

硕士学位论文

基于广义线性法的结构动响应数值算法设计

**DESIGN OF NUMERICAL ALGORITHMS
FOR STRUCTURAL DYNAMICS BASED
ON THE GENERAL LINEAR METHOD**

李金泽

哈尔滨工业大学
2017 年 12 月 19 日

国内图书分类号: O332
国际图书分类号: 62-1

学校代码: 10213
密级: 公开

工学硕士学位论文

基于广义线性法的结构动响应数值算法设计

硕士研究生: 李金泽

导师: 于开平教授

申请学位: 工学硕士

学科: 一般力学与力学基础

所在单位: 航天学院 航天工程与力学系

答辩日期: 2017 年 12 月 19 日

授予学位单位: 哈尔滨工业大学

Classified Index: O332

U.D.C: 62-1

Dissertation for the Master's Degree in Engineering

DESIGN OF NUMERICAL ALGORITHMS FOR STRUCTURAL DYNAMICS BASED ON THE GENERAL LINEAR METHOD

Candidate:	Li Jinze
Supervisor:	Professor Yu Kaiping
Academic Degree Applied for:	Master of Engineering
Specialty:	General Mechanics
Affiliation:	School of Astronautics
Date of Defence:	June 2018
Degree-Confering-Institution:	Harbin Institute of Technology

摘 要

摘要的字数（以汉字计），硕士学位论文一般为 500 ~ 1000 字，博士学位论文为 1000 ~ 2000 字，均以能将规定内容阐述清楚为原则，文字要精练，段落衔接要流畅。摘要页不需写出论文题目。英文摘要与中文摘要的内容应完全一致，在语法、用词上应准确无误，语言简练通顺。留学生的英文版博士学位论文中应有不少于 3000 字的“详细中文摘要”。

关键词是为了文献标引工作、用以表示全文主要内容信息的单词或术语。关键词不超过 5 个，每个关键词中间用分号分隔。（模板作者注：关键词分隔符不用考虑，模板会自动处理。英文关键词同理。）

关键词：T_EX；L^AT_EX；CJK；嗨！；thesis

Abstract

An abstract of a dissertation is a summary and extraction of research work and contributions. Included in an abstract should be description of research topic and research objective, brief introduction to methodology and research process, and summarization of conclusion and contributions of the research. An abstract should be characterized by independence and clarity and carry identical information with the dissertation. It should be such that the general idea and major contributions of the dissertation are conveyed without reading the dissertation.

An abstract should be concise and to the point. It is a misunderstanding to make an abstract an outline of the dissertation and words “the first chapter”, “the second chapter” and the like should be avoided in the abstract.

Key words are terms used in a dissertation for indexing, reflecting core information of the dissertation. An abstract may contain a maximum of 5 key words, with semi-colons used in between to separate one another.

Keywords: T_EX, L^AT_EX, CJK, template, thesis

物理量名称及符号表

表 1 国际单位制中具有专门名称的导出单位

量的名称	单位名称	单位符号	其它表示实例
频率	赫 [兹]	Hz	s-1

目 录

摘 要	I
Abstract	II
物理量名称及符号表	III
第 1 章 绪论	1
1.1 课题来源	1
1.2 课题研究背景、目的和意义	1
1.2.1 结构动响应问题的工程背景	1
1.2.2 结构动响应数值算法的数学力学背景	2
1.2.3 课题研究的意义	3
1.3 国内外研究现状	3
1.4 本文主要研究内容	4
第 2 章 积分算法的设计及基本分析	6
2.1 广义线性法	6
2.1.1 线性多步法	9
2.1.2 Runge-Kutta 法	14
2.1.3 One-leg 方法	16
2.1.4 扩展的向后微分格式	17
2.1.5 两步 Runge-Kutta 方法	19
2.1.6 多步 Runge-Kutta 方法	20
2.1.7 Peer 方法	21
2.2 相容性	21
2.3 稳定性	26
2.4 收敛性	32
2.5 精度分析	32
2.5.1 数值耗散	32
2.5.2 数值弥散	32
2.6 超调行为	32

2.7 虚假根分析	32
2.8 直接积分法的放大矩阵	32
2.9 本章小结	32
第 3 章 结构动响应数值算法	33
3.1 有限差分法 (线性多步法)	35
3.1.1 向前欧拉法	36
3.1.2 向后欧拉法	37
3.1.3 广义梯形算法	37
3.1.4 中点公式	38
3.1.5 中心差分法	39
3.1.6 Houbolt 算法	39
3.1.7 带终点加速度的中点公式	40
3.1.8 Newmark 算法	41
3.1.9 Wilson- θ 法	47
3.1.10 Park 算法	49
3.1.11 Hilber-Hughes-Taylor- α 算法	49
3.1.12 Wood-Bosak-Zienkiewicz- α 算法	50
3.1.13 基于速度更新的 γ_s 格式	51
3.1.14 带中点加速度的中点公式	52
3.1.15 三参数优化算法/CH- α 法	52
3.1.16 广义 α 算法	55
3.2 加权残值法	56
3.2.1 ρ -算法	56
3.2.2 单步隐式的 θ_1 格式	58
3.2.3 高阶导数的单步显式算法	60
3.2.4 单步单解算法框架/GSSSS 算法	63
3.3 时间子步策略	64
3.3.1 BEF 算法	64
3.3.2 BADF-TR 算法	65
3.3.3 GCD-BDF2 算法	65
3.3.4 Soares-Jr 型算法	66

3.4 预测-校正算法	70
3.5 结构依赖型直接积分法	70
3.6 复合型子步隐式算法	70
3.6.1 TR-BDF2 算法	71
3.6.2 TR-TR-Houbolt 算法	72
3.6.3 BDF2-Newmark 算法	73
3.6.4 BDF 型积分算法	73
3.6.5 TTBDf 算法	76
3.6.6 BDF2-BDF3 算法	77
3.6.7 BDF3BDF2-Newmark 算法	78
3.6.8 CTFEA 算法	78
3.6.9 TR-BDF2-Houbolt 算法	80
3.6.10 TR-NBDDF-N 算法	81
3.6.11 TR-RP 算法	82
3.7 复合型子步显式算法	82
3.7.1 Noh-Bathe 两子步算法	82
3.7.2 Kwon-Lee 四子步算法	84
3.8 本章小结	85
第 4 章 基于广义线性法的结构动响应数值算法设计	86
4.1 本章小结	86
第 5 章 数值算例	87
5.1 本章小结	87
结 论	88
参考文献	89
攻读硕士学位期间发表的论文及其他成果	101
哈尔滨工业大学学位论文原创性声明和使用权限	104
致 谢	105

第 1 章 绪论

1.1 课题来源

根据导师在研一期间的指导及个人兴趣、基础知识的储备出发，通过查阅相关资料并在导师的指导下共同商定此论文题目。

在大四毕业后的暑假期间根据导师推荐的几篇文章^[1]，我开始了学习结构动力学运动方程数值求解算法。同时，在研一上学期由刘伟老师讲解的《高等结构动力学》课程中，我再一次接触到了结构动力学方程的数值求解。同时在该课程实验中，我也第一次发现了满足一些看似可行的直接积分法，求解出的响应有时却并不可靠，例如 Wilson- θ 法。它的这种特性由于其本身的数值耗散和弥散特性造成的。这次学习让我对求解结构动力学方程的数值算法产生了兴趣。同时，我也知道力学对数学，尤其是计算数学的要求是很高的。于是在研究生期间，又选了很多数学专业的核心课程（非线性数值分析、现代常微分方程理论、小波分析）进行学习，为自己后面进一步的探究数值算法铺平道路。这些数学课程的学习中，我愈发的觉得结构动力学方程的直接积分法的研究还有很多工作可做。特别值得一提的是，似乎在力学中我们更多的是借鉴了线性多步法来发展直接积分法，而使用 Runge-Kutta 法的思想发展结构动力学求解算法少之甚少。特别就是由 Butcher J. C. 在 1966 年提出的更具一般性的算法框架去求解常微分方程^[2]。在该框架下，线性多步法和 Runge-Kutta 法都是其特例。这类算法应用到求解结构动力学运动方程还鲜有文献。这也是进行本文研究的一个主要目标。

1.2 课题研究背景、目的和意义

1.2.1 结构动响应问题的工程背景

动力学问题在国民经济和科学技术的发展中有着广泛的应用领域。最常见遇到的是结构动力学问题，它有两类研究对象。一类是在动力状态下工作的机械或结构，例如高速旋转的电机，汽轮机，离心压缩机，冲压机床，以及高速运行的车辆，飞行器等，它们承受本身惯性及与周围介质或结构相互作用的动力载荷。如何保证它们运行的平稳性及结构的安全性，是极为重要的课题。另一类是承受动力载荷作用的工程结构，例如建于地面的高层建筑，反应塔和管道，核电站的安全壳和热交换器。近海工程的海洋石油平台等，他们可能承受强风，水流，地震以及

波浪等各种动力载荷的作用。这些结构的破裂、倾覆和垮塌等事故的发生，将给人民的生命财产造成巨大的损失。正确分析和设计这类结构，在理论上和实际上都是具有重要意义的课题。动力学研究的另一重要领域是波在介质中的传播问题。它是研究短暂作用于介质边界或内部的载荷所引起的位移和速度的变化，如何在介质中向周围传播，以及在界面上如何反射，折射等规律。它的研究在结构的抗震设计，人工地震勘测，无损探伤等领域都有广泛的应用背景，因此也是近二十都年一直受到工程和科技界密切关注的课题。

1.2.2 结构动响应数值算法的数学力学背景

一般情况下，大多数工程问题的数学建模后得到的微分方程往往是不可能求出解析解的；或者说，花费很大的努力求得的解析解是不经济的。这也导致了在实际应用中，应用数值算法求解得到的微分方程是一种必要手段。在结构动力学运动方程的求解过程中更是这种情况。

结构动力学数值计算主要有两大类传统方法^[3]。一是在空间域使用有限元离散后，基于大多数工程结构动态响应主要以低频为主的假设，使用模态分解和叠加的步骤，给出模态截断后的动响应，可归类于近似解析法。这类方法适合于比例阻尼假设情况，适合于长时间、持续动载荷作用问题，以及大多数以低频响应为主的工程问题。但对中高频激励问题（基于有限元模型，航天工程更关注的中高频多数指 200-2000HZ 范围）的计算尚没有很好解决。主要原因是：2000HZ 以内可能有很多阶模态，高阶模态对参数变化更为敏感，同时要求更细密的有限元网格^[3]。此外，高阶模态阻尼比的确定也还没有合理的模型，存在精度和计算量两方面的问题。另外一种方法是在时间域使用有限差分，或者使用时间有限元离散，得到时间逐步递推的计算步骤，还有一类是时空域同时离散的时空有限元方法。这类方法属于数值方法，不仅适用于比例阻尼也适用于非比例阻尼、非线性情况。这类方法也将是本文分析和研究的重点。

目前在大型结构的瞬态动力学、非线性动力学响应数值计算问题上，比较有效的方法仍然是时域内的直接积分方法，又称逐步积分法。该类方法，有合适的精度、合适的计算量以及适合于大多数实际工程问题。也就是说算法数值计算的整体性能好、适用范围广，因此一直受到计算力学、计算数学工作者和工程界的重视。

1.2.3 课题研究的意义

目前, 用线性多步法的理论去发展直接积分法的理论比较完善。特别是经过 Dahlquist 的发展推广^[4], 得到了如下结论^[5]:

- (1) 一个显式的、A-稳定的线性多步法是不存在的。
- (2) 一个三阶及其以上精度的 A-稳定的线性多步法是不存在的。
- (3) 带有最小误差常数的二阶精度的 A-稳定的线性多步法是梯形积分规则。

于是, 想要利用线性多步法的理论去发展具有 A-稳定的算法已经是不可能的。然而, 在数学分支—常微分方程数值解的理论中, 我们也得知通过适当的构造, 广义线性法可以突破上述线性多步法的障碍, 理论上可以实现任意阶精度下, 其数值方法仍能保持一定好的数值特性, 例如 A-稳定性, 甚至 L-稳定性。特别地, 作为广义线性法的一个特例—Runge-Kutta 法, 也可以实现上述特性。故本文的研究意义在于, 将广义线性法应用到结构动力学运动方程求解中, 进而推导出新的求解格式。这些新的算法格式在数学分析框架下具有良好的性质, 但应用到力学问题上, 可能还需要进行一定的改进和完善。同时, 这些新的数值格式也将在力学背景下进行算法的优劣分析, 比如, 其数值高频耗散、相对周期误差等。特别地, 本文也将尝试利用广义线性法中对于非线性常微分方程、刚度问题的分析借鉴到力学中的非线性动力学、刚度硬化问题的分析上。这样可以对现有直接积分法在理论分析非线性问题时, 提供一个理论判断依据。

1.3 国内外研究现状

目前, 国内外在广义线性法和结构动力学数值算法的发展中, 几乎都是相对独立的。鲜有研究者将二者结合起来。但过去的几十年也有一些研究学者将 Runge-Kutta 法应用到结构动力学数值求解中, 进而构造了一些新的数值算法。

周树荃和高科华^[6] 在论文中利用 3 级 3 阶半隐式 Runge-Kutta 法求解结构动力学问题, 并利用多项式预处理共轭梯度法求解有关代数方程组, 提出了半隐式 RK 型并行直接积分法 (RK33P) . 与相应的串行算法 RK33S 算法进行比较, 发现当求解系统的阶数为 103-104, 其加速度比可达 24-27。

Christoph L. 和 Simeon B.^[7,8] 利用 Runge-Kutta-Nyström 方法求解动力学中的刚性力学系统, 该方法一般化了著名的 Störmer 规则, 并且使得求解的稳定域最大化。结果表明, 文章提出的 RKN 方法具有一定的可比性和高效性。

Dopico D. 和 Lugris U.^[9] 等分析了在实时多体结构系统中具有遗传 Runge-

Kutta 稳定性的广义线性方法 (IRK) 的两个应用。文章问答了 IRK 方法是否适合求解实时结构响应以及该方法是否比传统的 Newmark 家族方法更优。

吴志桥等人^[10]将几种具有不同稳定性的 Runge-Kutta 方法应用到结构动力学的数值求解中, 并且使用了减小计算量的两种方法: 使用单对角隐式 Runge-Kutta 方法和应用转化矩阵, 算例表明在精确解上较小的物理阻尼能有效抑制高频分量, 但对各种直接积分方法的影响较小, 较高精度的 L-稳定 Runge-Kutta 方法能有效抑制高频分量的同时高精度的求解低频振动。而且在他的博士论文^[11]中, 通过研究 Runge-Kutta 方法对结构动力学运动方程的求解格式, 得到 A-稳定与直接积分法中的无条件稳定是等价的, 而 L-稳定格式包含数值阻尼。

Yin S. H.^[12]通过在结构动力学数值求解格式中构造位移和速度显式更新公式, 推导出带四个参数的放大矩阵, 同时利用 Runge-Kutta 方法求解对应的结构动力学运动微分方程得到其状态向量的迭代格式, 进而得到其迭代矩阵, 最后通过让放大矩阵和迭代矩阵对应风量相等, 得到其位移速度更新公式中的四个参数, 结果表明其参数是结构依赖的。

郭静和邢誉峰^[13]在 2014 年利用 2 级 4 阶隐式 Guass-Legendre 辛 Runge-Kutta 方法 (GLSRK) 求解有阻尼和外荷载情况下的线性动力学运动方程。并首次给出了 Guass-Legendre 辛 RK 方法和经典 RK 方法的谱半径和单步相位误差的显式表达式。通过算例表明, 辛 RK 方法比经典 RK 方法优越, 尤其是在运动学特性和长时间数值模拟方面尤为显著; 但是该算法与传统的 Newmark- β 法相比, 计算量和储存量过于庞大, 这也在一定程度上限制了该方法的应用。

针对上述问题, 黄策、富明慧^[14]等人将郭静等人提出的四阶隐式 Guass-Legendre 辛 RK 法进行优化, 先通过消元, 降低问题的规模, 再利用消元后得到的方程系数矩阵正定对称的特点, 采用预处理共轭梯度法求解。与原算法相比, 改进后的算法大幅度降低了储存量和计算量; 同时与 Newmark- β 法相比, 在未明显增加计算量的前提下, 大幅度提高了计算精度。

可以看出, 国内外利用 Runge-Kutta 方法去求解结构动力学方程的研究也是较为零散的, 还有大量的工作可做。

1.4 本文主要研究内容

本文将从数学常微分方程数值方法—广义线性法和力学中结构动力学运动方程的结构依赖型直接积分法两个角度出发, 针对动力学运动方程的数值解法进行

探究。本论文的主要研究内容有：

- 系统学习和探究广义线性法的体系框架。总结广义线性法至今为止的研究成果和应用领域。同时，以广义线性法的知识框架去进一步学习和研究线性多步法。
- 对现阶段新兴的结构依赖型算法进一步系统研究。挖掘新的具有实用价值的算法。同时对利用其它构造思想去发展结构动力学数值解法的文献进行总结和归纳。
- 将广义线性法的几类主要分支算法进行研究，并尝试将其应用到结构动力学数值求解中，以期获得很好的数值算法。目前根据现有文献研究表明，这种思路是可行的，但可能需要做进一步的细致工作研究。
- 将广义线性法对非线性系统的刻画应用到直接积分法求解结构动力学非线性系统的分析。
- 将上述发展算法进行 C++ 或 MATLAB 代码实现，得到其相应的工具包。

第 2 章 积分算法的设计及基本分析

对于结构动力学运动方程

$$M\ddot{U}(t) + C\dot{U}(t) + KU(t) = F(t) \quad (2-1)$$

的数值求解及其数值算法的设计分析，许多的基本概念借助于数学上对一阶微分方程

$$\dot{y}(t) = f(y, t) \quad (2-2)$$

的数值算法的设计分析，如数值算法的相容性、稳定性和收敛性等。事实上，结构动力学运动方程 (2-1) 本质上就是一个二阶的线性微分方程，通过引入速度为独立的变量，可以将其二阶微分形式降维为具有形如 (2-2) 形式的更高阶一阶微分方程。因此，在这一节主要针对 (2-2) 进行数值算法性能指标分析，如果一些定义对于二阶微分方程有差异，再具体指出。

由于广义线性法是求解微分方程 (2-2) 更具一般性的方法，因此本节先通过最简单的数值算法 (Euler 法) 分析引出广义线性法，而后后续的算法性能分析都尽可能在广义线性的基础上进行定义及分析。而常见的线性多步法和 Runge-Kutta 法将是由其特例导出。

2.1 广义线性法

在 20 世纪初 Euler 在他的著作中，针对如下自治的一阶微分方程

$$\dot{y}(t) = f(y(x)) \quad (2-3)$$

带有合适的初始条件 $y(x_0) = y_0 \in \mathbb{X}$ 的数值求解，给出了如下的数值格式

$$y_n = y_{n-1} + h_n f(y_{n-1}), \quad n = 1, 2, \dots \quad (2-4)$$

对于每一个给定的 n 值，其 y_n 都可以根据上式由 y_{n-1} 计算得到。同时， h_n 为在第 n 的积分步长，一般情况下没必要要求在整个积分过程中相等，亦即变时间步长积分。但在本文整个分析过程中，我们假定其数值算法都是在相等的时间步长内进行积分及计算的。亦即

$$h_n \equiv h \quad n = 1, 2, \dots \quad (2-5)$$

显然，该数值格式的优劣很大程度依赖于积分步长 h 以及等式 (2-3) 右端项 $f(y)$ 随变量 y 的变化快慢程度。而后者通常由 $f(y)$ 满足的 Lipschitz 常数 L 来度量。同时，我们也需要提及该条件也保证了微分方程 (2-3) 解的存在唯一性。于是，我们在本全文中都假定分析的微分方程都满足该 Lipschitz 条件。关于该条件的对其微分方程解的存在唯一性的证明，读者可以翻阅任意一本常微分方程的书籍都可以查到。这里不再做过多累述。

事实上，Euler 格式 (2-4) 简单易用，但它却是条件稳定的且只有一阶精度。在很多时候，并不是很实用。因此，后续许多学者提出了很多改进 Euler 格式的算法。其中，至少有两个策略进行改进 Euler 方法

- ‡ 单步内使用更多、更复杂的计算，如 Runge-Kutta 法。
- ‡ 使用更多已知节点的逼近值。如线性多步法。
- ‡ 更高阶导数值的使用。如 Rosenbrock 方法。
- ‡ 多步-多级-多导数方法。如拟 Runge-Kutta 方法。

从简单的 Euler 格式出发，我们可以建立如图2-1的改进策略坐标系

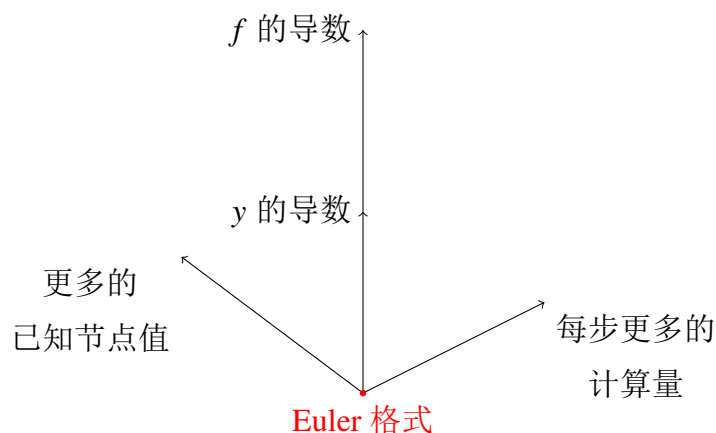


图 2-1 Euler 格式的改进策略示意图
Fig. 2-1 Generalizations of the Euler method

在图2-1的坐标系下，现阶段的大多数的数值算法都可以被建立在这个三维算法格式空间中，如图2-2所示。于是，可以知道广义线性法是线性多步法和 Runge-Kutta 法的结合，更是 Euler 格式的一般化推广。同时，从该图可以知道，广义线性法并不能包括所有的数值格式，它仅仅是现阶段数值算法的一大类。

广义线性方法即结合了线性多步法的多值特点，又使用了 Runge-Kutta 方法的多级属性。并且它首次由 Butcher 教授在 1966 年提出^[2]。下面提及的广义线性法

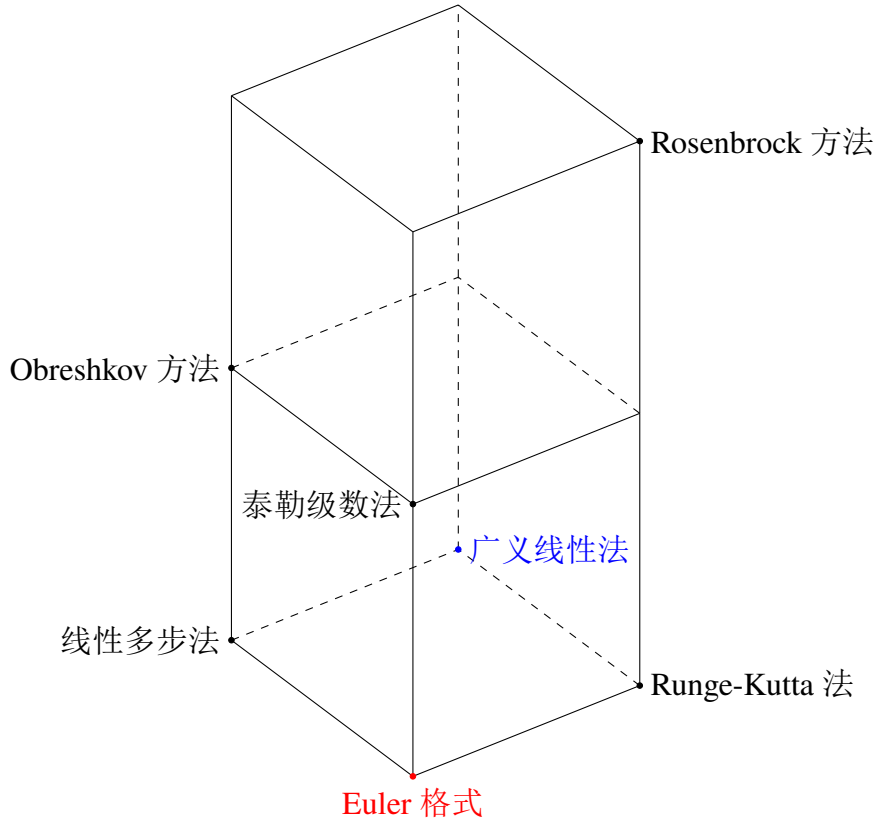


图 2-2 Euler 格式的改进算法框架

Fig.2-2 Generalized formworks of the Euler method

的矩阵表示形式则是由 Burrage 和 Butcher 在 1980 年引入而来^[15]。

假定在单步内进行转换的量有 r 个。在第 n 步的开始，这 r 个量被表示为 $y_1^{[n-1]}, y_2^{[n-1]}, \dots, y_r^{[n-1]}$ 。当第 n 步完成计算时，对应的 r 量分别为 $y_1^{[n]}, y_2^{[n]}, \dots, y_r^{[n]}$ ，并且这些量将作为下一个时间步内的起始值进行后续计算。同时，在单步内计算的 s 个级数值 Y_1, Y_2, \dots, Y_s 所对应的级数导数值为 F_1, F_2, \dots, F_s 。为了表示方便，引入如下的 r 或 s 维向量表示，即

$$y^{[n-1]} = \begin{bmatrix} y_1^{[n-1]} \\ y_2^{[n-1]} \\ \vdots \\ y_r^{[n-1]} \end{bmatrix}, \quad y^{[n]} = \begin{bmatrix} y_1^{[n]} \\ y_2^{[n]} \\ \vdots \\ y_r^{[n]} \end{bmatrix}, \quad Y^{[n]} = \begin{bmatrix} Y_1^{[n]} \\ Y_2^{[n]} \\ \vdots \\ Y_s^{[n]} \end{bmatrix}, \quad F(Y^{[n]}) = \begin{bmatrix} f(Y_1^{[n]}) \\ f(Y_2^{[n]}) \\ \vdots \\ f(Y_s^{[n]}) \end{bmatrix} \quad (2-6)$$

类似于 Runge-Kutta 方法，级数值 Y_i 是依赖于级数导数值 F_i 的线性组合来计算，但是现在的广义线性法将其推广为不仅依赖于级数导数值的线性组合，也依赖于前述已知节点逼近值 y_i 的线性组合。即

$$Y_i = \sum_{j=1}^s h a_{ij} f(Y_j^{[n]}) + \sum_{j=1}^r u_{ij} y_j^{[n-1]}, \quad i = 1, 2, \dots, s \quad (2-7)$$

同理，对于输出量 $y_i^{[n]}$ 也是不仅线性依赖于各个级数导数值 $F_i = f(Y_i)$ 而且也是线性依赖于前述已知节点逼近值 y_i ，即

$$y_i^{[n]} = \sum_{j=1}^s h b_{ij} f(Y_j^{[n]}) + \sum_{j=1}^r v_{ij} y_j^{[n-1]}, \quad i = 1, 2, \dots, r \quad (2-8)$$

令 $A = [a_{ij}]_{s \times s}$, $U = [u_{ij}]_{s \times r}$, $B = [b_{ij}]_{r \times s}$ 以及 $V = [v_{ij}]_{r \times r}$ ，同时使用 Kronecker 积符号 (\otimes)，广义线性法则可以表示为

$$Y^{[n]} = h(A \otimes I)F(Y^{[n]}) + (U \otimes I)y^{[n-1]} \quad (2-9)$$

$$y^{[n]} = h(B \otimes I)F(Y^{[n]}) + (V \otimes I)y^{[n-1]} \quad (2-10)$$

其中， I 表示维度相容的单位矩阵。而 Kronecker 积定义如下，若 $A \in \mathbb{R}^{m_1 \times n_1}$ 和 $B \in \mathbb{R}^{m_2 \times n_2}$ ，则有

$$A \otimes B = \begin{bmatrix} a_{11}B & a_{12}B & \cdots & a_{1,n_1}B \\ a_{21}B & a_{22}B & \cdots & a_{2,n_1}B \\ \vdots & \vdots & & \vdots \\ a_{m_1,1}B & a_{m_1,2}B & \cdots & a_{m_1,n_1}B \end{bmatrix} \in \mathbb{R}^{m_1 m_2 \times n_1 n_2} \quad (2-11)$$

进一步，广义线性法可以被表达为如下形式

$$\begin{bmatrix} Y^{[n]} \\ y^{[n]} \end{bmatrix} = \begin{bmatrix} A \otimes I & U \otimes I \\ B \otimes I & V \otimes I \end{bmatrix} \begin{bmatrix} hF(Y^{[n]}) \\ y^{[n-1]} \end{bmatrix} \quad (2-12)$$

通常情况下，广义线性法的数值性能就被这四个矩阵所决定，即 A, U, B 和 V 。因此对于一个广义线性法，可以用如下的分块矩阵刻画

$$\begin{bmatrix} A & U \\ B & V \end{bmatrix} \quad (2-13)$$

当四个矩阵取值不同时，对应着不同的广义线性法。特别地，线性多步法和 Runge-Kutta 都是其特例。

2.1.1 线性多步法

考虑对于一阶微分方程 (2-2) 的 k 步线性多步法

$$y_n = \sum_{j=1}^k \alpha_j y_{n-j} + h \sum_{j=0}^k \beta_j f(y_{n-j}) \quad (2-14)$$

令 $Y^{[n]} = y_n$, 并且

$$y^{[n]} = [y_n, y_{n-1}, \dots, y_{n-k+1}, hf(y_n), hf(y_{n-1}), \dots, hf(y_{n-k+1})]^T \quad (2-15a)$$

$$y^{[n-1]} = [y_{n-1}, y_{n-2}, \dots, y_{n-k}, hf(y_{n-1}), hf(y_{n-2}), \dots, hf(y_{n-k})]^T \quad (2-15b)$$

于是, 对于 k 步的线性多步法 (2-14) 可以写成广义线性法的形式^[15], 即 $r = 2k, s = 1$

$$\left[\begin{array}{c|cccccccc} \beta_0 & \alpha_1 & \cdots & \alpha_{k-1} & \alpha_k & \beta_1 & \cdots & \beta_{k-1} & \beta_k \\ \hline \beta_0 & \alpha_1 & \cdots & \alpha_{k-1} & \alpha_k & \beta_1 & \cdots & \beta_{k-1} & \beta_k \\ 0 & 1 & \cdots & 0 & 0 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 & 0 & \cdots & 0 & 0 \\ 1 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 0 & 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 1 & 0 \end{array} \right] = \quad (2-16)$$

特别地, k 步的线性多步法 (2-14) 也可以写成 $r = k, s = 1$ 的广义线性法形式^[16]。文 [16] 中定义

$$y_i^{[n-1]} = \sum_{j=k-i+1}^k (\alpha_j y_{n+k-i-j} + h\beta_j f(y_{n+k-i-j})), \quad i = 1, 2, \dots, k \quad (2-17)$$

其实, 公式 (2-17) 在文献 [17] 就已经被提出来了, 只不过没有涉及广义线性法的应用。于是, 线性多步法 (2-14) 则可以写成如下形式

$$y_n = h\beta_0 f(y_n) + \sum_{j=1}^k (\alpha_j y_{n-j} + h\beta_j f(y_{n-j})) = h\beta_0 f(y_n) + y_k^{[n-1]} \quad (2-18)$$

进一步, 在 n 步结束时

$$\begin{aligned} y_i^{[n]} &= \sum_{j=k-i+1}^k (\alpha_j y_{n+1+k-i-j} + h\beta_j f(y_{n+1+k-i-j})) \\ &= \alpha_{k-i+1} y_n + h\beta_{k-i+1} f(y_n) + \sum_{j=k-i+2}^k (\alpha_j y_{n+1+k-i-j} + h\beta_j f(y_{n+1+k-i-j})) \\ &= (\alpha_{k-i+1} \beta_0 + \beta_{k-i+1}) h f(y_n) + \alpha_{k-i+1} y_k^{[n-1]} + y_{i-1}^{[n-1]} \end{aligned} \quad (2-19)$$

其中, $i = 1, 2, \dots, k$ 。线性多步法 (2-14) 的另外一种广义线性法的表出形式为

$$\left[\begin{array}{c|c} A & U \\ \hline B & V \end{array} \right] = \left[\begin{array}{c|cccccc} \beta_0 & 0 & 0 & 0 & \cdots & 0 & 1 \\ \hline \alpha_k \beta_0 + \beta_k & 0 & 0 & 0 & \cdots & 0 & \alpha_k \\ \alpha_{k-1} \beta_0 + \beta_{k-1} & 1 & 0 & 0 & \cdots & 0 & \alpha_{k-1} \\ \alpha_{k-2} \beta_0 + \beta_{k-2} & 0 & 1 & 0 & \cdots & 0 & \alpha_{k-2} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \alpha_2 \beta_0 + \beta_2 & 0 & 0 & 0 & \cdots & 0 & \alpha_2 \\ \alpha_1 \beta_0 + \beta_1 & 0 & 0 & 0 & \cdots & 1 & \alpha_1 \end{array} \right] \quad (2-20)$$

其中, $Y^{[n]} = y_n$, 而 $y^{[n]}$ 则为

$$y^{[n]} = [y_1^{[n]}, y_2^{[n]}, \dots, y_{k-1}^{[n]}, y_k^{[n]}]^T \quad (2-21)$$

2.1.1.1 Adams 算法

线性多步法中, 求解非刚性问题时最常用的方法就是 Adams 方法族。对于线性多步法公式 (2-14) 中, 令

$$\alpha_1 = 1, \quad \alpha_i = 0, i > 1 \quad (2-22)$$

于是, Adams 算法的一般形式为

$$y_n = y_{n-1} + h \sum_{j=0}^k \beta_j f(y_{n-j}) \quad (2-23)$$

需要说明的是, 算法 (2-23) 的显式形式通常称为 Adams-Bashforth 算法; 而其隐式算法则称为 Adams-Moulton 算法。

于是将 Adams 算法改写成广义线性法的形式, 则有

$$\left[\begin{array}{c} Y_1 \\ \hline y_n \\ hf(y_n) \\ hf(y_{n-1}) \\ hf(y_{n-2}) \\ \vdots \\ hf(y_{n-k+1}) \end{array} \right] = \left[\begin{array}{c|cccccc} \beta_0 & 1 & \beta_1 & \beta_2 & \cdots & \beta_{k-1} & \beta_k \\ \hline \beta_0 & 1 & \beta_1 & \beta_2 & \cdots & \beta_{k-1} & \beta_k \\ 1 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 1 & 0 \end{array} \right] \left[\begin{array}{c} hf(Y_1) \\ \hline y_{n-1} \\ hf(y_{n-1}) \\ hf(y_{n-2}) \\ hf(y_{n-3}) \\ \vdots \\ hf(y_{n-k}) \end{array} \right] \quad (2-24)$$

向前显式 Euler 法对应的参数为

$$k = 1, \quad \beta_0 = 0, \quad \beta_1 = 1 \quad (2-25)$$

于是，从等式 (2-24) 可知其在广义线性法框架下的表示形式为

$$\left[\begin{array}{c|cc} 0 & 1 & 1 \\ \hline 0 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{array} \right] \quad (2-26)$$

事实上，向前显式 Euler 法也可以写为如下更加简洁的形式

$$\left[\begin{array}{c} Y_1 \\ -\frac{Y_1}{y_n} \end{array} \right] = \left[\begin{array}{c|c} 0 & 1 \\ \hline 1 & 1 \end{array} \right] \left[\begin{array}{c} hf(Y_1) \\ -\frac{hf(Y_1)}{y_n} \end{array} \right] \quad (2-27)$$

其中， $Y_1 = y_{n-1}$ 。

尽管 Adams-Moulton 算法是隐式的，由于其较小的稳定域，它们也仅仅只在求解非刚性问题时使用。除此之外，它们也常常在作为预测-校正格式使用。

2.1.1.2 预测-校正格式

预测-校正格式中，使用 Adams-Bashforth 方法作为一个预测逼近算法，而 Adams-Moulton 作为校正格式算法。同时，预测-校正格式通常简写为“PEC”或者“PECE”，其中“P”代表预测格式，而“E”表示计算，“C”表示校正格式。其算法一般形式可写为

$$y_n^* = y_{n-1} + h \sum_{j=1}^k \beta_j^* f(y_{n-j}) \quad (2-28a)$$

$$y_n = y_{n-1} + h\beta_0 f(y_n^*) + h \sum_{j=1}^k \beta_j f(y_{n-j}) \quad (2-28b)$$

