



Surface Description Using Bicubic B-splines Curvature Based Smoothing of Plane Cubic B-spline Curves

Hvid, Søren Lovmand; Larsen, Poul Scheel; Sørensen, Jens Nørkær

Publication date:
1993

Document Version
Publisher's PDF, also known as Version of record

[Link back to DTU Orbit](#)

Citation (APA):
Hvid, S. L., Larsen, P. S., & Sørensen, J. N. (1993). *Surface Description Using Bicubic B-splines: Curvature Based Smoothing of Plane Cubic B-spline Curves*. Technical University of Denmark. AFM No. 93-08, 93-10

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

AFM - 94-02

ISSN 0590-8809

Januar 1994

QM3D

3D Netgenerering

Brugermanual

Søren Lovmand Hvid



**AFDELINGEN
FOR
FLUID MEKANIK**

**DEN POLYTEKNISKE LÆREANSTALT
LYNGBY**

Forord

Denne rapport er affattet som en del af en afhandling til opnåelse af Ph.D. graden ved Danmarks Tekniske Højskole. Rapporten beskriver arbejde, som er udført ved Afdelingen for Fluid Mekanik, Danmarks Tekniske Højskole i perioden september 1990 til december 1993. Arbejdet er udført under vejledning af professor, Ph.D. P. Scheel Larsen, lektor, lic.techn. J. Nørkær Sørensen samt civ.ing., lic.techn. J. A. Michelsen.

Jeg vil gerne takke mine vejledere for deres støtte i gennemførelsen af dette projekt og ved udfærdigelsen af nærværende rapport.

Abstrakt

Det i denne manual beskrevne program er en netgenerator til fremstilling af tredimensionelle beregningsnet til brug for Finite-Volume og/eller Finite-difference baserede beregningskoder indenfor fluid mekanikken. Beregningsnettene kan dog anvendes helt generelt ved numerisk løsning af differential-ligninger. Programmets navn er QM3D, som står for *Quick Mesh 3D*.

Programmet er interaktivt, dvs. at en bruger ved en programkørsel kommer med input til programmet, og får en respons derpå. Brugerfladen er ret simpel, idet den består af en kommandolinie, hvorpå brugeren kan indtaste kommandoer og input. Programmet giver brugeren et grafisk respons på brugerens input i form af billeder af beregningsnettet.

Det anvendte programmeringssprog er FORTRAN 77, og til de grafiske præsentationer benyttes X-WINDOWS (Release 11). X-WINDOWS grafikrutinerne er udviklet på Afdelingen for Fluid Mekanik af Civ.Ing., Lic.Techn. Erik Andresen. QM3D-programmet er udviklet på og til UNIX arbejdsstationer og kører pt. på følgende typer:

- Hewlett Packard serie 9000 7XX
- Silicon Graphics Personal Iris 4DXX
- IBM serie 6000

Der er på ovennævnte maskiner enkelte forskelle, som skyldes variationer i systemafhængige udvidelser til FORTRAN 77 standarden og forskelle i de enkelte udgaver af UNIX styresystemet. Programmet er afprøvet på ovennævnte maskiner, men burde med en relativt beskedne indsats kunne bringes til at virke på UNIX arbejdsstationer fra andre producenter.

Indholdsfortegnelse

1 Brugervejledning til QM3D	2
1.1 Generel beskrivelse	2
1.1.1 Opbygning	3
1.1.2 Brugerfladen	4
1.1.3 Opsætning til en kørsel	6
1.2 Kommandobeskrivelser	6
2 Programdokumentation	9
2.1 Strækning og ortogonalitet	9
2.1.1 Strækningsfunktioner	9
2.1.2 Ortogonalitetsparametre	12
2.2 Filbeskrivelser	13
2.2.1 Inputfiler	13
2.2.2 Outputfiler	15
2.2.3 Interne filer	15
2.3 Flowcharts	17
2.4 Begrænsninger	20
Litteraturfortegnelse	28

Kapitel 1

Brugervejledning til QM3D

1.1 Generel beskrivelse

QM3D¹ er et netgenereringsprogram beregnet til at fremstille strukturerede beregningsnet i tre dimensioner, til brug for Finite Volume og/eller Finite Difference baserede beregningskoder eller andre strukturerede koder². Programmet er interaktivt, hvilket betyder at brugeren justerer på nettet med et løbende grafisk feedback på skærmen. Selve interaktionen foregår ved at brugeren afgiver kommandoer på en kommandolinie.

Netgenereringen sker efter en algebraisk metode³, hvilket dels betyder at fremstillingen af nettet sker forholdsvis hurtigt (deraf navnet på programmet) ved at interpolere mellem randdata og dels at brugeren i høj grad er ansvarlig for resultatet - dvs. netkvaliteten. Til at styre fremstillingen af nettet har brugeren en række værktøjer til rådighed, og disse beskrives i afsnit 1.2.

QM3D er udviklet til at fremstille enkeltblok-net, således at de geometriske variable x, y og z indiceres med tre indices i, j og k . Selve fremstillingen af nettet sker dog (oftest) i blokke - herefter benævnt *delblokke*⁴ - dels for at lette styringen af nettet og dels for at lette det grafiske feedback. Brugeren bestemmer hovedsageligt selv hvordan opdelingen i delblokke skal være. Brugeren bestemmer ligeledes selv nettets topologi, hvorved forstås det tredimensionelle domænes geometri - specielt singulariteter, specialpunkter mv. herunder nettyper (f.eks. O, H og/eller C net). Denne topologi fastlægges i en geometrifil, hvor delblokkene defineres og deres hjørner specificeres. I denne fil gives også koordinater for tre-dimensionelle overflader, f.eks. frembragt af programmet *Surf*, se [5]. Genereringen af volumennettet sker ved interpolation udfra delblokkens flader, og nettet på fladerne findes ligeledes ved interpolation mellem fladens rande. Geometrien og punkt-

¹Quick Mesh 3D

²Med *strukturerede koder* menes koder der baserer sig på implicit indicering, dvs. hvor nabopunkter er givet implicit som følge af arraystrukturen. Strukturerede net er (normalt) firkanter i 2D og "kasser" i 3D

³En implementering af *Transfinite Interpolation*

⁴Se side 3 for en nærmere beskrivelse af begrebet *delblok*

fordelingen på delblokkenes tolv kanter er derfor afgørende.

1.1.1 Opbygning

QM3D genererer enkelt-blok net, men er konstrueret således at fremstillingen af nettet sker i delblokke. At nettet er et enkeltblok net betyder at koordinaterne har tre indices i, j, k , hvor flerblok net normalt har et fjerde index til at angive bloknr. Fremstillingen af nettet i delblokke betyder at man er i stand til bedre at styre nettet, idet genereringen sker i én delblok ad gangen. Til hver delblok kan man da angive særskilte værdier for de forskellige parametre, der bruges til at styre nettet.

I en flerblok ("multiblock") netgenerator, er orientering og indicering i en blok normalt uafhængig af de andre. Begrebet *delblok* skyldes, at QM3D's ene blok underopdeles i flere blokke - ikke for at øge den geometriske fleksibilitet, men for at forbedre kontrollen over netgenereringen indenfor blokken. En delblok yderflade skal passe identisk sammen med en anden delblok yderflade, således at en delblok højst har fladesammenfald med seks andre delblokke (indre delblokke).

