Háskóli Íslands, raunvísindasvið

Tölvueðlisfræði - Verkefni 2

Haustmisseri 2013

Kennari - Viðar Guðmundsson

Guðjón Henning

17. október 2013

1 Jaynes Cummings Módelið

Hugsum kerfi með einangruðu atómi með tvo orkustig. Þessi orkustig svara til gunnástands $|g\rangle = \binom{0}{1}$ og örvaðs ástands $|e\rangle = \binom{1}{0}$. Orka þessara ástanda er $E_g = -\frac{1}{2}\hbar\omega$ og $E_e = \frac{1}{2}\hbar\omega$. Hamiltonvirki þessa kerfis er því

$$\hat{H}_{\text{atom}} = \frac{\hbar\omega}{2}\hat{\sigma}_z = \frac{\hbar\omega}{2}\begin{pmatrix}1 & 0\\ 0 & -1\end{pmatrix}$$

Skoðum nú skammtabundið rafsegulsvið í tómarúmi og ritum sviðið sem virkja,

$$\hat{H}_{\text{EM}} = \hbar\nu\left(\hat{a}^\dagger\hat{a} + \tfrac{1}{2}\right)$$

þar sem $\hat{a}^\dagger$ og $\hat{a}$ eru hækkunar- og lækkunarvirkjar ljóseinda og ν er horntíðni rafsegulsviðsins. Föst viðbót á orku breytir engu í virkni, svo hægt er að rita þetta sem

$$\hat{H}_{\text{EM}} = \hbar\nu\hat{a}^\dagger\hat{a}$$

Eigingildi þessarar jöfnu eru eigingildi töluvirkja ljóseindarinnar $\hat{N} = \hat{a}^\dagger\hat{a}$. Þessi eigingildi er hægt að rita sem $|M\rangle = |0\rangle, |1\rangle, |2\rangle, ...$ þar sem $\hat{H}_{\text{EM}}|M\rangle = M\hbar\nu|M\rangle$.

Hér hefur Hamiltonvirkinn verið ritaður fyrir einangrað tveggja stiga atóm og skammtabundið rafsegulsvið í tómarúmi. Ef þessum tveim kerfum er blandað saman mun rafsviðshluti EM sviðsins víxlverka við tvískaut atómsins. Þá verður Hamiltonvirki þessa blandaða sviðs

$$\hat{H} = \hat{H}_{\text{atom}} + \hat{H}_{\text{EM}} + \hat{H}_{\text{int}}$$

þar sem $\hat{H}_{\text{int}}$ er hluti víxlverkunarinnar, sem er

$$\hat{H}_{\text{int}} = \frac{\hbar\Omega}{2}\hat{E}\hat{S}$$

með $\hat{E} = \hat{a} + \hat{a}^\dagger$ (rafsviðsvirki) og $\hat{S} = \hat{\sigma}_+ + \hat{\sigma}_-$ (skautunarvirki). $\hbar\Omega$ táknar styrk víxlverkunarinnar.

$\hat{\sigma}_+$ og $\hat{\sigma}_-$ eru hækkunar- og lækkunarvirkjar Pauli sem eru

$$\hat{\sigma}_+ = |e\rangle\langle g| = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \qquad \text{og} \qquad \hat{\sigma}_- = |g\rangle\langle e| = \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix}$$

Þá verður heildar Hamiltonvirkinn

$$\hat{H} = \frac{\hbar\omega}{2}\hat{\sigma}_z + \hbar\nu\hat{a}^\dagger\hat{a} + \frac{\hbar\Omega}{2}(\hat{a}\hat{\sigma}_+ + \hat{a}\hat{\sigma}_- + \hat{a}^\dagger\hat{\sigma}_+ + \hat{a}^\dagger\hat{\sigma}_-)$$

Þessi Hamiltonvirki kallast Jaynes-Cummings Hamiltonvirkinn.

Byrjum á því að rita niður verkun virkjanna sem og heppilegar stærðir. Við viljum skala reikninga í einingum $\hbar\omega$, svo við skilgreinum

$$\hbar\omega \equiv 1$$

sem veldur því að

$$\hbar\nu = \frac{\nu}{\omega} \qquad ; \qquad \frac{\hbar\Omega}{2} = \frac{\Omega}{2\omega}$$

Og virkjarnir verka á eftirfarandi hátt

$$\begin{aligned} \sigma_+(g,N) &= (e,N) & a(e,N) &= \sqrt{N}(e,N-1) \\ \sigma_+(e,N) &= 0 & a(e,0) &= 0 \\ \sigma_-(e,N) &= (g,N) & a(g,0) &= 0 \\ \sigma_-(g,N) &= 0 & a(g,N) &= \sqrt{N}(g,N-1) \\ & & a^\dagger(g,N) &= \sqrt{N+1}(g,N+1) \\ & & a^\dagger(e,N) &= \sqrt{N+1}(e,N+1) \end{aligned}$$

Í verkefninu þarf að nota hornalínuaðferð (diagonalization method) til að reikna orkuróf Hamiltonvirkjans. Við notum grunninn $\{|i,M\rangle\}$, þar sem $|i,M\rangle = |i\rangle \otimes |M\rangle$ og $i \in \{g,e\}$. Góð leið til að númera ástönd grunnsins með einni tölu (til að byggja fylki) er

$$|g,0\rangle = |1\rangle,\ |e,0\rangle = |2\rangle,\ |g,1\rangle = |3\rangle,\ |e,1\rangle = |4\rangle,\ ...,\ |i,k\rangle = |\mu\rangle$$

Nú er hægt að skoða hvernig mynda má fylkjastökin

$$\langle i,M|\hat{H}|j,N\rangle$$

Skoðum fyrst $\hat{H}_{\text{atom}}$

$$\langle i,M|\hat{H}_{\text{atom}}|j,N\rangle = \frac{\hbar\omega}{2}\,\delta_{ij}\,\delta_{MN}\,\delta_{je} - \frac{\hbar\omega}{2}\,\delta_{ij}\,\delta_{MN}\,\delta_{jg}$$

og svo $\hat{E}_{\text{EM}}$, en rifjum fyrst upp að

$$\begin{aligned} \hat{a}^\dagger\hat{a}|n\rangle &= \hat{a}^\dagger\sqrt{n}|n-1\rangle \\ &= n|n\rangle \end{aligned}$$

sem gefur okkur

$$\langle i,M|\hat{H}_{\text{EM}}|j,N\rangle = \hbar\nu N\,\delta_{ij}\,\delta_{MN}$$

Að reikna víxlverkunina er aðeins snúnara. Fylkjastökin þar eru

$$\langle i,M|\tfrac{\hbar\Omega}{2}\{\hat{a}\hat{\sigma}_+ + \hat{a}\hat{\sigma}_- + \hat{a}^\dagger\hat{\sigma}_+ + \hat{a}^\dagger\hat{\sigma}_-\}|j,N\rangle$$

Skoðum einn lið í einu og munum eftir virkni virkjanna.

$$\begin{aligned}\langle i, M|\hat{a}\hat{\sigma}_+|j, N\rangle &= \sqrt{N}\,\delta_{M,N-1}\,\delta_{ie}\,\delta_{jg}\\ \langle i, M|\hat{a}\hat{\sigma}_-|j, N\rangle &= \sqrt{N}\,\delta_{M,N-1}\,\delta_{ig}\,\delta_{je}\\ \langle i, M|\hat{a}^\dagger\hat{\sigma}_+|j, N\rangle &= \sqrt{N+1}\,\delta_{M,N+1}\,\delta_{ie}\,\delta_{jg}\\ \langle i, M|\hat{a}^\dagger\hat{\sigma}_-|j, N\rangle &= \sqrt{N+1}\,\delta_{M,N+1}\,\delta_{ig}\,\delta_{je}\end{aligned}$$

Tökum þetta saman og fáum

$$\begin{aligned}\langle i, M|\hat{H}_{\mathrm{EM}}|j, N\rangle &= \frac{\hbar\Omega}{2}\left(\sqrt{N}\,\delta_{M,N-1}(\delta_{ie}\delta_{jg}+\delta_{ig}\delta_{je}) + \sqrt{N+1}\,\delta_{M,N+1}(\delta_{ie}\delta_{jg}+\delta_{ig}\delta_{je})\right)\\ &= \frac{\hbar\Omega}{2}\left(\sqrt{N}\,\delta_{M,N-1} + \sqrt{N+1}\,\delta_{M,N+1}\right)(\delta_{ie}\delta_{jg}+\delta_{ig}\delta_{je})\end{aligned}$$

Við viljum einnig skoða meðalfjölda ljóseinda hvers ástands sem fall af víxlverkunarstyrknum $\frac{\Omega}{2\omega}$. Fyrst skilgreinum við

$$|\alpha) = \sum_i C_{i\alpha}|i\rangle$$

þar sem $C_{i\alpha}$ eru stök eiginvigranna. Þá er einoka virki skilgreindur

$$|\alpha) = U|i\rangle$$

sem gefur fylkið

$$(\beta|\hat{N}|\alpha) = \langle j|U^\dagger\hat{N}U|i\rangle = \sum_{ij} C^*_{j\beta}C_{i\alpha}\langle j|\hat{N}|i\rangle$$

