

**Rakenteiden mekaniikan menetelmiä  
metallirakentajille  
OSA 2  
Ratkaisutekniikat**

Reijo Kouhia  
TKK  
Rakenteiden mekaniikka

12.10.2005 Metallirakentamisen tutkimuskeskus, Seinäjoki



# OSA 2: RATKAISUTEKNIIKAT

1. Lineaarisen algebrallisen yhtälöryhmän ratkaisu
2. Lineaaristen algebrallisten ominaisarvotehtävien ratkaisu
3. Aikaintegrointitekniikat



# LINEAARINEN YHTALÖRYHMA

## Ongelma

$$Ax = b$$

missä  $A$  on  $n \times n$  matriisi. 1-käsitteinen ratkaisu mikäli  $\det(A) \neq 0$

### $A$ :n ominaisuuksia

- symmetrinen / epäsymmetrinen
- positiivisesti definiitti / indefiniitti / positiivinen reaalinen
- rakenne / harvuus



# Joitain määritelmiä

Olkoon  $\mathbf{A}$   $n \times n$  neliömatriisi

Ominaisarvot

$$\mathbf{A}\phi_i = \lambda_i\phi_i \quad i = 1, \dots, n \quad \lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_{n-1} \leq \lambda_n$$

Determinantti  $\det(\mathbf{A}) = \prod_{i=1}^n \lambda_i$

Symmetrisen matriisin ominaisarvot ovat reaalisia

Jos  $\mathbf{A}$  on SPD niin kaikki  $\lambda_i > 0$

Jos  $\mathbf{A}$  on SPD niin  $\mathbf{A}$ :n itseisarvoltaan suurin alkio on diagonaalilla



# Algoritmien jaottelu

- Suorat Gaussin algoritmiin perustuvat menetelmät. Jaottelu matriisin varastointimuodon mukaan:
  - nauha (band)
  - muuttuva nauha (profiili, skyline, envelope)
  - aaltorintama (frontal solution)
  - harvat suorat ratkaisijat (sparse direct solvers)
- Iteratiiviset menetelmät
  - klassiset iteraatiot (Jacobi, Gauss-Seidel, SOR, SSOR)
  - Krylovin aliavaruusiteraatiot (CG, BiCG, BiCGStab, GMRES, QMR)
  - moniverkkoalgoritmit



# Yleisiä ominaisuuksia

- Suorat menetelmät
  - luotettavia
  - ratkaisuaika ennustettavissa
  - eivät ole parametririippuvia
  - vaativat paljon muistia
  - laskenta-aika kasvaa rajusti tehtävän koon kasvaessa
- Iteratiiviset menetelmät
  - luetettavia vain “hyvänlaatuisille” ongelmille
  - parametririippuvia
  - vaativat vähemmän muistia
  - nopeita (jos suppenevat hyvin)



# Suorat ratkaisutekniikat

Gaussin algoritmiin perustuvat hajotelmat

$$A = LDU, \quad \text{tai} \quad A = \bar{L}U. \quad \text{missä} \quad \bar{L} = LD$$

Choleskyn hajotelma SPD matriisille

$$A = \tilde{L}^T \tilde{L} \quad \text{missä} \quad \tilde{L} = LD^{\frac{1}{2}}$$



# Hajotelmaratkaisu

Ratkaisun

$$Ax = b$$

vaiheet Gaussin algoritmossa

- muodostetaan systeemimatriisin hajotelma

$$A = LDU$$

- kuormavektorin redusointi

$$Lz = b$$

- diagonaalilla skaalaus

$$Dy = z$$

- ja lopuksi takaisinsijoitus

$$Ux = y.$$





# LU-hajotelma

## Geneerinen muoto

```
do
  do
    do
       $A_{ij} = A_{ij} - L_{ik}/A_{kj}$ 
    end do
  end do
end do
```

do silmukoiden indeksit  $i, j$  ja  $k$  esiintyvät eri järjestyksessä.

Eri  $ijk$ -muodot viittaavat ja laskevat hajotelman  $L$  ja  $U$  tekijät eri tavalla.



# Rivimuotoinen kij-versio

```

do  $k = 1, n - 1$ 
  do  $i = k + 1, n$ 
     $L_{ik} = A_{ik}/A_{kk}$ 
    do  $j = k + 1, n$ 
       $A_{ij} = A_{ij} - L_{ik}/A_{kj}$ 
    end do
  end do
end do

```



# Sarakemuotoinen kji-versio

```

do  $k = 1, n - 1$ 
  do  $r = k + 1, n$ 
     $L_{rk} = A_{rk}/A_{kk}$ 
  end do
  do  $j = k + 1, n$ 
    do  $i = k + 1, n$ 
       $A_{ij} = A_{ij} - L_{ik}/A_{kj}$ 
    end do
  end do
end do

```



lask.  $U$ :n  $(k + 1)$ :s rivi  
päivitetään jäljelle  
jäänyt osa  $A$ :sta

lasketaan  $L$ :n  $k$ :s sarake



# Suorat ratkaisutekniikat

Jaottelu matriisin varastointimuodon mukaan:

- vakionauha (band)
- muuttuva nauha (profiili, skyline, envelope)
- aaltorintama (frontal solution)
- harvat suorat ratkaisijat (sparse direct solvers)

Symmetrisen kerroinmatriisin faktoroinnin työmäärä (kolmelle ensimmäiselle varastointitavalle) on

$$W_{\text{hajotelma}} \sim \frac{1}{2}nb_{\text{rms}}^2$$

$n$  on tuntemattomien lkm ja  $b_{\text{rms}}$  matriisin RMS puolinauhan leveys  
Kuormavektorin redusointi ja takaisinsijoitus

$$W_{\text{kuormav}} \sim 2nb_{\text{rms}}$$



# Vakionauharatkaisijat

Tuntemattomat pyrittävä järjestämään siten, että **maksimianhanleveys** minimoituisi.

Cuthill-McKee tai Gibbs-Poole-Stockmeyer algoritmit minimoivat **maksimianhanleveyttä**



# Profiiliratkaisijat

$$\begin{bmatrix}
 x & x & 0 & x & 0 & 0 & 0 & 0 \\
 & x & x & 0 & 0 & 0 & 0 & 0 \\
 & & x & x & 0 & x & 0 & 0 \\
 & & & x & x & x & 0 & 0 \\
 & & & & x & x & 0 & x \\
 & & & & & x & x & 0 \\
 & & & & & & x & x \\
 & & & & & & & x
 \end{bmatrix}$$

Tuntemattomat pyrittävä järjestämään siten, että **RMS-nauhanleveys** tai profiili minimoituisi.

Käänteinen Cuthill-McKee tai Gibbs-King algoritmit minimoivat profiilia.



# Nauharatkaisijat - vertailua

2-D lämmönjohtumisongelma

säännöllinen  $100 \times 100$  ja  $400 \times 400$  verkko bilineaarisia elementtejä

$$b_{\max} = b_{\text{rms}} + 1$$

ratkaisija	$n$	$b_{\text{rms}}$	faktorointi	kuormav.
nauha	9801	100	0.20	0.02
nauha lohkoalg.			0.15	0.02
profiili			0.24	0.02
profiili lohkoalg.			0.24	0.02
nauha	159201	400	40.78	4.72
nauha lohkoalg.			26.58	4.68
profiili			51.19	4.81
profiili lohkoalg.			36.12	1.60



# Nauharatkaisijat - vertailua

3-D kiinteän aineen malli

säännöllinen  $10 \times 10 \times 10$  ja  $20 \times 20 \times 20$  verkko trilineaarisia elementtejä

ratkaisija	$n$	$b_{\max}$	$b_{\text{rms}}$	faktorointi	kuormav.
nauha	3987	402	380	1.0	0.08
nauha lohkoalg.				0.6	0.08
profiili				1.1	0.07
profiili lohkoalg.				0.8	0.02
nauha	27777	1392	1355	157	2.8
nauha lohkoalg.				51	2.8
profiili				168	2.8
profiili lohkoalg.				62	1.7

lohkon koko  $64 \times 64$  nauharatkaisijalle ja  $64 \times 128$  profiiliratkaisijalle





# Nauhankavennusmenetelmät

Yhtälöiden uudelleennumerointi = matriisin rivien ja sarakkeiden permutointi

Graafiteoreettinen ongelma

Symmetristä  $n \times n$  matriisia vastaa  $n$ -solmuinen suuntaamaton graafi

Solmut yhdistetty tiellä jos  $a_{ij} = a_{ji} \neq 0$

**Tavoite:** minimoida naapurisolmujen erotusten itseisarvo



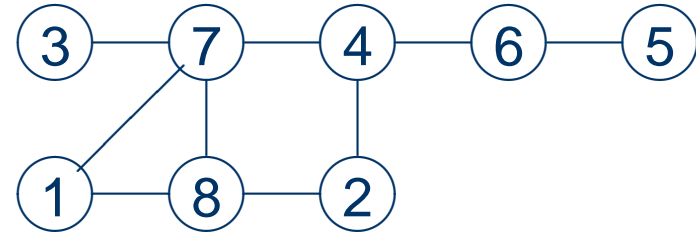
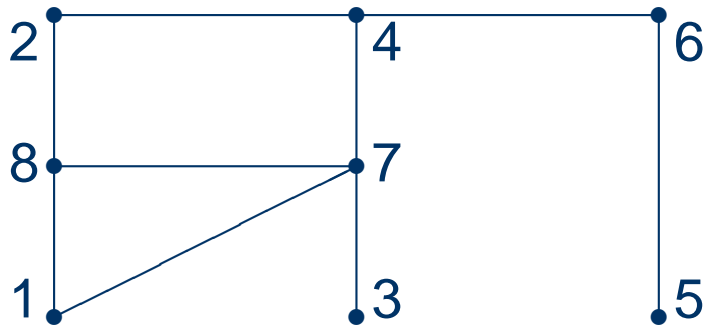
# Cuthill-McKee

1. Määrätään solmujen aste (= siihen liittyvien teiden lukumäärä), valitaan aloitussolmuksi 1 jokin minimiasteinen solmu
2. Solmun 1 naapurit asteen mukaisessa järjestyksessä, numeroidaan 2,3,... Näitä solmuja sanotaan tason 1 solmuiksi.
3. Tason 1 solujen naapurit, joilla ei ole vielä numeroa. Numeroidaan solmun 2 naapurit, sitten solmun 3 naapurit jne. Kun taso 1 käsitelty siirrytään tasolle 2, jne. Jatketaan kunnes kaikki solmut numeroitu.

