

波形松弛方法

蒋耀林 著



科学出版社
www.sciencep.com

波形松弛方法

蒋耀林 著

科学出版社

北京

内 容 简 介

本书主要讨论用于求解微分方程并具有广泛应用背景的波形松弛方法理论及应用. 除绪论外, 全书共 11 章, 基本内容包括初值问题与周期问题的连续及离散波形松弛方法的收敛性、波形松弛算子的谱理论、波形松弛方法的加速算法, 以及其他一些常用方法. 全书论证详尽, 系统性强, 各章内容自成体系, 又相互联系. 为便于读者理解和阅读, 在内容安排上, 由浅入深, 循序渐进, 详略得当.

本书可供计算数学、应用数学、电路与系统以及计算机相关专业研究生阅读, 同时也可作为理工类相关专业教师以及从事科学和工程计算的科研工作者的参考书.

图书在版编目(CIP)数据

波形松弛方法/蒋耀林著. —北京: 科学出版社, 2009

ISBN 978-7-03-023523-7

I. 波… II. 蒋… III. 松弛法 IV. O242.26

中国版本图书馆 CIP 数据核字 (2008) 第 186580 号

责任编辑: 赵彦超 / 责任校对: 陈玉凤

责任印制: 钱玉芬 / 封面设计: 杜剑平

科学出版社 出版

北京东黄城根北街 16 号

邮政编码: 100717

<http://www.sciencep.com>

双青印刷厂印刷

科学出版社发行 各地新华书店经销

*

2009 年 2 月第 一 版 开本: B5 (720×1000)

2009 年 2 月第一次印刷 印张: 23 3/4

印数: 1—2 500 字数: 466 000

定价: 68.00 元

(如有印装质量问题, 我社负责调换〈双青〉)

前 言

微分方程可用来描述自然规律,而微分方程求解则是现代大型科学与工程计算的核心.随着计算机的飞速发展,需求解问题的规模越来越大,而迭代法作为解决大规模问题的有效方法,也成为求解大型微分方程最重要的方法之一.波形松弛方法,有时也称为动力学迭代方法,作为一种典型的整体动态迭代方法,更受到越来越多科学工作者的关注.随着学者们的不断研究与探索,波形松弛方法在理论分析方面已经取得极大的发展.此方法最大的优点是将复杂的系统解耦,并使解耦后的子系统保持原系统的某些特性,而且能做到并行求解.在当今科技高速发展的年代,并行算法的求解得到广大学者的关注,而且实际问题的并行化也是大型工程应用领域的迫切需求.在这样的情况下,波形松弛方法作为一种可并行实现的方法日臻成熟.

波形松弛方法自 1982 年在电路模拟领域被提出以来,不但在电子工程界得到广泛的应用,而且在科学计算界也得到普遍重视.实际上,在波形松弛方法出现以前,微分方程领域就存在着著名的 Picard 迭代方法.现在我们知道,其实这仅仅是波形松弛方法的一种简单形式.一般而言,人们对微分方程的认识主要是基于定性分析和数值分析这两种方法.定性分析一般难于全面描述一般方程的性态,而且对于大型复杂系统也无能为力;而一般数值方法计算过程比较单一,不能充分利用计算资源,并且为了提高精度而改变步长时,需要对整个过程重新计算,从而消耗大量时间.现在,许多工业问题都要求快速、实时的响应,例如飞行器或舰船的模拟与控制、期权价格计算、分子动力学的模拟、病理学的模拟、流体结构的计算、长时间的天气预报以及卫星轨道控制等,单一的数值计算方法都难以完全胜任.利用波形松弛方法求解复杂问题不仅能够达到工业生产中所要求的精度,而且能够从松弛迭代过程看出问题的性态变化趋势.更为重要的是,松弛后的系统由不耦合或弱耦合的子系统组成,如果再对其进行并行处理,则可节省大量运算时间.

目前,波形松弛方法已经应用到更复杂的模型,比如随机微分方程、抛物型偏微分方程等.波形松弛方法的解耦与并行的思想更是被应用到其他算法中,形成许多高效的新算法,比如新型区域分解、新型多重网格以及其他的一些新型算法.波形松弛方法的实现过程相对简单,并且在这方面的研究已趋于成熟.近年,随着与各种算法的交叉融合,波形松弛方法的思想得到更广阔和更深入的发展.因此,作者认为目前有必要对波形松弛方法及时做一个总结.同时,考虑到它的发展现状,希望能对其进一步的研究与推广起到促进作用.