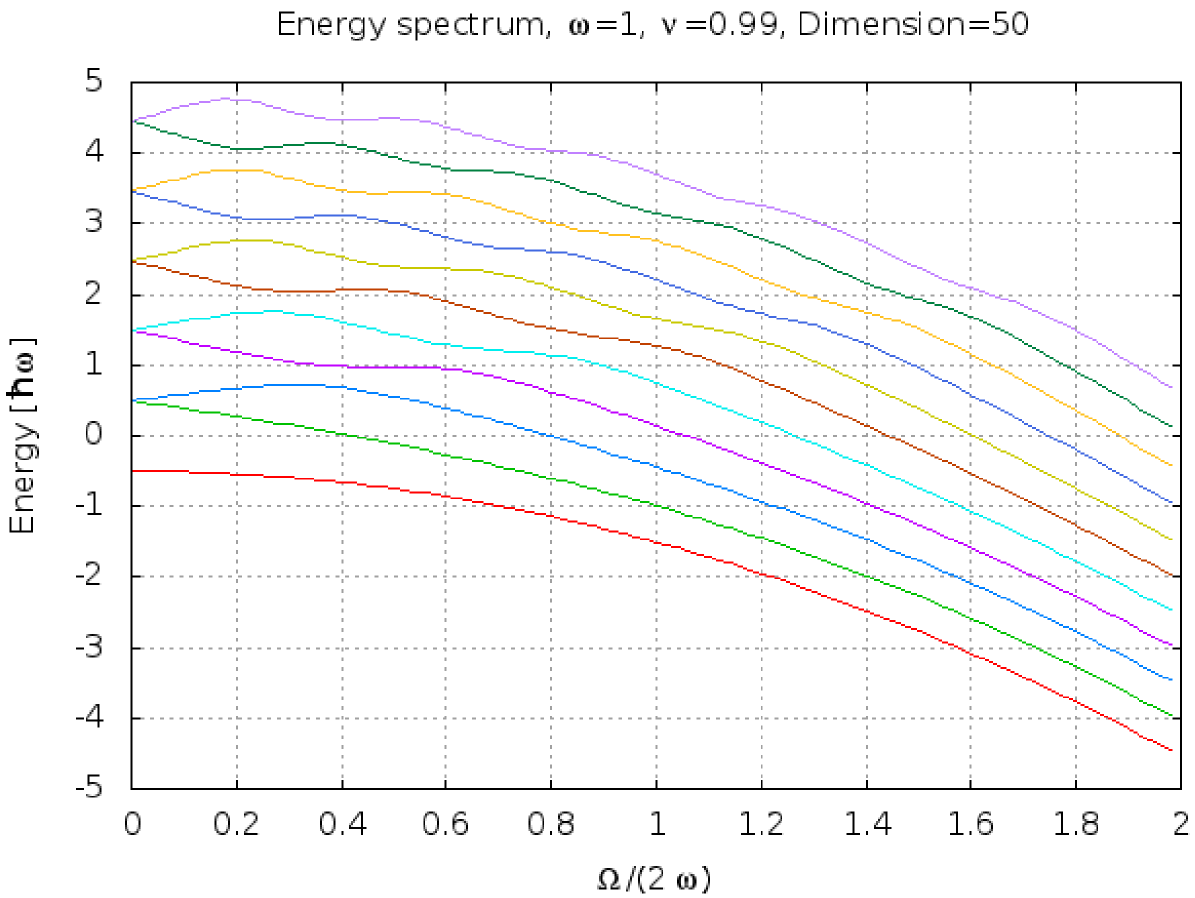
Þetta fylki les hornalínu til að sjá meðalfjölda ljóseinda í kerfinu. Hér er gott að nýta innbyggða forritið í Fortran: MATMUL, sem framkvæmir fylkjamargföldun. Reyndar eru hér notuð sambærileg stef sem gera kleift að reikna einnig tvinntölugild fylki.

1.1 Orkugildi ástandanna

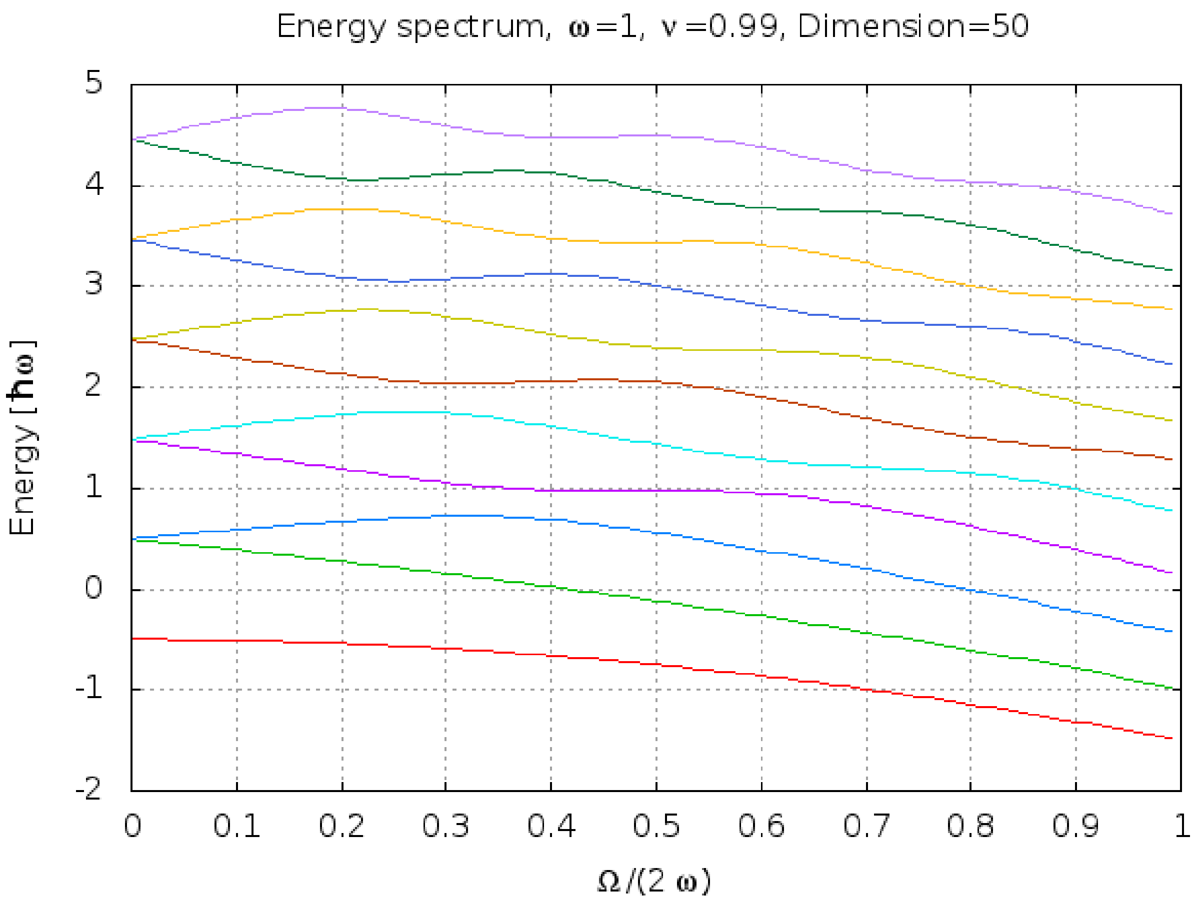
Eitt af skilyrðum JC módelsins er að ljóseindaorkan þarf að vera sambærileg við orkbil ástanda, eða $\omega \simeq \nu$. Því er valið

$$\delta = \frac{|\omega - \nu|}{\omega} \simeq 1\%$$

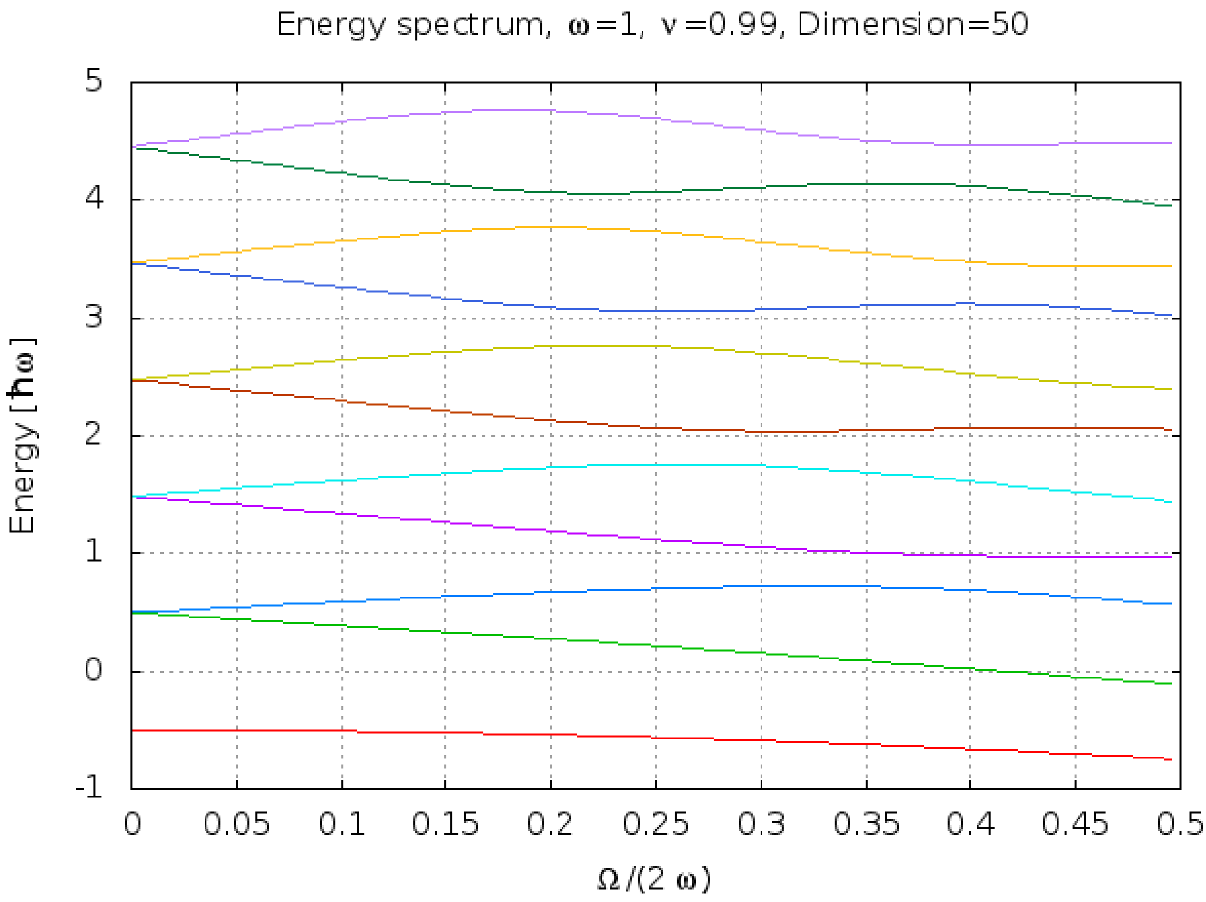
Þetta gefur augaleið að velja $\omega = 1$ og $\nu = 0.99$. Til gamans skoðum við líka niðurstöður þegar $\omega = 1$ og $\nu = 0.5$. Einnig prófum við að hækka víddina og skoða fleiri ástönd.



