# 有关数学知识汇总

## 拉普拉斯算子（Laplace operator）

### 定义

在数学和物理中，拉普拉斯算子或拉普拉斯算符（Laplace operator， Laplacian）是一个微分算子，通常写成△或▽2；拉普拉斯算子有许多用途，此外也是[椭圆型算子](http://zh.wikipedia.org/w/index.php?title=%E6%A4%AD%E5%9C%86%E5%9E%8B%E7%AE%97%E5%AD%90&action=edit&redlink=1)中的一个重要例子。在物理中，常用于[波方程](http://zh.wikipedia.org/wiki/%E6%B3%A2%E6%96%B9%E7%A8%8B)的[数学模型](http://zh.wikipedia.org/wiki/%E6%95%B0%E5%AD%A6%E6%A8%A1%E5%9E%8B)、[热传导方程](http://zh.wikipedia.org/wiki/%E7%86%B1%E5%82%B3%E5%B0%8E%E6%96%B9%E7%A8%8B)以及[亥姆霍兹方程](http://zh.wikipedia.org/wiki/%E4%BA%A5%E5%A7%86%E9%9C%8D%E5%85%B9%E6%96%B9%E7%A8%8B)。在[静电学](http://zh.wikipedia.org/wiki/%E9%9D%9C%E9%9B%BB%E5%AD%B8)中，[拉普拉斯方程](http://zh.wikipedia.org/wiki/%E6%8B%89%E6%99%AE%E6%8B%89%E6%96%AF%E6%96%B9%E7%A8%8B)和[泊松方程](http://zh.wikipedia.org/wiki/%E6%B3%8A%E6%9D%BE%E6%96%B9%E7%A8%8B)的应用随处可见。在[量子力学](http://zh.wikipedia.org/wiki/%E9%87%8F%E5%AD%90%E5%8A%9B%E5%AD%B8)中，其代表[薛定谔方程式](http://zh.wikipedia.org/wiki/%E8%96%9B%E4%B8%81%E6%A0%BC%E6%96%B9%E7%A8%8B%E5%BC%8F)中的[动能](http://zh.wikipedia.org/wiki/%E5%8B%95%E8%83%BD)项。在数学中，经拉普拉斯算子运算为零的[函数](http://zh.wikipedia.org/wiki/%E5%87%BD%E6%95%B8)称为[调和函数](http://zh.wikipedia.org/wiki/%E8%B0%83%E5%92%8C%E5%87%BD%E6%95%B0)；拉普拉斯算子是[霍奇理论](http://zh.wikipedia.org/wiki/%E9%9C%8D%E5%A5%87%E7%90%86%E8%AB%96)的核心，并且是[德拉姆上同调](http://zh.wikipedia.org/wiki/%E5%BE%B7%E6%8B%89%E5%A7%86%E4%B8%8A%E5%90%8C%E8%B0%83)的结果。

定义

拉普拉斯算子是n维[欧几里得空间](http://zh.wikipedia.org/wiki/%E6%AC%A7%E5%87%A0%E9%87%8C%E5%BE%97%E7%A9%BA%E9%97%B4)中的一个二阶微分算子，定义为[梯度](http://zh.wikipedia.org/wiki/%E6%A2%AF%E5%BA%A6)（▽*f*）的[散度](http://zh.wikipedia.org/wiki/%E6%95%A3%E5%BA%A6)（▽·*f）*。因此如果*f*是[二阶可微](http://zh.wikipedia.org/wiki/%E5%AF%BC%E6%95%B0)的[实函数](http://zh.wikipedia.org/wiki/%E5%AE%9E%E5%87%BD%E6%95%B0)，则*f*的拉普拉斯算子定义为：

\Delta f = \nabla^2 f = \nabla \cdot \nabla f, （1）

*f* 的拉普拉斯算子也是[笛卡儿坐标系](http://zh.wikipedia.org/wiki/%E7%AC%9B%E5%8D%A1%E5%84%BF%E5%9D%90%E6%A0%87%E7%B3%BB)*xi*中的所有非混合二阶[偏导数](http://zh.wikipedia.org/wiki/%E5%81%8F%E5%AF%BC%E6%95%B0)：

\Delta f = \sum_{i=1}^n \frac {\partial^2 f}{\partial x^2_i}.（2）

作为一个二阶微分算子，拉普拉斯算子把 *C*k 函数映射到 *C*k-2 函数，对于*k* ≥ 2。表达式(1)（或(2)）定义了一个算子Δ : *C*k(**R**n) → *C*k-2(**R**n)，或更一般地，定义了一个算子Δ : *C*k(Ω) → *C*k-2(Ω)，对于任何[开集](http://zh.wikipedia.org/wiki/%E5%BC%80%E9%9B%86)Ω。

函数的拉普拉斯算子也是该函数的[黑塞矩阵](http://zh.wikipedia.org/wiki/%E9%BB%91%E5%A1%9E%E7%9F%A9%E9%98%B5)的[迹](http://zh.wikipedia.org/wiki/%E8%BF%B9)：

\Delta f = \mathrm{tr}(H(f)).\,\!

