Parabolic equations

A. Derivation of diffusion equation. Heat diffuses from hot to cold, down gradient

Recall divergence (Gauss') theorem: $\int \nabla\cdot\mathbf{F}dV = \int \mathbf{n}\cdot\mathbf{F}\,dS$ **n** is outward normal

e.g. $\int_a^b d_x F dx = F(b)-F(a)$ because **n**= 1 at b and -1 at a.

Fick's law of diffusion $\boldsymbol{F} = -\kappa\rho C_v \nabla T$

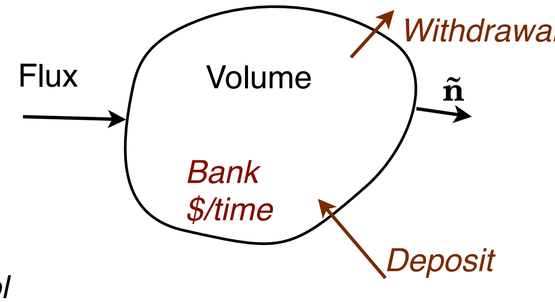
$$-\int \boldsymbol{F}\cdot\hat{n}dS = \int \rho C_v \partial_t T dV$$

$$-\int \nabla\cdot\boldsymbol{F}dV = \int \rho C_v \partial_t T dV$$

$$\int \nabla\cdot\kappa\nabla T dV = \int \partial_t T dV \qquad \text{<- Finite vol}$$

$$\Rightarrow \nabla\cdot\kappa\nabla T = \partial_t T \qquad \text{<-Finite diff}$$

General idea of conservation law



$$\boxed{\nabla\kappa\nabla T = \partial_t T}$$

$$1-D\ :\quad \partial_x(\kappa\partial_x T) = \partial_t T \ \text{ or, for constant } \kappa\text{:}\ \ \kappa\partial_x^2 T = \partial_t T$$

Grid in x : •← Δx →• • • •

1 2 ...... i I

March each point in time ---- I-2 o.d.e.'s (+2 b.c.'s)

B. Semi-discrete equations. Let κ be constant Second order central in x:

$$\frac{dT_i}{dt} = \kappa\delta_x^2 T = \frac{\kappa}{\Delta x^2}(T_{i+1} - 2T_i + T_{i-1})$$

i=2,3.. I-1. System of I-2 o.d.e.'s (+2 b.c.'s)

C. Can use any o.d.e. solver for system of equations --- just need RHS. Simplest: Euler explicit.

Solve at *i=2,3,4 ... I-1;* T_1 and T_I from boundary conditions. Marching solution for temperature evolution.

Variable k, something to think about

$$\frac{dT_i}{dt} = \delta_x(\kappa\delta_x T) = \frac{1}{\Delta x^2}(\kappa_{i+1/2}[T_{i+1} - T_i] - \kappa_{i-1/2}[T_i - T_{i-1}])$$

Variable Δx, something to think about

$$\frac{dT_i}{dt} = \delta_x(\kappa\delta_x T) = \frac{2}{x_{i+1} - x_{i-1}}\left(\kappa_{i+1/2}\frac{T_{i+1} - T_i}{x_{i+1} - x_i} - \kappa_{i-1/2}\frac{T_i - T_{i-1}}{x_i - x_{i-1}}\right)$$

Euler Explicit: $\delta_t T = \kappa\delta^2_x T$:

$$\frac{T_j^{n+1} - T_j^n}{\Delta t} = \frac{\kappa}{\Delta x^2}\left(T_{j+1}^n - 2T_j^n + T_{j-1}^n\right)$$

Let $\alpha = \kappa\Delta t/\Delta x^2$ only (non-dimensional) parameter.

$$T_j^{n+1} - T_j^n = \alpha\left(T_{j+1}^n - 2T_j^n + T_{j-1}^n\right)$$

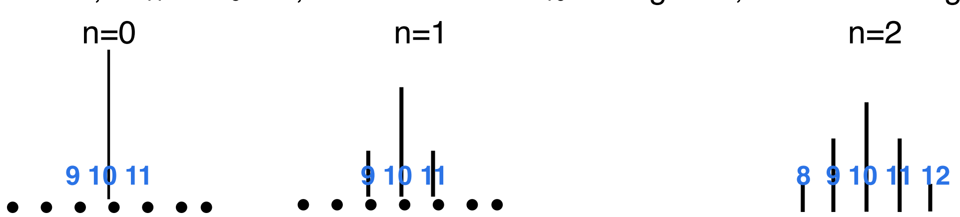
$$T_j^{n+1} = (1 - 2\alpha)T_j^n + \alpha\left(T_{j+1}^n + T_{j-1}^n\right)$$

Euler explicit code T^0 = initial distribution T(x). B.c. $T_1^n = Tw_0^n$, $T_J^n = Tw_1^n$, fixed values

Good idea to require $1-2\alpha > 0$ or $\Delta t < \Delta x^2/2\kappa$ ---- why?