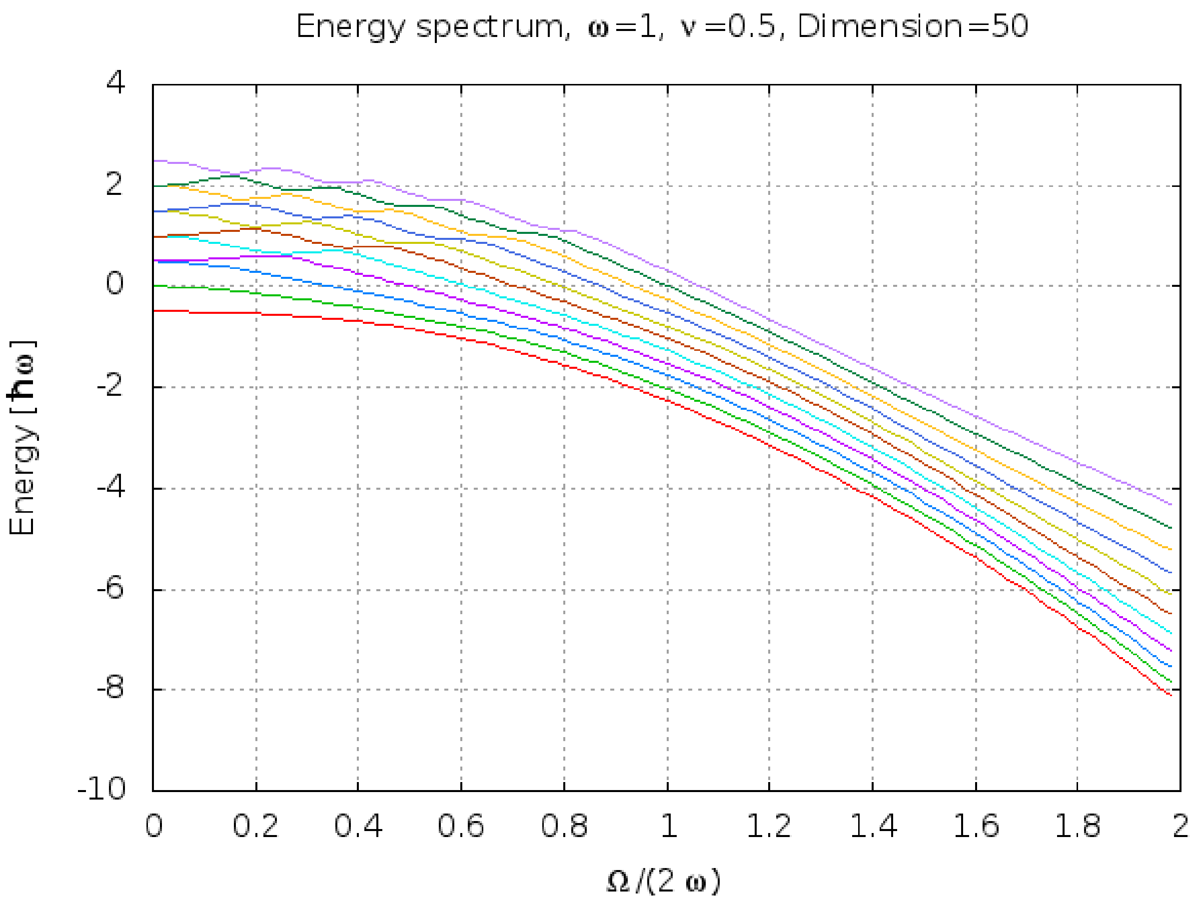
(a) $\frac{\Omega}{2\omega} = [0:2]$, $\nu = 0.99$.



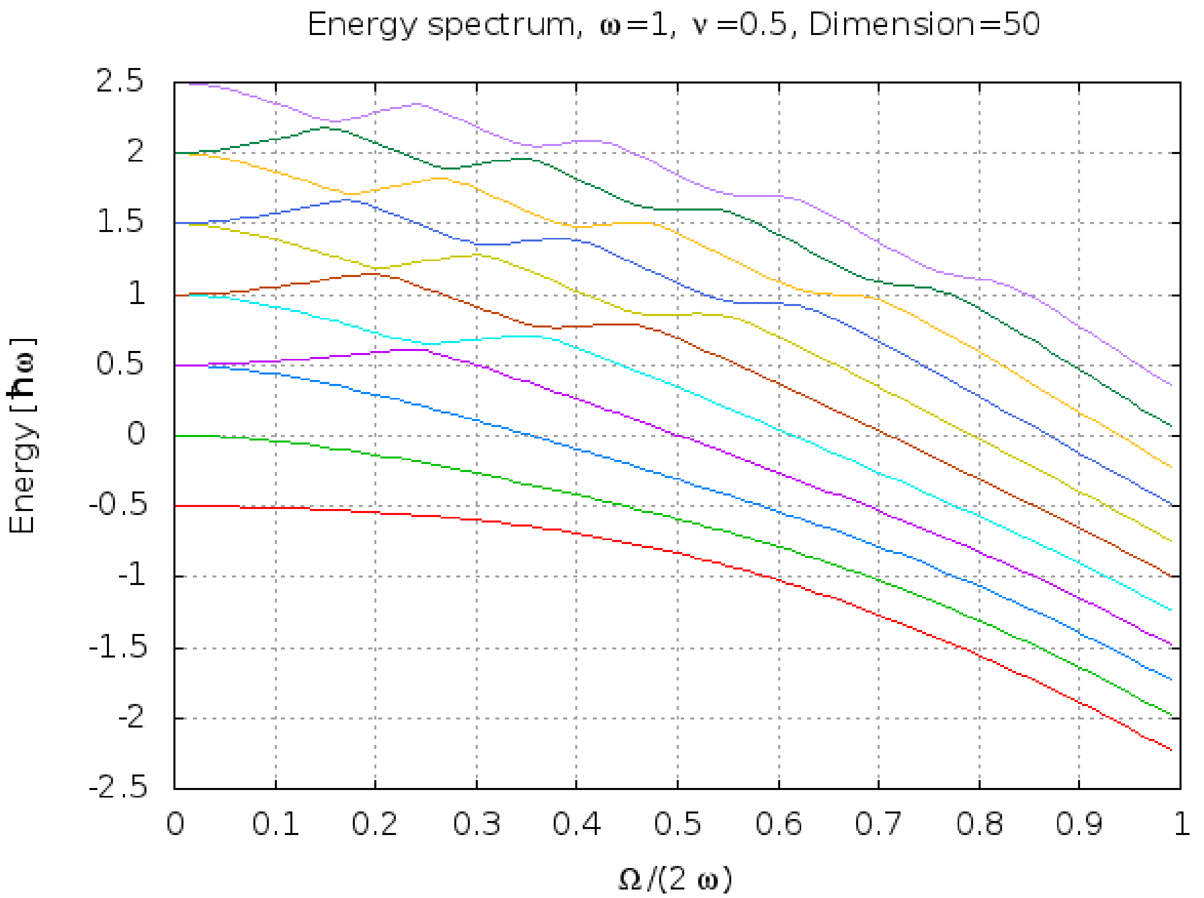
(b) $\frac{\Omega}{2\omega} = [0:1]$, $\nu = 0.99$.



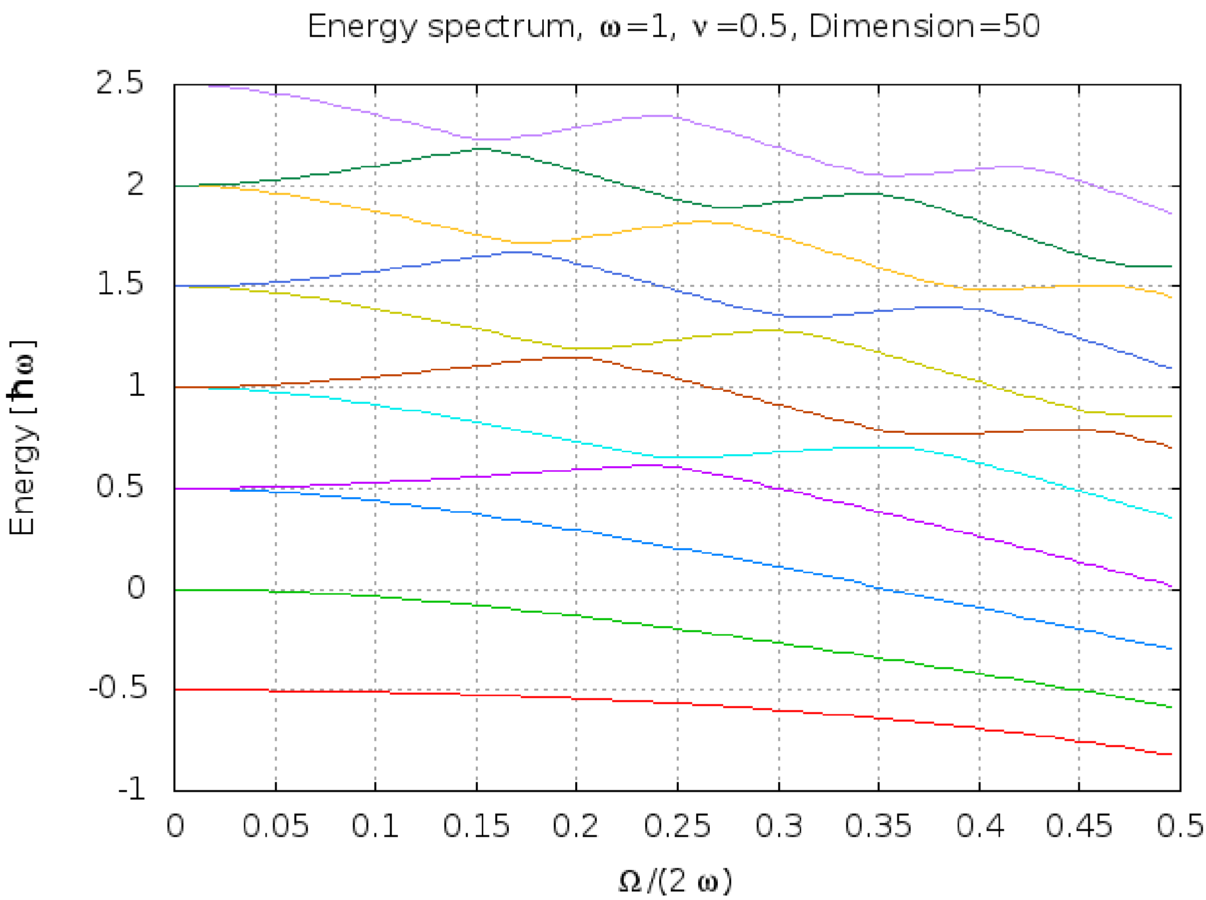
(c) $\frac{\Omega}{2\omega} = [0:0.5]$, $\nu = 0.99$.



(d) $\frac{\Omega}{2\omega} = [0:2]$, $\nu = 0.5$.

