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 numbers,
   Exercise: DYNEW

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

<u>Dynamo equation</u>

$$\frac{\partial \underline{B}}{\partial t} = \mathrm{rot}\,(\underline{u} \times \underline{B}) + \mathrm{rot}\,(\alpha \underline{B}) - \eta\, \mathrm{rot}\,\mathrm{rot}\,\underline{B}$$ linear homogenous

Cartesian coordinates: $(x, y, z) \hat{=} (\vartheta, \varphi, r)$

„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: $\underline{B} = (B_x, B_y, B_z) = \underset{\text{toroidal}}{(0, B_y, 0)} + \underset{\text{poloidal}}{\mathrm{rot}\,(0, A_y, 0)}$

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

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

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

$\alpha\Omega$ approximation

A_y, B_y only function of x (not realistic)

omit index y

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

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

Non-dimensional variables:

$$B = B_0 \tilde{B} \quad , \quad A = B_0 L \tilde{A} \quad , \quad x = L \tilde{x} \quad , \quad t = \frac{L^2}{\eta_0} \tilde{t}$$

$$G = G_0 \tilde{G} \quad , \quad \alpha = \alpha_0 \tilde{\alpha} \quad , \quad \eta = \eta_0 \tilde{\eta}$$

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

with $R_\Omega = \dfrac{G_0 L^2}{\eta_0}$ and $R_\alpha = \dfrac{\alpha_0 L}{\eta_0}$ Reynolds numbers

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

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

omit all ~

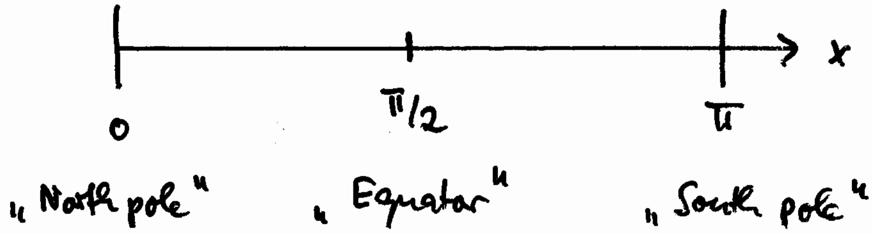
$$\frac{\partial B}{\partial t} = \sin x \frac{\partial A}{\partial x} + \frac{\partial^2 B}{\partial x^2} \quad , \quad \frac{\partial A}{\partial t} = P \cos x \, B + \frac{\partial^2 A}{\partial x^2}$$

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

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



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

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

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

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

In what follows: only antisymmetric solution

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

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

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

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)

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

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

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

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

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

Orthogonality relations:

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

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

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

$$\omega a_m = \frac{P}{2}\left(b_{m-1} + b_{m+1}\right) - m^2 a_m \,, \qquad m \text{ odd}$$

$$\omega b_m = \frac{1}{2}\left((m-1)\,a_{m-1} - (m+1)\,a_{m+1}\right) - m^2 b_m \,, \qquad m \text{ even}$$

$$\omega \begin{pmatrix} a_1 \\ b_2 \\ a_3 \\ b_4 \\ \vdots \\ a_{n-1} \\ b_n \end{pmatrix} = \begin{pmatrix} -1 & P/2 & & & & \\ 1/2 & -4 & -3/2 & & & \\ & P/2 & -9 & P/2 & & \\ & & & \ddots & & \\ & & & P/2 & -(n-1)^2 & P/2 \\ & & & & (n-1)/2 & -n^2 \end{pmatrix} \begin{pmatrix} a_1 \\ b_2 \\ a_3 \\ b_4 \\ \vdots \\ a_{n-1} \\ b_n \end{pmatrix}$$

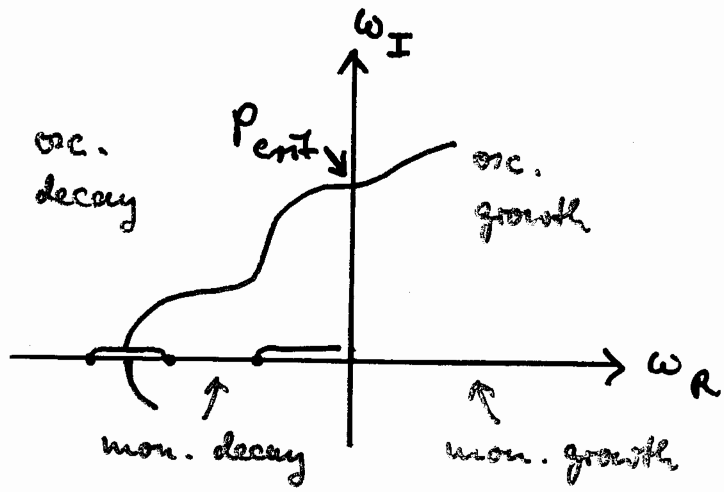
Matrix eigenvalue problem: DYNEW

Exercises:

