

Chapter 2: Numerical Methods

For wave equations in geophysical fluid dynamics

Xinyu Wen

Dept of Atmospheric & Oceanic Sciences
Peking University

Outline

1 气候模式与平流方程

2 时间积分

3 有限差分方法

4 谱方法

5 有限体积法

6 谱元法

7 半拉格朗日法

8 数值计算方法之比较

1 气候模式与平流方程

- What is Climate Model?
- Advection equation
- Implication of advection equation

2 时间积分

3 有限差分方法

4 谱方法

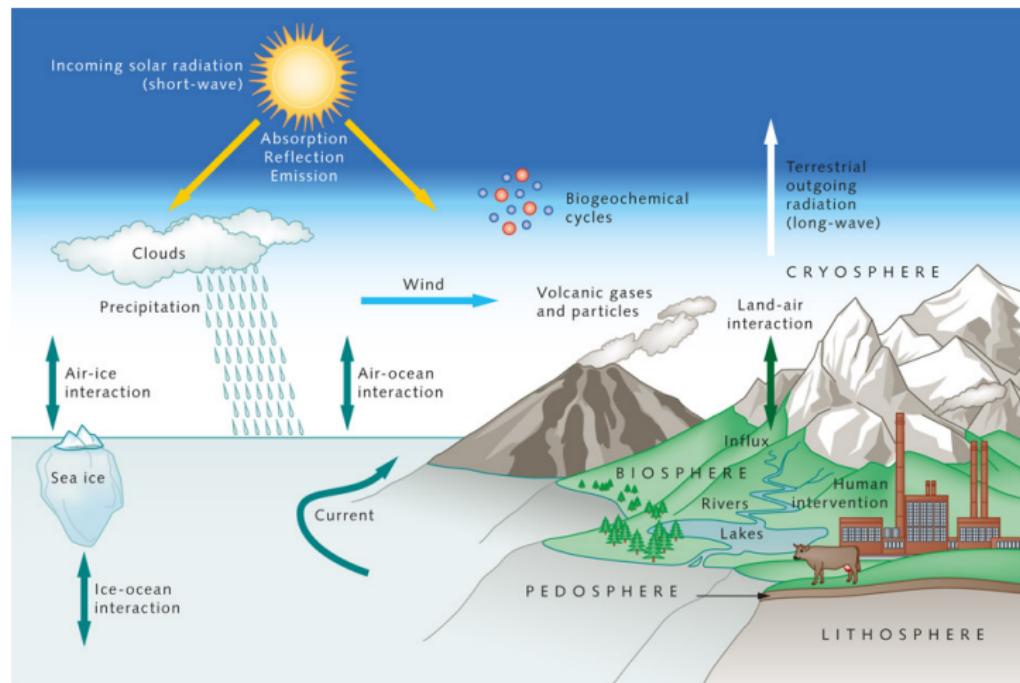
5 有限体积法

6 谱元法

7 半拉格朗日法

8 数值计算方法之比较

气候模式的基本图景



气候模式是什么？

从物理过程的角度看气候模式

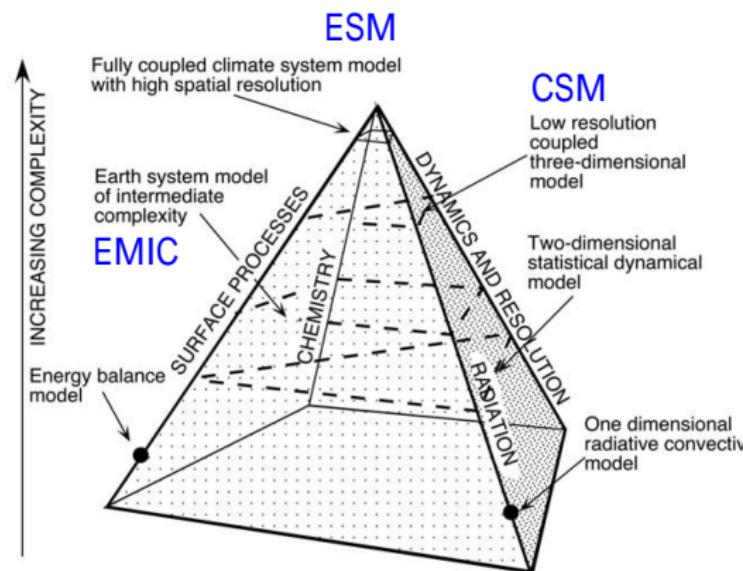


Figure: McGuffie and Henderson-Sellers, 2005

气候模式是什么？

从复杂维度的角度看气候模式

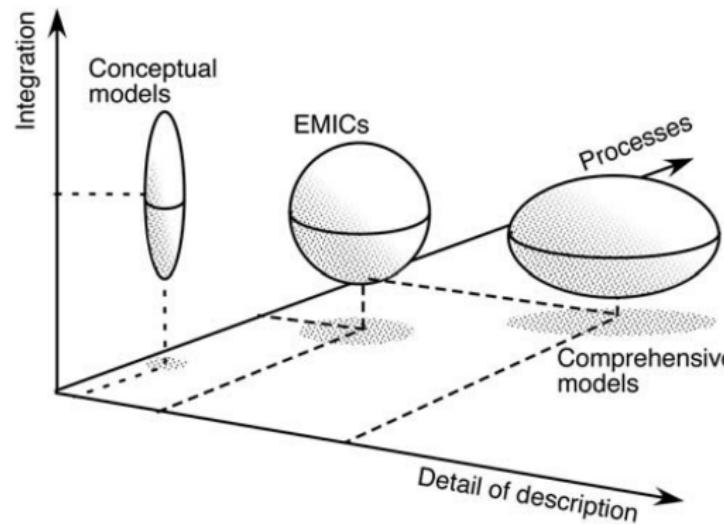


Figure: Claussen, 2002

气候模式的抽象形式

大气与海洋模式的最抽象形式

$$\frac{dx_i}{dt} = F(X) \quad (1)$$

其中 x_i 是某个状态变量，而由多个状态变量可构成能够代表整个系统的状态向量 $X = (x_1, x_2, x_3, \dots, x_N)$

$$F(X) = Dyn(X) + Phy(X) + Diff(X) \quad (2)$$

$Dyn(X)$ 是描述动力过程的显式方程

$Phy(X)$ 是次网格物理过程，采用参数化方式描述

$Diff(X)$ 是数值耗散，有的有物理基础，有的只为计算稳定

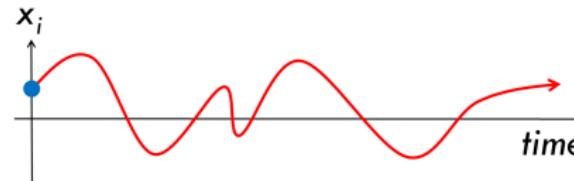
双曲线方程

1st-order Hyperbolic Equation

$$\frac{dx_i}{dt} = F(X) \quad (3)$$

$$\frac{\partial x_i}{\partial t} + \nabla \cdot \vec{v} x_i = F(X) \quad (4)$$

$$\frac{\partial x_i}{\partial t} + u \frac{\partial x_i}{\partial x} + v \frac{\partial x_i}{\partial y} + w \frac{\partial x_i}{\partial z} = F(X) \quad (5)$$



平流方程

Advection Equation

假设被平流物是 N , 定常平流流场是 $\vec{v} = (c_x, c_y, c_z)$

$$\frac{\partial N}{\partial t} + c_x \frac{\partial N}{\partial x} = 0 \quad (6)$$

$$\frac{\partial N}{\partial t} + c_x \frac{\partial N}{\partial x} + c_y \frac{\partial N}{\partial y} = 0 \quad (7)$$

$$\frac{\partial N}{\partial t} + c_x \frac{\partial N}{\partial x} + c_y \frac{\partial N}{\partial y} + c_z \frac{\partial N}{\partial z} = 0 \quad (8)$$

- 定常平流方程描述了熵守恒的平流过程
- 解为被平流物随平流流场向下游匀速平移的行波解

平流的物理意义

Table: 平流、湍流、分子扩散的对比

	平流	湍流	分子扩散
水平尺度	$10^3 km$	$1 km$	$0.1 \rightarrow 10 m$
垂直尺度	$10 km$	$1 km$	$0.1 \rightarrow 10 m$
主要维度	水平	垂直	各向同性
位置	PBL 之上	PBL 以内	梯度大的地方
量化指标	$v = 0.1 \rightarrow 10 m/s$	$K = 1 \rightarrow 10 m^2/s$	$\nu = 10^{-5} m^2/s$

平流方程在地球流体中的重要地位

常见的地球流体力学中出现的方程：

- 动量方程：“风速”的平流
- 热力学方程：“温度”的平流
- 水汽方程：“水汽”的平流
- 盐度方程：“盐度”的平流
- 化学预报方程：“化学物种”的平流
- TKE 方程：“湍流动能”的平流

平流过程的图景



Figure: 2012-10-6, 俄罗斯 Shiveluch 火山喷发 (MODIS/Aqua)

平流过程的图景

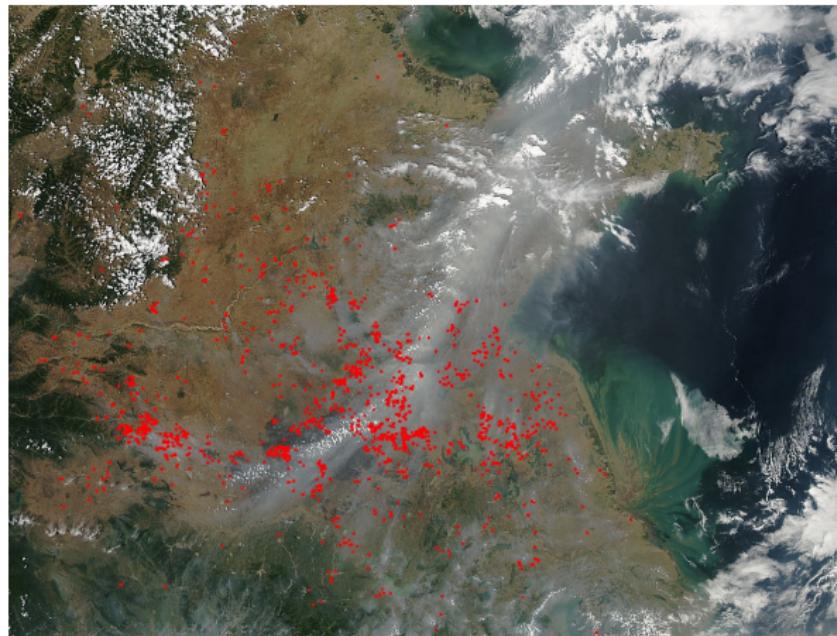


Figure: 2012-6-12, 中国东部野火, 河南、河北、山东、江苏、安徽进行
冬小麦收获 (MODIS/Aqua)

平流过程的图景

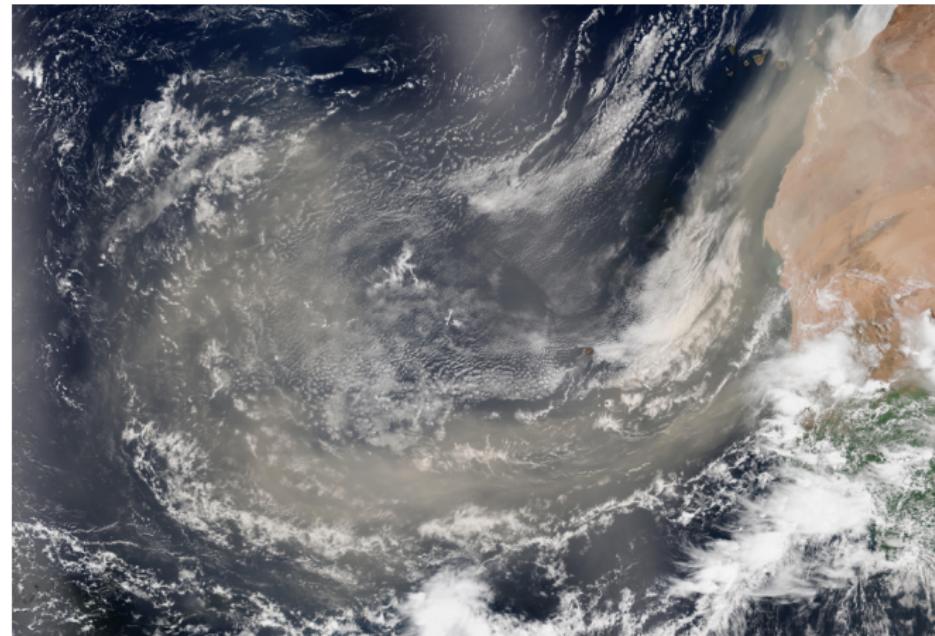


Figure: 2013-9, 非洲撒哈拉沙漠沙尘暴飘入大西洋 (MODIS)

1 气候模式与平流方程

2 时间积分

- Time-differencing picture
- One-stage method
- Multi-stage method

3 有限差分方法

4 谱方法

5 有限体积法

6 谱元法

7 半拉格朗日法

8 数值计算方法之比较

如果你是盲人...



Figure: 起点? 方向? 迈多大步?

方向：永远是个问题

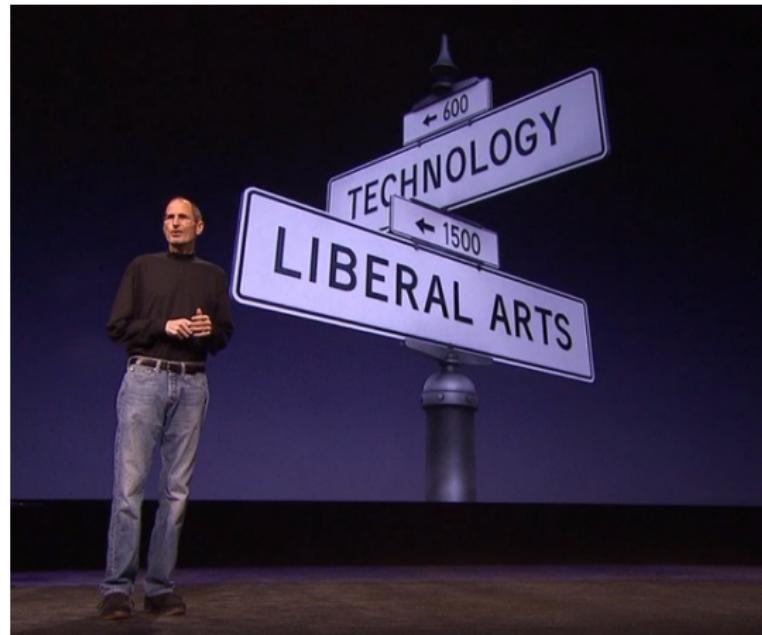


Figure: 当老乔来到十字路口时，他也曾艰难地徘徊

时间积分：决定起点、决定方向、决定迈步大小

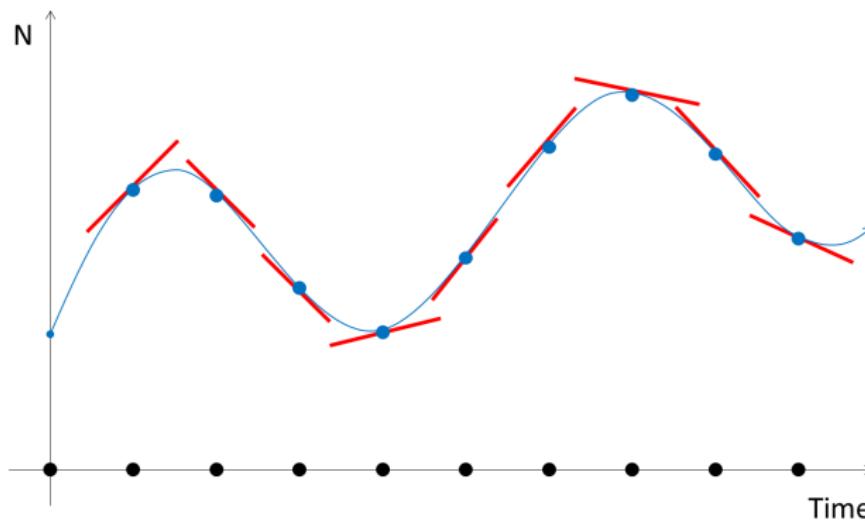


Figure: 即使斜率完全准确，“向前迈步”仍然是个难题

单时间片方案

只用一个时间片 t^n 的 X , 即可算出下一个时间片 t^{n+1} 的 X

- Forward Euler scheme
- Matsuno scheme
松野
- Heun scheme
- Runge-Kutta scheme family

Forward Euler Scheme (order=1)

算法

$$X^{n+1} = X^n + dt \cdot F(X^n)$$

Fortran 95 Code

```
1 subroutine forward(x0,x1,N)
2   implicit none
3   ! interface
4   integer, intent(in) :: N
5   real, dimension(N), intent(in) :: x0
6   real, dimension(N), intent(out) :: x1
7   ! local variables
8   real, dimension(N) :: xtend
9   call tend(x0,xtend,N)
10  x1 = x0 + dt * xtend
11 end subroutine
```

Matsuno Scheme (order=1 or 1.5)

算法

$$X^{est} = X^n + dt \cdot F(X^n)$$

$$X^{n+1} = X^n + dt \cdot F(X^{est})$$

Fortran 95 Code

```
1 subroutine matsuno(x0,x1,N)
2   implicit none
3   ! interface
4   integer, intent(in) :: N
5   real, dimension(N), intent(in) :: x0
6   real, dimension(N), intent(out) :: x1
7   ! local variables
8   real, dimension(N) :: xest, xtend
9   call tend(x0,xtend,N)
10  xtend = x0 + dt * xtend
11  call tend(xest,xtend,N)
12  x1    = x0 + dt * xtend
13 end subroutine
```