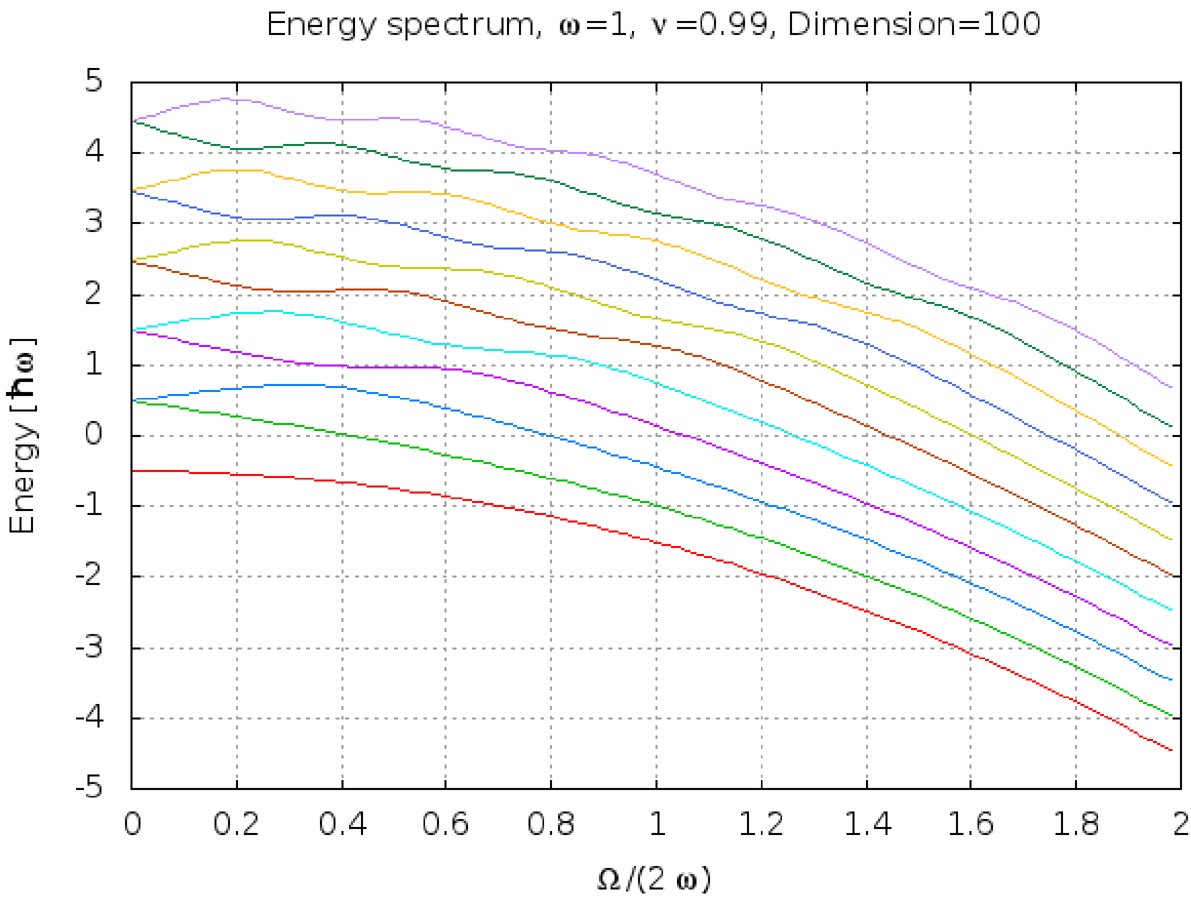


(e) $\frac{\Omega}{2\omega} = [0:1]$, $\nu = 0.5$.

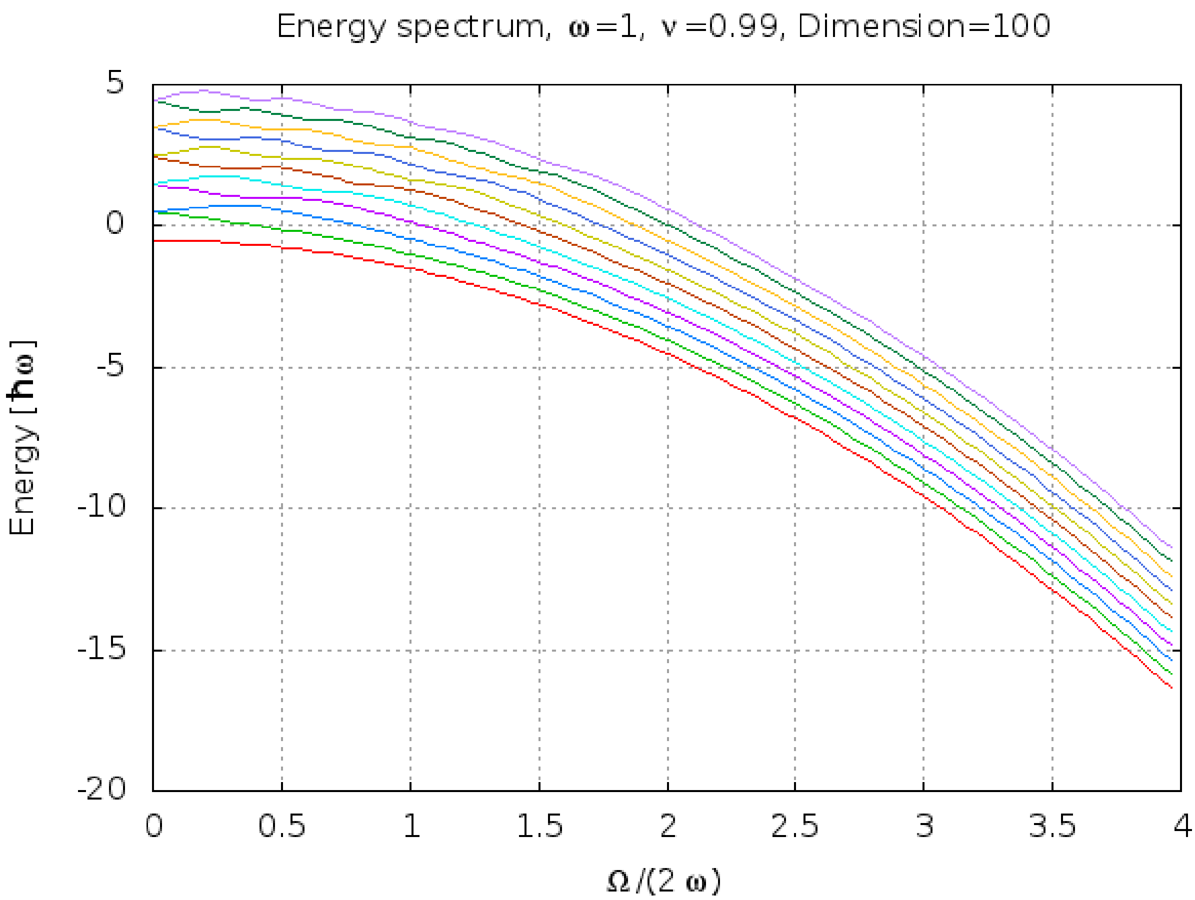


(f) $\frac{\Omega}{2\omega} = [0:0.5]$, $\nu = 0.5$.

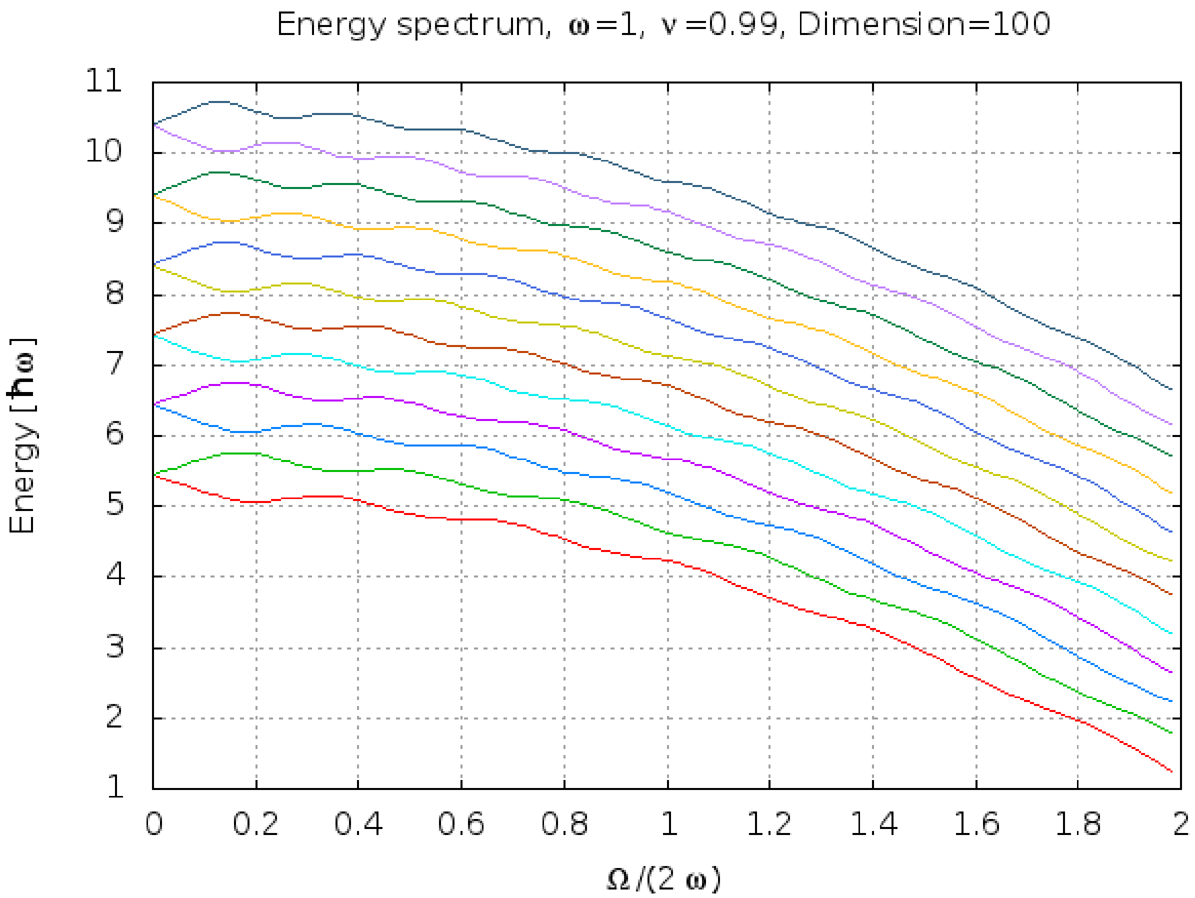
Mynd 1.1: Orka fyrstu 11 ástandanna sem fall af víxlverkunarstyrk.