### 坐标表示式

二维空间

\Delta f = \frac{\partial^2f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}

其中**x**与**y**代表 x-y 平面上的笛卡儿坐标

另外极坐标的表示法为：

 \Delta f 
= {1 \over r} {\partial \over \partial r} \left( r {\partial f \over \partial r} \right) 
+ {1 \over r^2} {\partial^2 f \over \partial \theta^2}

三维空间

**笛卡儿坐标系**下的表示法


\Delta f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} + \frac{\partial^2 f}{\partial z^2}.


**圆柱坐标系**下的表示法

 \Delta f 
= {1 \over \rho} {\partial \over \partial \rho}
  \left( \rho {\partial f \over \partial \rho} \right) 
+ {1 \over \rho^2} {\partial^2 f \over \partial \theta^2}
+ {\partial^2 f \over \partial z^2 }. 


**球坐标系**下的表示法

 \Delta f 
= {1 \over r^2} {\partial \over \partial r}
  \left( r^2 {\partial f \over \partial r} \right) 
+ {1 \over r^2 \sin \theta} {\partial \over \partial \theta}
  \left( \sin \theta {\partial f \over \partial \theta} \right) 
+ {1 \over r^2 \sin^2 \theta} {\partial^2 f \over \partial \phi^2}.


N维空间

在参数方程为*x=rθ*∈R*N*（其中*r*∈[0,+∞)以及*θ*∈*SN-1*）的***N* 维球坐标系**中，拉普拉斯算子为：

 \Delta f
= \frac{\partial^2 f}{\partial r^2}
+ \frac{N-1}{r} \frac{\partial f}{\partial r}
+ \frac{1}{r^2} \Delta_{S^{N-1}} f


其中△*SN-1*是*N-1*维球面上的[拉普拉斯-贝尔特拉米算子](http://zh.wikipedia.org/wiki/%E6%8B%89%E6%99%AE%E6%8B%89%E6%96%AF-%E8%B4%9D%E5%B0%94%E7%89%B9%E6%8B%89%E7%B1%B3%E7%AE%97%E5%AD%90)。我们也可以把{\partial^2 f \over \partial r^2}
+ \frac{N-1}{r} \frac{\partial f}{\partial r}的项写成\frac{1}{r^{N-1}} \frac{\partial}{\partial r} \Bigl(r^{N-1} \frac{\partial f}{\partial r} \Bigr)。

恒等式

如果*f*和*g*是两个函数，则它们的乘积的拉普拉斯算子为：

\Delta(fg)=(\Delta f)g+2((\nabla f)\cdot(\nabla g))+f(\Delta g).

*f*是径向函数*f*(*r*)且*g*是[球谐函数](http://zh.wikipedia.org/wiki/%E7%90%83%E8%B0%90%E5%87%BD%E6%95%B0)*Ylm*(θ,ϕ)，是一个特殊情况。这个情况在许多物理模型中有所出现。*f*(*r*)的梯度是一个径向矢量，而角函数的梯度与径向矢量相切，因此：

2(\nabla f(r))\cdot(\nabla Y_{lm}(\theta,\phi))=0.

球谐函数还是球坐标系中的拉普拉斯算子的角部分的特征函数：

\Delta Y_{\ell m}(\theta,\phi) = -\frac{\ell(\ell+1)}{r^2} Y_{\ell m}(\theta,\phi).

因此

\Delta( f(r)Y_{\ell m}(\theta,\phi) ) = \left(\frac{d^2f(r)}{dr^2} + \frac{2}{r} \frac{df(r)}{dr} - \frac{\ell(\ell+1)}{r^2} f(r)\right)Y_{\ell m}(\theta,\phi).

### 推广

#### 复杂空间上的实值函数

拉普拉斯算子可以用一定的方法推广到非欧几里得空间，这时它就有可能是[椭圆型算子](http://zh.wikipedia.org/w/index.php?title=%E6%A4%AD%E5%9C%86%E5%9E%8B%E7%AE%97%E5%AD%90&action=edit&redlink=1)，[双曲型算子](http://zh.wikipedia.org/w/index.php?title=%E5%8F%8C%E6%9B%B2%E5%9E%8B%E7%AE%97%E5%AD%90&action=edit&redlink=1)，或[超双曲型算子](http://zh.wikipedia.org/w/index.php?title=%E8%B6%85%E5%8F%8C%E6%9B%B2%E5%9E%8B%E7%AE%97%E5%AD%90&action=edit&redlink=1)。

