

目 录

第一章 绪论	(1)
1.1 钢筋混凝土结构非线性分析的意义	(1)
1.2 钢筋混凝土结构有限元分析的特点与现状	(2)
1.3 钢筋混凝土结构有限元分析的发展趋势	(4)
1.4 钢筋混凝土结构非线性分析中的几个基本概念	(5)
第二章 钢筋混凝土结构材料的本构关系	(7)
2.1 概述	(7)
2.2 钢筋的本构关系	(13)
2.3 混凝土的本构关系	(16)
2.4 钢筋与混凝土之间的粘结	(34)
第三章 钢筋混凝土结构有限元分析中的几种单元	(41)
3.1 钢筋混凝土结构有限元分析计算步骤	(41)
3.2 平面单元	(43)
3.3 杆系单元	(59)
3.4 联结单元	(63)
3.5 钢筋混凝土结构有限元模型的选择	(72)
第四章 非线性有限元分析的计算方法	(77)
4.1 混凝土的开裂与破坏	(77)
4.2 有限元非线性方程组的解法	(84)
4.3 单元开裂和屈服后的处理	(95)
4.4 结构进入负刚度后的处理	(103)
第五章 钢筋混凝土构件有限元分析	(109)
5.1 按杆系结构进行梁的有限元分析	(109)
5.2 钢筋混凝土构件荷载-挠度曲线计算	(113)
5.3 按平面应力问题进行梁的有限元分析	(139)
第六章 钢筋混凝土框架结构有限元分析	(142)
6.1 基本假定与结构简化	(142)
6.2 结构非线性计算模型	(143)
6.3 结构有限元非线性分析	(145)
第七章 钢筋混凝土剪力墙结构有限元分析	(166)
7.1 概述	(166)
7.2 钢筋混凝土剪力墙有限元分析的基本理论	(166)
7.3 钢筋混凝土剪力墙有限元分析实例	(176)

第八章 钢筋混凝土结构动力有限元分析	(181)
8.1 动力分析的基本要求	(181)
8.2 动力方程及单元特性	(185)
8.3 动力特性的求解方法	(188)
8.4 动力反应的求解方法	(196)
8.5 动力系统的简化方法	(211)
附录 A 钢筋混凝土剪力墙结构非线性有限元分析源程序	(215)
附录 B 钢筋混凝土构件裂缝及变形图绘制	(237)
参考文献	(246)

第一章 绪 论

1.1 钢筋混凝土结构非线性分析的意义

钢筋混凝土结构是目前工业与民用建筑中最主要的结构型式。由于钢筋混凝土是由两种性质不同的材料——混凝土和钢筋组合而成的,它的性能明显地依赖于这两种材料的性能,特别是在非线性阶段,混凝土和钢筋本身的各种非线性性能,都不同程度地在这种组合材料中反映出来,关于混凝土材料和钢筋材料的非线性特征,读者可以参考有关钢筋混凝土结构基本理论方面的论著。在这里,我们仅仅讨论与钢筋混凝土结构计算分析有关的一些问题。

(1) 由于钢筋和混凝土的抗拉强度相差很大,钢筋混凝土结构在正常使用状态下,大部分受弯构件都已经开裂而进入非线性状态,但钢筋并未屈服仍在弹性状态下工作,因此,作为一个结构或构件来说,必然是在非线性状态下工作,这时用弹性分析方法求得的结构内力和变形就不能反映结构的实际工作状态。

(2) 混凝土和钢筋在一个结构中共同工作的条件是两者之间的变形协调,没有相对滑移,但实际上,这种条件并不能完全满足,特别是在反复荷载下,光圆钢筋与混凝土之间的粘结往往会破坏,某些情况下,会导致变形过大,而传统的线弹性结构分析不能反映这些现象。

(3) 在钢筋混凝土结构的设计中,在内力分析时,往往按弹性计算,而在构件截面设计时,却按极限状态进行计算,其结果是内力分析和截面设计的结果都不能反映结构的实际受力状态,造成了钢筋混凝土结构内力分析和截面设计的严重脱节。

(4) 与其他任何形式的结构一样,节点和连接是保证钢筋混凝土结构能作为一个复杂体系承受外力的基本条件,而传统的弹性结构分析时将节点理想化为刚接(例如框架)或者铰接(例如桁架)均不能反映节点的复杂受力状态和变形情况,从而难以为设计提供正确的信息。

(5) 在长期荷载作用下,混凝土会产生一定的徐变变形,这时,结构的内力和变形就发生了变化,按弹性分析求得的内力和变形就不能反映实际情况了。

正由于存在着这些问题,钢筋混凝土结构的非线性分析就显得特别重要,并成为结构工程领域研究的一个热点,受到了越来越多研究和设计人员的重视。用传统的解析方法分析钢筋混凝土结构的非线性问题时只能解决一些非常简单的构件或结构的计算,对于大量的钢筋混凝土结构分析问题,只能用数值分析方法解决。有限元方法作为一个强有力的数值分析工具,在钢筋混凝土结构的非线性分析中起到了越来越大的作用。用有限元方法进行钢筋混凝土结构的非线性分析,至少有以下优点:

- (1) 可以在计算模型中分别反映混凝土和钢筋材料的非线性特性;
- (2) 可以考虑或模拟钢筋与混凝土之间的粘结;

(3) 可以在一定程度上模拟节点的构造和边界条件;

(4) 可以提供大量的结构反应信息,例如应力、变形的全过程,结构开裂以后的各种状态。借助于先进的计算机图形显示技术,还可以直观地看到结构受荷载后从弹性变形到开裂、破坏的全过程,为进行合理的设计提供形象的依据;

(5) 可以部分代替试验,进行大量的参数分析,为制定设计规范和标准提供依据。

正由于有上述这些优点,有限元非线性分析方法在钢筋混凝土结构的研究和设计中有着广泛的应用前景。首先是作为一种强有力的研究工具,可以用于计算分析在试验中难以搞清楚的各种问题,例如不同材料界面的特性、不同结构与介质之间的耦合特性、结构中的局部应力问题以及复杂体型结构的内力与变形性能等问题。其次是作为一种非线性设计方法的工具,可以应用于重要结构的设计工作,例如核反应堆的安全壳、海洋平台结构、大型隧洞结构以及复杂体型高层建筑结构等。还可以应用于模拟施工过程的计算分析,例如混凝土坝体,由于施工工序多,工期长,混凝土的徐变在施工过程中和交付使用后一直存在,因此用有限元分析方法就可以模拟全过程的受力性能、应力及变形分布以及徐变后的应力重分布,为设计和施工提供参考信息。又例如对于地下结构的开挖和支护过程,也需要用有限元非线性分析方法进行计算分析,等等。

1.2 钢筋混凝土结构有限元分析的特点与现状

钢筋混凝土结构的有限元分析有与其他固体力学有限元分析所不同的特点,主要表现在以下几个方面:

(1) 需要模拟混凝土的开裂和裂缝发展过程,特别是在反复荷载作用下裂缝的开裂和闭合过程。

(2) 需要在模型中适当地反映钢筋与混凝土之间的粘结和滑移机理。

(3) 需要模拟混凝土材料在达到峰值应力以后的性能,因为一个部位的混凝土达到峰值应力并不能说明整个结构达到极限状态。同理,也应模拟钢筋屈服以后的性能。

(4) 对于复杂的钢筋混凝土结构,材料非线性问题与几何非线性问题同时存在,使得计算分析的难度大大增加。

(5) 分析结果强烈地依赖于混凝土材料和钢筋材料的本构关系以及钢筋与混凝土之间粘结滑移的本构关系。因此对上述本构关系的深入研究和全面正确的描述是保证钢筋混凝土有限元分析结果正确可靠和能应用于工程实际的基本条件。

正由于有以上特点,钢筋混凝土结构有限元分析作为一个相对独立的研究领域,受到了土木工程界越来越广泛的重视。美国学者 D.Ngo 和 A.C.Scordelis 最早把有限元分析方法应用于钢筋混凝土简支梁的抗剪分析(1967年)。在他们的研究工作中,将钢筋和混凝土均划分为三角形单元,按平面应力问题和线弹性理论分析混凝土和钢筋的应力;针对钢筋混凝土结构的特点,在钢筋和混凝土之间附加了一种沿钢筋径向和切向都有一定刚度的粘结弹簧,从而可以分析粘结应力的变化情况;为了反映混凝土开裂的特性,提出了离散裂缝(Discrete Cracks)的模式,即在梁中预先设置裂缝,裂缝两边用不同的节点,裂缝间也附加了特殊的联结弹簧,以模拟混凝土裂缝间的骨料咬合力和钢筋的销栓作用。在以

后的近 30 年中,钢筋混凝土结构有限元分析大体上经历了三个发展阶段^[1]。

第一个阶段是 1967~1977 年这一时期,在这个阶段中,在 Ngo 和 Scordelis 的工作基础上,还提出了分布裂缝(Smeared Crack)模式,将钢筋分布在混凝土单元中,假定钢筋和混凝土之间有效粘结;有些研究中还用“拉伸强化(Tension Stiffening)”的概念以考虑裂缝之间混凝土对受拉的贡献。由于分布裂缝模式计算相对简单并具有较高的精度,这一模式已被应用于平面应力、平面应变、板弯曲、壳体、轴对称和三维实体问题。对于板弯曲和壳体问题,使用了分层单元,并假定每一个混凝土单元处于双向受力状态,裂缝沿板厚逐层地发展。这些单元已经被应用于核反应完全壳、存储容器和海洋石油平台等大型混凝土结构的非线性分析中。这一阶段的研究和应用都取得了很大的进展,但总的来说,不管是理论研究还是工程应用,都比较粗糙,处于探索阶段。

第二个阶段是 1977—1985 年这段时期,这个时期的研究工作主要可分为两个方面。一方面是继续在单元模式的选取、混凝土的本构关系和破坏理论,裂缝的模拟和拉伸强化,骨料咬合和销栓作用以及粘结方面进行深入的研究。另一方面是系统性的总结和交流工作,美国土木工程师协会组织了一个 20 人的委员会,花了 5 年的时间,总结和分析了钢筋混凝土结构有限元分析领域的大量研究资料和信息,在 1982 年 5 月发表了长达 545 页的综述报告,内容涉及本构关系与破坏理论;钢筋模拟及粘结的表示;混凝土开裂;剪力传递;时间效应;动力分析;数值算例和应用;还在附录中发表了钢筋混凝土结构非线性分析的有限元源程序。在这一时期,欧洲和亚洲一些学者也在钢筋混凝土结构的有限元分析方面进行了大量的研究工作,1987 年 7 月在联邦德国召开了“钢筋混凝土空间结构非线性性能”的国际会议;1981 年,国际桥梁与结构工程协会在荷兰召开了“高等混凝土力学”的国际会议;1984 年,在前南斯拉夫召开了“混凝土结构的计算机辅助分析与设计”国际会议。在这一时期,日本学者的研究工作在起步较晚的情况下很快地达到了应用阶段,并且在与试验的结合方面取得了很大的进展^[2]。他们在梁、柱、梁柱节点、剪力墙、核反应堆结构等方面都进行深入细致的研究,并部分地应用于工程设计或为制订规范提供了依据。在这一时期,我国学者在资料较少的情况下也开始进行了研究工作,发表了一些论文和专著^{[3],[4]}。

第三个阶段是 1985 年到现在这段时期,除在混凝土的本构关系的表述和试验研究方面继续进行更深入的研究之外,钢筋混凝土结构非线性有限元分析进一步向实用方向发展,努力把现有的分析方法与工程设计结合起来。研究的领域也进一步扩展到动力、冲击荷载下的非线性分析,分析模型和材料参数成为预测钢筋混凝土结构在动力与冲击荷载下性能的研究热点;高强混凝土和受约束混凝土结构的非线性有限元分析也受到了重视;材料非线性、几何非线性以及时间因素的综合考虑也溶入了钢筋混凝土结构的非线性有限元分析。在混凝土结构中,与时间因素有关的效应包括荷载、预应力、环境条件以及随着时间推移而变化的徐变、收缩、老化、热效应和预应力筋的松弛。在这一时期中,我国在钢筋混凝土结构非线性有限元分析的大部分领域开展了研究工作,取得了很大进展。我国虽然没有专门召开过钢筋混凝土非线性有限元分析方面的会议,但这方面的研究工作在计算力学、结构工程、地震工程等全国性的学术会议中都有所反映,也出版了钢筋混凝土结构非线性有限元分析方面的专门著作^[5,6],反映了我国在这一领域的研究成果。近

几年来我国在地震作用下钢筋混凝土结构的非线性有限元时程分析方面的所做的工作也在国际上产生了一定的影响^{[7],[8]}。

1.3 钢筋混凝土结构有限元分析的发展趋势

科学研究的基本目的是认识世界和改造世界。钢筋混凝土结构非线性有限元分析的目的,第一是认识钢筋混凝土结构在各类荷载作用下的受力性能和破坏机理;第二是为钢筋混凝土结构的合理设计提供依据,今后的研究也将是在这两个方面开展工作。

为了更好地认识钢筋混凝土结构的受力性能和破坏机理,对下列一些基本问题的深入研究是必需的^[1,2,9]:

- (1) 在多轴应力状态下混凝土本构关系的试验及分析模型的建立。
- (2) 在反复荷载作用下混凝土受力性能的试验及滞回关系模型的描述。
- (3) 粘结、开裂及剪切机理的进一步研究和有限元描述。
- (4) 各种单元的最优选择,例如是选择钢筋与混凝土的组合单元还是单独选择混凝土单元和钢筋单元等。
- (5) 裂缝模式的最优选择,例如选择离散裂缝模式或者分布裂缝模式的条件和适用场合,这种选择对计算结果的影响等等。
- (6) 非线性计算的各种算法及其稳定性问题。

为了更好地用非线性有限元分析方法解决钢筋混凝土结构设计中的各种复杂问题,这也需要在下列一些问题上进行深入研究:

(1) 大型、开放的钢筋混凝土结构非线性分析程序的研制。这种程序应经过试验数据的检验,应包含混凝土在各种受力状态下的本构关系,应为用户提供灵活方便的接口,允许用户使用具体工程的试验数据来建立混凝土和钢筋的本构关系,允许用户使用按最新试验数据建立的破坏准则等等。

(2) 各种分析模型、计算方法的比较和选择。在基本模型和方法的研究中,一般是用有限的试验数据检验分析的正确性,而作为在设计中的应用,由于工作的复杂性和大型化,不可能直接用试验检验某种分析方法的正确性。这时就必须选择不同的分析模型和方法,对同一类结构进行计算分析,再通过进行参数研究和比较分析,从中选择适合某一类结构的分析模型和方法。也可以通过这种有效的计算,使得验证性试验的数量大大减少。

(3) 对现有的规范设计方法的评价和改进。其目标是,借助于有限元分析方法,重新评价现有的设计公式和宏观分析模型,通过大量的参数分析来改进宏观模型,进一步完善设计方法;希望发展一种有效的单元,例如像钢筋混凝土组合单元一样,这种单元具有较少的自由度,可以用于分析大尺度的结构而仍然具有较高的精度,等等。

(4) 不完整结构的分析和模拟。正在施工中的结构、分段装配式结构、在全部施加予应力之前的预应力混凝土结构,这些结构都是不完整的结构,在实际施工中经常存在。对这些结构,用钢筋混凝土结构有限元方法进行非线性总体分析、区域分析以及局部分析是具有重要现实意义的课题,近几年来受到了人们的重视,这也是有限元分析方法应用于工

程实际的一个重要领域。

1.4 钢筋混凝土结构非线性分析中的几个基本概念

在钢筋混凝土结构非线性有限元分析中,下列一些概念或名词术语经常会遇到,有必要在这里作以下介绍。

1. 本构关系

在研究材料的力学性质时,从由试验和经验中观察到的特性出发,在某些理论的假设下,找出描述材料力学性质的数学表达式,它们被称为材料的本构方程或本构关系。由于得出材料本构关系的过程中带有经验性和某些假定,因此,对于相同的材料,特别是混凝土材料,不同的研究人员可能会得到不同的本构关系。由于材料的本构关系仅是强调了材料在某些方面的性质,因此,对于像混凝土这样的各向异性的材料,受力性质不同,得到的本构关系不同,例如受压和受拉时的本构关系不同。

2. 屈服极限

物体在受力以后的变形过程中,当外力增加到一定程度时,变形由弹性变成非弹性(塑性),即开始产生永久变形。这种由弹性变形变为非弹性变形的转折点的应力就称为材料的屈服极限。结构上使用的软钢一类材料,具有明显的屈服阶段,因此这种材料具有明显的屈服极限。但对于混凝土材料,在受压时,一般没有明显的屈服极限,而在受拉时的屈服极限与强度极限基本一致。

3. 屈服条件 屈服函数 屈服面

判断物体内某一点开始出现塑性变形时其应力状态所应满足的条件,称为屈服条件,有时也称为塑性条件。从自然状态开始第一次产生塑性变形时的条件称为初始屈服条件,由于产生了塑性变形,材料的内部结构发生了变化,这时的屈服条件称为后继屈服条件。对于结构上使用的钢材,一般都有初始屈服和后继屈服的现象。对于混凝土材料,这种现象不明显。但对于钢筋与混凝土的组合材料,在研究反复荷载作用下的特性时,有时应考虑初始屈服和后继屈服的差别。

表示屈服条件的函数关系式称为屈服函数。屈服函数在应力空间中表示的曲面,称为屈服面。它是弹性与塑性阶段之间的界限,应力落在此曲面内的应力状态称为弹性状态,应力落在此曲面上的状态称为塑性状态。屈服面是由达到屈服条件的各种应力状态点集合而成的,在简单拉伸的情况下,它相当于拉伸曲线上的屈服点。

4. 强化 软化

为了叙述简单起见,我们以弹塑性材料(例如结构用钢材)简单拉伸试验的应力应变曲线图 1.1 为例,说明这两个概念。

在钢材的拉伸试验中,当应力未超过屈服极限 σ_s 时,应力应变关系呈直线,超过这一点时,应力应变之间就不再是线性关系,材料就发生了塑性变形,当塑性变形发展到一定程度时,材料的内部结构发生了变化,材料又重新获得了继续抵抗外载的能力,这时必须继续增大应力才能使它产生新的塑性变形,这种屈服极限提高的现象叫做强化,这个变形阶段称为强化阶段。当达到曲线最高点 C 时,应力达到最大值,这个值即材料的强度极

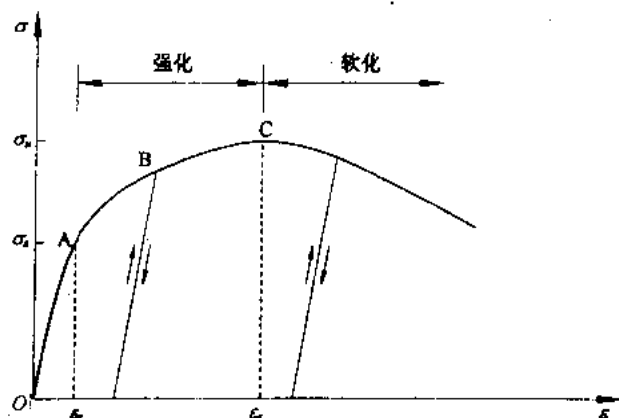


图 1.1 钢材简单拉伸 σ - ϵ 曲线示意图

限。此时应力开始下降而塑性变形不断增加,这种应力降低、应变增加的现象叫做软化,这个变形阶段称为软化阶段。

混凝土材料在单向受压时没有明显的屈服极限,因而没有像钢材那样明显的强化阶段。但混凝土材料有明显的强度极限,在强度极限之后有明显的软化阶段。混凝土材料软化阶段特征的描述和利用,一直是混凝土基本性能的试验研究和非线性有限元分析的热点和难点。

5. 钢筋混凝土构件的拉伸强化(Tension Stiffening)

在钢筋混凝土受拉构件中,当混凝土受拉开裂后,虽然主裂缝处混凝土不能承受拉应力,但主裂缝之间的混凝土仍承担一部分拉应力,这种现象称为拉伸强化(Tension Stiffening)。在钢筋混凝土结构有限元分析中,如采用钢筋与混凝土的组合单元,则在计算单元刚度和应力时需考虑拉伸强化效应,以反映单元开裂后混凝土与钢筋的共同工作。

第二章 钢筋混凝土结构材料的本构关系

2.1 概述

钢筋混凝土结构材料的本构关系对钢筋混凝土结构有限元分析结果有重大的影响,如果所选用的本构关系不能很好地反映材料的各项力学性能,那么其他计算再精确也无法反映结构的实际受力特征。

所谓材料的本构关系,主要是指描述材料力学性质的数学表达式。用什么样的表达式来描述材料受力后的变化规律呢?不同的学者根据材料的性质,受力的条件和大小,试验方法,不同的理论模型等因素综合考虑,建立了许多种钢筋混凝土材料的本构关系表达方式。

本构关系所基于的理论模型主要有:弹性理论、非线性弹性理论、弹塑性理论、粘弹性、粘塑性理论、断裂力学理论、损伤力学理论、内时理论等。迄今为止,由于钢筋混凝土材料的复杂因素,还没有一种理论模型被公认为可以完全描述混凝土材料的本构关系。有些本构关系虽然能比较好地反映材料在复杂应力状态下的应力应变关系,但是其表达形式很复杂,所采用的参数过多,使用不方便。一般来讲一个好的本构关系表达式应该是:数学公式推导方便;关系式中的有关主要参数有明确的物理意义;基于有关的试验和经验。

考虑到钢筋混凝土结构的特点及经济、实用,在钢筋混凝土结构非线性有限元分析中应用得较多的是非线性弹性理论和弹塑性理论。

本节对各种理论模型作一简单介绍,对于钢筋混凝土结构有限元分析中钢筋、混凝土及两者之间粘结所常常采用的本构关系在下面几节中具体讨论。

2.1.1 线弹性本构关系

在加载或卸载中,应力应变呈线性关系(以 E 为斜率的直线),见图 2.1,其表达式为:

$$\sigma = E\varepsilon \quad (2.1)$$

应力与应变之间呈线性比例关系,对于一维问题,其比例系数即为弹性模量 E ,它是常数。对于二维、三维问题,联系应力和应变之间关系的则是弹性矩阵。弹性矩阵中每一项均为常数,与应力水平和加载路径均无关。

当混凝土所受的力很小,无裂缝时,可将混凝土看成弹性匀质材料,采用线弹性的本构关系,但混凝土常常是带裂缝工作的,混凝土的应力应变关系显然与线弹性计算模型相差较大,然而由于线弹性模型简单,程序编制容易,应用方便,所以至今仍有相当广泛的应用。

2.1.2 非线性弹性关系

由于混凝土材料在加载过程中应变增加明显地快于应力的增加,为描述这种非线性关系,工程上常采用非线性弹性模型,如图 2.2 所示,其特点是应力应变不成正比,但仍有一一对应关系,卸载时沿加载路径返回,没有残余变形。应力状态由应变状态唯一确定,应力应变的关系式为:

$$\sigma = E(\sigma)\epsilon \quad (2.2)$$

式中弹性模量 E (对于二三维问题,则是物理矩阵)是应力水平(也可表示为应变大小)的函数,而不再是常数。如果找到适合于钢筋混凝土材料 $E(\sigma)$ 的表达式,且求解问题只是考虑单调加载的情况,就可以得到较为理想的计算结果。该关系式在钢筋混凝土结构非线性有限元分析中应用最广,后面讨论的几种混凝土本构关系均属于非线性弹性范畴。

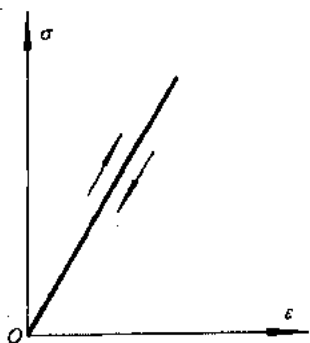


图 2.1 线弹性的应力-应变关系

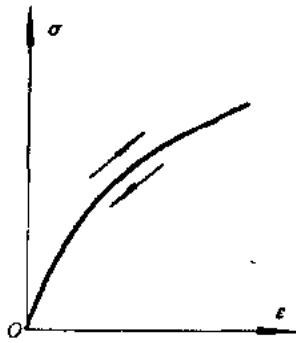


图 2.2 非线性弹性的应力-应变关系

2.1.3 弹塑性关系

前述变弹性常数的非线性弹性模型,考虑了钢筋混凝土材料的非线性,较线弹性模型前进了一大步。但这类模型的缺点是在材料屈服后的变形规律的描述并不符合塑性流动法则,使塑性变形的计算带有任意性。因此,对于受荷载作用后,处于较高应力水平下的钢筋混凝土结构,常常采用弹塑性的本构关系。

弹塑性理论的许多概念是根据材料单向拉压试验的应力-应变曲线所得到的,再推广到二维或三维的应力应变情况。以典型的钢筋单向拉伸试验为例,图 2.3(a)为软钢单向拉伸试验的应力应变曲线,曲线包括了线弹性阶段(OA 段),流动阶段(AB 段)及硬化阶段(BC 段),计算时,根据材料的不同条件作不同的简化,常用的简化模型为:

1. 理想弹塑性模型

见图 2.3(b),一旦进入塑性,其应力保持不变,而应变不断增加。

2. 线性强化弹塑性模型

当考虑材料进入塑性后的应变强化,可选用图 2.3(c)所示的线性强化弹塑性模型。

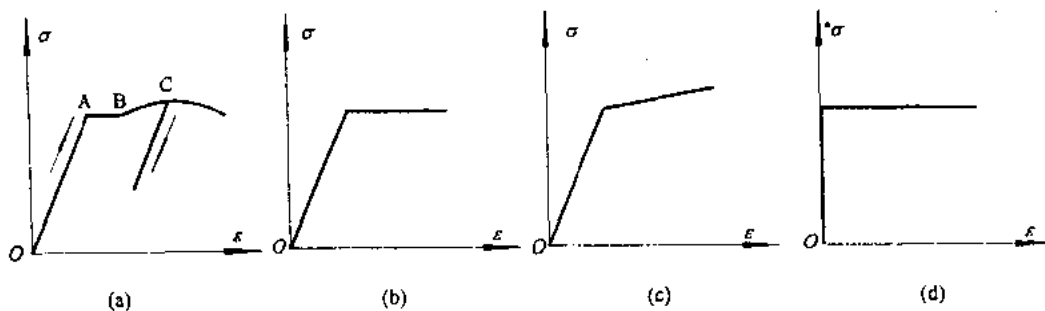


图 2.3 钢筋单向拉伸试验曲线及不同的简化模型

3. 刚塑性模型

若可恢复的弹性变形在总的变形中所占比例很小时,为简化计算也可忽略弹性变形部分,而只考虑塑性变形,可取图 2.3(d)所示的刚塑性模型。

弹塑性体的重要特点是,材料进入塑性状态的条件不仅与材料的物理力学性质有关,而且与加载历史及其应力水平有关。因此,为了在弹塑性分析时综合考虑上述因素,建立本构关系(除上面介绍的模型外,还有强化模型等)时,应同时考虑以下三个基本要求:

- (a) 假定一个符合材料特性的屈服准则;
- (b) 建立合适的塑性变形流动法则;
- (c) 建立与材料变形特征相应的硬化和软化定律。

2.1.4 粘弹性和粘塑性的流变模型

钢筋混凝土结构在外荷载作用下,其应变不仅与应力状态和加载历史、路径有关,而且还与时间因素有关。按照弹性力学或塑性力学观点,结构的内力和变形(无论是弹性变形还是塑性变形)均在加载的瞬时发生,不再随时间变化。但混凝土材料的徐变和预应力构件中钢筋应力损失等现象,说明需要考虑应力应变关系与时间的联系。如果引用流变学的观点(参见文献[12]),就能对上述现象作出较为合理的解释。

1. 流变学采用的三个简单流变元件

流变学计算模型往往是由若干个简单的流变元件组合而成的,如图 2.4 所示的三种理想化的简单流变元件,能形象化地表征物体的三个基本力学性质—弹性,粘性和塑性。

(a) 理想弹性元件

图 2.4(a)为弹簧元件又称虎克(Hooke)体,能模拟物体的弹性。应力应变关系为:

$$\sigma = E\epsilon$$

(b) 粘性元件

图 2.4(b)为粘性元件,又称牛顿(Newton)体或

阻尼器。它可以想象为一个活塞在充满了粘性液体的圆筒中运动,其应力与应变速率成

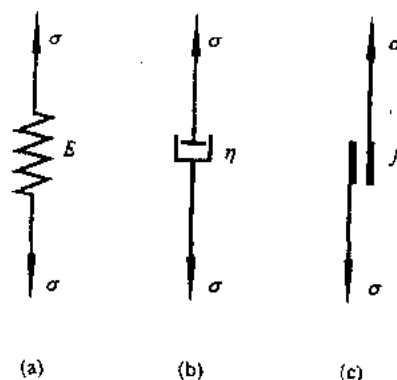


图 2.4 理想化的简单流变元件

正比:

$$\sigma = \eta \dot{\epsilon} \quad (2.3)$$

式中: η ——粘滞系数;

$\dot{\epsilon}$ ——应变速率($\dot{\epsilon} = d\epsilon/dt$)。

(c) 理想塑性元件

图 2.4(c) 为理想塑性元件, 又称圣维南(St. Venant)体或滑块元件。它可以想象为具有摩擦阻力 f 的两个滑块。当应力 σ 小于摩擦阻力时, 不产生变形, 一旦应力达到起始摩擦阻力时就产生等应力下塑性流动, 即有如下关系式:

当 $\sigma < f$ 时, $\dot{\epsilon} = 0$

当 $\sigma = f$ 时, $\dot{\epsilon} = \text{任意值}$ (取决于其它有关的条件)

2. 粘弹性流变模型

弹性材料实际上也具有流变性, 具有代表性的粘弹性模型称为广义凯尔文模型, 如图 2.5 所示, 它是由一个粘性元件(η)和一个弹簧(E_k)并联而成的凯尔文模型中再串联一个弹簧(E_H)组成。

一维问题的本构方程可表示为:

$$\frac{\eta}{E_k} \dot{\epsilon} + \epsilon = \frac{\eta \dot{\sigma}}{E_H \cdot E_k} + \frac{E_H + E_k}{E_H \cdot E_k} \sigma \quad (2.4)$$

式中: $\epsilon, \dot{\epsilon}$ ——应变和应变速率;

$\sigma, \dot{\sigma}$ ——应力和应力对时间 t 的导数;

E_H, E_k ——分别为两个弹簧的弹性常数。

图 2.5 广义凯尔文模型

当受到常应力 $\sigma = \sigma_0$ 作用, 显然在初始时刻 $t = 0$ 时, 应变为瞬时

应变 $\epsilon_0 = \frac{\sigma_0}{E_H}$, 以此作初始条件可以得到式(2.4)的解:

$$\epsilon = J(t) \sigma_0 \quad (2.5)$$

式中, $J(t)$ 称为蠕变柔量, 它是时间的函数:

$$J(t) = \frac{1}{E_H} + \frac{1}{E_k} \left[1 - \exp\left(-\frac{E_k}{\eta} t\right) \right] \quad (2.6)$$

显然, 式(2.5)表示了弹性材料在应力不变的情况下, 应变随时间不断增长的特性——弹性蠕变, 其应变值最终收敛于一个定值。

如果改用在 $t = 0$ 时 $\epsilon = \epsilon_0$ 的初始条件, 保持应变为常量, 又可得到式(2.4)的另一形式的解:

$$\sigma = E(t) \epsilon_0 \quad (2.7)$$

式中, $E(t)$ 称为松弛模量, 它是时间的函数:

$$E(t) = \frac{E_H E_k}{E_H + E_k} + \frac{E_H^2}{E_H + E_k} \left[\exp\left(-\frac{E_H + E_k}{\eta} t\right) \right] \quad (2.8)$$

此解反映了弹性材料在应变保持不变的情况下应力逐渐减小趋向于一个定值

$$\frac{E_H E_k}{E_H + E_k} \cdot \epsilon_0$$

3. 粘塑性流变模型

进入塑性后的材料,流变性比弹性阶段更明显,比较典型的粘塑性流变模型为宾哈姆(Bingham)模型,如图 2.6 所示。它是由一个粘性元件与一个滑块并联后,再串联一个弹簧组成的。

当材料应力低于屈服强度时(不超过滑块之间的起动摩阻力),宾哈姆模型只表现出弹性。但应力超过屈服强度后,宾哈姆模型就要发生粘性流动,一维问题的本构关系为:

$$\text{当 } \sigma < \sigma_y \text{ 时} \quad \sigma = E \varepsilon \quad (2.9)$$

$$\text{当 } \sigma \geq \sigma_y \text{ 时} \quad \dot{\varepsilon} = \frac{1}{E} \dot{\sigma} + \frac{1}{\eta} (\sigma - \sigma_y) \quad (2.10)$$

式中: σ_y ——材料的屈服强度;

$\dot{\sigma}$ 、 $\dot{\varepsilon}$ ——分别为应力和应变对时间 t 的导数;

E ——弹性模量;

η ——粘滞系数。

当初始条件 $t = 0$ 时,加载常量应力 $\sigma_0 > \sigma_y$,则式(2.10)的解为:

$$\varepsilon = \frac{\sigma_0}{E} + \frac{\sigma_0 - \sigma_y}{\eta} t$$

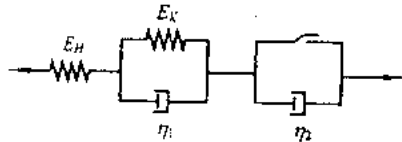


图 2.7 粘弹粘塑性模型

当同时考虑材料在弹性阶段和塑性阶段的流变时,可将上述的广义凯尔文模型和宾哈姆模型串联组成粘弹粘塑性模型,如图 2.7 所示。受荷载作用后,其总的应变应包括瞬时弹性应变、弹性阶段的粘性流变和进入塑性以后的粘塑流变,如下式所示:

$$\varepsilon = \varepsilon_e + \varepsilon_{ve} + \varepsilon_{vp}$$

式中: ε_e ——瞬时弹性应变;

ε_{ve} ——粘弹效应产生的应变;

ε_{vp} ——粘塑效应产生的应变。

粘弹粘塑模型的一维流变方程可写为:

$$\varepsilon = \left[\frac{1}{E_H} + \frac{1}{E_1} \left(1 - \exp \left(-\frac{E_1}{\eta_1} t \right) \right) \right] \sigma + \frac{t}{\eta_2} (\sigma - \sigma_y)$$

粘弹粘塑模型通常用于钢筋混凝土结构在长期荷载作用下的混凝土徐变和钢筋应力松弛等与时间相关的结构分析。

在粘弹粘塑体的有限元分析中,一般采用时步—初应变法计算,将整个粘性流变过程划分为若干个时步,把粘性应变看成为初应变。对每一个时步将流变引起的“初应变”转化为相应的结点荷载,并迭加到总的结点荷载中,从而建立起总体平衡方程。

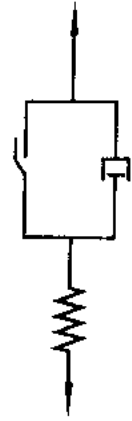


图 2.6 宾哈姆模型

4. 粘弹塑性流变模型

2.1.5 断裂力学模式

近十几年来,在钢筋混凝土结构中,断裂力学的应用得到了进一步的研究。国内、国际上也开过多次专门的混凝土断裂力学会议,研讨断裂力学在岩石与混凝土材料中的应用。

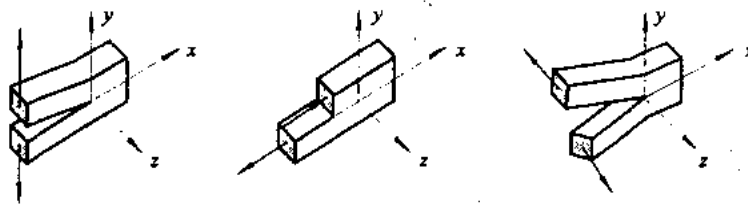
断裂力学的应用需要满足下述条件,而且该结构会发生低应力脆断:

- (a) 研究的结构应是含有裂缝的缺陷体;
- (b) 结构中有拉(剪、扭)应力的作用;
- (c) 材料本身的微观结构对脆断敏感。

钢筋混凝土结构能满足以上的条件,如混凝土的抗拉强度很低,受荷载作用后,在拉应力很小时就会产生裂缝;在一般受力构件中(除轴压构件外)均有拉(剪、扭)应力在起作用等。

断裂力学对于处理混凝土单条裂缝的扩展是很有效的,但其应用与它在金属结构中的应用相比较,相差甚远,主要原因是混凝土材料本身的不均匀性和非线性。

在一般应力场中,按裂缝位置与应力方向之间的关系,裂缝扩展可分为如图 2.8 所示的三种类型:张开型、剪切型和扭转型。下面以工程中最常见的张开型裂缝为例,简单介绍断裂力学的基本方法。



(a) 张开型

(b) 剪切型

(c) 扭转型

图 2.8 三种裂缝形态

对于张开型裂缝,在裂缝顶点附近的应力分量及位移分量为(图 2.9):

$$\left. \begin{aligned} \sigma_x &= \frac{K_I}{\sqrt{2\pi r}} \cos \frac{\theta}{2} \left[1 - \sin \frac{\theta}{2} \sin \frac{3\theta}{2} \right] \\ \sigma_y &= \frac{K_I}{\sqrt{2\pi r}} \cos \frac{\theta}{2} \left[1 + \sin \frac{\theta}{2} \sin \frac{3\theta}{2} \right] \\ \tau_{xy} &= \frac{K_I}{\sqrt{2\pi r}} \sin \frac{\theta}{2} \cos \frac{\theta}{2} \cos \frac{3\theta}{2} \\ u &= \frac{(1+\nu)K_I}{2E} \sqrt{\frac{r}{2\pi}} \left[(2S-1) \cos \frac{\theta}{2} - \cos \frac{3\theta}{2} \right] \end{aligned} \right\}$$

$$\left. \begin{aligned} v &= \frac{(1+\nu)K_I}{2E} \sqrt{\frac{r}{2\pi}} \left[(2S+1) \sin \frac{\theta}{2} - \sin \frac{3\theta}{2} \right] \\ S &= \frac{3-\nu}{1+\nu} \quad (\text{平面应力条件下}) \\ S &= 3-4\nu \quad (\text{平面应变条件下}) \end{aligned} \right\} \quad (2.11)$$

式中: r, θ ——以裂缝顶点为原点的极坐标

K_I ——I型裂缝的应力强度因子。 K_I 正比于外力,并取决于外力分布、材料和裂缝形状。

对于张开型裂缝而言,裂缝尖端的应力场就决定于强度因子 K_I ,当 K_I 达到临界值 K_{IC} 时,裂缝就扩展, K_I 是通过结构分析求出的。在线弹性断裂力学中,可以用有限元法计算裂缝尖端的应力强度因子 K_I 。

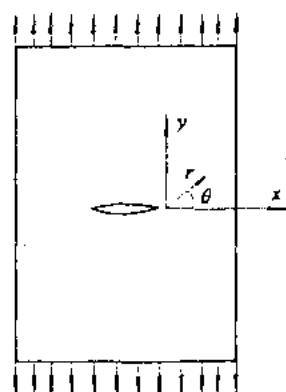


图 2.9 张开型裂缝前端应力

K_{IC} 为断裂韧性,仅仅取决于材料性能,表征材料对裂缝扩展的抗力,是从断裂力学引出的衡量材料韧性的指标,显然,当 $K_I > K_{IC}$ 时裂缝将扩展,结构将破裂。 K_{IC} 是通过材料试验测定的。对于线弹性均质材料,它是个常量,但由于混凝土材料的不均匀性,以及切口断裂效应,尺寸、时间效应等因素对 K_{IC} 均有影响。

2.2 钢筋的本构关系

在有限元分析中,常采用的钢筋本构关系如下所述。

2.2.1 单向加载下,钢筋的应力-应变关系

软钢的应力-应变曲线可以分三段:弹性段,屈服平台和强化段。

如图 2.10(a)所示,弹性段是以 E (钢筋弹性模量)为斜率的直线;屈服平台是斜率为

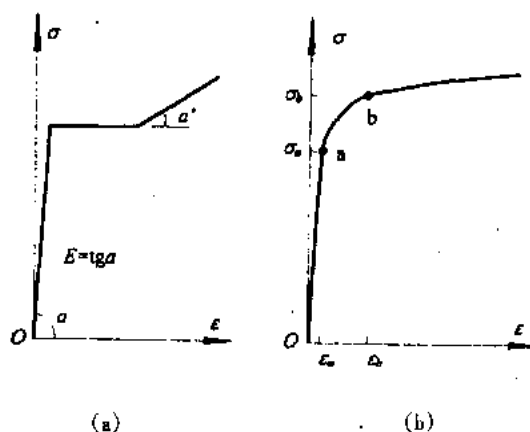


图 2.10 软钢及硬钢的应力-应变关系

零的水平线(也可把屈服台阶一段改为平坦的倾斜线);强化段可用曲线或直线表示。实

际上在钢筋混凝土构件形成塑性铰以后,由于塑性区段混凝土的极限变形很少超过 0.006,所以钢筋受拉以后的变形即使越过屈服平台进入强化段,也达不到很大范围,从而强化段可以简化为直线(斜率为 $E' = \tan \alpha'$),同级钢筋 E' 值也很分散的, Y. Higashibata 建议取 $E' = 0.01E$ 。

硬钢的应力-应变也可以分为三段:弹性段、软化段、后续段。如图 2.10(b)所示,曲线 ab 段为软化段,根据试验资料得到的应力应变关系式^[18]为:

$$\sigma = \frac{\sigma_b \varepsilon_b - \sigma_a \varepsilon_a}{\varepsilon_b - \varepsilon_a} + \frac{\varepsilon_a \varepsilon_b (\sigma_a - \sigma_b)}{\varepsilon (\varepsilon_b - \varepsilon_a)} \quad (2.12)$$

式中: $\sigma_a, \sigma_b, \varepsilon_a, \varepsilon_b$ ——分别为图 2.10(b)所示 a 和 b 点的应力及应变。

2.2.2 反复加载下,钢筋的应力-应变关系

在反复荷载作用下,软钢的卸载段往往采用以 E 为斜率的直线。Kato^[4]给出了软化段的应力应变关系式(见图 2.11(a)):

$$\left(\frac{\sigma}{\sigma_s} - a \right) \left(\frac{\varepsilon}{\varepsilon_B} + a - 1 \right) - a(1 - a) = 0 \quad (2.13)$$

式中:

$$a = \frac{E}{E - E_B};$$

$$E_B = -\frac{E}{6} \log_{10} 10 \varepsilon_s;$$

$$\varepsilon_s = \sum \Delta \varepsilon_s (= \Delta \varepsilon_{s1} + \Delta \varepsilon_{s2})$$

公式(2.13)考虑到历史上曾出现过的应变。

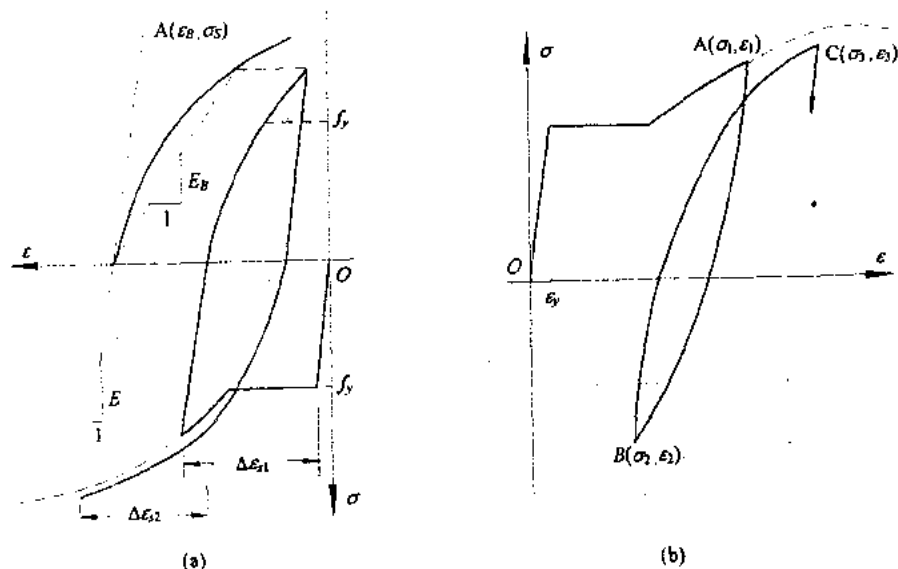


图 2.11 软钢的应力-应变模型

文献[4]介绍了除卸载线以外的软化段(包括延伸的骨架曲线)的应力应变关系式:

$$\sigma = \pm \sigma_u \frac{\epsilon - \epsilon_r}{\epsilon - \epsilon_r \pm c} \quad (2.14)$$

式中: σ_u ——钢材的极限应力, $\sigma_u = k_2 \cdot f_y$;
 k_2 ——系数;
 ϵ_r ——历史上出现过的最大残余变形;
 c ——参数。

按下式取值:

如 $\epsilon_r < 0.02$

$$c = 0.0014 + 0.0393 \frac{|\epsilon - \epsilon_r|(|\epsilon_r| + 0.001)}{(|\epsilon - \epsilon_r| + 0.06)(|\epsilon_r| + 0.0035)} \leq 0.007;$$

如 $\epsilon_r \geq 0.002$:

对应于图 2.12(a) $c = 0.007$;

对应于图 2.12(b) $c = 0.005$ 。

符号: 当 $\epsilon_r - \epsilon$ 为正时用负号, 反之则用正号。

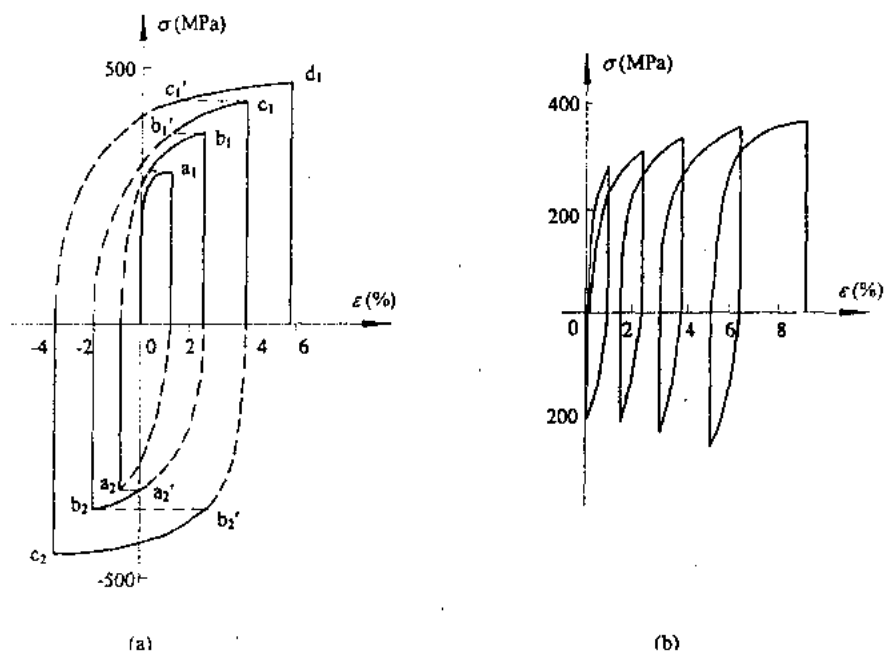


图 2.12 钢筋在反复荷载作用下的应力应变滞回环

Sozen^[4]以 Ramberg-Osgood 方程为基础给出下述应力应变关系式(见图 2.11(b)):

$$\frac{\epsilon - \epsilon_i}{\epsilon_{os}} = \frac{\sigma - \sigma_i}{\sigma_{os}} + \left(\frac{\sigma - \sigma_i}{\sigma_{os}} \right)^\alpha \quad (2.15)$$

式中, σ_i 及 ϵ_i 为半周的初始值(如 AB 的 σ_1 与 ϵ_1 , BC 的 σ_2 和 ϵ_2 等), σ_{os} 、 ϵ_{os} 及 α 为三个参数, Sozen 按不同情况分别给出了有关的数据。

2.3 混凝土的本构关系

对于混凝土的应力应变关系,不同的研究者采用不同的表达式,如多项式、二次抛物线加直线、有理分式等。本节主要讨论在有限元分析中所采用的几种表达式。

2.3.1 单调加载应力-应变关系式

1. 单向受压

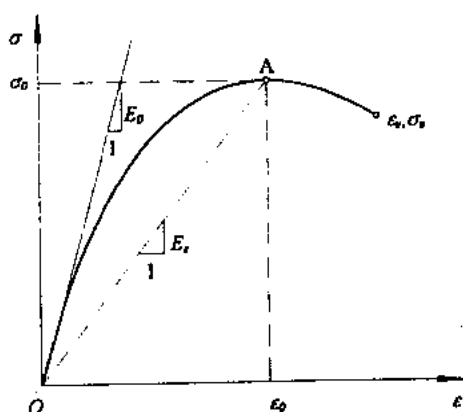


图 2.13 混凝土在轴向受压下一般的应力-应变关系

混凝土在轴向受压下一般的应力-应变关系曲线见图 2.13。

Saenz 建立了一般性的公式:

$$\sigma = \frac{E\epsilon}{A + B\left(\frac{\epsilon}{\epsilon_0}\right) + C\left(\frac{\epsilon}{\epsilon_0}\right)^2 + D\left(\frac{\epsilon}{\epsilon_0}\right)^3} \quad (2.16)$$

式中: E 为弹性模量; A, B, C 和 D 为常数,可由下列条件确定(图 2.13):

$$(a) \epsilon = 0 \quad \sigma = 0$$

$$(b) \epsilon = 0 \quad \frac{d\sigma}{d\epsilon} = E_0 \quad (\text{图中原点的切线模量})$$

$$(c) \epsilon = \epsilon_0 \quad \sigma = \sigma_0$$

$$(d) \epsilon = \epsilon_0 \quad \frac{d\sigma}{d\epsilon} = 0 \quad (\text{图中最高点的 } E \text{ 值等于零})$$

$$(e) \epsilon = \epsilon_u \quad \sigma = \sigma_u \quad \epsilon_u = 0.003 \sim 0.004$$

可以得到:

$$\sigma = \frac{E_0\epsilon}{1 + (R + R_E - 2)\frac{\epsilon}{\epsilon_0} - (2R - 1)\left(\frac{\epsilon}{\epsilon_0}\right)^2 + R\left(\frac{\epsilon}{\epsilon_0}\right)^3} \quad (3.17)$$

式中:

$$R = \frac{R_E(R_\sigma - 1)}{(R_\epsilon - 1)^2} - \frac{1}{R_\epsilon}$$

$$R_E = \frac{E_0}{E_s}$$

$$R_\sigma = \frac{\sigma_0}{\sigma_u}$$

$$R_\epsilon = \frac{\epsilon_u}{\epsilon_0}$$

如果忽略条件(e)可得下式:

$$\sigma = \frac{E_0\epsilon}{1 + \left[\frac{E_0}{E_s} - 2\right]\frac{\epsilon}{\epsilon_0} + \left(\frac{\epsilon}{\epsilon_0}\right)^2} \quad (2.18)$$

式(2.18)只适用于应力-应变曲线的上升段,在有限元计算中,常采用此式。

2. 单向受拉

混凝土轴向受拉应力-应变的试验研究较少,文献[4]所提出的部分试验结果见图 2.14。

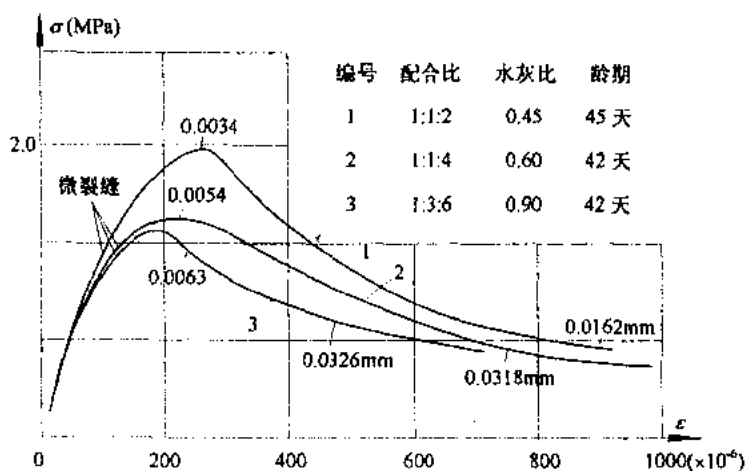


图 2.14 混凝土单向受拉时的应力-应变关系

混凝土单向受拉的应力-应变关系式可采用:两折线(图 2.15)、三段斜直线(Madu 提出的,图 2.16)和曲线型等。其表达式为:

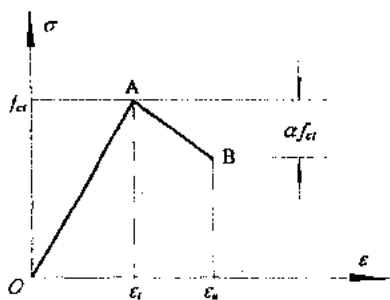


图 2.15 混凝土受拉的 σ - ϵ 关系(两折线)

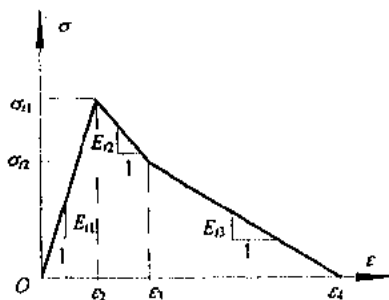


图 2.16 混凝土受拉的 σ - ϵ 关系(三段斜直线)

(a) 按两折线假定(图 2.15)

$$\left. \begin{aligned} \text{当 } 0 \leq \epsilon \leq \epsilon_t \text{ 时, } \sigma &= f_{ct} \frac{\epsilon}{\epsilon_t}; \\ \text{当 } \epsilon_t \leq \epsilon \leq \epsilon_u \text{ 时, } \sigma &= f_{ct} \left(1 - \alpha \frac{\epsilon - \epsilon_t}{\epsilon_u - \epsilon_t} \right) \end{aligned} \right\} \quad (2.19)$$

式中: ϵ_t ——对应于抗拉强度 f_{ct} 的应变;

ϵ_u ——受拉最大可用应变。

(b) 按三段斜直线假定(图 2.16)

$$\left. \begin{aligned} & \text{当 } 0 \leq \varepsilon \leq \varepsilon_2 \text{ 时} \\ & \text{当 } \varepsilon_2 < \varepsilon < \varepsilon_3 \text{ 时} \\ & \text{当 } \varepsilon_3 < \varepsilon < \varepsilon_4 \text{ 时} \end{aligned} \right\} \begin{aligned} & \sigma(\varepsilon) = \phi(\varepsilon)\varepsilon \\ & \phi_1(\varepsilon) = E_{t1} \\ & \phi_2(\varepsilon) = E_{t1} \frac{\varepsilon_2}{\varepsilon} + E_{t2} \left(1 - \frac{\varepsilon_2}{\varepsilon}\right) \\ & \phi_3(\varepsilon) = \phi_2(\varepsilon_3) \frac{\varepsilon_2}{\varepsilon} + E_{t3} \left(1 - \frac{\varepsilon_3}{\varepsilon}\right) \end{aligned} \quad (2.20)$$

(c) 按曲线假定

$$\left. \begin{aligned} & \varepsilon \leq 0.0001 \\ & 0.0001 < \varepsilon \leq 0.00015 \\ & \varepsilon > 0.00015 \end{aligned} \right\} \begin{aligned} & \sigma = f_{ct} \frac{2\varepsilon}{\varepsilon + 0.0001} \\ & \sigma = f_{ct} \\ & \sigma = 0 \end{aligned} \quad (2.21)$$

在有限元分析中,混凝土单向受拉下的应力-应变关系一般简化为线性关系。

2.3.2 重复加载应力-应变关系式

混凝土重复加载应力-应变的关系式包括骨架曲线、卸载及再加载曲线方程。骨架曲线可按上一节讨论的方法建立其关系式,本节主要讨论卸载及再加载的关系式。卸载及再加载方程可分为直线和曲线两类。

1. 直线方程:

直线方程比较简单,见 Blakeley 的模型(图 2.17)。在应用中,必须有这些线段的方程式和加、卸载的变化规则。

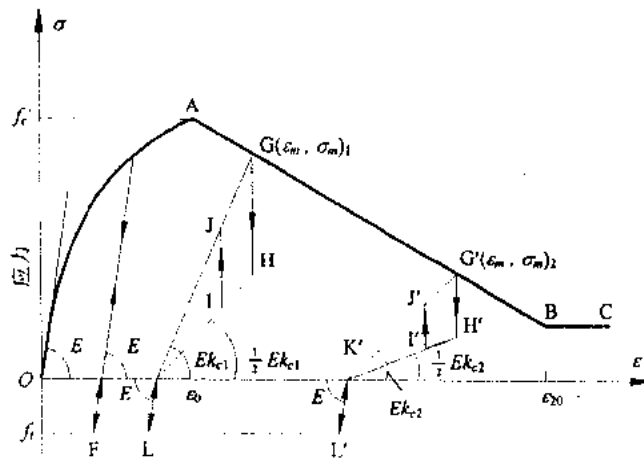


图 2.17 Blakeley 混凝土的应力-应变模型

在 Blakeley 模型中,虽考虑了卸载至混凝土受拉,但由于未超过抗拉强度,所以仍属重复加载。在该模型中,当应变小于 ε_0 时,卸载和再加载曲线都是以初始弹性模量 E 为斜率的直线;当应变大于 ε_0 时,以卸载点(G 或 G')垂直向下卸到一半(H 或 H'),然后考虑刚度退化系数 k_c 进行卸载和再加载。

k_c 与卸载点坐标有关,其计算式为:

$$k_c = 0.8 - \frac{(\epsilon_m - \epsilon_0) \times 0.7}{\epsilon_{20} - \epsilon_0} \geq 0.1 \quad (2.22)$$

式中: ϵ_{20} ——相应于最大应力只剩 20% 时的应变;

ϵ_m, ϵ_0 ——见图 2.17 所示。

2. 曲线方程

Takiguchi 及 Tanigawa 都是以 Umemura 的 e 函数曲线为骨架曲线,卸载和再加载也是用曲线方程。

文献[4]建立的卸载及再加载曲线方程(图 2.18)如下:

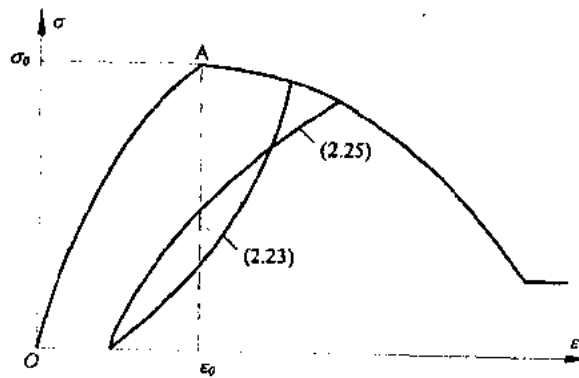


图 2.18 混凝土应力-应变模型

卸载:

$$\sigma = \frac{(\epsilon - 0.2\epsilon_1)\sigma_1}{1.8\epsilon_1 - \epsilon} \quad (2.23)$$

再加载:

$$\epsilon_1 \leq \epsilon_0: \quad \sigma = 2k_1\sigma_0 \frac{\epsilon - 0.2\epsilon_1}{\epsilon_0 + \epsilon - 0.2\epsilon_1} \quad (2.24)$$

$$2\epsilon_0 \geq \epsilon_1 > \epsilon_0: \quad \sigma = 2\sigma_1 \left(\frac{\epsilon - 0.2\epsilon_1}{\epsilon_0 + \epsilon - 0.2\epsilon_1} \right) \left(\frac{\epsilon_0}{\epsilon_1} \right) \quad (2.25)$$

$$\epsilon_1 > 2\epsilon_0: \quad \sigma = \sigma_1 \left(\frac{\epsilon - 0.2\epsilon_1}{\epsilon_0 + \epsilon - 0.2\epsilon_1} \right) \quad (2.26)$$

式中: ϵ_1, σ_1 ——卸载时的应变和应力;

k_1 ——系数, 0.8~1;

σ_0 ——混凝土的极限应力。

2.3.3 反复加载应力-应变关系

在承受反复荷载的钢筋混凝土结构中,对于重新受压的已开裂截面和新受压的未裂截面,有不同的受力特性。裂面重新受压时,在裂缝闭合的过程中,由于骨料的咬合作用而传递压力,所以在周期性反复荷载下,应力-应变曲线方程要考虑裂面效应。

考虑了裂面效应的等效应力-应变模型见图 2.19。该模型的骨架曲线相同于图 2.18, 下面主要讨论卸载和再加载关系式。

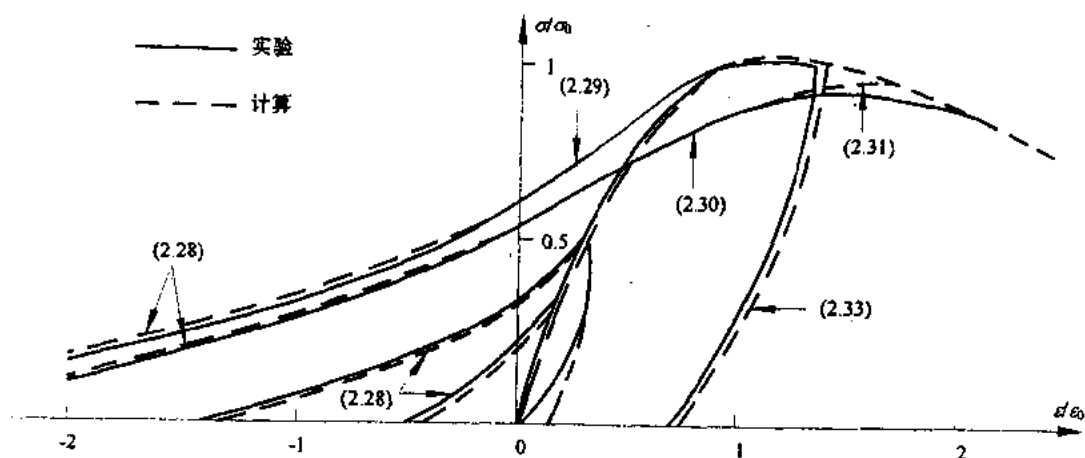


图 2.19 混凝土考虑裂面效应的等效应力-应变模型

设再加载时对应于最大裂缝宽度的应变为 $\epsilon_{\max}$, 对应于开始产生裂缝效应的裂缝宽度为 ϵ_w ; 由试验资料可得两者的关系:

$$\epsilon_w = \epsilon_{\max} \left(0.1 + \frac{0.9\epsilon_0}{\epsilon_0 + \epsilon_{\max}} \right) \quad (2.27)$$

1. 再加载方程:

$$\text{当 } \epsilon_w \leq \epsilon \leq 0 \quad \sigma = \sigma_{\text{con}} \left(1 - \frac{2\epsilon}{\epsilon_w + \epsilon} \right) \quad (2.28)$$

当 $\epsilon > 0$:

$$\text{如 } \epsilon_1 \leq \epsilon_0 \quad \sigma = \sigma_{\text{con}} (1 - \epsilon/\epsilon_0) + \frac{2\epsilon}{\epsilon_0 + \epsilon} \sigma_0 \quad (2.29)$$

$$\text{如 } \epsilon_1 > \epsilon_0 \text{ 及 } \epsilon < \epsilon_1 \quad \sigma = \sigma_{\text{con}} \left(1 - \frac{\epsilon}{\epsilon_1} \right) + \frac{2\epsilon}{\epsilon_1 + \epsilon} \left(\frac{2\epsilon_0}{\epsilon_1 + \epsilon_0} \right) \sigma_1 \quad (2.30)$$

$$\text{如 } \epsilon_1 > \epsilon_0 \text{ 及 } \epsilon > \epsilon_1 \quad \sigma = \frac{2\epsilon}{\epsilon_1 + \epsilon} \left(\frac{2\epsilon_0}{\epsilon_1 + \epsilon_0} \right) \sigma_1 \quad (2.31)$$

式中: σ_{con} ——对应于 $\epsilon = 0$ 时的接触应力:

$$\sigma_{\text{con}} = 0.3\sigma_0 \left[2 + \frac{(\epsilon_w/\epsilon_0 - 4)}{(\epsilon_w/\epsilon_0 + 2)} \right] \quad (3.32)$$

2. 卸载方程:

$$\text{如 } \epsilon_1 > \epsilon_0 \quad \sigma = \left(\frac{2\epsilon - \epsilon_1}{3\epsilon_1 - 2\epsilon} \right) \sigma_1 \quad (2.33)$$

2.3.4 混凝土双向受力下应力-应变关系

混凝土在双向受力下的试验数据比较多, 试验结果表明, 混凝土的应变大小与应力状态的性质(受压或受拉)等有关。为了既方便又能较好地反映混凝土在双向受力下的应力

应变关系,可采用非线性弹性模式,该模式属于经验型,计算简单,适用于单调加载和混凝土受压区非线性变形阶段,在钢筋混凝土结构有限元分析中应用较多。此外,也可采弹塑性模式,断裂力学模式等。

1. 非线性弹性本构关系

非线性弹性本构模型有两种基本形式:

- (a) 全量式应力应变形式;
- (b) 增量式应力应变形式。

全量式应力应变形式仅仅适用按比例一次加载的情况。在钢筋混凝土有限元分析中,非线性弹性本构关系——增量型的模式应用较多,下面介绍其中的几个模式。

(1) Darwin 和 Pecknold 模式^[19]