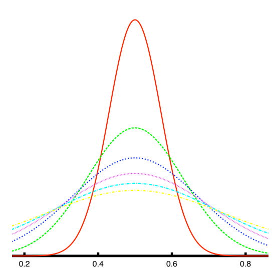
Example, by hand: t=0, $T_j=0$ unless j=10: $T_{10} = 1$. Then

$T^1_{10} = 1-2\alpha$, $T^1_{11} = T^1_9 = \alpha$, If $2\alpha > 1$ then T^1_{10} is negative, which is wrong.



$$\alpha \quad 1 - 2\alpha \quad \alpha \qquad (1-2\alpha)^2 + 2\alpha^2,\ 2\alpha(1-2\alpha),\ \alpha^2$$

$\delta \sim \sqrt{4\kappa t}$



$$\int_{-x}^{x} T dx = \kappa\partial_x T|_{-x}^{x} \to 0 \; x \to \infty$$

unbounded hot spot

total heat is conserved

Pseudo code, without storing time levels (*n*) -- as in CFD codes. For constant temperature.

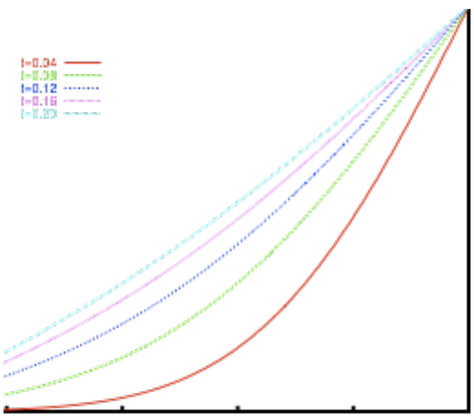
Initialize T(x): `T(:)=T0(:)`

```
FOR t=ΔT to NΔt

S1=T0
DO j=2,J-1
  SS = T(j)
  T(j) = (1-2α)T(j)+α(T(j+1)+S1)
  S1=SS
ENDDO
```
Plot T(x) if desired
end of time loop

or

```
FOR t=ΔT to NΔt

SS(:)  = T(:)
DO j=2,J-1
  T(j) = (1-2α)SS(j)+α(SS(j+1)+SS(j-1))
ENDDO
```
Plot T(x) if desired
end of time loop

Illustration with insulated wall



D. Could use RK-2 approach:

```
Half-step
DO j=2,J-1
  Th(j) = T(j) + 0.5 α [ T(j+1)-2T(j)+T(j-1) ]
ENDDO
Th(1)=Th(2)+QΔ x   !  dT/dx =-Q,  Impose b.c.'s at each RK step
Th(J)=T_wall
Midpoint rule
DO j=2,J-1
  T(j) = T(j) +  α [ Th(j+1)-2Th(j)+Th(j-1) ]
ENDDO
T(1)=T(2)+QΔ x
T(J)=T_wall
```

E. Generally, to use ODE solver. Write as a system of *semi-discrete* equations

$$d_t T_j = \kappa \delta^2_x T = \kappa\,(T_{j+1} - 2T_j + T_{j-1}) / \Delta x^2 \equiv RHS_j$$

This is a system of J-2 o.d.e.'s $d_t T_j = RHS_j$

======================

```
!---- How to solve heat equation by RKn  (For constant wall temperatures = initial values)
REAL(8) :: T(200)
NT = tend/Δt ;   initialize T(:)=T0(:)
Time loop: DO it = 1,NT
   tim = it*dt
   call rkn(200,RHS,T,tim,tim+Δt)   ---- plot results at desired times
 END DO Time loop
```

yp is the RHS of heat equation

```
SUBROUTINE RHS(N,y,yp,Δt)
REAL :: y(N),yp(N)
    yp(1) = 0. ; yp(N)=0.        ! const. Temperature B.C.: don't change wall value
  DO i= 2,N-1
    yp(i) = alpha/Δt *( y(i+1)+y(i-1)-2.*y(i) )   ! interior points recall  alpha/Δt = κ/Δx^2
RETURN
```

2nd order Runge-Kutta routine

```
 SUBROUTINE rk2(N,RHS,t,y,tend)
 REAL :: y(N),yp(N),y1(N),y1p(N)
     h = tend-t
     CALL RHS(N,y,yp,h)
         y1 = y+.5*yp*h
    CALL RHS(N,y1,y1p,h)
         y = y+y1p*h
RETURN
```

4th order Runge-Kutta routine

```
SUBROUTINE rk4(N,RHS,y,t,tend)
REAL :: y(N),yp(N),y1(N),y1p(N),y2(N),y2p(N),y3(N),y3p(N)
   h  = tend-t
  CALL RHS(n,y,yp,h)
      y1 = y+0.5*h*yp
  CALL RHS(n,y1,y1p,h)
      y2 = y+0.5*h*y1p
  CALL RHS(n,y2,y2p,h)
      y3 = y+h*y2p
  CALL RHS(n,y3,y3p,h)
      y = y+h/6.0*(y3p+2.0d0*y2p+2.0d0*y1p+yp)
 RETURN
```