HUOM: Aloitussolmun määrittely voidaan tehdä paremmin



# Esimerkki



graafin solmu	1	2	3	4	5	6	7	8
solmun aste	2	2	1	3	1	2	4	3

## Cuthill-McKee uudelleennumerointi

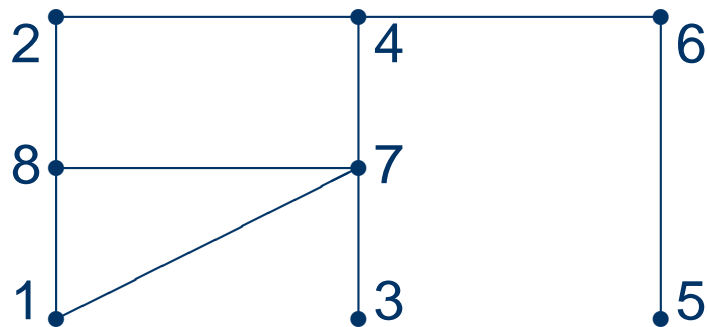
taso	$\mathcal{L}_0$	$\mathcal{L}_1$	$\mathcal{L}_2$	$\mathcal{L}_3$	$\mathcal{L}_4$
solmut	3	7	1, 4, 8	2, 6	5
uusi numerointi	1	2	3, 4, 5	6, 7	8



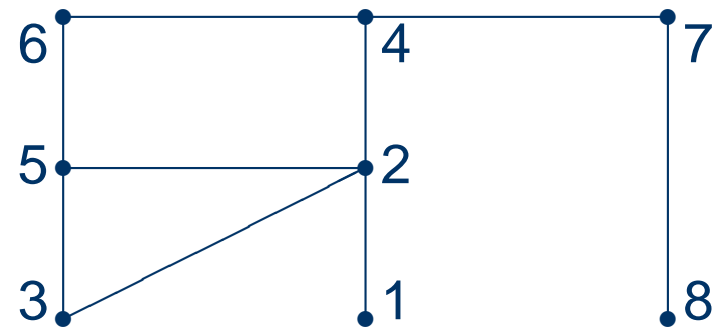
# Esimerkki, CM-numerointi

taso	$\mathcal{L}_0$	$\mathcal{L}_1$	$\mathcal{L}_2$	$\mathcal{L}_3$	$\mathcal{L}_4$
solmut	3	7	1, 4, 8	2, 6	5
uusi numerointi	1	2	3, 4, 5	6, 7	8

## Alkuperäinen

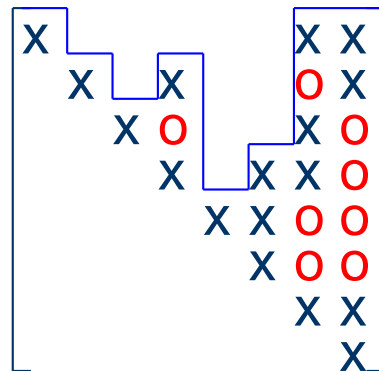
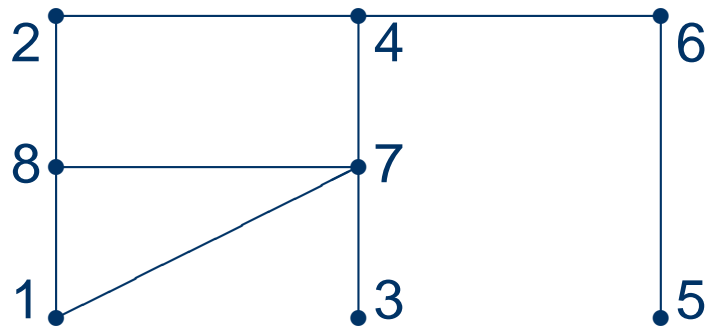


## CM numeroitu



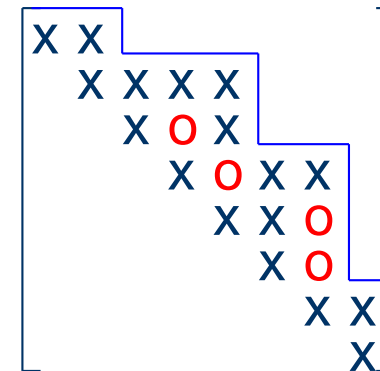
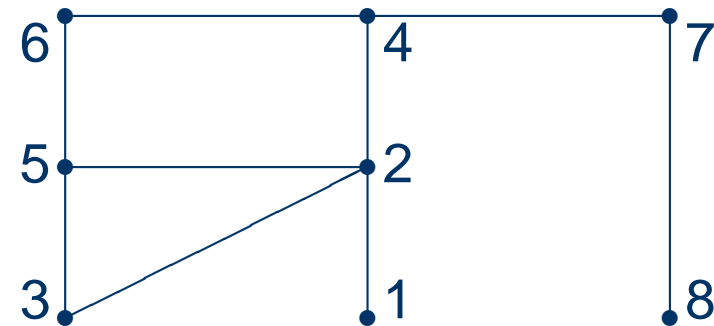
# Esimerkki, CM-numerointi

Alkuperäinen



$$b_{\max} = 8, \quad \text{prof} = 25$$

CM numeroitu

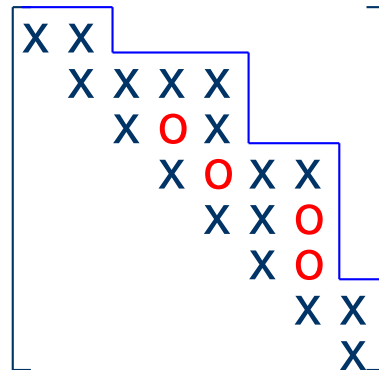
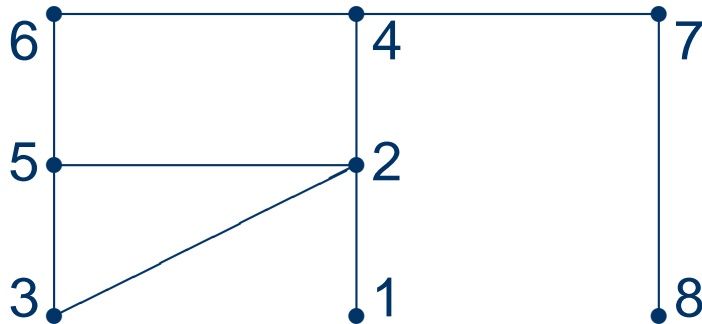


$$b_{\max} = 4, \quad \text{prof} = 21$$



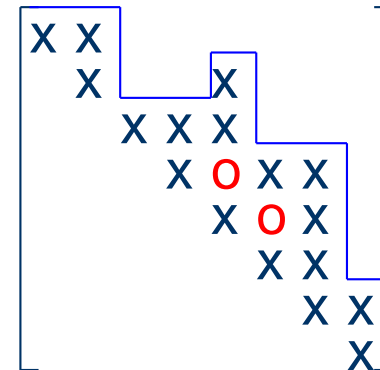
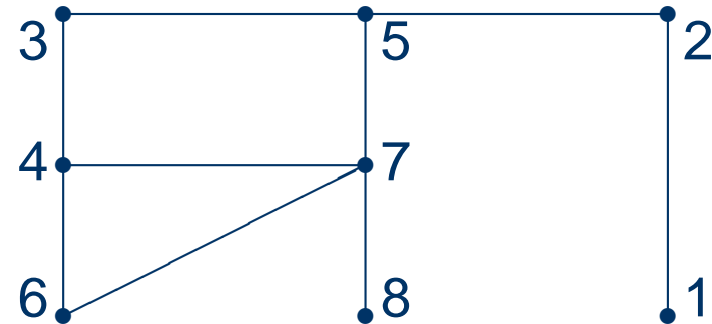
# Esimerkki, käännteinen CM

## CM numerointi



$$b_{\max} = 4, \quad \text{prof} = 21$$

## käännteinen CM numerointi



$$b_{\max} = 4, \quad \text{prof} = 19$$



# Harvat suorat ratkaisijat

Hajotelman teko kaksivaiheinen

1. symbolinen kolmiointi = toimenpiteet, joilla selvitetään  $\text{fil}(\mathbf{A})$  ja muodostetaan tietorakenteet  $\mathbf{L}$ :n tallentamiseksi
2. numeerinen hajotelman laskenta (monirintamamenetelmä)

Ennen symbolista kolmiointia on tuntemattomat pyrittävä järjestämään siten, että täytyminen minimoituisi, t.s. etsitään permutaatio  $\mathbf{P}$  ja

$$\mathbf{PAP}^T = \mathbf{LL}^T \quad \text{tai} \quad \mathbf{LU}$$



# Täyttymisen minimointi

Ongelma on löytää yhtälöiden numerointi joka minimoi täyttymisen.

- minimiaste (minimum degree), variantteja esim. Quotient MD ja Multiple MD
- alirakenne (nested dissection)

Minimiastemenetelmät ovat yleensä parempia





# Aaltorintamaratkaisija

Yhtälöiden kokoaminen ja eliminointi suoritetaan “samanaikaisesti”.

Tuntematonta ei kuitenkaan voida eliminoida ennen kuin kaikki siihen liittyvät elementit ovat koottuna.

Myös elementit on numeroitava järkevästi.

Harvoille matriiseille on kehitetty monirintamaratkaisija.



# Iteratiiviset ratkaisijat

- klassiset iteraatiot (Jacobi, Gauss-Seidel, SOR, SSOR)
- moniverkkoalgoritmi
- Krylovin aliavaruusiteraatiot (CG, BiCG, BiCGStab, GMRES, QMR)



# Klassiset iteraatiot

Stationäärinen iteraatio

$$\mathbf{x}_{k+1} = \mathbf{B}\mathbf{x}_k + \mathbf{c}$$

sekä  $\mathbf{B}$  että  $\mathbf{c}$  eivät saa riippua iteraatiokierroksesta  $k$ .

Idea

$$\mathbf{M}\mathbf{x} = (\mathbf{M} - \mathbf{A})\mathbf{x} + \mathbf{b}$$

Virhe  $\mathbf{e}_k = \mathbf{x} - \mathbf{x}_k$  toteuttaa

$$\mathbf{M}\mathbf{e}_{k+1} = (\mathbf{M} - \mathbf{A})\mathbf{e}_k \quad \Rightarrow \quad \mathbf{e}_{k+1} = \mathbf{M}^{-1}(\mathbf{M} - \mathbf{A})\mathbf{e}_k = \mathbf{B}\mathbf{e}_k$$

joten

$$\mathbf{e}_k = \mathbf{B}\mathbf{e}_{k-1} = \mathbf{B}^2\mathbf{e}_{k-2} = \cdots = \mathbf{B}^k\mathbf{e}_0$$

suppenee jos  $|\lambda_i(\mathbf{B})| < 1$



# Klassiset iteraatiot, jatkoa

Kolme ehdokasta matriisiksi  $M$  ovat:

1.  $M$  on  $A$ :n diagonaali (Jacobin menetelmä),
2.  $M$  on  $A$ :n alakolmio (Gaussin-Seidelin menetelmä),
3.  $M$  on edellisten kombinaatio, ylirelaksaatiomenetelmä, (SOR = successive overrelaxation)



# Moniverkkoratkaisija

**IDEA** useita diskretoinnin tasoja, ratkaisun matalataajuuksiset komponentit vangitaan harvassa verkossa ja korkeataajuuksikomponentit hienossa verkossa.

Algoritmin kaksi oleellista vaihetta ovat:

- relaksaatiovaihe, korkeataajuuksiset komponentit vangitaan yksinkertaisella iteraatiolla, esim. Jacobin tai Gauss-Seidelin iteraatiolla (relaksaatiota suoritetaan useilla diskreetointitasoilla),
- korjausvaihe, ratkaisun matalataajuuksiset komponentit vangitaan ratkaisemalla ongelma hyvin harvassa verkossa.

Teoriassa optimaalinen, työmääräestimaatti

$$W \sim Cn$$



# Yksinkertainen kaksiverkkoalgoritmi

1. ratkaistaan systeemi harvassa verkossa  $A_1 x_1 = b_1$ ,
2. interpoloidaan hienon verkon arvot harvasta verkosta  $x_2 = F x_1$ ,
3. suoritetaan muutama relaksaatioiteraatio hienossa verkossa,
4. muunnetaan jäännös  $r_2 = f_2 - A_2 x_2$  harvempaan verkkoon  $r_1 = C r_2$ ,
5. ratkaistaan harvan verkon korjaus  $A_1 \Delta x_1 = r_1$ ,
6. siirretään korjaus hienoon verkkoon ja lisätään edellisiin hienon verkon arvoihin  $\Delta x_2 = F \Delta x_1$ ,  $x_2^{uusi} = x_2^{vanha} + \Delta x_2$ ,
7. siirrytään kohtaan 3.

