*x* + 1*.*

The Hermite interpolation formula is

*f* = *Hnf* + *Rnf,*

where *Rnf* denotes the remainder term (the error).

Theorem 4 *If f ∈ Cn*[*α, β*] *and f*(*n*)*is derivable on* (*α, β*)*, with α* = min*{x, x*0*, ..., xm} and β* = max*{x, x*0*, ..., xm}, then there exists ξ ∈* (*α, β*) *such that*

(*Rnf*)(*x*) = *u*(*x*)

(*n* + 1)!*f*(*n*+1)(*ξ*)*.* (3)

Proof. Consider

*F*(*z*) =

*u*(*z*) (*Rnf*)(*z*) *u*(*x*) (*Rnf*)(*x*)

*.*

*F ∈ Cn*[*α, β*] and there exists *F*(*n*+1) on (*α, β*)*.*

We have

*F*(*x*) = 0*, F*(*j*)(*xk*) = 0*, k* = 0*, ..., m*; *j* = 0*, ..., rk*;

COURSE 5

Hermite interpolation with double nodes

Example 1 *In the following table there are some data regarding a moving car. We may estimate the position (and the speed) of the car when the time is t* = 10 *using Hermite interpolation.*

*Time* 0 3 5 8 13

*Distance* 0 225 383 623 993

*Speed* 75 77 80 74 72

Consider *f* : [*a, b*] *→* R*, x*0*, x*1*, ..., xm ∈* [*a, b*]

and the values *f*(*x*0)*, f*(*x*1)*, ..., f*(*xm*)*, f′*(*x*0)*, f′*(*x*1)*, ..., f′*(*xm*)*.*

The Hermite interpolation polynomial with double nodes, *H*2*m*+1*,* sat isfies the interpolation properties:

*H*2*m*+1(*xi*) = *f*(*xi*)*, i* = ~~0~~*~~, m~~,*

*H′*2*m*+1(*xi*) = *f′*(*xi*)*, i* = ~~0~~*~~, m~~.*

It is a polynomial of *n* = 2*m* + 1 degree*.*

For computation: use Lagrange polynomial written in Newton form, with divided differences table having each node *xi* written twice.

Consider *z*0 = *x*0*, z*1 = *x*0*, z*2 = *x*1*, z*3 = *x*1*,* . . . , *z*2*m* = *xm, z*2*m*+1 = *xm.*

Form divided differences table: each node appear twice, in the first column write the values of *f* for each node twice; in the second column*,* at the odd positions put the values of the derivatives of *f*; the other elements are computed using the rule from divided differences.

We obtain the following table:

*z*0

*z*1 *z*2

*z*3 ...

*z*2*m z*2*m*+1

| *f*(*z*0) | (*D*1*f*)(*z*0) = *f′*(*x*0) | (*D*2*f*)(*z*0) |  | (*D*2*mf*)(*z*0) | (*D*2*m*+1*f*)(*z*0) |
| --- | --- | --- | --- | --- | --- |
| *f*(*z*1) | (*D*1*f*)(*z*1) | ... |  | (*D*2*mf*)(*z*1) |  |
| *f*(*z*2) | (*D*1*f*)(*z*2) = *f′*(*x*1) |  |  |  |  |
| *f*(*z*3) | ... |  |  |  |  |
| ... | (*D*1*f*)(*z*2*m−*1) | (*D*2*f*)(*z*2*m−*1) | ... |  |  |
| *f*(*z*2*m*) | (*D*1*f*)(*z*2*m*) = *f′*(*xm*) |  | . . . |  |  |
| *f*(*z*2*m*+1) |  |  | . . . |  |  |

Newton interpolation polynomial for the nodes *x*0*, ..., xn* is

(*Nnf*)(*x*) = *f*(*x*0) + X*n i*=1

(*x − x*0)*...*(*x − xi−*1)(*Dif*)(*x*0)*,*

and similarly, Hermite interpolation polynomial is

(*H*2*m*+1*f*)(*x*) = *f*(*z*0) +

2*m*X+1 *i*=1

(*x − z*0)*...*(*x − zi−*1)(*Dif*)(*z*0)*,*

where (*Dif*)(*z*0)*, i* = 1*, ...,* 2*m* + 1 are the elements from the first line and columns 2*, ...,* 2*m* + 1.