- $\omega(P)$ diagrams

  for $P > 0$ and

  $P < 0$



- critical dynamo numbers : $\omega_R(P_{crit}) = 0$
- convergence
- display $B(x,t)$ for various cases
- same for quadrupolar solutions

Alternative solution of eigenvalue problem:

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

- for a given P
  - estimate ω, compute det ($\neq 0$)
  - iterate ω until det $= 0$ (Regula Falsi, Newton-Raphson, ...)

- vary P until $\omega_R = 0$ $\rightsquigarrow$ 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),iv1(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+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

      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)
```

matrix elements

```
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'

      do p=pa,pe,dp
```

loop over dynamo numbers

```
         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,iv1,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
1000     format (f20.10,2d20.10)
1010     format (20x,2d20.10)
```

output

```
      enddo
```

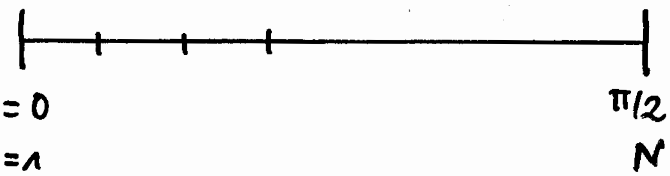
end of loop

```
      goto 10
      end
```

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

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

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



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

FTCS

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

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

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

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

$$A_i^{n+1} = \Delta t\,(DA_i^n) + A_i^n$$

$$i = 2,\ N-1$$

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

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

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

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

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

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

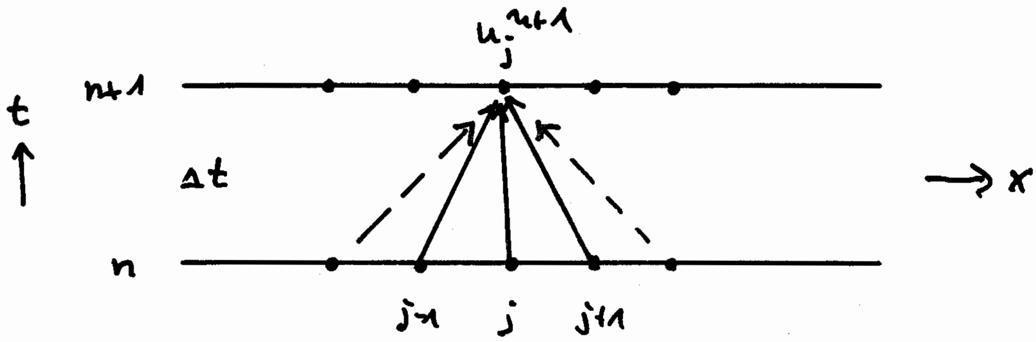
Fourier mode $u = \hat{u}(t)\, e^{ikx}$

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

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

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

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



Conditions on algorithm:

Consistency, exactness, stability, efficiency

```
      program dynex_core
      implicit real*8 (a-h,o-z)                                  declaration of
      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),        variables
     .          anew(nxd),bnew(nxd)

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

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

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

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


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

      do k=1,ntim                                                begin of time loop

c        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                    rhs
            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
         do i=2,nx-1                                             new variables
            a(i)=anew(i)
            b(i)=bnew(i)
         enddo
         a(1)=0.d0
         b(1)=0.d0                                               boundary values
         a(nx)=(4.d0*a(nx-1)-a(nx-2))/3.d0
         b(nx)=0.d0