本书大部分内容取自作者多年来在波形松弛方法方面所做的科学研究工作,并

且为兼顾全书的完整性,筛选了少量该领域的成熟成果.自1995年起,作者一直从事波形松弛方法的理论分析以及在电路模拟领域内的应用等研究工作.当前,作者正致力于波形松弛方法与其他方法的交叉应用以及波形松弛方法的并行实现等问题的研究.本书的部分内容曾作为西安交通大学研究生学位课程多次讲授,受到学生的欢迎.书中基本内容散见于有关波形松弛方法的诸多文献中.为了便于读者阅读,我们尽量用比较通俗易懂的语句叙述,内容方面注重条理性和系统性.同时,既重视基础理论,也注意最新进展,努力使读者阅读本书后能很快地进入到波形松弛方法的前沿研究中,而且也容易利用此方法去解决实际问题.经过整理、归纳,并总结学生的反馈意见等,在不断修改、丰富和完善后,最终完成了本书的定稿.

本书主要内容包括三个部分,除绪论外共11章.具体安排如下:

绪论部分介绍波形松弛方法的基本思想,内容比较简单,为后面部分做一些铺垫.

第1~6章为第一部分,遵循由简单到复杂的顺序,介绍波形松弛方法在不同系统初值问题中的应用,这些初值问题可以看作工程应用领域中瞬态响应问题的抽象模型.

第7~9章为第二部分,与第一部分相对应,这一部分针对工程应用领域中普遍关注的稳态响应问题,集中介绍周周期问题的波形松弛方法.

第10~11章为第三部分,主要介绍波形松弛方法的加速技术,以及在特征值求解和模型降阶等方法中的应用.

在本书的写作过程中,作者的学生刘军、陈芳、李荣建、张辉、孔旭和李一鹏等同学付出了许多辛勤劳动,尤其是刘军同学,他长期负责材料的收集和整理等繁杂工作.在多年的研究工作和本书的写作中,作者的家人一直为作者营造着温馨和谐的家庭环境,使作者无后顾之忧.本书的出版得到了西安交通大学腾飞特聘教授科研配套经费的支持.在本书出版之际,衷心感谢所有支持和帮助过作者的人和机构.

由于作者水平有限,书中不妥与错误之处在所难免,希望广大读者和同仁不吝赐教.

蒋耀林

2009年1月于西安

目 录

绪论	1
0.1 波形松弛方法的基本思想	2
0.2 波形松弛方法的简单分类	4
第 1 章 常微分方程的波形松弛方法	7
1.1 泛函分析预备知识	7
1.1.1 Banach 空间	7
1.1.2 线性算子谱与谱半径	8
1.1.3 压缩映射原理	9
1.2 线性微分方程的波形松弛方法	10
1.2.1 迭代格式	10
1.2.2 连续时间情形	11
1.2.3 离散时间情形	17
1.3 非线性微分方程的波形松弛方法	23
1.3.1 一阶微分方程情形	23
1.3.2 二阶微分方程情形	28
1.4 波形松弛算子谱与伪谱	32
第 2 章 线性微分代数方程的波形松弛方法	40
2.1 微分代数方程简介	40
2.2 波形松弛方法	41
2.2.1 连续波形松弛方法	42
2.2.2 离散波形松弛方法	47
2.2.3 波形 Krylov 子空间方法	50
2.3 波形松弛算子谱与伪谱	55
2.3.1 波形松弛算子谱	55
2.3.2 波形松弛算子伪谱	63
第 3 章 非线性微分代数方程的波形松弛方法	73
3.1 典型微分代数方程的波形松弛方法	73
3.1.1 半显式微分代数方程	73
3.1.2 简单隐式微分代数方程	80

