Métodos de diferenças finitas.

MET-576-4

Modelagem Numérica da Atmosfera

Dr. Paulo Yoshio Kubota

Os métodos numéricos, formulação e parametrizações utilizados nos modelos atmosféricos serão descritos em detalhe.

3 Meses
24 Aulas (2 horas cada)

Dinâmica:

Métodos numéricos amplamente utilizados na solução numérica das equações diferencias parciais que governam os movimentos na atmosfera serão o foco, mas também serão analisados os novos conceitos e novos métodos.

Métodos de diferenças finitas.

- ✓ Métodos de diferenças finitas.
- ✓ Acurácia.
- ✓ Consistência.
- ✓ Estabilidade.
- ✓ Convergência.
- ✓ Grades de Arakawa A, B, C e E.
- ✓ Domínio de influência e domínio de dependência.
- ✓ Dispersão numérica e dissipação.
- ✓ Definição de filtros monótono e positivo.
- ✓ Métodos espectrais.
- ✓ Métodos de volume finito.
- ✓ Métodos Semi-Lagrangeanos.
- ✓ Conservação de massa local.
- ✓ Esquemas explícitos versus semi-implícitos.
- ✓ Métodos semi-implícitos.

Difusão

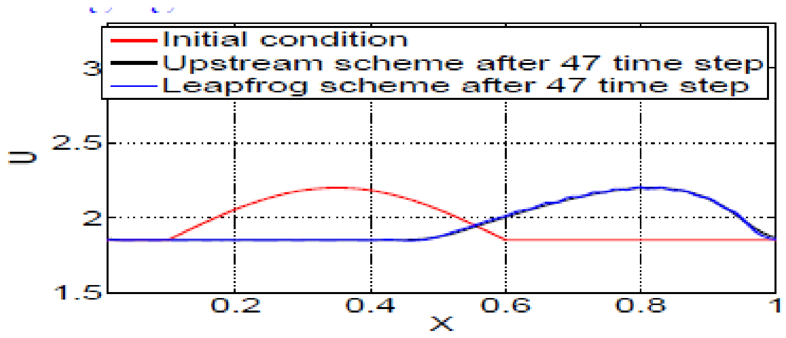
"a discretização dos termos de fricção e difusão e como isso afeta a estabilidade da solução. Esses termos estão incluídos na maioria dos modelos, desde os mais simples baseados nas equações de águas rasas até modelos de circulação geral oceano-atmosfera altamente complexos.

A advecção regime não-linear

Equação de Advecção Linear pode ser usada para modelar a advecção de algumas quantidades (por exemplo, temperatura, umidade) pelo vento, mas A equação para a energia eólica propriamente dita não é linear ou seja

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = 0$$

A não-linearidade permite mais interesse na dinâmica. Mas pode causar problemas numéricos tanto por meio de truncamento e a estabilidade. Ele gera grande gradiente



A solução mais simples é a de incluir a difusão que vai Evitar a formação de grandes gradientes

Paulo Yoshio Kubota

A equação de difusão

Considerar a difusão equação linear

$$\frac{\partial \emptyset}{\partial t} = K\frac{\partial^2 \emptyset}{\partial^2 x}, \qquad onde, K = constante$$

Assumir o domínio é periódicos em x. Se o estado inicial $(t = 0)$ é

$$\emptyset(x, 0) = e^{ikx}$$

Em seguida, a solução exata pode ser encontrado da seguinte forma. Assuma a solução $\emptyset(x,t) = \Phi(t)e^{ikx}$. E substitua na equação de Difusão

$$\Phi\ (t) = \Phi_0 e^{-k^2Kt}$$

$$t = 0$$

$$\Phi\ (0) = \Phi_0$$

$$\emptyset(x,t) = \Phi_0 e^{-k^2Kt} e^{ikx}$$

Considere a amplitude $\Phi_0 = 1$, Portanto a solução da equação de difusão torna-se:

$$\emptyset(x,t) = e^{-k^2Kt} e^{ikx}$$

Substituindo na equação de difusão

$$\frac{\partial\emptyset}{\partial t} = K\frac{\partial^2\emptyset}{\partial^2 x}, \qquad onde, K = constante$$

$$\frac{\partial\emptyset(x,t)}{\partial t} = -k^2K\Phi_0 e^{-k^2Kt}e^{ikx}$$

$$\frac{\partial\emptyset(x,t)}{\partial x} = ik\Phi_0 e^{-k^2Kt}e^{ikx}$$

$$\frac{\partial^2\emptyset(x,t)}{\partial^2 x} = -k^2\Phi_0 e^{-k^2Kt}e^{ikx}$$

$$\frac{\partial\emptyset}{\partial t} = K\frac{\partial^2\emptyset}{\partial^2 x}$$

$$-k^2K\Phi_0 e^{-k^2Kt}e^{ikx} = -k^2K\Phi_0 e^{-k^2Kt}e^{ikx}$$

Métodos de diferenças finitas.

Assim a solução para a equação de difusão satisfaz a equação diferencial

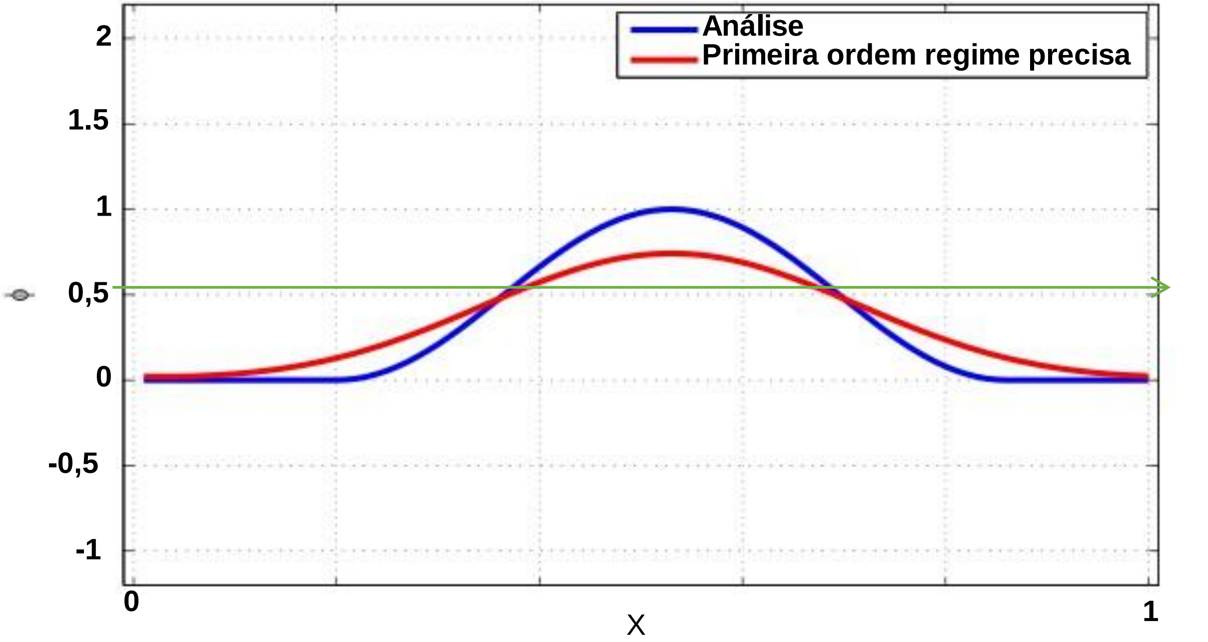
$$\frac{\partial \Phi(t) e^{ikx}}{\partial t} = -k^2 K \Phi(t) e^{ikx}$$

A solução é uma onda estacionária diminuindo a amplitude. O amortecimento é mais rápido para ondas curtas de que ondas longas.