      enddo                                                      end of time loop

      stop
      end
```

Implicit treatment

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

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

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

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

$$\begin{pmatrix} \ddots & & \\ -\frac{\Delta t}{\Delta x^2} & \left(1 + \frac{2\Delta t}{\Delta x^2}\right) & -\frac{\Delta t}{\Delta x^2} \\ & & \ddots \end{pmatrix} \begin{pmatrix} A_2 \\ \vdots \\ A_{i-1} \\ A_i \\ A_{i+1} \\ \vdots \\ A_{N-1} \end{pmatrix}^{n+1} = A_i^n + \Delta t\, P \cos_i B_i^n$$

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)                        declaration of
      dimension x(nxd),sinx(nxd),cosx(nxd),a(nxd),b(nxd)
     .          ,at(nxd-2),bt(nxd-2),ct(nxd-2),dta(nxd-2)       variables
     .          ,dtb(nxd-2),xa(nxd-2),xb(nxd-2)

      read *, nx,amp,dtim,ntim,rp                                input of parameters
      read *, rauf,quen
      if (nx.gt.nxd) stop 'nx>nxd'
      nx1=nx-1
      nx2=nx-2
      pih=dasin(1.d0)                                            auxilliary variables
      pi=2.d0*pih
      dx=pih/dfloat(nx1)
      dx1=1.d0/(dx*dx)                                           and
      dx2=dtim/(2.d0*dx)
      dx3=-dtim*dx1
      dx4=1.+2.d0*dtim*dx1                                       grid
      do i=1,nx
         x(i)=dfloat(i-1)*dx
         sinx(i)=dsin(x(i))
         cosx(i)=dcos(x(i))
      enddo

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

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

      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))       rhs
     .             -dtim*rauf*dsign(1.d0,b(i1))*b(i1)*b(i1)
         enddo

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

      enddo                                                      end of time loop
      stop
      end
```

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

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

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

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

requires solution of system of linear equations

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

$O(\Delta t^2, \Delta x^2)$, alway 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, alway 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

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

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

$$\left(\begin{array}{c|c|c|c|c}
\diagdown\!\diagdown\!\diagdown & \diagdown & 0 & 0 & 0 \\ \hline
\diagdown & \diagdown\!\diagdown\!\diagdown & \diagdown & 0 & 0 \\ \hline
0 & \diagdown & \diagdown\!\diagdown\!\diagdown & \diagdown & 0 \\ \hline
0 & 0 & \diagdown & \diagdown\!\diagdown\!\diagdown & \diagdown \\ \hline
0 & 0 & 0 & \diagdown & \diagdown\!\diagdown\!\diagdown
\end{array}\right)
\begin{pmatrix} u_{2j} \\ u_{3j} \\ u_{4j} \\ \vdots \end{pmatrix}^{n+1} = \text{rhs}$$

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

Alternate Direction Implicit ADI

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

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

$M \cdot$ (N-tridiagonal) + $N \cdot$ (M-tridiagonal)

Nonlinear dynamo

- α-quenching : $\alpha \longrightarrow \dfrac{\alpha}{1+B^2}$
- magnetic buoyancy : additional term in B-equation:

$$-\operatorname{sign}(B)\, B^2$$

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

  stable for $\Delta t \leq \frac{\Delta x}{v}$, CFL
- leapfrog : $\dfrac{u_i^{n+1} - u_i^{n-1}}{2\Delta t} = -v \dfrac{u_{i+1}^n - u_{i-1}^n}{2\Delta x}$, $\Delta t \leq \frac{\Delta x}{v}$
- Upwind: $\dfrac{u_i^{n+1} - u_i^n}{\Delta t} = -v \begin{cases} \dfrac{u_i^n - u_{i-1}^n}{\Delta x} & \text{for } v > 0 \\ \dfrac{u_{i+1}^n - u_i^n}{\Delta x} & \text{for } v < 0 \end{cases}$

  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. nx = 11, 26, 51, 101, ...) 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 recources 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.

Dynamo equation – exercises

- pcpublic / schmitt / dyn.tar
- www.linmpi.mpg.de /~schmitt / dyn.tar
- tar xvf dyn.tar ⇝ dynew, dynex, dynim
- compile and link, make

dynewp: eigenvalues

n	rg
pa, pe, dp, iflag – loop	inderd

dynew: eigenvalue and eigenvector for one mode

dynew.ps or butt.pro ⇝ butt.ps

dynex:

nbn, nx, dtim, ntim, rp	dynex < inex
ranf, quen	dynex.ps
iwr, nstwr, nendwr, nwr	butt.pro

dynim:

nbn, nsw, nanf, amp, nx, dtim, ntim, rp

ranf, quen

idr, nstdr, nendd, ndr, iwr, nstwr, nendw, nwr

iim, nstim, nendi, nim, ipl, nstpl, nendp, npl