Darwin 和 Pecknold 模式的基本假定为:混凝土为正交各向异性材料,并且在各级荷载增量内应力-应变成线弹性关系,其应力增量与应变增量关系式为:

$$\begin{Bmatrix} d\sigma_1 \\ d\sigma_2 \\ d\tau_{12} \end{Bmatrix} = \frac{1}{1-\nu_1\nu_2} \begin{bmatrix} E_1 & \nu_2 E_1 & 0 \\ \nu_1 E_2 & E_2 & 0 \\ 0 & 0 & (1-\nu_1\nu_2)G \end{bmatrix} \begin{Bmatrix} d\epsilon_1 \\ d\epsilon_2 \\ d\gamma_{12} \end{Bmatrix} \quad (2.34)$$

式中: E_1, E_2 ——施加一级荷载后在主应力方向的等效切线模量;

$d\sigma_1, d\sigma_2, d\tau_{12}$ ——由荷载增量引起的应力增量;

$d\epsilon_1, d\epsilon_2, d\gamma_{12}$ ——由荷载增量引起的应变增量;

ν_1, ν_2 ——由于在方向 1,2 受力而对方向 2,1 所产生影响的泊松比值。

各向异性体弹性力学的基本关系式为:

$$\nu_1 E_2 = \nu_2 E_1$$

并近似取:

$$\nu^2 = \nu_1 \nu_2$$

这样,式(2.34)就成为:

$$\begin{Bmatrix} d\sigma_1 \\ d\sigma_2 \\ d\tau_{12} \end{Bmatrix} = \frac{1}{1-\nu^2} \begin{bmatrix} E_1 & \nu \sqrt{E_1 E_2} & 0 \\ \nu \sqrt{E_1 E_2} & E_2 & 0 \\ 0 & 0 & (1-\nu^2)G \end{bmatrix} \begin{Bmatrix} d\epsilon_1 \\ d\epsilon_2 \\ d\gamma_{12} \end{Bmatrix} \quad (2.35)$$

公式(2.35)中, E_1, E_2, G, ν 的取值方法如下所述。

(a) 剪切模量 G 的取值

由于缺少有关的试验资料,作出了一些假定后, G 与 E_1, E_2 值的关系式可取为:

$$(1-\nu^2)G = \frac{1}{4}(E_1 + E_2 - 2\nu\sqrt{E_1 E_2})$$

则式(2.35)变为:

$$\begin{Bmatrix} d\sigma_1 \\ d\sigma_2 \\ d\tau_{12} \end{Bmatrix} = \frac{1}{1-\nu^2} \begin{bmatrix} E_1 & \nu \sqrt{E_1 E_2} & 0 \\ \nu \sqrt{E_1 E_2} & E_2 & 0 \\ 0 & 0 & \frac{1}{4}(E_1 + E_2 - 2\nu\sqrt{E_1 E_2}) \end{bmatrix} \begin{Bmatrix} d\epsilon_1 \\ d\epsilon_2 \\ d\gamma_{12} \end{Bmatrix} \quad (2.36)$$

(b) 泊松比 ν 的取值

ν 值可由试验得出, 在 0.15~0.20 之间, ν 值的大小对于计算结果影响相对较小。

Darwin 和 Pecknold 建议:

双向受压时:

$$\nu = 0.2$$

$$\text{一向受压、一向受拉时: } \nu = 0.2 + 0.6 \left(\frac{\sigma_2}{f_c} \right)^4 + 0.4 \left(\frac{\sigma_1}{f_c} \right)^4 \quad (2.37)$$

(c) E_1 及 E_2 的取值

E_1 及 E_2 可由试验得出, 但 E_1, E_2 究竟是取单向加载的试验结果还是取双向受荷下的试验结果, 两种不同的取法, 均存在一些问题。

试验研究表明, 双向受压时, 微裂缝受约束, 其刚度增长远比泊松比单独影响引起的增长大得多。如果取用正交方向单向加荷试验得出的切线模量, 公式(2.36)就意味着任一方向有效刚度由于另一方向压应力作用而产生的增长, 仅与泊松比 ν 值有关, 没有考虑微裂缝受约束的影响。如果取用双向受荷下得出的切线模量值, 将该值代入应力增量与应变增量式(2.36)中计算, 这样, 泊松比就会再一次得到重复考虑, 其原因是双向受荷下的应力应变关系式已考虑到泊松比和微裂缝受约束的影响。这在理论上是不正确的。

鉴于上述原因, 如果取用双向受荷下得出的切线模量值, 就必须在双向受荷试验得出的应力应变关系式中消除泊松比的影响因素。为此, Darwin 和 Pecknold 提出了等效单向受力应力应变关系式, 用以代表消除泊松比影响后(没有消除微裂缝影响)的应力应变关系。这些应力应变曲线随 $\alpha = \frac{\sigma_1}{\sigma_2}$ 比值的变化而异。

混凝土在双向受压应力状态下, 主应力各为 σ_1, σ_2 , 假定在每级荷载增量内应力-应变成线弹性关系, 于是可得出下式:

$$\left. \begin{aligned} d\epsilon_1 &= \frac{d\sigma_1}{E_1} - \frac{\gamma d\sigma_2}{E_2} \\ d\epsilon_2 &= \frac{d\sigma_2}{E_2} - \frac{\gamma d\sigma_1}{E_1} \end{aligned} \right\} \quad (2.38)$$

由式(2.38)中第二式加整理后可得:

$$d\sigma_2 = E_2 \frac{d\epsilon_2}{1 - \gamma \alpha n} = E_2 d\epsilon_{2u} \quad (2.39)$$

式中: $\alpha = \frac{\sigma_1}{\sigma_2}$; $n = \frac{E_1}{E_2}$; $d\epsilon_{2u} = \frac{d\epsilon_2}{1 - \gamma \alpha n}$ 为方向 2 的等效单向受力应变。

于是, 式(2.39)中的 E_2 由于等效单向受力应变 $d\epsilon_{2u}$ 的引入而消除了泊松比的影响, $d\sigma_2$ 方向的应力可由等效单向受力应变 $d\epsilon_{2u}$ 及相应的弹性模量 E_2 求得。

图 2.20 表示了这样一条曲线, 从这曲线上取得切线模量, 再加上考虑泊松比影响(通过代入式(2.36)中的泊松比计算), 就可以得到双向受力下真实的应力增量与应变增量的关系。

根据 Darwin 和 Pecknold 所作的分析, 从双向受力应力应变曲线中消除泊松比影响后所得的等效单向应力应变关系仍可用 Saenz 公式, 其形式表示为

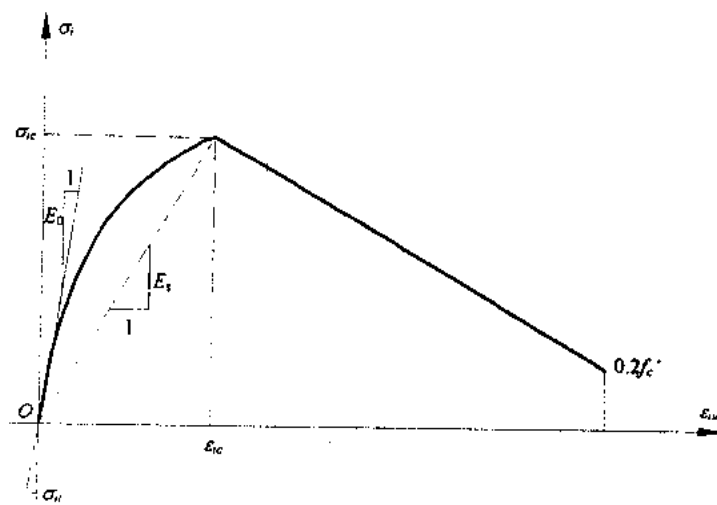


图 2.20 等效单向受力应力·应变曲线

$$\sigma_i = \frac{E_0 \epsilon_{iu}}{1 + \left[\frac{E_0}{E_s} - 2 \right] \left(\frac{\epsilon_{iu}}{\epsilon_{ic}} \right) + \left(\frac{\epsilon_{iu}}{\epsilon_{ic}} \right)^2} \quad (2.40)$$

式中: i ——主应力方向($i=1,2$);

ϵ_{ic} ——相应于最大压应力 σ_{ic} 的等效单向受力应变;

ϵ_{iu} ——等效单向受力应变, 即 $\epsilon_{iu} = \sum \frac{\Delta \sigma_i}{E_i}$;

E_s ——相应于最大压应力 σ_{ic} 的割线模量, $E_s = \frac{\sigma_{ic}}{\epsilon_{ic}}$;

E_0 ——原点切线模量, 取单向受力时的 E_0 值。

关于 σ_{ic} 的计算, Darwin 和 Pecknold 建议采用 Kupfer 和 Gerstle 所提出的公式, 即在压-压区域:

$$\left(\frac{\sigma_1}{f_c'} + \frac{\sigma_2}{f_c'} \right)^2 - \frac{\sigma_1^2}{f_c'^2} - 3.65 \frac{\sigma_1}{f_c'} = 0 \quad (2.41a)$$

$$\left. \begin{aligned} \sigma_{2c} &= \frac{1 + 3.65a}{(1+a)^2} f_c' \\ \sigma_{1c} &= a\sigma_{2c} \end{aligned} \right\} \quad (2.41b)$$

式中: f_c' ——为圆柱体抗压强度;

σ_1, σ_2 及 f_c' ——压应力均取负号。

在压-拉区域:

$$\sigma_{1c} = \left(1 - 0.8 \frac{\sigma_2}{f_c'} \right) f_c' \quad (2.42a)$$

$$\sigma_{2c} = \frac{1 + 3.28a}{(1+a)^2} f_c' \quad (2.42b)$$

式中: f'_t ——单向受力时的抗拉强度;

σ_{1t} ——拉应力;

σ_{2c} ——压应力。

在拉-拉区域:

$$\left. \begin{aligned} \sigma_{1t} &= f'_t \\ \sigma_{2t} &= f'_t \end{aligned} \right\} \quad (2.43)$$

在上述公式的基础上,作了修改后的计算式为:

在压-压区域($\sigma_1 = \text{压}, \sigma_2 = \text{压}, 0 \leq a \leq 1$)

$$\left. \begin{aligned} \sigma_{2c} &= \frac{1+3.65a}{(1+a)^2} f'_c \\ \sigma_{1c} &= a\sigma_{2c} \end{aligned} \right\} \quad (2.44)$$

在压-拉区域($\sigma_1 = \text{拉}, \sigma_2 = \text{压}, -0.17 \leq a \leq 0$)

$$\left. \begin{aligned} \sigma_{2c} &= \frac{1+3.28a}{(1+a)^2} f'_c \\ \sigma_{1t} &= a\sigma_{2c} \end{aligned} \right\} \quad (2.45)$$

在拉-压区域($\sigma_1 = \text{拉}, \sigma_2 = \text{压}, -\infty < a < -0.17$)

$$\left. \begin{aligned} \sigma_{2c} &\leq 0.65f'_t \\ \sigma_{1t} &= f'_t \end{aligned} \right\} \quad (2.46)$$

在拉-拉区域($\sigma_1 = \text{拉}, \sigma_2 = \text{拉}, 1 < a < \infty$)

$$\left. \begin{aligned} \sigma_{1t} &= f'_t \\ \sigma_{2t} &= f'_t \end{aligned} \right\} \quad (2.47)$$

对于 ϵ_{ic} 值,则按下式确定:

$$\text{当 } |\sigma_{ic}| > |f'_c|, \quad \epsilon_{ic} = \epsilon_p \left[3 \left(\frac{\sigma_{ic}}{f'_c} \right) - 2 \right] \quad (2.48a)$$

$$\text{当 } |\sigma_{ic}| < |f'_c|, \quad \epsilon_{ic} = \epsilon_p \left[-1.6 \left(\frac{\sigma_{ic}}{f'_c} \right)^3 + 2.25 \left(\frac{\sigma_{ic}}{f'_c} \right)^2 + 0.35 \left(\frac{\sigma_{ic}}{f'_c} \right) \right] \quad (2.48b)$$

对 Saenz 式(2.40)微分,得

$$E_i = \frac{d\sigma_i}{d\epsilon_{iu}} = \frac{E_i \left[1 - \left(\frac{\epsilon_{iu}}{\epsilon_{ic}} \right)^2 \right]}{\left[1 + \left(\frac{E_0}{E_i} - 2 \right) \left(\frac{\epsilon_{iu}}{\epsilon_{ic}} \right) + \left(\frac{\epsilon_{iu}}{\epsilon_{ic}} \right)^2 \right]^2} \quad (2.49)$$

式中, E_i 为主应力方向 i 的切线模量($i = 1, 2$)。

上述切线模量值可用在双向受压、一向受拉一向受压时的受压方向。

对于双向受拉、一向受拉一向受压时的受拉方向切线模量,则仍按单向受拉一样,沿用原点的切线模量值。

在计算过程中等效单向受力应变值由加载过程中累积求得,即

$$\epsilon_{ib} = \sum \frac{\Delta \sigma_i}{E_i}$$

式中: $\Delta \sigma_i$ ——每一级荷载增量在主应力方向 i 所引起的应力增量;

E_i ——上一级荷载增量施加后,在主应力方向 i 的切线模量值;

(2) Liu, Nilson 和 Slate 模式

Liu 等假定混凝土为正交各向异性材料,并认为在每级荷载增量中,应力-应变成线弹性关系。该模式适用于应力-应变曲线的上升段。

根据上述假定, Liu 等也采用了同公式(2.34)的应力增量与应变增量关系式。

Liu 等采用的双向受压下的应力应变关系表达式为:

$$\sigma_i = \frac{E_0 \epsilon_i}{(1 - \nu a_i) \left[1 + \left(\frac{1}{1 - \nu a_i} \cdot \frac{E_0}{E_s} - 2 \right) \left(\frac{\epsilon_i}{\epsilon_{ic}} \right) + \left(\frac{\epsilon_i}{\epsilon_{ic}} \right)^2 \right]} \quad (2.50)$$

式中: E_0 ——单向受力时的初始切线模量;

ν ——单向受力时的泊松比;

σ_{ic} ——主要主应力(数值较大的应力值),单位 MPa。

a_i ——垂直于 i 方向主应力与 i 方向主应力的比值;

ϵ_{ic} ——相应于最大应力 σ_{ic} 的应变;

当 $a_i \leq 1$ 时, $\epsilon_{ic} = -0.0025$;

当 $a_i > 1$ 时, $\epsilon_{ic} = (500 - 78\sigma_{ic}) \times 10^{-6}$;

E_s ——割线模量, $E_s = \frac{\sigma_{ic}}{\epsilon_{ic}}$

将公式(2.50)对 ϵ_i 微分,得出:

$$E_{ib} = \frac{d\sigma_i}{d\epsilon_i} = \frac{E_0 \left[1 - \left(\frac{\epsilon_i}{\epsilon_{ic}} \right)^2 \right]}{(1 - \nu a_i) \left[1 + \left(\frac{1}{1 - \nu a_i} \cdot \frac{E_0}{E_s} - 1 \right) \left(\frac{\epsilon_i}{\epsilon_{ic}} \right) + \left(\frac{\epsilon_i}{\epsilon_{ic}} \right)^2 \right]^2} \quad (2.51)$$

如果将 E_{ib} 直接代入应力增量与应变增量公式(2.35)中计算,就会出现如上节所讲到的泊松比被重复考虑的情况,所以应消除这种影响。

Liu 等将公式(2.51)中的泊松比进行分离,分离了泊松比影响后的计算公式见(2.52)。实际上,在公式(2.52)中还留有 ν 值的部分影响,并未把 ν 值完全分离出去。

$$E'_{ib} = E_{ib} (1 - \nu a_i) = \frac{E_0 \left[1 - \left(\frac{\epsilon_i}{\epsilon_{ic}} \right)^2 \right]}{\left[1 + \left(\frac{1}{1 - \nu a_i} \cdot \frac{E_0}{E_s} - 1 \right) \left(\frac{\epsilon_i}{\epsilon_{ic}} \right) + \left(\frac{\epsilon_i}{\epsilon_{ic}} \right)^2 \right]^2} \quad (2.52)$$

E'_{ib} 为 i 方向($i = 1, 2$)有效切线模量(已分离了泊松比影响)。

把 E'_{ib} 代入应力增量与应变增量公式(2.35)中并对剪切项作了不同于 Darwin 模式的处理,可得出公式(2.53)。

$$\begin{Bmatrix} d\sigma_1 \\ d\sigma_2 \\ d\tau_{12} \end{Bmatrix} = \begin{bmatrix} \lambda' \frac{E'_{1b}}{E'_{2b}} & \lambda' \nu & 0 \\ \lambda' \nu & \lambda' & 0 \\ 0 & 0 & \frac{E'_{1b} E'_{2b}}{E'_{1b} + E'_{2b} + 2E'_{2b} \nu} \end{bmatrix} \begin{Bmatrix} d\epsilon_1 \\ d\epsilon_2 \\ d\gamma_{12} \end{Bmatrix} \quad (2.53)$$

式中: E'_{1b} ——双向受压时,用式(2.52),并取 $i=1$;

双向受拉,一向受拉、一向受压时用 E_0 ;

E'_{2b} ——双向受压时,用式(2.52),并取 $i=2$;

双向受拉时,用 E_0 ;

一向受拉、一向受压时,用单向受压切线模量,即(2.52)式按 $\alpha_i=0$ 求算,并取 $i=2$ 。

$$\lambda' = \frac{E'_{1b}}{\frac{E'_{1b}}{E'_{2b}} - \nu^2}$$

泊松比 ν 值,由单向受力试验结果取为 0.2。

对于一向受拉、一向受压情况,当受拉方向开裂后, $E'_{1b}=0$ 时,无法再利用式(2.53)求算应力增量,此时可采用 Cedolin 和 Poli 有关剪切模量的表达式,公式(2.53)改为:

$$\begin{Bmatrix} d\sigma_1 \\ d\sigma_2 \\ d\tau_{12} \end{Bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & E'_{2b} & 0 \\ 0 & 0 & 0.1E_0 \left(1 - \frac{\epsilon_1}{\epsilon_p}\right) \end{bmatrix} \begin{Bmatrix} d\epsilon_1 \\ d\epsilon_2 \\ d\gamma_{12} \end{Bmatrix} \quad (2.54)$$

其中: ϵ_1 为垂直于裂缝方向的应变; ϵ_p 为单向受压的极限应变;其他符号同前。

(3) Kupfer 和 Gerstle 模式

Kupfer 等模式将混凝土视作各向同性体,根据试验结果,采用非线性的剪切模量和体积模量得到非线性弹性本构关系。

按弹性体假定,剪切模量 G 和体积模量 K 分别为:

$$\left. \begin{aligned} G &= E/2(1+\nu) \\ K &= E/3(1-2\nu) \end{aligned} \right\} \quad (2.55)$$

式中: ν ——泊松比。

Kupfer 等人根据试验资料,得出 K_s 及 G_s 的变化规律有如下的关系式:

$$\left. \begin{aligned} \frac{G_s}{G_0} &= 1 - a \left(\frac{\tau_0}{f_{cu}} \right)^m \\ \frac{K_s}{K_0} &= \frac{G_s/G_0}{e^{-s}} \end{aligned} \right\} \quad (2.56)$$

式中: $s = (c\gamma_0)^p$;

G_0, K_0 ——原点的剪切模量与体积模量;

G_s, K_s ——剪切割线模量与体积割线模量,它随八面体应变的增大而减少,见图

2.21;

τ_0, γ_0 ——八面体剪应力及八面体剪应变。可按下式计算：

$$\tau_0 = \frac{\sqrt{2}}{3} \sqrt{\sigma_1^2 - \sigma_1 \sigma_2 + \sigma_2^2}$$

$$\gamma_0 = \frac{1}{3} \sqrt{(\epsilon_1 - \epsilon_2)^2 + (\epsilon_2 - \epsilon_3)^2 + (\epsilon_3 - \epsilon_1)^2}$$

f_{cu} ——混凝土单向受压强度；

m, a, c, p ——常数, 由试验数据统计求出的结果见表 2-1 及表 2-2。

表 2-1

f_{cu} (MPa)	G_0/f_{cu}	a	m
19.1	650	2.9	2.0
32.4	425	3.4	2.4
61.9	261	5.3	4.0

表 2-2

f_{cu} (MPa)	K_0/f_{cu}	C	p
19.1	830	470	1.90
32.4	556	420	2.20
61.9	415	500	3.35

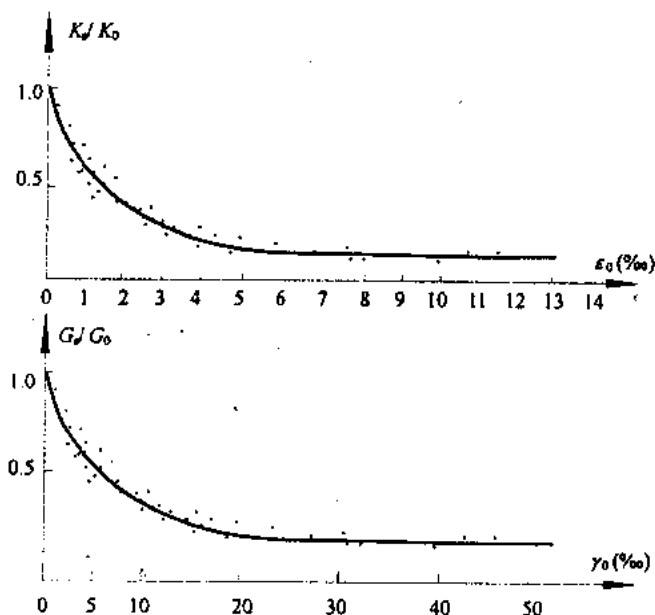


图 2.21 K 及 G 的变化规律

Kupfer 等人同时又提出下列公式：

$$\left. \begin{aligned} \frac{G_T}{G_0} &= \frac{\left[1 - a \left(\frac{\tau_0}{f_{cu}} \right)^m \right]^2}{1 + (m-1) a \left(\frac{\tau_0}{f_{cu}} \right)^m} \\ \frac{K_T}{K_0} &= \frac{G_T/G_0}{e^{-s} [1 - ps]} \end{aligned} \right\} \quad (2.57)$$

式中: $s = (c\gamma_0)^p$;

G_T ——剪切切线模量;

K_T ——体积切线模量。

利用公式(2.55)可得到 E, ν 与 G, K 关系:

$$\left. \begin{aligned} E &= \frac{6GK}{2(G+3K)} \\ \nu &= \frac{3K-2G}{2(G+3K)} \end{aligned} \right\} \quad (2.58)$$

在有限元计算中,求解非线性问题的计算方法有增量法,迭代法(有关求解方法将在第四章中详细说明)等。如果采用迭代法解非线性问题,需采用割线模量;如果采用增量法求解,需采用切线模量。

用迭代法求解时,可用 G_s, K_s 代替公式(2.58)中的 G 和 K ,再代入公式 $\{\sigma\} = [D]\{\epsilon\}$ 及 $[D]$ 的计算式,可得出应力-应变关系式:

$$\begin{Bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{Bmatrix} = 4G_s \frac{3K_s + G_s}{3K_s + 4G_s} \begin{bmatrix} 1 & \frac{3K_s - 2G_s}{2(3K_s + G_s)} & 0 \\ \frac{3K_s - 2G_s}{2(3K_s + G_s)} & 1 & 0 \\ 0 & 0 & \frac{3K_s + 4G_s}{4(3K_s + G_s)} \end{bmatrix} \begin{Bmatrix} \epsilon_x \\ \epsilon_y \\ \gamma_{xy} \end{Bmatrix} \quad (2.59)$$

用增量法求解时,可用 G_T, K_T 代替公式(2.59)中的 G_s 和 K_s ,即可求出应力-应变关系式。Kupfer 等模式的形式简单,但在实用上受到较大的限制,其原因主要是根据部分试验数据得出的参数和规律有一定的适用范围,当应力水平较高时,本模式的计算值与试验值相差较大。

2. 弹塑性本构模型

经典的塑性理论目前主要有两种,一是形变理论,二是增量理论。形变理论是弹塑性小变形理论的简称,该理论试图直接建立全量式应力应变关系,形变理论的数学处理比较简单,但理论上并非最好,而且仅适用于简单加载情况(在按比例加载的情况下可以得到较理想的结果)。在电子计算机得到广泛应用的情况下,形变理论已较少采用。

增量理论又称流动理论,是描述材料在塑性状态时应力与应变速度或应变增量之间关系的理论,这一理论在实际应用中需要按加载过程积分,计算比较复杂,但随着电子计算机的发展和计算方法的改进,增量理论得到广泛的应用。

(1) Mises 模式

在讨论该模式之前,先简单介绍塑性力学中的几个定义以及增量塑性理论的几个基本假定。

(a) 等效应力 $\bar{\sigma}$ 和等效塑性应变增量 $d\bar{\epsilon}_p$

仅仅是为了应用方便而引用这两个量。 $\bar{\sigma}$ 并不表示作用在那个面上的应力, $d\bar{\epsilon}_p$ 也不是塑性应变强度的全微分。

$$\begin{aligned}\bar{\sigma} &= \frac{1}{\sqrt{2}} \sqrt{(\sigma_x - \sigma_y)^2 + (\sigma_y - \sigma_z)^2 + (\sigma_z - \sigma_x)^2 + 6(\tau_{xy}^2 + \tau_{yz}^2 + \tau_{zx}^2)} \\ &= \frac{1}{\sqrt{2}} \sqrt{(\sigma_1 - \sigma_2)^2 + (\sigma_2 - \sigma_3)^2 + (\sigma_3 - \sigma_1)^2}\end{aligned}\quad (2.60)$$

其中: $\sigma_x, \sigma_y, \sigma_z, \tau_{xy}, \tau_{yz}, \tau_{zx}$ 为各个应力分量; $\sigma_1, \sigma_2, \sigma_3$ 为主应力。

$\bar{\sigma}$ 与八面体剪应力关系为:

$$\bar{\sigma} = \frac{3}{\sqrt{2}} \tau_0 \quad (2.61)$$

$$\begin{aligned}d\bar{\epsilon}_p &= \frac{\sqrt{2}}{3} \sqrt{(d\epsilon_{xp} - d\epsilon_{yp})^2 + (d\epsilon_{yp} - d\epsilon_{zp})^2 + (d\epsilon_{zp} - d\epsilon_{xp})^2 + \frac{3}{2}(d\gamma_{xyp}^2 + d\gamma_{yzp}^2 + d\gamma_{zxp}^2)} \\ &= \frac{\sqrt{2}}{3} \sqrt{(d\epsilon_{1p} - d\epsilon_{2p})^2 + (d\epsilon_{2p} - d\epsilon_{3p})^2 + (d\epsilon_{3p} - d\epsilon_{1p})^2}\end{aligned}\quad (2.62)$$

其中: $d\epsilon_{xp}, d\epsilon_{yp}, d\epsilon_{zp}, d\gamma_{xyp}, d\gamma_{yzp}, d\gamma_{zxp}$ 为各个塑性应变分量的增量; $d\epsilon_{1p}, d\epsilon_{2p}, d\epsilon_{3p}$ 为各个主应变为的塑性应变增量。

(b) 屈服条件

Mises 模式的屈服条件为:

$$F(\bar{\sigma}) = \bar{\sigma} - K = 0 \quad (2.63)$$

式中: K 为单向受力时的屈服应力。它表示在复合应力作用下,当等效应力达到单向受力的屈服应力时,材料屈服。

在主应力空间中 Mises 模式的屈服面为一个圆柱面,如图 2.22 所示。

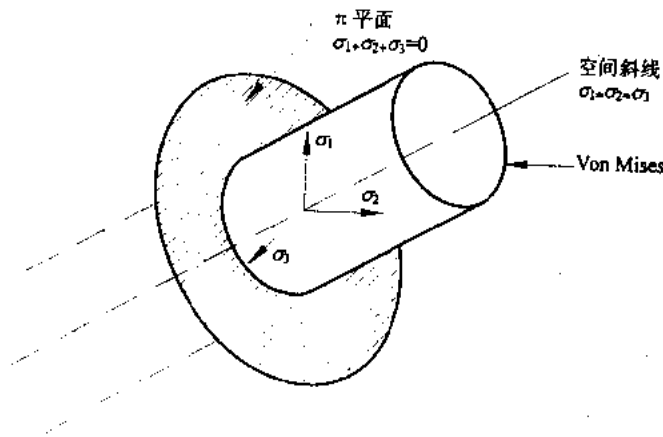


图 2.22 主应力空间中 Mises 模式的屈服面

(c) 硬化规律

材料屈服后,随着塑性变形增加,应力也同时增大。卸载后再加载,后继屈服极限将比初始屈服极限为高,这就是硬化现象。此处仅引用各向同性硬化理论,即后继屈服面随着硬化程度逐渐发生大小变化,但不发生形状和位置变化。

各向同性硬化理论有一种假设认为硬化程度只是总塑性功 W_p 的函数,而与应力路径无关。

$$\bar{\sigma} = F(W_p) \quad (2.64)$$

实际上,硬化程度不仅取决于单元体的最初和最终状态之差,也与应变路径有关,所以采用塑性应变增量 $d\bar{\epsilon}_p$,并沿应变路径积分 $\int d\bar{\epsilon}_p$ 来反映硬化程度。

$$\bar{\sigma} = H_0' \left(\int d\bar{\epsilon}_p \right) \quad (2.65)$$

式中: H_0' 为应变硬化系数,具体数值可由单向受力时的应力与塑性应变关系确定之。新的屈服只有在等效应力满足上述关系时才发生,利用微分关系,可得出下列硬化规律:

$$d\bar{\sigma} = H_0' d\bar{\epsilon}_p \quad (2.66)$$

(d) 流动法则

流动法则也称正交定律,它描述了塑性应变增量各分量与应力间的相互关系,也就是确定塑性应变增量方向的一条规定。在主应力空间中可将塑性应变能相同的点连起来形成的曲面,称之为塑性势面。流动法则规定任意点处的塑性应变增量,其方向总是与塑性势面正交。流动法则是为计算而作的一种假定,其塑性势面可以假设为各种不同的形式。在塑性理论中,如果塑性势面与屈服面相重合,那么这个流动法则就称为相关联(亦称相适应)的流动法则,反之称为不相关联的流动法则。

相关联的流动法则比较适用于金属材料,而对于混凝土,不少学者认为应该采用不相关联的流动法则,但为了方便,在实际结构分析中往往仍然采用相关联的流动法则。

如果采用相关联的流动法则,可以将塑性应变增量表示成与屈服函数对应力微分成比例的关系式,即:

$$d\{\epsilon\}_p = \lambda \left\{ \frac{\partial F}{\partial \sigma} \right\} \quad (2.67a)$$

若采用 Mises 模式的屈服函数(见公式 2.63),则上式可改写为:

$$d\{\epsilon\}_p = \lambda \left\{ \frac{\partial \bar{\sigma}}{\partial \sigma} \right\} \quad (2.67b)$$

其中: $d\{\epsilon\}_p$ 为 $\{d\epsilon_{xp}, d\epsilon_{yp}, d\epsilon_{zp}, d\gamma_{xyp}, d\gamma_{xyp}, d\gamma_{xyp}\}^T$; $\left\{ \frac{\partial \bar{\sigma}}{\partial \sigma} \right\}$ 为 $\left\{ \frac{\partial \bar{\sigma}}{\partial \sigma_x}, \frac{\partial \bar{\sigma}}{\partial \sigma_y}, \frac{\partial \bar{\sigma}}{\partial \sigma_z}, \frac{\partial \bar{\sigma}}{\partial \sigma_{xy}}, \frac{\partial \bar{\sigma}}{\partial \sigma_{yz}}, \frac{\partial \bar{\sigma}}{\partial \sigma_{zx}} \right\}^T$; λ 为待定常数。

将等效应力 $\bar{\sigma}$ 分别对 $\sigma_x, \sigma_y, \sigma_z, \tau_{xy}, \tau_{yz}, \tau_{zx}$ 求偏导数,可得出下列关系:

$$\begin{Bmatrix} \frac{\partial \bar{\sigma}}{\partial \sigma_x} \\ \frac{\partial \bar{\sigma}}{\partial \sigma_y} \\ \frac{\partial \bar{\sigma}}{\partial \sigma_z} \\ \frac{\partial \bar{\sigma}}{\partial \tau_{xy}} \\ \frac{\partial \bar{\sigma}}{\partial \tau_{yz}} \\ \frac{\partial \bar{\sigma}}{\partial \tau_{zx}} \end{Bmatrix} = \begin{Bmatrix} \frac{3\sigma'_x}{2\bar{\sigma}} \\ \frac{3\sigma'_y}{2\bar{\sigma}} \\ \frac{3\sigma'_z}{2\bar{\sigma}} \\ \frac{3\tau_{xy}}{\bar{\sigma}} \\ \frac{3\tau_{yz}}{\bar{\sigma}} \\ \frac{3\tau_{zx}}{\bar{\sigma}} \end{Bmatrix} \quad (2.68)$$

式中: $\sigma'_x = \sigma_x - \sigma_m = \sigma_x - \left(\frac{\sigma_x + \sigma_y + \sigma_z}{3} \right)$

$\sigma'_y = \sigma_y - \sigma_m = \sigma_y - \left(\frac{\sigma_x + \sigma_y + \sigma_z}{3} \right)$

$\sigma'_z = \sigma_z - \sigma_m = \sigma_z - \left(\frac{\sigma_x + \sigma_y + \sigma_z}{3} \right)$

由公式(2.67a), (2.67b)及(2.68)可以得出:

$$\left. \begin{aligned} d\epsilon_{xp} &= \lambda \cdot \frac{3}{2} \frac{\sigma'_x}{\bar{\sigma}} \\ d\epsilon_{yp} &= \lambda \cdot \frac{3}{2} \frac{\sigma'_y}{\bar{\sigma}} \\ d\epsilon_{zp} &= \lambda \cdot \frac{3}{2} \frac{\sigma'_z}{\bar{\sigma}} \\ d\gamma_{xyp} &= \lambda \cdot \frac{3\tau_{xy}}{\bar{\sigma}} \\ d\gamma_{yxp} &= \lambda \cdot \frac{3\tau_{yz}}{\bar{\sigma}} \\ d\gamma_{zxp} &= \lambda \cdot \frac{3\tau_{zx}}{\bar{\sigma}} \end{aligned} \right\} \quad (2.69)$$

将式(2.69)代入(2.62)式可得出:

$$d\bar{\epsilon}_p = \lambda \quad (2.70)$$

现在可以导出弹塑性本构模型,

达到塑性阶段以后,应力增量与应变增量的关系为:

$$d\{\sigma\} = [D]_e \cdot d\{\epsilon\}_e = [D]_e (d\{\epsilon\} - d\{\epsilon\}_p) \quad (2.71)$$

其中: $d\{\sigma\}$ 为 $\{d\sigma_x \ d\sigma_y \ d\sigma_z \ d\tau_{xy} \ d\tau_{yz} \ d\tau_{zx}\}^T$;

$d\{\epsilon\}$ 为总应变增量 $\{d\epsilon_x \ d\epsilon_y \ d\epsilon_z \ d\gamma_{xy} \ d\gamma_{yz} \ d\gamma_{zx}\}^T$;

$d\{\epsilon\}_e$ 为总应变增量的弹性部分;

$d\{\epsilon\}_p$ 为总应变增量的塑性部分。

由式(2.66)可得

$$d\bar{\sigma} = \left\{ \frac{\partial \bar{\sigma}}{\partial \sigma} \right\}^T d\{\sigma\} = H'_0 \cdot d\bar{\epsilon}_p \quad (2.72)$$

将式(2.71)代入式(2.72),可得

$$\left\{ \frac{\partial \bar{\sigma}}{\partial \sigma} \right\}^T [D]_e (d\{\epsilon\} - d\{\epsilon\}_p) = H'_0 \cdot d\bar{\epsilon}_p \quad (2.73)$$

将式(2.67)代入式(2.73),并且利用 $\lambda = d\bar{\epsilon}_p$, 可得

$$\left\{ \frac{\partial \bar{\sigma}}{\partial \sigma} \right\}^T [D]_e \left(d\{\epsilon\} - d\bar{\epsilon}_p \left\{ \frac{\partial \bar{\sigma}}{\partial \sigma} \right\} \right) = H'_0 \cdot d\bar{\epsilon}_p \quad (2.74)$$

从而可得

$$d\bar{\epsilon}_p = \frac{\left\{ \frac{\partial \bar{\sigma}}{\partial \sigma} \right\}^T [D]_e d\{\epsilon\}}{H'_0 + \left\{ \frac{\partial \bar{\sigma}}{\partial \sigma} \right\}^T [D]_e \left\{ \frac{\partial \bar{\sigma}}{\partial \sigma} \right\}} \quad (2.75)$$

而

$$d\{\epsilon\}_p = \frac{\left\{ \frac{\partial \bar{\sigma}}{\partial \sigma} \right\} \left\{ \frac{\partial \bar{\sigma}}{\partial \sigma} \right\}^T [D]_e}{H'_0 + \left\{ \frac{\partial \bar{\sigma}}{\partial \sigma} \right\}^T [D]_e \left\{ \frac{\partial \bar{\sigma}}{\partial \sigma} \right\}} d\{\epsilon\} \quad (2.76)$$

$$\begin{aligned} d\{\sigma\} &= [D]_e (d\{\epsilon\} - d\{\epsilon\}_p) \\ &= \left\{ [D]_e - \frac{[D]_e \left\{ \frac{\partial \bar{\sigma}}{\partial \sigma} \right\} \left\{ \frac{\partial \bar{\sigma}}{\partial \sigma} \right\}^T [D]_e}{H'_0 + \left\{ \frac{\partial \bar{\sigma}}{\partial \sigma} \right\}^T [D]_e \left\{ \frac{\partial \bar{\sigma}}{\partial \sigma} \right\}} \right\} d\{\epsilon\} \\ &= [D]_{ep} \cdot d\{\epsilon\} \end{aligned} \quad (2.77)$$

可知弹塑性矩阵为:

$$[D]_{ep} = [D]_e - \frac{[D]_e \left\{ \frac{\partial \bar{\sigma}}{\partial \sigma} \right\} \left\{ \frac{\partial \bar{\sigma}}{\partial \sigma} \right\}^T [D]_e}{H'_0 + \left\{ \frac{\partial \bar{\sigma}}{\partial \sigma} \right\}^T [D]_e \left\{ \frac{\partial \bar{\sigma}}{\partial \sigma} \right\}} \quad (2.78)$$

在平面应力情况下, 只要将公式(2.78)中 $[D]_e$ 及 $\frac{\partial \bar{\sigma}}{\partial \sigma}$ 改成:

$$[D]_e = \frac{E}{1-\nu^2} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{bmatrix}$$

$$\left\{ \frac{\partial \bar{\sigma}}{\partial \sigma} \right\} = \left[\frac{3}{2} \frac{\sigma'_x}{\bar{\sigma}}, \frac{3}{2} \frac{\sigma'_y}{\bar{\sigma}}, \frac{3\tau_{xy}}{\bar{\sigma}} \right]^T$$

经演化后, 可得出

$$[D]_{ep} = \begin{bmatrix} \frac{E}{1-\nu^2} & \frac{\nu E}{1-\nu^2} & 0 \\ \frac{\nu E}{1-\nu^2} & \frac{E}{1-\nu^2} & 0 \\ 0 & 0 & \frac{E}{2(1+\nu)} \end{bmatrix} - \begin{bmatrix} \frac{\bar{S}_1^2}{\bar{S}} & \frac{\bar{S}_1 \bar{S}_2}{\bar{S}} & \frac{\bar{S}_1 \bar{S}_3}{\bar{S}} \\ \frac{\bar{S}_1 \bar{S}_2}{\bar{S}} & \frac{\bar{S}_2^2}{\bar{S}} & \frac{\bar{S}_2 \bar{S}_3}{\bar{S}} \\ \frac{\bar{S}_1 \bar{S}_3}{\bar{S}} & \frac{\bar{S}_2 \bar{S}_3}{\bar{S}} & \frac{\bar{S}_3^2}{\bar{S}} \end{bmatrix} \quad (2.79)$$

其中:

$$\bar{S} = \frac{4}{9} \bar{\sigma}^2 H_0' + \bar{S}_1 \bar{\sigma}_x' + \bar{S}_2 \bar{\sigma}_y' + 2 \bar{S}_3 \bar{\tau}_{xy}'$$

$$\bar{S}_1 = \frac{E}{1-\nu^2} (\sigma_x' + \nu \sigma_y')$$

$$\bar{S}_2 = \frac{E}{1-\nu^2} (\nu \sigma_x' + \sigma_y')$$

$$\bar{S}_3 = \frac{E}{1+\nu} \tau_{xy}'$$

$\sigma_x', \sigma_y', \tau_{xy}'$ 为应力偏量。

将式(2.79)代入式(2.77)即可得出逐级加载下按切线刚度法的应力增量-应变增量关系。

(2) Drucker 和 Prager 模式

对于岩土、混凝土材料,不少学者采用由 Drucker-Prager 所建议的条件,即屈服条件为:

$$F(\sigma) = \alpha I_1 + \sqrt{J_2} - K = 0 \quad (2.80)$$

式中: I_1 ——应力第一不变量, $I_1 = \sigma_x + \sigma_y + \sigma_z$;

J_2 ——应力偏量第二不变量;

$$J_2 = \frac{1}{6} [(\sigma_x - \sigma_y)^2 + (\sigma_y - \sigma_z)^2 + (\sigma_z - \sigma_x)^2 + 6\tau_{xy}^2 + 6\tau_{yx}^2 + 6\tau_{zx}^2]$$

α, K ——与材料性质有关的常数。

在主应力空间中,Drucker-Prager 屈服准则的屈服面是一个圆锥面,如图 2.23 所示。

式(2.80)中的常数 α, K 的取值对模式是否能较好地符合混凝土在双向受压下破坏的试验结果影响甚大。日本学者野口博^[4]建议:

$$\alpha = 0.07, \quad K = -\frac{0.88}{\sqrt{3}} |f_c'|$$

式中: f_c' ——混凝土棱柱体抗压强度。

采用 Drucker-Prager 屈服条件,同时又假定塑性势面与屈服面相重合,即采用相关联(亦称相适应)的流动法则也可推导相应的弹塑性矩阵 $[D]_{ep}$ 。

弹塑性矩阵 $[D]_{ep}$ 的推导过程与 Mises 模式相类似,硬化规律仍可用式(2.66) $d\bar{\sigma} = H_0' d\bar{\epsilon}_p$, 并采用相关联的流动法则。塑性应变增量与应力状态之间的关系式可写成:

$$d\{\epsilon\}_p = \lambda \left\{ \frac{\partial F}{\partial \sigma} \right\}$$

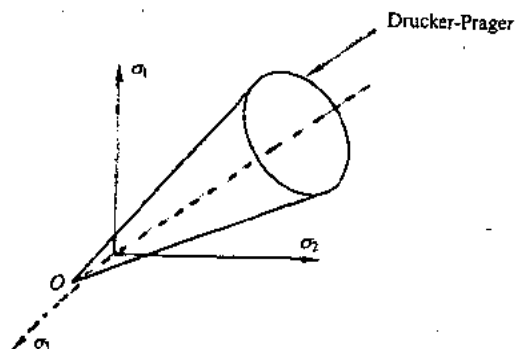


图 2.23 主应力空间中, Drucker-Prager 屈服准则的屈服面

式中: F 为屈服函数, 由式(2.80)确定,

并可推导得到

$$\lambda = d\bar{\epsilon}_p$$

塑性阶段的应力增量和应变增量的关系式为:

$$d\{\sigma\} = [D]_{ep} d\{\epsilon\}$$

其中弹塑性矩阵的推导过程类似于前面的推导过程。

$$[D]_{ep} = [D]_e - \frac{[D]_e \left\{ \frac{\partial F}{\partial \sigma} \right\} \left\{ \frac{\partial F}{\partial \sigma} \right\}^T [D]_e}{H_0 + \left\{ \frac{\partial F}{\partial \sigma} \right\}^T [D]_e \left\{ \frac{\partial F}{\partial \sigma} \right\}} \quad (2.81)$$

对于二维问题的弹塑性矩阵仍可利用式(2.79), 但此处 $\bar{S}, \bar{S}_1, \bar{S}_2, \bar{S}_3$ 需按下式计算:

$$\left. \begin{aligned} \bar{S} &= \frac{4}{9} \bar{\sigma}^2 H' + \bar{S}_1 \sigma'_x + \bar{S}_2 \sigma'_y + 2\bar{S}_3 \tau'_{xy} + \alpha \bar{\sigma} (\bar{S}_1 + \bar{S}_2) \\ \bar{S}_1 &= \frac{E}{1-\nu^2} (\sigma'_x + \nu \sigma'_y) + \bar{\alpha} \frac{E}{1-\nu} \bar{\sigma} \\ \bar{S}_2 &= \frac{E}{1-\nu^2} (\nu \sigma'_x + \sigma'_y) + \bar{\alpha} \frac{E}{1-\nu} \bar{\sigma} \\ \bar{S}_3 &= \frac{E}{1+\nu} \tau'_{xy} \end{aligned} \right\} \quad (2.82)$$

而

$$H' = \left(1 - \frac{3}{2} \bar{\alpha}\right)^2 H_0 \quad \bar{\alpha} = \frac{2}{\sqrt{3}} \alpha$$

2.4 钢筋与混凝土之间的粘结

随着有限元法在钢筋混凝土结构非线性分析中的应用, 钢筋与混凝土之间粘结和滑移的研究更显重要。

2.4.1 概述

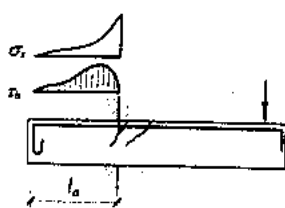
钢筋与混凝土两种不同材料共同工作的基础是它们之间具有足够的粘结强度。粘结

应力通常是指在钢筋与混凝土接触界面上所产生的沿钢筋纵向分布的剪应力,通过这种粘结作用使钢筋与混凝土两者之间可应力传递并变形协调。在钢筋混凝土非线性有限元分析中必须考虑到钢筋与混凝土之间的粘结滑移关系,考虑粘结力可能发生破坏的情况。

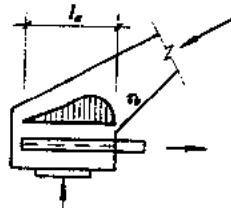
粘结应力按其在钢筋混凝土结构构件中的作用性质可分为如图 2.24,2.25 所示的两种类型:

(1) 锚固粘结应力:如图 2.24(a),(b)所示,钢筋必须有足够的锚固长度,通过这段长度上粘结应力的积累,钢筋被“锚固”在混凝土中,这样可使钢筋在未充分发挥其作用前不被拔出,以保证钢筋强度的充分发挥。

(2) 开裂截面处的粘结应力:如图 2.25 所示,开裂截面处的钢筋应力通过裂缝两侧的粘结应力向混凝土传递。



(a) 梁端锚固粘结应力



(b) 桁架端节点锚固粘结应力

图 2.24 锚固粘结应力

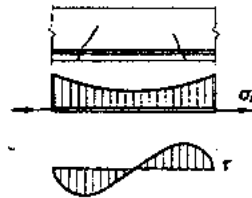


图 2.25 开裂截面处的粘结应力

量测粘结应力有不同的试验方法,如:中心拔出试验,拉伸试验、梁式试验。试验方法的选择取决于所研究的粘结应力作用的性质,试验方法对粘结应力、平均粘结应力($\bar{\tau}$)——滑移(S)关系曲线均有影响。

通过试验可以得到平均粘结应力,加载端滑移量等数据,但在本构关系上还有待进一步研究。本节主要讨论粘结机理和目前所采用的一些粘结滑移本构关系。

2.4.2 粘结机理

钢筋与混凝土之间的粘结强度主要由三部分组成:

- (a) 混凝土中水泥凝胶体与钢筋表面的化学胶着力,也称胶结力;
- (b) 混凝土收缩,将钢筋紧紧握固而产生的摩擦力;
- (c) 钢筋表面粗糙、凹凸不平与混凝土之间产生的机械咬合作用力,也称咬合力。咬合力又包括表面咬合力(光圆钢筋,螺旋钢筋)和齿肋咬合力(螺旋钢筋)。

由于钢筋表面形状(光圆或变形钢筋)的不同,在粘结受力过程中所发生的物理现象也不同。

光圆钢筋的粘结强度,在滑移前主要取决于化学胶结力,发生滑移后则取决于摩擦力和咬合力。对于变形钢筋的粘结强度,虽然胶着力和摩擦力仍然存在,但其粘结强度主要依靠钢筋表面凸出的肋与混凝土的机械咬合力。

加载初期,由胶着力承担界面上的剪应力。随着荷载增大,胶着力破坏后,钢筋开始滑移,此时粘结强度主要由钢筋表面肋对混凝土的斜向挤压力和钢筋与周围混凝土之间

的摩擦力所组成(图 2.26)。

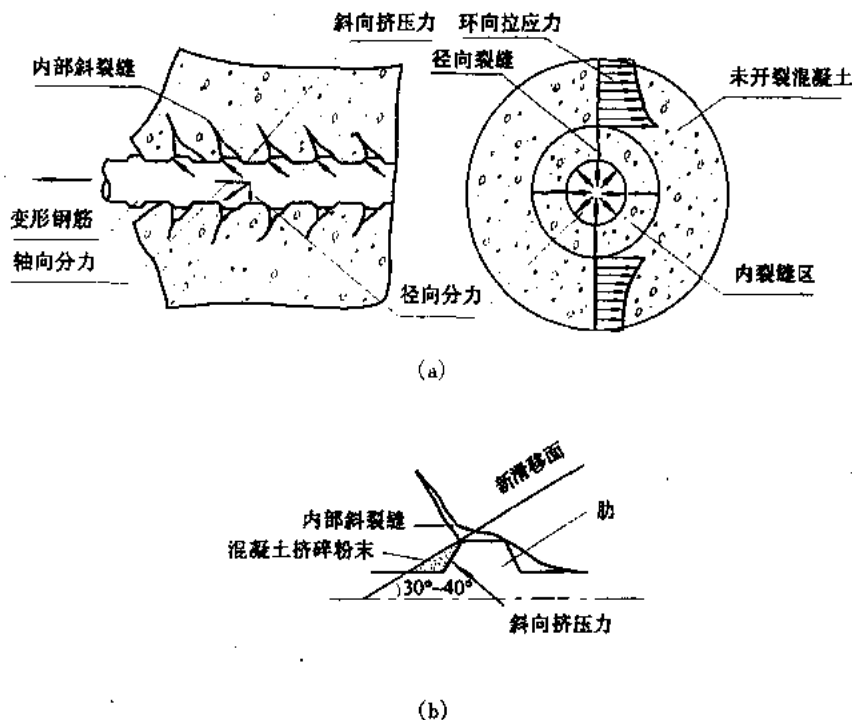


图 2.26 变形钢筋肋处的挤压力及内部裂缝

从图 2.26 中可看出:斜向挤压力沿钢筋轴线的分力使肋间的混凝土受弯受剪,其径向分力使钢筋四周的混凝土产生环向拉力。在这些力的综合作用下,当荷载引起的在主拉应力和环向拉应力方向上的应变超过混凝土的极限拉应变时,将产生如图所示的内部斜向裂缝和径向裂缝。随着荷载再增大,钢筋肋处受挤压一面的混凝土被压碎,使钢筋有可能沿混凝土挤碎后的压实粉末形成新的较平滑的滑移面而滑动。如果钢筋外围混凝土较薄,又未配置环向箍筋来约束混凝土的变形,则径向内裂缝很容易发展到构件表面,形成沿纵向钢筋的劈裂裂缝,这种裂缝发展到一定长度时,将使外围混凝土保护层崩裂,从而丧失粘结能力,此类破坏一般称之为劈裂粘结破坏。

若钢筋外围混凝土较厚,或有环向箍筋约束混凝土的变形,则纵向劈裂裂缝的发展受到抑制,使荷载能继续增加,直到肋间混凝土的剪切强度被耗尽,钢筋连同肋间的破碎混凝土逐渐被拔出,此时相对滑移量可达 1~2 mm。由于滑移面上还残存有相当大的骨料摩擦阻力和咬合力,粘结应力并不降低,粘结-滑移曲线出现较长的水平段,表现出较好的粘结延性。这种破坏的粘结强度比劈裂粘结强度大很多。

钢筋与混凝土的局部粘结应力和局部滑移关系如图 2.27 所示。

图中从 0 至 τ_0 一段主要是胶着力,滑移较小, τ_0 有 0.6 MPa 左右。图中粘结应力为 τ_A 时,由于环向拉应力小于主拉应力,初始次生斜裂缝可能出现, τ_A 点可作为初始斜裂缝出现的临界值。当粘结应力超过 τ_A 时,不仅次生斜裂缝发展,劈裂也将发生;对于光面钢筋来说,如果保护层很薄,劈裂裂缝有可能穿透保护层使粘结失效,粘结力由图中 B 点迅

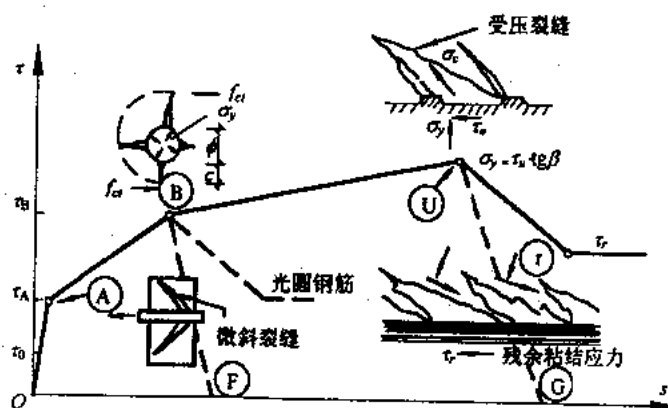


图 2.27 局部粘结应力-局部滑移关系

速下降至 F 。

对于螺纹变形钢筋, τ_A 至 τ_B 段咬合力是占主导地位; 越过 τ_B 螺纹钢筋主要靠齿肋的咬合力起作用, 受压裂缝出现在螺纹钢筋粘结应力达到 τ_u 时; 如果劈裂穿过保护层, 粘结强度则由 U 点下降至 G , 否则下降至 r 点; 而粘结强度从 τ_u 下降至残余粘结强度 τ_r 。

2.4.3 单调加载情况下, 钢筋与混凝土之间粘结和滑移的本构关系

根据单调加载试验资料和粘结机理分析, 影响粘结强度的因素有混凝土强度、浇筑混凝土的方向(与钢筋走向关系)、保护层厚度、钢筋表面形状、直径、间距、横向钢筋的配置方式等。为了方便公式推导及应用, 考虑了部分影响因素后所得到的粘结强度计算和局部粘结应力与局部滑移计算公式介绍如下。

1. Tassios 给出的粘结应力计算公式(见图 2.27)

(1) 光圆钢筋

$$\tau_A = \lambda \zeta f_{ct} \quad (2.83)$$

式中: λ ——位置系数, $\lambda = l_0/2.5c$, 当 $l_0/c > 2.5$ 时 $\lambda = 1$;

l_0 ——锚固长度;

ζ ——系数, 取决于作用在混凝土的三向应力场;

c ——净保护层厚度;

f_{ct} ——混凝土平均抗拉强度。

$$\tau_u = \tau_B < 0.4 \zeta \lambda \left(\frac{2}{3} \frac{c}{d_s} + \frac{1}{3} \right) f_{ct} + 30 \gamma \frac{A_t}{d_s a_t} + 0.4 P_y \quad (2.84)$$

式中: ζ ——系数, 混凝土浇灌方向与主筋方向一致时 $\zeta = 1$, 与主筋方向垂直时 $\zeta = 2/3$ 。

γ ——几何系数, 取决于箍筋形式, 例如: 环箍为 1, 井式箍为 1/2 或平行箍为 1/4;

A_t, a_t ——箍筋的面积和间距;

d_s ——主筋直径;

P_y ——作用于混凝土与钢筋交界面上的外部压应力。

式(2.84)中,第二项系数 30 的单位为 MPa,并要求 $\frac{c}{d_s} \leq 6$ 以及 $0.4\zeta\left(\frac{2}{3}\frac{c}{d_s} + \frac{1}{3}\right) \geq \zeta$ 。

(2) 螺纹钢筋

$$\tau_A = \lambda \zeta f_{ct} \quad (2.85)$$

$$\tau_B = \zeta \lambda \left(\frac{2}{3} \frac{c}{d_s} + \frac{1}{3} \right) f_{ct} + 80 \gamma \frac{A_t}{d_s a_t} + \frac{1}{4} P_r \quad (2.86)$$

$$\tau_u = \frac{1}{3} \zeta f_c + \frac{8}{3} \gamma \frac{A_t}{d_s a_t} f_{st} + \frac{1}{3} P_r \quad (2.87)$$

$$\tau_r = 0.12 \zeta f_c + 0.3 \frac{A_t}{d_s a_t} f_{st} \sqrt{f_{st}} + 1.4 P_r \quad (2.88)$$

$$\left(P_r \leq \frac{f_c}{6} \right)$$

式中: d_s ——主筋周围箍筋的等效直径;

f_{st} ——箍筋的屈服强度。

2. Nilson 等给出的局部粘结应力-局部滑移公式

在有限元分析中,Nilson 采用了 Bresler 和 Bertero 对试验资料分析所得的局部粘结应力-局部滑移公式,即局部 τ - s 非线性关系表达式:

$$\tau = 9.78 \times 10^2 s - 5.72 \times 10^4 s^2 + 8.35 \times 10^5 s^3 \quad (2.89)$$

式中, τ 的单位为 N/mm^2 ; s 的单位为 mm 。

Houdle 和 Mirga 通过试验提出

$$\tau = (5.3 \times 10^2 s - 2.52 \times 10^4 s^2 + 5.86 \times 10^5 s^3 - 5.47 \times 10^6 s^4) \sqrt{\frac{f_c}{40.7}} \quad (2.90)$$

式中: f_c ——混凝土抗压强度,以 N/mm^2 计; s 以 mm 计。

Nilson 所建议的公式没有考虑混凝土强度的影响。图 2.28 对公式(2.89)与(2.90)

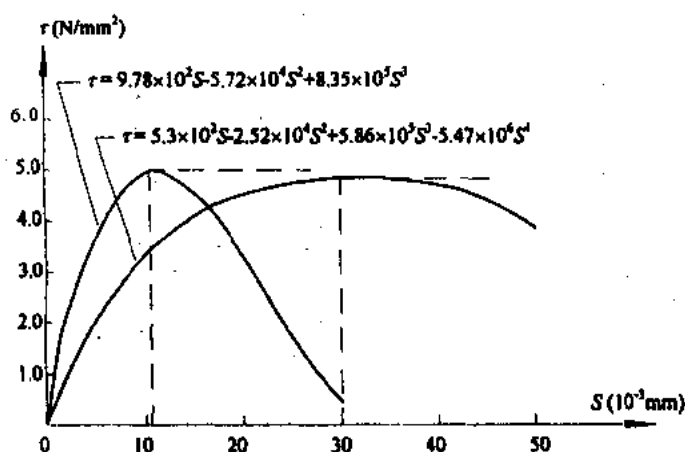


图 2.28 τ - s 公式(2.89)与(2.90)的比较

(取 $f_c = 40.7 \text{ N/mm}^2$) 所表示的粘结应力和滑移量的关系进行了比较。除上面讨论的计算公式外,多年来许多学者提出了不同形式的粘结应力,滑移表达式,例如用指数函数、三角函数、高次抛物线等表示。

Hawkins 也根据试验,给出了如图 2.29 所示的计算模型,其关系式分为三个阶段来讨论,具体计算见原文^[21]。

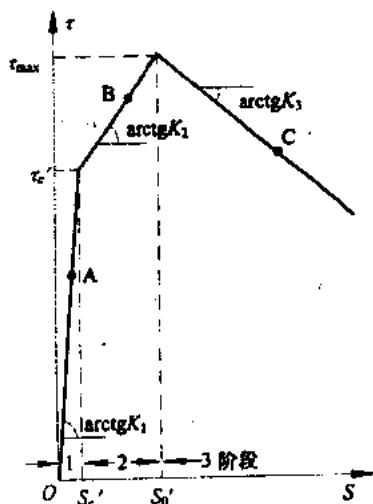


图 2.29 τ - S 计算模型

2.4.4 反复加载情况下,钢筋与混凝土之间粘结滑移的本构关系

Tassios, Hawkins 等进行了反复加载情况下的粘结和滑移试验,并与单调加载进行了对比。显然,反复加载对粘结的性能是不利的,强度降低,延性减少。根据试验资料提出了反复加载情况下的局部粘结—局部滑移的本构关系。

Tassios 提出了如图 2.30 的局部粘结应力-滑移计算模型,图中的骨架曲线取这一模型没有考虑在反复荷载下混凝土的裂面效应,只考虑钢筋的拉压。该模型在卸载后加压时,原有次生裂缝闭合,并取降低了的强度为正向的 $\frac{2}{3}$ 。 τ_{01} , τ_Q 的计算式为:

$$\tau_{01} = \frac{-1}{4}(\tau_1 + P_y); \quad (2.91)$$

$$\tau_Q = \frac{1}{4} \left(\frac{2}{3} \tau_1 + P_y \right) \quad (2.92)$$

Hawkins 也提出了出了一种粘结滑移计算模型,该模型上下大体对称,单调加载与反复加载的包络线是分开的,强度有所降低,具体详见文献[21]。

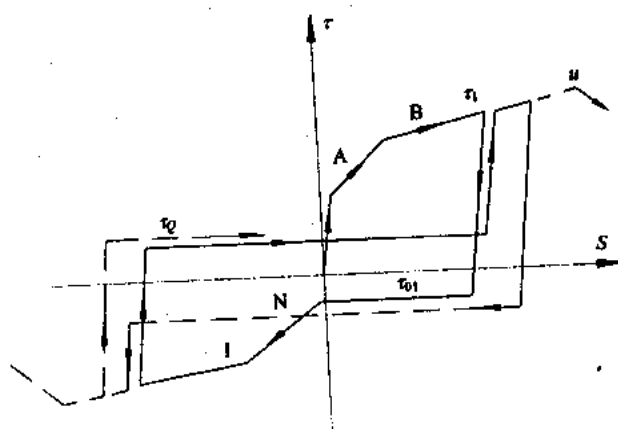


图 2.30 Tassios 粘结滑移计算模型

第三章 钢筋混凝土结构有限元分析中的几种单元

3.1 钢筋混凝土结构有限元分析计算步骤

钢筋混凝土结构有限元分析虽然有其特殊性,但与一般有限元方法一样,其计算步骤可分为三步:(1)结构的离散化;(2)单元分析;(3)总体分析。

3.1.1 结构的离散化

为了解题的需要,应对需求解的结构物进行离散化。把一个连续的实际结构物变换成离散化的有限元计算模型。这个离散化的结构计算模型由有限多个有限大小的构件,在有限多个结点处相互联系而构成。这些有限大小的构件就称为有限单元。例如图 3.1 为一开洞的简支梁,图 3.2 为离散化以后的有限元网格,图中所采用的有限单元为最简单的三结点三角形单元。

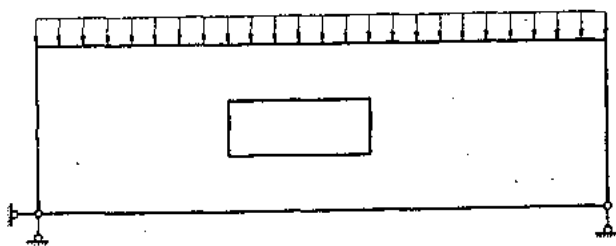


图 3.1 开洞简支梁

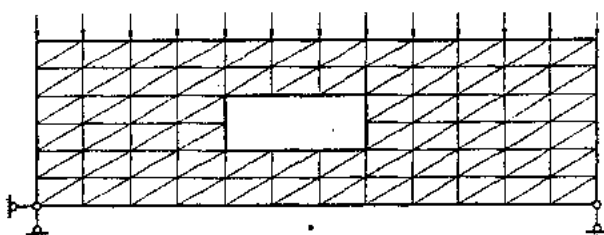


图 3.2 有限元网格

离散化工作实际上包括了以下三个方面:

1. 划分网格:选择合适的有限单元,划分网格后,将单元和结点编号。单元与单元之间用结点相连。有限元是以结点的位移作为基本未知量的。
2. 引入位移边界条件:确定支座形式和位置,在位移值为零的结点上安置铰支座或

链杆支座。

3. 荷载的移置:对受到外荷载作用的单元,为了便于分析计算,将作用在结构上的实际荷载(包括体力、面力、集中力以及温度、收缩等引起的荷载效应)按静力等效的原则移置到单元结点上,成为等效集中荷载(结点荷载)。

3.1.2 单元分析

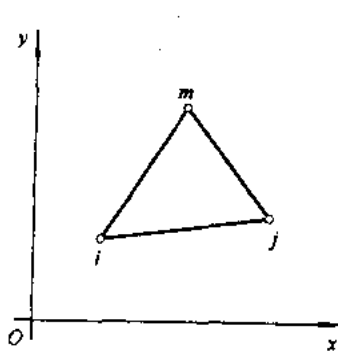
单元分析的最终目的是要建立单元结点力和单元结点位移之间的关系式。以平面问题的三结点三角形单元为例,见图 3.3,每个结点有 2 个自由度,3 个结点共有 6 个自由度,单元的结点位移列阵为

$$\{\delta\}^e = \{u_i \quad v_i \quad u_j \quad v_j \quad u_m \quad v_m\}^T \quad (3.1)$$

与 6 个基本未知量对应的 6 个结点力分量组成的单元结点力列阵为

$$\{F\} = \{x_i \quad y_i \quad x_j \quad y_j \quad x_m \quad y_m\}^T \quad (3.2)$$

