



UPPSALA
UNIVERSITET

Friktionsförluster i induktionsmotorer

Nicolas Forsberg

Kristian Rönnerberg

Rapport i Teknisk-vetenskapliga datorberäkningar

Maj 2008

PROJEKTRAPPORT

Abstract

This project models the air flow through a large electrical motor. The target physical phenomena is the flow of air which is used to cool parts of a motor, which are heated during operation. The air flow is driven by the rotating parts of the engine. The goal of this project is to find how much mechanical energy is required to drive the flow at different rounds per minute. Navier-Stokes equations are solved using FEM (Finite element method). The software which is used for defining the geometry and computing the results is COMSOL Multiphysics 3.4. The flow is assumed to be incompressible and isothermal. Models in 2D and 3D are evaluated. The amount of available main memory (16 GB) limits the complexity of the geometries which are studied.

In order to compute the air flow, and thereby the transmitted energy as a function of the rotational velocity, a 3D model must be used. None of the evaluated models could correctly estimate the mechanical energy transmitted since even the most complex models were a too coarse reduction from the real machine.

Sammanfattning

I detta projekt har luftflödet genom en stor elektrisk motor simulerats. Detta luftflöde används för att kyla de delar av maskinen som blir varma vid bruk. Luftflödets drivs av den elektriska maskinens roterande delar. Förhoppningen var att finna hur mycket mekanisk energi som överförs från motorns roterande delar till den omgivande luften vid olika varvtal. Navier-Stokes ekvation löstes med FEM (Finita elementmetoden). Programvaran som användes för geometridefinition och beräkning var COMSOL Multiphysics 3.4. Flödet har antagits vara inkompressibelt och isothermt. Modeller i 2D och 3D utvärderades. Hur pass komplicerad geometri som kunde användas i simuleringarna begränsades av mängden tillgängligt arbetsminne (16 GB).

För att beräkna luftflöde, och därmed överförd energi, som en funktion av motorns varvtal måste en 3D-modell användas. Ingen av de prövade modellerna kunde korrekt uppskatta överföringen av mekanisk energi. De mest komplicerade 3D-modeller som testades var för grova förenklingar av verkligheten.

Keywords: Navier-Stokes, CFD, Fluid dynamics, rotating frame of reference, FEM.

Success is the ability to go from one failure to another with no loss of enthusiasm.

– *Winston Churchill*

Contents

1	Förord	5
1.1	Tack till	5
2	Inledning	6
2.1	Mål	6
2.2	Problemet	6
2.3	Navier-Stokes ekvation och FEM	8
2.3.1	FEM på Poissons ekvation	8
2.3.2	Navier-Stokes	11
2.3.3	Roterande referenssystem	12
2.4	Bromsande kraften och effekten	13
2.5	Resonemang runt det numeriska	13
2.6	Tillvägagångssätt	14
2.7	Mjukvara och hårdvara	14
3	Modeller	15
3.1	Modeller i 2D	15
3.1.1	Statisk 2D-modell	15
3.1.2	2D-modell med roterande beräkningsnät	15
3.1.3	Statisk 2D-modell med maskinstomme	16
3.2	Modeller i 3D	17
3.2.1	3D-modell med roterande beräkningsnät	17
3.2.2	Statisk 3D-modell med ansatta Coriolis- och centrifugalkrafter	18
3.2.3	Statisk 3D-modell ‘tårtbit’ med ansatta Coriolis- och centrifugalkrafter	18
3.2.4	Statisk 3D-modell platt ‘tårtbit’	19
3.2.5	Statisk 3D-modell med roterande väggar som randvillkor	19
4	Resultat	20
4.1	Modeller i 2D	20
4.1.1	Statisk 2D-modell	20
4.1.2	2D-modell med roterande beräkningsnät	20
4.1.3	Statisk 2D-modell med maskinstomme	21
4.2	Modeller i 3D	22
4.2.1	3D-modell med roterande beräkningsnät	22
4.2.2	Statisk 3D-modell med ansatta Coriolis- och Centrifugalkrafter	22
4.2.3	Statisk 3D-modell ‘tårtbit’ med ansatta Coriolis- och centrifugalkrafter	22
4.2.4	Statisk 3D-modell platt ‘tårtbit’	23
4.2.5	Statisk 3D-modell med roterande väggar som randvillkor	24
5	Diskussion	25
5.1	Analys av modeller	25
6	Slutsatser	26

List of Figures

1	Komplett maskin	7
2	Rotor	7
3	Stator	7
4	Maskinens kylkrets	8
5	Poissons Ekvation, beräkningsnät.	9
6	Poissonsekvation, lösning	11
7	2D-Geometri, roterande nät	16
8	2D-Geometri, maskinstomme	17
9	3D-Geometri, roterande nät.	18
10	3D-Geometri, 'roterande nättårtbit'.	18
11	3D-Geometri, platt 'tårtbit'	19
12	3D-Geometri, roterande väggar	19
13	Resultat statisk 2D-modell	20
14	Resultat 2D-modell med roterande beräkningsnät	21
15	Resultat 2D-modell med maskinstomme	21
16	Resultat 3D-modell med ansatta Coriolis- och Centrifugal-krafter	22
17	Resultat 3D-modell 'tårtbit' med ansatta krafter	23
18	Resultat "tårtbits"-modell i 3D	24
19	Resultat 3D-modell med roterande väggar	24

List of Tables

1	Data - Statisk 2D-modell	20
2	Data - Statisk 2D-modell med roterande beräkningsnät	20
3	Data - Statisk 2D-modell med maskinstomme	21
4	Data - Statisk 3D-modell med ansatta Coriolis- och Centrifugalkrafter	22
5	Data - Statisk 3D-modell 'tårtbit' med ansatta Coriolis- och Centrifugalkrafter	23
6	Data - Statisk 3D-modell platt 'tårtbit'	23
7	Data - Statisk 3D-modell med roterande väggar	24

1 Förord

Detta arbete utfördes under tio veckor våren 2008 inom ramen för kursen “Tekniskt-vetenskapliga datorberäkningar” vid Uppsala Universitet. Själva uppgiften kom till genom att undertecknade tog kontakt med ABB Machines och frågade om de hade något som de ville ha undersökt, och som låg inom ramen för ovan nämnd kurs. Det man ville ha reda på var om det gick att ta fram en beräkningsmodell för hur stor effektförlusten blev för en elektrisk maskin på grund av att den driver sin egen kyl-luft.

1.1 Tack till

Vi vill rikta ett tack till, utan inbördes ordning, följande:

Jonas Hagberg och Tore Sundqvist, och resten av personalen vid UPPMAX, för hjälpen med mjukvaran på UPPMAX resurser.

Therese Gustafsson från COMSOL AB för att hon bidragit med tillfälliga licenser för våra arbetsstationer och laptops. Detta har underlättat vårt arbete.

Henrik Hedlund och Joel Fredrikson, systemadministratörer vid IT-Institutionen, förtjänar ett tack för hjälpen med hårdvara och nätverksfrågor.

Vi vill också rikta ett tack till Per Lötstedt, professor i numerisk analys, för att ha agerat bollplank och kommit med tips och idéer.

Maya Neytcheva som varit ansvarig för kursen, sett till att allt fungerat smidigt och kommit med tips och idéer.

Djordje Savinovic, vår handledare på ABB Machines, och Holger Persson, utvecklingschef på ABB Machines, som fixade det här projektet åt oss.



*Nicolas Forsberg och Kristian Rönnberg
Polacksbacken rum 2445, Uppsala - Våren, nådens år 2008*

2 Inledning

Då ABB Machines konstruerar sina maskiner används vissa konstruktionsparametrar, i form av dimensioner på ingående maskinkomponenter, för att göra en teoretisk uppskattning av fysikaliska storheter som t.ex. resistans, lossryckningsmoment, kortslutningsmoment och liknande. Beräkningarna av dessa värden sker i ABB:s egna beräkningsprogram OSKAR. Idag saknas en modell, för en specifik maskintyp, för hur mycket effekt som går åt för att transportera kyl-luft genom maskinen vid drift, d.v.s. hur stora effektförluster som uppstår till följd av att maskinen själv står för att driva cirkulationen av luften eftersom ingen extern kraftkälla används. I fortsättningen kommer denna effektförlust att refereras till som "friktionsförlust" för att använda den terminologi som används inom ABB Machines. Målet med detta projekt är att finna en modell för denna friktionsförlust.

2.1 Mål

Projektets mål är att:

- Utgående från givna mått på ingående maskinkomponenter konstruera en ekvivalent maskinmodell som kan användas för att uppskatta friktionsförlusterna från kyl-luften.
- Korrekt uppskatta friktionsförlusterna utgående från den ekvivalenta maskinmodellen.
- Utvärdera om den ekvivalenta maskinmodellen kan användas för maskiner med olika dimensioner hos de ingående komponenterna.

Då den verkliga geometrin för maskinen är komplicerad är förhoppningen att hitta en ekvivalent maskinmodell, d.v.s. en maskinmodell med samma beteende som den verkliga men i någon mening förenklad geometri. Denna modell ska användas som utgångspunkt för att beräkna den efterfrågade friktionsförlusten. Denna ekvivalenta modell ska ge korrekta resultat, för den specifika maskintypen ifråga, vid alla rimligt tänkbara varvtal. Det är också en målsättning att modellen ska kunna utökas för att användas på andra konstruktioner än den nuvarande.

2.2 Problemet

Förenklat kan man säga att maskinen, som syns i figur 1, består av fyra komponenter: rotor, stator, maskinstomme och kylartopp. Rotorn är den del av maskinen som snurrar vid drift och det är rotorns axel som överför mekanisk energi från maskinen till den applikation man vill driva, till exempel en vattenpump.