于是，一个 PEC 格式可以由下列的广义线性法表示

$$\begin{bmatrix} Y_1 \\ \hline y_n \\ hf(y_n) \\ hf(y_{n-1}) \\ hf(y_{n-2}) \\ \vdots \\ hf(y_{n-k+1}) \end{bmatrix} = \begin{bmatrix} 0 & 1 & \beta_1^* & \beta_2^* & \cdots & \beta_{k-1}^* & \beta_k^* \\ \beta_0 & 1 & \beta_1 & \beta_2 & \cdots & \beta_{k-1} & \beta_k \\ 1 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 1 & 0 \end{bmatrix} \begin{bmatrix} hf(Y_1) \\ \hline y_{n-1} \\ hf(y_{n-1}) \\ hf(y_{n-2}) \\ hf(y_{n-3}) \\ \vdots \\ hf(y_{n-k}) \end{bmatrix} \quad (2-29)$$

其中, $Y_1 = y_n^*$ 。而 PECE 格式则可以写为

$$y_n^* = y_{n-1} + h \sum_{j=1}^k \beta_j^* f(y_{n-j}) \quad (2-30a)$$

$$y_n = y_{n-1} + h\beta_0 f(y_n^*) + h \sum_{j=1}^k \beta_j f(y_{n-j}) \quad (2-30b)$$

于是, 一个 PEC 格式可以由下列的广义线性法表示

$$\begin{bmatrix} Y_1 \\ Y_2 \\ \hline y_n \\ hf(y_n) \\ hf(y_{n-1}) \\ hf(y_{n-2}) \\ \vdots \\ hf(y_{n-k+1}) \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & \beta_1^* & \beta_2^* & \cdots & \beta_{k-1}^* & \beta_k^* \\ \beta_0 & 0 & 1 & \beta_1 & \beta_2 & \cdots & \beta_{k-1} & \beta_k \\ \beta_0 & 0 & 1 & \beta_1 & \beta_2 & \cdots & \beta_{k-1} & \beta_k \\ 0 & 1 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & \cdots & 1 & 0 \end{bmatrix} \begin{bmatrix} hf(Y_1) \\ hf(Y_2) \\ \hline y_{n-1} \\ hf(y_{n-1}) \\ hf(y_{n-2}) \\ hf(y_{n-3}) \\ \vdots \\ hf(y_{n-k}) \end{bmatrix} \quad (2-31)$$

2.1.1.3 向后微分公式

向后微分公式 (BDF) 是第一个被提出来求解刚性问题的数值算法。为了克服 Adams 算法在求解刚性问题的较小稳定域问题, Curtiss 和 Hirschfelder 在 1952 年提出了向后微分公式^[18]。后来经过 Gear 的推广^[19], 向后微分公式得到广泛的应用。

向后微分公式可以由格式 (2-14) 中, 令

$$\beta_j = 0, \quad j > 0 \quad (2-32)$$

得到。亦即逼近解仅仅依赖于当前步的导数值 $f(y_n)$ 。其算法格式为

$$y_n = \sum_{j=1}^k \alpha_j y_{n-j} + h\beta_0 f(y_n) \quad (2-33)$$

特别地，向后微分公式的精度可以达到 7 阶。7 阶更高的向后微分公式是不稳定的^[19]。其中，仅仅只有 $k = 1$ 和 $k = 2$ 是具有 A 稳定的。对于其他 k 值事实上不适合求解刚性问题。对于 $k = 1 \rightarrow 6$ ，其具体的算法格式如下

$$k = 1 : y_n = y_{n-1} + hf(y_n) \quad (2-34)$$

$$k = 2 : y_n = \frac{4}{3}y_{n-1} - \frac{1}{3}y_{n-2} + \frac{2}{3}hf(y_n) \quad (2-35)$$

$$k = 3 : y_n = \frac{18}{11}y_{n-1} - \frac{9}{11}y_{n-2} + \frac{2}{11}y_{n-3} + \frac{6}{11}hf(y_n) \quad (2-36)$$

$$k = 4 : y_n = \frac{48}{25}y_{n-1} - \frac{36}{25}y_{n-2} + \frac{16}{25}y_{n-3} - \frac{3}{25}y_{n-4} + \frac{12}{25}hf(y_n) \quad (2-37)$$

$$k = 5 : y_n = \frac{300}{137}y_{n-1} - \frac{300}{137}y_{n-2} + \frac{200}{137}y_{n-3} - \frac{75}{137}y_{n-4} + \frac{12}{137}y_{n-5} + \frac{60}{137}hf(y_n) \quad (2-38)$$

$$k = 6 : y_n = \frac{120}{49}y_{n-1} - \frac{150}{49}y_{n-2} + \frac{400}{147}y_{n-3} - \frac{75}{49}y_{n-4} + \frac{24}{49}y_{n-5} - \frac{10}{147}y_{n-6} + \frac{20}{49}hf(y_n) \quad (2-39)$$

在广义线性法的表示形式下有

$$\begin{bmatrix} Y_1 \\ y_n \\ hf(y_n) \\ hf(y_{n-1}) \\ hf(y_{n-2}) \\ \vdots \\ hf(y_{n-k+1}) \end{bmatrix} = \begin{bmatrix} \beta_0 & \alpha_1 & \alpha_2 & \alpha_3 & \cdots & \alpha_{k-1} & \alpha_k \\ \beta_0 & \alpha_1 & \alpha_2 & \alpha_3 & \cdots & \alpha_{k-1} & \alpha_k \\ 0 & 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 1 & 0 \end{bmatrix} \begin{bmatrix} hf(Y_1) \\ y_{n-1} \\ hf(y_{n-1}) \\ hf(y_{n-2}) \\ hf(y_{n-3}) \\ \vdots \\ hf(y_{n-k}) \end{bmatrix} \quad (2-40)$$

其中， $Y_1 = y_n$ 。同时， $k = 1$ 对应向后微分的隐式 Euler 算法。其广义线性法的表示形式为

$$\begin{bmatrix} 1 & | & 1 \\ - & + & - \\ 1 & | & 1 \end{bmatrix} \quad (2-41)$$

2.1.2 Runge-Kutta 法

对于求解微分方程 (2-3) 的一般形式的 Runge-Kutta 方法可描述如下

$$Y_i^{[n]} = y_{n-1} + h \sum_{j=1}^s a_{ij} f(Y_j^{[n]}), \quad i = 1, 2, \dots, s \quad (2-42a)$$

$$y_n = y_{n-1} + h \sum_{j=1}^s b_j f(Y_j^{[n]}) \quad (2-42b)$$

一般情况下，Runge-Kutta 法可通过 Butcher 表格表示，即

$$\begin{array}{c|c} c & A \\ \hline & b^T \end{array} = \begin{array}{c|ccc} c_1 & a_{11} & \cdots & a_{1s} \\ \vdots & \vdots & \ddots & \vdots \\ c_s & a_{s1} & \cdots & a_{ss} \\ \hline & b_1 & \cdots & b_s \end{array} \quad (2-43)$$

其中，向量 c 表示在单步内极值的位置；而矩阵 A 表示极值对其他极值导数的依赖性；向量 b 表明了积分权重值，暗示了最后的输出量如何依赖于先前计算的极导数值。当然，也可以通过广义线性法的形式表出，即

$$\left[\begin{array}{c|c} A & e \\ \hline b^T & 1 \end{array} \right] = \left[\begin{array}{cccc} a_{11} & \cdots & a_{1s} & 1 \\ \vdots & \ddots & \vdots & \vdots \\ a_{s1} & \cdots & a_{ss} & 1 \\ \hline b_1 & \cdots & b_s & 1 \end{array} \right] \quad (2-44)$$

有意思的是，前述提及的 Euler 法也可以看作一阶的 Runge-Kutta 法，其 Butcher 表格为

$$\begin{array}{c|c} 0 & \\ \hline & 1 \end{array} \quad (2-45)$$

而一般性的二阶精度带任意参数 θ 的 Runge-Kutta 方法可用 Butcher 表格描述为

$$\begin{array}{c|c} 0 & \\ \theta & \theta \\ \hline & 1 - \frac{1}{2\theta} \quad \frac{1}{2\theta} \end{array} \quad (2-46)$$

当参数 θ 取值分别为 $\frac{1}{2}$ 和 1，分别对应中点公式和 Trapezoidal 规则。在 Runge-Kutta 的框架下，这二者分别有 RK22 和 RK21 表示。其对应的 Butcher 表格为

$$\text{RK21: } \begin{array}{c|c} 0 & \\ 1 & 1 \\ \hline & \frac{1}{2} \quad \frac{1}{2} \end{array} \quad (2-47)$$

$$\text{RK22:} \quad \begin{array}{c|c} 0 & \\ \hline \frac{1}{2} & \frac{1}{2} \\ \hline & 0 \quad 1 \end{array} \quad (2-48)$$

而我们最常使用的四阶 Runge-Kutta 方法，也就是 MATLAB 软件中 ode 求解器中的“ode45”，可表示为

$$\text{RK41:} \quad \begin{array}{c|ccc} 0 & & & \\ \hline \frac{1}{2} & \frac{1}{2} & & \\ \hline \frac{1}{2} & 0 & \frac{1}{2} & \\ \hline 1 & 0 & 0 & 1 \\ \hline & \frac{1}{6} & \frac{1}{3} & \frac{1}{3} & \frac{1}{6} \end{array} \quad (2-49)$$

由等式 (2-44) 不难将这些 Runge-Kutta 方法使用广义线性法的形式表出，即

$$\text{RK21:} \quad \left[\begin{array}{cc|c} 0 & 0 & 1 \\ 1 & 0 & 1 \\ \hline \frac{1}{2} & \frac{1}{2} & 1 \end{array} \right] \quad (2-50)$$

$$\text{RK41:} \quad \left[\begin{array}{cccc|c} 0 & 0 & 0 & 0 & 1 \\ \frac{1}{2} & 0 & 0 & 0 & 1 \\ 0 & \frac{1}{2} & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 \\ \hline \frac{1}{6} & \frac{1}{3} & \frac{1}{3} & \frac{1}{6} & 1 \end{array} \right] \quad (2-51)$$

关于 Runge-Kutta 方法更加详细的分析和应用可以参考文献或专著 [20–26]。

2.1.3 One-leg 方法

为了简化线性多步法 (2-14) 在求解刚性问题时的误差分析，Dahlquist 引入了如下格式的 One-leg 方法^[27,28]。

$$y_n = \sum_{j=1}^k \alpha_j y_{n-j} + h\beta f \left(\frac{1}{\beta} \sum_{j=0}^k \beta_j y_{n-j} \right) \quad (2-52)$$

其中, 参数 β 满足 $\beta = \sum_{j=0}^k \beta_j$ 。因此, 从公式 (2-52) 可以知道, 每步内只需要计算一次 f 的函数值。同时, 也需要提及的时, 对于对应的线性多步法 (2-14), One-leg 方法可能具有更强的非线性稳定性特点, 如 G 稳定性以及对于非一致性网格上更好的鲁棒性^[29]。

令

$$Y^{[n]} = \frac{1}{\beta} \sum_{j=1}^k \beta_j y_{n-j} \quad (2-53)$$

将等式 (2-52) 带入到上式就有

$$\begin{aligned} Y^{[n]} &= \frac{1}{\beta} \left(\beta_0 y_n + \sum_{j=1}^k \beta_j y_{n-j} \right) \\ &= \frac{1}{\beta} \left[\beta_0 \left(\sum_{j=1}^k \alpha_j y_{n-j} + h\beta f(Y^{[n]}) \right) + \sum_{j=1}^k \beta_j y_{n-j} \right] \\ &= \frac{1}{\beta} \sum_{j=1}^k (\beta_0 \alpha_j + \beta_j) y_{n-j} + h\beta_0 f(Y^{[n]}) \end{aligned} \quad (2-54)$$

当建立如下的递推向量

$$y^{[n]} = [y_n \quad y_{n-1} \quad \cdots \quad y_{n-k+1}]^T \quad (2-55)$$

时, One-leg 方法 (2-52) 就可以用广义线性法表出如下 $r = k, s = 1$

$$\left[\begin{array}{c|c} A & U \\ \hline B & V \end{array} \right] = \left[\begin{array}{c|cccccc} \beta_0 & \frac{\beta_0 \alpha_1 + \beta_1}{\beta} & \frac{\beta_0 \alpha_2 + \beta_2}{\beta} & \cdots & \frac{\beta_0 \alpha_{k-1} + \beta_{k-1}}{\beta} & \frac{\beta_0 \alpha_k + \beta_k}{\beta} \\ \hline \beta & \alpha_1 & \alpha_2 & \cdots & \alpha_{k-1} & \alpha_k \\ 0 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & 0 \end{array} \right] \quad (2-56)$$

关于 One-leg 方法的另外一种书写方式可以参见文献 [16]。

2.1.4 扩展的向后微分格式

Cash 提出了扩展到向后微分公式 (Extended backward differentiation formulas:EBDFs)^[30] 求解刚性微分方法。该格式涉及到 t_{n+k+1} 时刻 f 的计算, 其算法可描述为

$$\sum_{j=0}^k \alpha_j y_{n+j} = h\beta_k f(y_{n+k}) + h\beta_{k+1} f(y_{n+k+1}) \quad (2-57)$$

其中, 参数 $\alpha_j, j = 0, 1, \dots, k, \beta_k, \beta_{k+1}$ 可通过算法格式的 $k+1$ 阶精度条件确定, 而不失一般性的可以假定

$$\alpha_k = 1 \quad (2-58)$$

对于 $k = 1, 2, 3$, 上述 EBDFs 算法格式可实现 A 稳定和 L 稳定; 而对于 $k = 4, 5, 6, 7, 8$, 仅仅是 $A(\alpha)$ 稳定的。其稳定域大小可见文 [23,30]。

假设变量逼近值 $y_n, y_{n+1}, \dots, y_{n+k-1}$ 已知, 则基于 EBDF 格式的计算步骤可描述为

i. 基于经典的向后微分格式计算 \bar{y}_{n+k}

$$\bar{y}_{n+k} + \sum_{j=0}^{k-1} \hat{\alpha}_j y_{n+j} = h\hat{\beta}_k f(\bar{y}_{n+k}) \quad (2-59)$$

ii. 继续用上式中的向后微分公式计算 \bar{y}_{n+k+1} 的值, 即

$$\bar{y}_{n+k+1} + \hat{\alpha}_{k-1} \bar{y}_{n+k} + \sum_{j=0}^{k-2} \hat{\alpha}_j y_{n+j+1} = h\hat{\beta}_k f(\bar{y}_{n+k+1}) \quad (2-60)$$

iii. 丢弃 \bar{y}_{n+k} , 在 EBDF 格式 (2-57) 中引入 \bar{y}_{n+k+1} 求解 y_{n+k} 可得

$$y_{n+k} + \sum_{j=0}^{k-1} \alpha_j y_{n+j} = h\beta_k f(y_{n+k}) + h\beta_{k+1} f(\bar{y}_{n+k+1}) \quad (2-61)$$

正如文 [31] 所指出的, 上述 EBDF 格式的缺点是在步骤 (i,ii) 中需要求解带有相同雅可比矩阵 $(I - h\hat{\beta}_k J, J = \partial f / \partial y)$ 的非线性方程组, 但在第 iii 步骤中, 却又有一个不同的雅可比矩阵 $(I - h\beta_k J)$ 进行非线性求解, 这就导致了额外的计算量。为了弥补这个缺点, Cash 又提出了修改版的 EBDF 格式 (MEBDF)^[31], 即

$$\sum_{j=0}^k \alpha_j y_{n+j} = h\hat{\beta}_k f(y_{n+k}) + h(\beta_k - \hat{\beta}_k) f(\bar{y}_{n+k}) + h\beta_{k+1} f(\bar{y}_{n+k+1}) \quad (2-62)$$

数值格式 (MEBDF) 对于 $k = 1, 2, 3$ 也是 A 稳定和 L 稳定的; 而对于 $k = 4, 5, 6, 7, 8$ 都是带有较对应的 EBDF 格式更大 α 角度的 $A(\alpha)$ 稳定算法。对于 BDF、EBDF 和 MEBDF 算法所对应的 α 角度列于表2-1。注意, 表格中带有 “*” 表示不是 $A(\alpha)$ 稳定的。

将步骤 i 中的 \bar{y}_{n+k} 带入到步骤 ii 中的方程中, 得

表 2-1 BDF、EBDF 和 MEBDF 算法所对应的 $A(\alpha)$ 稳定的 α 角度值
 Table2-1 Angles α of $A(\alpha)$ -stability for BDF, EBDF and MEBDF schemes

k	1	2	3	4	5	6	7	8
BDF	90°	90°	88°	73°	51°	18°	*	*
EBDF	90°	90°	90°	87.61°	80.21°	67.73°	48.82°	19.98°
MEBDF	90°	90°	90°	88.36°	83.07°	74.48°	61.98°	42.87°

$$\begin{aligned} \bar{y}_{n+k+1} = & \hat{\alpha}_{k-1} \hat{\alpha}_0 y_n + \sum_{j=1}^{k-1} (\hat{\alpha}_{k-1} \hat{\alpha}_j - \hat{\alpha}_{j-1}) y_{n+j} \\ & - h \hat{\alpha}_{k-1} \hat{\beta}_k f(\bar{y}_{n+k}) + h \hat{\beta}_k f(\bar{y}_{n+k+1}) \end{aligned} \quad (2-63)$$

于是，基于 MEBDF 格式，其广义线性法的表示形式中的四个矩阵可写为

$$Y^{[n]} = \begin{bmatrix} \bar{y}_{n+k} \\ \bar{y}_{n+k+1} \\ y_{n+k} \end{bmatrix}, \quad F(Y^{[n]}) = \begin{bmatrix} f(\bar{y}_{n+k}) \\ f(\bar{y}_{n+k+1}) \\ f(y_{n+k}) \end{bmatrix}, \quad y^{[n]} = \begin{bmatrix} y_{n+k} \\ y_{n+k-1} \\ \vdots \\ y_{n+1} \end{bmatrix} \quad (2-64)$$

其系数矩阵 A, B, U 和 V 分别为

$$A = \begin{bmatrix} \hat{\beta}_k & 0 & 0 \\ -\hat{\alpha}_{k-1} \hat{\beta}_k & \hat{\beta}_k & 0 \\ \beta_k - \hat{\beta}_k & \beta_{k+1} & \hat{\beta}_k \end{bmatrix} \quad (2-65a)$$

$$U = \begin{bmatrix} -\hat{\alpha}_{k-1} & -\hat{\alpha}_{k-2} & \cdots & -\hat{\alpha}_1 & -\hat{\alpha}_0 \\ \hat{\alpha}_{k-1} \hat{\alpha}_{k-1} - \hat{\alpha}_{k-2} & \hat{\alpha}_{k-1} \hat{\alpha}_{k-2} - \hat{\alpha}_{k-3} & \cdots & \hat{\alpha}_{k-1} \hat{\alpha}_1 - \hat{\alpha}_0 & \hat{\alpha}_{k-1} \hat{\alpha}_0 \\ -\alpha_{k-1} & -\alpha_{k-2} & \cdots & -\alpha_1 & -\alpha_0 \end{bmatrix} \quad (2-65b)$$

$$B = \begin{bmatrix} \beta_k - \hat{\beta}_k & \beta_{k+1} & \hat{\beta}_k \\ 0 & 0 & 0 \\ \vdots & \vdots & \vdots \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad V = \begin{bmatrix} -\alpha_{k-1} & -\alpha_{k-2} & \cdots & -\alpha_1 & -\alpha_0 \\ 1 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 1 & 0 \end{bmatrix} \quad (2-65c)$$

2.1.5 两步 Runge-Kutta 方法

考虑文 [32] 中提出的两步 Runge-Kutta 方法 (Two-step Runge-Kutta: TSRK)。该类算法依赖于两个连续步的极数值，其算法可以描述为

$$Y_i^{[n]} = (1 - \mu_i)y_{n-1} + \mu_i y_{n-2} + h \sum_{j=1}^s \left(a_{ij} f(Y_j^{[n]}) + b_{ij} f(Y_j^{[n-1]}) \right) \quad (2-66a)$$

$$y_n = (1 - \vartheta)y_{n-1} + \vartheta y_{n-2} + h \sum_{j=1}^s \left(v_j f(Y_j^{[n]}) + w_j f(Y_j^{[n-1]}) \right) \quad (2-66b)$$

其中, $i = 1, 2, \dots, s$, 且 $Y_i^{[n]}$ 是 $y(t_{n-1} + c_i h)$ 的逼近值。TSRK 也可以由下列的 Butcher 表格给出

$$\begin{array}{c|c|c} u & A & B \\ \hline \vartheta & v^T & w^T \end{array} \quad (2-67)$$

在广义线性法的表示形式下, 则有 $r = s + 2$

$$\begin{bmatrix} Y^{[n]} \\ y_n \\ y_{n-1} \\ hF(Y^{[n]}) \end{bmatrix} = \begin{bmatrix} A & e - u & u & B \\ v^T & 1 - \vartheta & \vartheta & w^T \\ 0 & 1 & 0 & 0 \\ I & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} hF(Y^{[n]}) \\ y_{n-1} \\ y_{n-2} \\ hF(Y^{[n-1]}) \end{bmatrix} \quad (2-68)$$

其中, I 是维度为 s 的单位矩阵, 而 0 是带有合适维度的全零矩阵, $A = [a_{ij}]_{s \times s}$, $B = [b_{ij}]_{s \times s}$, $v = [v_j]_{s \times 1}$, $w = [w_j]_{s \times 1}$, 而 e 为 s 维元素全为 1 的列向量。

2.1.6 多步 Runge-Kutta 方法

Burrage 和 Sharp 等人研究如下形式的多步 Runge-Kutta 方法 (Multistep Runge-Kutta: MRK) [33-35]。

$$Y_i^{[n]} = h \sum_{j=1}^s a_{ij} f(Y_j^{[n]}) + \sum_{j=1}^k u_{ij} y_{n+1-j} \quad (2-69a)$$

$$y_{n+1} = h \sum_{j=1}^s b_j f(Y_j^{[n]}) + \sum_{j=1}^k v_j y_{n+1-j} \quad (2-69b)$$

其中, $i = 1, 2, \dots, s$ 。而 $Y_i^{[n]}$ 是 $y(t_n + c_i h)$ 的逼近值。对于 $k = 1$, 上述的 MRK 方法就退化为普通的 Runge-Kutta 算法; 而对于 $k = 2$, MRK 方法并不能退化为前述提及的 TSRK 方法 (2-66)。这是因为 MRK 方法仅仅依赖于当前步内的级数值 $Y_i^{[n]}$, 而 TSRK 方法却依赖于连续两个时间步长内的级数值 $Y_i^{[n]}, Y_i^{[n-1]}$ 。

通过令

$$y^{[n]} = [y_{n+1} \quad y_n \quad \cdots \quad y_{n-k+2}]^T \quad (2-70)$$

MRK 方法 (2-69) 可以写成广义线性法的形式如下

$$\left[\begin{array}{c|c} A & U \\ \hline B & V \end{array} \right] = \left[\begin{array}{cccc|cccc} a_{11} & a_{12} & \cdots & a_{1s} & u_{11} & u_{12} & \cdots & u_{1,k-1} & u_{1k} \\ a_{21} & a_{22} & \cdots & a_{2s} & u_{21} & u_{22} & \cdots & u_{2,k-1} & u_{2k} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{s1} & a_{s2} & \cdots & a_{ss} & u_{s1} & u_{s2} & \cdots & u_{s,k-1} & u_{sk} \\ \hline b_1 & b_2 & \cdots & b_s & v_1 & v_2 & \cdots & v_{k-1} & v_k \\ 0 & 0 & \cdots & 0 & 1 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 1 & 0 \end{array} \right] \quad (2-71)$$

2.1.7 Peer 方法

Weiner 等人提出了 Peer 方法^[36], 该方法使得所有的级数值都有相同的特点, 同时没有额外的求解向量使用。对于两步 Peer 方法, 在一致网格划分下有如下形式

$$Y_i^{[n]} = \sum_{j=1}^s b_{ij} Y_j^{[n]} + h \sum_{j=1}^s a_{ij} f(Y_j^{[n-1]}) + h \sum_{j=1}^s r_{ij} f(Y_j^n) \quad (2-72)$$

其中, $i = 1, 2, \dots, s$, 在向量形式下有

$$Y^{[n]} = (\mathbf{B} \otimes \mathbf{I}) Y^{[n-1]} + h(\mathbf{A} \otimes \mathbf{I}) F(Y^{[n-1]}) + h(\mathbf{R} \otimes \mathbf{I}) F(Y^{[n]}) \quad (2-73)$$

而 $Y^{[n]}$ 是 $y(t_n + c_i h)$, $i = 1, 2, \dots, s$ 的逼近值。上述格式在广义线性法的形式下表出为

$$\left[\begin{array}{c} Y^{[n]} \\ Y^{[n]} \\ hF(Y^{[n]}) \end{array} \right] = \left[\begin{array}{c|c|c} \mathbf{R} & \mathbf{B} & \mathbf{A} \\ \hline \mathbf{R} & \mathbf{B} & \mathbf{A} \\ \hline \mathbf{I} & \mathbf{0} & \mathbf{0} \end{array} \right] \left[\begin{array}{c} hF(Y^{[n]}) \\ Y^{[n-1]} \\ hF(Y^{[n-1]}) \end{array} \right] \quad (2-74)$$

2.2 相容性

这一节, 将建立广义线性法 (2-12) 的相容性条件。当然, 从广义线性法的相容条件可以退化为线性多步法和 Runge-Kutta 法的相容性条件。假设下列两个向量存在

$$\mathbf{q}_0 = [q_{1,0} \quad q_{2,0} \quad \cdots \quad q_{r,0}]^T \quad (2-75)$$

$$\mathbf{q}_1 = [q_{1,1} \quad q_{2,1} \quad \cdots \quad q_{r,1}]^T \quad (2-76)$$

使得输入向量 $y^{[n-1]}$ 满足

$$y_i^{[n-1]} = q_{i,0}y(t_{n-1}) + q_{i,1}hy'(t_{n-1}) + O(h^2), \quad i = 1, 2, \dots, r \quad (2-77)$$

同时, 要求级数值 $Y^{[n]}$ 和输出向量 $y^{[n]}$ 的分量满足

$$Y_i^{[n]} = y(t_{n-1} + c_i h) + O(h^2), \quad i = 1, 2, \dots, s \quad (2-78)$$

$$y_i^{[n]} = q_{i,0}y(t_n) + q_{i,1}hy'(t_n) + O(h^2), \quad i = 1, 2, \dots, r \quad (2-79)$$

正如文 [20,37,38] 所说, 将这些关系带入广义线性法 (2-12) 可得

$$y(t_{n-1}) + hc_i y'(t_{n-1}) = h \sum_{j=1}^s a_{ij} y'(t_{n-1}) \quad (2-80)$$

$$+ \sum_{j=1}^r u_{ij} (q_{j,0}y(t_{n-1}) + hq_{j,1}y'(t_{n-1})) + O(h^2), \quad i = 1, 2, \dots, s$$

$$q_{i,0}y(t_n) + q_{i,1}hy'(t_n) = h \sum_{j=1}^s b_{ij} y'(t_n) \quad (2-81)$$

$$+ \sum_{j=1}^r v_{ij} (q_{j,0}y(t_{n-1}) + hq_{j,1}y'(t_{n-1})) + O(h^2), \quad i = 1, 2, \dots, r$$

在上述两式总比较 $O(1)$ 和 $O(h)$ 项的系数可得

$$\sum_{j=1}^r u_{ij} q_{j,0} = 1, \quad i = 1, 2, \dots, s \quad \sum_{j=1}^r v_{ij} q_{j,0} = q_{i,0}, \quad i = 1, 2, \dots, r \quad (2-82)$$

和

$$\sum_{j=1}^s a_{ij} + \sum_{j=1}^r u_{ij} q_{j,1} = c_i, \quad i = 1, 2, \dots, s \quad (2-83a)$$

$$\sum_{j=1}^s b_{ij} + \sum_{j=1}^r v_{ij} q_{j,1} = q_{i,0} + q_{i,1}, \quad i = 1, 2, \dots, r \quad (2-83b)$$

于是, 就有如下定义

定义 2.1 (预相容) 广义线性法 (c, A, U, B, V) 是预相容的^[20,37,38], 如果存在向量 q_0 使得

$$Uq_0 = e, \quad Vq_0 = q_0 \quad (2-84)$$

其中, $e = [1, \dots, 1]^T \in \mathbb{R}^s$ 。而向量 q_0 称为预相容向量。

定义 2.2 (相容) 广义线性法 (c, A, U, B, V) 是相容的^[20,37,38], 如果它带有预相容向量 q_0 且存在向量 q_1 使得

$$Be + Vq_1 = q_0 + q_1 \quad (2-85)$$

其中, $e = [1, \dots, 1]^T \in \mathbb{R}^r$ 。而向量 q_1 称为相容向量。

从相容性的定义可以看出, 相容是建立在预相容的基础上做了进一步的要求。亦即算法是相容的, 也暗示了算法是预相容的。

定义 2.3 (级相容) 广义线性法 (c, A, U, B, V) 是级相容的^[38], 如果满足

$$Ae + Uq_1 = c \quad (2-86)$$

实质上, 上述的预相容、相容条件都是保证了广义线性法 (2-12) 在求解一阶微分方程 $y'(t) = 0$ 和 $y'(t) = 1$ 时能获得精确解。对于其更加详细的介绍可以参见文献 [39–41]。

针对 Runge-Kutta 方法, 可以很容易判断其预相容和相容向量分别为 $q_0 = 1$ 和 $q_1 = 0$, 并且其级相容和相容性条件就等价于

$$Ae = c, \quad b^T e = 1 \quad (2-87)$$

特别地, 级相容条件 $Ae = c$ 并不是必需条件, 不满足此条件的 Runge-Kutta 方法已经有学者进行了研究^[42–44]。因此就剩余相容性条件 $b^T e = 1$ 被必须满足。

而对于线性多步法 (2-14) 在 $r = 2k, s = 1$ 的广义线性法表示形式下, 其预相容和相容向量分别为

$$\begin{aligned} q_0 &= \left[1 \quad \dots \quad 1 \mid 0 \quad \dots \quad 0 \right]^T \in \mathbb{R}^{2k} \\ q_1 &= \left[0 \quad -1 \quad \dots \quad -(k-1) \mid 1 \quad \dots \quad 1 \right]^T \in \mathbb{R}^{2k} \end{aligned} \quad (2-88)$$

将其带入预相容、相容和级相容条件可得

$$\sum_{j=1}^k \alpha_j = 1, \quad \sum_{j=1}^k j\alpha_j = \sum_{j=0}^k \beta_j, \quad c = 1 \quad (2-89)$$

而对于线性多步法定另外一种较为紧凑的广义线性法表示形式下, 其预相容和相容向量分别为

$$\mathbf{q}_0 = \begin{bmatrix} \alpha_k \\ \alpha_{k-1} + \alpha_k \\ \vdots \\ \sum_{j=1}^k \alpha_j \end{bmatrix} \in \mathbb{R}^k, \quad \mathbf{q}_1 = \begin{bmatrix} \beta_k \\ \beta_{k-1} + \beta_k - \alpha_k \\ \vdots \\ \sum_{j=1}^k \beta_j - \sum_{j=1}^k (j-1)\alpha_j \end{bmatrix} \in \mathbb{R}^k \quad (2-90)$$

尽管预相容和相容向量不一致，但是得到的对应的条件却是一致的，都是等式 (2-89)。

对于 One-leg 方法 (2-52)，我们也可以得到其预相容和相容向量

$$\mathbf{q}_0 = \mathbf{e} \in \mathbb{R}^k, \quad \mathbf{q}_1 = [0 \quad -1 \quad \cdots \quad -(k-1)]^T \in \mathbb{R}^k \quad (2-91)$$

而对应的预相容、相容和级相容条件分别为

$$\sum_{j=1}^k \alpha_j = 1, \quad \sum_{j=1}^k j\alpha_j = \sum_{j=0}^k \beta_j, \quad c = 1 - \frac{1}{\beta} \sum_{j=1}^k j\beta_j \quad (2-92)$$

对于修改的扩展向后微分格式 (MEBDF)(2-65)，其预相容和相容向量分别为

$$\mathbf{q}_0 = \mathbf{e}, \quad \mathbf{q}_1 = [0 \quad -1 \quad \cdots \quad -(k-1)]^T \quad (2-93)$$

这两个向量与 One-leg 方法所对应的预相容、相容向量相等。进而其预相容条件则可写为

$$1 + \sum_{j=0}^{k-1} \hat{\alpha}_j = 0, \quad 1 + \sum_{j=0}^{k-1} \alpha_j = 0 \quad (2-94)$$

相容性条件为

$$\beta_k + \beta_{k+1} + \sum_{j=0}^{k-1} (k-j)\alpha_j = 0 \quad (2-95)$$

以及级相容条件则为

$$c_1 = \hat{\beta}_k + \sum_{j=0}^{k-1} (k-j)\hat{\alpha}_j + 1 \quad (2-96a)$$

$$c_2 = (\hat{\alpha}_{k-1} - 1) \left(2 - \hat{\beta}_k + \sum_{j=0}^{k-1} (k-j)\hat{\alpha}_j \right) \quad (2-96b)$$

$$c_3 = \beta_k + \beta_{k+1} + \sum_{j=0}^{k-1} (k-j)\alpha_j + 1 \quad (2-96c)$$

对于两步的 Runge-Kutta 方法 (2-66)，其预相容和相容向量分别为

$$\mathbf{q}_0 = \left[\begin{array}{cc|cc} 1 & 1 & 0 & \dots & 0 \end{array} \right]^T \in \mathbb{R}^{s+2}, \quad \mathbf{q}_1 = \left[\begin{array}{cc|cc} 0 & -1 & 1 & \dots & 1 \end{array} \right]^T \in \mathbb{R}^{s+2} \quad (2-97)$$

而对应的预相容条件自动满足，其相容和级相容条件则为

$$(\mathbf{v}^T + \mathbf{w}^T)\mathbf{e} = 1 + \theta, \quad (\mathbf{A} + \mathbf{B})\mathbf{e} - \mathbf{u} = \mathbf{c} \quad (2-98)$$

对于多步 Runge-Kutta 方法 (2-69)，其预相容和相容向量也与 One-leg 方法和 MEBDF 方法的对于向量一致，即等式 (2-91)。而预相容条件为

$$\mathbf{U}\mathbf{e} = \mathbf{e}, \quad \mathbf{v}^T \mathbf{e} = 1 \quad (2-99)$$

相容性条件为

$$\sum_{j=1}^s b_j - \sum_{j=1}^{k-1} jv_{j+1} = 1 \quad (2-100)$$

级相容性条件为

$$c_i = \sum_{j=1}^s a_{ij} - \sum_{j=1}^k ju_{ij}, \quad i = 1, 2, \dots, s \quad (2-101)$$

对于两步的 Peer 方法 (2-72)，其预相容和相容向量为

$$\mathbf{q}_0 = \begin{bmatrix} \mathbf{e} \\ \mathbf{0} \end{bmatrix} \in \mathbb{R}^{2s}, \quad \mathbf{q}_1 = \begin{bmatrix} \mathbf{c} - \mathbf{e} \\ \mathbf{e} \end{bmatrix} \in \mathbb{R}^{2s} \quad (2-102)$$

而预相容条件为

$$\mathbf{B}\mathbf{e} = \mathbf{e} \quad (2-103)$$

同时，相容和级相容条件退化为一组，为

$$(\mathbf{R} + \mathbf{A})\mathbf{e} + \mathbf{B}(\mathbf{c} - \mathbf{e}) = \mathbf{c} \quad (2-104)$$

总之，对于可以转化为广义线性法形式的数值算法，其相容性都可以由其前述定义导出。不过，对于一个具体的算法，其预相容和相容向量的显式表达式需要小心计算，而由其预相容定义可以得知，其预相容向量 \mathbf{q}_0 是矩阵 \mathbf{V} 的特征值为 1 所对应的一个特征向量。