3.2	一般微分代数方程的波形松弛方法	84
3.2.1	完全隐式微分代数方程	84
3.2.2	高指标微分代数方程	95
3.3	单调波形松弛方法	101
3.3.1	初始值与输入函数的单调依赖性	102
3.3.2	收敛性分析	105
3.3.3	初始迭代选取	106
第 4 章	积分微分代数方程的波形松弛方法	111
4.1	线性积分微分代数方程的波形松弛方法	111
4.1.1	连续波形松弛方法	111
4.1.2	离散波形松弛方法	113
4.1.3	多重分裂波形松弛方法	114
4.1.4	波形 Krylov 子空间方法	118
4.1.5	矩阵分裂方法	120
4.2	非线性积分微分代数方程的波形松弛方法	122
4.2.1	连续波形松弛方法	123
4.2.2	离散波形松弛方法	134
第 5 章	时滞微分方程的波形松弛方法	141
5.1	显式时滞常微分方程的波形松弛方法	141
5.1.1	简单时滞微分方程	141
5.1.2	典型时滞微分方程	143
5.1.3	广义时滞常微分方程	149
5.2	隐式时滞常微分方程的波形松弛方法	152
5.3	时间域无损传输线方程的波形松弛方法	158
5.3.1	无损传输线方程模型	158
5.3.2	波形松弛方法	161
第 6 章	偏微分方程的波形松弛方法	166
6.1	多重网格波形松弛方法	166
6.1.1	多重网格方法	166
6.1.2	连续时间情形	169
6.1.3	离散时间情形	182
6.2	区域分解波形松弛方法	189
6.2.1	区域分解方法介绍	189
6.2.2	传统 Schwarz 波形松弛方法	193
6.2.3	优化 Schwarz 波形松弛方法	199

第 7 章 常微分方程的周期波形松弛方法	204
7.1 线性微分方程的周期波形松弛方法	204
7.1.1 周期多重分裂波形松弛方法	204
7.1.2 周期多重打靶波形松弛方法	207
7.2 非线性微分方程的周期波形松弛方法	211
7.2.1 强耗散情形	212
7.2.2 一般情形	215
7.2.3 基于谐波平衡的波形松弛方法	218
7.3 非线性微分方程的拟线性周期波形松弛方法	220
7.3.1 拟线性化过程	220
7.3.2 收敛性分析	222
7.4 非线性时滞常微分方程的周期波形松弛方法	226
第 8 章 微分代数方程的周期波形松弛方法	231
8.1 线性微分代数方程的周期波形松弛方法	231
8.1.1 连续周期波形松弛方法	231
8.1.2 离散周期波形松弛方法	247
8.2 非线性微分代数方程的周期波形松弛方法	250
8.2.1 周期波形松弛方法	250
8.2.2 周期 Newton 波形松弛方法	254
第 9 章 偏微分方程的周期波形松弛方法	256
9.1 周期多重网格波形松弛方法	256
9.1.1 收敛性分析	257
9.1.2 模型问题	262
9.2 周期区域分解波形松弛方法	267
9.2.1 两个重叠子区域情形	269
9.2.2 有限个重叠子区域情形	275
第 10 章 波形松弛的加速方法	283
10.1 窗口加速方法	283
10.1.1 非线性情形	283
10.1.2 特殊情形: 线性方程	291
10.2 超松弛加速方法	293
10.2.1 逐次超松弛加速方法	293
10.2.2 卷积逐次超松弛加速方法	304
10.3 其他加速方法	311
10.3.1 优化波形松弛方法	311

10.3.2	预处理加速方法	317
10.3.3	多项式加速方法	319
第 11 章	波形松弛方法的一些应用	321
11.1	特征值问题中的波形松弛方法	321
11.1.1	特征值问题的并行算法	321
11.1.2	特征值问题的并行实现	331
11.2	模型降阶中的波形松弛方法	335
11.2.1	主成分分析与模型降阶	335
11.2.2	降阶与分解的基本过程	341
11.2.3	线性时不变情形	344
11.3	抽象空间中的波形松弛方法	350
11.3.1	发展方程	351
11.3.2	空间分解	352
11.3.3	收敛性分析	354
参考文献		362

绪 论

近年来,在数值模拟或数值计算中出现的方程的规模越来越大,形式也越来越复杂,有微分方程、积分方程、代数方程等各种不同形式.传统的数值方法在求解其中很多问题时都存在着或多或少的不足.如一般情况下,对于给定的时间步长,求含时间的刚性非线性常微分方程的标准数值解通常分为以下三个步骤:

(1) 首先对常微分方程的时间变量进行离散,得到一个非线性代数系统.此系统是由在特定离散时间点上趋近于常微分方程原始解的一系列代数方程组成的;

(2) 利用 Newton-Raphson 方法或相关的技术,对所得非线性系统进行线性化,这样得到一个大规模的线性代数系统;

(3) 最后用 Gauss 直接消元法或其变形形式求解线性化得到的大型线性代数系统.