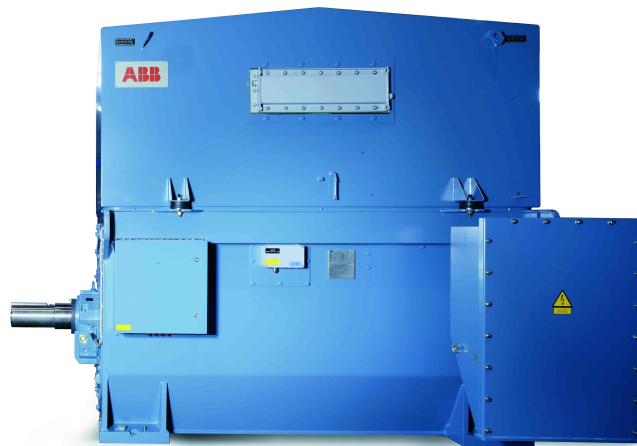


Figure 1: Så här ser en färdigmonterad maskin ut. Det som sticker ut till vänster är rotoraxeln, som överför mekanisk energi till omgivningen. (Reproducerad med tillstånd från ABB Machines)

På rotoraxeln, som kan ses på figur 2, är balkar monterade och på dessa balkar krympts plåtpaket bestående av elektroplåt fast genom att först hettas upp och sedan avkylas. Balkarna fungerar också som fläktvingar och är den del av maskinen som står för att suga in luft och trycka ut den genom kylkanalerna i rotorn.

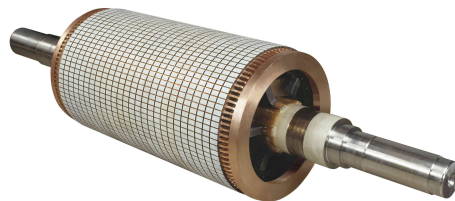


Figure 2: Detta är en bild på en komplett rotor. Innanför den kopparfärgade ringen kan balkarnas ändrar skönjas. Dessa balkar löper längs hela insidan av plåtpaketet och fungerar som fläktvingar. De svarta linjer som löper längs omkretsen är kylkanalernas utlopp. (Reproducerad med tillstånd från ABB Machines)

När rotorn är monterad i den kompletta maskinen sitter den så att vissa delar befinner sig i statorn. Statorn visas i figur 3. Magnetfälten som induceras i rotor och stator ger upphov till rotoraxelns rotation. På statorn finns kylkanaler placerade på ungefär samma sätt som på rotorn.

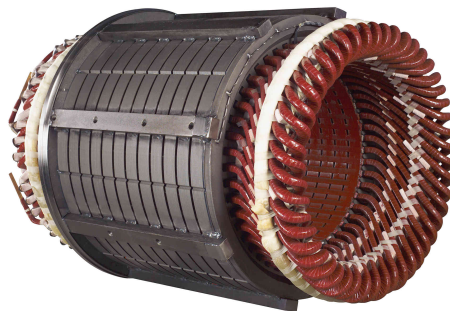


Figure 3: På bilden syns en stator. De svarta linjer som löper längs omkretsen är statorns kylkanaler. (Reproducerad med tillstånd från ABB Machines)

Den kylluft som pressas genom rotorn skall passera genom statorn på sin väg ut till kylartoppen där en värmväxlare sitter. Denna värmväxlare kyler ner luften innan den leds tillbaka in i rotorn där den återigen får verka som kylmedium se figur 4.

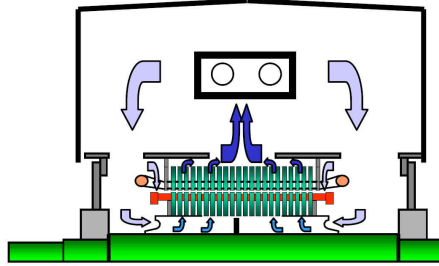


Figure 4: En principbild över maskinens kylkrets. (Reproducerad med tillstånd från ABB Machines)

Rotor och stator är monterade inuti maskinstomme. Geometrin hos maskinstommen påverkar hur kyl-luften kan cirkulera inuti maskinen och komplicerar luftflödet ytterligare.

Det är dessa delar som utgör maskinens. Det finns ett stort antal föremål som obstruerar luftflödet både i stator och rotor. I och med att rotorn roterar ändras lufkanalerna i rotor och stator position relativt varandra vilket påverkar luftflödet.

På denna geometri kommer Navier-Stokes inkompressibla ekvationer att lösas. Dessa är en uppsättning icke-linjära partiella differentialekvationer som sammankopplar hastighet, krafter och tryck i fluider. I detta projekt används Navier-Stokes ekvationer för att bestämma de efterfrågade friktionsförlusterna. Fler detaljer om Navier-Stokes ekvationer tas upp nedan, avsnitt 2.3.

2.3 Navier-Stokes ekvation och FEM

2.3.1 FEM på Poissons ekvation

För att illustrera Finita Element Metoder (FEM) löses här *Poissons ekvation*. Ekvationen löses på enhetskvadraten, Ω med homogent *Dirchlet*-randvillkor, här innebär det att u föreskrivs ett specifikt värde längs med randen, $\partial\Omega$. Det man gör med FEM är att interpolera lösningen med hjälp av så kallade basfunktioner. I detta exempel används styckvis linjära basfunktioner.

$$\nabla^2 u = f \quad x, y \in \Omega \quad (1a)$$

$$u = 0 \quad x, y \in \partial\Omega \quad (1b)$$

Det första steget är att definiera variationsformuleringen. Antag att u löser (1) då löser den också (2).

$$\int_{\Omega} \nabla^2 u \phi \, d\Omega = \int_{\Omega} f \phi \, d\Omega \quad (2)$$

Där (1a) har multiplicerats med en än så länge godtycklig *testfunktion*, ϕ , vilken uppfyller (1b). Med hjälp av *Greens sats* kan (2) skrivas på symmetrisk form.

$$\int_{\partial\Omega} (\nabla u \phi) d\partial\Omega - \int_{\Omega} \nabla u \nabla \phi d\Omega = \int_{\partial\Omega} f \phi d\Omega \quad (3)$$

Första integranden är noll över hela randen så variationsformuleringen är: Finn u sådan att (4) uppfylls för varje ϕ och sådan att randvillkoret (1b) uppfylls.

$$- \int_{\Omega} \nabla u \nabla \phi d\Omega = \int_{\partial\Omega} f \phi d\Omega \quad (4)$$

Uppenbart är att om u uppfyller variationsformuleringen, uppfyller den också *poissons ekvation* (1). Fördelen med variationsformuleringen är inte alltid helt uppenbar men den tillåter formulering av en svag form av lösningen d.v.s. istället för kräva att u uppfyller variationsformuleringen över hela domänen lättas kravet på u . Det räcker om u uppfyller variationsformuleringen på ett ändligt antal punkter, *noder*, i domänen. Denna svaga form av lösningen kommer hädanefter att betecknas u^h . Det har visats att skillnaden mellan "den riktiga lösningen" och den svaga kan göras godtyckligt liten genom att öka antalet noder och minska avståndet mellan dessa.

Nästa steg är att skapa ett beräkningsnät. Hur detta går till i praktiken ligger utanför denna rapport. I detta fall används ett av Comsol Multiphysics genererat beräkningsnät bestående av 13 noder, n_i . Utifrån dessa noder bestäms element, $e^{(i)}$, i detta fall 16 stycken. Hur man bäst skapar elementen ligger också utanför denna rapport men en metod som ska nämnas är Voronoï-triangulering.

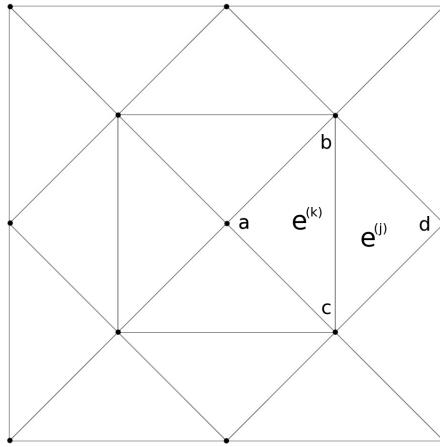


Figure 5: Beräkningsnät, element och noder på enhetskvadraten.

Som nämnts ovan kommer här att användas styckvis linjära basfunktioner för att beskriva den svaga lösningen, u^h . Varje nod kommer att få en egen basfunktion, ϕ_j ; Denna väljs sådan att den är 1 endast vid sin associerade nod, n_j och 0 vid alla andra noder. Det visar sig att när basfunktionerna väljs på detta sätt är de ortogonala och u^h kan skrivas som en linjär kombination av dessa basfunktioner.

$$u^h = \sum_j u_j^h \phi_j \quad (5)$$

Här behövs en distinktion; \mathbf{u}^h är en vektor bestående av komponenterna u_j^h medan u^h enligt tidigare är approximationen till lösningen. Kan man nu bestämma samtliga u_j^h sådana att u^h uppfyller variationsformuleringen (4) är lösningen klar. Detta görs med *Petrov-Galerkinmetoden*, man bestämmer \mathbf{u}^h genom att se till att variationsformuleringen uppfylls av u^h för alla ϕ_i .

$$-\int_{\Omega} \nabla \left(\sum_j u_j^h \phi_j \right) \nabla \phi_i \, d\Omega = \int_{\partial\Omega} f \phi_i \, d\Omega \quad (6)$$

Vilket skrivs om till

$$-\sum_j u_j^h \int_{\Omega} \nabla \phi_j \nabla \phi_i \, d\Omega = \int_{\partial\Omega} f \phi_i \, d\Omega \quad (7)$$

Detta ger upphov till ett ekvationsystem, $A\mathbf{u}^h = \mathbf{b}$, där följande gäller

$$A = \{a_{i,j}\} \quad a_{i,j} = -\int_{\Omega} \nabla \phi_j \nabla \phi_i \, d\Omega \quad \mathbf{b} = \int_{\partial\Omega} f \phi_i \, d\Omega \quad (8)$$