在[闵可夫斯基空间](http://zh.wikipedia.org/wiki/%E9%97%B5%E5%8F%AF%E5%A4%AB%E6%96%AF%E5%9F%BA%E7%A9%BA%E9%97%B4)中，拉普拉斯算子变为[达朗贝尔算子](http://zh.wikipedia.org/wiki/%E8%BE%BE%E6%9C%97%E8%B4%9D%E5%B0%94%E7%AE%97%E5%AD%90)：

\square = 
{\partial^2 \over \partial x^2 } +
{\partial^2 \over \partial y^2 } +
{\partial^2 \over \partial z^2 } -
\frac {1}{c^2}{\partial^2 \over \partial t^2 }.


达朗贝尔算子通常用来表达[克莱因-高登方程](http://zh.wikipedia.org/wiki/%E5%85%8B%E8%8E%B1%E5%9B%A0-%E9%AB%98%E7%99%BB%E6%96%B9%E7%A8%8B)以及四维[波动方程](http://zh.wikipedia.org/wiki/%E6%B3%A2%E5%8A%A8%E6%96%B9%E7%A8%8B)。第四个项前面的符号是负号，而在欧几里德空间中则是正号。因子 *c* 是需要的，这是因为时间和空间通常用不同的单位来衡量；如果 *x* 方向用寸来衡量，*y* 方向用厘米来衡量，也需要一个类似的因子。

值域为复杂空间

#### 矢量值函数的拉普拉斯算子

拉普拉斯算子作用在矢量值函数上，其结果被定义为一个矢量，这个矢量的各个分量分别为矢量值函数各个分量的拉普拉斯，即

\nabla^2 \mathbf{A} = (\nabla^2 A_x, \nabla^2 A_y, \nabla^2 A_z)

更一般地，对没有坐标的矢量，我们用下面的方式定义（受[矢量恒等式](http://zh.wikipedia.org/wiki/%E5%90%91%E9%87%8F%E6%81%86%E7%AD%89%E5%BC%8F)的启发）：

 \nabla^2 \mathbf{A} = \nabla(\nabla \cdot \mathbf{A}) - \nabla \times (\nabla \times \mathbf{A}) 

也可用类似于[拉普拉斯－德拉姆算子](http://zh.wikipedia.org/wiki/%E6%8B%89%E6%99%AE%E6%8B%89%E6%96%AF%EF%BC%8D%E5%BE%B7%E6%8B%89%E5%A7%86%E7%AE%97%E5%AD%90)的方式定义，然后证明“[旋度](http://zh.wikipedia.org/wiki/%E6%97%8B%E5%BA%A6)的旋度”矢量恒等式．

#### 拉普拉斯－贝尔特拉米算子

拉普拉斯算子也可以推广为定义在黎曼流形上的椭圆型算子，称为拉普拉斯-贝尔特拉米算子。达朗贝尔算子则推广为伪黎曼流形上的双曲型算子。拉普拉斯–贝尔特拉米算子还可以推广为运行于张量场上的算子（也称为拉普拉斯–贝尔特拉米算子）。

另外一种把拉普拉斯算子推广到伪黎曼流形的方法，是通过拉普拉斯–德拉姆算子，它作用在微分形式上。这便可以通过外森比克恒等式来与拉普拉斯–贝尔特拉米算子联系起来。

## 导数与偏导数的区别

设有[二元函数](http://zhidao.baidu.com/search?word=%E4%BA%8C%E5%85%83%E5%87%BD%E6%95%B0&fr=qb_search_exp&ie=utf8)z=f(x,y)，点(x0,y0)是其定义域D内一点.把y固定在y0而让x在x0有增量△x，相应地函数z=f(x,y)有增量(称为对x的偏增量)△z=f(x0+△x,y0)-f(x0,y0)。 如果△z与△x之比当△x→0时的极限存在，那么此[极限值](http://zhidao.baidu.com/search?word=%E6%9E%81%E9%99%90%E5%80%BC&fr=qb_search_exp&ie=utf8)称为函数z=f(x,y)在(x0,y0)处对x的[偏导数](http://zhidao.baidu.com/search?word=%E5%81%8F%E5%AF%BC%E6%95%B0&fr=qb_search_exp&ie=utf8)。  
你这里[一元函数](http://zhidao.baidu.com/search?word=%E4%B8%80%E5%85%83%E5%87%BD%E6%95%B0&fr=qb_search_exp&ie=utf8)y=f(x)中求导称导数，和[偏导数](http://zhidao.baidu.com/search?word=%E5%81%8F%E5%AF%BC%E6%95%B0&fr=qb_search_exp&ie=utf8)的结果是一样的。[多元函数](http://zhidao.baidu.com/search?word=%E5%A4%9A%E5%85%83%E5%87%BD%E6%95%B0&fr=qb_search_exp&ie=utf8)中，才可以理解为真正的求[偏导数](http://zhidao.baidu.com/search?word=%E5%81%8F%E5%AF%BC%E6%95%B0&fr=qb_search_exp&ie=utf8)，比如你[多元函数](http://zhidao.baidu.com/search?word=%E5%A4%9A%E5%85%83%E5%87%BD%E6%95%B0&fr=qb_search_exp&ie=utf8)你必须说对某一个未知数求偏导数。