Heun Scheme (order=2)

算法

$$X^{est} = X^n + dt \cdot F(X^n)$$

$$X^{n+1} = X^n + \frac{1}{2} dt \cdot [F(X^n) + F(X^{est})]$$

Fortran 95 Code

```

1 subroutine heun(x0,x1,N)
2   implicit none
3   ! interface
4   integer, intent(in) :: N
5   real, dimension(N), intent(in) :: x0
6   real, dimension(N), intent(out) :: x1
7   ! local variables
8   real, dimension(N) :: xtend0, xest xtendest
9   call tend(x0,xtend0,N)
10  xest = x0 + dt * xtend0
11  call tend(xest,xtendest,N)
12  x1 = x0 + 0.5 * dt * ( xtend0 + xtendest )
13 end subroutine

```

Runge-Kutta Scheme Family

算法 (order=2), 亦称 mid-point Runge-Kutta

$$q_1 = dt \cdot F(X^n)$$

$$q_2 = dt \cdot F(X^n + q_1) - q_1$$

$$X^{n+1} = X^n + q_1 + \frac{1}{2}q_2$$

Runge-Kutta Scheme Family

算法 (order=3)

$$q_1 = dt \cdot F(X^n)$$

$$q_2 = dt \cdot F\left(X^n + \frac{1}{3}q_1\right) - \frac{5}{9}q_1$$

$$q_3 = dt \cdot F\left(X^n + \frac{15}{16}q_2\right) - \frac{153}{128}q_2$$

$$X^{n+1} = X^n + \frac{15}{16}q_1 + \frac{8}{15}q_2$$

Runge-Kutta Scheme Family

算法 (order=4)

$$q_1 = dt \cdot F(X^n)$$

$$q_2 = dt \cdot F\left(X^n + \frac{1}{2}q_1\right)$$

$$q_3 = dt \cdot F\left(X^n + \frac{1}{2}q_2\right)$$

$$q_4 = dt \cdot F(X^n + q_3)$$

$$X^{n+1} = X^n + \frac{1}{6}(q_1 + 2q_2 + 2q_3 + q_4)$$

以 Runge-Kutta (order=4) 为例

Fortran 95 Code

```
1 subroutine rk4(x0,x1,N)
2   implicit none
3   ! interface
4   integer, intent(in) :: N
5   real, dimension(N), intent(in) :: x0
6   real, dimension(N), intent(out) :: x1
7   ! local variables
8   real, dimension(N) :: xtend, q1, q2, q3, q4
9
10  call tend(x0,xtend,N)
11  q1 = dt * xtend
12
13  call tend(x0+0.5*q1,xtend,N)
14  q2 = dt * xtend
15
16  call tend(x0+0.5*q2,xtend,N)
17  q3 = dt * xtend
18
19  call tend(x0+q3,xtend,N)
20  q4 = dt * xtend
21
22  x1 = x0 + 1.0/6.0 * ( q1 + 2*q2 + 2*q3 + q4 )
23 end subroutine
```

多时间片方案

使用当前 t^n 及过去若干时间片的 X , 才可算出下一个时间片 t^{n+1} 的 X

- Leapfrog scheme
- Adams-Bashforth-Matsuno Predictor-Corrector scheme
- Adams-Bashforth scheme family

Leapfrog Scheme (order=2)

蛙跃格式

算法

$$X^{n+1} = X^{n-1} + 2dt \cdot F(X^n)$$

Fortran 95 Code

```
1 subroutine forward(x0,x1,x2,N)
2   implicit none
3   ! interface
4   integer, intent(in) :: N
5   real, dimension(N), intent(in) :: x0, x1
6   real, dimension(N), intent(out) :: x2
7   ! local variables
8   real, dimension(N) :: xtend
9   call tend(x1,xtend,N)
10  x2 = x0 + 2.0 * dt * xtend
11 end subroutine
```

ABM Predictor-Corrector Scheme (order=3)

算法

$$X^{est} = X^n + \frac{1}{2} dt \cdot [3F(X^n) - F(X^{n-1})]$$

$$X^{n+1} = X^n + \frac{1}{12} dt \cdot [5F(X^{est}) + 8F(X^n) - F(X^{n-1})]$$

Fortran 95 Code

```

1 subroutine abm(x0,x1,x2,N)
2   implicit none
3   ! interface
4   integer, intent(in) :: N
5   real, dimension(N), intent(in) :: x0, x1
6   real, dimension(N), intent(out) :: x2
7   ! local variables
8   real, dimension(N) :: xtend0, xtend1, xest xtendest
9   call tend(x0,xtend0,N)
10  call tend(x1,xtend1,N)
11  xest = x1 + 0.5 * dt * (3*xtend1-xtend0)
12  call tend(xest,xtendest,N)
13  x2 = x1 + 1.0/12.0 * dt * ( 5*xtendest + 8*xtend1 - xtend0 )
14 end subroutine

```

Adams-Basforth Scheme Family

算法 (order=2, 3, 4)

$$X^{n+1} = X^n + \frac{1}{2} dt \cdot [3F(X^n) - F(X^{n-1})]$$

$$X^{n+1} = X^n + \frac{1}{12} dt \cdot [23F(X^n) - 16F(X^{n-1}) + 5F(X^{n-2})]$$

$$X^{n+1} = X^n + \frac{1}{24} dt \cdot [55F(X^n) - 59F(X^{n-1}) + 37F(X^{n-2}) - 9F(X^{n-3})]$$

以 Adams-Bashforth (order=4) 为例

Fortran 95 Code

```
1 subroutine ab4(x0,x1,x2,x3,x4,N)
2   implicit none
3   ! interface
4   integer, intent(in) :: N
5   real, dimension(N), intent(in) :: x0, x1, x2, x3
6   real, dimension(N), intent(out) :: x4
7   ! local variables
8   real, dimension(N) :: xtend0, xtend1, xtend2, xtend3
9   call tend(x0,xtend0,N)
10  call tend(x1,xtend1,N)
11  call tend(x2,xtend2,N)
12  call tend(x3,xtend3,N)
13  x4 = x3 + 1.0/24.0 * dt * ( 55*xtend3 - 59*xtend2 + 37*xtend1 - 9*xtend0 )
14 end subroutine
```

1 气候模式与平流方程

2 时间积分

3 有限差分方法

- Space-Differencing basics
- Demo
- Numerical errors

4 谱方法

5 有限体积法

6 谱元法

7 半拉格朗日法

8 数值计算方法之比较

空间偏微分方程的基本称谓

n^{th} -order n 阶; m^{th} -degree m 次

$$\left(\frac{\partial^n N}{\partial x^n} \right)^m$$

Examples

$$\frac{\partial N}{\partial t} = -c \frac{\partial N}{\partial x} \quad (9)$$

$$\frac{\partial N}{\partial t} = -c_x \frac{\partial N}{\partial x} - c_y \frac{\partial N}{\partial y} \quad (10)$$

$$\frac{\partial N}{\partial t} = -c \frac{\partial N}{\partial x} + k \frac{\partial^2 N}{\partial N^2} \quad (11)$$

常见空间偏微分的差分算法

一阶精度 $O(\Delta)$

- $\frac{\partial N}{\partial x}$
 - $Forward = \frac{N_1 - N_0}{dx}$
 - $Backward = \frac{N_0 - N_{-1}}{dx}$
- $\frac{\partial^2 N}{\partial x^2}$ N/A
- $\frac{\partial^3 N}{\partial x^3}$ N/A
- $\frac{\partial^4 N}{\partial x^4}$ N/A

常见空间偏微分的差分算法

二阶精度 $O(\Delta^2)$

- $\frac{\partial N}{\partial x} = \frac{N_1 - N_{-1}}{2dx}$
- $\frac{\partial^2 N}{\partial x^2} = \frac{N_1 - 2N_0 + N_{-1}}{dx^2}$
- $\frac{\partial^3 N}{\partial x^3} = \frac{N_2 - 2N_1 + 2N_{-1} - N_{-2}}{2dx^3}$
- $\frac{\partial^4 N}{\partial x^4} = \frac{N_2 - 4N_1 + 6N_0 - 4N_{-1} + N_{-2}}{dx^4}$

常见空间偏微分的差分算法

四阶精度 $O(\Delta^4)$

- $\frac{\partial N}{\partial x} = \frac{-N_2 + 8N_1 - 8N_{-1} + N_{-2}}{12dx}$
- $\frac{\partial^2 N}{\partial x^2} = \frac{-N_2 + 16N_1 - 30N_0 + 16N_{-1} - N_{-2}}{12dx^2}$
- $\frac{\partial^3 N}{\partial x^3} = \frac{-N_3 + 8N_2 - 13N_1 + 13N_{-1} - 8N_{-2} + N_{-3}}{8dx^3}$
- $\frac{\partial^4 N}{\partial x^4} = \frac{-N_3 + 12N_2 - 39N_1 + 56N_0 - 39N_{-1} + 12N_{-2} - N_{-3}}{6dx^4}$

举例

在 $x \in [0 \rightarrow 2\pi]$ 范围，假设原始的 N 的分布是

$$N(x) = \sin(x)$$

那么 N 的空间偏导数的解析解是

$$\frac{\partial N}{\partial x} = \cos(x)$$

而如果采用离散形式，将 $x \in [0 \rightarrow 2\pi]$ 分解为 N_x 个格点，则数值解与解析解的差别有多大？请试用多种差分方法尝试回答。

Fortran Code 1/2

```
1  program main
2
3      implicit none
4
5      real , parameter :: PI = 3.1415926
6      integer , parameter :: N = 16           ! number of points
7      real , dimension(N) :: x,sinx,cosx
8      integer , dimension(N) :: pr, pl       ! pointer
9
10     integer :: i
11     real :: d
12     real :: backward1, forward1, central2, central4
13     real :: bw1,fw1,cen2,cen4
14
15     !— init —
16     d      = 2.0*PI/real(N)
17     do i   = 1, N
18         x(i)   = (real(i)-0.5)*d
19         sinx(i) = sin(x(i))
20         cosx(i) = cos(x(i))
21         pl(i)   = i-1
22         pr(i)   = i+1
23     end do
24     pl(1) = N
25     pr(N) = 1
```

Fortran Code 2/2

```
27      !—  $d(\sin x)/dx = \cos x$ —
28      do i      = 1, N
29
30          backward1    = (sinx(i) - sinx(pi(i)))/d
31          forward1     = (sinx(pr(i)) - sinx(i))/d
32          central2     = (sinx(pr(i)) - sinx(pi(i)))/(2.0*d)
33          central4     = (sinx(pi(pi(i)))) - 8.0*sinx(pi(i)) + 8.0*sinx(pr(i)) \\
34                                - sinx(pr(pr(i)))/(12.0*d)
35
36          bw1           = backward1 - cosx(i)
37          fw1           = forward1 - cosx(i)
38          cen2           = central2 - cosx(i)
39          cen4           = central4 - cosx(i)
40
41      ! 输出绝对值
42      !print "(7f10.4)", x(i), sinx(i), cosx(i), backward1, forward1, \\
43                           central2, central4
44      ! 输出差异值
45      print "(7f10.4)", x(i), sinx(i), cosx(i), bw1, fw1, cen2, cen4
46
47      end do
48
49 end program
```

Compiling

```
1  #!/bin/tcsh
2  # Compile script
3  # Xinyu Wen, Peking Univ, Mar 2015
4
5  gfortran -o run main.f90
6  ./run
7  ./run > result
8
9  #gnuplot draw-absvalue.gnuplot
10 # OR
11 gnuplot draw-diff.gnuplot
```

Plotting

```
1 # Plotting script
2 # Xinyu Wen, Peking Univ, Mar 2015
3
4 set term post eps color solid enh
5 set output "diff.eps"
6
7 set xlabel "x"
8 set ylabel "sin(x)"
9
10 set yrange [-0.5:0.5]
11 set xrange [0:6.28]
12
13 set title "d(sin(x))/dx"
14
15 plot "result" u 1:4 t "Backward" w l lw 4, "result" u 1:5 t "Forward" w l lw 4,
16     "result" u 1:6 t "Central-2" w l lw 4, "result" u 1:7 t "Central-4" w l lw 4
17
18 # in case you want to stop for a sec
19 #pause -1
```

Solution

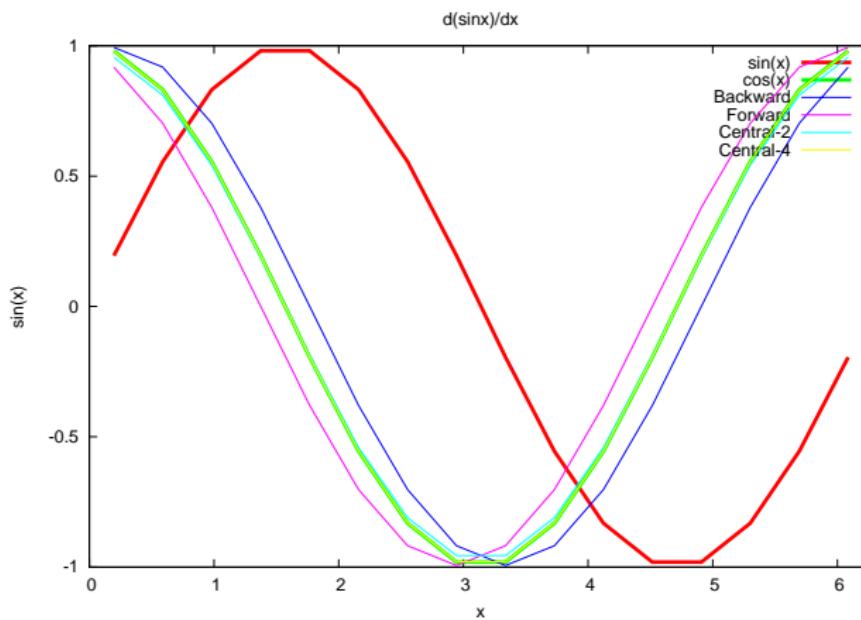


Figure: $Nx = 16$, Absolute values from diverse space-differencing schemes

Solution

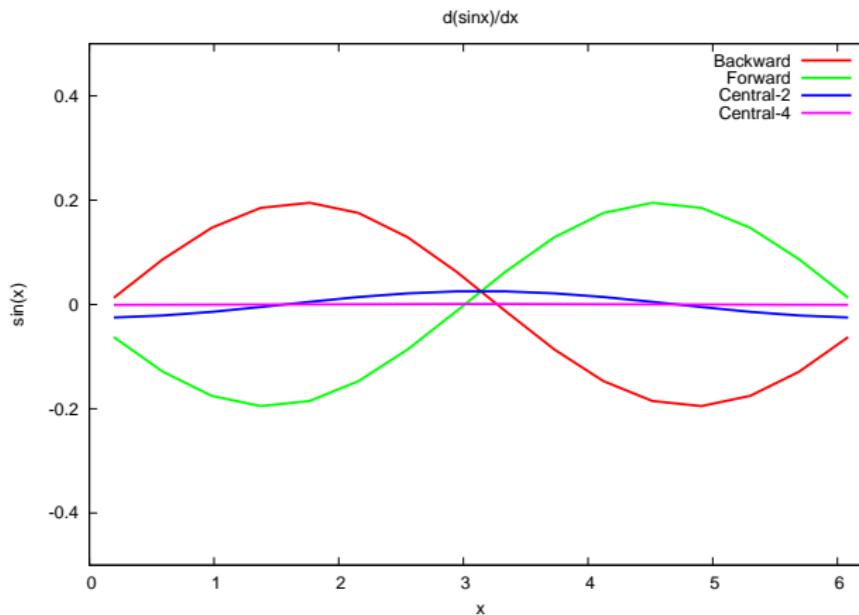


Figure: $Nx = 16$, Differences

Truncation error

截断误差

由于只保留了微分方程 Taylor 展式的有限项，而丢失了其高阶项所导致的误差。也称离散误差（Discretization error）。

从正方向在 x 点展开

$$f(x + dx) = f(x) + f'(x)dx + f''(x)\frac{dx^2}{2!} + f'''(x)\frac{dx^3}{3!} + f''''(x)\frac{dx^4}{4!} + \dots \quad (12)$$

从负方向在 x 点展开

$$f(x - dx) = f(x) - f'(x)dx + f''(x)\frac{dx^2}{2!} - f'''(x)\frac{dx^3}{3!} + f''''(x)\frac{dx^4}{4!} + \dots \quad (13)$$

Truncation error

截断误差

如果用12减去13，并把 $f'(x)$ 移到左边，其余移到右边

$$f'(x) = \frac{f(x+dx) - f(x-dx)}{2dx} - \frac{f'''(x)}{3!} \frac{dx^3}{3!} - \dots \quad (14)$$

这正是一阶导中央差公式，可见其截断误差是 $O(\Delta x^2)$

Truncation error

截断误差

如果用12加上13，并把 $f''(x)$ 移到左边，其余移到右边

$$f''(x) = \frac{f(x+dx) - 2f(x) + f(x-dx)}{dx^2} - \frac{2f'''(x)}{dx^2} \frac{dx^4}{4!} - \dots \quad (15)$$

这正是二阶导中央差公式，可见其截断误差是 $O(\Delta x^2)$

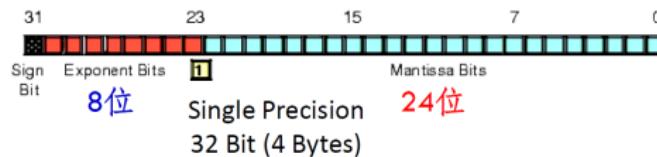
Round-off error

舍入误差

舍入误差

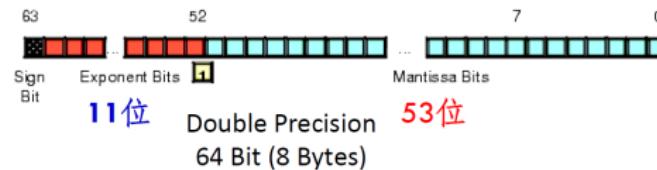
由于计算机的物理条件限制，在计算过程中对浮点数不能进行准确运算，通常只用有限个有效数字进行表达，有效数字之外的计算精度损失长期积累导致的误差，称为舍入误差，也称稳定性误差（Stability error）。

IEEE 754 Standard, 1985, 2008



尾数
 $\pm 2^{23}$
 ± 8388608
指数
 $2^{\pm 2^7}$
 $10^{\pm 38}$

精度 6-7 位



$\pm 2^{52}$
 ± 4503599627370496
精度 15-16 位

$2^{\pm 2^{10}}$
 $10^{\pm 308}$

Scales of The Universe

单精度 ($10^{\pm 38}$) 和双精度 ($10^{\pm 308}$) 够用吗? IEEE 怎么不更新一下?