单元内任一点 \$(x, y)\$ 的应力、应变和位移可表示为



$$\{\sigma\} = \begin{Bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{Bmatrix}$$

$$\{\epsilon\} = \begin{Bmatrix} \epsilon_x \\ \epsilon_y \\ \gamma_{xy} \end{Bmatrix}$$

$$\{f\} = \begin{Bmatrix} u \\ v \end{Bmatrix}$$

图 3.3 三结点三角形单元

由于有限元素法一般是以结点位移作为基本未知量,一旦得到了单元结点位移列阵,就能求得该单元中任意点的位移,应变和应力。单元分析的任务是确定以下几组关系:

(1) 单元内任意一点的位移 \$\{f\}\$ 和单元结点位移 \$\{\delta\}^e\$ 之间的关系为

$$\{f\} = [N]\{\delta\}^e \quad (3.3)$$

(2) 单元内任意一点的应变 \$\{\epsilon\}\$ 和单元结点位移 \$\{\delta\}^e\$ 之间的关系为

$$\{\epsilon\} = [B]\{\delta\}^e \quad (3.4)$$

(3) 单元内任意一点的应力 \$\{\sigma\}\$ 和单元结点位移 \$\{\delta\}^e\$ 之间的关系为

$$\{\sigma\} = [S]\{\delta\}^e \quad (3.5)$$

(4) 单元结点力 \$\{F\}^e\$ 和单元结点位移 \$\{\delta\}^e\$ 之间的关系为

$$\{F\}^e = [K]^e\{\delta\}^e \quad (3.6)$$

其中, \$[N]\$、\$[B]\$、\$[S]\$ 分别称为形函数矩阵、几何矩阵、应力矩阵。

单元分析得到的(3.6)式是有限元中最重要的关系式,得到了单元结点力与结点位移之间的关系式,是形成总体平衡方程的基础。式中的 \$[K]^e\$ 为单元刚度矩阵,它的矩阵阶数和元素取值与选择的单元形式有关,考虑到的钢筋混凝土结构构件受力后几何非线性及材料非线性的本构关系,因此在分析中需要随应力状态的变化不断调整刚度矩阵中的

有关元素。

3.1.3 总体分析

对每一个单元进行单元分析,找出每一个单元的单元刚度矩阵 $[K]^e$,在此基础上,可将各个单元组合成结构,进行总体分析。总体分析包括以下四个步骤:

- (1) 根据各个单元刚度矩阵 $[K]^e$ 组装成总体刚度矩阵 $[K]$ 。
- (2) 引入支承约束条件(边界条件)。
- (3) 建立平衡方程

$$[K]\{\delta\} = \{R\} \quad (3.7)$$

式中 $\{R\}$ ——总体外荷载列阵,由各单元外荷载列阵组装而成;

$\{\delta\}$ ——总体结点位移列阵;

$[K]$ ——由各单元刚度矩阵组装而成的总体刚度矩阵。

- (4) 求解总体平衡方程(3.7)

在求得基本未知量结点位移后,可根据式(3.3)~(3.6)式求得各单元中任意一点的位移,应变和应力。

3.2 平面单元

用有限元方法分析钢筋混凝土结构,除了要考虑钢筋混凝土结构的一些特殊性外,其基本原理和方法是和一般固体力学有限元是一致的。求解过程中所采用的有限单元类型一般也是直接采用固体力学有限元常用的类型,或者是这些类型的组合。下面讨论在钢筋混凝土结构中常用的有限单元的单元刚度矩阵。

3.2.1 三结点三角形单元

三结点三角形单元又称为简单三角形单元,是平面单元中最简单的单元形式。在早期有限元计算中是最广泛被采用的一种单元,由于其构造简单,使用方便,即使在目前,仍在有限元分析中占有一席之地。

图 3.4 表示一个在平面坐标系中的一个三结点三角形单元,按逆时针方向,其结点编号分别为 i, j, m 。单元内任意一点的 x 和 y 方向的位移分别用 u, v 来表示,简单三角形单元的结点位移向量 $\{\delta\}^e$ (如图 3.4(a)所示)可写成

$$\{\delta\}^e = \begin{Bmatrix} u_i \\ v_i \\ u_j \\ v_j \\ u_m \\ v_m \end{Bmatrix}$$

与其相对应的六个结点力分量(如图 3.4(b)所示),结点力向量 $\{F\}^e$ 可写成

$$\{F\}^e = \begin{Bmatrix} X_i \\ Y_i \\ X_j \\ Y_j \\ X_m \\ Y_m \end{Bmatrix}$$

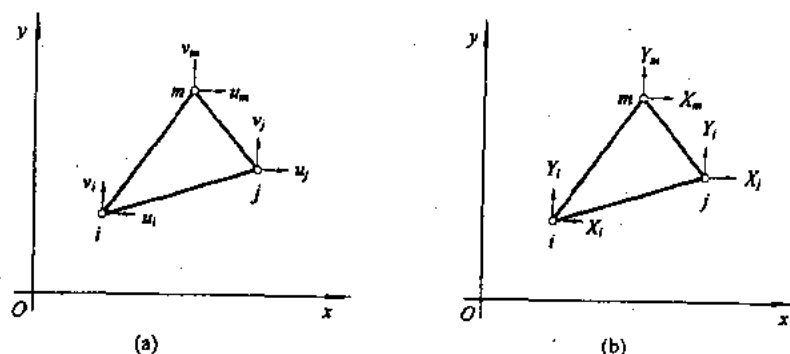


图 3.4 三结点三角形单元结点位移及结点力

单元分析的主要任务是推导 $\{\delta\}^e$ 与 $\{F\}^e$ 之间的转换关系,即

$$\{F\}^e = [K]\{\delta\}^e$$

推导上式的目的是找出下列各个量之间的转换关系:

- (1) 结点位移向量 $\{\delta\}^e$ 与单元内任意点位移向量 $\{f\}$;
- (2) 结点位移向量 $\{\delta\}^e$ 与单元应变 $\{\epsilon\}$;
- (3) 单元应变 $\{\epsilon\}$ 与应力 $\{\sigma\}$;
- (4) 单元应力 $\{\sigma\}$ 与结点力 $\{F\}$ 。

最后综合起来即可得出由结点位移求结点力的关系式,从而求出单元刚度矩阵 $[K]$ 。

1. 位移模式的选择

为了求单元内任意点 (x, y) 的位移 u 和 v , 可把 u, v 假设为坐标 x, y 的某种函数, 最简便的位移模式是假设位移为坐标的线性函数。

对简单三角形单元的位移模式可选择一次多项式, 即

$$\begin{cases} u = \alpha_1 + \alpha_2 x + \alpha_3 y \\ v = \alpha_4 + \alpha_5 x + \alpha_6 y \end{cases} \quad (3.8)$$

式中 α_1 至 α_6 是待定常数, 方程(3.8)表明位移在单元内和沿单元的边界都按线性变化的。待定常数可用结点位移值来表示。以 $i, j, m, 3$ 点的坐标值代入(3.8)式后, 得:

$$\begin{cases} u_i = \alpha_1 + \alpha_2 x_i + \alpha_3 y_i \\ v_i = \alpha_4 + \alpha_5 x_i + \alpha_6 y_i \end{cases} \quad (i, j, m) \quad (3.9)$$

(i, j, m) 表示轮换码, 实际上(3.9)式共表示 6 个方程式。待定常数 $\alpha_1 \sim \alpha_6$

$$\left. \begin{aligned} \alpha_1 &= \frac{1}{2A} \sum a_i u_i \\ \alpha_2 &= \frac{1}{2A} \sum b_i u_i \\ \alpha_3 &= \frac{1}{2A} \sum c_i u_i \end{aligned} \right\} \quad (a)$$

式中, $a_i = x_j y_m - x_m y_i; b_i = y_i - y_m; c_i = x_m - x_j \quad (i, j, m)$ (b)

$$A = \frac{1}{2} \begin{vmatrix} 1 & x_i & y_i \\ 1 & x_j & y_j \\ 1 & x_m & y_m \end{vmatrix}$$

按照解析几何, A 等于三角形 ijm 的面积, 为了得出的面积不致成为负值, 则 i, j, m 的次序必须逆时针排列。

同理

$$\left. \begin{aligned} \alpha_4 &= \frac{1}{2A} \sum a_i v_i \\ \alpha_5 &= \frac{1}{2A} \sum b_i v_i \\ \alpha_6 &= \frac{1}{2A} \sum c_i v_i \end{aligned} \right\} \quad (c)$$

将(a), (c)两式代入(3.8)式, 得

$$\left. \begin{aligned} u &= N_i u_i + N_j u_j + N_m u_m \\ v &= N_i v_i + N_j v_j + N_m v_m \end{aligned} \right\} \quad (3.10a)$$

式中, $N_i = \frac{a_i + b_i x + c_i y}{2A} \quad (i, j, m)$ (d)

位移模式表达式(3.10a)也可改写为矩阵形式如下:

$$\{f\} = \begin{Bmatrix} u \\ v \end{Bmatrix} = [N] \{\delta\}^e = [N_i N_j N_m] \{\delta\}^e \quad (3.10b)$$

式中, $[N] = \begin{bmatrix} N_i & 0 & N_j & 0 & N_m & 0 \\ 0 & N_i & 0 & N_j & 0 & N_m \end{bmatrix}$ (e)

$[N]$ 称为形态矩阵或形函数矩阵。 N_i, N_j, N_m 称为形函数或插值函数, 它们反映单元的位移形态。

可以证明形函数有以下两个性质: ① 形函数和位移函数是同阶次的; ② 形函数在本身结点上其值为1, 在其他结点上其值为零, 即 $N_i(x_i, y_i) = 1, N_i(x_j, y_j) = N_i(x_m, y_m) = 0 \quad (i, j, m)$ 。

上述位移模式满足下列两方面的条件:

- (1) 位移模式中包含有单元的刚体位移状态和常量应变;
- (2) 位移模式保证相邻单元的公共边界处(交界处)位移的连续性。

满足上述两个条件, 有限元解答一般是收敛的。

2. 单元的几何矩阵 $[B]$

对于平面应力问题,可通过弹性力学中的几何方程,将式(3.10a)代入后得:

$$\{\epsilon\} = \begin{Bmatrix} \frac{\partial u}{\partial x} \\ \frac{\partial v}{\partial y} \\ \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \end{Bmatrix} = \begin{Bmatrix} \frac{1}{2A} \sum b_i u_i \\ \frac{1}{2A} \sum c_i v_i \\ \frac{1}{2A} \sum (c_i u_i + b_i v_i) \end{Bmatrix} = \frac{1}{2A} \begin{bmatrix} b_i & 0 & b_j & 0 & b_m & 0 \\ 0 & c_i & 0 & c_j & 0 & c_m \\ c_i & b_i & c_j & b_j & c_m & b_m \end{bmatrix} \begin{Bmatrix} u_i \\ v_i \\ u_j \\ v_j \\ u_m \\ v_m \end{Bmatrix} \quad (3.11)$$

矩阵可以分块,记

$$[B_i] = \frac{1}{2A} \begin{bmatrix} b_i & 0 \\ 0 & c_i \\ c_i & b_i \end{bmatrix} \quad (i, j, m) \quad (3.12)$$

$$\text{则} \quad \{\epsilon\} = [B_i \quad B_j \quad B_m] \{\delta\}^e = [B] \{\delta\}^e \quad (3.13)$$

式中, $[B]$ 为几何矩阵; $\{\delta\}^e$ 为结点位移列阵。

$$[B] = \frac{1}{2A} \begin{bmatrix} b_i & 0 & b_j & 0 & b_m & 0 \\ 0 & c_i & 0 & c_j & 0 & c_m \\ c_i & b_i & c_j & b_j & c_m & b_m \end{bmatrix} \quad (3.14)$$

$$\{\delta\}^e = [u_i, v_i, u_j, v_j, u_m, v_m]^T$$

因为几何矩阵 $[B]$ 中每一个元素均为常数,故三结点三角形单元内各点应变均相同,即为常量应变。

3. 单元的应力矩阵 $[S]$

平面应力问题的物理方程为

$$\{\sigma\} = [D] \{\epsilon\} \quad (3.15)$$

其中 $[D]$ 称为弹性矩阵,它的元素与弹性模量 E 及泊松比 ν 有关。

$$[D] = \frac{E}{1-\nu^2} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{bmatrix} \quad (3.16)$$

对于混凝土和钢筋单元, E 和 ν 值分别采用相应于各材料的数值,混凝土的弹性模量可近似按规范根据不同强度取值,泊松比一般取用 0.15 ~ 0.2。钢筋的弹性模量可按规范根据不同钢种取值,泊松比一般取用 0.3。

将式(3.13)代入式(3.15),即可得出单元结点位移 $\{\delta\}^e$ 与单元应力 $\{\sigma\}$ 的关系式如下:

$$\{\sigma\} = \begin{Bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{Bmatrix} = [D] \{\epsilon\} = [D][B] \{\delta\}^e = [S] \{\delta\}^e \quad (3.17)$$

其中, $[S]$ 称为应力矩阵。

$$[S] = [D][B]$$

它可以分块写成:

$$[S] = [S_i \quad S_j \quad S_m]$$

其中子矩阵为

$$[S_i] = \frac{E}{2(1-\nu^2)\Delta} \begin{bmatrix} b_i & \nu c_i \\ \nu b_i & c_i \\ \frac{1-\nu}{2} c_i & \frac{1-\nu}{2} b_i \end{bmatrix}; (i, j, m)$$

因为 $[S]$ 矩阵中每一个元素都是常数, 因而简单三角形单元称为常应力单元。

对于平面应变问题, 只要把 E 换成 $\frac{E}{1-\nu^2}$, ν 换成 $\frac{\nu}{1-\nu}$, 就可用上面的结果。

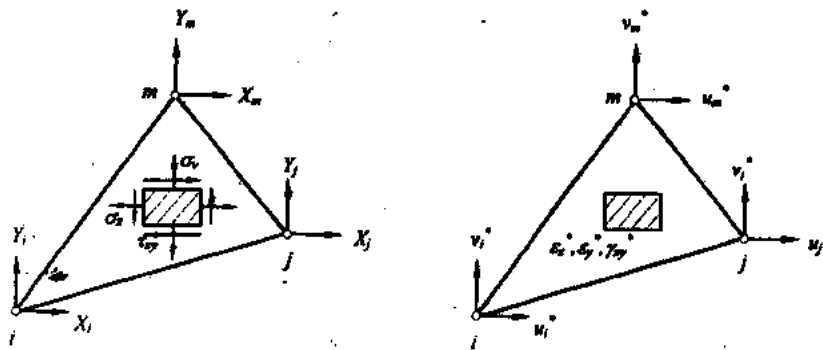
4. 单元的刚度矩阵 $[K]^e$

假设在某单元 ijm 中发生了虚位移, 相应的结点虚位移记为 $\{\delta^*\}^e$, 引起的虚应变记为 $\{\epsilon^*\}$, 因为每一个单元所受的荷载都按等效静载的原则移置到相应的结点上, 所以单元受到的外力只有结点力, 由虚功方程可得

$$(\{\delta^*\}^e)^T \{F\}^e = \iint t \{\epsilon^*\}^T \{\sigma\} dx dy$$

式中, t 为单元厚度。

图 3.5 所示为三角形单元的实际力系和虚设位移



(a) 实际力系

(b) 虚设位移

图 3.5 三角形单元的实际力系和虚设的位移状态

将 $\{\epsilon^*\} = [B]\{\delta^*\}^e$, $\{\sigma\} = [D][B]\{\delta^*\}^e$ 代入上式后, 得:

$$(\{\delta^*\}^e)^T \{F\}^e = \iint t ([B]\{\delta^*\}^e)^T [D][B]\{\delta^*\}^e dx dy$$

根据矩阵乘积的逆序法则, 上式可化为:

$$(\{\delta^*\}^e)^T \{F\}^e = \iint t (\{\delta^*\}^e)^T [B]^T [D][B]\{\delta^*\}^e dx dy$$

式中 $\{\delta^*\}^e$ 是任意的, 等式两边左乘 $((\{\delta^*\}^e)^T)^{-1}$, 且把 $\{\delta^*\}^e$ 提到积分号外后, 得:

$$\{F\}^e = \iint t[B]^T[D][B]dxdy \cdot \{\delta\}^e \quad (3.18)$$

$$\text{记} \quad [K]^e = \iint t[B]^T[D][B]dxdy \quad (3.19)$$

把(3.19)式代入(3.18)式后,得:

$$\{F\}^e = [K]^e \{\delta\}^e \quad (3.20)$$

在简单三角形单元中,积分号内的矩阵都是常量,且注意到 $\iint dxdy = A$, (3.19)式,可改写为

$$[K]^e = [B]^T[D][B]At = [B]^T[S]At \quad (3.21)$$

由于 $[B] = [B_i B_j B_m]$, $[S] = [S_i S_j S_m]$, 单元刚度矩阵也可写成分块形式:

$$[K]^e = \begin{bmatrix} k_{ii} & k_{ij} & k_{im} \\ k_{ji} & k_{jj} & k_{jm} \\ k_{mi} & k_{mj} & k_{mm} \end{bmatrix} \quad (3.22)$$

式中,某一个子块

$$[k_{rs}] = \frac{Et}{4(1-\nu^2)\Delta} \begin{bmatrix} b_r b_s + \frac{1-\nu}{2} c_r c_s & \nu b_r c_s + \frac{1-\nu}{2} c_r b_s \\ \nu c_r b_s + \frac{1-\nu}{2} b_r c_s & c_r c_s + \frac{1-\nu}{2} b_r b_s \end{bmatrix}; (r=i, j, m \quad s=i, j, m)$$

对于平面应变问题,只须将公式中的 E 换成 $\frac{E}{1-\nu^2}$, ν 换成 $\frac{\nu}{1-\nu}$ 即可。

3.2.2 平面问题矩形单元

简单三角形单元是常应变单元,较突出的缺点是精度不够理想,为了能较好地反映单元的实际受力情况,对于规则形状的对象,可采用矩形单元。矩形单元可采用4个结点,8个结点,9个结点等多种形式。

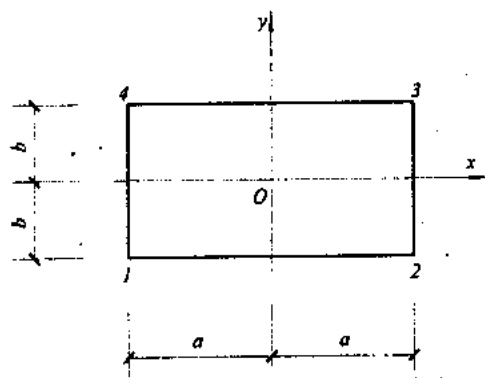


图 3.6 四结点矩形单元

对于图 3.6 所示的四结点矩形单元,为了计算简便,以平行于两邻边的 2 个对称轴为 x 及 y 轴,该矩形沿 x 及 y 轴方向的边长分别用 $2a$ 及 $2b$ 表示。

设矩形单元的结点力和结点位移向量为

$$\{F\}^e = \{X_1 Y_1 X_2 Y_2 X_3 Y_3 X_4 Y_4\}^T$$

$$\{\delta\}^e = \{u_1 v_1 u_2 v_2 u_3 v_3 u_4 v_4\}^T$$

取位移插值函数为

$$\begin{cases} u = a_1 + a_2 x + a_3 y + a_4 xy \\ v = a_5 + a_6 x + a_7 y + a_8 xy \end{cases} \quad (3.23)$$

(3.23)式的位移模式包含了刚体位移和常量应变项。在单元边界上,位移分量是按线性

变化的,从而保证了相邻单元在其公共边界上的位移连续性,上述位移模式满足了收敛条

件。

将四个结点处的 8 个位移值分别代入式(3.23)可得到 8 个方程,联立求解可解出 8 个待定常数 $\alpha_1, \alpha_2, \dots, \alpha_8$,从而确定了位移插值函数式(3.23)。经整理后可表示为

$$\begin{Bmatrix} u \\ v \end{Bmatrix} = \begin{bmatrix} N_1 & 0 & N_2 & 0 & N_3 & 0 & N_4 & 0 \\ 0 & N_1 & 0 & N_2 & 0 & N_3 & 0 & N_4 \end{bmatrix} \begin{Bmatrix} u_1 \\ v_1 \\ u_2 \\ v_2 \\ u_3 \\ v_3 \\ u_4 \\ v_4 \end{Bmatrix} \quad (3.24a)$$

式中

$$N_1 = \frac{1}{4} \left(1 - \frac{x}{a} \right) \left(1 - \frac{y}{b} \right)$$

$$N_2 = \frac{1}{4} \left(1 + \frac{x}{a} \right) \left(1 - \frac{y}{b} \right)$$

$$N_3 = \frac{1}{4} \left(1 + \frac{x}{a} \right) \left(1 + \frac{y}{b} \right)$$

$$N_4 = \frac{1}{4} \left(1 - \frac{x}{a} \right) \left(1 + \frac{y}{b} \right)$$

称为形函数,写成矩阵形式有

$$\{f\} = \begin{Bmatrix} u \\ v \end{Bmatrix} = [N] \{\delta\}^e \quad (3.24b)$$

形函数矩阵 $[N]$ 中各元素的确定,除了上述联立方程求解待定常数 $\alpha_1, \alpha_2, \dots, \alpha_8$ 的方法外,还可以用形函数的基本性质确定。

在确定了位移模式和形函数矩阵后,不难找出单元应变和结点位移的关系式。

$$\{\epsilon\} = \begin{Bmatrix} \epsilon_x \\ \epsilon_y \\ \gamma_{xy} \end{Bmatrix} = \begin{Bmatrix} \frac{\partial u}{\partial x} \\ \frac{\partial v}{\partial y} \\ \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \end{Bmatrix} = [B] \{\delta\}^e \quad (3.25)$$

其中, $[B]$ 称为几何矩阵,其具体表达式为

$$[B] = \frac{1}{4ab} \times \begin{bmatrix} -(b-y) & 0 & b-y & 0 & b+y & 0 & -(b+y) & 0 \\ 0 & -(a-x) & 0 & -(a+x) & 0 & a+x & 0 & a-x \\ -(a-x) & -(b-y) & -(a+x) & b-y & a+x & b+y & a-x & -(b+y) \end{bmatrix} \quad (3.26)$$

由应力、应变关系可以求得单元的应力

$$\{\sigma\} = \begin{Bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{Bmatrix} = [D]\{\varepsilon\} = [D][B]\{\delta\}^e = [S]\{\delta\}^e \quad (3.27)$$

其中

$$[s] = \frac{E}{4ab(1-\nu^2)} \begin{bmatrix} s_{11} & s_{12} & s_{13} & s_{14} & s_{15} & s_{16} & s_{17} & s_{18} \\ s_{21} & s_{22} & s_{23} & s_{24} & s_{25} & s_{26} & s_{27} & s_{28} \\ s_{31} & s_{32} & s_{33} & s_{34} & s_{35} & s_{36} & s_{37} & s_{38} \end{bmatrix} \quad (3.28)$$

$$s_{11} = -(b-y)$$

$$s_{12} = -\nu(a-x)$$

$$s_{13} = -s_{11} = b-y$$

$$s_{14} = -\nu(a+x)$$

$$s_{15} = b+y$$

$$s_{16} = -s_{14} = \nu(a+x)$$

$$s_{17} = -s_{15} = -(b+y)$$

$$s_{18} = -s_{12} = \nu(a-x)$$

$$s_{21} = -\nu(b-y)$$

$$s_{22} = -(a-x)$$

$$s_{23} = -s_{21} = \nu(b-y)$$

$$s_{24} = -(a+x)$$

$$s_{25} = \nu(b+y)$$

$$s_{26} = -s_{24} = a+x$$

$$s_{27} = -s_{25} = -\nu(b+y)$$

$$s_{28} = -s_{22} = a-x$$

$$s_{31} = -\frac{1-\nu}{2}(a-x)$$

$$s_{32} = -\frac{1-\nu}{2}(b-y)$$

$$s_{33} = -\frac{1-\nu}{2}(a+x)$$

$$s_{34} = -s_{32} = \frac{1-\nu}{2}(b-y)$$

$$s_{35} = -s_{33} = \frac{1-\nu}{2}(a+x)$$

$$s_{36} = \frac{1-\nu}{2}(b+y)$$

$$s_{37} = -s_{31} = \frac{1-\nu}{2}(a-x)$$

$$s_{38} = -s_{36} = -\frac{1-\nu}{2}(b+y) \quad (3.29)$$

可见采用矩形单元时,单元中的应力是随坐标变化的,因而它比三角形常应变单元能更好地反映实际应力的变化。

利用虚功原理,可以推导出结点力与结点位移的关系为

$$\{F_e\} = t \int_{-a}^a \int_{-b}^b [B]^T [D] [B] dx dy \{\delta\}^e = [K] \{\delta\}^e \quad (3.30)$$

将 $[B]$ 和 $[D]$ 代入后,对每一元素积分,可以得出单元刚度矩阵的具体表达式为

$$[K] = \frac{Et}{1-\nu^2} \begin{bmatrix} k_1 & & & & & & & & & \\ k_2 & k_3 & & & & & & & & \\ k_4 & k_5 & k_1 & & & & & & & \\ -k_5 & k_6 & -k_2 & k_3 & & & & & & \\ k_7 & -k_2 & k_8 & k_5 & k_1 & & & & & \\ -k_2 & k_9 & -k_5 & k_{10} & k_2 & k_3 & & & & \\ k_8 & -k_5 & k_7 & k_2 & k_4 & k_5 & k_1 & & & \\ k_5 & k_{10} & k_2 & k_9 & -k_5 & k_6 & -k_2 & k_3 & & \end{bmatrix} \quad \text{对 称} \quad (3.31)$$

式中

$$k_1 = \frac{1}{3} \frac{b}{a} + \frac{1-\nu}{6} \frac{a}{b}$$

$$k_2 = \frac{1+\nu}{8}$$

$$k_3 = \frac{1}{3} \frac{a}{b} + \frac{1-\nu}{6} \frac{b}{a}$$

$$k_4 = -\frac{1}{3} \frac{b}{a} + \frac{1-\nu}{12} \frac{a}{b}$$

$$k_5 = \frac{1-3\nu}{8}$$

$$k_6 = -\frac{1}{6} \frac{a}{b} - \frac{1-\nu}{6} \frac{b}{a}$$

$$k_7 = \frac{1}{6} \frac{b}{a} - \frac{1-\nu}{12} \frac{a}{b}$$

$$k_8 = \frac{1}{6} \frac{b}{a} - \frac{1-\nu}{6} \frac{a}{b}$$

$$k_9 = -\frac{1}{6} \frac{a}{b} - \frac{1-\nu}{12} \frac{b}{a}$$

$$k_{10} = -\frac{1}{3} \frac{a}{b} + \frac{1-\nu}{12} \frac{b}{a}$$

3.2.3 四结点平面等参单元

在二维分析中,具有三个结点的简单三角形单元较容易进行网格划分和逼近边界形

状,应用比较灵活,但由于单元内的应力和应变是常数,故精度不够理想。如果采用四结点矩形单元,单元内应力和应变是线性变化的,所以计算精度比简单三角形单元有较大提高,但它不能适应曲线边界和斜边界,其适用性非常有限。当采用任意四边形的等参单元,既能大大提高计算精度又能很好地适应几何形状不规则的区域和边界。对于高阶的平面等参单元(如八结点平面等参单元)还能很好地拟合曲线边界。

本节先讨论四结点任意四边形等参单元。

1. 四结点平面等参单元的位移模式、几何矩阵及雅可比矩阵

在四边形等参单元推导过程中,采用两套坐标系:总体坐标系 xoy ,它用于被分析物体的实际单元;局部坐标系 $\xi\eta$,它是标准化的正方形单元,在等参单元分析中又称为母单元,母单元在边界上有 $\xi = \pm 1$ 和 $\eta = \pm 1$,见图 3.7。

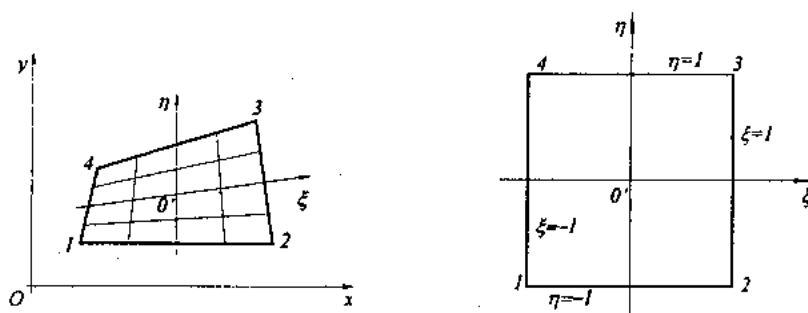


图 3.7 四边形单元

显然,在母单元上建立位移插值函数是方便的,现在的问题是要建立两种单元间的一一对应关系,这可以通过图象变换中映射的方法解决。如取实际单元中任一点的坐标与结点坐标有如下关系,即

$$\left. \begin{aligned} x &= \sum_{i=1}^4 N_i x_i \\ y &= \sum_{i=1}^4 N_i y_i \end{aligned} \right\} \quad (3.32)$$

其中, $x_i, y_i (i=1, 2, 3, 4)$ 为结点坐标,而

$$\left. \begin{aligned} N_1 &= \frac{1}{4}(1-\xi)(1-\eta) \\ N_2 &= \frac{1}{4}(1+\xi)(1-\eta) \\ N_3 &= \frac{1}{4}(1+\xi)(1+\eta) \\ N_4 &= \frac{1}{4}(1-\xi)(1+\eta) \end{aligned} \right\} \quad (3.33a)$$

式(3.33a)可合并写成如下统一形式:

$$N_i = \frac{1}{4}(1 + \xi \xi_i)(1 + \eta \eta_i) \quad (3.33b)$$

式中, ξ_i, η_i 表示第 i 结点的相应局部坐标。

N_i 是 ξ, η 的函数, 也可称为形函数。通过这一变换, 对母单元上每一点 (ξ, η) 都可以在实际单元中找到一一对应点 (x, y) , 即两单元间有一一对应关系。并且, 4 个角点与 4 边界上的点也均有一一对应关系。

于是, 我们在母单元中可建立位移插值函数:

$$\left. \begin{aligned} u &= \sum_{i=1}^4 N_i u_i \\ v &= \sum_{i=1}^4 N_i v_i \end{aligned} \right\} \quad (3.34)$$

其中, $u_i, v_i (i=1, 2, 3, 4)$ 为结点位移; $N_i (i=1, 2, 3, 4)$ 为形函数, 它与坐标变换式 (3.32) 中的形函数相同。因为坐标变换与位移变换中取同一形函数, 所以这种单元便称为等参单元。

有了位移分布函数, 可以求得单元内各点的应变和结点位移的关系式, 由几何方程

$$\{\epsilon\} = \begin{Bmatrix} \epsilon_x \\ \epsilon_y \\ \gamma_{xy} \end{Bmatrix} = \begin{Bmatrix} \frac{\partial u}{\partial x} \\ \frac{\partial v}{\partial y} \\ \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \end{Bmatrix}$$

$$= \begin{bmatrix} \frac{\partial N_1}{\partial x} & 0 & \frac{\partial N_2}{\partial x} & 0 & \frac{\partial N_3}{\partial x} & 0 & \frac{\partial N_4}{\partial x} & 0 \\ 0 & \frac{\partial N_1}{\partial y} & 0 & \frac{\partial N_2}{\partial y} & 0 & \frac{\partial N_3}{\partial y} & 0 & \frac{\partial N_4}{\partial y} \\ \frac{\partial N_1}{\partial y} & \frac{\partial N_1}{\partial x} & \frac{\partial N_2}{\partial y} & \frac{\partial N_2}{\partial x} & \frac{\partial N_3}{\partial y} & \frac{\partial N_3}{\partial x} & \frac{\partial N_4}{\partial y} & \frac{\partial N_4}{\partial x} \end{bmatrix} \begin{Bmatrix} u_1 \\ v_1 \\ u_2 \\ v_2 \\ u_3 \\ v_3 \\ u_4 \\ v_4 \end{Bmatrix} = [B] \{\delta\} \quad (3.35)$$

上式中, $[B]$ 称为几何矩阵, 在几何矩阵中要用到形函数对整体坐标 x, y 的微分。然而, 等参单元中的形函数是用局部坐标 ξ, η 表示的, 直接对整体坐标 x, y 微分有困难, 需找出形函数对局部坐标微分和形函数对整体坐标微分的关系式。利用复合函数求导公式, 得

$$\begin{aligned} \frac{\partial N_i}{\partial \xi} &= \frac{\partial N_i}{\partial x} \frac{\partial x}{\partial \xi} + \frac{\partial N_i}{\partial y} \frac{\partial y}{\partial \xi} \\ \frac{\partial N_i}{\partial \eta} &= \frac{\partial N_i}{\partial x} \frac{\partial x}{\partial \eta} + \frac{\partial N_i}{\partial y} \frac{\partial y}{\partial \eta} \end{aligned}$$

写成矩阵形式, 则为

$$\begin{Bmatrix} \frac{\partial N_i}{\partial \xi} \\ \frac{\partial N_i}{\partial \eta} \end{Bmatrix} = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} \end{bmatrix} \begin{Bmatrix} \frac{\partial N_i}{\partial x} \\ \frac{\partial N_i}{\partial y} \end{Bmatrix} = [J] \begin{Bmatrix} \frac{\partial N_i}{\partial x} \\ \frac{\partial N_i}{\partial y} \end{Bmatrix} \quad (3.36)$$

式中 $[J]$ 即为雅可比矩阵,再将公式(3.32)关系代入,则得:

$$[J] = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^4 \frac{\partial N_i}{\partial \xi} x_i & \sum_{i=1}^4 \frac{\partial N_i}{\partial \xi} y_i \\ \sum_{i=1}^4 \frac{\partial N_i}{\partial \eta} x_i & \sum_{i=1}^4 \frac{\partial N_i}{\partial \eta} y_i \end{bmatrix} \quad (3.37)$$

由于形函数 $N_i(\xi, \eta)$ 是局部坐标 ξ, η 的函数,不难对 ξ, η 的求导,四结点等参单元的形函数对 ξ, η 的偏导数由下表3.1列出。

表 3.1

i	ξ_i	η_i	N_i	$\frac{\partial N_i}{\partial \xi}$	$\frac{\partial N_i}{\partial \eta}$
1	-1	-1	$\frac{1}{4}(1-\xi)(1-\eta)$	$-\frac{1}{4}(1-\eta)$	$-\frac{1}{4}(1-\xi)$
2	1	-1	$\frac{1}{4}(1+\xi)(1-\eta)$	$\frac{1}{4}(1-\eta)$	$-\frac{1}{4}(1+\xi)$
3	1	1	$\frac{1}{4}(1+\xi)(1+\eta)$	$\frac{1}{4}(1+\eta)$	$\frac{1}{4}(1+\xi)$
4	-1	1	$\frac{1}{4}(1-\xi)(1+\eta)$	$-\frac{1}{4}(1+\eta)$	$\frac{1}{4}(1-\xi)$

于是,可确定雅可比矩阵:

$$[J] = \frac{1}{4} \begin{bmatrix} -(1-\eta) & (1-\eta) & (1+\eta) & -(1+\eta) \\ -(1-\xi) & -(1+\xi) & (1+\xi) & (1-\xi) \end{bmatrix} \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \\ x_3 & y_3 \\ x_4 & y_4 \end{bmatrix} \quad (3.38)$$

对式(3.36)求逆后可得

$$\begin{Bmatrix} \frac{\partial N_i}{\partial x} \\ \frac{\partial N_i}{\partial y} \end{Bmatrix} = [J]^{-1} \begin{Bmatrix} \frac{\partial N_i}{\partial \xi} \\ \frac{\partial N_i}{\partial \eta} \end{Bmatrix} \quad (3.39)$$

由于雅可比矩阵的逆阵 $[J]^{-1}$ 不难求得,进而由式(3.39)确定了形函数对整体坐标 x, y 的微分。代入式(3.35)不难确定几何矩阵 $[B]$ 中的每一项元素。

2. 单元刚度矩阵与高斯积分

几何矩阵 $[B]$ 确定后,可由虚功原理得到的平面问题单元刚度矩阵一般表达式,积分求得单元刚度矩阵:

$$[K] = t \iint [B]^T [D] [B] dx dy$$

由于被积函数 $[B], [B]^T$ 是由局部坐标 ξ, η 表示,积分时的微分面积 $dx dy$ 要转换成局部坐标表示,并相应地确定积分上、下限。

根据矢量代数知识,由图3.8可知,积分微元面积:

$$dA = |a \times b| = |a||b|\sin\theta \quad (3.40)$$

$$\begin{aligned} \text{设: } a &= a_x i + a_y j \\ b &= b_x i + b_y j \end{aligned}$$

其中, a_x, a_y 为边长 a 在 x 轴、 y 轴上的投影; b_x, b_y 为边长 b 在 x 轴与 y 轴上的投影; i, j 分别为沿 x, y 方向的单位向量。则

$$dA = |(a_x i + a_y j) \times (b_x i + b_y j)| = \begin{vmatrix} a_x & b_x \\ a_y & b_y \end{vmatrix} \quad (3.41)$$

由图 3.8 可知:

点 1 坐标为 $x(\xi, \eta) = x_1, y(\xi, \eta) = y_1$;

点 2 坐标为 $x(\xi + d\xi, \eta) = x_1 + \frac{\partial x}{\partial \xi} d\xi, y(\xi + d\xi, \eta) = y_1 + \frac{\partial y}{\partial \xi} d\xi$;

点 3 坐标为 $x(\xi, \eta + d\eta) = x_1 + \frac{\partial x}{\partial \eta} d\eta, y(\xi, \eta + d\eta) = y_1 + \frac{\partial y}{\partial \eta} d\eta$ 。

于是有

$$\begin{aligned} a_x &= x_2 - x_1 = \frac{\partial x}{\partial \xi} d\xi & a_y &= y_2 - y_1 = \frac{\partial y}{\partial \xi} d\xi \\ b_x &= x_3 - x_1 = \frac{\partial x}{\partial \eta} d\eta & b_y &= y_3 - y_1 = \frac{\partial y}{\partial \eta} d\eta \\ dA &= \begin{vmatrix} a_x & b_x \\ a_y & b_y \end{vmatrix} = \begin{vmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial x}{\partial \eta} \\ \frac{\partial y}{\partial \xi} & \frac{\partial y}{\partial \eta} \end{vmatrix} d\xi d\eta = |J| d\xi d\eta \end{aligned} \quad (3.42)$$

式中, $|J|$ 为雅可比矩阵的行列式

$$|J| = \begin{vmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial x}{\partial \eta} \\ \frac{\partial y}{\partial \xi} & \frac{\partial y}{\partial \eta} \end{vmatrix}$$

显然, 变量 ξ, η 的积分上下限均为 -1 到 $+1$, 于是四结点等参单元的单元刚度矩阵公式变换为:

$$[K] = \int_{-1}^1 \int_{-1}^1 [B]^T [D] [B] |J| d\xi d\eta \quad (3.43)$$

从理论上讲, 求解上述积分已没有什么困难, 但由于被积函数为 ξ, η 的高次项, 直接求积相当繁, 一般不是采用求原函数代入上下限的方法求积, 而是常采用数值积分的方法。有限元分析中最常用的是高斯数值积分法。

在积分区间内, 选取若干个有代表性的样点(称为高斯积分样点), 算出被积函数 f 在这些积分点处的函数值, 然后用相应的加权系数乘上这些函数值, 将乘积累加后作为近似积分值, 这种方法称为高斯求积法。

对于积分区在 -1 至 $+1$ 之间的一维高斯求积公式为

$$I = \int_{-1}^1 f(s) ds = \sum_{i=1}^n H_i f(s_i) \quad (3.44)$$

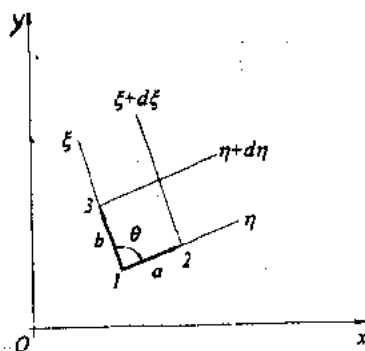


图 3.8

式中: $f(s_i)$ ——被积函数 f 在积分点 S_i 处的函数值;

H_i ——对应 $f(s_i)$ 的加权系数;

n ——所取高斯积分点的个数。

高斯积分样点坐标 s_i , 以及对应加权系数 H_i 的确定, 可用数学方法推导得出, 现将结果列于表 3.2 中。

表 3.2 高斯积分公式中的积分点坐标和加权系数

高斯积分点数	高斯点坐标 s_i	加权系数 H_i
2	$\pm \sqrt{1/3} = \pm 0.5773503$	1.0
3	± 0.7745967 0.0	0.555556 0.888889
4	± 0.8611363 ± 0.3399810	0.3478548 0.6521452
5	± 0.9061798 ± 0.5384693 0.0	0.2369269 0.4786287 0.5688889

如图 3.9 所示, 被积函数 $f(x)$ 在区间 $(-1, 1)$ 连续, 当选用 2 个高斯点积分时, 由表 3.2 中查得这 2 个高斯点的坐标分别为

$$x_1 = -\sqrt{\frac{1}{3}} \quad H_1 = 1.0$$

$$x_2 = \sqrt{\frac{1}{3}} \quad H_2 = 1.0$$

代入式(3.44)可用高斯积分法得到该被积函数的近似积分值:

$$I = \int_{-1}^1 f(x) dx = \sum_{i=1}^2 H_i f(x_i) \\ = H_1 f(x_1) + H_2 f(x_2) = 1.0 \times f\left(-\sqrt{\frac{1}{3}}\right) + 1.0 f\left(\sqrt{\frac{1}{3}}\right)$$

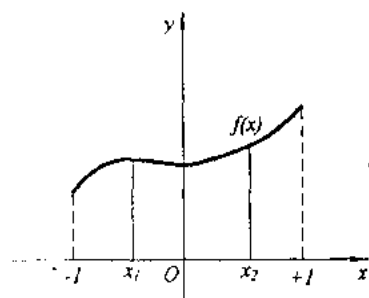


图 3.9

可以证明, 选用 n 个高斯点作数值积分时, 当被积函数为多项式时, 其最高次幂 $\leq 2n - 1$, 数值积分的结果是完全精确的。但在有限元应用中并不是增加积分点的个数, 就能大大提高计算精度, 考虑了各种因素后, 通常在有限元计算时二维问题选用 2×2 个高斯点, 三维问题选用 $3 \times 3 \times 3$ 个高斯点。

以一维高斯求积公式为基础, 不难推广到二维和三维的求积公式。

求二重积分 $\int_{-1}^1 \int_{-1}^1 f(s, t) ds dt$ 的值时, 可以先对变量 s 积分, 这时把 t 看作常量, 由式(3.44)有

$$\int_{-1}^1 f(s, t) ds dt = \sum_{i=1}^n H_i f(s_i, t) = \phi(t)$$

再对 t 进行积分, 得

$$\int_{-1}^1 \int_{-1}^1 f(s, t) ds dt = \int_{-1}^1 \phi(t) dt = \sum_{j=1}^n H_j \phi(t_j) = \sum_{j=1}^n H_j \left[\sum_{i=1}^n H_i f(s_i, t_j) \right]$$

整理后得二维高斯求积公式:

$$\int_{-1}^1 \int_{-1}^1 f(s, t) ds dt = \sum_{i=1}^n \sum_{j=1}^n H_i H_j f(s_i, t_j) \quad (3.45)$$

3.2.4 八结点曲线四边形等参单元

在求解二维问题时, 更多的是采用 8 结点曲线四边形等参单元来代替 4 结点四边形等参单元。8 结点等参单元比 4 结点等参单元精度更高, 更能适应各种不规则形状的边界情况。

8 结点等参单元的单元分析可由 4 结点等参单元直接推广而得。

如图 3.10(a) 所示的 8 结点曲线四边形单元, 其母单元为如图 3.10(b) 所示的边长为 2 的正方形单元, 只要引入局部坐标 ξ, η 就可把母单元的位移模式用到曲线四边形的实际单元中去。

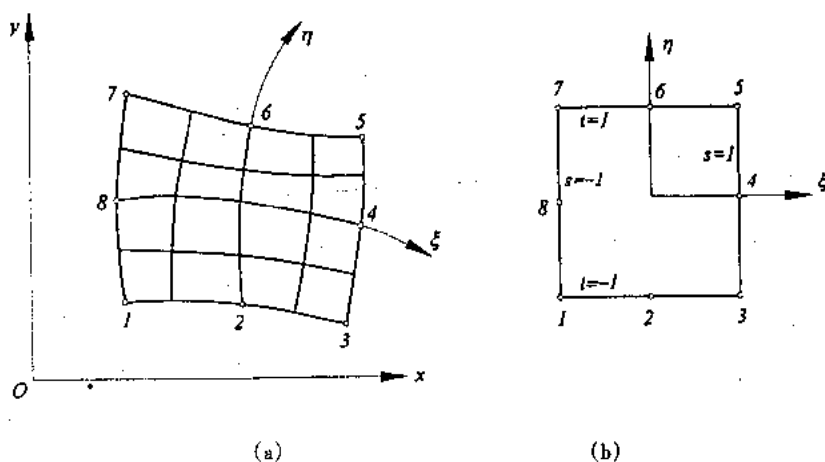


图 3.10 8 结点曲线四边形单元

位移模式仍可取为

$$\left. \begin{aligned} u &= \sum_{i=1}^8 N_i u_i \\ v &= \sum_{i=1}^8 N_i v_i \end{aligned} \right\} \quad (3.46)$$

其中形函数 N_i 可通过形函数基本性质直接求得。

四个角结点上形函数:

$$N_i = \frac{1}{4}(1 + \xi\xi_i)(1 + \eta\eta_i)(\xi\xi_i + \eta\eta_i - 1) \quad (i = 1, 3, 5, 7)$$

边中结点形函数:

$$N_i = \frac{1}{2}(1 - \xi^2)(1 + \eta\eta_i) \quad (i = 2, 6)$$

$$N_i = \frac{1}{2}(1 + \eta\eta_i)(1 - \eta^2) \quad (i = 4, 8)$$

坐标变换式也采用与位移模式一样的形函数:

$$\left. \begin{aligned} x &= \sum_{i=1}^8 N_i x_i \\ y &= \sum_{i=1}^8 N_i y_i \end{aligned} \right\} \quad (3.47)$$

式中: N_i 是局部坐标 ξ, η 的函数。

几何矩阵:

$$[B] = [B_1 \quad B_2 \quad B_3 \quad B_4 \quad B_5 \quad B_6 \quad B_7 \quad B_8]$$

其中:

$$[B_i] = \begin{bmatrix} \frac{\partial N_i}{\partial x} & 0 \\ 0 & \frac{\partial N_i}{\partial y} \\ \frac{\partial N_i}{\partial y} & \frac{\partial N_i}{\partial x} \end{bmatrix} \quad (i = 1, 2, \dots, 8)$$

式中 $\frac{\partial N_i}{\partial x}, \frac{\partial N_i}{\partial y}$ 为形函数对整体坐标的微分, 可直接利用式(3.36), (3.37)和(3.39)

确定, 但在应用式(3.37)求雅可比矩阵时, 求和要有 8 项, 即

$$[J] = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^8 \frac{\partial N_i}{\partial \xi} x_i & \sum_{i=1}^8 \frac{\partial N_i}{\partial \xi} y_i \\ \sum_{i=1}^8 \frac{\partial N_i}{\partial \eta} x_i & \sum_{i=1}^8 \frac{\partial N_i}{\partial \eta} y_i \end{bmatrix}$$

8 结点等参单元的形函数及导数表示式列于表 3.3 中

表 3.3

结点	局部坐标		形函数	导数	
i	ξ_i	η_i	N_i	$\frac{\partial N_i}{\partial \xi}$	$\frac{\partial N_i}{\partial \eta}$
1	-1	-1	$\frac{1}{4}(1 - \xi)(1 - \eta)$	$-\frac{1}{4}(1 - \eta)$	$-\frac{1}{4}(1 - \xi)$
			$-\frac{1}{4}(1 - \xi^2)(1 - \eta)$	$\frac{1}{2}\xi(1 - \eta)$	$\frac{1}{4}(1 - \xi^2)$
			$-\frac{1}{4}(1 - \xi)(1 - \eta^2)$	$\frac{1}{4}(1 - \eta^2)$	$\frac{1}{2}\eta(1 - \xi)$
2	0	-1	$-\frac{1}{2}(1 - \xi^2)(1 - \eta)$	$-\xi(1 - \eta)$	$-\frac{1}{2}(1 - \xi^2)$

续表

结点	局部坐标		形函数	导数	
3	1	-1	$\frac{1}{4}(1+\xi)(1-\eta)$	$\frac{1}{4}(1-\eta)$	$-\frac{1}{4}(1+\xi)$
			$-\frac{1}{4}(1-\xi^2)(1-\eta)$	$\frac{1}{2}\xi(1-\eta)$	$\frac{1}{4}(1-\xi^2)$
			$-\frac{1}{4}(1+\xi)(1-\eta^2)$	$-\frac{1}{4}(1-\eta^2)$	$\frac{1}{2}\eta(1+\xi)$
4	1	0	$\frac{1}{2}(1+\xi)(1-\eta^2)$	$\frac{1}{2}(1-\eta^2)$	$-\eta(1+\xi)$
5	1	1	$\frac{1}{4}(1+\xi)(1+\eta)$	$\frac{1}{4}(1+\eta)$	$\frac{1}{4}(1+\xi)$
			$-\frac{1}{4}(1-\xi^2)(1+\eta)$	$\frac{1}{2}\xi(1+\eta)$	$-\frac{1}{4}(1-\xi^2)$
			$-\frac{1}{4}(1+\xi)(1-\eta^2)$	$-\frac{1}{4}(1-\eta^2)$	$\frac{1}{2}\eta(1+\xi)$
6	0	1	$\frac{1}{2}(1-\xi^2)(1+\eta)$	$-\xi(1+\eta)$	$\frac{1}{2}(1-\xi^2)$
7	-1	1	$\frac{1}{4}(1-\xi)(1+\eta)$	$-\frac{1}{4}(1+\eta)$	$\frac{1}{4}(1-\xi)$
			$-\frac{1}{4}(1-\xi^2)(1+\eta)$	$\frac{1}{2}\xi(1+\eta)$	$-\frac{1}{4}(1-\xi^2)$
			$-\frac{1}{4}(1-\xi)(1-\eta^2)$	$\frac{1}{4}(1-\eta^2)$	$\frac{1}{2}\eta(1-\xi)$
8	-1	0	$\frac{1}{2}(1-\xi)(1-\eta^2)$	$-\frac{1}{2}(1-\eta^2)$	$-\eta(1-\xi)$

在确定几何矩阵后,不难得到单元刚度矩阵

$$[K]^e = \int_{-1}^1 \int_{-1}^1 [B]^T [D] [B] |J| d\xi d\eta \quad (3.48)$$

当有外荷载作用时,需将该单元所受的每一个荷载都向结点移置(分解)而成为结点荷载。

3.3 杆系单元

为了描述钢筋的受力状态,可采用一维杆系单元,常用的一维单元为一维线性桁架单元和一维等参单元。

3.3.1 一维桁架单元

所谓一维桁架单元,是指两力杆,只能承受轴力,而不能受弯和受剪的杆件单元。

设桁架单元两端结点分别为 i, j , 杆截面积为 A , 杆长为 L , 与水平轴夹角为 θ , 并取 $C = \cos\theta, S = \sin\theta$, 见图 3.11。

杆端结点位移列阵 $\{\delta\}^e$ 和杆端结点力列阵 $\{F\}^e$ 分别为

$$\{\delta\}^e = [u_i \quad v_i \quad u_j \quad v_j]^T$$

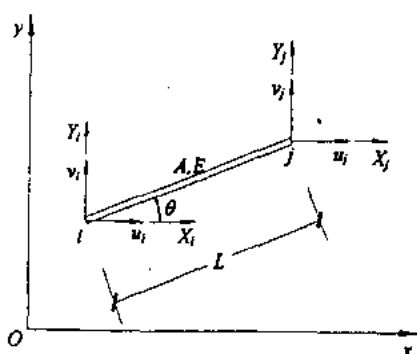


图 3.11 桁架单元

$$\{F\}^e = [X_i \quad Y_i \quad X_j \quad Y_j]^T$$

桁架单元的应变与结点位移之间的关系可用下式表示之：

$$\epsilon = \frac{1}{L} [-C \quad -S \quad C \quad S] \begin{Bmatrix} u_i \\ v_i \\ u_j \\ v_j \end{Bmatrix}$$

或简写为：

$$\epsilon = [B] \{\delta\}^e \quad (3.49)$$

由应力应变关系可得：

$$\sigma = E \epsilon = \frac{E}{L} [-C \quad -S \quad C \quad S] \begin{Bmatrix} u_i \\ v_i \\ u_j \\ v_j \end{Bmatrix}$$

或简写为

$$\sigma = [S] \{\delta\}^e \quad (3.50)$$

式中： σ ——桁架单元的应力；

$[S]$ ——应力矩阵；

利用虚功原理，可得结点和结点位移的关系式如下：

$$\{F\}^e = \begin{Bmatrix} X_i \\ Y_i \\ X_j \\ Y_j \end{Bmatrix} = \frac{AE}{L} \begin{bmatrix} C^2 & CS & -C^2 & -CS \\ CS & S^2 & -CS & -S^2 \\ -C^2 & -CS & C^2 & CS \\ -CS & -S^2 & CS & S^2 \end{bmatrix} \begin{Bmatrix} u_i \\ v_i \\ u_j \\ v_j \end{Bmatrix}$$

或简写为

$$\{F\}^e = [k] \{\delta\}^e \quad (3.51)$$

此处，单元刚度矩阵 $[k]$ 为：

$$[k] = \frac{AE}{L} \begin{bmatrix} C^2 & CS & -C^2 & -CS \\ CS & S^2 & -CS & -S^2 \\ -C^2 & -CS & C^2 & CS \\ -CS & -S^2 & CS & S^2 \end{bmatrix} \quad (3.52)$$

单元刚度矩阵，也可分块写成

$$[k] = \begin{bmatrix} k_{ii} & k_{ij} \\ k_{ji} & k_{jj} \end{bmatrix}$$

其中，子矩阵为

$$\left. \begin{aligned} [k_{rs}] &= \frac{AE}{L} \begin{bmatrix} C^2 & CS \\ CS & S^2 \end{bmatrix}; & (r=i, j; s=i, j; r=s) \\ [k_{rs}] &= \frac{AE}{L} \begin{bmatrix} -C^2 & -CS \\ -CS & -S^2 \end{bmatrix}; & (r=i, j; s=i, j; r \neq s) \end{aligned} \right\}$$

3.3.2 等参杆单元

在分析钢筋混凝土结构时,常把平面单元和杆件单元组合使用,特别是平面单元采用 8 结点等参元时,为变形协调杆单元可选用等参杆单元,下面推导等参杆单元的基本公式。

1. 等参杆单元的位移模式

如图 3.12 所示的平面曲杆,有三个结点,其参数坐标为 s ,以杆中结点 3 为坐标原点,结点 3 不一定位于杆的几何中心。

杆中任一点的位移为

$$\left. \begin{aligned} u &= N_1 u_1 + N_2 u_2 + N_3 u_3 \\ v &= N_1 v_1 + N_2 v_2 + N_3 v_3 \end{aligned} \right\} \quad (3.53)$$

坐标变换式选用相同的形函数:

$$\left. \begin{aligned} x &= N_1 x_1 + N_2 x_2 + N_3 x_3 \\ y &= N_1 y_1 + N_2 y_2 + N_3 y_3 \end{aligned} \right\} \quad (3.54)$$

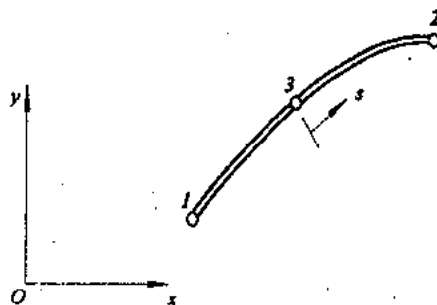


图 3.12 平面曲杆

式中

$$N_1 = \frac{1}{2} s(s-1)$$

$$N_2 = \frac{1}{2} s(s+1)$$

$$N_3 = (1-s^2)$$

2. 等参杆单元的应变和应力

设单元的应变为 ϵ ,当单元的微分元 δl 变化到 $(\delta l + \epsilon \delta l)$ 时,新的增量可表示为

$$(\delta l + \epsilon \delta l)^2 = \left(\delta x + \frac{\partial u}{\partial s} \delta s \right)^2 + \left(\delta y + \frac{\partial v}{\partial s} \delta s \right)^2 \quad (3.55)$$

式中 s 为局部坐标。

(3.55)式两边除以 $(\delta l)^2$ 得

$$(1 + \epsilon)^2 = \left(\frac{\delta x}{\delta l} + \frac{\partial u}{\partial s} \frac{\delta s}{\delta l} \right)^2 + \left(\frac{\delta y}{\delta l} + \frac{\partial v}{\partial s} \frac{\delta s}{\delta l} \right)^2$$

注意到 $\left(\frac{\delta x}{\delta l} \right)^2 + \left(\frac{\delta y}{\delta l} \right)^2 = 1$ 并略去高阶微分小量,即

$$\epsilon^2 = 0; \quad \left(\frac{\partial u}{\partial s} \frac{\delta s}{\delta l} \right)^2 = 0; \quad \left(\frac{\partial v}{\partial s} \frac{\delta s}{\delta l} \right)^2 = 0,$$

得

$$\epsilon = \left(\frac{\delta x}{\delta l} \right) \left(\frac{\partial u}{\partial s} \frac{\delta s}{\delta l} \right) + \left(\frac{\delta y}{\delta l} \right) \left(\frac{\partial v}{\partial s} \frac{\delta s}{\delta l} \right) \quad (3.56)$$

式中 $\delta x = \frac{\partial x}{\partial s} \delta s$; $\delta y = \frac{\partial y}{\partial s} \delta s$; $\frac{\delta x}{\delta l} = \frac{\partial x}{\partial s} \frac{\delta s}{\delta l}$; $\frac{\delta y}{\delta l} = \frac{\partial y}{\partial s} \frac{\delta s}{\delta l}$

$$\delta l = \sqrt{(\delta x)^2 + (\delta y)^2} = \sqrt{\left(\frac{\partial x}{\partial s}\right)^2 + \left(\frac{\partial y}{\partial s}\right)^2} \delta s$$

令 $\Psi(s) = \left(\frac{\partial x}{\partial s}\right)^2 + \left(\frac{\partial y}{\partial s}\right)^2$ (3.57)

则 $\frac{\delta s}{\delta l} = \frac{1}{\sqrt{\Psi(s)}}$

代入式(3.56)得

$$\epsilon = \frac{\left(\frac{\partial x}{\partial s}\right) \left(\frac{\partial u}{\partial s}\right) + \left(\frac{\partial y}{\partial s}\right) \left(\frac{\partial v}{\partial s}\right)}{\Psi(s)}$$

式中: $\frac{\partial x}{\partial s} = \frac{\partial \sum_{i=1}^3 N_i x_i}{\partial s} = N'_1 x_1 + N'_2 x_2 + N'_3 x_3$

$$\frac{\partial u}{\partial s} = \frac{\partial \sum_{i=1}^3 N_i u_i}{\partial s} = N'_1 u_1 + N'_2 u_2 + N'_3 u_3$$

$$N'_1 = \frac{\partial N_1}{\partial s} = \left(s - \frac{1}{2}\right); \quad N'_2 = \frac{\partial N_2}{\partial s} = \left(s + \frac{1}{2}\right); \quad N'_3 = \frac{\partial N_3}{\partial s} = -2s$$

$$\begin{aligned} \left(\frac{\partial x}{\partial s}\right) \left(\frac{\partial u}{\partial s}\right) &= (N'_1 x_1 + N'_2 x_2 + N'_3 x_3)(N'_1 u_1 + N'_2 u_2 + N'_3 u_3) \\ &= (N'_1 N'_1 x'_1 + N'_1 N'_2 x'_2 + N'_1 N'_3 x'_3) u_1 + (N'_1 N'_2 x'_1 + N'_2 N'_2 x'_2 + N'_2 N'_3 x'_3) u_2 \\ &\quad + (N'_1 N'_3 x'_1 + N'_2 N'_3 x'_2 + N'_3 N'_3 x'_3) u_3 \end{aligned}$$

同理也可以得到 $\left(\frac{\partial y}{\partial s}\right) \left(\frac{\partial v}{\partial s}\right)$ 的表达式。另外,式(3.57)可写为

$$\begin{aligned} \Psi(s) &= (N'_1)^2 (x_1^2 + y_1^2) + (N'_2)^2 (x_2^2 + y_2^2) + (N'_3)^2 (x_3^2 + y_3^2) \\ &\quad + 2(N'_1 N'_2)(x_1 x_2 + y_1 y_2) + 2(N'_1 N'_3)(x_1 x_3 + y_1 y_3) \\ &\quad + 2N'_2 N'_3 (x_2 x_3 + y_2 y_3) \end{aligned} \quad (3.58)$$

于是可得到应变 ϵ 的表达式,写成矩阵表达式:

$$\{\epsilon\} = [B]_{1 \times 6} \{\delta\}_{6 \times 1}^e \quad (3.59)$$

式中 $\{\delta\}^e = [u_1 \ v_1 \ u_2 \ v_2 \ u_3 \ v_3]^T$

$$[B] = \frac{1}{\Psi(s)} \begin{bmatrix} \left(s - \frac{1}{2}\right)^2 x_1 + \left(s - \frac{1}{2}\right) \left(s + \frac{1}{2}\right) x_2 - 2s \left(s - \frac{1}{2}\right) x_3 \\ \left(s - \frac{1}{2}\right)^2 y_1 + \left(s - \frac{1}{2}\right) \left(s + \frac{1}{2}\right) y_2 - 2s \left(s - \frac{1}{2}\right) y_3 \\ \left(s - \frac{1}{2}\right) \left(s + \frac{1}{2}\right) x_1 + \left(s + \frac{1}{2}\right)^2 x_2 - 2s \left(s + \frac{1}{2}\right) x_3 \\ \left(s - \frac{1}{2}\right) \left(s + \frac{1}{2}\right) y_1 + \left(s + \frac{1}{2}\right)^2 y_2 - 2s \left(s + \frac{1}{2}\right) y_3 \\ - 2s \left(s - \frac{1}{2}\right) x_1 - 2s \left(s + \frac{1}{2}\right) x_2 + 4s^2 x_3 \\ - 2s \left(s - \frac{1}{2}\right) y_1 - 2s \left(s + \frac{1}{2}\right) y_2 + 4s^2 y_3 \end{bmatrix}^T \quad (3.60)$$

杆件横截面一般取为常量,如果杆件为变截面时,也可用下式表示截面积:

$$A(s) = N_1 A_1 + N_2 A_2 + N_3 A_3$$

应力与结点位移关系:

$$\{\sigma\} = [D]_{1 \times 1} [B]_{1 \times 6} \{\delta\}_{6 \times 1}^e \quad (3.61)$$

对一维杆而言,弹性矩阵即为 $[D] = E$ 。

按虚功原理,不难得到杆单元的刚度矩阵。

$$[K]^e = \int_l EA [B]^T [B] dl = \int_{-1}^1 EA [B]^T [B] \Psi(s)^{\frac{1}{2}} ds \quad (3.62)$$

由于单元刚度矩阵的积分比较复杂,可用前述的高斯求积法进行数值积分。

3.4 联结单元

对于钢筋混凝土结构,由于它是由钢筋和混凝土两种不同材料所组成,如果混凝土和钢筋两者之间的相互粘结很好,不会发生相对滑移,则分别采用代表混凝土和钢筋的单元就足够了,这些混凝土单元相互之间及它们与钢筋单元之间都只是在结点处相互铰结。

如果要考虑混凝土和钢筋之间的相对滑移,考虑到粘结力可能发生破坏,那么还需要引入另一种能反映钢筋与混凝土两者间界面性能的单元,即联结单元。联结单元的特点是,它能沿着与联结面垂直方向传递压应力,也能沿着与联结面平行方向传递剪应力,但不能传递拉应力。

钢筋与混凝土的界面性能是极其复杂的,它不仅与混凝土的材性有关,而且受到很多因素的影响,如钢筋的外形特征,直径粗细,混凝土保护层厚度、横向约束钢筋以及受力状况等等。在钢筋混凝土有限元分析中,近十几年来已提出了多种不同形式的联(粘)结单元模式,这方面的研究情况反映在 1981 年、1987 年 Delft 的两次混凝土力学会议,1982 年、1992 年的两次“Bond in Concrete”国际会议,以及 1985 年、1992 年的两次日、美 RC 有限元分析讨论会的论文集中。

在钢筋混凝土有限元分析中,粘结模型种类较多,有双垂直弹簧粘结模型,粘结区单元,斜压杆单元,四结点线性节理单元,六结点曲边节理单元等。本节主要讨论双弹簧联结单元,四结点线性节理单元和六结点曲边节理单元,对有些粘结单元作简单介绍。

3.4.1 双弹簧联结单元

如图 3.13 所示,(也称为粘结链)在垂直于钢筋和平行于钢筋表面方向设置互相垂直的一组弹簧。这组弹簧是设想的力学模型,具有弹性刚度,但并无实际几何尺寸,所以它可以放置在需要设置联系的任何地方。在沿钢筋长度方向上肯定不会产生粘结破坏的结点,例如简支梁端部(当粘结锚固得较好时)以及对称情况下的跨中,可以不设置联结单元,其余部位可根据具体情况按需要设置。