Example 2 *Consider the double nodes x*0 = *−*1 *and x*1 = 1*, and f*(*−*1) = *−*3*, f′*(*−*1) = 10*, f*(1) = 1*, f′*(1) = 2*. Find the Hermite inter polation polynomial, that approximates the function f, in both forms, using the classical formula and using divided differences.*

Sol. We present here the method with divided differences. We have *m* = 1*, r*0 = *r*1 = 1 *⇒ n* = 3

*z*0 = *−*1

*z*1 = *−*1 *z*2 = 1 *z*3 = 1

| *f*(*−*1) = *−*3 | *f′*(*−*1) = 10 | *f*(1)*−f*(*−*1)  2 *−f′*(*−*1)  *z*2*−z*0 = *−*4 | 0*−*(*−*4)  *z*3*−z*0= 2 |
| --- | --- | --- | --- |
| *f*(*−*1) = *−*3 | *f*(1)*−f*(*−*1)  *z*2*−z*1= 2 | *f′*(1)*−f*(1)*−f*(*−*1)  *z*3*−z*1= 0  2 |  |
| *f* (1) = 1 | *f′*(1) = 2 |  |  |
| *f*(1) = 1 |  |  |  |

The Hermite interpolation polynomial is

(*H*3*f*)(*x*) = *f*(*z*0) + X3 *i*=1

(*x − z*0)*...*(*x − zi−*1)(*Dif*)(*z*0)

= *f*(*z*0) + (*x − z*0)(*D*1*f*)(*z*0) + (*x − z*0)(*x − z*1)(*D*2*f*)(*z*0) + (*x − z*0)(*x − z*1)(*x − z*2)(*D*3*f*)(*z*0)

i.e.,

(*H*3*f*)(*x*) =*f*(*−*1) + (*x* + 1)*f′*(*−*1) + (*x* + 1)2*f*(1)*−f*(*−*1)*−*2*f′*(*−*1)

~~4~~

+ (*x* + 1)2(*x −* 1)2*f′*(1)*−f*(1)+*f*(*−*1) ~~4~~

= *−* 3 + 10(*x* + 1) *−* 4(*x* + 1)2 + 2(*x* + 1)2(*x −* 1) =2*x*3 *−* 2*x*2 + 1*.*

2.4. Birkhoff interpolation

Let *xk ∈* [*a, b*]*, k* = 0*,* 1*, ..., m, xi ̸*= *xj*for *i ̸*= *j, rk ∈* N and *Ik ⊂ {*0*,* 1*, ..., rk}, k* = 0*,* 1*, ..., m, f* : [*a, b*] *→* R s.t. *∃f*(*j*)(*xk*)*, k* = 0*, ..., m, j ∈ Ik,* and denote *n* = *|I*0*|* + *...* + *|Im| −* 1*,* where *|Ik|* is the cardinal of the set *Ik.*

The Birkhoff interpolation problem (BIP) consists in determining the polynomial *P* of the smallest degree such that

*P*(*j*)(*xk*) = *f*(*j*)(*xk*)*, k* = 0*, ..., m*; *j ∈ Ik.*

Remark 4 *If Ik* = *{*0*,* 1*, ..., rk}, k* = 0*, ..., m, then (BIP) reduces to a (HIP). Birkhoff interpolation is also called* lacunary Hermite interpola tion.

In order to check if (BIP) has an unique solution, we consider the polynomial *P*(*x*) = *anxn* + *...* + *a*0 and the (*n* + 1) *×* (*n* + 1) linear system

*P*(*j*)(*xk*) = *f*(*j*)(*xk*)*, k* = 0*, ..., m*; *j ∈ Ik,* (1)

having as unknowns the coefficients of the polynomial. If the determi nant of the system (1) is nonzero than (BIP) has an unique solution.

Definition 5 *A solution of (BIP), if exists, is called* Birkhoff inter polation polynomial*, denoted by Bnf.*

Birkhoff interpolation polynomial is given by

*m*

(*Bnf*)(*x*) = X *k*=0

X

*j∈Ik*

*bkj*(*x*)*f*(*j*)(*xk*)*,* (2)

where *bkj*(*x*) denote the Birkhoff fundamental interpolation polynomi als. They fulfill relations:

*b*(*p*)

*kj* (*xν*) = 0*, ν ̸*= *k, p ∈ Iν* (3) *b*(*p*)

*kj* (*xk*) = *δjp, p ∈ Ik,* for *j ∈ Ik* and *ν, k* = 0*,* 1*, ..., m,*

with *δjp* =

(1*, j* = *p* 0*, j ̸*= *p.*

Remark 6 *Because of the gaps of the interpolation conditions, it is hard to find an explicit expression for bkj, k* = 0*, ..., m*; *j ∈ Ik. They are found using relations (3).*

Birkhoff interpolation formula is

*f* = *Bnf* + *Rnf,*

where *Rnf* denotes the remainder term.