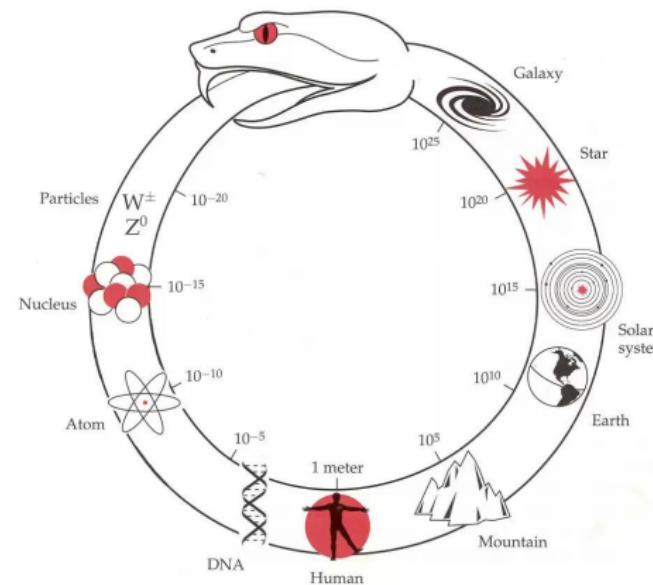


Figure: 我们宇宙的尺度只有 $10^{-30} \rightarrow 10^{30}$ 的差距

三种最常见的 Round-off error 情况

以单精度（32bit）为例

- 计算出界: 当计算超越 10^{38} 时
- 大吃小: 相差 7 个量级以上的两个数的加、减法
- 双胞胎相减: 相近到 7 个有效数字以上的两个数的减法

Demo 1: 计算机算得准确吗?

```
1 program sqrt2
2
3     implicit none
4
5     real :: a, b, c, x
6
7     a = sqrt(2.0)
8     b = sqrt(2.0)
9     c = a*b
10    x = 2.0
11
12    print "(3f20.10)", a, b, c
13    print "(1f20.10)", x
14
15 end program
```

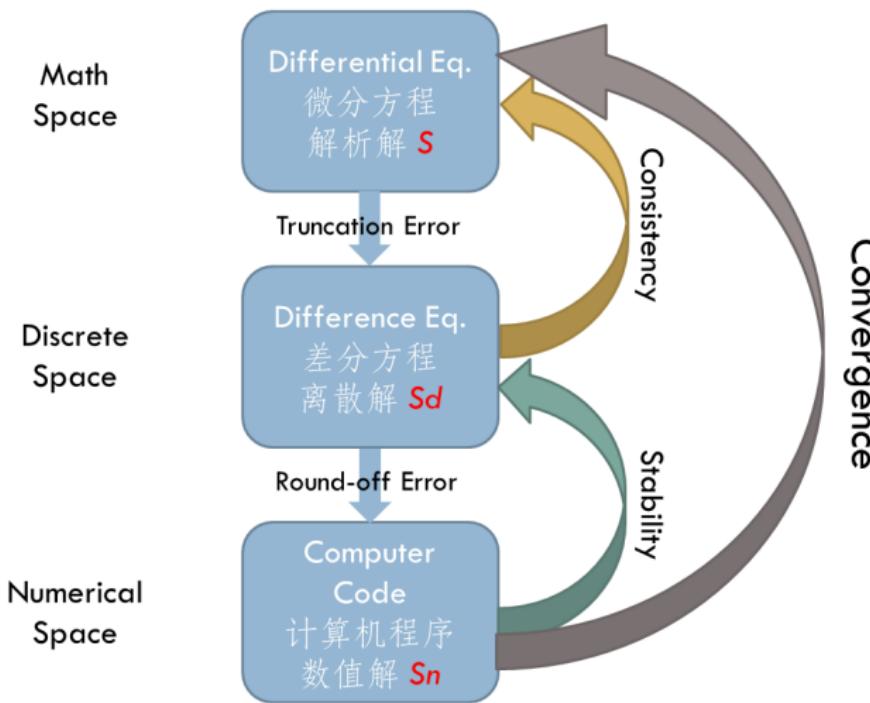
Demo 2: 大吃小?

```
1 program bigplussmall
2
3     implicit none
4
5     real :: big, small, plus, minus
6     integer :: i
7
8     big = 1.23456789
9     small = 0.00000001
10    plus = big + small
11    minus = big - small
12
13    print "(3f20.10)", big, small, plus
14    print "(3f20.10)", big, small, minus
15    print "(1f20.10)", 1.2345679
16    print *, "Another Test:"
17    print "(1f20.10)", plus+small*100000
18
19    do i = 1, 100000
20        plus = plus + small
21    end do
22
23    print "(1f20.10)", plus
24
25 end program
```

Demo 3: 双胞胎相减?

```
1 program smalldiff
2
3     implicit none
4
5     real :: a, b, diff
6
7     a = 1.2345678
8     b = 1.2345673
9     diff = a-b
10
11    print "(3f20.10)", a,b,diff
12
13 end program
```

数值解的误差来源



一点思考：两种误差的互相掣肘

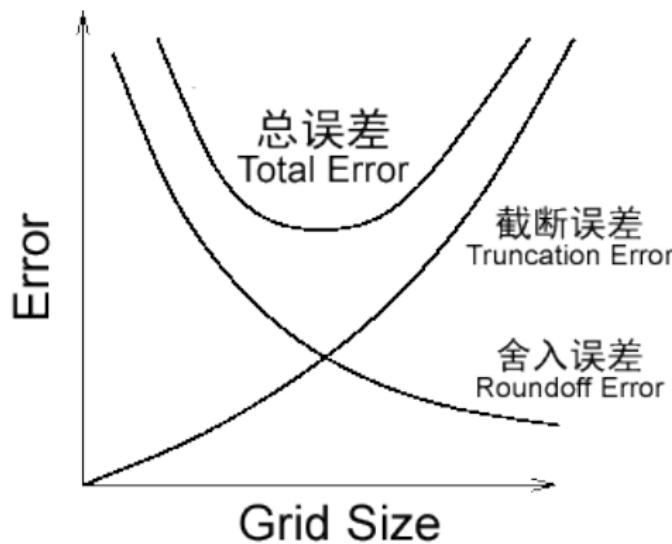


Figure: 更小的网格一定好吗?

1 气候模式与平流方程

2 时间积分

3 有限差分方法

4 谱方法

- History and Basic Concepts
- Discrete Fourier Transform
- Pesudo Spectral Method vs. Spectral Method
- Legendre polynomials and Associated Legendre polynomials
- Spherical harmonic transform

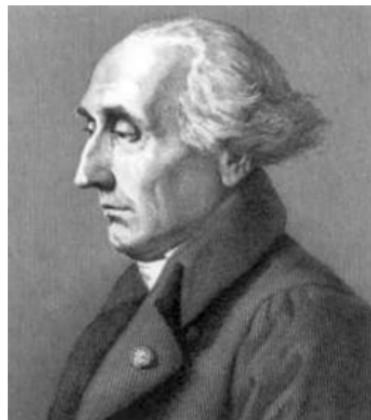
5 有限体积法

6 谱元法

7 半拉格朗日法

8 数值计算方法之比较

傅里叶的故事



Joseph Lagrange
1736-1813

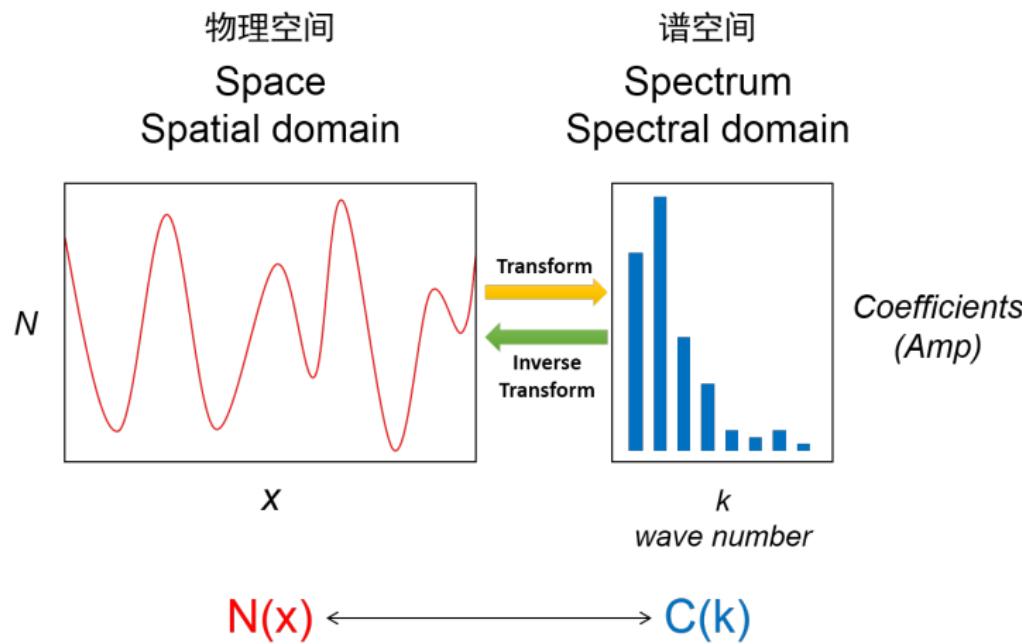


Joseph Fourier
1768-1830



Pierre-Simon Laplace
1749-1827

物理空间 vs. 谱空间



波数的概念

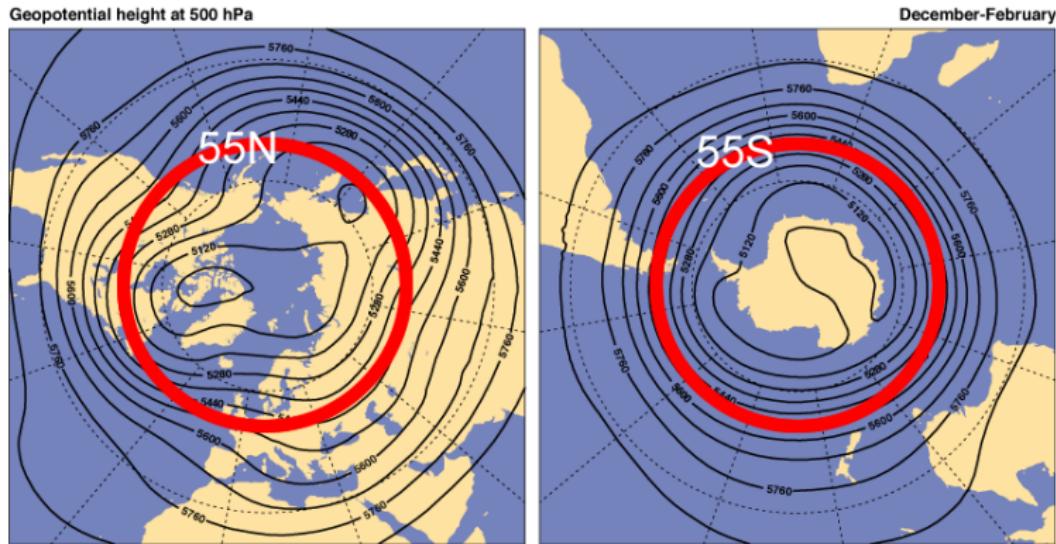


Figure: Climatological mean of DJF Z500 in ERA40

波数的概念

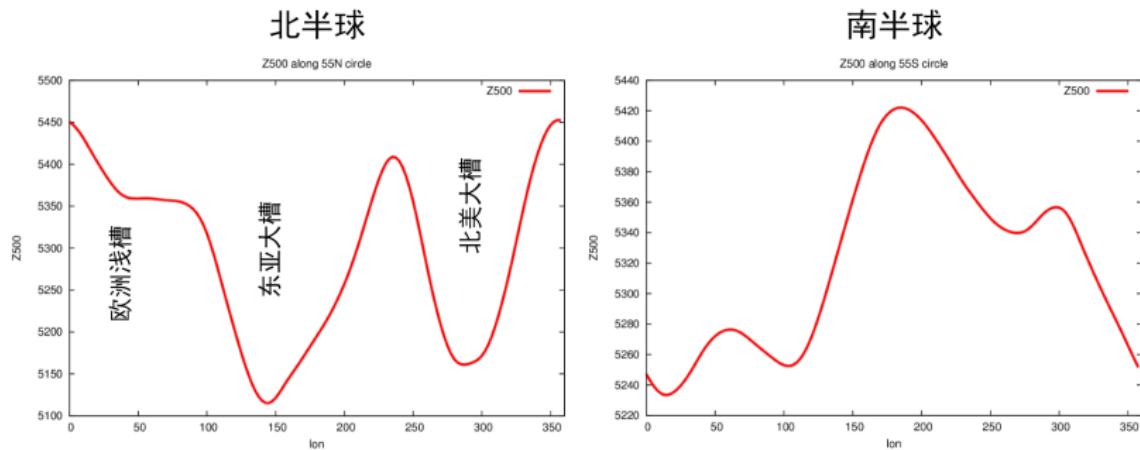


Figure: 沿 55N/55S 纬线的高度场 (DJF Z500 in ERA40)

波数的概念

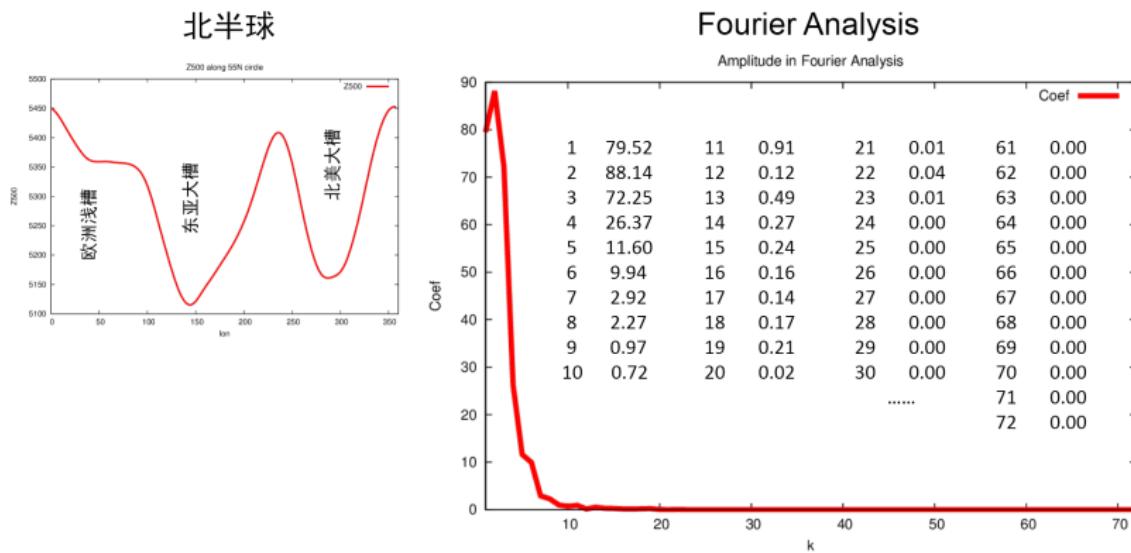


Figure: 谱空间系数的振幅（左图 \times 方向 144 个数，右图分解 1-72 波）

最大波数 k_{max}

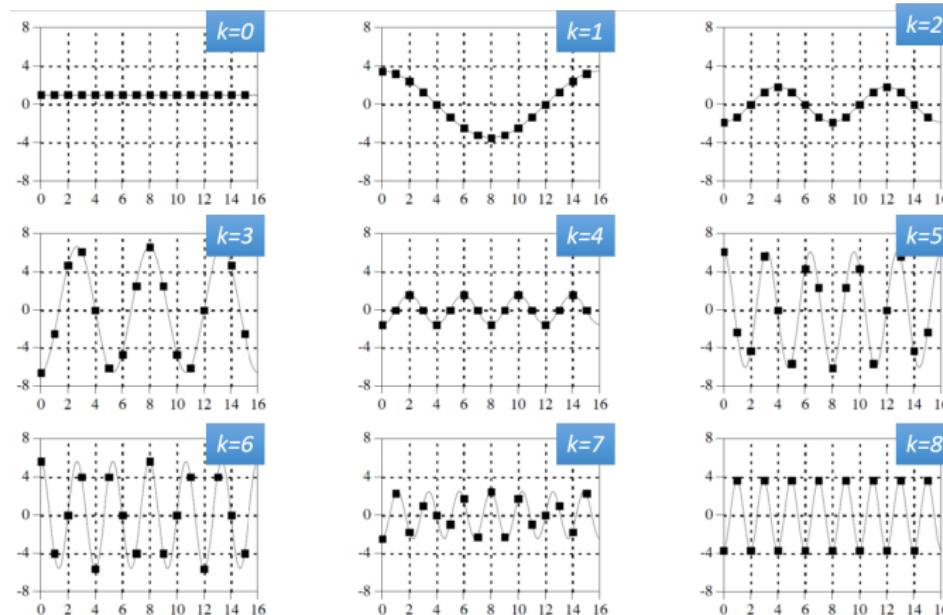


Figure: \times 方向离散点 $N = 16$, 可表达最大波数 $k_{max} = N/2 = 8$

Gibbs Phenomenon

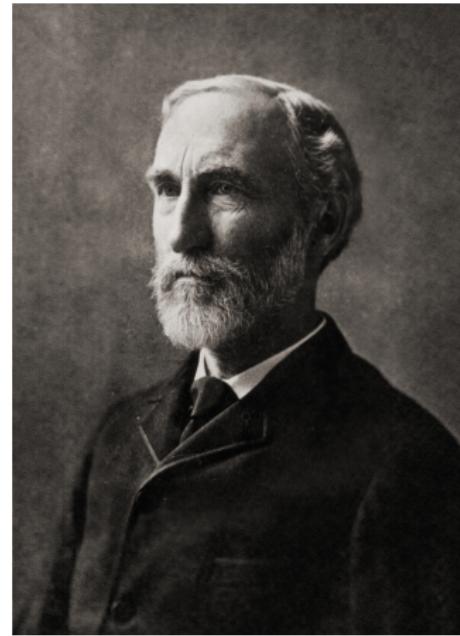


Figure: 美国数学家 Josiah W. Gibbs (1839-1903)

