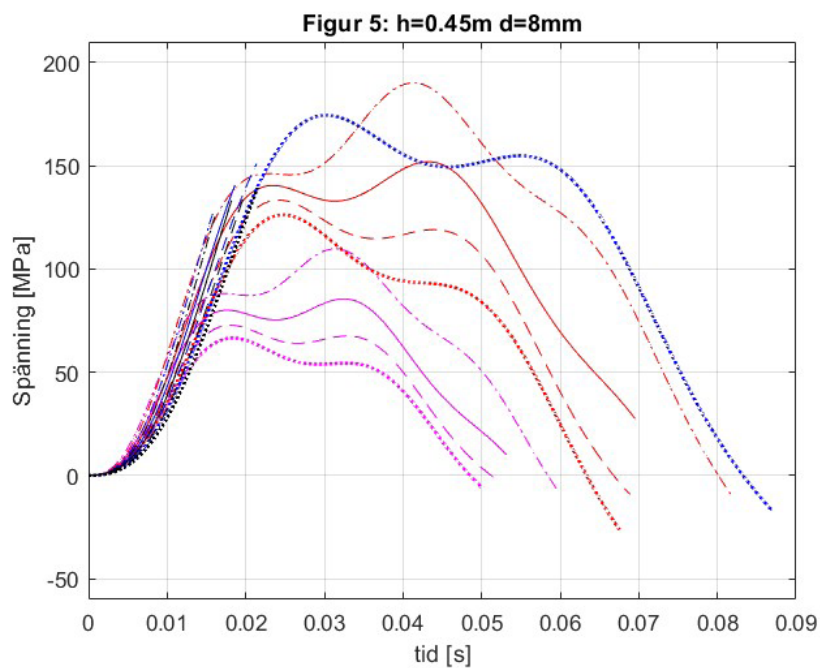




IMPLEMENTERING AV BERÄKNINGSMETODER FÖR ANALYS AV GLASPANELER BELASTADE AV STÖTLASTER

OSKAR LAGERGREN

*Bachelor's Dissertation
at Structural Mechanics*

DEPARTMENT OF CONSTRUCTION SCIENCES
DIVISION OF STRUCTURAL MECHANICS

ISRN LUTVDG/TVSM--25/6003--SE (1-57) | ISSN 0281-6679

BACHELOR'S DISSERTATION

IMPLEMENTERING AV BERÄKNINGSMETODER FÖR ANALYS AV GLASPANELER BELASTADE AV STÖTLASTER

OSKAR LAGERGREN

Supervisor: **LINUS ANDERSSON**, Assistant Professor, Div. of Structural Mechanics, LTH, Lund.

Examiner: Professor **KENT PERSSON**, Div. of Structural Mechanics, LTH, Lund.

Copyright © 2025 Division of Structural Mechanics
Faculty of Engineering (LTH), Lund University, Sweden.

Printed by V-husets tryckeri LTH, Lund, Sweden, July 2025 (*PI*).

For information, address:

Div. of Structural Mechanics, LTH, Lund University, Box 118, SE-221 00 Lund, Sweden.

Homepage: <http://www.byggmek.lth.se>

Abstract

This study investigates the dynamic response of a glass pane under impact loading utilizing modal analysis and Newmark's method. The benefits and drawbacks of modal analysis as well as Newmark's method in this context are also investigated. The study aims to simulate the testing of glass panes according to the testing method described in SS-EN 12600 where an impact device is swung in a pendulum motion towards the glass pane. Linear models of the system are used. The results indicate that modal analysis is sufficient for linear models of the system. Newmark's method is, however, necessary for the implementation of non-linear models. The results also highlight the need for further development of the model. More specifically, so that it can handle the situation of loss of contact between the glass pane and the impact device.

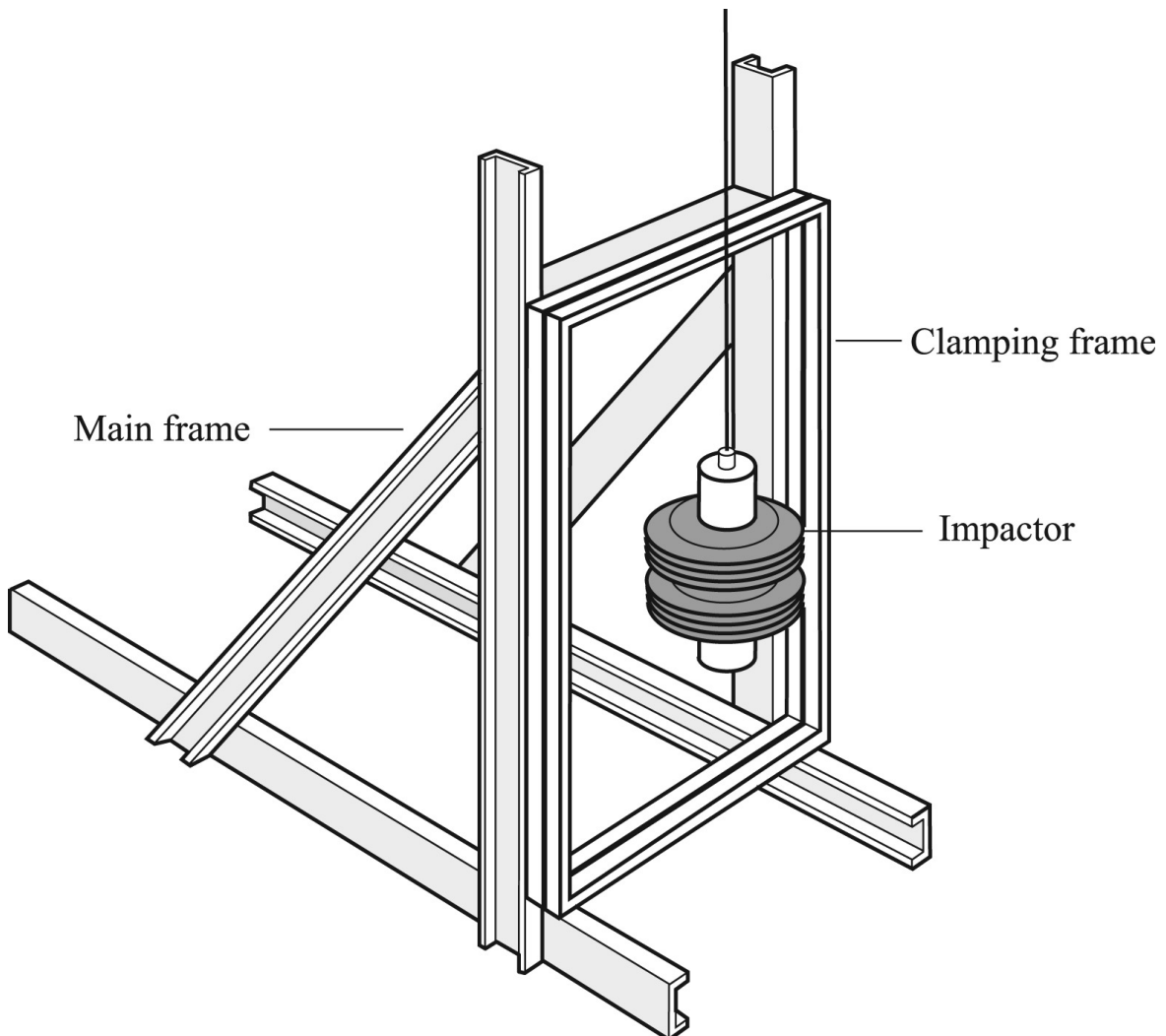
Innehåll

Abstract	I
Innehåll	III
1 Introduktion	1
2 Metod	5
2.1 Modell	5
2.1.1 Kontaktvillkor	7
2.1.2 Randvillkor	7
2.1.3 Bidragande massa	8
2.2 Modal analys	8
2.2.1 Odämpad svängning	8
2.2.2 Dämpad svängning	10
2.3 Newmarks metod	11
2.4 Spänningar i glaset	12
2.5 Program	13
2.5.1 Program 1: modalnd	15
2.5.2 Program 2: modalndsigma	16
2.5.3 Program 3: modald	17
2.5.4 Program 4: modalndsigma	17
2.5.5 Program 5: newmark	18
3 Resultat	21
3.1 Program 1: modalnd	22
3.2 Program 2: modalndsigma	23
3.3 Program 3: modald	31
3.4 Program 4: modalndsigma	32
3.5 Program 5: newmark	39
4 Diskussion	41
Litteratur	43
Appendix	45

1 Introduktion

Bakgrund

Vid dimensionering av glas i byggnader ska glasytor tåla dynamisk påverkan av en människa enligt plan- och bygglagen. Boverket definierar dynamisk påverkan av en människa i detta sammanhang som krafter som uppstår av personer i snabb, kraftig rörelse som hopp, språng, fall eller liknande Boverket [1]. Boverket specificerar inte hur denna belastning skall modelleras utan hänvisar till provningsmetoden SS-EN 12600. Denna provningsmetod går ut på att släppa en impaktor i en pendelrörelse mot en glastruta som är monterad på en ram SIS [2], se figur 1.1 Andersson m.fl. [3]. Impaktorn består av två däck samt två vikter. Vikterna väljs så att impaktorns totala



Figur 1.1: Provningsanordning SS-EN 12600

massa är 50 kg. Under stöten ska vikterna på impaktorn inte vara i kontakt med glaset. Dimensionerna på ramen kan ses i figur 1.2. Impaktorn höjs upp till en specificerad

höjd för att sedan släppas i en pendelrörelse mot glaset. Höjden varierar beroende på vilken typ av klassificering som glaset skall testas för. I tabell 1.1 kan ses vilken höjd som svarar mot en viss klassificering. För att en viss klassificering ska uppnås måste glaset hålla för ett test på varje höjd upp till och med den avsedda klassificeringens höjd.

Klassificering	Fallhöjd [mm]
3	190
2	450
1	1200

Tabell 1.1: Fallhöjder

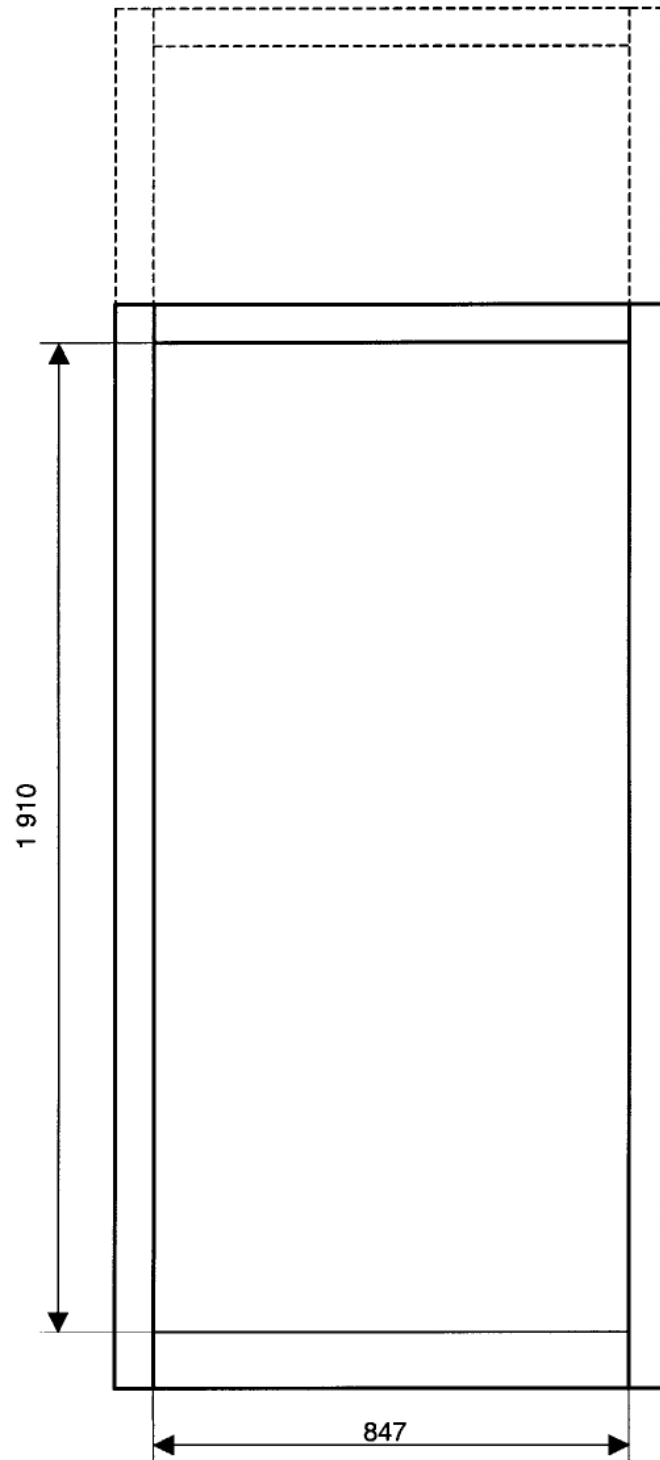
Syfte

Syftet med denna rapport är att ta fram modeller för glasrutans dynamiska respons samt implementera dynamiska analysmetoder och utföra beräkningar utifrån de modeller som tagits fram. Utifrån denna respons kan sedan spänningen i glaset beräknas. Rapporten har även som syfte att utforska fördelar och nackdelar med de modeller som tas fram samt de analysmetoder som implementeras. Utöver detta är syftet även att undersöka inverkan av glasets dimensioner samt fallhöjden på impaktorn i form av en parameterstudie.

Begränsningar

Denna rapport utgår ifrån en förenklad modell av glasrutan i form av en balk. Denna förenkling gör det möjligt att ta fram modeller, utföra parameterstudier och studera principerna bakom glasrutans dynamiska respons. Förenklingen är dock för stor för att kunna få tillförlitliga siffror på utböjning och spänning. Sekundärstrukturen antas vara helt styv. Även detta är en förenkling.

Ytterligare en begränsning med rapporten är att den utgår från linjära modeller. Det går att implementera icke-linjära modeller med hjälp av Newmarks metod som diskuteras nedan. Detta är emellertid utanför rapportens omfattning.



Figur 1.2: SS-EN 12600, ramens dimensioner [mm]

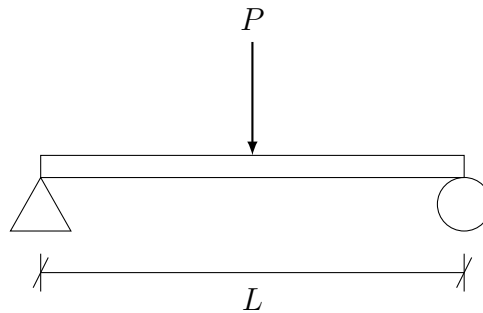
2 Metod

2.1 Modell

Modellering av glasrutan

I denna rapport kommer glasrutan att modelleras som en fritt upplagd balk, se figur 2.1. Balkens dimensioner kommer att frångå dem som syns i figur 1.2 då syftet är att undersöka principen av att modellera stötlaster mot glas, inte att nödvändigtvis efterlikna provningsmetoden i så hög grad som möjligt. Genom användning av platteori hade en större grad av efterlevnad kunnat uppnås men detta är utanför rapportens omfattning. En effekt av att modellera glasskivan som en balk är att förskjutningen av glaset i den punkt där impaktorn stöter i är linjärt beroende av den kraft som impaktorn verkar på glasrutan med. Om impaktorns kraft modelleras som en punktlast i balkens mitt beskrivs glasets förskjutning i den punkten med följande samband Heyden m. fl. [4]:

$$u = \frac{PL^3}{48EI} \quad (2.1)$$



Figur 2.1: Modellering av glasrutan

Ekvation (2.1) kan skrivas om på följande sätt:

$$P = ku$$

Med

$$k = \frac{48EI}{L^3} \quad (2.2)$$

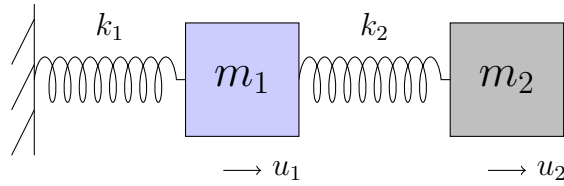
Där E är materialets elasticitetsmodul, L är balkens längd och I är tvärsnittets ytt-röghetsmoment. För rektangulära tvärsnitt kan I beräknas med följande samband:

$$I = \frac{bd^3}{12} \quad (2.3)$$

Där b är tvärsnittets bredd och d dess höjd.

Modellering av stöten

Modelleringen av händelseförloppet då impaktorn stöter i glaset görs i denna rapport med hjälp av en två-frihetsgrads-system. En frihetsgrad representerar glasets förskjutning i den punkt där impaktorn stöter i. Den andra frihetsgraden representerar impaktorns förskjutning från sitt jämviktsläge. Interaktionen mellan de två frihetsgraderna kan göras på flera olika sätt med varierande grad av efterlevnad av verkligheten samt beräkningskomplexitet. Det enklaste fallet kan ses i figur 2.2. I denna modell antas att materialresponserna från både glaset och impaktorn är linjär och dämpning beaktas inte. Den vänstra massan är i denna modell glasets bidragande massa (se rubrik 2.1.3) och



Figur 2.2: Odämpat system

den högra är impaktorns massa. Fjädrarna modellerar materialresponserna från glaset respektive impaktorn. Den vänstra fjäderns styvhet ges av ekvation (2.2). Den högra fjäderns styvhet kan bestämmas från empiriska undersökningar. Genom att ställa upp jämviktsvillkor för systemet i figur 2.2 erhålles följande ekvation:

$$\mathbf{m}\ddot{\mathbf{u}} + \mathbf{k}\mathbf{u} = \mathbf{0} \quad (2.4)$$

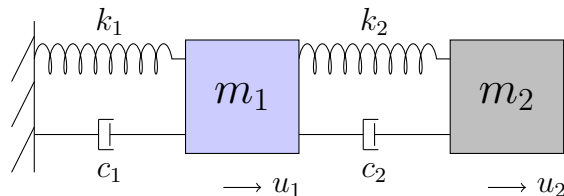
Med

$$\mathbf{m} = \begin{bmatrix} m_1 & 0 \\ 0 & m_2 \end{bmatrix} \quad \mathbf{k} = \begin{bmatrix} k_1 + k_2 & -k_2 \\ -k_2 & k_2 \end{bmatrix}$$

och

$$\mathbf{u}(t) = \begin{bmatrix} u_1(t) \\ u_2(t) \end{bmatrix} \quad \ddot{\mathbf{u}}(t) = \frac{d^2}{dt^2} \begin{bmatrix} u_1(t) \\ u_2(t) \end{bmatrix}$$

Om viskös dämpning tas i beaktning fås modellen som ses i figur 2.3. Dämpningen i



Figur 2.3: Dämpat system

systemet gör att en till term i jämviktsekvationen läggs till.

$$\mathbf{m}\ddot{\mathbf{u}} + \mathbf{c}\dot{\mathbf{u}} + \mathbf{k}\mathbf{u} = \mathbf{0} \quad (2.5)$$

Utseendet på $\mathbf{c}$ -matrisen kan variera beroende på vilken typ av dämpningsmodell som används i modellen. Den typ av dämpning som ses i figur 2.3 ger upphov till följande utseende på $\mathbf{c}$:

$$\mathbf{c} = \begin{bmatrix} c_1 + c_2 & -c_2 \\ -c_2 & c_2 \end{bmatrix}$$

2.1.1 Kontaktvillkor

Båda modeller som nämns ovan förutsätter att impaktorn är i kontakt med glaset. När impaktorn inte längre är i kontakt med glaset upphör deras interaktion och modellerna som beskrivits är då inte längre rimliga. Detta kriteriet kan, för det odämpade fallet, uttrycks matematiskt på följande sätt:

$$u_2 - u_1 \leq 0$$

Samma kontaktvillkor gäller för det dämpade fallet. Här yttrar sig villkoret som att summan av fjäderkraften från den högra fjädern och kraften från den högra dämparen är en tryckande kraft. Detta uttrycks matematiskt på följande sätt:

$$k_2(u_2 - u_1) + c_2(\dot{u}_2 - \dot{u}_1) \leq 0$$

Detta kriterium kan brytas innan dess att glaset har hunnit vända och är på väg tillbaka till sitt jämviktsläge. Om detta händer kommer glasets utböjning att fortsätta öka utan inverkan av impaktorn. Utböjningen fortsätter att öka tills dess att ett maximum har nåtts eller tills dess att impaktorn får kontakt med glaset igen varvid de interagerar med varandra igen.