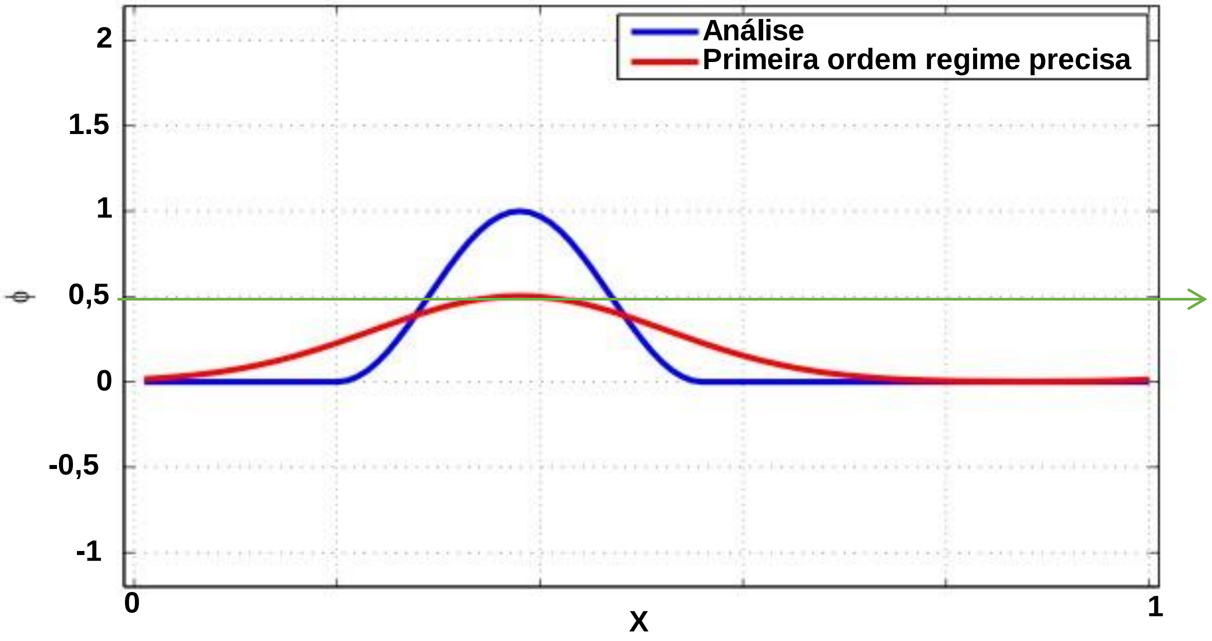
Solução analítica

$$\boxed{\emptyset(x,t) = \Phi_0 e^{-k^2 K t} e^{ikx}}$$

Solução apos 200 tempo passo



Solução apos 400 tempo passo



Estabilidade

Considere uma Equação diferencial ordinária usando o esquema diferença centralizada de segunda ordem (Leap_Frog)

$$\frac{\partial \emptyset}{\partial t} = K\frac{\partial^2 \emptyset}{\partial^2 x}, \qquad onde, K = constante$$

$$\frac{\emptyset_j^{n+1} - \emptyset_j^{n-1}}{2\Delta t} = K\frac{\emptyset_{j+1}^{n-1} - 2\emptyset_j^{n-1} + \emptyset_{j-1}^{n-1}}{\Delta x^2}$$

Supõem-se que a solução é dada pela equação: $\emptyset_j^n = A^n e^{ikj\Delta x}$, substituindo o esquema numérico

$$\frac{A^{n+1}e^{ikj\Delta x} - A^{n-1}e^{ikj\Delta x}}{2\Delta t} = K\left(\frac{A^n e^{ik(j+1)\Delta x} - 2A^n e^{ik(j)\Delta x} + A^n e^{ik(j-1)\Delta x}}{\Delta x^2}\right)$$

$$A^{n+1}e^{ikj\Delta x} = A^{n-1}e^{ikj\Delta x} + \frac{2K\Delta t}{\Delta x^2}\left(A^n e^{ik(j+1)\Delta x} - 2A^n e^{ik(j)\Delta x} + A^n e^{ik(j-1)\Delta x}\right)$$

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

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

Eliminando os termos $A^n e^{ikj\Delta x}$

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

Multilica a equação por A

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

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

$$\cos(k\Delta x) = \frac{e^{ik\Delta x} + e^{-ik\Delta x}}{2}$$

$$A^2 = 1 + A\frac{2K\Delta t}{\Delta x^2}\left(-2 + 2\cos(k\Delta x)\right)$$

$$A^2 - A\frac{2K\Delta t}{\Delta x^2}\left(-2 + 2\cos(k\Delta x)\right) - 1 = 0$$

$$A^2 - A\frac{2K\Delta t}{\Delta x^2}(-2+2\cos(k\Delta x)) - 1 = 0$$

$$A^2 + A\frac{2K\Delta t}{\Delta x^2}(2-2\cos(k\Delta x)) - 1 = 0$$

$$\boxed{ax^2 + bx + c = 0}$$

$$\boxed{x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}}$$

$$A = \frac{-\frac{2K\Delta t}{\Delta x^2}(2-2\cos(k\Delta x)) \pm \sqrt{\left(\frac{2K\Delta t}{\Delta x^2}(2-2\cos(k\Delta x))\right)^2 + 4}}{2}$$

$$A = \frac{-\frac{4K\Delta t}{\Delta x^2}(1-\cos(k\Delta x)) \pm \sqrt{\left(\frac{4K\Delta t}{\Delta x^2}(1-\cos(k\Delta x))\right)^2 + 4}}{2}$$

$$A = -\frac{4K\Delta t}{2\Delta x^2}(1-\cos(k\Delta x)) \pm \sqrt{\frac{1}{4}\left(\frac{4K\Delta t}{\Delta x^2}(1-\cos(k\Delta x))\right)^2 + \frac{4}{4}}$$

$$A = -\frac{4K\Delta t}{2\Delta x^2}(1-\cos(k\Delta x)) \pm \sqrt{\frac{1}{4}\left(\frac{4K\Delta t}{\Delta x^2}(1-\cos(k\Delta x))\right)^2 + \frac{4}{4}}$$

$$A = -\frac{2K\Delta t}{\Delta x^2}(1-\cos(k\Delta x)) \pm \sqrt{\left(\frac{4K\Delta t}{2\Delta x^2}(1-\cos(k\Delta x))\right)^2 + 1}$$

$$A = -\frac{2K\Delta t}{\Delta x^2}(1-\cos(k\Delta x)) \pm \sqrt{\left(\frac{2K\Delta t}{\Delta x^2}(1-\cos(k\Delta x))\right)^2 + 1}$$

$$A = -\frac{2K\Delta t}{\Delta x^2}(1-\cos(k\Delta x)) \pm \sqrt{\left(\frac{2K\Delta t}{\Delta x^2}\right)^2 (1-\cos(k\Delta x))^2 + 1}$$