Det är nu lätt att bestämma $a_{i,j}$ och \mathbf{b} eftersom ϕ_i och f är kända funktioner. Integralerna kan evalueras numeriskt eller analytiskt. Med tanke på basfunktionernas struktur är det lätt att inse att $a_{i,j} = 0$ då ϕ_i och ϕ_j tillhör olika element. Detta är bra då det ger en gles matris som kräver förhållandevis lite minne att arbeta med. Det är enklast att skapa A genom att koncentrera sig på ett element i taget. Betrakta element $e^{(k)}$ i figur 5. Elementet består av noderna a, b och c . De enda basfunktioner som är nollskilda på detta element är just ϕ_a, ϕ_b och ϕ_c därför bygger man först upp en elementmatris, $E^k = \{e_{i,j}^{(k)}\}$ för element $e^{(k)}$ blir E^k .

$$E = \begin{pmatrix} \int_{e^k} (\nabla \phi_a)^2 \, d\Omega & \int_{e^k} \nabla \phi_a \nabla \phi_b \, d\Omega & \int_{e^k} \nabla \phi_a \nabla \phi_c \, d\Omega \\ \int_{e^k} \nabla \phi_b \nabla \phi_a \, d\Omega & \int_{e^k} (\nabla \phi_b)^2 \, d\Omega & \int_{e^k} \nabla \phi_b \nabla \phi_c \, d\Omega \\ \int_{e^k} \nabla \phi_c \nabla \phi_a \, d\Omega & \int_{e^k} \nabla \phi_c \nabla \phi_b \, d\Omega & \int_{e^k} (\nabla \phi_c)^2 \, d\Omega \end{pmatrix} \quad (9)$$

Varje $e_{i,j}^k$ adderas till A på respektive plats. Detta gör man en gång per element i domänen. Vektorn \mathbf{b} består endast av kända storheter. Nu består ekvationsystemet av lika många obekanta som ekvationer och man kan vänta sig en unik lösning. Men först ska randvillkoren implementeras. Även för detta finns det olika metoder att göra detta på. Det enklaste är att ersätta de ekvationer som hör till en viss nod med randvillkoret för den noden. Betrakta nod n_d i figur 5 som hör till randen. Då blir $a_{d,d} = 1$ och $b_d = 0$.

Det som återstår är att lösa ekvationsystemet $A\mathbf{u}^h = \mathbf{b}$. Detta görs med fördel med en numerisk lösare. I figur 6 är Poissons ekvation löst med $f = x^2 + y^2$ och med beräkningsnätet i figur 5.

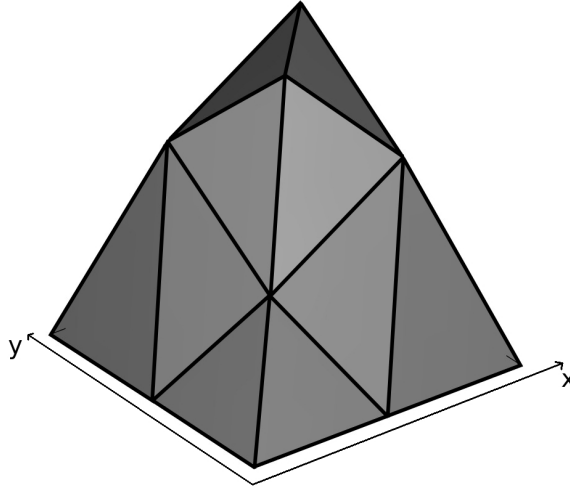


Figure 6: Poissonsekvation löst med linjära basfunktioner och 16 element.

2.3.2 Navier-Stokes

I detta projekt används Navier-Stokes inkompressibla ekvation tillsammans med kontinuitetsekvationen för inkompressibelt flöde för att lösa problemen. Här visas hur variationsformuleringen ser ut för dessa ekvationer samt hur basfunktioner används för att approximera lösningen.

$$\rho \left[\frac{\partial \mathbf{v}}{\partial t} + (\mathbf{v} \cdot \nabla) \mathbf{v} \right] = -\nabla p + \mathbf{f} + \eta \nabla^2 \mathbf{v} \quad (10a)$$

$$\nabla \cdot \mathbf{v} = 0 \quad (10b)$$

Här är ρ densitet, \mathbf{v} är hastighetsfältet, p är trycket, \mathbf{f} är volymskraft och η är viskositeten.

Tillvägagångssättet är i stort sett samma som för Poissons ekvation i avsnitt 2.3.1. En uppenbar skillnad är att vi nu löser för fyra variabler, en för tryck och en för varje hastighetskomponent. Ytterligare en skillnad är att man väljer att för hastighetsfältet använda basfunktioner av en ordning högre. Detta behövs för att få rimlig konvergens. För trycket räcker det med basfunktioner av samma typ som användes i Poissons ekvation. Mer om olika typer av basfunktioner och FEM i allmänhet beskrivs av R. Wait i [5].

Då man löser för stationärt flöde, som ofta är fallet i detta projekt, är $\frac{\partial \mathbf{v}}{\partial t} = 0$. För att fortsätta med variationsformuleringen: Multiplicera Navier-Stokes ekvation, (10a), med basfunktionerna av högre ordning, ψ_i och multiplicera sedan kontinuitetsekvationen, (10b) med vanliga basfunktioner, ϕ_j och integrera dessa över domänen, Ω .

$$\int_{\Omega} \left(-\eta \nabla^2 \mathbf{v} + \rho (\mathbf{v} \cdot \nabla) \mathbf{v} + \nabla p - \mathbf{f} \right) \psi_i d\Omega = 0 \quad (11a)$$

$$\int_{\Omega} \nabla \cdot \mathbf{v} \phi_j d\Omega = 0 \quad (11b)$$

Partialintegrera och man får

$$\int_{\Omega} \eta \nabla \mathbf{v} \nabla \psi_i + (\rho(\mathbf{v} \cdot \nabla) \mathbf{v} + \nabla p - \mathbf{f}) \psi_i d\Omega = 0 \quad (12a)$$

$$\int_{\Omega} \nabla \mathbf{v} \phi_j d\Omega = 0 \quad (12b)$$

För att illustrera svårigheterna med Navier-Stokes ekvationer introduceras här dimensionslösa variabler. Varför detta är nödvändigt visas strax. Låt $\mathbf{v} = v_0 \mathbf{v}'$, $\nabla = \frac{1}{D} \nabla'$, och $p = v_0^2 \rho_0 p'$. De primmade variablerna saknar nu dimension. Konstanterna v_0 , D och ρ_0 väljes som karakteristiska storheter. Basfunktionerna är redan dimensionslösa. Det är brukligt att här använda Reynolds tal:

$$Re = \frac{v_0 D \rho_0}{\eta} \quad (13)$$

Skriv om (12a) och efter några steg fås:

$$\int_{\Omega} \nabla' \mathbf{v}' \nabla' \psi_i + Re((\mathbf{v}' \cdot \nabla') \mathbf{v}' + \nabla' p' - \mathbf{f}') \psi_i d\Omega = 0 \quad (14)$$

Utifrån denna ekvation kan man dra ett par viktiga slutsatser. För högre Reynoldstal växer den term som är icke-linjär, $(\mathbf{v} \cdot \nabla) \mathbf{v}$. Detta gör att ekvationen är betydligt svårare att lösa numeriskt. I praktiken resulterar ett högt Reynoldstal i att man behöver tätare nät, vilket ställer höga krav på hårdvara. Reynoldstal ökar med högre hastighet och med låg viskositet. Induktionsmotorn roterar med 1500 *rpm* vilket på sina ställen ger upphov till väldigt hög hastighet. Luft har en låg viskositet. Detta tillsammans gör att det krävs väldigt fint nät. En annan viktig slutsats är att om två olika problem delar samma geometri fast skalad, tänk vindtunnel, är ekvationerna som löses exakt samma så länge Reynoldstal hålls konstant. Detta kallas för *Reynolds likhetsprincip* och är det faktum som gör att man kan dra slutsatser av vindtunnelexperiment. Mer om hur Navier–Stokes Ekvation löses med hjälp av FEM beskrivs på ett lättfattligt sätt av J. Burckardt i [1]

2.3.3 Roterande referenssystem

I detta projekt har det i vissa fall visat sig bättre att använda roterande referenssystem istället för att använda inbyggda funktionen “rotating mesh” som finns i Comsol Multiphysics. När man transformerar från ett statiskt referenssystem till ett som roterar med konstant vinkelhastighet uppkommer ’fiktiva’ krafter i form av *Centrifugalkraften* och *Corioliskraften*, $\omega \times \omega \times \mathbf{r}$ respektive $2\omega \times \mathbf{v}$, där \mathbf{r} är lägesvektorn. Detta förklaras mer utförligt i [4]. Centrifugalkraften ändrar inte ekvationen nämnvärt men Corioliskraften är hastighetsberoende och kommer därför att ha hög inverkan på hur ekvationen beter sig, framför allt vid hög rotationshastighet och vid hög radiell hastighet. Navier-Stokes ekvation i roterande referenssystem blir enligt [2].

$$\rho \left[\frac{\partial \mathbf{v}}{\partial t} + (\mathbf{v} \cdot \nabla) \mathbf{v} + 2\omega \times \mathbf{v} + \omega \times \omega \times \mathbf{r} \right] = -\nabla p + \eta \nabla^2 \mathbf{v} + \mathbf{f} \quad (15)$$

Utförs samma beräkningar som i det fixa referenssystemet fast man också introducerar den di-

mensionslösa vinkelhastigheten $\omega = \frac{v_0}{D}$ blir den dimensionslösa variationsformuleringen:

$$\int_{\Omega} \nabla' \mathbf{v}' \nabla' \psi_i + Re \left[(\mathbf{v}' \cdot \nabla') \mathbf{v}' + \nabla' p' + 2\omega' \times \mathbf{v}' + \omega' \times \omega' \times \mathbf{r}' - \mathbf{f}' \right] \psi_i d\Omega = 0 \quad (16a)$$

$$\int_{\Omega} \nabla' \mathbf{v}' \phi_j d\Omega = 0 \quad (16b)$$

Samma effekt uppstår här, med högre Reynoldstal blir den icke linjära delen dominant. Dessutom syns effekten att Centrifugal- och Corioliskraften blir större vid högre varvtal.