$F$  interpolaatio hienoon,  $C$  "keskiarvoistus" harvaan verkkoon,  $C = F^T$



# Krylovin aliavaruusiteraatiot

## Krylovin aliavaruus

$$\mathcal{K}_m(\mathbf{A}, \mathbf{r}_0) = \text{span} \{ \mathbf{r}_0, \mathbf{A}\mathbf{r}_0, \mathbf{A}^2\mathbf{r}_0, \dots, \mathbf{A}^{m-1}\mathbf{r}_0 \}, \quad \text{missä } \mathbf{r}_0 = \mathbf{b} - \mathbf{A}\mathbf{x}_0$$

Ratkaisu  $m$ -dimensioisesta aliavaruudesta  $\mathbf{x}_0 + \mathcal{K}_m$ .  
Vaatimus (Petrov-Galerkin ehto)

$$\mathbf{r}_m = \mathbf{b} - \mathbf{A}\mathbf{x}_m \perp \mathcal{L}_m$$



# Liittogradienttimenetelmä

Soveltuu SPD systeemeille

Ortogonaaliprojektio  $\mathcal{L} = \mathcal{K} = \mathcal{K}_m(\mathbf{A}, \mathbf{r}_0)$

Hakusuunnat  $\mathbf{d}_k$  konjugoivat, eli

$$\mathbf{d}_i^T \mathbf{A} \mathbf{d}_j = 0, \quad i \neq j.$$

Residuaalit ovat ortogonaalisia

$$\mathbf{r}_i^T \mathbf{r}_j = 0, \quad i \neq j.$$





# Algoritmi

- laske  $\mathbf{r}_0 = \mathbf{b} - \mathbf{A}\mathbf{x}_0$ ,
- aseta  $\mathbf{d}_0 = \mathbf{r}_0$ , ja laske  $\tau_0 = \mathbf{r}_0^T \mathbf{d}_0$
- iteroi  $i = 0, 1, 2, \dots$  kunnes iteraatio suppenee:
  1. laske:  $\mathbf{s} = \mathbf{A}\mathbf{d}_i$ ,  $\alpha_i = \tau_i / \mathbf{d}_i^T \mathbf{s}$
  2. päivitä  $\mathbf{x}_{i+1} = \mathbf{x}_i + \alpha_i \mathbf{d}_i$   $\mathbf{r}_{i+1} = \mathbf{r}_i - \alpha_i \mathbf{s}$
  3. laske  $\tau_{i+1} = \mathbf{r}_{i+1}^T \mathbf{r}_{i+1}$   $\beta_i = \tau_{i+1} / \tau_i$
  4. päivitä  $\mathbf{d}_{i+1} = \mathbf{z} + \beta_i \mathbf{d}_i$



# Suppeneminen

Teoriassa tarkka ratkaisu viimeistään  $n$ :n iteraation jälkeen.

$$\|\mathbf{x}_k - \mathbf{x}\|_A \leq 2 \left( \frac{\sqrt{\text{Cond}(\mathbf{A})} - 1}{\sqrt{\text{Cond}(\mathbf{A})} + 1} \right)^k \|\mathbf{x}_0 - \mathbf{x}\|_A$$

Suhteelliseen tarkkuuteen  $TOL$  vaaditaan iteraatioita

$$I \geq \frac{1}{2} \sqrt{\text{Cond}(\mathbf{A})} \log \frac{2}{TOL}.$$



# Pohjustus

Lineaariselle systeemille  $Ax = b$ , ekvivalentti pohjustettu systeemi

$$M_1^{-1}AM_2^{-1}y = M_1^{-1}b,$$

missä  $M = M_1M_2$  on pohjustinmatriisi  
 $M_1, M_2$  ovat vasemman- ja oikeanpuoleinen pohjustin

Pohjustimen  $M$  pitäisi olla

- hyvä approksimaatio  $A$ :lle, siten että  $M^{-1}A \approx I$
- edullinen muodostaa ja käyttää



# Pohjustustekniikat

Voidaan jaotella pääpiirteittäin

1. klassisiin iteraatioihin (Jacobi, SSOR) perustuvat pohjustimet
2. epätäydelliset harvat  $LU$ -hajotelmat (ILU tai IC, symmetrisille)
3. polynomipohjustimet
4. inverssipohjustimet
5. moniverkko tai monitasopohjustimet



# Algoritmi

- muodosta  $M$ , tai suoraan  $M^{-1}$  (tai faktoroitu muoto)
- laske  $r_0 = b - Ax_0$ ,
- suorita pohjustinoperaatio  $d_0 = M^{-1}r_0$ , ja laske  $\tau_0 = r_0^T d_0$
- iteroi  $i = 0, 1, 2, \dots$  kunnes supennut
  1. laske:  $s = Ad_i$ ,  $\alpha_i = \tau_i / d_i^T s$ ,
  2. päivitä  $x_{i+1} = x_i + \alpha_i d_i$ ,  $r_{i+1} = r_i - \alpha_i s$ ,
  3. pohjustinoperaatio  $z = M^{-1}r_{i+1}$  ja laske  $\tau_{i+1} = r_{i+1}^T z$ ,  $\beta_i = \tau_{i+1} / \tau_i$ ,
  4. päivitä  $d_{i+1} = z + \beta_i d_i$ .



# Epätäydellinen Cholesky

Valitaan

- harvuusrakenne samaksi kuin  $A$ :lla  
IC(0)
- harvuusrakenne dynaamisesti tiputtamalla hajotelman pienet alkiot  
IC(droptol)

**ONGELMA:** Onko epätäydellistä hajotelmaa olemassa?

On olemassa vain muutama varma algoritmi kaikille harvuusrakenteille

- Ajiz-Jennings
- Tismenetsky
- Benzi-Tuma



# Menetelmien vertailua

Asymptoottisia laskutoimitusmääräarvioita  $W \sim Cn^k$

DY k.l.	$k$ - 2 dim.	$k$ - 3 dim.	menetelmä
2	1.5	1.33	konjugaattigradietti
	1.25	1.17	pohjustettu konj. grad.
	2	2.33	Choleskyn hajotelma
	1.5	2	alirakennetekniikka
	1	1	moniverkkoalgoritmi
4	2	1.67	konjugaattigradietti
	1.67	1.5	pohjustettu konj. grad.
	2	2.33	Choleskyn hajotelma
	1.5	1.67	alirakennetekniikka
	1	1	moniverkkoalgoritmi



# Esimerkkejä - vertailua

3-D kiinteän aineen malli

säännöllinen  $10 \times 10 \times 10$  ja  $20 \times 20 \times 20$  verkko trilineaarisia elementtejä

ratkaisija	$n$	$b_{\max}$	$b_{\text{rms}}$	faktorointi	kuormav.
nauha lohkoalg.	3987	402	380	0.6	0.08
profiili lohkoalg.				0.8	0.02
nauha lohkoalg.	27777	1392	1355	51	2.8
profiili lohkoalg.				62	1.7
PCG - IC(0)				0.5	8.4





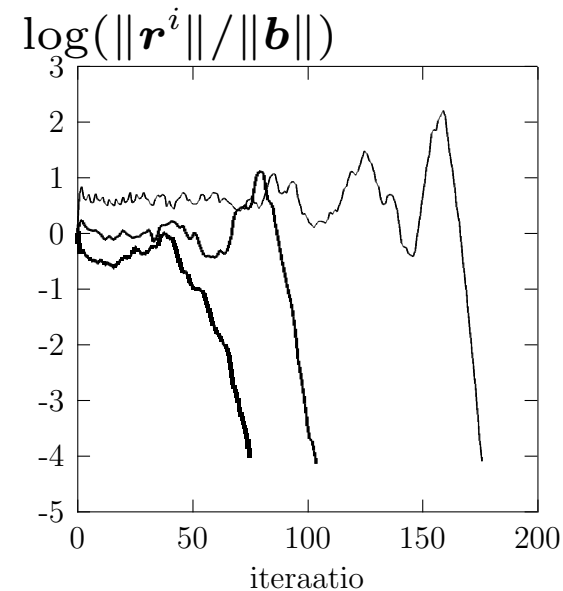
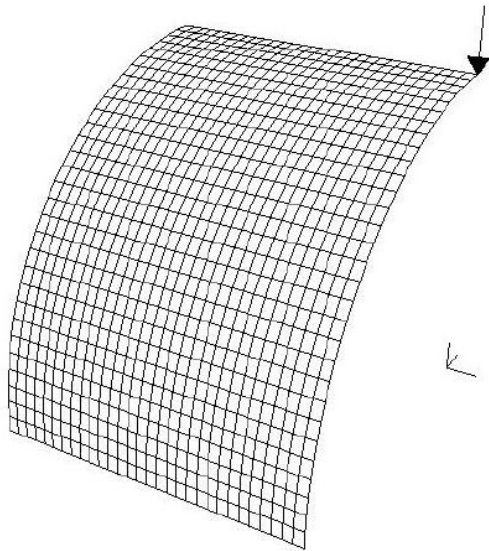
# Esimerkkejä

Sylinterikuori 30 x 30 stabiloituja MITC4 elementtejä  $n = 5489$

Kuoren suhteellisen paksuuden  $t/R$  vaikutus

IC(0)-pohjustetun konjugaattigradienttimenetelmän konvergenssiin

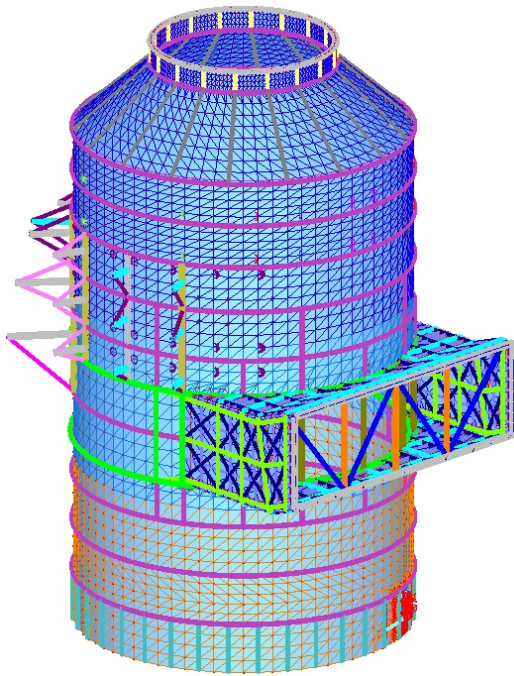
$t/R = 10^{-1}, 10^{-2}, 10^{-3}$



# Käytännön esimerkki

Juha Airola, A&S Virtual Systems Oy, ohjelma Robot Millennium

$n = 57348$ ,  $b = 1512$  (RCM optimoitu),



ratkaisija	CPU-aika**	keskusm.	levym.
profiili	451	84.0	445.8
aaltorintama	483	84.0	445.8
harva, monirint., MD	99	30.8	97.2
harva, monirint., ND	130	52.0	131.5
PCG-IC <sup>†</sup>	210	100.0	8.0
PCG-Monitasop. <sup>†,*</sup>	425	100.0	8.0

MD = minimiaste, ND = alirakenne

<sup>†</sup>  $\|\mathbf{r}_i\|_\infty < TOLR \|\mathbf{b}\|_\infty$ ,  $TOLR = 10^{-4}$

\* 3-tasoa, 3-sisäistä it.