上述过程已经被证明对求解小规模或中等规模的系统是完全适用的,例如成百上千个方程组成的系统.然而对于上万个以上方程组成的系统,这种方法的优点就不太明显了.原因有两方面:首先,每一个变量是在固定的时间点上进行离散的,这要求时间步长的选择必须适应不断迅速变化的函数值.在时间足够长时,大规模或大型微分方程系统中的变量各自变化速度不同.时间步长的加细对于变化慢的变量未必合适,这样就浪费了大量计算时间.其次, Gauss 消元法所花费的时间随着变量个数的急剧增加而迅速增长,即使避免使用刚性矩阵而采用稀疏矩阵技术,消元法在每个时间点上的消耗也是很惊人的.

工业应用中许多数学物理模型都是通过大规模的常微分或微分代数系统来描述的,并且系统中许多变量随时间的变化差异很大.为克服传统的数值方法存在的不足,人们已经尝试了多种方法,大部分的求解技术都是迭代技术,它们以“松弛”过程为基础.对于松弛过程而言,基于求解线性系统的方法称为线性松弛方法,基于求解非线性系统的方法称为非线性松弛方法,基于求解微分方程的方法称为波形松弛 (waveform relaxation, WR) 方法,有时文献中也称 WR 方法为动力学迭代 (dynamic iteration) 方法.

下面以常微分方程系统为例,简单介绍一下 WR 方法的一般而基本的格式,同时给出几种常用的 WR 方法.

0.1 波形松弛方法的基本思想

考虑一般的非线性常微分方程 (ODEs) 初值问题

$$\begin{cases} \frac{dx(t)}{dt} = f(x(t), t), \\ x(0) = x_0, \quad t \in [0, T], \end{cases} \quad (0.1.1)$$

其中 $T > 0$, 非线性函数 $f: \mathbf{R}^n \times [0, T] \rightarrow \mathbf{R}^n$, $x_0 = [x_{1,0}, x_{2,0}, \dots, x_{n,0}]^T \in \mathbf{R}^n$ 为给定的初值, $x(t) = [x_1(t), x_2(t), \dots, x_n(t)]^T \in \mathbf{R}^n$ 为所求向量. 我们可以将系统 (0.1.1) 写成如下分量形式

$$\begin{cases} \frac{dx_1(t)}{dt} = f_1(x_1(t), x_2(t), x_3(t), \dots, x_n(t), t), \\ \frac{dx_2(t)}{dt} = f_2(x_1(t), x_2(t), x_3(t), \dots, x_n(t), t), \\ \dots\dots\dots \\ \frac{dx_n(t)}{dt} = f_n(x_1(t), x_2(t), x_3(t), \dots, x_n(t), t), \\ x_1(0) = x_{1,0}, x_2(0) = x_{2,0}, \dots, x_n(0) = x_{n,0}. \end{cases}$$

求解系统 (0.1.1) 所用的 WR 方法是一种连续时间迭代方法. 也就是说, 给定一个逼近系统 (0.1.1) 的解的近似函数, WR 方法会在给定的时间区间内计算出一个新的逼近解. 之所以称为 WR 方法, 是因为这是关于解函数的迭代过程, 这种解函数在工业应用中通常表现为波形函数, 尤其在现代集成电路模拟领域更是如此. 很明显, WR 不同于大多数标准的迭代方法, 而是以时间为自变量的函数的迭代. 迭代形式的选取, 应使分解后的微分方程相对简单.