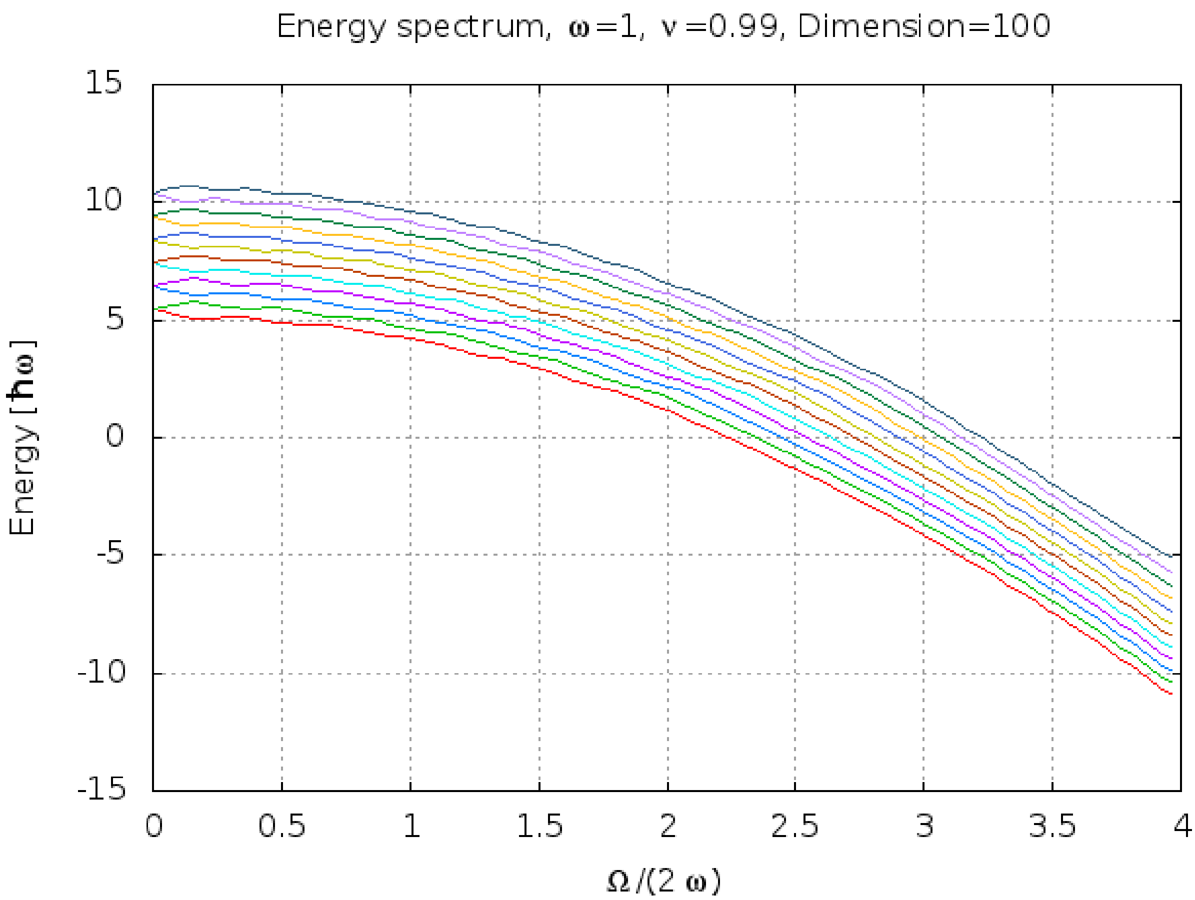
(a) Fyrstu 11 ástönd. $\frac{\Omega}{2\omega} = [0:2]$, $\nu = 0.99$.



(b) Fyrstu 11 ástönd. $\frac{\Omega}{2\omega} = [0:4]$, $\nu = 0.99$.



(c) Næstu 11 ástönd. $\frac{\Omega}{2\omega} = [0:2]$, $\nu = 0.99$.

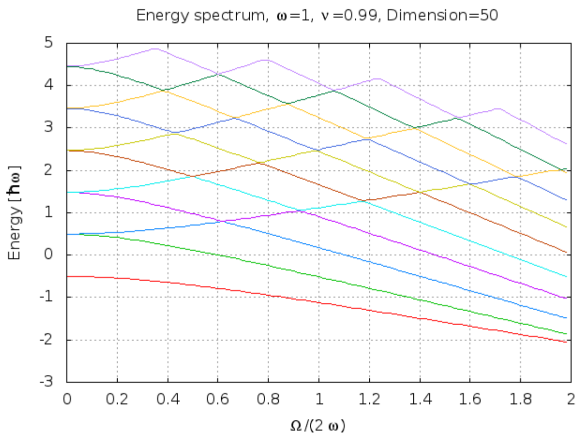


(d) Næstu 11 ástönd. $\frac{\Omega}{2\omega} = [0:4]$, $\nu = 0.99$.

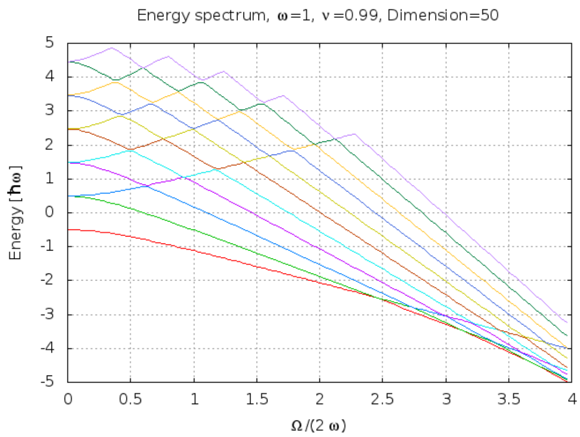
Mynd 1.2: Orka fyrstu 22 ástandanna sem fall af víxlverkunarstyrk með vídd 100.

1.2 Víxlverkun: Herma og andherma

Í víxlverkunarlið Hamiltonvirkjans eru fjórir liðir. Tveir þeirra, $\hat{a}\hat{\sigma}_+$ og $\hat{a}^\dagger\hat{\sigma}_-$, kallast hermuliðir. Hinir tveir, $\hat{a}^\dagger\hat{\sigma}_+$ og $\hat{a}\hat{\sigma}_-$, kallast andhermuliðir. Hér skoðum við áhrif þeirra í þrem tilvikum þar sem: hermuliðum er sleppt, andhermuliðum er sleppt, og víxlverkun er sleppt.

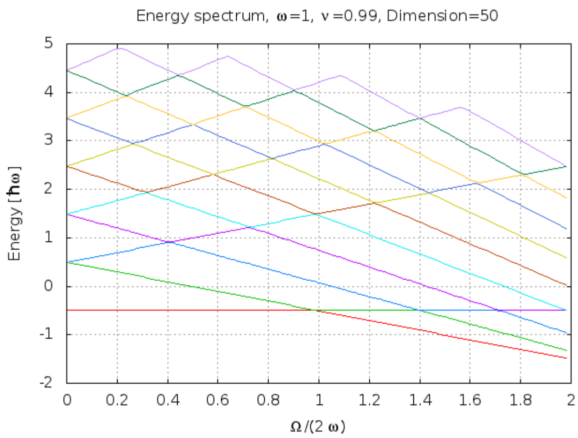


(a) $\frac{\Omega}{2\omega} = [0:2]$, $\nu = 0.99$.

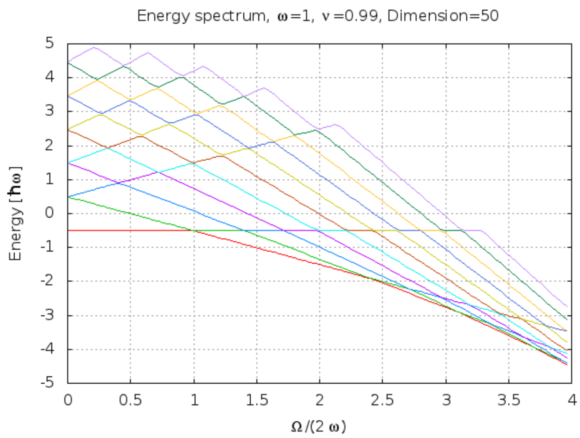


(b) $\frac{\Omega}{2\omega} = [0:4]$, $\nu = 0.99$.

Mynd 1.3: Orkuróf skoðað með engum andhermuliðum.

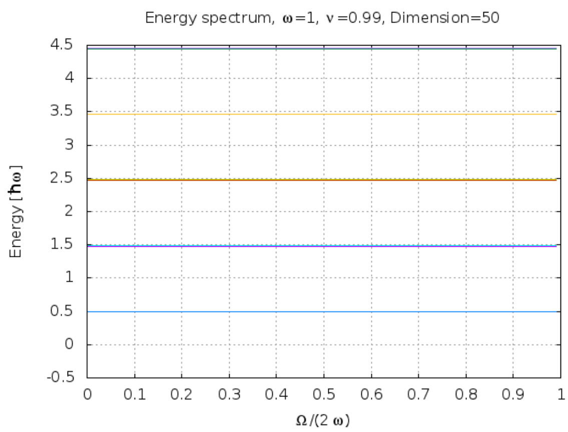


(a) $\frac{\Omega}{2\omega} = [0:2]$, $\nu = 0.99$.



(b) $\frac{\Omega}{2\omega} = [0:4]$, $\nu = 0.99$.

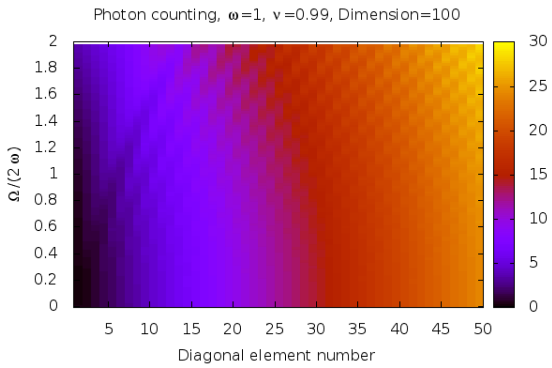
Mynd 1.4: Orkuróf skoðað með engum hermuliðum.



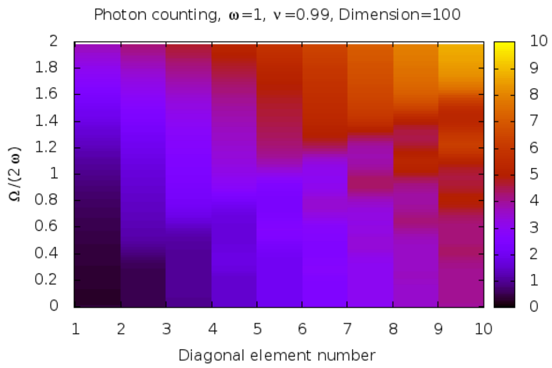
Mynd 1.5: Orkuróf skoðað með engri víxverkun.