Ambas as raízes A_+ e A_- são reais ($(1-\cos(k\Delta x)) > 0$) e o seu produto é A_+A_- = -1. $A_- \leq -1$

Métodos de diferenças finitas.

$$A_+A_- = \left(-\frac{2K\Delta t}{\Delta x^2}(1-\cos(k\Delta x)) + \sqrt{\left(\frac{2K\Delta t}{\Delta x^2}\right)^2(1-\cos(k\Delta x))^2+1}\right)\left(-\frac{2K\Delta t}{\Delta x^2}(1-\cos(k\Delta x)) - \sqrt{\left(\frac{2K\Delta t}{\Delta x^2}\right)^2(1-\cos(k\Delta x))^2+1}\right)$$

Ambas as raízes A+ e A- são reais ($(1-\cos(k\Delta x)) > 0$) e o seu produto é A+A- = -1. A- ≤ -1

Métodos de diferenças finitas.

$$A_+A_- = \left(-\frac{2K\Delta t}{\Delta x^2}(1-\cos(k\Delta x)) + \sqrt{\left(\frac{2K\Delta t}{\Delta x^2}\right)^2 (1-\cos(k\Delta x))^2 + 1}\right)\left(-\frac{2K\Delta t}{\Delta x^2}(1-\cos(k\Delta x)) - \sqrt{\left(\frac{2K\Delta t}{\Delta x^2}\right)^2 (1-\cos(k\Delta x))^2 + 1}\right)$$

Métodos de diferenças finitas.

$$
\begin{aligned}
A_+A_- = &\left(\frac{2K\Delta t}{\Delta x^2}(1-\cos(k\Delta x))\right)^2 \\
&+\left(\frac{2K\Delta t}{\Delta x^2}(1-\cos(k\Delta x))\right)\sqrt{\left(\frac{2K\Delta t}{\Delta x^2}\right)^2(1-\cos(k\Delta x))^2+1} \\
&-\left(\frac{2K\Delta t}{\Delta x^2}(1-\cos(k\Delta x))\right)\sqrt{\left(\frac{2K\Delta t}{\Delta x^2}\right)^2(1-\cos(k\Delta x))^2+1} \\
&-\left(\sqrt{\left(\frac{2K\Delta t}{\Delta x^2}\right)^2(1-\cos(k\Delta x))^2+1}\right)^2
\end{aligned}
$$

Métodos de diferenças finitas.

$$A_+A_- = \left(\frac{2K\Delta t}{\Delta x^2}\right)^2 (1-\cos(k\Delta x))^2$$
$$+\left(\frac{2K\Delta t}{\Delta x^2}(1-\cos(k\Delta x))\right)\sqrt{\left(\frac{2K\Delta t}{\Delta x^2}\right)^2 (1-\cos(k\Delta x))^2+1}$$
$$-\left(\frac{2K\Delta t}{\Delta x^2}(1-\cos(k\Delta x))\right)\sqrt{\left(\frac{2K\Delta t}{\Delta x^2}\right)^2 (1-\cos(k\Delta x))^2+1}$$
$$-\left(\left(\frac{2K\Delta t}{\Delta x^2}\right)^2 (1-\cos(k\Delta x))^2+1\right)$$

$$A_+A_- = \cancel{\left(\frac{2K\Delta t}{\Delta x^2}\right)^2 (1-\cos(k\Delta x))^2}$$

$$+\cancel{\left(\frac{2K\Delta t}{\Delta x^2}(1-\cos(k\Delta x))\right)\sqrt{\left(\frac{2K\Delta t}{\Delta x^2}\right)^2 (1-\cos(k\Delta x))^2+1}}$$

$$-\cancel{\left(\frac{2K\Delta t}{\Delta x^2}(1-\cos(k\Delta x))\right)\sqrt{\left(\frac{2K\Delta t}{\Delta x^2}\right)^2 (1-\cos(k\Delta x))^2+1}}$$

$$-\cancel{\left(\frac{2K\Delta t}{\Delta x^2}\right)^2 (1-\cos(k\Delta x))^2} - 1$$

$$A_+A_- = -1$$

Ambas as raízes A_+ e A_- São reais e o seu produto é $A_+A_- = -1$.

$A_+A_- = -1$

A- ≤ -1

$$A_- = -\left(\frac{2K\Delta t}{\Delta x^2}\right)(1-\cos(k\Delta x)) - \sqrt{\left(\left(\frac{2K\Delta t}{\Delta x^2}\right)(1-\cos(k\Delta x))\right)^2+1}$$

$$\left(1-cos(k\Delta x)\right) > 0$$

$$A_- = -\left(\frac{2K\Delta t}{\Delta x^2}\right)(1-\cos(k\Delta x)) - \sqrt{\left(\left(\frac{2K\Delta t}{\Delta x^2}\right)(1-\cos(k\Delta x))\right)^2+1}$$

<0 <0

A- ≤ -1

Paulo Yoshio Kubota

$$A_+ = -\left(\frac{2K\Delta t}{\Delta x^2}\right)(1-\cos(k\Delta x)) + \sqrt{\left(\left(\frac{2K\Delta t}{\Delta x^2}\right)(1-\cos(k\Delta x))\right)^2 + 1}$$

(Usando $\sqrt{(y+1)} \approx 1 + \frac{1}{2}y$,

$$Série\ de\ Taylor\ \sqrt{x+1}\ com\ centro\ em\ 0 = 1 + \frac{1}{2}x - \frac{1}{8}x^2 + \frac{1}{16}x^3 +++ +$$

$$\sqrt{\left(\left(\frac{2K\Delta t}{\Delta x^2}\right)(1-\cos(k\Delta x))\right)^2 + 1} = 1 + \frac{1}{2}\left(\left(\frac{2K\Delta t}{\Delta x^2}\right)(1-\cos(k\Delta x))\right)^2$$

$$A_+ = -\left(\frac{2K\Delta t}{\Delta x^2}\right)(1-\cos(k\Delta x)) + 1 + \frac{1}{2}\left(\left(\frac{2K\Delta t}{\Delta x^2}\right)(1-\cos(k\Delta x))\right)^2$$

$$\left(1 - cos(k\Delta x)\right) > 0$$

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

> 0 <0 >0

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

> 0 <0 >0

Resultando em $A_{+} > 0$; $A_{+} < 1$.

A_{+} situa-se entre 0 e 1. E isso corresponde ao modo físico. Resultando em $|A| < 1$

Resultando em $A_{+} \geq 1$.