2.4 Bromsande kraften och effekten

Momentet som en ytkraft ger upphov till ges av

$$\mathbf{M} = \int_A (\mathbf{r} \times \mathbf{f}) dA \quad (17)$$

där \mathbf{f} är ytkraften som verkar på arean A . För att få det totala momentet som luften påverkar rotorn med måste integralen tas över hela rotorns area. Effekten räknas sedan ut som

$$P = \mathbf{M} \cdot \boldsymbol{\omega} \quad (18)$$

Ytkraften \mathbf{f} ges genom att ta produkten mellan Cauchys spänningstensor, $\boldsymbol{\sigma}$, och den utåtriktade normalen för arean som betraktas.

$$\mathbf{f} = \boldsymbol{\sigma} \hat{\mathbf{n}} = \eta \left(\begin{array}{ccc} 2 \frac{\partial v_x}{\partial x} & \left(\frac{\partial v_x}{\partial y} + \frac{\partial v_y}{\partial x} \right) & \left(\frac{\partial v_x}{\partial z} + \frac{\partial v_z}{\partial x} \right) \\ \left(\frac{\partial v_y}{\partial x} + \frac{\partial v_x}{\partial y} \right) & 2 \frac{\partial v_y}{\partial y} & \left(\frac{\partial v_y}{\partial z} + \frac{\partial v_z}{\partial y} \right) \\ \left(\frac{\partial v_z}{\partial x} + \frac{\partial v_x}{\partial z} \right) & \left(\frac{\partial v_z}{\partial y} + \frac{\partial v_y}{\partial z} \right) & 2 \frac{\partial v_z}{\partial z} \end{array} \right) \hat{\mathbf{n}} \quad (19)$$

Här är η fluidens kinematiska viskositet.

2.5 Resonemang runt det numeriska

Det ekvationssystem som ska lösas är

$$Ax = b \quad (20)$$

Om en lösning har konvergerat eller inte testas med

$$\rho \|M^{-1}(b - Ax)\| < tol \|M^{-1}b\| \quad (21)$$

där $\rho > 1$ och $tol < 1$. För de iterativa lösarna är M en 'preconditioner'. Detta bygger på att lösa ekvationssystemet (20) är ekvivalent med att lösa följande ekvationssystem:

$$MAx = Mb \quad (22)$$

För de direkta lösarna som finns inbyggda i COMSOL ges M av LU-faktoriseringen till koefficientmatrisen A . Om vi kallar den exakta lösningen till systemet för \hat{x} fås att $A\hat{x} = b$. *Residualen* r beräknas som

$$r = A\hat{x} - Ax = A(\hat{x} - x) \quad (23)$$

Det som ska vara litet är skillnaden mellan \hat{x} och x . För att få en uppskattning på denna skillnad tas normen av

$$\hat{x} - x = A^{-1}r \quad (24)$$

vilket blir

$$\|\hat{x} - x\| = \|A^{-1}r\| \leq \|A^{-1}\| \|r\| \quad (25)$$

Storleken på $\|r\|$ kan vara runt 10^{-6} medan storleken på $\|A^{-1}\|$ lätt vara 10^{10} . Så bara för att ekvation (25) är uppfylld behöver inte skillnaden mellan den approximerade lösningen x och den exakta lösningen \hat{x} vara liten. För att komma runt detta används en ekvation av typen (21). Hur M väljs sköts av programmet.

2.6 Tillvägagångssätt

Den kommersiella programvaran COMSOL Multiphysics 3.4 användes för modelleringen. Valet föll på denna programvara då det fanns tillgängligt på universitetet, geometrin kan definieras via ett grafiskt användargränssnitt samt att nätgenerator och ett antal olika lösare för numeriska problem finns inbyggda i detta program. Att börja från ingenting och skriva nätgeneratorer, ekvationslösare, integratorer med mera är ett arbete som inte skulle vara möjligt inom tidsramen, de tio veckor som detta projekt omfattar. Mer om nät (beräkningsnät) och ekvationsssystem som uppstår finns att läsa under sektion 2.3, "Navier-Stokes och FEM". Att geometrin kan modelleras grafiskt underlättar arbetet då det är komplexa geometrier som berörs.

Det naturliga valet att närma sig det här problemet, med utgångspunkt från de verktyg vi hade till hands, var "tillämparens" tillvägagångssätt. Vi valde att arbeta utifrån det vi kunde prestera med hjälp av de inbyggda verktygen i COMSOL. Med detta tillvägagångssätt betraktades beräkningsprogrammet som en svart låda som tar indata från användaren och genererar ett svar. Vägen från indata till svar, det vill säga vad de numeriska lösarna gör och hur de betar sig ligger utanför detta projekts omfattning. Målsättningen har varit att generera resultat som kan tolkas fysikaliskt och vara praktiskt tillämpbara.

Den genomgående tanken i arbetet har varit att försöka skapa en modell som endast tar motorns varvtal som parameter och därefter genererar hastigheter och tryck. Mer om modellerna som använts finns under sektion 3, "Modeller".

2.7 Mjukvara och hårdvara

Den mjukvara som använts är COMSOL Multiphysics 3.4, ett kommersiellt program som utvecklas av COMSOL AB i Stockholm. Programvaran finns tillgänglig för Windows, Unix/Linux och Mac OS X. COMSOL Multiphysics använder sig av *finita elementmetoden* (FEM) för att lösa differentialekvationer. Mer om FEM finns att läsa under sektion 2.3, "Navier-Stokes och FEM". Hårdvaran som funnits tillgänglig är:

- Noder i UPPMAX Grad-cluster:
2.66 GHz dual Intel Xenon quad core, 16 GB RAM - Scientific Linux
- Arbetsstationer:
2.4 GHz Intel single core, 2 GB RAM - Fedora Core 8
AMD Athlon 64 3000+ (1.8 GHz), 1 GB RAM - Fedora Core 8
- Laptop:
2.4 GHz Intel Core2 Duo, 2GB RAM - Mac OS X

Arbetsstationerna och laptopen har använts för att bygga modellerna och för postprocessning av resultaten. Noderna i beräkningsclustret har använts för att utföra själva beräkningarna. Beräkningarna har körts i “batch mode” där vi har laddat upp modellerna lokalt till noderna och sedan startat beräkningarna från terminalen. BASH-scriptet, `com_run` som använts för att starta beräkningarna återfinns i bilagan.

3 Modeller

Under arbetets gång har flera olika modeller vuxit fram. Nya modeller har skapats allt eftersom tidigare modeller har funnits otillräckliga.

3.1 Modeller i 2D

Då målet med projektet är att hitta en enkel modell har flera 2-D modeller undersökts. Ett snitt, med normal i rotoraxelns riktning, gjordes genom luftkanalerna på rotor och stator. I försöken gjordes antagandet att rotorernas och statorernas luftkanaler befann sig mitt för varandra i z-led. Modellerna i 2D är en grov reduktion av verkligheten för de geometrier som är berörda i detta projekt. Detta beror på att snitten ser olika ut beroende på var längs axeln de läggs, d.v.s. det finns ingen klar axiell symmetri som kan motivera 2D-snitten. Fördelarna med att modellera i 2D är att det är illustrativt och att ekvationssystemen som ska lösas blir mycket mindre. Motsvarande upplösning i 3D är med dagens hårdvara inte möjlig.

3.1.1 Statisk 2D-modell

I denna modell har rotorernas och statorernas luftkanaler ritats ut i detalj, se figur 7. I den här modellen bortses från effekterna av rotorernas axel och balkar för att endast undersöka luftflödet genom kylkanalerna. Hastigheten som luften har när den kommer in i rotorernas luftkanaler ansätts som ett randvillkor på den inre randen. Detta baseras på mätdata från en existerande modell. Längs den inre randen ansätts ett homogent radiellt inflöde trots att detta är en förenkling av hur det riktiga flödet ser ut.

3.1.2 2D-modell med roterande beräkningsnät

Samma geometri som för den statiska 2D modellen används i detta fall. Dock roterar rotorernas beräkningsnät. Flödes hastigheten ansattes på samma sätt som den statiska 2D-modellen. Användandet av det roterande beräkningsnätet gjordes för att statorn och rotorernas kylkanaler ändrar position relativt varandra då maskinen är i drift. Dessutom påverkar maskinens rotationshastighet i hög grad luftflödet i kylkanalerna. Genom att beräkningsnätet roterar kan rotorernas geometri följa med och på så vis illustrera hur flödet ändras med tiden.

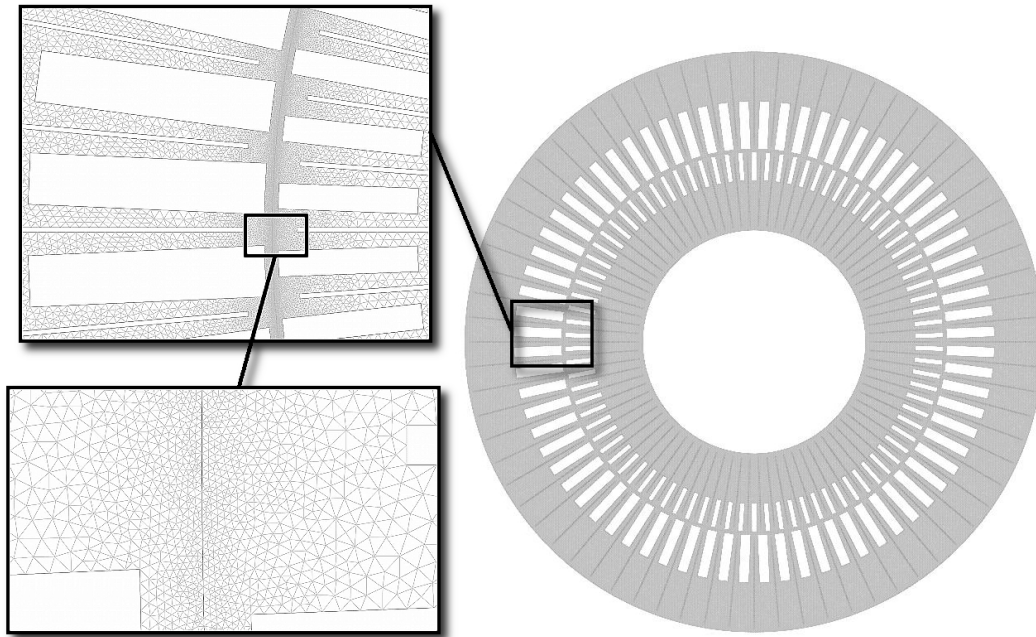


Figure 7: Högupplöst modell av rotor och stator. Man kan skönja gränsen mellan roterande nät och fixt.