\*\* koko ratkaisuaika yhdelle kuormitustapaukselle



# Menetelmien valinnasta

Käytä aina suoraa menetelmää jos se on mahdollista.

Iteratiivinen ratkaisu vain hyvin suurille systeemeille.

Valitse pohjustin luotettavuuden perusteella.



# Kirjallisuutta

- I.S. Duff, A.M. Erisman, J.K. Reid, *Direct Methods for Sparse Matrices*, Oxford Science Publications, 1986
- A. George, J. W. Liu, *Computer Solution of Large Sparse Positive Definite Systems*, Prentice Hall, 1981
- G. Meurant, *Computer Solution of Large Linear Systems*, Elsevier, 1999
- J.M. Ortega, *Introduction to Parallel and Vector Solution of Linear Systems*, Plenum Press, 1988
- Y. Saad, *Iterative Methods for Sparse Linear Systems*, PWS Publishing, 1996
- H.A. van der Vorst, *Iterative Krylov Methods for Large Linear Systems*, Cambridge University Press, 2003



# Algebralliset ominaisarvotehtävät

- värähtelytehtävät

$$K\phi = \omega^2 M\phi$$

$K$  jäykkyyismatriisi, SPD

$M$  massamatriisi, positiivisesti (semi)definiitti

- stabiiliusongelma, yksinkertaisimmillaan alkutilan suhteen linearisoitu

$$K_0\phi = -\lambda K_1\phi$$

$K_0$  alkutilan jäykkyyismatriisi, SPD (voi sisältää omapainon vaikutuksen)

$K_1$  alkujännitysmatriisi eli geometrinen jäykkyyismatriisi  
voi olla pos. def., neg. def. tai indefiniitti



# Käsitteitä

## Lineaariset ominaisarvotehtävät

- tavallinen (standardi) ominaisarvotehtävä  $A\phi = \lambda\phi$   
*Huom: symmetrisen matriisin ominaisarvot ovat reaalisia*
- yleistetty ominaisarvotehtävä  $Ax = \lambda Bx$   
missä  $A$  ja  $B$  ovat  $n \times n$  neliömatriiseja  
*Huom: symmetrisellä yleistetyllä ominaisarvotehtävällä voi olla kompleksisia ominaisarvoja*  
Riittävä (ei välttämätön) ehto reaalisille ominaisarvoille on

$$aA + bB = SPD, \quad \text{joillekin } a, b \in \mathbb{R}$$



# Käsitteitä

## Polynominen ominaisarvotehtävä

$$(\mathbf{K}_0 + \lambda \mathbf{K}_1 + \lambda^2 \mathbf{K}_2 + \cdots + \lambda^r \mathbf{K}_r) \phi = \mathbf{0}$$

Voidaan saattaa lineaariseksi yleistetyksi ominaisarvotehtäväksi:

$$\mathbf{x}_1 = \lambda \phi, \mathbf{x}_2 = \lambda \mathbf{x}_1, \dots$$

$$\left( \begin{bmatrix} \mathbf{K}_0 & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \cdots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{I} \end{bmatrix} + \lambda \begin{bmatrix} \mathbf{K}_1 & \mathbf{K}_2 & \cdots & \mathbf{K}_{r-1} & \mathbf{K}_r \\ -\mathbf{I} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & -\mathbf{I} & \mathbf{0} \end{bmatrix} \right) \begin{Bmatrix} \phi \\ \mathbf{x}_1 \\ \vdots \\ \mathbf{x}_{r-1} \end{Bmatrix} = \mathbf{0}.$$



# Ominaisuuksia

Jos  $A$ :n ominaisarvot ovat  $\lambda_i$  niin  $A^{-1}$ :n ominaisarvot ovat  $\lambda_i^{-1}$

Jos  $\phi$  on tehtävän  $A\phi = \lambda B\phi$  jokin ominaisvektori sitä vastaava ominaisarvo on

$$\lambda = \frac{\phi^T A \phi}{\phi^T B \phi} \quad (\text{Rayleighin osamäärä})$$





# Potenssi-iteraatio

## OLETUKSET

$A$  diagonalisoituva, eli  $\Phi^{-1}A\Phi = \text{diag}(\lambda_1, \dots, \lambda_n)$  missä  $\Phi = [\phi_1, \dots, \phi_n]$  ja ominaisarvoille pätee  $|\lambda_1| \leq |\lambda_2| \leq \dots \leq |\lambda_{n-1}| < |\lambda_n|$ .

## ALGORITMI

Alkuarvauksesta  $x_0$ ,  $\|x_0\|_2 = 1$   
iteroidaan  $i = 1, 2, \dots$

1. lasketaan  $z_i = Ax_{i-1}$  ja  $\tilde{\lambda}_i = x_{i-1}^T z_i = x_{i-1}^T Ax_{i-1}$

2. normeerataan  $x_i = z_i / \|z_i\|_2$

Jos  $x_0^T \phi_n \neq 0$  niin  $\tilde{\lambda}_i \rightarrow \lambda_n$  ja  $x_i \rightarrow \phi_n$  kun  $i \rightarrow \infty$



# Suppeneminen

Määräytyy ominaisarvojen  $\lambda_{n-1}$  ja  $\lambda_n$  suhteesta

$$|\lambda_n - \tilde{\lambda}_k| = \mathcal{O} \left( \left| \frac{\lambda_{n-1}}{\lambda_n} \right|^k \right).$$

Suppeneminen on hidasta mikäli  $\lambda_{n-1}/\lambda_n \approx 1$



Otaksutaan, että aloitusvektori on muotoa

$$\mathbf{x}_0 = \alpha_1 \boldsymbol{\phi}_1 + \alpha_2 \boldsymbol{\phi}_2 + \cdots + \alpha_n \boldsymbol{\phi}_n$$

oletetaan  $\alpha_n \neq 0$

$$\mathbf{A}\mathbf{x}_0 = \sum_{i=1}^n (\lambda_i \boldsymbol{\phi}_i \boldsymbol{\phi}_i^T) \sum_{j=1}^n (\alpha_j \boldsymbol{\phi}_j) = \sum_{i=1}^n \alpha_i \lambda_i \boldsymbol{\phi}_i = \alpha_n \lambda_n \left( \boldsymbol{\phi}_n + \sum_{i=1}^{n-1} \frac{\alpha_i \lambda_i}{\alpha_n \lambda_n} \boldsymbol{\phi}_i \right)$$

$k$ :n iteraation jälkeen

$$\mathbf{A}^k \mathbf{x}_0 = \alpha_n \lambda_n^k \left[ \boldsymbol{\phi}_n + \sum_{i=1}^{n-1} \frac{\alpha_i}{\alpha_n} \left( \frac{\lambda_i}{\lambda_n} \right)^k \boldsymbol{\phi}_i \right].$$



# Käänteinen potenssi-iteraatio

## ALGORITMI

Alkuarvauksesta  $\mathbf{x}_0$ ,  $\|\mathbf{x}_0\|_2 = 1$   
iteroidaan  $i = 1, 2, \dots$

1. lasketaan  $\mathbf{z}_i = \mathbf{A}^{-1}\mathbf{x}_{i-1}$  ja  $\tilde{\lambda}_i = \mathbf{x}_{i-1}^T \mathbf{z}_i$

2. normeerataan  $\mathbf{x}_i = \mathbf{z}_i / \|\mathbf{z}_i\|_2$

Jos  $\mathbf{x}_0^T \phi_1 \neq 0$  niin  $\tilde{\lambda}_i \rightarrow \lambda_1$  ja  $\mathbf{x}_i \rightarrow \phi_1$  kun  $i \rightarrow \infty$



# Käänteinen iteraatio yleistetyille systeemeille

$$A\phi = \lambda B\phi, \quad \implies \quad \lambda = \frac{\phi^T A\phi}{\phi^T B\phi} \quad \text{yleensä} \quad \phi^T B\phi = 1$$

Sovelletaan potenssi-iteraatiota systeemiin

$$A^{-1}B\phi = \mu\phi, \quad \mu = \lambda^{-1}$$

eli valitaan jokin alkuarvaus  $x_0$ , lasketaan  $y_0 = Bx_0$  ja iteroidaan  $i = 1, 2, \dots$

1. lasketaan  $z_i = A^{-1}y_{i-1} = A^{-1}Bx_{i-1}$
2. normeerataan  $x_i = \frac{z_i}{(z_i^T B z_i)^{1/2}}$



# Tehokkaampi algoritmi

Valitaan alkuarvaus  $x_0$ , lasketaan  $y_0 = Bx_0$  ja iteroidaan  $i = 1, 2, \dots$

1. ratkaistaan  $Az_i = y_{i-1}$

2. lasketaan  $\bar{y}_i = Bz_i$

3. lasketaan  $\rho_i = \frac{z_i^T y_{i-1}}{z_i^T \bar{y}_i}$

4. normeerataan  $y_i = \frac{\bar{y}_i}{(z_i^T \bar{y}_i)^{1/2}}$

Olettaen, että  $y_0^T \phi_1 \neq 0$  ja että  $\lambda_1$  on yksinkertainen

$$\rho_i \longrightarrow \lambda_1 \quad y_i \longrightarrow B\phi_1 \quad \text{kun} \quad i \longrightarrow \infty$$



# Rayleighin osamääräiteraatio

## ALGORITMI

Alkuarvaus  $x_0$ , lasketaan  $y_0 = Bx_0$ , valitaan alkusiirros (shift)  $\rho_0$  (yleensä nolla) iteroidaan  $i = 1, 2, \dots$

1. ratkaistaan  $(A + \rho_{i-1}B)z_i = y_{i-1}$

2. lasketaan  $\bar{y}_i = Bz_i$

3. lasketaan  $\rho_i = \frac{z_i^T y_{i-1}}{z_i^T \bar{y}_i} + \rho_{i-1}$

4. normeerataan  $y_i = \frac{\bar{y}_i}{(z_i^T \bar{y}_i)^{1/2}}$



# Yleistetty lineaarinen ominaisarvotehtävä

Ratkaistavana muutamia ominaisarvopareja ( $p$ -kappaletta) systeemistä (dimensio  $n \gg p$ )

$$A\Phi = B\Phi\Lambda,$$

“Ominaisarvomatriisi”  $\Lambda = \text{diag}(\lambda_i)$  Ominaisvektorimatriisi:  $\Phi = [\phi_1, \dots, \phi_p]$

Suurten symmetristen ominaisarvotehtävien ratkaisualgoritmeja:

- aliavaruusiteraatio (subspace, simultaneous iteration)
- Lanczosin menetelmä





# Aliavaruusiteraatio

$p$ :n alimman ominaisarvoparin  $(\lambda_i, \phi_i)$  löytämiseksi

Valitse aloitusvektorit  $\Phi_0$  ja iteroi  $k = 1, 2, \dots$ ,

1. ratkaise  $A\bar{\Phi}_k = B\Phi_{k-1}$       $\Phi = [\phi_1, \dots, \phi_q]$       $q > p$
2. projisoi  $A, B$  aliavaruuteen      $A_k^* = \bar{\Phi}_k^T A \bar{\Phi}_k$       $B_k^* = \bar{\Phi}_k^T B \bar{\Phi}_k$
3. ratkaise projisoitu tehtävä      $A_k^* Q_k = B_k^* Q_k \Lambda_k$
4. päivitä ominaisvektorit      $\Phi_k = \bar{\Phi}_k Q_k$

Jos vektorit  $\Phi_0$  eivät ole ortogonaalisia haluttujen ominaisvektoreiden kanssa, niin iteraatio konvergoi  $\Lambda_k \longrightarrow \Lambda$  ja  $\Phi_k \longrightarrow \Phi$  kun  $k \longrightarrow \infty$ .