Gibbs Phenomenon

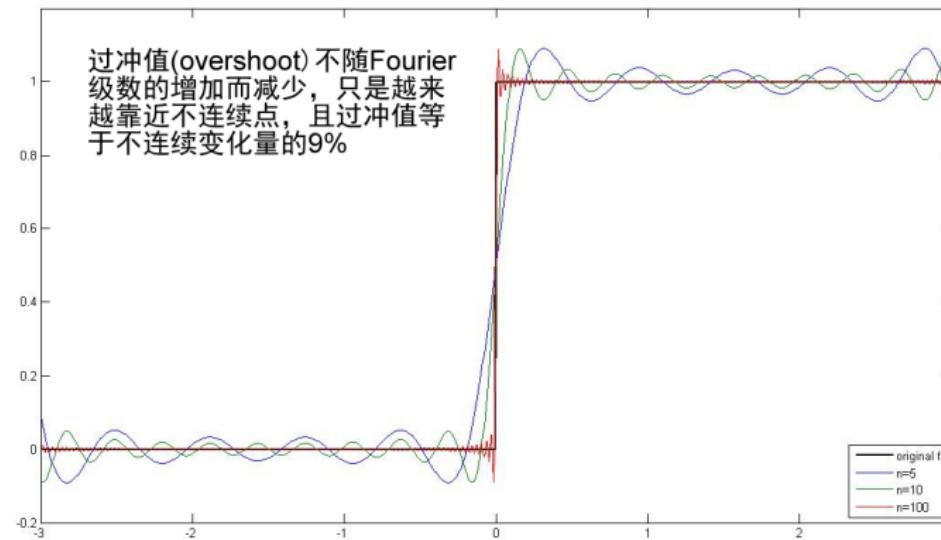


Figure: Gibbs 同学说: $\frac{\text{overshoot}}{\|\text{discontinuity gap}\|} = 0.089490\dots$

Gibbs Phenomenon

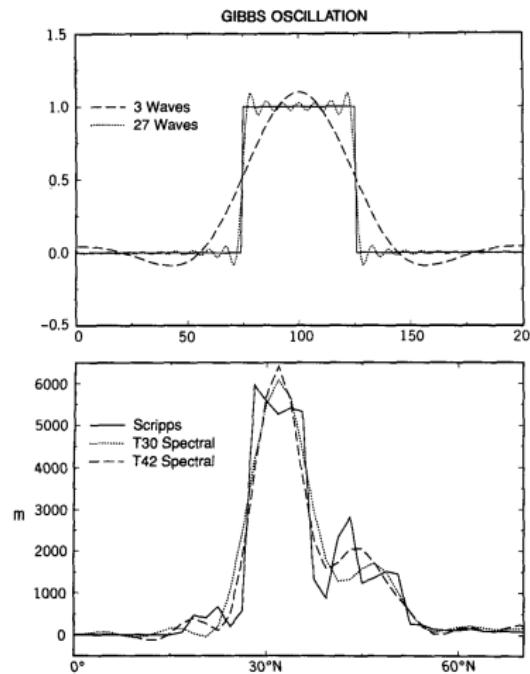


Figure: 下图为沿 90E 穿越喜马拉雅山脉的南北剖面 (Navarra, 1994)

Gibbs Phenomenon

影响常常体现在

- 地形：陡峭、突变处
- 水汽：负水汽问题
- 云：云边界处
- 参数化订正过大处

Gibbs Phenomenon

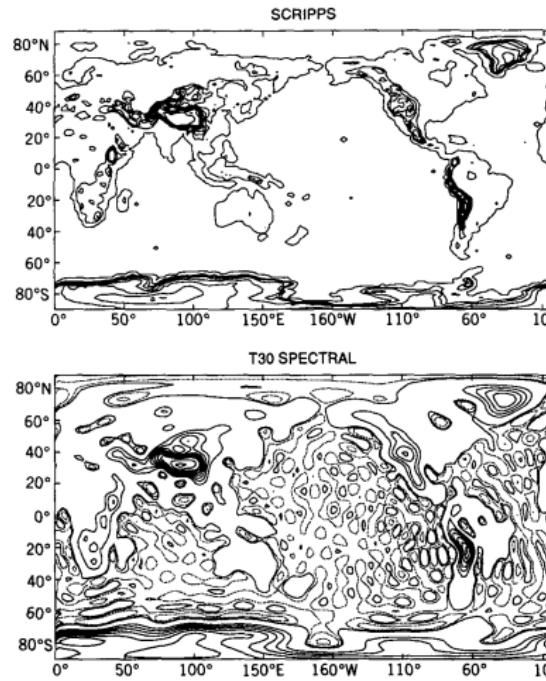


Figure: Scripps 格点模式与谱模式（T30）对地形的表达

Gibbs Phenomenon

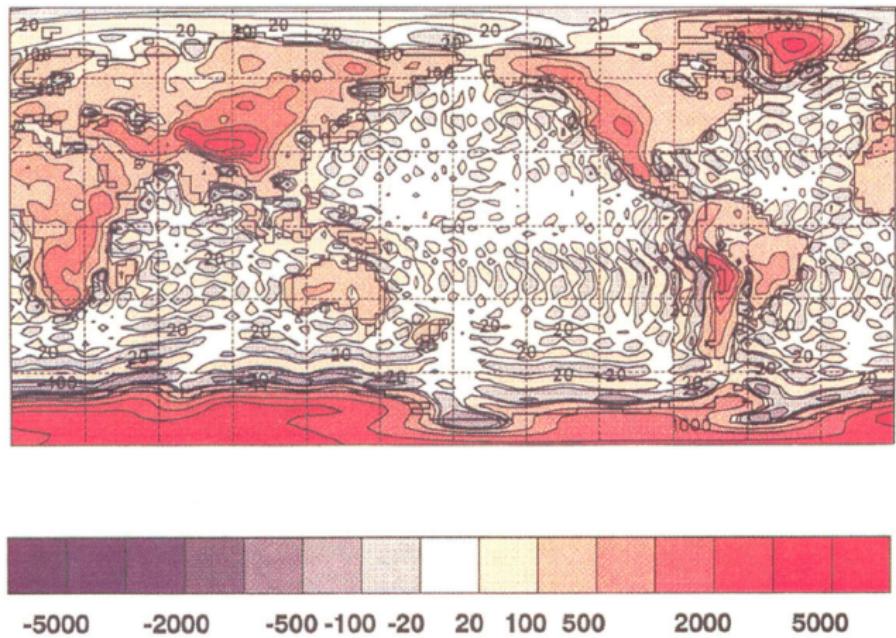


FIG. 2. R30 climate model surface elevation (m) computed using the standard procedure.

Figure: 菱形截断谱模式（R30）对地形的表达

Gibbs Phenomenon

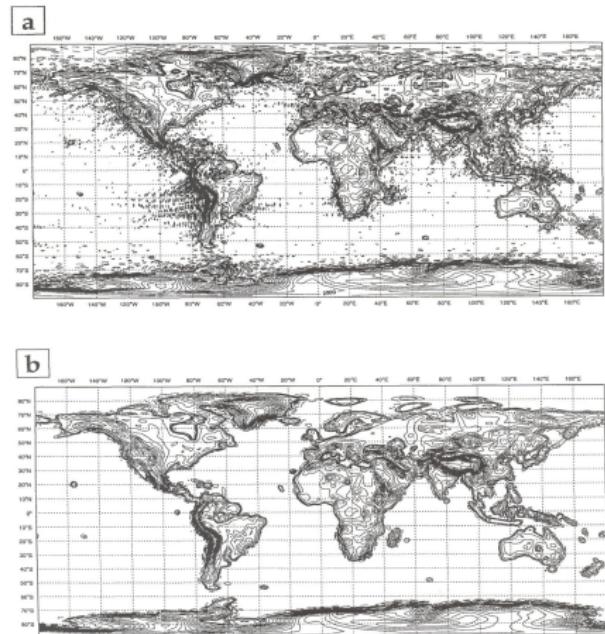


FIG. 3. Global representation of the orography in the T126 constant-resolution spectral model. (a) Classic spectral projection, (b) classic spectral projection smoothed by the filter introduced by Sardeshmukh and Hoskins. The contour intervals are the same as in Fig. 1.

Figure: 三角形截断谱模式（T126）对地形的表达以及滤波改进

连续傅里叶变换（实数）

Fourier Transform (real)

Analysis

$$a_k = 2 \int_{-\infty}^{\infty} f(x) \cos 2\pi kx \, dx \quad (16)$$

$$b_k = 2 \int_{-\infty}^{\infty} f(x) \sin 2\pi kx \, dx \quad (17)$$

Synthesis

$$f(x) = \int_0^{\infty} (a_k \cos 2\pi kx + b_k \sin 2\pi kx) \, dk \quad (18)$$

连续傅里叶变换（复数）

Fourier Transform (complex)

Analysis

$$c_k = \int_{-\infty}^{\infty} f(x) e^{-2\pi i k x} dx \quad (19)$$

Synthesis

$$f(x) = \int_{-\infty}^{\infty} c_k e^{2\pi i k x} dk \quad (20)$$

离散傅里叶变换（实数）

Discrete Fourier Transform (real)

Analysis

$$a_k = \frac{2}{N} \sum_{n=1}^N f_n \cdot \cos(2\pi kn/N) \quad (21)$$

$$b_k = \frac{2}{N} \sum_{n=1}^N f_n \cdot \sin(2\pi kn/N) \quad (22)$$

其中 $k \in [0 \rightarrow N/2]$, 特别地:

$$a_0 \Leftarrow a_0/2 \quad (23)$$

$$a_{N/2} \Leftarrow a_{N/2}/2 \quad (24)$$

离散傅里叶变换（实数）

Discrete Fourier Transform (real)

Analysis

```
1 subroutine dft_1d_real(x,N,a,b)
2
3     implicit none
4
5     integer, intent(in) :: N
6     real, dimension(1:N), intent(in) :: x
7     real, dimension(0:N/2), intent(out) :: a, b
8
9     integer :: i, k
10
11    do k = 0, N/2
12        a(k) = 0.0
13        b(k) = 0.0
14        do i = 1, N
15            a(k) = a(k) + x(i)*cos(2.0*Pi*k*i/N)
16            b(k) = b(k) + x(i)*sin(2.0*Pi*k*i/N)
17        end do
18        a(k) = 2.0*a(k)/N
19        b(k) = 2.0*b(k)/N
20    end do
21    a(0) = a(0)/2.0
22    a(N/2) = a(N/2)/2.0
23
24 end subroutine
```

离散傅里叶变换（实数）

Discrete Fourier Transform (real)

Synthesis

$$f_n = \sum_{k=0}^{N/2} a_k \cos(2\pi kn/N) + \sum_{k=0}^{N/2} b_k \sin(2\pi kn/N) \quad (25)$$

离散傅里叶变换（实数）

Discrete Fourier Transform (real)

Synthesis

```
1 subroutine idft_1d_real(a,b,N,x)
2
3     implicit none
4
5     integer, intent(in) :: N
6     real, dimension(0:N/2), intent(in) :: a, b
7     real, dimension(1:N), intent(out) :: x
8
9     integer i, k
10
11    do i = 1, N
12        x(i) = 0.0
13        do k = 0, N/2
14            x(i) = x(i) + a(k)*cos(2.0*Pi*k*i/N) + b(k)*sin(2.0*Pi*k*i/N)
15        end do
16    end do
17
18 end subroutine
```

离散傅里叶变换（复数）

Discrete Fourier Transform (complex)

Analysis

$$c_k = \frac{1}{N} \sum_{n=1}^N f_n \cdot e^{-i2\pi kn/N}, \quad k \in [-N/2, N/2 - 1] \quad (26)$$

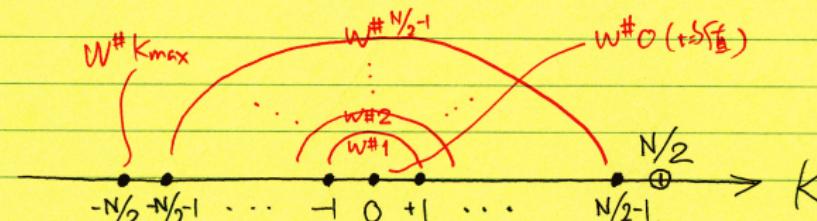
Synthesis

$$f_n = \sum_{k=-N/2}^{N/2-1} c_k \cdot e^{i2\pi kn/N}, \quad n \in [1, N] \quad (27)$$

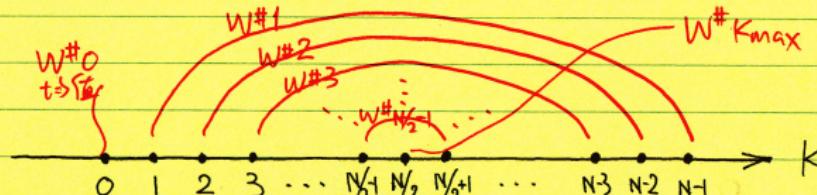
上述公式中波数范围也可以取 $k \in [-N/2 + 1, N/2]$ or $[0, N - 1]$

注意 k 的取值范围与波数的对应关系

在做 DFT 截断时， C_k 是关于 $k=0$ 留样的。



Regular DFT, $K \in [-\frac{N}{2}, \frac{N}{2}-1]$



Note: $K_{\max} = N/2$

FFT-type DFT, $k \in [0, N-1]$

离散傅里叶变换（复数）

Discrete Fourier Transform (complex)

Analysis

```
1 subroutine dft_1d_cplx(x,N,coef)
2
3 implicit none
4
5 integer, intent(in) :: N
6 complex, dimension(1:N), intent(in) :: x
7 complex, dimension(-N/2:N/2-1), intent(out) :: coef
8
9 complex, parameter :: jj=(0,1)
10 integer :: i,k
11
12 do k = -N/2, N/2-1
13   coef(k) = (0,0)
14   do i = 1, N
15     coef(k) = coef(k) + x(i)*exp(-2.0*jj*Pi*k*i/N)
16   end do
17   coef(k) = coef(k)/N
18 end do
19
20 end subroutine
```

离散傅里叶变换（复数）

Discrete Fourier Transform (complex)

Synthesis

```
1 subroutine idft_1d_cplx(coef,N,x)
2
3     implicit none
4
5     integer, intent(in) :: N
6     complex, dimension(-N/2:N/2-1), intent(in) :: coef
7     complex, dimension(1:N), intent(out) :: x
8
9     complex, parameter :: jj=(0,1)
10    integer :: i, k
11
12    do i = 1, N
13        x(i) = (0,0)
14        do k = -N/2, N/2-1
15            x(i) = x(i) + coef(k)*exp(2.0*jj*pi*k*i/N)
16        end do
17    end do
18
19 end subroutine
```

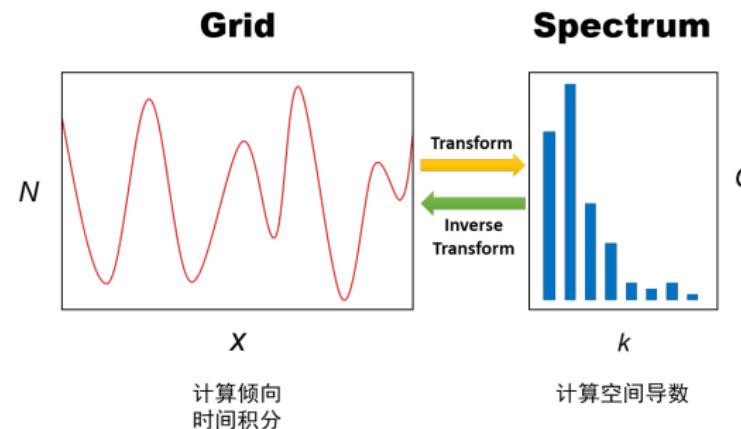
Demo

对一个纬圈的位势高度场做傅里叶变换

- 数据：NCEP/NCAR Reanalysis, 2016 年 1 月 21 日的瞬时值，500mb 高度场，55N 纬圈
- 编写子程序：复数形式的 DFT+iDFT & 实数形式的 DFT+iDFT
- 在 $0 \rightarrow k_{max}$ 波数范围内，对数据应用 DFT+iDFT，看是否与观测相当？
- 令 $k_{max} = 0, 1, 2, 3\dots$ ，尝试人为截断，体会谱变换收敛速度的快慢