Brugeren interagerer med programmet via en kommandofortolker, som derved bliver den centrale del af programmet. Herfra foretages de relevante kald af underrutiner som respons på kommandoer fra brugeren. Overførsel af information mellem de forskellige dele af programmet sker i det væsentligste via common blokke, idet der til programmet hører én include fil QM3D.INC, som inkluderes i (næsten) alle rutiner. I denne include fil er så godt som alle de variable, som programmet bruger, erklæret og placeret i common blokke. Valget af denne metode til argumentoverførsel har følgende fordele:

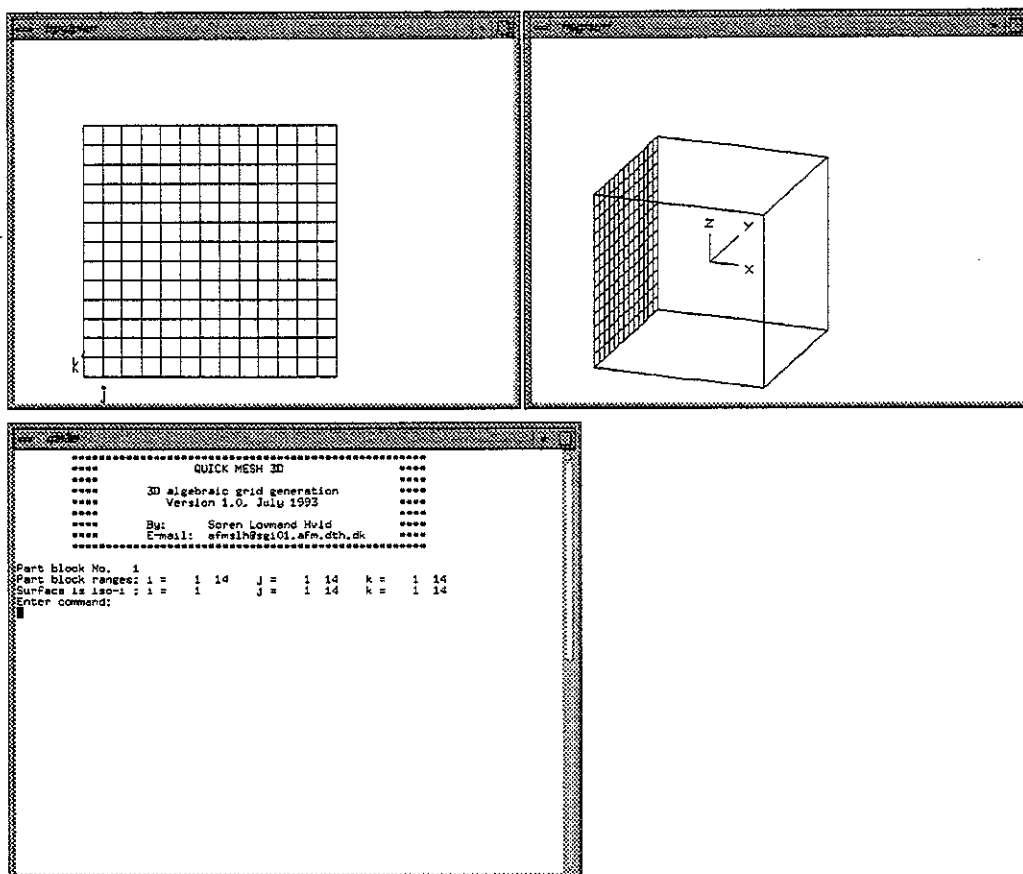
- Det eliminerer fejl ved argumentoverførsel
- Det letter implementering af ændringer i globale parametre
- Det gør programmet lettere læseligt, idet identiske størrelser har samme navn i alle rutiner

Til gengæld har metoden også en række ulemper:

- Enhver ændring af en variabel i én rutine påvirker hele programmet
- De enkelte underrutiner kan ikke uden videre bruges af andre programmer
- Variabelnavne som er logiske i én underrutine forekommer ulogiske i deres brug i en anden underrutine.

Ændringer i variable foretages på én af to måder:

- Ved at ændre en parametersætning i QM3D.INC for programspecifikke globale variable



Figur 1.1: Skærmens udseende under programudførslen

- Ved at ændre variabelværdien via kommandofortolkeren for brugerjusterbare variable

Ud over at ændre i QM3D.INC kan brugeren være nødt til at rette i en eller flere af følgende filer:

special.f Heri specificeres en specialbehandling af dele af nettet. Bruges typisk til at implementere singulariteter eller til at fastholde f.eks. randdata.

qm3d.dat Denne fil angiver hvilke(n) geometrifil(er) der skal indlæses

1.1.2 Brugersiden

Ved kald af QM3D startes programmet op i et separat vindue, som nedlægges igen ved afslutningen af kørslen. Brugeren interagerer med QM3D via en kommandofortolker, som kører i dette vindue. Til at give brugeren et feedback starter QM3D to vinduer, der grafisk viser nettet i den aktuelle delblok. I figur 1.1 ses et billede af skærmen under en kørsel med QM3D. Kommandovinduet er placeret nederst og de to grafikvinduer øverst.

I det ene vindue vises hele delblokken i en ortogonal projektion (dimetri), således at delblokkens rande og den yderflade på delblokken der er under behandling vises. Delblokken i dette vindue kan roteres om en lodret akse så man kan få en bedre synsvinkel. I dette vindue ser man altså det "sande" net på en given flade.

I det andet vindue vises den blokflade der er under behandling i en plan projektion. Fladen projiceres på det plan der går igennem fladens fire hjørnepunkter⁵. Denne plane projektion benyttes, da den giver et meget let forståeligt grafisk billede af nettet til brugeren, men er naturligvis farlig derved, at misinformation kan finde sted som følge af projektionen.

Endelig har brugeren den mulighed at se vilkårlige udsnit af nettet på tværs af delblokke og ikke begrænset til disses yderflader. Disse udsnit fås ved brug af kommandoen "Drawgrid" som gennemgås i afsnit 1.2. Udsnitene vises i plan projektion, hvor projektionen foretages som nævnt ovenfor.

Ved hjælp af en række kommandoer har brugeren mulighed for at tilpasse nettet efter behov. Der er mulighed for at foretage følgende indgreb:

- Flytning af de 8 hjørnepunkter i delblokken
- Strækning af nettet på delblokkens 12 kanter samt ændring af kantes geometri⁶
- ændring af ortogonalitetsfaktorer for såvel fladeinterpolation som rumlig interpolation

For at brugeren kan se effekten af sine indgreb, er det en fordel at se den flade på delblokken, der er under behandling. Brugeren fortæller selv, hvilken blokflade han vil se. Endelig kan brugeren iværksætte volumennetgenerering af delblokken, og programmet melder tilbage hvis der i volumennettet er negative cellevoluminer (krydsende netlinier). Er dette tilfældet, genereres en datafil "metric.dat" som lægges i det directory, hvorfra QM3D blev kaldt. I denne fil lagres resultatet af volumenberegningen med oplysning om indices (kun for de negative voluminer). Brugeren kan så ændre på fladenettene og ortogonalitetsfaktorerne indtil netoverlap undgås. Beregning af voluminer er den eneste netkontrol der udføres af programmet; brugeren må selv vurdere om nettet er af tilfredsstillende kvalitet.

⁵De fire hjørnepunkter vil ikke altid tilhøre det samme plan. Planet findes som det plan, der har en normalvektor, som er parallel med krydsvektoren til de to diagonalvektorer.

⁶QM3D antager som begyndelsesbetingelse rette linier mellem delblokkens hjørnepunkter

1.1.3 Opsætning til en kørsel

Skal man igang med en netgenereringsopgave vil arbejdsgangen normalt være følgende:

1. Fastlæg geometrien og topologien af beregningsdomænet.
2. Generer et overfladenet på alle ikke-trivielle overflader. Det kan være såvel beregningsdomænets yderflader som indre geometrier. Overfladenettene kan f.eks. genereres v.hj.a. programmet *Surf*, se [5].
3. De fremstillede overfladenet placeres i en *geometrifil* med et ganske bestemt format. I denne fil defineres tillige domænets topologi. Formatet er beskrevet i afsnit 2.2.1 og et eksempel vist i appendix side 26.
4. Editer filen *qm3d.dat*. Filen skal indeholde antal blokke i beregningsnettet (QM3D er i denne implementering begrænset til enkeltblok net, dvs. antal blokke er "1"), samt navnet på den fremstillede geometrifil.
5. Includefilen *qm3d.inc* editeres. Her skal opløsningen *ni, nj, nk* angives til de ønskede værdier.
6. Hvis en speciel behandling af visse dele af nettet skal foretages, programmeres dette direkte i filen *special.f*. Denne fil kan typisk anvendes til at generere singulariteter, til at sikre numerisk identitet mellem identiske punkter med forskellig lagerplads etc.
7. Programmet oversættes med kommandoen *make qm3d*, og kan herefter afvikles med kommandoen *qm3d*.