平行于两种单元接触面的弹簧用以计算相对滑移和粘结应力,垂直于两种单元接触面的弹簧用以考虑钢筋的销栓作用。两种弹簧刚度分别为 k_h 和 k_v 。

1. 应变矩阵

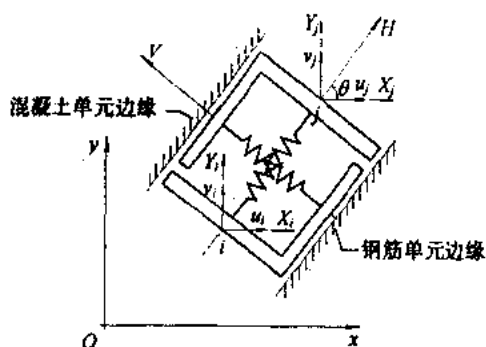


图 3.13 双弹簧联结单元

联结单元的应变(此外应变的涵义不是一般所常用的无量纲的单位应变;而是 i 和 j 点分别在 H 和 V 方向位移的差额,其单位与位移相同)与结点位移之间的关系可用下式表示之。

$$\{\epsilon\} = \begin{Bmatrix} \epsilon_h \\ \epsilon_v \end{Bmatrix} = \begin{bmatrix} -C & -S & C & S \\ S & -C & -S & C \end{bmatrix} \begin{Bmatrix} u_i \\ v_i \\ u_j \\ v_j \end{Bmatrix} \quad (3.63a)$$

或简写为

$$\{\epsilon\} = [B]\{\delta\}^e \quad (3.63b)$$

其中, $[B]$ 为应变矩阵:

$$[B] = \begin{bmatrix} -C & -S & C & S \\ S & -C & -S & C \end{bmatrix} \quad (3.64)$$

应变矩阵 $[B]$ 也可以分块写成:

$$[B] = [B_i \quad B_j]$$

其中,子矩阵为

$$[B_i] = \begin{bmatrix} -C & -S \\ S & -C \end{bmatrix}$$

$$[B_j] = \begin{bmatrix} C & S \\ -S & C \end{bmatrix}$$

2. 应力矩阵

由应力应变关系可得

$$\{\sigma\} = \begin{Bmatrix} \sigma_h \\ \sigma_v \end{Bmatrix} = [D]\{\epsilon\} = \begin{bmatrix} k_h & 0 \\ 0 & k_v \end{bmatrix} \begin{Bmatrix} \epsilon_h \\ \epsilon_v \end{Bmatrix} = \begin{bmatrix} k_h & 0 \\ 0 & k_v \end{bmatrix} \begin{bmatrix} -C & -S & C & S \\ S & -C & -S & C \end{bmatrix} \begin{Bmatrix} u_i \\ v_i \\ u_j \\ v_j \end{Bmatrix} \quad (3.65)$$

上式中,应力 $\{\sigma\}$ 代表 H 和 V 方向的弹簧总应力(内力) σ_h 和 σ_v , 可以简写为

$$\{\sigma\} = [S]\{\delta\}^e$$

其中, $[S]$ 为应力矩阵

$$[S] = \begin{bmatrix} k_h & 0 \\ 0 & k_v \end{bmatrix} \begin{bmatrix} -C & -S & C & S \\ S & -C & -S & C \end{bmatrix} = \begin{bmatrix} -k_h C & -k_h S & k_h C & k_h S \\ k_v S & -k_v C & -k_v S & k_v C \end{bmatrix} \quad (3.66)$$

应力矩阵可以分块写成:

$$[S] = [S_i \quad S_j]$$

其中,子矩阵为

$$[S_i] = \begin{bmatrix} -k_h C & -k_h S \\ k_v S & -k_v C \end{bmatrix}$$

$$[S_j] = \begin{bmatrix} k_h C & k_h S \\ -k_v S & k_v C \end{bmatrix}$$

3. 单元刚度矩阵

$$\{F\}^e = [K] \{\delta\}^e \quad (3.67)$$

其中, $\{F\}^e$ 为代表结点力的列阵:

$$\{F\}^e = \begin{Bmatrix} X_i \\ Y_i \\ X_j \\ Y_j \end{Bmatrix}$$

而 $\{\delta\}^e$ 为代表结点位移的列阵:

$$\{\delta\}^e = \begin{Bmatrix} u_i \\ v_i \\ u_j \\ v_j \end{Bmatrix}$$

$[k]$ 为单元刚度矩阵:

$$\begin{aligned} [k] &= [B]^T [D] [B] = \begin{bmatrix} -C & S \\ -S & -C \\ C & -S \\ S & C \end{bmatrix} \begin{bmatrix} k_h & 0 \\ 0 & k_v \end{bmatrix} \begin{bmatrix} -C & -S & C & S \\ S & -C & -S & C \end{bmatrix} \\ &= \begin{bmatrix} k_h C^2 + k_v S^2 & k_h SC - k_v SC & -k_h C^2 - k_v S^2 & -k_h SC + k_v SC \\ k_h SC - k_v SC & k_h S^2 + k_v C^2 & -k_h SC + k_v SC & -k_h S^2 - k_v C^2 \\ -k_h C^2 - k_v S^2 & -k_h SC + k_v SC & k_h C^2 + k_v S^2 & k_h SC - k_v SC \\ -k_h SC + k_v SC & -k_h S^2 - k_v C^2 & k_h SC - k_v SC & k_h S^2 + k_v C^2 \end{bmatrix} \end{aligned} \quad (3.68a)$$

单元刚度矩阵 $[k]$ 可以分块写成:

$$[k] = \begin{bmatrix} k_{ii} & k_{ij} \\ k_{ji} & k_{jj} \end{bmatrix} \quad (3.68b)$$

其中, 子矩阵为

$$[k_{rs}] = \begin{bmatrix} k_h C^2 + k_v S^2 & k_h SC - k_v SC \\ k_h SC - k_v SC & k_h S^2 + k_v C^2 \end{bmatrix}; \quad (r = i, j; s = i, j; r = s)$$

$$[k_{rs}] = \begin{bmatrix} -k_h C^2 - k_v S^2 & -k_h SC + k_v SC \\ -k_h SC + k_v SC & -k_h S^2 - k_v C^2 \end{bmatrix}; \quad (r=i, j; s=i, j; r \neq s)$$

由式(3.67)及(3.68b)可得到作用在结点 i 的结点力为

$$\{F_i\} = \begin{Bmatrix} X_i \\ Y_i \end{Bmatrix} = [K_{ii} \quad K_{ij}] \begin{Bmatrix} \delta_i \\ \delta_j \end{Bmatrix}$$

其中, $\{F_i\}$ 为联结单元由于结点位移施加于结点 i 的力; $[K_{ii} \quad K_{ij}]$ 同式(3.68b); $\{\delta_i\} = \begin{Bmatrix} u_i \\ v_i \end{Bmatrix}; (i, j)$, 为结点位移。

弹簧刚度 k_h 和 k_v 受钢筋表面性质、直径和间距、混凝土的品种、强度、构件尺寸、单元划分等许多因素的影响, 所以应从试验数据出发, 根据具体不同的情况确定。

双弹簧联结单元具有形式简单, 可以很方便地设置在钢筋与混凝土单元之间, 而不影响单元划分的优点。但也有明显的不足, 不能反映变形钢筋对混凝土的楔作用。

3.4.2 四结点线性节理单元

这种单元是一种退化了的四边形单元, 即宽度等于零的四边形单元。它首先由 Goodman 用于岩石力学中作为节理单元, 后又引伸用于各种边界接触面的单元, 如钢筋与混凝土间的粘结滑移单元。由于这种单元宽度等于零, 所以可以很方便地放置于钢筋和混凝土之间而不影响钢筋与混凝土单元的几何划分。又由于这种单元从四边形单元退化而来, 可以与 4 结点平面等参单元建立更为协调的关系^[10]。

1. 水平方向的单元刚度矩阵

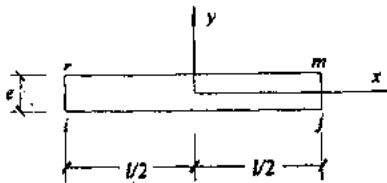


图 3.14 节理单元

图 3.14 表示一水平方向的节理单元, 单元长度为 l , 厚度(垂直于 x, y 平面)为 t , 单元有 4 个结点 i, j, m, r 。把坐标原点放在单元形心上。

单元结点力是

$$\{F\}^e = [U_i \ V_i \ U_j \ V_j \ U_m \ V_m \ U_r \ V_r]^T \quad (3.69a)$$

单元结点位移是

$$\{\delta\}^e = [u_i \ v_i \ u_j \ v_j \ u_m \ v_m \ u_r \ v_r]^T \quad (3.69b)$$

假设单元上缘 rm 和下缘 ij 上的位移是线性分布

的, 即

$$u_{\pm} = \frac{1}{2} \left(1 - \frac{2x}{l} \right) u_r + \frac{1}{2} \left(1 + \frac{2x}{l} \right) u_m$$

$$u_{\mp} = \frac{1}{2} \left(1 - \frac{2x}{l} \right) u_i + \frac{1}{2} \left(1 + \frac{2x}{l} \right) u_j$$

由此可知单元上缘与下缘的水平位移差为

$$\Delta u = u_{\pm} - u_{\mp} = \frac{1}{2} \left[(u_r - u_i) \left(1 - \frac{2x}{l} \right) + (u_m - u_j) \left(1 + \frac{2x}{l} \right) \right] \quad (a)$$

同理, 单元上缘与下缘的垂直位移差为

$$\Delta v = v_{\text{上}} - v_{\text{下}} = \frac{1}{2} \left[(v_r - v_i) \left(1 - \frac{2x}{l} \right) + (v_m - v_j) \left(1 + \frac{2x}{l} \right) \right] \quad (\text{b})$$

假设单元内剪应力与水平位移差成正比,即

$$\tau_s = \lambda_s \Delta u \quad (3.70)$$

单元内正应力与垂直位移差成正比,即

$$\sigma_n = \lambda_n \Delta v \quad (3.71)$$

其中 λ_s, λ_n 分别是节理的切向和法向劲度系数。

令

$$\{w\} = \begin{Bmatrix} \Delta u \\ \Delta v \end{Bmatrix}, \quad [\lambda] = \begin{bmatrix} \lambda_s & 0 \\ 0 & \lambda_n \end{bmatrix} \quad (3.72)$$

式(3.70), (3.71)写成矩阵形式:

$$\{\sigma\} = \begin{Bmatrix} \tau_s \\ \sigma_n \end{Bmatrix} = [\lambda] \{w\} \quad (3.73)$$

由(a), (b)两式可得

$$\{w\} = \begin{Bmatrix} \Delta u \\ \Delta v \end{Bmatrix} = [M] \{\delta\}^e \quad (3.74)$$

式中:

$$[M] = \frac{1}{2} \begin{bmatrix} -z_1 & 0 & -z_2 & 0 & z_2 & 0 & z_1 & 0 \\ 0 & -z_1 & 0 & -z_2 & 0 & z_2 & 0 & z_1 \end{bmatrix} \quad (3.75)$$

其中

$$z_1 = 1 - \frac{2x}{l}, \quad z_2 = 1 + \frac{2x}{l} \quad (\text{c})$$

假若单元各结点产生虚位移 $\{\delta^*\}$, 那么单元内的虚位移差为

$$\{w^*\} = [M] \{\delta^*\}$$

在单位长度上, 单元应力所作虚功是

$$t \{w^*\}^T \{\sigma\} = \{\delta^*\}^T t [M]^T [\lambda] \{w\}$$

沿单元长度积分后, 即得到单元应力所做虚功, 它必须等于单元结点力所做虚功, 由此得

$$\{F\}^e = \left(t \int_{-\frac{l}{2}}^{\frac{l}{2}} [M]^T [\lambda] [M] dx \right) \{\delta\}^e \quad (\text{d})$$

即

$$\{F\}^e = [k']^e \{\delta\}^e \quad (3.76)$$

$$[k']^e = t \int_{-\frac{l}{2}}^{\frac{l}{2}} [M]^T [\lambda] [M] dx \quad (3.77)$$

把 $[M]$ 的表达式(3.75)代入式(3.77), 并注意到

$$\int_{-\frac{l}{2}}^{\frac{l}{2}} z_1^2 dx = \frac{4l}{3}, \quad \int_{-\frac{l}{2}}^{\frac{l}{2}} z_1 z_2 = \frac{2l}{3}, \quad \int_{-\frac{l}{2}}^{\frac{l}{2}} z_2^2 dx = \frac{4l}{3}$$

得到水平节理单元的单元刚度矩阵如下:

$$[k']^e = \frac{lt}{6} \begin{bmatrix} 2\lambda_s & 0 & \lambda_s & 0 & -\lambda_s & 0 & -2\lambda_s & 0 \\ 0 & 2\lambda_n & 0 & \lambda_n & 0 & -\lambda_n & 0 & -2\lambda_n \\ \lambda_s & 0 & 2\lambda_s & 0 & -2\lambda_s & 0 & -\lambda_s & 0 \\ 0 & \lambda_n & 0 & 2\lambda_n & 0 & -2\lambda_n & 0 & -\lambda_n \\ -\lambda_s & 0 & -2\lambda_s & 0 & 2\lambda_s & 0 & \lambda_s & 0 \\ 0 & -\lambda_n & 0 & -2\lambda_n & 0 & 2\lambda_n & 0 & \lambda_n \\ -2\lambda_s & 0 & -\lambda_s & 0 & \lambda_s & 0 & 2\lambda_s & 0 \\ 0 & -2\lambda_n & 0 & -\lambda_n & 0 & \lambda_n & 0 & 2\lambda_n \end{bmatrix} \quad (3.78)$$

2. 整体坐标系中的单元刚度矩阵

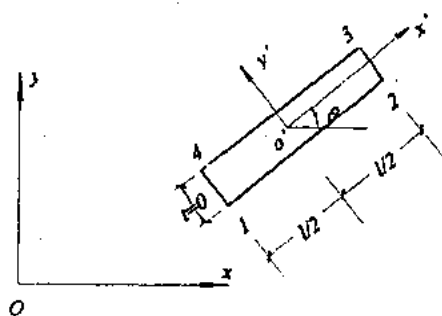


图 3.15 斜向 4 结点平面节理单元

当节理单元的方向与整体坐标系非正交,而是有一个夹角时,只要采用坐标转换的方法,可以导出其单元刚度矩阵。

设 4 结点节理单元的局部坐标系 (x', y') 和整体坐标系 (x, y) 之间的夹角为 θ , 如图 3.15 所示, 显然两种坐标系符合下列变换式:

$$\begin{Bmatrix} x \\ y \end{Bmatrix} = [L] \begin{Bmatrix} x' \\ y' \end{Bmatrix}$$

图 3.15 斜向 4 结点平面节理单元

$$\text{式中} \quad [L] = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \quad (e)$$

设局部坐标中的单元结点力为 $\{F'\}^e$, 则整体坐标中的单元结点力为

$$\{F\}^e = [T] \{F'\}^e \quad (f)$$

式中:

$$[T] = \begin{bmatrix} L & 0 & 0 & 0 \\ 0 & L & 0 & 0 \\ 0 & 0 & L & 0 \\ 0 & 0 & 0 & L \end{bmatrix} \quad (g)$$

根据坐标变换, 不难推出整体坐标系中四结点节理单元的单元刚度矩阵。

$$[k]^e = [T][k']^e[T]^T \quad (3.79)$$

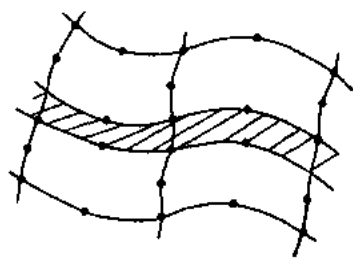
式中 $[k']^e$ 为局部坐标系中(即水平方向)的单元刚度矩阵, 见式(3.78)。

3.4.3 六结点曲边节理单元

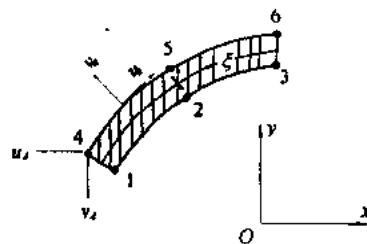
如果节理单元要与八结点等参单元联结时, 为保证单元边界的连续性, 宜采用六结点曲边节理单元。六结点节理单元的上下边界都是曲线, 见图 3.16 所示, 单元的宽度可以采用有限厚度, 也可假定为零, 此处假定单元宽度为零。

由于单元边界是曲线, 故推导单元刚度矩阵时需要类似于平面等参单元进行坐标变换。

例如, 节理下边界的坐标为



(a)



(b)

图 3.16 6 结点曲边节理单元

$$\begin{cases} x = N_1 x_1 + N_2 x_2 + N_3 x_3 \\ y = N_1 y_1 + N_2 y_2 + N_3 y_3 \end{cases} \quad (3.80)$$

其中 (x_i, y_i) 是结点 i 的坐标, $i = 1, 2, 3$ 。

其形函数 N_i 为

$$N_1 = -\xi \frac{(1-\xi)}{2}, \quad N_2 = 1 - \xi^2, \quad N_3 = \xi \frac{(1+\xi)}{2} \quad (3.81)$$

沿节理单元下边的水平位移为

$$u_{\text{下}} = N_1 u_1 + N_2 u_2 + N_3 u_3$$

沿节理单元上边的水平位移为

$$u_{\text{上}} = N_1 u_4 + N_2 u_5 + N_3 u_6$$

节理单元上、下边界的水平位移差为

$$\Delta u = N_1 (u_4 - u_1) + N_2 (u_5 - u_2) + N_3 (u_6 - u_3)$$

同样, 节理单元上、下边界的铅直位移差为

$$\Delta v = N_1 (v_4 - v_1) + N_2 (v_5 - v_2) + N_3 (v_6 - v_3)$$

所以

$$\begin{Bmatrix} \Delta u \\ \Delta v \end{Bmatrix} = [N] \{\delta\}^e$$

其中 $[N] = \begin{bmatrix} -N_1 & 0 & -N_2 & 0 & -N_3 & 0 & N_1 & 0 & N_2 & 0 & N_3 & 0 \\ 0 & -N_1 & 0 & -N_2 & 0 & -N_3 & 0 & N_1 & 0 & N_2 & 0 & N_3 \end{bmatrix}$

设沿单元上、下边界切线方向的位移差为 Δu_s , 沿单元上、下边界法线方向的位移差为 Δv_n , 则 $\Delta u_s, \Delta v_n$ 可由 Δu 和 Δv 经坐标转换而求得如下:

$$\begin{Bmatrix} \Delta u_s \\ \Delta v_n \end{Bmatrix} = [L] \begin{Bmatrix} \Delta u \\ \Delta v \end{Bmatrix} = [L][N] \{\delta\}^e \quad (3.82)$$

式中

$$[L] = \begin{bmatrix} \alpha & \beta \\ -\beta & \alpha \end{bmatrix}$$

$$\alpha = \frac{\frac{\partial x}{\partial \xi}}{\sqrt{\left(\frac{\partial x}{\partial \xi}\right)^2 + \left(\frac{\partial y}{\partial \xi}\right)^2}}, \quad \beta = \frac{\frac{\partial y}{\partial \xi}}{\sqrt{\left(\frac{\partial x}{\partial \xi}\right)^2 + \left(\frac{\partial y}{\partial \xi}\right)^2}}$$

与四结点节理单元相同,引入节理的切向和法向的刚度系数 λ_s, λ_n , 可以写出单元内应力与结点位移差之间的关系式:

$$\{\sigma\} = \begin{Bmatrix} \tau_s \\ \sigma_n \end{Bmatrix} = \begin{bmatrix} \lambda_s & 0 \\ 0 & \lambda_n \end{bmatrix} \begin{Bmatrix} \Delta u_s \\ \Delta v_n \end{Bmatrix}$$

$$\text{令} \quad [\lambda] = \begin{bmatrix} \lambda_s & 0 \\ 0 & \lambda_n \end{bmatrix}; \quad \{w\} = \begin{Bmatrix} \Delta u_s \\ \Delta v_n \end{Bmatrix} = [L][N]\{\delta\}^e$$

$$\text{则} \quad \{\sigma\} = [\lambda]\{w\}$$

若单元各结点产生虚位移 $\{\delta^*\}$, 相应单元内的虚位移差为

$$\{w^*\} = [L][N]\{\delta^*\}$$

单元应力所作的虚功为

$$\int t \{w^*\}^T \{\sigma\} dV = t \int \{\delta^*\}^T [N]^T [L]^T [\lambda] [L] [N] \{\delta\}^e dV$$

根据虚功原理, 结点力 $\{F\}^e$ 在虚位移 $\{\delta^*\}$ 上作的虚功要与单元内的虚功相等, 于是可得到结点力和结点位移的关系式:

$$\{F\}^e = [k]^e \{\delta\}^e$$

式中, 单元刚度矩阵为

$$[k]^e = \int [N]^T [L]^T [\lambda] [L] [N] dV$$

$$[k]^e = t \int_{-1}^1 [N]^T [L]^T [\lambda] [L] [N] S d\xi \quad (3.83)$$

其中

$$S = \sqrt{\left(\frac{\partial x}{\partial \xi}\right)^2 + \left(\frac{\partial y}{\partial \xi}\right)^2}$$

$\frac{\partial x}{\partial \xi}$ 和 $\frac{\partial y}{\partial \xi}$ 应根据 (3.80) 式计算。

式 (3.83) 可用数值积分法算出, 必须指出, 由此式计算的是整体坐标系 (x, y) 中的单元刚度矩阵, 不必再进行坐标转换。

3.4.4 粘结区单元和斜压杆单元

1. 粘结区单元 (Bond Zone Element)^[14]

Groot 在 1981 年提出了粘结区单元, 这种单元假定钢筋外围有一厚度为 β_r 的圆柱体滑移层 (图 3.17), 它由一系列平行的受压锥体所组成, 用以模拟出现内部斜裂缝后, 混凝土被分割成锥状的软化层。

τ 与 σ_r 是相关的, 其合力与受压锥体的内力相平衡, 根据压杆的刚度可导出:

$$\begin{Bmatrix} \tau \\ \sigma_r \end{Bmatrix} = \frac{E_t \cos^3 \phi}{\beta_r} \begin{bmatrix} \tan^2 \phi & -\tan \phi \\ -\tan \phi & 1 \end{bmatrix} \begin{Bmatrix} \Delta u \\ \Delta v \end{Bmatrix} \quad (3.84)$$

式中, E_l 为滑移层的刚度, ϕ 为锥体倾斜角。应该说粘结区单元, 在模拟粘结作用上比双弹簧模型有本质上的改进, 其最大特点是反映了变形钢筋的楔作用, 但在实际应用中也存在如下问题:

(1) Groot 没有给出滑移层参数 β , E_l 及 ϕ 的取值方法, 在分析时难以合理选用;

(2) 粘结区单元是有实际尺寸的轴对称单元, 用于二维或三维有限元分析在单元划分上将带来很大的复杂性;

(3) 计算工作量大, 但结果也并不理想。

1984 年 Dragosavic 对 Groot 的模型作了改进, 明确滑移层厚度为 $0.5d$ (d 为钢筋直径), 并修正了 Groot 取 τ 及 σ_r 在滑移层内为常数的不合理假设, 给出了与式 (3.84) 相似的刚度矩阵。

2. 斜压杆单元 (Inclined Strut Bond Element)

在分析了各种粘结单元的优缺点后, 文献 [14] 中介绍了一种新的斜压杆单元。

变形钢筋与混凝土之间的力的传递主要是肋对混凝土的斜向挤压力, 它使混凝土产生如图 3.18 所示的内部锥状裂缝及径向裂缝, 图中 $W_{s,c}$ 和 $W_{c,c}$ 分别为钢筋处和外部观察到的裂缝宽度。随着荷载增大, 肋处混凝土逐渐被挤碎, 钢筋与混凝土将沿着挤碎后粉末堆积形成的滑移面发生剪切位移。显然当钢筋与混凝土产生轴向位移 Δu (滑移) 时, 将同时产生径向相对位移 (图 3.19), 两者的关系是 $\tan \theta \approx -\Delta v / \Delta u$ 。

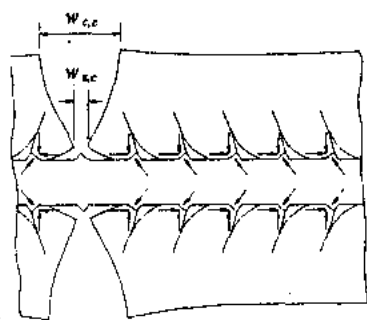


图 3.18 内部锥状裂缝

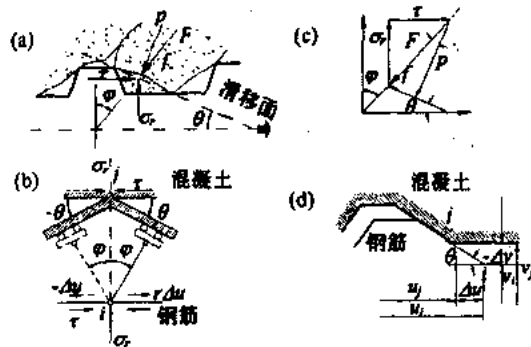


图 3.19 斜压杆单元

变形钢筋的这种粘结滑移表现可用图 3.19 所示斜压杆模型来模拟, 它有弹性刚度, 但无实际尺寸, 能设置在单元结点处或界面处构成连续的接触单元。其结点力与结点相对位移的关系为:

$$\begin{Bmatrix} F_u \\ F_v \end{Bmatrix} = k \begin{bmatrix} \tan^2 \phi & -\tan \phi \\ -\tan \phi & 1 \end{bmatrix} \begin{Bmatrix} \Delta u \\ \Delta v \end{Bmatrix}$$

式中取 $k = k_u / \tan \phi (\tan \phi - \tan \theta)$ 为斜压杆刚度, 式中 k_u 为切向弹簧刚度。根据试验分

析,计算时可取 $\theta = 40^\circ$, $\phi = 60^\circ$ 。

3.5 钢筋混凝土结构有限元模型的选择

用有限元方法来分析钢筋混凝土结构与一般固体力学中的有限元分析,在基本原理与方法上是一样的,但如何进行结构离散化时,又有其特殊性。因为钢筋混凝土结构由钢筋和混凝土两种不同的材料所组成,因此,建立钢筋混凝土有限元模型时必须考虑到由于材料的不均匀性和下列各种因素的影响:

- (1) 钢筋一般被包裹于混凝土之中,而且相对体积较小;
- (2) 混凝土应力应变的非线性性能及在复合应力作用下的本构关系;
- (3) 由于混凝土受拉性能很差,开裂荷载较小,而且裂缝会连续不断地出现。此时局部位移及应力对整个结构的强度及位移会产生影响,结构计算图式随时都可按需要而改变;
- (4) 随着荷载增加,混凝土和钢筋之间发生相对滑移,粘结力可能发生破坏;
- (5) 混凝土的收缩和徐变的影响等。

另外,钢筋混凝土有限元分析也应根据求解问题的具体要求,计算精度和计算机硬件的具体情况来选择相应的有限元计算模型。目前构成钢筋混凝土结构的有限元模型一般主要有三种方式:整体式、组合式和分离式,现介绍如下。

3.5.1 整体式模型

当分析区域较大,受计算机软件 and 硬件的限制,无法将钢筋和混凝土分别划分单元,同时人们所关心的计算结果是结构物在外荷载作用下的宏观反映(如结构的总体位移和应力分布情况等),这种情况下采用整体式模型比较合适。

在整体式有限元模型中,将钢筋弥散于整个单元中,并把单元视为连续均匀的材料。钢筋对整个结构的贡献,可以通过调整单元的材料力学性能参数来体现,例如提高材料的屈服强度 σ_y ,材料的弹性模量等。钢筋对整个结构的贡献作用的另一种处理方法是一次求得综合的单元刚度矩阵,把弹性矩阵改为由钢筋和混凝土两部分组成,其具体表达式为:

$$[D] = [D_c] + [D_s] \quad (3.85)$$

式(3.85)中, $[D_c]$ 为混凝土材料的弹性矩阵,在开裂前可按一般匀质体计算,其表达式为

$$[D_c] = \begin{bmatrix} d_{11} & d_{12} & d_{13} & 0 & 0 & 0 \\ & d_{22} & d_{23} & 0 & 0 & 0 \\ & & d_{33} & 0 & 0 & 0 \\ & & & d_{44} & 0 & 0 \\ & & & & d_{55} & 0 \\ & & & & & d_{66} \end{bmatrix} \quad (3.86)$$

$$\begin{aligned}
 \text{式中} \quad d_{11} = d_{22} = d_{33} &= \frac{E_c(1-\nu)}{(1+\nu)(1-2\nu)} \\
 d_{12} = d_{13} = d_{23} &= \frac{\nu E_c}{(1+\nu)(1-2\nu)} \\
 d_{44} = d_{55} = d_{66} &= \frac{E_c}{2(1+\nu)}
 \end{aligned}$$

随着荷载不断增加,混凝土开裂,裂缝不断出现,上述计算式应作相应的调整。式(3.85)中的 $[D_s]$ 为钢筋的弹性矩阵,其表达式为

$$[D_s] = E_s \begin{bmatrix} \rho_x & 0 & 0 & 0 & 0 & 0 \\ & \rho_y & 0 & 0 & 0 & 0 \\ & & \rho_z & 0 & 0 & 0 \\ \text{对 称} & & & 0 & 0 & 0 \\ & & & & 0 & 0 \\ & & & & & 0 \end{bmatrix} \quad (3.87)$$

式中, E_s 为钢筋的弹性模量; ρ_x, ρ_y 和 ρ_z 分别为沿 x, y 和 z 方向的配筋率。

整体式模型可采用各种平面单元,如三结点三角形单元,平面矩形单元,四结点或八结点等参单元,也可根据需要采用三维单元。

整体式模型的缺点是显而易见的,它无法揭示钢筋和混凝土之间相互作用的微观机理。

3.5.2 分离式模型

与整体式模型相反,将混凝土和钢筋各自划分成足够小的单元,按照混凝土和钢筋不同的力学性能,选择多种不同的单元形式,这就是分离式模型。

对于平面问题,混凝土材料可采用三角形单元,矩形单元,四结点或八结点等参单元。钢筋同样也可采用三角形单元,矩形单元或平面等参单元,但考虑到钢筋几何形状相对于混凝土是细长的,通常可采用一维杆件单元,等参杆单元,这样处理的好处是可以大大减少单元和结点数目,并且可以避免因钢筋单元划分过细,而在钢筋和混凝土的交界面采用太多的过渡单元,分离式钢筋混凝土有限元模型见图3.20。

受到外力作用后,构件中的钢筋和混凝土之间在相互约束的同时会产生相对滑移,为模拟两者之间这种粘结约束和相对滑移,可插入前述的联结单元(包括双弹簧的联结单元、四结点节理单元、六结点节理单元)。

经分析,双弹簧单元和四边形节理单元作为粘结单元均可用于实际结构分析,但当沿钢筋的单元网格划分不均匀时,或者在混凝土和钢筋应用高次插值单元的情况下,四结点和六结点节理单元可以更自然地得到比较协调的刚度矩阵。

分离式模型可揭示钢筋和混凝土之间相互作用的微观机理,这是整体式模型无法做到的。当然,分离式模型对计算机容量和速度的要求比较高,在需要对结构件内微观受力机理进行分析研究时,分离式模型的优点显得尤为突出。

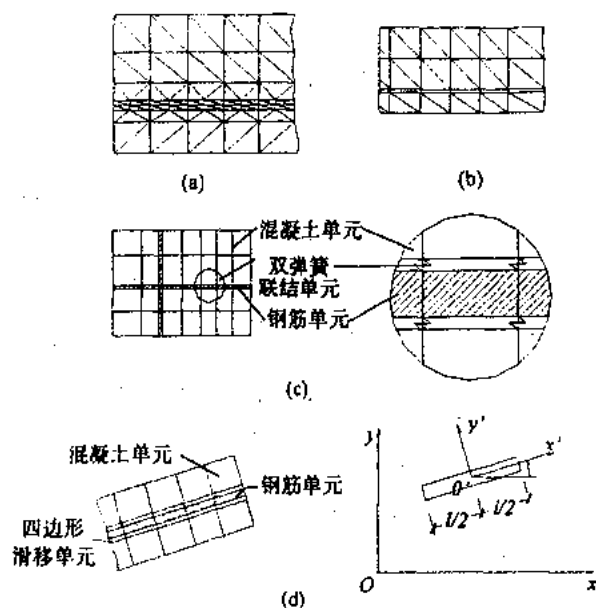


图 3.20 分离式钢筋混凝土有限元模型

3.5.3 组合式模型

组合式模型介于整体式和分离式模型之间。组合式模型假定钢筋和混凝土两者之间的相互粘结很好,不会有相对滑移。于是在单元分析时,可分别求得混凝土和钢筋对单元刚度矩阵的贡献,组成一个复合的单元刚度矩阵。最常用的组合式模型有两种方式,第一种为分层组合式,该组合方式在杆件系统中应用较广。第二种为混凝土和钢筋复合单元、现简述如下。

1. 分层组合式

对于受弯构件,该方法是将构件沿纵向分成若干单元,每个单元在其横截面上分成许多混凝土条带和钢筋条带,并假定每一条带上的应力 $\sigma_{c,i}$ 和 $\sigma_{s,i}$ 是均匀分布的。在条带划分中,为了考虑破坏阶段压区高度较小的情况,需要增加条带数才能达到一定的精度,具体的分析计算步骤见第五章。

2. 钢筋、混凝土复合单元

在建立单元刚度矩阵时,不但要考虑混凝土材料的作用,而且要考虑钢筋的刚度贡献,常用的钢筋混凝土复合单元有:带钢筋的四边形单元,带钢筋膜的八结点六面体单元。下面介绍带钢筋的四边形单元,如图 3.21 所示。

采用四结点任意四边形单元,单元结点力和结点位移分别由下式表示:

$$\{F\}^e = \{X_1 \ Y_1 \ X_2 \ Y_2 \ X_3 \ Y_3 \ X_4 \ Y_4\}^T$$

$$\{\delta\}^e = \{u_1 \ v_1 \ u_2 \ v_2 \ u_3 \ v_3 \ u_4 \ v_4\}^T$$

由前述可知,不考虑钢筋作用时,任意四边形单元的结点力和结点位移的关系为

$$\{F\}^e = [K_c] \{\delta\}^e \quad (3.88)$$

式中,混凝土单元刚度矩阵即为前述的任意四边形单元刚度矩阵:

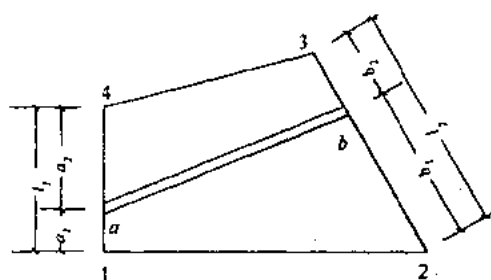


图 3.21 带钢筋的任意四边形单元

$$[K_c] = \int_{-1}^{+1} \int_{-1}^{+1} [B]^T [D] [B] |J| t d\xi d\eta$$

钢筋两端 a, b 两个结点的结点力和结点位移为

$$\{\bar{F}_s\} = [X_a \ Y_a \ X_b \ Y_b]^T \quad (a)$$

$$\{\bar{\delta}_s\} = [u_a \ v_a \ u_b \ v_b]^T \quad (b)$$

对于单根钢筋,其结点力和结点位移之间关系为

$$\{\bar{F}_s\} = [\bar{K}_s] \{\bar{\delta}_s\}^e \quad (3.89)$$

式中,钢筋单元刚度矩阵为:

$$[\bar{K}_s] = \frac{EA}{l} \begin{bmatrix} c^2 & cs & -c^2 & -cs \\ cs & s^2 & -cs & -s^2 \\ -c^2 & -cs & c^2 & cs \\ -cs & -s^2 & cs & s^2 \end{bmatrix} \quad (3.90)$$

由于钢筋两端 a, b 处的结点位移 $\{\bar{\delta}_s\}$ 并非是独立变量,它们与结点 1, 2, 3, 4 的位移是紧密相关的。由图 3.20 中的几何关系可知,当单元结点 1 发生单位位移 $u_1 = 1$,而其余结点位移为零时,有

$$\frac{u_a}{a_2} = \frac{u_1}{l_1}$$

即

$$u_a = \frac{a_2}{l_1}$$

同理,当 $v_1 = 1$,其余结点位移均为 0 时,可推得

$$v_a = \frac{a_2}{l_1}$$

按照上述的虚位移原理,可推得钢筋结点位移 $\{\bar{\delta}_s\}$ 与单元四个结点位移 $\{\delta\}^e$ 之间的关系式:

$$\{\bar{\delta}_s\} = \begin{Bmatrix} u_a \\ v_a \\ u_b \\ v_b \end{Bmatrix} = \begin{bmatrix} \frac{a_2}{l_1} & 0 & 0 & 0 & 0 & 0 & \frac{a_1}{l_1} & 0 \\ 0 & \frac{a_2}{l_1} & 0 & 0 & 0 & 0 & 0 & \frac{a_1}{l_1} \\ 0 & 0 & \frac{b_2}{l_2} & 0 & \frac{b_1}{l_2} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{b_2}{l_2} & 0 & \frac{b_1}{l_2} & 0 & 0 \end{bmatrix} \begin{Bmatrix} u_1 \\ v_1 \\ u_2 \\ v_2 \\ u_3 \\ v_3 \\ u_4 \\ v_4 \end{Bmatrix}$$

即 $\{\bar{\delta}_s\} = [R]\{\delta\}^e \quad (3.91)$

其中, $[R]$ 称为坐标转换矩阵,式(3.91)表达了 4×1 阶的钢筋结点位移与 8×1 阶的四边形单元结点位移之间的转换关系。

根据力的平衡,用上述类似的方法可以推得单元结点力与钢筋结点力之间的关系:

$$\{F\}^e = [R]^T \{\bar{F}_s\} \quad (3.92)$$

将式(3.89)和式(3.90)代入式(3.91),可得到

$$\{F\}^e = [R]^T [\bar{K}_s] [R] \{\delta\}^e = [K_s] \{\delta\}^e \quad (3.93)$$

式中: $[K_s] = [R]^T [\bar{K}_s] [R] \quad (3.94)$

式(3.94)为钢筋对整个四边形单元的刚度贡献矩阵。作为混凝土和钢筋的复合单元,其单元刚度矩阵即为两者的叠加:

$$[K]^e = [K_c] + [K_s] \quad (3.95)$$

第四章 非线性有限元分析的计算方法

4.1 混凝土的开裂与破坏

4.1.1 概述

混凝土的破坏准则是判别混凝土是否破坏的依据。在单轴应力状态下,可以用混凝土轴心抗压强度或混凝土轴心抗拉强度等一种指标来表示,建立其强度破坏条件是比较方便的。在复杂应力状态下,如何建立结构材料强度破坏条件的问题早就得到了研究。古典早期的强度破坏理论曾对促进现代技术水平的发展起过重大的作用,由于古典的强度理论均具有明确的力学概念,如最大正应力,最大剪应力等达到某一数值时,即产生破坏,所以有些理论至今仍全部或部分地被应用着。早期的理论主要有:最大正应力理论;最大剪应力理论(Tresca 理论);第四强度理论(Von Mises 理论);莫尔强度理论;Drucker-Prager 破坏准则等。

70 年代以来,随着钢筋混凝土有限元分析方法的发展,对复杂应力状态下混凝土破坏准则的研究日益得到重视。混凝土在复杂受力状态下的强度是一个比较复杂的问题,目前仍未建立起比较完善的能解释不同破坏物理现象的混凝土强度理论,所以须依赖强度试验结果。混凝土在复杂应力状态下的强度要考虑不同应力分量之间的相互影响,一般也可用混凝土的应力状态 σ_{ij} 的函数来表示,即

$$f(\sigma_{ij}, k_1, k_2, \dots, k_n) = 0 \quad (4.1)$$

式中, $k_1, k_2, \dots, k_n$ 为反映材料性质的参数,由材料强度试验结果确定。包含有 n 个参数的破坏准则称为 n 参数破坏准则,为了能更好地描述混凝土强度性能,近年来强度准则的参数有愈来愈多的趋势。

本节将对混凝土材料在单轴和多轴应力下的强度问题,介绍几个主要的破坏准则。

4.1.2 单轴受力下的应力-应变关系

1. 单轴受压 σ - ϵ 关系

从试验可以得到混凝土应力应变全过程曲线,混凝土在单向受压下的破坏条件,一般可取用混凝土的极限应变达到 0.003 ~ 0.004 为准。根据试验结果,不同的研究者采用不同的数学表达式,下面介绍几种在钢筋混凝土有限元分析中应用较广的数学表达式。

(1) Saenz 等人的表达式

Saenz 等人提出的表达式见第二章(2.3.1 节),这一公式能很好地表达混凝土的 σ - ϵ 曲线,特别是上升段曲线公式较简单。为了更好地描述下降段曲线的性质(参见图 4.1),Elwi 和 Murray 又对其表达式进行了改进,图中破坏点的 $\epsilon_u = 0.003 \sim 0.004$ 。

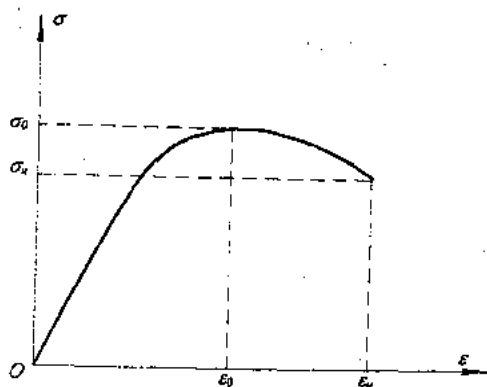


图 4.1 Elwi 和 Murray 改进的混凝土应力-应变表达式

(2) Sargin 表达式

在对 Saenz 公式作了修改后, Sargin 提出了下列公式:

$$\sigma = k_3 f_c \frac{A \left(\frac{\epsilon}{\epsilon_0} \right) + (D-1) \left(\frac{\epsilon}{\epsilon_0} \right)^2}{1 + (A-2) \frac{\epsilon}{\epsilon_0} + D \left(\frac{\epsilon}{\epsilon_0} \right)^2} \quad (4.2)$$

其中 $A = E_0/E_s$

式中 E_0 ——混凝土初始弹性模量;

E_s ——应力达到峰值时的割线模

量;

$k_3 = \sigma_0/f_c$ ——侧向约束对强度的影

响系数, $k_3 = 1$ 时, 适

合于无侧向约束的混

凝土;

D ——主要影响下降段的参数, 不

同 D 值的 $\sigma-\epsilon$ 曲线示于图 4.2。

(3) 我国学者提出的表达式。

国内许多单位和研究者对混凝土 $\sigma-\epsilon$

全曲线进行了研究, 清华大学提出了分项

的表达式, 四川建筑科学研究院建议的公式为

$$\sigma = \sigma_0 \left[2 \frac{\epsilon}{\epsilon_0} - \left(\frac{\epsilon}{\epsilon_0} \right)^2 \right] \quad \text{当 } \epsilon \leq \epsilon_0 \quad (4.3)$$

$$\sigma = \sigma_0 \left[1 - \alpha \left(\frac{\epsilon - \epsilon_0}{\epsilon_u - \epsilon_0} \right) \right] \quad \text{当 } \epsilon_0 < \epsilon \leq \epsilon_u \quad (4.4)$$

$$\text{式中 } \epsilon_0 = (0.463 \sqrt[3]{\sigma_0} - 1.22) \times 10^{-3} \quad (4.5)$$

$$\alpha = 0.062 \sqrt[3]{\sigma_0} - 0.09 \quad (4.6)$$

$\sigma_0 = 0.85 f'_c$ (f'_c 为混凝土圆柱体抗压强度)。

这一公式对混凝土的下降段不再取定值,该公式是在 Hongnestad 的曲线基础上得出的一个建议公式。

2. 单轴受拉 σ - ϵ 关系

混凝土轴向受拉应力-应变关系的试验研究较少,其破坏条件一般取用拉应力达到抗拉强度为准,在有限元分析中,混凝土单向受拉下的应力-应变关系一般简化为线性关系,其表达式可采用两折线,三段斜直线等。对于混凝土受拉时 σ - ϵ 曲线的下降段,各国学者也提出了不同形式的表达式。有单直线下降式,分段下降式,多段下降式,指数或曲线下降式等。

有关按两直线、三段斜直线假定的表达式可参见第二章所介绍的,下面介绍几种下降段的表达式。

(1) 多段直线下降式

在单直线下降式的基础上,美国的 Kang 和 Lin 提出了如图 4.3 所示的多段斜直线下降式。

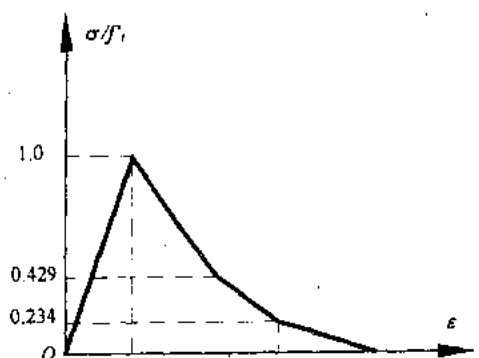


图 4.3 多段斜直线的下降式

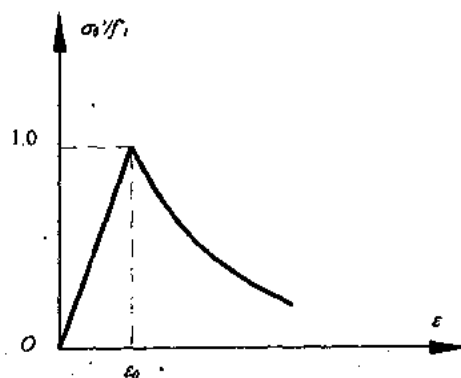


图 4.4 用指数式表示的下降段

(2) 指数下降式

文献[6]介绍了用指数式表示的下降段,即

$$\frac{\sigma}{f_t} = e^{-a(\epsilon - \epsilon_0)} \quad (4.7)$$

式中, a 是控制下降斜率的一个重要参数,该式能适应不同性质的混凝土受拉时的下降曲线,见图 4.4。

4.1.3 混凝土双向受力下的破坏条件

对于双向受力状态,根据混凝土在不同应力比 $k = \sigma_1/\sigma_2$ 下所做的强度破坏试验的结果,提出了较多的破坏准则,图 4.5 所示为 Kupfer 等人建议的计算公式强度包络线。

从图中可看到,双轴受压时混凝土在不同应力比值 $k = \sigma_1/\sigma_2$ 下的强度变化规律。由于一个方向的压应力会对另一方向压应力所引起的侧向变形形成一定程度的约束,限制了内部微裂缝的发展,从而提高了混凝土的抗压强度。双轴抗压强度的最大值并不发生

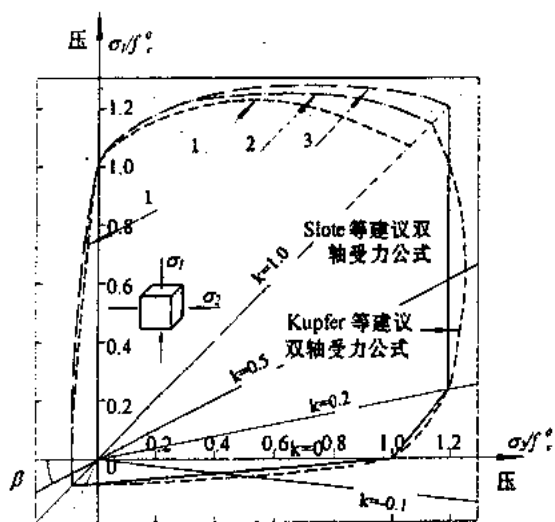


图 4.5 Kupfer 等人建议的计算公式强度包络线

在 $\sigma_1 = \sigma_2$ 时,而是发生在 σ_1/σ_2 约等于 2 或 0.5 的情况下,强度提高的幅度约为 25%。

混凝土处于一向受压、另一向受拉状态时(第二和第四象限),其强度均低于单轴受力(拉或压)时的强度。这是由于一个方向作用的拉应力会加大另一方向压应力所引起的侧向变形,从而加速内部微裂缝的发展,使受压方向混凝土强度降低。

双轴受拉时,不论两个方向的应力比值 σ_1/σ_2 如何变化,混凝土抗拉强度均基本保持单轴受拉时的强度不变。

现将常用的几种强度计算公式介绍如下:

下:

1. Kupfer 公式

Kupfer 等建议的双轴受力强度公式常被用于有限元分析中,具体公式见第二章公式 (2.41a) ~ (2.43)。

2. 修正的莫尔库伦准则

莫尔-库伦屈服准则可写为

$$\tau = c + \sigma \tan \varphi \quad (4.8)$$

式中 c ——内聚力;

φ ——内摩擦角;

σ, τ ——分别为破坏面上的法向应力和抗剪强度。

对于混凝土材料,很少去测定 c 和 φ ,而是用混凝土的抗拉强度 f_t 和抗压强度 f_c 来表示,根据图 4.6 可推导莫尔破坏准则的修正表达式。

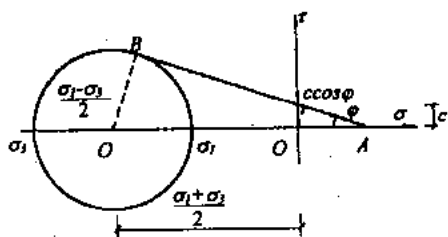


图 4.6 莫尔圆

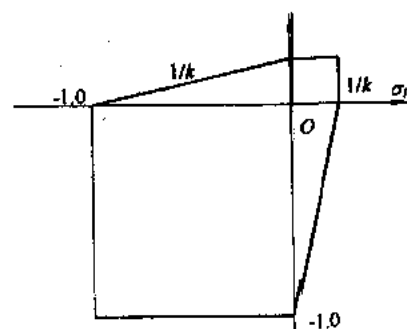


图 4.7 破坏包络线

由图 4.6 可得

$$OB = OA \cdot \sin \varphi = \left(C \frac{\cos \varphi}{\sin \varphi} + \frac{\sigma_1 + \sigma_3}{2} \right) \sin \varphi$$

$$\frac{1}{2}(\sigma_1 - \sigma_3) = c \cos \varphi - \frac{1}{2}(\sigma_1 + \sigma_3) \sin \varphi$$

将 $\mu = \operatorname{tg} \varphi$ 代入得

$$(\mu + \sqrt{1 + \mu^2})^2 \sigma_1 - \sigma_3 = 2C(\mu + \sqrt{1 + \mu^2})$$

令

$$K = (\mu + \sqrt{1 + \mu^2})^2$$

则可得

$$K \sigma_1 - \sigma_3 = 2C \sqrt{K} \quad (4.9)$$

单轴抗压时: $\sigma_1 = \sigma_2 = 0$, 式(4.9)可写为 $-\sigma_3 = 2c \sqrt{K} = f_c$

单轴抗拉时: $\sigma_1 = f_t$ 时破坏, $K \sigma_1 = K f_t = f_c$ 可得: $f_t = \frac{f_c}{K}$

这就是修正的莫尔库伦准则, 对于平面应力状态, 当

$$\text{双向压拉时:} \quad \sigma_1 > \sigma_2 > 0 \quad K \sigma_1 = f_c \quad (4.10)$$

$$\text{一拉一压时:} \quad \sigma_1 > 0 > \sigma_2 \quad K \sigma_1 - \sigma_2 = f_c \quad (4.11)$$

$$\text{双向受压时:} \quad 0 > \sigma_1 > \sigma_2 \quad -\sigma_2 = f_c \quad (4.12)$$

用图表示破坏包络线见图 4.7。用莫尔准则求得的强度值比试验值小, 由于该公式简单, 求得的强度计算结果偏于安全, 所以仍被应用。

3. Liu, Nilson 和 Slate 公式(多折线公式)

双向受压(在双压区采用双折线代替莫尔准则中的单直线):

$$k = \frac{\sigma_1}{\sigma_2} < 0.2 \quad \begin{cases} \sigma_{2c} = \left(1 - \frac{k}{1.2 - k}\right) f_c \\ \sigma_{1c} = k \sigma_2 \end{cases} \quad (4.13)$$

$$0.2 \leq k \leq 1.0 \quad \begin{cases} \sigma_{2c} = 1.2 f_c \\ \sigma_{1c} = k \sigma_2 \end{cases} \quad (4.14)$$

一拉一压时, 其表达式与莫尔准则相同。

4. Nilson 的多折线修正公式:

双向受压

$$0 \leq k \leq 0.5 \quad \sigma_{2c} = \left(0.46 \frac{\sigma_1}{f_c} - 0.9\right) f_c \quad (4.15)$$

$$0.5 < k \leq 1 \quad \sigma_{2c} = \left(-0.18 \frac{\sigma_1}{f_c} - 1.28\right) f_c \quad (4.16)$$

一拉一压:

$$\sigma_{2c} = \left(-1.6 \frac{\sigma_1}{f_c} - 0.9\right) f_c \quad (4.17)$$

双向受拉:

$$\sigma_{1t} = f_t = 0.055 f_c \quad (4.18)$$

5. Lin 和 Scordelis 公式^[4]

混凝土破坏条件的另一种表示式是用主应变作为参数。例如对于双向受压情况取用

$$\sqrt{\epsilon_1^2 - \epsilon_1 \epsilon_2 + \epsilon_2^2} - \epsilon_p = 0 \quad (4.19)$$

其中: ϵ_1, ϵ_2 为双向受压时的主应变; ϵ_p 为单向受压时的极限应变值。

这一关系式近似利用了塑性力学等效应变的概念。又如对于一向受压, 一向受拉情况, Robins 和 Kong 公式

$$\epsilon_t = 10^{-8} (a - b \sigma_m) \quad (4.20)$$

其中: ϵ_t 为混凝土极限拉应变; a, b 为常数, 取 $a = 105, b = 14.5$;

σ_m 为平均应力, $\sigma_m = \frac{\sigma_1 + \sigma_2 + \sigma_3}{3}$, (对平面应力问题, $\sigma_3 = 0$)。

4.1.4 混凝土三轴受压状态下的破坏条件

典型的三轴受压强度试验是通过液体静压力对混凝土圆柱体施加径向等压应力后再对圆柱体施加纵向压力直到破坏。试件由于受到侧向压力, 混凝土内部微裂缝的发展受到阻碍。图 4.8 所示为混凝土在三轴受压状态下的极限强度实测结果, 由图可见在一定的 σ_3/f_c^0 比值下, σ_1/f_c^0 和 σ_2/f_c^0 的变化类似于双轴受压情况, 但混凝土三轴受压强度随侧向等压值的增大而提高。

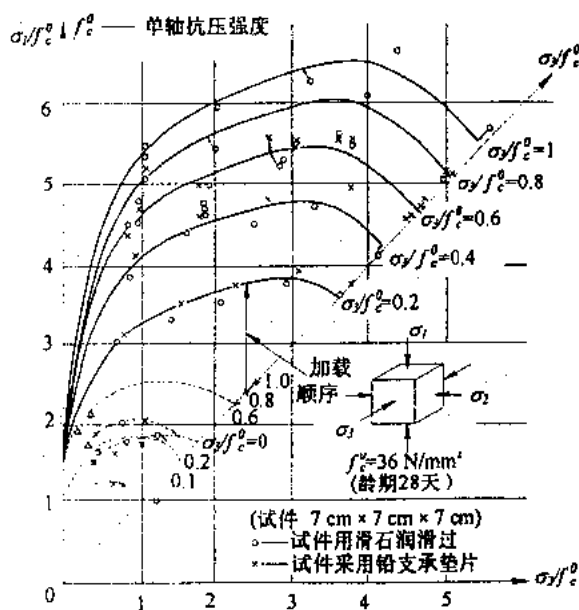


图 4.8 三轴应力状态下混凝土的极限强度变化

在复杂的应力状态下, 某一点的应力状态是由六个应力分量确定的, 不能任意选取某一个应力分量的数值作为判断材料是否破坏的标准。混凝土的强度要考虑不同应力分量之间的相互影响, 就要用应力状态的某种函数表达, 在三维空间可用一个破坏包络曲面来表示。

根据材料强度试验得到的一些破坏准则, 均是应力状态或其主应力 ($\sigma_1, \sigma_2, \sigma_3$) 的函数, 为了用数学方法表达方便, 又可用应力不变量 I_1, J_2, J_3 来表示, 或用圆柱坐标系亦

称为 Haigh-Westergaard 坐标(即 ξ, ρ, θ)来表示,也用八面体应力坐标轴来表示。因此,破坏曲面的函数方程式可表达为:

$$f(\sigma_1, \sigma_2, \sigma_3) = 0$$

$$f(I_1, J_2, J_3) = 0$$

$$f(\xi, \rho, \theta) = 0$$

$$f(\sigma_{oct}, \tau_{oct}, \theta) = 0$$

式中, σ_{oct}, τ_{oct} 为八面体正应力和剪应力。

混凝土在多轴应力下的强度问题,各国学者提出过多种不同的强度准则数学模型(如 Argyris 等, 1974; Kotsovos 和 Newman, 1978; Ottosen, 1977; 以及 Willam 和 Warnke, 1974), 有关这些破坏准则的具体表达式及其各自的优缺点和适用性可参阅 Chen W. F “Plasticity in Reinforced Concrete” 一书。现将简单介绍几种。

1. 三参数模型

(1) Bresler-Pister 的破坏面方程为

$$\frac{\sigma_{oct}}{f_c} = a - b \frac{\sigma_{oct}}{f_c} + c \left(\frac{\sigma_{oct}}{f_c} \right)^2 \quad (4.21)$$

(2) Willam-Warnke 的三参数破坏曲面方程为

$$f(\sigma_m, \tau_m, \theta) = \frac{1}{\rho} \cdot \frac{\sigma_m}{f_c} + \frac{1}{r(\theta)} \frac{\tau_m}{f_c} - 1 = 0 \quad (4.22)$$

(3) 文献[6]介绍的清华大学提出的三参数破坏准则为

$$a\rho^{1.5} + b\cos\theta\rho + c\xi = 1 \quad (4.23)$$

以上公式均包含了 3 个参数,这些参数由试验确定。

2. 四参数模型

(1) Ottosen 公式

$$f(I_1, J_2, \theta) = A \frac{J_2}{f_c^2} + \lambda(\theta) \frac{\sqrt{J_2}}{f_c} + B \frac{I_1}{f_c} - 1 = 0 \quad (4.24)$$

Ottosen 公式基本上能反映出混凝土破坏曲面的特征,但公式中的 4 个参数较难确定,4 个参数是由 4 组强度试验来确定的。

(2) 文献[6]介绍的四参数公式

$$a \frac{J_2}{f_c^2} + (b + c \cdot \cos\theta) \frac{\sqrt{J_2}}{f_c} + d \frac{I_1}{f_c} - 1 = 0 \quad (4.25)$$

3. 五参数模型

(1) Willam-Warnke 在三参数模型的基础上,提出了用平均剪应力及平均正应力表示的五参数公式

$$\begin{cases} \frac{\tau_{mc}}{f_c} = a_0 + a_1 \frac{\sigma_m}{f_c} + a_2 \left(\frac{\sigma_m}{f_c} \right)^2 & \theta = 60^\circ \\ \frac{\tau_{mt}}{f_c} = b_0 + b_1 \frac{\sigma_m}{f_c} + b_2 \frac{\sigma_m^2}{f_c^2} & \theta = 0^\circ \end{cases} \quad (4.26)$$

式中

$$\sigma_m = \frac{I}{3}, \tau_m = \frac{\rho}{\sqrt{5}} = \sqrt{\frac{2}{5}} J_2$$

公式中有 5 个独立的参数,这 5 个参数可由材料试验确定,具体计算可参阅有关的书。

(2) 清华大学王传志、过镇海等提出的五参数破坏准则

$$\tau_0 = a \left(\frac{b - \sigma_0}{c - \sigma_0} \right)^d \quad (4.27)$$

式中:

$$c = c_t \left(\cos \frac{3\theta}{2} \right)^{1.5} + c_c \left(\sin \frac{3\theta}{2} \right)^{1.5}$$

$$\sigma_0 = \frac{\sigma_{oct}}{f_c} \quad \tau_0 = \frac{\tau_{oct}}{f_c}$$

$$\theta = \frac{1}{3} \arccos \frac{3\sqrt{3}J_3}{2\sqrt{J_2^3}}$$

公式中的 5 个参数 a, b, c_t, c_c 和 d 均有明确的物理意义。

(3) 文献[22]公式

在分析了 Willam 等人提出的混凝土破坏准则存在问题的基础上,文献[22]介绍了一个改进了的五参数混凝土破坏准则。

$$\frac{\sigma_m}{f_c} = a_0 + a_1 \frac{\tau_m}{f_c} + a_2 \left(\frac{\tau_m}{f_c} \right)^2 \quad \theta = 0 \quad (4.28)$$

$$\frac{\sigma_m}{f_c} = b_0 + b_1 \frac{\tau_m}{f_c} + b_2 \left(\frac{\tau_m}{f_c} \right)^2 \quad \theta = \frac{\pi}{3} \quad (4.29)$$

该模式符合混凝土破坏曲面的主要特征,与试验数据吻合较好,参数确定较容易,具体计算可参阅文献[22]。

4.2 有限元非线性方程组的解法

4.2.1 概述

非线性问题可以分为三类:几何非线性、材料非线性及边界条件非线性。几何非线性问题是由于结构的位移或应变相当大,以致于必须按照变形后的几何位置来建立平衡方程;材料非线性问题是由材料本身的非线性应力应变关系引起的,例如钢筋混凝土结构中混凝土弹塑性变形,钢筋屈服和硬化,混凝土的收缩、徐变等;边界非线性问题是指边界条件的变化引起的非线性问题。如图 4.9(a)所示的梁,下边有一根柱子,柱与梁底之间有一间隙 Δ ,当梁受荷后挠度大于 Δ 时,柱子才对梁起支承作用,其荷载挠度曲线如图 4.9(b)所示,就整个加载过程看,是非线性问题。

在有些情况下,要同时考虑上述三种情况的某种组合,例如要同时考虑材料非线性和几何非线性,即既要涉及非线性的本构关系,又要涉及大变形问题,这是更为复杂的非线性问题。

无论哪一种非线性问题,总是最终归结为求解一个非线性的平衡方程组,例如用位移

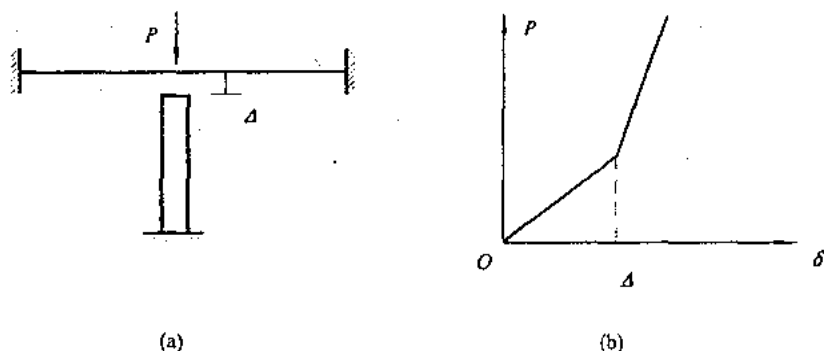


图 4.9 边界条件非线性

作未知数的有限元分析结构时,最后可得到一组总体平衡方程组:

$$[K(\delta)]\{\delta\} = \{R\} \quad (4.30)$$

式中 $\{\delta\}$ ——结点位移列阵;

$\{R\}$ ——结点荷载列阵;

$[K(\delta)]$ ——总体刚度矩阵,是结点位移的函数。

式(4.30)中的总体刚度矩阵 $[K(\delta)]$ 不同于线性问题中的总体刚度矩阵,矩阵中各元素不是常量,而是随结构的应力和位移的变化而变化的。对于材料非线性问题(最常遇到的一种非线性问题),由于联系应力与应变之间关系的物理矩阵 $[D]$ 是随应变或应力的变化而变化,不再像线弹性问题的弹性矩阵是一常数,而应力或应变的大小又与结点位移有关,实际上,物理矩阵也随结点位移大小的变化而变化,是结点位移的函数,即

$$[D] = [D(\delta)] = f(\delta) \quad (4.31)$$

由单元刚度矩阵集成的总体刚度矩阵的表达式为

$$[K(\delta)] = \sum \int [B]^T [D(\delta)] [B] dV \quad (4.32)$$

从式(4.32)中可知,总体刚度矩阵 $[K(\delta)]$ 也是结点位移 $\{\delta\}$ 的函数,于是就会产生了一个矛盾问题,要求解总体平衡方程式(4.30)以得到未知位移列阵 $\{\delta\}$,须先确定荷载列阵 $\{R\}$ 和总体刚度矩阵 $[K(\delta)]$,但总体刚度矩阵 $[K(\delta)]$ 的确定,又必须先要知道位移列阵 $\{\delta\}$,才能进而确定各单元的应变、应力和单元刚度矩阵,于是直接求解非线性方程有困难。为此,人们采用数值法求解非线性方程,在多种非线性方程求解方法中,大致可以归纳为三类:迭代法、增量法以及增量迭代混合法。

增量法是将荷载划分为许多增量,每次施加一个荷载增量。在一个荷载增量中,假定刚度矩阵是常数;在不同的荷载增量中,刚度矩阵可以有不同的数值,并与应力-应变关系相对应。迭代法在每次迭代过程中都施加全部荷载,但逐步修改位移和应变,使之满足非线性的应力-应变关系。混合法同时采用了增量法和迭代法,即荷载也划分为荷载增量,但增量的个数较少;而对每一个荷载增量,进行迭代计算。

