

Dynamo equation - example of a parabolic PDE

1. Modeling and simplification of equations, non-dimensional variables, boundary conditions, free decay
2. Solution as eigenvalue problem, expansion in decay modes, matrix eigenvalue problem, critical dynamo number,

Exercise: DYNEM

3. Solution as initial value problem, discretisation, explicit scheme, stability condition, butterfly diagram,

Exercise: DYNEX

4. Implicit scheme, Exercise: DYNIM, various methods: Crank-Nicolson, Dufort-Frankel, advantages and disadvantages

5. Miscellaneous:

2D, nonlinear effects, advective equation

Dynamo equation

$$\frac{\partial \underline{B}}{\partial t} = \text{rot}(\underline{u} \times \underline{B}) + \text{rot}(\alpha \underline{B}) - \eta \text{rot} \text{rot} \underline{B} \quad \text{linear homogeneous}$$

Cartesian coordinates: $(x, y, z) \stackrel{!}{=} (\varrho, \varphi, r)$

"Axial symmetry": $\frac{\partial}{\partial y} = 0$

"Rotation": $\underline{u} = (0, u_y, 0)$ with shear $G_x = \frac{\partial u_y}{\partial x}$, $G_z = \frac{\partial u_y}{\partial z}$

Magnetic field: $\underline{B} = (B_x, B_y, B_z) = (0, B_y, 0) + \text{rot}(0, A_y, 0)$

toroidal poloidal

$$\frac{\partial B_y}{\partial t} = G_z \frac{\partial A_y}{\partial x} - G_x \frac{\partial A_y}{\partial z} - \left(\frac{\partial}{\partial x} (\alpha \frac{\partial A_y}{\partial x}) + \frac{\partial}{\partial z} (\alpha \frac{\partial A_y}{\partial z}) \right) + \eta \left(\frac{\partial^2 B_y}{\partial x^2} + \frac{\partial^2 B_y}{\partial z^2} \right)$$

$$\frac{\partial A_y}{\partial t} = \alpha B_y + \eta \left(\frac{\partial^2 A_y}{\partial x^2} + \frac{\partial^2 A_y}{\partial z^2} \right)$$

simplifying assumptions: $G_x = 0$, $G_z = G$

$\alpha \Omega$ approximation

A_y, B_y only function of x (not realistic)

omit index y

$$\frac{\partial B}{\partial t} = G \frac{\partial A}{\partial x} + \eta \frac{\partial^2 B}{\partial x^2}, \quad \frac{\partial A}{\partial t} = \alpha B + \eta \frac{\partial^2 A}{\partial x^2}$$

$$\frac{\partial B}{\partial t} = G \frac{\partial A}{\partial x} + \gamma \frac{\partial^2 B}{\partial x^2}, \quad \frac{\partial A}{\partial t} = \alpha B + \gamma \frac{\partial^2 A}{\partial x^2}$$

Non-dimensional variables:

$$B = B_0 \tilde{B}, \quad A = B_0 L \tilde{A}, \quad x = L \tilde{x}, \quad t = \frac{L^2}{\gamma_0} \tilde{t}$$

$$G = G_0 \tilde{G}, \quad \alpha = \alpha_0 \tilde{\alpha}, \quad \gamma = \gamma_0 \tilde{\gamma}$$

$$\frac{\partial \tilde{B}}{\partial \tilde{t}} = R_\Omega \tilde{G} \frac{\partial \tilde{A}}{\partial \tilde{x}} + \tilde{\gamma} \frac{\partial^2 \tilde{B}}{\partial \tilde{x}^2}, \quad \frac{\partial \tilde{A}}{\partial \tilde{t}} = R_\alpha \tilde{\alpha} \tilde{B} + \tilde{\gamma} \frac{\partial^2 \tilde{A}}{\partial \tilde{x}^2}$$

with $R_\Omega = \frac{G_0 L^2}{\gamma_0}$ and $R_\alpha = \frac{\alpha_0 L}{\gamma_0}$ Reynolds numbers