# Projisointi ominaisarvot tehtävä

$B_k^*$  ei välttämättä pos. def.

$$B_k^* Q_k = A_k^* Q_k \Lambda_k^{-1}.$$

Yleensä projisointi matriisi  $A_k$  on pos. def.

Choleskyn decompositio  $A_k = LL^T$  on mahdollinen.

Projisointi yleistetty ongelma voidaan kirjoittaa standardimuodossa

$$CX = X\Lambda^{-1},$$

missä  $C = L^{-1} B_k^* (L^{-1})^T$  ja  $X = L^T Q_k$ .



# Käytännön seikkoja

Suppenemista mitataan yleensä

$$\frac{|\lambda_i^{(k)} - \lambda_i^{(k-1)}|}{|\lambda_i^{(k)}|} \leq \text{tol}, \quad i = 1, \dots, p$$

Suosittelava määrä iteraatiovektoreita  $q = \min(2p, p + 8)$  (K.-J. Bathe)  
riittämätön mikäli ominaisarvot ovat ryppäessä,  $q = p + \text{int}(p^{0.8}) + 3$  (K. Bell)

Ominaisvektorit suppenevat lineaarisesti nopeudella  $|\lambda_i|/|\lambda_{q+1}|$   
Suppenemista voi kiihdyttää käyttämällä spektrin siirrosta  $\sigma$ , eli ratkaistaan  $\mu, \phi$

$$(A - \sigma B)\phi = \mu B\phi \quad \lambda = \mu + \sigma$$

Monissa ohjelmissa vaaditaan  $\sigma < \min \lambda_i$ , jos  $\lambda_i > 0$  tai  $\sigma > \max \lambda_i$  jos  $\lambda_i < 0$



# Lanczosin menetelmä

Idea perustuu Schurin hajotelmaan:

$$Q^T A Q = T = \begin{bmatrix} \alpha_1 & \beta_1 & & & \\ \beta_1 & \alpha_2 & \beta_2 & & \\ & \beta_2 & \cdot & \cdot & \\ & & \cdot & \cdot & \\ & & & & \alpha_n \end{bmatrix}$$

$Q = [\mathbf{q}_1, \dots, \mathbf{q}_n]$  ortogonaalinen, siten  $AQ = QT$  eli

$$A\mathbf{q}_i = \beta_{i-1}\mathbf{q}_{i-1} + \alpha_i\mathbf{q}_i + \beta_i\mathbf{q}_{i+1}$$



# Lanczosin algoritmi

Koostuu kahdesta osasta:

1. Lanczosin iteraatiosta tridiagonaalimatriisiin  $T_m$  muodostamiseksi
2. tridiagonaalimatriisin ominaisarvojen laskennasta

$m:n$  Lanczos askeleen jälkeen  $T_m$  on  $m \times m$ -matriisi, jonka ominaisarvot ovat hyviä approksimaatioita  $A:n$  ( $n \times n$ -matriisi, ja  $m \ll n$ ) uloimmille ominaisarvoille.



# Lanczosin iteraatio

Ominaisarvotehtävälle  $A\phi = \lambda\phi$

## Alustus

Aseta:  $i = 0$ ,  $\mathbf{q}_0 = \mathbf{0}$ ,  $\beta_0 = 1$  valitse  $\mathbf{q}_1$  s.e.  $\|\mathbf{q}_1\| = 1$

**Iteroi jos**  $\beta_i \neq 0$ ,  $i = 1, 2, \dots$

1.  $\alpha_i = \mathbf{q}_i^T \mathbf{A} \mathbf{q}_i$
2.  $\mathbf{r}_i = (\mathbf{A} - \alpha_i \mathbf{I}) \mathbf{q}_i - \beta_{i-1} \mathbf{q}_{i-1}$
3.  $\beta_i = \|\mathbf{r}_i\|$
4.  $\mathbf{q}_{i+1} = \mathbf{r}_i / \beta_i$

Generoi Krylovin aliavaruuden  $\mathcal{K}_i(\mathbf{A}, \mathbf{q}_1) = \text{span} \{ \mathbf{q}_1, \mathbf{A} \mathbf{q}_1, \mathbf{A}^2 \mathbf{q}_1, \dots, \mathbf{A}^{i-1} \mathbf{q}_1 \}$

**HUOM:** Lanczosin vektorit  $\mathbf{q}_i$  eivät ole ominaisvektoreita  $\phi_i$ !!



# Ongelmia Lanczosin iteraatiossa

Moninkertaiset ominaisarvot  $\Rightarrow \beta_i = 0$

- pyöristysvirheet pelastajana
- lohko-Lanczos (lohkon koko  $\geq$  ominaisarvon kertaluku)

Ortogonaalisuuden häviäminen Lanczosin vektoreilta

- osittainen ja/tai valikoiva uudelleenortogonalisointi



# Lanczos yleistetylle tehtävälle

## Spektrin muunnos värähtelytehtävässä

$$K\phi = \lambda M\phi$$

nyt  $M$  on SPD tai symmetrinen pos. semidefiniitti

Muunnetaan muotoon

$$M(K - \sigma M)^{-1}M\phi = \theta M\phi \quad \implies \quad A\phi = \theta B\phi$$

missä  $\theta = 1/(\lambda - \sigma)$  ja  $\sigma$  on siirros (shift)

Alkuperäinen ominaisarvo on siten

$$\lambda = \sigma + \frac{1}{\theta}$$





## Spektrin muunnos lineaarisessa stabiiliustehtävässä

$$\mathbf{K}_0\phi = -\lambda\mathbf{K}_1\phi$$

geometrinen jäykkymatriisi  $\mathbf{K}_1$  ei välttämättä ole pos.def.

Muunnetaan muotoon ( $\mathbf{K}$  on SPD)

$$\mathbf{K}_0(\mathbf{K}_0 + \sigma\mathbf{K}_1)^{-1}\mathbf{K}_0\phi = \theta\mathbf{K}_0\phi \quad \implies \quad \mathbf{A}\phi = \theta\mathbf{B}\phi$$

missä  $\theta = \lambda/(\lambda - \sigma)$  ja  $\sigma \neq 0$  on siirros (shift)

Alkuperäinen ominaisarvo on siten

$$\lambda = \frac{\sigma\theta}{\theta - 1}$$



## Esimerkkejä – sylinterikuori

Lineaarinen stabiiliusanalyysi  $20 \times 120$  stab. MITC4 elementtejä  
 $n = 13681$ , halutaan 5 alinta ominaisarvoa tai arvot väliltä  $1, \dots, 1,3$   
 alin ominaisarvo 1.028 (1 = teorettinen arvo)

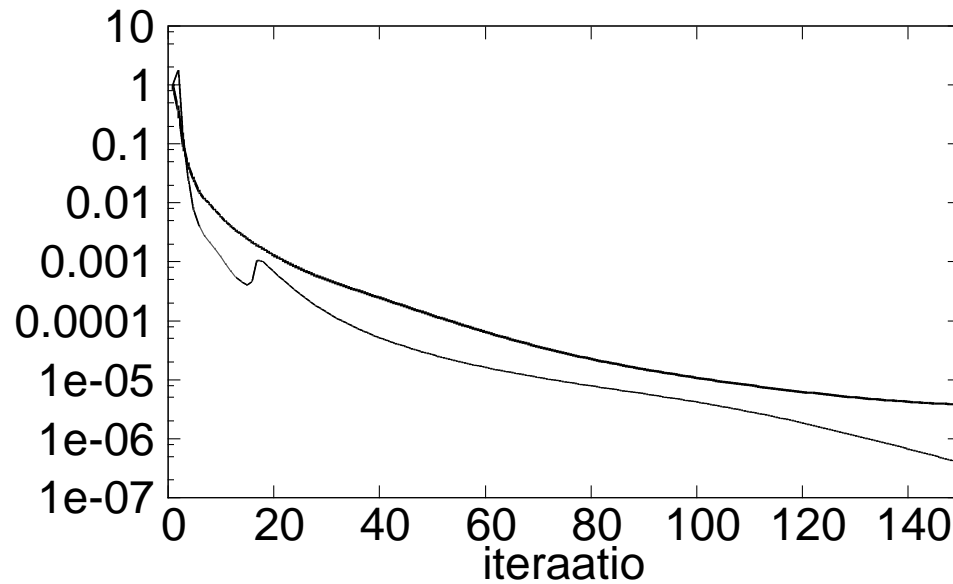
ratkaisija	$q$	$\sigma$	it	CPU	huom.
aliavaruus	10	1.0	> 150	487	1. om.arvo suppeni 131 it
lohko Lanczos	2	1.0	6	72	8 om.arvo suppeni

$q$  on aliavaruuden koko tai lohkon koko Lanczosin menetelmässä  
 it on aliavaruusiteraatioiden tai Lanczosin askeleiden lkm.



# Sylinterikuori

Alimman ominisarvon suppeneminen aliavaruusiteraatioissa



## Esimerkkejä – T-palkki

Lineaarinen stabiiliusanalyysi 880 stab. MITC4 elementtiä ( $P_{kr} = 1212N$ )  
 $n = 5563$ , halutaan 7 alinta ominaisarvoa tai arvot väliltä  $\sigma, \dots, 1500 N$

ratkaisija	$q$	$\sigma$	it	$m$	CPU	huom.
aliavaruus	14	1200	8	-	4.9	
	7	1200	39	-	8.9	
	14	500	19	-	9.1	
lohko Lanczos	1	1200	28	28	2.4	9 om.arvo suppeni
	2	1200	18	36	2.6	9 om.arvo suppeni
	4	1200	14	56	3.8	9 om.arvo suppeni
	1	500	14	14	4.3	8 om.arvo suppeni
	2	500	13	26	5.9	8 om.arvo suppeni
	4	500	22	88	4.3	9 om.arvo suppeni

$q$  on aliavaruuden koko tai lohkon koko Lanczosin menetelmässä (input)

$m$  on generoituvan aliavaruuden koko Lanczosin menetelmässä

it on aliavaruusiteraatioiden tai Lanczosin askeleiden lkm.



# Alivaruusiteraatio vs. Lanczos

- Lanczosin menetelmä yleensä nopeampi
- Joissain ohjelmissa rajoituksia Lanczosin menetelmän käytölle

## HUOMIOITAVAA

Suorita aina laskettujen ominaisarvojen tarkistus spektrin siirroksella (nimitetään myös Sturmin jonoon perustuvaksi tarkistukseksi)



# Kirjallisuutta

- K.-J. Bathe, E.L. Wilson, *Numerical Methods in Finite Element Analysis*, Prentice-Hall, 1976, luvut 10-12.
- T.J.R. Hughes, *The Finite Element Method*, Prentice-Hall, 1987, luku 10.6 “The Lanczos Algorithm for Solution of Large Generalized Eigenproblems” kirjoittanut B. Nour-Omid.
- B.N. Parlett, *The Symmetric Eigenvalue Problem*, Prentice-Hall, 1980
- Y. Saad, *Numerical Methods for Large Eigenvalue Problems*, Manchester University Press, 1992



# Suora aikaintegrointi

## KÄSITTEITÄ

Alkuarvot tehtävän

$$\dot{y} + f(y) = 0,$$

ratkaisumenetelmää kutsutaan *eksplisiittiseksi* eli *avoimeksi*, mikäli

$$y_{n+1} = y_n - \Delta t f(y_n).$$

Menetelmä *suljettu* eli *implisiittinen* jos

$$y_{n+1} = y_n - \Delta t f(y_{n+1}).$$



# FEM approksimaatio

Liikkeyhtälön FE-diskretointi paikkakoordinaattien suhteen tuottaa semi-diskreetin liikkeyhtälön.