Example 7 *Let f ∈ C*2[0*,* 1]*, the nodes x*0 = 0*, x*1 = 1 *and we sup pose that we know f*(0) = 1 *and f′*(1) = 1~~2~~*. Find the corresponding interpolation formula.*

Sol. *We have m* = 1*, I*0 = *{*0*}, I*1 = *{*1*}, so n* = 1 + 1 *−* 1 = 1*. We check if there exists a solution of the problem. Consider P*(*x*) = *a*1*x* + *a*0 *∈* P1 *and the system*

(*P*(0) = *f*(0)

*P′*(1) = *f′*(1) *⇐⇒*

*The determinat of the system is*

(*a*0 = *f*(0) *a*1 = *f′*(1) *.*

0 1 1 0

= *−*1 *̸*= 0*,*

*so the problem has an unique solution.*

*The Birkhoff polynomial is*

(*B*1*f*)(*x*) = *b*00(*x*)*f*(0) + *b*11(*x*)*f′*(1) *∈* P1*.*

*We have b*00(*x*) = *ax* + *b ∈* P1 *and*

(*b*00(*x*0) = 1

*b′*00(*x*1) = 0 *⇐⇒*

(*b*00(0) = 1 *b′*00(1) = 0 *⇔*

(*b* = 1 *a* = 0 *,*

*whence*

*b*00(*x*) = 1*.*

*For b*11(*x*) = *cx* + *d ∈* P1 *we have*

(*d* = 0

*c* = 1

*whence*

(*b*11(*x*0) = 0

*b′*11(*x*1) = 1 *⇐⇒*

(*b*11(0) = 0 *b′*11(1) = 1 *⇔*

*b*11(*x*) = *x.*

*So,*

(*B*1*f*)(*x*) = *f*(0) + *xf′*(1) = 1 + 12*x.*

Example 8 *Considering f′*(0) = 1*, f*(1) = 2 *and f′*(2) = 1*. Find the approximative value of f*(1~~2~~)*.*

*•* spline interpolation: piecewise polynomials that require no specific derivative information, except perhaps at the endpoints of the in terval.

Definition 9 *The piecewise-polynomial approximation that uses cubic spline polynomials between each successive pair of nodes is called* cubic spline interpolation.

(The word “spline” was used to refer to a long flexible strip, generally of metal, that could be used to draw continuous smooth curves by forcing the strip to pass through specified points and tracing along the curve.)

Definition 10 *Let f* : [*a, b*] *→* R *and the nodes a* = *x*0 *< x*1 *< ... < xn* = *b, a* cubic spline interpolant *S for f is the function that satisfies the following conditions:*

(a) *S*(*x*) *is a cubic polynomial, denoted Sj*(*x*) *on the subinterval* [*xj, xj*+1]*,∀j* = 0*,* 1*, ..., n −* 1*, i.e.,*

*S*(*x*) =

 

*S*0(*x*)*, x ∈* [*x*0*, x*1] *S*1(*x*)*, x ∈* [*x*1*, x*2] *...*

*Sn−*1(*x*)*, x ∈* [*xn−*1*, xn*]

(b) *Sj*(*xj*) = *f*(*xj*) *and Sj*(*xj*+1) = *f*(*xj*+1)*, ∀j* = 0*,* 1*, ..., n −* 1; (c) *Sj*(*xj*+1) = *Sj*+1(*xj*+1)*, ∀j* = 0*,* 1*, ..., n −* 2;

(d) *S′j*(*xj*+1) = *S′j*+1(*xj*+1)*, ∀j* = 0*,* 1*, ..., n −* 2;

(e) *S′′j*(*xj*+1) = *S′′j*+1(*xj*+1)*, ∀j* = 0*,* 1*, ..., n −* 2;

(f) *One of the following boundary conditions is satisfied:*

(i) *S′′*(*x*0) = *S′′*(*xn*) = 0 *(⇐⇒ S′′*0(*x*0) = *S′′n−*1(*xn*) = 0 *natural (or free) boundary)* natural spline*;*

(ii) *S′*(*x*0) = *f′*(*x*0) *and S′*(*xn*) = *f′*(*xn*) *(⇐⇒ S′*0(*x*0) = *f′*(*x*0) *and S′n−*1(*xn*) = *f′*(*xn*) *clamped boundary)* clamped spline*;*

(iii) *S*0(*x*) = *S*1(*x*) *and Sn−*2 = *Sn−*1 *(*de Boor spline*).*

Remark 11 *A cubic spline function defined on an interval divided into n subintervals will require determining* 4*n constants.*

We have the following expression of a cubic spline:

*Sj*(*x*) = *aj*+*bj*(*x−xj*)+*cj*(*x−xj*)2+*dj*(*x−xj*)3*, ∀j* = 0*,* 1*, ..., n−*1*.* (4)

Theorem 12 *If f is defined at a* = *x*0 *< x*1 *< ... < xn* = *b, then f has an unique natural spline interpolant S on the nodes x*0*, x*1*, ..., xn*; *that satisfies the natural boundary conditions S′′*(*a*) = 0 *and S′′*(*b*) = 0*.*

Example 17 *Construct* a natural cubic spline *that passes through the points* (1*,* 2)*,* (2*,* 3) *and* (3*,* 5)*.*

Sol. (Sketch of the solution) *We follow Definition 10: Here S*(*x*) *consists of two cubic splines, Sj*(*x*) *on the subinterval* [*xj, xj*+1]*,*