## 二阶混合偏导数



称为 *z* = *f* (*x*, *y*)的二阶偏导数



## 梯度

设体系中某处的物理参数（如[温度](http://baike.baidu.com/view/8193.htm)、[速度](http://baike.baidu.com/view/36819.htm)、[浓度](http://baike.baidu.com/view/63062.htm)等）为w，在与其垂直距离的dy处该[参数](http://baike.baidu.com/view/327406.htm)为w+dw，则称为该物理参数的梯度，也即该物理参数的变化率。如果参数为速度、浓度、温度或空间，则分别称为[速度梯度](http://baike.baidu.com/view/1216655.htm)、[浓度梯度](http://baike.baidu.com/view/268314.htm)、[温度梯度](http://baike.baidu.com/view/708561.htm)或[空间梯度](http://baike.baidu.com/view/3849753.htm)。

在[向量微积分](http://baike.baidu.com/view/7823334.htm)中，[标量场](http://baike.baidu.com/view/550627.htm)的梯度是一个[向量场](http://baike.baidu.com/view/550639.htm)。标量场中某一点上的梯度指向标量场增长最快的方向，梯度的长度是这个最大的变化率。更严格的说，从欧氏空间Rn到R的函数的梯度是在Rn某一点最佳的线性近似。在这个意义上，梯度是雅戈比矩阵的一个特殊情况。

在单变量的[实值函数](http://baike.baidu.com/view/1120153.htm)的情况，梯度只是[导数](http://baike.baidu.com/view/30958.htm)，或者，对于一个[线性函数](http://baike.baidu.com/view/2169890.htm)，也就是线的[斜率](http://baike.baidu.com/view/271319.htm)。

梯度一词有时用于[斜度](http://baike.baidu.com/view/479644.htm)，也就是一个[曲面](http://baike.baidu.com/view/324917.htm)沿着给定方向的倾斜程度。可以通过取向量梯度和所研究的方向的[点积](http://baike.baidu.com/view/2744555.htm)来得到斜度。梯度的数值有时也被称为梯度。

在[二元函数](http://baike.baidu.com/view/2268148.htm)的情形，设函数z=f(x,y)在平面区域D内具有一阶连续[偏导数](http://baike.baidu.com/view/1029405.htm)，则对于每一点P(x,y)∈D，都可以定出一个向量

(δf/x)\*i+(δf/y)\*j

这向量称为[函数](http://baike.baidu.com/view/15061.htm)z=f(x,y)在点P(x,y)的梯度，记作gradf(x,y)

类似的对三元[函数](http://baike.baidu.com/view/15061.htm)也可以定义一个：(δf/x)\*i+(δf/y)\*j+(δf/z)\*k 记为grad[f(x,y,z)]

梯度本意是一个向量（[矢量](http://baike.baidu.com/view/77474.htm)），当某一函数在某点处沿着该方向的[方向导数](http://baike.baidu.com/view/2094467.htm)取得该点处的最大值，即函数在该点处沿方向变化最快，变化率最大（为该[梯度的模](http://baike.baidu.com/view/3035084.htm)）。

## 散度（divergence）

div F=▽·F 在[矢量场](http://baike.baidu.com/view/241272.htm)F中的任一点M处作一个包围该点的任意闭合[曲面](http://baike.baidu.com/view/324917.htm)S，当S所限定的区域直径趋近于0时，比值∮F·dS/ΔV的极限称为矢量场F在点M处的散度，并记作div F

由散度的定义可知，div F表示在点M处的单位体积内散发出来的矢量F的[通量](http://baike.baidu.com/view/832399.htm)，所以div F描述了通量源的密度。

散度的重要性在于，可用于表征空间各点矢量场发散的强弱程度，当div F>0 ，表示该点有散发通量的正源；当div F<0 表示该点有吸收通量的负源；当div F=0，表示该点为无源场。

从定义中可以看出，散度是向量场的一种强度性质，就如同密度、浓度、温度一样，它对应的广延性质是一个封闭区域表面的通量，所以说散度是通量的体密度。物理上，散度的意义是场的有源性。某一点或某个区域的散度大于零，表示向量场在这一点或这一区域有新的通量产生，小于零则表示向量场在这一点或区域有通量湮灭。这样的点或区域分别称为向量场的正源（发散源）和负源（洞）。举例来说，假设将太空中各个点的热辐射强度向量看做一个向量场，那么某个热辐射源（比如太阳）周边的热辐射强度向量都指向外，说明太阳是不断产生新的热辐射的源头，其散度大于零。