Assumption: $\tilde{G} = \sin x$, $\tilde{\alpha} = \cos x$, $\tilde{\gamma} = 1$

New variable: $\hat{\tilde{A}} = R_\Omega \tilde{A}$

omit all \sim

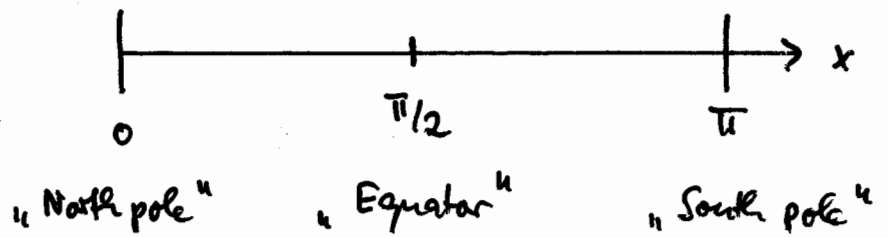
$$\frac{\partial B}{\partial t} = \sin x \frac{\partial A}{\partial x} + \frac{\partial^2 B}{\partial x^2}, \quad \frac{\partial A}{\partial t} = P \cos x B + \frac{\partial^2 A}{\partial x^2}$$

with dynamo number $P = R_\alpha R_\Omega$

in principle both equations combinable in one equation of higher order for A or B

Boundary conditions

$$L = \frac{\pi}{2}$$



$$x = 0 : A = B = 0$$

$$x = \pi : A = B = 0$$

$$x = \frac{\pi}{2} : \text{antisymmetric solution ("Dipole") : } \frac{\partial A}{\partial x} = B = 0$$

$$\text{symmetric solution ("Quadrupole") : } A = \frac{\partial B}{\partial x} = 0$$

In what follows: only antisymmetric solution

$$\text{Free decay : } \frac{\partial B}{\partial t} = \frac{\partial^2 B}{\partial x^2}, \quad \frac{\partial A}{\partial t} = \frac{\partial^2 A}{\partial x^2}$$

$$B_n = e^{\omega_n t} \sin nx \quad \text{with } \omega_n = -n^2, \quad n = 2, 4, 6, \dots$$

$$A_n = e^{\omega_n t} \sin nx \quad \text{with } \omega_n = -n^2, \quad n = 1, 3, 5, \dots$$

Solution as eigenvalue problem

Discretisation or better: Expansion in decay modes
(or any other complete, orthogonal system of functions
which already satisfies the boundary conditions)

$$B = e^{\omega t} \sum_{n=2,4,6,\dots} b_n \sin nx, \quad A = e^{\omega t} \sum_{n=1,3,5,\dots} a_n \sin nx$$

$$\omega \sum_{n=2,4,\dots} b_n \sin nx = \sin x \sum_{n=1,3,\dots} a_n n \cos nx - \sum_{n=2,4,\dots} b_n n^2 \sin nx$$

$$\omega \sum_{n=1,3,\dots} a_n \sin nx = \rho \cos x \sum_{n=2,4,\dots} b_n \sin nx - \sum_{n=1,3,\dots} a_n n^2 \sin nx$$

$$\sin x \cos nx = \frac{1}{2} (\sin(n+1)x - \sin(n-1)x)$$

$$\cos x \sin nx = \frac{1}{2} (\sin(n+1)x + \sin(n-1)x)$$

Orthogonality relations:

$$\int_0^{\pi/2} \sin nx \sin mx \, dx = \frac{\pi}{4} \delta_{nm} \quad \text{for } n, m \text{ both odd or both even}$$

$$\int_0^{\pi/2} \sin mx, \quad \frac{4}{\pi} \int_0^{\pi/2} dx, \quad m \text{ even}$$

$$\int_0^{\pi/2} \sin mx, \quad \frac{4}{\pi} \int_0^{\pi/2} dx, \quad m \text{ odd}$$