4.2.2 材料非线性问题的迭代解法

1. 一维问题非线性方程的迭代解

通过一个简单的一维非线性问题求解过程,以便于理解迭代法的求解思路,其概念不难推广到二维和三维问题。

如图 4.10(a)所示的非线性材料直杆,其材料的应力-应变关系为已知,如图 4.10(b)所示曲线,杆端作用已知轴力为 R ,试求杆端的位移 δ 。

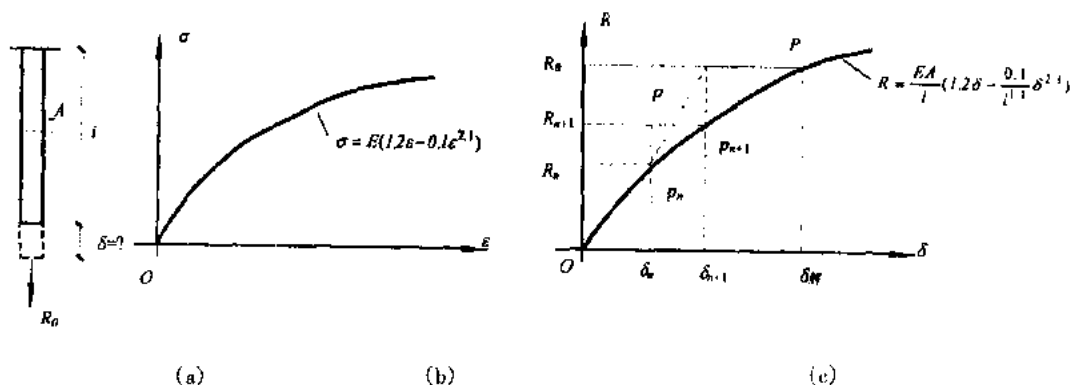


图 4.10 一维非线性问题求解

解:取杆单元的位移为线性模式,即

$$\epsilon = \frac{\delta}{l} \quad (\text{线性几何关系})$$

又

$$\sigma = E(1.2\epsilon - 0.1\epsilon^{2.1}) \quad (\text{非线性物理关系})$$

和

$$\sigma A = R_0 \quad (\text{用 } \sigma \text{ 表达为线性平衡关系})$$

从以上的前两式代入第三式,可得到关于用结点位移 δ 表示的非线性的平衡方程:

$$\frac{EA}{l} \left(1.2\delta - \frac{0.1}{l^{1.1}} \delta^{2.1} \right) = R_0 \quad (4.33a)$$

从几何上看,(4.33a)式的解是直角坐标系 (R, δ) 中曲线和直线 $R = R_0$ 的交点 P 的横坐标(见图 4.10c)

$$R = \frac{EA}{l} \left(1.2\delta - \frac{0.1}{l^{1.1}} \delta^{2.1} \right) \quad (4.33b)$$

用迭代法解以上非线性方程的具体迭代形式如下所述:

解决问题的基本思路是逐次逼近。如果找到了一个试探值 δ_n ,但不满足(4.33a)式,

如图 4.10(c)所示,它在曲线上的纵坐标值 $R_n = \frac{EA}{l} \left(1.2\delta_n - \frac{0.1}{l^{1.1}} \delta_n^{2.1} \right)$ 低于 R_0 ,用 $P_n(R_n, \delta_n)$ 表示。现在可以通过点 $P_n(R_n, \delta_n)$ 绘一条直线 p ,令直线 p 的斜率为 k_n ,则

$$k_n(\delta_{n+1} - \delta_n) = R_0 - R_n$$

这样,这条直线 p 和 $R = R_0$ 交点的横坐标 δ_{n+1} 就是一个改进了的试探解。如果按 δ_{n+1} 由(4.33b)式计算得的 R_{n+1} 仍小于 R_0 ,就可再从点 $P_{n+1}(R_{n+1}, \delta_{n+1})$ 出发,重复前面过程,又可得到进一步的改进试探解 δ_{n+2} 。这样,循环迭代,直至前后两次的位移值 δ 接近相等(在允许误差范围以内)为止。即当 $\delta_{m+1} \approx \delta_m$,或取 $R_m \approx R_0$ 时,最后得到了问题的解: $\delta = \delta_m$ 。

上述的逐次逼近过程, k_n 值可按不同方式选取, 从而有以下不同的迭代方法。可以看出, k_n 表示结点位移与结点力之间的关系, 可称做单元刚度。

(1) 常刚度迭代法

见图 4.11 所示, 用一系列平行直线与 $R = R_0$ 的交点去逼近曲线和 $R = R_0$ 的交点, 取 $k_n = k_0 = \text{常数}$, 每一次迭代的 k_n 都取第一次迭代用的常数 k_0 , $k_0 = \frac{EA}{l}$ 在物理意义上为直杆的抗拉刚度, 迭代过程中, 那些平行线族 p 的斜率, 也就是刚度始终保持不变, 所以这个过程称为常刚度迭代法。

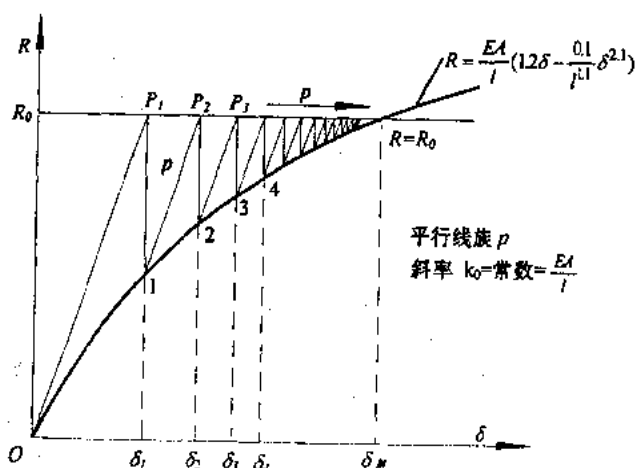


图 4.11 常刚度迭代法

当采用常刚度迭代法时, 式(a)可写为:

$$k_0(\delta_{n+1} - \delta_n) = R_0 - \left[\frac{EA}{l} \left(1.2\delta_n - \frac{0.1}{l^{1.1}} \delta_n^{2.1} \right) \right]$$

或写成迭代形式:

$$\left. \begin{aligned} k_0 \delta_{n+1} &= k_0 \delta_n + (R_0 - R_n) \\ R_n &= \frac{EA}{l} \left(1.2\delta_n - \frac{0.1}{l^{1.1}} \delta_n^{2.1} \right) \end{aligned} \right\} \quad (4.34)$$

每给一个 δ_n , 可由(4.34)式算出一个 δ_{n+1} 作为下一次的试探值。对于第一次的近似试探值, 可取 $\delta_n = 0$, 则 $R_n = 0$ 。

式中, k_0 为图 4.11 中 OP_1 直线的斜率。

在物理含义上, $k_0 = \frac{EA}{l}$ 代表直杆在弹性阶段的抗拉刚度, 于是 $R_0 = \frac{EA}{l} \cdot \delta_1$, 在上述迭代的过程中, 那些平行族 p 的斜率(即刚度), 始终保持不变, 因而这个过程称作常刚度迭代法。

(2) 割线(变)刚度迭代法

如图 4.12 所示, 在用一系列直线 p 与 $R = R_0$ 直线的交点 $P'_1, P'_2, P'_3, \dots$ 去逐次逼近 $R-\delta$ 曲线和 $R = R_0$ 直线的交点 P^* 时, 这些直线 p 的斜率并不相同。如用 k_n 代表其中某

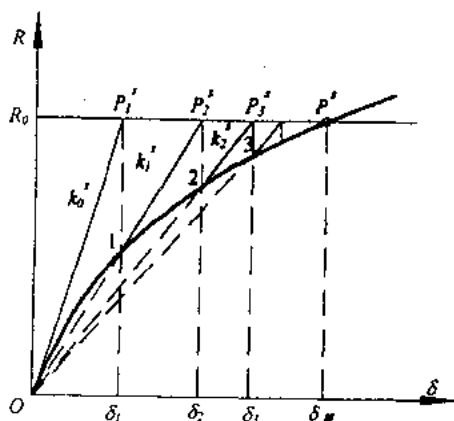


图 4.12 割线刚度迭代法

根直线 p 的斜率, 那么, 迭代中的每一步的直线方程可改写为

$$R_0 = R_n + k_n(\delta_{n+1} - \delta_n)$$

因而它与直线 $R = R_0$ 交点的横坐标, 即 $n+1$ 次近似解为

$$\delta_{n+1} = \delta_n + \frac{1}{k_n}(R_0 - R_n)$$

通常, k_n 可按两种不同方式选取。

第一种常用的变刚度迭代方式, 即所谓割

线刚度 $k_n = k_n^s = \frac{R_n}{\delta_n}$; k_n^s 在物理上代表从原点出发到所迭代时与曲线交点割线的斜率, 因而这个过程称为割线刚度迭代法。

于是, 将 k_n^s 代入前式, 可得割线刚度法的迭代公式为

$$\begin{cases} \delta_{n+1} = \frac{R_0}{k_n^s}; & k_n^s = \frac{R_n}{\delta_n} \\ R_n = \frac{EA}{l} \left(1.2\delta_n - \frac{0.1}{l^{1.1}}\delta_n^{2.1} \right) \end{cases} \quad (4.35)$$

当第一次迭代时, 取 $\delta_{n=0} = \delta_0 = 0$, $R_{n=0} = 0$, 由式(4.35)知, k_0^s 为不定值, 故第一次 k_0^s 可取任意值。

(3) 切线(变)刚度迭代法

第二种常用的变刚度迭代方式可另取

$$k_n = k_n^t = \left(\frac{dR}{d\delta} \right)_{\delta=\delta_n}$$

这时, k_n^t 在物理上代表切线刚度, 因而这个迭代过程称作切线刚度法。

将 k_n^t 代入前式, 得到切线刚度法的迭代公式:

$$\begin{cases} \delta_{n+1} = \delta_n + \frac{1}{k_n^t}(R_0 - R_n) \\ k_n^t \frac{dR}{d\delta} \Big|_{\delta=\delta_n} = \frac{EA}{l} \left(1.2 - \frac{0.21}{l^{1.1}}\delta_n^{1.1} \right) \\ R_n = \frac{EA}{l} \left(1.2\delta_n - \frac{0.1}{l^{1.1}}\delta_n^{2.1} \right) \end{cases} \quad (4.36)$$

因为 $\delta_{n=0} = 0$, $R_{n=0} = 0$, 故作为第一次的 k_0^t 可取任意值。

切线刚度法的迭代过程, 如图 4.13

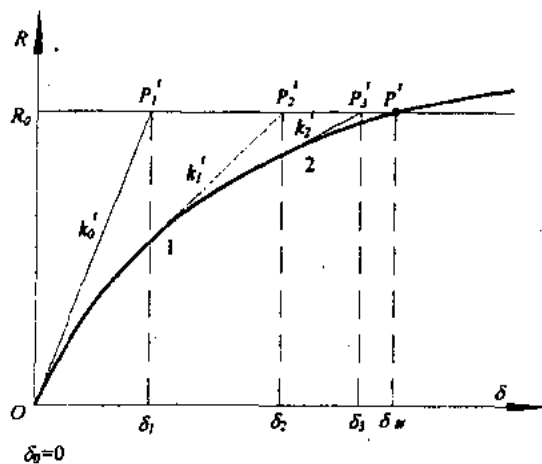


图 4.13 切线刚度迭代法

所示。

割线刚度迭代法和切线刚度迭代法统称为变刚度迭代法。

切线刚度法和前述的常刚度迭代法等价于数学上求解非线性方程的牛顿(Newton)-莱富松(Raphson)方法和修正的牛顿-莱富松法。

2. 两维问题非线性方程的几种迭代解法

(1) 常刚度迭代法

前述一维问题的非线性迭代解法,很容易地推广到两维或三维问题。如给定的应变值能够确定出相应的单元应力值时,采用常刚度迭代法比较适宜。

已知非线性的应力-应变关系为

$$\{\sigma\} = f(\epsilon) \quad (4.37)$$

引入假想的“弹性应力” $\{\sigma_0\}$,它由真实应变 $\{\epsilon\}$ 按线弹性关系计算得到,即

$$\{\sigma_0\} = [D]\{\epsilon\} = [D][B]\{\delta\}^e$$

式中 $\{\sigma_0\}$ ——假想弹性应力;

$[D]$ ——单元的弹性矩阵,仅由材料的弹性常数 E, μ 确定,为常数矩阵;

$[B]$ ——单元应变矩阵;

$\{\delta\}^e$ ——单元的结点位移列阵。

由有限元法的关系式可知:

$$\sum \int [B]^T \{\sigma_0\} dV = [k_0] \{\delta\} \quad (4.38)$$

上式中, $[k_0]$ 为线弹性问题的总刚度矩阵,由各单元刚度矩阵集合而成。 $[k_0]$ 可以看作是材料非线性问题的初始总体刚度矩阵。

将一维问题的常刚度迭代公式(4.34)中的 R_0 替换成总体荷载 $\{R\}$, k_0 由 $[k_0]$ 替换。其它记号均改为矩阵形式,可得到二维问题的非线性迭代公式。

$$\begin{aligned} [k_0] \{\delta\}_{n+1} &= [k_0] \{\delta\}_n + (\{R\} - \{R\}_n) \\ &= \{R\} + [k_0] \{\delta\}_n - \{R\}_n \\ &= \{R\} + \sum \int [B]^T (\{\sigma_0\} - \{\sigma\}_n) dV \end{aligned}$$

迭代公式即为

$$[k_0] \{\delta\}_{n+1} = \{R\} + \{\Delta R\}_n \quad (4.39a)$$

$$\{\Delta R\}_n = \sum \int [B]^T (\{\sigma_0\} - \{\sigma\}_n) dV \quad (4.39b)$$

式中, $\{\sigma\}_n$ 为第 n 次迭代时的各单元应力,由应力应变关系式(4.37)求得,记号 $\sum$ 为对各个单元求和。

常刚度迭代的步骤如下:

① 由材料的初始切线模量 E_0 ,泊松系数 μ_0 ,求得初始弹性矩阵 $[D]$,进而可得到初始总体刚度矩阵 $[k_0]$;

② 施加全部荷载 $\{R\}$,并令 $\{\Delta R\}_{n=0} = 0$,按式(4.39a)解得节点位移的第一次近似值 $\{\delta\}_1$,实际上这是线弹性问题的解;

③ 由结点位移得到单元应变, $\{\epsilon\}_1 = [B]\{\delta\}_1$,再得假想弹性应力 $\{\sigma_0\}_1 = [D]\{\epsilon\}_1$;

由非线性应力应变关系式(4.37)求得对应于 $\{\epsilon\}_1$ 的应力 $\{\sigma\}_1$;

④ 由迭代公式(4.39b)得到修正荷载项 $\{\Delta R\}$;

$$\{\Delta R\}_1 = \sum_e \int [B]^T (\{\sigma_0\}_1 - \{\sigma\}_1) dV$$

⑤ 仍采用第一次的 $[k_0]$ 作为总体刚度矩阵,已知 $\{R\}$ 和 $\{\Delta R\}_1$ 用迭代式(4.39a)得到第二次近似值 $\{\delta\}_2$;

⑥ 重复(3),(4),(5)迭代过程,直到前后两次计算值小于允许误差,即 $\{\delta\}_{n+1} - \{\delta\}_n \leq \epsilon$ (ϵ 为允许误差值),由此得到最后的解答。

需指出,有些文献在介绍求解非线性问题的迭代法中,常用的方法有初应力法和初应变法。其实这两种方法都属于常刚度迭代法,仅仅是从不同的角度给予物理意义上的解释。

(2) 割线刚度迭代法

如果材料的应力应变关系能写成如下形式:

$$\{\sigma\} = [D'_{(\epsilon)}] \{\epsilon\} \quad (4.40)$$

式中,矩阵 $[D'_{(\epsilon)}]$ 可以表达为应变的函数,则可采用割线刚度法进行迭代运算。非线性总体平衡方程式为

$$[k'] \{\delta\} = \{R\} \quad (4.41)$$

式中总体刚度矩阵 $[k'] = \sum \int [B]^T [D'] [B] dV \quad (4.42)$

总体刚度矩阵 $[k']$ 是应力和应变的函数,实际上也是结点位移的函数,随结点位移变化而变化。式(4.41)可直接用于迭代计算,因此割线刚度迭代法又称为直接迭代法,其迭代过程如下:

① 先令初始位移 $\{\delta\}_{n=0} = 0$,由初始切线模量 E_0 和泊松比 μ_0 算出物理矩阵 $[D'_{(\epsilon)}]$ 和初始总体刚度矩阵 $[k'_0]$;

② 施加全部荷载 $\{R\}$,由式(4.41)求得第一次位移近似值 $\{\delta\}_1$;

③ 由 $\{\delta\}_1$ 求出相应的应变 $\{\epsilon\}_1$ 和应力 $\{\sigma\}_1$ 以及物理矩阵 $[D'_{(\epsilon)}]$,并由式(4.42)求得相应的总体刚度矩阵 $[k']$;

④ 重复(2),(3)直至前后两次迭代计算结果(结点位移值)差值小于给定的允许误差。

(3) 切线刚度迭代法

如果材料的应力-应变关系能表示如下的增量关系

$$d\{\sigma\} = [D_t(\epsilon)] d\epsilon \quad (4.43)$$

式中, $[D_t(\epsilon)]$ 为切线刚度,是应变的函数。

可以采用切线刚度迭代法,其迭代公式如下:

$$[k_t]_n \Delta \{\delta\}_{n+1} = \{R\} - \{R\}_n \quad (4.44a)$$

$$[k_t]_n = \sum \int [B]^T [D_t(\epsilon)] [B] dV \quad (4.44b)$$

$$\{\sigma\}_n = [D_t(\epsilon)]_n [B] \{\delta\}_n \quad (4.44c)$$

$$\{R\}_n = \sum \int [B]^T \{\sigma\}_n dV \quad (4.44d)$$

$$\{\delta\}_{n+1} = \{\delta\}_n + \Delta\{\delta\}_{n+1} \quad (4.44e)$$

切线刚度迭代法的步骤如下:

① 由初始弹性常数 E_0, μ_0 求得物理矩阵 $[D_0]$, 按式(4.44b)得到初始总体刚度矩阵 $[k_0]$ 。

② 施加全部荷载 $\{R\}$, 并令 $\{\delta\}_{n=0} = 0$; $\{R\}_n = 0$ 由迭代式(4.44a)求得位移增量 $\Delta\{\delta\}_1$, 且总的位移 $\{\delta\}_1 = 0 + \Delta\{\delta\}_1 = \Delta\{\delta\}_1$, 如图 4.14(b)中 M_1 所示。并由此求得应变 $\{\epsilon\}_1$ 和应力 $\{\sigma\}_1$, 如图 4.14(a)中的 N_1 点所示;

③ 根据 σ - ϵ 非线性关系, 由 $\{\epsilon\}_1$ 求出相应的 $\{\sigma\}'_1$, 为图 4.14(a)中的 N'_1 点。并由 $\{\sigma\}'_1$ 按式(4.44d)求得与 $\{\delta\}_1$ 相对应的荷载 $\{R\}'_1$, 即

$$\{R\}'_1 = \sum \int [B]^T \{\sigma\}'_1 dV$$

④ 由 $\{\epsilon\}_1, \{\sigma\}'_1$ 求切线弹性常数 E_1, μ_1 和物理矩阵 $[D_1]$, 相当于图 4.14(a)曲线上 N'_1 点的切线斜率, 并按式(4.44b)求得 $[k_1]$, 即图 4.14(b)中的 $M_1 M_2$ 斜率;

⑤ 由迭代式(4.44a)求解 $[k_1] \Delta\{\delta\}_2 = \{R\} - \{R\}'_1$, 得 $\Delta\{\delta\}_2$ 和 $\{\delta\}_2 = \{\delta\}_1 + \Delta\{\delta\}_2$, 即得图 4.14(b)中的 M_2 点的位移值。

⑥ 重复(3),(4),(5)步骤, 直到接近真实解 δ_m , 使 $\Delta\{\delta\}_m \leq \epsilon$, ϵ 为允许误差。

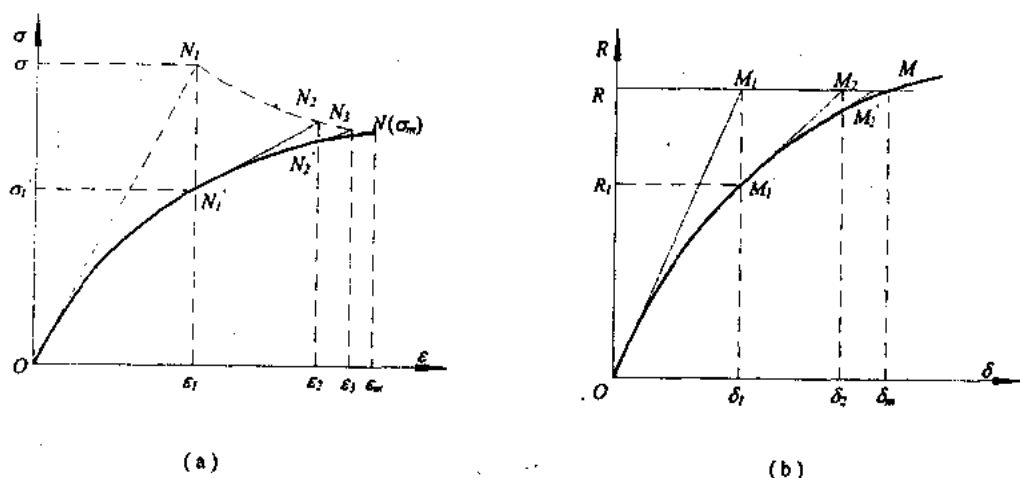


图 4.14 切线刚度迭代法

在迭代法中, 变刚度法, 特别是切线刚度法, 达到最后收敛所需的迭代次数较少。但在一般的工程计算中, 采用变刚度法每次迭代都要重新形成新的总体刚度矩阵, 并解算一次线性方程组, 耗费的计算机时较大。而常刚度法虽然收敛很慢, 但因各次迭代均采用相同的刚度矩阵, 当对刚度矩阵一次作好三角分解后, 后续的迭代计算只需要回代计算, 则每次迭代的时间比变刚度法要短得多。因此, 一般情况下, 在达到同等精度的要求下, 采用常刚度迭代法计算时间可短些。但是在求算塑性问题中, 如果进入塑性的区域很大, 有

时常刚度法收敛较慢,在这种情况下可改用切线刚度法。

4.2.3 材料非线性问题的增量解法

增量法是求解非线性问题的又一种有效方法。增量法的基本思想是将荷载划分成许多小的荷载部分(称为增量)。工程中这些荷载增量一般取成大小相等,但也可根据需要改为不相等。计算时每次只施加一级荷载增量。在施加每级增量的区间内,假设方程是线性的,即假设刚度矩阵 $[K]$ 为固定的值,但在各级荷载增量中, $[K]$ 则取不同值。每施加一级荷载增量 $\{\Delta R\}$ 可以得到一个相应的位移增量 $\{\Delta\delta\}$,累加这些位移增量可以得到任一级荷载时的总位移。增量法是用一系列线性问题去逼近非线性问题,实质上是用分段线性去代替非线性曲线。

如果结构初始状态的初荷载和初位移分别用 $\{R_0\}$ 和 $\{\delta_0\}$ 表示,(一般情况下,它们均为零矢量)。并将总的荷载分成 M 个增量,则总荷载为

$$\{R\} = \{R_0\} + \sum_{j=1}^M \{\Delta R_j\} \quad (4.45a)$$

而施加了第 i 级增量之后,此时荷载列阵为

$$\{R_i\} = \{R_0\} + \sum_{j=1}^i \{\Delta R_j\} \quad (4.45b)$$

同样,在第 i 级加载后,位移、应变和应力分别为

$$\left. \begin{aligned} \{\delta_i\} &= \{\delta_0\} + \sum_{j=1}^i \{\Delta\delta_j\} \\ \{\epsilon_i\} &= \{\epsilon_0\} + \sum_{j=1}^i \{\Delta\epsilon_j\} \\ \{\sigma_i\} &= \{\sigma_0\} + \sum_{j=1}^i \{\Delta\sigma_j\} \end{aligned} \right\} \quad (4.46)$$

具体求解时增量法又可分为始点刚度增量法、平均刚度增量法和中点刚度法三种,分述如下:

1. 始点刚度增量法

将荷载分成 M 级增量,在第 i 级荷载增量下,增量法的总体平衡方程为

$$[K_{i-1}]\{\Delta\delta_i\} = \{\Delta R_i\} \quad i = 1, 2, 3, \dots, M \quad (4.47)$$

式中,刚度矩阵 $[K_{i-1}]$ 为前级荷载终了时的切线刚度矩阵。因为在前一级荷载终了时的应力和应变已经知道,根据前一级荷载终了时的应力 $\{\sigma_{i-1}\}$ 或应变 $\{\epsilon_{i-1}\}$ 以及它们的应力应变关系可以确定物理矩阵 $[D_{i-1}]$,从而可以计算相应的刚度矩阵 $[K_{i-1}]$,此时假定在当前的第 i 级荷载增量中取刚度矩阵 $[K_{i-1}]$ 并保持不变。始点刚度增量法的实质是,在当前一级荷载增量的计算时,采用前一级末了时的刚度矩阵值。始点刚度法的计算公式见式(4.47)、(4.45b)、(4.46)。

2. 平均刚度增量法

由于在每一级荷载增量过程中,随应力和应变的变化,结构的刚度实际上是变化的。用初始刚度法计算虽然比较简单,但由于刚度矩阵在一级荷载中是不变化的,故计算误差比较大。特别是荷载增量幅度较大时更是如此。为了提高计算精度,在每级增量计算中可以采用平均的刚度值,即根据式(4.47)先计算出这一级末了时的 $\{\Delta\delta_i\}$ 及 $\{\delta_i\}$ 作为临时

值,再根据这临时的位移和应力-应变关系计算出临时的 $[k_i]$,然后将这一级初始时的刚度矩阵与这一级结束时的刚度矩阵(临时值)取平均值,当前这一级增量用于计算的平均刚度矩阵为

$$[\bar{K}_i] = \frac{1}{2}([K_{i-1}] + [K_i]) \quad (4.48)$$

于是第 i 级的位移增量计算式可改为

$$[\bar{K}_i]\{\Delta\delta_i\} = \{\Delta R_i\} \quad (4.49)$$

3. 中点刚度增量法

平均刚度增量法虽然能改善计算精度,但计算机要存贮两套刚度矩阵,占有较大的存贮量。为此,可改用中点刚度法(也称中点龙格-库塔法)如图 4.15 所示。对于第 i 级增量,首先施加荷载增量的一半,即 $\frac{1}{2}|\Delta R_i|$,在已知第 $i-1$ 级荷载增量结束时的刚度矩阵 $[K_{i-1}]$ 情况下,按下式求出临时的位移增量 $|\Delta\delta_{i-\frac{1}{2}}|$:

$$[K_{i-1}]\{\Delta\delta_{i-\frac{1}{2}}\} = \frac{1}{2}\{\Delta R_i\} \quad (4.50)$$

此时中点位移为

$$\{\delta'_{i-\frac{1}{2}}\} = \{\delta_{i-1}\} + \{\Delta\delta_{i-\frac{1}{2}}\}$$

并由位移及应力-应变关系,确定中点的刚度矩阵 $[K_{i-\frac{1}{2}}]$,以此作为当前这一级荷载增量的刚度矩阵以下式来代替式(4.47)重新求算当前第 i 级的位移增量。

$$[K_{i-\frac{1}{2}}]\{\Delta\delta_i\} = \{\Delta R_i\} \quad (4.51)$$

图 4.15 也表示了始点刚度法与中点刚度法的区别。图中 δ_i 是按刚度矩阵 $[K_{i-\frac{1}{2}}]$ 根据中点刚度法计算得到的位移值; δ'_i 是按刚度矩阵 $[K_{i-1}]$ 根据始点刚度法计算得到的位移值;显然 δ'_i 的精度差于 δ_i 。那么,将增量荷载减小到原来的一半,也就是将增量级数增加一倍,这时,用初始刚度法是否会得到与中点刚度法同样精度呢?实际上两者还是有差别。图中 δ''_i 就是荷载增量分二级后由始点刚度法算得的位移,尽管同样按刚度矩阵 $[K_{i-\frac{1}{2}}]$ 进行计算,但由于始点不同,中点刚度法由 A 点作为始点,而始点刚度法此时则从 B 点为始点,虽然精度比原来的始点刚度法有所提高,但还是没有达到中点刚度法的精度,尽管两者的计算工作量基本相同。

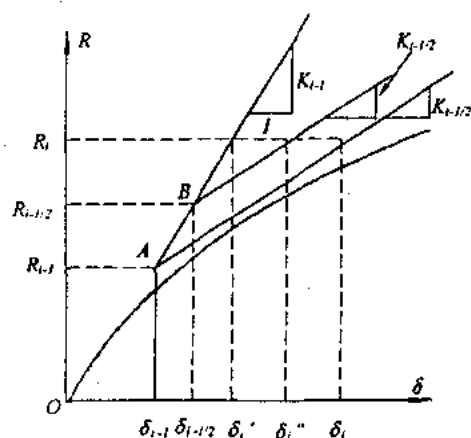


图 4.15 中点刚度法与始点刚度法比较

4.2.4 两种计算方法比较和增量迭代法

在分析非线性问题时,上述的增量法和迭代法各自具有优缺点。

增量法的主要优点是其普遍适用性。除了加工软化材料外,它能够用于几乎一切类型的非线性性态。所以它是非线性问题有限元解析中经常采用的方法。增量法的特点还在于能够比较全面地描述荷载一位移整个过程的性态,对材料性态与加载路径有关时最适用。由于增量法是将荷载逐级施加的,所以也可用于模拟施工加荷过程,能清楚地解得施工各阶段的变形和应力情况。

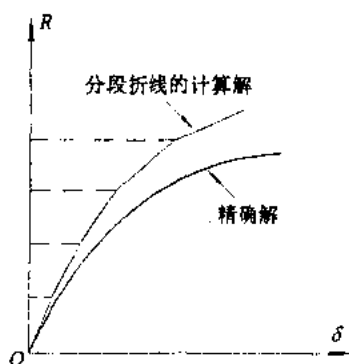


图 4.16 增量法的解与精确解的比较

但是增量法比常刚度迭代法通常要多费计算时间,这是因为对于分析中的每一步,都需要计算新的本构矩阵和单元刚度,再重新集合总刚度。增量法的另一个缺点是难以判别解答偏离真实解的程度。图 4.16 为增量解法偏离精确解的示意。由于增量法采用分段线性化,使得后一级增量计算是在前一级已产生误差的解的基础上进行。各级产生误差会累计。当增量分析的总级数少,每级荷载增量较大时,计算解的分段折线会偏离精确解曲线较远,误差将变得较大。虽然增加荷载增量的数量,减少每级增量的大小,可以达到减小误差的目的,但将会增加计算工作量。与增量法相比,迭代法在使用上较容易,较易编制电算程序和耗费较小的计算工作量;对于只需分析全量荷载作用下的结构反应是比较适合的,对于拉伸和压缩时弹性性质不同的所谓双模量材料,迭代法更为适用。此外,割线刚度迭代法,还可以成功地用于分析具有加工软化材料特性的结构,而增量法则不宜采用。

迭代法的缺点之一是不能保证它的收敛性,之二是不能用于动态问题和性态与加载路径有关的材料;缺点之三是所得到的位移、应力和应变是仅对总荷载而言,不能够描述加载过程中的性态。

增量-迭代法(又称混合法)是综合了增量法和迭代法两者的优点后的一种解法。增量-迭代法仍将荷载分成若干级增量,但荷载分级数可比增量法大为减少,即每级荷载增量的大小增加了。在每一级增量荷载作用下,又进行迭代计算,如图 4.17 所示,使得每一级增量中的计算误差可控制在很小的范围之内。虽然增量加迭代的混合算法耗用较多的计算时间。但它吸收了增量法和迭代法各自的优点,减小了两者的缺点,是分析非线性问题的非常有效方法,因此在工程中得到广泛的应用。

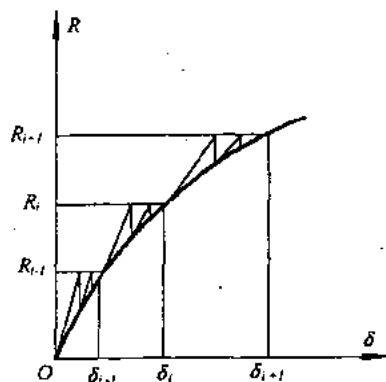


图 4.17 增量-迭代法示意图

4.3 单元开裂和屈服后的处理

钢筋混凝土有限元分析中非线性阶段需要解决下述问题:混凝土单元开裂或破坏后的处理;钢筋单元达到屈服后的处理;联结单元破坏后处理等。在处理这些问题中主要考虑结构构件在短期、单调荷载作用下的材料非线性性能。

4.3.1 混凝土单元开裂或破坏后的处理

1. 裂缝模式

混凝土的抗拉强度很低,拉应力引起开裂,混凝土裂缝对结构构件基本性能的影响是很大的,裂缝的处理始终是有限元分析的关键问题。过去已提出过几种不同的模式来处理钢筋混凝土有限元分析中的裂缝问题,各种模式的适用性随着计算机技术的不断发展,在不同的时期,有其各自的优缺点。

裂缝模式主要有:离散裂缝模式(Discrete cracking model)和片状裂缝模式(Smeared cracking model)。裂缝模式的选择应根据分析研究的目的来确定,如果只需要了解结构构件总的荷载-位移性能时,选择片状裂缝模式比较好,如果要了解构件实际的裂缝形态和详细的局部性能,则采用离散裂缝模式比较好。

(1) 离散裂缝模式

离散裂缝模式最初由 Ngo 和 Scordelis 提出的,该模式假定裂缝置于单元之间,将结点分离在裂缝两侧(图 4.18)。

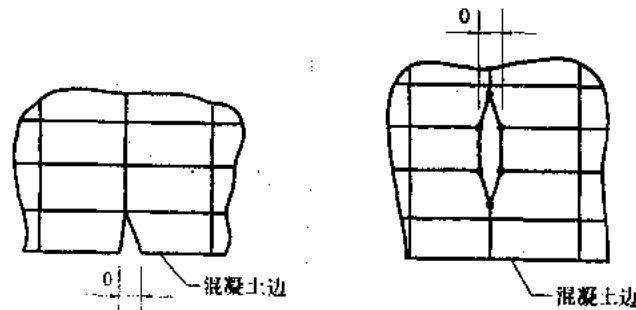


图 4.18 裂缝模式示意图

该模式可在梁中预先确定裂缝位置,并假定梁为线弹性体,裂缝沿相邻单元边界开展,在裂缝两侧单元边界各设置结点。目的是在已知裂缝位置和方向后,算出构件中混凝土应力、钢筋应力及粘结应力。以后,他们又开展了新的研究,考虑了钢箍、暗销力、咬合力及沿纵筋水平劈裂等影响。在裂缝两侧设置了联结单元。用来分析承受相同荷载的同一根梁在不同裂缝情况下的应力状态,对比不同裂缝情况下应力状态的差别,从而可知各种裂缝对内力的影响。

此模式缺点是明显的,因它只能用来研究梁内部的局部性能,不能作连续的全过程分析。

Nilson 改进了上述方法,不再用事先确定裂缝位置后按线性方法计算应力。不事先假定裂缝而按梁原始状态计算,直到相邻单元平均应力超过混凝土抗拉强度,在共同边界处开裂(图 4.18)时才停止加荷载。此时已形成新的有限元计算图式,需重新确定结点数量和相应的坐标值,在此新的有限元图式上再加载计算,如此反复直到破坏为止。

这种方法的优点在于能反映裂缝宽度及长度,并较易于表现出裂缝面之间力的传递。缺点是一旦有新的裂缝出现就要重新划分网格,即要不断改变单元形状,不能自动连续进行全过程分析计算,使计算过程很复杂,计算也很费时,所以该模式的应用曾受到一定的限制,但是随着计算机技术的发展,目前这种方法又有被推广使用的趋势。

(2) 片状裂缝模式

这种模式假定裂缝发生在单元体内部,以分布的裂缝来替代单独的裂缝,即出现裂缝后仍假定材料是连续的,仍然可用处理连续体介质力学的方法来处理,这种方法假定:某一单元内的应力(实质上是某一代表点的应力)超过了开裂的应力后,则认为整个单元开裂,在等应变三角形单元内,可认为在此单元范围内,按同一角度均呈开裂状态。即认为裂缝是分布于整个单元内部的,是微小的,彼此平行的,而且是“连续”的。这样,可以把开裂单元处理为正交异性材料。采用高次单元时,由于单元内应变是变化的,那么裂缝的延展就有问题了。为此,不同的研究者采用的处理方法不同,有些人回避这个问题,好在对裂缝的位置和长度的精确计算在大多数问题中并不十分重要。有关的处理方法可见文献[23]。

这种处理方法由于不必增加结点和重新划分单元,很容易由计算机自动进行处理,因而得到了广泛的应用。

2. 混凝土单元开裂、破坏后的处理

混凝土单元达到开裂、破坏后,应根据单元不同的受力情况作相应的处理。

(1) 单元受拉开裂的处理

当混凝土单元主拉应力或主拉应变达到开裂条件时,混凝土开裂,开裂后在垂直于裂缝方向一般假定不能再传递拉力(考虑受拉应力-应变曲线下降段为例),因此,需作相应处理。现分别按一向受拉、一向受压及双向受拉两种情况加以说明。

① 一向受拉、一向受压单元(受压方向没压坏)

单元内最大主拉应力 σ_1 或主拉应变 ϵ_1 未达开裂条件时,除了需按应力增加对单元刚度矩阵 $[K]$ 调整之外,不需作任何处理。

单元内最大主拉应力 σ_1 或主拉应变 ϵ_1 达开裂条件时,则处理如下。

在开裂的 σ_1 方向,将 σ_1 调整为零。由于单元内应力调整,单元应力失去平衡,需将不平衡应力转换成单元结点力,在整个构件或结构中重新进行分配,以达到新的平衡,也可将这些结点力与下一级荷载增量加在一起计算。

在 σ_2 方向,由于 σ_1 调整为零后,变成单向受压状态,与未开裂时的一向受拉、一向受压应力状态不同,故其压应力也需调整。即压应变 ϵ_2 仍用原来计算值,但按单向受力的应力-应变关系式求其应力。

$$\sigma_{2s} = \frac{E_0 \epsilon_2}{1 + \left[\frac{E_0}{E_s} - 2 \right] \frac{\epsilon_2}{\epsilon_c} + \left(\frac{\epsilon_2}{\epsilon_c} \right)^2} \quad (4.52)$$

将原有应力 σ_2 与调整后的应力 σ_{2u} 的差额 $\sigma_{2,ex}$ 转换成单元结点力进行重分配。

$$\sigma_{2,ex} = \sigma_2 - \sigma_{2u} \quad (4.53)$$

现以三角形单元为例说明转换力的具体计算方法。

(a) 将两部份超额应力转换到 x, y 轴上。

$$\begin{Bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{Bmatrix} = \begin{bmatrix} c^2 & s^2 \\ s^2 & c^2 \\ sc & -sc \end{bmatrix} \begin{Bmatrix} \sigma_1 \\ \sigma_{2,ex} \end{Bmatrix} \quad (4.54)$$

式中, c, s 分别是 $\cos\theta$ 及 $\sin\theta$;

$$\sigma_{1,ex} = \sigma_1;$$

$$\sigma_{2,ex} = \sigma_2 - \sigma_{2u}.$$

(b) 计算单元结点力

$$\{F\}^e = [k]\{\delta\}^e = [B]^T[D][B]t\Delta\{\delta\}^e = [B]^T\{\sigma\}_{ex}t\Delta \quad (4.55)$$

(c) 单元体除一部份应力转换成结点力之外,留下的主应力也需转换到 x, y 轴,以便与下一级荷载增量引起的应力值叠加。

$$\begin{Bmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{Bmatrix} = \begin{bmatrix} c^2 & s^2 \\ s^2 & c^2 \\ sc & -sc \end{bmatrix} \begin{Bmatrix} 0 \\ \sigma_{2u} \end{Bmatrix} \quad (4.56)$$

(d) 此时, σ_1 方向的切线模量 $E'_{1b} = 0$, 表示该方向已不再能承担拉力。 σ_2 方向的切线模量 E'_{2b} 按单向受力公式计算。用此 E'_{1b} 及 E'_{2b} 值代入弹塑性应力增量与应变增量公式中的 $[D']$, 以求出下一级荷载增量计算时的单元刚度矩阵 $[k]$ 。

② 双向受拉单元

(a) 如果最大主拉应力 σ_1 或最大主应变 ϵ_1 未达开裂条件,除调整单元刚度矩阵 $[k]$ 外,不需作任何处理。

(b) 如果最大主拉应力 σ_1 或最大主应变 ϵ_1 达到开裂条件,与上节相同,使 $\sigma_1 = 0$,并将 σ_1 转换成结点力,此时超额应力为:

$$\{\sigma\}_{ex} = \begin{Bmatrix} \sigma_{1,ex} \\ \sigma_{2,ex} \end{Bmatrix} = \begin{Bmatrix} \sigma_1 \\ 0 \end{Bmatrix}$$

代入公式(4.55)可算出结点力。单元内留下的应力在 σ_1 方向为零, σ_2 方向保持原来应力值不变。

$$\{\sigma\} = \begin{Bmatrix} \sigma_1 \\ \sigma_2 \end{Bmatrix} = \begin{Bmatrix} 0 \\ \sigma_2 \end{Bmatrix}$$

代入公式(4.56)求出转到 x, y 轴上的应力值。求单元刚度矩阵时,用 $E'_{1b} = 0$ 代入矩阵 $[D']$ 进行调整。

(c) 如果 σ_2 应力也随后达开裂条件,同样将 σ_2 转换为结点力,此时超额应力为

$$\{\sigma\}_{ex} = \begin{Bmatrix} \sigma_{1,ex} \\ \sigma_{2,ex} \end{Bmatrix} = \begin{Bmatrix} 0 \\ \sigma_2 \end{Bmatrix}$$

代入公式(4.55)求出结点力。单元内应力为零。 $E'_{1b} = E'_{2b} = 0$, 即该单元刚度矩阵 $[k] = 0$ 。

(d) 如果 σ_1 及 σ_2 同时到达开裂条件, 此时超额应力为

$$\{\sigma\}_{ex} = \begin{Bmatrix} \sigma_{1,ex} \\ \sigma_{2,ex} \end{Bmatrix} = \begin{Bmatrix} \sigma_1 \\ \sigma_2 \end{Bmatrix}$$

代入公式(4.55)求出结点力。单元内应力为零。 $E'_{1b} = E'_{2b} = 0$ 。 $[k] = 0$ 。

(e) 如果混凝土受拉时的应力应变关系采用如图 4.19 的形式, 即其中应力应变曲线的开始阶段为一直线; 但在达到抗拉强度而开裂后, 混凝土没有立即完全丧失抗拉能力, 而是逐渐减弱抗拉能力。在这一阶段的应力应变关系由 $\sigma = f_2(\epsilon)$ 表示。

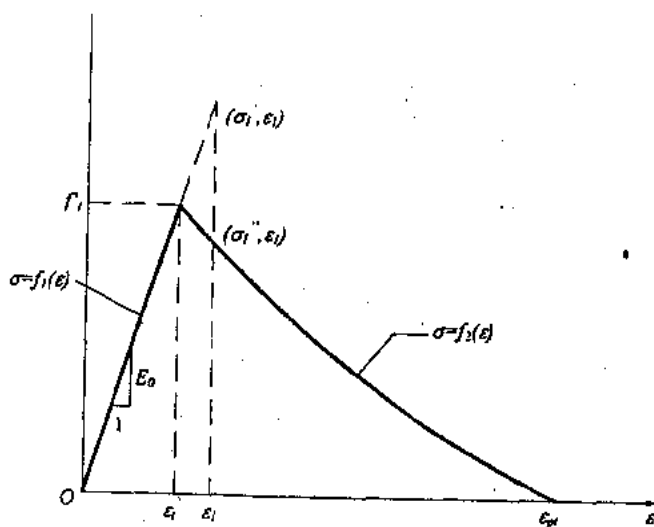


图 4.19 混凝土受拉时的应力应变关系

当在施加某一级荷载增量过程中, 主拉应力方向的主应变达到 ϵ_1 , 按弹性模量 E_0 计算的主拉应力值 $\sigma'_1 = E_0 \epsilon_1$ 超过了 f_t , 则实际上能存在于混凝土中的主拉应力应该调整为 $\sigma'_1 = f_2(\epsilon_1)$ 。其超额部分 $\sigma_{ex} = \sigma'_1 - \sigma_1$ 应转换为结点力。在重新分配结点力及以后的加载过程中, 对于 E_1 值应改用 $E_1 = f'_2(\epsilon)$, 此处 $f'_2(\epsilon)$ 为函数 $f_2(\epsilon)$ 的一次导数。

(2) 单元受压破坏后的处理

① 双向受压单元

如果最大主压应力 σ_2 或最大主压应变 ϵ_2 均未达到破坏条件, 不需作任何处理(除调整 $[k]$ 外)。

当最大主压应力 σ_2 或最大主压应变 ϵ_2 达到开裂条件时, 认为该单元完全破坏, 不能再承担任何外力, 将全部应力转换为结点力在整个构件或结构内进行重分配。

此时超额应力

$$\{\sigma\}_{ex} = \begin{Bmatrix} \sigma_{1,ex} \\ \sigma_{2,ex} \end{Bmatrix} = \begin{Bmatrix} \sigma_1 \\ \sigma_2 \end{Bmatrix}$$

代入公式(4.55)求出结点力。

单元内各应力均为零, $E'_{1b} = E'_{2b} = 0, [k] = 0$ 。

② 一向受拉、一向受压单元(受压方向压坏)

这种单元一般先是受拉方向开裂,先按本节 1.(1)那样作开裂处理,随后 σ_2 方向达受压破坏条件。此时认为该单元全部破坏,不能再承受任何外力。将单元内应力转换成结点力进行重分配。

单元超额应力为

$$\{\sigma\}_{ex} = \begin{Bmatrix} \sigma_{1,ex} \\ \sigma_{2,ex} \end{Bmatrix} = \begin{Bmatrix} 0 \\ \sigma_2 \end{Bmatrix}$$

代入公式(4.55),求出结点力。

单元内各应力均为零。 $E'_{1b} = E'_{2b} = 0, [k] = 0$ 。

4.3.2 钢筋单元达到屈服条件后的处理

在计算过程中,当钢筋单元应力超过屈服条件时,需根据钢筋采用的单元形式和钢筋的应力-应变关系作出相应调整。

1. 线形单元

为了计算方便,钢筋一般采用线形单元,因而不存在复合受力问题(忽略了钢筋受剪)。可以按照具体应力状态所处的直线段而采用相应的 E 值,在水平段上, $E = 0$ 。

如果在施加某一级荷载增量过程中,应力状态由一直线段转入另一直线段时,则需要进行如下处理。

对于线形单元,当钢筋应力 σ_s 超过屈服应力 f_y 时,需把超过屈服应力的部分 $\sigma_{ex} = \sigma_s - f_y$ 转化为结点力 F^e 。

$$F^e = \sigma_{ex} \cdot A_s = (\sigma_s - f_y) \cdot A_s \quad (4.57)$$

此时单元应力为 f_y , 单元刚度为 $E = 0$ 。

如钢筋采用图 4.20(b)所示应力-应变关系,除了需控制钢筋应力外,还需控制钢筋应变。当钢筋应力 σ_s 超过屈服应力 f_y 及钢筋应变 $\epsilon_1 \leq \epsilon \leq \epsilon_2$ 时,可与图 4.20(a)的情况同样处理,一旦钢筋应变 $\epsilon > \epsilon_2$,弹性模量就需取用相应强化段的 E 值。

2. 三角形单元

假定钢筋应力-应变关系取用图 4.20(a)模式,并采用塑性模式 Mises 屈服条件,在每级荷载增量施加后,求出等效应力 $\bar{\sigma}$ 。

$$\begin{aligned} \bar{\sigma} &= \frac{1}{\sqrt{2}} \sqrt{(\sigma_x - \sigma_y)^2 + (\sigma_y - \sigma_z)^2 + (\sigma_z - \sigma_x)^2 + 6(\tau_{xy}^2 + \tau_{yz}^2 + \tau_{zx}^2)} \\ &= \frac{1}{\sqrt{2}} \sqrt{(\sigma_1 - \sigma_2)^2 + (\sigma_2 - \sigma_3)^2 + (\sigma_3 - \sigma_1)^2} \end{aligned} \quad (4.58)$$

如果 $\bar{\sigma} < f_y$, 则单元未屈服,不需作处理,

如果 $\bar{\sigma} \geq f_y$, 需作如下处理。

(1) 调整单元应力,按比例 $\frac{f_y}{\bar{\sigma}}$ 缩小;

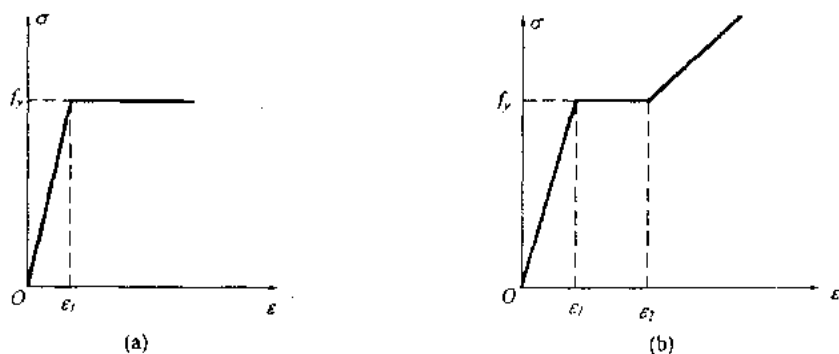


图 4.20 钢筋的应力-应变关系

(2) 调整单元刚度,用弹塑性矩阵 $[D]_{ep} = 0$,相应于单元刚度矩阵 $[k] = 0$,在后续荷载增量作用下,应力不再增大;

(3) 将超额应力 $\{\sigma\}_{ex} = \left(1 - \frac{f_y}{\bar{\sigma}}\right)\{\sigma\}$ 代入公式(4.55)求出结点力。

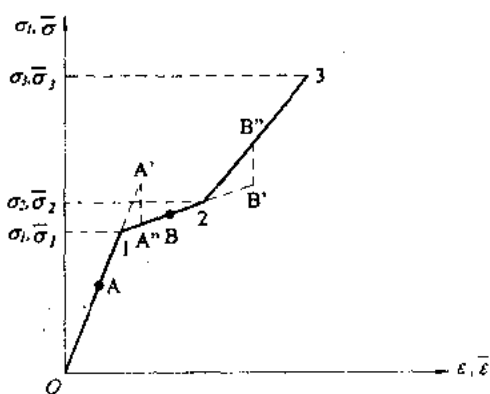


图 4.21 钢筋应力-应变关系

假定钢筋应力-应变关系取用图 4.21 模式,为避免由于 $E = 0$ 而使矩阵变成奇异,屈服台阶一段为平坦的倾斜段,屈服条件仍取用 Mises 条件。

每加一级荷载增量后,用公式(4.58)求出 $\bar{\sigma}$,如果 $\bar{\sigma} < \bar{\sigma}_1$ 不需处理,即可加下一级荷载增量。如果 $\bar{\sigma} \geq \bar{\sigma}_1$,则钢筋屈服,需作如下处理:

(1) 调整单元刚度,用平坦倾斜段(1-2)的弹塑性矩阵 $[D]_{ep:1-2}$ 代入刚度矩阵公式中的 $[D]$,可求出单元刚度矩阵 $[k]_1$ 。

(2) 调整应力增量及应力总量

过渡区 A—A' 的处理:引用应变强度 $\bar{\epsilon}$ 及比例系数 m 。

$$\bar{\epsilon} = \frac{\sqrt{2}}{2(1+\nu)} \sqrt{(\epsilon_x - \epsilon_y)^2 + (\epsilon_y - \epsilon_z)^2 + (\epsilon_z - \epsilon_x)^2 + \frac{3}{2}(\gamma_{xy}^2 + \gamma_{yz}^2 + \gamma_{zx}^2)} \quad (4.59)$$

$$m = \frac{\bar{\epsilon}_1 - \bar{\epsilon}_A}{\bar{\epsilon}_{A'} - \bar{\epsilon}_A} \quad (4.60)$$

式中 m ——比例系数;

$\bar{\epsilon}_1$ ——刚达到屈服应力时相应的应变强度;

$\bar{\epsilon}_A$ ——前一级荷载增量中,按弹性常数 $[D]_e$ 求出的 A 点相应的应变强度;

$\bar{\epsilon}_{A'}$ ——A' 点的垂直线与平坦倾斜段交点处相应的应变强度。

$$[D]_{ep:A-A'} = m[D]_e + (1-m)[D]_{ep:A'} \quad (4.61)$$

式中, $[D]_{ep:A'}$ 为相应于在平坦倾斜段上点 A' 的弹塑性矩阵。

于是,可用下式求出应力增量。

$$\{d\sigma\} = [D]_{ep:A-A'} \{d\epsilon\}$$

再求出应力总量。

$$\{\sigma\} = \{\sigma\}_{II} + \{d\sigma\}$$

(3) 将相应于图 4.21 中 $A'A''$ 一段的超额应力转换成结点力。

$$\{\bar{\sigma}\}_{ex} = [[D]_e - [D]_{ep:A-A'}] \{d\epsilon\} \quad (4.62)$$

以此代入公式(4.55)求出结点力。

当 $\bar{\sigma}$ 进入 1—2 区段后,每加一级荷载增量,用上一级求得的弹塑性矩阵 $[D]_{ep}$ 求应力增量 $\{d\sigma\}$ 及等效应力 $\bar{\sigma}$,如 $\bar{\sigma} < \sigma_2$,即可加下一级荷载增量,不需作任何处理。如果 $\bar{\sigma} \geq \sigma_2$,说明已进入 2—3 区段,就作如下处理。

(1) 调整单元刚度,用 $[D]_{ep:2-3}$ 求出单元刚度矩阵 $[k]$ 。

(2) 调整应力增量及应力总量。

过渡区处理:引用比例系数 m 。

$$m = \frac{\bar{\epsilon}_2 - \bar{\epsilon}_H}{\bar{\epsilon}_{B'} - \bar{\epsilon}_B} \quad (4.63)$$

弹塑性矩阵可用下式求出

$$[D]_{ep:B-B'} = m[D]_{ep:B} + (1-m)[D]_{ep:B'}$$

用下式求出应力增量

$$\{d\sigma\} = [D]_{ep:B-B'} \{d\epsilon\}$$

用下式求出应力总量

$$\{\sigma\} = \{\sigma\}_{II} + \{d\sigma\}$$

进入 2—3 区段后,即用 $[D]_{ep:2-3}$ 继续进行计算。

4.3.3 联结单元破坏后的处理

1. 联结单元的非线性刚度

联结单元可用来模拟单元之间的粘结力和咬合力,其刚度是非线性的,它随着单元间的相对滑移量或裂缝宽度而变化。刚度矩阵中有两个弹簧刚度,一个是平行于钢筋方向的弹簧刚度 k_h ,另一个是垂直于钢筋方向的弹簧刚度 k_v 。由于模拟咬合力的研究不太成熟,而且采用联结单元并非是考虑裂缝上咬合作用的唯一方法,所以下面仅介绍模拟粘结力联结单元的刚度。

模拟粘结力联结单元的弹簧刚度 k_h 与 k_v 的数值受到钢筋表面性质、直径和间距;混凝土的品种、强度等许多因素的影响。 k_h 主要依赖试验结果,而 k_v 的取值到目前为止还缺乏研究。

(1) 模拟粘结力联结单元的弹簧刚度 k_h

有关 k_h 的计算,不同的研究者根据各自的试验所得的 τ - s 关系表达式,通过微分导出不同计算公式,目前国内外提出的有以下几种计算式。

① Nilson 公式

弹簧刚度 k_h 为

$$k_h = \frac{d\tau}{ds} A = (9.8 \times 10^2 - 11.44 \times 10^4 s + 25.05 \times 10^5 s^2) A \quad (4.64)$$

式中, A 为从属于一个弹簧的钢筋表面积;

k_h 的单位为 N/mm 。

② Houde 和 Mirza 公式

$$k_h = \frac{d\tau}{ds} A = (529 - 50.4 \times 10^3 s + 17.5 \times 10^5 s^2 - 2.19 \times 10^7 s^3) \sqrt{\frac{f_c}{42.3}} A \quad (4.65)$$

Nilson 所建议的公式没有考虑混凝土强度的影响, 而 Houde 和 Mirza 公式则考虑了这一点。

③ 东南大学的公式

东南大学根据所做的试验, 提出了一个 τ - s 的计算式。

$$\tau = 6.58 \times 10^2 s - 2.12 \times 10^4 s^2 + 2.15 \times 10^5 s^3$$

$$k_h = \frac{d\tau}{ds} A = (6.58 \times 10^2 - 4.24 \times 10^4 s + 6.45 \times 10^5 s^2) A \quad (4.66)$$

(2) 模拟粘结力联结单元 k_v

k_v 的取值较复杂, 文献[4]介绍: 在实际梁中, 当裂缝扩展较大时, 垂直于纵向钢筋的粘结力与钢筋的锚固作用和梁的向下挤压作用均有很大关系。在此情况下, 粘结力不仅与化学性的粘结力和钢筋与混凝土之间的机械咬合力有关, 而且还与钢筋四周的混凝土是否能很好握住钢筋有关。实际上是一个三向受力问题。

当裂缝宽度不大, 或在裂缝两侧附近有钢箍时, 向下挤压作用较小, 还不会使混凝土发生水平劈裂, 此时钢筋在垂直方向的移动很小, 甚至可以忽略不计, 可认为混凝土和钢筋在垂直方向是刚接的, 于是可假定 k_v 为与混凝土受拉弹性模量成正比的一个很大值。

$$k_v = \frac{Eb_n l}{b} \quad (4.67)$$

式中 E ——混凝土受拉弹性模量;

b_n ——梁在钢筋高度处的净宽;

b ——梁宽;

l ——联结单元沿钢筋纵向的间距。

当然, 以上处理仅是一种近似方法。

2. 联结单元破坏后的处理

Nilson 等都认为, 在具体处理中, 对邻近裂缝处与离裂缝较远处的弹簧应按外部弹簧和内部弹簧两种不同情况来处理。一般可以取离裂缝一倍钢筋直径处作为外部弹簧和内部弹簧的分界线。

(1) 外部弹簧

根据试验可知, 在邻近裂缝处的外部弹簧, 在粘结力达到最大值后, 将产生粘结破坏, 在进一步滑移时, 粘结力取为零。

当平行于钢筋方向的弹簧应变 $|\epsilon_h| < \epsilon_{h\max} (=0.003)$, 不需作处理。

当 $|\epsilon_h| \geq \epsilon_{h,\max}$ 时,需作如下处理:

① 将水平向应力 σ_h 全部作为超额应力,转换成结点力

$$\{F\}^e = [B]^T \{\sigma\}_{ex} = \begin{bmatrix} -c & s \\ -s & -c \\ c & -s \\ s & c \end{bmatrix} \begin{Bmatrix} \sigma_h \\ 0 \end{Bmatrix} \quad (4.68)$$

式中, c 及 s 为 $\cos\theta$ 及 $\sin\theta$

② 调整单元本身内力为 $\sigma_h = 0, \quad \sigma_v = \sigma_v$

③ 调整刚度,按 $k_h = 0$ 代入联结单元刚度矩阵公式,求出新的刚度矩阵 $[k]$

(4) 内部弹簧

根据试验资料可知,当粘结力达到最大值后,内部弹簧不会发生破坏,如果再进一步发生滑移,粘结力也将继续保持原来峰值。

当 $|\epsilon_h| < \epsilon_{h,\max}$ 时,不需作处理,

当 $|\epsilon_h| \geq \epsilon_{h,\max}$ 时,需作如下处理。

① 将超额应力转换成结点力。

$$\sigma_{hex} = k_h (\epsilon_h - 0.003) \quad (4.69)$$

$$\{F\}^e = \begin{bmatrix} -c & s \\ -s & -c \\ c & -s \\ s & c \end{bmatrix} \begin{Bmatrix} \sigma_{hex} \\ 0 \end{Bmatrix} \quad (4.70)$$

② 调整单元本身内力为

$$\sigma_h = k_h \times 0.003$$

$$\sigma_v = \sigma_v$$

③ 调整刚度,按 $k_h = 0$, 求出新的刚度矩阵 $[k]$ 。

4.4 结构进入负刚度后的处理

4.4.1 概述

负刚度情况在钢筋混凝土结构分析中会经常遇到,混凝土材料在达到抗拉或抗压极限强度后,其 $\sigma-\epsilon$ 曲线有一下降段或称软化段,如图 4.22(a)。大多数钢筋混凝土结构在达到最大承载力后也存在有下降段,或称负刚度阶段,如图 4.22(b)所示。在有限元分析中,对于局部混凝土开裂或压屈后下降段的处理方法通常是将原有应力逐级分段下调,如图 4.23 所示,将下调应力作为超额应力并转化为结点力并入下一计算步骤予以释放。对于结构在负刚度情况下的分析,各国学者提出了许多种处理方法^[10],常见的有:位移控制法、附加虚拟弹簧法、强制迭代法、弧长法等,这些处理方法已在许多教材中作了较详细地论述,本书不再赘述。本书仅这些方法的运用及适用范围加以说明。

1. 位移控制法

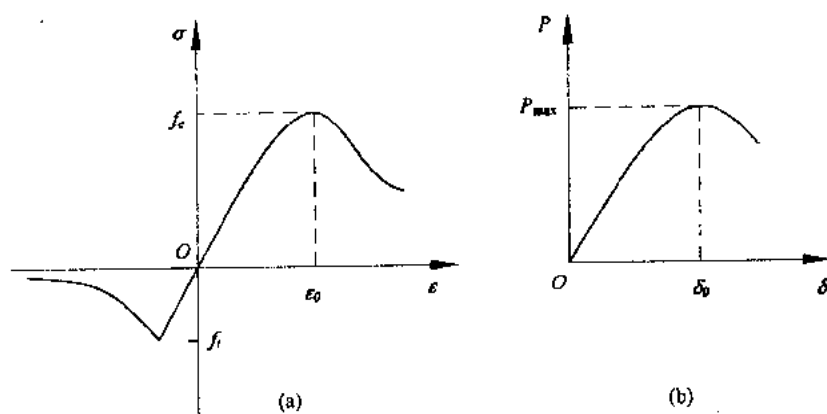


图 4.22 应力-应变、受力-变形曲线

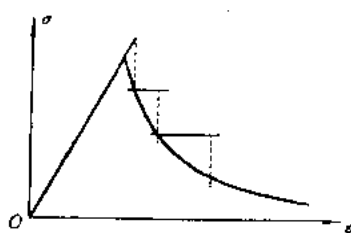


图 4.23 逐级卸载模式

位移控制法相对比较容易得到负刚度下的结构受力-变形关系,第五章中许多构件的受力-变形全过程分析均采用这一方法求得。位移控制法用于单自由度求解比较容易,而用于多自由度、多构件的结构分析比较困难。在一般情况下所需了解的是一定外力作用下的结构变形及性能,而不是通过已知位移来研究结构的性能。

2. 加虚拟弹簧法

如图 4.24 所示,这一处理方法的步骤在以往的文献中是这样进行的:在适当的部位加上虚拟弹簧,弹簧在全部变形范围内都呈线性,图 4.24 (a)中的曲线②。加弹簧后结构的 $P-\delta$ 曲线如图 4.24 中曲线③,该曲线已无负刚度问题,由曲线③、②相减可得曲线①,即钢筋混凝土构件的 $P-\delta$ 曲线。

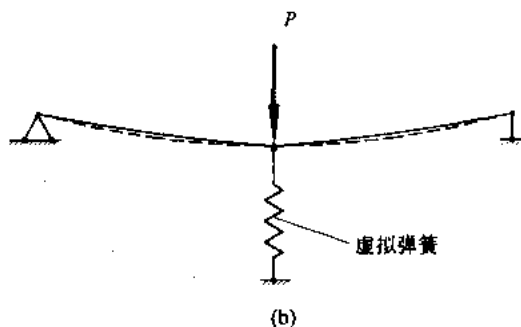
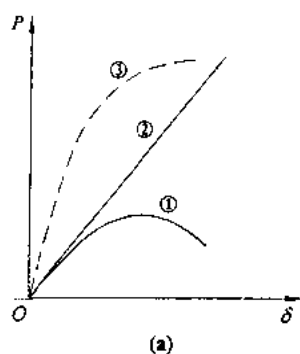


图 4.24 虚拟弹簧法示意图

然而在实际计算过程中,问题并不如此简单。图 4.24(b)中的梁在达到极限承载力后,其进入负刚度的部位仅是跨中部位的一段,而梁其它各段将沿图 4.25 中的 1 路径卸载,梁将产生图 4.24(b)中所示的变形(虚线至实线),而变形的多少则需根据平衡条件、边界条件另作计算。

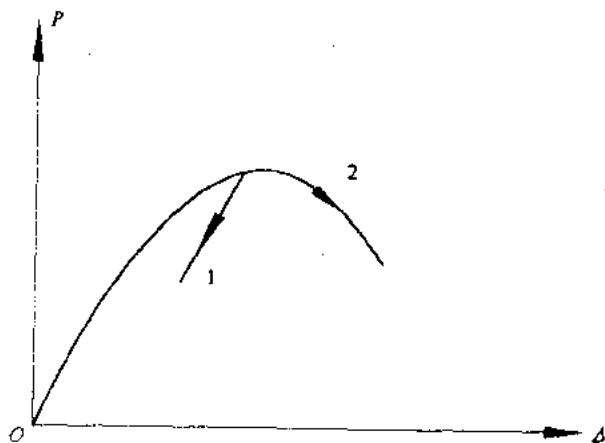


图 4.25 不同卸载路径广义力和广义变形图

3. 强制迭代法和弧长法

在强制迭代法和弧长法中一般仅考虑图 4.25 中路径 2 的情况,而许多结构在达到极限承载力后,一部分构件或一部分构件区域,例如图 4.24(b)梁跨中以外的区段,要沿路径 1 卸载,而在路径 1 方向上的受力-变形关系则无法用迭代法或弧长法求得。