2.3 稳定性

这一节，我们将重点阐述数值算法的稳定性，以期可以收集和整理截至目前为止的各种数值算法稳定定义的收集，因为只有对各种稳定性的定义了解和对比之后，才能发现其差异性。

在进一步阐述稳定性之前，我们先给出一些关于一个矩阵 A 的稳定性定义。

定义 2.4 (矩阵的稳定性/幂有界) 一个矩阵 A 是稳定的或者幂有界的^[20,37]，如果存在一个常数 C 使得对于任意的 $n = 0, 1, \dots$ 都有

$$\|A^n\| \leq C \quad (2-105)$$

定义 2.5 (矩阵的收敛性) 一个矩阵 A 是收敛的^[20,37]，如果当 $n \rightarrow \infty$ 有

$$\|A^n\|_{\infty} \rightarrow 0 \quad (2-106)$$

定义 2.6 (多项式的根条件) 一个多项式 p 满足根条件^[24,37]，如果多项式 p 的所有零根都位于闭的单位圆内，而仅单根可能位于单位圆上。

定理 2.1 (矩阵收敛的等价条件) 对于两个矩阵 A 和 B ，下列的陈述等价^[20,37]：

- (i) 矩阵 A 是收敛的。
- (ii) 矩阵 A 的谱半径小于 1，即 $\rho(A) < 1$ 。
- (iii) 存在一个相似于 A 的矩阵 B ，使得 $\|B\|_{\infty} < 1$ 。

证明 参见文献 [20]。 □

定理 2.2 (矩阵稳定的等价条件) 对于两个矩阵 A 和 B ，下列的陈述等价^[20,37]：

- (i) 矩阵 A 是稳定的。
- (ii) 矩阵 A 的最小多项式满足根条件要求。
- (iii) 存在一个相似于 A 的矩阵 B ，使得 $\|B\|_{\infty} \leq 1$ 。

证明 参见文献 [20]。 □

从矩阵的稳定性和收敛性的等价条件可知，其收敛性强于稳定性。下面给出两个很有用的不等式条件。

定理 2.3 (稳定矩阵的鲁棒性) 一个稳定的矩阵 A ，对于任意具有相同维度的矩阵 B 都会存在一个常数 C 使得对于任意的 $n = 1, 2, \dots$ 都有

$$\left\| \left(A + \frac{1}{n} B \right)^n \right\| \leq C \quad (2-107)$$

成立。

证明 参见文献 [20]。 \square

上述定理说明了对任意稳定的矩阵进行微扰动后仍然是稳定的。

定理 2.4 矩阵 \mathbf{A} 是 $m \times m$ 维稳定的矩阵, 且函数 $\Phi: \mathbb{R}^m \rightarrow \mathbb{R}^m$ 满足带有 Lipschitz 常数 L 的 Lipschitz 条件。若存在两个序列 $w = (w_1, w_2, \dots, w_n)$ 和 $v = (v_1, v_2, \dots, v_n)$ 满足条件

$$v_i = \mathbf{A}v_{i-1} + \frac{1}{n}\Phi(v_{i-1}) + w_i, \quad i = 1, 2, \dots, n \quad (2-108)$$

则存在独立于 n 的 C 使得

$$\|v_n\| \leq C \left(\|v_0\| + \sum_{j=1}^n \|w_j\| \right) \quad (2-109)$$

证明 参见文献 [20]。 \square

考虑一个非常简单的一阶微分方程

$$y'(t) = 0, \quad y(t_0) = y_0 \quad (2-110)$$

当使用广义线性法 (2-12) 进行求解时, 可以得到

$$y^{[n]} = \mathbf{V}y^{n-1} = \mathbf{V}^n y^{[0]}, \quad n = 0, 1, \dots \quad (2-111)$$

由此可以知道, 为了使得广义线性法求解得到的数值解稳定, 则需要对广义线性法的矩阵 \mathbf{V} 满足一定的条件。即有如下定义

定义 2.7 (零稳定性) 广义线性法 $(c, \mathbf{A}, \mathbf{U}, \mathbf{B}, \mathbf{V})$ 是零稳定的^[20,37,38], 如果存在一个常数 C 使得对于任意的 $n = 0, 1, \dots$ 都有

$$\|\mathbf{V}^n\| \leq C \quad (2-112)$$

可以看出, 广义线性法的零稳定性就是其矩阵 \mathbf{V} 的零稳定性。对于线性多步法, 其对应的零稳定性就是矩阵 \mathbf{V} 的最小多项式 p 满足根条件。亦即有下述定理

定理 2.5 (零稳定性的等价转述) 广义线性法 $(c, \mathbf{A}, \mathbf{U}, \mathbf{B}, \mathbf{V})$ 是零稳定的^[38], 如果其系数矩阵 \mathbf{V} 的最小多项式满足根条件。

定义 2.8 (弱零稳定性) 广义线性法 $(c, \mathbf{A}, \mathbf{U}, \mathbf{B}, \mathbf{V})$ 是弱零稳定的^[19], 如果它是零稳定的, 且矩阵 \mathbf{V} 有两个或以上的特征值位于单位圆上。此时, 也称矩阵 \mathbf{V} 的最小多项式满足弱根条件。

定义 2.9 (强零稳定性) 广义线性法 (c, A, U, B, V) 是强零稳定的^[19,24,38], 如果它是零稳定的, 且矩阵 V 有且仅有一个特征值位于单位圆上。此时, 也称矩阵 V 的最小多项式满足强根条件。

对于 Runge-Kutta 方法 (2-42), 其最小多项式为 $p(w) = w - 1$, 因此 Runge-Kutta 方法都是零稳定的, 而且还是严格零稳定的。而对于线性多步法 (2-14) 在广义线性表示形式 $(r = 2k, s = k)$ 下, 其最小多项式为

$$p(w) = w^k \rho(w), \quad \rho(w) = w^k - \sum_{j=1}^k \alpha_j w^{k-j} \quad (2-113)$$

而对于 One-leg 方法 (2-52) 的最小多项式为 $\rho(w)$ 。因此对于线性多步法和 One-leg 方法在多项式 $\rho(w)$ 满足根条件时实现零稳定性。对于两步 Runge-Kutta 方法 (2-66), 其最小多项式为

$$p(w) = w(w^2 - (1 - \vartheta)w - \vartheta) \quad (2-114)$$

其零根为 $w = 0, w = 1$ 和 $w = -\vartheta$ 。因此, 当条件 $-1 < \vartheta \leq 1$ 满足时, TSRK 实现零稳定性。

接下来, 我们通过一个著名的 Dahlquist 测试微分方程

$$y'(t) = \lambda y(t), \quad y(t_0) = y_0 \quad (2-115)$$

来研究广义线性法 (2-12) 的稳定性特点。其中, λ 是一个复数值。对微分方程 (2-115) 使用广义线性法 (2-12) 可以得到

$$Y^{[n]} = h\lambda AY^{[n]} + UY^{[n-1]} \quad (2-116a)$$

$$y^{[n]} = h\lambda BY^{[n]} + Vy^{[n-1]} \quad (2-116b)$$

令 $z = h\lambda$, 且假设矩阵 $I - zA$ 可逆, 则有

$$Y^{[n]} = (I - zA)^{-1}UY^{[n-1]} \quad (2-117)$$

进一步, 可以得到

$$y^{[n]} = M(z)y^{[n-1]} \quad (2-118)$$

其中, $M(z)$ 被定义为广义线性法 (2-12) 的稳定矩阵, 其显式表达为

$$M(z) = V + zB(I - zA)^{-1}U \quad (2-119)$$

进而也可以定义稳定函数 $p(w, z)$ 如下

$$p(w, z) = \det(wI - M(z)) \quad (2-120)$$

其中, $w \in \mathbb{C}$ 。

定义 2.10 (广义稳定的) 广义线性法 (c, A, U, B, V) 对给定的 $z \in \mathbb{C}$ 是广义稳定的^[24], 如果对于这样的 z , 稳定函数 $p(w, z)$ 满足根条件。

定义 2.11 (绝对稳定性) 广义线性法 (c, A, U, B, V) 对给定的 $z \in \mathbb{C}$ 是绝对稳定的^[24,38], 如果对于这样的 z , 稳定函数 $p(w, z)$ 的所有零根 $w_i = w_i(z), i = 1, 2, \dots, r$ 都位于单位圆内 (亦或定义为, 稳定矩阵 $M(z)$ 的谱半径都严格小于 1)。

需要说明的是, Gear 在他的书 [19] 中定义绝对稳定的算法为其稳定函数满足根条件, 亦指这里的广义稳定性。

定义 2.12 (相对稳定性) 广义线性法 (c, A, U, B, V) 对给定的 $z \in \mathbb{C}$ 是相对稳定的^[19,45], 如果对于这样的 z , 稳定函数 $p(w, z)$ 的主根模值大于等于其他虚假根的模值。

定义 2.13 (绝对稳定域/区间) 广义线性法 (c, A, U, B, V) 的绝对稳定域 \mathcal{A} 是所有 $z \in \mathbb{C}$ 使其广义线性法绝对稳定的集合, 即

$$\mathcal{A} = \{z \in \mathbb{C} : |w_i(z)| < 1, i = 1, 2, \dots, r\} \quad (2-121)$$

而绝对稳定区间 \mathcal{I} 就是绝对稳定域 \mathcal{A} 与实轴的交集, 即 $\mathcal{I} = \mathcal{A} \cap \mathbb{R}$ 。

同理, 我们也可以定义对应广义稳定域和相对稳定域, 由于使用较少, 故这里不再显式给出。

定义 2.14 (A 稳定性) 广义线性法 (c, A, U, B, V) 是 A 稳定的^[38], 如果它的绝对稳定域包含复平面的左半平面, 即

$$\{z \in \mathbb{C} : \operatorname{Re}(z) < 0\} \subseteq \mathcal{A} \quad (2-122)$$

当使用或构造一个数值积分算法时，我们很自然的希望它是 A 稳定的，因为 A 稳定的算法表示它对任意大的积分步长都是稳定的，稳定性的要求对积分步长 h 并没有构成任何的限制，这是我们仅需要依据精度条件来选择积分步长 h 。因此，在构造一个求解刚性微分方程的数值算法时，往往应对其进行 A 稳定性分析。但是，Dahlquist 在引入 A 稳定性的同时，得到了如下的两个结论^[5,23,45-47]：

‡ 显式的线性多步法 (包括显式的 Runge-Kutta 方法) 不可能是 A 稳定的。

‡ A 稳定的隐式线性多步法的阶数不可能超过 2，且在所有 A 稳定的二阶方法中，Trapeoidal 公式具有最小的局部截断误差常数 ($C = \frac{1}{12}$)。

由于上述结论的存在和推动，以及为了突破这些结论的限制，刚性微分方程的数值算法的研究主要想两个方面进行发展：

(i) 减弱 A 稳定性的定义，也就是说要求设计的数值算法的稳定区域不必包括整个开左半平面，而只需要包括其中的一部分。这样就先后由 Widlund 提出了 $A(\alpha)$ 稳定性^[48]，由 Gear 引进的刚性稳定性^[19] 以及由 Cryer 提出的 A_0 稳定性^[49]。构造具有这些稳定性的数值方法仍然适合求解一大类刚性问题。它们的最高精度阶也可以超过 2 阶，进而突破了前述限制。

(ii) 保持 A 稳定性的定义，构造具有新形式的数值积分算法。至少在结构动力学数值求解算法里面，可以通过引入结构依赖于的参数实现这一目的。

定义 2.15 (刚性 A 稳定) 一个数值算法被称为是刚性 A 稳定的^[19]，如果对于微分方程 (2-115) 的数值解满足，当 $\text{Re}(z) \rightarrow -\infty$ 时，有

$$\frac{y_{n-1}}{y_n} \rightarrow 0 \quad (2-123)$$

成立。

一般情况下，A 稳定是很困难或是不可能实现的，因此一个相对较弱的稳定性条件被提出来，即 $A(\alpha)$ 稳定性。

定义 2.16 ($A(\alpha)$ 稳定性) 一个数值算法是 $A(\alpha)$ 稳定的^[20,24,48]，如果存在一个角度 α 使得

$$S_\alpha = \{z \in \mathbb{C} : |\arg(-z)| < \alpha\} \subseteq \mathcal{A} \quad (2-124)$$

其中， $\arg(z)$ 表示复数 z 的角度。

另外可代替 A 稳定性的是刚性稳定^[19]，其定义如下

定义 2.17 (刚性稳定的) 广义线性法 (c, A, U, B, V) 是刚性稳定的^[19]，如果在复平面区域 $R_1 \triangleq \{z : \operatorname{Re}(z) \leq D\}$ 它是绝对稳定的；而在区域 $R_2 \triangleq \{z : D < \operatorname{Re}(z) < \alpha, |\operatorname{Im}(z)| < \theta\}$ 它是精确的。

其在复平面内的示意图如图2-3所示，

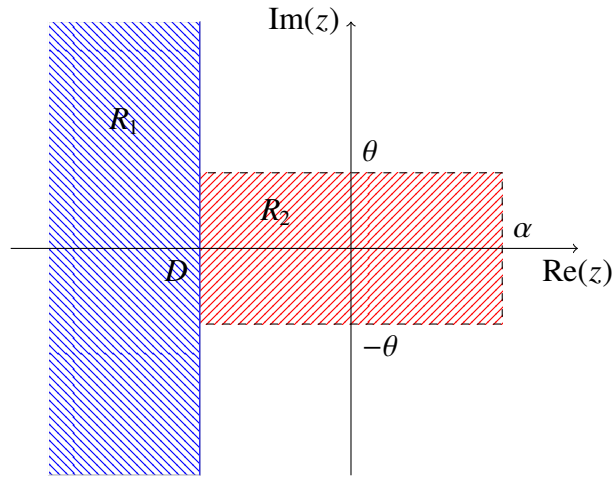


图 2-3 刚性稳定域
Fig.2-3 Stiffly stability

刚性稳定的算法可以推导出该算法也是 $A(\alpha)$ 稳定的，其中 $\alpha = \tan^{-1}(\theta/D)$ ^[50]。

定义 2.18 (极端稳定的) 广义线性法 (c, A, U, B, V) 在 ∞ 是极端稳定的^[24]，如果有

$$\lim_{z \rightarrow \infty} \rho(M(z)) = 0 \quad (2-125)$$

其中， $\rho(M(z))$ 表示稳定矩阵 $M(z)$ 的谱半径。

定义 2.19 (L 稳定性) 广义线性法 (c, A, U, B, V) 是 L 稳定的^[20,24,38]，如果它是 A 稳定且是极端稳定的。

2.4 收敛性

2.5 精度分析

2.5.1 数值耗散

2.5.2 数值弥散

2.6 超调行为

2.7 虚假根分析

2.8 直接积分法的放大矩阵

定义 2.20 直接积分算法的一致性^[51]。如果两个直接积分算法的放大矩阵 A 的对应元素分别相等^①，则称这两个积分算法是一致的。

定义 2.21 直接积分算法的相似性/谱等价^[51]。如果两个直接积分算法的放大矩阵 A 的不变量分别对应相等，则称这两个积分算法是相似的/谱等价的。

需要说明的是，一般情况下，两个谱等价的算法并不一定能导出一致的数值算法特性；而两个算法的一致性可以保证。这样的例子可以参见文献 [52]。

2.9 本章小结

第 3 章 结构响应数值算法

对于结构动力学问题，在经过合适的空间离散离散离散之后，得到一个关于时间的二阶常微分方程，即结构动力学运动方程 (3-1)

$$M\ddot{U}(t) + C\dot{U}(t) + KU(t) = F(t) \quad (3-1)$$

带有初值条件 (3-2)

$$U(0) = {}^0U \quad \dot{U}(0) = {}^0\dot{U} \quad (3-2)$$

对于该微分方程的求解，最常用的技术手段就是直接积分法，该方法的要求运动方程 (3-1) 在离散的时刻精确满足。同时假定位移和速度的更新方程，利用离散时刻的平衡方程求得对应时刻的加速度，进而求得速度和位移。特别地，自从 Newmark- β 算法^[53]、Wilson- θ 法^[54] 和 Houbolt 法^[55] 等直接积分法提出以来，许多的研究学者在此之后基于各种原理提出了许多性能优良的积分算法。各种不同类型的直接积分法可以参见书籍^[5,22,23,56,57] 以及两篇关于显、隐式直接积分法的综述文章^[58,59]。

在方程 (3-1) 中的 M, C, K 分别表示质量、阻尼和刚度矩阵；而 $F(t)$ 表示系统所受到外力荷载，它一般情况下是关于时间 t 的已知函数；而 $U(t)$ 表示系统的位移向量，而其上标的 (\cdot) 代表对时间的微分，亦即 $\dot{U}(t), \ddot{U}(t)$ 分别表示其速度和加速度。

特别地，若引入中间变量 $\dot{U}(t) = V(t)$ 可将二阶的微分方程 (3-1) 转换为一阶微分方程组

$$\begin{bmatrix} \dot{U}(t) \\ \dot{V}(t) \end{bmatrix} = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ -M^{-1}K & -M^{-1}C \end{bmatrix} \begin{bmatrix} U(t) \\ V(t) \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ M^{-1}F(t) \end{bmatrix} \quad (3-3)$$

其初始条件为

$$U(0) = {}^0U \quad V(0) = {}^0\dot{U} \quad (3-4)$$

其中， $\mathbf{0}$ 和 \mathbf{I} 分别为零矩阵和单位矩阵^①。于是，在常微分方程理论中的数值算法

① 其矩阵维度与系统自由度数相同。特别地，在不引起混淆的情况下，后面都不再特别指明矩阵的维度问题。

都可以运用到该一阶微分方程中求解。但是在转换后的位移-速度场情况下进行求解，系数矩阵的维度增加，同时位移、速度的微分关系只能以弱形式满足。需要说明的是，在后文中提及算法设计时所说的位移-速度场亦指该算法的设计是以一阶运动方程 (3-3) 为基础的；而 (单) 位移场则说明算法设计是以二阶运动方程 (3-1) 为基础的。

一般情况下，对多自由度运动方程进行算法的设计和分析是一件非常繁琐和复杂的工作。幸运的是，借助模态正交性原则^[5]且假设阻尼为经典的粘性阻尼，可以将多自由度运动方程 (3-1) 解耦为单自由度模态方程

$$\ddot{u}(t) + 2\xi\omega\dot{u}(t) + \omega^2u(t) = f(t) \quad (3-5)$$

其中， ξ, ω 和 $f(t)$ 分别为阻尼率、系统的固有频率和模态外激励力； $u(t)$ 为未知的位移。

类似的，对于一阶解耦的运动方程也可以由上述一阶位移场的运动方程经引入中间变量转换得到。这里给出另外一种等价表述方式^[60,61]

$$\begin{cases} \dot{y}_1(t) = \lambda_2 y_1(t) + f(t) \\ \dot{y}_2(t) = \lambda_1 y_2(t) + f(t) \end{cases} \quad (3-6)$$

其中，

$$\lambda_{1,2} = -\xi\omega \pm j\sqrt{1-\xi^2}\omega \quad (3-7)$$

而 $j = \sqrt{-1}$ 为虚数单位且 $0 \leq \xi < 1$ 。原始的一阶微分方程的解 $u(t)$ 与该一阶方程组的解 $y_1(t), y_2(t)$ 的关系为

$$\begin{cases} y_1(t) = \dot{u}(t) - \lambda_1 u(t) \\ y_2(t) = \dot{u}(t) - \lambda_2 u(t) \end{cases} \quad (3-8)$$

于是就有

$$u(t) = -\frac{y_1(t) - y_2(t)}{\lambda_1 - \lambda_2} \quad (3-9)$$

注意，变量 $\lambda_{1,2}$ 和 $y_{1,2}(t)$ 为复数。而最终的计算结果 $u(t)$ 为实数。

迄今为止，构造直接积分算法的主要框架有

★ 有限差分法 (线性多步法)[Finite Difference Methods/Linear Multistep Methods]。

- ★ 多级法 (Runge-Kutta 法)[Multistage Methods/Runge-Kutta Methods]。
- ★ 加权残值法 [Weighted Residual Methods]。
- ★ 泰勒级数配点法 [Taylor Series Collocation Methods]。
- ★ 哈密顿原理/定律 [Hamilton's Principle/Law]。
- ★ 最小二乘法 [Least-Squares Methods]。
- ★ 时间子步策略 [Composite Sub-steps Techinques]。
- ★ 结构依赖型参数策略 [Structure-dependent Parameters Techinques]。

本章后续内容，将对这些不同的策略和框架算法进行简单的总结和归纳。同时，在作者力所能及的范围内尽可能阐述清楚。

3.1 有限差分法 (线性多步法)

考虑带有初值条件 $y(0) = y(t_0 = 0)$ 的单变量一阶微分方程

$$\dot{y}(t) = f(y, t) \quad (3-10)$$

在时间 $t \in [0, T]$ 上的数值解。将时间区间 $[0, T]$ 划分为互不重叠的相等的小区间，即

$$[0, T] = \bigcup_{n=0}^{N-1} [t_n, t_{n+1}] \quad (3-11)$$

而 $\Delta t = t_{n+1} - t_n$ 为时间步长大小，在本文中都假定为常值不可变，亦即本文将不考虑变时间步长的积分算法。而时间积分算法一般是假定在 t_n 时刻之前的响应已知，求 t_{n+1} 时刻的响应。

针对一阶微分方程 (3-10) 的 k -步线性多步法^[5,19] 被定义如下

$$\begin{aligned} & \alpha_0 y^{t+\Delta t} + \alpha_1 y^t + \cdots + \alpha_k y^{t+(1-k)\Delta t} \\ & = \Delta t [\beta_0 f(t+\Delta t, y, t_{n+1}) + \beta_1 f(t, y, t_n) + \cdots + \beta_k f(t+(1-k)\Delta t, y, t_{n+1-k})] \end{aligned} \quad (3-12)$$

其中， α_i, β_i 为算法参数。上述方程可以改成为

$$\sum_{i=0}^k \{ \alpha_i y^{t+(1-i)\Delta t} + \Delta t \beta_i f(t+(1-i)\Delta t, y, t_{n+1-i}) \} = 0 \quad (3-13)$$

注意到各项之间是线性关系，且当 $\beta_0 = 0$ 时，该线性 k -步法为显式格式；否则为 k -步隐式算法。

考虑到在求解结构动力学问题时，经过有限元离散之后往往等到一个二阶微

分方程，故针对一般形式的二阶微分方程

$$\ddot{y} = f(\dot{y}(t), y(t), t) = g_0 y(t) + g_1 \dot{y}(t) + h(t) \quad (3-14)$$

其中， $g_0, g_1 \in \mathbf{R}$ 是常数。同时该微分方程也是假定为线性的。对于二阶微分方程 (3-14) 的 k -步的线性多步法^[62]

$$\sum_{i=0}^k \{ \alpha_i {}^{t+(1-i)\Delta t}y + \Delta t \beta_i g_1 {}^{t+(1-i)\Delta t}y + \Delta t^2 \gamma_i [g_0 {}^{t+(1-i)\Delta t}y + h(t_{n+1-i})] \} = 0 \quad (3-15)$$

其中， α_i, β_i 和 γ_i 均为算法参数。当参数 $\beta_0 = 0$ ，上述算法得到显式算法；否则是隐式的。关于该公式的更多的应用可以参考原文 [62]。

3.1.1 向前欧拉法

向前欧拉法可以通过将 $y(t_{n+1}) = {}^{n+1}y$ 在 t_n 时刻进行泰勒级数展开获得。即

$${}^{n+1}y = {}^ny + \Delta t {}^n\dot{y} + \frac{\Delta t^2}{2!} {}^n\ddot{y} + \frac{\Delta t^3}{3!} {}^ny^{(3)} + \dots \quad (3-16)$$

上式可以写为

$${}^{n+1}y = {}^ny + \Delta t {}^n\dot{y} + O(\Delta t^2) \quad (3-17)$$

其中， $O(\Delta t^2)$ 表示为局部截断误差。因此一阶导数可以由下列的向前差分得到

$${}^n\dot{y} = \frac{{}^{n+1}y - {}^ny}{\Delta t} + E_{trunc}(y, \Delta t) \quad (3-18)$$

于是，向前欧拉法由下式给出

$$\frac{{}^{n+1}y - {}^ny}{\Delta t} = {}^n\dot{y} \quad (3-19)$$

亦即

$${}^{n+1}y = {}^ny + \Delta t {}^n\dot{y} = {}^ny + \Delta t f({}^ny, t_n) \quad (3-20)$$

可以看出，当在 t_n 时刻的 ny 已知时，根据上述就可以求得在 t_{n+1} 时刻的 ${}^{n+1}y$ 值，而不需要求解 (非) 线性方程组。故向前欧拉法是单步显式方法，同时该算法是一阶条件稳定算法。

向前欧拉法是最简单也是最早被提出的一个数值算法。后续的许多学者基于此提出了许多改进欧拉法的方案，进而引出了许多其他性能较优的算法。感兴趣的读者可以参考书籍 [20,38]。

3.1.2 向后欧拉法

同理，若关于时间的一阶导数 $\dot{y}(t)$ 用向后差分公式进行逼近，则可以得到向后欧拉法

$${}^{n+1}\dot{y} = {}^n\dot{y} + \Delta t {}^{n+1}\ddot{y} = {}^n\dot{y} + \Delta t f({}^{n+1}y, t_n) \quad (3-21)$$

可以看出，为了求解 ${}^{n+1}\dot{y}$ ，需要求解一个线性方程组或者非线性方程组（取决于 $f(y, t)$ 的形式），于是向后欧拉法是单步隐式算法。同时，向后欧拉法是无条件稳定带渐进消弭特性的算法。即是 L-稳定的。

3.1.3 广义梯形算法

通过将 ${}^n\dot{y}$ 和 ${}^{n+1}\dot{y}$ 进行凸组合^[5,56]，即 $(0 \leq \alpha \leq 1)$

$${}^{n+\alpha}\dot{y} = (1 - \alpha){}^n\dot{y} + \alpha {}^{n+1}\dot{y} \quad (3-22)$$

结合

$${}^{n+1}y - {}^ny = \Delta t {}^{n+\alpha}\dot{y} \quad (3-23)$$

就可以得到具有如下形式的广义梯形公式

$${}^{n+1}y = {}^ny + \Delta t [(1 - \alpha){}^n\dot{y} + \alpha {}^{n+1}\dot{y}] \quad (3-24)$$

需要说明的时，当 $\alpha = 0$ 时，广义梯形公式退化为向前欧拉法；当 $\alpha = 1$ 时，广义梯形公式退化为向后欧拉法；而当 $\alpha = \frac{1}{2}$ 时，得到梯形公式

$${}^{n+1}y = {}^ny + \frac{\Delta t}{2}({}^n\dot{y} + {}^{n+1}\dot{y}) \quad (3-25)$$

该梯形公式也称为 Crank-Nicolson 公式。特别地，类似于梯形公式的运用在后续算法设计中使用频率相当高，读者应引起注意。

对于如下的半离散化方程

$$M\dot{d} + Kd = F \quad (3-26)$$

的广义梯形公式形式为

$$M{}^{n+1}\dot{d} + K{}^{n+1}d = {}^{n+1}F \quad (3-27)$$

$${}^{n+1}d = {}^nd + \Delta t {}^{n+\alpha}\dot{d} \quad (3-28)$$

$${}^{n+\alpha}\dot{d} = {}^n\dot{d}(1 - \alpha) + {}^{n+1}\dot{d}\alpha \quad (3-29)$$

将上述三式化为位移 d 和 F 的关系，则为

$$(M + \alpha \Delta t K)^{n+1} d = [M - (1 - \alpha) \Delta t K]^n d + \Delta t [{}^{n+1} F \alpha + {}^n F (1 - \alpha)] \quad (3-30)$$

当 $\alpha = \theta$ 时，上述算法也称为 θ -格式^[56]。

3.1.4 中点公式

当考虑变量 y 的如下 α 凸组合时

$${}^{n+\alpha} y = {}^{n+1} y \alpha + {}^n y (1 - \alpha) \quad (3-31)$$

同时，关于时间 t 的 β 凸组合有

$$t_{n+\beta} = \beta t_{n+1} + (1 - \beta) t_n \quad (3-32)$$

则变量 y 在区间 $[t_n, t_{n+1}]$ 的中点值 ${}^{n+1/2} y$ 为

$${}^{n+1/2} y = \frac{{}^{n+1} y + {}^n y}{2} \quad (3-33)$$

与此同时，关于时间的中点值为

$$t_{n+\frac{1}{2}} = \frac{t_{n+1} + t_n}{2} \quad (3-34)$$

对于一般的非线性情况，下式一般情况下不成立

$${}^{n+1/2} y \neq y(t_{n+\frac{1}{2}}) \quad (3-35)$$

根据 ${}^{n+1/2} y$ 是否做近似处理时，可以得到两个中点公式：

显式中点公式^[19]

$${}^{n+1} y = {}^n y + \Delta t {}^{n+1/2} \dot{y} = {}^n y + \Delta t f({}^{n+1/2} y, t_{n+\frac{1}{2}}) \quad (3-36)$$

$${}^{n+1/2} y = {}^n y + \frac{\Delta t}{2} f({}^n y, t_n) \quad (3-37)$$

$$t_{n+\frac{1}{2}} = \frac{t_{n+1} + t_n}{2} \quad (3-38)$$

隐式中点公式^[22]

$${}^{n+1} y = {}^n y + \Delta t {}^{n+1/2} \dot{y} = {}^n y + \Delta t f({}^{n+1/2} y, t_{n+\frac{1}{2}}) \quad (3-39)$$

$$t_{n+\frac{1}{2}} = \frac{t_{n+1} + t_n}{2} \quad (3-40)$$

对于线性系统，中点公式将退化为梯形公式 (Crank-Nicolson 公式)。

3.1.5 中心差分法

对于一阶系统 (3-10)，使用中心差分公式计算其一阶导数项，则有其中心差分格式为

$${}^n\dot{y} = \frac{{}^{n+1}y - {}^{n-1}y}{2\Delta t} \quad (3-41)$$

亦即

$${}^{n+1}y = {}^{n-1}y + 2\Delta t {}^n\dot{y} = {}^{n-1}y + 2\Delta t f({}^ny, t_n) \quad (3-42)$$

而对于二阶微分方程

$$\ddot{y}(t) = f(y(t), \dot{y}(t), t) \quad (3-43)$$

其二阶导数的中心差分仍可由泰勒级数展开得到，即

$${}^n\ddot{y} = \frac{{}^{n+1}y - 2{}^ny + {}^{n-1}y}{\Delta t^2} \quad (3-44)$$

于是，二阶微分方程 (3-43) 的中心差分格式为

$${}^{n+1}y = -{}^{n-1}y + 2{}^ny + \Delta t^2 {}^n\ddot{y} = -{}^{n-1}y + 2{}^ny + \Delta t^2 f({}^ny, {}^n\dot{y}, t) \quad (3-45)$$

对于位移场的运动方程 (3-1)，其中心差分格式则可描述如下

$$M^t\ddot{U} + C^t\dot{U} + K^tU = {}^tF \quad (3-46)$$

$${}^t\dot{U} = \frac{1}{2\Delta t}({}^{t+\Delta t}U - {}^{t-\Delta t}U) \quad (3-47)$$

$${}^t\ddot{U} = \frac{1}{\Delta t^2}({}^{t+\Delta t}U - 2{}^tU + {}^{t-\Delta t}U) \quad (3-48)$$

对于给定的初始条件，中心差分法并没有办法进行自启动；于是还需要引入额外的启动机制去计算 ${}^{-\Delta t}U$ 才能进行后续计算。常用的一个启动机制^[63] 为

$${}^{-\Delta t}U = {}^0U - {}^0\dot{U}\Delta t + \frac{1}{2}{}^0\ddot{U}\Delta t^2 \quad (3-49)$$

3.1.6 Houbolt 算法

Houbolt 算法^[62] 是最早计算结构动力学数值响应的算法之一。该算法使用了在 $t + \Delta t$ 时刻的运动平衡方程和两个向后微分公式。而向后微分公式通过使用连续四个节点的值来构造三次多项式完成。其算法格式可以描述如下

$$M^{t+\Delta t}\ddot{U} + C^{t+\Delta t}\dot{U} + K^{t+\Delta t}U = {}^{t+\Delta t}F \quad (3-50a)$$

$${}^{t+\Delta t}\dot{U} = \frac{1}{6\Delta t}(11{}^{t+\Delta t}U - 18{}^tU + 9{}^{t-\Delta t}U - 2{}^{t-2\Delta t}U) \quad (3-50b)$$

$${}^{t+\Delta t}\ddot{U} = \frac{1}{\Delta t^2}(2{}^{t+\Delta t}U - 5{}^tU + 4{}^{t-\Delta t}U - {}^{t-2\Delta t}U) \quad (3-50c)$$

通过使用速度和加速度的更新公式，在平衡方程中消去速度和加速度得到仅关于位移的差分方程如下

$$\left[\frac{2}{\Delta t^2}M + \frac{11}{6\Delta t} + K \right] {}^{t+\Delta t}U = {}^{t+\Delta t}F + \left[\frac{5}{\Delta t^2}M + \frac{3}{\Delta t}C \right] {}^tU - \left[\frac{4}{\Delta t^2}M + \frac{3}{2\Delta t}C \right] {}^{t-\Delta t}U + \left[\frac{1}{\Delta t^2}M + \frac{1}{3\Delta t}C \right] {}^{t-2\Delta t}U \quad (3-51)$$

显然，Houbolt 算法不是自启动的。对于在第一个时间步长内需要知道 ${}^{t-\Delta t}U$ 和 ${}^{t-2\Delta t}U$ ，这就意味着需要引入额外的启动机制。同时该算法是无条件稳定的、二阶精度的隐式算法。Houbolt 算法在高频段引入了很强的数值耗散。严格地说，它具有渐进消弭特性，即当 $\omega\Delta t \rightarrow \infty$ 时，有 $\rho_\infty = 0$ 。它的无条件稳定性加消弭特性意味着 Houbolt 算法是 L-稳定的。同时它的低频段耗散和相对周期误差在常用的数值算法中也是最高的，这也可以看成它除了非自启动外的另外一个重要的缺陷。

当考虑 Geradin 给出的对于二阶微分方程的如下逼近格式^[64]时，

$$M^{t+\Delta t}\ddot{U} + C^{t+\Delta t}\dot{U} + K^{t+\Delta t}U = {}^{t+\Delta t}F \quad (3-52a)$$

$$\sum_{i=0}^k \alpha_i {}^{t+(1-i)\Delta t}U = \Delta t \sum_{i=0}^k \beta_i {}^{t+(1-i)\Delta t}\dot{U} \quad (3-52b)$$

$$\sum_{i=0}^k \gamma_i {}^{t+(1-i)\Delta t}U = \Delta t^2 \sum_{i=0}^k \eta_i {}^{t+(1-i)\Delta t}\ddot{U} \quad (3-52c)$$

Houbolt 算法则可以由以下参数表示：($k = 3$)

$$\begin{cases} \alpha_0 = 11, \alpha_1 = -18, \alpha_2 = 9, \alpha_3 = -2 \\ \beta_0 = 6, \beta_1 = \beta_2 = \beta_3 = 0 \\ \gamma_0 = 2, \gamma_1 = -5, \gamma_2 = 4, \gamma_3 = -1 \\ \eta_0 = 1, \eta_1 = \eta_2 = \eta_3 = 0 \end{cases} \quad (3-53)$$

3.1.7 带终点加速度的中点公式

当半离散化的位移场二阶运动微分方程 (3-1) 在 $t_{n+\frac{1}{2}}$ 时，有

$$M^{t+\Delta t/2}\ddot{U} + C^{t+\Delta t/2}\dot{U} + K^{t+\Delta t/2}U = {}^{t+\Delta t/2}F \quad (3-54)$$

同时，其速度和位移的更新公式为

$${}^{t+\Delta t}U = {}^tU + \Delta t {}^{t+\Delta t/2}\dot{U} \quad (3-55a)$$

$${}^{t+\Delta t}\dot{U} = {}^t\dot{U} + \Delta t {}^{t+\Delta t/2}\ddot{U} \quad (3-55b)$$

在方程 (3-54) 中的位移、速度、加速度以及外力在时刻 $t_{n+\frac{1}{2}}$ 的逼近表达式为

$$\begin{cases} {}^{t+\Delta t/2}U = \frac{1}{2}({}^tU + {}^{t+\Delta t}U) \\ {}^{t+\Delta t/2}\dot{U} = \frac{1}{2}({}^t\dot{U} + {}^{t+\Delta t}\dot{U}) \\ {}^{t+\Delta t/2}\ddot{U} = \frac{1}{2}({}^t\ddot{U} + {}^{t+\Delta t}\ddot{U}) \\ {}^{t+\Delta t/2}F = \frac{1}{2}({}^tF + {}^{t+\Delta t}F) \end{cases} \quad (3-56)$$