准谱方法

Pesudo Spectral Method



准谱方法：在谱空间中，利用谱方法解析地估计空间导数项，再返回格点空间进行倾向项的计算，并最终在格点空间中完成积分。

准谱方法

以平流方程为例

$$\frac{\partial f}{\partial t} = -a \frac{\partial f}{\partial x}$$

- 格 – 谱变换 (Analysis):

$$c_k = \frac{1}{N} \sum_{n=1}^N f_n \cdot e^{-i2\pi kn/N}, \quad k \in [-N/2, N/2 - 1]$$

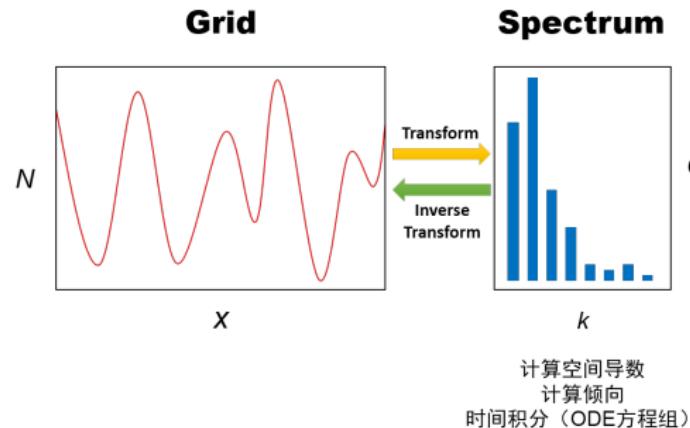
- 在谱空间估计导数项，并做谱 – 格变换 (Synthesis):

$$\frac{\partial f}{\partial x} = \sum_{k=-N/2}^{N/2-1} c_k \cdot \frac{i2\pi k}{N} \cdot e^{i2\pi kn/N}, \quad n \in [1, N]$$

- 在格点空间中得到倾向: $-a \frac{\partial f}{\partial x}$
- 时间积分

谱方法

Spectral Method



谱方法：只在谱空间中，利用谱方法解析地估计空间导数项，再计算倾向项（线性项简单；非线性项很复杂，要做矩阵相乘），时间积分退化为一组 ODE 方程组，可显式或半隐式时间积分。谱方法只在一开始需要做一次格谱变换，之后的计算都在谱空间中展开，原则上不再需要返回格点空间。

谱方法：线性平流方程

仍以定常风速 a 的一维平流方程为例

$$\frac{\partial f(x, t)}{\partial t} = -a \frac{\partial f(x, t)}{\partial x}$$

考虑谱空间中 $f(x \in [0, 2\pi], t)$ 可表述为谱系数

$$c_k(t) = \frac{1}{2\pi} \sum_x f(x, t) \cdot e^{-ikx}$$

则 $f(x, t)$ 可表述为在谱空间中时空分离的谱系数的函数

$$f(x, t) = \sum_k c_k(t) \cdot e^{ikx}$$

将此谱表达代入平流方程

$$\sum_k \frac{\partial}{\partial t} c_k(t) \cdot e^{ikx} = -a \sum_k c_k(t) \cdot ik \cdot e^{ikx}$$

在谱空间中，对于某波数 k 而言

$$\frac{\partial}{\partial t} c_k(t) \cdot e^{ikx} = -a \cdot c_k(t) \cdot ik \cdot e^{ikx}$$

化简可得

$$\frac{dc_k(t)}{dt} = -iak \cdot c_k(t)$$

我们再仔细观察这两种形式

格点空间的平流方程

$$\frac{\partial f(x, t)}{\partial t} = -a \frac{\partial f(x, t)}{\partial x}$$

谱空间的平流方程

$$\frac{dc_k(t)}{dt} = -iak \cdot c_k(t)$$

在波数 k 空间中，此方程实际是一套ODE 方程组，可直接积分

谱方法：非线性平流方程

以一维动量 $u(x, t)$ 平流方程为例

$$\frac{\partial u(x, t)}{\partial t} = -u(x, t) \frac{\partial u(x, t)}{\partial x}$$

与线性平流方程类似， $u(x, t)$ 可表述为在谱空间中时空分离的谱系数的函数

$$u(x, t) = \sum_k c_k(t) e^{ikx}$$

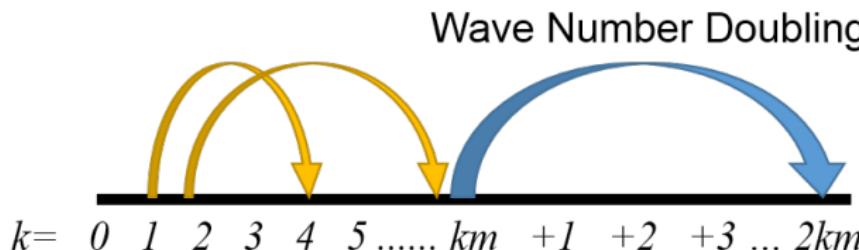
将此谱表达代入平流方程

$$\sum_k \frac{\partial}{\partial t} c_k(t) e^{ikx} = - \sum_k c_k(t) e^{ikx} \cdot \sum_k ik \cdot c_k(t) e^{ikx}$$

$$\sum_k \frac{\partial}{\partial t} c_k(t) e^{ikx} = -ik \cdot \left(\sum_k c_k(t) e^{ikx} \times \sum_k c_k(t) e^{ikx} \right)$$

右边构成 $k_{max} \times k_{max}$ 的矩阵，计算量瞬间平方，其物理意义是波 - 波相互作用

非线性不稳定：波波相互作用



- 波波相互作用导致能量向短波（长波）串级，谱空间中最大波数加倍 (Doubling)
- 截断：此时发挥了决定性作用，把一定 k 以上的高波数波全部滤掉，抑制短波增长
- 由于波波相互作用的计算是 $k_{max} \times k_{max}$ 的矩阵，使得谱方法在处理非线性项时代价极其昂贵，与差分方法相比在计算效率上相距甚远
- FFT(Cooley&Tukey, 1965) 的出现使得谱方法的计算效率显著提高

对非线性不稳定的理解

举例 Burger's 方程（动量平流方程）

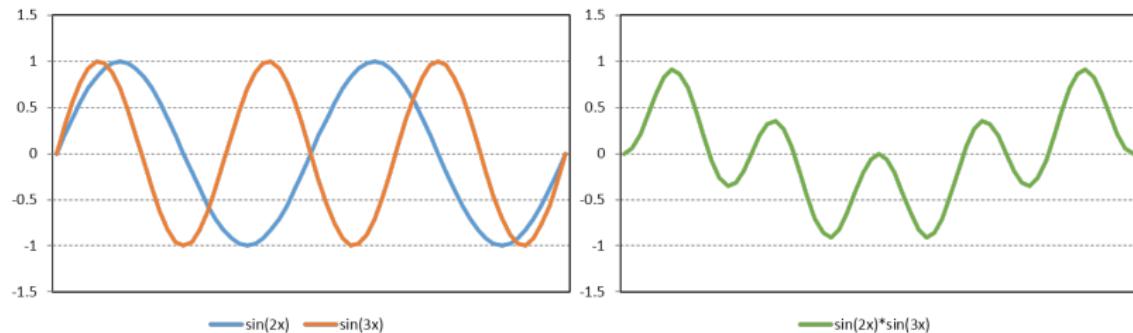
$$\frac{\partial u}{\partial t} = -u \frac{\partial u}{\partial x}$$

在 $x \in [0, 2\pi]$ 的空间上考虑谐波 $u = \sin kx$, k 为波数, 则非线性项的解析解为

$$u \frac{\partial u}{\partial x} = k \cdot \sin kx \cdot \cos kx = \frac{k}{2} \sin 2kx$$

可见, 波波相互作用产生的非线性加倍 (Doubling) 是连续函数本身的性质, 而并非离散误差。事实上, 正是由于离散系统不能很好地再现连续系统中的 Doubling 过程, 才导致的非线性不稳定

对波波相互作用的理解



$$A = \sin(2x) \quad k_a = 2$$

$$B = \sin(3x) \quad k_b = 3$$

$$A \times B = \sin(2x)\sin(3x)$$

$$|k_a - k_b| = 1$$

$$k_a + k_b = 5$$

准谱方法 vs. 谱方法

Table: 准谱方法与谱方法的对比

	准谱方法	谱方法
主要立足点	格点空间	谱空间
谱空间里计算什么	偏导数	全部倾向
线性项计算量	$O(k_{max})$	$O(k_{max})$
非线性项计算量	$O(k_{max})$	$O(k_{max} \times k_{max})$
时间积分	标准 PDE 积分	退化为对谱系数的 ODE 积分

Strengths and Weaknesses

Strengths

- 相速度与解析解一致，不会变慢（差分方法相速度会变慢）
- 在波数空间中收敛快，只用很少的几个波数即可以很高的精度表达大尺度特征
- 对于导数的估计精度远高于差分方法

Weaknesses

- 没有任何的守恒性（谱方法本身是没有任何守恒考量的）
- Gibbs 现象广泛存在，对不连续点表达很差，比如地形
- 为了考虑物理过程对变量的修正，要频繁做格谱变换、谱格变换
- 水汽平流不能用谱方法（出现负水汽问题）
- 若提高分辨率，则计算代价呈指数增长，远超差分方法

弦振动

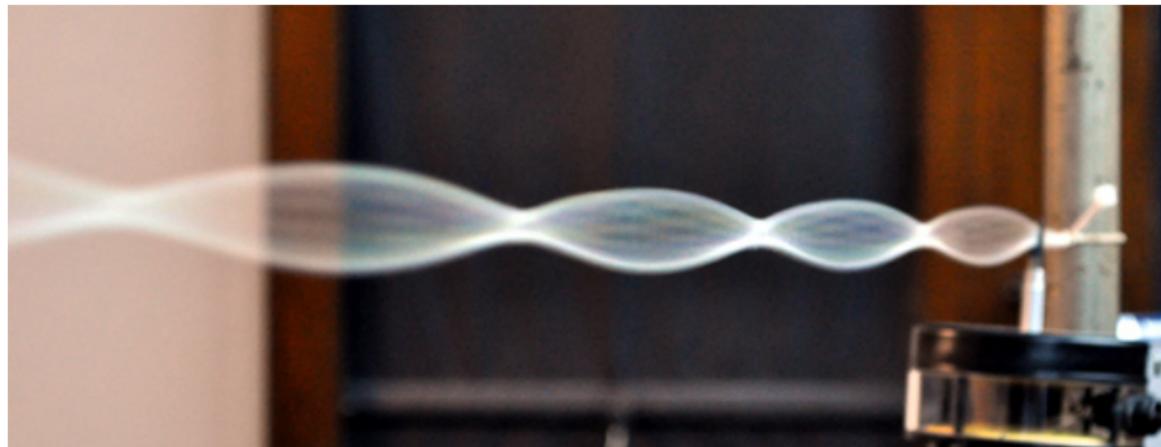


Figure: 弦振动的延长曝光时间摄影照片

弦振动方程

Vibrating String Equation or Classic Wave Equation

数学形式

$$\frac{\partial^2 \Phi}{\partial t^2} = c^2 \frac{\partial^2 \Phi}{\partial x^2}$$

$$c = \sqrt{\frac{T}{\rho}}$$

- Φ 是弦上每个点距平衡态的相对位移
- c 是波沿弦传播的速度
- T 是作用在弦上的张力
- ρ 是单位长度弦的质量，即弦密度

拉普拉斯方程

Laplace's Equation

- 考虑驻波形式，忽略时间项，弦振动方程退化为 Laplace 方程

$$\nabla^2 \Phi = 0$$

- 在笛卡尔 - 球坐标下 ($x = r \cos \lambda \sin \theta$, $y = r \sin \lambda \sin \theta$) 展开

$$\frac{1}{\cos \theta} \frac{\partial}{\partial \theta} \left(\cos \theta \frac{\partial \Phi}{\partial \theta} \right) + \frac{1}{\cos^2 \theta} \frac{\partial^2 \Phi}{\partial \lambda^2} = 0$$

- 对经度 λ 和余纬 θ 实行分离变量，令 $\Phi = L(\lambda)P(\theta)$ ，代入上式得
- 东西方向 (Fourier)

$$\frac{d^2 L}{d \lambda^2} + m^2 L = 0$$

- 南北方向 (Associated Legendre Eq)

$$\frac{1}{\cos \theta} \frac{d}{d \theta} \left(\cos \theta \frac{d P}{d \theta} \right) + \left(n(n+1) - \frac{m^2}{\cos^2 \theta} \right) P = 0 \quad (28)$$

勒让德多项式

Legendre Polynomials

- 令 $\mu = \cos \theta \in [-1, 1]$, 其中 θ 是 colatitude, 且令 $m = 0$
- Legendre 方程

$$\frac{d}{d\mu} \left((1 - \mu^2) \frac{dP}{d\mu} \right) + n(n+1)P = 0$$

- Legendre 多项式

$$P_n(\mu) = \sum_{r=0}^M (-1)^r \frac{(2n-2r)!}{2^n r! (n-r)! (n-2r)!} \mu^{n-2r}, M = \frac{n}{2}$$

- L 多项式的另一种形式: 罗德里格斯公式 (Rodrigues' formula)

$$P_n(\mu) = \frac{1}{2^n n!} \frac{d^n}{d\mu^n} (\mu^2 - 1)^n$$

Legendre 多项式的基本性质

- 在 $\mu = 1$ 处为固定点: $P_n(\mu = 1) \equiv 1$
- n 是偶数次时, $P_n(\mu)$ 为偶对称
- n 是奇数次时, $P_n(\mu)$ 为奇对称
- 和 x 轴有 n 个交点, 称为根 (root)

Legendre 多项式的几何图景

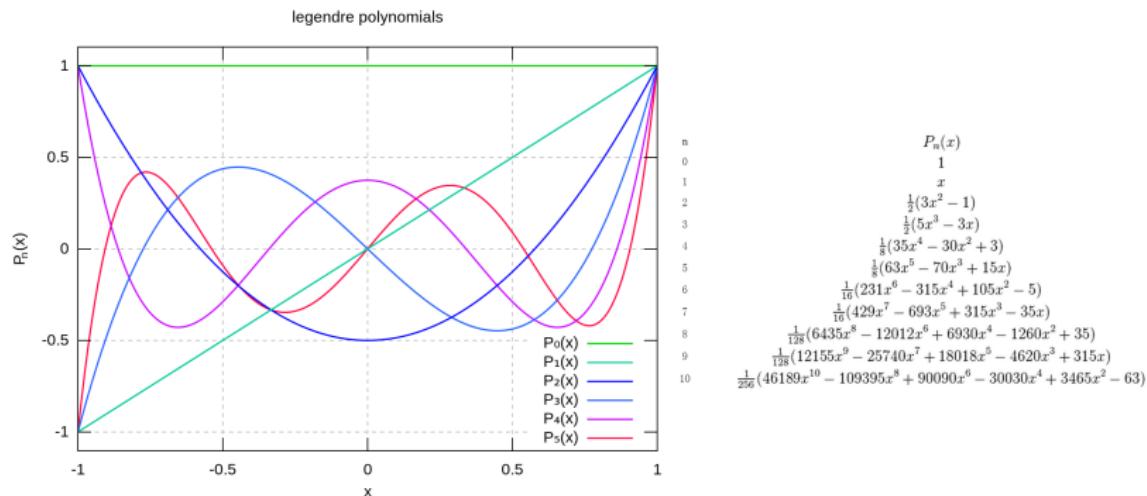


Figure: Degree $0 \rightarrow 5$ 的 Legendre 多项式

Legendre 多项式的递归建立

已知前两个

$$P_0(\mu) = 1$$

$$P_1(\mu) = \mu$$

利用递归关系

$$(n+1)P_{n+1}(\mu) = (2n+1)\mu P_n(\mu) - nP_{n-1}(\mu)$$

Fortran Subroutine

```
1  function lp_fast(n,x) result(lpreturn)
2
3      implicit none
4      integer, intent(in) :: n
5      real, intent(in) :: x
6      real :: lpreturn
7
8      !—— local
9      real :: lp0, lp1
10     integer :: i
11
12    lp0 = 1.000000
13    if (n==0) then
14        lpreturn = lp0;  return
15    end if
16
17    lp1 = x
18    if (n==1) then
19        lpreturn = lp1;  return
20    end if
21
22    !——  $P_n(x)$ , for  $n > 1$ 
23    do i = 2, n
24        lpreturn = ((2.0*(i-1)+1.0)*x*lp1-(i-1)*lp0)/i
25        lp0 = lp1
26        lp1 = lpreturn
27    end do
28
29    end function
```

连带勒让德多项式

Associated Legendre Polynomials

- 令 $\mu = \cos \theta \in [-1, 1]$, 其中 θ 是 colatitude, 则方程28可写为
- Associated Legendre 方程

$$\frac{d}{d\mu} \left((1 - \mu^2) \frac{dP}{d\mu} \right) + \left(n(n+1) - \frac{m^2}{1 - \mu^2} \right) P = 0$$

- Associated Legendre 多项式 (Rodrigues' formula)

$$P_n^m(\mu) = \frac{(1 - \mu^2)^{m/2}}{2^n n!} \frac{d^{n+m}}{d\mu^{n+m}} (\mu^2 - 1)^n$$