1.2 Kommandobeskrivelser

De følgende kommandobeskrivelser indeholder kommandoens fulde navn og mindste tilladelige forkortelse (understreget), samt evt. argumenter der skal gives til kommandoen. Derefter følger en kort beskrivelse af kommandoens formål samt af argumenternes betydning. Bortset fra kommandoerne *Quit* og *quit* er det uden betydning, om der anvendes store eller små bogstaver.

Angle α Sætter synsvinklen i vinduet hvor delblokken er vist i dimetrisk projektion. Rotation sker om en lodret akse, og vinklen angives i grader. Startværdi for α er 0° hver gang der skiftes delblok.

Corner $i j k$ Refererer til hjørnet med indices i, j, k (som skal tilhøre den aktuelle delblok). Koordinaterne for hjørnet skrives til skærmen og brugeren kan angive en flytning af hjørnet enten relativt eller absolut v.hj.a. kommandoerne:

Absolut $x y z$ Her flyttes hjørnet til de nye koordinater x, y, z .

Relativ $x y z$ Her flyttes hjørnet afstandende x, y, z relativt i forhold til den nuværende position.

Drawgrid Denne kommando starter en interaktiv session, hvor brugeren kan få vist en vilkårlig del af nettet i plan projektion, ved brug af *hpglview*.⁷ Det er en forudsætning den del af nettet man vil se også er blevet genereret (hvis man f.eks. vil se et udsnit af volumennettet i den aktuelle delblok genereres dette først med kommandoen *grid vol* (eller blot "*g v*"), se denne kommando). I *drawgrid* bedes brugeren om at specificere indices for den pågældende del af nettet.

Edge $ie m1 m2$ Ændring af en kant i en den aktuelle delblok. Argumentet ie beskriver om det er en i, j eller k linie (for argumentværdierne henh. 1, 2 og 3), og de to øvrige argumenter giver liniens indices i de to andre variable. Hér angives argumenterne altid "i rækkefølge", dvs. for en j -linie angives de faste i og k indices i den rækkefølge etc. Ved hjælp af kommandoen *edge* kan der ændres på følgende parametre:

- Netstrækning v.h.j.a. parametrene p og w
- Kantens geometri v.h.j.a. parameteren fac samt tangentvektoren i kantliniens endepunkter $bn1$ og $bn2$.

Strækningsparametrene beskrives nærmere i afsnit 2.1. De to tangentvektorer $bn1$ og $bn2$ må brugeren give explicit. Længden af vektorerne har betydning for deres indflydelse på kurvens geometri - lille længde, lille indflydelse og vice versa. Parameteren fac bruges til at skalere tangentvektorerne, og er som sådan overflødig, idet man blot kan indtaste større eller mindre værdier for disse. Det er dog bekvemt at kunne ændre skaleringen v.h.j.a. vægten fac , der iøvrigt skalerer begge vektorer lige meget. Initialværdier af parametrene er:

p	w	fac	$bn1$	$bn2$
0.001	2.0	1.0	(0, 0, 0)	(0, 0, 0)

Grid C Genererer et net på en delblok flade eller i delblokkens volumen, afhængig af argumentet C , som er en streng. Argumentet C kan være **surface** eller **volume**, hvor kun den understregede del af argumentet er nødvendig. Ved argumentet s genereres overfladenettet på den aktuelle flade i delblokken, og ved argumentet v genereres et volumennet udfra delblokkens 6 flader.

⁷*hpglview* er et X-windows baseret grafikprogram udviklet på Afdelingen for Fluid Mekanik.

Ortho *C* Ændring af ortogonalitetsfaktorer for enten den aktuelle delblok flade eller for delblokkens volumen. Ortogonalitetsfaktoren bruges ved generering af flade- og/eller volumennet. Argumentet *C* er, som beskrevet for kommandoen *grid*, enten *s* eller *v*. Se afsnit (2.1.2).

quit Afslutter behandlingen af den aktuelle delblok. Programmet fortsætter derefter selv med den næste delblok, eller spørger brugeren hvilken delblok der skal behandles, afhængig af hvor langt i netgenereringsprocessen man er. Bemærk at kommandoen gives specifikt med et lille "q".

Quit Afslutter QM3D. Efter afgivelse af denne kommando genererer QM3D volumennet i alle delblokke ud fra delblokfladerne, og melder tilbage om negative volumener. Derefter spørges om brugeren ønsker at gemme nettet i en QM3D-fil, som kan være enten en ASCII-fil eller en binær fil, og om brugeren ønsker at gemme en *drawgrid*-fil til brug for programmet *drawgrid*, der virker på nøjagtig samme måde som brug af *drawgrid* kommandoen i QM3D. Filer beskrives nærmere i afsnit 2.2. Bemærk at kommandoen gives med et stort "Q".

Ranges Oplyser om indexgrænserne for den aktuelle delblok. Disse grænser bruges ved kald af visse kommandoer, som f.eks. *edge*.

Surface *ind isurf* Angiver hvilken flade af delblokken der skal være den aktuelle. Argumenterne er af integer type. Første argument *ind* angiver hvilken flade af delblokken brugeren ønsker at se:

ind = 1 Angiver en iso-*i* flade på delblokken

ind = 2 Angiver en iso-*j* flade på delblokken

ind = 3 Angiver en iso-*k* flade på delblokken

Andet argument *isurf* angiver hvilken af de to mulige yderflader af delblokken der refereres til. Værdien af argumentet er delblokfladens index som returneret af kommandoen *ranges*.

Kapitel 2

Programdokumentation

2.1 Strækning og ortogonalitet

2.1.1 Strækningsfunktioner

Strækning af nettet har til formål at koncentrere netpunkter i områder, hvor der er store gradienter, eller hvor en høj opløsning af domænet af andre årsager er ønskelig. Til strækning af nettet bruges strækningsfunktioner. For disse funktioner skal gælde, at de skal være monotone, da der i modsat fald vil forekomme netoverlap. Strækningsfunktionerne foreskriver en afbildning fra den ækvidistante inddeling i computeren til den ønskede variation i det fysiske domæne. Argumentet ξ til funktionen vil normalt være normeret til intervallet $[0, 1]$, generelt

$$\xi = \frac{i - i_1}{i_2 - i_1} ; i \in [i_1, i_2] \quad (2.1)$$

Ud over at tilvejebringe en ønsket koncentration af punkter vælges strækningsfunktionen med henblik på at give en så glat variation i punktafstanden som mulig - set ud fra et numerisk synspunkt. Det er vist af Vinokur [9] at den hyperbolske tangens giver det glatteste resultat for en given strækning, og Thompson [1] opsummerer følgende for valget af strækningsfunktioner:

- Hyperbolske funktioner er bedre end rene exponentialfunktioner
- Exponentialfunktioner er bedre end polynomier

Det er dog vanskeligt at formulere den hyperbolske tangens så den bliver fleksibel og nem i brug, og på denne baggrund er det i QM3D valgt at basere strækning på følgende simple exponentialfunktion:

$$f(\xi) = \frac{\exp^{p\xi} - 1}{\exp^p - 1} , p \neq 0 , \quad (2.2)$$

hvor værdien af parameteren p bestemmer strækningen - større p større strækning og vice versa. For $p \rightarrow 0$ fås uniform (eller ingen) strækning.