4.4.2 结构负刚度情况下荷载增量法的运用及适用条件

根据非线性方程的各种计算方法的特点可以看出,对于有卸载情况或反复加载下的结构的受力分析,较有效的求解方法应是荷载增量法。增量法的特点在于分段线性化,将整个计算分析过程的应力-应变过程分成若干增量段,取每一增量段的切线刚度或校正切线刚度代替不断变化的结构刚度,逐段计算并累加,直到求得最终结果。由于增量法求解过程中,每一步均将结构视作线性,并作一次独立的加荷过程,因此,结构整体需处于正刚度状态,若结构在某阶段转化成负刚度或零刚度,那么这一阶段的增量法的受力-位移模型就找不到能量平衡点,即无法求得实际解。因此,增量法只适用于结构达到极限状态前的求解。对处于极限状态后的构件计算(如图 4.26(a)所示),则可采用上述加附加虚拟弹簧或附加虚拟单元的方法,使整体结构刚度为正(如图 4.26(b)所示),则增量法继续有效。但这一处理方法是有限制条件的,图 4.24(b)中附加虚拟弹簧的结构体系是不能用增量法来进行求解的,因为结构部分区域进入负刚度后,该区域的弹性势能无处释放,即找不到能量平衡点(从静力学角度看)。

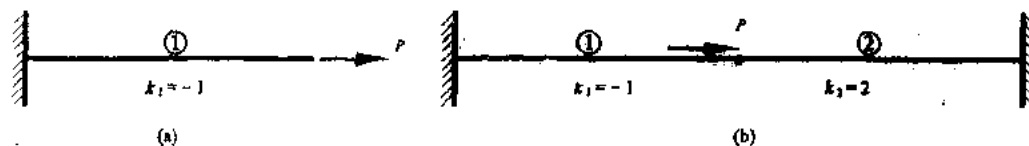


图 4.26 加虚拟单元算法

根据增量法求解其实质是寻找能量平衡点这一原理,增量法的可解性以及所得解的

正确性就可根据能量法的极值原理来进行判别。

根据单元势能公式为

$$\Pi^e = \int_e W(\{\epsilon\}) dV - \{u\}^T \{p\} \quad (4.71)$$

$$W(\{\epsilon\}) = \int \{\sigma\}^T d\epsilon \quad (4.72)$$

由切线刚度得应力-应变关系

$$\{\sigma\} = \int [D]_T d\epsilon \quad (4.73)$$

将式(4.72)、(4.73)代入式(4.71)得

$$\Pi^e = \int_e \left(\iint \{d\epsilon\}^T [D]_T \{d\epsilon\} \right) dV - \{u\}^T \{p\} \quad (4.74)$$

由单元应变与结点位移关系

$$\{\epsilon\} = [B] \{u\}^e; \quad \{d\epsilon\} = [B] \{du\}^e$$

代入式(4.74)

$$\Pi^e = \int_e \left(\iint \{du\}^e{}^T [B]^T [D]_T [B] \{du\}^e \right) dV - \{u\}^T \{p\} \quad (4.75)$$

又由切线刚度矩阵

$$[k]_T^e = \int_e [B]^T [D]_T [B] dV$$

则

$$\Pi^e = \iint \{du\}^e{}^T [k]_T^e \{du\}^e - \{u\}^T \{p\} \quad (4.76)$$

总体势能为单元势能之和,即

$$\Pi = \sum \Pi^e \quad (4.77)$$

将单元势能(4.76)按单元结点和总体结点对应关系代入上式,可得

$$\Pi = \iint \{dU\}^T [k]_T \{dU\} - \{U\}^T \{p\} \quad (4.78)$$

根据最小势能原理,有

$$\delta \Pi = 0 \quad (4.79)$$

总体势能增量

$$\Delta \Pi = \frac{1}{2} \delta^2 \Pi \quad (4.80)$$

$\delta^2 \Pi > 0$ Π 为极小,结构为稳定平衡状态,未进入负刚度阶段。

$\delta^2 \Pi < 0$ Π 为极大,为不稳定平衡状态,结构已过承载力极值点,进入负刚度阶段。

$\delta \Pi^2 = 0$ Π 为驻值,临界平衡状态。

由式(4.78)和(4.79)

$$\delta \Pi = \{\delta U\}^T \int [k]_T \{dU\} - \{\delta U\}^T \{p\} = 0 \quad (4.81)$$

$$\int [K]_T \{dU\} - \{P\} = 0$$

$$\begin{aligned}\{dP\} &= [K]_T \{dU\} \\ \{\Delta P\} &= [K]_T \{\Delta U\}\end{aligned}\quad (4.82)$$

式(4.82)即为增量法求解公式

$$\begin{aligned}\delta^2\Pi &= \{\delta^2 U\}^T \int [K]_T \{dU\} + \{\delta U\}^T [K]_T \{\delta U\} - \{\delta^2 U\}^T \{P\} \\ &= \{\delta^2 U\} \left(\int [K]_T \{dU\} - \{P\} \right) + \{\delta U\}^T [K]_T \{\delta U\} \\ &= \{\delta U\}^T [K]_T \{\delta U\}\end{aligned}\quad (4.83)$$

根据上述能量准则

$$\begin{aligned}\{\delta U\}^T [K]_T \{\delta U\} &> 0 \quad \text{稳定平衡状态;} \\ \{\delta U\}^T [K]_T \{\delta U\} &< 0 \quad \text{不稳定平衡状态;} \\ \{\delta U\}^T [K]_T \{\delta U\} &= 0 \quad \text{临界平衡状态.}\end{aligned}$$

可见,结构是否进入负刚度,最终取决于切线刚度矩阵 $[K]_T$ 和位移向量 $\{\Delta U\}$;当 $[K]_T$ 正定时, $\{\delta U\}^T [K]_T \{\delta U\} > 0$,结构未进入负刚度阶段,用增量法可解;当 $[K]_T$ 负定时,结构进入负刚度阶段,增量法求得非实际解;当 $[K]_T$ 奇异,即 $\det[K]_T = 0$,结构处于临界平衡状态,增量法方程(4.82)无解。 $[K]_T$ 由正定过渡到负定,即结构由正刚度发展到负刚度的过程之间,由于局部负刚度的出现, $[K]_T$ 存在有一个不定阶段,增量法能否求得实际解取决于 ΔP 向量。当由增量方程(4.82)求得的位移向量增量 ΔU ,使 $\{\Delta U\}^T [K]_T \{\Delta U\} > 0$,说明结构处于稳定平衡状态,增量法可求得实际解;若 $\{\Delta U\}^T [K]_T \{\Delta U\} \leq 0$,则增量法求不到实际解。

4.4.3 结构负刚度状态下增量法求解实例

钢筋混凝土受拉杆件(如图 4.27)所示,混凝土在受拉开裂后,主裂缝之间的混凝土仍能承担一部分拉应力,这一效应称之为“拉伸强化”(Tension Stiffening),混凝土平均受拉应力-应变关系为图 4.23 形式。由于有的受拉构件的配筋率较小,在混凝土开裂后,构件会有一负刚度阶段产生(图 4.28 中 A~B 点),在这一阶段混凝土将原有的承载力逐步释放并传递给钢筋,这一过程过去一般采用图 4.23 逐步卸载模式或阶段下降模式进行,这里直接将混凝土应力-应变的负刚度组合进单元整体刚度中,并附加两个虚拟单元(图 4.27),以确保增量法计算过程中结构的整体刚度为正。图中的有限元模型采用四结点等参单元,钢筋为分布模式单元。

计算结果如图 4.28 所示,图中 0~A 为混凝土开裂前阶段,构件呈一定的线性刚度,混凝土开裂后构件刚度产生突变,刚度急剧减弱,并进入负刚度,随着变形加大,混凝土应力逐步释放,钢筋应力不断增加,整个受拉杆件刚度趋近于单个钢筋的受拉刚度。图 4.28 中的计算结果较好地反映了钢筋混凝土受拉构件这种受力-变形关系机理,同实际试验情况相一致。

4.4.4 结语

结构负刚度是结构受力-变形分析的一个特殊阶段,也是一个重要阶段,结构极限承

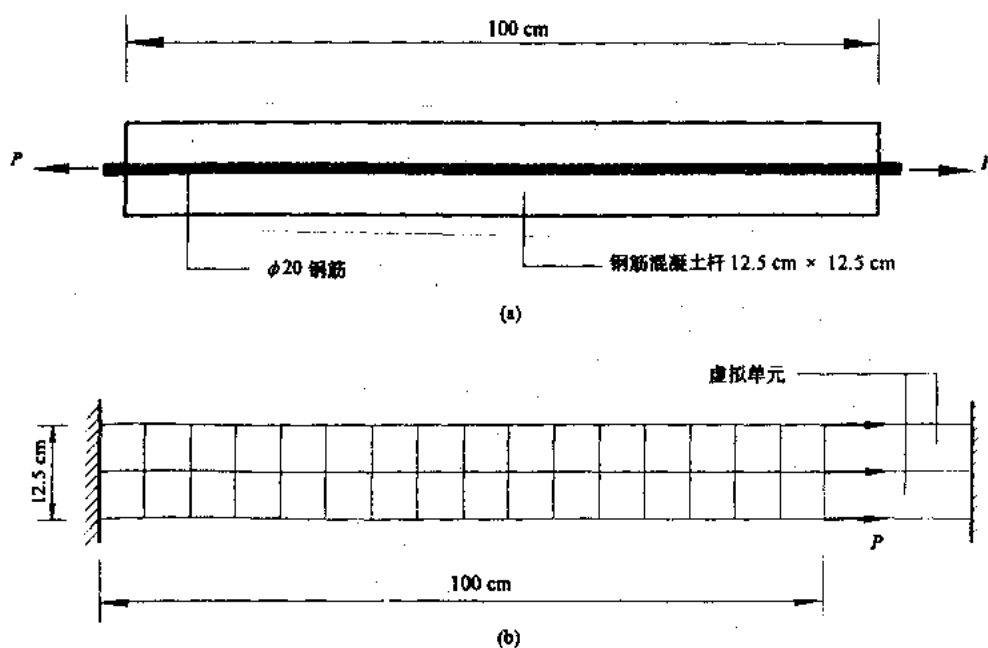


图 4.27 受拉杆及单元计算模型

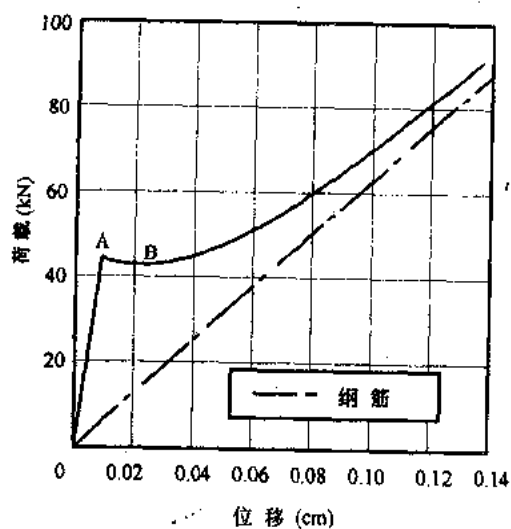


图 4.28 拉杆受力-变形曲线

载下的受力分析、应力重分布过程等均会遇到这一阶段,它是反映结构性能的重要指标之一。但这一阶段也是计算分析中较难处理的部分。目前所提出的结构负刚度的求解模型虽能解决部分问题,但也受到许多条件的限制,全面有效的结构负刚度状态下的求解方法还有待进一步研究。

第五章 钢筋混凝土构件有限元分析

本章论述钢筋混凝土构件的有限元分析,包括梁在弯矩作用下及弯矩和剪力共同作用下的强度计算和变形计算,柱在弯矩和轴力作用下的强度计算和变形计算。

5.1 按杆系结构进行梁的有限元分析

通过平截面假定和钢筋、混凝土材料应力-应变关系可求得构件正截面弯矩-曲率($M-\varphi$)关系曲线,有了 $M-\varphi$ 关系以后,在进行梁受力变形全过程分析时,将梁视为包含着若干微段的变刚度杆单元,并假定在结点之间的每一小段内各截面处的曲率是线性变化的(图 5.1)。

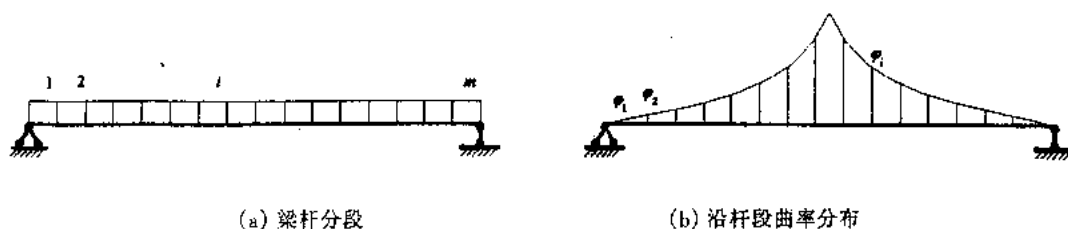


图 5.1 梁杆分段及杆段曲率分布

这样,利用虚功原理和共轭梁法,可计算出任意截面的转角 θ_i 和 δ_i ,采用分级加载或分级加位移的方法,就能获得任意一级加载或加位移后的位移和相应的内力,从而达到对梁进行全过程分析目的。

5.1.1 计算假定

1. 平截面假定

梁正截面变形后仍保持平面,截面应变为直线分布,不考虑钢筋与混凝土之间的相对位移。从理论上来讲平截面假定仅适用于跨高比较大的连续均质弹性材料的构件。对由混凝土及钢筋组成的构件,由于材料的非均质性,以及混凝土开裂,特别是在纵筋屈服,受压区高度减小而临近破坏的阶段,在开裂截面上的平截面假定已不能适用。但是,考虑到构件破坏是产生在某一区段长度内的,而且试验结果表明,只要应变量测标距有一定长度,量测的截面平均应变值从加荷开始直到构件破坏,都能较好地符合平截面假定^[4]。

2. 钢筋的应力-应变关系

受拉或受压钢筋可采用第二章中所介绍的钢筋的应力-应变关系。应力-应变关系的不同将对计算结果产生一定的影响,在研究工作和实际工程计算中应采用根据试验结果获得的参数。

3. 混凝土应力-应变关系

混凝土应力-应变关系式选用第二章中所述的单轴作用下的混凝土本构关系模型,且在混凝土的受压区带有下降段。

由于受压区沿截面高度的应变梯度的影响,每一截面高度的应力-应变关系是不相同的^[4]。尽管如此,考虑到试验实测的数据不多,还不足以建立反映其规律的表达式,因此,暂且假定各层混凝土仍遵循同一的应力-应变关系。

4. 不考虑剪切变形的影响。

5.1.2 截面分层

图 5.2 所示为矩形截面钢筋混凝土构件,在正截面受力作用下,截面的应变、应力分布。为了能进行数值计算,将混凝土的截面分成有限条带,并假定每一条带上的应力 $\sigma_{c,i}$ 均匀分布,混凝土和钢筋的应力均以压力为负,拉力为正。

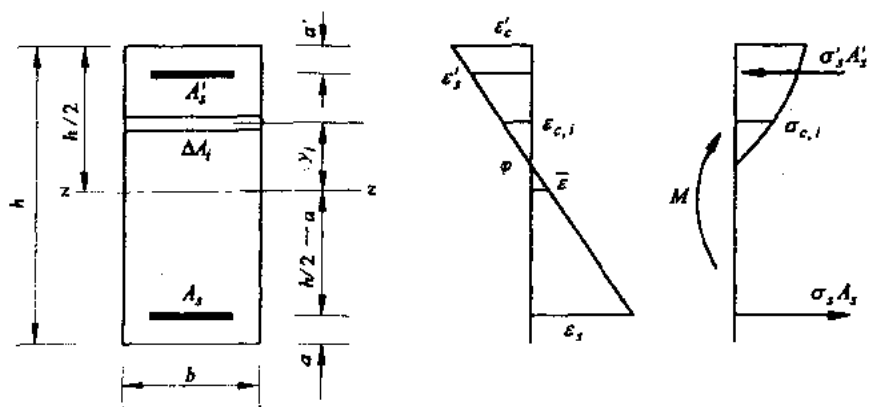


图 5.2 矩形截面梁应变和应力图

在条带划分时,为了能使计算模型中截面上的应力分布接近实际情况,一般采用多条带划分。

根据平截面假定可得截面曲率为

$$\varphi = \frac{\epsilon'_c + \epsilon_s}{h_0} \quad (5.1)$$

式中 ϵ'_c ——截面受压区边缘混凝土应变;

ϵ_s ——受拉钢筋应变;

h_0 ——截面有效高度;

φ ——截面曲率。

截面上任意一条条带的应变为 $\epsilon_{c,i}$

$$\epsilon_{c,i} = \varphi y_i \quad (5.2)$$

式中, y_i 为任意一条带的高度中心距截面中心轴($z-z$)的距离(截面中心轴以上为正,以下为负)。

按已知的混凝土和钢筋的应力-应变关系,可得截面上任一条带的混凝土和钢筋的应

力 $\sigma_{c,i}$, σ'_s 及 σ_s 。则每一条带上作用的力为

$$\text{混凝土:} \quad N_i = bd\sigma_{c,i} = \Delta A\sigma_{c,i} \quad (5.3)$$

$$\begin{aligned} N'_s &= A'_s\sigma'_s \\ \text{钢筋:} \quad N_s &= A_s\sigma_s \end{aligned} \quad (5.4)$$

每一条带混凝土及钢筋对截面高度 $h/2$ 的力矩为

$$M_i = N_i y_i \quad (5.5)$$

$$M'_s = N'_s y'_s, \quad M_s = N_s y_s \quad (5.6)$$

根据力的平衡条件可以得到

$$\sum_i^n \sigma_{c,i} \Delta A_i + \sigma'_s A'_s - \sigma_s A_s = 0 \quad (5.7)$$

若有轴向力作用在截面中心轴处时,上式应为

$$\sum_i^n \sigma_{c,i} \Delta A_i + \sigma'_s A'_s - \sigma_s A_s = N \quad (5.8)$$

根据力矩平衡条件有

$$M = \sum_i^n \sigma_{c,i} \Delta A_i y_i + \sigma'_s A'_s \left(\frac{h}{2} - a' \right) + \sigma_s A_s \left(\frac{h}{2} - a \right) \quad (5.9)$$

5.1.3 弯矩-曲率计算

弯矩-曲率求解,主要是求得弯矩和曲率的对应关系,因此首先从弯矩或曲率两者之间选定一个作为已知,来确定另一个。由于弯矩-曲率存在着下降段,某些区段的弯矩值对应两个曲率,为了方便起见,可先假定曲率为已知,然后求相应的内力,具体求解步骤如下:

- (1) 取曲率 $\varphi: = \varphi + \Delta\varphi$;
- (2) 假定梁截面受压区边缘的混凝土应变 ϵ'_c ;
- (3) 求各混凝土条带和钢筋的应变;
- (4) 按混凝土和钢筋的应力-应变关系求与应变相对应的应力值;
- (5) 按式(5.8)求内力总和,判别是否满足平衡条件;
- (6) 若不满足平衡条件,则需调整应变值 ϵ'_c ,重复步骤(3)~(5);
- (7) 满足平衡条件后,按式(5.9)求内力弯矩,从而得出 φ 所对应的弯矩 M ;
- (8) 循环步骤(1)~(7),直至得出整个的 M - φ 关系。

图 5.3 为相应的计算框图。

在上述计算中会遇到数值计算的逐次逼近问题,以矩形截面偏心受压构件为例,在轴向力 N 作用下第一次假定的 ϵ'_c 为 ϵ_1 ,则通过 ϵ_1 求得各条带的应变值 $\epsilon_{1,i}$ 及相应的应力 $\sigma_{1,i}$ 后,可求得各条带的内力总和 N_1 。然而, ϵ_1 不可能一次假定就正好满足平衡条件, $N_1 - N = 0$,而需要调整 ϵ_1 。

设函数

$$y = N'(\epsilon) - N \quad (5.10)$$

通过调整 ϵ ,使 $y = 0$,即 $N'(\epsilon) = N$,则 ϵ 调整到位。

式(5.10)的插值函数为

$$\varphi(\varepsilon) = y_1 + \frac{y_2 - y_1}{\varepsilon_2 - \varepsilon_1}(\varepsilon - \varepsilon_1) \quad (5.11)$$

式中 $y_1 = N_1 - N$, $y_2 = N_2 - N$ 。要使 $\varphi(\varepsilon) = 0$, 根据式(5.11), 有

$$\varphi(\varepsilon) = y_1 + \frac{y_2 - y_1}{\varepsilon_2 - \varepsilon_1}(\varepsilon - \varepsilon_1) = N_1 - N + \frac{N_2 - N_1}{\varepsilon_2 - \varepsilon_1}(\varepsilon - \varepsilon_1) = 0$$

$$\varepsilon_3 = \varepsilon = \varepsilon_1 - \frac{N_1 - N}{N_2 - N_1}(\varepsilon_2 - \varepsilon_1) \quad (5.12)$$

按上式 ε_3 再进行复算, 看 $y_3 = N_3 - N \rightarrow 0$, 如果不小于某一规定值, 可重复上述步骤进行运算, 直至 $y_i = N_i - N$ 接近于零。

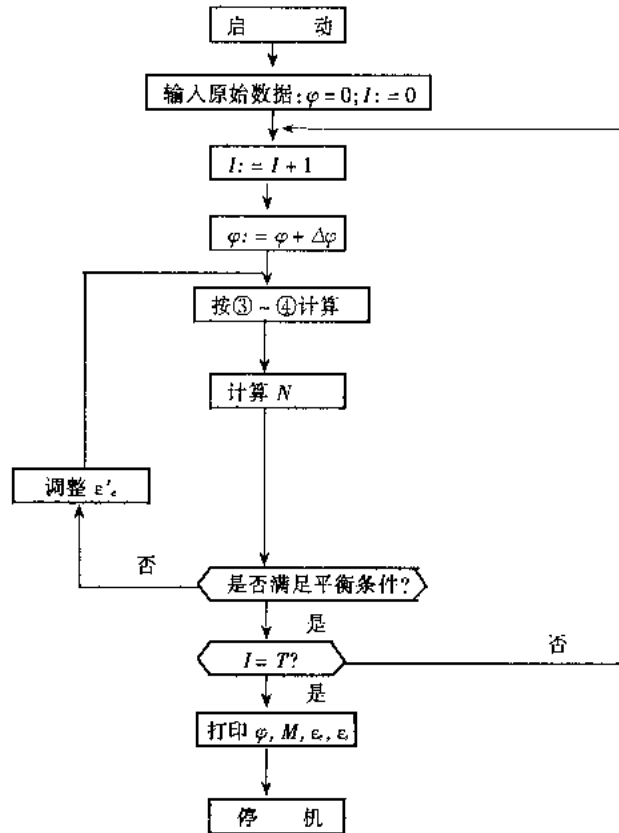


图 5.3 弯矩-曲率计算框图(注:框图中 T 为规定的计算次数)

在某些构件的计算中,弯矩或弯矩和轴力作为已知条件,求相应的曲率 φ , 以往的这一求解过程,比已知曲率 φ 求弯矩和轴力要麻烦得多,因为,在已知 φ 的前提下求弯矩和轴力,只需调整 ε 一个变量,而在已知弯矩和轴力的前提下求 φ ,则需要调整 ε 和 φ 进行试算,以满足平衡方程式(5.8)和(5.9)。此时需采用二元插值法搜索 ε 和 φ 。然而,在目前计算机速度大大提高了的情况下可采用已知 M 求 φ 的步骤,通过较大范围内的试算求得 M 所对应的 φ 。

5.2 钢筋混凝土构件的荷载-挠度曲线计算

5.2.1 梁的荷载-挠度曲线计算

以上讨论了弯矩-曲率($M-\varphi$)的计算。在求得构件各截面处的曲率以后,就可求出构件任意点的挠度。为了便于数值计算,将梁分成 m 个小段,相应结点 $m+1$ 个,为了保证足够的精度,通常 m 取 16 以上,在结点之内的每一小段内的曲率假定是线性变化的(图 5.1)。运用虚梁法(或称共轭梁法)可容易地求得任一截面的转角 θ_i 和挠度 δ_i ($i=0,1,\dots,m$)。

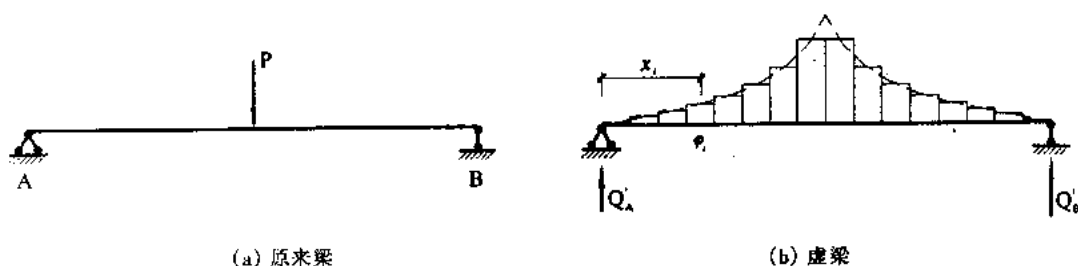


图 5.4 简支梁及其对应的虚梁

图 5.4(b)为简支梁(图 5.4(a))所对应的虚梁,根据虚荷载平衡条件

$$\phi_A = Q'_A = \sum_{i=1}^m \varphi_i \Delta x (l - x_i) / l \quad (5.13)$$

跨中位移 $\delta_{l/2}$ 为

$$\delta_{l/2} = M'_{l/2} = Q'_A \cdot \frac{l}{2} - \sum_{i=1}^m \varphi_i \Delta x \left(\frac{l}{2} - x_i \right) \quad (5.14)$$

5.1.3 中所论述的 $M-\varphi$ 关系曲线计算可采用分级加变形或分级加荷载方法进行,与此相似 $P-\delta$ 关系求解也可采用分级加变形或分级加荷载的两种方法之一。

1. 分级加变形求 $P-\delta$ 关系

分级加变形又可分为分级加曲率和分级加挠度。

(1) 分级加曲率

对于简支梁以其跨中弯矩最大截面 m 处的曲率作为控制值,逐级增加曲率。具体计算步骤如下:

- ① 每一级: $\varphi_m := \varphi_m + \Delta\varphi$;
 - ② 由 φ_m 确定跨中弯矩 M_m ;
 - ③ 由 M_m 计算 P ;
 - ④ 由 P 计算各截面 M_i ;
 - ⑤ 由 M_i 确定 φ_i ;
 - ⑥ 由 φ_i 求得挠度 δ_i ,重复步骤①~⑥,直到跨中位移 δ 达到所预定的值。
- (2) 分级加挠度

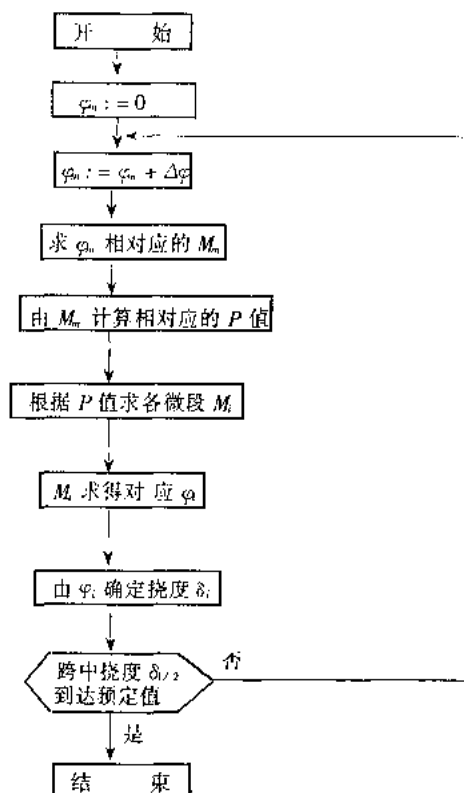


图 5.5 分级加曲率求 P - δ 曲线流程图

- ① 以简支梁跨中挠度 $\delta_{l/2}$ 作为控制值,每一级: $\delta_{l/2} = \delta_{l/2} + \Delta\delta$;
- ② 假定一个 P ,由 P 计算各截面 M_i ;
- ③ 由 M_i 确定 φ_i ;
- ④ 由 φ_i 求挠度 $\delta_{i,1}$,得到跨中 $\delta_{l/2,1}$;
- ⑤ 若 $\delta_{l/2,1}$ 与 $\delta_{l/2}$ 存在差值,调整 P ,再由步骤②开始求 $\delta_{l/2,2}, \delta_{l/2,3}, \dots, \delta_{l/2,k}$,直至 $\delta_{l/2,k}$ 与 $\delta_{l/2}$ 相差小于一规定值;
- ⑥ 重复步骤①~⑤,直至求得完整的 P - δ 曲线。

2. 分级加荷载求 P - δ 关系

- (1) P 由零开始,每一级加: $P_i = P + \Delta P$;
- (2) 由 P 计算各截面 M_i ;
- (3) 同 M_i 确定 φ_i ;
- (4) 由 φ_i 确定挠度 δ_i 。

与求 M - φ 相似,由于受力变形关系曲线存在下降段,一个力对应两个变形值,当构件达到极限承载力后 P - δ 进入下降段,计算时 ΔP 将转为负值,因此需在计算程序中增加判别过程。然而,在极限荷载区域内, P - δ 曲线的切线基本呈水平向,按上述步骤 ΔP 计算 δ 时,精度较难控制,容易溢出。

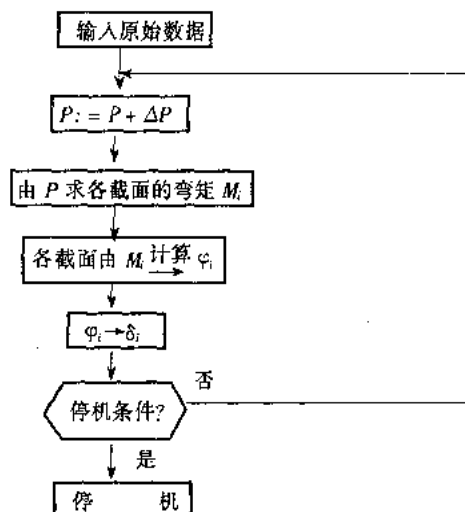


图 5.6 分级加荷载求 $P-\delta$ 曲线流程图

5.2.2 构件的卸载计算处理

构件的 $P-\delta$ 全过程计算要涉及到下降段, 构件的局部区域出现塑性铰, 构件承载力 P 相应减小, 这时构件塑性铰区段以外的区域同样出现卸载。对于塑性铰区段 $M-\varphi$ 按下降段负刚度方向取值。而对于塑性铰区段外 $M-\varphi$ 则应按正刚度方向取值。一般情况下, 卸载刚度同初始刚度比较接近, 可按初始刚度卸载, 如图 5.7。

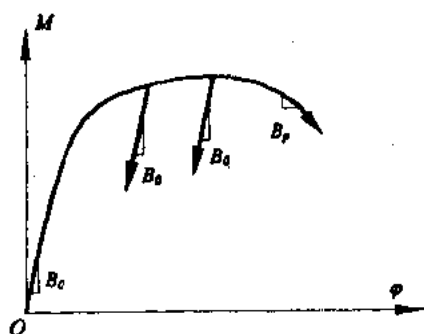


图 5.7 在弯矩-曲率图上的卸载路径

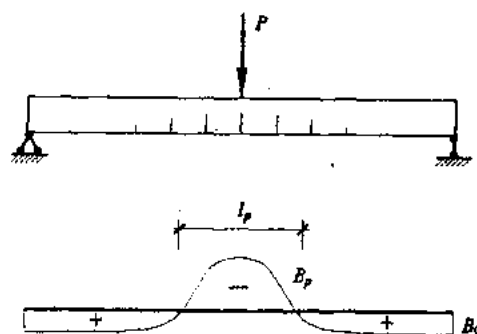


图 5.8 弯矩过 M_u 后梁的刚度分布

5.2.3 受弯构件 $P-\delta$ 曲线计算实例

1. 计算构件及钢筋、混凝土本构关系

钢筋混凝土简支梁及截面配筋示意如图 5.9 所示, 这根梁受一集中荷载作用, 作用位置离梁端距离为 a 。

程序中钢筋应力-应变关系采用理想弹塑性模型, 并考虑其屈服硬化, 钢筋应变超过 ϵ_{sh} 后取 $E'_s = 0.01E_s$ (如图 5.10 所示)。混凝土的应力-应变关系在受压区采用 Hognestad 关系式, 下降段为线性 (图 5.11), 受拉区混凝土按图中所附的公式计算。



图 5.9 简支梁受力简图及截面配筋

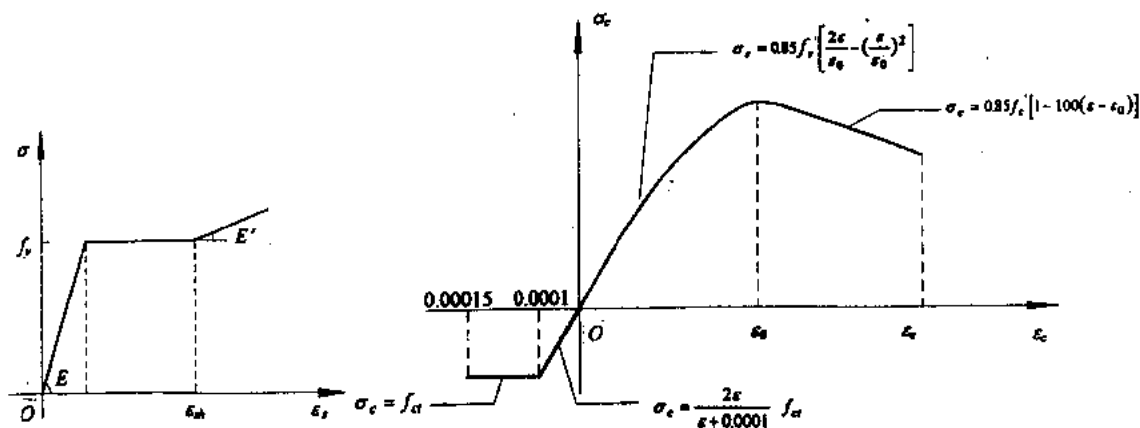


图 5.10 钢筋应力-应变关系

图 5.11 混凝土应力-应变关系

2. 源程序中各主要符号及意义:

- sn ——截面划分条带数;
- ln —— a 长度上的分段数;
- st ——钢筋型号;
- ee ——截面中间处应变;
- eo —— ϵ_0 ;
- eu —— ϵ_u ;
- z ——条带到截面中间的距离;
- $a[k]$ ——钢筋到截面上边缘距离;
- e ——应变;
- s ——应力;
- $p[j]$ ——外力;
- $d[j]$ ——挠度;
- $m[j]$ ——弯矩;
- $c[j]$ ——曲率;
- $sf1, sf2$ ——截面合力;
- ffc ——混凝土合力;
- fs ——钢筋合力;

mic——混凝土合弯矩(截面内);

mis——截面钢筋合弯矩。

注:钢筋、混凝土的应力-应变以受拉为正,受压为负。

3. 源程序、输入数据及计算结果

```
#include <stdio.h>
#include <math.h>
float abstr(float);
int sign(float);
void main(void)
{
    float m[300],c[300],p[300],d[300];
    float mom[100],coc[100];
    int i;
    for (i=0;i<300;i++)
    {
        m[i]=0.0;
        c[i]=0.0;
        p[i]=0.0;
        d[i]=0.0;
    }
    for (i=0;i<100;i++)
    {
        mon[i]=0.0;
        coc[i]=0.0;
    }
    // * * * * * Enter Data To Store In Input.dat * * * * *

    FILE * file1, * file2, * file3;
    float fy,es,esh;
    float fc,fct;
    float asl,as;
    float l,a,h,b;
    float ao[2];
    int sn,ln,st;
    file1 = fopen("input.dat","r");
    fscanf(file1,"%f%f%f", &fy, &es, &esh);
    fscanf(file1,"%f%f", &fc, &fct);
    fscanf(file1,"%f%f", &asl, &as);
    fscanf(file1,"%f%f%f%f",&l, &a,&h,&b);
    fscanf(file, "%f%f", &ao[0], &ao[1]);
```

```

fscanf(file1, "%d%d%d", &sn, &ln, &st);
fclose(file1);
// * * * * * End of Inputing Data * * * * *

float dc = 0.0000002, de = 0.00005, ee, em;
float sf1 = 0.0, sf2 = 0.0, dsf;
float ffc = 0.0, fs = 0.0;
float mi = 0.0, mic, mis;
float z, e, s, r;
float eo, eu, cc = 0.0;
float lp, hh, hn, aas, etop;
float esy, da;
float dd, dsn, d1;
float mm, mo, co, dp, tan;
int j, k, n, ii, jj;
int jmax1 = 500, jmax2 = 0, jmax3 = 0;
esy = fy/es;
hn = h/sn;
eu = -0.004;
e = -0.002;
ee = -0.0001;
j = 0;
ii = 0;
do
{
    j + +;
    c[j] = c[j - 1] + dc;
}
do
{
    ii + +;
    mic = 0.0;
    mis = 0.0;
}

// * * * * * Calculate The Force Of Concrete * * * * *

ffc = 0.0;
for(i = 0; i < sn; i + +)
{
    z = h/2 - i * hn - hn/2;
    ee = ee + z * c[j];
    if(e > 0.00015) s = 0.0;
    else if (e > 0.0001) s = fct;
}

```



```

    else if (e>0.0) s = (2 * fct * e)/(e + 0.0001);
    else if (e>eo) s = -0.85 * fc * (2 * e/eo - (e/eo) * (e/eo));
    else if (e>eu) s = -0.85 * fc * (1 - 100 * (eo - e));
    else s = 0.0;
    ffc = ffc + s * b * (h/sn);
    mic = mic + s * b * z * (h/sn);
}
* * * * * Calculate The Force Of Steel * * * * *

```

```
fs = 0.0;
```

```

for(k = 0, k<2; k + + )
{
    z = ao[k] - h/2;
    e = ee + z * c[j];
    if(abstr(e)<= esy) s = e * es;
    else if (abstr(e)<= esh) s = sign(e) * fy;
    else s = sign(e) * (fy + 0.01 * es * (abstr(e) - esh));
    if(z<0.0) aas = as1;
    else aas = as;
    if(abstr(e)>esy) de = 0.0000003;
    fs = fs + s * aas;
    mis = mis + s * aas * z;
}
if (ii == 1)
{
    sf1 = ffc + fs;
    ee = ee + de;
    em = de;
}
else
{
    sf2 = ffc + fs;
    dsf = sf2 - sf1;
    em = - sf2 * em/dsf;
    if(em == 0.0) em = de;
    ee = ee + em;
    sf1 = sf2;
}
mm = mic + mis;

```

```

    }
    while( abstr(sf2) >= 0.1);
    if( e >= esy)
    {
        if( j <= jmax1) jmax1 = j;
    }

    etop = ee - (h/2 - ln/2) * c[j];
    m[j] = mm;
    if( m[j] >= mi)
    {
        mi = m[j];
        jmax2 = j;
    }
    while( etop >= 1.2 * eu);
    jmax3 = j;
// * * * * * Put Out The Answer of Moment And Curvature * * * * *

```

```

file2 = fopen("out1.dat", "w");
for(j = 0; j <= jmax3; j++)
{
    fprintf(file2, "%18.8g, %18.8g \n", c[j], m[j]);
}
fclose(file2);

```

```

if(st == 1) lp = 0.35 * h;
if(st == 2) lp = 0.5 * h;
tan = m[jmax1]/c[jmax1];
da = a/ln;
dc = 0.00000022;
j = 0;
do
{
    cc = cc + dc;
    j = j + 1;
    if(cc <= c[jmax1])
    {
        i = -1;
        do
        {
            i++;

```

```

        if((c[i]<=cc)&&(cc<c[i+1]))
            mm = m[i] + ((cc - c[i]) * (m[i+1] - m[i])) / (c[i+1] - c[i]);
        }
    while(i<=jmax1+1);
    r = mm/a;
    p[j] = (mm * l) / (a * (1 - a));
    for(n = 1; n<=ln; n++)
    {
        mom[n] = r * n * da;
        jj = -1;
        do
        {
            jj++;
            if((m[jj]<=mom[n])&&(mom[n]<m[jj+1]))
                coc[n] = c[jj] + ((mom[n] - m[jj]) * (c[jj+1] - c[jj])) / (m[jj+1] - m[jj]);
        }
        while(jj<jmax1);
        dd = coc[n] * da * (n * da - da/2);
        d[j] = d[j] + dd;
    }
}

// * * * * * To Deal With The Plastic Hinge * * * * *

else if(cc<=c[jmax2])
{
    dsu = 0.0;
    i = -1;
    do
    {
        i++;
        if((c[i]<=cc)&&(cc<c[i+1]))
            mm = m[i] + (cc - c[i]) * (m[i+1] - m[i]) / (c[i+1] - c[i]);
    }
    while(i<=jmax2+1);
    r = mm/a;
    p[j] = (mm * l) / (a * (1 - a));
    da = (a - lp) / ln;
    for(n = 1; n<=ln; n++)
    {
        mom[n] = r * n * da;
        for(jj = 0; jj<=jmax2; jj++)

```

```

        {if((m[jj]<=mom[n]&&(mom[n]<m[jj+1]))
            coc[n]=c[jj]+((mom[n]-m[jj])*(c[jj+1]-c[jj]))/(m[jj+1]
                -m[jj]));
        }
        dd=coc[n]*da*(n*da-da/2);
        dsn=dsn+dd;
    }
    mo=r*a;
    for(jj=jmax1;jj<=jmax2;jj++)
        {if((m[jj]<=mo)&&(mo<m[jj+1]))
            co=c[jj]+((mo-m[jj])*(c[jj+1]-c[jj]))/(m[jj+1]-m[jj]);
        }
        dl=co*lp*(aa-lp/2);
        d[j]=dsn+dl;
    }
// * * * * * To Deal With The Unloading * * * * *

```

else

```

{
    dsn=0.0;
    i=-1;
    do
        {i++;
            if((c[i]<=cc)&&(cc<c[i+1]))
                mm=m[i]+((cc-c[i])*(m[i+1]-m[i]))/(c[i+1]-c[i]);
        }
    while(i<=jmax3);
    r=mm/a;
    p[j]=(mm*1)/(a*(1-a));
    dp=p[j-1]-p[j];
    da=(a-lp)/ln;
    for(n=1;n<=ln;n++)
        {mom[n]=dp*n*da;
            coc[n]=mom[n]/tan;
            dd=coc[n]*da*(n*da-da/2);
            dsn=dsn+dd;
        }
    dl=dc*lp*(a-lp/2);
}

```

```

        d[j] = d[j - 1] - dsn + d1;
    }
    while(cc <= c[jmax3]);
// * * * * * Put Out The Answer Of Force And Displacement * * * * *

    file3 = fopen("out2.dat", "w");
    for(j = 0; j < jmax3; j++)
        {fprintf(file3, "%15.4f, %15.4f\n", d[j], p[j]);
        }
    }
int sign(float num)
{int q;
    if(num < 0.0) q = -1;
    if(num >= 0.0) q = 1;
    return(q);
}
float abstr(float x)
{if(x >= 0) x = x;
    else x = -x;
    return x;
}
输入数据
* * * * * INPUT.DAT * * * * *
276.    200000.    0.05
20.7    2.83
1019.    2039.
5000.    2500.    635.    254.
35.    565.
30    30    1

```

5.2.4 压弯构件荷载-挠度($P-\delta$)全过程分析

压弯构件是具有轴向力作用的受弯构件。在轴力作用下,由于构件挠度的存在,会产生二次矩,而二次矩又将加大挠度。因此,压弯构件的 $P-\delta$ 计算既包括材料非线性又包括几何非线性。

如图 5.13 所示。在柱底处由 P 、 N 引起的弯矩 M_m 为

$$M_m = Pl + N\delta_0 = M_1 + M_2 \quad (5.15)$$

式中的第一项为水平 P 引起的弯矩,第二项即为二次矩,也称为附加弯矩。 $M_m - \varphi_m$ 关系曲线如图 5.14 所示。

从式(5.15)及图 5.14 可以看出,随着截面曲率的逐步增加,附加弯矩 M_2 不断加大。

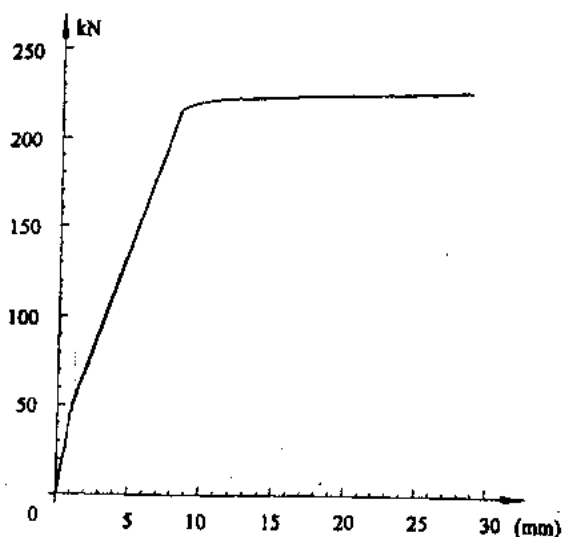


图 5.12 钢筋混凝土梁受力-变形曲线

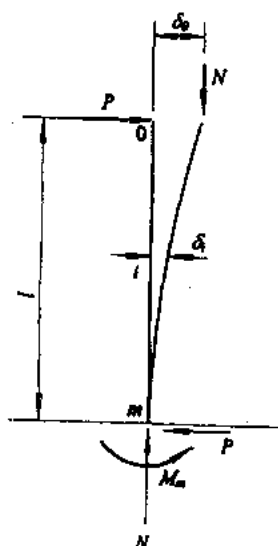


图 5.13 压弯构件受力-变形示意图

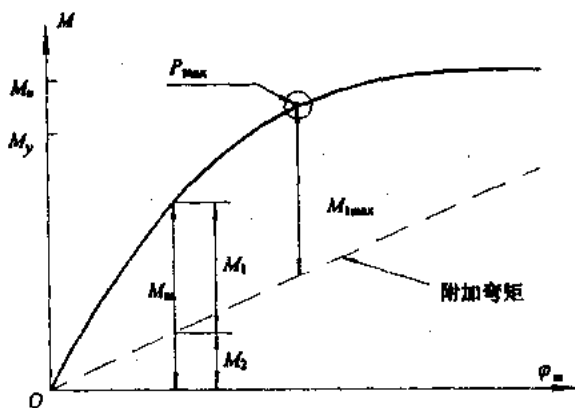


图 5.14 M_m - φ_m 关系曲线

根据图 5.13 有

$$P = (M_m - N\delta_0)/l \quad (5.16)$$

由于二次矩与挠度的相互连动影响,在数值分析中必须采用反复迭代,逐次逼近,从而求得实际解。

全过程计算分析中会涉及构件的下降段,对于单个构件来说,采用位移控制,即加变形的方法比较简便。但是,计算过程中需注意某杆段进入下降段后,其它杆段会卸载, M - φ 关系须循线性卸载线取值。

与受弯构件相似,加变形也有加曲率和加位移两种方法。

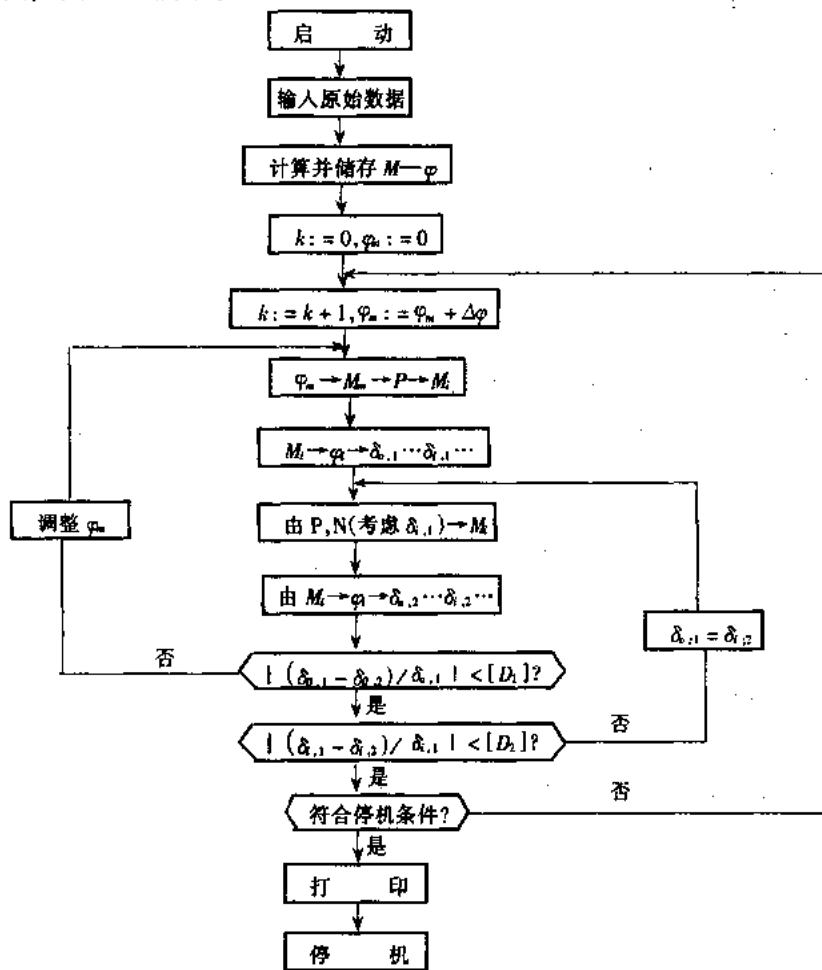
1. 分级加曲率求 P - δ 关系曲线

压弯构件的轴力 N 为不变量,因此 $M-\varphi$ 为一确定的曲线,其具体的计算步骤为:

- (1) φ_m 由零开始,每一级加: $\varphi_m := \varphi_m + \Delta\varphi$;
- (2) 由 φ_m 得出 M_m ;
- (3) 假定挠曲线 $\delta_{i,0}$ ($i=0,1,\dots,m$) (每次可取上一次的挠曲线,取第一次的挠曲线时,假定杆的曲率呈三角形分布, φ_m 为一较小值);
- (4) 由 $M_m, N, \delta_{i,0}$ 求 P (按式 5.16);
- (5) 由 $P, N, \delta_{i,0}$ 求各截面的 M_i ,并由 M_i 查得 φ_i ;
- (6) 由 φ_i 求得新的挠曲线 $\delta_{i,1}$,求各截面的 M_i ,并由 M_i 查得 φ_i ;
- (7) 以新的挠曲线 $\delta_{i,1}$ 为假定值重复步骤(4)~(6)求得 $\delta_{i,2}, \delta_{i,3}, \dots, \delta_{i,k}$ 。

若 $\delta_{i,(k-1)}$ 与 $\delta_{i,k}$ 接近且小于一允许值,则可认为计算结果有效,并进入下一循环计算,直至得出整个 $P-\delta$ 曲线。

计算框图如图 5.15 所示。



(框图中 $[D_1], [D_2]$ 为允许限值)

图 5.15 分级加曲率求 $P-\delta$ 曲线计算框图

2. 分级加挠度

- (1) δ_0 由零开始, 每一级加: $\delta_0 := \delta_0 + \Delta\delta$;
- (2) 假定 $P_{i,0}$ 和挠曲线 $\delta_{i,0}$ (每次可取上一次的挠曲线乘以一个比例系数, 取第一次的挠曲线时, 假定杆的曲率分布为三角形);
- (3) 由 $N, P_{i,0}, \delta_{i,0}$ 求各杆段 M_i ;
- (4) 由 M_i 查得 φ_i , 由 φ_i 求 $\delta_{i,1}$;
- (5) 由新求得的挠曲线 $\delta_{i,1}$ 为假定值, 重复步骤(3)、(4)求得 $\delta_{i,2}, \delta_{i,3}, \dots, \delta_{i,k}$ 。直至若 $\delta_{i,(k-1)}$ 与 $\delta_{i,k}$ 接近且小于一允许值。
- (6) 将 $\delta_{0,k}$ 与 δ_0 相比较, 若接近且小于一允许值, 则可认为计算结果有效并加以记录, 然后进入下一循环计算。否则, 调整 P 并重复步骤(2)~(6), 直至 $\delta_{0,k}$ 与 δ_0 相接近为止。

5.2.5 偏压构件轴力-位移曲线全过程分析

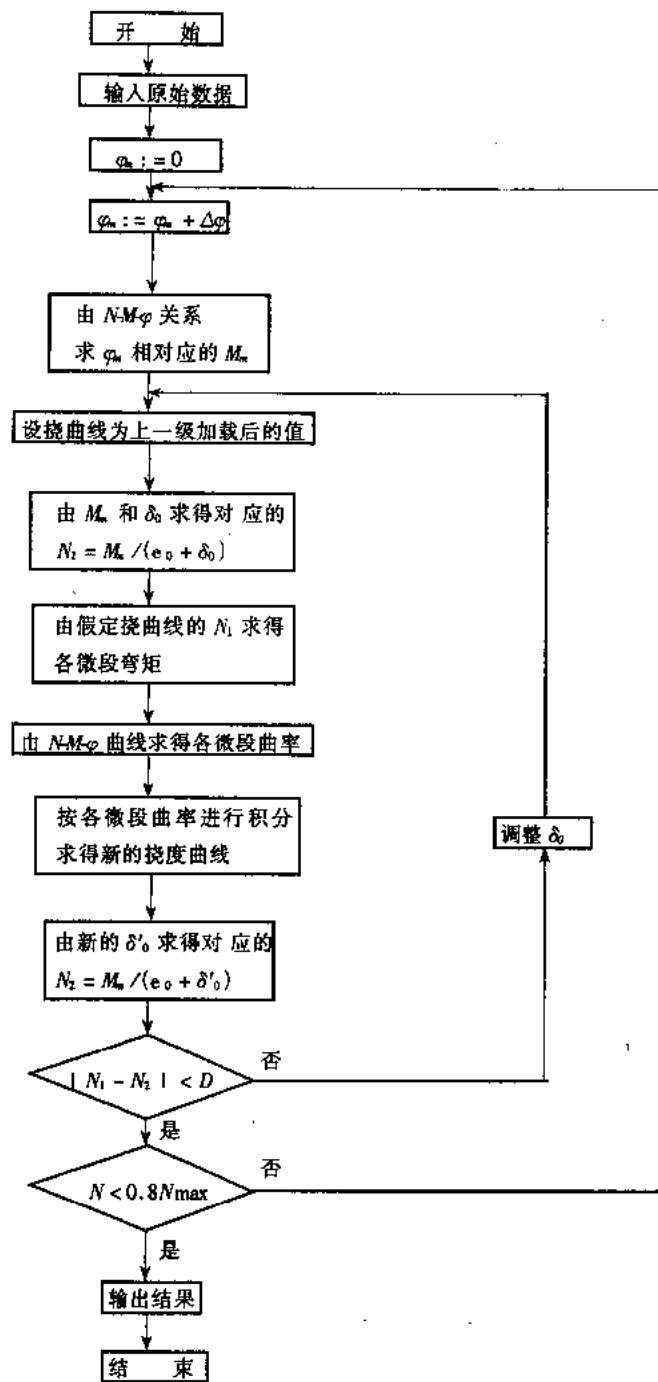
偏压构件与压弯构件的不同在于, 偏压构件的轴力 N 为变量, 初始偏心矩 e_0 为常量; 而压弯构件的轴力 N 为不变量, 横向力为变量。两者的共同之处在于均受二次弯矩作用的影响, 截面的破坏均是压力、弯矩共同作用下的破坏, 由于变量的不同, 偏压构件的全过程分析按 $N-\delta$ 关系曲线进行, 压弯构件则按 $P-\delta$ 曲线计算。

在偏压构件的全过程分析, 早期的计算中通常采用预先计算存储, 后调用 $M-\varphi$ 的方法, 但由于偏压构件的轴力 N 为变量, 因此, 需要计算一族 $N-M-\varphi$ 关系曲线, 并以二维数组的形式存储起来, 以备调用。然而, 随着计算机计算速度的不断提高和内存容量的加大, $N-M-\varphi$ 的对应关系可采用随调随算的方式进行。

与其它具有下降段的单个构件受力分析计算方法相似, 偏压构件的全过程分析通常采用加位移的方法。具体计算步骤如下:

- (1) 由 $\varphi_m = 0$ 开始, 每次 $\varphi_m := \varphi_m + \Delta\varphi$;
- (2) 由 $N-M-\varphi$ 关系求 φ_m 相对应的 M_m , 其中 N 取上一级的计算值, 开始时 N 取零;
- (3) 根据上一次计算的 δ_0 值及 M_m 反算出纵向轴力 N_1
$$N_1 = M_m / (e_0 + \delta_0) \quad (5.17)$$
- (4) 由 N_1 和挠曲线计算各微段弯矩 M_i , 再由 M_i 求出 φ_i ;
- (5) 由 φ_i 计算 δ'_i , 并由 δ'_i, e_0, M_m 反算出纵向轴力 N_2 , 若 N_2 和 N_1 相差较大, 则调整 δ_0 , 重复计算步骤(4)~(5), 直至 N_2 和 N_1 相当接近且小于某一限定值。

计算框图如图 5.16 所示。



(框图中 D 为允许限值)

图 5.16 偏压构件轴力-位移曲线计算框图

5.2.6 偏压构件计算分析实例

1. 计算构件及计算简图

钢筋混凝土柱截面配筋及计算简图如图 5.17 所示, 钢筋屈服强度 $f_y = 315 \text{ MPa}$, 弹性

模量 $E_s = 2.03 \times 10^5 \text{ MPa}$, 混凝土圆柱体抗压强度 $f'_c = 15 \text{ MPa}$ 。

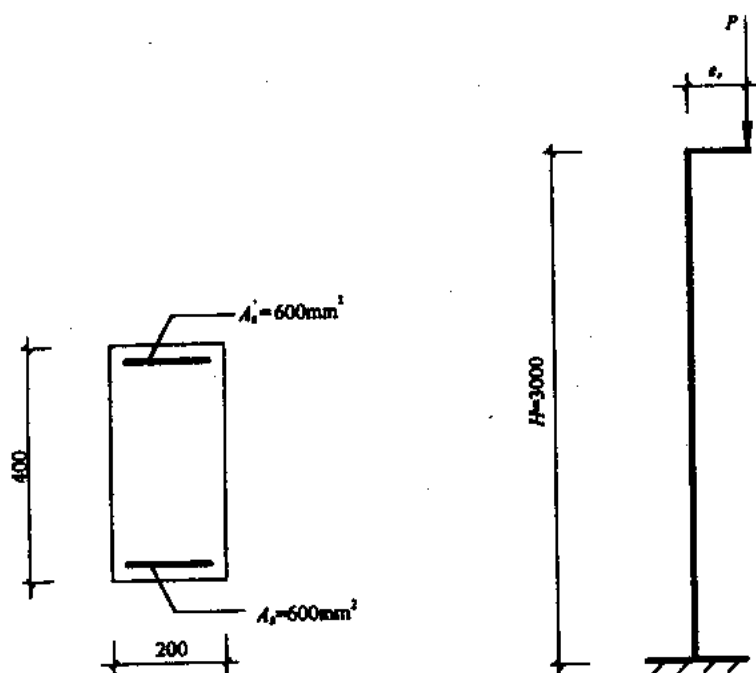


图 5.17 偏压构件受力简图及截面配筋

程序中钢筋、混凝土应力-应变关系与图 5.10、图 5.11 的钢筋、混凝土应力-应变关系相似, 钢筋在受压区不考虑其屈服后的强化。

偏压构件计算实例的轴力-位移曲线结果如图 5.18 所示。

2. 偏压构件计算源程序

```
#include <stdio.h>
#include <math.h>
#include <iostream.h>
#include <process.h>
#include <fstream.h>
```

```
FILE * OutputFile;
struct Const{
    double Es;
    double HardenEs;
    double YieldStress;
    double YieldLength;
    double fet;
    double fc;
```

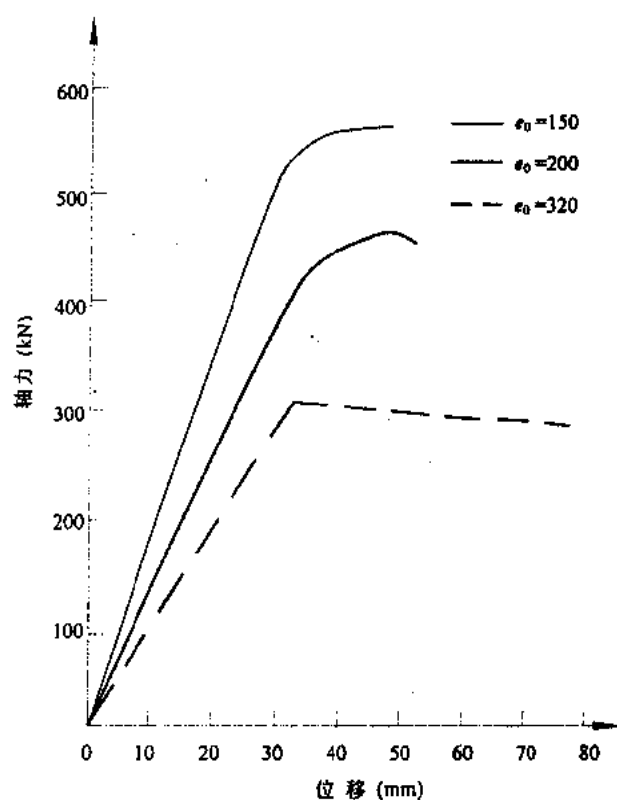


图 5.18 偏压构件轴力-位移计算结果

```

}
struct data{
    int StripeNum, UnitNum;
    double b,h,as,Ast, Asc,H;
};
// - - - - - class StrainStress
Class StrainStress{
    double CPStrain;
    double CUStrain;
public:
    Const AConst;
    StrainStress(Const _ Const){
        AConst = _ Const;
        CPStrain = 0.002;
        CUStrain = 0.004;
    }
    double SteelStress(double);

```

```

    double ConcreteStress(double);
};

double StrainStress::SteelStress(double Strain){//根据钢筋应变计算钢筋应力
    int isTension = 0;
    if((Strain + 1.0) >= 1.0) isTension = 1;
    double result = 0.0;
    double YieldStrain = AConst.YieldStress/AConst.Es;
    double HardenStrain = YieldStrain + AConst.YieldLength;
    if(fabs(Strain) <= YieldStrain)
        result = fabs(Strain) * AConst.Es;
    else if(fabs(Strain) <= HardenStrain)
        result = AConst.YieldStress;
    else
    {
        if (isTension)
            result = (fabs(Strain) - HardenStrain) * AConst.HardenEs
                + AConst.YieldStress;
        else
            result = AConst.YieldStress;//受压钢筋不考虑强化作用
    }
    if((Strain + 1.0) > 1.0)
        return result;
    else
    {
        if(fabs(Strain) > 0.004) //当受压混凝土应变大于 0.004 时,混凝土压碎,
            return 0.0;
        else return -result;
    }
}

double StrainStress::ConcreteStress(double Strain){//根据混凝土应变计算应力,
                                                    //受拉为正

    int isTension = 1;
    if(Strain < 0) isTension = 0;
    if(isTension){ //不考虑混凝土受拉
        return 0.0;
    }
    else{
        Strain = fabs(Strain);
        double Stress0 = 0.85 * AConst.fc;

```

```

        if (Strain <= CPStrain){
            double ratio = Strain/CPStrain;
            return -Stress0 * (2 * ratio - ratio * ratio);
        }
        else if(Strain <= CUStrain){
            double ratio = (Strain-CPStrain)/(CUStrain-CPStrain);
            return -Stress0 * (1 - 0.15 * ratio);
        }
        else if(Strain <= 0.01) return 0.1 * Stress0; //当混凝土压碎后,
                                                    //假设剩下 10% 的
                                                    //剩余强度
        else{
            cout << endl << "Concrete is callapse!" << endl;
            cout << "Program terminated!" << endl;
            fclose(OutputFile);
            exit(1);
        }
    }
}

// - - - - - class MomentCurvature - - - - -
class MomentCurvature: public StrainStress{
    double * Z;
    double * pCStrain;
    double * pCStress;
    int StripeNum, UnitNum;
    double __moment[1024];
    double b, h, as, Ast, Asc, H;
    double SStressC, SStressT;
    double TotalForce, TotalMoment;
    double AxialForce;
    double _totalForce(double, double);
public:
    double M _ FiStiffness;
    int isYield;
    MomentCurvature(data, Const);
    ~ MomentCurvature();
    double _moment(double);
    double Moment(double, double);

```

```

void Curvature(double * ,double * ,double);
double Moment _ Table(double, double);
};

MomentCurvature::MomentCurvature(data Adata, Const AConst):StrainStress(AConst){
    //构造函数
    StripeNum = Adata.StripeNum;
    UnitNum = Adata.UnitNum;
    b = Adata.b;
    h = Adata.h;
    as = Adata.as;
    Ast = Adata.Ast;
    Asc = Adata.Asc;
    H = Adata.H;
    Z = new double[StripeNum];
    pCStrain = new double[StripeNum];
    pCStress = new double[StripeNum];
    isYield = 0;
}

MomentCurvature::~MomentCurvature(){} //析构函数
    delete Z;
    delete pCStrain;
    delete pCStress;
}

double MomentCurvature::_totalForce(double StrainAve, double Curvature){
    //计算截面合力
    for(int i = 0; i < StripeNum; i++){
        Z[i] = -h/2 + (i + 0.5) * h/StripeNum;
        pCStrain[i] = StrainAve + Z[i] * Curvature;
        pCStress[i] = ConcreteStress(pCStrain[i]);
    }
    SStressC = SteelStress(StrainAve - (h/2 - as) * Curvature);
    SStressT = SteelStress(StrainAve + (h/2 - as) * Curvature);
    TotalMoment = 0;
    TotalForce = 0;
    for(i = 0; i < StripeNum; i++){
        TotalForce + = pCStress[i] * b * H/StripeNum;
    }
}