将等式 (3-56) 带入上两式，就可得到

$$\begin{aligned} \left[\frac{1}{2}M + \frac{\Delta t}{4}C + \frac{\Delta t^2}{8}K \right] {}^{t+\Delta t}\ddot{U} &= \frac{1}{2}({}^tF + {}^{t+\Delta t}F) \\ &- \left[\frac{1}{2}M + \frac{\Delta t}{4}C + \frac{\Delta t^2}{8}K \right] {}^t\ddot{U} - \left[C + \frac{\Delta t}{2}K \right] {}^t\dot{U} - K {}^tU \end{aligned} \quad (3-57)$$

带有位移和速度更新公式

$${}^{t+\Delta t}U = {}^tU + {}^t\dot{U}\Delta t + \frac{\Delta t^2}{4}({}^t\ddot{U} + {}^{t+\Delta t}\ddot{U}) \quad (3-58a)$$

$${}^{t+\Delta t}\dot{U} = {}^t\dot{U} + \frac{\Delta t}{2}({}^t\ddot{U} + {}^{t+\Delta t}\ddot{U}) \quad (3-58b)$$

这个算法被 K. K. Tamma 等人称为带终点加速度的中点公式^[64][Midpoint Rule method with end-point acceleration](MPR-EPA)。同时在上一个时间步长内的加速度在当前的计算中起着很重要的作用。特别地，该算法对线性结构动力学系统是能量、动量以及角动量都是守恒的。

3.1.8 Newmark 算法

在众多的求解结构动力学数值响应算法中，Newmark 家族算法^[53]可能是被公认使用最为广泛的算法。该算法在时间区间 $[t_n, t_{n+1}]$ 内使用加速度的凸组合进行计算，即

$$\hat{\ddot{U}} = (1 - \gamma){}^t\ddot{U} + \gamma {}^{t+\Delta t}\ddot{U} \quad (3-59)$$

其中, γ 表示自由参数。通过使用在 t_{n+1} 时刻的平衡方程, 同时对上述进行积分得到下列的 Newmark 算法家族:

$$M^{t+\Delta t}\ddot{U} + C^{t+\Delta t}\dot{U} + K^{t+\Delta t}U = {}^{t+\Delta t}F \quad (3-60a)$$

$${}^{t+\Delta t}U = {}^tU + {}^t\dot{U}\Delta t + \frac{\Delta t^2}{2}[(1-2\beta){}^t\ddot{U} + 2\beta{}^{t+\Delta t}\ddot{U}] \quad (3-60b)$$

$${}^{t+\Delta t}\dot{U} = {}^t\dot{U} + \Delta t[(1-\gamma){}^t\ddot{U} + \gamma{}^{t+\Delta t}\ddot{U}] \quad (3-60c)$$

其中, 参数 γ, β 控制着算法的数值性能。同时 Newmark 算法运用十分广泛。在线性结构动力学中, 使用最广泛的是 a-形式。即先显式的求解位移和速度

$${}^{t+\Delta t}\tilde{U} = {}^tU + {}^t\dot{U}\Delta t + \frac{\Delta t^2}{2}(1-2\beta){}^t\ddot{U} \quad (3-61a)$$

$${}^{t+\Delta t}\tilde{\dot{U}} = {}^t\dot{U} + \Delta t(1-\gamma){}^t\ddot{U} \quad (3-61b)$$

其初始加速度 ${}^0\ddot{U}$ 由初始时刻的平衡方程得到

$${}^0\ddot{U} = M^{-1}({}^0F - C{}^0\dot{U} - K{}^0U) \quad (3-62)$$

于是, 在 t_{n+1} 时刻的加速度由求解下来方程组得到

$$(M + \gamma\Delta tC + \beta\Delta t^2K){}^{t+\Delta t}\ddot{U} = {}^{t+\Delta t}F - C{}^{t+\Delta t}\tilde{\dot{U}} - K{}^{t+\Delta t}\tilde{U} \quad (3-63)$$

同时, 在 t_{n+1} 时刻的速度和位移由

$${}^{t+\Delta t}U = {}^{t+\Delta t}\tilde{U} + \beta\Delta t^2{}^{t+\Delta t}\ddot{U} \quad (3-64a)$$

$${}^{t+\Delta t}\dot{U} = {}^{t+\Delta t}\tilde{\dot{U}} + \gamma\Delta t{}^{t+\Delta t}\ddot{U} \quad (3-64b)$$

计算而得。

Newmark 算法的无条件稳定性的要求

$$2\beta \geq \gamma \geq \frac{1}{2} \quad (3-65)$$

对于条件稳定性, 应满足条件

$$\gamma \geq \frac{1}{2}, \quad \beta < \frac{1}{2}\gamma \quad \text{和} \quad \Omega \leq \Omega_{cr} \quad (3-66)$$

其中, $\Omega = \omega\Delta t$ 并且

$$\Omega_{cr} = \frac{\xi(\gamma - \frac{1}{2}) + \sqrt{\frac{1}{2}\gamma - \beta + \xi^2(\gamma - \frac{1}{2})^2}}{\frac{1}{2}\gamma - \beta} \quad (3-67)$$

需要注意的是，其稳定性条件应对系统的每个模态系统均成立，故一般情况下，最高阶模态系统应满足条件 (3-67)。特别地，对于参数 $\gamma < \frac{1}{2}$ ，Newmark 算法将产生负阻尼，进而产生无界振动的数值解；而对于 $\gamma > \frac{1}{2}$ ，将产生正的数值阻尼，且此时系统的粘性阻尼不能扩大系统的临界采样频率，故对于系统的模态阻尼估计不充分时，临界采样频率建议取值如下

$$\Omega_{cr} = \frac{1}{\sqrt{\gamma/2 - \beta}} \quad (3-68)$$

同时，当 $\gamma > \frac{1}{2}$ ，Newmark 算法尽管引入了数值耗散，但是算法的精度却退化为一阶。同时对于给定的 $\gamma > \frac{1}{2}$ ，可以通过选择 β 来实现最优耗散，即高频耗散最大化。其 β 取值为

$$\beta = \frac{(\gamma + \frac{1}{2})^2}{4} \quad (3-69)$$

Newmark 算法带有参数 $\beta = 1$ 和 $\gamma = \frac{3}{2}$ 提出了可能的最大高频耗散，但该算法对低频求解精度较低，一般很少使用。

于是，大多数情况下， $\gamma = \frac{1}{2}$ 被使用；此时算法没有数值耗散，但实现二阶精度。

当参数 γ, β 选择不同的值时，可以产生不同的数值直接算法，如下表所示：

表 3-1 Newmark 算法家族的著名成员
Table 3-1 Some well-known members for the Newmark family

γ	β	名称	精度	稳定性
1/2	0	中心差分法/Leap-Frog 格式	2	$\Omega_{cr} \leq 2$
1/2	1/12	Fox-Goodwin 格式	3	$\Omega_{cr} \leq \sqrt{6}$
1/2	1/6	线平均加速度法	2	$\Omega_{cr} \leq 2\sqrt{3}$
1/2	1/4	常平均加速度法/Trapeoidal 规则	2	$\Omega_{cr} \leq \infty$
1/2	1/2	Lee's 算法 ^[65]	2	--
3/2	4/5	Galerkin 方法	1	--
3/2	1	向后微分公式	1	--

其中，中心差分法是显式的，也是被认为最为经济、求解方便的积分算法，在显式算法中使用较广；对于常平均加速度算法，该算法在二阶的隐式算法中，拥有最小的误差主项^[46]，同时该算法也是 Trapeoidal 规则在二阶运动方程中的体现；对于 Fox-Goodwin 格式，当 $C \neq 0$ 时，实现三阶精度，而当无阻尼时，可以实现四阶精度。

3.1.8.1 基于有限元离散的加权残差推导

Newmark 算法除了使用在 $t + \Delta t$ 时刻的位移和速度的泰勒级数展开来获得其计算格式外，还可以通过假设在 $t - \Delta t, t, t + \Delta t$ 时刻的有限元离散，在进行加权残差推导亦可获得其计算格式^[66]，而且后者更可以获得其他新的高阶精度算法。

消去 Newmark 算法 (3-60) 中的速度和加速度，可以得到如下的关于位移的迭代式

$$\begin{aligned}
 & [M + \gamma \Delta t C + \beta \Delta t^2 K]^{t+\Delta t} U \\
 & + [-2M + (1 - 2\gamma) \Delta t C + (\frac{1}{2} + \gamma - 2\beta) \Delta t^2 K]^{tU} \\
 & + [M + (-1 + \gamma) \Delta t C + (\frac{1}{2} - \gamma + \beta) \Delta t^2 K]^{t-\Delta t} U \\
 & + \beta^{t+\Delta t} F \Delta t^2 + (\frac{1}{2} + \gamma - 2\beta)^t F \Delta t^2 + (\frac{1}{2} - \gamma + \beta)^{t-\Delta t} F \Delta t^2 = 0
 \end{aligned} \tag{3-70}$$

在时间域 $t - \Delta t, t, t + \Delta t$ 上，位移 $U(t)$ 可以表示为

$$U(t) = \sum_i N_i(t)^i U, \quad i = t - \Delta t, t, t + \Delta t \tag{3-71}$$

使用权重残差在时间域 $0, 2\Delta t$ 内积分得到

$$\int_0^{2\Delta t} W(t) [M \sum \ddot{N}_i^i U + C \sum \dot{N}_i^i U + K \sum N_i^i U - \sum^i F N_i] dt = 0 \tag{3-72}$$

其中，假定外力荷载 $F(t)$ 也使用了相同的形函数进行插值计算。而形函数的显式表达为 $(-1 \leq \xi = t/\Delta t \leq 1)$

$$\left. \begin{aligned}
 N_{t+\Delta t} &= \xi(1 + \xi)/2 \\
 N_t &= (1 - \xi)(1 + \xi) \\
 N_{t-\Delta t} &= -\xi(1 - \xi)/2
 \end{aligned} \right\} \tag{3-73}$$

其形函数如图3-1所示。

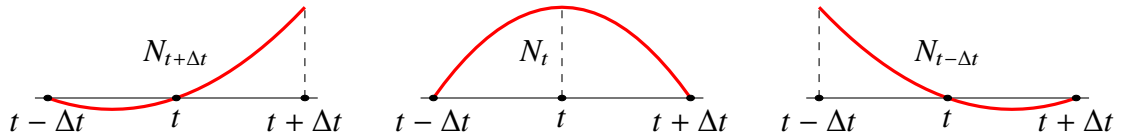


图 3-1 $N_{t+\Delta t}, N_t, N_{t-\Delta t}$ 二次形函数
Fig.3-1 Parabolic shape functions

于是，经过上述计算可以得到类似于 (3-70) 的表达式。进一步，若作如下参

数规定

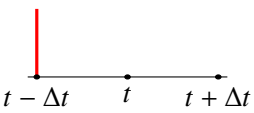
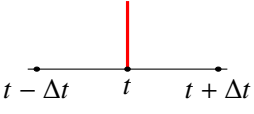
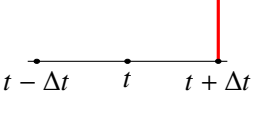
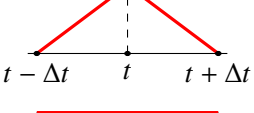
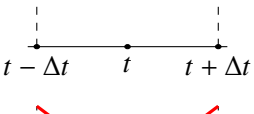
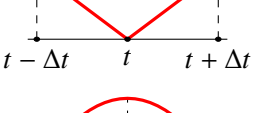
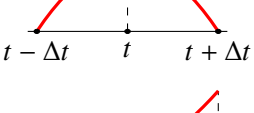
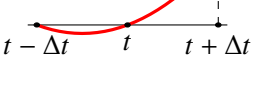
$$\gamma = \frac{\int_{-1}^1 W(t)\xi d\xi}{\int_{-1}^1 W(t)d\xi} + \frac{1}{2} \quad (3-74a)$$

$$\beta = \frac{\frac{1}{2} \int_{-1}^1 W(t)\xi(1+\xi)d\xi}{\int_{-1}^1 W(t)d\xi} \quad (3-74b)$$

则，推导得到的关系将与等式 (3-70) 一致，意味着得到了 Newmark 算法。特别地，表3-2给出了选择不同的权重函数 $W(t)$ 时，将分别得到不同参数 β, γ 所对应的积分算法。从表3-2可以发现，当 $\gamma = \frac{1}{2}$ 时，所对应的权重函数都是对称的。

表 3-2 Newmark 算法成员所对应的权重函数 $W(t)$

Table3-2 The weighted function $W(t)$ for some members of the Newmark family

γ	β	名称	$W(t)$
$-\frac{1}{2}$	0	--	
$\frac{1}{2}$	0	中心差分法	
$\frac{3}{2}$	1	向后微分公式	
$\frac{1}{2}$	$\frac{1}{12}$	Fox-Goodwin 格式	
$\frac{1}{2}$	$\frac{1}{6}$	线平均加速度法	
$\frac{1}{2}$	$\frac{1}{4}$	常平均加速度法/Trapeoidal 规则	
$\frac{1}{2}$	$\frac{1}{10}$	--	
$\frac{3}{2}$	$\frac{4}{5}$	Galerkin 方法	

当使用更多的时间节点时, 该有限元-加权残差方法可以推导出更高阶的算法格式。文 [66] 中给出了时间节点 $t - 2\Delta t, t - \Delta t, t, t + \Delta t$ 四个节点的使用三次多项式插值的离散格式。其算法格式为

$$\begin{aligned}
 & [M(\gamma - 1) + (\frac{1}{2}\beta - \gamma + \frac{1}{3})\Delta t C + (\frac{1}{6}\alpha - \frac{1}{2}\beta + \frac{1}{3}\gamma)\Delta t^2 K]^{t+\Delta t} U \\
 & + [(4 - 3\gamma)M + (4\gamma - \frac{3}{2}\beta - \frac{3}{2})\Delta t C + (2\beta - \frac{1}{2}\alpha - \frac{3}{2}\gamma)\Delta t^2 K]^t U \\
 & + [(3\gamma - 5)M + (\frac{3}{2} - 5\gamma + 3)\Delta t C + (\frac{1}{2}\alpha - \frac{5}{2}\beta + 3\gamma)\Delta t^2 K]^{t-\Delta t} U \\
 & + [(2 - \gamma)M + (2\gamma - \frac{1}{2}\beta - \frac{11}{6})\Delta t C + (\beta - \frac{1}{6}\alpha - \frac{11}{6}\gamma + 1)\Delta t^2 K]^{t-2\Delta t} U \\
 & + (\frac{1}{6}\alpha - \frac{1}{2}\beta + \frac{1}{3}\gamma)^{t+\Delta t} F \Delta t^2 + (2\beta - \frac{1}{2}\alpha - \frac{3}{2}\gamma)^t F \Delta t^2 \\
 & + (\frac{1}{2}\alpha - \frac{5}{2}\beta + 3\gamma)^{t-\Delta t} F \Delta t^2 + (\beta + 1 - \frac{1}{6}\alpha - \frac{11}{6}\gamma)^{t-2\Delta t} F \Delta t^2
 \end{aligned} \tag{3-75}$$

其中, 参数 α, β, γ 被定义如下

$$\begin{cases} \alpha = \frac{\int_0^3 W \xi^3 d\xi}{\int_0^3 W d\xi} \\ \beta = \frac{\int_0^3 W \xi^2 d\xi}{\int_0^3 W d\xi} \\ \gamma = \frac{\int_0^3 W \xi d\xi}{\int_0^3 W d\xi} \end{cases} \tag{3-76}$$

有意思的是, 当权重函数 $W(t)$ 使用在 $t + \Delta t$ 时刻的单位脉冲函数时, 有参数 $\alpha = 27, \beta = 9, \gamma = 3$, 可以得到 Houbolt 算法^[67]。而当参数满足如下关系时

$$\begin{cases} \alpha = 2 + 4\theta + 3\theta^2 + \theta^3 \\ \beta = \frac{4}{3} + 2\theta + \theta^2 \\ \gamma = 1 + \theta \end{cases} \tag{3-77}$$

算法格式 (3-75) 将退化为 Wilson- θ 法。

文 [66] 中也考虑了在分段线性时间元的离散, 而离散的形函数更加一般化, 亦即在时刻 $t + i\Delta t$ 有 $W_{t+i\Delta t} = 1$, 其他时刻均为零。其分段线性时间元和形函数表示如图3-2。

然后在一个更大的时间域 $[0, m\Delta t]$ 上进行加权残差积分后得到

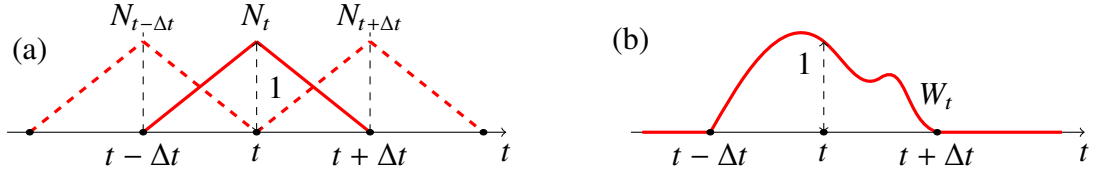


图 3-2 (a): $N_{t+\Delta t}, N_t, N_{t-\Delta t}$ 分段线性的形函数; (b): 一般性的加权函数 $W(t)$ 示意图
Fig.3-2 (a): The piecewise linear shape functions; (b): The general weighting function $W(t)$

$$\begin{aligned}
 & [M + \gamma_2 \Delta t C + \beta_2 \Delta t^2 K]^{t+\Delta t} U \\
 & + [-2M + (\gamma_1 - \gamma_2) \Delta t C + (\beta_1 - \beta_2 + \gamma_2) \Delta t^2 K]^{t} U \\
 & + [M - \gamma_1 \Delta t C + (\gamma_1 - \beta_1) \Delta t^2 K]^{t-\Delta t} U \\
 & + \beta_2^{t+\Delta t} F \Delta t^2 + (\beta_1 - \beta_2 + \gamma_2)^t F \Delta t^2 + (\gamma_1 - \beta_1)^{t-\Delta t} F \Delta t^2 = 0
 \end{aligned} \tag{3-78}$$

其中,

$$\begin{cases} \gamma_1 = \int_{n-1}^n W d\xi & \gamma_2 = \int_n^{n+1} W d\xi \\ \beta_1 = \int_{n-1}^n W \xi d\xi & \beta_2 = \int_n^{n+1} W \xi d\xi \end{cases} \tag{3-79}$$

这里的 $n-1, n, n+1$ 分别代表 $t - \Delta t, t, t + \Delta t$ 时刻, 后文也将继续这样进行混用而不再特别指明, 一般情况下是很容易分清的。当参数满足

$$\begin{aligned}
 \gamma_2 &= \gamma & \beta_2 &= \beta \\
 \gamma_1 &= 1 - \gamma & \beta_1 &= \frac{1}{2} - \beta
 \end{aligned} \tag{3-80}$$

算法格式 (3-78) 将退化为经典的 Newmark 算法 (3-70)。同时, 对算法 (3-78) 进行相容性分析将使得四个独立的参数退化为两个。

需要特别指明的是, 如果将上述有限元离散的加权残差方法应用到位移-速度场的一阶运动微分方程 (3-3), 则可以使用两个不同的加权函数 $W_1(t), W_2(t)$ 进而推导出新的算法格式, 如高阶导数的显式算法^[68,69]。

最后, 需要说明的是对于 Newmark 算法的位移和速度更新格式是现阶段直接积分法中使用最为广泛的更新格式。后续很多学者基于此提出了许多新的积分算法^[70-73]。

3.1.9 Wilson- θ 法

Wilson- θ 法^[54] 是第一个无条件稳定的带可控数值耗散的积分算法, 其本质上是平均加速度法的拓展。该算法假设在时间区间 $(t_n, t_n + \theta \Delta t)$ 内, 加速度线性变换,

即

$${}^{t+\tau}\ddot{U} = {}^t\ddot{U} + \frac{\tau}{\theta\Delta t}({}^{t+\theta}\ddot{U} - {}^t\ddot{U}) \quad (3-81)$$

其中, θ 为自由参数, 控制着算法的稳定性和精度, 并且 $0 \leq \tau \leq \theta\Delta t$ 。在区间内对等式 (3-81) 向变量 τ 积分两次得到

$${}^{t+\tau}\dot{U} = {}^t\dot{U} + {}^t\ddot{U}\tau + \frac{\tau^2}{2\theta\Delta t}({}^{t+\theta}\ddot{U} - {}^t\ddot{U}) \quad (3-82a)$$

$${}^{t+\tau}U = {}^tU + {}^t\dot{U}\tau + \frac{1}{2}{}^t\ddot{U}\tau^2 + \frac{1}{6\theta\Delta t}\tau^3({}^{t+\theta}\ddot{U} - {}^t\ddot{U}) \quad (3-82b)$$

其中, ${}^{t+\tau}U = U(t_n + \tau)$ 并且 ${}^{t+\theta}U = U(t_n + \theta\Delta t)$ 。在上两式中雇佣 $\tau = \theta\Delta t$ 并使用在 $t + \theta\Delta t$ 时刻的平衡运动方程可得 Wilson- θ 的算法格式如下

$$M{}^{t+\theta\Delta t}\ddot{U} + C{}^{t+\theta\Delta t}\dot{U} + K{}^{t+\theta\Delta t}U = {}^{t+\theta\Delta t}F \quad (3-83a)$$

$${}^{t+\theta\Delta t}\ddot{U} = \frac{6}{\theta^2\Delta t^2}({}^{t+\theta\Delta t}U - {}^tU) - \frac{6}{\theta}{}^t\dot{U} - 2{}^t\ddot{U} \quad (3-83b)$$

$${}^{t+\theta\Delta t}\dot{U} = \frac{3}{\theta\Delta t}({}^{t+\theta\Delta t}U - {}^tU) - 2{}^t\dot{U} - \frac{\theta\Delta t}{2}{}^t\ddot{U} \quad (3-83c)$$

其中, ${}^{t+\theta\Delta t}F = {}^tF + \theta({}^{t+\Delta t}F - {}^tF)$ 代表荷载的线性逼近。将 Wilson- θ 改写为线性多步法的形式即有

$$\begin{aligned} \left[\frac{6}{\theta^2\Delta t^2}M + \frac{3}{\theta\Delta t}C + K \right] {}^{t+\theta\Delta t}U &= {}^tF + \theta({}^{t+\Delta t}F - {}^tF) \\ &+ M \left[\frac{6}{\theta^2\Delta t^2}{}^tU + \frac{6}{\theta\Delta t}{}^t\dot{U} + 2{}^t\ddot{U} \right] + C \left[\frac{3}{\theta\Delta t}{}^tU + 2{}^t\dot{U} + \frac{\theta\Delta t}{2}{}^t\ddot{U} \right] \end{aligned} \quad (3-84)$$

在计算得到了 ${}^{t+\theta\Delta t}U$ 之后, 经过等式 (3-83) 就可以计算得到 ${}^{t+\theta\Delta t}\dot{U}$ 和 ${}^{t+\theta\Delta t}\ddot{U}$ 。进而由等式 (3-82) 在 $\tau = \Delta t$ 时计算得到在 $t + \Delta t$ 时刻的位移和速度。即

$${}^{t+\Delta t}\ddot{U} = \frac{6}{\theta^3\Delta t^2}({}^{t+\theta\Delta t}U - {}^tU) - \frac{6}{\theta^2\Delta t}{}^t\dot{U} + \left(1 - \frac{3}{\theta}\right){}^t\ddot{U} \quad (3-85a)$$

$${}^{t+\Delta t}\dot{U} = {}^t\dot{U} + \frac{\Delta t}{2}({}^{t+\Delta t}\ddot{U} + {}^t\ddot{U}) \quad (3-85b)$$

$${}^{t+\Delta t}U = {}^tU + {}^t\dot{U}\Delta t + \frac{\Delta t^2}{6}({}^{t+\Delta t}\ddot{U} + 2{}^t\ddot{U}) \quad (3-85c)$$

Wilson- θ 法是一个隐式算法。该算法是自启动的, 不需要引入额外的启动机制。同时当 $\theta = 1$ 时, 对应 Newmark 算法家族的线性加速度法; 当 $\theta \geq 1.37$ 时, 算法是无条件稳定的, 一般 θ 取值为 1.4。Wilson- θ 实现二阶精度, 且为带有可控的数值耗散, 其耗散能力由参数 θ 控制。但当 $\theta > 2$ 时, 算法将减小数值耗散, 同

时增加相对周期误差。特别需要提及的是 Wilson- θ 法很严重的超调行为^[74]，具体地，存在位移上的二阶超调和速度上的一阶超调行为。这也是该算法最大的一个缺点。有意思的是，文 [75] 中通过引入一个结构依赖性的参数 α 可以很好的处理 Wilson- θ 的超调行为，而且文 [76] 也分析了 Wilson- θ 法严重的超调行为是由于初值加速度的不协调引起的，如果引入协调的初始加速度，该算法也将没有超调行为。

3.1.10 Park 算法

Park 算法是一个线性多步算法。它是隐式的、无条件稳定的二阶精度算法。而且带有渐进消弭特性，因此是 L-稳定的。类似于 Houbolt 算法，Park 算法也是基于在 $t + \Delta t$ 的平衡运动方程和两个向后微分公式逼近速度和加速度得到。其算法格式描述如下

$$M^{t+\Delta t}\ddot{U} + C^{t+\Delta t}\dot{U} + K^{t+\Delta t}U = {}^{t+\Delta t}F \quad (3-86a)$$

$${}^{t+\Delta t}\dot{U} = \frac{1}{6\Delta t}(10{}^{t+\Delta t}U - 15{}^tU + 6{}^{t-\Delta t}U - {}^{t-2\Delta t}U) \quad (3-86b)$$

$${}^{t+\Delta t}\ddot{U} = \frac{1}{6\Delta t}(10{}^{t+\Delta t}\dot{U} - 15{}^t\dot{U} + 6{}^{t-\Delta t}\dot{U} - {}^{t-2\Delta t}\dot{U}) \quad (3-86c)$$

该算法是刚性稳定的。除此之外，Park 算法也可以通过 Gear 两步法^[19]和三步刚性稳定算法^[19]结合产生。因此该算法可以使用较大的时间步长求解刚性系统。它也拥有在高频段的强数值耗散，但在低频段其耗散较 Houbolt 算法低，同时也提高了相对周期误差。对于一阶微分系统 (3-10) 的线性多步法公式 (3-13)，Park 算法对应的参数取值为 ($k = 3$)

$$\begin{aligned} \alpha_0 &= 10, \alpha_1 = -15, \alpha_2 = 6, \alpha_3 = -1 \\ \beta_0 &= 6, \beta_1 = \beta_2 = \beta_3 = 0 \end{aligned} \quad (3-87)$$

3.1.11 Hilber-Hughes-Taylor- α 算法

Hilber-Hughes-Taylor- α 算法 (HHT- α)^[71] 是基于设计无条件稳定的带可控数值耗散的直接积分法。该算法在时间区间 $[t_n, t_{n+1}]$ 内使用 Newmark 格式的位移和速度的更新公式，也就是说 Newmark 格式是被使用作为一个基础和起始点。即：

$${}^{t+\Delta t}U = {}^tU + {}^t\dot{U}\Delta t + \frac{\Delta t^2}{2}[(1 - 2\beta){}^t\ddot{U} + 2\beta{}^{t+\Delta t}\ddot{U}] \quad (3-88a)$$

$${}^{t+\Delta t}\dot{U} = {}^t\dot{U} + \Delta t[(1 - \gamma){}^t\ddot{U} + \gamma{}^{t+\Delta t}\ddot{U}] \quad (3-88b)$$

并且离散的平衡方程被加权满足，即

$$M^{t+\Delta t}\ddot{U} + (1-\alpha)C^{t+\Delta t}\dot{U} + \alpha C^t\dot{U} + (1-\alpha)K^{t+\Delta t}U + \alpha K^tU = {}^{t+\alpha}F \quad (3-89)$$

对于外荷载的激励加权值 ${}^{t+\alpha}F$ 通常情况下可以选择为线性叠加，即

$${}^{t+\alpha}F = (1-\alpha){}^{t+\Delta t}F + \alpha{}^tF \quad (3-90)$$

但需要说明的是，文 [77] 中还分析了另外一种组合方式的影响。即

$${}^{t+\alpha}F = F((1-\alpha)t_{n+1} + \alpha t_n) \quad (3-91)$$

将位移和速度更新公式带入加权组合的离散运动方程，可得

$$\begin{aligned} & [M + \Delta t\gamma(1-\alpha)C + \Delta t^2\beta(1-\alpha)K] {}^{t+\Delta t}\ddot{U} \\ & = -(1-\alpha)C^{t+\Delta t}\tilde{\dot{U}} - \alpha C^t\dot{U} + (1-\alpha)K^{t+\Delta t}\tilde{U} - \alpha K^tU + {}^{t+\alpha}F \end{aligned} \quad (3-92)$$

其中， ${}^{t+\Delta t}\tilde{\dot{U}}$ 和 ${}^{t+\Delta t}\tilde{U}$ 来自于等式 (3-61)。于是，下一时刻的加速度可由求解上式得到，而下一时刻的位移和速度则有等式 (3-88) 计算得到。

HHT- α 算法是隐式的、二阶精度算法。当下列参数满足时，可实现无条件稳定性：

$$-\frac{1}{3} \leq \alpha \leq 0, \quad \beta = \frac{(1-\alpha)^2}{4} \quad \text{和} \quad \gamma = \frac{1}{2} - \alpha \quad (3-93)$$

其中，参数 α 控制算法的耗散特性。 α 取更大的负值将增加数值耗散和相对周期误差；相反，取更小的负值将会减小耗散和相对周期误差。无论情况下，其数值耗散和相对周期误差都比 Houbolt 算法和 Wilson- θ 算法低。同时，它并不是最优的耗散性算法。除此之外，它具有速度上的一阶超调行为。

3.1.12 Wood-Bosak-Zienkiewicz- α 算法

Wood-Bosak-Zienkiewicz- α 算法 (WBZ- α)^[72] 是基于 HHT- α 法的思想，修改了加权平衡方程的惯性力项，即保持 Newmar 的位移和速度更新公式不变，使用如下的加权平衡方程

$$(1-\alpha)M^{t+\Delta t}\ddot{U} + \alpha M^{t+\Delta t}\ddot{U} + C^{t+\Delta t}\dot{U} + K^{t+\Delta t}U = {}^{t+\Delta t}F \quad (3-94)$$

其中 α 为算法参数。仍将 Newmark 算法的位移和速度更新公式 (3-88) 带入上式得到

$$[(1-\alpha)M + \Delta t\gamma C + \Delta t^2\beta K] {}^{t+\Delta t}\ddot{U} = {}^{t+\Delta t}F - \alpha M^{t+\Delta t}\ddot{U} - C^{t+\Delta t}\tilde{\dot{U}} - K^{t+\Delta t}\tilde{U} \quad (3-95)$$

其中, ${}^{t+\Delta t}\tilde{U}$ 和 ${}^{t+\Delta t}\tilde{U}$ 来自于等式 (3-61)。当参数满足如下条件时, WBZ- α 算法实现二阶精度和无条件稳定性:

$$\alpha \leq 0, \quad \gamma = \frac{1}{2} - \alpha \quad \text{和} \quad \beta = \frac{1}{4}(1 - \alpha)^2 \quad (3-96)$$

WBZ- α 算法的数值特点类似于 HHT- α 算法。如, 不是最优的耗散性算法以及 WBZ- α 也具有速度上的一阶超调行为。

3.1.13 基于速度更新的 γ_s 格式

Namburu 和 Tamma 在 20 世纪 90 年代左右基于一般形式的守恒律微分方程的 Lax-Wendroff/Taylor-Galerkin 离散得到了一类速度更新的显式/隐式直接积分算法^[78-84]。其中, 具有一般表达性的是 γ_s 族算法^[84]

$$[M + \Delta t \gamma_1 C + \Delta t^2 \gamma_1 \gamma_2 K]{}^{t+\Delta t}\dot{U} = [M - \Delta t(1 - \gamma_1)C - \Delta t^2 \gamma_1(1 - \gamma_2)K]{}^t\dot{U} - \Delta t K^t U + \Delta t[(1 - \gamma_1)^t F + \gamma_1{}^{t+\Delta t} F] \quad (3-97)$$

$${}^{t+\Delta t}U = {}^tU + \Delta t[\gamma_3{}^{t+\Delta t}\dot{U} + (1 - \gamma_3){}^t\dot{U}] \quad (3-98)$$

对于参数选取条件, 在无阻尼情况下 $\xi = 0$ 时,

$$\gamma_1 + \gamma_3 = 1 \quad 1 - 2\gamma_1 = 0 \quad (3-99)$$

可保证实现二阶精度。在有阻尼时, 仅需要 $\gamma_1 + \gamma_3 = 1$ 即可。

同时, 对于具体的参数取值, 原文 [84] 给出如下情况 ($0 \leq \xi \leq 1$)

‡ 当 $\gamma_1 = \gamma_2 = \gamma_3 = \frac{1}{2}$ 时, 可以实现无条件稳定的二阶隐式算法^[82]。

‡ 当 $\gamma_1 = \gamma_3 = \frac{1}{2}, \gamma_2 < 0.5$ 时, 可以实现条件稳定的二阶隐式算法。

‡ 当 $\gamma_1 = \gamma_3 = \frac{1}{2}, \gamma_2 = 0$ 时, 可以实现条件稳定的二阶显式算法^[78-81,83], 其稳定域为 $\Omega_{cr} = 2$ 。

对于无阻尼的线性守恒系统, 上述算法是能量、动量以及角动量守恒的。该格式的原始性工作由 Simo 和 Tarnow 针对非线性系统在位移-速度场中提出^[85]。

非线性的应用上述格式在求解非线性动力学方程时, 可采用如下更具一般性的算法格式^[86]

$$M{}^{t+\Delta t}\Delta\dot{U} = \Delta t({}^{t+\Delta t/2}F_{int} + {}^{t+\Delta t/2}F_{ext}) \quad (3-100)$$

而位移更新公式则仍使用公式 (3-98)。实现上, 上述方程的物理含义就是: 系统动量的增量 ($M\Delta\dot{U}$) 等于系统所受到外力的冲量 (和 $\Delta t F_{\text{合}}$)。其中, F_{int} 和 F_{ext} 分别为

系统的内力和外力。对于线性系统，二者可分别看作 $-C\dot{U}(t) - KU(t)$ 和 $F(t)$ 。而上标 $t + \Delta t/2$ 表示在 $t + \Delta t/2$ 时刻进行计算。因此对于不同的内力和外力在 $t + \Delta t/2$ 时刻的逼近表达式可以导出不同的积分格式。如采用下列等式计算

$${}^{t+\Delta t/2}F_{int} = \frac{1}{2}[(-C^t\dot{U} - K^tU) + (-C^{t+\Delta t}\dot{U} - K^{t+\Delta t}U)] \quad (3-101)$$

$${}^{t+\Delta t/2}F_{ext} = \frac{1}{2}({}^tF + {}^{t+\Delta t}F) \quad (3-102)$$

于是，算法格式 (3-100) 将导出 $\gamma_1 = \gamma_2 = \gamma_3 = \frac{1}{2}$ 的算法^[82]。

该算法的一个主要特点是真正意义上的自启动算法，并不需要任何形式的初始加速度启动机制。特别地，当上述参数依赖于结构的特性时，将导出一类依赖于结构的直接积分法。如后面的 Soares-Jr 型算法族。

3.1.14 带中点加速度的中点公式

对于如下形式的二阶精度的隐式算法^[64](MPR-MPA)

$$\left[M + \frac{1}{2}C\Delta t + \frac{1}{4}K\Delta t^2\right] {}^{t+\Delta t}\ddot{U} = -C^t\dot{U} - K\left[{}^tU + \frac{\Delta t}{2}{}^t\dot{U}\right] + \frac{1}{2}({}^tF + {}^{t+\Delta t}F) \quad (3-103a)$$

$${}^{t+\Delta t}U = {}^tU + {}^t\dot{U}\Delta t + \frac{\Delta t^2}{2}{}^{t+\Delta t}\ddot{U} \quad (3-103b)$$

$${}^{t+\Delta t}\dot{U} = {}^t\dot{U} + \frac{\Delta t}{2}({}^t\ddot{U} + {}^{t+\Delta t}\ddot{U}) \quad (3-103c)$$