Linearisessa tapauksessa se voidaan kirjoittaa muodossa

$$M\ddot{q} + C\dot{q} + Kq = f$$

Tutkitaan y.o. liikkeyhtälön aikaintegrointimenetelmiä kuitenkin yksivapausasteisen systeemin avulla.

Mikäli otaksutaan Rayleighin vaimennus

$$C = aM + bK$$

voidaan liikkeyhtälö diagonalisoida ominaismuotojen avulla.





# Diagonalisoinnin ja aikadiskretoinnin kommutatiivisuus

$$\begin{array}{ccc}
 M\ddot{q} + C\dot{q} + Kq = f & \xrightarrow{\text{aikadiskretointi}} & M\ddot{q}_{n+1} + C\dot{q}_{n+1} + Kq_{n+1} = f_{n+1} \\
 \downarrow \text{ominaismuotohajotelma} & & \downarrow \text{ominaismuotohajotelma} \\
 \ddot{y} + 2\zeta\omega\dot{y} + \omega^2y = f & \xrightarrow{\text{aikadiskretointi}} & \ddot{y}_{n+1} + 2\zeta\omega\dot{y}_{n+1} + \omega^2y_{n+1} = f_{n+1}
 \end{array}$$



# Värähtelevä lanka

DY

$$\rho A \frac{\partial^2 u}{\partial t^2} - \sigma A \frac{\partial^2 u}{\partial x^2} = 0,$$

$$u(x, 0) = \bar{u}_0 \sin(\pi x/L),$$

$$\frac{\partial u(x, 0)}{\partial t} = v(x, 0) = \bar{v}_0 \sin(\pi x/L)$$

Yrite

$$u(x, t) = Y(t) \sin \frac{\pi x}{L},$$



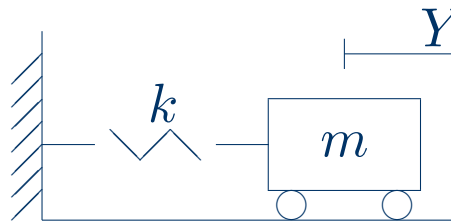
# Yksivapausasteinen liikeyhtälö

$$\begin{cases} \ddot{Y} + \omega^2 Y = 0, \\ Y(0) = \bar{u}_0, \\ \dot{Y}(0) = \bar{v}_0, \end{cases}$$

missä  $\omega^2 = \pi^2 \sigma / (\rho L^2)$

Ratkaisu

$$Y(t) = \frac{\bar{v}_0 L}{c\pi} \sin \omega t + \bar{u}_0 \cos \omega t$$



# Keskeisdifferenssimenetelmä

## Nopeuksien keskeisdifferenssiapproksimaatio

$$\dot{Y}_{n-\frac{1}{2}} \approx \dot{y}_{n-\frac{1}{2}} = \frac{y_n - y_{n-1}}{\Delta t_n},$$

$$\dot{Y}_{n+\frac{1}{2}} \approx \dot{y}_{n+\frac{1}{2}} = \frac{y_{n+1} - y_n}{\Delta t_{n+1}}.$$

## Kiihtyvyyksien keskeisdifferenssiapproksimaatiot

$$\ddot{Y}_n \approx \ddot{y}_n = \frac{\dot{y}_{n+\frac{1}{2}} - \dot{y}_{n-\frac{1}{2}}}{t_{n+\frac{1}{2}} - t_{n-\frac{1}{2}}} = \frac{\dot{y}_{n+\frac{1}{2}} - \dot{y}_{n-\frac{1}{2}}}{\Delta t_{n+\frac{1}{2}}}.$$

## Yhdistämällä

$$\ddot{y}_n = \frac{\Delta t_n (y_{n+1} - y_n) - \Delta t_{n+1} (y_n - y_{n-1})}{\Delta t_n \Delta t_{n+\frac{1}{2}} \Delta t_{n+1}}.$$



# Keskeisdifferenssimenetelmä

Jos aika-askel vakio

$$\ddot{y}_n = \frac{y_{n+1} - 2y_n + y_{n-1}}{\Delta t^2}$$

Sovelletaan yhtälöön  $\ddot{Y} + \omega^2 Y = 0$

$$y_{n+1} = (2 - \Delta t^2 \omega^2) y_n - y_{n-1}$$

Entä stabiilius ??

Sijoitetaan yrite  $y_n = C\lambda^n$ , saadaan karakteristinen yhtälö

$$\lambda^2 - (2 - \Delta t^2 \omega^2) \lambda + 1 = 0,$$

Juuret ovat

$$\lambda_{1,2} = 1 - \frac{1}{2} \Delta t^2 \omega^2 \pm \sqrt{\left(1 - \frac{1}{2} \Delta t^2 \omega^2\right)^2 - 1}.$$



# CD-stabiilius

Juurten oltava kompleksisia

$$\left(1 - \frac{1}{2}\Delta t^2\omega^2\right)^2 < 1,$$

Rajoite aika-askeleen pituudelle

$$\Delta t < \Delta t_{\text{kr}} = \frac{2}{\omega}.$$

$$\lambda_{1,2} = a \pm \left(\sqrt{1 - a^2}\right) i,$$

missä  $i^2 = -1$  ja  $a = 1 - \frac{1}{2}\Delta t^2\omega^2$ .

$|\lambda_{1,2}| = 1$  *ei amplitudivirhettä*



# Newmarkin menetelmä

$$y_{n+1} = y_n + \Delta t \dot{y}_n + \frac{1}{2} \Delta t^2 [(1 - 2\beta)\ddot{y}_n + 2\beta\ddot{y}_{n+1}],$$

$$\dot{y}_{n+1} = \dot{y}_n + \Delta t [(1 - \gamma)\ddot{y}_n + \gamma\ddot{y}_{n+1}],$$

parametrit  $\beta$  ja  $\gamma$ .

Sovelletaan liikeyhtälöön  $\ddot{Y} + \omega^2 Y = 0$

$$\begin{bmatrix} 1 + \beta\Omega^2 & 0 \\ \gamma\Omega^2 & 1 \end{bmatrix} \begin{Bmatrix} y_{n+1} \\ \Delta t \dot{y}_{n+1} \end{Bmatrix} = \begin{bmatrix} 1 - \frac{1}{2}(1 - 2\beta)\Omega^2 & 1 \\ (\gamma - 1)\Omega^2 & 1 \end{bmatrix} \begin{Bmatrix} y_n \\ \Delta t \dot{y}_n \end{Bmatrix},$$

missä  $\Omega^2 = \Delta t^2 \omega^2$ .



Kirjoitetaan muodossa

$$\mathbf{x}_{n+1} = \mathbf{A}\mathbf{x}_n,$$

missä  $\mathbf{x}_n = [y_n, \Delta t \dot{y}_n]^T$  ja  $\mathbf{A}$  stabiiliusmatriisi (vahvennusmatriisi)

$$\mathbf{A} = \frac{1}{1 + \beta\Omega^2} \begin{bmatrix} 1 - \frac{1}{2}(1 - 2\beta)\Omega^2 & 1 \\ -\Omega^2 - (\beta - \frac{1}{2}\gamma)\Omega^4 & 1 + (\beta - \gamma)\Omega^2 \end{bmatrix}.$$

$$\lambda^2 - 2A_1\lambda + A_2 = 0,$$

$$A_1 = \frac{1}{2}\text{trace}\mathbf{A} = \frac{1}{2}(A_{11} + A_{22}) = 1 - \frac{1}{2} \frac{(\frac{1}{2} + \gamma)\Omega^2}{1 + \beta\Omega^2},$$

$$A_2 = \det \mathbf{A} = A_{11}A_{22} - A_{12}A_{21} = 1 + \left(\frac{1}{2} - \gamma\right) \frac{\Omega^2}{1 + \beta\Omega^2}.$$

Juuret ovat  $\lambda_{1,2} = A_1 \pm \sqrt{A_1^2 - A_2}$ .





Jotta numeerinen ratkaisu esittäisi oskilloivaa vastetta

$$A_1^2 - A_2 < 0$$
$$\implies -4\Omega^2 + \left[ \left( \frac{1}{2} + \gamma \right)^2 - 4\beta \right] \Omega^4 < 0.$$

Mikäli  $\Omega^4$ :n kerroin on negatiivinen, ovat juuret aina kompleksisia kaikilla mahdollisilla  $\Omega^2$ :n arvoilla ( $\Omega^2 = \omega^2 \Delta t^2 > 0$ ).

$$\beta > \frac{1}{4} \left( \gamma + \frac{1}{2} \right)^2.$$

Stabiiliusmatriisin ominaisarvot

$$\lambda_{1,2} = A_1 \pm \left( \sqrt{A_2 - A_1^2} \right) i.$$



# Trapetsikaava

Juurten itseisarvo

$$|\lambda|^2 = A_2 = 1 + \left(\frac{1}{2} - \gamma\right) \frac{\Omega^2}{1 + \beta\Omega^2},$$

Mikäli  $\gamma = \frac{1}{2}$  on  $|\lambda| = 1 \quad \forall \quad \Delta t$

Kompleksisuusehto on nyt  $-4\Omega^2 + (1 - 4\beta)\Omega^4 < 0$ ,

Toteutuu jos  $\beta = \frac{1}{4}$

*Trapetsikaava tai keskimääräisen kiihtyvyyden menetelmä*

$$y_{n+1} = y_n + \Delta t \dot{y}_n + \frac{1}{4} \Delta t^2 (\ddot{y}_n + \ddot{y}_{n+1}),$$

$$\dot{y}_{n+1} = \dot{y}_n + \frac{1}{2} \Delta t (\ddot{y}_n + \ddot{y}_{n+1}).$$

Ei amplitudivirhettä.