2.1.2 Randvillkor

Randvillkoren till ekvation (2.4) samt (2.5) är på formen:

$$\mathbf{u}(0) = \mathbf{u}_0 \quad \dot{\mathbf{u}}(0) = \mathbf{v}_0$$

De initiala positionerna, $\mathbf{u}_0$, är båda noll då modellen börjar vara giltig precis när impaktorn kommer i kontakt med glaset.

$$\mathbf{u}_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Glasets initiala hastighet är även den noll då glaset är i vila innan impaktorn kommer i kontakt med det. Impaktorns initiala hastighet kan beräknas med hjälp av principen av energins bevarande. Den potentiella energi som impaktorn har innan den släpps är:

$$E_{pot} = mgh$$

Där m är impaktorns massa, g är gravitationsaccelerationen och h är höjden från vilken impaktorn släpps. Precis innan impaktorn stöter i glaset har impaktorn samma mängd energi fast nu i form av rörelseenergi.

$$E_k = \frac{mv^2}{2}$$

Genom att sätta dessa lika med varandra kan den initiala hastigheten beräknas.

$$\begin{aligned}\frac{mv^2}{2} &= mgh \\ v &= \sqrt{2gh} \\ \mathbf{v}_0 &= \begin{bmatrix} 0 \\ -\sqrt{2gh} \end{bmatrix}\end{aligned}\tag{2.6}$$

Minustecknet i ekvation kommer från den valda riktningen på frihetsgrad 2.

2.1.3 Bidragande massa

Vid övergången mellan balken i figur 2.1 och modellerna i figurerna 2.2 och 2.3 behövs massan för glaset justeras. Detta beror på att delar av balkens massa tas upp av dess stöd istället för att bidra till tröghet vid rörelse. För att justera massan används det värde som togs fram av Nyström [5].

$$m_1 = 0.486m_{glas}\tag{2.7}$$

2.2 Modal analys

Modal analys handlar om att uttrycka lösningen till ett system av linjära differentialekvationer som en linjärkombination av egenmoderna till systemet. Detta tillgängliggör exakta lösningar till systemet. Metoden är även beräkningsmässigt effektivt.

2.2.1 Odämpad svängning

Rörelseekvationen för systemet som syns i figur 2.2 är ekvation (2.4). För att lösa denna ekvation ansätts $\mathbf{u}$ på följande form.

$$\mathbf{u}(t) = \phi q(t)\tag{2.8}$$

Där ϕ är en tidsberoende vektor och $q(t)$ är en skalär funktion på följande form:

$$\begin{aligned}q(t) &= A \cos(\omega t) + B \sin(\omega t) \\ \ddot{q} &= -\omega^2 q\end{aligned}\tag{2.9}$$

Genom att sätta in ekvation (2.8) och (2.9) i (2.4) fås följande.

$$\begin{aligned}\mathbf{m}\phi\ddot{q} + \mathbf{k}\phi q &= \mathbf{0} \\ -\omega^2\mathbf{m}\phi q + \mathbf{k}\phi q &= \mathbf{0}\end{aligned}$$

Då $q(t) \neq 0$ för alla t kan båda led divideras med $q(t)$.

$$(\mathbf{k} - \omega^2\mathbf{m})\phi = \mathbf{0}\tag{2.10}$$

Ekvation (2.10) beskriver egenvärdesproblemet där ω^2 är egenvärden och ϕ är egenvektorer. Egenvektorerna kallas i detta sammanhang för systemets egenmoder. Ekvation (2.10) kan för enkla system lösas algebraiskt men det vanligaste är att det löses numeriskt. Egenvektorerna, ϕ , kan normeras på flera sätt, i denna rapport kommer de att normeras på följande sätt:

$$\phi^T \mathbf{m} \phi = \mathbf{M} = \mathbf{I} \quad (2.11)$$

Denna normalisering ger följande samband:

$$\phi^T \mathbf{k} \phi = \mathbf{K} = \begin{bmatrix} \omega_1^2 & 0 & 0 & \dots & 0 \\ 0 & \omega_2^2 & 0 & \dots & 0 \\ 0 & 0 & \omega_3^2 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & \omega_N^2 \end{bmatrix}$$

Det går att visa att den allmänna lösningen till ekvation (2.4) kan uttryckas som en linjärkombination av egenmoderna till systemet, ϕ Chopra [6].

$$\mathbf{u}(t) = \sum_{i=1}^N \phi_i q_i(t)$$

För varje egenmod, ϕ_i , finns det en motsvarande funktion, $q_i(t)$. Detta kan skrivas på matrisform.

$$\mathbf{u}(t) = \phi \mathbf{q}(t) \quad (2.12)$$

Där ϕ är en matris med systemets egenmoder som kolonner.

$$\phi = \begin{bmatrix} [\phi_1] & [\phi_2] & [\phi_3] & \dots & [\phi_N] \end{bmatrix}$$

Vektorn $\mathbf{q}$ innehåller funktioner på formen (2.9) med en funktion per egenmod.

$$\mathbf{q}(t) = \begin{bmatrix} q_1(t) \\ q_2(t) \\ q_3(t) \\ \vdots \\ q_N(t) \end{bmatrix}$$

$$q_i(t) = A_i \cos(\omega_i t) + B_i \sin(\omega_i t) \quad (2.13)$$

Speciellt gäller

$$q_i(0) = A_i \quad \dot{q}_i = \omega_i B_i \quad (2.14)$$

Problemet har nu reducerats till att bestämma konstanterna i ekvation (2.13). Detta kan göras genom att utgå från ekvation (2.12) och multiplicera med $\phi^T \mathbf{m}$ från vänster.

$$\phi^T \mathbf{m} \mathbf{u}(t) = \phi^T \mathbf{m} \phi \mathbf{q}(t) = \mathbf{I} \mathbf{q}(t) = \mathbf{q}(t) \quad (2.15)$$

$$\mathbf{q}(0) = \phi^T \mathbf{m} \mathbf{u}(0)$$

$$q_i(0) = A_i \quad \phi^T \mathbf{m} \mathbf{u}(0) = \begin{bmatrix} A_1 \\ A_2 \\ A_3 \\ \vdots \\ A_N \end{bmatrix}$$

Genom att derivera (2.15) och sätta in $t = 0$ kan resterande konstanter bestämmas

$$\dot{\mathbf{q}}(0) = \boldsymbol{\phi}^T \mathbf{m} \dot{\mathbf{u}}(0)$$

$$\dot{q}_i(0) = \omega_i B_i \quad \dot{\mathbf{q}}(0) = \begin{bmatrix} \omega_1 B_1 \\ \omega_2 B_2 \\ \omega_3 B_3 \\ \vdots \\ \omega_N B_N \end{bmatrix}$$

Vektorn $\mathbf{q}(t)$ är nu bestämd och förskjutningarna kan bestämmas med ekvation (2.12).

2.2.2 Dämpad svängning

Vid beaktande av dämpning är rörelseekvationen för systemet ekvation (2.5). Samma ansättning av lösning görs här som för det odämpade fallet.

$$\mathbf{u}(t) = \boldsymbol{\phi} \mathbf{q}(t)$$

De egenmoder som togs fram för det odämpade systemet används även här. Anledningen till detta är att beräkningarna för egenmoderna i det dämpade systemet är betydligt mer omständliga och egenmoderna för det odämpade systemet är en bra approximation då dämpningen är liten. Genom att sätta in ekvation (2.12) i (2.5) och multiplicera med $\boldsymbol{\phi}^T$ från vänster erhålles

$$\boldsymbol{\phi}^T \mathbf{m} \boldsymbol{\phi} \ddot{\mathbf{q}}(t) + \boldsymbol{\phi}^T \mathbf{c} \boldsymbol{\phi} \dot{\mathbf{q}}(t) + \boldsymbol{\phi}^T \mathbf{k} \boldsymbol{\phi} \mathbf{q}(t) = \mathbf{0}$$

$$\mathbf{M} \ddot{\mathbf{q}}(t) + \mathbf{C} \dot{\mathbf{q}}(t) + \mathbf{K} \mathbf{q}(t) = \mathbf{0} \quad (2.16)$$

Där

$$\mathbf{C} = \boldsymbol{\phi}^T \mathbf{c} \boldsymbol{\phi}$$

För att dra nytta av modal analys krävs att $\mathbf{C}$ är diagonal. Detta är inte nödvändigtvis sant. Om det antas att $\mathbf{C}$ är diagonal blir (2.16) ett system av okopplade differentialekvationer som lätt kan lösas. I denna rapport kommer $\mathbf{C}$ antas vara diagonal. De okopplade ekvationerna blir då

$$\ddot{q}_i(t) + 2\zeta_i \omega_i \dot{q}_i(t) + \omega_i^2 q_i(t) = 0 \quad (2.17)$$

Där ω_i är egenfrekvenserna till systemet och ζ_i definieras på följande sätt:

$$\zeta_i = \frac{c_i}{2m\omega_i}$$

Lösningen till ekvation (2.17) är

$$q_i(t) = e^{-\zeta_i \omega_i t} \left(q_i(0) \cos(\omega_{iD} t) + \frac{\dot{q}_i(0) + \zeta_i \omega_i q_i(0)}{\omega_{iD}} \sin(\omega_{iD} t) \right) \quad (2.18)$$

Där ω_{iD} är de dämpade egenfrekvenserna för systemet.

$$\omega_{iD} = \omega_i \sqrt{1 - \zeta_i^2}$$

Den allmänna lösningen till ekvation (2.5) kan uttryckas som en linjärkombination av egenmoderna:

$$\mathbf{u}(t) = \sum_{i=1}^N \phi_i q_i(t)$$

$\mathbf{C}$ -matrisen får följande utseende:

$$\mathbf{C} = \begin{bmatrix} 2\zeta_1\omega_1 & 0 & 0 & \dots & 0 \\ 0 & 2\zeta_2\omega_2 & 0 & \dots & 0 \\ 0 & 0 & 2\zeta_3\omega_3 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 2\zeta_N\omega_N \end{bmatrix} \quad (2.19)$$

Problemet har nu återigen reducerats till att bestämma konstanterna till funktionen $\mathbf{q}(t)$. Detta görs på samma sätt som i föregående kapitel.

$$\mathbf{q}(0) = \boldsymbol{\phi}^T \mathbf{m} \mathbf{u}(0) \quad \dot{\mathbf{q}}(0) = \boldsymbol{\phi}^T \mathbf{m} \dot{\mathbf{u}}(0)$$

Därefter kan förskjutningarna bestämmas med ekvation (2.12).

2.3 Newmarks metod

Newmarks metod är en algoritm för att lösa differentialekvationer numeriskt Chopra [6]. Fördelen med en numerisk metod är att icke-linjära effekter kan implementeras. Newmarks metod är en time-stepping metod, vilket innebär att lösningen beräknas approximativt vid diskreta tidsintervall. Proceduren för Newmarks metod är följande:

$$\dot{\mathbf{u}}_{i+1} = \dot{\mathbf{u}}_i + (1 - \gamma)\Delta t \ddot{\mathbf{u}}_i + \gamma\Delta t \ddot{\mathbf{u}}_{i+1} \quad (2.20)$$

$$\mathbf{u}_{i+1} = \mathbf{u}_i + \Delta t \dot{\mathbf{u}}_i + ((0.5 - \beta)(\Delta t)^2) \ddot{\mathbf{u}}_i + (\beta(\Delta t)^2) \ddot{\mathbf{u}}_{i+1} \quad (2.21)$$

Om materialresponsen är linjär ser rörelseekvationen för systemet ut på följande vis:

$$\mathbf{m} \ddot{\mathbf{u}}_{i+1} + \mathbf{c} \dot{\mathbf{u}}_{i+1} + \mathbf{k} \mathbf{u}_{i+1} = \mathbf{0} \quad (2.22)$$

Där indexet syftar på iterationen som avses. I detta fall kan $\mathbf{u}_{i+1}$ lösas ut genom att först lösa ut $\ddot{\mathbf{u}}_{i+1}$ ur ekvation (2.21).

$$\ddot{\mathbf{u}}_{i+1} = \frac{1}{\beta(\Delta t)^2}(\mathbf{u}_{i+1} - \mathbf{u}_i) - \frac{1}{\beta\Delta t}\dot{\mathbf{u}}_i - \left(\frac{1}{2\beta} - 1\right)\ddot{\mathbf{u}}_i \quad (2.23)$$

Därefter kan (2.23) sättas in i (2.20).

$$\dot{\mathbf{u}}_{i+1} = \frac{\gamma}{\beta\Delta t}(\mathbf{u}_{i+1} - \mathbf{u}_i) + \left(1 - \frac{\gamma}{\beta}\right)\dot{\mathbf{u}}_i + \Delta t \left(1 - \frac{\gamma}{2\beta}\right)\ddot{\mathbf{u}}_i \quad (2.24)$$

Frihetsgradernas acceleration och hastighet för $i+1$ är nu uttryckt i termer av $\mathbf{u}_{i+1}$, $\mathbf{u}_i$, $\dot{\mathbf{u}}_i$ och $\ddot{\mathbf{u}}_i$. Genom att sätta in (2.23) och (2.24) i (2.22) och lösa ut $\mathbf{u}_{i+1}$ erhålls:

$$\hat{\mathbf{K}} \mathbf{u}_{i+1} = \hat{\mathbf{P}}_{i+1}$$

Där

$$\hat{\mathbf{K}} = \mathbf{k} + \frac{1}{\beta(\Delta t)^2} \mathbf{m} + \frac{\gamma}{\beta \Delta t} \mathbf{c} + \quad (2.25)$$

$$\begin{aligned} \hat{\mathbf{P}}_{i+1} = & \left(\frac{1}{\beta(\Delta t)^2} \mathbf{m} + \frac{\gamma}{\beta \Delta t} \mathbf{c} \right) \mathbf{u}_i + \left(\frac{1}{\beta \Delta t} \mathbf{m} + \left(\frac{\gamma}{\beta} - 1 \right) \mathbf{c} \right) \dot{\mathbf{u}}_i \\ & + \left(\left(\frac{1}{2\beta} - 1 \right) \mathbf{m} + \Delta t \left(\frac{\gamma}{2\beta} - 1 \right) \mathbf{c} \right) \ddot{\mathbf{u}}_i \end{aligned} \quad (2.26)$$

$\mathbf{q}_{i+1}$ kan nu beräknas enligt:

$$\mathbf{u}_{i+1} = \hat{\mathbf{K}}^{-1} \hat{\mathbf{P}}_{i+1}$$

Om begynnelsevärdena till problemet är kända kan denna process itereras för att erhålla en lösning.

Constant average acceleration

Ett vanligt specialfall av Newmarks metod är där $\gamma = \frac{1}{2}$ och $\beta = \frac{1}{4}$. Detta specialfall kallas för constant average acceleration. Detta fall innebär att accelerationen mellan två punkter sätts till genomsnittet av värdena på accelerationen vid de avsedda punkterna.

I denna rapport kommer enbart constant average acceleration att användas.

Stabilitet

Newmarks metod med constant average acceleration är stabil för alla värden på Δt Chopra [6]. Detta innebär att lösningen är begränsad för begränsade värden på tiden. Detta innebär att den enda begränsningen på storleken av Δt är beroende av vilken upplösning på lösningen som önskas.

2.4 Spänningar i glaset

Momentet i den tänkta balken förhåller sig till kraften P enligt Heyden m. fl. [4]:

$$M(x) = \frac{P}{2}x \quad , \quad 0 \leq x \leq \frac{L}{2}$$

Speciellt gäller:

$$M\left(\frac{L}{2}\right) = M_{max} = \frac{PL}{4} \iff P = M_{max} \frac{4}{L} \quad (2.27)$$

Balkens förskjutning i sin mittpunkt förhåller sig till kraften P enligt:

$$u = \frac{PL^3}{48EI} \iff P = u \frac{48EI}{L^3} \quad (2.28)$$

Genom att kombinera (2.27) och (2.28) erhålls följande:

$$u \frac{48EI}{L^3} = M_{max} \frac{4}{L}$$

$$M_{max} = u \frac{12EI}{L^2} \quad (2.29)$$

När momentet är känt kan den maximala spänningen tas fram genom följande samband:

$$\sigma = \frac{M_{max}}{W} \quad (2.30)$$

Där W är balkens böjmotstånd. Genom att kombinera ekvation (2.29) och (2.30) erhålls följande:

$$\sigma = u \frac{12EI}{WL^2} \quad (2.31)$$

Böjmotståndet, W , kan för rektangulära tvärsnitt beräknas enligt:

$$W = \frac{bd^2}{6} \quad (2.32)$$

Spänningen i ekvation (2.31) representerar både drag i underkant av balken samt tryck i ovensidan av balken eftersom att det är ren böjning.

2.5 Program

För att utföra beräkningarna har kod skrivits i MATLAB. Totalt har fem olika program skrivits. Tre av programmen beräknar och redovisar förskjutningar i en graf. De andra två programmen beräknar och redovisar spänningar. I de fall där spänningar beräknas görs en parameterstudie för att undersöka hur spänningen varierar beroende på glasets längd, bredd, tjocklek samt fallhöjden från vilken impaktorn släpps. I tabellen nedan presenteras de värden som testas i parameterstudien.