下面是一种基本而有效的 WR 方法. 这种方法利用上一步“旧的”迭代值计算下一步“新的”迭代值, 迭代形式如下

$$\begin{cases} \frac{dx_1^{(k+1)}(t)}{dt} = f_1(x_1^{(k+1)}(t), x_2^{(k)}(t), x_3^{(k)}(t), \dots, x_n^{(k)}(t), t), \\ \frac{dx_2^{(k+1)}(t)}{dt} = f_2(x_1^{(k+1)}(t), x_2^{(k+1)}(t), x_3^{(k)}(t), \dots, x_n^{(k)}(t), t), \\ \dots\dots\dots \\ \frac{dx_n^{(k+1)}(t)}{dt} = f_n(x_1^{(k+1)}(t), x_2^{(k+1)}(t), x_3^{(k+1)}(t), \dots, x_n^{(k+1)}(t), t), \\ x_1^{(k+1)}(0) = x_{1,0}, x_2^{(k+1)}(0) = x_{2,0}, \dots, x_n^{(k+1)}(0) = x_{n,0}. \end{cases}$$

这种方法称为 Gauss-Seidel WR 方法, 此方法与求解线性代数方程的 Gauss-Seidel 迭代方法十分相似. 它将求解 n 个变量微分方程的任务转化成按顺序求解一系列单变量微分方程的任务.

此外, Jacobi WR 方法是另一种常用的迭代方法, 其迭代形式更简单, 更适合并行计算, 其具体形式如下

$$\left\{ \begin{array}{l} \frac{dx_1^{(k+1)}(t)}{dt} = f_1(x_1^{(k+1)}(t), x_2^{(k)}(t), x_3^{(k)}(t), \dots, x_n^{(k)}(t), t), \\ \frac{dx_2^{(k+1)}(t)}{dt} = f_2(x_1^{(k)}(t), x_2^{(k+1)}(t), x_3^{(k)}(t), \dots, x_n^{(k)}(t), t), \\ \dots\dots\dots \\ \frac{dx_n^{(k+1)}(t)}{dt} = f_n(x_1^{(k)}(t), x_2^{(k)}(t), x_3^{(k)}(t), \dots, x_n^{(k+1)}(t), t), \\ x_1^{(k+1)}(0) = x_{1,0}, x_2^{(k+1)}(0) = x_{2,0}, \dots, x_n^{(k+1)}(0) = x_{n,0}. \end{array} \right.$$

以上两种 WR 方法的初始迭代函数 $x^{(0)}(t)$ 都定义在整个时间区间上. 很自然地选取此初始迭代函数向量为与初值相关的常向量

$$x_i^{(0)}(t) \equiv x_{i,0}, \quad t \in [0, T], \quad i = 1, 2, \dots, n.$$

当然, 初始函数向量也可以选取为其他相关问题的解. 比如, 可以将对方程右端做稍微的简化得到的解作为初始迭代函数, 或者用略有不同的初始条件得到的解作为初始迭代函数等.

Gauss-Seidel WR 方法的计算过程如下面的算法所示, Jacobi WR 方法与之类似. 不难发现, Gauss-Seidel WR 方法是典型的顺序方法, 方程依次求解. 相反地, Jacobi WR 方法则是完全并行的方法, 每次迭代, 各方程彼此相互独立, 能够同时求解.

算法 (Gauss-Seidel WR 方法)

第一步. 令 $k = 0$, 选择初始函数 $x_i^{(0)}(t)$, $i = 1, 2, \dots, n$;

第二步. 令 $k = k + 1$, 对每一个 $i = 1, 2, \dots, n$, 求解方程

$$\frac{dx_i^{(k+1)}(t)}{dt} = f_i(x_1^{(k+1)}(t), \dots, x_{i-1}^{(k+1)}(t), x_i^{(k+1)}(t), x_{i+1}^{(k)}(t), \dots, x_n^{(k)}(t), t),$$

其初值为 $x_i^{(k+1)}(0) = x_{i,0}$; 迭代直至收敛.

例 0.1.1 利用 Gauss-Seidel WR 方法求解下面微分方程组, 其中 $t \geq 0$,

$$\begin{cases} \frac{dx_1(t)}{dt} = x_2(t), \\ \frac{dx_2(t)}{dt} = -x_1(t), \\ x_1(0) = 0, x_2(0) = 1. \end{cases}$$

关于迭代 $x_1^{(k)}(t)$ 和 $x_2^{(k)}(t)$ 的计算过程如下, 这里 $x_1^{(k)}(0) = 0, x_2^{(k)}(0) = 1$.