# Stabiilius

Newmarkin menetelmä on **ehdoitta stabiili** ja sen **vahvennusmatriisin ominaisarvot ovat kompleksisia**, mikäli seuraavat ehdot toteutuvat

$$\gamma \geq \frac{1}{2} \quad \text{ja} \quad \beta > \frac{1}{4} \left( \gamma + \frac{1}{2} \right)^2.$$

Stabiiliusalueetta voidaan hieman laajentaa, mikäli juurien sallitaan olevan reaaliset. Tällöin  $A_1^2 - A_2 > 0$  ja kriittinen ehto seuraa epäyhtälöistä

$$\begin{cases} -1 \leq A_1 \leq 0 \\ \max |\lambda| = A_1 - \sqrt{A_1^2 - A_2} \end{cases} \quad \text{tai} \quad \begin{cases} 0 < A_1 \leq 1 \\ \max |\lambda| = A_1 + \sqrt{A_1^2 - A_2} \end{cases}$$

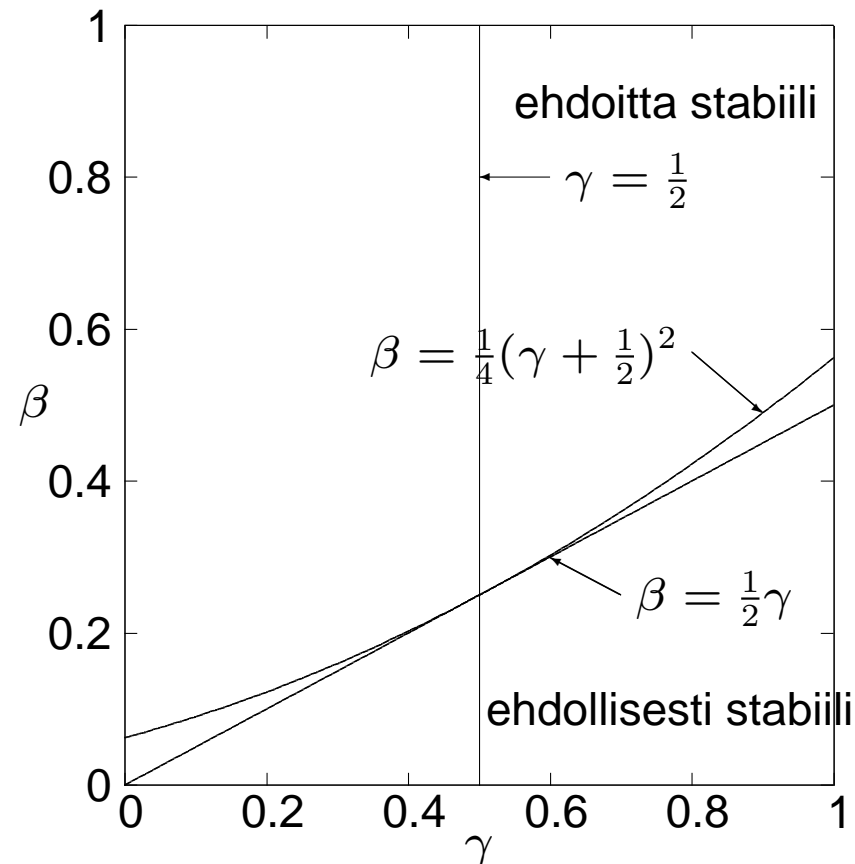
$$2A_1 + A_2 + 1 \geq 0 \quad \implies \quad 4 + 2(2\beta - \gamma)\Omega^2 \geq 0.$$

Newmarkin menetelmä on **ehdoitta stabiili** kun

$$2\beta \geq \gamma \geq \frac{1}{2}$$



# Newmarkin menetelmän stabiiliusalue



# Ehdollinen stabiilius

Mikäli  $\beta < \gamma/2$  menetelmä *ehdollisesti stabiili*

*Kriittinen aika-askel* on

$$\omega \Delta t \leq \frac{1}{\sqrt{\frac{1}{2}\gamma - \beta}} = \Omega_{\text{kr}}.$$

havainnollisempi esitys värähtelyn *jaksonajan*  $T = 2\pi/\omega$  avulla:

$$\frac{\Delta t}{T} \leq \frac{\Omega_{\text{kr}}}{2\pi}.$$

Valinnalla  $\beta = 0, \gamma = \frac{1}{2}$  saadaan keskeisdifferenssimenetelmä

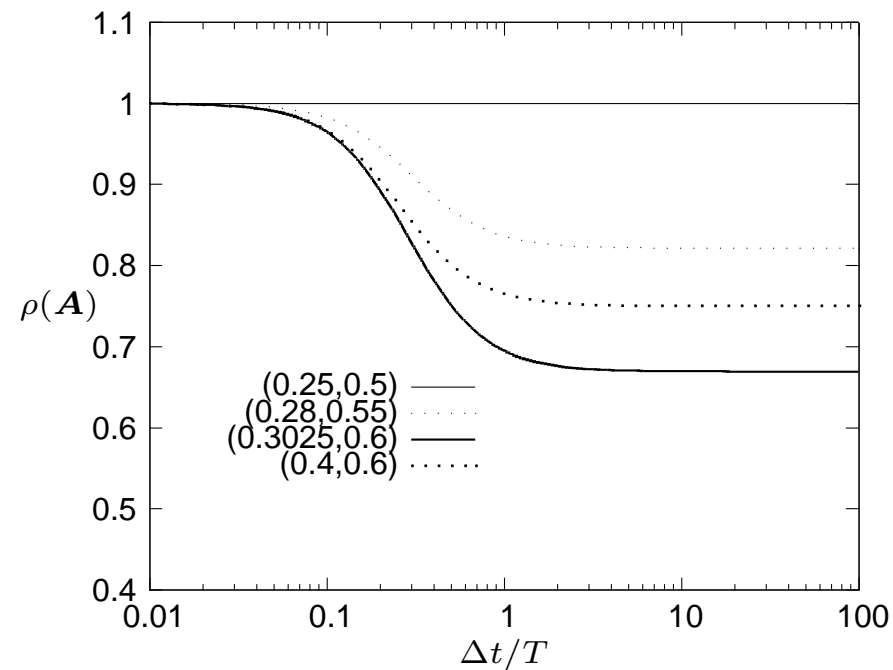
$$y_{n+1} = y_n + \Delta t \dot{y}_n + \frac{1}{2} \Delta t^2 \ddot{y}_n,$$

$$\dot{y}_{n+1} = \dot{y}_n + \frac{1}{2} \Delta t (\ddot{y}_n + \ddot{y}_{n+1}).$$



# Korkeataajuuskäyttäytyminen

Newmarkin menetelmän stabiiliusmatriisin spektraalisäde muutamilla  $\beta, \gamma$  pareilla aika-askeleen koon funktiona



# Tunnettuja menetelmiä

Newmarkin menetelmäperheen edustajia ovat myös

- lineaarisen kiihtyvyyden menetelmä,  $\beta = \frac{1}{6}$ ,  $\gamma = \frac{1}{2}$
- Foxin ja Goodwinin menetelmä,  $\beta = \frac{1}{12}$ ,  $\gamma = \frac{1}{2}$

Nämä ovat *implisiittisiä ehdollisesti stabiileja* menetelmiä.

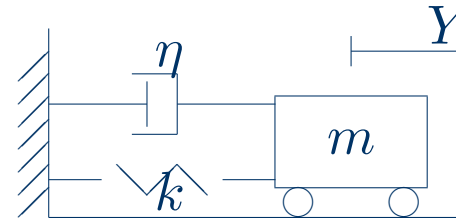
Eivät siten ole taloudellisia suurten systeemien analysointiin.



# Vaimennettu värähtelijä

Kuvan värähtelijän liikeyhtälö on

$$m\ddot{Y} + c\dot{Y} + kY = \bar{f}.$$



Liikeyhtälö voidaan kirjoittaa myös muodossa

$$\ddot{Y} + 2\zeta\omega\dot{Y} + \omega^2Y = \frac{\bar{f}}{m}\bar{F},$$

missä  $\zeta = c/(2m\omega)$  on suhteellinen vaimennuskerroin.

Homogeenisen osan ratkaisu on muotoa  $Y_h = C \exp(rt)$ , ja  $r$  voidaan ratkaista karakteristisen polynomin  $r^2 + 2\zeta\omega r + \omega^2 = 0$  nollakohdista:

$$r_{1,2} = (-\zeta \pm \sqrt{\zeta^2 - 1})\omega.$$

Homogeenisen osan ratkaisun luonne riippuu nyt suhteellisen vaimennuksen  $\zeta$  arvosta





## Kolme mahdollisuutta:

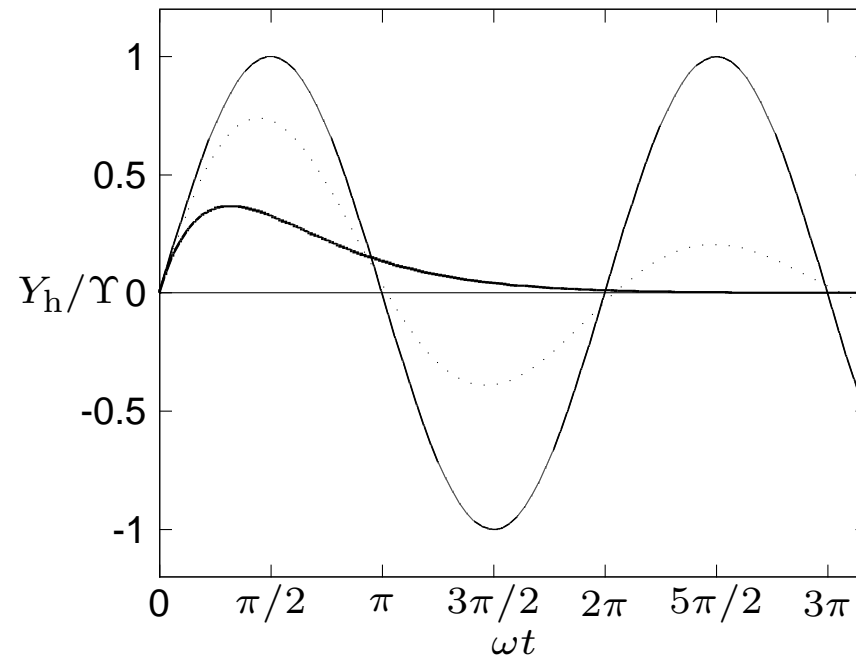
1. Ylikriittinen vaimennus kun  $\zeta > 1$ . Liike vaimenee monotonisesti ja koska ratkaisu ei esitä värähtelyä kutsutaan sitä aperiodiseksi liikkeeksi.
2. Kun  $\zeta = c/(2m\omega) = 1$  on karakteristisella yhtälöllä reaalin kaksoisjuuri.
3. Tärkein tapaus on alikriittinen vaimennus,  $\zeta < 1$ , juuret kompleksisia konjugaattilukuja. Ratkaisu esittää taajuudella  $\omega_d$  tapahtuvaa harmonista värähtelyä jonka amplitudi pienenee eksponentiaalisesti ajan mukana

$$Y_h(t) = e^{-\zeta\omega t}(C_1 \sin \omega_d t + C_2 \cos \omega_d t) = \Upsilon e^{-\zeta\omega t} \sin(\omega_d t + \phi),$$

missä  $\Upsilon = \sqrt{C_1^2 + C_2^2}$ ,  $\omega_d = \sqrt{1 - \zeta^2}\omega$  ja  $\phi = \arctan(C_2/C_1)$ .