Fallhöjd (h) [m]	Tjocklek (d) [mm]	Längd (L) [m]	Bredd (b) [m]
0.19	6.0	0.75	0.75
0.45	8.0	1.0	1.0
1.20	10.0	1.25	1.25
	12.0	1.5	1.5

Tabell 2.1: Värden som testas i parameterstudien

Alla värden testas oberoende av varandra vilket innebär att det totalt blir $3 \cdot 4 \cdot 4 \cdot 4 = 192$ fall per program som ingår i parameterstudien.

Nedan presenteras programmens namn och principiella funktion. Koden för programmen återfinns i appendix. Två av programmen simulerar det odämpade systemet och tre av dem simulerar det dämpade.

Odämpat system

Program 1: modalnd

Detta programmet simulerar det odämpade systemet med hjälp av modal analys. Programmet redovisar förskjutningarna i en graf och meddelar om kontaktvillkoret bröts och i sådana fall när.

Program 2: modalndsigma

Detta program bygger på 'modalnd' med skillnaden att det beräknar spänningen i glaset där impaktorn slår i. Enligt dem antaganden som gjorts är det där spänningen är störst. Spänningen i denna punkt beräknas för varje tidpunkt. En parameterstudie görs med hjälp av detta program.

Dämpat system

Program 3: modald

Detta program beräknar förskjutningarna för det dämpade systemet med hjälp av modal analys. Förskjutningarna presenteras i en graf och programmet meddelar om bilvillkoret bröts och i sådana fall när.

Program 4: modaldsigma

Detta program beräknar spänningen i glaset där impaktorn slår i under simuleringens gång. En parameterstudie görs med hjälp av detta program.

Program 5: newmark

Detta program simulerar det dämpade systemet med hjälp av Newmarks metod. Förskjutningarna presenteras i en graf. Programmet meddelar även om kontaktvillkoret bröts och i sådana fall när.

Simuleringens längd

För att få den maximala mängden information från simuleringarna bör beräkningarna göras fram till dess att kontaktvillkoret bryts. Efter att kontaktvillkoret bryts är inte beräkningarna av större intresse. Det är därför eftersträvänsvärt att låta simuleringen pågå fram till dess att kontaktvillkoret bryts, men inte längre än så. När kontaktvillkoret bryts beror dock på vilken indata som programmet får och det går således inte att i förväg veta vilken längd som bör väljas. En lösning på detta problem är att sätta en lång tid för simuleringen och sedan se när kriteriet bröts. Därefter går det att ställa in en lämplig längd på simuleringen. Denna metod används för programmen 'modalnd', 'modald' samt 'newmark'. Figur 2.4 visar hur detta ser ut i programmet 'modalnd'.

```

Modellkravet bröts vid t=48 ms
Warning: Modellkriteriet bryts tidigt i simuleringen
> In modalnd (line 66)
Maximal utböjning av glaset: 24.3 mm      (t=20.4 ms)

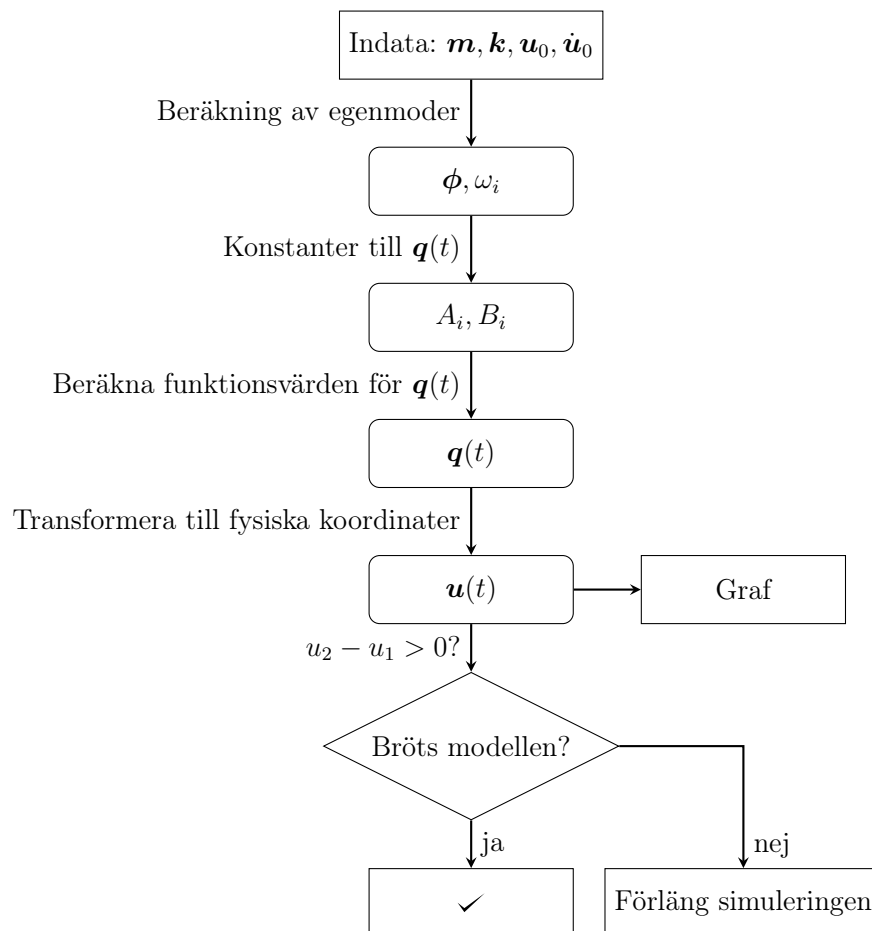
```

Figur 2.4: Konsolmeddelande från modalnd då en lång simuleringslängd valts.

I de resterande programmen redovisas endast spänningen då kontaktvillkoret inte är brutet. För dessa program kan en lämplig simuleringslängd väljas utifrån de grafer som erhålls när programmet körs.

2.5.1 Program 1: modalnd

Detta program beräknar och redovisar förskjutningarna för glaset och impaktorn i det odämpade systemet. Även den maximala utböjningen samt när denna inträffar redovisas. Programmet utnyttjar modal analys för att ta fram förskjutningarna. Koden till programmet finns under appendix A. Den principiella funktionen för 'modalnd' kan ses i figur 2.5.



Figur 2.5: Principiell funktion, modalnd.

Konstanterna A_i och B_i är de konstanter som beskrivs i ekvation (2.14).

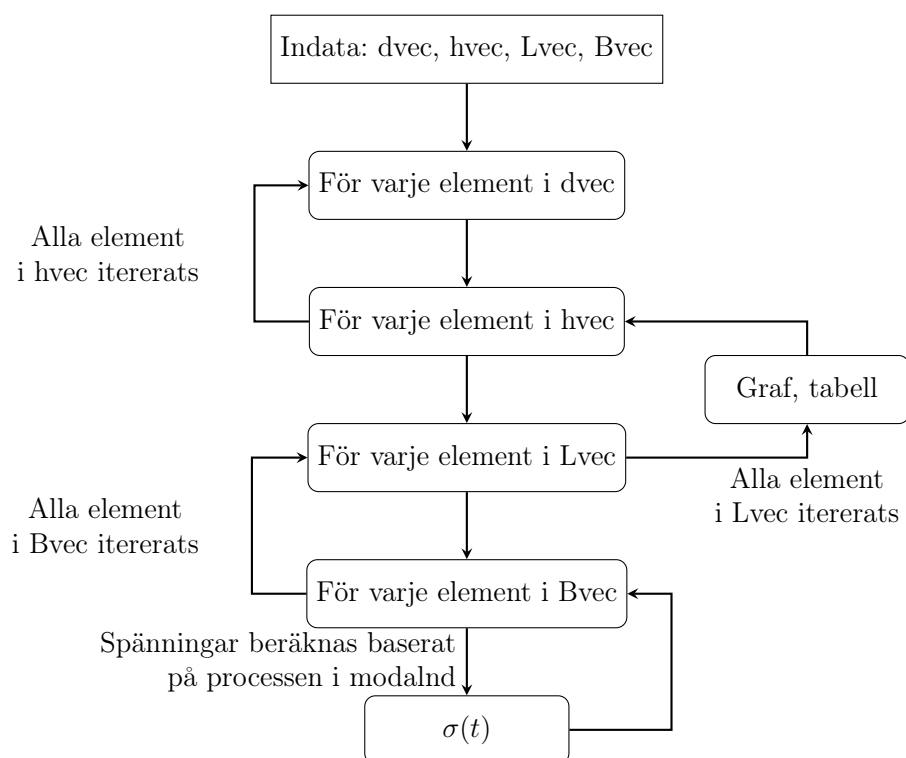
Programmet kommer att köras för följande fall:

Fallhöjd	0.45 m
Tjocklek	12 mm
Längd	1 m
Bredd	1 m

Tabell 2.2: Värden som indata baseras på till modalnd

2.5.2 Program 2: modalndsigma

Detta program bygger på 'modalnd'. För varje kombination av parametrarna som nämns i tabell 2.1 körs en modifierad version av 'modalnd'. Istället för att redovisa förskjutningarna för frihetsgraderna beräknas istället spänningen i glaset för impaktorn slår i. Hur denna spänning varierar i tiden redovisas sedan i en graf. Programmet ger även tabeller där den största spänningen för varje kombination av parametrar anges. Koden för detta program redovisas i appendix B. Programmet principiella funktion kan ses i figur 2.6.



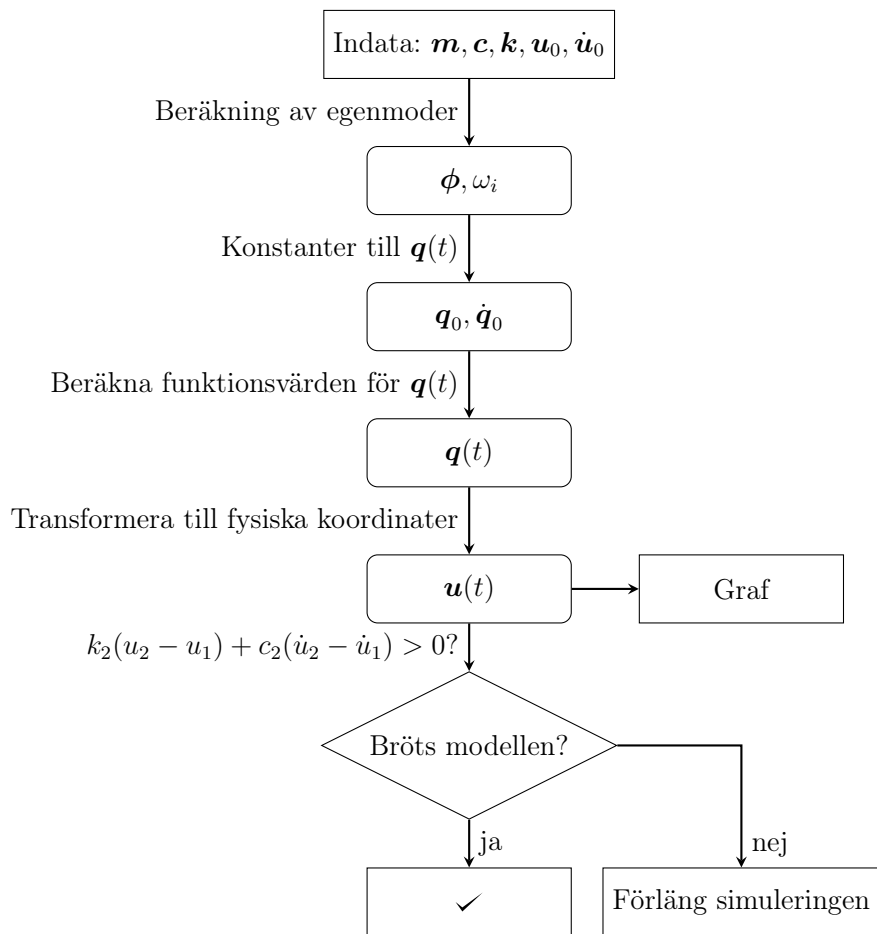
Figur 2.6: Principiell funktion, modalndsigma

Med hjälp av detta program görs en parameterstudie för de värden som presenteras i tabell 2.1. För att göra alla 192 fall överskådliga kommer de presenteras i grupper av 16. Varje graf som presenteras kommer att innehålla 16 kurvor som svarar mot alla kombinationer av längd och bredd. Varje graf svarar således mot en kombination av

fallhöjd och en tjocklek på glaset. Totalt blir det 12 grafer. Enligt Burström [7] är böjhallfastheten för glas 30–100 MPa. Vid kortvariga laster är hållfastheten större. I programmet 'modalndsigma' redovisas endast spänningar om de understiger 200 MPa.

2.5.3 Program 3: modald

Detta program redovisar förskjutningarna på samma sätt som 'modalnd' samt meddelar den maximala spänningen och när den inträffade. Även detta program utnyttjar modal analys för att ta fram förskjutningarna. Skillnaden mot 'modalnd' är att dämpning tas i beaktande så som det beskrivits i metoden. Koden finns i appendix C. Den principiella funktionen av programmet ses i figur 2.7.



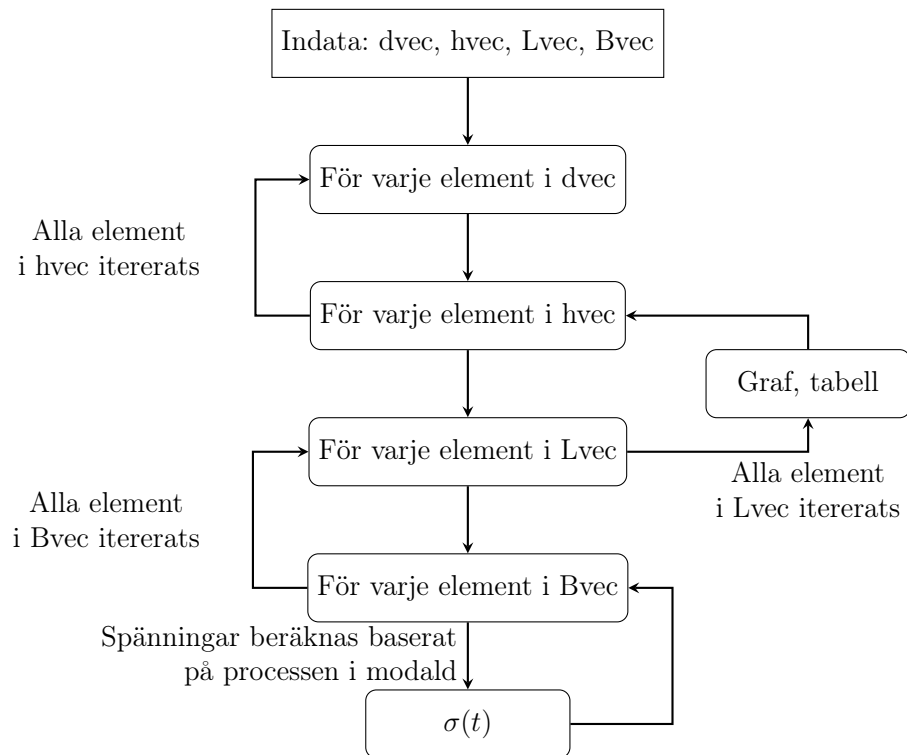
Figur 2.7: Principiell funktion, modald.

Programmet körs för samma fall som 'modalnd' kördes för (se tabell 2.2).

2.5.4 Program 4: modalndsigma

Detta program bygger på 'modald' likt hur 'modalndsigma' bygger på 'modalnd'. Spänningen beräknas för varje kombination av parameter. Den maximala spänningen för varje fall ges i en tabell. Samma figurförklaring som för 'modalndsigma' gäller

även för detta program. Koden till detta program finns i appendix D. Den principiella funktionen av programmet kan ses i figur 2.8.

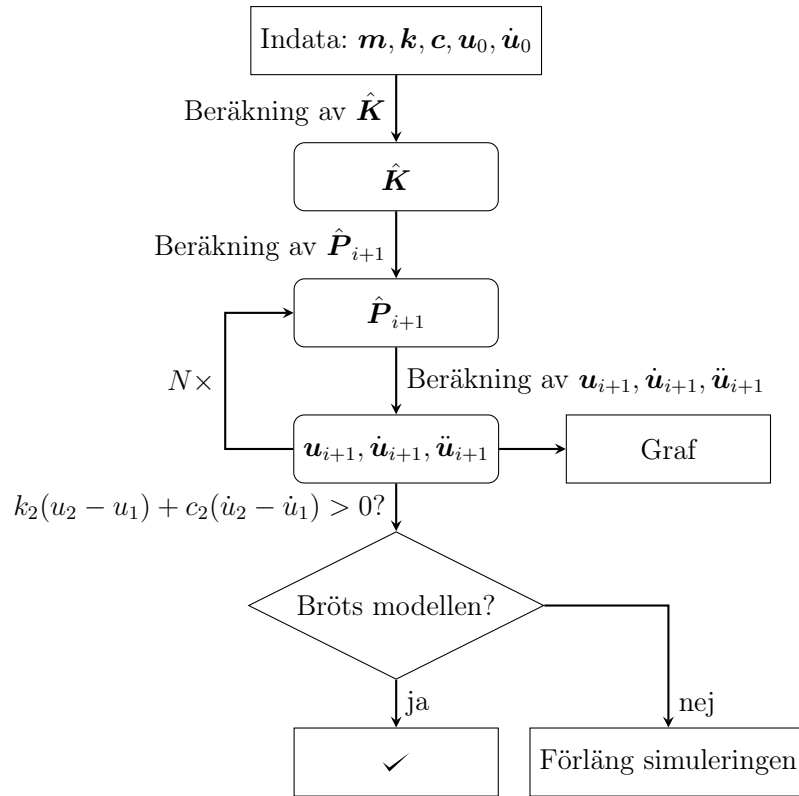


Figur 2.8: Principiell funktion, modalsigma

En parameterstudie utförs på detta program likt 'modalndsigma'. Samma 192 fall undersöks. Samma värden på dämpningsförhållande används som i 'modald'.

2.5.5 Program 5: newmark

Detta program beräknar förskjutningarna för det dämpade systemet i figur 2.3. Programmet gör detta med hjälp av Newmarks metod som beskrevs under rubrik 2.3. Koden till programmet återfinns i appendix E. i figur 2.9 visas den principiella funktionen av programmet.



Figur 2.9: Principiell funktion, newmark.

Där $\hat{K}$ och $\hat{P}_{i+1}$ är de koefficienter som beskrevs i ekvation (2.25) och (2.26). Samma fall kommer att testas i detta program som testades i 'modalnd' och 'modald'.