1.3 Talning ljóseinda

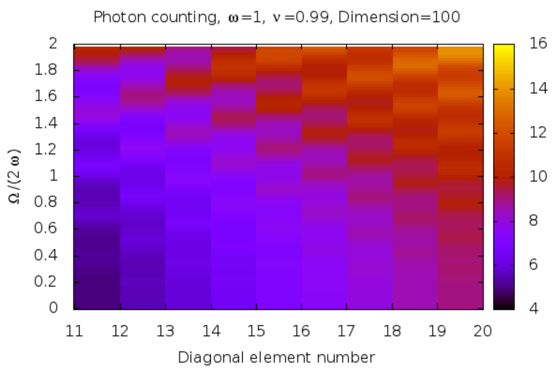
Við það að reikna fylkið $(\beta|\hat{N}|\alpha)$ og skoða hornalínu þess, $(\alpha|\hat{N}|\alpha)$, gefur beint meðalfjölda ljóseinda kerfisins. Þó svo að reiknað sé með vídd fylkis sem 100x100 er áhugaverðast að fylgjast með fyrstu hornalínustökunum, þar sem þau virðast hafa augljósari og viðameiri breytingu með auknum víxlverkunarstyrk. Við skoðum þrjár mismunandi útfærslur á þessum niðurstöðum á formi „hitagrafs", „línugrafs", og „girðingagrafs". Erfitt er að lesa úr gröfum sem innihalda 50 hornalínustök, þannig að yfirleitt eru skoðuð nokkur hornalínustök á hverri mynd til að sjá breytingar betur.



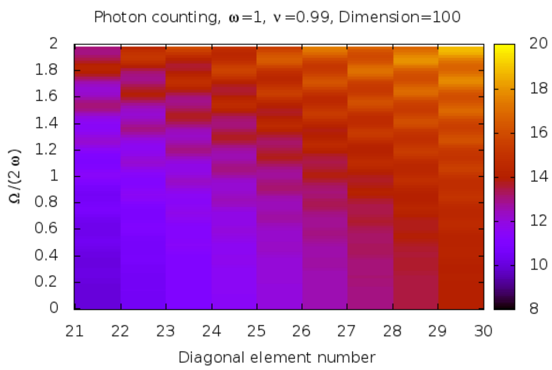
(a) Hornalínustök 1-50.



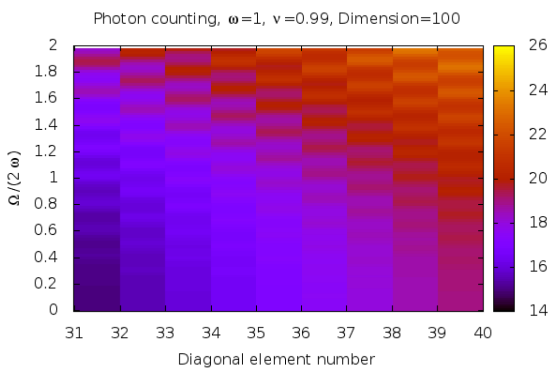
(b) Hornalínustök 1-10.



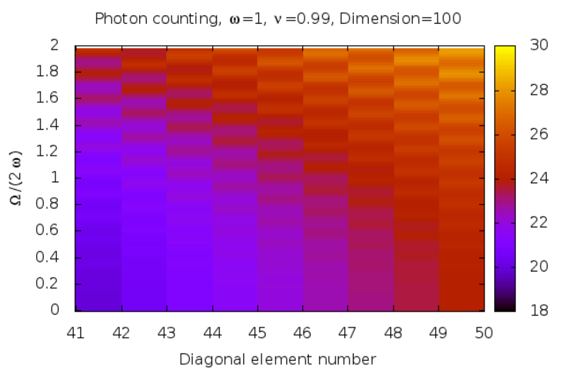
(c) Hornalínustök 11-20.



(d) Hornalínustök 21-30.

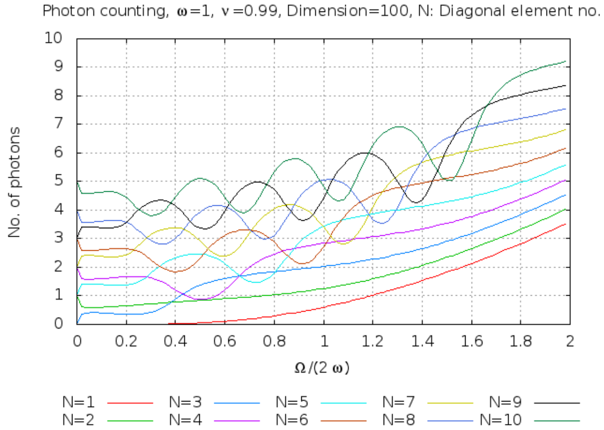


(e) Hornalínustök 31-40.

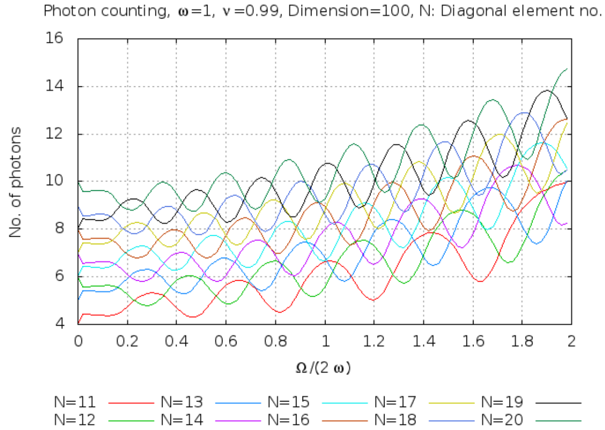


(f) Hornalínustök 41-50.

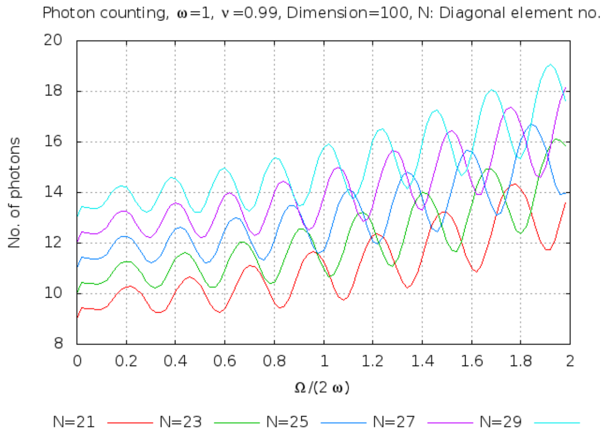
Mynd 1.6: Hitagraf sem sýnir meðalfjölda ljóseinda fyrir hækkandi víxlverkunarstyrk.



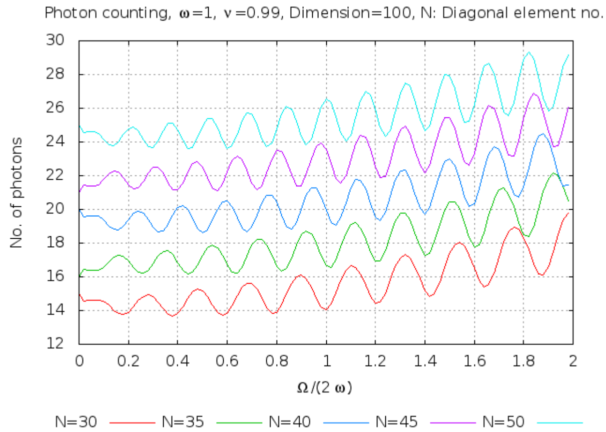
(a) Hornalínustök 1-10.



(b) Hornalínustök 11-20.

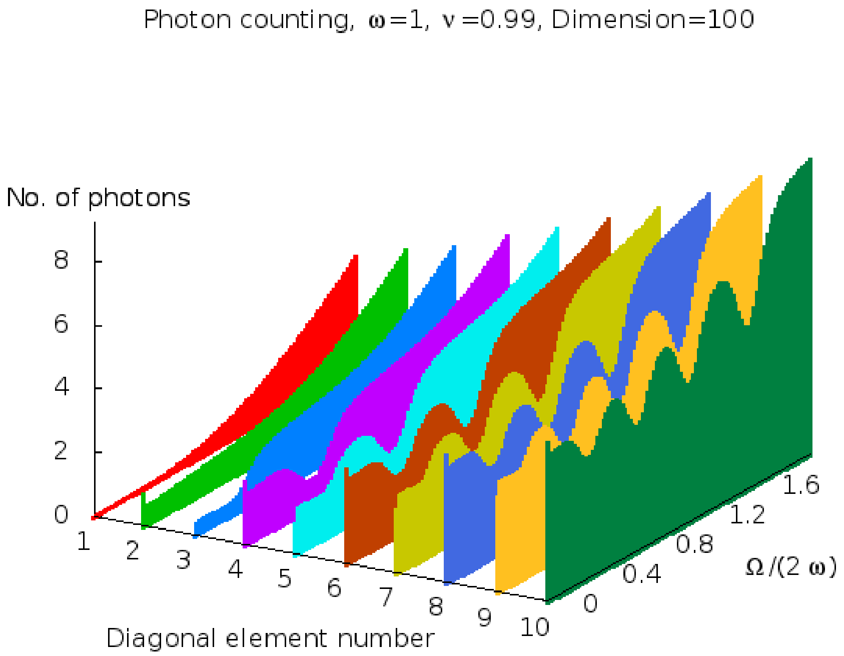


(c) Hornalínustök á bilinu 21-30.