3.1.3 Statisk 2D-modell med maskinstomme

För att få en bild av hur luftflödet påverkas av maskinstommen gjordes en enkel modell av detta. Maskinstommen ritades upp och statorns geometri placerades däri. För att hålla nere storleken på modellen modelleras rotorn här som en slät cylinder. Längs den inre randen i figur 8 ansätts ett rotationsflöde i kombination med ett radiellt inflöde. Det radiella inflödet beskriver motorns fläktverkan, rotationsflödet uppkommer för att rotorn roterar. Längst upp i maskinstommen sätts ett utflöde. Matematiskt sätter man normalkraften till noll. Här frångår modellen verkligheten något. I den verkliga maskinstommen sitter utblåset längst bak i maskinen, i den verkliga motorn blåser det alltså inte ut längs hela axeln som det gör i den här modellen. Däremot visar denna modell på hur ett asymmetriskt utflöde påverkar det övriga flödet i statorn.

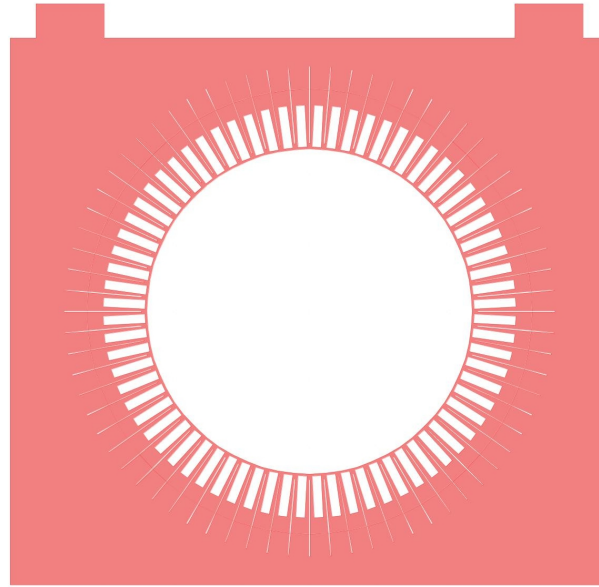


Figure 8: 2D-geometrin för stator och maskinstomme. Utblåsen sitter högst upp i bild

3.2 Modeller i 3D

Ett tillkortakommande med de tvådimensionella modellerna är att z -komponenten av hastigheten inte inkorporeras i lösningen. Detta gör det omöjligt att uppskatta den fläktverkan motorn har. Detta är viktigt för att kunna uppskatta friktionsförlusterna. Därför har även ett flertal 3D-modeller undersökts. Förhoppningen är att kunna uppskatta hur mycket luft motorn suger i sig vid ett visst varvtal. Detta kan sedan användas för att beräkna den effekt motorn använder för att vispa runt luften. För att hålla storleken på beräkningarna har luftkanalernas inverkan helt utelutits. Tanken var att ersätta luftkanalerna med ett poröst material med samma flödesmotstånd som luftkanalerna skulle ge upphov till.

3.2.1 3D-modell med roterande beräkningsnät

En första ansats var att modellera rotorns axel och balkar enligt ritning. Dessa omsluts i en cylinder som motsvarar rotorns inre radie. Denna cylinder sträcker sig längre ut i axiellt led än vad axeln gör. Detta för att fungera som insugsträtt. Längst ner på denna cylinder fogas ytterligare en cylinder som ska fungera som rotorns och statorns luftkanaler. I dessa områden kan man använda ett poröst material. I detta fall flödar dock luften helt obehindrat genom luftkanalerna. I det område som roterar används ett roterande beräkningsnät och i det område som är stillastående används ett fixerat beräkningsnät. Se figur 9

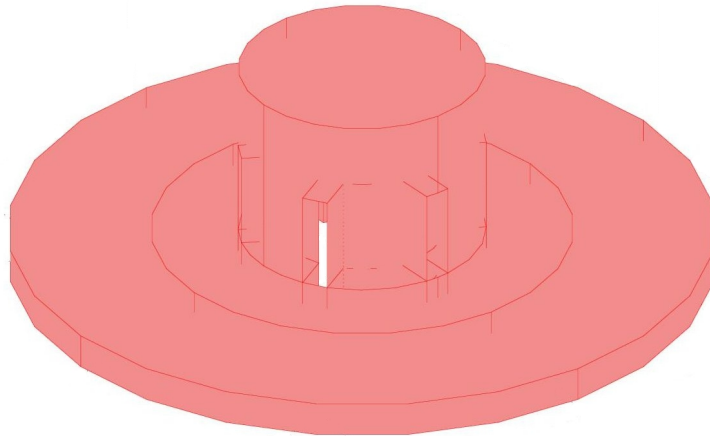


Figure 9: Geometrin för 3D-modellen med roterande beräkningsnät.

3.2.2 Statisk 3D-modell med ansatta Coriolis- och centrifugalkrafter

För att komma bort från tidsberoendet och det roterande beräkningsnätet används i den här modellen i stället ett roterande referenssystem. Det roterande referenssystemet används för att beräkna luftflödet i rotorn och ett inertialsystem för statorn. På detta vis hålls beräkningsnäten fixa relativt varandra och man kan lösa för ett stationärt fall. Man behåller effekten av rotation och centrifugalkraften blir den ingrediens som skapar själva luftflödet. Därför kan man vid insugen ansätta en 'öppen rand' d.v.s. normalkraften vid den ytan sätts till noll. Geometrin är den samma som för 3D-modellen med det roterande beräkningsnätet.

3.2.3 Statisk 3D-modell 'tårtbit' med ansatta Coriolis- och centrifugalkrafter

I och med frångången av det roterande beräkningsnätet kan man ytterligare utnyttja symmetrin. Detta reducerar storleken av beräkningarna med en sjättedel. På så vis reduceras minnesanvändning och tiden det tar att lösa systemet. Denna modell liknar modellen ovan fast med andra randvillkor. Modellen är uppbyggd av tre domäner, a , b och c . I domänen b är Coriolis- och centrifugalkrafter ansatta. Rand A och B är av typen öppen rand, övriga ränder är 'no-slip'.

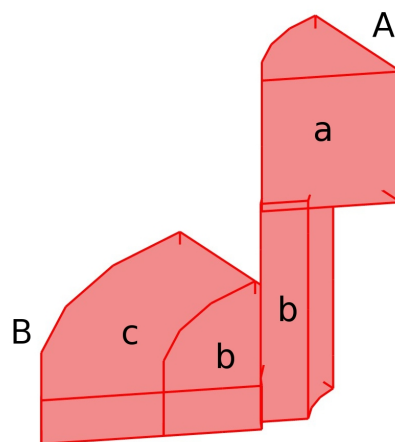


Figure 10: Geometrin för 3D-'tårtbit' med Coriolis- och Centrifugalkraft.

3.2.4 Statisk 3D-modell platt 'tårtbit'

Denna modell saknar den 'tratt' som finns i den tidigare modellen. Den har däremot rätt höjd på luftkanalen och är utskuren för att motsvara en två halva plåtsektioner och luftkanalen dem emellan. De randvillkor som ansätts är öppen rand längs ränderna A , B , C och D och 'no-slip' på alla övriga ränder.

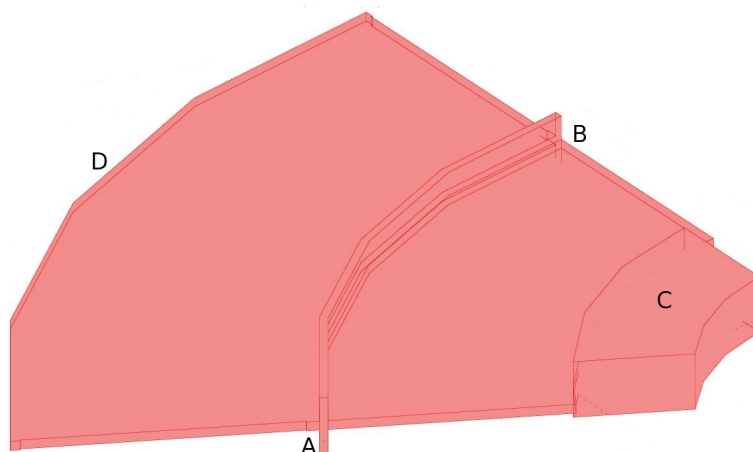


Figure 11: En 'tårtbit' plåtsektion.

3.2.5 Statisk 3D-modell med roterande väggar som randvillkor

Ett annat sätt att ta sig an problemet gjordes genom att ansätta randvillkor som motsvarar rörliga väggar i ett statiskt system. Längs med de ränder som roterar ansätter man hastigheten $\mathbf{v}_{\text{rand}} = \boldsymbol{\omega} \times \mathbf{r}$.