Fortegnet for p bestemmer om der strækkes mod det ene eller andet endepunkt. I figur 2.1 ses resultatet af en strækning med funktionen f givet ved ligning (2.2) med p sat til henh. $+3$ og -3 . Lille hældning på kurven svarer til høj koncentration af netpunkter og omvendt.

For at øge fleksibiliteten af strækningsfunktionen (2.2), er den i QM3D udbygget med en ekstra parameter w , som anvendes ved tosidig strækning. En sådan strækning vil ofte være ønskelig - bl.a. ved interne strømninger. Parameteren w angiver funktionsværdien for $\xi = 0.5$, altså

$$f(0.5) = w .$$

Beregningen af den tosidige strækningsfunktion bliver:

$$f(\xi) = \begin{cases} w \frac{\exp^{2p\xi} - 1}{\exp^p - 1} & \text{for } \xi \leq 0.5 \\ 1 - (1 - w) \frac{\exp^{2p(1-\xi)} - 1}{\exp^p - 1} & \text{for } \xi \geq 0.5 \end{cases} \quad (2.3)$$

Desværre bliver $f'(0.5)$ diskontinuert for $w \neq 0.5$ for denne funktion, og for at råde bod på dette forhold anvendes to andengradspolynomier i intervallerne $\xi \in [0.45, 0.5]$ og $\xi \in]0.5, 0.55]$ til at sammensætte de to exponentialfunktioner således at den første afledede overalt er kontinuert. Disse to andengradspolynomier skrives:

$$g_1(\xi) = a_1\xi^2 + b_1\xi + c_1 \quad (2.4)$$

$$g_2(\xi) = a_2\xi^2 + b_2\xi + c_2 \quad (2.5)$$

og de skal opfylde betingelserne:

$$\begin{aligned} g_1(0.45) &= f(0.45) & g_2(0.55) &= f(0.55) \\ g_1'(0.45) &= f'(0.45) & g_2'(0.55) &= f'(0.55) \\ g_1(0.5) &= g_2(0.5) = w^* & g_1'(0.5) &= g_2'(0.5) \end{aligned} \quad (2.6)$$

hvor w^* bliver den nye funktionsværdi for $\xi = 0.5$ som vil være en anelse forskellig fra det givne argument w . Systemet (2.6) må løses for hvert nyt valg af p og/eller w . Den endelige beregning af strækningsfunktionen ser nu således ud:

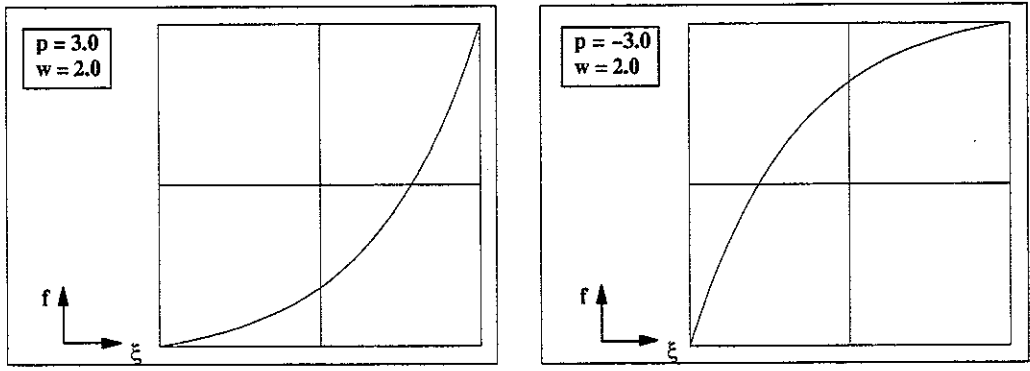
For $w \ni [0, 1]$:

$$f(\xi) = \frac{\exp^{p\xi} - 1}{\exp^p - 1} \quad (2.7)$$

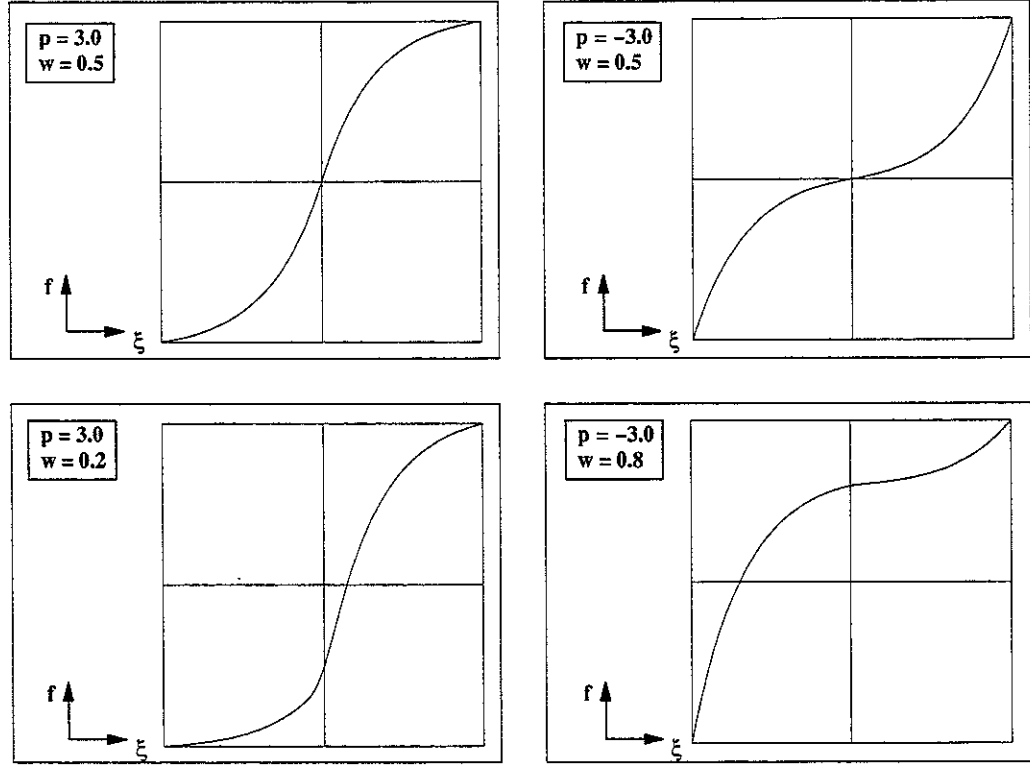
og for $w \in [0, 1]$:

$$f(\xi) = \begin{cases} w \frac{\exp^{2p\xi} - 1}{\exp^p - 1} & \text{for } \xi \in [0, 0.45] \\ a_1\xi^2 + b_1\xi + c_1 & \text{for } \xi \in]0.45, 0.5] \\ a_2\xi^2 + b_2\xi + c_2 & \text{for } \xi \in]0.5, 0.55] \\ 1 - (1 - w) \frac{\exp^{2p(1-\xi)} - 1}{\exp^p - 1} & \text{for } \xi \in]0.55, 1] \end{cases} \quad (2.8)$$

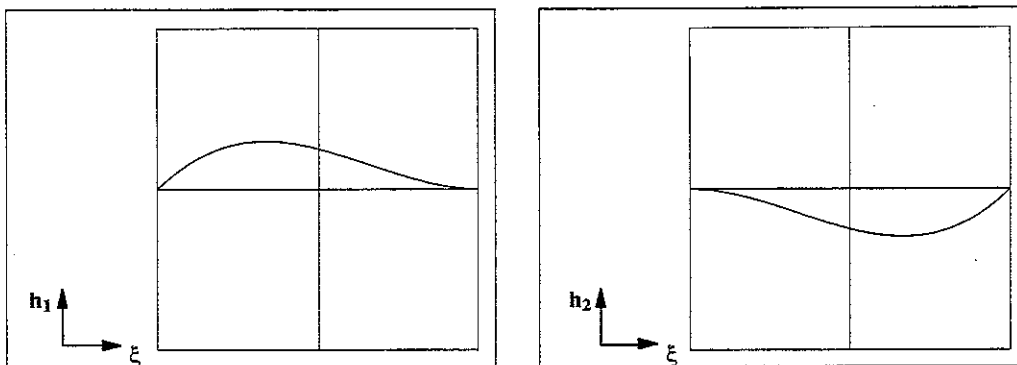
I figur 2.1 og figur 2.2 ses resultatet af beregning af f for forskellige valg af p og w .