上式公式的加速度在时刻 $t_{n+\frac{1}{2}}$ 严格成立。需要说明的是，在大多数非耗散性算法中，该算法的鲁棒性较好，而且在线性多步法的框架下是最佳选择。同时在位移和速度上都没有超调行为。

3.1.15 三参数优化算法/CH- α 法

在结构动力学数值响应求解算法中，使用最为广泛、研究最为深入的另外一个算法就是 Chung 和 Hulbert 提出的广义 α 法 (The generalized- α method)^[87]。特别地，为了与后续更加一般化的 α 方法^[73] 区别开来，本文都将该算法注释为：CH- α 算法。

事实上，在 Chung 和 Hulbert 等人提出 CH- α 算法之前，国内的邵慧萍等人就已经提出了该算法，其算法命令为三参数优化算法^[88]。其算法格式结合 HHT- α 算法和 WBZ- α 算法的特点，定义了如下的算法格式：

$$\begin{aligned} (1 - \xi){}^{t+\Delta t}F + \xi{}^tF = M[(1 - \alpha){}^{t+\Delta t}\ddot{U} + \alpha{}^t\ddot{U}] + C[(1 - \eta){}^{t+\Delta t}\dot{U} + \eta{}^t\dot{U}] \\ + K[(1 - \delta){}^{t+\Delta t}U + \delta{}^tU] \end{aligned} \quad (3-104a)$$

$${}^{t+\Delta t}U = {}^tU + {}^t\dot{U}\Delta t + \frac{\Delta t^2}{2}[(1-2\beta){}^t\ddot{U} + 2\beta{}^{t+\Delta t}\ddot{U}] \quad (3-104b)$$

$${}^{t+\Delta t}\dot{U} = {}^t\dot{U} + \Delta t[(1-\gamma){}^t\ddot{U} + \gamma{}^{t+\Delta t}\ddot{U}] \quad (3-104c)$$

显然，该三参数优化算法仍然雇用了 Newmark 算法的位移和速度更新公式。当参数取不同的值，将退化为前述提及的一些算法，如

‡ 当 $\alpha = \delta = \eta = \xi = 0$ 时，该算法退化为 Newmark 家族算法。

‡ 当 $\alpha = 0$ 时，该算法退化为 HHT- α 算法。

‡ 当 $\delta = \eta = \xi = 0$ 时，该算法退化为 WBZ- α 算法。

事实上，上述涉及的六个参数中，只有三个参数被选择以实现二阶精度和无条件稳定性。即

$$\begin{cases} \alpha \leq \delta \leq \frac{1}{2} & \beta \geq \frac{1}{2}\gamma = \frac{1}{2}(\frac{1}{2} - \alpha + \delta) \\ \eta = \xi = \delta & \xi \geq 0 \end{cases} \quad (3-105)$$

如果将这些参数与高频极限下的谱半径 ρ_∞ 相联系，则可以定义如下的单参数簇算法

$$\begin{cases} \alpha = \frac{2\rho_\infty - 1}{\rho_\infty + 1} \\ \beta = \frac{1}{(\rho_\infty + 1)^2} \\ \gamma = \frac{3 - \rho_\infty}{2(1 + \rho_\infty)} \\ \eta = \xi = \delta = \frac{\rho_\infty}{\rho_\infty + 1} \end{cases} \quad (3-106)$$

3.1.15.1 最优的 χ 格式

邵慧萍在文 [88] 中给出了一组最优的 χ 算法格式，其参数取值如下

$$\begin{cases} \alpha = -\frac{2\rho_\infty - 1}{\rho_\infty + 1} & \beta = -\frac{1}{(\rho_\infty + 1)^2} \\ \gamma = -\frac{3 - \rho_\infty}{2(1 + \rho_\infty)} & \eta = \xi = \delta = -\frac{\rho_\infty}{\rho_\infty + 1} \end{cases} \quad (3-107)$$

3.1.15.2 HHT- α 算法

当上述算法被定义如下时，将退化为 HHT- α 算法。

$$\begin{cases} \alpha = 0 & \beta = \frac{1}{(\rho_\infty + 1)^2} \\ \gamma = \frac{3 - \rho_\infty}{2(1 + \rho_\infty)} & \eta = \xi = \delta = \frac{\rho_\infty}{\rho_\infty + 1} \end{cases} \quad (3-108)$$

3.1.15.3 WBZ- α 算法

当上述算法被定义如下时，将退化为 WBZ- α 算法。

$$\begin{cases} \alpha = \frac{\rho_\infty - 1}{\rho_\infty + 1} & \beta = -\frac{1}{(\rho_\infty + 1)^2} \\ \gamma = -\frac{3 - \rho_\infty}{2(1 + \rho_\infty)} & \eta = \xi = \delta = 0 \end{cases} \quad (3-109)$$

3.1.15.4 CH- α 算法

Chung 和 Hulbert 于 1993 年在应用力学杂志发表了 CH- α 算法^[87]。该算法实质上与前述的三参数优化算法一致。其算法的平衡方程加权组合如下

$$M^{n+(1-\alpha_m)}\ddot{U} + C^{n+(1-\alpha_f)}\dot{U} + K^{n+(1-\alpha_f)}U = {}^{n+(1-\alpha_f)}F \quad (3-110)$$

其中

$${}^{n+(1-\alpha_m)}\ddot{U} = (1 - \alpha_m)^{t+\Delta t}\ddot{U} + \alpha_m {}^t\ddot{U} \quad (3-111a)$$

$${}^{n+(1-\alpha_f)}\dot{U} = (1 - \alpha_f)^{t+\Delta t}\dot{U} + \alpha_f {}^t\dot{U} \quad (3-111b)$$

$${}^{n+(1-\alpha_f)}U = (1 - \alpha_f)^{t+\Delta t}U + \alpha_f {}^tU \quad (3-111c)$$

对于外力荷载的加权组合 ${}^{n+(1-\alpha_f)}F$ 类似于三参数优化算法，有两种组合模式。其位移和速度更新公式使用 Newmark 算法的位移和速度更新公式 (3-88)。于前者一致的是，参数 $\alpha_m = 0$ 和 $\alpha_f = 0$ 分别对应着 HHT- α 和 WBZ- α 算法；而 $\alpha_m = \alpha_f = 0$ 对应于原始的 Newmark 家族算法。

当满足下列条件时，CH- α 算法实现二阶精度

$$\gamma = \frac{1}{2} - \alpha_m + \alpha_f \quad (3-112)$$

而无条件稳定性要求

$$\alpha_m \leq \alpha_f \leq \frac{1}{2} \quad \text{和} \quad \beta \geq \frac{1}{4} + \frac{1}{2}(\alpha_f - \alpha_m) \quad (3-113)$$

为了实现最优的数值耗散，算法在高频极限情况下的三个特征值 ($\lambda_{1,2,3}^\infty$) 应满足下列条件

$$\text{Im}(\lambda_{1,2}^\infty) = 0 \quad |\lambda_3^\infty| = |\lambda_{1,2}^\infty| \quad (3-114)$$

其中 Im 表示取复数的虚部。于是，参数 β 应满足

$$\beta = \frac{1}{4}(1 - \alpha_m + \alpha_f)^2 \quad (3-115)$$

同样的，如果以高频极限下的谱半径值 ρ_∞ 表示算法的耗散特性，则 CH- α 算法可以表示为

$$\alpha_m = \frac{2\rho_\infty - 1}{\rho_\infty + 1} \quad \alpha_f = \frac{\rho_\infty}{\rho_\infty + 1} \quad (3-116)$$

当 $\rho_\infty = 1$ 时，CH- α 算法退化为非耗散性的带终点加速度的中点公式 (MPR-EPA); 而 $\rho_\infty = 0$ 退化为带有渐进消弭特性的 WBZ- α 算法。特别地，CH- α 算法被证明是数值耗散特性最优的算法，同时也是存在虚假根的算法中，虚假根影响数值解最小的算法^[89]。

3.1.16 广义 α 算法

尽管三参数优化算法/CH- α 算法具有很好的耗散特性，但是却存在速度上的一阶超调行为。因此，基于加权组合平衡方程引入数值耗散的思想，于开平教授于 2008 年提出了更具一般性的广义 α 算法^[73]。该算法不仅使得前述的 WBZ- α 、HHT- α 以及 CH- α 算法都是该算法的一个特例，而且还分别提出了三个算法对应的无超调行为的新算法。其广义 α 算法的算法描述如下

$$\begin{aligned} (1 - \alpha)M^{t+\Delta t}\ddot{U} + \alpha M^t\ddot{U} + (1 - \delta)C^{t+\Delta t}\dot{U} + \delta C^t\dot{U} + (1 - \eta)K^{t+\Delta t}U + \eta K^tU \\ = (1 - \eta)^{t+\Delta t}F + \eta^tF \end{aligned} \quad (3-117a)$$

$${}^{t+\Delta t}U = {}^tU + {}^t\dot{U}\Delta t + \Delta t^2(\epsilon^t\ddot{U} + \beta^{t+\Delta t}\ddot{U}) \quad (3-117b)$$

$${}^{t+\Delta t}\dot{U} = {}^t\dot{U} + \Delta t(\mu^t\ddot{U} + \gamma^{t+\Delta t}\ddot{U}) \quad (3-117c)$$

为了实现二阶精度，参数需要满足条件

$$\gamma + \mu = 1 \quad \alpha + \beta + \epsilon - \eta - \mu = 0 \quad \beta + \epsilon + \delta - \eta = \frac{1}{2} \quad (3-118)$$

同时，无条件稳定性要求

$$\gamma \geq \frac{1}{2} \geq \eta \geq \delta \quad \beta \geq \frac{1}{2}(\eta + \gamma - \delta) \quad (3-119)$$

当分别满足以下三个条件时，可设计对应的三个算法

‡ $\text{Im}(\lambda_{1,2}^\infty) = 0$ 并且 $|\lambda_3^\infty| = |\lambda_{1,2}^\infty|$ 时，可以得到最优的数值耗散性算法。

‡ 算法在低频段的数值耗散尽可能小，即当 $\omega\Delta t \rightarrow 0$ 时，谱半径 $\rho \rightarrow 0$ 。此

时，可设计出类似于 HHT- α 算法。

‡ 在高频极限下，其虚假根为 0，即 $\lambda_3^\infty \rightarrow 0$ 。此时，可设计出类似于 WBZ- α 算法。

特别地，参数 μ 满足

$$\mu = \frac{\rho_\infty}{\rho_\infty + 1} \quad (3-120)$$

时，广义 α 算法没有超调行为。其广义 α 算法对应的不同算法成员如下表3-3所示

表 3-3 广义 α 积分算法成员的参数取值
Table3-3 Algorithmic parameters of the generalized- α method family

参数	NOCH- α	CH- α	NOHHT- α	HHT- α	NOWBZ- α	WBZ- α
α	$\frac{2\rho_\infty-1}{\rho_\infty+1}$	$\frac{2\rho_\infty-1}{\rho_\infty+1}$	0	0	$\frac{\rho_\infty-1}{\rho_\infty+1}$	$\frac{\rho_\infty-1}{\rho_\infty+1}$
δ	$\frac{3\rho_\infty-1}{2(\rho_\infty+1)}$	$\frac{\rho_\infty}{\rho_\infty+1}$	$\frac{1-\rho_\infty}{2(\rho_\infty+1)}$	$\frac{1-\rho_\infty}{\rho_\infty+1}$	$\frac{\rho_\infty-1}{2(\rho_\infty+1)}$	0
η	$\frac{\rho_\infty}{\rho_\infty+1}$	$\frac{\rho_\infty}{\rho_\infty+1}$	$\frac{1-\rho_\infty}{(\rho_\infty+1)}$	$\frac{1-\rho_\infty}{(\rho_\infty+1)}$	0	0
ϵ	$\frac{\rho_\infty}{(\rho_\infty+1)^2}$	$\frac{\rho_\infty^2+2\rho_\infty-1}{2(\rho_\infty+1)^2}$	$\frac{\rho_\infty}{(\rho_\infty+1)^2}$	$\frac{\rho_\infty^2+2\rho_\infty-1}{2(\rho_\infty+1)^2}$	$\frac{\rho_\infty}{(\rho_\infty+1)^2}$	$\frac{\rho_\infty^2+2\rho_\infty-1}{2(\rho_\infty+1)^2}$
β	$\frac{1}{(\rho_\infty+1)^2}$	$\frac{1}{(\rho_\infty+1)^2}$	$\frac{1}{(\rho_\infty+1)^2}$	$\frac{1}{(\rho_\infty+1)^2}$	$\frac{1}{(\rho_\infty+1)^2}$	$\frac{1}{(\rho_\infty+1)^2}$
μ	$\frac{\rho_\infty}{\rho_\infty+1}$	$\frac{3\rho_\infty-1}{2(\rho_\infty+1)}$	$\frac{\rho_\infty}{\rho_\infty+1}$	$\frac{3\rho_\infty-1}{2(\rho_\infty+1)}$	$\frac{\rho_\infty}{\rho_\infty+1}$	$\frac{3\rho_\infty-1}{2(\rho_\infty+1)}$
γ	$\frac{1}{\rho_\infty+1}$	$\frac{3-\rho_\infty}{2(\rho_\infty+1)}$	$\frac{1}{\rho_\infty+1}$	$\frac{3-\rho_\infty}{2(\rho_\infty+1)}$	$\frac{1}{\rho_\infty+1}$	$\frac{3-\rho_\infty}{2(\rho_\infty+1)}$
ρ_∞	[0, 1]	[0, 1]	[0.5, 1]	[0.5, 1]	[0, 1]	[0, 1]

注意到 NOCH- α 、NOHHT- α 和 NOWBZ- α 是三个新的广义 α 成员，其前缀“NO”表示在速度和位移上都没有超调行为。而且这三个算法满足 Hughes 所提出的一个具有竞争性算法的六个属性^[5,70]。

3.2 加权残值法

3.2.1 ρ -算法

ρ -算法^[90] 是通过假设在时间区间 $[t_n, t_{n+1}]$ 内的位移满足

$$U(t) = N_1(t)^t U + N_2(t)^t \dot{U} + N_3(t)^t \ddot{U} + N_4(t)^{t+\Delta t} U \quad (3-121)$$

其中， $N_i(t), i = 1, 2, 3, 4$ 是时间域上的形函数，亦即表明上式实质上是进行了时间有限元的离散。同时，通过引入权重函数 $W(t) \equiv 1$ 在区间内进行残差积分，并限制其值为零。即

$$\int_{t_n}^{t_{n+1}} [M\ddot{U}(t) + C\dot{U}(t) + KU(t) - F(t)] = 0 \quad (3-122)$$

经过复杂的计算^①，就可以得到如下的位移、速度和加速度的更新公式

$${}^{t+\Delta t}U = {}^tU + \Delta U \quad (3-123a)$$

$${}^{t+\Delta t}\dot{U} = -{}^t\dot{U} + \frac{1 - \rho_\infty}{1 + \rho_\infty} {}^t\ddot{U}\Delta t + \frac{2}{\Delta t}\Delta U \quad (3-123b)$$

$${}^{t+\Delta t}\ddot{U} = \frac{(1 + \rho_\infty)(1 + \rho_\infty^2)}{\Delta t^2}(\Delta U - {}^t\dot{U}\Delta t) + (1 - \rho_\infty - \rho_\infty^3){}^t\ddot{U} \quad (3-123c)$$

其位移的增量 ΔU 的计算公式为

$$\left[M + \frac{\Delta t}{2}C + \frac{\Delta t^2}{2(\rho_\infty + 1)}K\right]\Delta U = -\frac{1}{2}K{}^tU\Delta t^2 + M{}^t\dot{U}\Delta t + \frac{\rho_\infty - 1}{2(\rho_\infty + 1)}M{}^t\ddot{U}\Delta t^2 + \frac{\Delta t^2}{2}{}^tF^* \quad (3-124)$$

其中，

$${}^tF^* = \frac{1}{\Delta t} \int_{t_n}^{t_{n+1}} F(t)dt \quad (3-125)$$

表示在时间区间内的外力平均值。其参数 ρ_∞ 实质上表示算法在高频极限下的谱半径值，亦即表示算法的耗散能力大小。当 $\rho_\infty = 1$ 时， ρ -算法退化为 Trapezoidal 规则。

特别需要说明的是， ρ -算法在无阻尼情况下实现二阶精度；而在有阻尼情况仅有一阶精度，这也是它的缺点。同时，文中 [90] 陈述该 ρ -算法满足 Hughes 所提的六个准则^[5,70]，但是经过简单的计算，可以发现该算法是存在速度上的超调行为。除此之外， ρ -算法另外一个重大的缺陷是其糟糕的虚假根；其虚假根的模值异常大，导致其算法精度降低^[52,89]。此时，针对异常的虚假根存在，常用的数值阻尼和相对周期误差来测量算法的精度将不再适用^[51]。

3.2.1.1 ρ 算法的显式格式

文 [90] 中也给出了 ρ -算法的显式格式，其算法描述为

$$\left[M + \frac{\Delta t}{2}C\right]\Delta U = M{}^t\dot{U}\Delta t - \frac{\Delta t^2}{2}K\left({}^tU + \frac{\Delta t}{2}{}^t\dot{U}\right) + \frac{\Delta t^2}{2}{}^tF^* \quad (3-126a)$$

$${}^{t+\Delta t}U = {}^tU + \Delta U \quad (3-126b)$$

$${}^{t+\Delta t}\dot{U} = \frac{2}{\Delta t}\Delta U - {}^t\dot{U} \quad (3-126c)$$

其中， ${}^tF^*$ 来源于等式 (3-125)。该显式算法的特点是

^① 事实上，原文作者也是空降到下面的计算格式，而且关于其增量的计算格式根据后文的精度分析反推可知有误，在此更正。

‡ 相容且实现二阶精度。

‡ 条件稳定，且其稳定域满足 $\Omega_{cr} \leq 2\sqrt{1-\xi^2}$ 。当 $\xi = 0$ 时，其稳定域大小与中心差分法一致。

‡ 无数值耗散，且引入负相对周期误差。

‡ 真正意义上的自启动，不需要引入额外的启动机制，包括初始加速度的计算。

3.2.2 单步隐式的 θ_1 格式

单步隐式的 θ_1 算法^[51] 是利用加权残值的思想推导得到。通过假定位移、速度和加速度在时间区间 $[t, t + \Delta t]$ 内的变化关系，带入其平衡方程得到其残差值，最后通过利用任意的权重函数限定在时间区间 $[0, \Delta t]$ 内的积分为零得到其加速度增量 $\Delta\ddot{U}$ 的关系式，如下

$$\begin{aligned} (\theta_1\eta M + \theta_2\gamma\Delta t C + \theta_3\beta\Delta t^2 K)\Delta\ddot{U} = (1 - \theta_0)^t F + \theta_0^{t+\Delta t} F \\ - M^t\ddot{U} - C(\dot{U} + \theta_1\Delta t\ddot{U}) - K(U + \theta_1\Delta t\dot{U} + \frac{1}{2}\theta_2\Delta t^2\ddot{U}) \end{aligned} \quad (3-127)$$

其位移、速度和加速度的更新公式为

$${}^{t+\Delta t}\ddot{U} = {}^t\ddot{U} + \Delta\ddot{U} \quad (3-128)$$

$${}^{t+\Delta t}\dot{U} = {}^t\dot{U} + \Delta t\ddot{U} + \gamma\Delta t\Delta\ddot{U} \quad (3-129)$$

$${}^{t+\Delta t}U = {}^tU + \Delta t\dot{U} + \frac{1}{2}\Delta t^2\ddot{U} + \beta\Delta t^2\Delta\ddot{U} \quad (3-130)$$

其中包含 7 个算法参数。该算法的特是

‡ 使用了带任意权重函数的子空间配点法推导算法。

‡ 荷载加权组合参数 θ_0 是独立于算法的其他参数值的选取。

值得一提的是，当参数选择不同值，单步隐式的 θ_1 算法将退化为前述的一些积分算法。具体情况见表3-4。

无论阻尼存在与否时，当下列条件被满足时，该算法实现二阶精度

$$\gamma = \theta_1(\eta - 1) + \frac{1}{2} \quad (3-131)$$

类似地，Hoff 在文 [51] 中要求，当 $\omega\Delta t \rightarrow 0$ 时，虚假根 $\lambda_3 \rightarrow 0$ ，即

$$\theta_1\eta = 1 \quad (3-132)$$

而 $\theta_2 = \theta_3$ 则使得算法没有超调行为。同时，当

表 3-4 单步隐式的 θ_1 算法的算法成员
 Table 3-4 Some members of the one-step implicit θ_1 algorithms

算法	θ_1	θ_2	θ_3	β	γ	η
Newmark 算法	1	1	1	β	γ	1
Trapeoidal 规则/Newmark 算法 ($\beta = \frac{1}{2}, \gamma = \frac{1}{2}$)	1	1	1	$\frac{1}{4}$	$\frac{1}{2}$	1
Wilson- θ 算法	θ	θ^2	θ^3	$\frac{1}{6}$	$\frac{1}{2}$	1
Collocation 格式 ^[70]	θ	θ^2	θ^3	β	$\frac{1}{2}$	1
WBZ- α 算法	1	1	1	$\frac{(1-\alpha)^2}{4}$	$\frac{1}{2} - \alpha$	$1 - \alpha$
SSpj 算法家族 ^[91,92]	θ_1	θ_2	θ_3	$\frac{1}{6}$	$\frac{1}{2}$	1

$$\theta_2 = 1 \quad \beta = \frac{1}{4\theta_1^2} \quad (3-133)$$

时, 单步隐式的 θ_1 算法在谱半径、数值耗散以及相对周期误差上都拥有最好的结果。此时, 算法的数值格式可重新写为

$$\left[M + \left(\frac{3}{2} - \theta_1 \right) \Delta t C + \frac{1}{4\theta_1^2} \Delta t^2 K \right] \Delta \ddot{U} = (1 - \theta_0) {}^t F + \theta_0 {}^{t+\Delta t} F \quad (3-134a)$$

$$- M {}^t \ddot{U} - C({}^t \dot{U} + \theta_1 \Delta t {}^t \ddot{U}) - K({}^t U + \theta_1 \Delta t {}^t \dot{U} + \frac{1}{2} {}^t \ddot{U} \Delta t^2)$$

$${}^{t+\Delta t} \ddot{U} = {}^t \ddot{U} + \Delta \ddot{U} \quad (3-134b)$$

$${}^{t+\Delta t} \dot{U} = {}^t \dot{U} + \Delta t {}^t \ddot{U} + \left(\frac{3}{2} - \theta_1 \right) \Delta t \Delta \ddot{U} \quad (3-134c)$$

$${}^{t+\Delta t} U = {}^t U + \Delta t {}^t \dot{U} + \frac{1}{2} \Delta t^2 {}^t \ddot{U} + \frac{1}{4\theta_1^2} \Delta t^2 \Delta \ddot{U} \quad (3-134d)$$

并且参数 θ_1 满足 $0.95 \leq \theta_1 \leq 1.0$ 。实现上, 对于无条件稳定域要求, 其参数 θ_1 取值可为 $0.5 \leq \theta_1 \leq 1.0$, 但只有前者可以达到最优的数值耗散性能。当 $\theta_1 = 1$ 时, 单步隐式的 θ_1 算法将退化为 Trapeoidal 规则。

单步隐式的 θ_1 算法的特点总结如下

- ‡ 实现无条件稳定的二阶精度。
- ‡ 在低频段, 没有虚假根, 高频段具有可控的数值阻尼。
- ‡ 谱半径曲线没有分岔点出现。
- ‡ 没有超调行为。

事实上, 该算法可以看作是隐式的预测校正格式 (为了求解 ${}^{t+\Delta t} \ddot{U}$ 的速度和位移与最终在 $t + \Delta t$ 时刻的位移和速度表达式不一致)。即

$$M^{t+\Delta t}\ddot{U} + C^{t+\Delta t}\dot{\widetilde{U}} + K^{t+\Delta t}\widetilde{U} = {}^{t+\theta_0}F \quad (3-135a)$$

$${}^{t+\Delta t}\dot{\widetilde{U}} = {}^t\dot{U} + (2\theta_1 - \frac{3}{2}){}^t\ddot{U}\Delta t + (\frac{3}{2} - \theta_1){}^{t+\Delta t}\ddot{U}\Delta t \quad (3-135b)$$

$${}^{t+\Delta t}\widetilde{U} = {}^tU + \theta_1{}^t\dot{U}\Delta t + \frac{1}{4}(2 - \frac{1}{\theta_1^2}){}^t\ddot{U}\Delta t^2 + \frac{1}{4\theta_1^2}{}^{t+\Delta t}\ddot{U}\Delta t^2 \quad (3-135c)$$

可以看出 ${}^{t+\Delta t}\dot{\widetilde{U}}$ 和 ${}^{t+\Delta t}\widetilde{U}$ 是为了求解 ${}^{t+\Delta t}\ddot{U}$ 而存在的, 其真正的位移和速度用使用下式求解:

$${}^{t+\Delta t}\dot{U} = {}^t\dot{U} + (\theta_1 - \frac{1}{2}){}^t\ddot{U}\Delta t + (\frac{3}{2} - \theta_1){}^{t+\Delta t}\ddot{U}\Delta t \quad (3-136a)$$

$${}^{t+\Delta t}U = {}^tU + {}^t\dot{U}\Delta t + (\frac{1}{2} - \frac{1}{4\theta_1^2}){}^t\ddot{U}\Delta t^2 + \frac{1}{4\theta_1^2}{}^{t+\Delta t}\ddot{U}\Delta t^2 \quad (3-136b)$$

关于单步隐式的 θ_1 算法在非线性系统中的应用可以参考文献 [52]。

3.2.3 高阶导数的单步显式算法

当单位移场的二阶运动微分方程 (3-1) 转换为位移-速度场的一阶微分方程 (3-3) 后, 再使用加权残差法可以得到新的算法格式。其中高阶导数的单步显式算法^[68,69]就是其中之一。其算法格式推导概括如下。

首先, 在时间区间 $[t_n, t_{n+1}]$ 内引入变量 $\tau = t - t_n$, 再进行位移和速度泰勒级数展开得到

$$\begin{cases} U(\tau) = {}^tU + {}^t\dot{U}\tau + \beta_2{}^t\ddot{U}\tau^2 + \beta_3\frac{\Delta\ddot{U}}{\Delta t}\tau^3 \\ \dot{U}(\tau) = {}^t\dot{U} + {}^t\ddot{U}\tau + \gamma_2\frac{\Delta\ddot{U}}{\Delta t}\tau^2 \\ \ddot{U}(\tau) = {}^t\ddot{U} + \frac{\Delta\ddot{U}}{\Delta t}\tau \end{cases} \quad (3-137)$$

$$\begin{cases} V(\tau) = {}^tV + {}^t\dot{V}\tau + \eta_2{}^t\ddot{V}\tau^2 + \eta_3\frac{\Delta\ddot{V}}{\Delta t}\tau^3 \\ \dot{V}(\tau) = {}^t\dot{V} + {}^t\ddot{V}\tau + \kappa_2\frac{\Delta\ddot{V}}{\Delta t}\tau^2 \\ \ddot{V}(\tau) = {}^t\ddot{V} + \frac{\Delta\ddot{V}}{\Delta t}\tau \end{cases} \quad (3-138)$$

其中, 荷载也进行泰勒级数展开得到

$$F(\tau) \triangleq p(\tau) = {}^tp + {}^t\dot{p}\tau + \frac{1}{2}{}^t\ddot{p}\tau^2 + \frac{1}{6}\frac{\Delta\ddot{p}}{\Delta t}\tau^3 \quad (3-139)$$

然后, 选择两个独立的加权函数 $w_1(\tau), W_2(\tau)$ 进行加权残差积分得到

$$\frac{1}{\Delta t} \int_0^{\Delta t} [M\dot{V}(\tau) + CV(\tau) + KU(\tau) - p(\tau)]w_1(\tau)d\tau = 0 \quad (3-140)$$

$$\frac{1}{\Delta t} \int_0^{\Delta t} [\dot{U}(\tau) - V(\tau)]w_2(\tau)d\tau = 0 \quad (3-141)$$

这里，并没有显式的给出加权函数的表达式，而是将其列为算法的积分参数进行后续的分析 (这个技巧在其他的加权残差方法用也会用到)。

$$\begin{aligned} \frac{1}{\Delta t} \int_0^{\Delta t} w_j(\tau)d\tau &= 1 \quad j = 1, 2 \\ \frac{1}{\Delta t} \int_0^{\Delta t} w_1(\tau)\tau^i d\tau &= \Psi_i \Delta t^i \quad i \geq 1 \\ \frac{1}{\Delta t} \int_0^{\Delta t} w_2(\tau)\tau^i d\tau &= \Pi_i \Delta t^i \quad i \geq 1 \end{aligned} \quad (3-142)$$

于是，对上述的两个残差函数进行积分计算得到如下的两个表达式

$$\begin{aligned} (\kappa_2 \Psi_2 \Delta t M + \eta_3 \Psi_3 \Delta t^2 C) \Delta \ddot{V} &= {}^t p + \Psi_1 {}^t \dot{p} \Delta t + \frac{1}{2} \Psi_2 {}^t \ddot{p} \Delta t^2 + \frac{1}{6} \Psi_3 \Delta {}^t \ddot{p} \Delta t^2 \\ &\quad - M({}^t \dot{V} + \Psi_1 {}^t \ddot{V} \Delta t) - C({}^t V + \Psi_1 {}^t \dot{V} \Delta t + \eta_2 \Psi_2 {}^t \ddot{V} \Delta t^2) \\ &\quad - K({}^t U + \Psi_1 {}^t \dot{U} \Delta t + \beta_2 \Psi_2 {}^t \ddot{U} \Delta t^2 + \beta_3 \Psi_3 \Delta {}^t \ddot{U} \Delta t^2) \end{aligned} \quad (3-143)$$

$$\gamma_2 \Pi_2 \Delta \ddot{U} \Delta t = {}^t V - {}^t \dot{U} + \Pi_1 {}^t \dot{V} \Delta t - \Pi_1 {}^t \ddot{U} \Delta t + \eta_2 \Pi_2 {}^t \ddot{V} \Delta t^2 + \eta_3 \Pi_3 \Delta {}^t \ddot{V} \Delta t^2 \quad (3-144)$$

在 $t + \Delta t$ 时刻的响应值就由上述两式求得，而主要的未知量 $\Delta \ddot{V}$ 和 $\Delta \ddot{U}$ 可以通过上述公式计算得到。

最后，令 $\Psi_3 = 0$ 就可以获得显式算法如下

$$\begin{aligned} \kappa_2 \Psi_2 M \Delta \ddot{V} \Delta t &= {}^t p + \Psi_1 {}^t \dot{p} \Delta t + \frac{1}{2} \Psi_2 {}^t \ddot{p} \Delta t^2 - M({}^t \dot{V} + \Psi_1 {}^t \ddot{V} \Delta t) \\ &\quad - C({}^t \dot{V} + \Psi_1 {}^t \dot{V} \Delta t + \eta_2 \Psi_2 {}^t \ddot{V} \Delta t^2) \\ &\quad - K({}^t U + \Psi_1 {}^t \dot{U} \Delta t + \beta_2 \Psi_2 {}^t \ddot{U} \Delta t^2) \end{aligned} \quad (3-145)$$

$$\gamma_2 \Pi_2 \Delta \ddot{U} = \eta_2 \Pi_2 {}^t \ddot{V} \Delta t + \eta_3 \Pi_3 \Delta {}^t \ddot{V} \Delta t \quad (3-146)$$

其中下一时刻的状态量由

$${}^{t+\Delta t} U = {}^t U + {}^t V \Delta t + \beta_2 {}^t \dot{V} \Delta t^2 + \beta_3 \Delta {}^t \ddot{U} \Delta t^2 \quad (3-147a)$$

$${}^{t+\Delta t} V = {}^t V + {}^t \dot{V} \Delta t + \eta_2 {}^t \ddot{V} \Delta t^2 + \eta_3 \Delta {}^t \ddot{V} \Delta t^2 \quad (3-147b)$$

$${}^{t+\Delta t} \ddot{V} = {}^t \ddot{V} + \Delta \ddot{V} \quad (3-147c)$$

$${}^{t+\Delta t} \dot{V} = M^{-1}({}^{t+\Delta t} F - C {}^{t+\Delta t} V - K {}^{t+\Delta t} U) \quad (3-147d)$$

计算得到。

通过相容性分析，得到如下的精度条件

$$\text{二阶精度: } \beta_2 = 1 - \eta_2 \quad (3-148)$$

$$\text{三阶精度: } \frac{\beta_3}{\gamma_2} = \frac{\eta_3 - \frac{1}{12} - \eta_2(1 - \eta_2)(1 - \frac{1}{\kappa_2}) - \eta_2}{\eta_3 \frac{\Pi_3}{\Pi_2} (1 - \frac{1}{\Psi_1}) - \eta_2} \quad (3-149)$$

$$\begin{aligned} \text{四阶精度: } \frac{\Pi_3}{\Pi_2} = \frac{1}{\eta_3 \frac{\beta_3}{\gamma_2} (2 - \frac{1}{\Psi_1})} & \left\{ \frac{\beta_3}{\gamma_2} \eta_2 \left[2 - \frac{1}{\kappa_2} (1 - \eta_2) \right] + \eta_3 (2 - \eta_2) \right. \\ & \left. - \eta_2 (3 - 2\eta_2) - (1 - \eta_2) \frac{1}{\kappa_2} (\eta_3 - 2\eta_2) \right\} \end{aligned} \quad (3-150)$$

同时，为了获得最大的稳定性极限，令

$$\Psi_1 = 1 \quad (3-151)$$

以及在低频段令谱半径 $\rho = 1$ 有

$$\frac{\Psi_1}{\Psi_2 \kappa_2} = 1 \quad (3-152)$$

最后，剩余两个参数 γ_2, Π_2 ，文 [68] 中令其取值为 1。即

$$\gamma_2 = \Pi_2 = 1 \quad (3-153)$$

将参数的限定条件带入原始的算法格式化简后，得到如下更加简洁的显式积分格式

$$\begin{aligned} M \Delta \ddot{V} \Delta t = & {}^t p + {}^t \dot{p} \Delta t + \frac{1}{2} \Psi_2 {}^t \ddot{p} \Delta t^2 - M({}^t \dot{V} + {}^t \ddot{V} \Delta t) \\ & - C({}^t \dot{V} + {}^t \ddot{V} \Delta t + \eta_2 \Psi_2 {}^t \ddot{V} \Delta t^2) \\ & - K({}^t U + {}^t V \Delta t + (1 - \eta_2) \Psi_2 {}^t \dot{V} \Delta t^2) \end{aligned} \quad (3-154)$$

其中下一时刻的状态量由

$${}^{t+\Delta t} U = {}^t U + {}^t V \Delta t + (1 - \eta_2) {}^t \dot{V} \Delta t^2 + \beta_3 \eta_2 {}^t \ddot{V} \Delta t^3 + \beta_3 \eta_3 \Pi_3 \Delta \ddot{V} \Delta t^3 \quad (3-155a)$$

$${}^{t+\Delta t} V = {}^t V + {}^t \dot{V} \Delta t + \eta_2 {}^t \ddot{V} \Delta t^2 + \eta_3 \Delta \ddot{V} \Delta t^2 \quad (3-155b)$$

$${}^{t+\Delta t} \dot{V} = {}^t \dot{V} + \Delta \ddot{V} \quad (3-155c)$$

$${}^{t+\Delta t} \dot{V} = M^{-1}({}^{t+\Delta t} F - C {}^{t+\Delta t} V - K {}^{t+\Delta t} U) \quad (3-155d)$$

该高阶导数的单步显式算法 (3-154-3-155) 至少是二阶精度的。为了实现更高的精度，需要引入新的参数条件，如等式 (3-149-3-150) 以及考虑稳定性要求。特别地，

当参数选取不同时，提出的高阶导数的单步显式算法将与一些算法谱等价。具体参数取值见表3-5。其中，“HDSSE”代表高阶导数的单步显式算法，而后面的数

表 3-5 高阶导数的单步显式算法参数取值

Table3-5 Parameter adjusting for the higher derivative single step explicit algorithms