Alternative solution of eigenvalue problem:

$$\det (K_{ij} - \omega \delta_{ij}) = 0$$

- for a given P
 - estimate ω , compute $\det (\neq 0)$
 - iterate ω until $\det = 0$ (Regula Falsi, Newton-Raphson, ...)
- vary P until $\omega_R = 0 \approx P_{crit}$

```

program dynewp_core
parameter (nd=50,nnd=nd+nd)
implicit real*8 (a-h,o-z)
dimension gmat (nnd,nnd), ewr (nnd), ewi (nnd), ev (1,1), ivl (nnd),
          fv1 (nnd), ind (nnd), gmat1 (nnd,nnd)

```

} declaration of var

```

print *, 'Input: n / number of expansion coefficients'
read *, n
if (n.gt.nd) stop 'n>nd'
nn=n+.

```

} input parameters

```

do i=1,nn
  do j=1,nn
    gmat(i,j)=0.d0
  enddo
enddo

```

} initialisation of matrix

```

do i=1,nn
  ri=dfloat(i)
  gmat(i,i)=-ri*ri
enddo

```

} matrix elements

```

do i=2,nn-2,2
  gmat(i,i-1)=0.5d0*dfloat(i-1)
  gmat(i,i+1)=-0.5d0*dfloat(i+1)
enddo
gmat(nn,nn-1)=0.5d0*dfloat(nn-1)

```

```

10 print *, 'Input: pa,pe,dp,iflag / stops if iflag<0 / ',
      'dynamo numbers'
read *, pa,pe,dp,iflag
if (iflag.lt.0) stop 'normal stop'

```

} loop over
dynamo numbers

```

do p=pa,pe,dp

```

```

  ph=0.5d0*p

```

```

  do i=3,nn-1,2
    gmat(i,i-1)=ph
    gmat(i,i+1)=ph
  enddo
  gmat(1,2)=ph

```

} P-dep. matrix elements

```

  do i=1,nn
    do j=1,nn
      gmat1(i,j)=gmat(i,j)
    enddo
  enddo

```

} copy of matrix

```

  call rg (nnd,nn,gmat1,ewr,ewi,0,ev,ivl,fv1,ierr)
  if (ierr.ne.0) stop 'ierr ne 0 in rg'

```

} computation of eigen-
values (EISPACK)

```

  call indexd (nn,ewr,ind)

```

← ordering of eigenvalues
(RECIPES)

```

  print *
  print 1000, p, ewr(ind(nn)), ewi(ind(nn))
  do i=1,4
    print 1010, ewr(ind(nn-i)), ewi(ind(nn-i))
  enddo

```

} output

```

1000 format (f20.10,2d20.10)
1010 format (20x,2d20.10)

```

```

enddo

```

end of loop

```

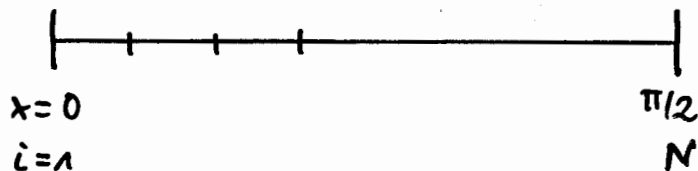
goto 10
end

```

Initial value problem: explicit, Δx^2 , Euler, Δt

$$\frac{\partial A}{\partial t} = P \cos x B + \frac{\partial^2 A}{\partial x^2}, \quad \frac{\partial B}{\partial t} = \sin x \frac{\partial A}{\partial x} + \frac{\partial^2 B}{\partial x^2}$$

$$f(x_i, t^n) = f_i^n$$



$$\frac{\partial f_i^n}{\partial t} = \frac{f_i^{n+1} - f_i^n}{\Delta t}$$

$$\frac{\partial f_i^n}{\partial x} = \frac{f_{i+1}^n - f_{i-1}^n}{2\Delta x}, \quad \frac{\partial^2 f_i^n}{\partial x^2} = \frac{f_{i+1}^n - 2f_i^n + f_{i-1}^n}{\Delta x^2}$$