Corresponde ao modo computacional e é incondicionalmente instável. Resultando em $|A| \geq 1$

Estabilidade

Agora, considere a diferença forward de primeira ordem para a derivada no tempo.

$$\frac{\partial \emptyset}{\partial t} = K\frac{\partial^2 \emptyset}{\partial^2 x}, \qquad onde, K = constante$$

$$\frac{\emptyset_j^{n+1} - \emptyset_j^n}{\Delta t} = K\left(\frac{\emptyset_{j+1}^n - 2\emptyset_j^n + \emptyset_{j-1}^n}{\Delta x^2}\right)$$

Supõem-se que a solução é dada pela equação: $\emptyset_j^n = A^n e^{ikj\Delta x}$**, substituindo o esquema numérico. A análise de estabilidade de Von Neumann leva à seguinte equação para o fator de ampliação**

$$\frac{A^{n+1}e^{ikj\Delta x} - A^n e^{ikj\Delta x}}{\Delta t} = K\left(\frac{A^n e^{ik(j+1)\Delta x} - 2A^n e^{ik(j)\Delta x} + A^n e^{ik(j-1)\Delta x}}{\Delta x^2}\right)$$

$$A^{n+1}e^{ikj\Delta x} = A^n e^{ikj\Delta x} + \frac{K\Delta t}{\Delta x^2}\left(A^n e^{ik(j+1)\Delta x} - 2A^n e^{ik(j)\Delta x} + A^n e^{ik(j-1)\Delta x}\right)$$

$$A^n A e^{ikj\Delta x} = A^n e^{ikj\Delta x} + \frac{K\Delta t}{\Delta x^2}\left(A^n e^{ikj\Delta x}e^{ik\Delta x} - 2A^n e^{ikj\Delta x} + A^n e^{ikj\Delta x}e^{-ik\Delta x}\right)$$

Eliminado os termos: $A^n e^{ikj\Delta x}$

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

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

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

$$A = 1 + \frac{K\Delta t}{\Delta x^2}\left(-2 + 2\cos(k\Delta x)\right)$$

análise de estabilidade de Von Neumann leva à seguinte equação para o fator de ampliação

$$A = 1 - \frac{2K\Delta t}{\Delta x^2}\left(1 + \cos(k\Delta x)\right)$$

$$A = \left(1 - \frac{2K\Delta t}{\Delta x^2}(1+\cos(k\Delta x))\right) = \left(1 + i^2\frac{2K\Delta t}{\Delta x^2}(1+\cos(k\Delta x))\right)$$

Agora $0 \leq (1+\cos(k\Delta x)) \leq 2$, Então $|A| \leq 1$ se $\frac{2K\Delta t}{\Delta x^2} \leq 1$, Ounseja se $\Delta t \leq \frac{\Delta x^2}{2K}$

$$|A|^2 = AA^*$$

$$|A|^2 = \left(1 + i^2\frac{2K\Delta t}{\Delta x^2}(1+\cos(k\Delta x))\right)\left(1 - i^2\frac{2K\Delta t}{\Delta x^2}(1+\cos(k\Delta x))\right)$$

$$|A|^2 = \left(1 + \frac{2K\Delta t}{\Delta x^2}(1+\cos(k\Delta x)) - \frac{2K\Delta t}{\Delta x^2}(1+\cos(k\Delta x)) - \left(\frac{2K\Delta t}{\Delta x^2}(1+\cos(k\Delta x))\right)^2\right)$$

$$|A|^2 = \left(1 - \left(\frac{2K\Delta t}{\Delta x^2}(1+\cos(k\Delta x))\right)^2\right)$$

Este regime é condicionalmente estável. Note-se que devido à dependência do quadrado de Δx, um Aumento moderado na resolução podem exigir um intervalo de tempo Δt muito menor. Ex. fazer a análise de estabilidade para o esquema de versões anteriores da derivada temporal

```fortran
MODULE Difusion1D
 IMPLICIT NONE
 PRIVATE
 PUBLIC ::  Init,Run,Finalize
 INTEGER                        :: NX
 REAL                           :: X0
 REAL                           :: DX
 REAL                           :: DT
 REAL                           :: kappa
 REAL                           :: total_time
 REAL   , ALLOCATABLE :: X  (:)
 REAL   , ALLOCATABLE :: T  (:)
 REAL   , ALLOCATABLE :: Tp (:)
 CHARACTER (LEN=2), ALLOCATABLE :: CT (:)

CONTAINS
SUBROUTINE Init(NX_IN,X0_IN,DX_IN,kappa_in,DT_IN,total_time_in)
 INTEGER, INTENT (IN   ) :: NX_IN
 REAL      , INTENT (IN   ) :: X0_IN
 REAL      , INTENT (IN   ) :: DX_IN
 REAL      , INTENT (IN   ) :: kappa_in
 REAL      , INTENT (IN   ) :: DT_IN
 REAL      , INTENT (IN   ) :: total_time_in
 INTEGER :: i
 NX=NX_IN
 X0=X0_IN
 DX=DX_IN
 kappa=kappa_in
 DT=DT_IN
 total_time=total_time_in
 ALLOCATE (X (1:NX  ));X=0.0
 ALLOCATE (T (0:NX+1));T=0.0
 ALLOCATE (Tp(0:NX+1));Tp=0.0
 ALLOCATE (CT(0:NX+1));CT='  '
 FORALL (i=1:NX)X(i) = X0 + (i-1)*DX
 T=273.16
 T(0:1) = 300.0
 CT='. '
END SUBROUTINE Init
```

```fortran
SUBROUTINE Run()
!
! dT          d dT
!----- = k * ---------
! dt          dx dx
!
! k=1
!
 IMPLICIT NONE
 INTEGER :: itr
 INTEGER :: i

 DO itr=1,(total_time/DT)
   DO i=1,NX
      Tp(i) = T(i) + DT*(kappa*((T(i+1) + T(i-1) - 2*T(i))/2*DX))
   END DO
   T(1:NX)=Tp(1:NX)
   WHERE (T /= 273.16)CT='X '
   PRINT *,'CT'  ,MAXVAL(T),MINVAL(T)
   WRITE(*,'(50A)')(CT(i),i=0,NX-1)

 END DO

END SUBROUTINE Run

SUBROUTINE Finalize()
 IMPLICIT NONE
 DEALLOCATE (X )
 DEALLOCATE (T )
 DEALLOCATE (Tp)
 DEALLOCATE (CT)

END SUBROUTINE Finalize

END MODULE Difusion1D
```

Paulo Yoshio Kubota

```
!------------
PROGRAM MAIN
 USE Difusion1D, Only :Init,Run,Finalize
 IMPLICIT NONE
 INTEGER, PARAMETER :: NX   =50
 REAL   , PARAMETER :: DX   = 5.0/REAL(NX)!m
 REAL   , PARAMETER :: alfa   =10.0 ! 0.1 -> 1
 REAL   , PARAMETER :: total_time   =300.0
 REAL   , PARAMETER :: kappa=2.0
 REAL   , PARAMETER :: DT=  (alfa*(DX**2))/kappa
 REAL               :: X0=1!m
 PRINT*,total_time/DT
 CALL Init(NX,X0,DX,kappa,DT,total_time)
 CALL Run()
 CALL Finalize()
END PROGRAM MAIN
```

Exercício de difusão

Resolver numericamente o problema de difusão que tem sido discutido na seção anterior usando o esquema de diferença finitas para a derivada temporal. Use uma resolução espacial de $\Delta x = 10^{-2}$ m e Coeficiente de difusão $K = 2{,}9 \times 10^{-5}$. Integrar para pelo menos