$$\text{初始化: } \begin{cases} x_1^{(0)}(t) \equiv 0, \\ x_2^{(0)}(t) \equiv 1. \end{cases}$$

$$\text{第 1 次迭代: } \begin{cases} \frac{dx_1^{(1)}(t)}{dt} = x_2^{(0)}(t), \\ \frac{dx_2^{(1)}(t)}{dt} = -x_1^{(1)}(t). \end{cases} \Rightarrow \begin{cases} x_1^{(1)}(t) = t, \\ x_2^{(1)}(t) = 1 - \frac{t^2}{2}. \end{cases}$$

$$\text{第 2 次迭代: } \begin{cases} \frac{dx_1^{(2)}(t)}{dt} = x_2^{(1)}(t), \\ \frac{dx_2^{(2)}(t)}{dt} = -x_1^{(2)}(t). \end{cases} \Rightarrow \begin{cases} x_1^{(2)}(t) = t - \frac{t^3}{3!}, \\ x_2^{(2)}(t) = 1 - \frac{t^2}{2} + \frac{t^4}{4!}. \end{cases}$$

$$\text{第 3 次迭代: } \begin{cases} \frac{dx_1^{(3)}(t)}{dt} = x_2^{(2)}(t), \\ \frac{dx_2^{(3)}(t)}{dt} = -x_1^{(3)}(t). \end{cases} \Rightarrow \begin{cases} x_1^{(3)}(t) = t - \frac{t^3}{3!} + \frac{t^5}{5!}, \\ x_2^{(3)}(t) = 1 - \frac{t^2}{2} + \frac{t^4}{4!} - \frac{t^6}{6!}. \end{cases}$$

⋮

$$\text{第 } k+1 \text{ 次迭代: } \begin{cases} \frac{dx_1^{(k+1)}(t)}{dt} = x_2^{(k)}(t), \\ \frac{dx_2^{(k+1)}(t)}{dt} = -x_1^{(k+1)}(t). \end{cases} \Rightarrow \begin{cases} x_1^{(k+1)}(t) = \sum_{i=0}^k (-1)^i \frac{t^{2i+1}}{(2i+1)!}, \\ x_2^{(k+1)}(t) = \sum_{i=0}^{k+1} (-1)^i \frac{t^{2i}}{(2i)!}. \end{cases}$$

这个迭代过程明显收敛于方程的解 $x_1^*(t) = \sin t$ 和 $x_2^*(t) = \cos t$, 并且每一步迭代增加一项 Taylor 展开项.

0.2 波形松弛方法的简单分类

Jacobi WR 方法和 Gauss-Seidel WR 方法只是众多一般的 WR 方法中的两种. 一般 WR 方法的共同点在于它们是函数的迭代, 也就是说, 这类方法在迭代之前不必对时间进行离散. 同理, 也可以定义离散波形迭代, 即方程在离散时间点上的迭代格式. 我们分别称这些方法为连续时间 WR 方法和离散时间 WR 方法.

用 \mathfrak{R} 表示 WR 算子, WR 迭代过程可以简单地表示成以下形式

$$x^{(k+1)} = \mathfrak{R}(x^{(k)}), \quad k = 0, 1, \dots$$

此处, 我们只考虑与迭代次数无关的迭代算子, 而且只考虑单步迭代, 即 $x^{(k+1)}$ 只显式地依赖于 $x^{(k)}$, 与更前一步的迭代无关. 一类重要的 WR 方法可以用以下迭代公式来表示

$$\begin{cases} \frac{dx^{(k+1)}(t)}{dt} = F(x^{(k+1)}(t), x^{(k)}(t), t), \\ x^{(k+1)}(0) = x_0, \end{cases} \quad (0.2.1)$$

其中, 函数 F 称为函数 f 的 WR 分裂函数或简称分裂函数, 它应满足 $F(w, w, t) = f(w, t)$. 值得注意的是, 分裂函数 $F(u, v, t)$ 的选择必须使得微分方程 $\frac{du}{dt} = F(u, v, t)$ 容易求解, 并且迭代公式 (0.2.1) 应有快速的收敛性.

下面给出几个比较重要的常用波形迭代方法.