*∀j* = 0*,* 1*, i.e.,*

*S*(*x*) =

*given by* (*4*)*,*

(*S*0(*x*)*, x ∈* [*x*0*, x*1] *S*1(*x*)*, x ∈* [*x*1*, x*2]

*S*0(*x*) = *a*0 + *b*0(*x −* 1) + *c*0(*x −* 1)2 + *d*0(*x −* 1)3*,*

*S*1(*x*) = *a*1 + *b*1(*x −* 2) + *c*1(*x −* 2)2 + *d*1(*x −* 2)3*.*

*There are 8 constants (ai, bi, ci, di, i* = 0*,* 1*) to be determined, which requires 8 conditions, that come from (b),(c),(d),(e),(i).*

 

*s*0(*x*0) = *f*(*x*0) *s*0(*x*1) = *f*(*x*1) *s*1(*x*1) = *f*(*x*1) *s*1(*x*2) = *f*(*x*2) *s′*0(*x*1) = *s′*1(*x*1)

*s′′*0(*x*1) = *s′′*1(*x*1) *s′′*0(*x*0) = 0

*s′′*1(*x*2) = 0

Example 18 *Construct* a clamped spline *S that passes through the points* (1*,* 2)*,* (2*,* 3) *and* (3*,* 5) *and that has S′*(1) = 2 *and S′*(3) = 1*.*

General case. Solution of the least squares problem is

*ϕ*(*x*) = X*n i*=1

*aigi*(*x*)*,*

where *{gi, i* = 1*, ..., n}* is a basis of the space and the coefficients *ai* are obtained solving the normal equations:

X*n*

*i*=1

In the discrete case

*aihgi, gki* = *hf, gki , k* = 1*, ..., n. m*

*hf, gi* =X

*k*=0

and in the continuous case

*w* (*xk*) *f* (*xk*) *g* (*xk*)

*hf, gi* =

where *w* is a weight function.

Z *b*

*aw* (*x*) *f* (*x*) *g* (*x*) *,*

3. Numerical integration of functions

The need: for evaluating definite integrals of functions that has no explicit antiderivatives or whose antiderivatives are not easy to obtain.

Let *f* : [*a, b*] *→* R be an integrable function, *xk, k* = 0*, ..., m,* distinct nodes from [*a, b*]*.*

Definition 4 *A formula of the form*

Z *b*

*m*

*af*(*x*)*dx* =X *k*=0

*Akf*(*xk*) + *R*(*f*)*,*

*is* a numerical integration formula *or* a quadrature formula*.*

*Ak*- the coefficients; *xk−*the nodes; *R*(*f*) - the remainder (the error).

COURSE 7

3.1. Interpolatory quadrature formulas Definition 1 *A quadrature formula*

Z *b*

*m*

*af*(*x*)*dx* =X *k*=0

*Akf*(*xk*) + *R*(*f*)*,*

*is* an interpolatory quadrature formula *if it is obtained by integrating each member of an interpolation formula regarding the function f and the nodes xk.*

Remark 2 *An interpolatory quadrature formula has its degree of ex actness at least the degree of the corresponding interpolation polyno mial.*

Definition 3 *The quadrature formulas with equidistant nodes, xk* = *a* + *kh, h* = *b−a*

*~~m~~, are called* Newton-Cotes formulas.

have that there exist *ξ ∈* (*a, b*) such that Z *b*

*af*(*x*)*dx* =*b − a*

2[*f*(*a*) + *f*(*b*)]

# *ba*

+*f00*(*ξ*) 2

"*x*3

3*−*(*a* + *b*)*x*2

2+ *abx*

We obtain the trapezium’s quadrature formula

Z *b*

*af*(*x*)*dx* =*b − a*

2[*f*(*a*) + *f*(*b*)] + *R*1(*f*)*,* (1)

where the remainder (the error) is

*R*1(*f*) = *−*(*b − a*)3

12*f00*(*ξ*)*, ξ ∈* (*a, b*)*.*

This formula is called the trapezium’s formula because the integral is approximated by the area of a trapezium.

Remark 4 *The error from (1) involves f00, so the rule gives exact result when is applied to function whose second derivative is zero (polynomial of first degree or less). So its degree of exactness is 1.*

Example 5 *Approximate the integral* R 31(2*x* + 1)*dx using the trapez ium’s formula.*

(*Remark.* The result is the exact value of the integral because *f*(*x*) = 2*x* + 1 is a linear function and the degree of exactness of the trapezium’s formula is 1*.*)

For *m* = 2 ((*x*0 = *a, x*1 = *a* + *b−a*

~~2~~*, x*2 = *b, h* = *b−a*

Simpson’s quadrature formula

~~2~~) one obtains the

+ *R*2(*f*)*,* (2)

where

Z *b*

*af*(*x*)*dx* =*b − a* 6

*f*(*a*) + 4*f*

*a* + *b* 2

+ *f*(*b*)