算法简称	η_2	η_3	β_3	Ψ_2	Π_3
中心差分法	0	0	0	1	1
NF ^[93]	η_3	$\frac{\beta_3}{2} [\sqrt{1 + \frac{4}{\beta_3}} - 1]$	β_3	$\frac{1}{1-\eta_3}$	$\frac{\beta_3}{\beta_3 - \eta_3^2}$
HDSSE4	η_3	η_3	$\frac{2}{\eta_3} (\frac{1}{24} - \eta_3^2)$	$\frac{1+\eta_3}{1-\eta_3}$	$\frac{1}{1-24\eta_3^2} - \eta_3$
HDSSE3D	$\frac{7}{30}$	$\frac{1}{4}$	$\frac{341}{630}$	$\frac{2}{3}$	$\frac{39}{40}$

字表示其对应的阶数，字母“D”表示存在数值阻尼。同时对于不同的高阶算法的数值性能比较在表3-6给出。

表 3-6 高阶导数的单步显式算法子成员的数值性能比较

Table3-6 Characteristics of the higher order methods covered by the higher derivative single step explicit algorithms

算法简称	参数取值	精度	数值耗散	函数计算	稳定域 Ω_{cr}	注释
中心差分法		2	No	1	2	谱等价
NF2	$\beta_3 = \frac{1}{16}$	2	No	2	4	谱等价
NF4	$\beta_3 = \frac{1}{12}$	4	No	2	$2\sqrt{3}$	谱等价
HDSSE4	$\beta_3 = \frac{1}{2} [\sqrt{\frac{7}{6}} - 1]$	4	No	2	$2\sqrt{3}$	无数值阻尼
HDSSE4D	$\beta_3 = \frac{3}{8} [\sqrt{\frac{7}{6}} - 1]$	4	轻微	2	$2\sqrt{3}$	数值阻尼不大
HDSSE3D		3	Yes	2	3.88	存在数值阻尼

最后，上述的高阶导数单步显式算法也可以以预测-校正形式书写，且与其他算法的性能比较可以参见文献 [69]。

3.2.4 单步单解算法框架/GSSSS 算法

Zhou 和 Tamma 等人针对线性多步法在结构动力学数值响应算法设计中的应用，提出了一个一般性的单步单解算法^[74](The generalized single step single solve algorithms)[GSSSS]。该算法不仅可以将现阶段基于单位移场二阶运动微分方程提出的单步算法^①作为其特例，而且也提供了一些新的算法家族。同时，Zhou 等人也在文 [74] 中证明了 GSSSS 算法是线性多步法在结构动力学数值算法应用中单步算法的体现。换言之，GSSSS 算法是与线性多步法一一对应的。因此由于 Dahlquist

① 这里的单步算法是指其发展思想是在线性多步法的理论框架之下发展的，而非单步多级法，如 Runge-Kutta 法。

定理^[46]的存在, GSSSS 算法仅实现了二阶精度的无条件稳定性。该算法通过使用时间上的加权残值得到其一般性算法格式, 其算法格式的 a -形式描述如下

$$\begin{aligned} & (\Lambda_6 W_1 M + \Lambda_5 W_2 C \Delta t + \Lambda_3 W_3 K \Delta t^2) \Delta \ddot{U} \\ & = -M^t \ddot{U} - C(^t \dot{U} + \Lambda_4 W_1 {}^t \ddot{U} \Delta t) - K(^t U + \Lambda_1 W_1 {}^t \dot{U} \Delta t + \Lambda_2 W_2 {}^t \ddot{U} \Delta t^2) \quad (3-156) \\ & + (1 - W_1) {}^t F + W_1 {}^{t+\Delta t} F \end{aligned}$$

其位移、速度和加速度更新公式为

$${}^{t+\Delta t} U = {}^t U + \lambda_1 {}^t \dot{U} \Delta t + \lambda_2 {}^t \ddot{U} \Delta t^2 + \lambda_3 \Delta \ddot{U} \Delta t^2 \quad (3-157a)$$

$${}^{t+\Delta t} \dot{U} = {}^t \dot{U} + \lambda_4 {}^t \ddot{U} \Delta t + \lambda_5 \Delta \ddot{U} \Delta t \quad (3-157b)$$

$${}^{t+\Delta t} \ddot{U} = {}^t \ddot{U} + \Delta \ddot{U} \quad (3-157c)$$

需要说明的是,

‡ 当参数 $\Lambda_i = \lambda_i, i = 1, 2, \dots, 5$ 时, 该算法将退化为

3.3 时间子步策略

3.3.1 BEF 算法

清华大学的刘天云副研究员给出了一个利用向后 Euler 公式 (BEF) 求解结构动响应^[94]。该算法具有

‡ 易求解性。非常方便的运用到线性系统和非线性系统的求解中。

‡ 不涉及额外的算法参数选择, 如拉格朗日乘子。

‡ 单解性。在离散的时刻具有对称的有效刚度矩阵。

‡ 易编程。非常容易的编程到已有的软件中。

其算法格式如下

$${}^{t+\Delta t} U = {}^t \dot{U} + \frac{\Delta t^2}{2} {}^{t+\Delta t} \ddot{U} \quad (3-158)$$

$${}^{t+\Delta t} \dot{U} = \frac{3}{2\Delta t} {}^{t+\Delta t} U - \frac{2}{\Delta t} {}^t U + \frac{1}{2\Delta t} {}^{t-\Delta t} U \quad (3-159)$$

且在 $t + \Delta t$ 时刻满足平衡方程 (3-1), 即

$$M {}^{t+\Delta t} \ddot{U} + C {}^{t+\Delta t} \dot{U} + K {}^{t+\Delta t} U = {}^{t+\Delta t} F \quad (3-160)$$

在第一个时间步长内, 需要额外的速度启动机制

$$\Delta {}^t \dot{U} = \frac{3}{2\Delta t} (\Delta {}^t U - {}^0 U) - \frac{1}{2} {}^0 \dot{U} + \frac{\Delta t}{4} {}^0 \ddot{U} \quad (3-161)$$

3.3.2 BADF-TR 算法

后来, 刘天云副研究员在 2013 年通过考虑 Newmark 算法家族^[53] 中的参数 $\beta = \gamma = 1/2$ 来分析得到了一个新的保辛算法 (BADF-TR 算法)^[95]。该算法仍然拥有前面提及的 BEF 算法的优点, 同时还是自启动的。其算法格式如下

$${}^{t+\Delta t}U = {}^tU + {}^t\dot{U}\Delta t + \frac{\Delta t^2}{2} {}^{t+\Delta t}\ddot{U} \quad (3-162)$$

$${}^{t+\Delta t}\dot{U} = {}^t\dot{U} + \frac{\Delta t}{2} ({}^t\ddot{U} + {}^{t+\Delta t}\ddot{U}) \quad (3-163)$$

更新公式 (3-162) 是利用了二阶的向后加速度公式 (Backward Acceleration Formulas), 而 (3-163) 则是利用 Trapezoidal 规则。

其实, 参数 $\beta = \gamma = \frac{1}{2}$ 早被 Lee 所推导出来求解拟线性双曲微分方程^[65,66]。

3.3.3 GCD-BDF2 算法

清华大学的李庆斌等人^[96] 利用广义中心差分法和三点向后微分公式结合 (其速度更新公式利用三点向后微分公式而加速度则利用广义中心差分法), 提出了一个新的复合型算法。该算法是二阶收敛的、无条件稳定的单解算法。而且每一个时间步长内只需要求解一次非线性方程组, 其计算量较 Trapezoidal 规则相当。

在 $t + \Delta t$ 时刻的加速度 ${}^{t+\Delta t}\ddot{U}$ 可以由向前 Euler 差分公式得到

$${}^{t+\Delta t}\ddot{U} = {}^t\ddot{U} + \Delta t {}^tU^{(3)} \quad (3-164)$$

其中二阶、三阶段微分表达式则利用中心差分公式得到

$${}^t\ddot{U} = \frac{{}^{t+\Delta t}U - 2{}^tU + {}^{t-\Delta t}U}{\Delta t^2}, \quad {}^tU^{(3)} = \frac{{}^{t+\Delta t}\dot{U} - 2{}^t\dot{U} + {}^{t-\Delta t}\dot{U}}{\Delta t^2} \quad (3-165)$$

将公式 (3-165) 带入到 (3-164) 可以得到其加速度 ${}^{t+\Delta t}\ddot{U}$ 的更新公式

$${}^{t+\Delta t}\ddot{U} = \frac{{}^{t+\Delta t}U - 2{}^tU + {}^{t-\Delta t}U}{\Delta t^2} + \frac{{}^{t+\Delta t}\dot{U} - 2{}^t\dot{U} + {}^{t-\Delta t}\dot{U}}{\Delta t} \quad (3-166)$$

同时, 其速度 ${}^{t+\Delta t}\dot{U}$ 的更新公式则利用三点向后微分公式

$${}^{t+\Delta t}\dot{U} = \frac{1}{2\Delta t} (3{}^{t+\Delta t}U - 4{}^tU + {}^{t-\Delta t}U) \quad (3-167)$$

于是, 公式 (3-166)-(3-167) 结合在 $t + \Delta t$ 时刻的平衡方程, 就得到了 GCD-BDF2 算法。

3.3.4 Soares-Jr 型算法

Soares Jr 近来提出了一些耗散性的时间积分算法^[97-103]，其特点是算法参数依赖于系统的频率值。

3.3.4.1 SJ 算法格式 1^[100]

考虑带有合适初值问题的运动方程 (3-1) 的数值积分算法时，在时间步长 Δt 时积分则有

$$M \int_{t-\Delta t/2}^{t+\Delta t/2} \ddot{U}(\tau) d\tau + C \int_{t-\Delta t/2}^{t+\Delta t/2} \dot{U}(\tau) d\tau + K \int_{t-\Delta t/2}^{t+\Delta t/2} U(\tau) d\tau = \int_{t-\Delta t/2}^{t+\Delta t/2} F(\tau) d\tau \quad (3-168)$$

其中，等式左边的积分可近似为

$$\int_{t-\Delta t/2}^{t+\Delta t/2} \ddot{U}(\tau) d\tau \approx {}^{t+\Delta t}\dot{U} - {}^t\dot{U} \quad \int_{t-\Delta t/2}^{t+\Delta t/2} \dot{U}(\tau) d\tau \approx {}^{t+\Delta t}U - {}^tU \quad (3-169a)$$

$$\int_{t-\Delta t/2}^{t+\Delta t/2} U(\tau) d\tau \approx {}^tU\Delta t + \frac{\Delta t^2}{2}(\alpha_1 {}^{t+\Delta t}\dot{U} + \alpha_2 {}^t\dot{U}) \quad (3-169b)$$

而右边的外力荷载积分项可以使用数值积分方法计算，如 Trapezoidal 积分规则和 Simpson 积分规则等^[104]。

在 $t + \Delta t$ 时刻的位移使用 Trapezoidal 规则^[5] 计算，即

$${}^{t+\Delta t}U = {}^tU + \frac{\Delta t}{2}({}^{t+\Delta t}\dot{U} + {}^t\dot{U}) \quad (3-170)$$

将等式 (3-169)-(3-170) 带入到 (3-168) 可得

$$[M + \frac{1}{2}\Delta t C + \frac{1}{2}\alpha_1 \Delta t^2 K] {}^{t+\Delta t}\dot{U} = [M {}^t\dot{U} - \frac{1}{2}\Delta t C {}^t\dot{U} - K(\Delta t {}^tU + \frac{1}{2}\alpha_2 \Delta t^2 {}^t\dot{U})] + F_t \mathfrak{J} \quad (3-171)$$

其中， $F_t \mathfrak{J}$ 表示等式 (3-168) 右端的荷载积分项。于是在 $t + \Delta t$ 时刻的速度可以通过求解等式 (3-171) 获得。结合等式 (3-170)，一个完整的直接积分格式可以被获得。算法参数 α_1, α_2 控制着算法的各项数值性能，其选取规则如下

$$\alpha_1 = \frac{1}{2} + \frac{3}{2} \tanh(a\omega\Delta t) \quad \alpha_2 = 2\sqrt{2\alpha_1} - \alpha_1 - 1 \quad (3-172)$$

其中 a 为控制参数， ω 为系统的固有频率。在求解多自由度系统时，由 $\omega_i = \sqrt{K_{ii}/M_{ii}}$ 近似求解，此时的参数 $\alpha_i, i = 1, 2$ 是两个向量。

特别地，算法的耗散能力的大小与控制参数 a 成正比，当 $a = 0$ 且荷载积分项选择适当的积分规则时，上述算法将退化为标准的常平均加速度算法^[5]。当 $a \geq 0$ 时，算法实现无条件稳定性。该算法的特点是

‡ 真正意义上的自启动算法。省略了求解加速度响应值。

- ‡ 基于速度和位移格式的单步单解算法。
 - ‡ 实现无条件稳定性 (L-稳定)。
 - ‡ 实现二阶精度
 - ‡ 尽管理论分析得出在位移上存在超调，但其数值阻尼将其完全滤去。故实际上，该算法没有超调现象发生。
 - ‡ 在中频段，数值阻尼大小可控。
- 文 [102] 中着重分析了该算法的一个条件稳定子域

$$(\alpha_1, \alpha_2) \in \{0 < \alpha_1 < \frac{1}{2}, \alpha_2 \geq 1 - \alpha_1\} \quad (3-173)$$

当 $\alpha_2 = 1 - \alpha_1$ 时，该条件稳定算法没有数值阻尼，而对于 $\alpha_2 > 1 - \alpha_1$ 时，数值阻尼则被引入。具体的参数 α_1, α_2 的取值建议为

$$\alpha_1 = \frac{1}{2} \tanh\left(\frac{1}{4}\Omega_{max}\right) \quad \begin{cases} \alpha_2 = 1 - \alpha_1 \text{ 无数值阻尼} \\ \alpha_2 = 2\sqrt{2\alpha_1 + (2/\Omega_s)^2} - (1 + \alpha_1) \text{ 最大数值阻尼} \end{cases} \quad (3-174)$$

其中， Ω_{max}, Ω_s 分别为系统的无量纲最大采样频率和谱半径分岔点处的采样频率（此时谱半径实现最大化）。

3.3.4.2 SJ 算法格式 2^[97]

若在 $t + \Delta t$ 时刻的位移计算公式 (3-170) 改写成一般性的计算格式

$${}^{t+\Delta t}U = {}^tU + \delta_1 {}^{t+\Delta t}\dot{U} + \delta_2 {}^t\dot{U} \quad (3-175)$$

则计算 $t + \Delta t$ 时刻的速度公式变为

$$[M + \delta_1 \Delta t C + \frac{1}{2} \alpha_1 \Delta t^2 K] {}^{t+\Delta t}\dot{U} = [M {}^t\dot{U} - \delta_2 \Delta t C {}^t\dot{U} - K(\Delta t {}^tU + \frac{1}{2} \alpha_2 \Delta t^2 {}^t\dot{U})] + F_i \mathfrak{J} \quad (3-176)$$

其中 $F_i \mathfrak{J}$ 仍表示等式 (3-168) 右端的荷载积分项。其算法参数 $\alpha_i, \delta_i, i = 1, 2$ 可取值为

♣ 显式算法

$$\delta_1 = \frac{1}{2} + d\omega\Delta t \quad \alpha_1 = 0 \quad \delta_2 = 1 - \delta_1 \quad \alpha_2 = 2\delta_2 \quad (3-177)$$

♣ 隐式算法

$$\begin{cases} \delta_1 = \frac{1}{2} + d\omega\Delta t & \alpha_1 = \frac{1}{2} + \frac{3}{2} \tanh(a\omega\Delta t) \\ \delta_2 = \frac{1}{2} - d\omega\Delta t & \alpha_2 = 2\sqrt{2\alpha_1 - \alpha_1 - 2\delta_1} \end{cases} \quad (3-178)$$

其中, 参数 d, a 控制算法的数值特性, 如周期误差、中频段的数值阻尼等。该算法的特点类似于 SJ 算法格式 1。

需要指出的是, 当 $\delta_1 + \delta_2 = 1$ 且 $a = 0$ 时, 上述隐式算法将于 Tamma 等人在 1990 年提出的一个隐式自启动无条件稳定算法^[82]一致, 即当 $C = 0$ 时

$$[M + \frac{\Delta t^2}{4}K]^{t+\Delta t}\dot{U} = [M - \frac{\Delta t^2}{4}K]^t\dot{U} - \Delta t K^t U + \frac{\Delta t}{2}(^tF + {}^{t+\Delta t}F) \quad (3-179)$$

$${}^{t+\Delta t}U = {}^tU + \Delta t[(1 - \gamma)^t\dot{U} + \gamma^{t+\Delta t}\dot{U}] \quad (3-180)$$

对于参数 $\gamma = 0.5$ 可实现无条件稳定和二阶精度^[82]。

3.3.4.3 SJ 算法格式 3^[101]

当 SJ 算法格式 1^[100] 中的公式 (3-169b) 修改为

$$\int_{t-\Delta t/2}^{t+\Delta t/2} U(\tau) d\tau \approx {}^tU\Delta t + \frac{\Delta t^2}{2}[\alpha^{t+\Delta t}\dot{U} + (1 - \alpha)^t\dot{U}] \quad (3-181)$$

其余计算格式不变, 此时在 $t + \Delta t$ 时刻的速度计算方程为

$$[M + \frac{1}{2}\Delta t C + \frac{1}{2}\alpha\Delta t^2 K]^{t+\Delta t}\dot{U} = [M^t\dot{U} - \frac{1}{2}\Delta t C^t\dot{U} - K(\Delta t^tU + \frac{1}{2}(1 - \alpha)\Delta t^2\dot{U})] + {}_{F_i}\mathfrak{J} \quad (3-182)$$

同理, 其 ${}_{F_i}\mathfrak{J}$ 表示等式 (3-168) 右端的荷载积分项。其自适应参数 α 取值为

$$\alpha = \frac{1}{2} \tanh(a\omega\Delta t), \quad a \geq 0.24567002 \quad (3-183)$$

3.3.4.4 SJ 算法格式 4^[98]

当在公式 (3-169b) 中取 $\alpha_1 = 0$ 时, SJ 算法格式 1^[100] 将退化为一个显式的直接积分格式^[98]。同时令其余计算格式不变, 此时在 $t + \Delta t$ 时刻的速度计算方程为

$$[M + \frac{1}{2}\Delta t C]^{t+\Delta t}\dot{U} = [M^t\dot{U} - \frac{1}{2}\Delta t C^t\dot{U} - K(\Delta t^tU + \frac{1}{2}\alpha_2\Delta t^2\dot{U})] + {}_{F_i}\mathfrak{J} \quad (3-184)$$

${}_{F_i}\mathfrak{J}$ 依旧表示等式 (3-168) 右端的荷载积分项。其自适应参数 α_2 取值为

$$\alpha_2 = 1 + a\omega\Delta t \quad (3-185)$$

当 $\alpha_2 \geq 1$ 时, 算法是条件稳定的; 否则, 算法是无条件不稳定的。当 $\alpha_2 = 1$ 时, 算法具有中心差分法的数值特性; 当 $\alpha_2 > 1$ 时, 算法引入数值阻尼, 此时基于稳定性理论的临界采样频率为

$$\Omega_c = \frac{2}{\sqrt{\alpha_2}} \quad (3-186)$$

而谱半径分岔点处的采样频率为

$$\Omega_b = \frac{4}{1 + \alpha_2} \quad (3-187)$$

一般情况下，参数 α_2 建议取值范围在 $1 \leq \alpha_2 \leq 3$ 。

特别地，当参数 $a = 0$ ，且 $_{F_i}\mathfrak{J} = \Delta t {}^tF + \frac{1}{2} {}^t\dot{F} \Delta t^2$ 时，上述算法将与 Kumar K. Tamma 等人在 20 世纪八、九十年代提出的基于 Lax-Wendroff/Taylor-Galerkin 方法的显式速度格式^[78-81]一致。即

$$[M + \frac{1}{2} C \Delta t] {}^{t+\Delta t}\dot{U} = -\Delta t {}^tU K + [M - \frac{1}{2} \Delta t C - \frac{1}{2} \Delta t^2 K] {}^t\dot{U} + \Delta t {}^tF + \frac{1}{2} {}^t\dot{F} \Delta t^2 \quad (3-188)$$

$${}^{t+\Delta t}U = {}^tU + \Delta t [(1 - \gamma) {}^t\dot{U} + \gamma {}^{t+\Delta t}\dot{U}] \quad (3-189)$$

其中，参数 $\gamma \in [0.5, 1]$ 保证了算法的稳定性。当 $\gamma = 0.5$ 时，可以实现二阶精度和最大的稳定域，即 $\Omega_{cr} = 2$ 。

3.3.4.5 SJ 算法格式 4^[103]

Soares Jr 基于前面提出的 SJ 算法格式 2^[97]，提出了一个新的显式积分算法^[103]。针对无阻尼矩阵情况 ($C = 0$)，该算法格式为

$$M {}^{t+\Delta t}\dot{U} = {}_{F_i}\mathfrak{J} + M {}^t\dot{U} - K({}^tU \Delta t + \frac{1}{2} {}^t\dot{U} \Delta t^2) \quad (3-190)$$

$$M {}^{t+\Delta t}U = M[{}^tU + \frac{\Delta t}{2}({}^t\dot{U} + {}^{t+\Delta t}\dot{U})] - K[(\beta b_1 b_2) {}^t\dot{U} \Delta t^3 + (\frac{1}{16} + \beta b_1) {}^{t+\Delta t}\dot{U} \Delta t^3] \quad (3-191)$$

其中， $_{F_i}\mathfrak{J}$ 则为前述提及的荷载积分项。同时参数 β, b_1 和 b_2 取值分别为

$$b_1 = 8.567 \cdot 10^{-3} \quad b_2 = 8.590 \cdot 10^{-1} \quad 0 \leq \beta \leq 1 \quad (3-192)$$

此时，算法的谱半径曲线光滑，无凸起。而参数 $\beta = 0$ 对于无数值阻尼情形，其基于稳定性理论的采样频率极限为 $\Omega_c = 4$ ；而当 $\beta = 1$ 时，谱半径在分岔点达到最小值 $\rho_b = 0$ ，其最大采样频率为 $\Omega_c \approx 3.571$ 。

该算法的优点主要是

- ‡ 真正意义上的自启动，不需要任何求解加速度的机制。
- ‡ 算法简单，求解方便且实现二阶精度。
- ‡ 显式的条件稳定算法，其稳定域是中心差分法的二倍，但计算量几乎相当。

‡ 实现数值阻尼的可控性 (由参数 $\beta \in [0, 1]$ 控制)。

对于有阻尼系统 ($\xi \neq 0$) 时，该算法格式如下

$$[M + \frac{1}{2} \Delta t C] {}^{t+\Delta t}\dot{U} = {}_{F_i}\mathfrak{J} + M {}^t\dot{U} - \frac{1}{2} {}^t\dot{U} C \Delta t - K({}^tU \Delta t + \frac{1}{2} {}^t\dot{U} \Delta t^2) \quad (3-193)$$

$$[M + \frac{1}{2}\Delta t C]^{t+\Delta t} U = [M + \frac{1}{2}\Delta t C] \cdot [{}^t U + \frac{\Delta t}{2}({}^t \dot{U} + {}^{t+\Delta t} \dot{U})] - \frac{1}{2} {}^{t+\Delta t} \dot{U} \Delta t^2 C - K[(\beta b_1 b_2) {}^t \dot{U} \Delta t^3 + (\frac{1}{16} + \beta b_1) {}^{t+\Delta t} \dot{U} \Delta t^3] \quad (3-194)$$

3.3.4.6 稳定化的中心差分法

Großholz 和 Soares Jr 等人^[99] 利用一个依赖于结构的参数引入人工质量矩阵进而将传统的中心差分法^[55] 实现了无条件稳定。该算法针对单自由度无阻尼自由振动系统

$$\ddot{U}(t) + \omega^2 U(t) = 0 \quad (3-195)$$

引入依赖于结构的参数后

$$(1 + a(\Delta t \omega)^2) \ddot{U}(t) + \omega^2 U(t) = 0 \quad (3-196)$$

对 $\ddot{U}(t)$ 使用二阶中心差分后并带入上式得

$${}^{t+\Delta t} U = \left(2 - \frac{(\Delta t \omega)^2}{1 + a(\Delta t \omega)^2} \right) {}^t U - {}^{t-\Delta t} U \quad (3-197)$$

如果需要，其速度和加速度仍使用传统的中心差分法求解格式^[55] 计算。

参数 a 满足 $0 \leq a \leq 0.25$ ；当 $a = 0$ 时，该稳定化的中心差分法退化为传统的中心差分法^[55]；当 $a = 0.25$ 时，该方法与常平均加速度算法^[5] 是谱等价的。数值实验表明，引进的稳定化中心差分法精度较传统的中心差分法和常平均加速度算法高。但引入的技术也需要事先对待分析的系统的频率进行估计和计算，尤其是求解大规模多自由度系统中，可能是一件比较麻烦的事情。

3.4 预测-校正算法

3.5 结构依赖型直接积分法

3.6 复合型子步隐式算法

Tarnow 和 Simo 在 1994 年提出了一个子步复合方法^[105]，将一般的二阶直接积分法的精度提高到四阶。在该方法中，一个时间步长内需要三次的子步计算。同时原来的二阶方法的稳定、守恒性质仍保持不变。需要注意的是，当该方法运用到二阶的 Newmark- β 方法^[53] 时，导出的高阶算法是没有数值耗散的。为了引入高频耗散特性，需要对求得的响应进行一定的技术处理，如后处理的滤波技术^[106]。后来，又有许多研究学者进行进一步的探索，发展了许多可靠优质的直接积分算法。

3.6.1 TR-BDF2 算法

麻省理工的 K.J. Bathe 教授在 2005 年提出了一个两子步算法^[107, 108]。假设在 t 时刻的状态量, 即 ${}^tU, {}^t\dot{U}, {}^t\ddot{U}$, 都是已知的。为了求在时刻 $t + \Delta t$ 的未知量 ${}^{t+\Delta t}U, {}^{t+\Delta t}\dot{U}, {}^{t+\Delta t}\ddot{U}$, 该方法将一个时间步长 Δt 划分为两个子时间步长 $\gamma\Delta t, (1 - \gamma)\Delta t$ 。

在第一个时间子步 $\gamma\Delta t$ 内, 采用 Trapezoidal 规则^[5] 进行计算, 即

$${}^{t+\gamma\Delta t}\dot{U} = {}^t\dot{U} + \frac{{}^t\ddot{U} + {}^{t+\gamma\Delta t}\ddot{U}}{2}\gamma\Delta t \quad (3-198)$$

$${}^{t+\gamma\Delta t}U = {}^tU + \frac{{}^t\dot{U} + {}^{t+\gamma\Delta t}\dot{U}}{2}\gamma\Delta t \quad (3-199)$$

为了求得在 $t + \Delta t$ 时刻的加速度, 需要引入在该时刻的离散形式的平衡方程

$$M^{t+\gamma\Delta t}\ddot{U} + C^{t+\gamma\Delta t}\dot{U} + K^{t+\gamma\Delta t}U = {}^{t+\gamma\Delta t}F \quad (3-200)$$

利用方程 (3-198)-(3-200) 就可以求得在中间时刻 $t + \gamma\Delta t$ 处的位移 ${}^{t+\gamma\Delta t}U$ 、速度 ${}^{t+\gamma\Delta t}\dot{U}$ 和加速度 ${}^{t+\gamma\Delta t}\ddot{U}$ 。然后对于第二子步内的更新则是利用了三点的向后微分公式^[107]

$${}^{t+\Delta t}\dot{f} = c_1 {}^t\dot{f} + c_2 {}^{t+\gamma\Delta t}\dot{f} + c_3 {}^{t+\Delta t}\dot{f} \quad (3-201)$$

其中, 系数 c_1, c_2 和 c_3 分别为

$$c_1 = \frac{1 - \gamma}{\Delta t \gamma} \quad (3-202)$$

$$c_2 = \frac{-1}{(1 - \gamma)\gamma\Delta t} \quad (3-203)$$

$$c_3 = \frac{2 - \gamma}{(1 - \gamma)\Delta t} \quad (3-204)$$

于是, 当 f 分别表示位移和速度时, 则有:

$${}^{t+\Delta t}\dot{U} = c_1 {}^t\dot{U} + c_2 {}^{t+\gamma\Delta t}\dot{U} + c_3 {}^{t+\Delta t}\dot{U} \quad (3-205)$$

$${}^{t+\Delta t}\ddot{U} = c_1 {}^t\ddot{U} + c_2 {}^{t+\gamma\Delta t}\ddot{U} + c_3 {}^{t+\Delta t}\ddot{U} \quad (3-206)$$

同时, 利用 $t + \Delta t$ 时刻的平衡方程

$$M^{t+\Delta t}\ddot{U} + C^{t+\Delta t}\dot{U} + K^{t+\Delta t}U = {}^{t+\Delta t}F \quad (3-207)$$

通过利用方程 (3-205)-(3-160) 就可以求出在时刻 $t + \Delta t$ 的位移 ${}^{t+\Delta t}U$ 、速度 ${}^{t+\Delta t}\dot{U}$ 和加速度 ${}^{t+\Delta t}\ddot{U}$ 。这样就完成了状态量从 t 时刻到 $t + \Delta t$ 的转换, 进而可以求解到满意的时间内的位移、速度和加速度值。

K.J. Bathe 教授在文 [107, 108] 并没有给出在第二个子步内求解非线性问题时, 迭代的位移初值选择规则。而文 [109] 中给出了下列的预测值计算格式

$${}^{t+\Delta t}U = {}^{t+\gamma\Delta t}U + (1 - \gamma)\Delta t {}^{t+\gamma\Delta t}\dot{U} + \frac{(1 - \gamma)^2\Delta t^2}{4} {}^{t+\gamma\Delta t}\ddot{U} \quad (3-208)$$

需要说明的是, 章杰等人^[110]已经证明了, 若在第一子步内使用 Newmark 算法^[5], 则 K.J. Bathe 所提出的复合型子步算法在众多算法中具有最优性。同时, K.J. Bathe 在文章 [107] 中使用了 $\gamma = 0.5$ 来求解非线性问题较 Trapezoidal 规则取得了一定的优势。另外一个值得推荐的 γ 取值是 $2 - \sqrt{2}$ 。 $\gamma = 2 - \sqrt{2}$ 的使用具有以下几方面的意义:

- 在求解非线性问题时, 可以使得在两个子步内的有效刚度矩阵是一致的, 进而降低计算量^[108, 111]。如求解线性问题, 该值可以给出问题的最优数值响应值^[108]。
- 该值给出了最小的截断误差常数^[110, 111]和最大的线性化稳定域^[111]。
- 在众多合适的 γ 中, 该值能实现谱半径和周期误差曲线最低; 同时, 数值能量耗散最大^[112]。

有意思的是, $\gamma = 0.731$ 被 Klarmann 等人所建议^[112]。该值能提升加速度响应的计算精度。

当然, 上述的复合子步技巧也可以用于多个子步, 亦即在一个时间步长 Δt 内, 将其分为 n 份, 每份不一定要求相等, 而在每一个子步内使用不同的积分方法进行求解, 进而复合成新的直接积分法。利用该技巧, 许多学者又提出了新的复合型子步算法。

3.6.2 TR-TR-Houbolt 算法

K.J. Bathe 等人将上述思想一般化, 提出了一个三子步复合型算法^[107]。将时间步长均分为三个相等的子步长 $\Delta t/3$, 在前两个子步长内均使用 Trapezoidal 规则, 在第三个子步长内, 利用已知的 $t, t + \Delta t/3, t + 2\Delta t/3$ 时刻的信息雇用 Houbolt 算法^[5, 55]进行计算获得在 $t + \Delta t$ 时刻的状态量。在三个子步内的更新公式分别如下

$$\text{第一个子步} \begin{cases} {}^{t+\Delta t/3}\dot{U} = {}^t\dot{U} + \frac{1}{6}\Delta t({}^t\ddot{U} + {}^{t+\Delta t/3}\ddot{U}) \\ {}^{t+\Delta t/3}U = {}^tU + \frac{1}{6}\Delta t({}^t\dot{U} + {}^{t+\Delta t/3}\dot{U}) \end{cases} \quad (3-209)$$

$$\text{第二个子步} \begin{cases} {}^{t+2\Delta t/3}\dot{U} = {}^{t+\Delta t/3}\dot{U} + \frac{1}{6}\Delta t({}^{t+\Delta t/3}\ddot{U} + {}^{t+2\Delta t/3}\ddot{U}) \\ {}^{t+2\Delta t/3}U = {}^{t+\Delta t/3}U + \frac{1}{6}\Delta t({}^{t+\Delta t/3}\dot{U} + {}^{t+2\Delta t/3}\dot{U}) \end{cases} \quad (3-210)$$

$$\text{第三个子步} \begin{cases} \Delta t \cdot {}^{t+\Delta t}\dot{U} = d_1 {}^tU + d_2 {}^{t+\Delta t/3}U + d_3 {}^{t+2\Delta t/3}U + d_4 {}^{t+\Delta t}U \\ \Delta t \cdot {}^{t+\Delta t}\ddot{U} = d_1 {}^t\dot{U} + d_2 {}^{t+\Delta t/3}\dot{U} + d_3 {}^{t+2\Delta t/3}\dot{U} + d_4 {}^{t+\Delta t}\dot{U} \end{cases} \quad (3-211)$$

其系数取值为

$$d_1 = -1 \quad d_2 = \frac{9}{2} \quad d_3 = -9 \quad d_4 = \frac{11}{2} \quad (3-212)$$

3.6.3 BDF2-Newmark 算法

利用复合子步的技巧,许多学者^[110,113,114]将时间步长 Δt 一份为二,在前一个子步内使用 Newmark 算法^[53],后一个使用三点向后微分公式 (BDF2)^[19]。同时假定在两个子步不一定均分。进而分析了子步的不均等划分以及 Newmark 参数对数值性能的影响。在第一个子步内使用 Newmark 算法有

$${}^{t+p\Delta t}\dot{U} = {}^t\dot{U} + p\Delta t[(1-\gamma){}^t\ddot{U} + \gamma{}^{t+p\Delta t}\ddot{U}] \quad (3-213)$$

$${}^{t+p\Delta t}U = {}^tU + p\Delta t{}^t\dot{U} + (p\Delta t)^2 \left[\left(\frac{1}{2} - \beta \right) {}^t\ddot{U} + \beta {}^{t+p\Delta t}\ddot{U} \right] \quad (3-214)$$

在第二个子步内使用三点向后微分公式 (3-205)-(3-206)。

文 [110] 已经给出结论,当参数取值满足

$$p = 2 - \sqrt{2} \quad \gamma = \frac{1}{2} \quad \beta = \frac{1}{4} \quad (3-215)$$

TR-Newmark 算法给出了位移和加速度的最小误差主项。此时,TR-Newmark 算法将退化为 TR-BDF2 算法。

3.6.4 BDF 型积分算法

对 K.J. Bathe 提出的复合型子步算法进行扩展的策略是利用下列基于向后微分公式的广义四步线性多步法^[115]:

$$\alpha_1 {}^{t+\Delta t}y + \alpha_2 {}^ty + \left(\frac{7}{2} - 6\alpha_1 - 3\alpha_2 \right) {}^{t-\Delta t}y + (8\alpha_1 + 3\alpha_2 - 6) {}^{t-2\Delta t}y + \left(\frac{5}{2} - 3\alpha_1 - \alpha_2 \right) {}^{t-3\Delta t}y = {}^{t+\Delta t}\dot{y}\Delta t \quad (3-216)$$

其零稳定性要求参数 α_1 和 α_2 需要满足下列条件

$$\alpha_2 \leq -2\alpha_1 + \frac{3}{2} \quad \alpha_2 > -\frac{18}{5}\alpha_1 + \frac{5}{2} \quad (3-217)$$

当参数 α_2 满足 $\alpha_2 = \frac{5}{2} - 3\alpha_1$ 时，四步向后微分公式 (3-216) 将退化为三步形式，即

$$\alpha_1^{t+\Delta t} \ddot{y} + \left(\frac{5}{2} - 3\alpha_1\right)^t \dot{y} + (3\alpha_1 - 4)^{t-\Delta t} y + \left(\frac{3}{2} - \alpha_1\right)^{t-2\Delta t} y = {}^{t+\Delta t} \dot{y} \Delta t \quad (3-218)$$