FTCS

at the boundary: $\frac{\partial f_N^n}{\partial x} = \frac{3f_N^n - 4f_{N-1}^n + f_{N-2}^n}{2\Delta x}$

derived by Taylor expansions

$$\frac{A_i^{n+1} - A_i^n}{\Delta t} = P \cos_i B_i^n + \frac{A_{i+1}^n - 2A_i^n + A_{i-1}^n}{\Delta x^2} \equiv DA_i^n \quad i=2, N-1$$

$$\frac{B_i^{n+1} - B_i^n}{\Delta t} = \sin_i \frac{A_{i+1}^n - A_{i-1}^n}{2\Delta x} + \frac{B_{i+1}^n - 2B_i^n + B_{i-1}^n}{\Delta x^2} \equiv DB_i^n \quad i=2, N-1$$

$$A_i^{n+1} = \Delta t (DA_i^n) + A_i^n \quad i=2, N-1$$

$$B_i^{n+1} = \Delta t (DB_i^n) + B_i^n$$

Boundary values: $A_1^n = 0, B_1^n = 0$

$$A_N^n = \frac{4}{3} A_{N-1}^n - \frac{1}{3} A_{N-2}^n, \quad B_N^n = 0$$

Stability condition: $\Delta t \leq \frac{1}{2} \Delta x^2$

Stability for $\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2}$

Amplification factor: $\varepsilon^{n+1} = g \varepsilon^n$, condition: $|g| \leq 1$

Fourier mode $u = \hat{u}(t) e^{ikx}$

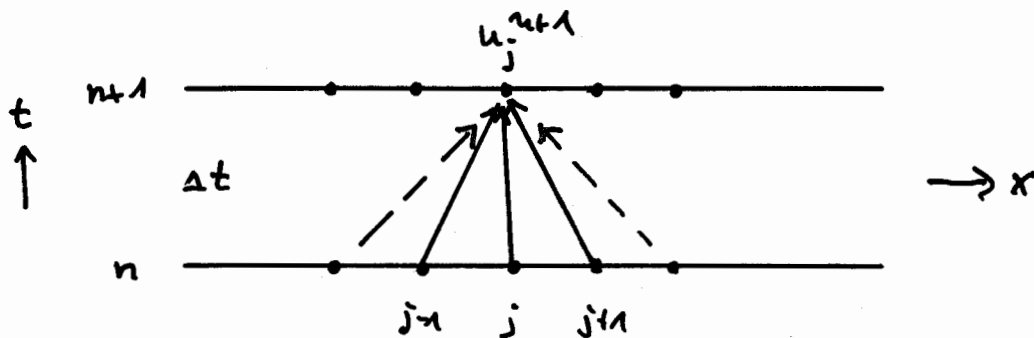
$$\hat{u}^{n+1} e^{ikx_j} = \hat{u}^n e^{ikx_j} + \frac{\Delta t}{\Delta x^2} \hat{u}^n (e^{ikx_{j+1}} - 2e^{ikx_j} + e^{ikx_{j-1}})$$

$$\hat{u}^{n+1} = \hat{u}^n \left(1 + \frac{2\Delta t}{\Delta x^2} \left(\frac{1}{2} e^{ik\Delta x} + \frac{1}{2} e^{-ik\Delta x} - 1 \right) \right)$$