Längden på tidsstegen

Det är av intresse hur långa tidssteg, Δt , som väljs för simuleringen. Då Newmarks metod är stabil för alla värden på Δt innebär detta att storleken på tidsstegen styrs av vilken upplösning som efterfrågas av lösningen. För att undersöka detta körs programmet i fyra iterationer med olika stora tidssteg. Simuleringens längd sätts till $T = 0.055$ s. För att få en bra bild av hur simuleringen påverkas av storleken på tidsstegen är antalet tidssteg i simuleringen ett bra mått. Om antalet tidssteg ökar exponentiellt täcks ett stort intervall på ett överskådligt sätt. Låt N beteckna antalet tidssteg i simuleringen. De väljs enligt följande samband:

$$N(n) = 10^{\frac{n+2}{3}} \quad \text{avrundat till närmaste heltal}$$

Där n är vilken iteration som avses.

n	N
1	10
2	22
3	46
4	100

Tabell 2.3: Antalet tidssteg för varje iteration

I koden beräknas antalet tidssteg enligt:

$$N = \left\lceil \frac{T}{\Delta t} \right\rceil$$

Storleken på tidsstegen kan således bestämmas genom:

$$\Delta t = \frac{T}{N}$$

Resultaten av beräkningarna presenteras i tabell 2.4.

n	Δt [s]
1	0.0055
2	0.0025
3	0.0012
4	0.00055

Tabell 2.4: Storlek på tidsstegen

3 Resultat

I denna del presenteras de grafer och tabeller som fås från programmen som nämns i metoden. Här redovisas även de beräkningar som görs för att få rätt indata till programmen.

Randvillkor

Randvillkoren till samtliga system beror enbart på fallhöjden. De initiala positionerna för frihetsgraderna är i båda fallen 0.

$$\mathbf{u}_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

De initiala hastigheterna kan beräknas med hjälp av ekvation (2.6). Enbart impaktorns initiala hastighet är nollskild. Värdena presenteras i tabell 3.1.

h [m]	$-\sqrt{2gh}$ [m/s]
0.19	-1.931
0.45	-2.971
1.20	-4.852

Tabell 3.1: Impaktorns initiala hastighet

Gemensam indata

Viss indata är gemensam för alla program och oberoende av vilka värden som används i parameterstudien. Denna indata presenteras här för överskådlighetens skull. Glasets elasticitetsmodul sätts till 70 000 MPa och dess densitet till 2500 kg/m³ enligt Burström [7]. Styvheten för impaktorn sätts till 350 kN/m. Detta värde är uppskattat utifrån de resultat som presenterats av Andersson m. fl. [3]. Impaktorns massa sätts till 50 kg enligt förutsättningarna för provningsmetoden (se kapitel 1).

E	70 000 MPa
ρ	2500 kg/m ³
k_2	350 kN/m
m_2	50 kg

Tabell 3.2: Gemensam indata

Värden på dämpningsförhållandena sätts till följande:

$$\zeta_1 = 0.02 \quad \zeta_2 = 0.08$$

Dessa värden är en uppskattning av de värden som togs fram av Andersson m. fl. [3].

3.1 Program 1: modalnd

Indata

Den indata som behövs beräknas är glasets massa och styvhet. Massan kan beräknas enligt:

$$m_{glas} = \rho \cdot L \cdot b \cdot d$$

Där L är glasets längd, b dess bredd och d dess tjocklek. ρ är glasets densitet. L , b och d kan läsas av i tabell 2.2.

$$m_{glas} = 2500 \text{ kg/m}^3 \cdot 1.0 \text{ m} \cdot 1.0 \text{ m} \cdot 0.012 \text{ m} = 30 \text{ kg}$$

Den bidragande massan kan nu beräknas enligt ekvation (2.7).

$$m_1 = 0.486 \cdot 30 \text{ kg} = 14.58 \text{ kg}$$

Glasets styvhet beräknas enligt ekvation (2.2) och (2.3).

$$I = \frac{1.0 \text{ m} \cdot (0.012 \text{ m})^3}{12} = 1.44 \cdot 10^{-7} \text{ m}^4$$

$$k_1 = \frac{48 \cdot 70\,000 \text{ MPa} \cdot 1.44 \cdot 10^{-7} \text{ m}^4}{(1.0 \text{ m})^3} = 483.84 \text{ kN/m}$$

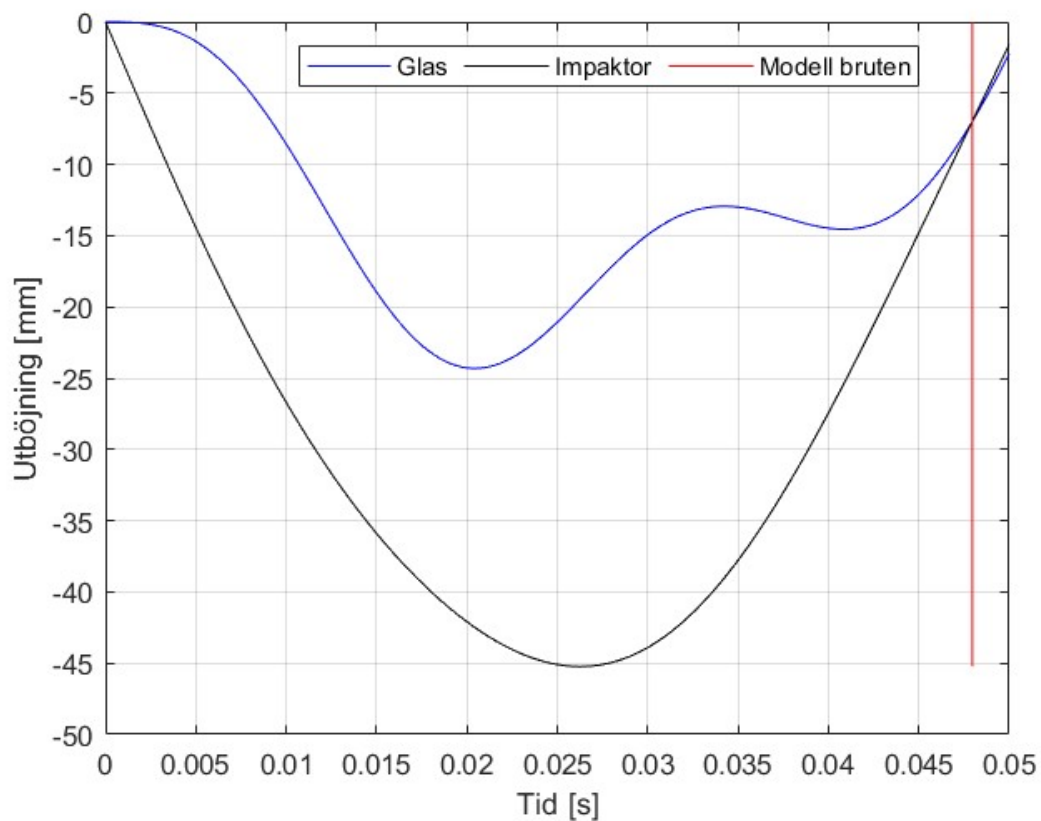
Värdena sammanfattas i tabell 3.3.

m_1	14.58 kg
I	$1.44 \cdot 10^{-7} \text{ m}^4$
k_1	483.84 kN/m

Tabell 3.3: Indata för modalnd

Utdata

Genom att köra programmet som redovisas i appendix A fås följande graf och konsolmeddelande:



```
>> modalnd
Modellkravet bröts vid t=48 ms
Maximal utböjning av glaset: 24.3 mm      (t=20.4 ms)
```

Figur 3.1: Utdata för modalnd

3.2 Program 2: modalndsigma

Indata

Indatan som behövs för parameterstudien är den i tabell 2.1. Utifrån de värdena kan följande beräknas:

$$W = \frac{bd^2}{6}$$

$$I = \frac{bd^3}{12}$$

$$m_1 = 0.486 \cdot \rho \cdot L \cdot b \cdot d$$

$$k_1 = \frac{48EI}{L^3}$$

Detta är tillräckligt för att köra programmet. Värdena för dem beräkningarna redovisas inte för överskådlighetens skull. Se appendix B för hur de tas fram.

Figurförklaring

Tabell 3.4 förklarar hur graferna från modalndsigma skall tolkas. Om spänningarna i beräkningarna för en kurva skulle överstiga 200 MPa kommer resten av grafen inte att redovisas. Om detta är fallet markeras kurvan med en röd stjärna i sitt slut. Om det inte finns någon röd stjärna i slutet av kurvan innebär det att kontaktvillkoret bröts där kurvan tar slut.

L	Färg
0.75	Lila
1.0	Röd
1.25	Blå
1.5	Svart

(a) Färger

b	Linjetyp
0.75	Punktstreckad
1.0	Heldragen
1.25	Streckad
1.5	Prickad

(b) Linjetyp

Tabell 3.4: Figurförklaring

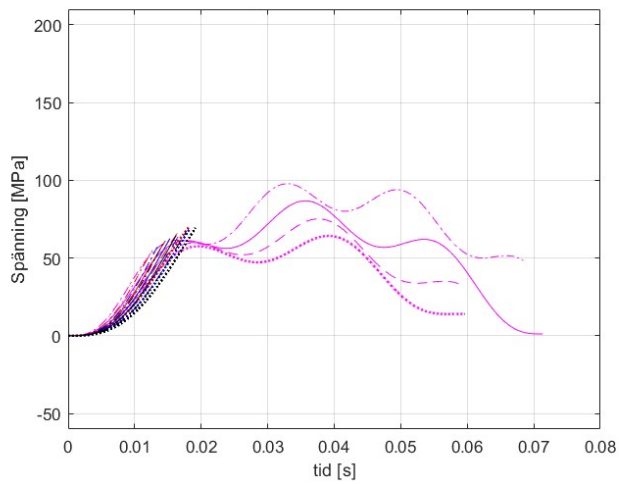
Utdata

Genom att köra programmet i appendix B fås följande grafer och tabeller:

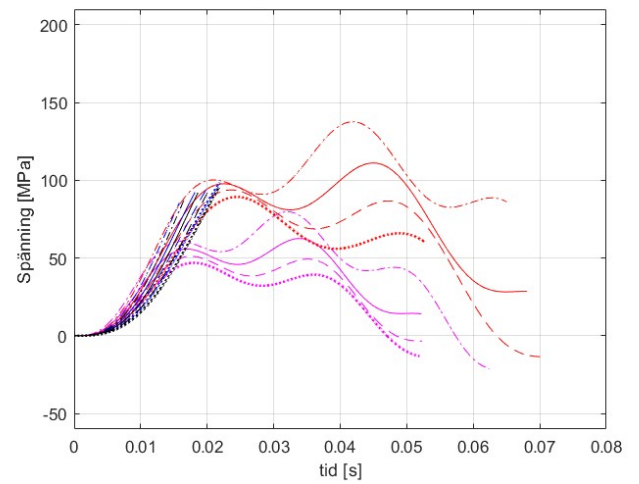
Fallhöjd:

0.19 m

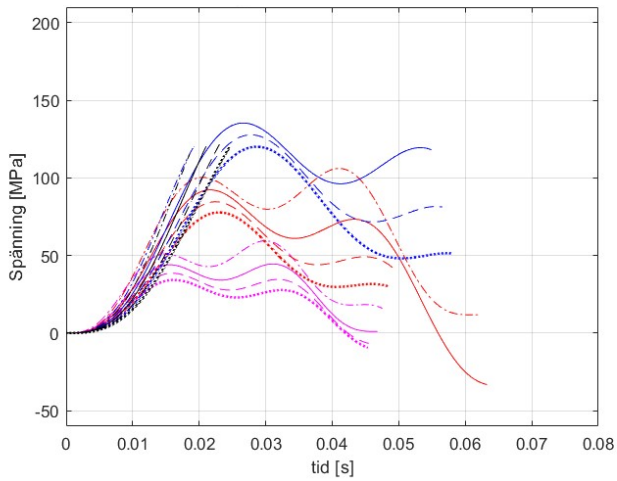
Spänningar



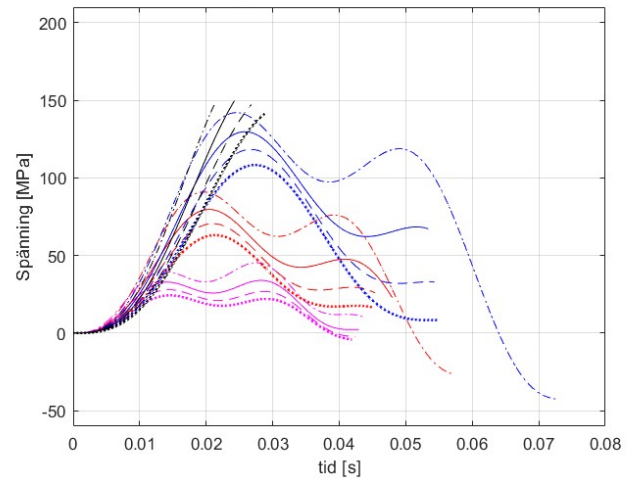
(a) Glastjocklek: 6 mm



(b) Glastjocklek: 8 mm



(c) Glastjocklek: 10 mm



(d) Glastjocklek: 12 mm

Figur 3.2: modalnsigma: spänning i glaset för olika glastjocklekar med 0.19 m i fallhöjd.

Maximala spänningar

		σ_{max} [MPa]			
$L \backslash b$		0.75	1.0	1.25	1.5
0.75		97.8	86.8	75.2	64.2
1.0		-	-	-	-
1.25		-	-	-	-
1.5		-	-	-	-

$d = 6 \text{ mm}$

		σ_{max} [MPa]			
$L \backslash b$		0.75	1.0	1.25	1.5
0.75		79.8	62.6	51.1	46.9
1.0		137.6	111.1	93.8	89.3
1.25		-	-	-	-
1.5		-	-	-	-

$d = 8 \text{ mm}$

		σ_{max} [MPa]			
$L \backslash b$		0.75	1.0	1.25	1.5
0.75		59.5	44.5	38.6	34.3
1.0		106.1	92.3	84.6	77.7
1.25		-	135.4	127.7	120.1
1.5		-	-	-	-

$d = 10 \text{ mm}$

		σ_{max} [MPa]			
$L \backslash b$		0.75	1.0	1.25	1.5
0.75		45.4	34.0	28.0	24.3
1.0		90.9	79.7	70.5	63.1
1.25		142.0	129.8	118.3	108.3
1.5		-	-	-	-

$d = 12 \text{ mm}$

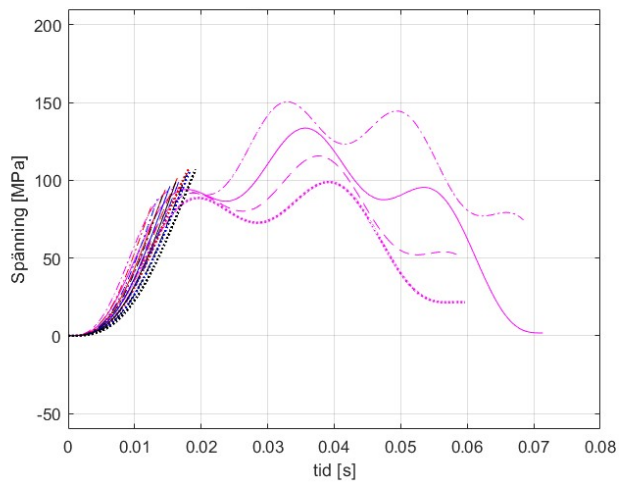
Tabell 3.5: modalndsigma: maximala spänningar med fallhöjd: 0.19 m

De maximala spänningarna redovisas i tabellerna ovan endast om kontaktvillkoret bröts efter att den maximala spänningen uppnåddes. Vilka fall detta gäller går att urskilja från graferna.

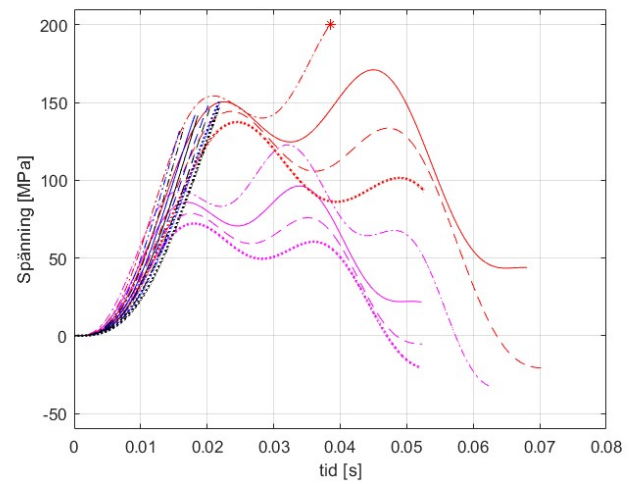
Fallhöjd:

0.45 m

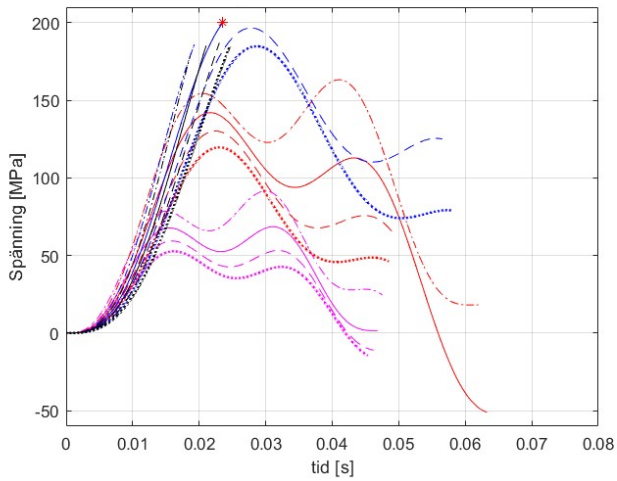
Spänningar



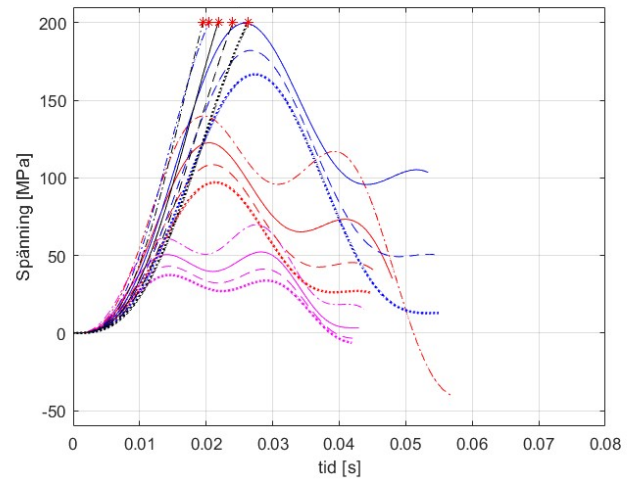
(a) Glastjocklek: 6 mm



(b) Glastjocklek: 8 mm



(c) Glastjocklek: 10 mm



(d) Glastjocklek: 12 mm

Figur 3.3: modalndsigma: spänning i glaset för olika glastjocklekar med 0.45 m i fallhöjd.