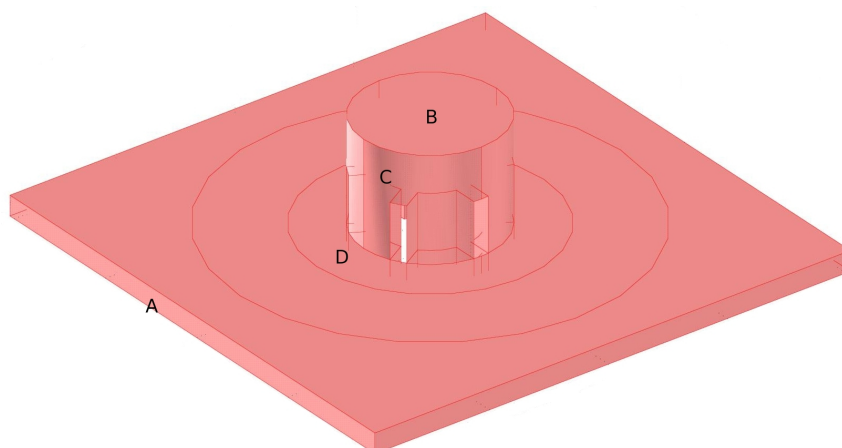


Figure 12: Geometri för modellen med rörliga väggar.

De randvillkor som ansätts är öppen rand vid A och B , rörlig vägg längs cylinderns mantelyta, C , och alla ränder som innesluts av denna. Rörlig vägg ansätts också vid D , och D :s motsvarighet på "undersidan" (ej synlig i bild). Övriga ränder har 'no-slip'-villkor.

4 Resultat

4.1 Modeller i 2D

Nedan visas resultat från simuleringarna av 2D-modellerna.

4.1.1 Statisk 2D-modell

I denna modell roterar inte rotorn relativt statorn. Ett inflöde är ansatt vid den inre randen på rotorn.

Table 1: Data - Statisk 2D-modell

Egenskap	Värde	Lösarinställningar		Beräkningsnät	
Inflödes hastighet rpm	18 [m/s]	Lösare	UMFPACK	Element	52160
	0 [1/min]	Typ	Stationär	Frihetsgrader	265212
		Tolerans	10^{-3}	Basfunktioner	P2-P1
		ρ	400		

Här är ρ feluppskattningsfaktorn. Se ekvation (21).

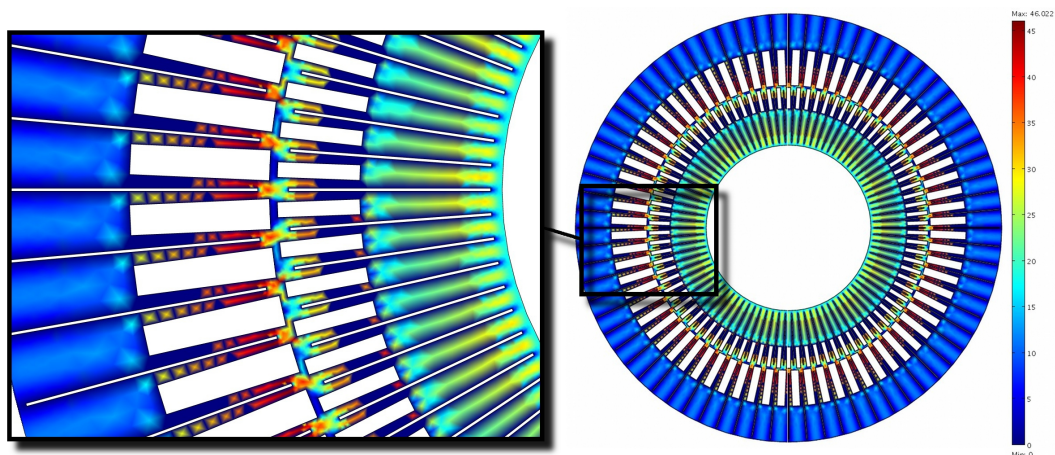


Figure 13: Här ses magnituden av hastighetsfältet för den statiska 2D-modellen.

4.1.2 2D-modell med roterande beräkningsnät

Här används ett roterande beräkningsnät.

Table 2: Data - Statisk 2D-modell med roterande beräkningsnät

Egenskap	Värde	Lösarinställningar		Beräkningsnät	
Inflödes hastighet rpm	18 [m/s]	Lösare	UMFPACK	Element	511540
	1000 [1/min]	Typ	Tidsberoende	Frihetsgrader	2408847
Tidsintervall	$[0, 10^{-6}]$	Tolerans	10^{-6}	Basfunktioner	P2-P1
		ρ	400		

Här är ρ feluppskattningsfaktorn. Se ekvation (21).

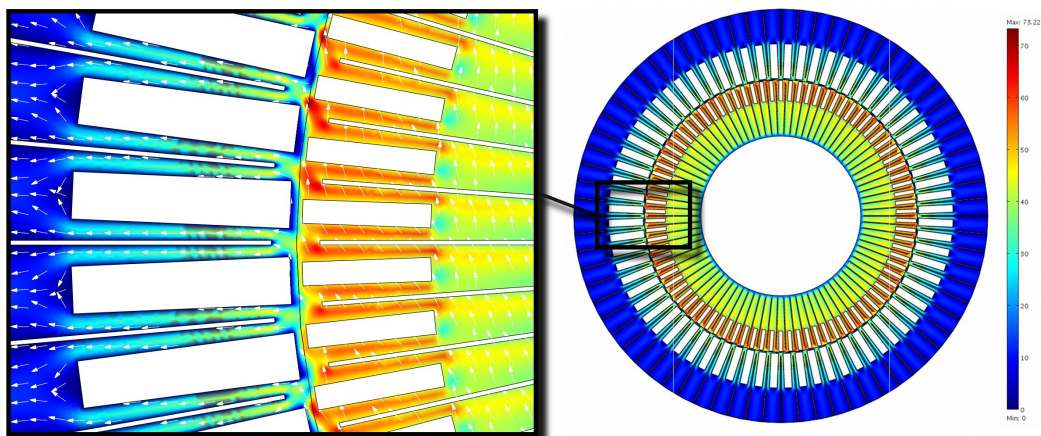


Figure 14: Bilden visar magnituden av hastighetsfältet för 2D-modellen med roterande beräkningsnät. Hastighetsvektorena ses utritade i den inzoomade delen.

4.1.3 Statisk 2D-modell med maskinstomme

Resultat för simulering på 2D-modellen med maskinstomme. Rotorn har modellerats som en slät cylinder med inflöde och cirkulationsflöde.

Table 3: Data - Statisk 2D-modell med maskinstomme

Egenskap	Värde	Lösarinställningar		Beräkningsnät	
Inflödeshastighet	25 [m/s]	Lösare	UMFPACK	Element	279757
rpm	1500 [1/min]	Typ	Stationär	Frihetsgrader	1403802
		Tolerans	10^{-6}	Basfunktioner	P2-P1
		ρ	400		

Här är ρ feluppskattningsfaktorn. Se ekvation (21).

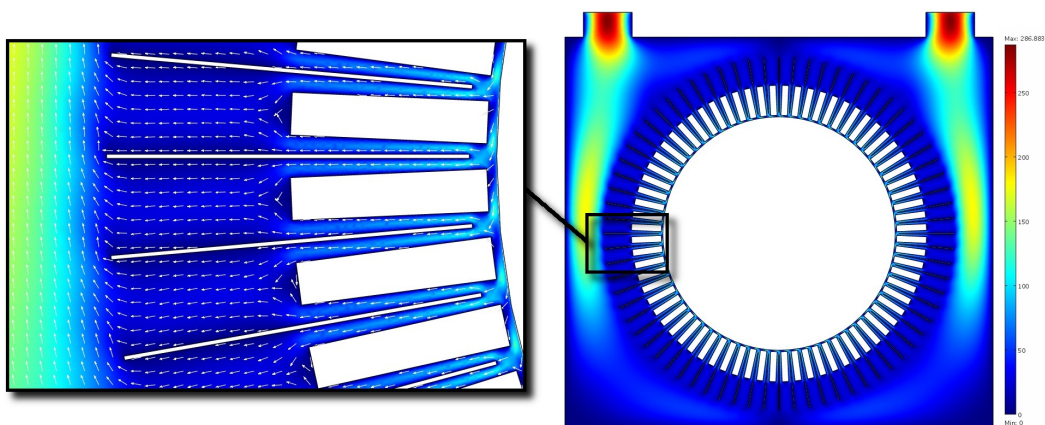


Figure 15: I bilden ses magnituden av hastighetsfältet för 2D-modellen med maskinstomme. Hastighetsvektorena ses utritade i den inzoomade delen.

4.2 Modeller i 3D

Nedan visas resultat från simuleringarna av 3D-modellerna.

4.2.1 3D-modell med roterande beräkningsnät

Denna modell övergavs då det inte gick att hitta en lösning.

4.2.2 Statisk 3D-modell med ansatta Coriolis- och Centrifugalkrafter

Beräkningar gjordes på en 3D-modell där det ansattes Coriolis- och Centrifugalkrafter enligt vad som beskrivs i avsnitt 3.2.2.

Table 4: Data - Statisk 3D-modell med ansatta Coriolis- och Centrifugalkrafter

Egenskap	Värde	Lösarinställningar		Beräkningsnät	
Inflöde(beräknat)	0,2 [m^3/s]	Lösare	UMFPACK	Element	101581
rpm	1500 [1/min]	Typ	Stationär	Frihetsgrader	522638
		Tolerans	10^{-6}	Basfunktioner	P2-P1
		ρ	400		

Här är ρ feluppskattningsfaktorn. Se ekvation (21).