Figur 2.1: Strækningssfunktioner



Figur 2.2: To-sidige Strækningssfunktioner



Figur 2.3: Blandingsfunktioner for normalvektorer

2.1.2 Ortogonalitetsparametre

QM3D anvender orthogonalitetsparametre til at styre orthogonaliteten ved delblokkens rand, både for fladenettet på delblokkens overflader og for volumennettet. Initialværdien for alle orthogonalitetsparametrene er nul (0), svarende til at der ikke påtvinges yderligere orthogonalitet ved randene. Årsagen til dette er dels, at selve interpoleringen af nettet foretages sådan at en rimelig orthogonalitet opnås ved randene uden hjælp fra orthogonalitetsfaktorerne, og dels at specificering af (ekstra) orthogonalitet v. hj. af disse faktorer ændrer på netstrækningen - specielt ved randen, hvor indflydelsen fra orthogonalitetsfaktorerne er størst, se [6].

Ortogonalitetsstyring med orthogonalitetsparametrene virker ved at normalvektorerne på randen medtages i interpolation af nettet, og parametrenes størrelse bestemmer graden af orthogonalitet, svarende til en skalering af normalvektorerne. Heri ligger også årsagen til ændringen af netstrækningen, idet bidraget fra normalvektorerne adderes til den øvrige beregning. Dette betyder også, at man risikerer netoverlap ved for høj skalering. For at opnå, at indvirkningen af normalvektorer begrænses til den lokale rand benyttes et sæt blandingsfunktioner. Disse funktioner antager værdien nul i hver ende, da man ellers ikke vil interpolere eksakt til randene. Endvidere er gradienten af blandingsfunktionen 1 ved den rand hvor normalen er beregnet, og nul ved den anden. Valget af blandingsfunktioner er faldet på tredjegradspolynomierne:

$$h_1(\xi) = \xi^3 - 2\xi^2 + \xi \quad (2.9)$$

$$h_2(\xi) = \xi^3 - \xi^2 \quad (2.10)$$

Disse to blandingsfunktioner er afbildet i figur 2.3.

2.2 Filbeskrivelser

Input til QM3D sker dels interaktivt v.h.j.a. kommandofortolkeren og dels via filer, hvorigennem nogle faste parametre samt geometri- og topologidata kommunikerer. Input via filer sker i underrutinerne INPUTPAR, GEOMETRY samt READQM3D. Endelig sker input ved at ændre i include-filen QM3D.INC, hvor bl.a. netopløsningen ($n_i \times n_j \times n_k$) fastsættes.

Output fra QM3D foregår ligeledes interaktivt via kommandofortolkeren, dvs. til skærmen, samt til filer. Output til filer sker i underrutinerne WRITQM3D og METRIC.

Internt bruger QM3D en række filer til grafikpræsentation samt til systemarbejde.

I det følgende gennemgås alle anvendte filer med en beskrivelse af anvendelsen og filformatet.

2.2.1 Inputfiler

QM3D.INC Her angives globale parametre og impliciterklæringer. Kun de beskrevne parametre bør ændres: n_i , n_j og n_k fastsættes til den ønskede opløsning af nettet, parametrene *maxblock*, der angiver det maximale antal blokke i nettet¹, samt *maxpart*, der angiver det maksimale antal delblokke pr. blok, fastsættes. Parameteren "eps" bør ikke ændres fra standardværdien 10^{-4} . IMPLICIT-erklæringen bruges til at angive, om der ønskes anvendt enkelt- eller dobbeltpræcision i flydende taloperationer. Da netgeneratoren er algebraisk, vil det normalt ikke være nødvendigt at anvende dobbeltpræcision.

Når QM3D.INC er ændret, skal programmet recompileres med kommandoen *make qm3d*. Denne strategi er valgt, idet tidsforbruget til kompilering normalt vil være forsvindende i forhold til den tid, der medgår til selve netgenereringen. En arbitrær (høj) indexgrænse på parametrene undgås herved.

Et eksempel på filen QM3D.INC er vist i appendix side 22.

QM3D.DAT Denne fil skal altid forefindes i det directory, hvorfra QM3D kaldes. Filen indeholder brugeroplysninger til programmet og indlæses i underrutinen INPUTPAR. Første linie i QM3D.DAT indeholder integer konstanten *nblock*, der angiver antallet af blokke ($nblock \leq maxblock$). På de efterfølgende *nblock* linier angives filnavne (én fil pr. block) på geometri/topologi filer. Brugeren editerer selv QM3D.DAT og specificerer *nblock* samt anfører filnavne (evt. incl. stier).

Hvis sessionen er en viderebearbejdning af en påbegyndt opgave, og der er lavet en QM3D-fil (enten ASCII eller binær) skal denne fil angives i QM3D.DAT. I så fald anføres kun denne ene fil uanset værdien

¹Netgeneratoren er i sin nuværende form kun *forberedt* til multiblok netgenerering, hvorfor *maxblock* normalt blot kan sættes til 1.

af *nblock*. Brugeren skal selv sørge for at angive den korrekte fil. QM3D-filer har extension ".qm3d.asc" for ASCII-filer og ".qm3d.bin" for binære filer.

Et eksempel på filen QM3D.DAT er vist i appendix side 25.

GEOMETRY I underrutinen GEOMETRY indlæses geometri/topologi filer angivet af brugeren i QM3D.DAT (dog ikke QM3D-filer der læses i underrutinen READQM3D). Disse filer har et ganske bestemt format, som beskrives herunder. Et eksempel på en geometri/topologi fil ses i appendix side 26. Formatet er:

Linie 1-3 Kommentarløner. Kan anvendes som header med information.

Linie 4 *nsurf* : Integer, der angiver antal definerede overflader i denne fil ($nsurf \geq 0$).

Linie 5 Kommentarløner (kan benyttes af brugeren).

Herefter indlæses *nsurf* definerede overflader. Disse overflader skal være definerede som et rektangulært net af punkter $nrow \times ncol$, hvert punkt angivet ved dets Kartesiske (x, y, z) -værdi. Indlæsning af hver overflade foregår således:

Linie 1 Kommentarløner (linie 6 i eksemplet).

Linie 2 *surf ind*, *surf con* : Integer konstanter. *surf ind* angiver om der er tale om en konstant-*i*, konstant-*j* eller konstant-*k* overflade (for *nsurf* = henholdsvis 1, 2 eller 3). *surf con* er værdien af den konstante koordinat.

Linie 3 *i1*, *i2*, *j1*, *j2*: Integer konstanter der angiver indeksvariationerne for de ikke-konstante koordinater, således at $i2 - i1 + 1 = nrow$ og $j2 - j1 + 1 = ncol$. Hvis $i1 \geq i2$ henh. $j1 \geq j2$ indlæses bagfra for den pågældende variable. Dette gør indlæsningen mere fleksibel.

Herefter følger $nrow \times ncol$ linier med (x, y, z) -værdier for overfladen, efterfulgt af en tom linie. Denne løkke gentages for *nsurf* definerede overflader.

Efter geometriindlæsningen følger topologidefinitionen, hvor antallet af delblokke og disses indeksgrænser specificeres. Formatet er:

Linie 1 Kommentarløner.

Linie 2 *npart*: Integer der angiver antal delblokke i en blok.