散度等于零的区域称为无源场或管形场。流体力学中，散度为零的流体称为不可压缩流体，也就是说此流体中不会有一部分凭空消失或突然产生，每个微小时间间隔中流入一个微小体元的流体总量都等于在此时间间隔内流出此体元的流体总量。

## 通量

通量，是表示物质分子移动量的大小，指某种物质在每秒内通过每平方厘米的假想平面的摩尔或毫尔数。

## 交错网格（staggered grid）

交错网格就是将标量（如压力、温度和密度等）在正常的网格节点上存储和计算，而将速度的各分量分别放在错位后的网格上存储和计算，错位后的网格的中心位于原控制体积的界面上。

使用交错网格的目的，是为了解决在普通网格上离散控制方程时给计算带来的严重问题（一个高度方向非均匀的压力场在离散后的动量方程中的作用，与均匀压力场的作用一致，检测不出变化的压力场）。交错网格也是SIMPLE算法实现的基础。

## 共轭梯度法

共轭梯度法（Conjugate Gradient）是介于[最速下降法](http://baike.baidu.com/view/2568955.htm)与[牛顿](http://baike.baidu.com/view/1511.htm)法之间的一个方法，它仅需利用一阶[导数](http://baike.baidu.com/view/30958.htm)信息，但克服了最速下降法收敛慢的缺点，又避免了牛顿法需要存储和计算Hesse[矩阵](http://baike.baidu.com/view/10337.htm)并求逆的缺点，共轭梯度法不仅是解决大型[线性方程组](http://baike.baidu.com/view/325740.htm)最有用的方法之一，也是解大型非线性最优化最有效的[算法](http://baike.baidu.com/view/7420.htm)之一。 在各种优化算法中，共轭梯度法是非常重要的一种。其优点是所需存储量小，具有步收敛性，稳定性高，而且不需要任何外来参数。

### 简介

共轭梯度法最早是由Hestenes和Stiefle（1952）提出来的，用于解[正定](http://baike.baidu.com/view/20230.htm)[系数矩阵](http://baike.baidu.com/view/3503330.htm)的[线性方程组](http://baike.baidu.com/view/325740.htm)，在这个基础上，Fletcher和Reeves （1964）首先提出了解非线性最优化问题的共轭梯度法。由于共轭梯度法不需要[矩阵](http://baike.baidu.com/view/10337.htm)存储，且有较快的收敛速度和二次终止性等优点，现在共轭梯度法已经广泛地应用于实际问题中。

共轭梯度法是一个典型的[共轭方向](http://baike.baidu.com/view/6060360.htm)法，它的每一个搜索方向是互相共轭的，而这些搜索方向d仅仅是负梯度方向与上一次迭代的搜索方向的组合，因此，存储量少，计算方便

### 算法介绍

又称共轭斜量法，是解[线性代数](http://baike.baidu.com/view/32243.htm)[方程组](http://baike.baidu.com/view/314172.htm)和非[线性方程组](http://baike.baidu.com/view/325740.htm)的一种数值方法，例如对线性代数方程组 Ax=ƒ, (1)式中**A**为n阶[矩阵](http://baike.baidu.com/view/10337.htm)，x和ƒ为n维列向量,当**A**对称[正定](http://baike.baidu.com/view/20230.htm)时,可以证明求(1)的解X\*和求二次[泛函](http://baike.baidu.com/view/523281.htm)

[(2)](http://baike.baidu.com/picview/2565822/2565822/0/a583631ee9318dcb1bd576ce.html)

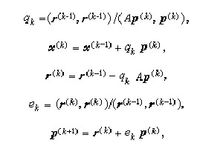
(2)

的 极小值问题是等价的。此处(x,у)表示向量x和у的内积。由此，给定了初始向量x（0），按某一方向去求(2)式取极小值的点x(1),就得到下一个迭 代值x(2),再由x(2)出发，求x(3)等等，这样来逼近x\*。若取求极小值的方向为F在 x(k=1,2,…)处的负梯度方向就是所谓最速下降法，然而理论和实际计算表明这个方法的收敛速度较慢，共轭梯度法则是在 x(k-1)处的梯度方向r(k-1)和这一步的修正方向p(k-1)所构成的二维平面内，寻找使F减小最快的方向作为下一步的修正方向p(k),即求极 小值的方向p（其第一步仍取负梯度方向）。计算公式为

[公式](http://baike.baidu.com/picview/2565822/2565822/0/b3f6cea2d5bbb292cbefd0b4.html)