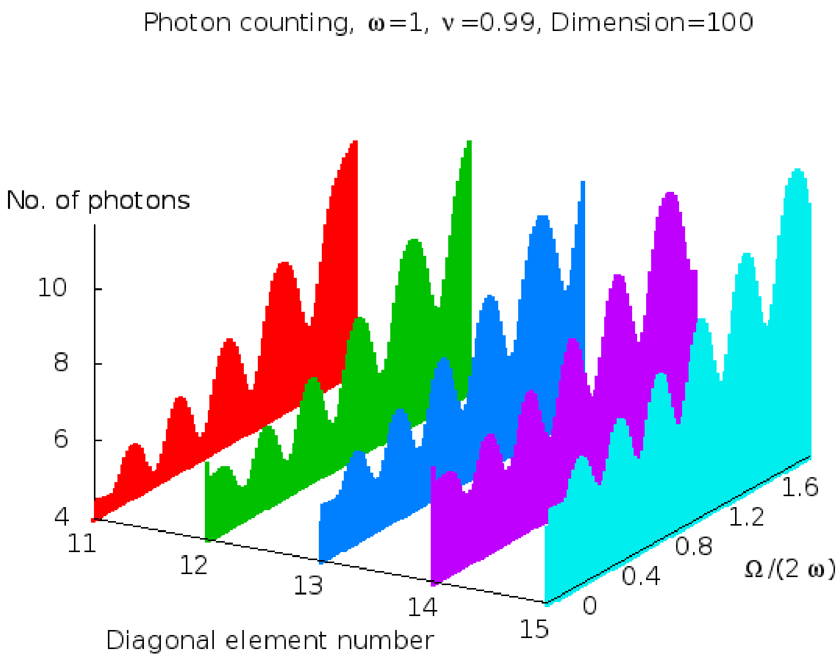


(d) Hornalínustök á bilinu 31-50.

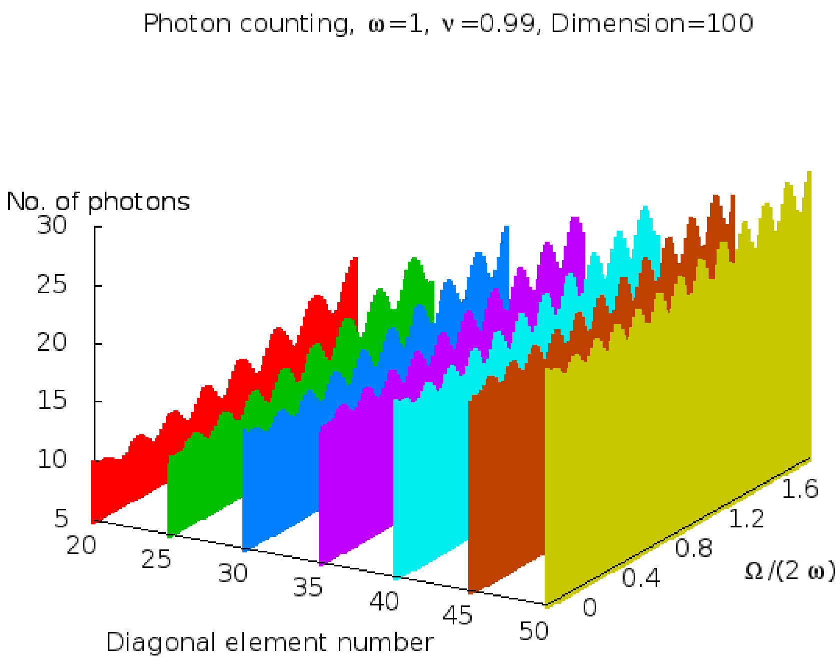
Mynd 1.7: Línugraf sem sýnir meðalfjölda ljóseinda fyrir hækkandi víxlverkunarstyrk.



(a) Hornalínustök 1-10.



(b) Hornalínustök 11-15.



(c) Hornalínustök á bilinu 20-50.

Mynd 1.8: Girðingagraf sem sýnir meðalfjölda ljóseinda fyrir hækkandi víxlverkunarstyrk.

1.4 Niðurstöður

1.4.1 Orkugildi ástandanna

Við skoðun á myndum 1.1 og 1.2 sést að orkan lækkar eftir því sem víxlverkunin er sterkari ásamt því að orkan sveiflast meira eftir því sem orkuástandið er hærra. Við það að lækka ν, og þar með hækka δ, lækkar orkan mun örar eftir því sem víxlverkunarstyrkur hækkar, sem sjá má t.d. með því að bera saman myndir 1.1d og 1.1a. Einnig sést það að sveiflur orkuástandanna stækka fyrir lægra ν með því að bera saman t.d. myndir 1.1e og 1.1.b.

1.4.2 Víxlverkun: Herma og andherma

Hér er merkilegt að taka eftir því að orka ástandanna helst stöðug fyrir stutt bil á víxverkunarstyrknum í orkugildinu $-\hbar\omega/2$ þegar engir hermuliðir eru til staðar, eins og sést á samanburði mynda 1.4b og 1.3b. Orkubreyting ástandanna, þegar hermu- eða andhermuliðum er sleppt, breytist í stökkum, en ekki sveiflum, líkt og gerðist þegar báðir liðir voru teknir með í reikninga. Síðan ætti það ekki að koma á óvart að engin orkubreyting verður á ástöndunum þegar víxlverkunin er ekki til staðar, eins og sjá má á mynd 1.5.

1.4.3 Talning ljóseinda

Hækkandi styrkur víxlverkunar hækkar og lækkar meðalfjölda ljóseinda kerfisins í sveiflum, sérstaklega þegar skoðað er lengra inn í hornalínustök fylkisins. Sennilega hefði maður áætlað að fjöldi ljóseinda myndi hækka nokkuð stöðugt fyrir hækkandi styrk víxlverkunar, áður en reikningar hófust.

Hins vegar er breytingin nánast línuleg aukning (réttara sagt stöðug aukning, engar sveiflur) fyrir fyrstu hornalínustökin, að því undanskildu þegar víxlverkunarstyrkurinn er lítill. Út frá þessu (og við athugun á myndum 1.6-1.8) er hægt að áætla að fjöldi ljóseinda hækki línulega eftir að ákveðnum víxlverkunarstyrksþröskuldi er náð. Áður en þessum þröskuldi er náð eykst meðalljóseindafjöldi með sveiflukenndum hætti.

Því er hægt að gera ráð fyrir að sjá línulega fjölgun ljóseinda fyrir lítinn víxlverkunarstyrk þegar víddin er lítil (eða fyrstu hornalínustök í stóru fylki), því þá er víxlverkunarstyrksþröskuldurinn lítill. Ef víddin er mikil og langt er farið inn í hornalínustök fylkisins þarf víxlverkunarstyrkurinn að vera mun sterkari til að sjá línulega aukningu ljóseinda, því þá hefur þröskuldurinn einnig hækkað.