$\underbrace{\hspace{10em}}_{\cos k\Delta x}$

$$g(\Delta t, \Delta x, k) = 1 + \frac{2\Delta t}{\Delta x^2} (\cos k\Delta x - 1) = 1 - \frac{4\Delta t}{\Delta x^2} \sin^2 \frac{k\Delta x}{2}$$

$$\left| 1 - \frac{4\Delta t}{\Delta x^2} \sin^2 \frac{k\Delta x}{2} \right| \leq 1 \quad \forall k, \quad -\frac{4\Delta t}{\Delta x^2} \geq -2, \quad \Delta t \leq \frac{1}{2} \Delta x^2$$



Conditions on algorithm:

Consistency, exactness, stability, efficiency

```

program dynex_core
implicit real*8 (a-h,o-z)
implicit integer (i-n)
parameter (nxd=201,ntd=500)
real*4 bf(ntd,nxd)
dimension x(nxd),sinx(nxd),cosx(nxd),a(nxd),b(nxd),
          anew(nxd),bnew(nxd)

```

} declaration of
variables

```

read *, nx,dtim,ntim,rp
read *, rauf,quen

```

} input parameters

```

if (nx.gt.nxd) stop 'nx>nxd'

```

```

nx1=nx-1
pih=dasin(1.d0)
dx=pih/dfloat(nx1)
dx1=1.d0/(dx*dx)
dx2=1.d0/(2.d0*dx)

```

} auxiliary variables

```

do i=1,nx
  x(i)=dfloat(i-1)*dx
  sinx(i)=dsin(x(i))
  cosx(i)=dcos(x(i))
enddo

```

} grid

```

do i=1,nx
  b(i)=dsin(2.d0*x(i))
  a(i)=0.d0
enddo

```

} initial fields

```

do k=1,ntim

```

begin of time loop

```

  Output of k, a, b

```

← Output of fields

```

do i=2,nx-1
  ip1=i+1
  im1=i-1
  da=cosx(i)*b(i)/(1.d0+quen*b(i)*b(i))
    +(a(ip1)-2.d0*a(i)+a(im1))*dx1
  db=rp*sinx(i)*(a(ip1)-a(im1))*dx2
    +(b(ip1)-2.d0*b(i)+b(im1))*dx1
    -rauf*dsign(1.d0,b(i))*b(i)*b(i)
  anew(i)=dtim*da+a(i)
  bnew(i)=dtim*db+b(i)
enddo

```

} rhs

```

do i=2,nx-1
  a(i)=anew(i)
  b(i)=bnew(i)
enddo

```

} new variables

```

a(1)=0.d0
b(1)=0.d0
a(nx)=(4.d0*a(nx-1)-a(nx-2))/3.d0
b(nx)=0.d0

```

} boundary values

```

enddo

```

end of time loop

```

stop
end

```

Implicit treatment

$$\frac{A_i^{n+1} - A_i^n}{\Delta t} = P \cos_i B_i^n + \frac{A_{i+1}^{n+1} - 2A_i^{n+1} + A_{i-1}^{n+1}}{\Delta x^2}$$

$$\frac{B_i^{n+1} - B_i^n}{\Delta t} = \alpha \mu_i \frac{A_{i+1}^n - A_{i-1}^n}{2\Delta x} + \frac{B_{i+1}^{n+1} - 2B_i^{n+1} + B_{i-1}^{n+1}}{\Delta x^2}$$

$$-\frac{\Delta t}{\Delta x^2} A_{i-1}^{n+1} + \left(1 + \frac{2\Delta t}{\Delta x^2}\right) A_i^{n+1} - \frac{\Delta t}{\Delta x^2} A_{i+1}^{n+1} = A_i^n + \Delta t P \cos_i B_i^n$$

$$-\frac{\Delta t}{\Delta x^2} B_{i-1}^{n+1} + \left(1 + \frac{2\Delta t}{\Delta x^2}\right) B_i^{n+1} - \frac{\Delta t}{\Delta x^2} B_{i+1}^{n+1} = B_i^n + \frac{\Delta t}{2\Delta x} \alpha \mu_i (A_{i+1}^n - A_{i-1}^n)$$

$$\begin{pmatrix} \vdots & & & & & & \\ & \ddots & & & & & \\ & & -\frac{\Delta t}{\Delta x^2} & & & & \\ & & & \left(1 + \frac{2\Delta t}{\Delta x^2}\right) & & & \\ & & & & -\frac{\Delta t}{\Delta x^2} & & \\ & & & & & \ddots & \\ & & & & & & A_{N-1}^{n+1} \end{pmatrix} = A_i^n + \Delta t P \cos_i B_i^n$$