1. Picard 方法

Picard 迭代方法中分裂函数的选取很简单, 即 $F(u, v, t) = f(v, t)$, 其分量形式为

$$F_i(u, v, t) = f_i(v_1, v_2, \dots, v_i, \dots, v_n, t), \quad i = 1, 2, \dots, n.$$

从这个分裂函数可以看出, 对于每一步迭代, Picard 迭代式的右端项只有上一步迭代的结果, 对于新的迭代, 可以看作已知函数. 每步迭代中的 n 个微分方程相互独立, 因此它们可以同时积分, 从而可以很容易地用并行算法实现. 在对应的离散时间迭代中, 每个方程的右端可以在时间离散点上平行地赋值.

2. 波形松弛方法

在这种松弛方法中, 选取分裂函数 $F(u, v, t)$ 使得它的 Jacobi 矩阵 $\partial F / \partial u$ 是块对角或块下三角的, 从而微分方程系统 (0.1.1) 可以有效地解耦成可以独立求解或顺次求解的子系统 (0.2.1). 前面介绍的逐点 Jacobi WR 方法和 Gauss-Seidel WR 方法中分裂函数的特征分别为

$$F_i(u, v, t) = f_i(v_1, v_2, \dots, v_{i-1}, u_i, v_{i+1}, \dots, v_n, t) \quad (\text{Jacobi WR}),$$

$$F_i(u, v, t) = f_i(u_1, u_2, \dots, u_{i-1}, u_i, v_{i+1}, \dots, v_n, t) \quad (\text{Gauss-Seidel WR}).$$

我们可以类似地直接定义逐块 Jacobi WR 方法和逐块 Gauss-Seidel WR 方法. 比如, 一个 2×2 的块 Jacobi WR 方法由下列分裂函数 $F(u, v, t)$ 确定

$$\begin{cases} F_{2i-1}(u, v, t) = f_{2i-1}(v_1, \dots, v_{2i-2}, u_{2i-1}, u_{2i}, v_{2i+1}, \dots, v_n, t), \\ F_{2i}(u, v, t) = f_{2i}(v_1, \dots, v_{2i-2}, u_{2i-1}, u_{2i}, v_{2i+1}, \dots, v_n, t). \end{cases}$$

3. Newton 波形松弛方法

Newton 波形松弛方法, 即 Newton WR 方法, 它的分裂函数 $F(u, v, t)$ 可取为

$$F(u, v, t) = f(v, t) + \frac{\partial f}{\partial u}(v, t)(u - v).$$

这是对函数 f 线性化的结果, 其中需要计算 Jacobi 矩阵 $\partial f / \partial u$. 形式上, Newton WR 方法并不能将原始系统解耦成简单的子系统, 但是它能够快速收敛. 此外, 也可以用一些方法逼近 Jacobi 矩阵 $\partial f / \partial u$, 这就是拟 Newton WR 法.

除了以上的几种典型而简单的 WR 方法, 还有很多具有特殊用途的 WR 方法, 在这里就不一一列举了.

第 1 章 常微分方程的波形松弛方法

我们在绪论中已简单介绍了波形松弛方法的用法和分类, 其并行求解的特点使得这种方法在近十年得到了越来越多的关注. 本章以简单的常微分方程 (ODEs) 初值问题为例, 深入研究 WR 方法的本质, 并详细讨论 WR 方法的收敛性. 这一章的内容是后面章节的理论基础.

1.1 泛函分析预备知识

首先, 为方便阅读, 我们有必要回顾一下相关的泛函分析的内容. 这些基本内容在后面的理论分析中应用非常广泛.

1.1.1 Banach 空间

下面给出几个常用的 Banach 空间及相应的范数.

连续函数空间是指 $[0, T]$ 上的所有连续向量值函数的集合, 记为 $C([0, T]; \mathbf{C}^n)$ 或 $C([0, T])$. 在此空间上可以定义如下两种不同范数, 即最大值范数

$$\|x\|_T = \max_{0 \leq t \leq T} \|x(t)\|$$

和指数型 λ 范数

$$\|x\|_{\lambda, T} = \max_{0 \leq t \leq T} \{e^{-\lambda t} \|x(t)\|\},$$

其中 λ 是某正数, $\|\cdot\|$ 表示 \mathbf{C}^n 中的任意范数.

j 阶连续可导函数空间是指 $[0, T]$ 上的所有 j 阶连续可导向量值函数的集合, 记为 $C^j([0, T]; \mathbf{R}^n)$ 或 $C^j([0, T])$.