公式

再逐次计算

[](http://baike.baidu.com/picview/2565822/2565822/0/b110e619f24eb545dab4bdb3.html)

公式

(k=1,2,…)。可以证明当i≠j时，

[公式](http://baike.baidu.com/picview/2565822/2565822/0/504ec7f96076e366252df2b9.html)

公式

从而平p(1),p(2)形成一共轭[向量](http://baike.baidu.com/view/77260.htm)组;r(0),r(1),…形成一[正交向量组](http://baike.baidu.com/view/5187897.htm)。后者说明若没有舍入误差的话，至多 n次迭代就可得到(1)的精确解。然而在实际计算中,一般都有舍入误差，所以r(0),r(1),…并不真正互相正交，而尣(0)尣(1),…等也只是逐步逼近(1)的真解,故一般将共轭梯度法作为[迭代法](http://baike.baidu.com/view/649495.htm)来使用。

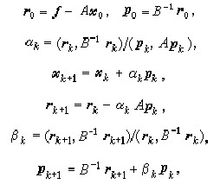
近来在解方程组(1)时,常将共轭梯度法同其他一些[迭代法](http://baike.baidu.com/view/649495.htm)结合作用。特别是对病态方程组这种方法往往能收到比较显著的效果。其方法是选取一对称[正定矩阵](http://baike.baidu.com/view/686970.htm)**B**并进行三角分解，得**B**=LLT。将方程组(1)化为 hу=b, (3)

此处y=lTx,b=l-1ƒ,h=l-1Al-T,而

[公式](http://baike.baidu.com/picview/2565822/2565822/0/948bcfc885e14a287f3e6f83.html)

公式

再对(3)用共轭梯度法，计算公式为

[](http://baike.baidu.com/picview/2565822/2565822/0/3bc6f750f24445631038c28b.html)

公式

k=0,1,2,…)适当选取**B**，当**B**很接近**A**时，h的[条件数](http://baike.baidu.com/view/1014731.htm)较之**A**大大减小,从而可使共轭梯度法的收敛[速度大](http://baike.baidu.com/view/892388.htm)为加快，由一些[迭代法](http://baike.baidu.com/view/649495.htm)的[矩阵](http://baike.baidu.com/view/10337.htm)分裂**A**=M -N，可选取M 为这里的**B**，例如对称超松弛迭代(SSOR)，强[隐式](http://baike.baidu.com/view/2852863.htm)迭代(SIP)等,这类方法常称为广义共轭梯度法或预条件共轭梯度法，它也可用于解代数特征值问题

## 多重网格方法(Multigridmethod)

多重网格方法是解微分方程的方法。这个方法的好处是在利用迭代法收敛结果的时候速度特别快。并且，不管是否对称，是否线性都无所谓。它的主要思想是在粗糙结果和精细结果之间插值。

前面介绍了Gauss–Seidel方法和Jacobi 方法，现在再用这两个方法来举例。尽管Gauss–Seidel (GS)方法converge更快一些，但其实对于维度很高的系统都很慢。Multigrid（MG）方法的思路是先把问题粗糙化，把原网格投影到一个比较简单的新网格上计算，等到快速收敛以后再经由Interpolation（插值）返回原来的系统。

对于某个工程数学问题（如泊松方程），可以归纳为线形方程Ax = b, A为n X n矩阵。那么，最终目的是得到所谓的x = A^(-1)b。定义e(t) = x – x(t)，当e(t)为小于某个值的时候，可以认为xconverge到了合适的值。但实际上我们比较的是相邻的值。

把A非奇异分解A = B – C，

Bx – Cx = b

x = B(-1) Cx + B^(-1) b

并分开求解x

Bx(t+1) - Cx(t)= b

x(t+1) = B^(-1) Cx(t) + B^(-1) b -（1）

把形如B^(-1) C的矩阵称作迭代矩阵，用M表示。

容易发现,

x(t+1) – x = Mx(t) + B^(-1) b – x = Mx(t) +Mx = M (x(t) – x) -（2）

以上式子与（1）式等价。

另外可令N = B^(-1):

x(t+1) = Mx(t) + Nb - (3)

不同的迭代方法其实就是A的不同分解法，反映到（3）式就是取不同的M和N值。

比如，在GS方法中B = D – L, C = U, 则(3)式为：

x(t+1) = (D - L)^(-1)Ux(t) + (D - L)^(-1)b - (4)

分析发现，当n的数值比较大时，以上收敛是极其缓慢的。假设p(i) 是序号为i的原系统有限元基函数（i = 1, 2, … , n），q(i)是粗糙化的网格(i = 1,2, … , m m<n).