Boundary conditions: $A_1 = 0$, $A_N = \frac{4}{3} A_{N-1} - \frac{1}{3} A_{N-2}$

$$-\frac{\Delta t}{\Delta x^2} A_{N-2} + \left(1 + \frac{2\Delta t}{\Delta x^2}\right) A_{N-1} - \frac{\Delta t}{\Delta x^2} \left(\frac{4}{3} A_{N-1} - \frac{1}{3} A_{N-2}\right) =$$

$$= -\frac{\Delta t}{\Delta x^2} \left(1 - \frac{1}{3}\right) A_{N-2} + \left(1 + \frac{\Delta t}{\Delta x^2} \left(2 - \frac{4}{3}\right)\right) A_{N-1}$$

Equation for B analogous,

boundary conditions: $B_1 = 0$, $B_N = 0$

```

program dynim_core
implicit real*8 (a-h,o-z)
parameter (nxd=201,ntd=50,nttd=500)
dimension x(nxd),sinx(nxd),cosx(nxd),a(nxd),b(nxd)
      ,at(nxd-2),bt(nxd-2),ct(nxd-2),dta(nxd-2)
      ,dtb(nxd-2),xa(nxd-2),xb(nxd-2)

```

} declaration of variables

```

read *, dtim,ntim,rp
read *, rauf,quen
if (nx.gt.nxd) stop 'nx>nxd'

```

} input of parameters

```

nx1=nx-1
nx2=nx-2
pih=dasin(1.d0)
pi=2.d0*pih
dx=pih/dfloat(nx1)
dx1=1.d0/(dx*dx)
dx2=dtim/(2.d0*dx)
dx3=-dtim*dx1
dx4=1.+2.d0*dtim*dx1
do i=1,nx
  x(i)=dfloat(i-1)*dx
  sinx(i)=dsin(x(i))
  cosx(i)=dcos(x(i))
enddo

```

} auxiliary variables and grid

```

do i=1,nx2
  at(i)=dx3
  bt(i)=dx4
  ct(i)=dx3
enddo
atn=at(nx2)
btn=bt(nx2)
atx=at(nx2)-ct(nx2)/3.d0
btx=bt(nx2)+4.d0*ct(nx2)/3.d0

```

} matrix elements

} boundary conditions

```

do i=1,nx
  a(i)=0.d0
  b(i)=amp*dsin(2.d0*x(i))
enddo

```

} initial fields

```

do k=1,ntim

```

← begin of time loop

c Output of variables

← Output of fields

```

do i=1,nx2
  i1=i+1
  dta(i)=a(i1)+dtim*cosx(i1)*b(i1)/(1.d0+quen*b(i1)*b(i1))
  dtb(i)=b(i1)+dx2*rp*sinx(i1)*(a(i1+1)-a(i1-1))
      -dtim*rauf*dsign(1.d0,b(i1))*b(i1)*b(i1)
enddo

```

} this

```

at(nx2)=atx
bt(nx2)=btx
call tridag (at,bt,ct,dta,xa,nx2)
at(nx2)=atn
bt(nx2)=btn
call tridag (at,bt,ct,dtb,xb,nx2)
do i=1,nx2
  i1=i+1
  a(i1)=xa(i)
  b(i1)=xb(i)
enddo
a(1)=0.d0
b(1)=0.d0
a(nx)=(4.d0*a(nx1)-a(nx2))/3.d0
b(nx)=0.d0