Herefter følger en løkke der indlæser grænser for de *npart* delblokke i rækkefølge:

Linie 1 Kommentarløner.

Linie 2 i_1, i_2 : Indeksgrænser for i .

Linie 3 j_1, j_2 : Indeksgrænser for j .

Linie 4 k_1, k_2 : Indeksgrænser for k .

Linie 5 Kommentarlínie.

Til sidst angives delblokkenes hjørnepunkter. Det er kun nødvendigt at specificere de hjørnepunkter, der ikke er definerede som en del af overfladeindlæsningen. Gentages disse punkter hér, skal man sikre sig at koordinaterne stemmer overens med de tidligere angivne. Programmet kontrollerer ikke om alle nødvendige hjørnepunkter til delblokkene er angivet. Hver linie angiver data for ét hjørnepunkt, og formatet er:

Linie 1 Kommentarlínie.

Linie 2 i, j, k, x, y, z : Tre integer konstanter, der angiver indices for hjørnepunktet, og tre reelle konstanter, der angiver positionen.

Linie 3 ... Som ovenfor.

Der læses til filslut (end-of-file).

READQM3D I denne underrutine indlæses QM3D-filer (filer genereret af QM3D) i ASCII eller binært format. ASCII-formatet er indkluderet så det er muligt at overføre QM3D filer mellem maskiner uden binær kompatibilitet. Foruden geometri- og topologiinformation gemmes i disse filer også netkontrolvariable. Ved indlæsning af en QM3D fil genoptages en session således i samme tilstand som ved en tidligere afslutning. Filformatet kan udledes af kildeteksten (underrutinerne READQM3D og WRITQM3D) og dokumenteres ikke yderligere her, da brugeren normalt ikke vil have behov for at ændre formatet eller editere i filerne.

2.2.2 Outputfiler

WRITQM3D Genererer QM3D-filer i enten ASCII eller binært format. I disse filer gemmes en igangværende netgenereringssession. Se i øvrigt beskrivelsen af READQM3D i afsnit 2.2.1.

METRIC Underrutinen METRIC beregner cellevoluminet for alle celler i nettet. Hvis der findes negative voluminer skrives indices for cellen samt cellens volumen til filen *metric.dat*. Brugeren informeres og kan herefter læse *metric.dat*, som er en ASCII-fil, og se hvor i nettet der er problemer med netoverlap.

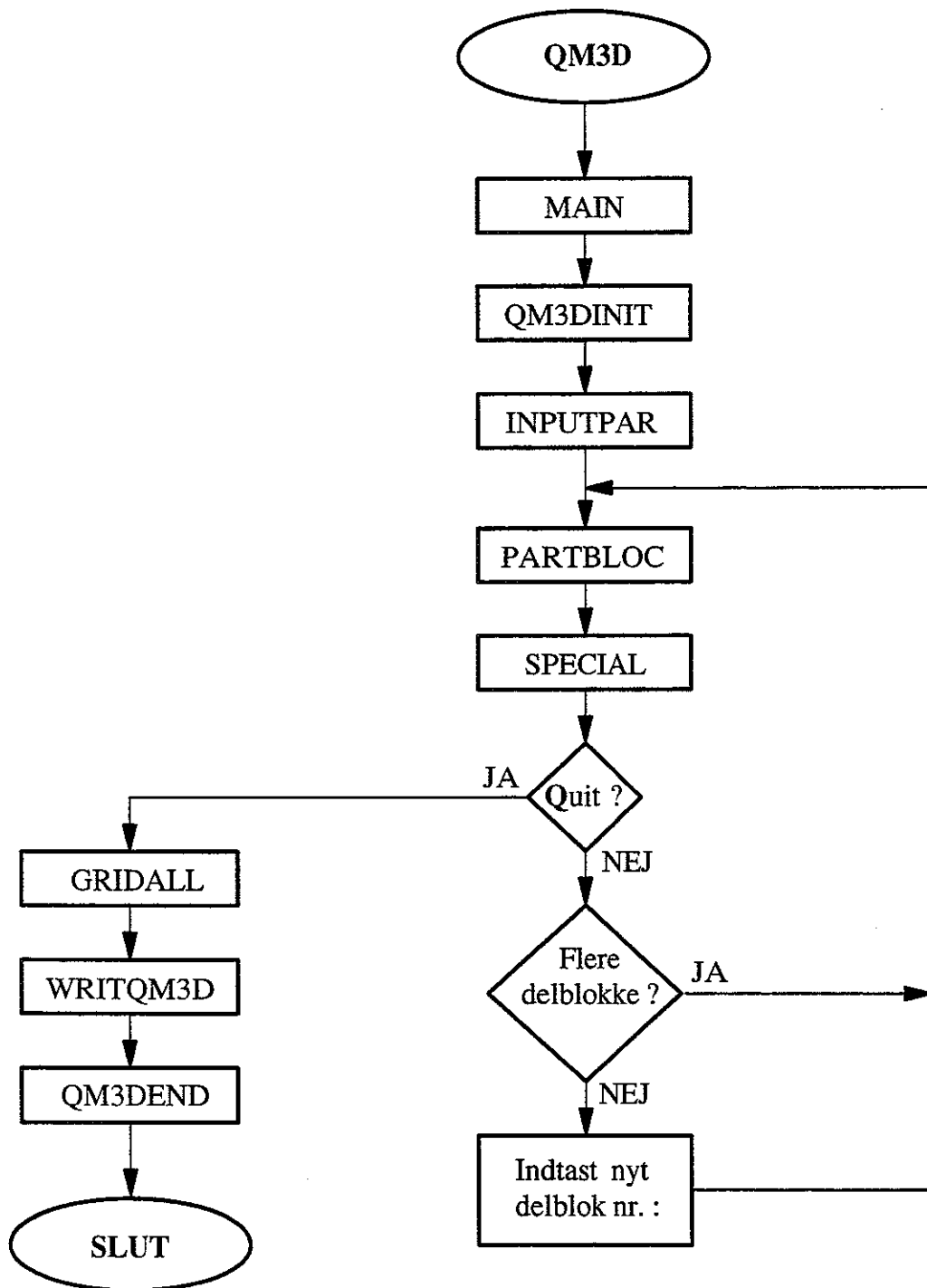
2.2.3 Interne filer

.userqm3d Indeholder brugernavn på brugeren. Bruges i.f.m. systemkald.