p 幂可积的 Lebesgue 可测函数空间是指满足

$$\int_0^T \|x(t)\|^p dt < +\infty, \quad 1 \leq p < +\infty$$

的向量值函数集合, 此空间记作 $L^p([0, T]; \mathbf{C}^n)$ 或 $L^p([0, T])$. 这里, T 可以是 $+\infty$, 相应空间为 $L^p([0, +\infty); \mathbf{C}^n)$ 或 $L^p([0, +\infty))$, 下同. 在空间 $L^p([0, T])$ 上可以定义如下范数

$$\|x\| = \sqrt[p]{\int_0^T \|x(t)\|^p dt}, \quad 1 \leq p < +\infty.$$

需要说明的是, $\|x\| = 0$ 并不能推导出在每一点都有 $x(t) = 0$. 因为 $x(t)$ 可能在某个零测度集合上非零, 为此, 通常并不区分只在零测度集合上不同的函数, 也就是说, “ $x = y$ ” 意味着 $x(t) = y(t)$ 几乎处处成立, 记作 “ $x(t) = y(t)$ a.e.”.

本性有界函数空间是指几乎处处满足 $\|x(t)\| \leq M < +\infty$ 的 Lebesgue 可测函数 $x(t)$ 的集合, 记为 $L^\infty([0, T]; \mathbf{C}^n)$ 或 $L^\infty([0, T])$. 在此空间上可以定义如下范数

$$\|x\|_\infty = \inf \{M \mid \|x(t)\| \leq M \text{ a.e., } t \in [0, T]\}.$$

p 幂序列空间是指满足

$$\sum_{i=0}^N \|x_i\|^p < +\infty, \quad 1 \leq p < +\infty$$

的序列 $x = \{x_i\}_{i=0}^N$ 的集合, 其中 $x_i \in \mathbf{C}^n$, N 可以是 $+\infty$, 此空间记为 $l^p(0 \cdots N; \mathbf{C}^n)$ 或 $l^p(N)$. 在此空间上可以定义范数

$$\|x\| = \sqrt[p]{\sum_{i=0}^N \|x_i\|^p}, \quad 1 \leq p < +\infty.$$

类似地, 可定义空间 $l^\infty(0 \cdots N; \mathbf{C}^n)$ 或 $l^\infty(N)$, 其范数为 $\|x\| = \max_{0 \leq i \leq N} \{\|x_i\|\}$.

在不引起混淆的情形下, 以上相应空间中的向量值函数也可以在 \mathbf{R}^n 中取值, 即空间 $C([0, T]; \mathbf{R}^n)$, $L^p([0, T]; \mathbf{R}^n)$, $L^\infty([0, T]; \mathbf{R}^n)$, $l^p(0 \cdots N; \mathbf{R}^n)$ 以及 $l^\infty(0 \cdots N; \mathbf{R}^n)$.

在本书中, 若无特殊说明, 矩阵或算子范数均是向量或函数向量范数相容的范数. 通常记 I 是单位矩阵或 Banach 空间中的恒等算子. 此外, 有时也简单记指数函数 e^t 为 $\exp(t)$ 等.

1.1.2 线性算子谱与谱半径

在有限维向量空间中, 线性系统迭代方法的收敛性由迭代算子 (矩阵) 的谱半径决定, 在 Banach 空间中的迭代方法也有类似的性质.

定义 1.1.1 线性赋范空间及相应的范数用符号 $(X, \|\cdot\|_X)$ 表示. 如果存在非负常数 c , 使得对所有 $x \in X$, 都有 $\|\mathfrak{R}x\|_X \leq c\|x\|_X$, 则称线性算子 \mathfrak{R} 是有界的.

定义 1.1.2 定义在赋范空间 X 上的有界线性算子 \mathfrak{R} 的范数为

$$\|\mathfrak{R}\|_X = \sup_{0 \neq x \in X} \{\|\mathfrak{R}x\|_X / \|x\|_X\} = \sup_{\|x\|_X=1} \|\mathfrak{R}x\|_X.$$

定义 1.1.3 定义在赋范空间 X 上的线性算子 \mathfrak{R} 的谱, 记为 $\sigma(\mathfrak{R})$, 其元素 λ 使得在 X 的稠密子集上, 算子 $\lambda I - \mathfrak{R}$ 不存在有界逆算子, 其中 I 是恒等算子.