```

} solution of tridiagonal system (RECIPES)

} new variables

} boundary values

```

enddo
stop
end

```

← end of time loop

Various algorithms for $\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2}$

$$(\delta^2 u)_i^n = u_{i+1}^n - 2u_i^n + u_{i-1}^n$$

1) Explicit: $u_i^{n+1} - u_i^n = \frac{(\delta^2 u)_i^n}{\Delta x^2}$, $O(\Delta t, \Delta x^2)$, $\Delta t \leq \frac{1}{2} \Delta x^2$

2) Implicit: $u_i^{n+1} - u_i^n = \frac{(\delta^2 u)_i^{n+1}}{\Delta x^2}$, $O(\Delta t, \Delta x^2)$, always stable

requires solution of system of linear equations

3) Crank-Nicholson: $u_i^{n+1} - u_i^n = \frac{(\delta^2 u)_i^{n+1} + (\delta^2 u)_i^n}{2\Delta x^2}$,

$O(\Delta t^2, \Delta x^2)$, always stable

4) DuFort-Frankel: $\frac{u_i^{n+1} - u_i^{n-1}}{2\Delta t} = \frac{u_{i+1}^n - u_i^{n+1} - u_i^{n-1} + u_{i-1}^n}{\Delta x^2}$

$O(\Delta t^2, \Delta x^2, (\frac{\Delta t}{\Delta x})^2)$, explicit, always stable

5) Leapfrog: $\frac{u_i^{n+1} - u_i^{n-1}}{2\Delta t} = \frac{(\delta^2 u)_i^n}{\Delta x^2}$, always unstable

Two-dimensional diffusion equation implicit

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}, \quad u(x_i, y_j, t^n) = u_{ij}^n$$

$$\frac{u_{ij}^{n+1} - u_{ij}^n}{\Delta t} = \frac{u_{i+1,j}^{n+1} - 2u_{ij}^{n+1} + u_{i-1,j}^{n+1}}{\Delta x^2} + \frac{u_{i,j+1}^{n+1} - 2u_{ij}^{n+1} + u_{i,j-1}^{n+1}}{\Delta y^2}$$

$$\begin{pmatrix} \text{///} & / & 0 & 0 & 0 \\ / & \text{///} & / & 0 & 0 \\ 0 & / & \text{///} & / & 0 \\ 0 & 0 & / & \text{///} & / \\ 0 & 0 & 0 & / & \text{///} \end{pmatrix} \begin{pmatrix} u_{2j} \\ u_{3j} \\ u_{4j} \\ \vdots \end{pmatrix}^{n+1} = \text{rhs}$$

$i = 1, N \quad j = 1, M \quad \longrightarrow \quad (N \cdot M)^2$ sparse block tridiagonal

Alternate Direction Implicit ADI

$$\frac{u_{ij}^{n+1/2} - u_{ij}^n}{\frac{1}{2}\Delta t} = \frac{u_{i+1,j}^{n+1/2} - 2u_{ij}^{n+1/2} + u_{i-1,j}^{n+1/2}}{\Delta x^2} + \frac{u_{i,j+1}^n - 2u_{ij}^n + u_{i,j-1}^n}{\Delta y^2}$$

implicit explicit

$$\frac{u_{ij}^{n+1} - u_{ij}^{n+1/2}}{\frac{1}{2}\Delta t} = \frac{u_{i+1,j}^{n+1/2} - 2u_{ij}^{n+1/2} + u_{i-1,j}^{n+1/2}}{\Delta x^2} + \frac{u_{i,j+1}^{n+1} - 2u_{ij}^{n+1} + u_{i,j-1}^{n+1}}{\Delta y^2}$$

explicit implicit

$M \cdot (N\text{-tridiagonal}) + N \cdot (M\text{-tridiagonal})$

Nonlinear dynamo

- α -quenching : $\alpha \longrightarrow \frac{\alpha}{1 + \beta^2}$
- magnetic buoyancy : additional term in β -equation:
 - $\text{sign}(\beta) \beta^2$
 - limit amplitude growth when supercritical
 - coupling of modes
 - mixed parity possible
 - dependent on initial field