Para 6 horas (cerca de 25000 segundos), e mostrar as soluções para T = 1 Hora, T = 2 Horas, T = 3 Horas, T = 4 Horas, T = 5 Horas, e T = 6 Horas. Comparar a solução com o Uma análise Fourier (retenção 1000 componentes). Escolha o passo de tempo sendo de tal ordem que o sistema seja estável. Deixe o primeiro setup em temperatura de 1m haste longa dada.

$$\emptyset(x) = \begin{cases} 273.15 + 20x & 0 \le x \le 0.5 \\ 273.15 + 20 - 20x & 0.5 < X < 1 \end{cases}$$

Ambas as extremidades são mantidos na mesma temperatura T_0 = 273.15K.

```fortran
MODULE Class_Fields
  IMPLICIT NONE
  PRIVATE
  INTEGER, PUBLIC      , PARAMETER  :: r8=8
  INTEGER, PUBLIC      , PARAMETER  :: r4=4
  REAL(KIND=r8),PUBLIC  ,ALLOCATABLE :: PHI_P(:)
  REAL(KIND=r8),PUBLIC  ,ALLOCATABLE :: PHI_C(:)
  REAL(KIND=r8),PUBLIC  ,ALLOCATABLE :: PHI_M(:)
  REAL(KIND=r8),PUBLIC               :: Coef_K
  INTEGER      ,PUBLIC               :: iMax
  PUBLIC :: Init_Class_Fields
CONTAINS
!-------------------------------------------------------------------------------------------
  SUBROUTINE Init_Class_Fields(xdim,Coef_K0)
    IMPLICIT NONE
    INTEGER      , INTENT(IN   ) :: xdim
    REAL(KIND=r8), INTENT(IN   ):: Coef_K0
    iMax=xdim
    Coef_K=Coef_K0
    ALLOCATE(PHI_P(-1:iMax+2))
    ALLOCATE(PHI_C(-1:iMax+2))
    ALLOCATE(PHI_M(-1:iMax+2))
  END SUBROUTINE Init_Class_Fields
!-------------------------------------------------------------------------------------------
END MODULE Class_Fields
```

```fortran
MODULE Class_NumericalMethod
  USE Class_Fields, Only : PHI_P,PHI_C,PHI_M,Coef_K,iMax
 IMPLICIT NONE
 PRIVATE
 INTEGER, PUBLIC      , PARAMETER :: r8=8
 INTEGER, PUBLIC      , PARAMETER :: r4=4
 REAL(KIND=r8) :: Dt
 REAL(KIND=r8) :: Dx

 PUBLIC :: InitNumericalScheme
 PUBLIC :: SchemeUpdate
 PUBLIC :: SchemeUpStream
CONTAINS
!-------------------------------------------------------------------------------------------
 SUBROUTINE InitNumericalScheme(dt_in,dx_in)
   IMPLICIT NONE
   REAL(KIND=r8), INTENT(IN   ) :: dt_in
   REAL(KIND=r8), INTENT(IN   ) :: dx_in
   INTEGER :: i
   REAL(KIND=r8) :: x
   Dt=dt_in
   Dx=dx_in
   PHI_C = 273.15
   x=0.0
   DO i=1,iMax
      IF(x >= 0.0 .and. x < 0.5) THEN
         PHI_C(i)= 273.15 + 2*X
      ELSE IF(x > 0.5 .and. x < 1.0) THEN
         PHI_C(i)= 273.15 + 2.0 - 2*X
      END IF
       x =  (i)* DX
   END DO
   PHI_M=PHI_C
   PHI_P=PHI_C
 END SUBROUTINE InitNumericalScheme
```

```
!------------------------------------------------------------------------------------------
 FUNCTION SchemeUpStream()  RESULT(ok)
   IMPLICIT NONE
   ! Utilizando a diferenciacao forward no tempo e
   ! backward no espaco (upstream)
   !
   ! F(j,n+1) - F(j,n)       F(j+1,n) -2F(j,n) + F(j-1,n)
   !------------------- + u ----------------------------- = 0
   !      dt                    dx
   !
   INTEGER :: ok
   INTEGER :: j
   DO j=1,iMax
      PHI_P(j) = PHI_C(j) - (Coef_K*Dt/(Dx**2))* &
                 (PHI_C(j+1) -2.0*PHI_C(j) + PHI_C(j-1))
   END DO
   CALL UpdateBoundaryLayer()
 END FUNCTION SchemeUpStream
!------------------------------------------------------------------------------------------
 SUBROUTINE UpdateBoundaryLayer()
   IMPLICIT NONE
   PHI_P(0)      =  273.15 !PHI_P(iMax )
   PHI_P(-1)     =  273.15 !PHI_P(iMax-1)
   PHI_P(imax+1) =  273.15 !PHI_P(1)
   PHI_P(iMax+2) =  273.15 !PHI_P(2)
 END  SUBROUTINE UpdateBoundaryLayer
!------------------------------------------------------------------------------------------
 FUNCTION SchemeUpdate()  RESULT(ok)
   IMPLICIT NONE
   INTEGER :: ok
   PHI_M=PHI_C
   PHI_C=PHI_P
   ok=0
 END FUNCTION SchemeUpdate
!------------------------------------------------------------------------------------------
END MODULE Class_NumericalMethod
```

```
MODULE Class_WritetoGrads
 USE Class_Fields, Only:PHI_P,PHI_C,PHI_M,Coef_K,iMax
 IMPLICIT NONE
 PRIVATE
  INTEGER, PUBLIC      , PARAMETER  :: r8=8
  INTEGER, PUBLIC      , PARAMETER  :: r4=4
 INTEGER             , PARAMETER :: UnitData=1
 INTEGER             , PARAMETER :: UnitCtl=2
 CHARACTER(LEN=400)              :: FileName
 LOGICAL                         :: CtrlWriteDataFile
 PUBLIC :: SchemeWriteCtl
 PUBLIC :: SchemeWriteData
 PUBLIC :: InitClass_WritetoGrads
CONTAINS
 SUBROUTINE InitClass_WritetoGrads()
   IMPLICIT NONE
  FileName=''
  FileName='Difusion1D'
  CtrlWriteDataFile=.TRUE.
 END SUBROUTINE InitClass_WritetoGrads

 FUNCTION SchemeWriteData(irec)  RESULT(ok)
   IMPLICIT NONE
   INTEGER   , INTENT(INOUT) :: irec
   INTEGER             :: ok
   INTEGER             :: lrec
   REAL(KIND=r4)       :: Yout(iMax)
   INQUIRE(IOLENGTH=lrec) Yout
   IF(CtrlWriteDataFile)OPEN(UnitData,FILE=TRIM(FileName)//'.bin', &
     FORM='UNFORMATTED',ACCESS='DIRECT',STATUS='UNKNOWN', &
     ACTION='WRITE',RECL=lrec)
   ok=1
   CtrlWriteDataFile=.FALSE.
   Yout=REAL(PHI_C(1:iMax),KIND=r4)
   irec=irec+1
   WRITE(UnitData,rec=irec)Yout
   ok=0
 END FUNCTION SchemeWriteData
```