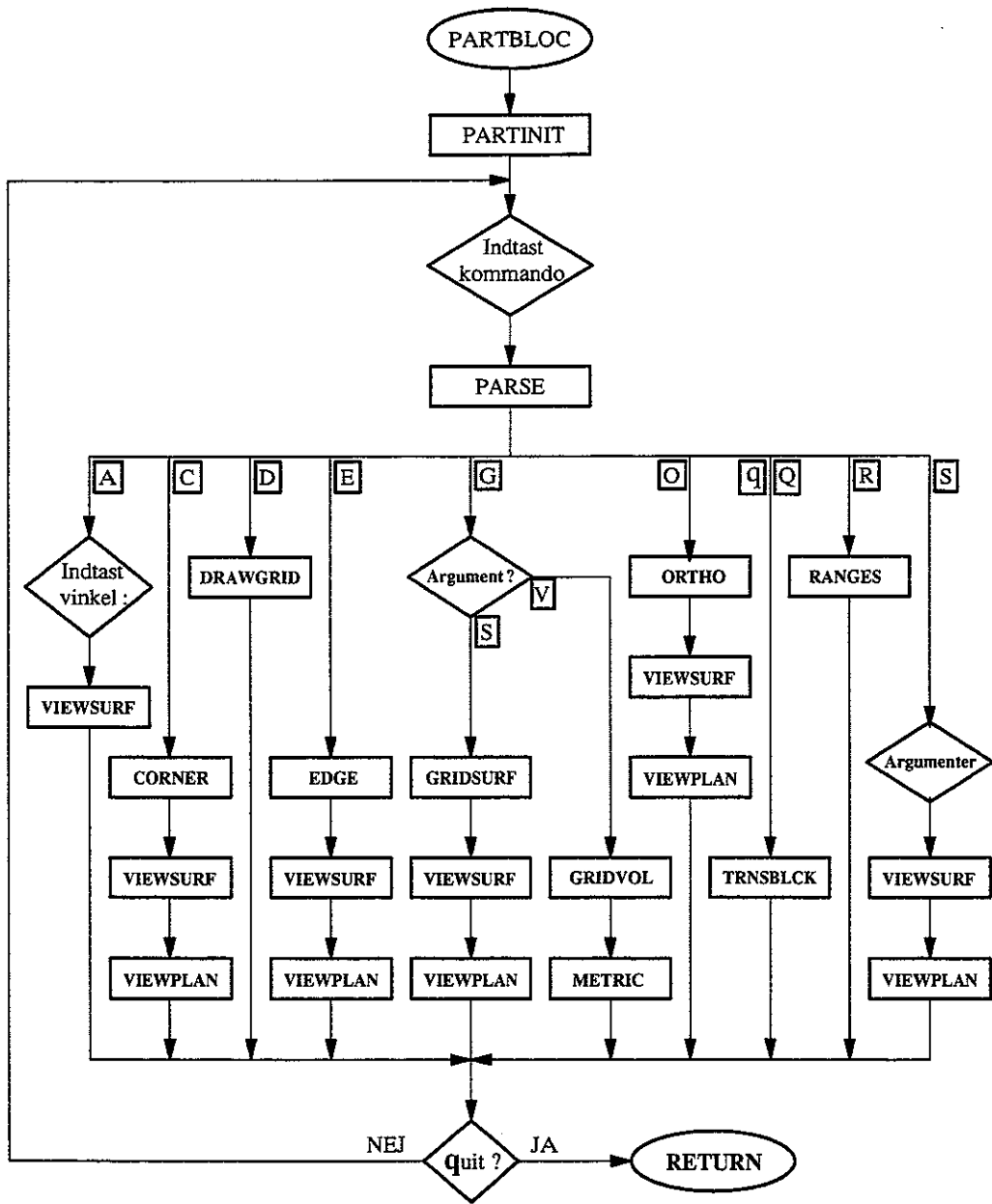
- .qm3dpln** Indeholder processnr. for igangværende kald af HPGLPLAN (en til QM3D tilpasset udgave af HPGLVIEW, AFMEA/1991).
- .qm3dsrf** Indeholder processnr. for igangværende kald af HPGLSURF (en til QM3D tilpasset udgave af HPGLVIEW, AFMEA/1991).
- viewplan.plt** Grafikfil genereret af VIEWPLAN til kald af HPGLPLAN.
- viewsurf.plt** Grafikfil genereret af VIEWSURF til kald af HPGLSURF.
- grdhppl** Grafikfil genereret af DRAWGRID til kald af HPGLVIEW.

2.3 Flowcharts

De følgende flowcharts viser hovedstrukturen i programudførslen. I figur 2.4 ses programudførslen for hovedprogramdelen. Her foretages al opsætning, initialisering og I/O via filer. Den væsentligste under rutine er PARTBLOC for hvilken et flowchart er vist i figur 2.5. Dette er kommandofortolkeren gennem hvilken brugeren kommunikerer med QM3D og nettet skabes og tilpasses.



Figur 2.4: Programudførelse for hovedprogramdelen



Figur 2.5: Programudførelse for kommandofortolkeren

2.4 Begrænsninger

I programmets nuværende form er der en række kedelige begrænsninger. Nogle af disse er "historiske" og knytter sig til de retningslinier der blev udstukket på et relativt tidligt tidspunkt i projektet. Andre begrænsninger er "tidsmæssige" og kan udelukkende henføres til, at der ikke har været tid til at implementere disse faciliteter i koden.

Af "historiske" begrænsninger er der stort set kun en enkelt, nemlig begrænsningen til enkeltblok net. Dette er dog en begrænsning, der er til at leve med, idet et flerblok net godt kan laves ved at køre selvstændige sessioner med QM3D for hver blok. Det kræver blot lidt mere manuelt arbejde af brugeren med at generere de nødvendige geometrifiler. Problemet med denne begrænsning er, at det kræver et forholdsvis stort omprogrammeringsarbejde at ændre koden for at understøtte flerblok netgenerering.

De "tidsmæssige" begrænsninger er af mere kritisk karakter, og de har betydning for programmets anvendelighed. Til gengæld kan disse mangler relativt nemt føjes til programmet, idet den eksisterende kodenstruktur ikke skal ændres afgørende. Det er overvejende et spørgsmål om at udvikle de nødvendige underprogrammer, og at indføre kald af disse i programmet. De mest kritiske begrænsninger i denne kategori er:

Netpunkter Der er ikke mulighed for at tilføje ekstra punkter i nettet.

Hvis man ønsker flere punkter må netgenerering ske "fra bunden", og altid med det ønskede nets opløsning. Det vil være en fordel hvis den indledende netgenerering sker på et groft net, hvorefter det endelige net genereres med samme parametre i en anden opløsning.

Indlæste geometrier De indlæste overflader betragtes som faste og kan ikke ændres (brugeren kan dog gennemtvinge en ændring). En interpolation af disse flader er ønskelig, så nettet også her kan justeres. Hensigten har været at implementere programmet *Surf* i QM3D, men indtil videre foretages overfladenetgenereringen altså separat for givne tre-dimensionelle overflader.

Netkvalitet Det er ønskeligt at give brugeren mere detaljeret information om nettets kvalitet end det sker nu, bla. fordi det kan være vanskeligt at vurdere tre-dimensionelle nets kvalitet visuelt. Tilføjelse af denne facilitet kan ske forholdsvis enkelt.

Delblokinddeling I programmets nuværende form skal der være en én-til-én overensstemmelse mellem delblokflader, dvs. en delblokflade skal være identisk med en nabodelblokflade. Denne begrænsning er meget stiv og betyder, at antallet af delblokke forøges unødigt til skade for overblikket. Det er dog forholdsvis besværligt at ændre dette forhold p.gr.a. parameteroverførslen mellem delblokke, der baserer sig på identiteten.

Strækningsfunktioner Der er et begrænset antal strækningsfunktioner til rådighed. Til gengæld er de forholdsvis fleksible og programmeringsmæssigt meget kompakte. Tilføjelse af nye eller andre strækningsfunktioner er enkelt, men i så fald skal brugeren igennem en længere interaktion for at vælge den ønskede funktion.