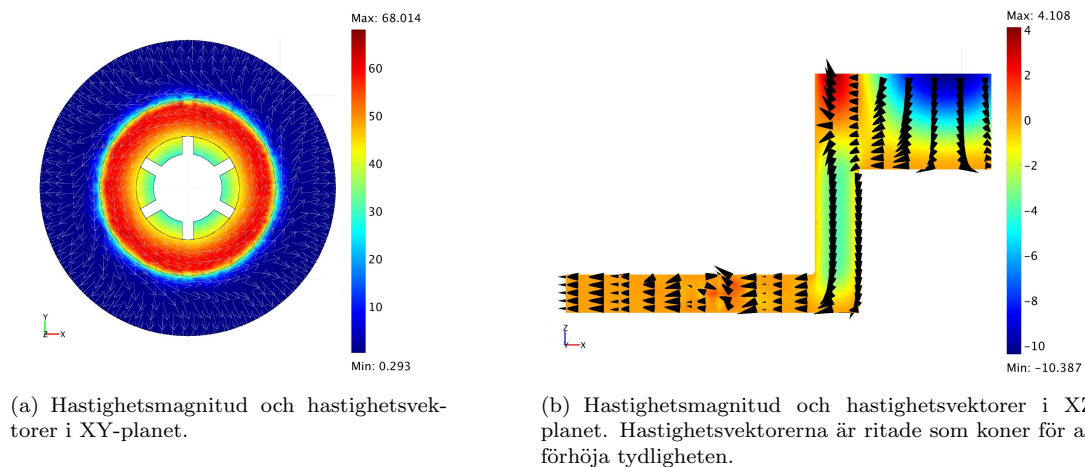


Figure 16: I figurerna ses hastighetsfältets magnitud och hastighetsvektorer utritade.

4.2.3 Statisk 3D-modell 'tårtbit' med ansatta Coriolis- och centrifugalkrafter

Resultat av 'tårtbits'-modellen där Coriolis- och centrifugalkrafter är ansatta i område b , enligt figur 10.

Table 5: Data - Statisk 3D-modell 'tårtbit' med ansatta Coriolis- och Centrifugalkrafter

Egenskap	Värde	Lösarinställningar		Beräkningsnät	
Inflöde (beräknat)	0,073 [m^3/s]	Lösare	UMFPACK	Element	14925
rpm	10 [1/min]	Typ	Stationär	Frihetsgrader	83036
		Tolerans	10^{-6}	Basfunktioner	P2-P1
		ρ	400		

Här är ρ feluppskattningsfaktorn. Se ekvation (21).

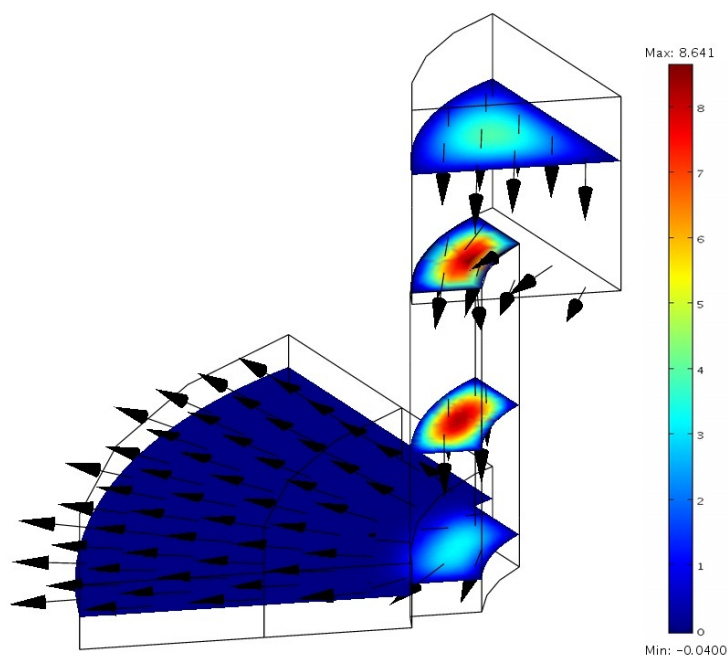


Figure 17: Resultatet för simuleringen syns ovan. Hastighetsfältets magnitud och normaliserade hastighetsvektorer.

4.2.4 Statisk 3D-modell platt 'tårtbit'

Resultat av simulering för den platta 'tårtbits'-modellen.

Table 6: Data - Statisk 3D-modell platt 'tårtbit'

Egenskap	Värde	Lösarinställningar		Beräkningsnät	
Inflöde(beräknat)	$3,3 \cdot 10^{-3}$ [m^3/s]	Lösare	UMFPACK	Element	58366
rpm	1500 [1/min]	Typ	Stationär	Frihetsgrader	1021466
		Tolerans	10^{-6}	Basfunktioner	P3-P2
		ρ	400		

Här är ρ feluppskattningsfaktorn. Se ekvation (21).

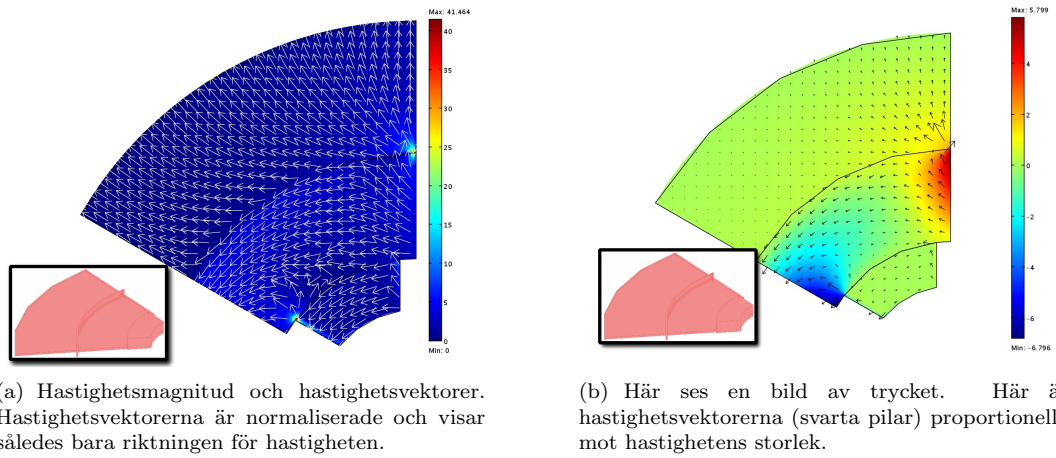


Figure 18: Hastighetsfältets storlek och trycket för "tärtbits"-modellen.

4.2.5 Statisk 3D-modell med roterande väggar som randvillkor

Table 7: Data - Statisk 3D-modell med roterande väggar

Egenskap	Värde	Lösarinställningar		Beräkningsnät	
Inflöde (beräknat)	$1,25 \cdot 10^{-3} [m^3/s]$	Lösare	UMFPACK	Element	46370
rpm	1800 [1/min]	Typ	Stationär	Frihetsgrader	242821
		Tolerans	10^{-6}	Basfunktioner	P2-P1
		ρ	400		

Här är ρ feluppskattningsfaktorn. Se ekvation (21).

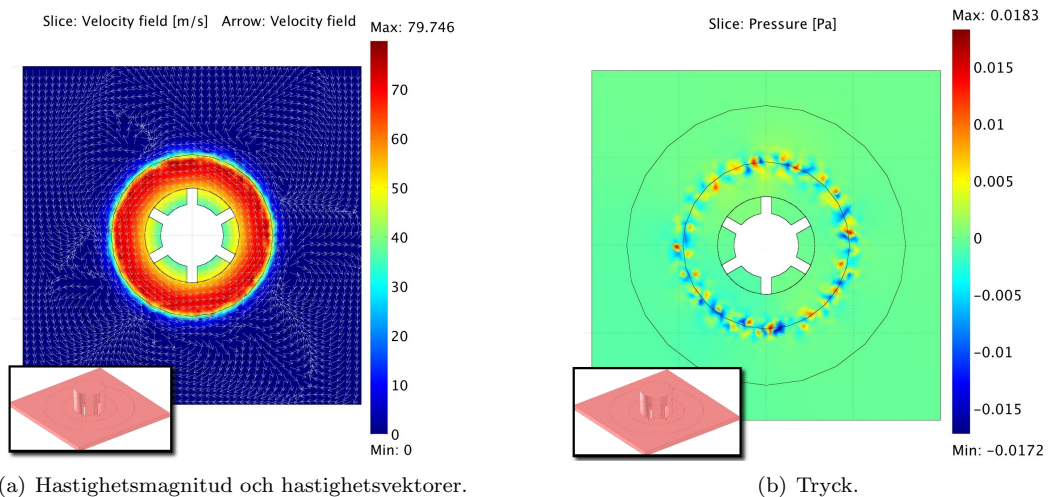


Figure 19: I figur a ses hastighetsfältets magnitud och hastighetsvektorer utritade. I figur b ses tryckets magnitud.

5 Diskussion

Att skapa en modell som genererar tillämpbara resultat visade sig väldigt svårt med det tillvägagångssätt som valts.

Ett annat tillvägagångssätt kan vara att titta på fysiken och formulera en starkare och mer specifik frågeställning att ha som utgångspunkt. Att börja med papper, penna och en formelsamling och ställa upp alla fysikaliska och kompatibilitets-samband som ska gälla bör ge en bättre förståelse för vad det är som eftersträvas.

Genom att studera de olika ekvationslösare som används kan problem som uppstår eventuellt isoleras till dessa. Då det går att göra flera inställningar för de olika lösarna för att öka stabilitet och prestanda, vilka ibland kan stå i motsats till varandra, kan kunskap om dessa vara väsentlig. I nuläget vet vi inte om vissa av infallsvinklarna vi använt oss av kunnat bära frukt om lösarinställningarna optimerats. Att veta hur lösarna fungerar ger också en fingervisning om vilken minnesåtgång som kan krävas.