```

```

        TotalMoment += pCStress[i] * b * h * Z[i]/StripeNum;
    }
    TotalForce += SStressT * Ast + SStressC * Asc;
    TotalMoment += (SStressT * Ast - SStressC * Asc) * (h/2 - as);
    return TotalForce;
}

double MomentCurvature::_moment(double Curvature){//根据曲率求弯矩
    double StrainAve1 = -0.004, StrainAve2 = -0.004;
    double Increment = 0.0002;
    double N1, N2;
    N1 = _totalForce(StrainAve1, Curvature);
    int i = 0;
    while (N1 + AxialForce < 0){
        StrainAve2 += Increment;
        N1 = _totalForce(StrainAve2, Curvature);
    }
    i = 0;
    while (fabs(N1 + AxialForce) > 10){
        double midAve = (StrainAve1 + StrainAve2)/2;
        N1 = _totalForce(midAve, Curvature);
        if(N1 + AxialForce > 0) StrainAve2 = midAve;
        else if(N1 + AxialForce < 0) StrainAve1 = midAve;
        else break;
    }
    return TotalMoment;
}

double MomentCurvature::Moment(double Curvature, double _ AxialForce){
    //根据曲率求弯矩
    AxialForce = _ AxialForce;
    return _moment(Curvature);
}

double MomentCurvature::Moment _ Table(double Curvature, double _ AxialForce){
    //刷新弯矩-曲率表
    double Curv = 0;
    AxialForce = _ AxialForce;
    double Increment = 0.0000002;
    double M0, M1 = 0;
    int isInitTable = 1;

```

```

int i = 0;
do {
    Curv + = Increment;
    M0 = M1;
    M1 = _moment(Curv);
    _moment[ + + i] = M1;
    if(M0 > M1)//判断截面是否屈服
    {
        if (isInitTable){
            isYield = 1; isInitTable = 0; }
    }
} while (Curv < = Curvature);
M_FiStiffness = _moment[10]/(10 * Increment);
return M1 - (M1 - M0) * (Curv-Curvature)/Increment;
}

void MomentCurvature::Curvature (double * Moment, double * _Curvature,
                                   double _AxialForce){
    //计算各截面的曲率
    if(isYield) cout < < "isYield" > > endl;
    double Curv = 0;
    AxialForce = _AxialForce;
    double Increment = 0.0000002;
    double M0, M1 = 0;
    int YieldUnitNum = int(1.5 * h * UnitNum/H); //计算塑性铰长度
    for(int i = 0; i < UnitNum; i + + ){
        int j = 0;
        do{
            Curv + = Increment;
            M0 = M1;
            M1 = _moment[ + + j];
        } while(M1 < = Moment[i]);
        if(M1 - M0 == 0) _Curvature[i] = Curv-Increment/2;
        else _Curvature[i] = Curv-Increment
            + Increment * (Moment[i] - M0)/(M1 - M0);
        Curv = 0;
    }
    if(isYield)//对于塑性铰进行处理
    {

```



```

        for(int j = 1;j < YieldUnitNum;j + + )
            _ Curvature[ UnitNum-j] = _ Curvature[ UnitNum];
    }
}

// - - - - - class of main program - - - - -
class MainClass{
    data Adata;
    Const AConst;
    int UnitNum;
    double H;
    double * Curvature;
    double * Moment;
    double * Deflection;
    double Nmax;
    double e0;
    double N1;
    MomentCurvature * MC;
    void _ moment(double, double *, double * );
    void _ deflection(double *, double * );
    int _ AxialForceOK(double *, double *, double *, double&, double&);
    void Kernal();
public:
    MainClass();
    ~ MainClass();
    void execute();
};

MainClass::MainClass(){//构造函数
    ifstream input;
    cout << endl << "unit Number:";
    cin >> UnitNum;
    Adata.UnitNum = UnitNum;
    cout << endl << "Stripe Number of Cross Section:";
    cin >> Adata.StripeNum;
    Curvature = new double[ UnitNum + 1];
    Moment = new double[ UnitNum + 1];
    Deflection = new double[ UnitNum + 1];
    for (int i = 0;i <= UnitNum;i + + ){

```

```

    Curvature[i] = 0;
    Moment[i] = 0;
    Deflection[i] = 0;
}
//输入计算参数
cout << endl << "Es, Hardened Es, Yield Stress, Yield Length, fct, fc:" << endl;
cin >> AConst.Es >> AConst.HardenEs >> AConst.YieldStess
    >> AConst.YieldLength >> AConst.fct >> AConst.fc;
cout << endl << "b, h, as Ast, Asc:" << endl;
cin >> Adata.b >> Adata.h >> Adata.as >> Adata.Ast >> Adata.Asc;
cout << endl << "Column Height:";
cin >> H;
Adata.H = H;
cout << endl << "e0:";
cin >> e0;
Nmac = AConst.fc * Adata.b * Adata.h + AConst.YieldStess * (Adata.Ast + Adata.Asc);
MC = ner MomentCurvature(Adata, Aconst);
}
MainClass::~MainClass(){//析构函数
    delete Curvature;
    delete Moment;
    delete Deflection;
    delete MC;
}
// - - - - - procedure: ger deflection form Curvature of each section - - - - -
void MainClass::_deflection(double * Curvature, double * deflection){
    //根据各截面曲率求挠曲线
    double total = 0;
    double level;
    double x = H/UnitNum;
    for(int i = 0; i < UnitNum; i++){
        for(int j = i; j < UnitNum; j++){
            level = (j - i + 0.5) * x;
            total += Curvature[j + 1] * X * level;
        }
        deflection[i] = total;
        total = 0;
    }
}

```

```

        deflection[UnitNum] = 0;
    }
    // - - - - form N0, 1 - - > Mi - - - -
    void MainClass::_moment(double AxialForce, double * deflection, double * moment){
        //计算二次效应后的各截面弯矩
        for(int i = 0, i <= UnitNum; i++){
            moment[i] = AxialForce * (e0 + deflection[0] - deflection[i]);
        }
    }
    int MainClass::_AxialForceOK(double * Curvature, double * Moment,
                                double * Deflection, double& N1, double& N2){
        //根据固定端曲率求轴力
        double _N0 = 0, _N1, _N2, _Moment2, G1, G2;
        double Increment = 50000;
        Moment[UnitNum] = MC -> Moment(Curvature[UnitNum], _N0);
        _N2 = Moment[UnitNum]/(e0 + Deflection[0]);
        while (_N2 - _N0 > 0){
            _N1 = _N0;
            _N0 += Increment;
            Moment[UnitNum] = MC -> Moment(Curvature[UnitNum], _N0);
            _N2 = Moment[UnitNum]/(e0 + Deflection[0]);
        }
        while(fabs((_N1 - _N0)/(_N1 + 300)) > 0.001){
            double temp = (_N1 + _N0)/2;
            Moment[UnitNum] = MC -> Moment(Curvature[UnitNum], temp);
            _N2 = Moment[UnitNum]/(e0 + Deflection[0]);
            if(_N2 - temp > 0) _N1 = temp;
            else _N0 = temp;
        }
        N1 = _N1;
        MC -> Moment _Table(Curvature[UnitNum], N1);
        _moment(N1, Deflection, Moment);
        MC -> Curvature(Moment, Curvature, N1);
        _deflection(Curvature, Deflection);
        N2 = Moment[UnitNum]/(e0 + Deflection[0]);
        double err;
        double errLimit = 0.001;
        err = fabs((N1 - N2)/(N1 + 100));
    }

```

```

    if(err < errLimit) return 1;
    else return 0;
}

void MainClass::Kernal() { //主循环
    double N2, N3, N4, Predeflex;
    while(! _ AxialForceOK(Curvature, Moment, Deflection, N1, N2)) {
        Predeflex = Deflection[0];
        Deflection[0] + = 0.05;
        N3 = N1;
        _ AxialForceOK(Curvature, Moment, Deflection, N3, N4);
        if(N4 - N3 - N2 + N1 = 0) cout < < "N4 - N3 - N2 + N1 = 0";
        Deflection[0] = Predeflex - 0.005 * (N2 - N1)/(N4 - N3 - N2 + N1);
        Predeflex = Deflection[0];
    }
}

void MainClass::execute() { //执行函数
    FILE * Outputfile;
    char string[100];
    cout < < endl < < "outfile name:";
    cin > > string;
    Outputfile = fopen(string, "w");
    long unsigned int loopIndex = 0;
    double lastD;
    lastD = 0;
    N1 = 0;
    fprintf(Outputfile, "%10.5f%15.5f \n", 0.0, 0.0);
    fclose(Outputfile);
    do {
        Outputfile = fopen(string, "a");
        Curvature[UnitNum] + = 0.0000005;
        Kernal();
        if(lastD > Deflection[0]) exit(2);
        lastD = Deflection[0];
        fprintf(Outputfile, "%15.5f%15.5f \n", Deflection[0], N1);
        cout < < "Loop Index:" < < + loopIndex < < endl;
        fclose (Outputfile);
    } while (N1 < 0.8 * Nmax);
}

```

```

void main() {
    MainClass AMainClass;
    AMainClass.execute();
}

```

5.3 按平面应力问题进行梁的有限元分析

上一节钢筋混凝土梁的计算模型建立在平截面假定的基础上,这一计算模型对以受弯变形为主的梁、柱进行强度计算及变形分析具有足够的精度。然而,对于深梁及所受剪力较大的杆或杆段以上模型则不太适用。若要对其进行较详细全面的数值分析,就需采用平面应力问题的有限元法。

钢筋混凝土梁的有限元模型如图 5.19 所示。根据梁的受力情况,可按平面应力有限元问题进行求解。混凝土一般划分成三角形单元、矩形单元或等参单元,钢筋可作为独立的单元,与混凝土单元通过联结单元相连接,组成分离式模型,也可与混凝土单元一同考虑,组成组合式或整体式模型(图 5.20)。

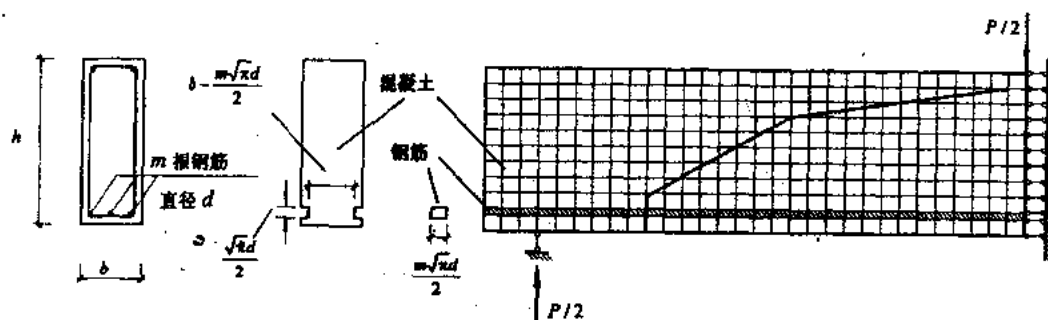
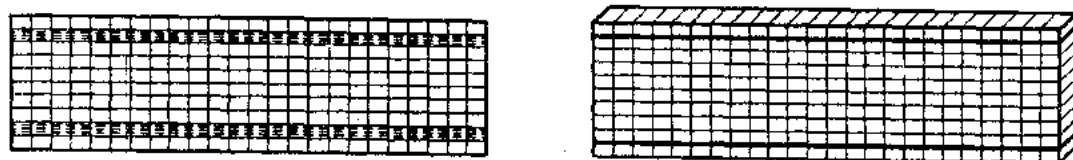


图 5.19 钢筋混凝土梁有限元计算模型



(a) 整体式

(b) 组合式

图 5.20 钢筋混凝土有限元计算模式

考虑到混凝土与钢筋之间可能出现的相对滑移,分离式模型中由联结单元来反映这

一现象,具体模型可参见 § 3.4。在整体式模型中混凝土与钢筋之间的粘结、滑移关系,则通过混凝土及钢筋的应力-应变关系来加以反映,关于这方面的论述详见 § 7.2.2。

参考文献[10]中列出了采用分离式、组合式和整体式有限元模型对钢筋混凝土简支梁的分析结果,并与试验进行了比较。通过比较认为,无论用哪一种有限元模式,均可得到良好的结果。

除了要反映钢筋与混凝土两种不同材料的共同工作特点外,钢筋混凝土的另一个主要特性是带裂缝工作。早先的钢筋混凝土梁的有限元分析,采用预先假定裂缝的方式,在划分单元时将裂缝走向作为单元的边界,计算过程中单元中的受拉应力达到抗拉强度后,认为混凝土在这一区域开裂,单元与相邻单元在边界处断开,原来相连的单元边界上的一个共同结点分成两个独立的结点。但是,由于裂缝的开裂方向不可能完全与划分好的单元边界相重合,在以后的分析中则采用边计算边划分单元的方法,待单元混凝土应力达到抗拉强度后,将构件的部分区域或整个构件重新进行单元划分,并增加新的结点。然而,这一调整工作非常复杂,虽然,平面有限元的单元划分的自动化程度已达到了较高的水平,但是,非线性计算过程中单元中的应力、应变状况需加以记录和追踪,单元的重新划分调整势必要牵涉到多方面的计算处理过程,增加许多繁琐的工作。因此,在目前这一种处理方式的运用较少。

另一种处理方法是以弥散的(或称分布的)裂缝模型来替代单独的裂缝,在单元开裂后,认为该单元仍是连接的,仍可按连续介质力学的方法对单元进行分析处理。这一方法将原来的一条单独的裂缝的受力-变形关系综合进整个单元的受力-变形中加以考虑。这种裂缝处理方法不必重新划分单元和增加新结点,计算程序相对简单易行,因此,得到了较为广泛的应用。

对于开裂混凝土之间的骨料咬合作用,在单元边界裂缝模式中,通过裂缝间的联结单元进行模拟(图 5.21)。而在弥散裂缝模式中骨料咬合作用由单元的剪切模量值的大小来加以反映。

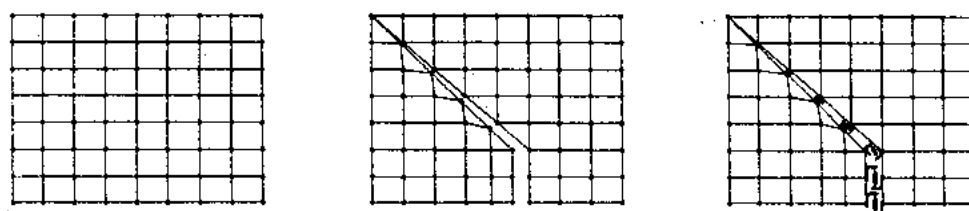


图 5.21 分离式裂缝处理

混凝土材料的本构关系可采用第三章所述的各种本构模型,混凝土的破坏准则可根据本构关系模型、应力状况、计算精度要求结合程序的效率和可行性进行选择。按平面应力问题进行钢筋混凝土梁的有限元分析可以适应于任何荷载作用的情况,特别是用于梁的抗剪计算分析时非常有效。计算中可以得出各个截面的应力和应变,同样也可以得到各种荷载下的 $P-\delta$ 曲线。

关于钢筋混凝土梁有限元非线性求解过程及其中的某些技术处理方法将在第七章的钢筋混凝土剪力墙分析中详细讨论,这里不加赘述。第七章所附的钢筋混凝土剪力墙非线性有限元计算程序同样可用于梁的分析。

第六章 钢筋混凝土框架结构有限元分析

框架结构主要由梁和柱组成,梁柱可以采用预制或现浇制成。由于框架结构的建筑平面布置灵活,立面也可变化,因此可做成需要较大空间的商场、餐厅、会议室、办公室、实验室等,也可加隔墙,做成不同用途的小房间。

框架结构的缺点是其侧向刚度较小,在地震作用下变形较大,容易引起梁或柱的破坏。因此,对框架结构详尽的试验研究和理论分析,有助于提高对框架结构性能认识,进行合理的结构设计,确保框架结构具有良好的抗震能力。

6.1 基本假定与结构简化

实际的框架结构是一个复杂的空间体系,加上很复杂的荷载,以及钢筋混凝土材料的非线性特性,对实际框架结构进行精确分析是相当困难的。因此,在计算时,必须作一些简化假定,忽略次要因素,反映主要问题。

1. 荷载作用方向

风载及地震作用方向是随意的。而在结构计算中通常假定水平力作用在结构的主轴方向,对相互正交的两个主轴 x 方向和 y 方向进行分解,分别进行内力分析。在矩形平面中,主轴分别平行于两个边长方向,如图 6.1。对于其他形状的结构平面,可根据实际平面几何形状和尺寸确定主轴方向。

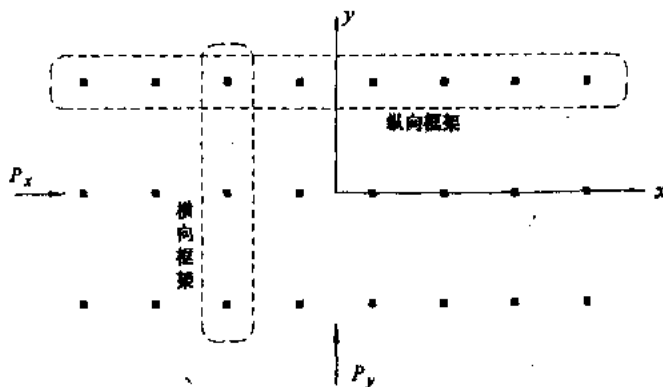


图 6.1 平面结构假定

2. 平面结构假定

任何结构都是一个空间结构,但对于框架结构体系而言,大多可以简化为平面结构,使计算大大简化。在简化中作了以下两个假定:

(1) 一片框架在其平面内承担侧向力的作用,而在其平面外的刚度很小,可以忽略。因此一个框架可以划分成若干个平面结构,共同承担与平面平行的侧向荷载,垂直与该平面方向的结构不参加受力。

(2) 各个平面框架结构之间通过楼板联系。楼板在其自身平面内刚度无穷大,而其平面外的刚度很小,可以忽略。

由以上两个基本假定出发,对于图 6.1 的结构,在 y 方向把结构简化为 8 榀框架,承担 y 方向的水平力。这 8 榀框架结构之间由无限刚度的楼板联系。当结构无扭转时,各榀框架在每层楼板处的侧移相等。当结构有扭转时,楼板仅作刚体转动,各榀框架在每层楼板处的侧移值呈直线关系,如图 6.2。同样,在 x 方向把结构简化为 3 片榀框架。

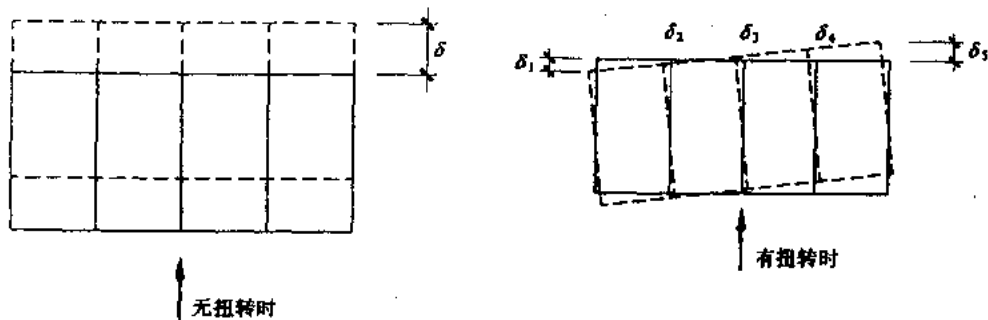


图 6.2 楼板刚度无限大假定

6.2 结构非线性计算模型

钢筋混凝土框架是受弯构件和压弯构件的组合物体。对于基本构件来说,它们的受力变形关系,即弯矩-曲率($M-\varphi$)和荷载-挠度($P-\delta$)关系都已在第五章中讨论过了。钢筋混凝土框架结构的计算即是这些非线性基本构件组合体的计算。而构件组合体的框架结构是一个超静定结构,各构件的受力-变形相互影响。由于弯矩会引起构件刚度的变化,构件刚度的变化又将引起内力重分布。因此,框架结构的整体刚度矩阵将随荷载和结构变形的变化而变化。与偏心受压柱相似,由于轴力的存在,框架的侧移变形将在框架中引起二次矩。二次矩是非线性的,对于静定的构件来说,问题较容易处理,而在超静定结构框架中的处理则较为烦琐。

对框架结构进行全过程分析的目的在于:①研究构件塑性铰出现过程和所处位置,发现结构的薄弱环节。认识结构内力分布规律,对结构进行优化设计;②通过对整个框架的受力-变形分析,得出框架柱的合理计算长度。

6.2.1 框架结构非线性计算基本框图

框架结构有限元计算,通常采用直接刚度法,其基本方程为

$$[K]\{\delta\} = \{P\} \quad (6.1)$$

式中 $[K]$ ——总刚度矩阵,若考虑二次矩的影响,需增加一个几何刚度矩阵;

$\{\delta\}, \{P\}$ ——位移及荷载列矩阵。

计算过程中,首先建立每个杆单元刚度矩阵;其次,将单元刚度矩阵组合成总刚度矩阵;第三,建立等效结点荷载列阵,然后求解位移;第四,计算单元杆端弯矩和各截面弯矩;第五,根据各截面弯矩求相应的曲率和刚度;重复以上步骤,直至求得框架受力-变形全过程。

图 6.3 是以加荷载增量进行全过程计算的框图。这一求解方法属于第四章所述的增量法,增量法只能用于结构整体正刚度情况下的求解,当结构承受的荷载达到最大值后,刚度将进入下降段。求解 $P-\delta$ 下降段,以往采用加位移方案(如图 6.4(a)),但求解过程非常复杂。目前相对较为简洁的求解方法是,根据第四章所论述的增量法在结构负刚度状态下的求解方法,在框架侧向加虚拟弹簧,求得 $P-\delta$ 下降段。典型框架的所加虚拟弹簧位置如图 6.4(b)所示。

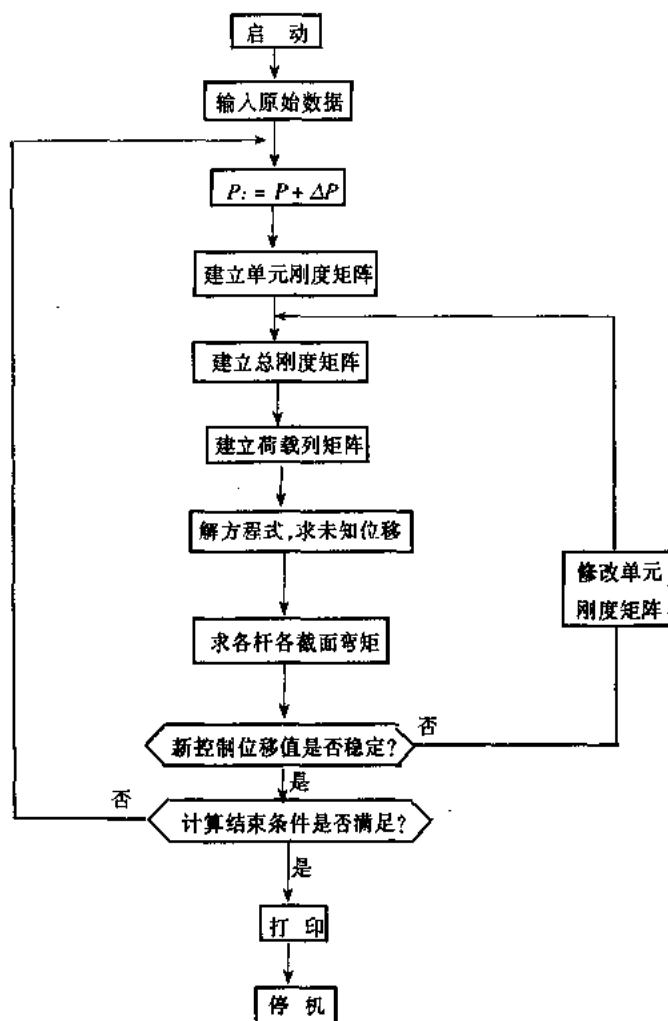


图 6.3 框架结构受力-变形计算框图

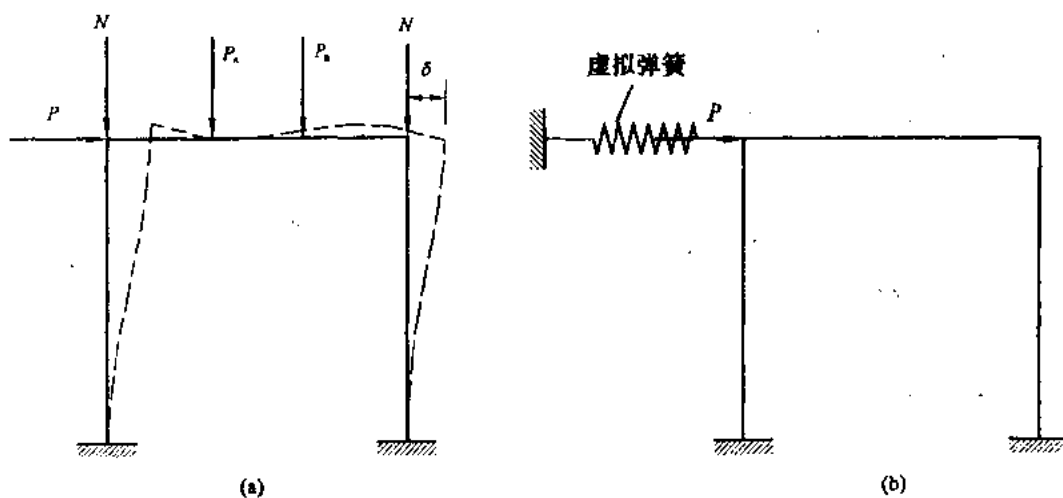


图 6.4 框架计算模型

6.3 结构有限元非线性分析

6.3.1 简化刚度矩阵法

1. 不考虑二次矩

简化刚度法就是对每根杆件单元的刚度给予一定的模型。如图 6.5 所示。当杆端塑性铰出现以前,杆件的截面刚度为常数,当弯矩到达屈服弯矩 M_y 时,刚度则下降进入另一常数。

为了计算方便,图 6.5 刚度模型可以用双分量的模型来表示。所谓双分量模型,就是假想每一杆件由两个平行的杆组成,一根是理想弹塑性杆(当杆端弯矩超出屈服弯矩 M_y 时,在该杆端出现塑性铰),另一根是弹性杆。如图 6.6 的弯矩-曲率图形所示,杆件的刚度 k 由弹性刚度分量 k_1 和弹塑性刚度分量 k_2 相加而成,即

$$k = k_1 + k_2$$

取图 6.7 所示的一根杆来推导一下其刚度矩阵,图中弯矩及位移均用增量表示。

杆单元弯矩、剪力与转角、位移之间的关系为

$$\begin{Bmatrix} \Delta M_i \\ \Delta M_j \\ \Delta Q_i \\ \Delta Q_j \end{Bmatrix} = [K] \begin{Bmatrix} \Delta \theta_i \\ \Delta \theta_j \\ \Delta \delta_i \\ \Delta \delta_j \end{Bmatrix} = p[K] \begin{Bmatrix} \Delta \theta_i \\ \Delta \theta_j \\ \Delta \delta_i \\ \Delta \delta_j \end{Bmatrix} + q[K] \begin{Bmatrix} \Delta \theta_i \\ \Delta \theta_j \\ \Delta \delta_i \\ \Delta \delta_j \end{Bmatrix} = [K_1] \begin{Bmatrix} \Delta \theta_i \\ \Delta \theta_j \\ \Delta \delta_i \\ \Delta \delta_j \end{Bmatrix} + [K_2] \begin{Bmatrix} \Delta \theta_i \\ \Delta \theta_j \\ \Delta \delta_i \\ \Delta \delta_j \end{Bmatrix} \quad (6.2)$$

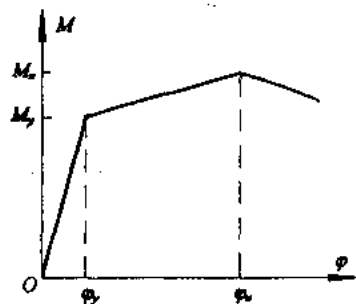


图 6.5 弯矩-曲率简化模型

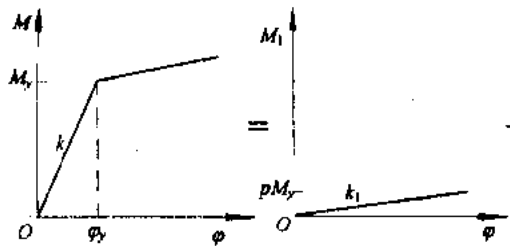


图 6.6 刚度双分量模型

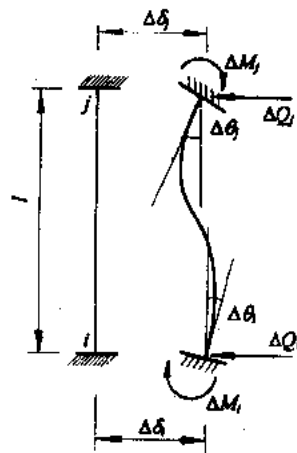


图 6.7 杆件受力-变形示意图

式中

$$[K_1] = [K_e] = \frac{k}{l} \begin{bmatrix} 4 & 2 & \frac{6}{l} & -\frac{6}{l} \\ 2 & 4 & \frac{6}{l} & -\frac{6}{l} \\ \frac{6}{l} & \frac{6}{l} & \frac{12}{l^2} & -\frac{12}{l^2} \\ -\frac{6}{l} & -\frac{6}{l} & -\frac{12}{l^2} & \frac{12}{l^2} \end{bmatrix} \quad (6.3)$$

$[K_2]$ 与杆端是否进入塑性有关,当杆的两端都未出现塑性铰时: $[K_2] = [k_e]$,其他情况:

(1) 当 i 端出现塑性铰,即 $|M_{2i}| \geq q|M_y| > |M_{2j}|$ 时,相应的弹塑性杆 i 端就成为铰接。这时,单元刚度矩阵为

$$[K_2] = [K_e] = \frac{k}{l} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 3 & \frac{3}{l} & -\frac{3}{l} \\ 0 & \frac{3}{l} & \frac{3}{l^2} & -\frac{3}{l^2} \\ 0 & -\frac{3}{l} & -\frac{3}{l^2} & \frac{3}{l^2} \end{bmatrix} \quad (6.4)$$

(2) 当 j 端出现塑性铰,即 $|M_{2i}| < q|M_y| \leq |M_{2j}|$ 时,相应的弹塑性杆 j 端就成为铰接。这时,单元刚度矩阵为

$$[K_2] = [K_e] = \frac{k}{l} \begin{bmatrix} 3 & 0 & \frac{3}{l} & -\frac{3}{l} \\ 0 & 0 & 0 & 0 \\ \frac{3}{l} & 0 & \frac{3}{l^2} & -\frac{3}{l^2} \\ -\frac{3}{l} & 0 & -\frac{3}{l^2} & \frac{3}{l^2} \end{bmatrix} \quad (6.5)$$

(3) 当 j 端出现塑性铰, 即 $|M_{2i}| \geq q|M_y|$, $|M_{2j}| \geq q|M_j|$ 时, 单元刚度矩阵为

$$[K_2] = 0 \quad (6.6)$$

2. 考虑二次矩的影响

由于框架结构相对来说受力变形较大, 在轴力 N 作用下, 将引起杆内弯矩的变化和位移的增长。在方程(6.1)中考虑二次矩的影响, 需增加一个几何刚度矩阵:

$$[K]\{\delta\} - [N]\{\delta\} = \{P\} \quad (6.7)$$

式中, $[N]$ 为几何总刚度矩阵。

几何总刚度矩阵 $[N]$ 是由单元几何刚度矩阵 $[N_0]$ 组成。单元几何刚度矩阵可以由假定的杆件挠曲线(假定为三次抛物线)按能量法推出。

(1) 当两端未出现塑性铰时, 单元几何刚度矩阵为

$$[N_0] = N \begin{bmatrix} \frac{2l}{15} & -\frac{l}{30} & \frac{1}{10} & -\frac{1}{10} \\ -\frac{l}{30} & \frac{2l}{15} & \frac{1}{10} & -\frac{1}{10} \\ \frac{1}{10} & \frac{1}{10} & \frac{6}{5l} & -\frac{6}{5l} \\ -\frac{1}{10} & -\frac{1}{10} & -\frac{6}{5l} & \frac{6}{5l} \end{bmatrix} \quad (6.8)$$

(2) 当 i 端出现塑性铰时, 单元几何刚度矩阵为

$$[N_0] = N \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & \frac{l}{5} & \frac{1}{5} & -\frac{1}{5} \\ 0 & \frac{1}{5} & \frac{6}{5l} & -\frac{6}{5l} \\ 0 & -\frac{1}{5} & -\frac{6}{5l} & \frac{6}{5l} \end{bmatrix} \quad (6.9)$$

(3) 当 j 端出现塑性铰时, 单元几何刚度矩阵为

$$[N_0] = N \begin{bmatrix} \frac{l}{5} & 0 & \frac{1}{5} & -\frac{1}{5} \\ 0 & 0 & 0 & 0 \\ \frac{1}{5} & 0 & \frac{6}{5l} & -\frac{6}{5l} \\ -\frac{1}{5} & 0 & -\frac{6}{5l} & \frac{6}{5l} \end{bmatrix} \quad (6.10)$$

(4) 当 i, j 两端出现塑性铰时, 单元几何刚度矩阵为

$$[N_0] = N \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -\frac{1}{l} & \frac{1}{l} \\ 0 & 0 & \frac{1}{l} & -\frac{1}{l} \end{bmatrix} \quad (6.11)$$

简化刚度法的优点是较为简便, 其缺点是与实际刚度相差较大。

6.3.2 实际刚度法^[25]

在以上所述的简化刚度法中, 将塑性铰出现前的整个杆单元按一个弹性刚度取值, 塑性铰假定出现在杆端, 用统一的模型化的 $M-\varphi$ 关系计算杆单元刚度矩阵。然而, 实际结构中, 塑性铰不一定都出现在杆端, 一些梁的塑性铰会产生在三分之一跨的集中荷载作用部位。

为了得到不同受力 - 变形情况下的杆单元刚度矩阵及相对精确的框架结构计算分析结果, 可采用实际刚度法来求杆单元的刚度, 实际刚度法是按框架的钢筋混凝土杆单元各截面的实际刚度出发, 推导各杆单元刚度矩阵。

计算单元刚度矩阵时, 将杆单元视为包含 n 小段的变刚度单元, 每段按截面的实际 $M-\varphi$ 关系赋予相应于弯矩的截面刚度 (图 6.8)

1. 刚度矩阵和等效荷载列阵

如图 6.8 所示, 杆单元 jk 沿杆长划分成相等的 n 个微段, 微段长度为 Δx 。设杆 k 端横向单位位移 $v_k = 1$, 弯矩沿杆长线性分布如图 6.8(b)。设任意微段的弯矩为 $M_i = \eta x_i$, η 为待定系数。由边界条件 $\varphi_k = 0$, 得

$$\varphi_k = \int \frac{M_i \bar{M}}{B(x)} dx \approx \sum_{i=1}^n \bar{M} \frac{M_i \Delta x}{B_i} = \eta \sum_{i=1}^n \frac{x_i \Delta x}{B_i} = 0 \quad (6.12)$$

式中 B_i 为微段刚度, 可由截面的 $N-M-\varphi$ 关系确定。设

$$A_i = \frac{\Delta x}{B_i}; \quad A = \sum_{i=1}^n \frac{1}{B_i} \Delta x = \sum_{i=1}^n A_i \quad (6.13)$$

$$S_0 = \sum_{i=1}^n \frac{\Delta x}{B_i} x_i = \sum_{i=1}^n A_i x_i = 0 \quad (6.14)$$

由式 (6.14) 可知 y 轴为 A_i 的形心轴, A_i 对 j 截面取矩:

$$\begin{aligned} S &= \sum_{i=1}^n \frac{\Delta x}{B_i} x_j = \sum_{i=1}^n \frac{\Delta x}{B_i} (c_1 + x_i) = c_1 \sum_{i=1}^n \frac{\Delta x}{B_i} + \sum_{i=1}^n \frac{\Delta x}{B_i} x_i \\ &= c_1 A + S_0 = c_1 A \\ c_1 &= \frac{S}{A}; \quad c_2 = l - c_1 \end{aligned} \quad (6.15)$$

又由边界条件 $v_k = 1$, 得

$$v_k = \int \frac{M_i \bar{M}}{B(x)} dx \approx \sum_{i=1}^n - (c_2 - x_i) \frac{\eta x_i \Delta x}{B_i}$$

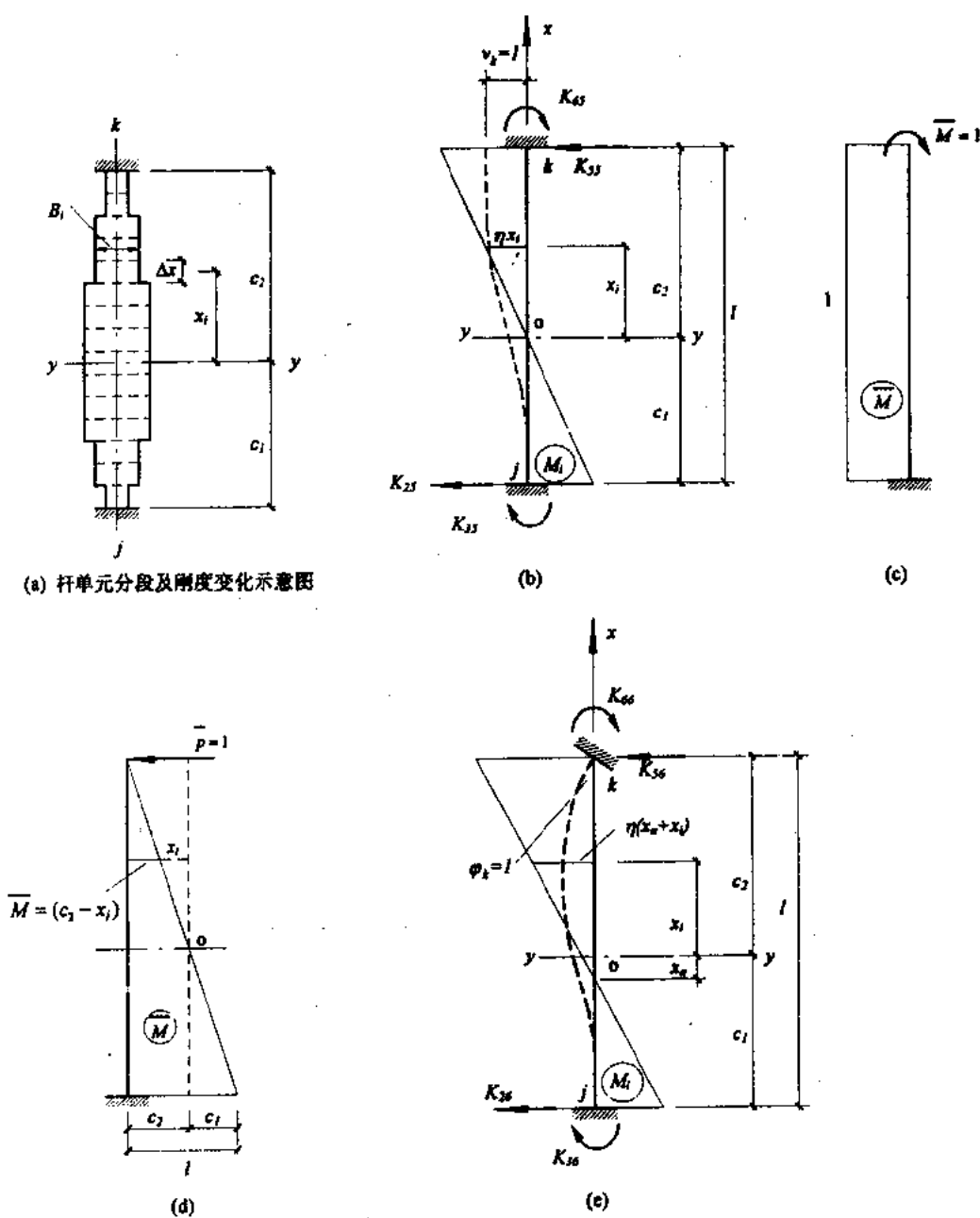


图 6.8

$$= \eta \sum_{i=1}^n \frac{x_i^2 \Delta x}{B_i} - \sum_{i=1}^n c_2 \eta \frac{x_i \Delta x}{B_i} = \eta I_y - 0 = \eta I_y = 1 \quad (6.16)$$

式中

由式(6.16),得

$$\eta = \frac{1}{I_y}; \quad M_i = \frac{1}{I_y} x_i$$

杆 k 端: $x_i = c_2$, $M = \frac{c_2}{I_y}$, 则刚度系数 $K_{65} = \frac{c_2}{I_y}$; 杆 j 端: $x_i = -c_1$, 则刚度系数 $K_{35} = \frac{c_1}{I_y}$; 相应地, $K_{55} = \frac{K_{65} + K_{35}}{l} = \frac{1}{I_y}$; $K_{25} = -K_{35} = -\frac{1}{I_y}$ 。

设杆 k 端单位转角 $\varphi_k = 1$, 弯矩沿杆长线性分布如图 6.8(e)。设任意微段的弯矩为 $M_i = \eta(x_a + x_i)$, η 为待定系数。由边界条件 $v_k = 0$, 及式(6.13)、(6.14)、(6.15) 得

$$\begin{aligned} v_k &= \int \frac{M_i \bar{M}}{B(x)} dx \approx \sum_{i=1}^n - (c_2 - x_i) \frac{\eta(x_a + x_i) \Delta x}{B_i} \\ &= \sum_{i=1}^n \frac{\eta x_a x_i \Delta x}{B_i} + \sum_{i=1}^n \frac{\eta x_i^2 \Delta x}{B_i} - \sum_{i=1}^n \frac{\eta c_2 x_a \Delta x}{B_i} - \sum_{i=1}^n \frac{\eta c_2 x_i \Delta x}{B_i} \\ &= \eta x_a S_0 + \eta I_y - \eta c_2 x_a A - \eta c_2 S_0 = \eta I_y - \eta c_2 x_a A = 0 \\ x_a &= \frac{I_y}{c_2 A} \end{aligned} \quad (6.17)$$

又由边界条件 $\varphi_k = 1$, 得

$$\begin{aligned} \varphi_k &= \int \frac{M_i \bar{M}}{B(x)} dx \approx \sum_{i=1}^n \frac{\eta(x_a + x_i) \Delta x}{B_i} \\ &= \eta \sum_{i=1}^n \frac{I_y \Delta x}{A c_2 B_i} + \eta \sum_{i=1}^n \frac{x_i \Delta x}{B_i} = \eta \frac{I_y}{A c_2} \sum_{i=1}^n \frac{\Delta x}{B_i} = \eta \frac{I_y}{c_2} = 1 \\ \eta &= \frac{c_2}{I_y} \end{aligned} \quad (6.18)$$

杆 k 端: $x_i = c_2$, $M = \eta(x_a + x_i) = \frac{c_2}{I_y} \left(\frac{I_y}{c_2 A} + c_2 \right) = \frac{1}{A} + \frac{c_2^2}{I_y}$, 则刚度系数 $K_{66} = \frac{1}{A} + \frac{c_2^2}{I_y}$; 杆 j 端: $x_i = -c_1$, $M = \eta(x_a + x_i) = \frac{c_2}{I_y} \left(\frac{I_y}{c_2 A} - c_1 \right) = -\frac{1}{A} + \frac{c_1 c_2}{I_y}$, 则刚度系数 $K_{36} = -\frac{1}{A} + \frac{c_1 c_2}{I_y}$;

同理可得 $K_{33} = \frac{1}{A} + \frac{c_1^2}{I_y}$; $K_{25} = -K_{35} = -\frac{1}{I_y}$ 。

设 a_i 为杆各微段的实际截面积, 在杆端加单位力 $N = 1$, 杆端位移为

$$\Delta = \sum_{i=1}^n \frac{l}{n E a_i} = \frac{1}{n E} \left(\sum_{i=1}^n \frac{1}{a_i} \right)$$

则杆单元的轴向刚度为

$$K_{11} = \frac{1}{\Delta} = \frac{n E}{l \left(\sum_{i=1}^n \frac{1}{a_i} \right)} \quad (6.19)$$

$$K_{44} = -K_{14} = -K_{41} = K_{11}$$

变刚度单元在局部坐标系中的刚度矩阵为

$$[\bar{K}]^e = \begin{bmatrix} \frac{nE}{l\left(\sum_{i=1}^n \frac{1}{a_i}\right)} & 0 & 0 & -\frac{nE}{l\left(\sum_{i=1}^n \frac{1}{a_i}\right)} & 0 & 0 \\ 0 & \frac{1}{I_y} & -\frac{c_1}{I_y} & 0 & -\frac{1}{I_y} & -\frac{c_1^2}{I_y} \\ 0 & -\frac{c_1}{I_y} & \frac{1}{A} + \frac{c_1^2}{I_y} & 0 & \frac{c_1}{I_y} & -\frac{1}{A} + \frac{c_1 c_2}{I_y} \\ -\frac{nE}{l\left(\sum_{i=1}^n \frac{1}{a_i}\right)} & 0 & 0 & \frac{nE}{l\left(\sum_{i=1}^n \frac{1}{a_i}\right)} & 0 & 0 \\ 0 & -\frac{1}{I_y} & \frac{c_2}{I_y} & 0 & \frac{1}{I_y} & \frac{c_2}{I_y} \\ 0 & -\frac{c_2}{I_y} & -\frac{1}{A} + \frac{c_1 c_2}{I_y} & 0 & \frac{c_2}{I_y} & \frac{1}{A} + \frac{c_2^2}{I_y} \end{bmatrix} \quad (6.20)$$

用矩阵位移法求解时,需将作用在杆单元上的各种荷载转换成等效结点荷载或固端力,变刚度杆单元在各种荷载作用下的固端力可以利用共轭梁上的虚荷载平衡条件求得(图 6.9)。

以作用于梁单元三分点处两个竖向集中荷载 $\lambda_1 P$ 和 $\lambda_2 P$ 为例。设任意微段的弯矩为

$$M_i = M_{0i} - (\gamma_0 + \eta x_i)$$

其相应的共轭梁虚荷载如图 6.9(b),由平衡条件 $\sum Y = 0$,得

$$\begin{aligned} \sum_{i=1}^n \frac{M_{0i} - (\gamma_0 + \eta x_i)}{B_i} \Delta x &= 0 \\ \sum_{i=1}^n \frac{M_{0i}}{B_i} \Delta x - \gamma_0 \sum_{i=1}^n \frac{\Delta x}{B_i} - \eta \sum_{i=1}^n \frac{x_i \Delta x}{B_i} &= \sum_{i=1}^n \frac{M_{0i}}{B_i} \Delta x - \gamma_0 A = 0 \end{aligned}$$

设 $R^v = \sum_{i=1}^n \frac{M_{0i}}{B_i} \Delta x$, 则

$$\gamma_0 = \frac{R^v}{A} \quad (6.22)$$

又由平衡条件 $\sum M = 0$, 得

$$\begin{aligned} \sum_{i=1}^n \frac{M_{0i} - (\gamma_0 + \eta x_i)}{B_i} \Delta x \cdot x_i &= \sum_{i=1}^n \frac{M_{0i} \Delta x}{B_i} x_i - \gamma_0 \sum_{i=1}^n \frac{\Delta x}{B_i} x_i - \eta \sum_{i=1}^n \frac{x_i^2 \Delta x}{B_i} \\ &= \sum_{i=1}^n \frac{M_{0i} \Delta x}{B_i} x_i - \eta I_y \end{aligned}$$

设 $M^v = \sum_{i=1}^n \frac{M_{0i} \Delta x}{B_i} x_i$, 则

$$\eta = \frac{M^v}{I_y} \quad (6.23)$$

根据式(6.21),杆端 j 的固端弯矩为

$$M_j = M_{0j} - (\gamma_0 + \eta x_j) = -\frac{R^v}{A} + \frac{M^v}{I_y} c_1 \quad (6.24)$$

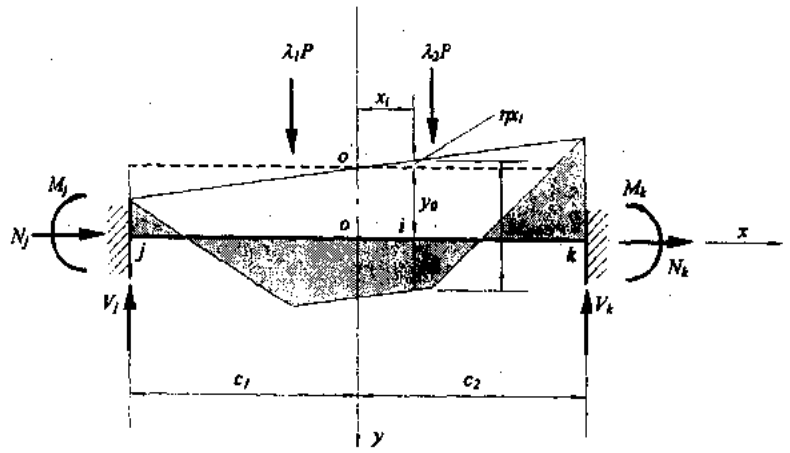
杆端 k 的固弯矩为

$$M_k = -M_{0k} + (\gamma_0 + \eta x_k) = \frac{R^v}{A} + \frac{M^v}{I_y} c_2 \quad (6.25)$$

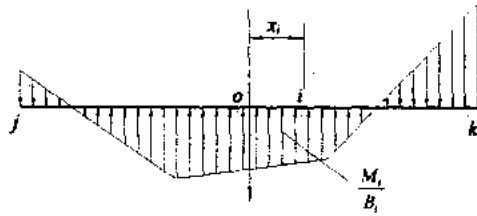
对任意点取矩, 可得杆端 j, k 的竖向固端力 V_j, V_k :

$$V_j = \frac{M^v}{I_y} - (2\lambda_1 + \lambda_2) \frac{P}{3} \quad (6.26)$$

$$V_k = -\frac{M^v}{I_y} - (\lambda_1 + 2\lambda_2) \frac{P}{3} \quad (6.27)$$



(a) 实梁的弯矩图



(b) 共轭梁虚荷载

图 6.9 变刚度杆单元固端力计算图示

则杆单元在局部坐标系中的结点等效荷载列阵可表示为

$$\{\bar{P}\}^e = \begin{Bmatrix} -N_j \\ -V_j \\ -M_j \\ -N_k \\ -V_k \\ -M_k \end{Bmatrix} = \begin{Bmatrix} 0 \\ -\frac{M^v}{I_y} + (2\lambda_1 + \lambda_2) \frac{P}{3} \\ \frac{R^v}{A} - \frac{M^c}{I_y} c_1 \\ \frac{M^v}{I_y} + (\lambda_1 + 2\lambda_2) \frac{P}{3} \\ -\frac{R^v}{A} - \frac{M^v}{I_y} c_2 \end{Bmatrix} \quad (6.28)$$

在框架全过程分析中,杆段的刚度是在变化的,从以上分析可以看出,杆单元的结点等效荷载的分配取决于杆段的相对刚度。因此,在增量法计算过程中结点等效荷载是在不断变化的,每一步计算后即需对结点等效荷载进行调整,调整方法是,本步计算完后,按杆段的实际刚度求结点等效荷载,将其与上一步求得的结点等效荷载相减,差值作为结点外荷载并入下一步计算:

$$\{\Delta \bar{P}\}^e = \{\bar{P}\}_i^e - \{\bar{P}\}_{i-1}^e \quad (6.29)$$

6.3.3 框架全过程分析中各弯矩 - 曲率段计算处理方法

1. 上升段(OA)

在第一次加变形或加荷载时,杆单元按混凝土弹性模量 E_c 计算刚度,以后均按“实际刚度”计算,即在求得杆单元各截面在上一级的弯矩的基础上,通过实际的 $M-\varphi$ 关系曲线求出各截面相对应的曲率,再由曲率来确定各截面的刚度。

2. 屈服点A附近

在屈服点附近(图 6.10 的 A 点),截面的刚度会有较大的变化,为了保持一定的计算精度,在屈服点附近的刚度计算可作以下处理[4]:如图 6.11 所示,当第 $i-1$ 次的 M_{i-1} 及第 i 次的 M_i 计算出来以后,计算第 $i+1$ 次的弯矩时,按刚度 B_i 求得的 M_G 可能太大而离

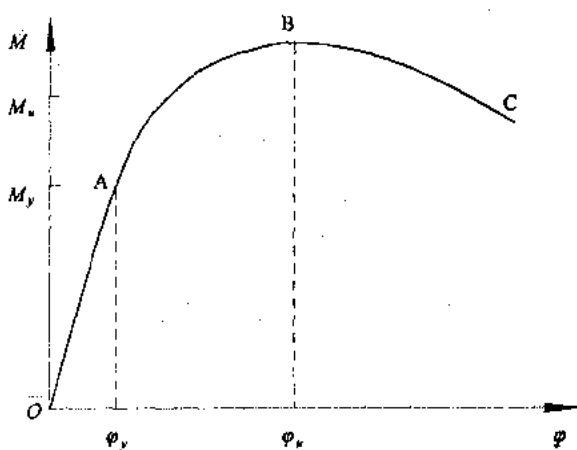


图 6.10 弯矩 - 曲率示意图

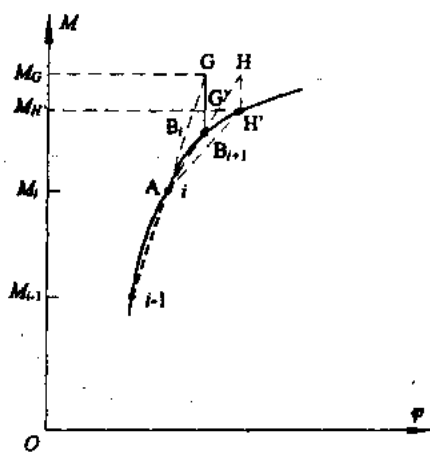


图 6.11 弯矩 - 曲率图中屈服点附近的处理

开了 $M-\varphi$ 曲线。这时,一方面要及时地把加变形的幅度减小,另一方面还要采取下列措施:把 G 点投影到 G' ,连接 AG' 作为新的刚度 B_{i+1} ,重新求得弯矩 M_H ,而把 M_G 作废。然后,再把 H 点投影到 H' , AH' 即作为下一步的刚度 B_{i+1} 。而高出 $M-\varphi$ 曲线的弯矩 HH' 值要作为作用在结点上的外荷载再求平衡。

以前在计算机运行速度相对较慢的情况下,常采用上述处理方法。目前计算机运行速度已大大提高,可采用减小计算步长,增加计算步数的方法来提高这一 $M-\varphi$ 区域的计算精度。计算精度可通过两种计算步长的计算结果比较来进行判定,若两种计算步长的计算结果基本相同,则说明计算精度满足要求。这样处理可使计算程序较为简捷。

3. 延性段(AB):

如图 6.10 所示,延性段的刚度随变形的增加而不断减小。除按前述方法沿 $M-\varphi$ 曲线前进外,当刚度小于某一很小值时(例如 $10^{-6} \text{kN} \cdot \text{m}^2$),刚度按此值取值,以免计算机除零溢出。

4. 下降段(AC):

某些杆段刚度、杆单元刚度进入负刚度,框架将逐步形成层间或整体几何可变机构, $P-\delta$ 曲线进入下降段,杆段一旦进入负刚度,单元刚度公式(6.20)就不能采用。因为在负刚度状态下能量原理已不再适用,单元刚度不能采用正刚度状态下的单元刚度求法,杆单元刚度矩阵只能通过给定杆端单位位移,经试算,求出杆端力的方法得到。

在第五章曾说过,当杆的某一段进入负刚度后(图 5.7、图 5.8),整个杆件就会开始卸载。同样,在框架结构中杆件的某一杆段进入负刚度,杆件将把它原来所承受的荷载卸给其它杆件。在计算分析中,这一过程的处理方法是:若本级加载后,某杆件的一杆段的弯矩到达极限,此刻杆端的内力为 $\{\bar{R}\}^e$ 。随后杆段进入负刚度,在不加荷载的情况下,杆件即产生变形,并卸载。计算出杆件产生卸载后的杆端内力,将其与本步计算所得杆端内力相减,差值作为外力,反方向加到杆端结点并入下一步计算。

根据图 6.12 杆件变形关系有

$$\phi_k = \phi_j - \varphi_a \cdot \Delta x + \int \left(\varphi - \frac{\Delta M}{B(x)} \right) dx \quad (6.30)$$

式中, φ_a 为杆的端部杆段软化变形后的曲率。

根据图 6.12,又有

$$\Delta M_j = M_j - M'_j = M_j - M_a \quad (6.31)$$

$$\Delta M_k = M_k - M'_k \quad (6.32)$$

式中 M_j ——杆端 j 卸载前弯矩;
 M'_j ——杆端 j 卸载后弯矩;
 M_a ——杆段 a 卸载后弯矩;
 M_k ——杆端 k 卸载前弯矩;
 M'_k ——杆端 k 卸载后弯矩。

$$\Delta M = \Delta M_j + \frac{\Delta M_k - \Delta M_j}{l} x \quad (\Delta x \leq x \leq l) \quad (6.33)$$

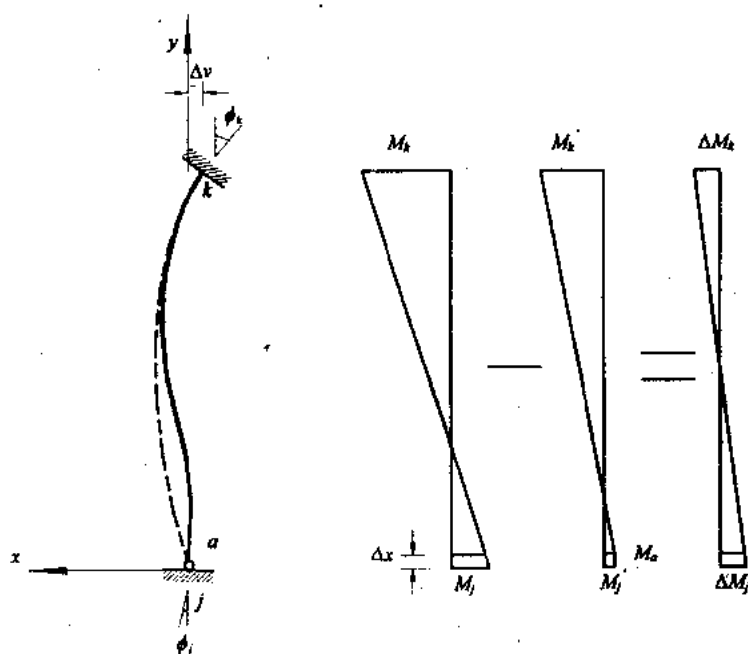


图 6.12 杆端屈服后弯矩变化情况

将式(6.33)代入式(6.30)

$$\begin{aligned}
 \phi_k &= \phi_j - \varphi_a \cdot \Delta x + \int \left[\varphi - \frac{1}{B(x)} \left(\Delta M_j + \frac{\Delta M_k - \Delta M_j}{l} x \right) \right] dx \\
 &= \phi_j - \varphi_a \cdot \Delta x + \sum_2^n \varphi_i \Delta x - \sum_2^n \frac{1}{B_i} \left(\Delta M_j + \frac{\Delta M_k - \Delta M_j}{l} x_i \right) \Delta x \\
 &= \phi_j - \varphi_a \cdot \Delta x + \sum_2^n \varphi_i \Delta x - \frac{1}{B_0} \Delta M_j (l - \Delta x) - \frac{\Delta M_k - \Delta M_j}{2B_0 l} (l^2 - \Delta x^2)
 \end{aligned} \quad (6.34)$$

在卸载时各杆段刚度取初始刚度, $B_i = B_0$ 。

$$\begin{aligned}
 v_k &= v_j - \varphi_a \cdot \Delta x \cdot l + \int \left[\varphi - \frac{1}{B(x)} \left(\Delta M_j + \frac{\Delta M_k - \Delta M_j}{l} x \right) \right] \bar{M} dx + \phi_j \cdot l \\
 &= v_j - \varphi_a \cdot \Delta x \cdot l + \int \left[\varphi - \frac{1}{B(x)} \left(\Delta M_j + \frac{\Delta M_k - \Delta M_j}{l} x \right) \right] \frac{(l-x)}{l} dx + \phi_j \cdot l \\
 &= v_j - \varphi_a \cdot \Delta x \cdot l + \sum_2^n \varphi \frac{(l-x_i)}{l} \Delta x - \sum_2^n \frac{\Delta M_j}{B_i} \frac{(l-x_i)}{l} \Delta x \\
 &\quad - \sum_2^n \frac{1}{B_i} (\Delta M_k - \Delta M_j) \frac{(l-x_i)x_i}{l^2} \Delta x + \phi_j \cdot l
 \end{aligned} \quad (6.35)$$

在试算时的步骤如下:先假定负刚度杆段 a 的弯矩 M_a , φ_a 由已有的 $M-\varphi$ 关系得到,由式(6.31)求得 ΔM_j ,再由式(6.34)求得 ΔM_k ,将 ΔM_k 代入式(6.35)求 v'_k ,将 v'_k 与实际 v_k 相比较,若不相等,则调整 M_a , φ_a ,重复上述计算步骤,直至 v'_k 与 v_k 相差小于一很小的规定值。

$$\Delta V_k = -\Delta V_j = \frac{\Delta M_k + \Delta M_j}{l} \quad (6.36)$$

将 $\Delta M_j, \Delta M_k, \Delta V_j, \Delta V_k$ 作为外力反向加载到杆端结点上。

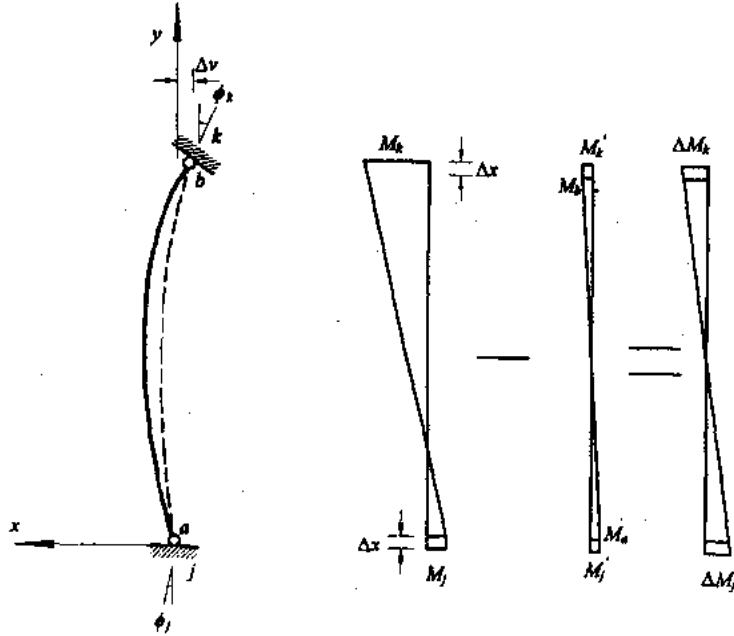


图 6.13 杆端屈服后弯矩变化情况

当杆的另一端端部杆段也到达极限强度开始卸载时(图 6.13),有

$$\Delta M_j = M_j - M_j' = M_j - M_a \quad (6.37)$$

$$\Delta M_k = M_k - M_k' = M_k - M_b \quad (6.38)$$

$$\begin{aligned} \phi_k &= \phi_j - \phi_a \cdot \Delta x + \int \left(\phi - \frac{\Delta M}{B(x)} \right) dx + \phi_b \cdot \Delta x \\ &= \phi_j - \phi_a \cdot \Delta x + \int \left[\phi - \frac{1}{B(x)} \left(\Delta M_j + \frac{\Delta M_k - \Delta M_j}{l} x \right) \right] dx + \phi_b \cdot \Delta x \\ &= \phi_j - \phi_a \cdot \Delta x + \sum_{i=1}^{n-1} \phi_i \Delta x - \sum_{i=1}^{n-1} \frac{1}{B_i} \left(\Delta M_j + \frac{\Delta M_k - \Delta M_j}{l} x_i \right) \Delta x + \phi_b \cdot \Delta x \end{aligned} \quad (6.39)$$

$$\begin{aligned} v_k &= v_j + \phi_a \cdot \Delta x \cdot l + \int \left[\phi - \frac{1}{B(x)} \left(\Delta M_j + \frac{\Delta M_k - \Delta M_j}{l} x \right) \right] \bar{M} dx - \frac{1}{2} \phi_b \cdot \Delta x^2 \\ &= v_j + \phi_a \cdot \Delta x \cdot l + \int \left[\phi - \frac{1}{B(x)} \left(\Delta M_j + \frac{\Delta M_k - \Delta M_j}{l} x \right) \right] \frac{(l-x)}{l} dx - \frac{1}{2} \phi_b \cdot \Delta x^2 \\ &= v_j + \phi_a \cdot \Delta x \cdot l + \sum_{i=1}^{n-1} \phi \frac{(1-x_i)}{l} \Delta x - \sum_{i=1}^{n-1} \frac{\Delta M_j}{B_i} \frac{(l-x_i)}{l} \Delta x \\ &\quad - \sum_{i=1}^{n-1} \frac{1}{B_i} (\Delta M_k - \Delta M_j) \frac{(l-x_i)x_i}{l^2} \Delta x - \frac{1}{2} \phi_b \cdot \Delta x^2 \end{aligned} \quad (6.40)$$