Appendix

Herunder vises includefilen *qm3d.inc*.

```
*****
* -----*
* | qm3d.inc |*
* | include file for qm3d (Quick Mesh 3D) program. |*
* | |*
* | Soren Lovmand Hvid |*
* | Dept. of Fluid Mechanics |*
* | Tech. Univ. of Denmark |*
* -----*
*****
implicit real*4(a-h,o-z)
integer orgsurf

c *** Global Parameters *** [ Note: maxn = max(imax,jmax,kmax) ! ]
parameter(imax=62,jmax=80,kmax=44)
parameter(maxn=80)
parameter(maxblock=1,maxpart=18,maxsurf=6*maxpart)
parameter(eps=1.0e-4)

c *** Definitions ***
dimension x(imax,jmax,kmax),y(imax,jmax,kmax),z(imax,jmax,kmax)
dimension x1(imax,jmax,kmax),y1(imax,jmax,kmax),z1(imax,jmax,kmax)
dimension x2(imax,jmax,kmax),y2(imax,jmax,kmax),z2(imax,jmax,kmax)
dimension xp(maxn,maxn),yp(maxn,maxn),zp(maxn,maxn)
dimension xp1(maxn,maxn),yp1(maxn,maxn),zp1(maxn,maxn)
dimension xp2(maxn,maxn),yp2(maxn,maxn),zp2(maxn,maxn)

dimension xi(maxn),eta(maxn)
dimension a1s(maxn),a2s(maxn),b1s(maxn),b2s(maxn)
dimension c1s(maxn),c2s(maxn),xis(maxn),etas(maxn)
dimension xi1s(maxn),xi2s(maxn),eta1s(maxn),eta2s(maxn)
dimension xnxis(2,maxn),ynxis(2,maxn),znxis(2,maxn)
dimension xnetas(2,maxn),ynetas(2,maxn),znetas(2,maxn)

dimension xiv(imax),etav(jmax),zetav(kmax)
dimension a1(imax),a2(imax),b1(jmax),b2(jmax),c1(kmax),c2(kmax)
dimension di1(imax),di2(imax),dj1(jmax),dj2(jmax)
dimension xi11(imax),xi12(imax),xi21(imax),xi22(imax)
dimension eta11(jmax),eta12(jmax),eta21(jmax),eta22(jmax)
dimension zeta11(kmax),zeta12(kmax),zeta21(kmax),zeta22(kmax)
dimension xnxi(2,jmax,kmax),xneta(imax,2,kmax),xnzeta(imax,jmax,2)
dimension ynxi(2,jmax,kmax),yneta(imax,2,kmax),ynzeta(imax,jmax,2)
```

```

dimension znxi(2,jmax,kmax),zeta(imax,2,kmax),znzeta(imax,jmax,2)

dimension xiface(maxblock,maxpart,6),xivol(maxblock,maxpart)
dimension etaface(maxblock,maxpart,6),etavol(maxblock,maxpart)
dimension zetaface(maxblock,maxpart,6),zetavol(maxblock,maxpart)

dimension rn1(maxblock,maxpart,12,3),rn2(maxblock,maxpart,12,3)
dimension dist(maxblock,maxpart,12,3)

dimension npart(maxblock),nsurf(maxblock),orgsurf(maxblock)
dimension iread(maxblock,maxsurf,6)
dimension irange(maxblock,maxpart,2)
dimension jrange(maxblock,maxpart,2)
dimension krange(maxblock,maxpart,2)

character*80 filename(maxblock)
dimension len(maxblock)

character string*80,cmd*1,carg*1
dimension iarg(3)
dimension arg(3)

character username*10,hpgplan*38,hpgsurf*38,askhpgl*65
character pidsurf*10,pidplan*10,killpid*18

logical fixcorn(maxblock,maxpart,8)
logical fixedge(maxblock,maxpart,12),fixsurf(maxblock,maxpart,6)
logical init,binqm3d,ascqm3d,quit

c *** Global common blocks ***
common/geomv/x,y,z,x1,y1,z1,x2,y2,z2
common/geoms/xp,yp,zp,xp1,yp1,zp1,xp2,yp2,zp2

common/dists/a1s,a2s,b1s,b2s,c1s,c2s,xis,etas,xi1s,xi2s,eta1s,
&          eta2s,xi,eta
common/norms/xnxis,ynxis,znxis,xnetas,ynetas,znetas

common/distv/xiv,etav,zetav,a1,a2,b1,b2,c1,c2,di1,di2,dj1,dj2,
&          xi11,xi12,xi21,xi22,eta11,eta12,eta21,eta22,
&          zeta11,zeta12,zeta21,zeta22
common/normv/xnxi,xneta,xnzeta,ynxi,yneta,ynzeta,znxi,zneta,znzeta

common/blockpar/xiface,etaface,zetaface,xivol,etavol,zetavol,
&          rn1,rn2,dist

common/range/nblock,npart,nsurf,orgsurf,

```

```
&          ni,nj,nk,irange,jrange,krange
common/read/iread
common/fixall/fixcorn,fixedge,fixsurf

common/view/iblock,ipart,indsurf,isurf,angle

common/chars/string,cmd,carg,filename
common/chpgl/username,hpgplan,hpgsurf,askhpgl
common/pids/pidsurf,pidplan,killpid
common/cmdargs/len,iarg,arg
common/logic/init,binqm3d,ascqm3d,quit
```

Herunder vises et eksempel på filen *qm3d.dat*.

```
1  No. of blocks
qm3dtest.3d
```

```
*** Any number of lines may be used below here for comments ***
The file qm3dtest.3d contains a simple geometry - a cube - where
one face has been fixed with a given surface mesh.
```

Herunder vises geometrifilen *qm3dtest.3d* til illustration af det anvendte format. Filen findes  vrigt sammen med QM3D kildeteksten, og er en behagelig geometri at  ve sig p  i brugen af QM3D.

```

*****
* Test geometryfile for QM3D. A box with one fixed surface. *
*****
  1          ! No. of surfaces in file
! ** This is a geometryfile for QM3D, test configuration **
! ** Surface of box **  nrow = 3 , ncol = 3
  1  1          ! surfind, surfcon
  1  3  1  3          ! j1,j2,k1,k2
  0.0  0.0  0.0
  0.0  2.0  0.0
  0.0  4.0  0.0
  0.0  0.0  2.0
  0.0  2.0  2.0
  0.0  4.0  2.0
  0.0  0.0  4.0
  0.0  2.0  4.0
  0.0  4.0  4.0

! ** specification of part blocks to follow **
  4          ! No. of part blocks in block
! ** part block #1 **
  1  7  irange
  1  3  jrange
  1  3  krange

! ** part block #2 **
  1  7  irange
  3  7  jrange
  1  3  krange

! ** part block #3 **
  1  7  irange
  1  3  jrange
  3  7  krange

! ** part block #4 **
  1  7  irange
  3  7  jrange
  3  7  krange

! ** Specification of corners.**
  1  1  7  0.0  0.0  14.0

```

1 3	7	0.0	4.0	14.0
1 7	1	0.0	14.0	0.0
1 7	3	0.0	14.0	4.0
1 7	3	0.0	14.0	14.0
7 1	1	14.0	0.0	0.0
7 3	1	14.0	4.0	0.0
7 7	1	14.0	14.0	0.0
7 1	3	14.0	0.0	4.0
7 3	3	14.0	4.0	4.0
7 7	3	14.0	14.0	4.0
7 1	7	14.0	0.0	14.0
7 3	7	14.0	4.0	14.0
7 7	7	14.0	14.0	14.0

Litteraturfortegnelse

- [1] Thompson, J., C.W.Mastin and Z.U.A. Warsi, *Numerical Grid Generation*, North Holland, 1985.
- [2] Farin, G., *Curves and Surfaces for Computer Aided Geometric Design*, Academic Press, 1988.
- [3] Faux, I. D. and M.J. Pratt, *Computational Geometry for Design and Manufacture*, Ellis Horwood Ltd., 1979.
- [4] Hvid, S.L., *Curvature Based Smoothing of Plane Cubic B-spline Curves*, Report AFM 93-08, 1993, Dept. of Fluid Mechanics, Techn. Univ. of Denmark.
- [5] Hvid, S.L., *Surface Description Using Bicubic B-splines*, Report AFM 93-10, 1993, Dept. of Fluid Mechanics, Techn. Univ. of Denmark.
- [6] Hvid, S.L., *Three Dimensional Algebraic Grid Generation*, To be published at Dept. of Fluid Mechanics, Techn. Univ. of Denmark.
- [7] Eiseman, P.R. and R.E. Smith, *Mesh Generation Using Algebraic Techniques*, Numerical Grid Generation, NASA CP 2166, 1980.
- [8] R.E. Smith and L-E. Eriksson, *Algebraic Grid Generation*, Comp. Meth. Appl. Mech. and Eng. **64**, 1987.
- [9] Vinokur, M., *On One-Dimensional Stretching Functions for Finite-Difference Calculations*, J. Comp. Phys. **50**, 1983.