- 其中 (order) $m \in [-M, M]$, (degree) $n \geq |m|$

Associated Legendre 多项式的基本性质

- 1 当 $n < \|m\|$ 时, $P_n^m(\mu) \equiv 0$
- 2 在两端 $\mu = \pm 1$ 处为固定点: $P_n^m(\mu = \pm 1) \equiv 0$
- 3 $P_n^{-m}(\mu)$ 与 $P_n^m(\mu)$ 的对称关系:

$$P_n^{-m}(\mu) = (-1)^m \frac{(n-m)!}{(n+m)!} P_n^m(\mu)$$

- 4 $P_n^m(-\mu)$ 与 $P_n^m(\mu)$ 的对称关系:

$$P_n^m(-\mu) = (-1)^{n-m} P_n^m(\mu)$$

Associated Legendre 多项式的参数空间

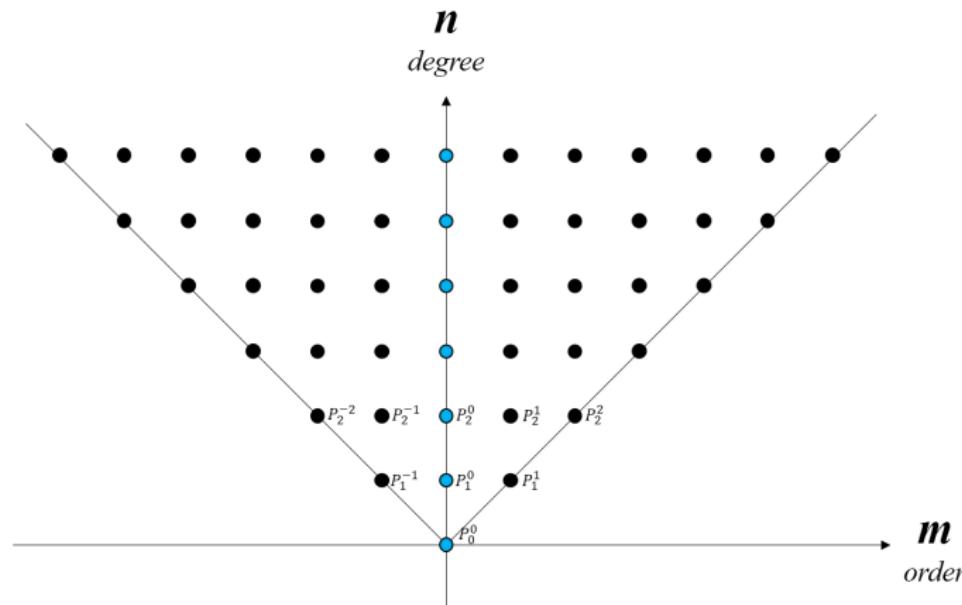


Figure: 只有 $n \geq \|m\|$ 的整数点，是 P_n^m 的参数空间

Associated Legendre 多项式的几何图景

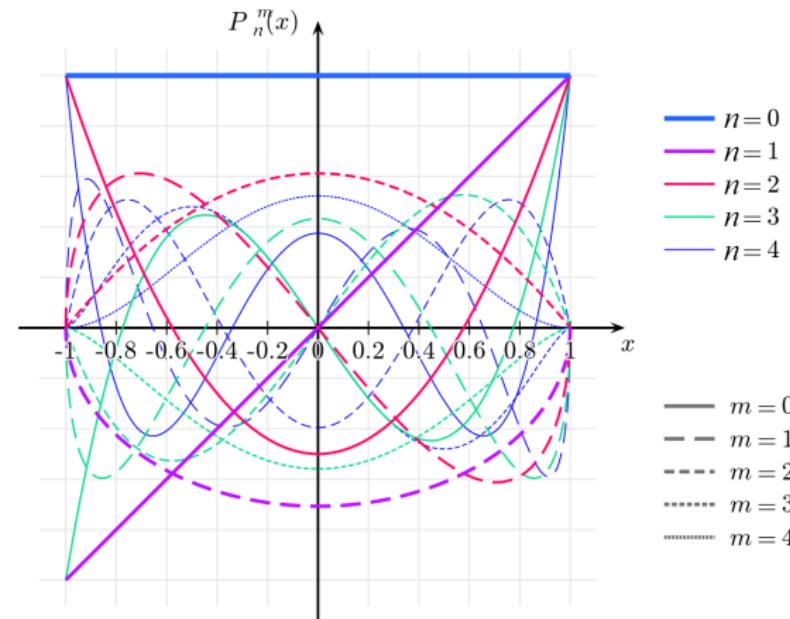


Figure: $n \in [0, 4]$, $m \in [0, 4]$ 的 Associated Legendre 多项式

Associated Legendre 多项式的递归建立

1 上下递归关系

$$\mu P_n^m(\mu) = e_{n+1}^m P_{n+1}^m(\mu) + e_n^m P_{n-1}^m(\mu), e_n^m = \left(\frac{n^2 - m^2}{4n^2 - 1} \right)^{1/2}$$

2 斜角递归关系

$$(1 - \mu^2)^{1/2} P_n^m(\mu) = g_n^m P_{n+1}^{m+1}(\mu) - h_n^m P_{n-1}^{m+1}(\mu)$$

$$g_n^m = \left(\frac{(n+m+1)(n+m+2)}{(2n+1)(2n+3)} \right)^{1/2}, h_n^m = \left(\frac{(n-m-1)(n-m)}{(2n+1)(2n-1)} \right)^{1/2}$$

3 导数双侧递归关系

$$(1 - \mu^2) \frac{dP_n^m(\mu)}{d\mu} = -ne_{n+1}^m P_{n+1}^m(\mu) + (n+1)e_n^m P_{n-1}^m(\mu)$$

4 导数单侧递归关系

$$(1 - \mu^2) \frac{dP_n^m(\mu)}{d\mu} = (2n+1)e_n^m P_{n-1}^m(\mu) - n\mu P_n^m(\mu)$$

Associated Legendre 多项式的递归建立

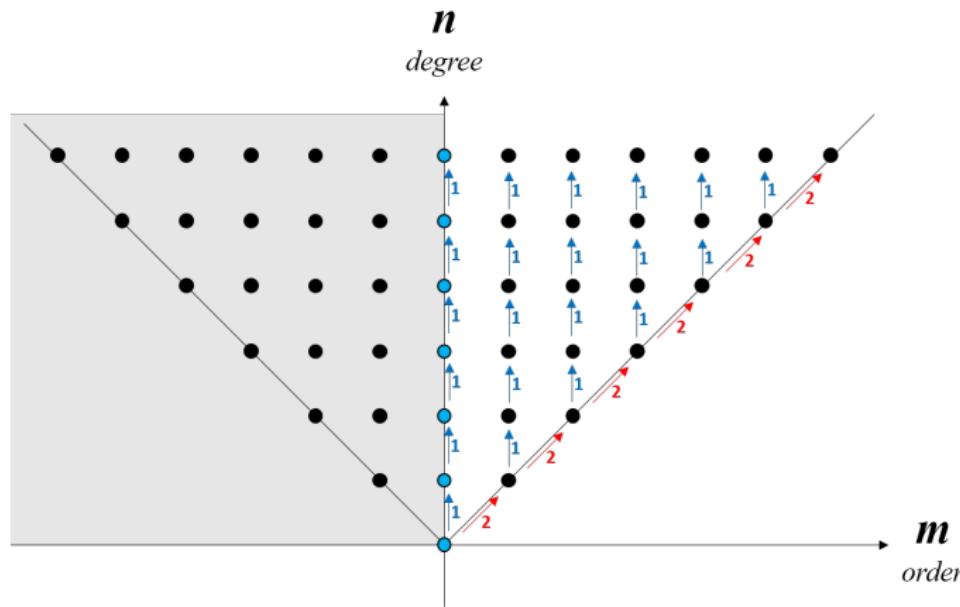


Figure: 反复利用递归关系 1 和 2，即可建立全部 $P_n^m(\mu)$ 多项式

Fortran Subroutine 1/2

```
1      function alf(l,m,x)
2
3          implicit none
4          integer, intent(in) :: l, m
5          real, intent(in) :: x
6          real :: alf
7
8          !— local variables
9          integer :: i, ll
10         real :: fact, pl1, pmm, pmmp1, somx2
11
12         !— start here
13         if (m.lt.0 .or. m.gt.l .or. abs(x).gt.1) then
14             stop "Wrong arguments in [alf] function !!!"
15         end if
16
17         !— compute P_m^m
18         pmm = 1.0
19         if (m.gt.0) then
20             somx2 = sqrt((1.0-x)*(1.0+x))
21             fact = 1.0
22             do i = 1, m
23                 pmm = -pmm*fact*somx2
24                 fact = fact+2.0
25             end do
26         end if
27         if (l.eq.m) then
28             alf = pmm;  return
29         end if
```

Fortran Subroutine 2/2

```
31          !—— compute  $P_{-}(m+1)^{\wedge}m$ 
32      pmmp1 = x*(2*m+1)*pmm
33      if (l.eq.m+1) then
34          alf = pmmp1
35          return
36      end if
37
38      !—— integrate until  $P_l^{\wedge}m$ 
39      do ll = m+2, l
40          pll = (x*(2*ll-1)*pmmp1-(ll+m-1)*pmm)/(ll-m)
41          pmm = pmmp1
42          pmmp1 = pll
43      end do
44      alf = pll
45
46      !—— normalize
47      alf = alf * sqrt(factorial(l-m)/factorial(l+m))
48  end function
```

其中 factorial 是一个求阶乘的函数，由于代码很简单这里从简

球谐函数变换

- 球坐标系下 Laplace 方程

$$\nabla^2 \Phi = 0$$

的解，称为球谐函数 (spherical harmonics) $Y_n^m(\lambda, \mu)$

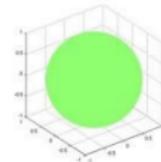
$$Y_n^m(\lambda, \mu) = P_n^m(\mu) e^{-im\lambda}$$

- 球谐函数的东西方向（经向）是 Fourier polynomials, m 是波数参数
- 球谐函数的南北方向（纬向）是 Associated Legendre polynomials, m 和 n 都是波数参数
- m 参数是 Fourier 和 Associated Legendre 共有的平衡参数，使得纬向波数和经向波数互相制衡：经向波数 $\rightarrow m$, 纬向波数 $\rightarrow (n - m)$
- 在球面上，用球谐函数作为基函数，对任意气象变量进行谱展开，称为球谐变换 (Spherical harmonic transform)

球谐函数的图景

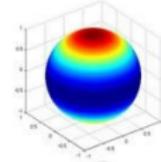
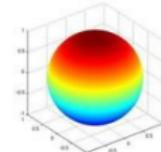
Single Harmonics

$n = 0$

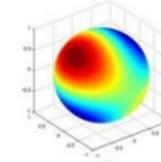
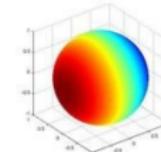


$$\cos(m\phi) P_n^m(\cos \theta)$$

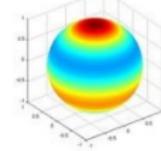
$n = 1$



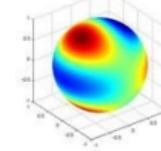
$n = 2$



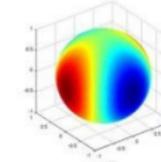
$n = 3$



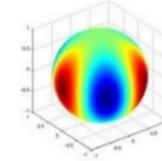
$m = 0$



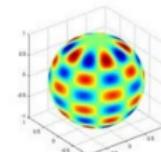
$m = 1$



$m = 2$



$m = 3$



$n = 10$
 $m = 5$

球谐函数的图景

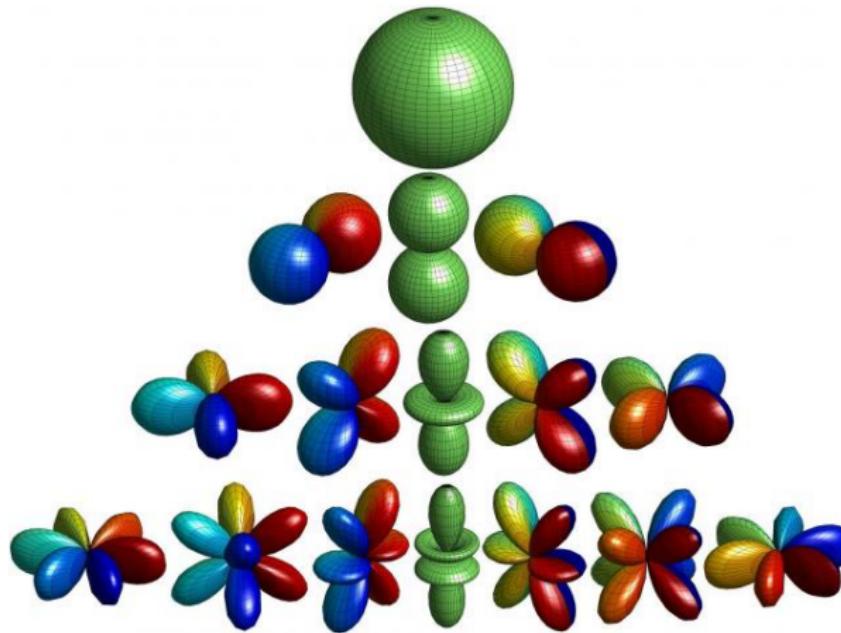


Figure: 非标准化的球谐函数

球谐函数变换的一般流程

两步变换法：以格 - 谱变换 $A(\lambda, \mu) \rightarrow A_n^m$ 为例

- 第一步：Fourier transform

对纬度 μ 循环，在每条纬线上执行 DFT 变换，得到每个纬度的 A^m

$$A^m(\mu) = \frac{1}{2\pi} \int_0^{2\pi} A(\lambda, \mu) e^{-im\lambda} d\lambda$$

- 第二步：Associated Legendre transform

沿经向做一次连带勒让德变换，得到 A 在球谐函数谱空间的系数 A_n^m

$$A_n^m = \int_{-1}^1 A^m(\mu) P_n^m(\mu) d\mu$$

反变换（谱 - 格变换）就是把两步反过来做即可：先做 reverse Associated Legendre transform，再做 reverse Fourier transform

球谐变换的截断方法

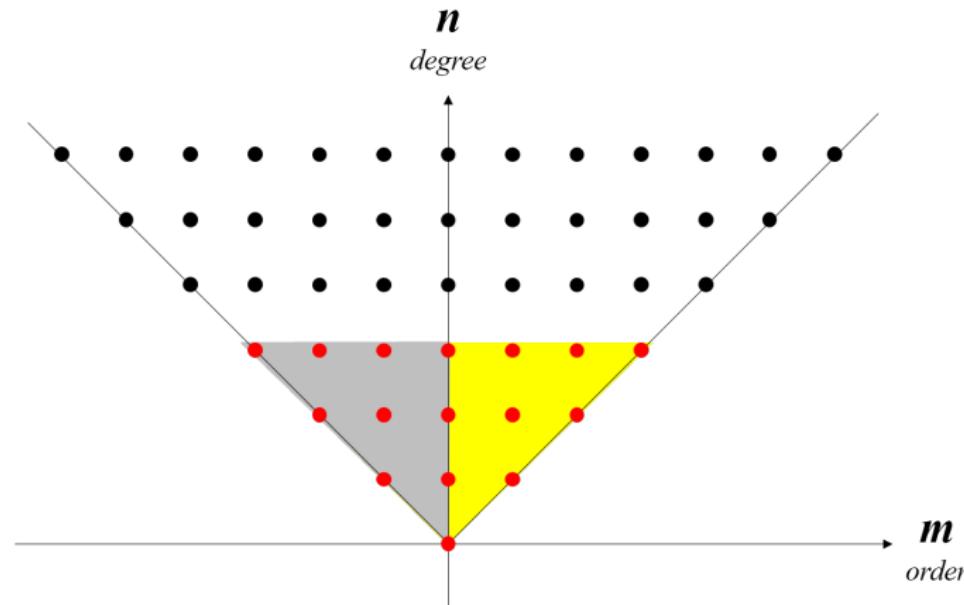


Figure: 三角形 (Triangular) 截断 T4

球谐变换的截断方法

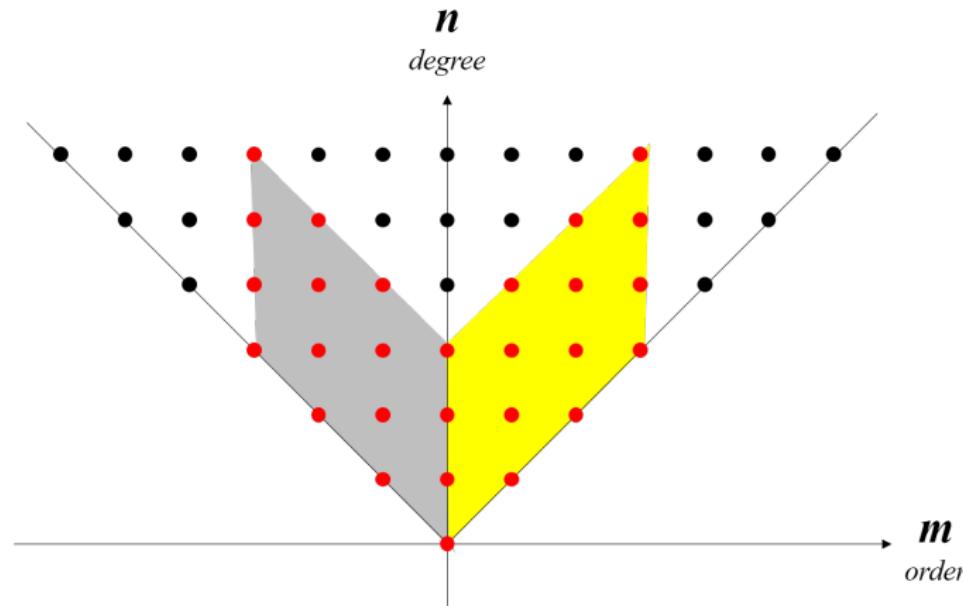
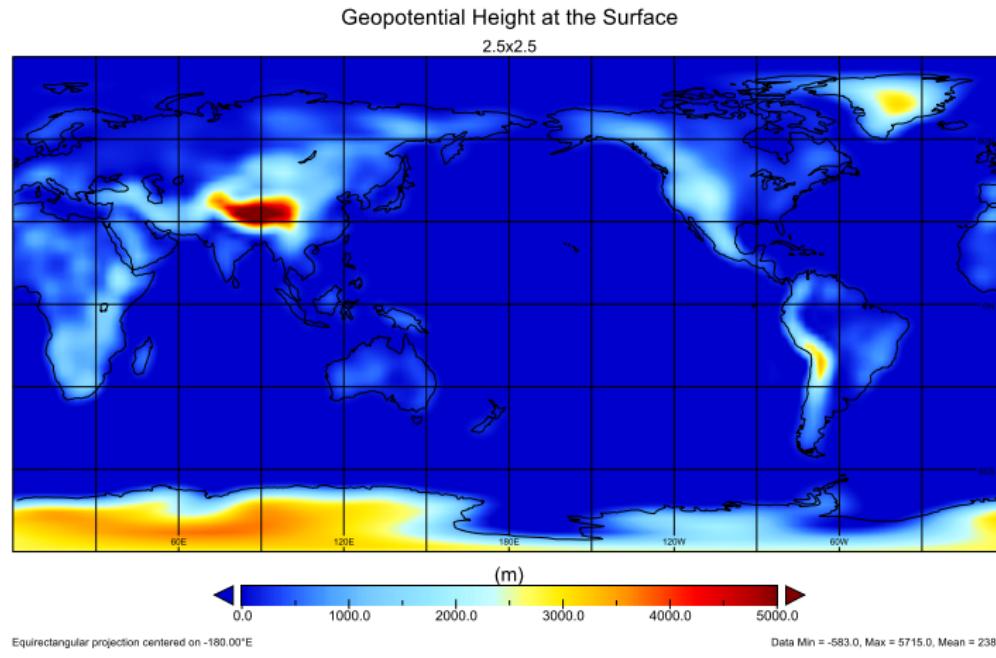