*R*2(*f*) = *−*(*b − a*)5

2880*f*(4)(*ξ*)*, a ≤ ξ ≤ b.* (3)

Example 6 *Approximate the integral* R 31(2*x* + 1)*dx using Simpson’s formula.*

Remark 7 *The error from (2) involves f*(4)*, so the rule gives exact result when is applied to any polynomial of third degree or less. So degree of exactness of Simpson’s formula is 3.*

Remark 8 *A Newton-Cotes quadrature formula has degree of exact*

*ness equal to*

(*m, if m is an odd number m* + 1*, if m is an even number.*

Remark 9 *The coefficients of the Newton-Cotes quadrature formulas have the symmetry property:*

*Ai* = *Am−i, i* = 0*, ..., m.*

For *m* = 3*,* Newton’s formula

Z *b*

*af*(*x*)*dx* =*b − a*

8

with

*f*(*a*) + 3*f*

2*a* + *b* 3

+ 3*f*

*a* + 2*b* 3

+ *f*(*b*)

+ *R*3(*f*)*,*

*R*3(*f*) = *−*(*b − a*)5

648*f*(4)(*ξ*)*.*

Example 10 *Compare the trapezium’s rule and Simpson’s rule ap proximations for*

Z 2

0*x*2*dx.*

Sol. *The exact value is* 2*.*667*; for trapezium rule the value is* 4*, for Simpson’s rule the value is* 2*.*667*. (The approximation from Simpson’s rule is exact because the error involves f*(4)(*x*) = 0*.*)

An efficient way of constructing a practical quadrature formula: re peated application of a simple formula.

Let *xk* = *a* + *kh, k* = 0*, ..., n* with *h* = *b−a*

*~~n~~,* be the nodes of a uniform

grid of [*a, b*]*.* By the additivity property of the integral we have

Z *b*

*af*(*x*)*dx* =X*n k*=1

*Ik,* with *Ik* =

Z *xk*

*xk−*1*f*(*x*)*dx*

Applying a quadrature formula to *Ik,* one obtains the repeated quadra ture formula.

Applying to each integral *Ik*the trapezium’s formula, we get

Z *b*

*af*(*x*)*dx* =X*n k*=1

(*xk − xk−*1 2

*f*(*xk−*1) + *f*(*xk*)*−*(*xk − xk−*1)3 12*f*”(*ξk*)

)

*,*

where *xk−*1 *≤ ξk ≤ xk,* or Z *b*

*af*(*x*)*dx* =*b − a*

2*n*



*f*(*a*) + *f*(*b*) + 2

*n*X*−*1 *k*=1

*f*(*xk*)



 + *Rn*(*f*)*,* (4)

with

*Rn*(*f*) = *−*(*b − a*)3

12*n*3

There exists *ξ ∈* (*a, b*) such that

X*n*

*k*=1

*f*”(*ξk*)*.*

1 *n*

X*n*

*k*=1

*f*”(*ξk*) = *f*”(*ξ*)*.*

So the repeated trapezium’s quadrature formula is



 + *Rn*(*f*)*,* (5)

with

Z *b*

*af*(*x*)*dx* =*b − a* 2*n*



*f*(*a*) + *f*(*b*) + 2

*n*X*−*1 *k*=1

*f*(*xk*)

*Rn*(*f*) = *−*(*b − a*)3

12*n*2*f*”(*ξ*)*, a < ξ < b* (6)

We have

*|Rn*(*f*)*| ≤* (*b − a*)3

12*n*2 *M*2*f,*

where *M*2*f* = max

*a≤x≤b|f*”(*x*)*| .* By

*|Rn*(*f*)*| ≤* (*b − a*)3

12*n*2 *M*2*f,* (7)

it follows that the repeated trapezium quadrature formula allows the approx. of an integral with arbitrary small given error, if *n* is taken sufficiently large. If we want that the absolute error to be smaller than *ε,* we determine the smallest solution *n* of the inequation

(*b − a*)3

12*n*2 *M*2*f < ε, n ∈* N*,*

and using this value in (4), leads to desired approximation.

Similarly, there is obtained the repeated Simpson’s quadrature for mula

Z *b*

*af*(*x*)*dx* = *b−a* ~~6~~*~~n~~*



*f*(*a*) + *f*(*b*) + 4 X*n*

*f*

*k*=1

*xk−*1+*xk* ~~2~~

+ 2

*n*X*−*1 *k*=1

*f*(*xk*)



+*Rn*(*f*) (8)

where

*Rn*(*f*) = *−*(*b − a*)5

2880*n*4*f*(4)(*ξ*)*, a < ξ < b,*

and

*|Rn*(*f*)*| ≤* (*b − a*)5

2880*n*4*M*4*f.*

Example 12 *1. Approximate the integral* R 31(2*x* + 1)*dx with repeated trapezium’s formula for n* = 2*.*