Maximala spänningar

$L \backslash b$		σ_{max} [MPa]			
		0.75	1.0	1.25	1.5
0.75		150.5	133.6	115.7	98.9
1.0		-	-	-	-
1.25		-	-	-	-
1.5		-	-	-	-

$d = 6 \text{ mm}$

$L \backslash b$		σ_{max} [MPa]			
		0.75	1.0	1.25	1.5
0.75		122.8	96.3	78.6	72.2
1.0		*	171.0	144.3	137.5
1.25		-	-	-	-
1.5		-	-	-	-

$d = 8 \text{ mm}$

$L \backslash b$		σ_{max} [MPa]			
		0.75	1.0	1.25	1.5
0.75		91.5	68.5	59.4	52.7
1.0		163.2	142.0	130.2	119.6
1.25		-	*	196.6	184.8
1.5		-	-	-	-

$d = 10 \text{ mm}$

$L \backslash b$		σ_{max} [MPa]			
		0.75	1.0	1.25	1.5
0.75		69.8	52.3	43.0	37.4
1.0		139.9	122.6	108.5	97.1
1.25		*	199.7	182.1	166.7
1.5		*	*	*	*

$d = 12 \text{ mm}$

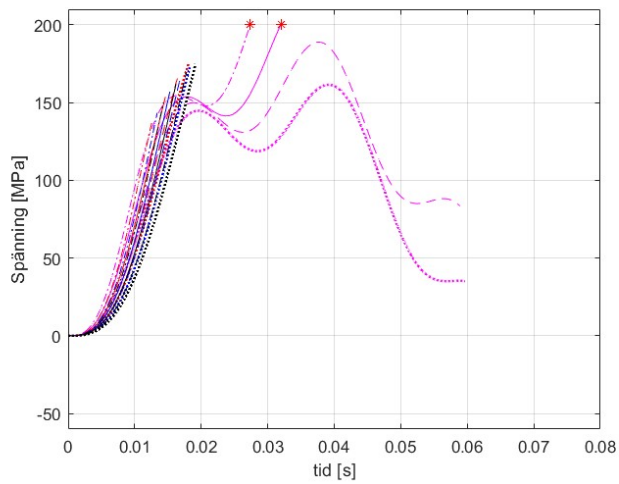
Tabell 3.6: modalndsigma: maximala spänningar med fallhöjd: 0.45 m

Här har vissa av fallen uppnått 200 MPa före kontaktvillkoret bröts. De markeras med '*’.

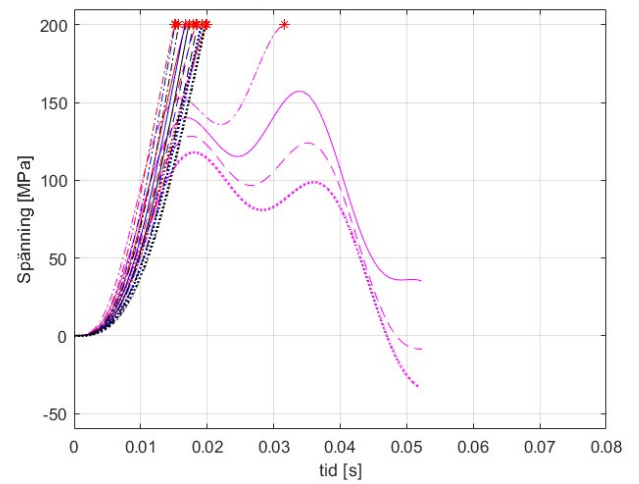
Fallhöjd:

1.20 m

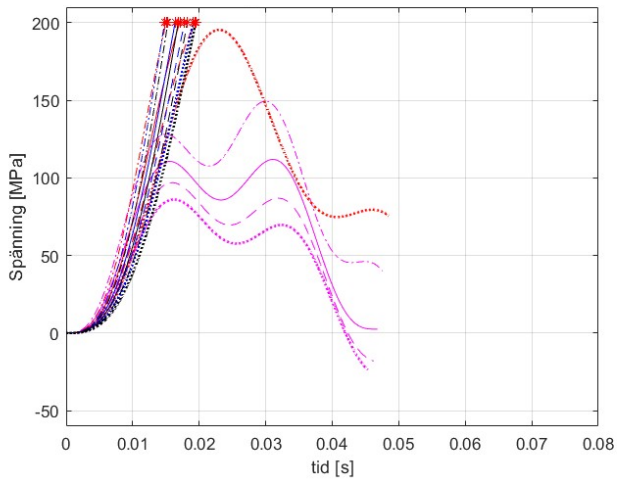
Spänningar



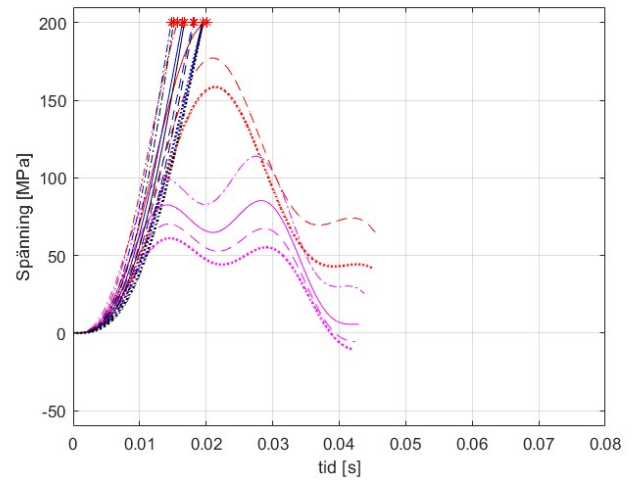
(a) Glastjocklek: 6 mm



(b) Glastjocklek: 8 mm



(c) Glastjocklek: 10 mm



(d) Glastjocklek: 12 mm

Figur 3.4: modalndsigma: spänning i glaset för olika glastjocklekar med 1.20 m i fallhöjd.

Maximala spänningar

σ_{max} [MPa]				
$L \backslash b$	0.75	1.0	1.25	1.5
0.75	*	*	188.9	161.4
1.0	-	-	-	-
1.25	-	-	-	-
1.5	-	-	-	-

$d = 6 \text{ mm}$

σ_{max} [MPa]				
$L \backslash b$	0.75	1.0	1.25	1.5
0.75	*	157.3	128.4	117.9
1.0	*	*	*	*
1.25	*	*	*	*
1.5	*	*	*	*

$d = 8 \text{ mm}$

σ_{max} [MPa]				
$L \backslash b$	0.75	1.0	1.25	1.5
0.75	149.4	111.9	96.9	86.1
1.0	*	*	*	195.4
1.25	*	*	*	*
1.5	*	*	*	*

$d = 10 \text{ mm}$

σ_{max} [MPa]				
$L \backslash b$	0.75	1.0	1.25	1.5
0.75	114.0	85.4	70.3	61.1
1.0	*	*	177.2	158.6
1.25	*	*	*	*
1.5	*	*	*	*

$d = 12 \text{ mm}$

Tabell 3.7: modalndsigma: maximala spänningar med fallhöjd: 1.20 m

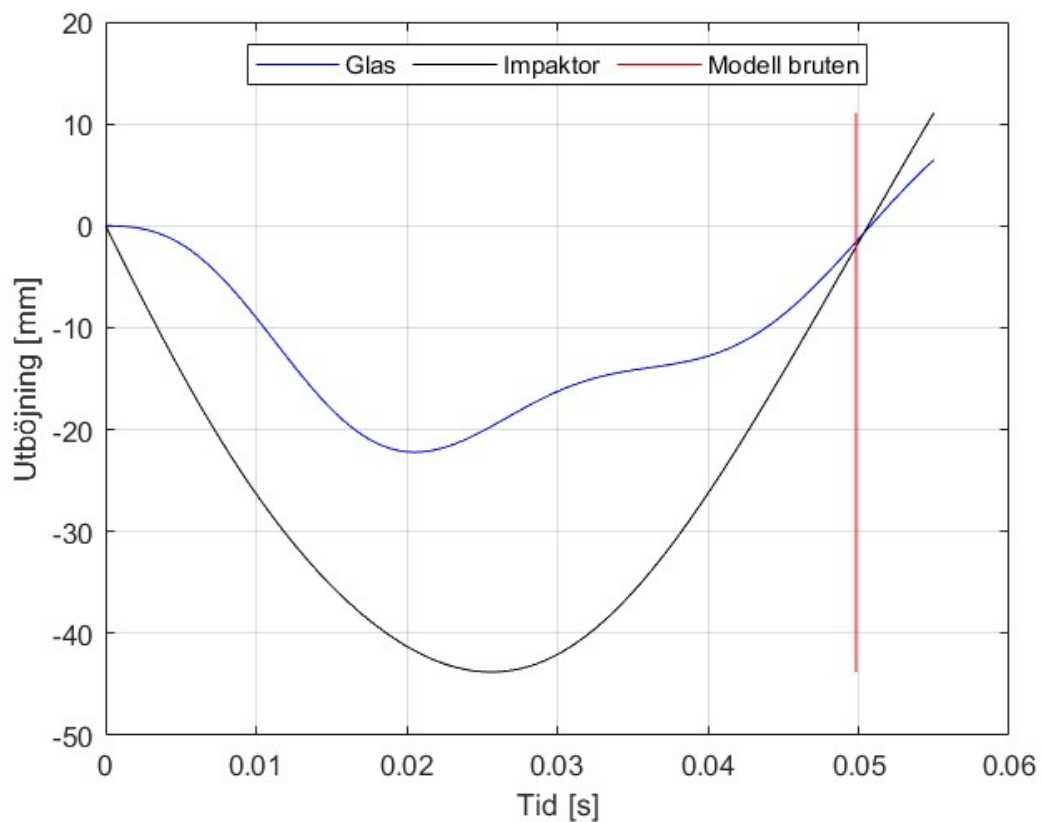
3.3 Program 3: modald

Indata

Den indata som behövs för detta program är densamma som för 'modalnd' (se tabell 3.3) så när som på dämpningsförhållandena vilka är sedan tidigare bestämda.

Utdata

Genom att köra koden i appendix C erhålls följande graf och konsolmeddelande:



```
>> modald
Modellkravet bröts vid t=49.8 ms
Maximal utböjning av glaset: 22.2 mm      (t=20.7 ms)
```

Figur 3.5: Utdata för modald

3.4 Program 4: modalndsigma

Indata

Indatan för detta program är densamma som för 'modalndsigma' så när som på dämpningsförhållandena.

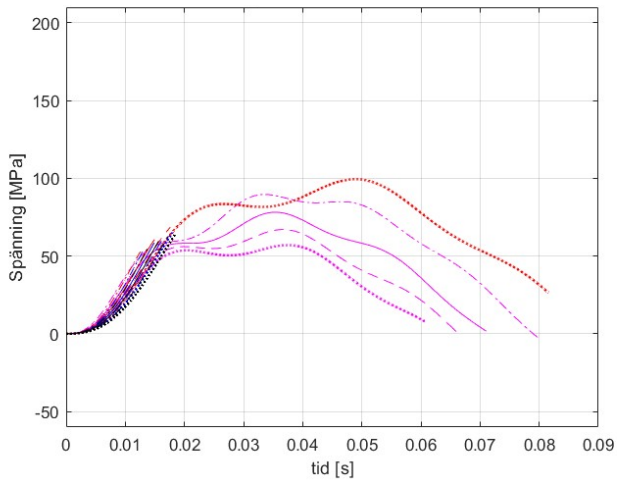
Utdata

Genom att köra programmet i appendix D fås följande grafer och tabeller:

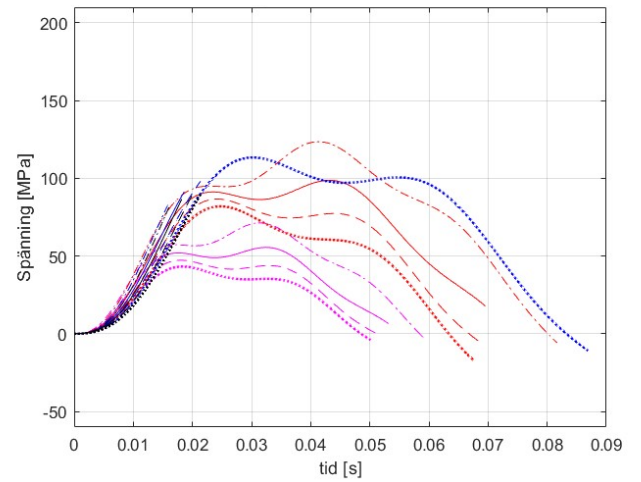
Fallhöjd:

0.19 m

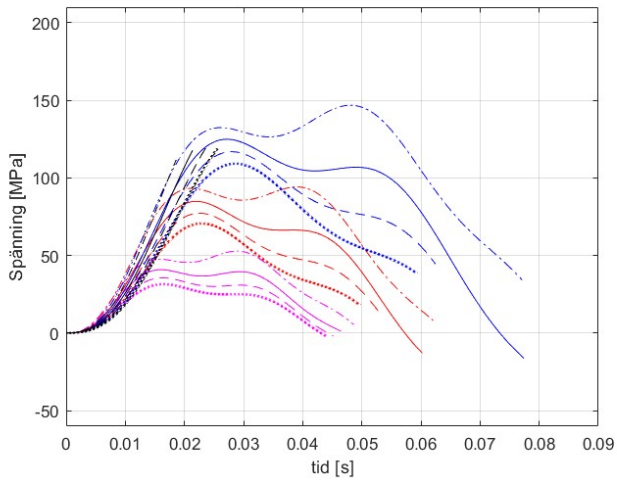
Spänningar



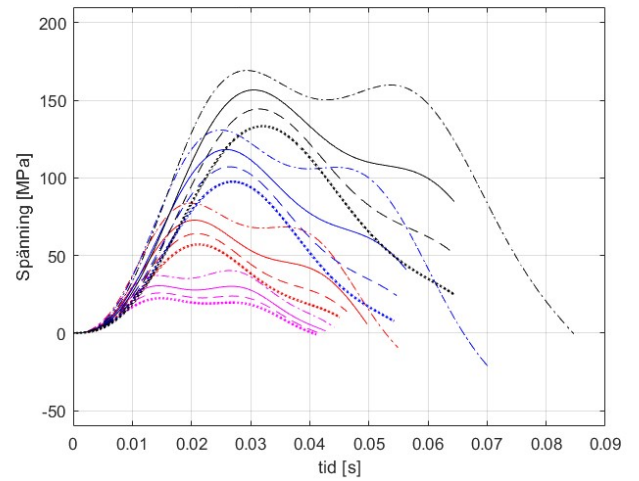
(a) Glastjocklek: 6 mm



(b) Glastjocklek: 8 mm



(c) Glastjocklek: 10 mm



(d) Glastjocklek: 12 mm

Figur 3.6: modal sigma: spänning i glaset för olika glastjocklekar med 0.19 m i fallhöjd.

Maximala spänningar

		σ_{max} [MPa]			
		0.75	1.0	1.25	1.5
$L \backslash b$					
0.75		89.6	78.2	67.0	57.0
1.0		-	-	-	99.4
1.25		-	-	-	-
1.5		-	-	-	-

$d = 6 \text{ mm}$

		σ_{max} [MPa]			
		0.75	1.0	1.25	1.5
$L \backslash b$					
0.75		71.3	55.6	47.3	43.2
1.0		123.5	98.7	86.7	82.0
1.25		-	-	-	113.4
1.5		-	-	-	-

$d = 8 \text{ mm}$

		σ_{max} [MPa]			
		0.75	1.0	1.25	1.5
$L \backslash b$					
0.75		52.84	40.84	35.6	31.5
1.0		94.2	84.9	77.3	70.6
1.25		146.8	124.9	116.9	109.1
1.5		-	-	-	-

$d = 10 \text{ mm}$

		σ_{max} [MPa]			
		0.75	1.0	1.25	1.5
$L \backslash b$					
0.75		40.3	30.5	25.9	22.5
1.0		83.7	72.8	64.1	57.1
1.25		130.8	118.3	107.1	97.6
1.5		169.2	156.7	144.4	133.3

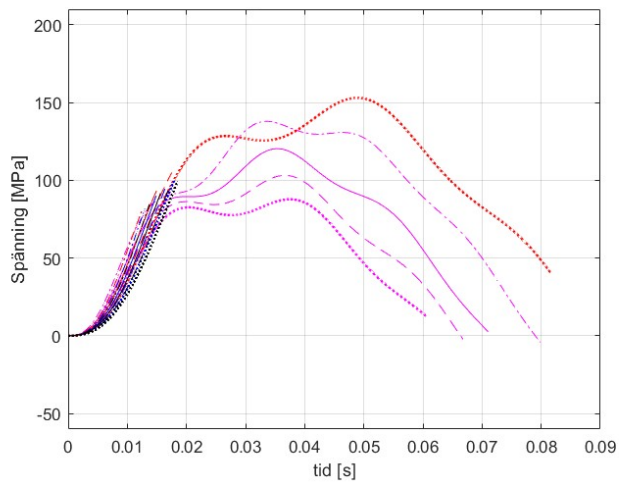
$d = 12 \text{ mm}$

Tabell 3.8: modalldsigma: maximala spänningar med fallhöjd: 0.19 m

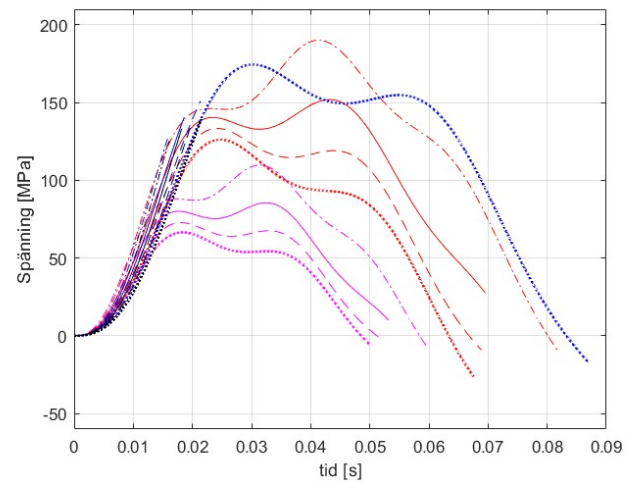
Fallhöjd:

0.45 m

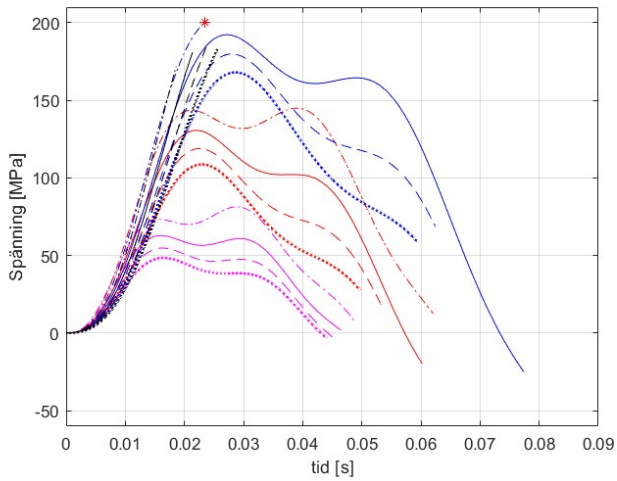
Spänningar



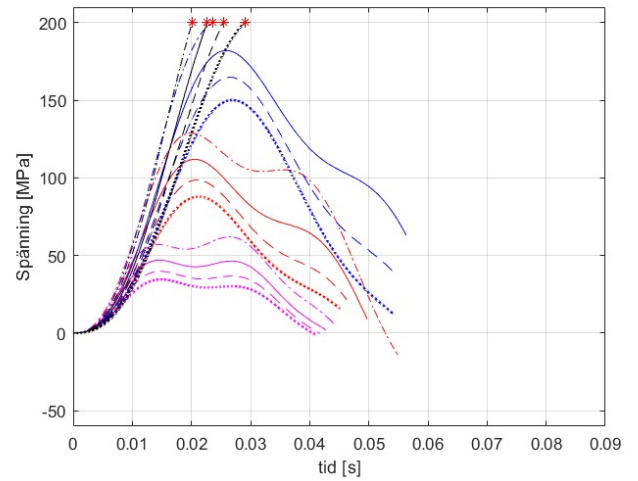
(a) Glastjocklek: 6 mm



(b) Glastjocklek: 8 mm



(c) Glastjocklek: 10 mm



(d) Glastjocklek: 12 mm

Figur 3.7: modal sigma: spänning i glaset för olika glastjocklekar med 0.45 m i fallhöjd.