## Vaimentumaton, kriittinen ja alikriittinen ( $\zeta = 0.2$ ) värähtely



# Newmark vaimennetulle systeemille

Sovelletaan Newmarkin menetelmää

$$y_{n+1} = y_n + \Delta t \dot{y}_n + \frac{1}{2} \Delta t^2 [(1 - 2\beta)\ddot{y}_n + 2\beta\ddot{y}_{n+1}],$$

$$\dot{y}_{n+1} = \dot{y}_n + \Delta t [(1 - \gamma)\ddot{y}_n + \gamma\ddot{y}_{n+1}],$$

vaimennetun värähtelijän yhtälöön askeleilla  $n$  ja  $n + 1$

$$\ddot{y}_n + 2\zeta\omega\dot{y}_n + \omega^2 y_n = F_n = f_n/m,$$

$$\ddot{y}_{n+1} + 2\zeta\omega\dot{y}_{n+1} + \omega^2 y_{n+1} = F_{n+1} = f_{n+1}/m,$$

saadaan

$$\begin{aligned} & \begin{bmatrix} 1 + \beta\Omega & 2\beta\zeta\Omega \\ \gamma\Omega^2 & 1 + 2\gamma\zeta\Omega \end{bmatrix} \begin{Bmatrix} y_{n+1} \\ \Delta t\dot{y}_{n+1} \end{Bmatrix} \\ &= \begin{bmatrix} 1 - \frac{1}{2}(1 - 2\beta)\Omega^2 & 1 - (1 - 2\beta)\zeta\Omega \\ (\gamma - 1)\Omega^2 & 1 + 2(\gamma - 1)\zeta\Omega \end{bmatrix} \begin{Bmatrix} y_n \\ \Delta t\dot{y}_n \end{Bmatrix} + \Delta t^2 \begin{Bmatrix} \frac{1}{2}(1 - 2\beta)\bar{F}_n + \beta\bar{F}_{n+1} \\ (1 - \gamma)\bar{F}_n + \gamma\bar{F}_{n+1} \end{Bmatrix}. \end{aligned}$$



## Kompakti muoto

$$\dot{\mathbf{x}}_{n+1} = \mathbf{A}\mathbf{x}_n + \mathbf{L}_n$$

missä  $\mathbf{x} = [y, \Delta t \dot{y}]^T$ ,  $\mathbf{A} = \mathbf{A}_1^{-1} \mathbf{A}_2$  ja

$$\mathbf{A}_1 = \begin{bmatrix} 1 + \beta\Omega^2 & 2\beta\zeta\Omega \\ \gamma\Omega^2 & 1 + 2\gamma\zeta\Omega \end{bmatrix}, \quad \mathbf{A}_2 = \begin{bmatrix} 1 - \frac{1}{2}(1 - 2\beta)\Omega^2 & 1 - (1 - 2\beta)\zeta\Omega \\ (\gamma - 1)\Omega^2 & 1 + 2(\gamma - 1)\zeta\Omega \end{bmatrix},$$

$$\mathbf{L}_n = \mathbf{A}_1^{-1} \Delta t^2 \left\{ \begin{array}{l} \frac{1}{2}(1 - 2\beta)F_n + \beta F_{n+1} \\ (1 - \gamma)F_n + \gamma F_{n+1} \end{array} \right\}.$$

Stabiiliusanalyysi periaatteessa samanlainen kuin vaimentamattomalle systeemille.

Ehdot juurien kompleksisuudelle ja ehdoitta stabiiliudelle parametrien  $\beta, \gamma$  suhteen

$$\gamma \geq \frac{1}{2}, \quad \beta \geq \frac{1}{4}\left(\gamma + \frac{1}{2}\right)^2$$

$$2\beta \geq \gamma \geq \frac{1}{2}$$

ovat samat kuin vaimentamattomalle tapaukselle.



# Ehdollinen stabiilius

Mikäli  $\beta < \gamma/2$  Newmarkin menetelmä on *ehdollisesti stabiili*

*Kriittinen aika-askel* on

$$\omega \Delta t \leq \Omega_{\text{kr,d}} = \frac{\zeta(\gamma - \frac{1}{2}) + \sqrt{\frac{1}{2}\gamma - \beta + \zeta^2(\gamma - \frac{1}{2})^2}}{\frac{1}{2}\gamma - \beta}.$$

**HUOM:** Vaimentamattoman systeemin kriittinen aika-askel on konservatiivinen arvio vaimennetun systeemin kriittiselle aika-askeleelle

Keskeisdifferenssimenetelmälle  $\beta = 0, \gamma = \frac{1}{2}$

$$\omega \Delta t = \Omega < \Omega_{\text{kr}} = 2\sqrt{1 - \zeta^2}.$$



# Tarkkuudesta

Aikaintegrointialgoritmi voi aiheuttaa

- **numeerista dissipaatiota**
  - mitataan algoritmisella vaimennussuhteella  $\zeta_{\text{alg}}$

$$\zeta_{\text{alg}} = \zeta + \frac{1}{2}(\gamma - \frac{1}{2})[\Omega + \mathcal{O}(\Omega^2)]$$

- **numeerista dispersiota**
  - mitataan suhteellisella jaksonajan virheellä (vaihevirhe)

$$\frac{T_{\text{alg}} - T}{T} = \mathcal{O}(\Omega^2) \quad T_{\text{alg}} = 2\pi/\omega_{\text{alg}}, \quad T = 2\pi/\omega$$

ja missä  $\zeta_{\text{alg}}$  ja  $\omega_{\text{alg}}$  ovat  $\zeta$ :n ja  $\omega$ :n algoritmisiä vastineita



Mikäli algoritmin vaimennussuhde  $\zeta_{\text{alg}}$  on pieni ja ajanhetkeksi valitaan algoritmin jaksonaika  $t_n = T_{\text{alg}}$

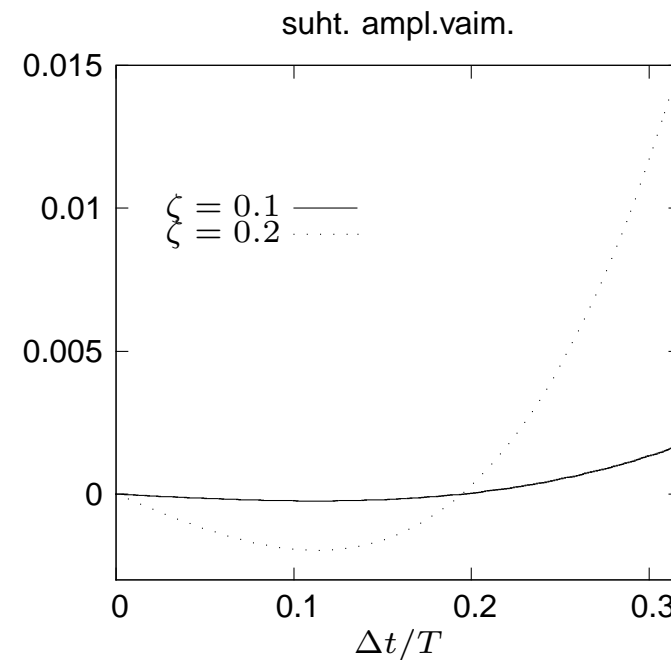
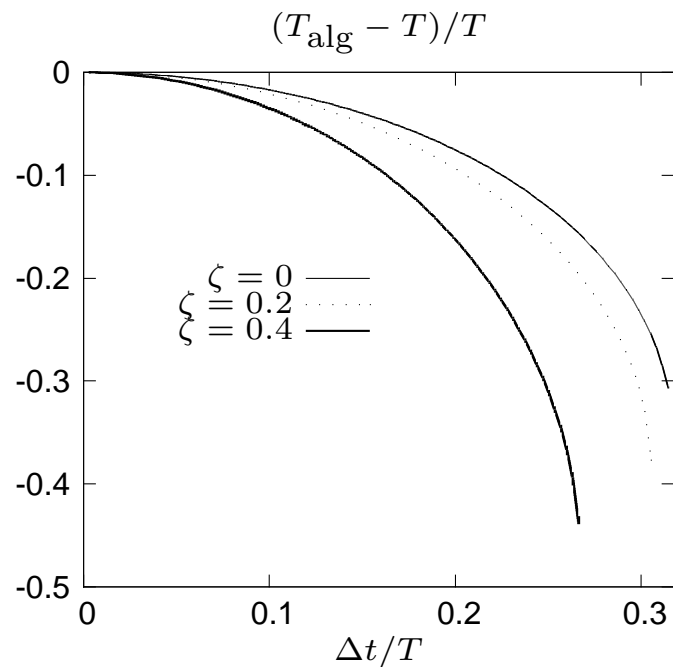
$$\exp(-\zeta_{\text{alg}}\omega_{\text{alg}}t_n) = \exp(-2\pi\zeta_{\text{alg}}) \approx 1 - 2\pi\zeta_{\text{alg}}$$

Yhdellä periodilla tapahtuu siten **numeerinen amplitudin vaimeneminen**

$$AV = 2\pi\zeta_{\text{alg}}$$



Keskeisdifferenssimenetelmän vaihevirheen ja yhden aika-askeleen suhteellisen amplitudivirheen (vaimennus positiivisena) riippuvuus aika-askeleen koosta.



**HUOM:** Keskeisdifferenssimenetelmällä ei ole amplitudivirhettä vaimentumattomalle systeemille.





## Yleistetty $\alpha$ -menetelmä

Yleistetyn  $\alpha$ -menetelmän siirtymän ja nopeuden kaavat ovat samat kuin Newmarkin menetelmässä, eli

$$y_{n+1} = y_n + \Delta t \dot{y}_n + \left(\frac{1}{2} - \beta\right) (\Delta t)^2 \ddot{y}_n + \beta (\Delta t)^2 \ddot{y}_{n+1},$$

$$\dot{y}_{n+1} = \dot{y}_n + (1 - \gamma) \Delta t \ddot{y}_n + \gamma \Delta t \ddot{y}_{n+1},$$

mutta nyt liikeyhtälö kirjoitetaan muodossa

$$m \ddot{y}_{n+1-\alpha_m} + c \dot{y}_{n+1-\alpha_f} + k y_{n+1-\alpha_f} = f(t_{n+1-\alpha_f}),$$

missä  $\alpha_m, \alpha_f$  ovat menetelmän parametrejä ja

$$y_{n+1-\alpha_f} = (1 - \alpha_f) y_{n+1} + \alpha_f y_n, \quad \dot{y}_{n+1-\alpha_f} = (1 - \alpha_f) \dot{y}_{n+1} + \alpha_f \dot{y}_n,$$

$$\ddot{y}_{n+1-\alpha_m} = (1 - \alpha_m) \ddot{y}_{n+1} + \alpha_m \ddot{y}_n, \quad t_{n+1-\alpha_f} = (1 - \alpha_f) t_{n+1} + \alpha_f t_n.$$



Voidaan osoittaa, että yleistetty  $\alpha$ -menetelmä on ehdoitta stabiili, jos

$$\alpha_m \leq \alpha_f \leq \frac{1}{2}, \quad \beta \geq \frac{1}{4} + \frac{1}{2}(\alpha_f - \alpha_m).$$

Matalien taajuuksien vaimennus minimoituu, kun  $\alpha_f = (\alpha_m + 1)/\sqrt{3}$ .

Katkaisuvirhe on  $\mathcal{O}[(\Delta t)^2]$  mikäli

$$\gamma = \frac{1}{2} - \alpha_m + \alpha_f.$$

Hilberin, Hughesin ja Taylorin  $\alpha$ -menetelmä saadaan kun  $\alpha_m = 0$ .



# Aikaintegrointimenetelmien vertailua

## Eksplisiittimenetelmien etuja:

- vaatii vähemmän laskentaa aika-askelta kohden
- algoritmit yksinkertaisia
- vaatii vähän muistitilaa.

Ainoa haitta on ehdollinen stabiilius

**Implisiittisten menetelmien** haitat ja edut ovat käänteiset eksplisiittimenetelmiin nähden. Haittoja ovat siten:

- algoritmit monimutkaisia
- vaativat enemmän muistia
- laskenta-aika askelta kohden huomattavasti suurempi
- pidemmästä aika-askeleesta johtuen virheet aikaintegroinnissa tulevat merkittäviksi



# Soveltuvuudesta

Rakenteiden dynamiikan ongelmat voidaan luokitella kahteen kategoriaan:

- aallon etenemisongelmat                      valitse **eksplisiittimenetelmä**
- värähtelyongelmat                              valitse **implisiittimenetelmä**



# Kirjallisuutta

- K.-J. Bathe, E.L. Wilson, *Numerical Methods in Finite Element Analysis*, Prentice-Hall, 1976, luvut 10-12.
- I. Fried, *Numerical Solution of Differential Equations*, Academic Press, 1979, luku 9
- T.J.R. Hughes, *The Finite Element Method*, Prentice-Hall, 1987, luvut 7 ja 9
- W.L. Wood, *Practical Time-stepping Schemes*, Clarendon Press, 1990