Kubota

```fortran
 FUNCTION SchemeWriteCtl(nrec)  RESULT(ok)
   IMPLICIT NONE
   INTEGER, INTENT(IN) :: nrec
   INTEGER             :: ok
   ok=1
   OPEN(UnitCtl,FILE=TRIM(FileName)//'.ctl',FORM='FORMATTED', &
   ACCESS='SEQUENTIAL',STATUS='UNKNOWN',ACTION='WRITE')
   WRITE(UnitCtl,'(A6,A        )')'dset ^',TRIM(FileName)//'.bin'
   WRITE(UnitCtl,'(A           )')'title  EDO'
   WRITE(UnitCtl,'(A           )')'undef  -9999.9'
   WRITE(UnitCtl,'(A6,I8,A18   )')'xdef  ',iMax,' linear 0.00 0.001'
   WRITE(UnitCtl,'(A           )')'ydef  1 linear  -1.27 1'
   WRITE(UnitCtl,'(A6,I6,A25   )')'tdef  ',nrec,' linear  00z01jan0001 1hr'
   WRITE(UnitCtl,'(A20         )')'zdef  1 levels 1000 '
   WRITE(UnitCtl,'(A           )')'vars 2'
   WRITE(UnitCtl,'(A           )')'phic 0 99 resultado da edol yc'
   WRITE(UnitCtl,'(A           )')'phia 0 99 solucao analitica ya'
   WRITE(UnitCtl,'(A           )')'endvars'
   CLOSE(UnitCtl,STATUS='KEEP')
   CLOSE(UnitData,STATUS='KEEP')
   ok=0
 END FUNCTION SchemeWriteCtl


END MODULE Class_WritetoGrads
```

```fortran
PROGRAM Main
 USE Class_Fields, Only:Init_Class_Fields
 USE Class_NumericalMethod, Only:InitNumericalScheme,&
                 SchemeUpdate,SchemeUpStream
 USE Class_WritetoGrads, Only :InitClass_WritetoGrads, &
                SchemeWriteData,SchemeWriteCtl
 IMPLICIT NONE
 INTEGER      , PARAMETER  :: r8=8
 INTEGER      , PARAMETER  :: r4=4
 REAL(KIND=r8)       , PARAMETER :: dx=0.01   !m
 REAL(KIND=r8)       , PARAMETER :: LX=1.0
 INTEGER             , PARAMETER :: xdim=LX/dx
 REAL(KIND=r8)       , PARAMETER :: Coef_K0=2.9e-5!m/s
 REAL(KIND=r8)       , PARAMETER :: dt=0.1*(dx**2)/Coef_K0   !s
                                 !  c*Dt/Dx < 1 => dt < dx/Coef_K0
 INTEGER             , PARAMETER :: ninteraction=20000
 CALL Init()
 CALL run()
CONTAINS
 SUBROUTINE Init()
  IMPLICIT NONE
  CALL Init_Class_Fields(xdim,Coef_K0)
  CALL InitNumericalScheme(dt,dx)
  CALL InitClass_WritetoGrads
 END SUBROUTINE Init
 SUBROUTINE Run()
  IMPLICIT NONE
  INTEGER :: test,tn,irec
  irec=0
  DO tn=0,ninteraction
     PRINT*,(tn)*dt
     test=SchemeUpStream()
     test=SchemeWriteData(irec)
     test=SchemeUpdate()
  END DO
  test=SchemeWriteCtl(ninteraction)
 END SUBROUTINE Run
END PROGRAM Main
```

Estabilidade

Agora, considere o esquema implícito.

$$\frac{\partial \emptyset}{\partial t} = K\frac{\partial^2 \emptyset}{\partial^2 x}, \qquad onde, K = constante$$

$$\frac{\emptyset_j^{n+1} - \emptyset_j^n}{\Delta t} = K\frac{\emptyset_{j+1}^{n+1} - 2\emptyset_j^{n+1} + \emptyset_{j-1}^{n+1}}{\Delta x^2}$$

Supõem-se que a solução é dada pela equação: $\emptyset_j^n = A^n e^{ikj\Delta x}$ **, substituindo o esquema numérico. A análise de estabilidade de Von Neumann leva à seguinte equação para o fator de ampliação**

Estabilidade

Agora, considere o esquema implícito.

$$\frac{\partial \emptyset}{\partial t} = K\frac{\partial^2 \emptyset}{\partial^2 x}, \qquad onde, K = constante$$

$$\frac{\emptyset_j^{n+1} - \emptyset_j^n}{\Delta t} = K\frac{\emptyset_{j+1}^{n+1} - 2\emptyset_j^{n+1} + \emptyset_{j-1}^{n+1}}{\Delta x^2}$$

Supõem-se que a solução é dada pela equação: $\emptyset_j^n = A^n e^{ikj\Delta x}$, substituindo o esquema numérico. A análise de estabilidade de Von Neumann leva à seguinte equação para o fator de ampliação

$$\frac{A^n A e^{ikj\Delta x} - A^n A e^{ikj\Delta x}}{\Delta t} = K\frac{A^n A e^{ik(j+1)\Delta x} - 2A^n A e^{ikj\Delta x} + A^n A e^{ik(j-1)\Delta x}}{\Delta x^2}$$

Estabilidade

$$\frac{A-1}{\Delta t}A^n e^{ikj\Delta x} = K\frac{A^n A e^{ikj\Delta x}e^{ik\Delta x} - 2A^n A e^{ikj\Delta x} + A^n A e^{ikj\Delta x}e^{-ik\Delta x}}{\Delta x^2}$$

$$\frac{A-1}{\Delta t}A^n e^{ikj\Delta x} = A^n A e^{ikj\Delta x} K\frac{Ae^{ik\Delta x} - 2A + Ae^{-ik\Delta x}}{\Delta x^2}$$

$$\frac{A-1}{\Delta t} = K\frac{Ae^{ik\Delta x} - 2A + Ae^{-ik\Delta x}}{\Delta x^2}$$

$$A-1 = \frac{\Delta t K}{\Delta x^2}\left(Ae^{ik\Delta x} - 2A + Ae^{-ik\Delta x}\right)$$

$$\frac{A-1}{(Ae^{ik\Delta x} - 2A + Ae^{-ik\Delta x})} = \frac{\Delta t K}{\Delta x^2}$$

Estabilidade

$$\frac{A-1}{(Ae^{ik\Delta x}-2A+Ae^{-ik\Delta x})}=\frac{\Delta tK}{\Delta x^2}$$

$$\frac{A-1}{(Ae^{ik\Delta x}+Ae^{-ik\Delta x}-2A)}=\frac{\Delta tK}{\Delta x^2}$$

$$\frac{A-1}{\left(2A\frac{e^{ik\Delta x}+e^{-ik\Delta x}}{2}-2A\right)}=\frac{\Delta tK}{\Delta x^2}$$

$$\frac{A-1}{(2Acos(k\Delta x)-2A)}=\frac{\Delta tK}{\Delta x^2}$$

Estabilidade

$$\frac{A-1}{(2A\cos(k\Delta x)-2A)}=\frac{\Delta tK}{\Delta x^2}$$

$$A-1=\frac{2\Delta tK}{\Delta x^2}(A\cos(k\Delta x)-A)$$

$$A-1=\left(\frac{2\Delta tK}{\Delta x^2}A\cos(k\Delta x)-\frac{2A\Delta tK}{\Delta x^2}\right)$$

$$A-\frac{2\Delta tK}{\Delta x^2}A\cos(k\Delta x)+\frac{2A\Delta tK}{\Delta x^2}=1$$

$$A\left(1-\frac{2\Delta tK}{\Delta x^2}\cos(k\Delta x)+\frac{2\Delta tK}{\Delta x^2}\right)=1$$

$$A=\frac{1}{\left(1-\frac{2\Delta tK}{\Delta x^2}\cos(k\Delta x)+\frac{2\Delta tK}{\Delta x^2}\right)}$$