Maximala spänningar

		σ_{max} [MPa]			
		0.75	1.0	1.25	1.5
$L \backslash b$					
0.75		137.9	120.3	103.2	87.8
1.0		-	-	-	153.0
1.25		-	-	-	-
1.5		-	-	-	-

$d = 6 \text{ mm}$

		σ_{max} [MPa]			
		0.75	1.0	1.25	1.5
$L \backslash b$					
0.75		109.8	85.5	72.8	66.5
1.0		190.1	151.9	133.4	126.2
1.25		-	-	-	174.5
1.5		-	-	-	-

$d = 8 \text{ mm}$

		σ_{max} [MPa]			
		0.75	1.0	1.25	1.5
$L \backslash b$					
0.75		81.3	62.9	54.8	48.5
1.0		144.9	130.6	118.9	108.7
1.25		*	192.2	179.8	168.0
1.5		-	-	-	-

$d = 10 \text{ mm}$

		σ_{max} [MPa]			
		0.75	1.0	1.25	1.5
$L \backslash b$					
0.75		62.0	47.0	39.8	34.6
1.0		128.8	112.0	98.6	87.9
1.25		*	182.0	164.9	150.1
1.5		*	*	*	*

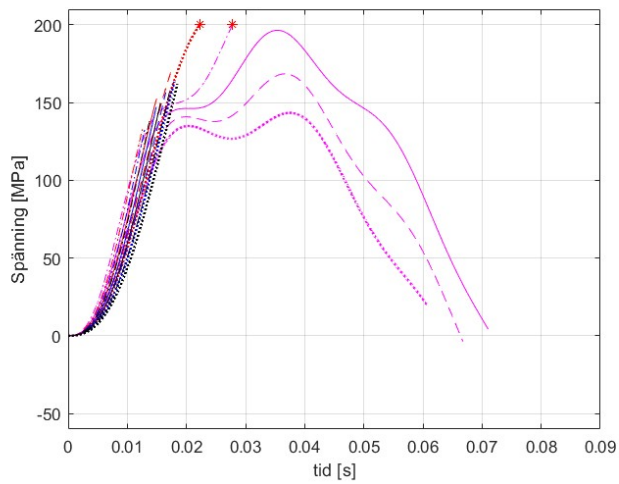
$d = 12 \text{ mm}$

Tabell 3.9: modaldsigma: maximala spänningar med fallhöjd: 0.45 m

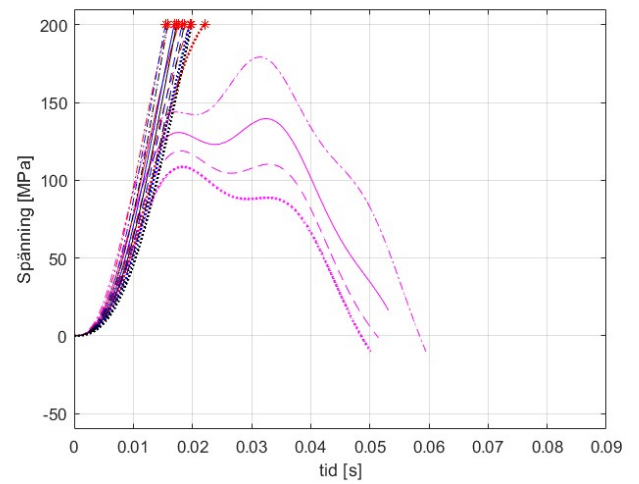
Fallhöjd:

1.20 m

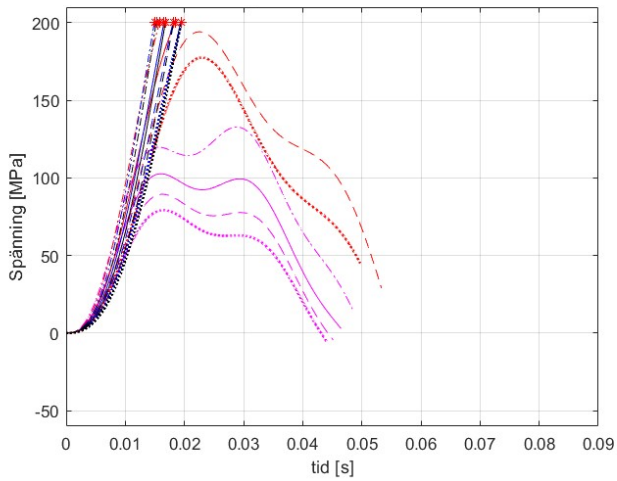
Spänningar



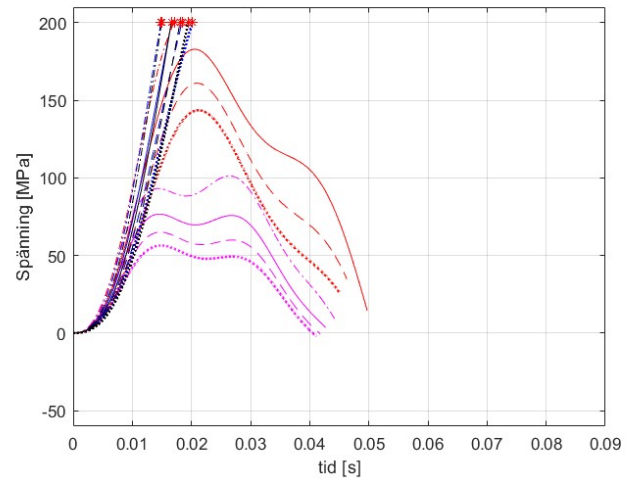
(a) Glastjocklek: 6 mm



(b) Glastjocklek: 8 mm



(c) Glastjocklek: 10 mm



(d) Glastjocklek: 12 mm

Figur 3.8: modal sigma: spänning i glaset för olika glastjocklekar med 1.20 m i fallhöjd.

Maximala spänningar

		σ_{max} [MPa]			
$L \backslash b$		0.75	1.0	1.25	1.5
0.75		*	196.5	168.4	143.4
1.0		-	-	-	*
1.25		-	-	-	-
1.5		-	-	-	-

$d = 6 \text{ mm}$

		σ_{max} [MPa]			
$L \backslash b$		0.75	1.0	1.25	1.5
0.75		179.2	139.6	118.9	108.6
1.0		*	*	*	*
1.25		*	*	*	*
1.5		*	*	*	*

$d = 8 \text{ mm}$

		σ_{max} [MPa]			
$L \backslash b$		0.75	1.0	1.25	1.5
0.75		132.8	102.6	89.5	79.2
1.0		*	*	194.1	177.5
1.25		*	*	*	*
1.5		*	*	*	*

$d = 10 \text{ mm}$

		σ_{max} [MPa]			
$L \backslash b$		0.75	1.0	1.25	1.5
0.75		101.2	76.7	65.0	56.4
1.0		*	182.9	161.0	143.6
1.25		*	*	*	*
1.5		*	*	*	*

$d = 12 \text{ mm}$

Tabell 3.10: modal sigma: maximala spänningar med fallhöjd: 1.20 m

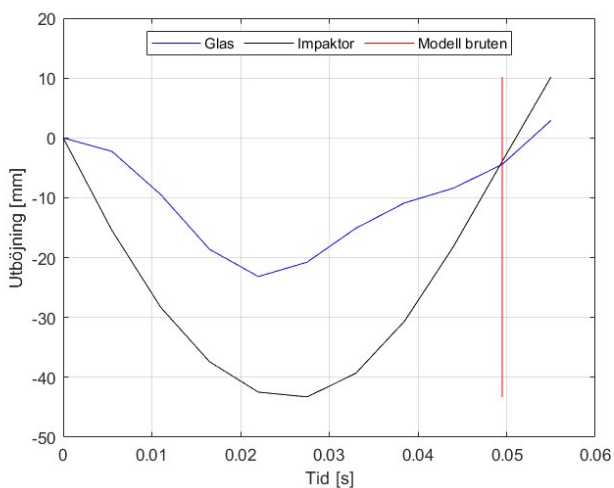
3.5 Program 5: newmark

Indata

Samma indata används som för 'modald'. Längden på tidsstegen bestämdes i metoden. Längden på tidsstegen varierar enligt tabell 2.4.

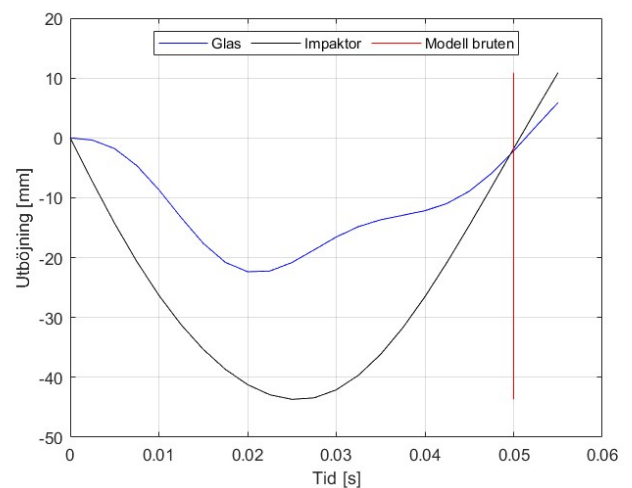
Utdata

De grafer och konsolmeddelanden som fås av att köra programmet med de ovan redovisade storlekarna på tidssteg presenteras nedan.



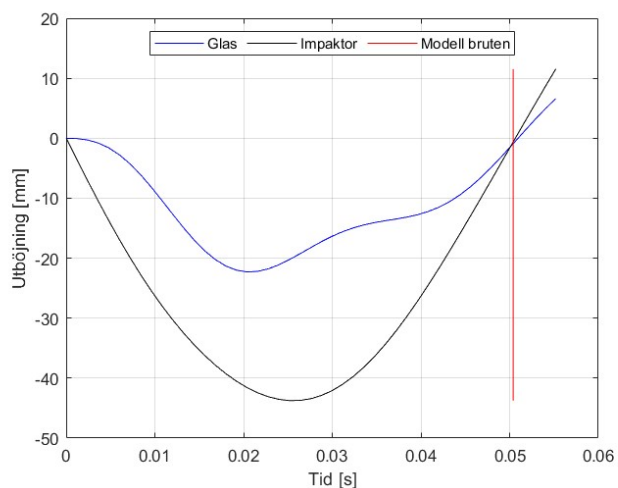
```
>> newmark
Modellkravet bröts vid t=49.5 ms
Maximal utböjning av glaset: 23.2 mm
Antal simuleringssteg: 10
```

(a) $\Delta t = 0.0055$ s



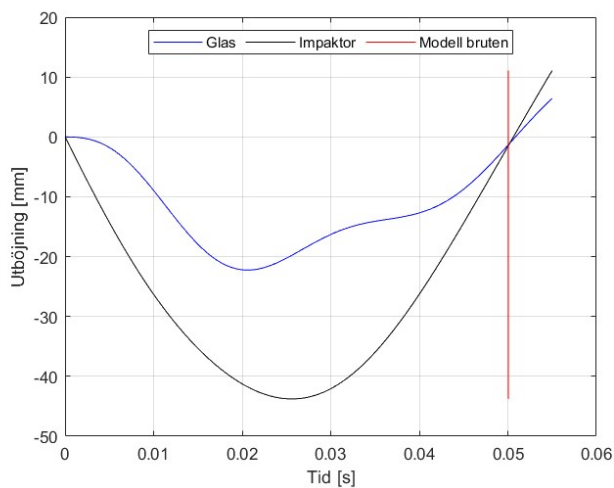
```
>> newmark
Modellkravet bröts vid t=50 ms
Maximal utböjning av glaset: 22.4 mm
Antal simuleringssteg: 22
```

(b) $\Delta t = 0.0025$ s



```
>> newmark
Modellkravet bröts vid t=50.4 ms
Maximal utböjning av glaset: 22.3 mm    (t=20.4 ms)
Antal simuleringssteg: 46
```

(c) $\Delta t = 0.0012$ s



```
>> newmark
Modellkravet bröts vid t=50.1 ms
Maximal utböjning av glaset: 22.2 mm    (t=20.4 ms)
Antal simuleringssteg: 100
```

(d) $\Delta t = 0.00055$ s

Figur 3.9: Grafer och konsolmeddelande från newmark för olika långa tidssteg

4 Diskussion

Syftet med rapporten är att ta fram modeller av den dynamiska responsen av glaspaneler samt program för att utföra beräkningar för de framtagna modellerna. För vissa fall har denna respons kunnat simuleras framgångsrikt inom ramen för de antaganden som gjorts. I de fall där kontaktvillkoret brutits innan den maximala spänningen uppnåtts har systemet inte kunnat simuleras framgångsrikt ur en spänningssynpunkt.

kontaktvillkoret

I en stor andel av de testade fallen i parameterstudien bryts kontaktvillkoret innan dess att den maximala spänningen uppnåtts. Detta belyser en stor brist som modellen har. För att lösa detta problem kan systemet simuleras vidare efter att kontaktvillkoret brutits utan någon interaktion mellan glaset och impaktorn. Systemet hade kunnat simuleras på detta sätt tills dess att den maximala spänningen uppnåtts eller impaktorn och glaset kommer i kontakt igen varvid systemet kan simuleras vidare såsom det gjorts i denna rapport.

Då spänningen enligt antaganden som gjorts i denna rapport är en omskalning av utböjningen har programmen 'modalnd', 'modald' och 'newmark' samma problem. För de fall där dessa program testades var modellen emellertid framgångsrik i att simulera systemet inom ramen för de antaganden som görs i rapporten.

Jämförelse mellan modal analys och Newmarks metod

Fördelar med modal analys är att det är beräkningsmässigt effektivt då en analytisk lösning av systemet erhålls. Modal analys tillåter dock inte för olinjära modeller. En analytisk lösning för sådana system finns inte i regel. Vid användning av Newmarks metod krävs ingen analytisk lösning av systemet. Det är således möjligt att simulera olinjära system. Implementering av olinjärt beteende skulle ge stora möjligheter till förbättring av modellen som använts i denna rapport.

Newmark: längden på tidssteg

Graferna från 'newmark' är mycket lika den från 'modald', likaså värdena som ges i konsolmeddelandena. Detta innebär att $\Delta t = 0.0055\text{ s}$ var i detta fall tillräckligt för att få tillfredsställande resultat vad gäller precisionen. Grafen har dock låg upplösning. Upplösningen är bättre för $\Delta t = 0.0025\text{ s}$ och än bättre för $\Delta t = 0.0012\text{ s}$. Det är ingen märkbar skillnad på upplösningen för $\Delta t = 0.00055\text{ s}$. Detta innebär att kortare tidssteg än $\Delta t = 0.0012\text{ s}$ är i detta fall inte nödvändigt då det varken ger nämnvärd

bättre precision eller bättre upplösning. Det är viktigt att ha i åtanke vid jämförelse mellan graferna från 'newmark' och 'modald' att den sistnämnda inte ger den exakta lösningen till systemet. Egenomderna som används i 'modald' är för det odämpade systemet, vilket är en bra approximation men inte en exakt lösning.

Parameterstudie

Utifrån den parameterstudie som gjorts går det att dra slutsatser för hur de ingående parametrarna påverkar systemet. Genom att studera kolumnerna i tabellerna som hör till 'modalndsigma' samt 'modaldsigma' ses att en ökad längd på balken ger en högre maximal spänning. Detta är att förvänta sig då en större spännvidd på balken gör att utböjningen blir större och således även spänningen.

Genom att istället studera raderna i samma tabeller inses att en ökad bredd på balkens tvärsnitt ger en minskning i den maximala spänningen. Även detta är att förvänta sig då böjmotsåndet, W , ökar snabbare än yttröghetsmomentet, I . Minskningen är inte lika stor som den motsvarande ökningen är för en ökad längd på balken.

En ökad fallhöjd ger en högre maximal spänning, likaså en minskad tjocklek på glaset. Sammanfattningsvis är resultaten av parameterstudien i linje med den teori som modellerna baseras på. Slutsatserna är därav inte av något större intresse med undantag för att modellerna är i viss mån rimliga.

Vidare studier

För att bygga vidare på de resultat som tagits fram i föreliggande rapport kan vidare studier göras. Simulering av systemet efter det att kontaktvillkoret brutits är av intresse för att kunna få en fullständig bild av responsen.

En mer detaljerad strukturmekanisk beskrivning av glaspanelen kan användas då den som använts i denna rapport är kraftigt förenklad.

Framför allt kan en olinjär modell implementeras. Implementeringen kan utgå från metoden och tillhörande program för Newmarks metod.