Figure: 菱形 (Rhomboidal) 截断 R4

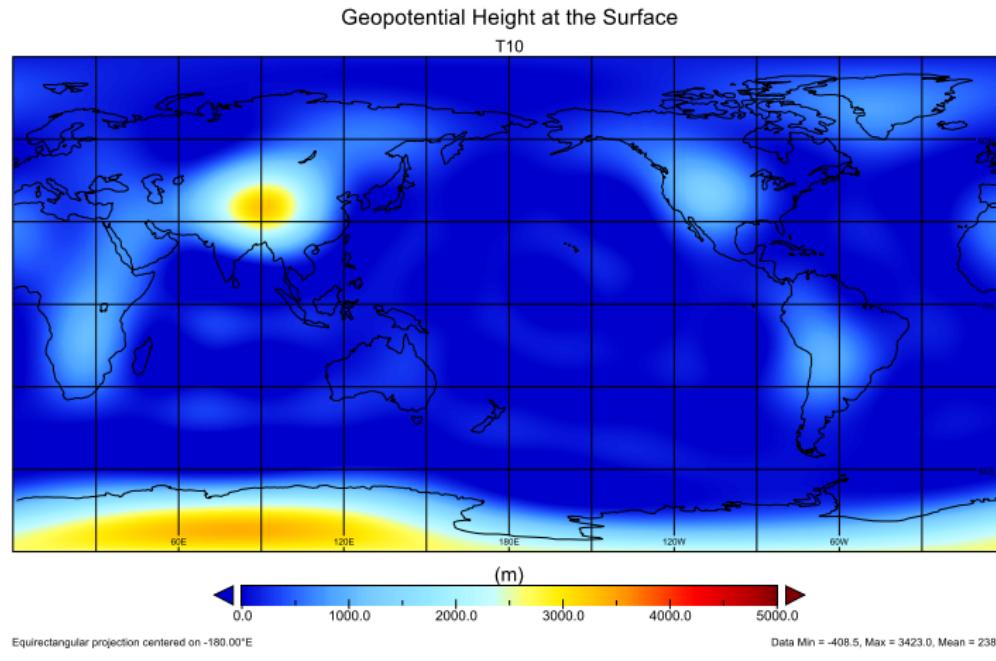
Demo

原始的格点地形分布



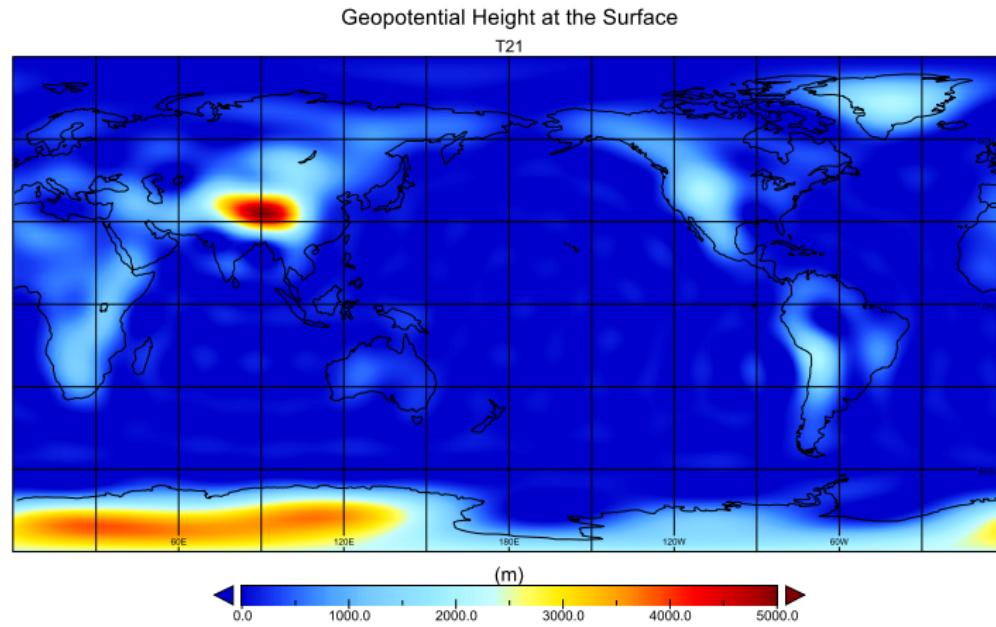
Demo

三角截断 T10



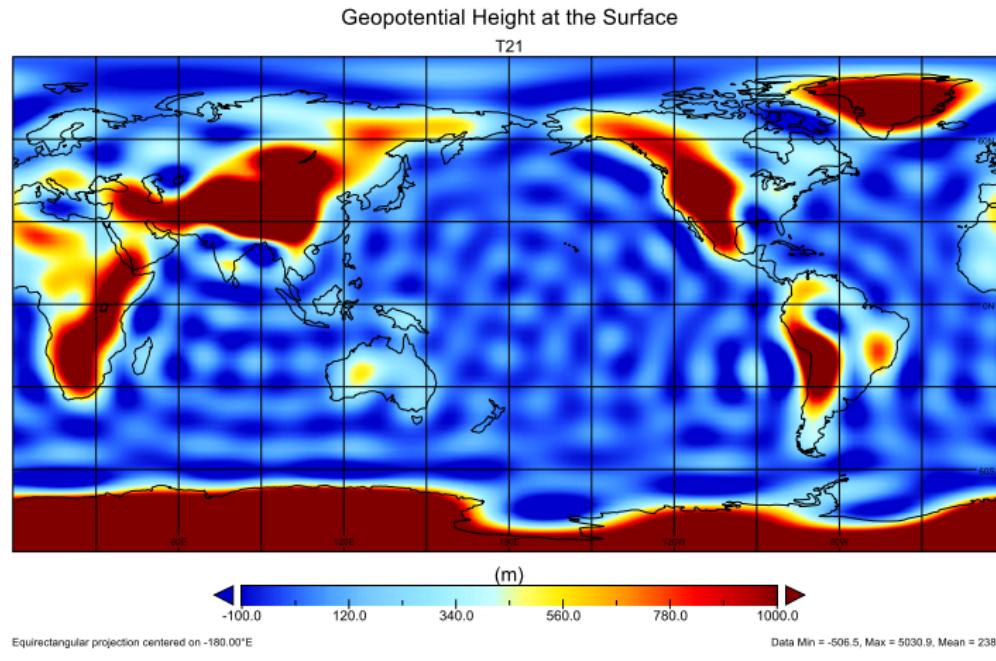
Demo

三角截断 T21



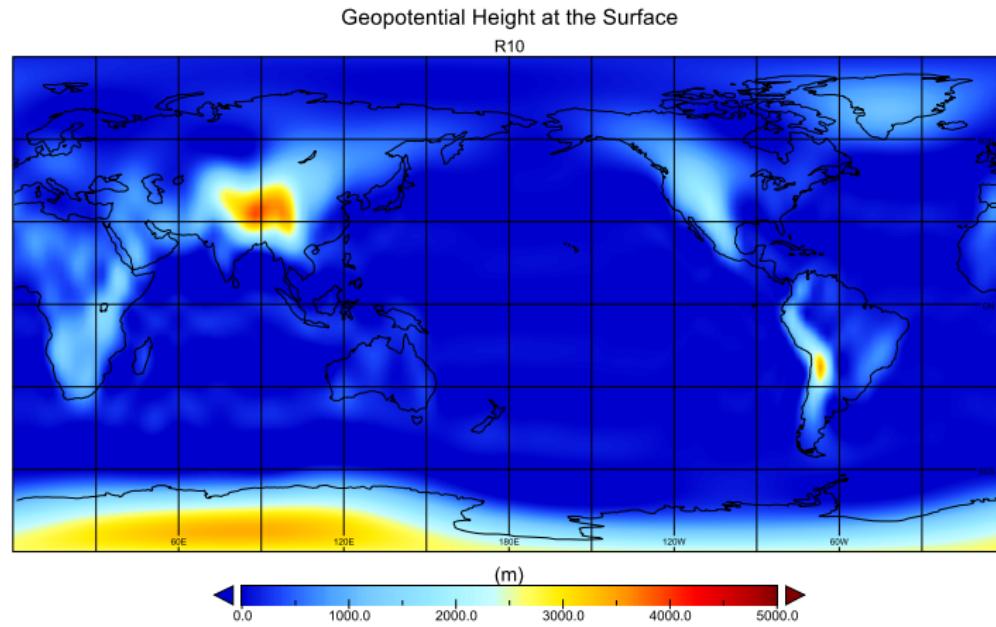
Demo

改变色阶即可看到 T21 中海上清晰的 Gibbs 现象



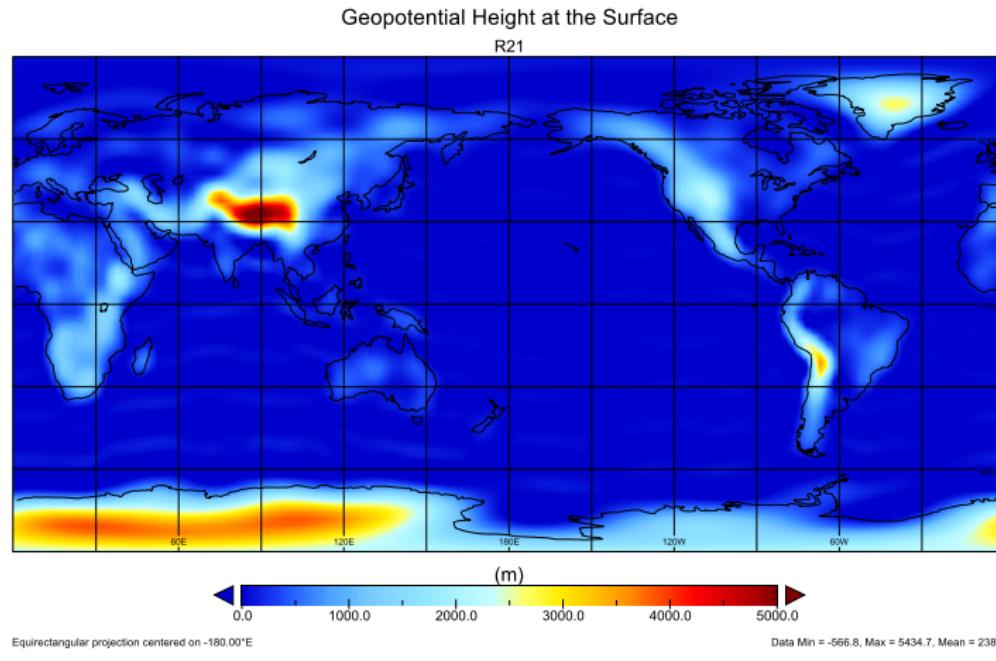
Demo

菱形截断 R10



Demo

菱形截断 R21



基于球谐变换的谱模式

最后值得一提的几句话

- 球谐变换将两极作为“固定点”，客观上避免了极点问题
- 球谐变换计算量很大，因此在计算非线性项的时候，通常采用返回格点空间计算的方法，称为变换法 (Transform Method)；只有在理论研究中，只考虑少数几个波数时，才直接计算波波相互作用矩阵
- 快速傅里叶变换 FFT 加速了经向 Fourier transform
- 高斯积分法 (Gaussian Quadrature) 加速了纬向 Associated Legendre transform
- 谱方法对应的格点空间，通常经向上需要 $2n \rightarrow 3n$ 个格点，还要考虑尽量是 2 的指数幂，以满足 FFT 的要求

1 气候模式与平流方程

2 时间积分

3 有限差分方法

4 谱方法

5 有限体积法

- The Riemann Problem
- Finite Volume Framework
- Simple FV Schemes
- Advanced FV Schemes

6 谱元法

7 半拉格朗日法

8 数值计算方法之比较

问题的提出

The Riemann Problem / Discontinuous Problem

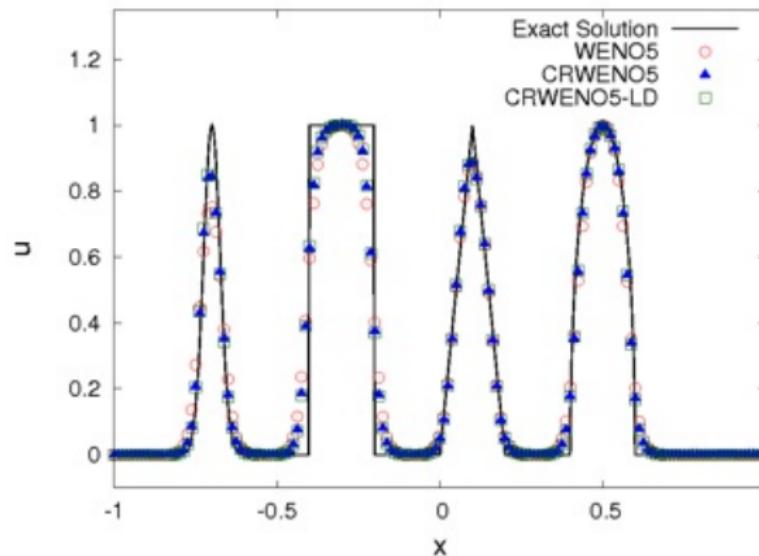


Figure: 平流方程 $\frac{\partial N}{\partial t} = -u \frac{\partial N}{\partial x}$ 对于不连续问题本质上是无效的

FV 的基本思想

2D mesh

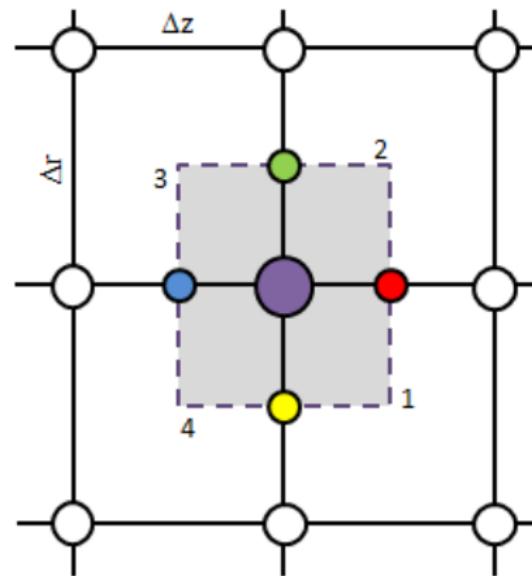


Figure: 以整个格点的积分为立脚点，考虑 4 个边界进出通量

FV 的基本思想

3D mesh

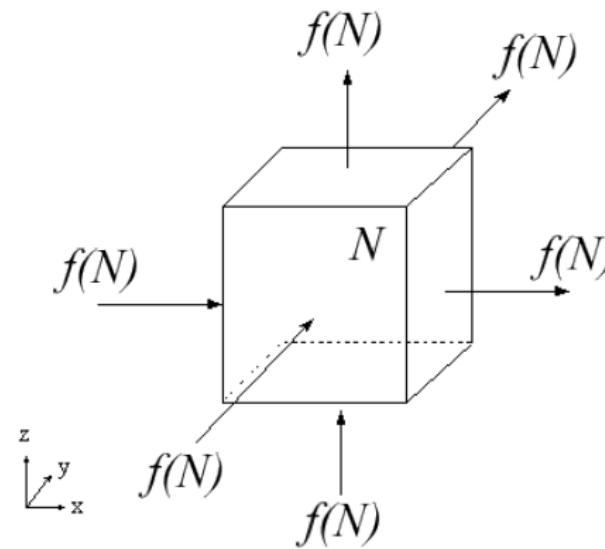


Figure: 这就是 Volume 概念的由来

平流方程：FV 等价性

一维 Advection 方程

$$\frac{\partial N_i}{\partial t} = -c \frac{\partial N}{\partial x} = -\frac{1}{\Delta x} (f_{i+1/2} - f_{i-1/2}), \quad f = f(N) = cN$$

三维 Advection 方程

$$\begin{aligned}\frac{\partial N_i}{\partial t} &= -c \left(\frac{\partial N}{\partial x} + \frac{\partial N}{\partial y} + \frac{\partial N}{\partial z} \right) \\ &= -\frac{1}{\Delta x} (f_{i+1/2,j,k} - f_{i-1/2,j,k}) \\ &\quad - \frac{1}{\Delta y} (f_{i,j+1/2,k} - f_{i,j-1/2,k}) \\ &\quad - \frac{1}{\Delta z} (f_{i,j,k+1/2} - f_{i,j,k-1/2})\end{aligned}$$

Finite Volume Framework

更一般的 1D 平流方程

$$\frac{\partial N}{\partial t} = - \frac{\partial f(N)}{\partial x}$$

有限体积框架

$$\frac{\partial N_i}{\partial t} = - \frac{1}{\Delta x} (f_{i+1/2} - f_{i-1/2})$$

Finite Volume Framework

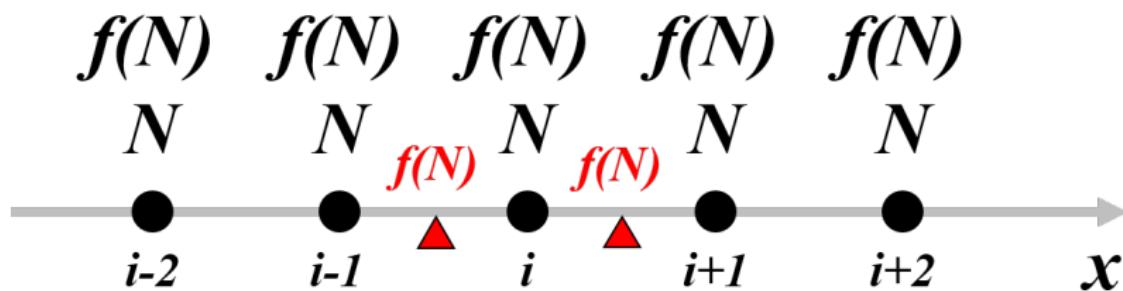


Figure: 问题转化为如何插值得到半格点的红色 $f(N)$?