Estabilidade

$$A = \frac{1}{\left(1 - \frac{2\Delta t K}{\Delta x^2} cos(k\Delta x) + \frac{2\Delta t K}{\Delta x^2}\right)}$$

$$\left(1 - \frac{2\Delta t K}{\Delta x^2} cos(k\Delta x) + \frac{2\Delta t K}{\Delta x^2}\right)^2 \geq (1)^2$$

$$1 - \frac{2\Delta t K}{\Delta x^2} cos(k\Delta x) + \frac{2\Delta t K}{\Delta x^2} \geq 1$$

$$-\frac{2\Delta t K}{\Delta x^2} cos(k\Delta x) \geq -\frac{2\Delta t K}{\Delta x^2}$$

$$\frac{2\Delta t K}{\Delta x^2} cos(k\Delta x) \leq \frac{2\Delta t K}{\Delta x^2}$$

Este regime é condicionalmente estável para $cos(k\Delta x) \leq 1$

Algoritmo Difusão implícito

Métodos de diferenças finitas.

$$\frac{\emptyset_j^{n+1} - \emptyset_j^n}{\Delta t} = K\frac{\emptyset_{j+1}^{n+1} - 2\emptyset_j^{n+1} + \emptyset_{j-1}^{n+1}}{\Delta x^2}$$

$$\emptyset_j^{n+1} - \emptyset_j^n = K\frac{\Delta t}{\Delta x^2}\left(\emptyset_{j+1}^{n+1} - 2\emptyset_j^{n+1} + \emptyset_{j-1}^{n+1}\right)$$

$$\emptyset_j^{n+1} - \emptyset_j^n = \left(K\frac{\Delta t}{\Delta x^2}\emptyset_{j+1}^{n+1} - 2K\frac{\Delta t}{\Delta x^2}\emptyset_j^{n+1} + K\frac{\Delta t}{\Delta x^2}\emptyset_{j-1}^{n+1}\right)$$

Métodos de diferenças finitas.

$$\emptyset_j^{n+1} - \emptyset_j^n = \left(K\frac{\Delta t}{\Delta x^2}\emptyset_{j+1}^{n+1} - 2K\frac{\Delta t}{\Delta x^2}\emptyset_j^{n+1} + K\frac{\Delta t}{\Delta x^2}\emptyset_{j-1}^{n+1}\right)$$

$$\emptyset_j^{n+1} - K\frac{\Delta t}{\Delta x^2}\emptyset_{j+1}^{n+1} + 2K\frac{\Delta t}{\Delta x^2}\emptyset_j^{n+1} - K\frac{\Delta t}{\Delta x^2}\emptyset_{j-1}^{n+1} = \emptyset_j^n$$

$$-K\frac{\Delta t}{\Delta x^2}\emptyset_{j-1}^{n+1} + \left(1 + 2K\frac{\Delta t}{\Delta x^2}\right)\emptyset_j^{n+1} - K\frac{\Delta t}{\Delta x^2}\emptyset_{j+1}^{n+1} += \emptyset_j^n$$

Métodos de diferenças finitas.

$$-K\frac{\Delta t}{\Delta x^2}\emptyset_{j-1}^{n+1}+\left(1+2K\frac{\Delta t}{\Delta x^2}\right)\emptyset_j^{n+1}-K\frac{\Delta t}{\Delta x^2}\emptyset_{j+1}^{n+1} += \emptyset_j^n$$

$$-Coef\_C*\emptyset_{j-1}^{n+1}+(1+2Coef\_C)*\emptyset_j^{n+1}-Coef\_C*\emptyset_{j+1}^{n+1} += \emptyset_j^n$$

$$Coef\_C = K\frac{\Delta t}{\Delta x^2}$$

Métodos de diferenças finitas.

$$-Coef\_C * \emptyset_{j-1}^{n+1} + (1 + 2Coef\_C) * \emptyset_{j}^{n+1} - Coef\_C * \emptyset_{j+1}^{n+1} += \emptyset_{j}^{n}$$

$$-Coef\_C * \emptyset_{0}^{n+1} + (1 + 2Coef\_C) * \emptyset_{1}^{n+1} - Coef\_C * \emptyset_{2}^{n+1} += \emptyset_{1}^{n}$$
$$-Coef\_C * \emptyset_{1}^{n+1} + (1 + 2Coef\_C) * \emptyset_{2}^{n+1} - Coef\_C * \emptyset_{3}^{n+1} += \emptyset_{2}^{n}$$
$$-Coef\_C * \emptyset_{2}^{n+1} + (1 + 2Coef\_C) * \emptyset_{3}^{n+1} - Coef\_C * \emptyset_{4}^{n+1} += \emptyset_{3}^{n}$$
$$-Coef\_C * \emptyset_{3}^{n+1} + (1 + 2Coef\_C) * \emptyset_{4}^{n+1} - Coef\_C * \emptyset_{5}^{n+1} += \emptyset_{4}^{n}$$
$$-Coef\_C * \emptyset_{4}^{n+1} + (1 + 2Coef\_C) * \emptyset_{5}^{n+1} - Coef\_C * \emptyset_{6}^{n+1} += \emptyset_{5}^{n}$$

....

$$-Coef\_C * \emptyset_{j-1}^{n+1} + (1 + 2Coef\_C) * \emptyset_{j}^{n+1} - Coef\_C * \emptyset_{j+1}^{n+1} += \emptyset_{j}^{n}$$

$$\begin{bmatrix} (1+2C) & -C & 0 & 0 & 0 & -C \\ -C & (1+2C) & -C & \dots\dots & \dots\dots & \dots\dots \\ 0 & -C & (1+2C) & -C & 0 & 0 \\ 0 & 0 & -C & (1+2C) & -C & 0 \\ \dots\dots & \dots\dots & \dots\dots & -C & (1+2C) & -C \\ -C & 0 & 0 & 0 & -C & (1+2C) \end{bmatrix} * \begin{bmatrix} \emptyset_{1}^{n+1} \\ \emptyset_{2}^{n+1} \\ \emptyset_{3}^{n+1} \\ \emptyset_{4}^{n+1} \\ \emptyset_{5}^{n+1} \\ \emptyset_{j}^{n+1} \end{bmatrix} = \begin{bmatrix} \emptyset_{1}^{n} \\ \emptyset_{2}^{n} \\ \emptyset_{3}^{n} \\ \emptyset_{4}^{n} \\ \emptyset_{5}^{n} \\ \emptyset_{j}^{n} \end{bmatrix}$$

Exercicio:

Construa um programa baseado no esquema implícito para a equação de difusão e integre a equação . Discuta o resultado

A advecção regime não-linear

(52)