Litteratur

- [1] Boverket. *Glas i byggnader*. Hämtad: 14 maj 2025. 2019. URL: <https://www.boverket.se/sv/PBL-kunskapsbanken/regler-om-byggande/boverkets-byggregler/sakerhet-vid-anvandning/glas-i-byggnader/>.
- [2] SIS. *SS-EN 12600 Byggnadsglas - Pendelprov - Motstånd mot tung stöt och klassindelning för planglas*. 2003.
- [3] Linus Andersson, Marcon Kozłowski, Peter Persson, Per-Erik Austrell och Kent Persson. “Reduced order modeling of soft-body impact on glass panels”. I: *Engineering Structures* (2022).
- [4] Susanne Heyden, Ola Dahlblom, Anders Olsson och Göran Sandberg. *Introduktion till strukturmekaniken*. Studentlitteratur, 2021.
- [5] Ulrika Nyström. “Deign with regard to explosions”. Examensarb. Chalmers University of Technology, 2006.
- [6] Anil K. Chopra. *Dynamics of Structures*. 5. utg. Pearson, 2020.
- [7] Per Gunnar Burström. *Byggnadsmaterial: tillverkning, egenskaper och användning*. Studentlitteratur, 2021.

Appendix A

```
function modalnd
%%%% Denna funktion beräknar förskjutningarna i ett 2DOF mass-fjäder-system.
%%%% Modal analys används.
%%%% Dämpning beaktas inte.

% — Inputs — %
% — Styvheter — %
k1 = 483.84*10^3; % Styvheten hos glaset [N/m]
k2 = 350*10^3; % Styvheten hos impaktorn [N/m]
K = [k1+k2 -k2; % Styvhetsmatris för systemet
     -k2 k2];

% — Massor — %
m1 = 14.58; % Glasets bidragande massa [kg]
m2 = 50; % Impaktorns massa [kg]
M = [m1 0; % Massmatrisen för systemet
     0 m2];

% — Randvillkor position — %
u1_0 = 0; % Glasets initiala position [m]
u2_0 = 0; % Impaktorns initiala position [m]
u_0 = [u1_0;
       u2_0];

% — Randvillkor hastighet — %
v1_0 = 0; % Glasets initiala hastighet [m/s]
v2_0 = -2.97; % Impaktorns hastighet vid stöten [m/s]
v_0 = [v1_0;
       v2_0];

% — Tid — %
T = 0.050; % Simuleringens längd [s]
N = 1500; % Simuleringssteg [-]
t = linspace(0,T,N);
% ————— %

% — Beräkningar — %
% Egenvektorer och egenvärden beräknas
[lambda,phi] = eigen(K,M); % phi'*M*phi = I (normaliserade)
omega = sqrt(lambda); % Svängningsfrekvenserna för egenmoderna

% Konstanterna till q(t)
Avec = phi'*M*u_0;
Bvec = phi'*M*v_0./omega;

% Funktionsvärden för q(t) samt u(t)
Q = q(t,Avec,Bvec,omega); % Se längst ner
u = phi*Q;
% ————— %

% — Modellkriteriet — %
mVar = 0;
for i = 1:N
    if u(2,i) > u(1,i) && mVar == 0
        mBreak = t(i);
        disp(['Modellkravet bröts vid t=',num2str(round(1000*mBreak,1)),' ms'])
        mVar = 1;
        break
    end
end

if mVar == 0
    warning('Modellkriteriet bröts ej! utöka simuleringens tidsspann.')
end
if mVar == 1 && mBreak < 0.5*t(N)
    warning('Modellkriteriet bryts tidigt i simuleringen')
end
% ————— %

% — Plot — %
plot(t,1000*u(1,:),color='blue') % *1000 för att få det i millimeter
hold on
grid on
xlabel('Tid [s]')
ylabel('Utböjning [mm]')
plot(t,1000*u(2,:),color='black')
if mVar == 0 % Modellkravet ej brutet
    legend('Glas','Impaktor',location='n')

    [dismax, dist] = max(abs(u(1,:))); % Glasets maximala utböjning
    dismaxStr = num2str(round(dismax*1000,1));
    distStr = num2str(round(1000*t(dist),1));
    disp(['Maximal utböjning av glaset: ', dismaxStr, ' mm (t=',distStr, ' ms)'])
elseif mVar == 1 % Modellkravet brutet
    plot([mBreak,mBreak],[1000*min(min(u)),1000*max(max(u))],color='red')
    legend('Glas','Impaktor','Modell bruten',location='n',orientation='horizontal')

    idx = find(t == mBreak); % Leta enbart efter värden innan modellen bröts
```

```

[dismax,dist] = max(abs(u(1,1:idx))); % Glasets maximala utböjning
dismaxStr = num2str(round(dismax*1000,1));
distStr = num2str(round(1000*t(dist),1));
disp(['Maximal utböjning av glaset: ', dismaxStr, ' mm      (t=',distStr, ' ms)'])
end
% ----- %

function qout = q(t,A,B,omega)
% Efter att diff. ekv. har diagonaliserats reduceras den till 2 stycken
% okopplade diff. ekv. vilka har lösning på formen A*cos(wt)+B*sin(wt). Denna
% funktionen beräknas värdena på lösningarna. Output blir en matris 2xN.
q1 = A(1)*cos(omega(1)*t)+B(1)*sin(omega(1)*t);
q2 = A(2)*cos(omega(2)*t)+B(2)*sin(omega(2)*t);
qout = [q1;q2];

```

Appendix B

```
function modalndsigma
%%%% Producerar spänningsgrafer för modalnd för olika:
%%%% Längder på glasets (balkens längd) [Lvec]
%%%% Bredder på glasets (balkens bredd på tvärsnittet) [Bvec]
%%%% Fallhöjder för impaktorn [hvec]
%%%% Tjocklekar på glasets [dvec]
%%%% G = 'J' ger grafer, annars bara tabeller
%%%% Y = 'J' ger ett fast spann på y-axeln

%%%% Spänningen plottas enbart tills dess att modellen bryts.

%%%% Grafförklaring:
% Lila: L=0.75
% Röd: L=1
% Blå: L=1.25
% Svart: L=1.5

% Punkt-streckad: B=0.75
% Heldragen: B=1
% Streckad: B=1.25
% Prickad: B=1.5

Lvec = [0.75;1;1.25;1.5];
Bvec = [0.75;1;1.25;1.5];
hvec = [0.19;0.45;1.2];
dvec = [0.006;0.008;0.01;0.012];

G = 'J';
Y = 'J';

% - Tabell-förberedelser - %
Lt = zeros(length(Lvec)*length(Bvec),1);
Bt = zeros(length(Lvec)*length(Bvec),1);
for a = 1:length(Lvec)
    Lt((a-1)*length(Bvec)+1:a*length(Bvec)) = Lvec(a)*ones(length(Bvec),1);
end
for a = 1:length(Bvec)
    Bt((a-1)*length(Lvec)+1:a*length(Lvec)) = Bvec;
end

% - Tid - %
T = 0.073; % Simuleringens längd [s]
N = 1500; % Simuleringssteg [-]
t = linspace(0,T,N);

% Alla kombinationer av inputs ska ge varsin graf
for i = 1:length(dvec)
    for j = 1:length(hvec)
        sigmavec = zeros(N,length(Lvec)*length(Bvec));
        breakvec = zeros(length(Lvec)*length(Bvec),1);
        idxvec = zeros(length(Lvec)*length(Bvec),1);
        reasonvec = cell(length(Lvec)*length(Bvec),1);
        disp(['Figur: ', num2str(3*(i-1)+j), ' (h=', num2str(hvec(j)), 'm ', 'd=', num2str(1000*dvec(i)),
            'mm)'])
        for k = 1:length(Lvec)
            for n = 1:length(Bvec)
                % --- Inputs --- %
                % - Materialegenskaper - %
                W = Bvec(n)*(dvec(i)^2)/6; % Glasets böjmostånd [m^3]
                I = Bvec(n)*(dvec(i)^3)/12; % Yttröghetsmoment [m^4]
                E = 70*10^9; % Elasticitetsmodul för glas [Pa]

                % - Styvheter - %
                k1 = 48*E*I/(Lvec(k)^3); % Styvheten hos glasets [N/m]
                k2 = 350*10^3; % Styvheten hos impaktorn [N/m]
                K = [k1+k2 -k2;
                    -k2 k2]; % Styvhetsmatris för systemet

                % - Massor - %
                rho = 2500; % Densitet för glas [kg/m^3]
                m1 = 0.486*rho*Bvec(n)*dvec(i)*Lvec(k); % Glasets bidragande massa [kg]
                m2 = 50; % Impaktorns massa [kg]
                M = [m1 0;
                    0 m2]; % Massmatrisen för systemet

                % - Randvillkor position - %
                u1_0 = 0; % Glasets initiala position [m]
                u2_0 = 0; % Impaktorns initiala position [m]
                u_0 = [u1_0;
                    u2_0];

                % - Randvillkor hastighet - %
                v1_0 = 0; % Glasets initiala hastighet [m/s]
                v2_0 = -sqrt(2*9.81*hvec(j)); % Impaktorns hastighet vid stöten [m/s]
                v_0 = [v1_0;
                    v2_0];

                % ----- %
            end
        end
    end
end
```

```

% — Beräkningar — %
% Egenvektorer och egenvärden beräknas
[lambda, phi] = eigen(K,M); % phi'*M*phi = I (normaliserade)
omega = sqrt(lambda); % Svängningsfrekvenserna för egenmoderna

% Konstanterna till q(t)
A = phi'*M*u_0;
B = phi'*M*v_0./omega;

% Funktionsvärden för q(t) samt u(t)
Q = q(t,A,B,omega); % Se längst ner
u = phi*Q;

% Spänning i glaset
sigmavec(:, length(Lvec)*(k-1)+n) = -u(1,:) * 12 * E * I / W;
% ————— %

mVar = 0;
for a = 1:N
    if abs(sigmavec(a, length(Lvec)*(k-1)+n)) >= 200000000
        mVar = -1;
        mBreak = t(a);
        breakvec(length(Lvec)*(k-1)+n) = mBreak;
        idxvec(length(Lvec)*(k-1)+n) = a;
        sigmavec(a, length(Lvec)*(k-1)+n) = 200000000;
        break
    end
    % — Modellkriteriet — %
    if u(2,a) > u(1,a) && mVar == 0
        mBreak = t(a);
        mVar = 1;
        breakvec(length(Lvec)*(k-1)+n) = mBreak;
        idxvec(length(Lvec)*(k-1)+n) = a;
        break
    end
end
if mVar == 0
    warning('Modellkriteriet bröts ej! utöka simuleringens tidsspann.')
end
% ————— %

if mVar == 0
    reasonvec{length(Lvec)*(k-1)+n} = 'Ej Bruten';
elseif mVar == 1
    reasonvec{length(Lvec)*(k-1)+n} = 'Bivillkor';
elseif mVar == -1
    reasonvec{length(Lvec)*(k-1)+n} = 'För hög spänning';
end
end
end
s = sigmavec/1000000;
if G == 'J'
    figure('Name', 'Spänning modalnd', 'Numbertitle', 'Off')
    plot(t(1:idxvec(1)), s(1:idxvec(1)), 1, '-m')
    hold on
    plot(t(1:idxvec(2)), s(1:idxvec(2)), 2, '-m')
    plot(t(1:idxvec(3)), s(1:idxvec(3)), 3, '-m')
    plot(t(1:idxvec(4)), s(1:idxvec(4)), 4, 'm', 'LineWidth', 1.5)
    plot(t(1:idxvec(5)), s(1:idxvec(5)), 5, '-r')
    plot(t(1:idxvec(6)), s(1:idxvec(6)), 6, '-r')
    plot(t(1:idxvec(7)), s(1:idxvec(7)), 7, '-r')
    plot(t(1:idxvec(8)), s(1:idxvec(8)), 8, 'r', 'LineWidth', 1.5)
    plot(t(1:idxvec(9)), s(1:idxvec(9)), 9, '-b')
    plot(t(1:idxvec(10)), s(1:idxvec(10)), 10, '-b')
    plot(t(1:idxvec(11)), s(1:idxvec(11)), 11, '-b')
    plot(t(1:idxvec(12)), s(1:idxvec(12)), 12, 'b', 'LineWidth', 1.5)
    plot(t(1:idxvec(13)), s(1:idxvec(13)), 13, '-black')
    plot(t(1:idxvec(14)), s(1:idxvec(14)), 14, '-black')
    plot(t(1:idxvec(15)), s(1:idxvec(15)), 15, '-black')
    plot(t(1:idxvec(16)), s(1:idxvec(16)), 16, 'black', 'LineWidth', 1.5)
    for a = 1:length(Lvec)*length(Bvec)
        if strcmp(reasonvec{a}, 'För hög spänning')
            plot(t(idxvec(a)), 200, 'r')
        end
    end
    grid on
    xlabel('tid [s]')
    ylabel('Spänning [MPa]')
    if Y == 'J'
        ylim([-60 210])
    end
    xlim([0 0.08])
    title(['Figur ', num2str(3*(i-1)+j), ': h=' , num2str(hvec(j)), 'm ', 'd=' , num2str(1000*dvec(i)), 'mm'])
end

sigmamax = zeros(length(Lvec)*length(Bvec), 1);
sigmamaxidx = zeros(length(Lvec)*length(Bvec), 1);
for a = 1:length(Lvec)*length(Bvec)
    [sigmamax(a), sigmamaxidx(a)] = max(abs(s(1:idxvec(a), a)));
end

sigmamaxT = t(sigmamaxidx);

T = table(Lt, Bt, round(sigmamax, 2), round(1000*sigmamaxT), round(1000*breakvec), reasonvec, 'VariableNames', {'L', 'B', 'Max spänning [MPa]', 'Tid för max spänning [ms]', 'Bruten vid [ms]', 'Brottanledning'});
disp(T)

```

```

    end
end

disp('Grafförklaring:')
disp('_____')
disp('Lila: L=0.75')
disp('Röd: L=1')
disp('Blå: L=1.25')
disp('Svart: L=1.5')
disp('_____')
disp('Punkt-streckad: B=0.75')
disp('Heldragen: B=1')
disp('Streckad: B=1.25')
disp('Prickad: B=1.5')
disp('_____')

function qout = q(t,A,B,omega)
% Efter att diff. ekv. har diagonaliserats reduceras den till 2 stycken
% okopplade diff. ekv. vilka har lösning på formen Acos(wt)+Bsin(wt). Denna
% funktionen beräknas värdena på lösningarna. Output blir en matris 2xN.
q1 = A(1)*cos(omega(1)*t)+B(1)*sin(omega(1)*t);
q2 = A(2)*cos(omega(2)*t)+B(2)*sin(omega(2)*t);
qout = [q1;q2];

```


Appendix C

```
function modald
%%%% Denna funktion simulerar ett 2DOF mass-fjädersystem med dämpning.
%%%% Funktionen använder sig av modal analys.

% — Inputs — %
% — Styvheter — %
k1 = 483.84*10^3; % Styvheten hos glaset [N/m]
k2 = 350000; % Styvheten hos impaktorn [N/m]
K = [k1+k2 -k2; % Styvhetsmatris för systemet
     -k2 k2];

% — Massor — %
m1 = 14.58; % Glasets bidragande massa [kg]
m2 = 50; % Impaktorns massa [kg]
M = [m1 0; % Massmatrisen för systemet
     0 m2];

% — Dämpning — %
zeta1 = 0.02; % Dämpning mod 1
zeta2 = 0.08; % Dämpning mod 2
zeta = [zeta1 zeta2];

% — Randvillkor position — %
u1_0 = 0; % Glasets initiala position [m]
u2_0 = 0; % Impaktorns initiala position [m]
u_0 = [u1_0; u2_0];

% — Randvillkor hastighet — %
v1_0 = 0; % Glasets initiala hastighet [m/s]
v2_0 = -2.97; % Impaktorns hastighet vid stöten [m/s]
v_0 = [v1_0; v2_0];

% — Tid — %
T = 0.055; % Simuleringens längd [s]
N = 150; % Simuleringssteg [-]
t = linspace(0,T,N);
% ————— %

% — Beräkningar — %
% Eigenvektorer och egenvärden beräknas
[lambda, phi] = eigen(K,M); % phi'*M*phi = I (normaliserade)
omega = sqrt(lambda); % Svängningsfrekvenserna för egenmoderna
omegaD = omega.*sqrt(1-zeta.^2); % Dämpad egenfrekvens

% q(0) samt q'(0) bestäms
q_0 = phi'*M*u_0;
qDot_0 = phi'*M*v_0;

% Funktionsvärden för q(t) och u(t)
Q = q(t, q_0, qDot_0, omega, zeta, omegaD); % Se längst ner
u = phi*Q;
% ————— %

% — Modellkriteriet — %
Cdiag = [2*zeta(1)*omega(1) 0; % Diagonaliserad C-matris
          0 2*zeta(2)*omega(2)];

C = inv(phi')*Cdiag*inv(phi);
c2 = -C(2,1);

coeff = q_0.*omegaD+(qDot_0.*zeta.*omega+(zeta.^2).*(omega.^2).*q_0)./omegaD;

mVar = 0;
for i = 1:N
    QDot = qDot(t(i), qDot_0, omega, zeta, omegaD, coeff);
    uDot = phi*QDot; % Detta är en vektor på formen 2x1
    if k2*(u(2,i)-u(1,i))+c2*(uDot(2)-uDot(1)) > 0 && mVar == 0
        mBreak = t(i);
        disp([ 'Modellkravet bröts vid t=', num2str(round(1000*mBreak,1)), ' ms' ])
        mVar = 1;
        break
    end
end
if mVar == 0
    warning('Modellkriteriet bröts ej! Utöka simuleringens tidsspann.')
end
if mVar == 1 && mBreak < 0.5*t(N)
    warning('Modellkriteriet bryts tidigt i simuleringen')
end
% ————— %

% — Plot — %
plot(t, 1000*u(1,:), color='blue')
hold on
grid on
xlabel('Tid [s]')
```

```

ylabel('Utböjning [mm]')
plot(t,1000*u(2,:),color='black')
if mVar == 0 % Modellkravet ej brutet
    legend('Glas','Impaktorn',location='n')

    [dismax,dist] = max(abs(u(1,:)));
    dismaxStr = num2str(round(dismax*1000,1));
    distStr = num2str(round(1000*t(dist),1));
    disp(['Maximal utböjning av glaset: ', dismaxStr, ' mm (t=',distStr, ' ms)'])
elseif mVar == 1 % Modellkravet brutet
    plot([mBreak,mBreak],[1000*min(min(u)),1000*max(max(u))],color='red')
    legend('Glas','Impaktorn','Modell bruten',location='n',orientation='horizontal')

    idx = find(t == mBreak);

    [dismax,dist] = max(abs(u(1,1:idx)));
    dismaxStr = num2str(round(1000*dismax,1));
    distStr = num2str(round(1000*t(dist),1));
    disp(['Maximal utböjning av glaset: ', dismaxStr, ' mm (t=',distStr, ' ms)'])
end
% ----- %

function qout = q(t,q_0,qDot_0,omega,zeta,omegaD)
% Efter att systemet har diagonaliserats återstår 2 stycken okopplade diff.
% ekv. Denna funktion ger ut värden på lösningen vid jämna tidsintervall i
% form av en matris på formen 2xN.

coeff = (qDot_0+zeta.*omega.*q_0)./omegaD; % Lite uppstädning i nedan

% Lösningen till den diagonaliserade ekvationen
q1 = exp(-zeta(1)*omega(1)*t).*(q_0(1)*cos(omegaD(1)*t)+coeff(1)*sin(omegaD(1)*t));
q2 = exp(-zeta(2)*omega(2)*t).*(q_0(2)*cos(omegaD(2)*t)+coeff(2)*sin(omegaD(2)*t));

qout = [q1;q2];

function qDotOut = qDot(t,qDot_0,omega,zeta,omegaD,coeff)
%% Denna funktion beräknar hastigheten av glaset och impaktorn. Detta
%% används för att kolla modellkriteriet.

qDot1 = exp(-zeta(1)*omega(1)*t).*(qDot_0(1)*cos(omegaD(1)*t)-coeff(1)*sin(omegaD(1)*t));
qDot2 = exp(-zeta(2)*omega(2)*t).*(qDot_0(2)*cos(omegaD(2)*t)-coeff(2)*sin(omegaD(2)*t));

qDotOut = [qDot1;qDot2];