Transport equation

$$\frac{\partial u}{\partial t} + v \frac{\partial u}{\partial x} = 0, \quad \text{general} \quad \frac{\partial u}{\partial t} + \frac{\partial f(u)}{\partial x} = 0 \quad \text{hyperbolic}$$

- FTCS always unstable

- Lax-Wendroff:
$$\frac{u_i^{n+1} - \frac{1}{2}(u_{i+1}^n + u_{i-1}^n)}{\Delta t} = -v \frac{u_{i+1}^n - u_{i-1}^n}{2\Delta x}$$

$$\text{stable for } \Delta t \leq \frac{\Delta x}{v}, \quad \text{CFL}$$

- leapfrog:
$$\frac{u_i^{n+1} - u_i^{n-1}}{2\Delta t} = -v \frac{u_{i+1}^n - u_{i-1}^n}{2\Delta x}, \quad \Delta t \leq \frac{\Delta x}{v}$$

- Upwind:
$$u_i \frac{u_i^{n+1} - u_i^n}{\Delta t} = -v \begin{cases} \frac{u_i^n - u_{i-1}^n}{\Delta x} & \text{for } v > 0 \\ \frac{u_{i+1}^n - u_i^n}{\Delta x} & \text{for } v < 0 \end{cases}$$

$$\text{stable for } \Delta t \leq \frac{\Delta x}{v}$$

⋮

Dynamo Equation – Exercises

D. Schmitt

DYNEW/P

For modes antisymmetric with respect to the equator, compute growth rate and frequency $\omega(P)$ for dynamo numbers $P < 0$ and $P > 0$ and display $\omega(P)$ diagrams in the complex ω -plane (rough hand drawing).

For the first excited oscillatory mode with $P < 0$, determine the critical dynamo number up to four significant digits and its frequency or period.

Check convergence of the frequency with respect to the number of expansion coefficients.

Display the magnetic field as a function of latitude and time for various critical modes and learn about the butterfly diagram of fundamental mode and overtones for positive and negative dynamo numbers.

Adapt the code for symmetric modes and repeat.

DYNEX

For the antisymmetric mode at $P = -102$, check stability and convergence for various spatial grids (e.g. $n_x = 11, 26, 51, 101, \dots$) and time increments. Compute at least 5 diffusion times and output only after the first diffusion time as it is spoiled by initial evolution.

DYNIM

As for DYNEX, check stability and convergence. Estimate the computing resources for DYNIM compared to DYNEX at similar accuracy for various grid spacings.

Optional

Adapt the code DYNEX for the Dufort-Frankel scheme and compare stability and convergence with the results from DYNEX and DYNIM.

Dynamics equations - exercises

- pepublic / schmitt / dyn.tar
- www.linmpi.mpg.de / ~schmitt / dyn.tar
- tar xvf dyn.tar \leadsto dynew, dynet, dynim
- compile and link, make

dynewp: eigenvalues

n		rg
pa, pe, dp, iflag - loop		indexd

dynew: eigenvalue and eigenvector for one mode

dynew.ps or butt.pro \leadsto butt.ps

dynet:

nbn, nx, dtim, ntim, rp		dynet < imex
tauf, quen		dynet.ps
iwr, nstwr, nendwr, nwr		butt.pro

dynim:

nbn, nsw, nantf, amp, nx, dtim, ntim, rp
tauf, quen

idr, nstdr, nendd, ndr, iwr, nstwr, nendwr, nwr
iim, nstim, nendi, nim, ipl, nstpl, nendp, npl