此时的零稳定要求 $\alpha_1 \geq 1$ 。需要注意到是

- 当 $\alpha_1 = \frac{3}{2}$ ，公式 (3-218) 退化为二阶的三点向后微分公式 (BDF2)。
- 当 $\alpha_1 = \frac{11}{6}$ ，公式 (3-218) 退化为三阶的四点向后微分公式 (BDF3)。

Dong 在文 [115] 中令 $\alpha_1 = \frac{11}{6} - \frac{\theta}{3}$ ，于是得到与公式 (3-218) 等价形式

$$A(\theta)^{t+\Delta t} \ddot{y} + B(\theta)^t \dot{y} + C(\theta)^{t-\Delta t} y + D(\theta)^{t-2\Delta t} y = {}^{t+\Delta t} \dot{y} \Delta t \quad (3-219)$$

其中

$$A(\theta) = \frac{11}{6} - \frac{\theta}{3} \quad B(\theta) = \theta - 3 \quad (3-220)$$

$$C(\theta) = \frac{3}{2} - \theta \quad D(\theta) = -\frac{1}{3} + \frac{\theta}{3} \quad (3-221)$$

此时的零稳定性对参数 θ 要求 $\theta \leq \frac{5}{2}$ 。

基于公式 (3-219)，第一个积分算法可构造如下：

$${}^{t+\Delta t} \dot{U} = \frac{A(\theta_1)}{\Delta t} {}^{t+\Delta t} U + \frac{B(\theta_1)}{\Delta t} {}^t U + \frac{C(\theta_1)}{\Delta t} {}^{t-\Delta t} U + \frac{D(\theta_1)}{\Delta t} {}^{t-2\Delta t} U \quad (3-222)$$

$${}^{t+\Delta t} \ddot{U} = \frac{A(\theta_2)}{\Delta t} {}^{t+\Delta t} \dot{U} + \frac{B(\theta_2)}{\Delta t} {}^t \dot{U} + \frac{C(\theta_2)}{\Delta t} {}^{t-\Delta t} \dot{U} + \frac{D(\theta_2)}{\Delta t} {}^{t-2\Delta t} \dot{U} \quad (3-223)$$

有意思的是，算法参数 θ_1 和 θ_2 可以取不同的值。当然根据前面的陈述，它们都应该满足零稳定性要求，即 $\theta_1, \theta_2 \leq \frac{5}{2}$ 。Dong 通过考虑一个单自由度振子分析了该算法获得无条件稳定性时的参数 θ_1, θ_2 的取值范围。

考虑更加一般性的四点向后微分公式 (3-216)，令下列的参数变换

$$\alpha_1 = -\frac{1}{3}\theta_1\theta_2 - \frac{1}{4}\theta_2 + \frac{25}{12} \quad \alpha_2 = \theta_1\theta_2 + \theta_2 - 4 \quad (3-224)$$

则，等式 (3-216) 可化为

$$\begin{aligned} & A_1(\theta_1, \theta_2)^{t+\Delta t} \ddot{y} + A_2(\theta_1, \theta_2)^t \dot{y} + A_3(\theta_1, \theta_2)^{t-\Delta t} y \\ & + A_4(\theta_1, \theta_2)^{t-2\Delta t} y + A_5(\theta_1, \theta_2)^{t-3\Delta t} y = {}^{t+\Delta t} \dot{y} \Delta t \end{aligned} \quad (3-225)$$

其中，

$$\begin{cases} A_1(\theta_1, \theta_2) = -\frac{1}{3}\theta_1\theta_2 - \frac{1}{4}\theta_2 + \frac{25}{12} \\ A_2(\theta_1, \theta_2) = \theta_1\theta_2 + \theta_2 - 4 \\ A_3(\theta_1, \theta_2) = -\theta_1\theta_2 - \frac{3}{2}\theta_2 + 3 \\ A_4(\theta_1, \theta_2) = \frac{1}{3}\theta_1\theta_2 + \theta_2 - \frac{4}{3} \\ A_5(\theta_1, \theta_2) = \frac{1}{4} - \frac{1}{4}\theta_2 \end{cases} \quad (3-226)$$

同理，其零稳定性要求

$$\begin{cases} (\theta_1 + \frac{3}{2})\theta_2 \leq 4 \\ (\theta_1 - \frac{1}{2})\theta_2 < 5 \end{cases} \quad (3-227)$$

于是，基于等式 (3-225) 可得第二个积分算法如下：

$$\begin{aligned} {}^{t+\Delta t}\dot{U} = & \frac{A_1(\theta_1, \theta_2)}{\Delta t} {}^{t+\Delta t}U + \frac{A_2(\theta_1, \theta_2)}{\Delta t} {}^tU + \frac{A_3(\theta_1, \theta_2)}{\Delta t} {}^{t-\Delta t}U \\ & + \frac{A_4(\theta_1, \theta_2)}{\Delta t} {}^{t-2\Delta t}U + \frac{A_5(\theta_1, \theta_2)}{\Delta t} {}^{t-3\Delta t}U \end{aligned} \quad (3-228)$$

$$\begin{aligned} {}^{t+\Delta t}\ddot{U} = & \frac{A_1(\theta_1, \theta_2)}{\Delta t} {}^{t+\Delta t}\dot{U} + \frac{A_2(\theta_1, \theta_2)}{\Delta t} {}^t\dot{U} + \frac{A_3(\theta_1, \theta_2)}{\Delta t} {}^{t-\Delta t}\dot{U} \\ & + \frac{A_4(\theta_1, \theta_2)}{\Delta t} {}^{t-2\Delta t}\dot{U} + \frac{A_5(\theta_1, \theta_2)}{\Delta t} {}^{t-3\Delta t}\dot{U} \end{aligned} \quad (3-229)$$

可以通过计算一个单自由度振子来获得该算法的无条件稳定域。Dong 在文 [115] 指出当参数 $(\theta_1, \theta_2) = (1/2, 4/5)$ 是无条件稳定的，且具有最小的耗散特性。

最后，Dong 通过利用 K.J. Bathe 的复合子步技巧，利用前面发展的三步、四步向后微分公式结合 Trapezoidal 规则提出了新的复合型子步算法^[115]。即将一个时间步长分为两个相等的子步长，在前一个子步长内利用 Trapezoidal 规则，即公式 (3-198)-(3-200) 带有 $\gamma = 1/2$ ，在第二个子步雇用前面发展的向后微分公式进行计算，即

$$\begin{aligned} {}^{t+\Delta t}\dot{U} = & \frac{A_1(\theta_1, \theta_2)}{\Delta t/2} {}^{t+\Delta t}U + \frac{A_2(\theta_1, \theta_2)}{\Delta t/2} {}^{t+\frac{1}{2}\Delta t}U + \frac{A_3(\theta_1, \theta_2)}{\Delta t/2} {}^tU \\ & + \frac{A_4(\theta_1, \theta_2)}{\Delta t/2} {}^{t-\frac{1}{2}\Delta t}U + \frac{A_5(\theta_1, \theta_2)}{\Delta t/2} {}^{t-\Delta t}U \end{aligned} \quad (3-230)$$

$$\begin{aligned} {}^{t+\Delta t}\ddot{U} = & \frac{A_1(\theta_1, \theta_2)}{\Delta t/2} {}^{t+\Delta t}\dot{U} + \frac{A_2(\theta_1, \theta_2)}{\Delta t/2} {}^{t+\frac{1}{2}\Delta t}\dot{U} + \frac{A_3(\theta_1, \theta_2)}{\Delta t/2} {}^t\dot{U} \\ & + \frac{A_4(\theta_1, \theta_2)}{\Delta t/2} {}^{t-\frac{1}{2}\Delta t}\dot{U} + \frac{A_5(\theta_1, \theta_2)}{\Delta t/2} {}^{t-\Delta t}\dot{U} \end{aligned} \quad (3-231)$$

通过零稳定性分析, 可以得到其参数的约束条件为

$$(\theta_1 + \frac{3}{2})\theta_2 < \frac{19}{4}, \quad \theta_2 \geq -2 \quad (3-232)$$

或者

$$(\theta_1 + \frac{3}{2})\theta_2 > \frac{19}{4}, \quad \theta_2 \leq -2 \quad (3-233)$$

若在第一个时间子步内, 使用 Newmark- $(\beta, \gamma = 1/2)$ 算法^[5] 进行计算, 即

$$\theta_2(t^{+\frac{1}{2}\Delta t}U - {}^tU) = \theta_2\left(\frac{\Delta t}{2}\right){}^t\dot{U} + \left(\frac{\Delta t}{2}\right)^2 t^{+\frac{1}{2}\Delta t}\ddot{U} + \left(\frac{\theta_2}{2} - 1\right)\left(\frac{\Delta t}{2}\right)^2 {}^t\ddot{U} \quad (3-234)$$

$$t^{+\frac{1}{2}\Delta t}\dot{U} - {}^t\dot{U} = \frac{\Delta t}{4}(t^{+\frac{1}{2}\Delta t}\ddot{U} - {}^t\ddot{U}) \quad (3-235)$$

参数 θ_2 实质上是扮演着 $1/\beta$ 的角色。当 $\theta_2 = 4$ 将退化为 Trapezoidal 规则。为了实现在第一个子步内地无条件稳定性, 参数 θ_2 应当满足^[5]

$$0 < \theta_2 \leq 4 \quad (3-236)$$

在第二个子步内, Dong 使用了如下的格式计算 $t + \Delta t$ 时刻的位移和速度, 即

$$\begin{aligned} (3 - \frac{3}{2}\theta_1)t^{+\Delta t}U - 6(1 - \theta_1)t^{+\frac{1}{2}\Delta t}U + \left(3 - \frac{9}{2}\theta_1\right){}^tU \\ = 2\frac{\Delta t}{2}t^{+\frac{1}{2}\Delta t}\dot{U} + (3\theta_1 - 2)\frac{\Delta t}{2}{}^t\dot{U} + \left(\frac{\Delta t}{2}\right)^2 t^{+\Delta t}\ddot{U} \end{aligned} \quad (3-237)$$

$$\frac{\Delta t}{2}t^{+\Delta t}\dot{U} = \frac{3}{2}t^{+\Delta t}U - 2t^{+\frac{1}{2}\Delta t}U + \frac{1}{2}{}^tU \quad (3-238)$$

通过 Taylor 级数展开可以证明该格式对于任意的 θ_1 取值都是二阶精度的^[115]。当参数 $\theta_1 = 1/2$ 时, 上式退化为二阶的向后微分公式 (BDF2)。于是, 当参数 $(\theta_1, \theta_2) = (1/2, 4)$ 时, 对于该复合型子步算法退化为前述提及的 K.J. Bathe 复合型子步算法^[107?, 108]。

3.6.5 TTBDf 算法

Chandra 和 Zhou 等人基于 Dong 提出的 BDF 型算法^[115] 提出了一个三子步方法^[116]。亦即将时间步长 Δt 分为三等份。在第一个子步内使用 Trapezoidal 规则, 即

$$t^{+\frac{1}{3}\Delta t}U = {}^tU + \frac{\Delta t}{6}({}^t\dot{U} + t^{+\frac{1}{3}\Delta t}\dot{U}) \quad (3-239)$$

$$t^{+\frac{1}{3}\Delta t}\dot{U} = {}^t\dot{U} + \frac{\Delta t}{6}({}^t\ddot{U} + t^{+\frac{1}{3}\Delta t}\ddot{U}) \quad (3-240)$$

且在 $t + \frac{1}{3}\Delta t$ 时刻处满足平衡方程 (3-1)。

在第二个子步内仍然雇用 Trapezoidal 规则进行计算，则有

$${}^{t+\frac{2}{3}\Delta t}U = {}^{t+\frac{1}{3}\Delta t}U + \frac{\Delta t}{6}({}^{t+\frac{1}{3}\Delta t}\dot{U} + {}^{t+\frac{2}{3}\Delta t}\dot{U}) \quad (3-241)$$

$${}^{t+\frac{2}{3}\Delta t}\dot{U} = {}^{t+\frac{1}{3}\Delta t}\dot{U} + \frac{\Delta t}{6}({}^{t+\frac{1}{3}\Delta t}\ddot{U} + {}^{t+\frac{2}{3}\Delta t}\ddot{U}) \quad (3-242)$$

且在 $t + \frac{2}{3}\Delta t$ 时刻处满足平衡方程 (3-1)。

在最后一个子步，也就是第三个子步内利用 Dong 提出的四点向后微分公式 (3-219)，

$${}^{t+\Delta t}\dot{U} = \frac{A(\theta_1)}{\Delta t/3}{}^{t+\Delta t}U + \frac{B(\theta_1)}{\Delta t/3}{}^{t+\frac{2}{3}\Delta t}U + \frac{C(\theta_1)}{\Delta t/3}{}^{t+\frac{1}{3}\Delta t}U + \frac{D(\theta_1)}{\Delta t/3}{}^tU \quad (3-243)$$

$${}^{t+\Delta t}\ddot{U} = \frac{A(\theta_2)}{\Delta t/3}{}^{t+\Delta t}\dot{U} + \frac{B(\theta_2)}{\Delta t/3}{}^{t+\frac{2}{3}\Delta t}\dot{U} + \frac{C(\theta_2)}{\Delta t/3}{}^{t+\frac{1}{3}\Delta t}\dot{U} + \frac{D(\theta_2)}{\Delta t/3}{}^t\dot{U} \quad (3-244)$$

其中， A, B, C, D 取值来自于等式 (3-220)-(3-221)。当参数 $(\theta_1, \theta_2) = (0.75, 0.75)$ 时，在众多的无条件稳定算法中，具有最小的耗散特性^[116]。

3.6.6 BDF2-BDF3 算法

根据文 [19]，可以知道二阶的向后微分公式 (BDF2) 可以表达为

$${}^{t+\Delta t}\dot{U} = \frac{3}{2\Delta t}{}^{t+\Delta t}U - \frac{2}{\Delta t}{}^tU + \frac{1}{2\Delta t}{}^{t-\Delta t}U \quad (3-245)$$

而三阶向后微分公式 (BDF3) 则为

$${}^{t+\Delta t}\dot{U} = \frac{11}{6\Delta t}{}^{t+\Delta t}U - \frac{3}{\Delta t}{}^tU + \frac{3}{2\Delta t}{}^{t-\Delta t}U - \frac{1}{3\Delta t}{}^{t-2\Delta t}U \quad (3-246)$$

于是，Dong 将 BDF2 和 BDF3 公式组合，可以得到如下的组合模式^[117]

$$\begin{aligned} {}^{t+\Delta t}\dot{U} = & (1 - \theta_1) \left(\frac{11}{6\Delta t}{}^{t+\Delta t}U - \frac{3}{\Delta t}{}^tU + \frac{3}{2\Delta t}{}^{t-\Delta t}U - \frac{1}{3\Delta t}{}^{t-2\Delta t}U \right) \\ & + \theta_1 \left(\frac{3}{2\Delta t}{}^{t+\Delta t}U - \frac{2}{\Delta t}{}^tU + \frac{1}{2\Delta t}{}^{t-\Delta t}U \right) \end{aligned} \quad (3-247)$$

$$\begin{aligned} {}^{t+\Delta t}\ddot{U} = & (1 - \theta_2) \left(\frac{11}{6\Delta t}{}^{t+\Delta t}\dot{U} - \frac{3}{\Delta t}{}^t\dot{U} + \frac{3}{2\Delta t}{}^{t-\Delta t}\dot{U} - \frac{1}{3\Delta t}{}^{t-2\Delta t}\dot{U} \right) \\ & + \theta_2 \left(\frac{3}{2\Delta t}{}^{t+\Delta t}\dot{U} - \frac{2}{\Delta t}{}^t\dot{U} + \frac{1}{2\Delta t}{}^{t-\Delta t}\dot{U} \right) \end{aligned} \quad (3-248)$$

当参数 (θ_1, θ_2) 取不同的值时，上述组合算法 BDF2-BDF3 将退化为不同的积分公式，如

- 当 $(\theta_1, \theta_2) = (0, 0)$ 时，BDF2-BDF3 退化为 BDF3。
- 当 $(\theta_1, \theta_2) = (1, 1)$ 时，BDF2-BDF3 退化为 BDF2。

- 当 $(\theta_1, \theta_2) = (1/2, 1/2)$ 时, BDF2-BDF3 退化为著名的 Park 积分算法^[118]。

当参数 (θ_1, θ_2) 在集合 $I = \{(\theta_1, \theta_2) | \theta_1 + \theta_2 \geq 1, 0 \leq \theta_1, \theta_2 \leq 1\}$ 时, BDF2-BDF3 实现无条件稳定^[117]。

3.6.7 BDF3BDF2-Newmark 算法

Dong 利用前述提及的 BDF2-BDF3 组合算法和 Trapezoidal 规则进行组合, 进而提出了一个新的复合型子步算法, BDF3BDF2-Newmark 算法^[117]。该算法将时间步长 Δt 分为两个相等的子步长 $\Delta t/2$, 在第一个子步长内使用 Trapezoidal 规则, 即

$${}^{t+\frac{1}{2}\Delta t}U = {}^tU + \frac{\Delta t}{4}({}^t\dot{U} + {}^{t+\frac{1}{2}\Delta t}\dot{U}) \quad (3-249)$$

$${}^{t+\frac{1}{2}\Delta t}\dot{U} = {}^t\dot{U} + \frac{\Delta t}{4}({}^t\ddot{U} + {}^{t+\frac{1}{2}\Delta t}\ddot{U}) \quad (3-250)$$

对于第二个子步长, 则使用 BDF2-BDF3 组合算法, 即

$${}^{t+\Delta t}\dot{U} = \frac{A(\theta_1)}{\Delta t/3}{}^{t+\Delta t}U + \frac{B(\theta_1)}{\Delta t/3}{}^{t+\frac{1}{2}\Delta t}U + \frac{C(\theta_1)}{\Delta t/3}{}^tU + \frac{D(\theta_1)}{\Delta t/3}{}^{t-\frac{1}{2}\Delta t}U \quad (3-251)$$

$${}^{t+\Delta t}\ddot{U} = \frac{A(\theta_2)}{\Delta t/3}{}^{t+\Delta t}\dot{U} + \frac{B(\theta_2)}{\Delta t/3}{}^{t+\frac{1}{2}\Delta t}\dot{U} + \frac{C(\theta_2)}{\Delta t/3}{}^t\dot{U} + \frac{D(\theta_2)}{\Delta t/3}{}^{t-\frac{1}{2}\Delta t}\dot{U} \quad (3-252)$$

其中,

$$\begin{cases} A(\theta) = \frac{11}{6}(1 - \theta) + \frac{3}{2}\theta \\ B(\theta) = 3(\theta - 1) - 2\theta \\ C(\theta) = \frac{3}{2}(1 - \theta) + \frac{1}{2}\theta \\ D(\theta) = \frac{1}{3}(\theta - 1) \end{cases} \quad (3-253)$$

显然, 该算法不是自启动的。在第一个时间步长内, 需要额外的启动机制, 如连续使用两次 Trapezoidal 规则。当参数 $(\theta_1, \theta_2) = (1, 1)$ 时, BDF3BDF2-Newmark 算法将退化为 K.J. Bathe 提出的 TR-BDF2 算法。

3.6.8 CTFEA 算法

Kim 和 Reddy 利用时间有限元配点方法导出了一个复合型子步算法 (CTFEA)^[119], 在第一个子步 $(t_s, t_s + \gamma\Delta t)$ 内, 使用线性插值形函数如下

$$\psi_s^L = \frac{\gamma\Delta t - t + t_s}{\gamma\Delta t} \quad \psi_{s+\gamma\Delta t}^L = \frac{t - t_s}{\gamma\Delta t} \quad (3-254)$$

其上标 L 表示使用了线性插值函数。于是使用上述形函数，在第一个子步内的状态变量值可以近似表达为

$$\begin{cases} U(t) \approx {}^tU = \psi_s^L(t){}^sU + \psi_{s+\gamma\Delta t}^L(t){}^{s+\gamma\Delta t}U \\ \dot{U}(t) \approx {}^t\dot{U} = \psi_s^L(t){}^s\dot{U} + \psi_{s+\gamma\Delta t}^L(t){}^{s+\gamma\Delta t}\dot{U} \\ \ddot{U}(t) \approx {}^t\ddot{U} = \psi_s^L(t){}^s\ddot{U} + \psi_{s+\gamma\Delta t}^L(t){}^{s+\gamma\Delta t}\ddot{U} \end{cases} \quad (3-255)$$

上述关系式只是近似成立，亦即其残差一般情况下不为 0。故其速度和加速度的残差可以表达为

$$\begin{cases} r_1(t) = \dot{U}(t) - {}^t\dot{U} \\ r_2(t) = \ddot{U}(t) - {}^t\ddot{U} \end{cases} \quad (3-256)$$

在区间 $t_s \leq t \leq t_{s+\gamma\Delta t}$ 通过利用配点法可以得到如下表达式

$$\begin{cases} 0 = \int_{t_s}^{t_{s+\gamma\Delta t}} \delta(t - \theta_1\gamma\Delta t) r_1(t) dt \\ 0 = \int_{t_s}^{t_{s+\gamma\Delta t}} \delta(t - \theta_1\gamma\Delta t) r_2(t) dt \end{cases} \quad (3-257)$$

其中， θ_1 表示在第一个子步内的配点参数。文 [119] 建议其 θ_1 取值应满足 $0.5 \leq \theta_1 \leq 1$ 来获得无条件稳定性。

于是，经过计算可以得到其速度和加速度的更新方程

$${}^{t+\gamma\Delta t}\dot{U} = c_1 {}^{t+\gamma\Delta t}\dot{U} + c_2 {}^t\dot{U} + c_3 {}^t\ddot{U} \quad (3-258)$$

$${}^{t+\gamma\Delta t}\ddot{U} = c_1 {}^{t+\gamma\Delta t}\ddot{U} + c_2 {}^t\ddot{U} + c_3 {}^t\ddot{U} \quad (3-259)$$

其中，参数 c_1, c_2 和 c_3 取值为

$$c_1 = \frac{1}{\gamma\theta_1\Delta t} \quad c_2 = -\frac{1}{\gamma\theta_1\Delta t} \quad c_3 = \frac{\theta_1 - 1}{\theta_1} \quad (3-260)$$

需要说明的是，在第一个子步内的计算实质上等价于广义的 Trapezoidal 规则^[5]。

- 当 $\theta_1 = 0.5$ 时，上述算法将退化为 Trapezoidal 规则。
- 当 $\theta_1 = 1.0$ 时，上述算法将退化为向后 Euler 公式。

类似地，在第二个子步长内，Kim 等人利用了二次插值函数进行逼近计算，然后将残差再利用配点进行计算，进而得到如下的计算格式：

$${}^{t+\Delta t}\dot{U} = d_1 {}^{t+\Delta t}\dot{U} + d_2 {}^{t+\gamma\Delta t}\dot{U} + d_3 {}^t\dot{U} + d_4 {}^{t+\gamma\Delta t}\ddot{U} + d_5 {}^t\ddot{U} \quad (3-261)$$

$${}^{t+\Delta t}\ddot{U} = d_1 {}^{t+\Delta t}\ddot{U} + d_2 {}^{t+\gamma\Delta t}\ddot{U} + d_3 {}^t\ddot{U} + d_4 {}^{t+\gamma\Delta t}\ddot{U} + d_5 {}^t\ddot{U} \quad (3-262)$$

其中，系数取值分别为

$$\begin{cases} d_1 = \frac{\gamma - 2\theta_2}{\theta_2(\gamma - \theta_2)\Delta t} & d_2 = \frac{2\theta_2 - 1}{\gamma\theta_2(\gamma - \theta_2)\Delta t} \\ d_3 = \frac{(\gamma - 1)(\gamma + 1 - 2\theta_2)}{\gamma\theta_2(\gamma - \theta_2)\Delta t} \\ d_4 = \frac{\theta_2 - 1}{\gamma(\theta_2 - \gamma)} & d_5 = \frac{(\theta_2 - 1)(\gamma - 1)}{\gamma\theta_2} \end{cases} \quad (3-263)$$

当 CTFEMA 算法中的参数 $(\theta_1, \theta_2) = (0.5, 1.0)$ 时，该算法将退化为 TR-BDF2 算法^[107, 108]。特别地， $\theta_1 = 0.5$ 时，CTFEMA 算法可实现二阶精度。同时对于参数 γ 的取值，需要注意的是

- γ 应满足 $0.5 \leq \gamma \leq 1.0$ ，此时线性系统的稳定性不受其影响，但对于非线性系统，其取值应尽量避免接近 1.0。
- γ 合适的选择能提高数值阻尼的同时也能增加周期误差。
- 当求解线性系统时， γ 的选择能减小计算量。如， γ 的合适选择可以使得两个子步内的等效刚度矩阵一致。

3.6.9 TR-BDF2-Houbolt 算法

北京大学的魏文彬等人提出了一个三子步复合型算法^[120]。亦即将一个时间步长分为三个子步长 $p\Delta t, (1-2p)\Delta t, p\Delta t$ ，其中 $0 < p < 0.5$ 。在第一个时间子步长内，Trapezoidal 规则被雇用，即

$${}^{t+p\Delta t}\dot{U} = {}^t\dot{U} + \frac{1}{2}p\Delta t({}^t\ddot{U} + {}^{t+p\Delta t}\ddot{U}) \quad (3-264)$$

$${}^{t+p\Delta t}U = {}^tU + \frac{1}{2}p\Delta t({}^t\dot{U} + {}^{t+p\Delta t}\dot{U}) \quad (3-265)$$

结合在 $t + p\Delta t$ 时刻的平衡方程

$$M{}^{t+p\Delta t}\ddot{U} + C{}^{t+p\Delta t}\dot{U} + K{}^{t+p\Delta t}U = {}^{t+p\Delta t}F \quad (3-266)$$

在第二个子步内，使用了三点向后微分公式

$$\Delta t \cdot {}^{t+(1-p)\Delta t}\dot{U} = c_1 {}^tU + c_2 {}^{t+p\Delta t}U + c_3 {}^{t+(1-p)\Delta t}U \quad (3-267)$$

$$\Delta t \cdot {}^{t+(1-p)\Delta t}\ddot{U} = c_1 {}^t\dot{U} + c_2 {}^{t+p\Delta t}\dot{U} + c_3 {}^{t+(1-p)\Delta t}\dot{U} \quad (3-268)$$

其中，系数取值为

$$c_1 = \frac{2p-1}{p(p-1)} \quad c_2 = \frac{1-p}{p(2p-1)} \quad c_3 = \frac{2-3p}{(p-1)(2p-1)} \quad (3-269)$$

仍然使用在 $t + (1 - p)\Delta t$ 时刻的平衡方程

$$M^{t+(1-p)\Delta t}\ddot{U} + C^{t+(1-p)\Delta t}\dot{U} + K^{t+(1-p)\Delta t}U = {}^{t+(1-p)\Delta t}F \quad (3-270)$$

最后, 在第三个子步内使用 Houbolt 方法^[5,55] 进行计算 $t + \Delta t$ 时刻的响应值。

$$\Delta t \cdot {}^{t+\Delta t}\dot{U} = d_1 {}^tU + d_2 {}^{t+p\Delta t}U + d_3 {}^{t+(1-p)\Delta t}U + d_4 {}^{t+\Delta t}U \quad (3-271)$$

$$\Delta t \cdot {}^{t+\Delta t}\ddot{U} = d_1 {}^t\dot{U} + d_2 {}^{t+p\Delta t}\dot{U} + d_3 {}^{t+(1-p)\Delta t}\dot{U} + d_4 {}^{t+\Delta t}\dot{U} \quad (3-272)$$

系数 $d_i, i = 1, 2, 3, 4$ 的取值如下

$$d_1 = -1 \quad d_2 = \frac{1}{(p-1)(2p-1)} \quad d_3 = \frac{1}{p(2p-1)} \quad d_4 = \frac{p^2 - p - 1}{p(p-1)} \quad (3-273)$$

再结合 $t + \Delta t$ 时刻的平衡方程 (3-160)。

文 [120] 中数值试验表明, 就精度和数值耗散特性而言, 参数 $p = 1/3$ 是众多可取 p 值中最佳选择方案。

3.6.10 TR-NBDDF-N 算法

文 [121] 将三点向后微分公式看作牛顿向后插值^[19] 的特例, 则可以得到其更加一般化的复合型子步算法。当时间步长均分为两份时, 该算法退化为 TR-BDF2 算法。对于均分为 m 份的复合型子步算法, 可参见原文^[121]。这里仅显式地给出均分为 3 份的三子步复合型算法。在第一个子步内, 使用 Trapezoidal 规则, 则有

$${}^{t+\Delta t/3}\dot{U} = \frac{6}{\Delta t}({}^{t+\Delta t/3}U - {}^tU) - {}^t\dot{U} \quad (3-274)$$

$${}^{t+\Delta t/3}\ddot{U} = \frac{6}{\Delta t}({}^{t+\Delta t/3}\dot{U} - {}^t\dot{U}) - {}^t\ddot{U} \quad (3-275)$$

在第二个子步内, 使用

$${}^{t+2\Delta t/3}\dot{U} = \frac{3}{2\Delta t}{}^tU - \frac{6}{\Delta t}{}^{t+\Delta t/3}U + \frac{9}{2\Delta t}{}^{t+2\Delta t/3}U \quad (3-276)$$

$${}^{t+2\Delta t/3}\ddot{U} = \frac{27}{4\Delta t^2}{}^tU - \frac{27}{\Delta t^2}{}^{t+\Delta t/3}U + \frac{81}{4\Delta t^2}{}^{t+2\Delta t/3}U + \frac{9}{2\Delta t}{}^t\dot{U} - \frac{6}{\Delta t}{}^{t+\Delta t/3}\dot{U} \quad (3-277)$$

而在第三个子步内, 计算格式则为

$${}^{t+\Delta t}\dot{U} = -\frac{1}{\Delta t}{}^tU + \frac{9}{2\Delta t}{}^{t+\Delta t/3}U - \frac{9}{\Delta t}{}^{t+2\Delta t/3}U + \frac{11}{2\Delta t}{}^{t+\Delta t}U \quad (3-278)$$

$$\begin{aligned} {}^{t+\Delta t}\ddot{U} = & -\frac{11}{2\Delta t^2}{}^tU + \frac{11}{4\Delta t^2}{}^{t+\Delta t/3}U - \frac{99}{2\Delta t^2}{}^{t+2\Delta t/3}U + \frac{121}{4\Delta t^2}{}^{t+\Delta t}U \\ & - \frac{1}{\Delta t}{}^t\dot{U} + \frac{9}{2\Delta t}{}^{t+\Delta t/3}\dot{U} - \frac{9}{\Delta t}{}^{t+2\Delta t/3}\dot{U} \end{aligned} \quad (3-279)$$

三个子步结合三个时刻的平衡方程即可求解在 $t + \Delta t$ 时刻的状态量。

3.6.11 TR-RP 算法

Rezaiee-Pajand 提出了一个两子步复合型算法^[122]。在第一个时间步长内，使用 Trapezoidal 规则进行计算。而在第二个子步内，引入更具一般性的算法格式

$${}^{t+\Delta t}U = {}^tU + \Delta t[A{}^t\dot{U} + B{}^{t+\Delta t}\dot{U}] + \Delta t^2[C{}^t\ddot{U} + D{}^{t+\Delta t/2}\ddot{U}] \quad (3-280)$$

$${}^{t+\Delta t}\dot{U} = {}^t\dot{U} + \Delta t[E{}^t\ddot{U} + F{}^{t+\Delta t/2}\ddot{U} + G{}^{t+\Delta t}\ddot{U}] \quad (3-281)$$

若满足二阶精度，则要求下列等式成立

$$\begin{cases} A + B = 1 & B + C + D = \frac{1}{2} \\ E + F + G = 1 & \frac{1}{2}F + G = \frac{1}{2} \end{cases} \quad (3-282)$$

于是，上述等式 (3-280)-(3-281) 整理为两参数算法

$${}^{t+\Delta t}U = {}^tU + \Delta t[\alpha{}^t\dot{U} + (1 - \alpha){}^{t+\Delta t}\dot{U}] + \Delta t^2[\beta{}^t\ddot{U} + (\alpha - \beta - \frac{1}{2}){}^{t+\Delta t/2}\ddot{U}] \quad (3-283)$$

$${}^{t+\Delta t}\dot{U} = {}^t\dot{U} + \Delta t[(1 - \alpha){}^t\ddot{U} + (2\alpha - 1){}^{t+\Delta t/2}\ddot{U} + (1 - \alpha){}^{t+\Delta t}\ddot{U}] \quad (3-284)$$

其中，参数 α, β 的选取依据为

$$\alpha = 2(\beta - 0.125) + 0.75 \quad 0 < \beta < 0.125 \quad 0.5 < \alpha < 0.75 \quad (3-285)$$

以满足算法实现

- ‡ 对线性系统无条件稳定。
- ‡ 复合型算法实现二阶精度的同时，实现最小化低频、最大化高频耗散。
- ‡ 新算法较 Trapezoidal 规则^[5] 和 TR-BDF2 算法^[107,108,123] 精度更高。
- ‡ 对 Trapezoidal 算法求解失败的非线性问题，新算法仍保持稳定性。

3.7 复合型子步显式算法

3.7.1 Noh-Bathe 两子步算法

K.J. Bathe 等人^[124] 提出了一个两子步复合型显式算法来求解波的传播问题。在第一个子步长 $p\Delta t$ 内，使用

$$M{}^{t+p\Delta t}\ddot{U} + C{}^{t+p\Delta t}\dot{\tilde{U}} + K{}^{t+p\Delta t}U = {}^{t+p\Delta t}F \quad (3-286)$$

$${}^{t+p\Delta t}U = {}^tU + (p\Delta t){}^t\dot{U} + \frac{1}{2}(p\Delta t)^2 \cdot {}^t\ddot{U} \quad (3-287)$$

$${}^{t+p\Delta t}\tilde{\ddot{U}} = (1-s)[{}^t\dot{U} + \frac{1}{2}(p\Delta t){}^t\ddot{U}] + s \cdot {}^t\dot{U} \quad (3-288)$$

$${}^{t+p\Delta t}\dot{U} = {}^t\dot{U} + \frac{1}{2}(p\Delta t)({}^t\ddot{U} + {}^{t+p\Delta t}\ddot{U}) \quad (3-289)$$

非常有意思的是，这里的带入平衡方程的速度更新公式并不是最后获得速度响应值的表达式。这二者分别由不同的速度更新公式计算。同时表达式 ${}^{t+p\Delta t}F$ 使用 Trapezoidal 规则^[77] 进行计算。

在第二个子步内，使用

$${}^{t+\Delta t}F = M{}^{t+\Delta t}\ddot{U} + C{}^{t+\Delta t}\dot{U} + K{}^{t+\Delta t}U \quad (3-290)$$

$${}^{t+\Delta t}U = {}^{t+p\Delta t}U + [(1-p)\Delta t]{}^{t+p\Delta t}\dot{U} + \frac{1}{2}[(1-p)\Delta t]^2 \cdot {}^{t+p\Delta t}\ddot{U} \quad (3-291)$$

$${}^{t+\Delta t}\tilde{\ddot{U}} = (1-s)[{}^{t+p\Delta t}\dot{U} + \frac{1}{2}[(1-p)\Delta t]{}^{t+p\Delta t}\ddot{U}] + s \cdot {}^{t+p\Delta t}\dot{U} \quad (3-292)$$

$${}^{t+\Delta t}\dot{U} = {}^{t+p\Delta t}\dot{U} + \frac{1}{2}[(1-p)\Delta t]{}^{t+p\Delta t}\ddot{U} + [(1-p)\Delta t](q_0{}^t\ddot{U} + q_1{}^{t+p\Delta t}\ddot{U} + q_2{}^{t+\Delta t}\ddot{U}) \quad (3-293)$$

为了在有无结构阻尼情况都实现二阶精度，上述参数应该满足

$$q_0 + q_1 + q_2 = \frac{1}{2} \quad q_2 = \frac{1}{2} - pq_1 \quad s = -1 \quad (3-294)$$

使用 Routh-Hurwitz 稳定性准则^[5] 可以得到其最大的稳定性极限值

$$\Omega_s^2 = \frac{1}{\gamma p(1-p)} \quad (3-295)$$

其中

$$\gamma = \frac{1}{4} - \frac{1}{2}(1-p)q_1 \quad -\frac{1}{4(1-p)} \leq q_1 \leq \frac{4p(1-p)-1}{8p(1-p)^2} \quad (3-296)$$