*2. Approximate π*~~4~~ *with repeated trapezium’s formula, considering pre cision ε* = 10*−*2*.*

Sol. 1. *Remark.* The result is the exact value of the integral because *f*(*x*) = 2*x* + 1 is a linear function and the degree of exactness of the trapezium’s formula is 1*.*

2. We have

*π*

4= *arctg*(1) =

Z 1 0

*dx*

1 + *x*2*,*

so *f*(*x*) = 1

1+*x*~~2~~*.* Using (7), we get

*|Rn*(*f*)*| ≤* (1 *−* 0)3

12*n*2 *M*2*f.*

We have

*f0*(*x*) = *−*2*x*

(1 + *x*2)2

*f00*(*x*) = 6*x*2 *−* 2

(1 + *x*2)3

and

*x∈*[0*,*1]*|f00*(*x*)*|* = 2*,*

*M*2*f* = max

so

6*n*2*<* 10*−*2 *⇒ n*2 *>*102

*|Rn*(*f*)*| ≤* 1

6= 16*.*66 *⇒ n* = 5*.*

We have *x*0 = 0*, x*1 = 1~~5~~*, x*2 = 2~~5~~*, x*3 = 3~~5~~*, x*4 = 4~~5~~*, x*5 = 1 (*h* = 1~~5~~)*.* The integral will be

*af*(*x*)*dx* ≈110 *f*(0) + *f*(1) + 2 *f*(15) + *f*(25) + *f*(35) + *f*(45)= 0*.*7837*.*Z *b*

3.5. Quadrature formulas of Gauss type All the previous rules can be written in the form

Z *b*

*m*

*af*(*x*)*dx* =X *k*=1

*Akf*(*xk*) + *Rm*(*f*)*,* (9)

where the coefficients *Ak, k* = 1*, ..., m*, do not depend on the function *f.* We have picked the nodes *xk, k* = 1*, ..., m* equispaced and have then calculated the coefficients *Ak, k* = 1*, ..., m.* This guarantees that the rule is exact for polynomials of degree *≤ m*.

It is possible to make such a rule exact for polynomials of degree *≤* 2*m −* 1, by choosing also the nodes appropriately. This is the basic idea of the gaussian rules.

Let *f* : [*a, b*] *→* R be an integrable function and *w* : [*a, b*] *→* R+ a weight function, integrable on [*a, b*]*.*

Definition 2 *A formula of the following form*

Z *b*

*m*

*aw*(*x*)*f*(*x*)*dx* =X *k*=1

*Akf*(*xk*) + *Rm*(*f*) (10)

*is called* a quadrature formula of Gauss type *or* with maximum degree of exactness *if the coefficients Ak and the nodes xk, k* = 1*, ..., m are determined such that the formula has the maximum degree of exactness.*

Remark 3 *The coefficients and the nodes are determined such that to minimize the error, to produce exact results for the largest class of polynomials.*

*Ak* and *xk, k* = 1*, ..., m* from (10) are 2*m* unknown parameters *⇒* 2*m* equations obtained such that the formula (10) is exact for any polynomial degree at most 2*m −* 1*.*

It is often possible to rewrite the integral R*bag*(*x*)*dx* as R*ba w*(*x*)*f*(*x*)*dx,* where *w*(*x*) is a nonnegative integrable function, and *f*(*x*) = *g*(*x*)

*w*(*x*)is

smooth, or it is possible to consider the simple choice *w*(*x*) = 1.

For the general case*,* consider the elementary polynomials *ek*(*x*) = *xk*; *k* = 0*, ...,* 2*m −* 1 and obtain the system s.t. *Rm*(*ek*) = 0 :

 

P*m*

*k*=1 P*m*

*k*=1 *...*

P*m*

*k*=1

*Ake*0(*xk*) = R*ba w*(*x*)*e*0 (*x*) *dx Ake*1(*xk*) = R*ba w*(*x*)*e*1 (*x*) *dx*

*Ake*2*m−*1(*xk*) = R*ba w*(*x*)*e*2*m−*1 (*x*) *dx*

*⇐⇒* 

with

*A*1 + *A*2 + *...* + *Am* = *µ*0

*A*1*x*1 + *A*2*x*2 + *...* + *Amxm* = *µ*1

*...*

*A*1*x*2*m−*1

1 + *A*2*x*2*m−*1

2 + *...* + *Amx*2*m−*1

*m*= *µ*2*m−*1

Z *b*

(11)

*µk* =

*aw*(*x*)*xkdx.*

As, the system (11) is difficult to solve, there have been found other ways to find the unknown parameters.

If *w*(*x*) = 1 (this case was studied by Gauss), then the nodes are the roots of Legendre orthogonal polynomial

*u*(*x*) = *m*!

(2*m*)! [(*x − a*)*m*(*x − b*)*m*](*m*)

and for finding the coefficients we use the first *m* equations from the system (11).