一种粗糙化方法是构造矩阵H，使p = H \* q, H为m X n矩阵。令A’ = HAH^T, x’ = Hx, b’ = Hb. 则 A’ x’ = b’ 是一个m维粗化的网格系统。

更加具体来说，对于一个k维的问题，如果k小于一个指定的维数，那么直接用jacob等方法解救可以了。否则，粗糙化为更低的维度比如变为原来的二分之一，最后再把维数变回来。把mesh粗糙化的过程叫做Downv-cycle (从k维到k/2维), 反之叫Up v-cycle.（从k维到2k维）。

## 柯朗-弗里德里希斯-列维条件（Courant–Friedrichs–Lewy condition）

In mathematics, the **Courant–Friedrichs–Lewy condition (CFL condition)** is a necessary condition for stability while solving certain partial differential equations (usually hyperbolic PDEs) numerically by the method of finite differences. It arises in the numerical analysis of explicit time-marching schemes, when these are used for the numerical solution. As a consequence, the time step must be less than a certain time in many explicit time-marching computer simulations, otherwise the simulation will produce incorrect results. The condition is named after Richard Courant, Kurt Friedrichs, and Hans Lewy who described it in their 1928 paper. This condition is an example of explicit time integration where the function that defines governing equation is evaluated at the current time

## 分裂算子（Split the operators）

[算子](http://zhidao.baidu.com/search?word=%E7%AE%97%E5%AD%90&fr=qb_search_exp&ie=utf8)分裂法（又称分数步法）是在计算有限差分的时候用到的。是苏联人提出的。它一般是对于多维问题提出来的。

对于二维问题：算法的基本思想是将一个时间[步长](http://zhidao.baidu.com/search?word=%E6%AD%A5%E9%95%BF&fr=qb_search_exp&ie=utf8)分成两个一半的时间[步长](http://zhidao.baidu.com/search?word=%E6%AD%A5%E9%95%BF&fr=qb_search_exp&ie=utf8)，然后在第一个半时间[步长](http://zhidao.baidu.com/search?word=%E6%AD%A5%E9%95%BF&fr=qb_search_exp&ie=utf8)里，只看x方向的影响，在后半个时间步长里，再看y方向的影响。

## Free field

In [physics](http://en.wikipedia.org/wiki/Physics) a **free field** is a [field](http://en.wikipedia.org/wiki/Field_%28physics%29) **without** [**interactions**](http://en.wikipedia.org/wiki/Fundamental_interaction), which is described by the terms of motion and mass.

In [classical physics](http://en.wikipedia.org/wiki/Classical_physics), a **free field** is a field whose [equations of motion](http://en.wikipedia.org/wiki/Equations_of_motion) are given by [linear](http://en.wikipedia.org/wiki/Linear) [partial differential equations](http://en.wikipedia.org/wiki/Partial_differential_equation). Such linear PDE's have a unique solution for a given initial condition.

In [quantum field theory](http://en.wikipedia.org/wiki/Quantum_field_theory), an [operator valued distribution](http://en.wikipedia.org/wiki/Operator_valued_distribution) is a **free field** if it satisfies some linear partial differential equations such that the corresponding case of the same linear PDEs for a classical field (i.e. not an operator) would be the [Euler-Lagrange equation](http://en.wikipedia.org/wiki/Euler-Lagrange_equation) for some [quadratic](http://en.wikipedia.org/wiki/Quadratic_polynomial) [Lagrangian](http://en.wikipedia.org/wiki/Lagrangian). We can differentiate distributions by defining their derivatives via differentiated [test functions](http://en.wikipedia.org/wiki/Test_function). See [Schwartz distribution](http://en.wikipedia.org/wiki/Schwartz_distribution) for more details. Since we are dealing not with ordinary distributions but operator valued distributions, it is understood these PDEs aren't constraints on states but instead a description of the relations among the smeared fields. Beside the PDEs, the operators also satisfy another relation, the commutation/anticommutation relations.

## Helmholtz-Hodge分解（Helmholtz-Hodge Decomposition）

## 显式积分和隐式积分

这是ansys里面的两种求解方法。

大多数非线性动力学问题一般多是采用显式求解方法，特别是在求解大型结构的瞬时高度非线性问题时，显示求解方法有明显的优越性。下面先简要对比一下隐式求解法和显示求解法。动态问题涉及到时间域的数值积分方法问题。在80年代中期以前，人们基本上采用纽曼法进行时间域的积分。根据纽曼法，位移、速度和加速度有着如下关系：

u(i+1)=u(i)+△t\*v(i)[(1—2p)a(i)+2p\*a(i+1)] (1)

v(i+1)=V(i)+△t[(1-2q)a(i)+2qa(i+1)] (2)