2 Kóði

```
# Python code to run make Adal and Adal
#! /usr/bin/env python
# -*- coding: utf-8 -*-
import os
os.system('make Adal')
os.system('rm *.dtx')
inc = 0.04
for x in range(0,100):
        os.system('echo "' + str(x*inc) + 'D0"' + ' | time nice ./Adal')
        os.system('mv H.dtx Eig/H' + str(x) + '.dtx' )
        os.system('mv Eigenval.dtx Eig/Eigenval' + str(x) + '.dtx')
        os.system('mv Eigenvect.dtx huehue/Eigenvect' + str(x) + '.dtx')
        os.system('mv ALLT.dtx a/ALLT' + str(x) + '.dtx')
        os.system('mv ENN.dtx a/ENN' + str(x) + '.dtx')
#       os.system('mv UPU.dtx a/UPU' + str(x) + '.dtx')
        os.system('mv Hornalinufylki.dtx a/Hornalinufylki' + str(x) + '.dtx
```

```
   PROGRAM Adal
!------------------------------------------------------------------------
!------------------------------------------------------------------------
   USE omp_lib                 ! For OpenMP parallel processing
   USE Mod_Precision           ! Module for setting double precision
   USE Mod_Init                ! Initial values
   USE Mod_Fields              ! Global variables

   USE mkl95_lapack            ! Subroutines
!    USE mkl95_blas             ! Subroutines

        USE blas95

        USE PAB
        USE PcAB
        USE PABZ

   IMPLICIT NONE

!------- Local variables ---------------------------------------------
!
   INTEGER            :: M, N, ierr, info
!-------- Uttak ------------------------------------------------------
        OPEN(UNIT=11,FILE=       'H_atom.dtx'      )!,STATUS='NEW')
        OPEN(UNIT=12,FILE=       'H_EM.dtx'                 )!,STATUS='NEW')
        OPEN(UNIT=13,FILE=       'H_int.dtx'                )!,STATUS='NEW')
        OPEN(UNIT=14,FILE=       'H.dtx'                    ,STATUS='NEW')
        OPEN(UNIT=15,FILE=     'Eigenval.dtx'            ,STATUS='NEW')
        OPEN(UNIT=16,FILE=       'Eigenvect.dtx'            ,STATUS='NEW')
        OPEN(UNIT=17,FILE=       'ALLT.dtx'        ,STATUS='NEW')
        OPEN(UNIT=18,FILE=       'ENN.dtx'         ,STATUS='NEW')
!       OPEN(UNIT=19,FILE=       'UPU.dtx'                  ,STATUS='NEW')
```

```
        OPEN(UNIT=20,FILE=        'Hornalinufylki.dtx' ,STATUS='NEW')

!  STATUS='NEW' the program will only run if these output files
!  do not exist
!------------------------------------------------------------------
! lesum OOmega fra keyrsla.py sem breytir thvi gildi i hverri
! keyrslu.
         READ(5,*) OOmega
   ierr = 0

   ALLOCATE(H(Nf,Nf),H_atom(Nf,Nf),H_EM(Nf,Nf),&
H_int(Nf,Nf),Ham(Nf,Nf), STAT=ierr)

H = Czero ! Zero-ing, Czero is defined in Mod_Init.f90
H_atom = Czero
H_EM = Czero
H_int = Czero
Ham = Czero

! Viljum skoda ahrif breytilegs Omega/omega
HLUTFALL = 0.5_dp*OOmega/omega

! Bua til H_atom fylki
DO M = 1, Nf
        DO N = 1, Nf
    IF ((M .EQ. N) .AND. (MOD(M,2) .EQ. 1)) THEN
                        H_atom(M,N) = (0.5_dp*-1.0_dp)
                ELSE IF (M .EQ. N) THEN
                        H_atom(M,N) = (0.5_dp*1.0_dp)
                END IF
                WRITE(11,FMT='(I4,2X,I4,2X,E15.8,1X,E15.8)') &
                M, N, H_atom(M,N)
        END DO
END DO
WRITE(11,FMT='')

! Bua til H_EM fylki
DO M=1,Nf
        DO N=1,Nf
                IF ((M .EQ. N) .AND. (MOD(N,2) .EQ. 0)) THEN
! Fjoldi ljoseinda haekkar i odru hverju skrefi.
! T.d. fyrir fyrstu tvo sletttoluskrefin, tha skodum vid N=2,
! sem gefur (N-2)/2=0 ljoseindir. Sidan faest fyrir
! N=4 ad (N-2)/2=1.
! Fyrir oddatoluskrefin tha er fjoldi ljoseinda
! (N-1)/2
                        H_EM(M,N) = (REAL(N-2.0_dp)/2)*(nu/omega)
                ELSE IF ((M .EQ. N) .AND. (MOD(N,2) .EQ. 1)) THEN
                        H_EM(M,N) = (REAL(N-1.0_dp)/2)*(nu/omega)
                END IF
        WRITE(12,FMT='(I4,2X,I4,2X,E15.8,1X,E15.8)') &
```

```
                M, N, H_EM(M,N)
        END DO
END DO
WRITE(12,FMT='')


! Bua til H_int fylki
DO M=1,Nf
        DO N=1,Nf
! M=N-1 og M=N+1 thydir ad i er odd og j er slett EDA i slett og j odd,
! thvi er otharfi ad skoda hvort baedi gildi. Her tharf bara ad athuga
! hvort N se slett eda odd upp a hvort thad thurfi ad draga 1 eda 2 fra
! i reikningum a fjolda ljoseinda. Setjum inn /2 i stad *0.5_dp
        IF ((M .EQ. (N-1)) .AND. (MOD(N,2) .EQ. 0)) THEN
        H_int(M,N) = SQRT((REAL(N-2.0_dp)/2) + 1.0_dp)*HLUTFALL
        ELSE IF ((M .EQ. (N-1)) .AND. (MOD(N,2) .EQ. 1)) THEN
        H_int(M,N) = SQRT(REAL(N-1.0_dp)/2)*HLUTFALL
        ELSE IF ((M .EQ. (N+1)) .AND. (MOD(N,2) .EQ. 0)) THEN
        H_int(M,N) = SQRT((REAL(N-2.0_dp)/2) + 1.0_dp)*HLUTFALL
        ELSE IF ((M .EQ. (N+1)) .AND. (MOD(N,2) .EQ. 1)) THEN
        H_int(M,N) = SQRT((REAL(N-1.0_dp)/2) + 1.0_dp)*HLUTFALL
        END IF
        WRITE(13,FMT='(I4,2X,I4,2X,E15.8,2X,E15.8)') M, N, H_int(M,N)
        END DO
END DO
WRITE(13,FMT='')

DO M=1,Nf
        DO N=1,Nf
                H(M,N)=H_atom(M,N)+H_EM(M,N)+H_int(M,N)
                WRITE(14,FMT='(I4,2X,I4,2X,E15.8,2X,E15.8)') M,N,H(M,N)
        END DO
                WRITE(14,FMT='')
END DO
WRITE(14,FMT='')

Ham = H

ALLOCATE(Eigval(Nf),Eigvect(Nf,Nf),U(Nf,Nf), &
ENN(Nf,Nf), ALLT(Nf,Nf),UPU(Nf,Nf),Hornalinufylki(Nf,Nf), STAT=ierr)

CALL HEEVD(H,Eigval,JOB,UPLO,info)    ! Subroutine from MKL for
! finding eigenvalues and vectors, input is a lower or upper
! triangular

U = H

DO M = 1, Nf
WRITE(15,FMT='(I4,2X,E15.8,2X,E15.8)') &
M, HLUTFALL, Eigval(M)
! Eigingildi a H
```

```
END DO
WRITE(15,FMT='')

! Eiginvigrar
DO M = 1, Nf
        DO N = 1,Nf
        WRITE(16,FMT='(I4,2X,I4,2XE15.8,2X,E15.8,2X,E15.8)') &
N,M, HLUTFALL, U(N,M)
        END DO
        WRITE(16,FMT='')
END DO

! Reikna ENN
DO M = 1,Nf
        DO N = 1,Nf
                IF ((M .EQ. N) .AND. (MOD(N,2) .EQ. 0)) THEN
                        ENN(M,N) = (N-2)/2
                ELSE IF ((M .EQ. N) .AND. (MOD(N,2) .EQ. 1)) THEN
                        ENN(M,N) = (N-1)/2
                END IF
                WRITE(18,FMT='(I4,2X,I4,2X,E15.8,2X,E15.8,1X,E15.8)') &
M, N, ENN(M,N)
        END DO
END DO

! reikna saman UP, U, og ENN
ALLT = MATMULVG(MATMULVGc(U,ENN),U)
DO M = 1,Nf
        DO N = 1,Nf
!               IF (M .EQ. N) THEN
                WRITE(17,FMT='(I4,2X,I4,2X,E15.8,2X,E15.8,1X,E15.8)') &
M, N, ALLT(M,N)
!               END IF
        END DO
                WRITE(17,FMT='')
END DO

! Hofum bara ahuga a hornalinunni, (alpha|N|alpha)
Hornalinufylki = Czero
        DO N = 1,Nf
         Hornalinufylki(N,N) = ALLT(N,N)
        WRITE(20,FMT='(I4,2X,E15.8,2X,E15.8,2X,E15.8,1X,E15.8)') &
N, HLUTFALL, Hornalinufylki(N,N)
        END DO
                WRITE(20,FMT='')

! reikna UPU til ad ga hvort einingafylkid faist ekki
!UPU = MATMULVGc(U,U)
!DO M = 1,Nf
!       DO N = 1,Nf
!               WRITE(19,FMT='(I4,2X,I4,2X,E15.8,2X,E15.8,1X,E15.8)') &
```

```
!M, N, UPU(M,N)
!        END DO
!                 WRITE(19,FMT='')
!END DO

DEALLOCATE(H,Eigval,U,Ham, STAT=ierr)

!-------------------------------------------------------------------------
!
END PROGRAM Adal

   MODULE Mod_Init

!-------------------------------------------
! Global parameters, parameters can not be changed in a program
!-------------------------------------------

    USE Mod_Precision

    IMPLICIT NONE

    INTEGER,           PARAMETER      :: NumThreads      = 2 !4

    INTEGER,           PARAMETER      :: Nf   = 100 !128     ! 128

!--------------------------------------------------------------------------------------------

    REAL(KIND=dp),      PARAMETER      :: pi   = 3.14159265358979324_dp
    REAL(KIND=dp),      PARAMETER      :: pi2i = 1.0_dp/(2.0_dp*pi)
    REAL(KIND=dp),      PARAMETER      :: pid2 = pi/2.0_dp

        REAL(KIND=dp),          PARAMETER                  :: omega = 1.0_dp
        REAL(KIND=dp),          PARAMETER                  :: nu = 0.99_dp
        REAL(KIND=dp)                                :: OOmega
        REAL(KIND=dp)                                :: HLUTFALL

    COMPLEX(KIND=dp),   PARAMETER      :: ci     = CMPLX(0.0_dp, 1.0_dp)
    COMPLEX(KIND=dp),   PARAMETER      :: CUnit  = CMPLX(1.0_dp, 0.0_dp)
    COMPLEX(KIND=dp),   PARAMETER      :: Czero  = CMPLX(0.0_dp, 0.0_dp)

    CHARACTER(LEN=1), PARAMETER       :: JOB    = 'V', UPLO  = 'U'
    CHARACTER(LEN=1), PARAMETER       :: TRANS = 'N', RANGO = 'I'


!--------------------------------------------------------------------------------------------
   END MODULE Mod_Init

   MODULE Mod_Fields

!-------------------------------------------
```

```
    USE Mod_Precision
    USE Mod_Init

    IMPLICIT NONE

    INTEGER                          :: Nmax

    REAL(KIND=dp)                    :: al

    COMPLEX(KIND=dp)                 :: cl

    CHARACTER(LEN=1)                 :: TransA, TransB


!--------------------------------------------------------------------------

  REAL(KIND=dp), ALLOCATABLE, DIMENSION(:)    :: Eigval, eiginfall

  COMPLEX(KIND=dp), ALLOCATABLE, DIMENSION(:,:)  :: H,H_atom, &
H_EM,H_int, Eigvect, U, ENN, ALLT, Ham, UPU, Hornalinufylki


!----------------------------------------------------------------------------------------------

   END MODULE Mod_Fields

   MODULE Mod_Precision

!------------------------------------------
! Determination of precision parameters
!------------------------------------------

    IMPLICIT NONE

    INTEGER, PARAMETER    :: dp = 8    ! Double precision parameter

!-------------------------------------------------
   END MODULE Mod_Precision

   MODULE PAB
   CONTAINS
   FUNCTION MATMULVG(Af,Bf)

   USE Mod_Precision

   USE mkl95_lapack
   USE blas95
   USE omp_lib

   IMPLICIT NONE
```

```
INTEGER :: Nd1, Nd2, ierr
COMPLEX(KIND=dp), ALLOCATABLE, DIMENSION(:,:) :: MATMULVG, Af, Bf

Nd1 = SIZE(Af,1)
Nd2 = SIZE(Bf,2)
ierr = 0

ALLOCATE(MATMULVG(Nd1,Nd2), STAT=ierr)

MATMULVG = (0.0_dp, 0.0_dp)
CALL GEMM3M(Af,Bf,MATMULVG)

!----------------------------------------------------------------
END FUNCTION MATMULVG
END MODULE PAB

MODULE PcAB
CONTAINS
FUNCTION MATMULVGc(Af,Bf)

USE Mod_Precision

USE mkl95_lapack
USE blas95
USE omp_lib

IMPLICIT NONE

INTEGER :: Nd1, Nd2, ierr
COMPLEX(KIND=dp), ALLOCATABLE, DIMENSION(:,:) :: MATMULVGc, Af, Bf

Nd1 = SIZE(Af,2)
Nd2 = SIZE(Bf,2)
ierr = 0

ALLOCATE(MATMULVGc(Nd1,Nd2), STAT=ierr)

MATMULVGc = (0.0_dp, 0.0_dp)
CALL GEMM3M(Af,Bf,MATMULVGc,'C','N')

!----------------------------------------------------------------
END FUNCTION MATMULVGc
END MODULE PcAB

MODULE PABz
CONTAINS
FUNCTION MATMULVGz(Af,Bf)

USE Mod_Precision
```

```
    USE mkl95_lapack
    USE blas95
    USE omp_lib

    IMPLICIT NONE

    INTEGER :: Nd1, Nd2, ierr
    COMPLEX(KIND=dp), ALLOCATABLE, DIMENSION(:,:) :: MATMULVGz, Af, Bf

    Nd1 = SIZE(Af,1)
    Nd2 = SIZE(Bf,1)
    ierr = 0

    ALLOCATE(MATMULVGz(Nd1,Nd2), STAT=ierr)

    MATMULVGz = (0.0_dp, 0.0_dp)
    CALL GEMM3M(Af,Bf,MATMULVGz,'N','C')

!--------------------------------------------------------------
    END FUNCTION MATMULVGz
    END MODULE PABz
```