特别地，当 $p = 0.5, q_1 = 0$ 时，其稳定性极限为 $\Omega = 4$ ，是中心差分法^[5] 的稳定性极限 ($\Omega_s = 2$) 的两倍。

为了使得分岔点处的频率值尽可能大，参数 q_1 需满足

$$q_1 = \frac{1-2p}{2p(1-p)} \quad (3-297)$$

此时的分岔点处的频率值 Ω_b ^①和稳定性极限 Ω_s 最大，分别为

$$\Omega_b = \sqrt{\frac{2}{p(1-p)}} \quad \Omega_s = \frac{2}{\sqrt{(1-p)(1-3p)}} \quad (3-298)$$

① 文 [124] 中计算结果为 $\Omega_b = \frac{2}{p}$ ，有误。

同时, 根据不等式关系 (3-296) 可得 p 的取值范围为

$$\frac{1}{2} \leq p \leq \frac{2}{3} \quad (3-299)$$

进一步, 若使用分岔点处的谱半径值 ρ_b 来刻画显式算法的耗散性能, 则有下列关系^①成立

$$p = \frac{1}{1 + \rho_b} \quad (3-300)$$

由等式 (3-299) 可知, 分岔点处的谱半径值应满足 $0.5 \leq \rho_b \leq 1$ 时。当 $\rho_b = 1$ 和 0.5, 该算法分别实现无耗散和最大的耗散特性, 但不能实现 $\rho_b = 0$ 的渐进消弭特性。

于是该复合型子步的显式算法实际为单参数算法 (p 或 ρ_b)。各参数取值如下

$$\begin{cases} s = -1 \\ q_0 = \frac{2p-1}{2p} \\ q_1 = \frac{1-2p}{2p(1-p)} \\ q_2 = \frac{p}{2(1-p)} \end{cases} \quad (3-301)$$

3.7.2 Kwon-Lee 四子步算法

Kwon 和 Lee 等人^[125], 基于四阶的中心差分格式和一个高阶导数项去最小化高频振动解, 提出了一阶精度的四子步显式复合型算法。该算法将时间步长 Δt 分为相等的四个子步长 $\Delta t/4$ 。

在第一个子步内使用

$$\begin{aligned} M^{t+\Delta t/4}\ddot{U} + C^{t+\Delta t/4}\dot{U} + K^{t+\Delta t/4}U &= {}^{t+\Delta t/4}\hat{R} \\ {}^{t+\Delta t/4}U &= {}^tU + \frac{\Delta t}{4}{}^t\dot{U} + \frac{1}{2}\left(\frac{\Delta t}{4}\right)^2{}^t\ddot{U} \\ {}^{t+\Delta t/4}\dot{U} &= {}^t\dot{U} + \frac{\Delta t}{4}{}^t\ddot{U} \end{aligned} \quad (3-302)$$

在第二个子步内则使用

① 文 [124] 中结果为 $p = \frac{2-\sqrt{2+2\rho_b}}{1-\rho_b}$, 有误, 原因是 Ω_b 的计算错误导致。

$$\begin{aligned}
 M^{t+\Delta t/2}\ddot{U} + C^{t+\Delta t/2}\dot{U} + K^{t+\Delta t/2}U &= {}^{t+\Delta t/2}\hat{R} \\
 {}^{t+\Delta t/2}U &= {}^tU + \frac{\Delta t}{2}{}^{t+\Delta t/4}\dot{U} + \frac{1}{2}\left(\frac{\Delta t}{2}\right)^2{}^{t+\Delta t/4}\ddot{U} \\
 {}^{t+\Delta t/2}\dot{U} &= {}^t\dot{U} + \frac{\Delta t}{2}{}^{t+\Delta t/4}\ddot{U}
 \end{aligned} \tag{3-303}$$

在第三个子步内则使用

$$\begin{aligned}
 M^{t+3\Delta t/4}\ddot{U} + C^{t+3\Delta t/4}\dot{U} + K^{t+3\Delta t/4}U &= {}^{t+3\Delta t/4}\hat{R} \\
 {}^{t+3\Delta t/4}U &= {}^{t+\Delta t/4}U + \frac{\Delta t}{2}{}^{t+\Delta t/2}\dot{U} + \frac{1}{2}\left(\frac{\Delta t}{2}\right)^2{}^{t+\Delta t/2}\ddot{U} \\
 {}^{t+3\Delta t/4}\dot{U} &= {}^{t+\Delta t/4}\dot{U} + \frac{\Delta t}{2}{}^{t+\Delta t/2}\ddot{U}
 \end{aligned} \tag{3-304}$$

最后，在第四个子步内使用

$$\begin{aligned}
 M^{t+\Delta t}\ddot{U} + C^{t+\Delta t}\dot{U} + K^{t+\Delta t}U &= {}^{t+\Delta t}R \\
 {}^{t+\Delta t}U &= 8{}^{t+3\Delta t/4}U - 8{}^{t+\Delta t/4}U + {}^tU - 3\Delta t{}^{t+\Delta t/2}\dot{U} + \alpha{}^{t+\Delta t/2}U^{(4)}(\Delta t)^4 \\
 {}^{t+\Delta t}\dot{U} &= 8{}^{t+3\Delta t/4}\dot{U} - 8{}^{t+\Delta t/4}\dot{U} + {}^t\dot{U} - 3\Delta t{}^{t+\Delta t/2}\ddot{U} + \alpha{}^{t+\Delta t/2}U^{(5)}(\Delta t)^4
 \end{aligned} \tag{3-305}$$

其中 $U^{(i)}$ 表示第 i 阶导数值。参数 α 控制算法的稳定性和精度，其取值为 $0.0409 \leq \alpha \leq 0.045$ 。同时 ${}^{t+\Delta t/2}U^{(4)}$ 和 ${}^{t+\Delta t/2}U^{(5)}$ 分别由

$${}^{t+\Delta t/2}U^{(4)} = \frac{{}^{t+3\Delta t/4}\dot{U} - 3{}^{t+\Delta t/2}\dot{U} + 3{}^{t+\Delta t/4}\dot{U} - {}^t\dot{U}}{(\Delta t)^3/32} \tag{3-306}$$

$${}^{t+\Delta t/2}U^{(5)} = \frac{{}^{t+3\Delta t/4}\ddot{U} - 3{}^{t+\Delta t/2}\ddot{U} + 3{}^{t+\Delta t/4}\ddot{U} - {}^t\ddot{U}}{(\Delta t)^3/32} \tag{3-307}$$

上述算法中的外荷载项在 $t + i\Delta/4, i = 1, 2, 3, 4$ 时刻的计算公式如下

$$\begin{cases} {}^{t+\Delta t/4}\hat{R} = \frac{1}{4}(3{}^tR + {}^{t+\Delta t}R) \\ {}^{t+\Delta t/2}\hat{R} = \frac{1}{2}({}^tR + {}^{t+\Delta t}R) \\ {}^{t+3\Delta t/4}\hat{R} = \frac{1}{4}({}^tR + 3{}^{t+\Delta t}R) \end{cases} \tag{3-308}$$

3.8 本章小结

第 4 章 基于广义线性法的结构动响应数值算法设计

4.1 本章小结

第 5 章 数值算例

5.1 本章小结

结 论

学位论文的结论作为论文正文的最后一章单独排写，但不加章标题序号。

结论应是作者在学位论文研究过程中所取得的创新性成果的概要总结，不能与摘要混为一谈。博士学位论文结论应包括论文的主要结果、创新点、展望三部分，在结论中应概括论文的核心观点，明确、客观地指出本研究内容的创新性成果（含新见解、新观点、方法创新、技术创新、理论创新），并指出今后进一步在本研究方向进行研究工作的展望与设想。对所取得的创新性成果应注意从定性和定量两方面给出科学、准确的评价，分（1）、（2）、（3）…条列出，宜用“提出了”、“建立了”等词叙述。

参考文献

- [1] 杨超, 肖守讷, 阳光武, 等. 一类非耗散的显式时间积分方法 [J]. 振动工程学报, 2015, 28(3): 441–448.
- [2] Butcher J C. On the convergence of numerical solutions to ordinary differential equations[J]. Mathematics of Computation, 1966, 20(93): 1–10.
- [3] 于开平, 邹经湘. 结构动力响应数值算法耗散和超调特性设计 [J]. 力学学报, 2005, 37(4): 467–476.
- [4] Dahlquist G. Convergence and stability in numerical integration of ordinary differential equations[J]. Mathematica Scandinavica, 1956, 19(4): 33–53.
- [5] Hughes T J R. Dover Civil and Mechanical Engineering: The Finite Element Method: Linear Static and Dynamic Finite Element Analysis[M]. [S.l.]: Dover Publications, 2000.
- [6] 周树荃, 高科华. 解结构动力问题的半隐式 Runge—Kutta 型并行直接积分法 [J]. 计算物理, 1992, 9(02): 133–138.
- [7] Lunk C, Simeon B. Runge-Kutta-Nyström-Methods with Maximized Stability Domain for Stiff Mechanical Systems[M]. [S.l.]: Springer Berlin Heidelberg, 2004: 139–144.
- [8] Lunk C, Simeon B. Runge-Kutta-Nyström methods with maximized stability domain in structural dynamics[J], 2005: 373–389.
- [9] Dopico D, Lugris U, Gonzalez M, et al. Two implementations of IRK integrators for realtime multibody dynamics[J]. International Journal for Numerical Methods in Engineering, 2010, 65(12): 2091–2111.
- [10] 吴志桥, 高普云, 任钧国. Runge-Kutta 方法求解结构动力学方程 [J]. 系统仿真学报, 2010(9): 2085–2090.
- [11] 吴志桥. L-稳定格式求解结构动力学方程和多体系统动力学方程 [D]. 长沙: 国防科学技术大学, 2009.
- [12] Yin S-H. A new explicit time integration method for structural dynamics[J/OL]. International Journal of Structural Stability and Dynamics, 2013, 13(3). <http://dx.doi.org/10.1142/S021945541250068X>.

- [13] 郭静, 邢誉峰. 结构动力学方程的辛 RK 方法 [J]. 应用数学和力学, 2014, 35(01): 12–21.
- [14] 黄策, 富明慧, 郑彬彬. 求解结构动力学方程的一种辛格式及其优化 [J]. 固体力学学报, 2016(s1): 45–49.
- [15] Burrage K, Butcher J C. Non-linear stability of a general class of differential equation methods[J/OL]. BIT Numerical Mathematics, 1980, 20(2): 185–203. <https://doi.org/10.1007/BF01933191>.
- [16] Butcher J C, Hill A T. Linear Multistep Methods as Irreducible General Linear Methods[J]. Bit Numerical Mathematics, 2006, 46(1): 5–19.
- [17] Skeel R D. Equivalent forms of multistep formulas[J]. Mathematics of Computation, 1979, 33(148): 1229–1250.
- [18] Curtiss C F, Hirschfelder J O. Integration of Stiff Equations[J]. Proceedings of the National Academy of Sciences of the United States of America, 1952, 38(3): 235–243.
- [19] Gear C W. Numerical initial value problems in ordinary differential equations[M]. [S.l.]: Prentice Hall, 1971.
- [20] Butcher J C. Numerical Methods for Ordinary Differential Equations[M]. 2nd ed. [S.l.]: Wiley, 2008.
- [21] Lapidus L, Seinfeld J H. Mathematics in Science and Engineering 74: Numerical Solution of Ordinary Differential Equations[M]. New York: Academic Press, 1971.
- [22] Gerhard Wanner Ernst Hairer S P N. Springer Series in Computational Mathematics 8: Solving Ordinary Differential Equations I: Nonstiff Problems[M]. 2nd ed. [S.l.]: Springer-Verlag Berlin Heidelberg, 1993.
- [23] Ernst Hairer G W. Springer Series in Computational Mathematics 14: Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems[M]. 2nd ed. [S.l.]: Springer-Verlag Berlin Heidelberg, 1996.
- [24] 李寿佛. 刚性常微分方程及刚性泛函微分方程数值分析 [M]. [S.l.]: 湘潭大学出版社, 2010.
- [25] Lambert J D. Computational Methods in Ordinary Differential Equations[M]. 1st ed. [S.l.]: John Wiley & Sons Inc, 1973.

- [26] Butcher J C. The numerical analysis of ordinary differential equations: Runge-Kutta and general linear methods[M]. [S.l.] : John Wiley and Sons, 1987.
- [27] Dahlquist G. Error analysis for a class of methods for stiff non-linear initial value problems[M] // Watson G. . Berlin : Springer, 1976 : 60 – 72.
- [28] Dahlquist G. On One-Leg Multistep Methods[J]. Siam Journal on Numerical Analysis, 1983, 20(6) : 1130 – 1138.
- [29] Hundsdorfer W H, Steiner B I. Convergence of linear multistep and one-leg methods for stiff nonlinear initial value problems[J]. Bit Numerical Mathematics, 1991, 31(1) : 124 – 143.
- [30] Cash J R. On the integration of stiff systems of O.D.E.s using extended backward differentiation formulae[J]. Numerische Mathematik, 1980, 34(3) : 235 – 246.
- [31] Cash J R. The integration of stiff initial value problems in ODEs using modified extended backward differentiation formulae[J]. Computers & Mathematics with Applications, 1983, 9(5) : 645 – 657.
- [32] Jackiewicz Z, Tracogna S. A General Class of Two-Step Runge-Kutta Methods for Ordinary Differential Equations[J]. Siam Journal on Numerical Analysis, 1995, 32(5) : 1390 – 1427.
- [33] Burrage K, Sharp P W. A class of variable-step explicit Nordsieck multivalue methods[J]. SIAM Journal on Numerical Analysis, 1994 : 1434 – 1451.
- [34] Burrage K. High order algebraically stable Runge-Kutta methods[J]. BIT, 1978, 18(4) : 373 – 383.
- [35] Burrage K. Order Properties of Implicit Multivalue Methods for Ordinary Differential Equations[J]. SIAM Journal of Numerical Analysis, 1988, 8(1) : 43 – 69.
- [36] Schmitt B A, Weiner R. Parallel Two-Step W-Methods with Peer Variables[J]. Siam Journal on Numerical Analysis, 2004, 42(1) : 265 – 282.
- [37] Burrage K. Parallel and sequential methods for ordinary differential equations[M]. [S.l.] : Clarendon Press ;, 1995.
- [38] Jackiewicz Z. General linear methods for ordinary differential equations[M]. First Edition. [S.l.] : Wiley, 2009.
- [39] Rattenbury N. Almost Runge-Kutta Methods for Stiff and Non-Stiff Problems[D]. New Zealand : University of Auckland, 2005.

- [40] Huang S J Y. Implementation of general linear methods for stiff differential equations[D]. New Zealand : University of Auckland, 2005.
- [41] Lee J H J. Numerical methods for ordinary differential equations: A survey of some standard methods[D]. New Zealand : University of Auckland, 2004.
- [42] Oliver J. A Curiosity of Low-Order Explicit Runge-Kutta Methods[J]. Mathematics of Computation, 1975, 29(132): 1032 – 1036.
- [43] Zlatev Z. Modified Diagonally Implicit Runge-Kutta Methods[J]. Siam Journal on Scientific & Statistical Computing, 1980, 2(3): 321 – 334.
- [44] Dekker K, Verwer J G. Stability of Runge-Kutta methods for stiff nonlinear differential equations[J]. Mathematics of Computation, 1984, 47(47).
- [45] 袁新鼎, 费景高, 刘德贵. 刚性常微分方程初值问题的数值解法 [M]. [S.l.]: 科学出版社, 1987: 19 – 20.
- [46] Dahlquist G G. A special stability problem for linear multistep methods[J/OL]. BIT Numerical Mathematics, 1963, 3(1): 27 – 43.
<http://dx.doi.org/10.1007/BF01963532>.
- [47] Wanner G. Dahlquist's classical papers on stability theory[J/OL]. BIT Numerical Mathematics, 2006, 46(3): 671 – 683.
<http://dx.doi.org/10.1007/s10543-006-0072-1>.
- [48] Widlund O B. A note on unconditionally stable linear multistep methods[J]. Bit Numerical Mathematics, 1967, 7(1): 65 – 70.
- [49] Cryer C W. A new class of highly-stable methods: A_0 -stable methods.[J]. Bit Numerical Mathematics, 1973, 13(2): 153 – 159.
- [50] Fatunla S O, Rheinboldt W, Siewiorek D. Computer Science and Scientific Computing: Numerical Methods for Initial Value Problems in Ordinary Differential Equations[M]. [S.l.]: Elsevier Inc, Academic Press, 1988.
- [51] Hoff C, Pahl P J. Development of an implicit method with numerical dissipation for time integration algorithms in structural dynamics[J]. Computer Methods in Applied Mechanics & Engineering, 1988, 67(3): 367 – 385.
- [52] Hoff C, Pahl P. Practical performance of the θ_1 -method and comparison with other dissipative algorithms in structural dynamics[J/OL]. Computer Methods in Applied Mechanics and Engineering, 1988, 67(1): 87 – 110.
<http://www.sciencedirect.com/science/article/pii/0045782588900709>.

- [53] Newmark N M. A method of computation for structural dynamics[J]. J. Engrg. Mech. Div., 1959, 85 : 67.
- [54] Wilson E L. A Computer Program for the Dynamic Stress Analysis of Underground Structure[J]. Dynamic Loads, 1968.
- [55] Chopra A K. Prentice-Hall International Series in Civil Engineering and Engineering Mechanics : Dynamics of Structures: Theory and applications to earthquake engineering[M]. 4th ed. [S.l.] : Prentice Hall, 2011.
- [56] Wood W L. Practical time-stepping schemes.[M]. [S.l.] : Oxford: Clarendon, 1990.
- [57] Zienkiewicz O C, Taylor R L. The finite element method : Vol 3[M]. 4th ed. New York : McGraw-hill London, 1991.
- [58] Dokainish M, Subbaraj K. A survey of direct time-integration methods in computational structural dynamics—I. Explicit methods[J/OL]. Computers & Structures, 1989, 32(6) : 1371 – 1386.
<http://www.sciencedirect.com/science/article/pii/0045794989903143>.
- [59] Subbaraj K, Dokainish M. A survey of direct time-integration methods in computational structural dynamics—II. Implicit methods[J/OL]. Computers & Structures, 1989, 32(6) : 1387 – 1401.
<http://www.sciencedirect.com/science/article/pii/0045794989903155>.
- [60] Fung T, Leung A. On the accuracy of discontinuous Galerkin methods in the time domain[J]. JVC/Journal of Vibration and Control, 1996, 2(2) : 193 – 217.
- [61] Fung T. Unconditionally stable higher-order accurate hermitian time finite elements[J/OL]. International Journal for Numerical Methods in Engineering, 1996, 39(20) : 3475 – 3495.
[http://dx.doi.org/10.1002/\(SICI\)1097-0207\(19961030\)39:20<3475::AID-NME10>3.0.CO;2-H](http://dx.doi.org/10.1002/(SICI)1097-0207(19961030)39:20<3475::AID-NME10>3.0.CO;2-H).
- [62] Geradin M. A classification and discussion of integration operators for transient structural response[J]. AIAA Journal, 1974, 108(6) : 1868 – 1876.
- [63] 张雄, 王天舒, 刘岩. 计算动力学 [M]. 2nd ed. 北京 : 清华大学出版社, 2007.
- [64] Har J, Tamma K K. Advances in Computational Dynamics of Particles, Materials and Structures[M/OL]. [S.l.] : Wiley, 2012.
<http://gen.lib.rus.ec/book/index.php?md5=534447fc6336c941295a051b4b4cf1e3>.

- [65] Lees M. A Linear Three-Level Difference Scheme for Quasilinear Parabolic Equations[J/OL]. *Mathematics of Computation*, 1966, 20(96): 516–522.
<http://www.jstor.org/stable/2003541>.
- [66] Zienkiewicz O C. A new look at the Newmark, Houbolt and other time stepping formulas: A weighted residual approach[J/OL]. *Earthquake Engineering & Structural Dynamics*, 1977, 5(4): 413–418.
<http://dx.doi.org/10.1002/eqe.4290050407>.
- [67] Houbolt J C. A Recurrence Matrix Solution for the Dynamic Response of Elastic Aircraft[J]. *Journal of the Aeronautical Sciences*, 1950, 17(9): 540–550.
- [68] Hoff C, Taylor R L. Higher derivative explicit one step methods for non-linear dynamic problems. Part I: Design and theory[J/OL]. *International Journal for Numerical Methods in Engineering*, 1990, 29(2): 275–290.
<http://dx.doi.org/10.1002/nme.1620290205>.
- [69] Hoff C, Taylor R L. Higher derivative explicit one step methods for non-linear dynamic problems. Part II: Practical calculations and comparisons with other higher order methods[J/OL]. *International Journal for Numerical Methods in Engineering*, 1990, 29(2): 291–301.
<http://dx.doi.org/10.1002/nme.1620290206>.
- [70] Hilber H M, Hughes T J R. Collocation, dissipation and ‘overshoot’ for time integration schemes in structural dynamics[J/OL]. *Earthquake Engineering & Structural Dynamics*, 1978, 6(1): 99–117.
<http://dx.doi.org/10.1002/eqe.4290060111>.
- [71] Hilber H M, Hughes T J R, Taylor R L. Improved numerical dissipation for time integration algorithms in structural dynamics[J/OL]. *Earthquake Engineering & Structural Dynamics*, 1977, 5(3): 283–292.
<http://dx.doi.org/10.1002/eqe.4290050306>.
- [72] Wood W, Bossak M, Zienkiewicz O. An alpha modification of Newmark’s method[J/OL]. *International Journal for Numerical Methods in Engineering*, 1980, 15(10): 1562–1566.
<http://dx.doi.org/10.1002/nme.1620151011>.

- [73] Yu K. A new family of generalized- α time integration algorithms without overshoot for structural dynamics[J/OL]. *Earthquake Engineering & Structural Dynamics*, 2008, 37(12): 1389 – 1409.
<http://dx.doi.org/10.1002/eqe.818>.
- [74] Zhou X, Tamma K K. Design, analysis, and synthesis of generalized single step single solve and optimal algorithms for structural dynamics[J]. *International Journal for Numerical Methods in Engineering*, 2004, 59(5): 597–668.
- [75] Mohammadzadeh S, Ghassemieh M, Park Y. Structure-dependent improved Wilson- method with higher order of accuracy and controllable amplitude decay[J/OL]. *Applied Mathematical Modelling*, 2017, 52: 417 – 436.
<http://www.sciencedirect.com/science/article/pii/S0307904X17305061>.
- [76] Benítez J M, Montáns F J. The value of numerical amplification matrices in time integration methods[J]. *Computers & Structures*, 2013, 128(5): 243 – 250.
- [77] Erlicher S, Bonaventura L, Bursi O S. The analysis of the Generalized- α method for non-linear dynamic problems[J]. *Computational Mechanics*, 2002, 28(2): 83 – 104.
- [78] Tamma K K, Namburu R R. A new finite element based Lax-Wendroff/Taylor-Galerkin methodology for computational dynamics[J]. *Computer Methods in Applied Mechanics & Engineering*, 1988, 71(2): 137 – 150.
- [79] Tamma K K, Namburu R R. An explicit velocity based Lax-Wendroff/Taylor-Galerkin methodology of computation for the dynamics of structures[J]. *Computers & Structures*, 1988, 30(5): 1017 – 1024.
- [80] Tamma K K, Namburu R R. A new unified explicit architecture of thermal/structural dynamic algorithms: Applications to coupled thermoelasticity[J]. *Computers & Structures*, 1989, 37(4): 535 – 545.
- [81] Tamma K K, Namburu R R. A robust selfstarting explicit computational methodology for structural dynamic applications: Architecture and representations[J]. *International Journal for Numerical Methods in Engineering*, 1990, 29(7): 1441 – 1454.
- [82] Tamma K K, Namburu R R. Applicability and evaluation of an implicit self-starting unconditionally stable methodology for the dynamics of structures[J]. *Computers & Structures*, 1990, 34(6): 835 – 842.

- [83] Tamma K K, D'Costa J F. A new explicit variable time-integration self-starting methodology for computational structural dynamics[J]. International Journal for Numerical Methods in Engineering, 1992, 33(6): 1165 – 1180.
- [84] Namburu R R, Tamma K K. A Generalized γ_s -family of self-starting algorithms for computational structural dynamics[J]. AIAA Journal, 1992.
- [85] Simo J C, Tarnow N. The discrete energy-momentum method: conserving algorithms for nonlinear elastodynamics[M]. [S.l.]: Birkhauser Verlag, 1992: 757 – 792.
- [86] Tamma K, Maocheng L I, Namburu R. Evaluation and applicability of explicit self-starting formulations for nonlinear structural dynamics[C] //34th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics & Materials Conference. 1993.
- [87] Chung J, Hulbert G M. A time integration algorithm for structural dynamics with improved numerical dissipation: The generalized- α method[J]. Journal of Applied Mechanics, 1993, 60: 371.
- [88] 邵慧萍, 蔡承文. 结构动力学方程数值积分的三参数算法 [J]. 应用力学学报, 1988(4): 78 – 83+139.
- [89] Hulbert G M, Chung J. The unimportance of the spurious root of time integration algorithms for structural dynamics[J/OL]. Communications in Numerical Methods in Engineering, 1994, 10(8): 591 – 597.
<http://dx.doi.org/10.1002/cnm.1640100803>.
- [90] Bazzi G, Anderheggen E. The ρ -family of algorithms for time-step integration with improved numerical dissipation[J/OL]. Earthquake Engineering & Structural Dynamics, 1982, 10(4): 537 – 550.
<http://dx.doi.org/10.1002/eqe.4290100404>.
- [91] Zienkiewicz O C, Wood W L, Hine N W, et al. A unified set of single step algorithms. Part 1: General formulation and applications[J/OL]. International Journal for Numerical Methods in Engineering, 1984, 20(8): 1529 – 1552.
<http://dx.doi.org/10.1002/nme.1620200814>.
- [92] Wood W L. A unified set of single step algorithms. Part 2: Theory[J/OL]. International Journal for Numerical Methods in Engineering, 1984, 20(12): 2303 – 2309.
<http://dx.doi.org/10.1002/nme.1620201210>.

- [93] Kujawski J, Gallagher R. A family of higher order explicit algorithms for transient dynamic analysis[J]. Transactions of the Society for Computer Simulation, 1984, 1 : 155 – 166.
- [94] Liu T, Zhao C, Li Q, et al. An efficient backward Euler time-integration method for nonlinear dynamic analysis of structures[J]. Computers & Structures, 2012, 106-107(5) : 20 – 28.
- [95] Liu T Y, Qingbin L I, Zhao C B. An efficient time-integration method for nonlinear dynamic analysis of solids and structures[J]. Science China, 2013, 56(4) : 798 – 804.
- [96] Zhang L, Liu T, Li Q. A Robust and Efficient Composite Time Integration Algorithm for Nonlinear Structural Dynamic Analysis[J]. Mathematical Problems in Engineering, 2015, (2015-5-20), 2015, 2015(1-4) : 1 – 11.
- [97] Soares Jr D. A simple and effective new family of time marching procedures for dynamics[J]. Computer Methods in Applied Mechanics & Engineering, 2014, 283 : 1138 – 1166.
- [98] Soares Jr D. An explicit family of time marching procedures with adaptive dissipation control[J]. International Journal for Numerical Methods in Engineering, 2014, 100(3) : 165 – 182.
- [99] Grosseholz G, Soares Jr D, von Estorff O. A stabilized central difference scheme for dynamic analysis[J/OL]. International Journal for Numerical Methods in Engineering, 2015, 102(11) : 1750 – 1760.
<http://dx.doi.org/10.1002/nme.4869>.
- [100] Soares Jr D. An implicit family of time marching procedures with adaptive dissipation control[J]. Applied Mathematical Modelling, 2015, 40(4) : 3325 – 3341.
- [101] Soares Jr D. A Second-order Time-marching Procedure with Enhanced Accuracy[J]. Computer Modeling in Engineering & Sciences, 2015, 105(5) : 341 – 360.
- [102] Soares Jr D. A simple and effective singlestep time marching technique based on adaptive time integrators[J]. International Journal for Numerical Methods in Engineering, 2016.
- [103] Soares Jr D. A novel family of explicit time marching techniques for structural dynamics and wave propagation models[J/OL]. Computer Methods in Applied Mechanics & Engineering, 2016, 311 : 838 – 855.
<http://dx.doi.org/10.1016/j.cma.2016.09.021>.

- [104] Quarteroni A, Sacco R, Saleri F. Numerical mathematics: Vol 37[M]. [S.l.]: Springer Science & Business Media, 2010.
- [105] Tarnow N, Simo J. How to render second order accurate time-stepping algorithms fourth order accurate while retaining the stability and conservation properties[J/OL]. Computer Methods in Applied Mechanics and Engineering, 1994, 115(3): 233 – 252.
<http://www.sciencedirect.com/science/article/pii/0045782594900612>.
- [106] Fung T C. Complex-time-step Newmark methods with controllable numerical dissipation[J/OL]. International Journal for Numerical Methods in Engineering, 1998, 41(1): 65 – 93.
[http://dx.doi.org/10.1002/\(SICI\)1097-0207\(19980115\)41:1<65::AID-NME270>3.0.CO;2-F](http://dx.doi.org/10.1002/(SICI)1097-0207(19980115)41:1<65::AID-NME270>3.0.CO;2-F).
- [107] Bathe K J, Baig M M I. On a composite implicit time integration procedure for nonlinear dynamics[J]. Computers & Structures, 2005, 83(31-32): 2513 – 2524.
- [108] Bathe K J. Conserving energy and momentum in nonlinear dynamics: A simple implicit time integration scheme[J]. Computers & Structures, 2007, 85(7 – 8): 437 – 445.
- [109] Silva W T M, Bezerra L M. Performance of Composite Implicit Time Integration Scheme for Nonlinear Dynamic Analysis[J]. Mathematical Problems in Engineering, 2008, 2008(4): 267 – 290.
- [110] Zhang J, Liu Y, Liu D. Accuracy of a composite implicit time integration scheme for structural dynamics[J/OL]. International Journal for Numerical Methods in Engineering, 2017, 109(3): 368 – 406.
<http://dx.doi.org/10.1002/nme.5291>.
- [111] Dharmaraja S, Wang Y, Strang G. Optimal stability for trapezoidal – backward difference split-steps[J]. Ima Journal of Numerical Analysis, 2009, 30(1): 141 – 148.
- [112] Klarmann S, Wagner W. Enhanced studies on a composite time integration scheme in linear and non-linear dynamics[J]. Computational Mechanics, 2015, 55(3): 455 – 468.

- [113] Kumar S, Gautam S S. Extension of A Composite Time Integration Scheme for Dynamic Problems[J], .
- [114] Guwahati. Analysis of A Composite Time Integration Scheme[J], .
- [115] Dong S. BDF-like methods for nonlinear dynamic analysis[J]. Journal of Computational Physics, 2010, 229(8): 3019 – 3045.
- [116] Chandra Y, Zhou Y, Stanciulescu I, et al. A robust composite time integration scheme for snap-through problems[J]. Computational Mechanics, 2015, 55(5): 1041 – 1056.
- [117] Dong S. Blended Time Integration Algorithms for Nonlinear Dynamics[C] // Aiaa/asme/asce/ahs/asc Structures, Structural Dynamics, and Materials Conference Aiaa/asme/ahs Adaptive Structures Conference. 2013.
- [118] Park K C. An improved stiffly stable method for direct integration of nonlinear structural dynamic equations[J]. Journal of Applied Mechanics, 1975, 42: 464.
- [119] Kim W, Reddy J N. An Improved Time Integration Algorithm: A Collocation Time Finite Element Approach[J]. International Journal of Structural Stability & Dynamics, 2016, 17(2): 1750024.
- [120] Wen W B, Wei K, Lei H S, et al. A novel sub-step composite implicit time integration scheme for structural dynamics[J/OL]. Computers & Structures, 2017, 182(C): 176 – 186.
<http://www.sciencedirect.com/science/article/pii/S0045794916313712>.
- [121] Rezaiee-Pajand M, Sarafrazi S R. A mixed and multi-step higher-order implicit time integration family[J]. ARCHIVE Proceedings of the Institution of Mechanical Engineers Part C Journal of Mechanical Engineering Science 1989-1996 (vols 203-210), 2010, 1(1): 1 – 12.
- [122] Rezaiee-Pajand M, Karimi-Rad M. More accurate and stable time integration scheme[J/OL]. Engineering with Computers, 2015, 31(4): 791 – 812.
<http://dx.doi.org/10.1007/s00366-014-0390-x>.
- [123] Bathe K-J, Noh G. Insight into an implicit time integration scheme for structural dynamics[J/OL]. Computers & Structures, 2012, 98-99: 1 – 6.
<http://www.sciencedirect.com/science/article/pii/S0045794912000107>.

- [124] Noh G, Bathe K-J. An explicit time integration scheme for the analysis of wave propagations[J/OL]. Computers & Structures, 2013, 129 : 178 – 193.
<http://dx.doi.org/10.1016/j.compstruc.2013.06.007>.
- [125] Kwon S-B, Lee J-M. A non-oscillatory time integration method for numerical simulation of stress wave propagations[J]. Computers & Structures, 2017, 192(Supplement C): 248 – 268.

攻读硕士学位期间发表的论文及其他成果

（一）发表的学术论文

- [1] XXX, XXX. Static Oxidation Model of Al-Mg/C Dissipation Thermal Protection Materials[J]. Rare Metal Materials and Engineering, 2010, 39(Suppl. 1): 520-524. (SCI 收录, IDS 号为 669JS, IF=0.16)
- [2] XXX, XXX. 精密超声振动切削单晶铜的计算机仿真研究 [J]. 系统仿真学报, 2007, 19 (4): 738-741, 753. (EI 收录号: 20071310514841)
- [3] XXX, XXX. 局部多孔质气体静压轴向轴承静态特性的数值求解 [J]. 摩擦学学报, 2007 (1): 68-72. (EI 收录号: 20071510544816)
- [4] XXX, XXX. 硬脆光学晶体材料超精密切削理论研究综述 [J]. 机械工程学报, 2003, 39 (8): 15-22. (EI 收录号: 2004088028875)
- [5] XXX, XXX. 基于遗传算法的超精密切削加工表面粗糙度预测模型的参数辨识以及切削参数优化 [J]. 机械工程学报, 2005, 41 (11): 158-162. (EI 收录号: 2006039650087)
- [6] XXX, XXX. Discrete Sliding Mode Control with Fuzzy Adaptive Reaching Law on 6-PEES Parallel Robot[C]. Intelligent System Design and Applications, Jinan, 2006: 649-652. (EI 收录号: 20073210746529)

（二）申请及已获得的专利（无专利时此项不必列出）

- [1] XXX, XXX. 一种温热外敷药制备方案: 中国, 88105607.3[P]. 1989-07-26.

（三）参与的科研项目及获奖情况

- [1] XXX, XXX. XX 气体静压轴承技术研究, XX 省自然科学基金项目. 课题编号: XXXX.
- [2] XXX, XXX. XX 静载下预应力混凝土房屋结构设计统一理论. 黑龙江省科学技术二等奖, 2007.

插图索引

图 2-1	Euler 格式的改进策略示意图	7
图 2-2	Euler 格式的改进算法框架	8
图 2-3	刚性稳定域	31
图 3-1	$N_{t+\Delta t}, N_t, N_{t-\Delta t}$ 二次形函数	44
图 3-2	(a): $N_{t+\Delta t}, N_t, N_{t-\Delta t}$ 分段线性的形函数;(b): 一般性的加权函数 $W(t)$ 示意图	47

表格索引

表 1	国际单位制中具有专门名称的导出单位	III
表 2-1	BDF、EBDF 和 MEBDF 算法所对应的 $A(\alpha)$ 稳定的 α 角度值.....	19
表 3-1	Newmark 算法家族的著名成员	43
表 3-2	Newmark 算法成员所对应的权重函数 $W(t)$	45
表 3-3	广义 α 积分算法成员的参数取值	56
表 3-4	单步隐式的 θ_1 算法的算法成员	59
表 3-5	高阶导数的单步显式算法参数取值	63
表 3-6	高阶导数的单步显式算法子成员的数值性能比较	63

致 谢

衷心感谢导师 XXX 教授对本人的精心指导。他的言传身教将使我终生受益。

.....

感谢哈工大 L^AT_EX 论文模板 hiThesis !