上面式子中 u(i+1),u(i)分别为当前时刻和前一时刻的位移，v(i+1)和V(i)为当前时刻和前一时刻的速度，a(i+1)和a(i)为当前时刻和前一时刻的加速度，p和q为两个待定参数，△t为当前时刻与前一时刻的时问差，符号 \* 为乘号。由式(1)和式(2)可知，在纽曼法中任一时刻的位移、速度、加速度都相互关联，这就使得运动方程的求解变成一系列相互关联的非线性方程的求解，这个求解过程必须通过迭代和求解联立方程组才能实现。这就是通常所说的隐式求解法。隐式求解法可能遇到两个问题。一是迭代过程不一定收敛，二是联立方程组可能出现病态而无确定的解。隐式求解法最大的优点是它具有无条件稳定性，即时间步长可以任意大。

如果采用中心差分法来进行动态问题的时域积分，则有如下位移、速度和加速度关系式：

u(i+1)=2u(i)-u(i-1)+a(i)(△t)^2 (3)

v(i+1)=[u(i+1)-u(i-1)]／2(△t) (4)

式中u(i-1)，为i-1时刻的位移。由式(3)可以看出，当前时刻的位移只与前一时刻的加速度和位移有关，这就意味着当前时刻的位移求解无需迭代过程。另外，只要将运动过程中的质量矩阵和阻尼矩阵对角化，前一时刻的加速度求解无需解联立方程组，从而使问题大大简化，这就是所谓的显式求解法。显式求解法的优点是它既没有收敛性问题，也不需要求解联立方程组，其缺点是时间步长受到数值积分稳定性的限制，不能超过系统的临界时间步长。

隐式求解法不考虑惯性效应[C]和[M]。对于线性问题，无条件稳定，可以用大的时间步。对于非线性问题，通过一系列线性逼近（Newton-Raphson）来求解；要求转置非线性刚度矩阵[K]，收敛时候需要小的时间步，对于高度非线性

问题无法保证收敛。因此，隐式求解一般用于线性分析和非线性结构静动力分析，包括结构固有频率和振型计算。 ansys使用的Newmark时间积分法即为隐式求解法。

显示求解法是ansys/ls-dyna中主要的求解方法，用于分析大变形、瞬态问题、非线性动力学问题等。对于非线性分析，显示求解法有一些基本的特点，如：块质量矩阵需要简单的转置；方程非耦合，可以直接求解；无须转置刚度矩阵，所有的非线性问题（包括接触）都包含在内力矢量中；内力计算是主要的计算部分；无效收敛检查；保存稳定状态需要小的时间步。（此处我也不是很理解，仅供你参考）。

## 单点积分和全积分

ansys作为一种有限单元法，它是一种离散化的数值解法。

有限单元法中，每一单元的特性用单元刚度矩阵来表示，每一结构构件的力与位移之间的关系不是精确推导出来的，而是利用每一单元中近似的位移函数得到节点位移，然后计算积分点应变和应力，输出时才根据用户请求将积分点结果复制或线性外推至单元的节点上。因此，有限单元法是一种近似的数值方法。先看一下积分点的概念：

计算刚度矩阵需要进行数值积分，Ansys采用高斯积分法，即采用各积分点处函数值与积分系数乘积之和，因此积分点也称高斯积分点。积分点位置的确定比较复杂，它是勒让德多项式Ln（x）的n个不同的实根，即需要求解勒让德多项式。对于面、体单元，在积分点处计算单元结果也比较精确。由此可知，积分点与节点完全不同，不同单元积分点位置也不一样，个别梁单元也没有积分点。

Gauss 积分阶数低于被积函数所有项次精确积分所需阶数的积分称为缩减积分,简单地说就是数值积分采用比精确积分要求少的积分点数。实际计算表明，采用缩减积分往往可以取得较完全精确积分更好的精度。

因此，所谓单点积分和全积分实际上指的是高斯积分时所采用的积分点的个数。这样说来，单点积分和全积分与显示求解法和隐式求解法没有本质的联系。

只不过，在显示动力分析中最消耗CPU的一项就是单元的处理。由于积分点的个数与CPU时间成正比，采用简化积分的单元便可以极大的节省数据存储量和运算次数，进而提高运算效率。除节省CPU外，单点积分单元在大变形分析中同样有效，ansys/ls-dyna单元能承受比标准ansys隐式单元更大的变形。因此，每种显示动力单元确省为单点积分。但单点积分有两个缺点：1.出现零能模型（沙漏模态）；2.应力结果精确度与积分点相关。为了控制沙漏，可以采用全积分单元。

总结一下，显示求解法、隐式求解法与单点积分、全积分不是一个层次上的概念。

我们在求解问题的时候应先根据我们的问题类型来决定是采用显示求解法还是隐式求解法。如果是采用显示求解法，默认是单点积分，如果产生了沙漏，改用全积分。