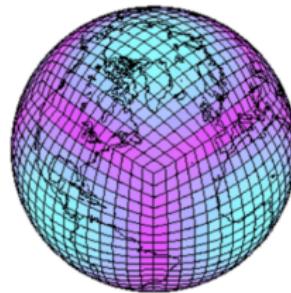
Advantages of FV Framework

一般意义上的优势

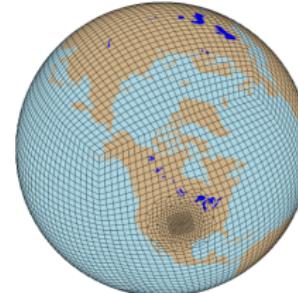
- 1 FV 思想是先进的哲学思想，是 Riemann 问题的弱解
- 2 解放了对格点形状的要求：任意形状网格都可以使用
- 3 守恒性能优异：只取决于半格点通量的插值的好坏，问题得到了有效转化和简化

八仙过海的格点设计

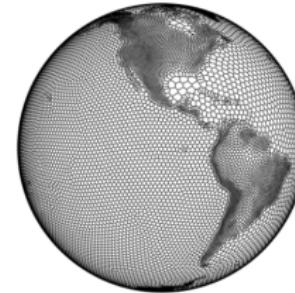
立方体投影



立方体投影自适应



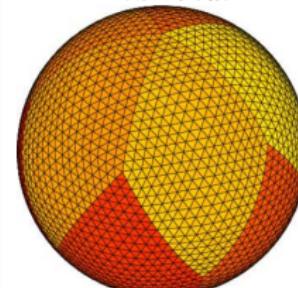
自适应



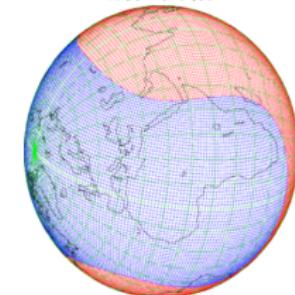
六边形网格



三角形网格

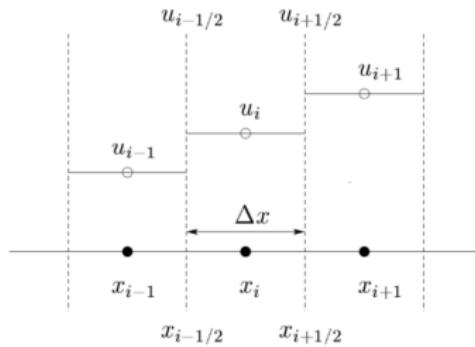


阴阳网格



分段常数插值

Piecewise-constant solution



Upwind-biased interface values

$$v > 0 \quad u_{i-1/2} \approx u_{i-1}, \quad u_{i+1/2} \approx u_i$$

$$I_c \approx v \frac{u_i - u_{i-1}}{\Delta x} \quad \text{backward difference}$$

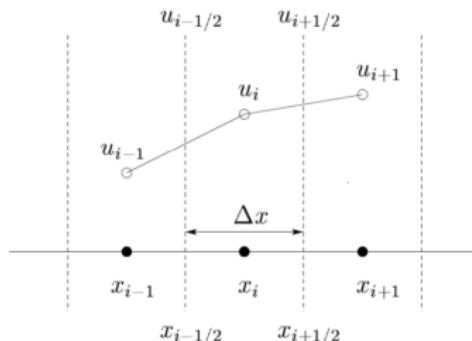
$$v < 0 \quad u_{i-1/2} \approx u_i, \quad u_{i+1/2} \approx u_{i+1}$$

$$I_c \approx v \frac{u_{i+1} - u_i}{\Delta x} \quad \text{forward difference}$$

Figure: 相当于迎风格式

分段线性插值

Piecewise-linear solution



Interpolation polynomial

$$p_1(x) = u_L \frac{x_R - x}{x_R - x_L} + u_R \frac{x - x_L}{x_R - x_L}$$

Averaged interface values

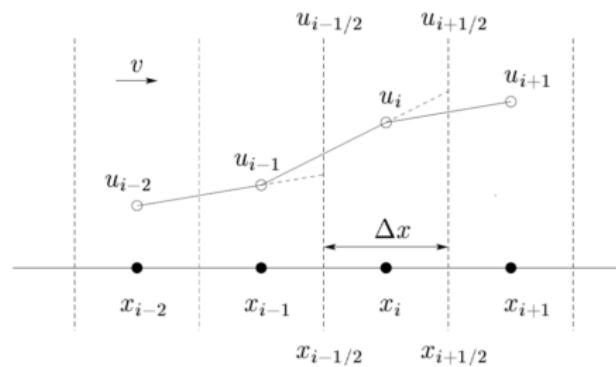
$$u_{i-1/2} \approx \frac{u_{i-1} + u_i}{2}, \quad u_{i+1/2} \approx \frac{u_i + u_{i+1}}{2}$$

$$I_c \approx v \frac{u_{i+1} - u_{i-1}}{2\Delta x} \quad \text{central difference}$$

Figure: 相当于中央差

线性外插

Piecewise-linear solution



Upwind-biased extrapolation

$$\boxed{v > 0}$$

$$u_{i-1/2} \approx \frac{3u_{i-1} - u_{i-2}}{2}$$

$$u_{i+1/2} \approx \frac{3u_i - u_{i-1}}{2}$$

$$I_c \approx v \frac{3u_i - 4u_{i-1} + u_{i-2}}{2\Delta x}$$

$$\boxed{v < 0}$$

$$u_{i-1/2} \approx \frac{3u_i - u_{i+1}}{2}$$

$$u_{i+1/2} \approx \frac{3u_{i+1} - u_{i+2}}{2}$$

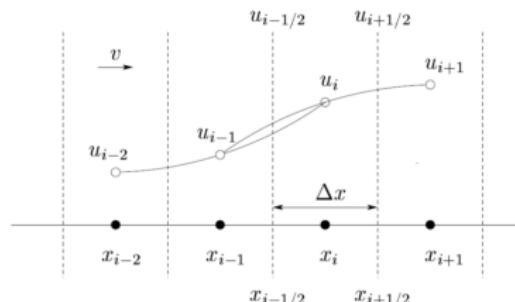
$$I_c \approx -v \frac{3u_i - 4u_{i+1} + u_{i+2}}{2\Delta x}$$

Figure: 相当于单侧差分

二次多项式插值

Quadratic Upwind Interpolation for Convective Kinematics

$$p_2(x) = u_L \frac{x-x_M}{x_L-x_M} \frac{x_R-x}{x_R-x_L} + u_M \frac{x-x_L}{x_M-x_L} \frac{x_R-x}{x_R-x_M} + u_R \frac{x-x_L}{x_R-x_L} \frac{x-x_M}{x_R-x_M}$$



Upwind-biased interface values

$$\boxed{v > 0} \quad u_{i-1/2} \approx \frac{3u_i + 6u_{i-1} - u_{i-2}}{8}$$

$$u_{i+1/2} \approx \frac{3u_{i+1} + 6u_i - u_{i-1}}{8}$$

$$I_c \approx v \frac{3u_{i+1} + 3u_i - 7u_{i-1} + u_{i-2}}{8\Delta x}$$

$$\boxed{v < 0} \quad u_{i-1/2} \approx \frac{3u_{i-1} + 6u_i - u_{i+1}}{8}$$

$$u_{i+1/2} \approx \frac{3u_i + 6u_{i+1} - u_{i+2}}{8}$$

$$I_c \approx -v \frac{3u_{i-1} + 3u_i - 7u_{i+1} + u_{i+2}}{8\Delta x}$$

Figure: 相当于 3 阶差分

Advanced Finite Volume Schemes

更复杂精确的插值方法带来了更先进的 FV 方案

- 1 Lax-Wendroff
- 2 MUSCL (Monotonic Upstream-centered Schemes for Conservation Laws)
- 3 FCT (Flux-Corrected Transport)
- 4 TVD (Total Variation Diminished)
- 5 ENO (Essential Non-Oscillation)
- 6 WENO (Weighted Essential Non-Oscillation)

其中后 4 种是带 Limiter 设计的，对平抑震荡，维持熵不减少起到重要作用

1 气候模式与平流方程

2 时间积分

3 有限差分方法

4 谱方法

5 有限体积法

6 谱元法

- Spectral Element Framework
- Orthogonal Basis Functions

7 半拉格朗日法

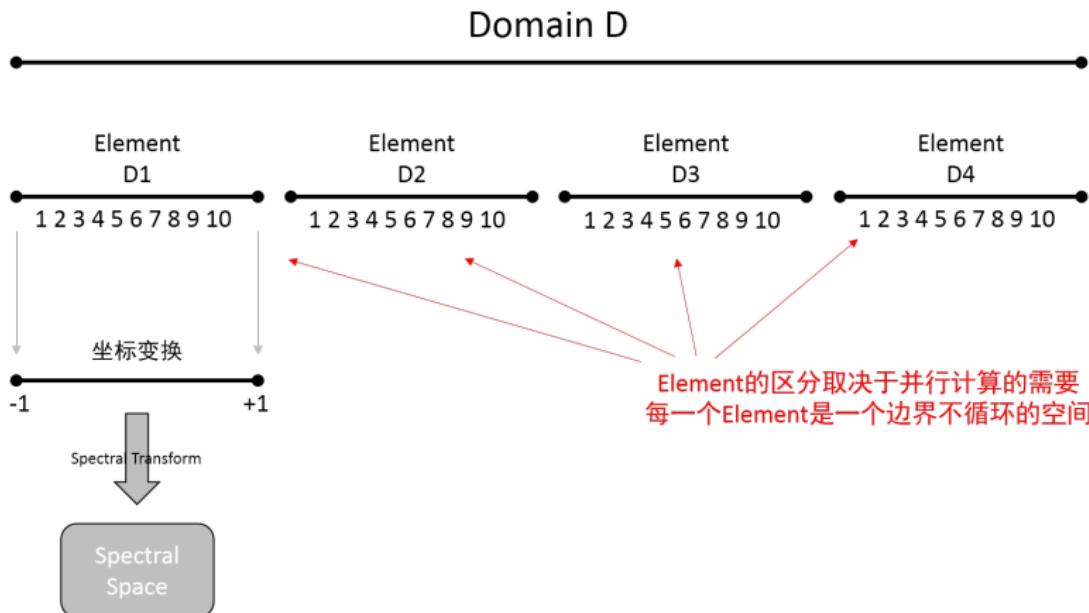
8 数值计算方法之比较

问题的提出

- 传统的 FD 方法容易分区，并行计算效率高，但计算精度低
- 传统的谱方法（Fourier, 球谐）要求变量定义在连续、循环空间上
- 传统的谱方法极大地限制了以分区为基础算法的并行计算
- 可否结合：FD 方法的并行计算高效性 + 谱方法的高精度？

谱元法的基本图景

SEM (Spectral Element Method)



SEM 中常用的基函数

正交基函数

- Lagrange Polynomials (拉格朗日多项式)
- Legendre Polynomials (勒让德多项式)
- Chebyshev Polynomials (切比雪夫多项式)

Chebyshev vs. Legendre Polynomials

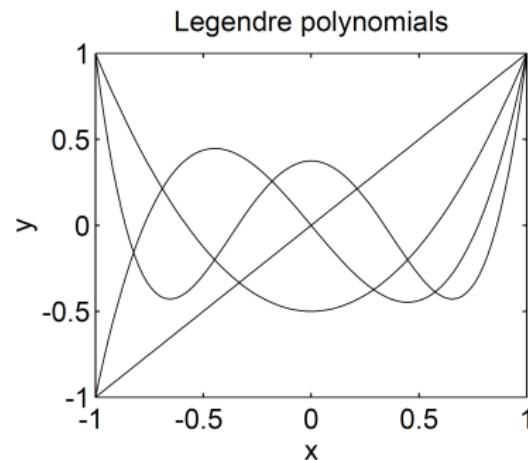
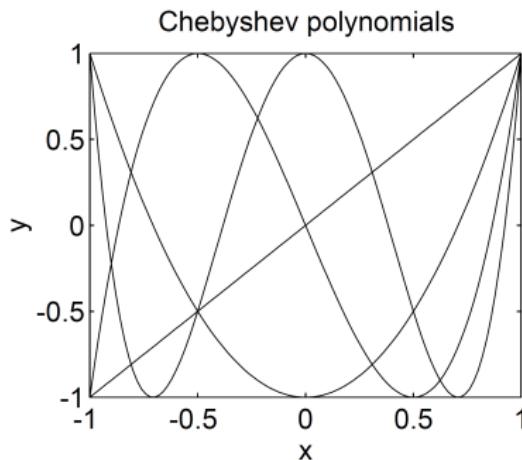


Figure: 切比雪夫多项式与勒让德多项式 ($n \in [0, 1, 2, 3, 4]$)

SEM 的优缺点

- 1 保留了谱方法的高精度
- 2 极大地提高了并行计算效率
- 3 但也仍保留了所有谱方法的缺点，比如缺少守恒性考量
- 4 Element 的边界连续性不好，谱方法在边界处计算精度有所降低

1 气候模式与平流方程

2 时间积分

3 有限差分方法

4 谱方法

5 有限体积法

6 谱元法

7 半拉格朗日法

- Semi-Lagrangian Method Framework
- Interpolation Approaches

8 数值计算方法之比较

问题的提出

- 针对平流问题，可否引入 Lagrangian 观点？
- FD 精度低、谱方法会出现 Gibbs 效应
- 更快、更准确地计算物质的输送
- 仍然留在 Grid Space 中，为了与其它过程的计算相融合

半拉格朗日法的基本图景

SL (Semi-Lagrangian Method)

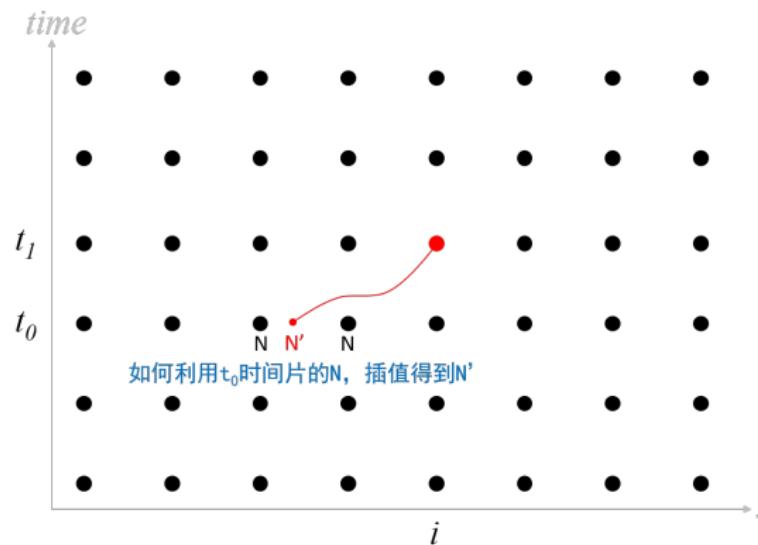


Figure: 如何插值是 SL 法的关键

SL 中常用的插值方法

- 3pt/4pt Lagrange Polynomials (3 点或 4 点拉格朗日多项式)
- Spline Interpolation (样条插值)
- Hermite Interpolation (埃尔米特插值)
- 可增加 Limiter 限制震荡，比如单调性限制、正定等

SL 的优缺点

- 计算速度快、精度高
- 时间步长不受 CFL 条件限制，可以拉开时间步长大踏步积分
- 容易构造正定特征，适合水汽和化学污染物等
- 缺少守恒性考量

1 气候模式与平流方程

2 时间积分

3 有限差分方法

4 谱方法

5 有限体积法

6 谱元法

7 半拉格朗日法

8 数值计算方法之比较

计算精度 vs. 计算代价