Advecção-difusão

Considere um processo de advecção-difusão

$$\frac{\partial \emptyset}{\partial t} + u\frac{\partial \emptyset}{\partial x} = K\frac{\partial^2 \emptyset}{dx^2} \qquad onde\ K = contante$$

Muito restritivo

$$\boxed{\frac{\emptyset_j^{n+1} - \emptyset_j^{n-1}}{2\Delta t}} + u\frac{\emptyset_{j+1}^{n} - \emptyset_{j-1}^{n}}{2\Delta x} = \boxed{K\frac{\emptyset_{j+1}^{n-1} - 2\emptyset_j^{n-1} + \emptyset_{j-1}^{n-1}}{\Delta x^2}}$$

Advecção-difusão

Considere um processo de advecção-difusão

$$\frac{\partial \phi}{\partial t} + u\frac{\partial \phi}{\partial x} = K\frac{\partial^2 \phi}{\partial^2 x}; \qquad onde\ \ K = cte \qquad (52)$$

Temos visto que leapfrog é condicionalmente estável para advecção e a aproximação forward é condicionalmente estável para a difusão

Pode-se utilizar a combinação do esquema forward para Difusão e leapfrog para advecção

$$\frac{\emptyset_j^{n+1} - \emptyset_j^{n}}{\Delta t} = K\frac{\emptyset_{j+1}^{n-1} - 2\emptyset_j^{n-1} + \emptyset_{j-1}^{n-1}}{\Delta x^2}$$

regime é condicionalmente estável

$$\frac{\emptyset_j^{n+1} - \emptyset_j^{n-1}}{2\Delta t} - u\frac{\emptyset_{j+1}^{n} - \emptyset_{j-1}^{n}}{2\Delta x}$$

regime é condicionalmente estável

Advecção-difusão

Para a difusão, use o passo de tempo $2\Delta t$, Portanto:

$$\frac{\emptyset_j^{n+1} - \emptyset_j^{n-1}}{2\Delta t} + u\frac{\emptyset_{j+1}^{n} - \emptyset_{j-1}^{n}}{2\Delta x} = K\frac{\emptyset_{j+1}^{n-1} - 2\emptyset_j^{n-1} + \emptyset_{j-1}^{n-1}}{\Delta x^2}$$

NB. O termo de difusão é calculado com base em um passo de tempo anterior em relação ao outros termos.

Métodos de diferenças finitas.

$$\frac{\emptyset_j^{n+1} - \emptyset_j^{n-1}}{2\Delta t} = -u\frac{\emptyset_{j+1}^{n} - \emptyset_{j-1}^{n}}{2\Delta x} + K\frac{\emptyset_{j+1}^{n-1} - 2\emptyset_j^{n-1} + \emptyset_{j-1}^{n-1}}{\Delta x^2}$$

$$\emptyset_j^{n+1} - \emptyset_j^{n-1} = -u\frac{2\Delta t}{2\Delta x}\left(\emptyset_{j+1}^{n} - \emptyset_{j-1}^{n}\right) + \frac{2\Delta t K}{\Delta x^2}\left(\emptyset_{j+1}^{n-1} - 2\emptyset_j^{n-1} + \emptyset_{j-1}^{n-1}\right)$$

$$\emptyset_j^{n+1} = \emptyset_j^{n-1} - u\frac{2\Delta t}{2\Delta x}\left(\emptyset_{j+1}^{n} - \emptyset_{j-1}^{n}\right) + \frac{2\Delta t K}{\Delta x^2}\left(\emptyset_{j+1}^{n-1} - 2\emptyset_j^{n-1} + \emptyset_{j-1}^{n-1}\right)$$

Exercicio:

Construa um programa baseado para a equação de Advecção-difusão e integre a equação . Discuta o resultado. Compara com o método de RageKutta de 4 ordem

Advecção-difusão

Considere um processo de advecção-difusão

$$\frac{\partial \phi}{\partial t} + u\frac{\partial \phi}{\partial x} = K\frac{\partial^2 \phi}{\partial^2 x}; \qquad onde\ \ K = cte \tag{52}$$

Temos visto que leapfrog (tempo) é condicionalmente estável para advecção e a aproximação forward (tempo) é condicionalmente estável para a difusão

Pode-se utilizar a combinação do esquema forward para Difusão e leapfrog para advecção

$$\frac{\emptyset_j^{n+1} - \emptyset_j^{n}}{\Delta t} = K\frac{\emptyset_{j+1}^{n+1} - 2\emptyset_j^{n+1} + \emptyset_{j-1}^{n+1}}{\Delta x^2}$$

regime é condicionalmente estável

$$\frac{\emptyset_j^{n+1} - \emptyset_j^{n-1}}{2\Delta t} - u\frac{\emptyset_{j+1}^{n+1} - \emptyset_{j-1}^{n+1}}{2\Delta x}$$

regime é condicionalmente estável

Advecção-difusão

Para a difusão, use o passo de tempo $2\Delta t$, Portanto:

$$\frac{\emptyset_j^{n+1} - \emptyset_j^{n-1}}{2\Delta t} + u\frac{\emptyset_{j+1}^{n+} - \emptyset_{j-1}^{n+}}{2\Delta x} = K\frac{\emptyset_{j+1}^{n+1} - 2\emptyset_j^{n+1} + \emptyset_{j-1}^{n+1}}{\Delta x^2}$$

NB. O termo de difusão é calculado com base em um passo de tempo anterior em relação ao outros termos.

$$\frac{\emptyset_j^{n+1} - \emptyset_j^{n}}{\Delta t} = K\frac{\emptyset_{j+1}^{n+1} - 2\emptyset_j^{n+1} + \emptyset_{j-1}^{n+1}}{\Delta x^2}$$

$$\frac{\emptyset_j^{n+1} - \emptyset_j^{n-1}}{2\Delta t} - u\frac{\emptyset_{j+1}^{n+1} - \emptyset_{j-1}^{n+1}}{2\Delta x}$$

Métodos de diferenças finitas.

$$\frac{\widehat{\emptyset}_j^{n+1} - \emptyset_j^{n-1}}{2\Delta t} = -u\frac{\emptyset_{j+1}^{n+1} - \emptyset_{j-1}^{n+1}}{2\Delta x} + TermoDifusão^{n+1}$$

$$TermoDifusão^{n+1} = K\frac{\emptyset_{j+1}^{n+1} - 2\emptyset_j^{n+1} + \emptyset_{j-1}^{n+1}}{\Delta x^2}$$

$$\widehat{\emptyset}_j^{n+1} - \emptyset_j^{n-1} = -u\frac{2\Delta t}{2\Delta x}\left(\emptyset_{j+1}^{n+1} - \emptyset_{j-1}^{n+1}\right) + 2\Delta t * TermoDifusão^{n+1}$$

$$\widehat{\emptyset}_j^{n+1} = \emptyset_j^{n-1} - u\frac{2\Delta t}{2\Delta x}\left(\emptyset_{j+1}^{n+1} - \emptyset_{j-1}^{n+1}\right) + 2\Delta t * TermoDifusão^{n+1}$$

Exercicio:

Construa um programa baseado para a equação de Advecção-difusão e integre a equação . Discuta o resultado. Compara com o método de RageKutta de 4 ordem