Den största svårigheten i arbetet har legat i att fluiden vi arbetat med är luft. Då luft har väldigt låg viskositet ger detta upphov till väldigt stora hastighetsgradienter i områden nära väggar med no-slip-randvillkor. Detta betyder att luftens hastighet är noll. I områden där stora gradienter finns måste ett tätt beräkningsnät användas för att korrekt uppskatta dessa. Detta har gett upphov till att ekvationssystemen som ska lösas blir väldigt stora. Att det just är luftens beteende som är boven i dramat har kunnat konstateras då modellerna har fungerat med andra värden på densitet och viskositet. I många fall har modellen bara berört en väldigt liten del av geometrin men ändå gett upphov till ekvationsystem som varit så stora att 16 GB RAM-minne varit för litet. Hur mycket arbetsminne som kan behövas kan inte vi uppskatta men vi kan som sagt konstatera att 16 GB inte räckte. Om det krävs 17 GB, 128 GB eller mer är en fråga som står öppen.

I samtliga modeller har fluiden antagits vara inkompressibel för att underlätta simuleringarna. Fluiden har också antagits ha samma temperatur i hela området, trots att den ska verka som kylmedel. Har hastighetsfältet väl bestämts går det att nyttja hastighetsfältet för att beräkna värmeöverföringen.

5.1 Analys av modeller

Den statiska 2D-modellen användes för att få fram en flödesprofil för ett snitt genom rotorn och statorns kylkanaler. Varvtalet är satt till noll främst för att se om det gick att erhålla en lösning med den här komplicerade geometrin. Det har visat sig att det är svårare att få konvergens ju högre varvantal som använts i fallet med de roterande beräkningsnäten. Tittar man noga i figur 13 i avsnitt 4.1.1 kan man notera fluktuationer i hastighetsfältet. Detta beror på för få interpolationspunkter. Betraktas figur 7 i avsnitt 3.1 ser man att i de områden där det är kortast avstånd mellan två väggar endast finns två till fyra element i bredd. Detta är för grovt. Det är för hårdvaruintensivt att lösa modeller med roterande beräkningsnät med finare beräkningsnät än i ovan nämnda fall.

För att få en bild av hur luftflödet påverkas av maskinstommens geometri gjordes en simulering enligt avsnitt 4.1.3. Som syns i tabell 4.1.3 kan man använda betydligt fler element eftersom denna modell inte är tidsberoende och därmed kan lösas med en stationär lösare. Betraktas figur 4.1.3 kan man se att det inte uppstår samma fluktuationer i hastighetsfältet som i figur 7. Detta är en effekt av att ha tätare nät. I de områden där det är kortast avstånd mellan två väggar finns det i den här modellen upp till 15 element i bredd.

Ett roterande beräkningsnät i 3D fungerade inte. Ett flertal modeller med olika geometriska konfiguration och olika parametrar, samt lösare, prövades. Inga av dessa gav resultat. I stället testades ett tillvägagångssätt med roterande referenssystem. Detta har haft delad framgång. I vissa modeller har det gett rimliga hastighetsfält. Däremot har för grova förenklingar gjorts på geometrin i

de fall där detta har testats. I modellen i avsnitt 4.2.2 saknas, till exempel, de delar som hindrar luftens väg. Dessa syns i figur 7. Utöver detta är luftkanalen för hög. Förhoppningen var att denna modell skulle ge en övre uppskattning på hur mycket luft som motorn suger in.

För att få upp tätheten i beräkningsnätet men ändå hålla minnesåtgången nere gjordes ytterligare utnyttjanden av de symmetrier som fanns. På detta vis kunde modellerna reduceras till "tårtbitar" som var en sjättedel av den ursprungliga geometrins storlek. Det uppstod då problem med hur symmetriränderna skulle hanteras. I modellen i avsnitt 4.2.3 bortsågs all verkan av dessa ränder då de gavs 'no-slip'-villkor. Detta betyder att hastigheten hos fluiden är noll på randen. I modellen 4.2.4 sattes öppen rand på symmetriränderna. Man kan se att vid den 'bakre' randen får man ett backflöde. Detta kan först tyckas konstigt men är rimligt eftersom fluider rör sig från områden med högt tryck mot områden med lågt tryck. I figur 18(a) kan noteras ett högre tryck vid den 'bakre' axeln. Även detta är rimligt eftersom ett föremål som rör sig i luft skapar högre tryck på 'framsidan' än på 'baksidan'.

En mer direkt reduktion av verkligheten är att ansätta rörliga väggar som randvillkor. Detta har gjorts i modell 4.2.5. För att illustrera vilken effekt omgivningen får på luftflödet gjordes en "låda" med utlopp uppåt som fick innesluta det som representerar statorn. Beräkningarna konvergerade men flödesprofilen ser konstig ut då luft verkar tryckas ut genom 'insuget', där luft rimligtvis bara borde komma in.

6 Slutsatser

Vill man ha luftflödet som funktion av varvtal måste en 3D-modell användas eftersom flödet till stor del beror av alla tre rumskoordinater. Det finns ingen 2D-modell som kan simulera alla de intressanta 3D fenomen som uppstår. Generellt kräver 3D väldigt mycket mer minnesresurser därför behövs en djupare studie i optimering av beräkningsnät.

Om man istället använder mätdata, från t.ex. en prototyp, kan en 2D-modell med fördel användas för att bestämma luftflödet i rotor och stator. Luftflödet kan i sin tur användas för att uppskatta värmeöverföringen. En frågeställning att gå vidare med är om det verkligen behövs ett så kraftigt luftflöde för att kyla maskinen?

Med tillgång till hårdvara med mer arbetsminne kan mer komplex geometri användas i 3D-modellerna. Med geometrier som ligger närmare verkligheten än de som använts här kan det tänkas att konkreta resultat kan erhållas. I detta projekt har uteslutande direkta lösare använts, dessa kräver mycket arbetsminne. Man bör istället använda iterativa lösare vilka kräver betydligt mindre arbetsminne. Dessa är dock inte lika stabila som de direkta lösarna.

En modell som ger en lösning till en frågeställning kan vara svårtolkad och påvisa fenomen som nödvändigtvis inte uppstår i verkligheten. Därför bör en kontinuerlig jämförelse mellan modell och verklighet göras för att se om det finns relevans i de resultat som erhålls från beräkningarna. I vissa fall kan modellen påvisa fenomen som kan tyckas konstiga men som verkligen uppstår, i andra fall är resultatet felaktigt för det man verkligen vill simulera.

References

- [1] John Burkardt. *Finite Element Treatment of the Navier–Stokes Equations*. School of Computational Science Florida State Univ., 2005.
- [2] Ramon Codina. Numerical solution of the incompressible navier-stokes equations with coriolis forces based on the discretization of the total time derivative. *J. Comput. Phys.*, 148(2):467–496, 1999.
- [3] Wei Michael Lai, David Rubin, and Erhard Krempl. *Introduction to continuum mechanics*. Pergamon Press, Oxford :, 3. ed. edition, 1993.
- [4] Stephen T. Thornton and Jerry B. Marion. *Classical dynamics of particles and systems*. Brooks/Cole - Thomson learning, Belmont, Calif. :, 5. ed. edition, 2004.
- [5] Richard Wait. *FINITE ELEMENT ALGORITHMS AND APPROXIMATIONS*. Dept. of Scientific Computing, Uppsala Univ., Uppsala, Sweden, Jan. 2007.

APPENDIX

com_run: Ett script som använts för att starta stora beräkningar på grad-clustret. Det skickar mail när beräkningen är startad och när beräkningen är avslutad. Lösar inställningar ska läggas i solerParam.m se exempel nedan.

```
#!/bin/bash

#Ett script som startar comsol batch $1 där $1 är
#m-filen som ska köras.
#OBS solverParam.m måste innehålla lösar
#inställningar från COMSOL
#Mailar personer i MAIL_LIST när jobbet startas
#och avslutas. Därefter skickas lösningen via scp
# till en plats. Tänk på att man
# måste skapa ssh-nycklar, dessa ska läggas i
# /home/user/.ssh/authorized_keysgen.

MAIL_LIST=person1@mail.com\ person2@mail.com
echo "flload $1.mph fem" > toSolve_$1.m
cat solverParam.m >> toSolve_$1.m
echo "flsave SOLVED_$1 fem" >> toSolve_$1.m

#Kontrollera om comsol redan körs.
if [ "$2" != 'fast' ]
then
    while [ -n "$(ps -A | grep comsol)" ]
    do
        sleep 3
    done
fi

echo "Startar comsol med script $1">message$HOSTNAME.txt
echo "Jobbet startades $(date)">> message$HOSTNAME.txt
mail -s "$HOSTNAME jobb:$1 STARTED" $MAIL_LIST< message$HOSTNAME.txt
/usr/local/comsol34/bin/comsol -np 8 -tmpdir /home/krro5599/comsol_tmp/ batch toSolve_$1

if [ -e SOLVED_$1.mph ]; then
    echo "Jobbet blev klart $(date) ">>message$HOSTNAME.txt
    mail -s "$HOSTNAME jobb:$1 FINISHED" $MAIL_LIST< message$HOSTNAME.txt
    scp SOLVED_$1.mph user@remotehost:/PATH/. > /dev/null
else
    echo "Det har blivit ett fel! $(date)">> message$HOSTNAME.txt
    mail -s "$HOSTNAME jobb:$1 FAILED" $MAIL_LIST< message$HOSTNAME.txt
fi
rm message$HOSTNAME.txt;
rm toSolve_$1.m

exit 0
```

solverParam.m: Fil som används av com_run för att köra en beräkning på grad-clustret.
 Lösarinställningar kan hämtas ur comsol via 'solver settings'.

```
% Extend mesh
fem.xmesh=meshextend(fem);

% Constants
%fem.const = {'T','20[degC]', 'INFLOW','4.5[m/s]','rpm','1000[1/min]'};

%solve da shit
fem.sol=femstatic(fem, ...
    'blocksize',1000, ...
    'symmetric','off', ...
    'solcomp',{'w','v','u','p'}, ...
    'outcomp',{'w','v','u','p'}, ...
    'nonlin','on', ...
    'hnlm','on', ...
    'uscale','none');

fem0=fem;
```