```

Appendix D

```
function modaldsigma
%%%% Producerar spänningsgrafer för modald för olika:
%%%% Längder på glaset (balkens längd) [Lvec]
%%%% Bredder på glaset (balkens bredd på tvärsnittet) [Bvec]
%%%% Fallhöjder för impaktorn [hvec]
%%%% Tjocklekar på glaset [dvec]
%%%% G = 'J' ger grafer, annars bara tabeller
%%%% Y = 'J' ger ett fast spann på y-axeln

%%%% Spänningen plottas enbart tills dess att modellen bryts.

%%%% Inmata alla inputs som kolonnvektorer! (Förutom de sista två)

%%%% Grafförklaring:
% Lila: L=0.75
% Röd: L=1
% Blå: L=1.25
% Svart: L=1.5

% Punkt-streckad: B=0.75
% Heldragen: B=1
% Streckad: B=1.25
% Prickad: B=1.5

Lvec = [0.75;1;1.25;1.5];
Bvec = [0.75;1;1.25;1.5];
hvec = [0.19;0.45;1.2];
dvec = [0.006;0.008;0.01;0.012];

G = 'J';
Y = 'J';

% - Tabell-förberedelser - %
Lt = zeros(length(Lvec)*length(Bvec),1);
Bt = zeros(length(Lvec)*length(Bvec),1);
for a = 1:length(Lvec)
    Lt((a-1)*length(Bvec)+1:a*length(Bvec)) = Lvec(a)*ones(length(Bvec),1);
end
for a = 1:length(Bvec)
    Bt((a-1)*length(Bvec)+1:a*length(Bvec)) = Bvec;
end

% - Tid - %
T = 0.09; % Simuleringens längd [s]
N = 1500; % Simuleringssteg [-]
t = linspace(0,T,N);

% - Dämpning - %
zeta1 = 0.02; % Dämpning mod 1
zeta2 = 0.08; % Dämpning mod 2
zeta = [zeta1 zeta2];

% För att underlätta avläsning av graf (Krånglar, ta det sen)
% colorvec = {'Lila','Lila','Lila','Lila','Röd','Röd','Röd','Röd','Blå','Blå','Blå','Blå','Svart','Svart','Svart','Svart'};
% lineTypevec = {'Punkt-streckad','Heldragen','Streckad','Prickad','Punkt-streckad','Heldragen','Streckad','Prickad','Punkt-streckad','Heldragen','Streckad','Prickad'};

% Alla kombinationer av inputs ska ge varsin graf
for i = 1:length(dvec)
    for j = 1:length(hvec)
        sigmavec = zeros(N,length(Lvec)*length(Bvec));
        breakvec = zeros(length(Lvec)*length(Bvec),1);
        idxvec = zeros(length(Lvec)*length(Bvec),1);
        reasonvec = cell(length(Lvec)*length(Bvec),1);
        disp(['Figur: ', num2str(3*(i-1)+j), ' (h=', num2str(hvec(j)), 'm ', 'd=', num2str(1000*dvec(i)), 'mm)'])
        for k = 1:length(Lvec)
            for n = 1:length(Bvec)
                % --- Inputs --- %
                % - Materialegenskaper - %
                W = Bvec(n)*(dvec(i)^2)/6; % Glaset böjmostånd [m^3]
                I = Bvec(n)*(dvec(i)^3)/12; % Yttreöghetsmoment [m^4]
                E = 70*10^9; % Elasticitetsmodul för glas [Pa]

                % - Styvheter - %
                k1 = 48*E*I/(Lvec(k)^3); % Styvheten hos glaset [N/m]
                k2 = 350*10^3; % Styvheten hos impaktorn [N/m]
                K = [k1+k2 -k2; -k2 k2]; % Styvhetsmatris för systemet

                % - Massor - %
                rho = 2500; % Densitet för glas [kg/m^3]
                m1 = 0.486*rho*Bvec(n)*dvec(i)*Lvec(k); % Glaset bidragande massa [kg]
                m2 = 50; % Impaktorns massa [kg]
                M = [m1 0; 0 m2]; % Massmatrisen för systemet
            end
        end
    end
end
```

```

% - Randvillkor position - %
u1_0 = 0; % Glasets initiala position [m]
u2_0 = 0; % Impaktorns initiala position [m]
u_0 = [u1_0;
        u2_0];

% - Randvillkor hastighet - %
v1_0 = 0; % Glasets initiala hastighet [m/s]
v2_0 = -sqrt(2*9.81*hvec(j)); % Impaktorns hastighet vid stöten [m/s]
v_0 = [v1_0;
        v2_0];

% ----- %
% ----- Beräkningar ----- %
% Egenvektorer och egenvärden beräknas
[lambda, phi] = eigen(K,M); % phi'*M*phi = I (normaliserade)
omega = sqrt(lambda); % Svängningsfrekvenserna för egenmoderna
omegaD = omega.*sqrt(1-zeta.^2); % Dämpad egenfrekvens

% q(0) samt q'(0) bestäms
q_0 = phi'*M*u_0;
qDot_0 = phi'*M*v_0;

% Funktionsvärden för q(t) och u(t)
Q = q(t,q_0,qDot_0,omega,zeta,omegaD); % Se längst ner
u = phi*Q;

% Spänning i glaset
sigmavec(:,length(Lvec)*(k-1)+n) = -u(1,:)*12*E*I/W;
% ----- %

% ----- Modellkriteriet ----- %
Cdiag = [2*zeta(1)*omega(1) 0;
          0 2*zeta(2)*omega(2)]; % Diagonaliserad C-matris

C = inv(phi')*Cdiag*inv(phi);
c2 = -C(2,1);

coeff = q_0.*omegaD+(qDot_0.*zeta.*omega+(zeta.^2).*(omega.^2).*q_0)./omegaD;

mVar = 0;
for a = 1:N
    if abs(sigmavec(a,length(Lvec)*(k-1)+n)) >= 200000000 % Kollar om spänningen
        överstiger 200MPa
        mVar = -1;
        mBreak = t(a);
        breakvec(length(Lvec)*(k-1)+n) = mBreak;
        idxvec(length(Lvec)*(k-1)+n) = a;
        sigmavec(a,length(Lvec)*(k-1)+n) = 200000000;
        break
    end
    QDot = qDot(t(a),qDot_0,omega,zeta,omegaD,coeff);
    uDot = phi*QDot; % Detta är en vektor på formen 2x1
    if k2*(u(2,a)-u(1,a))+c2*(uDot(2)-uDot(1)) > 0 && mVar == 0
        mBreak = t(a);
        mVar = 1;
        breakvec(length(Lvec)*(k-1)+n) = mBreak;
        idxvec(length(Lvec)*(k-1)+n) = a;
        break
    end
end
if mVar == 0
    warning('Modellkriteriet bröts ej! Utöka simuleringens tidsspann.')
end
% ----- %

if mVar == 0
    reasonvec{length(Lvec)*(k-1)+n} = 'Ej Bruten';
elseif mVar == 1
    reasonvec{length(Lvec)*(k-1)+n} = 'Bivillkor';
elseif mVar == -1
    reasonvec{length(Lvec)*(k-1)+n} = 'För hög spänning';
end
end
end
s = sigmavec/1000000;
if G == 'J'
    figure('Name','Spänning modald','NumberTitle','Off')
    plot(t(1:idxvec(1)),s(1:idxvec(1)),1),'-m')
    hold on
    plot(t(1:idxvec(2)),s(1:idxvec(2)),2),'-m')
    plot(t(1:idxvec(3)),s(1:idxvec(3)),3),'-m')
    plot(t(1:idxvec(4)),s(1:idxvec(4)),4),'m','LineWidth',1.5)
    plot(t(1:idxvec(5)),s(1:idxvec(5)),5),'-r')
    plot(t(1:idxvec(6)),s(1:idxvec(6)),6),'-r')
    plot(t(1:idxvec(7)),s(1:idxvec(7)),7),'-r')
    plot(t(1:idxvec(8)),s(1:idxvec(8)),8),'r','LineWidth',1.5)
    plot(t(1:idxvec(9)),s(1:idxvec(9)),9),'-b')
    plot(t(1:idxvec(10)),s(1:idxvec(10)),10),'-b')
    plot(t(1:idxvec(11)),s(1:idxvec(11)),11),'-b')
    plot(t(1:idxvec(12)),s(1:idxvec(12)),12),'b','LineWidth',1.5)
    plot(t(1:idxvec(13)),s(1:idxvec(13)),13),'-black')
    plot(t(1:idxvec(14)),s(1:idxvec(14)),14),'-black')
    plot(t(1:idxvec(15)),s(1:idxvec(15)),15),'-black')
    plot(t(1:idxvec(16)),s(1:idxvec(16)),16),'black','LineWidth',1.5)
    hold on
    for a = 1:length(Lvec)*length(Bvec)
        if strcmp(reasonvec{a},'För hög spänning')
            plot(t(idxvec(a)),200,'*r')
        end
    end
end

```

```

        end
        end
        grid on
        xlabel('tid [s]')
        ylabel('Spänning [MPa]')
        if Y == 'J'
            ylim([-60 210])
        end
        xlim([0 0.09])
        %legend('LxB=1x1','LxB=1x1.25','LxB=1x1.5','LxB=1.25x1','LxB=1.25x1.25','LxB=1.25x1.5',
        %      'LxB=1.5x1','LxB=1.5x1.25','LxB=1.5x1.5','location','sw')
        title(['Figur ',num2str(3*(i-1)+j),' : h= ',num2str(hvec(j)),' m ', 'd= ',num2str(1000*dvec(
            i)),' mm'])
    end

    sigmamax = zeros(length(Lvec)*length(Bvec),1);
    sigmamaxidx = zeros(length(Lvec)*length(Bvec),1);
    for a = 1:length(Lvec)*length(Bvec)
        [sigmamax(a), sigmamaxidx(a)] = max(abs(s(1:idxvec(a),a)));
    end

    sigmamaxT = t(sigmamaxidx);
    T = table(Lt,Bt,round(sigmamax,2),round(1000*sigmamaxT'),round(1000*breakvec), reasonvec, '
        VariableNames', {'L','B','Max spänning [MPa]','Vid tiden [ms]','Bruten vid [ms]','
        Brottanledning'});
    disp(T)
    %disp(['Graf: ', num2str(3*(i-1)+j), ' Bruten: ', num2str(max(breakvec)),' Max: ',num2str(
        max(max(abs(sigmavec)))/1000000)])
end
end

disp('Grafförklaring:')
disp('-----')
disp('Lila: L=0.75')
disp('Röd: L=1')
disp('Blå: L=1.25')
disp('Svart: L=1.5')
disp('-----')
disp('Punkt-streckad: B=0.75')
disp('Heldragen: B=1')
disp('Streckad: B=1.25')
disp('Prickad: B=1.5')
disp('-----')

function qout = q(t,q_0,qDot_0,omega,zeta,omegaD)
% Efter att systemet har diagonaliserats återstår 2 stycken okopplade diff.
% ekv. Denna funktion ger ut värden på lösningen vid jämna tidsintervall i
% form av en matris på formen 2xN.

coeff = (qDot_0+zeta.*omega.*q_0)./omegaD; % Lite uppstädning i nedan

% Lösningen till den diagonaliserade ekvationen
q1 = exp(-zeta(1)*omega(1)*t).*(q_0(1)*cos(omegaD(1)*t)+coeff(1)*sin(omegaD(1)*t));
q2 = exp(-zeta(2)*omega(2)*t).*(q_0(2)*cos(omegaD(2)*t)+coeff(2)*sin(omegaD(2)*t));

qout = [q1;q2];

function qDotOut = qDot(t,qDot_0,omega,zeta,omegaD,coeff)
%%% Denna funktion beräknar hastigheten av glaset och impaktorn. Detta
%%% används för att kolla modellkriteriet.

qDot1 = exp(-zeta(1)*omega(1)*t).*(qDot_0(1)*cos(omegaD(1)*t)-coeff(1)*sin(omegaD(1)*t));
qDot2 = exp(-zeta(2)*omega(2)*t).*(qDot_0(2)*cos(omegaD(2)*t)-coeff(2)*sin(omegaD(2)*t));

qDotOut = [qDot1;qDot2];

```


Appendix E

```
function newmark
%%%% Denna funktion simulerar ett 2DOF mass-fjädersystem.
%%%% Newmarks metod används med constant average acceleration.
%%%% Systemet löses i fysiska koordinater.

% — Inputs — %
% — Styvheter — %
k1 = 483.84*10^3; % Styvheten hos glaset [N/m]
k2 = 350000; % Styvheten hos impaktorn [N/m]
K = [k1+k2 -k2; % Styvhetsmatris för systemet
     -k2 k2];

% — Massor — %
m1 = 14.58; % Glasets bidragande massa [kg]
m2 = 50; % Impaktorns massa [kg]
M = [m1 0; % Massmatrisen för systemet
     0 m2];

% — Dämpning — %
zeta1 = 0.02; % Dämpning mod 1
zeta2 = 0.08; % Dämpning mod 2
zeta = [zeta1 zeta2];

% — Randvillkor position — %
u1_0 = 0; % Glasets initiala position [m]
u2_0 = 0; % Impaktorns initiala position [m]
u_0 = [u1_0;
       u2_0];

% — Randvillkor hastighet — %
v1_0 = 0; % Glasets initiala hastighet (sätts till noll) [m/s]
v2_0 = -2.97; % Impaktorns hastighet vid stöten (beräknas utifrån testets förutsättningar) [m/s]
v_0 = [v1_0;
       v2_0];

% — Tid — %
T = 0.055; % Simuleringens längd [s]
deltaT = 0.00055; % Simuleringsstegens längd [s]
N = ceil(T/deltaT); % Antal simuleringssteg [-]

% — Parametrar — %
gamma = 0.5; % Dessa värden gäller för
beta = 0.25; % constant average acceleration
% ————— %

% — Beräkningar — %
% Initierar variabler
uDdot = zeros(2,N+1);
uDot = zeros(2,N+1);
u = zeros(2,N+1);

% Tar fram konstanten c2
[lambda,phi] = eigen(K,M);
omega = sqrt(lambda);
Cdiag = [2*zeta(1)*omega(1) 0;
         0 2*zeta(2)*omega(2)];
C = inv(phi')*Cdiag*inv(phi);
c2 = -C(1,2);

% Randvillkor
uDdot(:,1) = -M\((C*v_0+K*u_0);
uDot(:,1) = v_0;
u(:,1) = u_0;

% Hjälpkoefficienter
A1 = gamma/(beta*deltaT)*C+1/(beta*(deltaT)^2)*M;
A2 = 1/(beta*deltaT)*M+(gamma/beta-1)*C;
A3 = (1/(2*beta)-1)*M+deltaT*(gamma/(2*beta)-1)*C;
Khat = K+A1;

mVar = 0;

for i = 1:N
    Phat = A1*u(:,i)+A2*uDot(:,i)+A3*uDdot(:,i);
    u(:,i+1) = Khat\Phat;
    uDot(:,i+1) = gamma/(beta*deltaT)*(u(:,i+1)-u(:,i))+(1-gamma/beta)*uDot(:,i)+deltaT*(1-gamma/(2*beta))*uDdot(:,i);
    uDdot(:,i+1) = 1/(beta*(deltaT)^2)*(u(:,i+1)-u(:,i))-1/(beta*deltaT)*uDot(:,i)-(1/(2*beta)-1)*uDdot(:,i);
end

% ————— %

% — Modellkriterie — %
for i = 1:N
    if k2*(u(2,i)-u(1,i))+c2*(uDot(2,i)-uDot(1,i)) > 0 && mVar == 0
        mBreak = deltaT*(i-1);
        disp(['Modellkravet bröts vid t=',num2str(round(1000*mBreak,1)), ' ms'])
    end
end
```

```

        mVar = 1;
        break
    end
end
if mVar == 0
    warning('Modellkriteriet bröts ej! Utöka simuleringens tidsspann.')
end
if mVar == 1 && mBreak < 0.5*deltaT*N
    warning('Modellkriteriet bryts tidigt i simuleringen')
end
% ----- %

% --- Plot --- %
tvec = linspace(0,deltaT*N,N+1);
plot(tvec,1000*u(1,:),color='blue')
hold on
grid on
xlabel('Tid [s]')
ylabel('Utböjning [mm]')
plot(tvec,1000*u(2,:),color='black')

if mVar == 0 % Ej Brutet modellkriterie
    legend('Glas','Impaktor',location='n')

    [dismax,dist] = max(abs(u(1,:)));
    dismaxStr = num2str(round(dismax*1000,1));
    distStr = num2str(round(1000*tvec(dist),1));
    disp(['Maximal utböjning av glaset: ',dismaxStr,' mm (t=',distStr,' ms)'])
elseif mVar == 1 % Brutet modellkriterie
    plot([mBreak,mBreak],[1000*min(min(u)),1000*max(max(u))],color='red')
    legend('Glas','Impaktor','Modell bruten',location='n',orientation='horizontal')

    idx = find(abs(tvec - mBreak) < 1e-10, 1);

    [dismax,dist] = max(abs(u(1,1:idx)));
    dismaxStr = num2str(round(1000*dismax,1));
    distStr = num2str(round(1000*tvec(dist),1));
    disp(['Maximal utböjning av glaset: ',dismaxStr,' mm (t=',distStr,' ms)'])
end
disp(['Antal simuleringsteg: ',num2str(N)])
% ----- %

```