Consider *m* = 1 and obtain the following Gauss type quadrature for mulaZ *b*

*af*(*x*)*dx* = *A*1*f*(*x*1) + *R*1(*f*)*.*

The system (11) becomes

 

*A*1 =R*badx* = *b − a A*1*x*1 =R*ba xdx* = *b*2*−a*2

~~2~~*.*

The unique solution of this system is *A*1 = *b − a, x*1 = *a*+*b* ~~2~~.

Example 4 *The same result is obtained considering x*1 *the root of the Legendre polynomial of the first degree,*

*u*(*x*) = 12[(*x − a*)(*x − b*)]*0*= *x −a* + *b*

2*.*

The Gauss type quadrature formula with one node is

+ *R*1(*f*)*,*

with

Z *b*

*af*(*x*)*dx* = (*b − a*)*f*

*a* + *b* 2

*R*1(*f*) = (*b − a*)3

24*f00*(*ξ*)*, ξ ∈* [*a, b*]

which is called the rectangle quadrature rule (also called the mid point rule).

The repeated rectangle (midpoint) quadrature formula is

Z *b*

*af*(*x*)*dx* =*b − a n*

X*n*

*i*=1

*f*(*xi*) + *Rn*(*f*)*,*

*Rn*(*f*) = (*b − a*)3

24*n*2*f00*(*ξ*)*, ξ ∈* [*a, b*]

~~2~~*~~n~~, xi* = *x*1 + (*i −* 1)*b−a*

with *x*1 = *a* + *b−a*

We have

*|Rn*(*f*)*| ≤* (*b − a*)3

*~~n~~, i* = 2*, ..., n.*

*x∈*[*a,b*]*|f00*(*x*)*|.*

24*n*2 *M*2*f,* with *M*2*f* = max

Remark 5 *Another* rectangle rule *is the following*: Z *b*

*af*(*x*)*dx* = (*b − a*)*f* (*a*) + *R*(*f*)*,*

*with*

*R*(*f*) = (*b − a*)2

2*f0*(*ξ*)*, ξ ∈* [*a, b*]*.*

Example 6 *Approximate* ln 2 = R 211*~~x~~dx, with ε* = 10*−*2*, using the re peated rectangle (midpoint) method.* Solution. We have

Z *b*

*af*(*x*)*dx* =*b − a n*

X*n*

*i*=1

*f*(*xi*) + *Rn*(*f*)*,*

*Rn*(*f*) = (*b − a*)3

24*n*2*f00*(*ξ*)*, ξ ∈* [*a, b*]*.*

*dx*

*x,*

so *f*(*x*) = 1*~~x~~*and we get

ln 2 =

Z 2 1



2*n*) + X*n*



ln 2 =*b − a n*

*f*(*a* +*b − a*

*i*=2

*f*(*a* +*b − a*

2*n*+ (*i −* 1)*b − a*

*n*)

+(*b − a*)3

24*n*2*f00*(*ξ*)

We have *f*(*x*) = 1*~~x~~, f0*(*x*) = *−* 1*x*~~2~~*, f00*(*x*) = 2*x*~~3~~*,* and *|f00*(*ξ*)*| ≤* 2*,* for *ξ ∈* [1*,* 2] so it follows

*|Rn*(*f*)*| ≤* 1

24*n*22 *<* 10*−*2 *⇒* 12*n*2 *>* 100 *⇒ n* = 3*.*

Therefore, 



ln 2 ≈13

1

1 + 1~~6~~+1

1 + 1~~6~~ + 1~~3~~+1

1 + 1~~6~~ + 2~~3~~

 =13 67+69+611= 0*.*6897

(real value is 0*.*693*...*)

Example 7 *For m* = 2*, Gauss quadrature formula is* Z *b*

*af*(*x*)*dx* = *A*1*f*(*x*1) + *A*2*f*(*x*2) + *R*2(*f*)*.*

*Find A*1*, A*2*, x*1*, x*2*.*

Sol. The corresponding Legendre polynomial is *u*(*x*) = 24!h(*x − a*)2(*x − b*)2i*00*

= *x*2 *−* (*a* + *b*)*x* +16(*a*2 + *b*2 + 4*ab*)*,*

with the roots

2*−*(*b − a*)*√*3

*x*1 =*a* + *b*

6*,*

2+(*b − a*)*√*3

*x*2 =*a* + *b*

6*.*

For finding *A*1 and *A*2 we use the first two equations: (*A*1 + *A*2 = *b − a*

*A*1*x*1 + *A*2*x*2 = (*b*2 *− a*2)*/*2*.*

We get

*A*1 = *A*2 = (*b − a*)*/*2*,*

so the quadrature formula of Gauss type with two nodes is

Z *b*

*af*(*x*)*dx* =*b − a* 2

*f*

*a* + *b*

2*−b − a* 6

*√*3+ *f**a* + *b*

2+*b − a*

6

*√*3+*R*2(*f*)*.*

For the interval [*−*1*,* 1] we get *A*1 = *A*2 = 1 and *x*1 = *−*

*√*3~~3~~*, x*2 =*√*3~~3~~*,*

which gives the fomula

Z 1

*−*1*f*(*x*)*dx ' f*(*−*

*√*3

3) + *f*(

*√*3

3)*.*

This formula has degree of precision 3*,* i.e., it gives exact result for every polynomial of the 3*−*rd degree or less.

Remark 8 *The resulting rules look more complicated than the interpo latory rules. Both nodes and weights for gaussian rules are, in general, irrational numbers. But, on a computer, it usually makes no difference whether one evaluates a function at x* = 3 *or at x* = 1*/√*3*. Once the nodes and weights of such a rule are stored, these rules are as easily used as the trapezium rule or Simpson’s rule. At the same time, these gaussian rules are usually much more accurate when compared with the last ones on the basis of number of function values used.*

Remark 9 *a) The coefficients Ak, k* = 1*, ..., m of a Gauss type formula are positive.*

*b) The coefficients Ak, k* = 1*, ..., m and the roots of the Legendre polynomials can be found in tables, for a* = *−*1*, b* = 1*. For example, for m* = 2 *and m* = 3 :

*m nodes coefficients*

2 0*.*577 1

*−*0*.*577 1

3 0*.*774 0*.*555

0 0*.*888

*−*0*.*774 0*.*555

*c) For different weight functions, tables are available for both the nodes and the coefficients.*

Example 10 *Approximate* R 1*−*1*ex* cos *xdx using a Gauss type quadra ture formula for m* = 3*.*

Sol.

Z 1

*−*1*ex* cos *xdx '* 0*.*55*e*0*.*77 cos 0*.*77 + 0*.*88 cos 0 + 0*.*55*e−*0*.*77 cos(*−*0*.*77) = 1*.*9333904

(Exact value is 1*.*9334214*.*) Absolute error is *<*3*.*2 *·* 10*−*5*.*

The integral R*baf*(*x*)*dx* for an arbitrary interval [*a, b*] could be transfomed in an integral on [*−*1*,* 1] using the change of variable

*t* =2*x − a − b*

*b − a⇔ x* =12[(*b − a*)*t* + *a* + *b*] *.*

The Gauss type quadrature formulas may be applied on the following

way:

Z *b*

*af*(*x*)*dx* =

Z 1

*−*1*f*

(*b − a*)*t* + (*b* + *a*) 2

!(*b − a*)

2*dt.* (13)

Example 11 *Consider the integral* R 31(*x*6*−x*2 sin(2*x*))*dx* = 317*.*3442466*.*

*a) Compare the result obtained for Newton-Cotes type formula with m* = 1 *(trapezium formula) and Gauss-Legendre formula for m* = 2;

*b) Compare the result obtained for Newton-Cotes type formula with m* = 2 *(Simpson formula) and Gauss-Legendre formula for m* = 3*.*

Sol. a) Each formula needs 2 evaluations of the function *f*(*x*) = *x*6 *− x*2 sin(2*x*). We have

Trapezium formula (*m* = 1) : 22[*f*(1) + *f*(3)] = 731*.*6054420; and a Gauss type formula for *m* = 2*,* using (13):

Z 3

1(*x*6 *− x*2sin(2*x*))*dx* =Z 1*−*1((*t* + 2)6 *−* (*t* + 2)2sin(2(*t* + 2)))*dt ' f*(*−*0*.*577 + 2) + *f*(0*.*577 + 2) = 306*.*8199344*.*

b) Each formula needs 3 evaluations of the function. We have F. Simpson (*m* = 2) : 13[*f*(1) + 4*f*(2) + *f*(3)] = 333*.*23; and a Gauss type formula for *m* = 3*,* using (13):

1(*x*6 *− x*2sin(2*x*))*dx* =Z 1*−*1((*t* + 2)6 *−* (*t* + 2)2sin(2(*t* + 2)))*dt*

Z 3

*'* 0*.*55*f*(*−*0*.*77 + 2) + 0*.*88*f*(2) + 0*.*55*f*(0*.*77 + 2)

= 317*.*2641516

3.6. General quadrature formulas

Using the interpolation formulas, there are obtained a large variety of quadrature formulas.

In the case of some concrete applications, the choosing of the quadra ture formula is made according to the information about the function *f.*

A general quadrature formula is given by:

Z *b*

*m*

*af*(*x*)*dx* =X *k*=0

X

*j∈Ik*

*Akjf*(*j*)(*xk*) + *R*(*f*)*.*

For example, consider the Hermite interpolation formula for *f* : [*a, b*] *→* R, the nodes *xk ∈* [*a, b*] *, k* = 0*, ..., m* multiple of orders *r*0*, ..., rm ∈* N,

*f* = *Hnf* + *Rnf,* (14)