在试算时,先假定负刚度杆段 a 的弯矩 M_a, ϕ_a 由已有的 $M-\phi$ 关系得到,由式(6.38)和式(6.39)及 $M-\phi$ 关系,试算求得 M_b, ϕ_b ,将 ΔM_k 代入式(6.40)求 v_k' ,将 v_k' 与实际 v_k 相比

较,若不相等则调整 M_a, φ_a , 重复上述计算步骤,直至 v_k 与 v_k 相差小于一很小的规定值。最后,将 $\Delta M_j, \Delta M_k, \Delta V_j, \Delta V_k$ 作为外力反向加载到杆端结点上。

当杆端进入负刚度及卸载后,设杆端 k 产生一单位转角 $\bar{\phi}$ (实际计算时取一较小值,图 6.14),杆端 j 的弯矩由原来的 M_a 减小至 M'_a ,杆端 k 的弯矩由原来的 M_k 减小至 M'_k 。参照式 (6.30),有

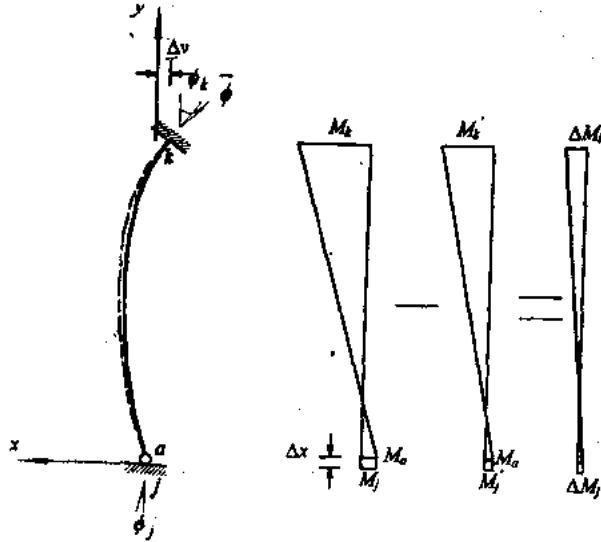


图 6.14 杆端变形增量引起弯矩变化情况

$$\phi_k + \bar{\phi} = \phi_j - \varphi_a \cdot \Delta x + \sum_2^n \varphi_i \Delta x - \frac{1}{B_0} \Delta M_j (l - \Delta x) - \frac{\Delta M_k - \Delta M_j}{2B_0 l} (l^2 - \Delta x^2) \quad (6.41)$$

与计算杆件卸载结点力步骤相同,由式(6.31)、(6.32)与式(6.41)求得 $\Delta M_j, \Delta M_k$, 则

$$K_{66} = -\Delta M_k \quad (6.42)$$

$$K_{36} = -\Delta M_j \quad (6.43)$$

采用同样计算方法可得到杆单元刚度矩阵各项系数。

在杆单元负刚度情况下进行计算时要注意以下两个方面:第一,要选取合适的虚拟单元刚度,刚度太小,则不能使计算结构的整体刚度大于零,若太大,则会增加计算的次数;第二,加虚拟单元后,结构实际承受的荷载是所施加的荷载减去虚拟单元所承受的荷载值;第三,在多层框架分析时,一般每层水平方向均加有虚拟单元。加载计算时,由于虚拟单元的影响,结构所承受侧向荷载与实际所需加载有一定差别,在计算时需不断调整各个楼层的加载数值。

6.3.4 实际刚度法中考虑二次矩的影响

在理想的情况下,单元几何刚度矩阵的各个系数可以通过假定的杆件挠曲线(例如三次抛物线)用能量法加以确定,当一端出现塑性铰时,其它部分仍当作弹性杆而导出刚度矩阵系数。然而实际的杆是一个变刚度构件,塑性铰具有一定的长度,为了给出较接近于

实际的解,可根据曲率通过数值积分或共轭梁法求出挠曲线,在每一次加位移或变形求得挠曲线以后,计算出单元几何刚度矩阵系数。在实际计算时也可采用下列近似方法来考虑二次矩:

- (1) 先不考虑二次矩影响,求出框架变形 $\{\delta\}$ 初值,并计算其实际挠度曲线;
 - (2) 求出轴力作用于已有位移框架而产生的各杆单元各截面的增量弯矩;
 - (3) 按照对应于增量弯矩的挠曲线,计算由 N 引起的固端弯矩和支座反力,将其并入该级荷载列矩阵;
 - (4) 重新求解基本方程,求出位移及内力。
- 由虚梁(图 6.15(c))得

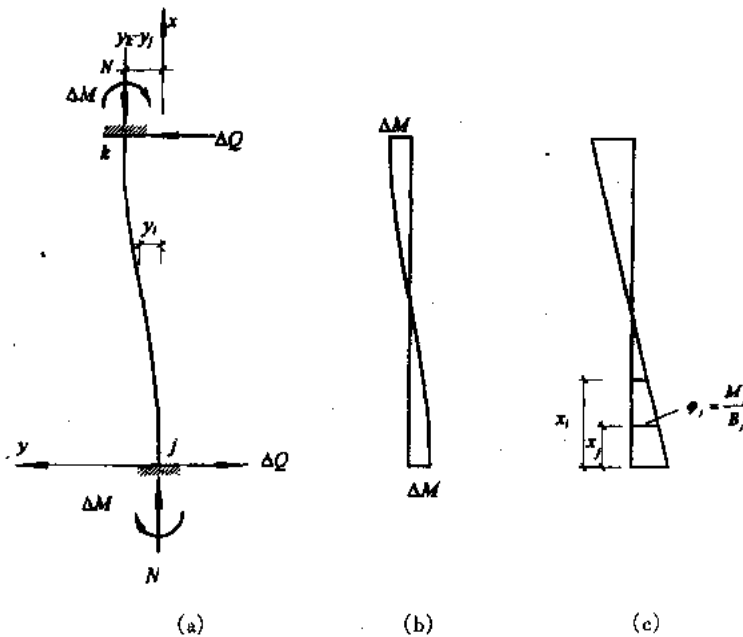


图 6.15 二次矩求解示意图

$$y_i \approx \sum_{j=1}^i \varphi_j \Delta x (x_i - x_j) \quad (6.44)$$

式中, φ_j 为上一步计算所得杆件曲率。

轴向力引起的二次弯矩

$$\Delta M_i = \Delta Q \cdot x_i + N \cdot y_i \quad (6.45)$$

又由 $\Delta y = 0$, 根据虚梁(图 6.15(c)), 可得

$$\Delta y = \sum_{i=1}^n \frac{\Delta M_i}{B_i} = \sum_{i=1}^n \frac{\Delta Q \cdot x_i + N \cdot y_i}{B_i} = \Delta Q \sum_{i=1}^n \frac{x_i}{B_i} + N \sum_{i=1}^n \frac{y_i}{B_i} = 0$$

$$\Delta Q = -N \sum_{i=1}^n \frac{y_i}{B_i} \bigg/ \sum_{i=1}^n \frac{x_i}{B_i} \quad (6.46)$$

$$\Delta M = \Delta Q \cdot l + N \cdot (y_k - y_j) \quad (6.47)$$

将 $\Delta Q, \Delta M$ 作为外荷载, 反向加到结点上。杆的各微段内力增加 ΔM_i 。

6.3.5 采用简化 $N-M-\varphi$ 关系曲线进行框架全过程分析^[25,26]

采用荷载增量法对框架结构全过程分析可求得 $P-\delta$ 曲线的下降段, 但在计算单元刚度时, 由于部分截面进入负刚度或刚度为零, 局部杆段软化, 未进入负刚度杆段则开始卸载, 求单元刚度系数的式(6.20)不再适用, 需采用试算方式求单位位移所对应的固端力作为刚度系数。可见, 其求解步骤非常复杂。即便是杆段在进入负刚度以前, 求各杆段的刚度 B_i 也需要占用很大的计算量。

若对 $M-\varphi$ 关系进行简化, 并忽略下降段, 采用增量法即荷载控制, 则框架结构全过程分析将大大简化, 且计算结果与试验也能较好地吻合。

图 6.16 为数值分析方法算得的某截面的 $N-M-\varphi$ 全过程曲线族。计算及试验表明, 受拉破坏(受弯及大偏心受压截面)的 $M-\varphi$ 关系曲线可以近似简化成图 6.17 所示的三折线。杆件在 3 个不同受力阶段的刚度可由下式确定:

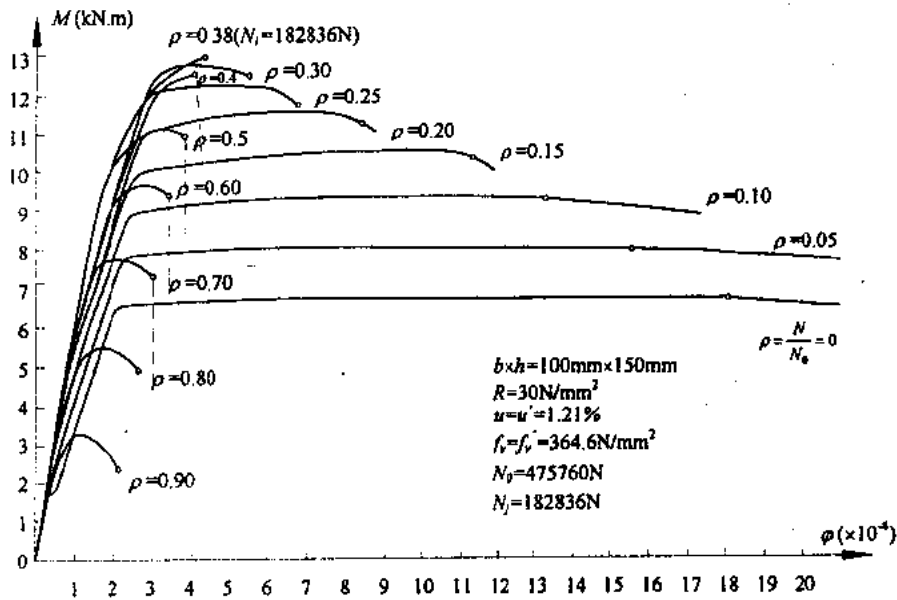


图 6.16 截面的 $N-M-\varphi$ 关系

$$\left. \begin{aligned} B_{\text{I}} &= \frac{M_c}{\varphi_c} \\ B_{\text{II}} &= \frac{M_y - M_c}{\varphi_y - \varphi_c} \\ B_{\text{III}} &= \frac{M_u - M_y}{\varphi_u - \varphi_y} \end{aligned} \right\} \quad (6.48)$$

式中: M_c, M_y, M_u 和 $\varphi_c, \varphi_y, \varphi_u$ 分别为杆件混凝土开裂、钢筋屈服、混凝土受压边缘达到极限应变 ϵ_u 时的弯矩和相应曲率。

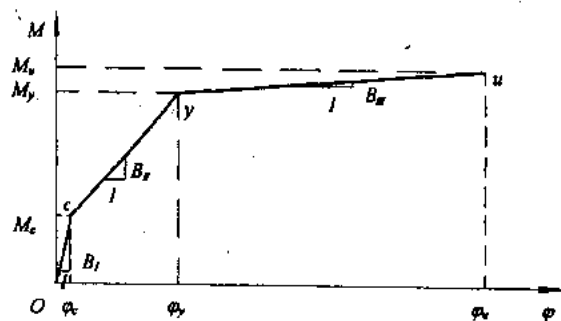


图 6.17 受拉破坏截面的简化 $M-\phi$ 曲线

1. 混凝土开裂的弯矩 M_c 和曲率 φ_c

$$M_c = \frac{\gamma f_t J_0}{y} + \frac{N J_0}{A_0 y} \quad (6.49)$$

$$\varphi_c = \frac{M_c}{0.85 E_c J_0} \quad (6.50)$$

式中 f_t ——混凝土极限抗拉强度；

A_0 ——换算截面积；

J_0 ——换算截面惯性矩；

y ——换算截面形心到受拉边缘的距离；

γ ——考虑混凝土截面受拉区弹塑性的系数,对于矩形截面,可近似取为 1.75。

当截面尺寸、混凝土强度等级及配筋确定后,每给定一个 N 值,即可由式(6.49)和式(6.50)求得 M_c 和 φ_c 。

2. 受拉钢筋屈服时的弯矩 M_y 和曲率 φ_y

当受拉钢筋达到屈服时,假定截面的应变及应力分布如图 6.18 所示。

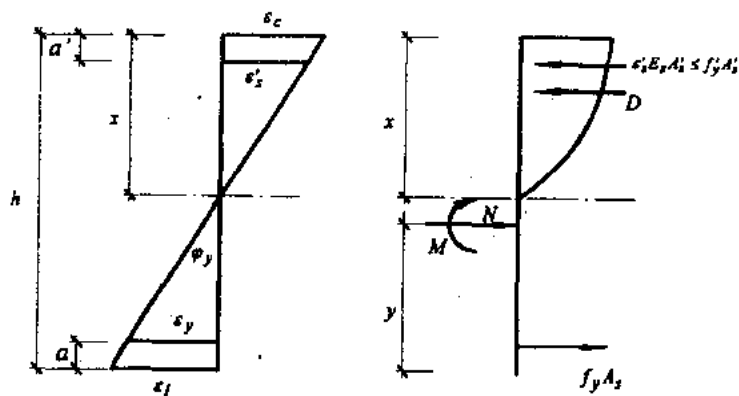


图 6.18 截面应力、应变分布

此时受拉钢筋的应变为 $\varepsilon_y = f_y / E_s$ 。如果假设受压区高度为 x ,则得

$$\varphi_y = \frac{\varepsilon_y}{h - a - s} \quad (6.51)$$

$$\varepsilon'_s = \varphi_y (x - a') \quad (6.52)$$

$$\varepsilon_c = \varphi_y x \quad (6.53)$$

$$D = \int b\sigma_c dx = b \sum_i \sigma_i \Delta x \quad (6.54)$$

$$N_s = D + \varepsilon'_s E_s A'_s - f_y A_s \quad (6.55)$$

对中和轴取矩,得

$$M_c = Dz = \int b\sigma_c x dx = b \sum_i \sigma_i x_i \Delta x \quad (6.56)$$

$$M_y = M_c + \varepsilon'_s E_s A'_s (x - a') + f_y A_s (h - x - a) + N(h - y - x) \quad (6.57)$$

根据公式(6.51) ~ (6.57),每给定一个 x 值更可得 M_y, φ_y 及相应的 N 。在框架分析中,通常是已知截面的 N 而要求计算相应的 M_y, φ_y 。其计算步骤与 5.3 中的求有轴力作用下的弯矩、曲率相似。

3. 混凝土受压边缘达到极限压应变 ε_u 时的弯矩 M_u 和曲率 φ_u

当混凝土受压边缘达到极限压应变 $\varepsilon_u = 0.004$ 时,就认为截面达到破坏状态,截面的应变分布如图 6.19 所示。混凝土受压应力图形近似地采用矩形,矩形应力图的高度取为 $0.85x$,矩形应力图的换算应力值 $\sigma_c = 0.77f_c$ 。

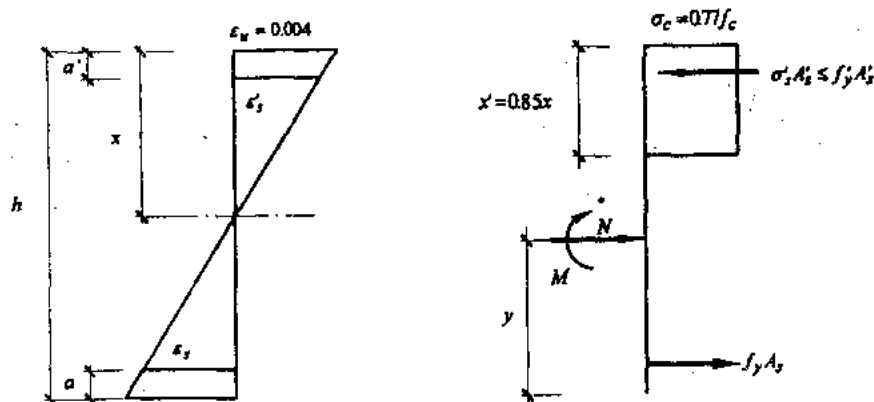


图 6.19 截面应力、应变分布

由截面上的平衡及变形条件可得

$$x' = 0.85x \quad (6.58)$$

$$bx'\sigma_c = f_y A_s - \sigma'_s A'_s + N \quad (6.59)$$

$$\sigma'_s = 0.004 E_s \frac{x - a'}{x} \quad (6.60)$$

由上三式每给定一个 N 值,便可解得相应的 x 及 σ'_s ,进而可求得 φ_u 和 M_u

$$\varphi_u = \frac{0.004}{x} \quad (6.61)$$

$$M_u = \sigma_c b x' \left(h - \frac{1}{2} x' - a \right) + \sigma_s' A_s' (h - a - a') - N(y - a) \quad (6.62)$$

值得注意的是:如公式(6.58)、(6.59)和(6.60)解得的 $\sigma_s' > f_y'$,则应取 $\sigma_s' = f_y'$ 直接由公式(6.58)和(6.59)求 x ,再按式(6.61)和式(6.62)求 φ_u 和 M_u 。

由图 6.16 可以看出,受压破坏截面(小偏心受压)的 $M-\varphi$ 关系曲线,从开始加荷起便逐渐弯曲,直至破坏,没有明显的转折点。通过对数值积分方法计算结果的分析,受压破坏截面(小偏心受压)的 $M-\varphi$ 关系可按下式计算^[20]

$$\varphi = \left(\frac{M}{M_u} \right) \varphi_e + \left(\frac{M}{M_u} \right)^5 (\varphi_u - \varphi_e) \quad (6.63)$$

式中 M_u 和 φ_u ——截面破坏时的弯矩和曲率;

φ_e 为截面破坏时曲率的弹性部分。

M_u 和 φ_u 可根据截面破坏时的平衡及变形条件求得(图 6.20)

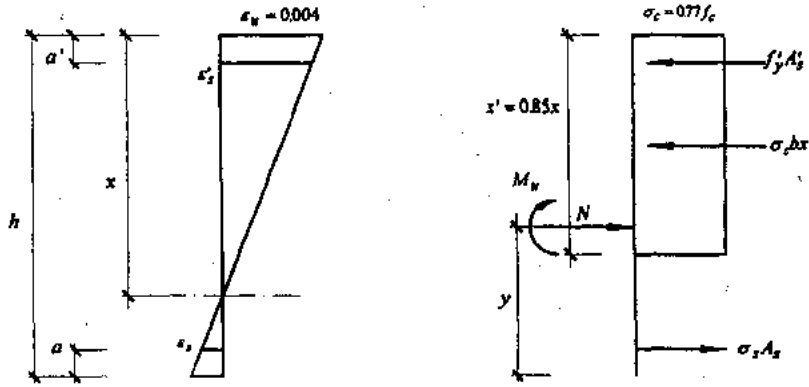


图 6.20 截面应力、应变分布

$$x' = 0.85x \quad (6.64)$$

$$\sigma_s = 0.004 E_s \left(\frac{h_0}{x} - 1 \right) \quad (6.65)$$

$$N = \sigma_c b x' + f_y' A_s' - \sigma_s A_s \quad (6.66)$$

由以上三式,给定轴力,便可解得 x ,而

$$\varphi_u = \frac{0.004}{x} \quad (6.67)$$

$$M_u = \sigma_c b x' \left(h_0 - \frac{x'}{2} \right) + f_y' A_s' (h_0 - a') - N(y - a) \quad (6.68)$$

在式(6.68)中当 $x' > h$ 时,应取 $x' = h$ 。

式(6.63)中的 φ_e 为截面破坏时曲率的弹性部分,可取为

$$\varphi_e = \frac{M_u}{0.85 E J_0} \quad (6.69)$$

大小偏心受压破坏的分界处,称为界限破坏,其破坏特征是受拉钢筋达到屈服时,压区混凝土也被压碎。此时,受拉钢筋应变为 $\epsilon_s = \epsilon_y = f_y / E_s$,混凝土受压边缘应变 $\epsilon_c = \epsilon_u$

= 0.004; 则相应的受压区高度 x_j 及曲率 φ_j 为

$$x_j = \frac{0.004(h-a)}{\epsilon_s + 0.004} \quad (6.70)$$

$$\varphi_j = \frac{0.004}{x_j} \quad (6.71)$$

利用公式(6.55)可算得界限破坏时的轴力 N_j 。界限破坏时的轴力 N_j 也可近似按下式求得

$$N_j = \sigma_c b x' - f_y A_s + f_y' A_s' \quad (6.72)$$

式中, $x' = 0.85x_j$ 。

在分析框架时, 当某一级荷载作用下载面的 M, N 求得后

如 $N < N_j$, 则按大偏心受压情况计算 $N-M-\varphi$ 关系, 确定截面刚度;

如 $N > N_j$, 则按小偏心受压情况计算 $N-M-\varphi$ 关系, 确定截面刚度。

6.3.6 结点刚域的影响

计算模型中框架梁、柱是一根理想的细杆, 梁、柱连接结点是一个点。而实际结构梁、柱连接结点是具有一定宽度和高度的区域。在实际计算中应视这一结点区为刚域, 梁、柱采用净跨作为单元计算长度进行微段划分。为了考虑结点刚域对框架变形和内力的影响, 必须将净跨部分的刚度矩阵 $[K_l]^e$ 移向刚端结点处, 形成带刚域的杆单元的刚度矩阵 $[K_r]^e$ 。

根据图 6.21 所示几何关系, 可以建立杆端位移和刚端位移的关系式:

$$\begin{Bmatrix} u_{ij} \\ v_{ij} \\ \theta_{ij} \\ u_{ik} \\ v_{ik} \\ \theta_{ik} \end{Bmatrix} = \begin{bmatrix} 1 & 0 & 0 & & & \\ 0 & 1 & -d_j & & 0 & \\ 0 & 0 & 1 & & & \\ & & & 1 & 0 & 0 \\ 0 & & & 0 & 1 & -d_k \\ & & & 0 & 0 & 1 \end{bmatrix} \begin{Bmatrix} u_{rj} \\ v_{rj} \\ \theta_{rj} \\ u_{rk} \\ v_{rk} \\ \theta_{rk} \end{Bmatrix} \quad (6.73)$$

$$\text{或简写成} \quad \{\delta_l\}^e = [H]\{\delta_r\}^e \quad (6.73)$$

式中 $\{\delta_l\}^e$ ——杆端位移列阵;

$[H]$ ——变换矩阵;

$\{\delta_r\}^e$ ——刚端位移列阵。

同理, 由图 6.22 可得杆件的刚端内力 $\{F_r\}^e$ 与杆端内力 $\{F_l\}^e$ 的关系:

$$\{F_r\}^e = [H]\{F_l\}^e \quad (6.74)$$

此外, 根据杆端、刚端结点力和结点位移关系:

$$[K_l]^e \{\delta_l\}^e = \{F_l\}^e \quad (6.75)$$

$$[K_r]^e \{\delta_r\}^e = \{F_r\}^e \quad (6.76)$$

由式(6.73) ~ 式(6.76), 经变换得到杆端刚度矩阵与刚端刚度矩阵的关系式:

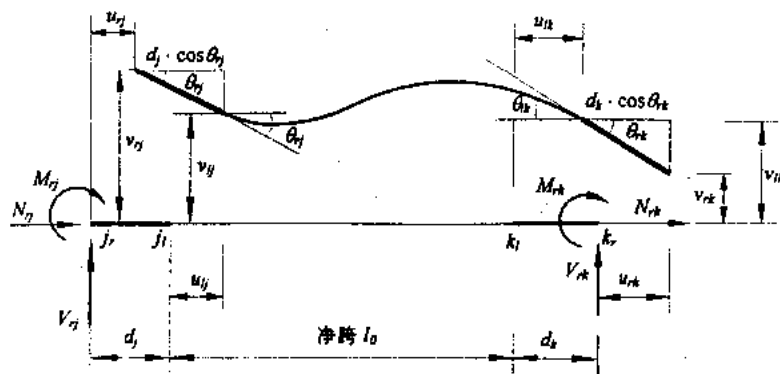


图 6.21 带刚域杆件的刚端内力及变形图

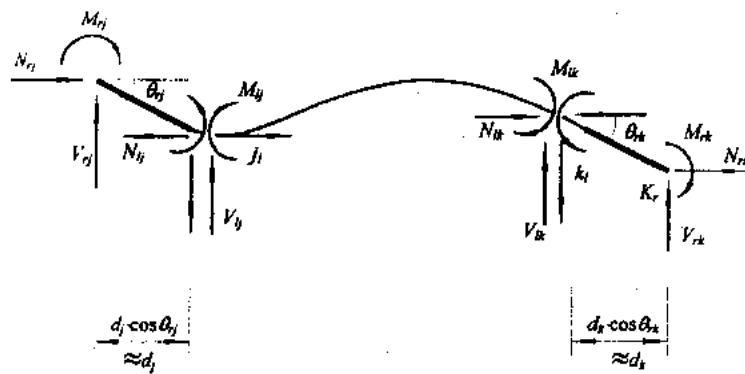


图 6.22 杆件的刚端内力与杆端内力的关系

$$[K_r]^e = [H]^T [K_f]^e [H] \quad (6.77)$$

在框架分析中,利用 $[K_r]^e$ 形成总刚度,通过解方程求得结点位移和每根杆的刚端位移 $\{\delta_r\}^e$,然后分别由式(6.73)和式(6.75)求杆端位移和杆端力。

综合以上所述,采用荷载增量法对框架进行全过程分析的计算框图见图 6.23。

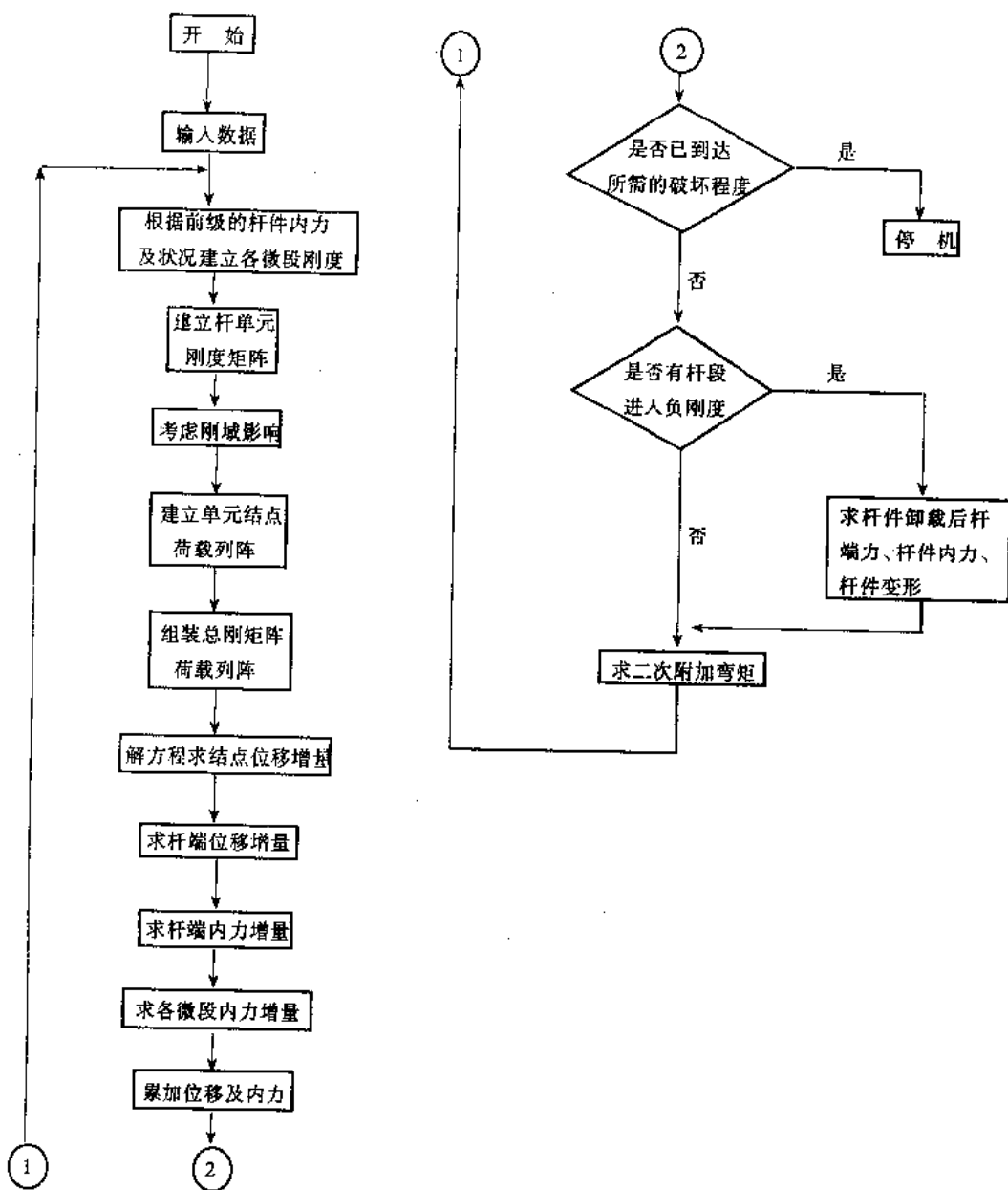


图 6.23 框架结构荷载增量法计算框图

第七章 钢筋混凝土剪力墙结构 有限元分析

7.1 概述

长期以来,人们采用解析的方法,将剪力墙简化成具有等效刚度的杆件,并按杆系和刚架在弹性假定或极限受力状态假定的基础上,对剪力墙结构进行内力分析和变形计算,这种分析方法采用了大量基于试验数据上的经验公式和弹性理论的简化推导,虽然这些经验公式和简化公式在宏观上能够反映剪力墙构件的部分非弹性性能,对常规设计来说也简便易行,但对于结构的详细研究分析显然存在不足之处,其局限性主要表现为:

(1) 只能从宏观上反映结构的弹性和极限状态时的性能,不能反映钢筋混凝土构件所特有的混凝土开裂、压碎、钢筋屈服等过程;

(2) 只能反映构件整体受力变形过程,不能较好地反映结构细部,如构件连接部位或刚度有较大变化部位的受力-变形性能;

(3) 适用于常规的剪力墙分析,不太适合于形状较复杂的剪力墙结构,如带有门窗洞口和竖缝剪力墙结构的受力分析。

随着有限元数值方法的发展和电子计算机技术的进步,有限元法在工程分析中得到了越来越广泛的应用,为克服传统分析方法的不足提供了条件。钢筋混凝土剪力墙有限元分析方法能给出结构和变形发展的全过程,能够描述或模拟混凝土裂缝的形成和扩展,以及结构的破坏过程、破坏形态及其滞回耗能特性;能够对剪力墙的极限承载能力作出评估;能够揭示出剪力墙结构的薄弱部位和环节,以利于结构优化设计。同时,它能广泛地适用于各种剪力墙结构类型和不同受力条件与受力过程分析,为人们对多种钢筋混凝土剪力墙结构类型特性的全面详细的认识提供了条件。因此,可以说钢筋混凝土非线性有限元分析不仅是一个面向工程设计的计算工具,更重要的是一个面向理论分析和研究的重要手段。

7.2 钢筋混凝土剪力墙非线性有限元分析的基本理论

7.2.1 钢筋混凝土剪力墙有限单元模型

由于剪力墙结构的受力主要是在其自身平面内,因此,其相应有限元模型采用平面应力单元。

钢筋混凝土结构有限元模型要考虑两种材料以及它们之间的相互关系。通常构成钢筋混凝土结构的有限元模型主要有第三章所述的三种形式:分离式、组合式和整体式。根据钢筋混凝土剪力墙结构配筋大体上均匀分布的特点,可采用整体式有限元模型。即假定钢筋弥散于整个有限单元中,并将单元视为连续均匀材料。混凝土和钢筋组合成一个

单元,其刚度矩阵表示为

$$[D] = [D_c] + \sum_i [D_s]_i \quad (7.1)$$

式中, $[D_c]$ 为混凝土刚度矩阵, $[D_s]_i$ 为第 i 方向等效分布钢筋的刚度矩阵。

7.2.2 反复加载下混凝土的应力-应变关系曲线

为了能在计算程序中得以有效运用,反复加载下混凝土应力-应变曲线的选用不仅要考虑到反映混凝土应力-应变关系的精确性,还要顾及程序实现的可能性、计算程序的效率以及计算机容量等情况。有关文献中^[4]虽然给出了混凝土在反复荷载下各区段的理论应力-应变曲线,但这些曲线要想被真正用于计算模型则非常难以实现,因为,在反复荷载下混凝土的应力-应变路径忽上忽下,跨越各个应力、应变区段,在计算过程中要对二次或二次以上应力-应变曲线进行追踪几乎无法进行。由于这个原因,许多已有的分析模型采用了分段直线来描述反复荷载下混凝土的应力-应变关系,其分析结果同样能与试验较好地吻合。综合多种因素,本书采用了图 7.1 所示的反复加载下混凝土应力-应变理想化曲线,其骨架曲线由曲线方程所组成,反复荷载应力-应变路径由折线加以描述^[38]。

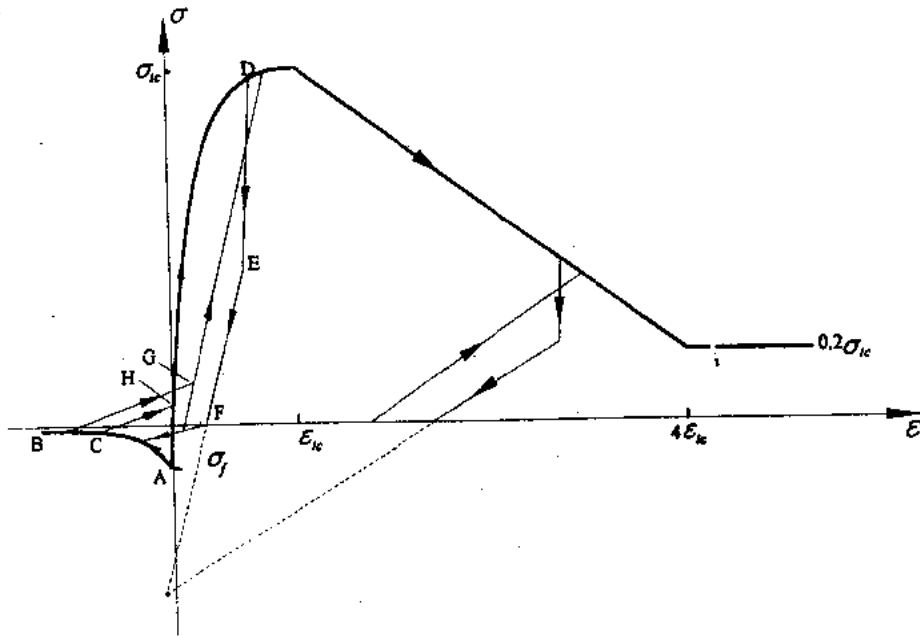


图 7.1 混凝土反复加载应力-应变关系模型

1. 压应变区域应力-应变骨架曲线

当 $\epsilon_i < \epsilon_{ic}$, 采用 Saenz 公式^[32]

$$\sigma_i = \frac{E_0 \epsilon_i}{1 + \left[\frac{E_0}{E_s} - 2 \right] \left(\frac{\epsilon_i}{\epsilon_{ic}} \right) + \left(\frac{\epsilon_i}{\epsilon_{ic}} \right)^2} \quad (7.2a)$$

$$E_i = \frac{d \sigma_i}{d \epsilon_i} = \frac{E_0 \left[1 - \left(\frac{\epsilon_i}{\epsilon_{ic}} \right)^2 \right]}{\left[1 + \left(\frac{E_0}{E_s} - 2 \right) \left(\frac{\epsilon_i}{\epsilon_{ic}} \right) + \left(\frac{\epsilon_i}{\epsilon_{ic}} \right)^2 \right]^2} \quad (7.2b)$$

式中 E_0 ——原点切线弹性模量；

E_s ——相应于最大压应力 σ_{ic} 的割线模量 $E_s = \sigma_{ic} / \epsilon_{ic}$ ；

ϵ_{ic} ——相应于最大压应力 σ_{ic} 的单向受力应变。

当 $\epsilon_i > \epsilon_{ic}, \epsilon_i < 4\epsilon_{ic}$

$$\sigma_i = \sigma_{ic} - \frac{\sigma_{ic} - 0.2\sigma_{ic}}{\epsilon_{ic} - 4\epsilon_{ic}} (\epsilon_{ic} - \epsilon_i) \quad (7.3a)$$

$$E_i = \frac{\sigma_{ic} - 0.2\sigma_{ic}}{\epsilon_{ic} - 4\epsilon_{ic}} \quad (7.3b)$$

当 $\epsilon_i > 4\epsilon_{ic}$, 取

$$\sigma_i = 0.2\sigma_{ic} \quad (7.4a)$$

$$E_i = 0 \quad (7.4b)$$

2. 拉应变区域骨架曲线

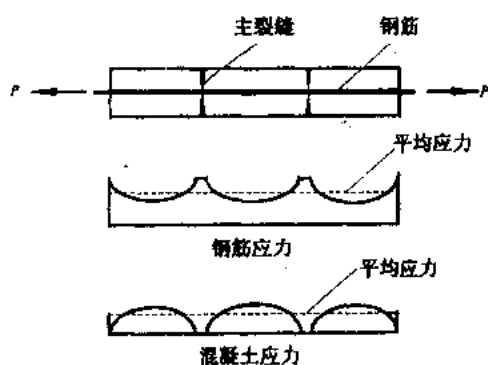


图 7.2 开裂后混凝土和钢筋应力分布

已有的计算模型通常假定混凝土开裂后,开裂单元内垂直于裂缝方向的混凝土弹性模量为零,混凝土不再承担拉应力,拉应力下降至零,而由钢筋负担全部受拉作用。混凝土原来承担的拉应力化作残余应力或超额应力,并转换成等效结点力,作为结点外荷载重新施加到整个结构上进行计算,求得新的平衡。然而,由于钢筋的存在,钢筋混凝土的实际开裂过程是,当混凝土应力达到受拉强度后,混凝土产生裂缝,混凝土和钢筋间产生滑移,由于钢筋表面和混凝土间的粘结和摩擦作用,裂缝之间的混凝土犹如附着在钢筋

上的小弹簧和钢筋一起承受拉应力,从一段钢筋混凝土开裂区间来看,开裂后钢筋混凝土所受应力可看作由图 7.2 中的平均钢筋受拉应力和平均混凝土受拉应力组合而成。随着受拉应变的增加,混凝土裂缝逐渐增多,混凝土和钢筋间的滑移充分发展,裂缝间混凝土中的拉应力逐步趋于零。

受拉破坏钢筋混凝土的混凝土等效应力-应变试验曲线如图 7.3 所示^[33]。

根据试验曲线本书建立了混凝土开裂后等效平均应力-应变关系模型:

$$E_i = - \alpha \frac{\sigma_i - 0.1\sigma_{ft}}{\sigma_{ft}} E_0 \quad (7.5a)$$

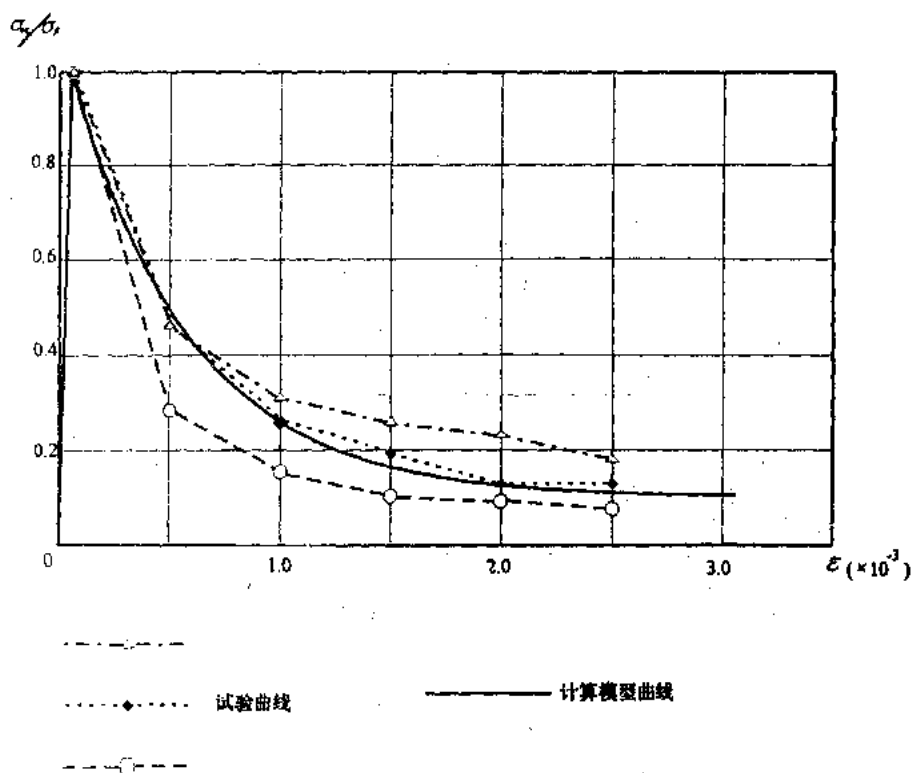


图 7.3 混凝土受拉等效平均应力-应变曲线

$$\sigma_i = \int E_i d\epsilon = - \int \alpha \frac{\sigma_i - 0.1\sigma_{ft}}{\sigma_{ft}} E_0 d\epsilon_i \quad (7.5b)$$

与图 7.3 试验曲线相拟合, 取 $\alpha = 0.1$ 。

上述混凝土开裂后的应力-应变关系模型构成图 7.1 中混凝土开裂后的应力-应变曲线 A-B, 用来作为对混凝土开裂后的应力-应变关系的描述。它通过平均应力的变化反映了混凝土与钢筋间的粘结和滑移的影响。

根据混凝土开裂前的线弹性假定及式(7.5), 拉应变区域骨架曲线定义为

当 $\epsilon_i > \epsilon_{i0}$

$$E_i = E_0 \quad (7.6a)$$

$$\sigma_i = E_0 \epsilon_i \quad (7.6b)$$

当 $\epsilon_i < \epsilon_{i0}$

$$E_i = -0.1 \frac{\sigma_i - 0.1\sigma_{ft}}{\sigma_{ft}} E_0 \quad (7.7a)$$

$$\sigma_i = \int E_i d\epsilon = -0.1 \int \frac{\sigma_i - 0.1\sigma_{ft}}{\sigma_{ft}} E_0 d\epsilon_i \quad (7.7b)$$

式中, σ_{ft} 为混凝土抗拉强度, ϵ_{i0} 为混凝土受拉达到开裂强度时对应的拉应变。

3. 压应变压域卸载曲线(以下曲线用切线模量加以描述)

当 $\sigma_i > 0.5\sigma_{i,\max}$, 图 7.1 中 D—E 段

$$E_i = E_0$$

式中, $\sigma_{i,\max}$ 为混凝土曾达到的最大等效单向受压应变所对应的压应力。

当 $0 < \sigma_i < 0.5\sigma_{i,\max}$, 图 7.1 中 E-F 段

$$E_i = \frac{\sigma_i - 4\sigma_{ft}}{\epsilon_i - 4\epsilon_{t0}} \quad (7.8)$$

当 $\sigma_i < 0$, 混凝土未曾开裂

$$E_i = \frac{\sigma_i - \sigma_{ft}}{\epsilon_i - \epsilon_{t0}} \quad (7.9)$$

当 $\sigma_i < 0$, 混凝土曾开裂

$$E_i = \frac{\sigma_i - \sigma_{i,\min}}{\epsilon_i - \epsilon_{i,\min}} \quad (7.10)$$

式中 $\sigma_{i,\min}$ —— 混凝土曾达到的最大受拉应变所对应的拉应力;

$\epsilon_{i,\min}$ —— 混凝土曾达到的最大受拉应变。

4. 压应变区域再加载曲线

$$\text{设 } E' = \frac{\sigma_{1.1m} - \sigma_i}{\epsilon_{1.1m} - \epsilon_i}$$

式中

$$\epsilon_{1.1m} = 1.1\epsilon_{i,\max} \quad \text{当 } \epsilon_{1.1m} \geq 0.05\epsilon_{ic}$$

$$\epsilon_{1.1m} = 0.05\epsilon_{ic} \quad \text{当 } \epsilon_{1.1m} \leq 0.05\epsilon_{ic}$$

$\sigma_{1.1m}$ 为混凝土受压骨架曲线中 $\epsilon_{1.1m}$ 所对应的应力。

当 $E' \geq \frac{\sigma_{1.1m} - 4\sigma_{ft}}{\epsilon_{1.1m} - 4\epsilon_{t0}}$ 时, 取

$$E_i = \frac{\sigma_{1.1m} - \sigma_i}{\epsilon_{1.1m} - \epsilon_i} \quad (7.11)$$

当 $E' < \frac{\sigma_{1.1m} - 4\sigma_{ft}}{\epsilon_{1.1m} - 4\epsilon_{t0}}$ 时, 则

当 $E' > 0.01E_0$ 时, 取

$$E_i = 0.01E_0 \quad (7.12)$$

当 $E' \leq 0.01E_0$ 时, 取

$$E_i = \frac{\sigma_{1.1m} - \sigma_i}{\epsilon_{1.1m} - \epsilon_i} \quad (7.13)$$

5. 拉应变区域卸载曲线

当 $\sigma_i < 0$ 时:

$$\left. \begin{aligned} E_i &= \frac{\sigma_a - \sigma_i}{\epsilon_a - \epsilon_i} \\ \sigma_a &= -\epsilon_{t0}E_0 \\ \epsilon_a &= -\epsilon_{t0} \end{aligned} \right\} \quad (7.14)$$

当 $\sigma_i > 0$ 时:

$$E_i = 0.01E_0 \quad (7.15)$$

6. 拉应变区域再加载曲线

$$E_i = \frac{\sigma_i - \sigma_{i,\min}}{\epsilon_i - \epsilon_{i,\min}} \quad (7.16)$$

7.2.3 反复加载下钢筋应力 - 应变关系

钢筋的反复荷载下应力 - 应变关系采用双折线理想弹塑性模型,并考虑钢筋屈服硬化,钢筋屈服后取 $E_s' = 0.01E_{s0}$ 。

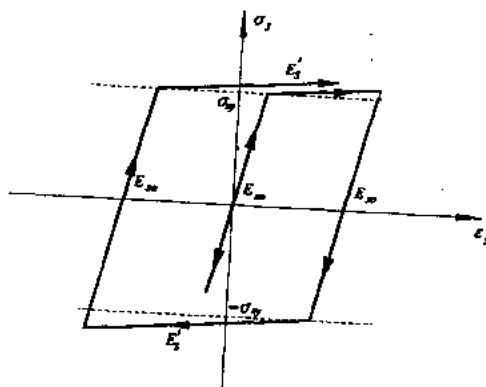


图 7.4 钢筋应力 - 应变关系模型

7.2.4 混凝土双向受力的本构关系

目前所采用的钢筋混凝土有限元本构模型大致可分为下列几种类型:

- (1) 以弹性理论为基础的线弹性和非线性弹性的本构关系;
- (2) 以经典塑性理论为基础的理想弹塑性和弹塑性硬化本构模型;
- (3) 采用断裂理论和塑性理论结合的塑性断裂理论,并考虑用应变空间建立的本构模型;
- (4) 粘性材料的本构关系发展起来的内时理论描述的本构模型;
- (5) 损伤理论和弹塑性损伤断裂混合建立的本构模型。

线弹性本构模型的应力 - 应变成线性关系,不能反映混凝土的实际应力 - 应变关系。非线性弹性混凝土本构关系模型建立在弹性理论基础上,应力 - 应变不成正比,但有一一对应关系。其它本构模型虽然很适合描述混凝土在一定的应力应变区域内的受力变形状况,或混凝土某一特定性能的应力应变状况,但目前还没有一种能全面反映混凝土在不同应力应变状态和不同应力应变发展过程下的本构关系模型。因此,在不同的应力应变区域,需要采用不同的本构关系模型。一些模型虽然具有理论上的可行性,但在实际应用方面会遇到许多数值计算和参数运用上的困难。几个复杂的模型组合在一起运用往往难以实现。综合上述情况,本章采用了 Darwin 和 Pecknold 提出的增量式正交混凝土本构模型,

它属非线性弹性本构模型,材料的常数根据试验采用当量单轴应力 - 应变关系来确定。

混凝土双向受力的本构关系详见第二章的有关内容。

7.2.5 钢筋单元本构关系

钢筋在有限元中可以被看作弥散于整个单元的一维连续介质(图 7.5), i 方向上的应力 - 应变关系为

$$d\sigma_{si} = \rho_i E_{si} d\epsilon_{si} \quad (7.17)$$

式中, ρ_i 为 i 方向配筋率。

i 方向上的刚度矩阵表示为

$$[D_s]_i = \begin{bmatrix} \rho_i E_{si} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (7.18)$$

由 i 方向局部坐标转至 xy 坐标

$$[D_s]_i = [T]^T [D_s]_i [T] \quad (7.19)$$

式中, $[T]$ 为局部坐标至整体坐标的转换矩阵:

$$[T] = \begin{bmatrix} \cos^2 \alpha & \sin^2 \alpha & \sin \alpha \cos \alpha \\ \sin^2 \alpha & \cos^2 \alpha & -\sin \alpha \cos \alpha \\ -2\sin \alpha \cos \alpha & 2\sin \alpha \cos \alpha & \cos^2 \alpha - \sin^2 \alpha \end{bmatrix} \quad (7.20)$$

上式的 α 值,当从 i 方向旋转至 x 轴为顺时针方向时取正号。

7.2.6 开裂钢筋混凝土单元的处理

带裂缝工作是混凝土结构的主要特性之一。混凝土的开裂会引起结构的刚度变化和降低以及应力的重分布,在混凝土的受力变形中占有较大比重。因此,对钢筋混凝土裂缝的计算模型设计合理与否是能否正确地进行结构分析的关键之一。

钢筋混凝土有限元分析中考虑裂缝的模型主要有分离式裂缝(discrete cracking)和弥散裂缝(smear cracking)模型两种,根据剪力墙裂缝分布的特点,以及计算程序的可行性,宜采用弥散裂缝模型对剪力墙的开裂过程加以描述。这里还需对已有的弥散裂缝模型增加以下假定:

(1) 某一单元内的高斯数值积分点上的拉应力达到开裂强度,则认为这一高斯积分点垂直于该主拉应力方向的混凝土开裂。开裂后的钢筋混凝土单元仍符合正交异性材料的假定。

(2) 混凝土开裂后考虑混凝土的拉伸强化效应(Tension Stiffening),开裂法向方向的切线模量采用图 7.1 中的 A—C 下降段,即 E_t 为负刚度。随着开裂方向应变加大,混凝土中原有的拉应力逐步释放。

(3) 在反复荷载作用下,开裂的混凝土裂缝有可能重新闭合,还有可能在裂缝的交叉方向形成第二条裂缝。第二条裂缝的开裂方向受到钢筋分布量与分布方向的影响和第一条裂缝的开展状况的影响,不一定和第一条裂缝完全相垂直^[34],但偏移量不大。考虑到程

序计算的效率和计算机容量等因素,一般可以假定第二条裂缝的开裂方向与第一条裂缝的开裂方向相垂直。

根据以上假定,在计算分析过程中,混凝土所处的开裂状态有以下7种(图7.6)。

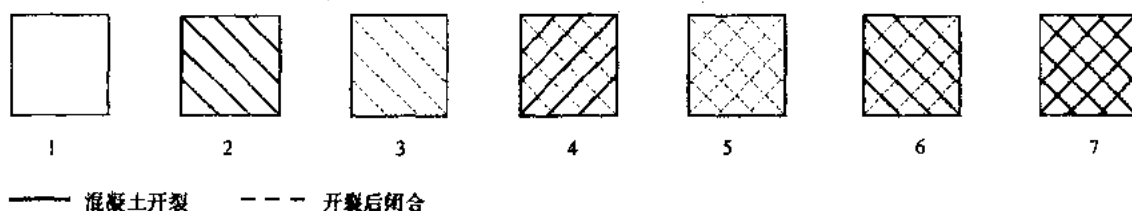


图 7.6 混凝土开裂模式

7.2.7 钢筋混凝土开裂后的剪切刚度

混凝土达到其抗拉强度出现开裂后,原来由混凝土承担的剪应力转而由混凝土开裂面的骨料咬合、摩擦和裂缝间的钢筋进行传递,这样钢筋混凝土有限元的剪切刚度会大大下降。

1. 混凝土开裂面的剪切刚度

混凝土开裂后的剪力传递能力取决于开裂面的几何性质、开裂面上的应力状况、开裂面错动情况等多种因素。这些因素目前还研究得很不充分,已有的开裂混凝土的剪力传递理论模型为数很少而且比较粗糙。反复剪切受力下的混凝土开裂面力学性能的研究较少。

根据钢筋混凝土开裂面的剪力传递机理,并作初步近似简化,在有限元分析中可以假定^[5]:

- (1) 混凝土剪切刚度仅与裂缝宽度有关;
- (2) 由剪切位移所引起的附加拉力可以由相应的钢筋承担;
- (3) 剪力的摩擦性质可以不考虑,在剪力和相对滑移之间存在着确定的对应关系。

由于这里的计算模型采用了“弥散裂缝”模型,开裂面的相对剪切位移将视为单元的剪应变,并相对地把开裂面之间的咬合作用视为单元内的剪应力与剪应变之间的关系。在早期的研究中,为简便起见,常采用的折算剪切模量 G' 来代表裂面之间的咬合作用:

$$G' = \beta G$$

式中, G 为混凝土开裂时刻前的剪切模量; β 为常数 ($0 \leq \beta \leq 1$)。然而,这一做法显得比较粗糙。由于开裂面的咬合作用主要与裂缝宽度有关。因此,将 G' 视为开裂单元的假想正应变(垂直于裂缝,表征裂缝宽度)的函数,将更能符合实际情况。

由于没有足够合理的关系可利用,AL-Manaidi^[35] 建议在开裂混凝土单元的本构关系中采用

$$G' = \frac{0.4G}{\epsilon_i / \epsilon_{i0}} \quad \text{当 } \epsilon_i \leq \epsilon_{i0} \text{ 时} \quad (7.21)$$

式中, G 为混凝土开裂时刻前的剪切模量; ϵ_i 为垂直于裂缝面的平均拉应变; ϵ_{i0} 为开裂时刻前的拉应变。

J. Houde^[36] 等建议采用

$$G' = \frac{G}{1 + k\omega^2}$$

式中, k 为参数, 且

$$\omega = (\epsilon_1 + \gamma\sigma_2/E) \cdot l_n$$

式中, ϵ_1 为垂直于裂缝的主应变; σ_2 为平行于裂缝的正应力; l_n 为通过单元形心且垂直于裂缝的线段在单元范围内的长度。

结合本章节的计算模型的特点, 本书选用了 AL-Manaidi 建议的开裂混凝土刚度模型, 即式(7.21)。

由于上式是被用来作为单调荷载作用下的开裂混凝土的剪切刚度模型, 而在反复荷载作用下, 混凝土的开裂面可能还会闭合, 因此对式(7.21) 需进行修正, 在混凝土开裂后, 当 $\epsilon_{i0} < \epsilon_i < 0$, 即开裂面重新趋向闭合, 剪切模量由 $0.4G \sim G$ 之间作线性插值

$$G' = \left(1 - \frac{0.6\epsilon_i}{\epsilon_{i0}}\right) G \quad (7.22)$$

当混凝土裂缝闭合后, 即 $\epsilon_i > 0$, 剪切模量按式(7.18) 取值。对于双向开裂混凝土, 认为该单元混凝土不再传递剪力, 剪切刚度为零。这里所采用的混凝土剪切刚度计算模型中未考虑反复受剪作用下混凝土剪切刚度的退化。

2. 开裂面钢筋的剪力传递

钢筋混凝土有限元开裂面上钢筋对剪切力的传递主要由同开裂面斜交的钢筋和通过开裂面钢筋的销栓力两部分组成, 对于前一种剪力传递作用已在钢筋单元模型中得以考虑。而后一种剪力传递作用即钢筋的销栓力作用, 由于其作用机理较复杂, 且对剪切刚度的影响相对来说较小^[33], 因此, 这里未考虑其影响。

7.2.8 非线性方程求解

鉴于钢筋混凝土应力 - 应变关系的复杂性以及反复荷载下加载过程的多变性, 一般情况下可采用增量法对非线性有限元方程进行求解, 方程的基本形式为

$$[K_{i-1}]\{\Delta\delta_i\} = \{\Delta R_i\}; (i = 1, 2, 3 \cdots) \quad (7.23)$$

式中, $[K_{i-1}]$ 为前一级荷载增量后的结构切线刚度矩阵。

7.2.9 加卸载判别

由于钢筋和混凝土在反复荷载作用下具有加载和卸载不同的应力 - 应变关系, 即加、卸载材料具有不同弹性模量。因此, 需要对加、卸载状况进行判别。在计算程序中需设置单元的加、卸载状态参数, 将主应力方向压应变增量设定为正向加载, 主应力方面拉应变增量设定为负向加载。在计算开始阶段, 材料为弹性状态, 即无加、卸载不同的弹性模量情况下, 求出结构向一侧水平加载时对应的各单元内主应力方向是拉应变、还是压应变, 以此来确定主应力方向的加、卸载状态参数, 在以后的计算步骤中, 以加、卸载状态参数作依据, 采用相应的加载或卸载弹性模量。当改变加荷方向时(从卸载开始), 将加、卸载状态参数作反向调整, 即原主应力方向加载过程变为卸载过程。由于钢筋混凝土开裂、屈服等非

线性应力 - 应变关系的存在,单元内的加、卸载状态不可能完全由初始状态来确定,一个单元的压屈或开裂往往会改变相邻单元原有的加、卸载状态,因此,在计算过程中还要将计算所求得的实际主应力方向加、卸载状态同加、卸载状态参数所预计的情况进行比较,若不一致则对状态参数进行调整,并取相应调整后的切线刚度模量再次进行计算,在调整过程中加、卸载状态往往会产生摆动,如果摆动多次而无其它影响产生,可以认为在此计算步骤中两种状态差别很小,可再继续下一步计算。

钢筋混凝土剪力墙结构按平面应力问题进行有限元非线性分析的计算框图如图 7.7 所示。

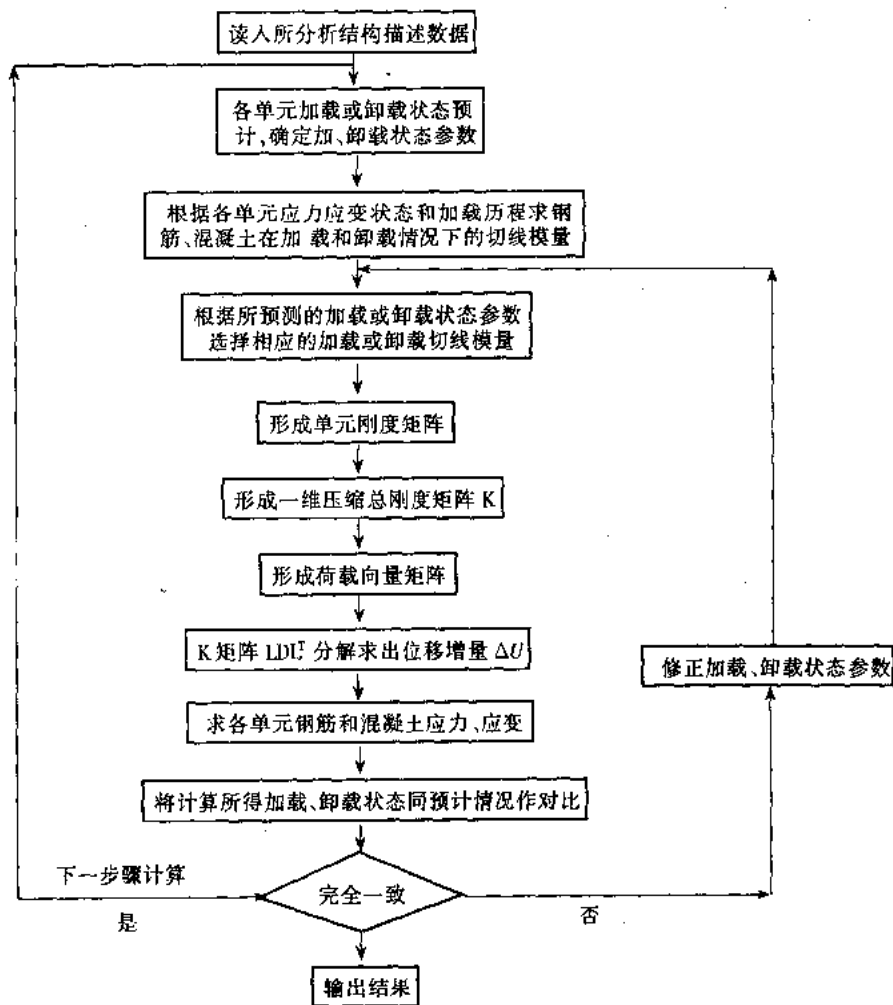


图 7.7 程序流程图

根据上述步骤用 C 语言编写的钢筋混凝土剪力墙结构非线性有限元分析的程序见附录 A。该程序除了用于钢筋混凝土剪力墙结构的分析以外,同样也可以用于钢筋混凝土平面应力状态下的结构和构件,例如梁、柱、框架结构、框支剪力墙结构等的单调或反复荷载作用下的分析。

7.3 钢筋混凝土剪力墙有限元分析实例

现以两个四层带竖缝剪力墙为例,说明钢筋混凝土剪力墙分析的全过程。

带竖缝剪力墙试验模型为两个 1/4 比例的四层带竖缝墙体^[37],图 7.8 所示。两个模型的外形尺寸和墙肢配筋完全相同,只是连系梁的高度和配筋不同。其中模型 SM-1 的连系梁尺寸为 $75\text{ mm} \times 50\text{ mm}$,配筋率约为 3%;模型 SM-2 的连系梁尺寸为 $75\text{ mm} \times 150\text{ mm}$,配筋率为 1.51%。混凝土圆柱体抗压强度和弹性模量分别为 $f'_c = 34.64\text{ MPa}$; $E_c = 3.6 \times 10^4\text{ MPa}$;钢筋屈服强度和弹性模量分别为 $\sigma_s = 342\text{ MPa}$; $E_s = 2.0 \times 10^5\text{ MPa}$ 。

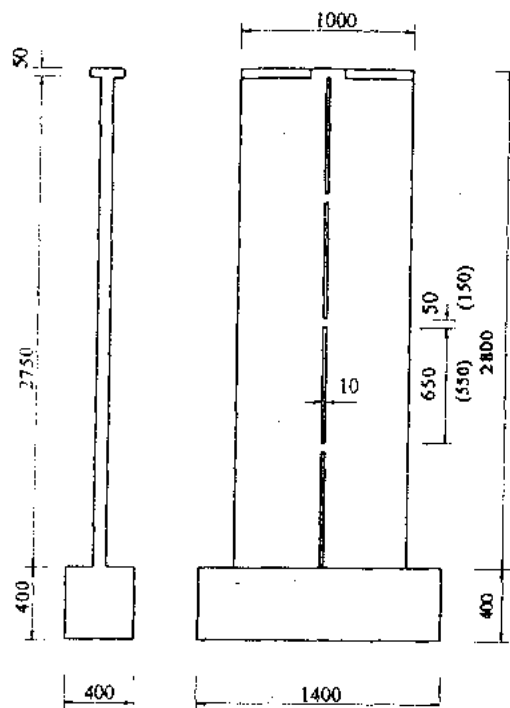


图 7.8 试验模型(括号内数值为模型 SM-2 尺寸)

模型 SM-1 在水平荷载作用下,第一层的连系梁与墙面的交界处首先出现裂缝,并发展为向墙内延伸的裂缝,这一点与双肢剪力墙的连系梁的开裂形式不同。随着水平荷载的增加,受拉边底部墙肢出现水平开裂,随后裂缝沿墙高度逐步增加,第二、第三层以及顶层连系梁与墙面的交界处接连开裂,墙面上的裂缝逐渐由边缘向中间延伸并不断扩展。到达极限荷载时,所有连系梁受压劈裂和剥落,受压墙肢出现水平开裂,其外侧底部边缘混凝土压碎,出现一小部分混凝土剥落(图 7.9(a))。

模型 SM-2 在水平向反复荷载作用下,其首次开裂也出现在第一层的连系梁与墙面的交界处,随后受拉边底部墙肢出现水平开裂。由于受水平荷载的反复作用,模型的裂缝左右比较对称,随着反复水平荷载峰值的增加,裂缝沿墙面高度逐步增加,第二、第三层连系梁与墙面的交界处相继开

裂。到达极限荷载时,由于连系梁尺寸较大,混凝土未被压碎。剪力墙破坏时以底部墙肢的过大变形和受压墙肢底部边缘混凝土压碎而破坏。

根据图 7.8 所示的带竖缝剪力墙的几何尺寸和配筋情况,本书给出了图 7.10 的有限元计算模型,共计 404 个单元,476 结点。在模型的连系梁部位应力相对较为集中,应力变化较大,计算模型对此周围的单元划分进行了细化。模型 SM-1 的连系梁较模型 SM-2 的小,在对模型 SM-1 分析时将有限元连系梁部位的几个单元的厚度设置为零。这样同样的单元、结点编号可用于两个模型的分析,以利于节省数据输入的工作量。

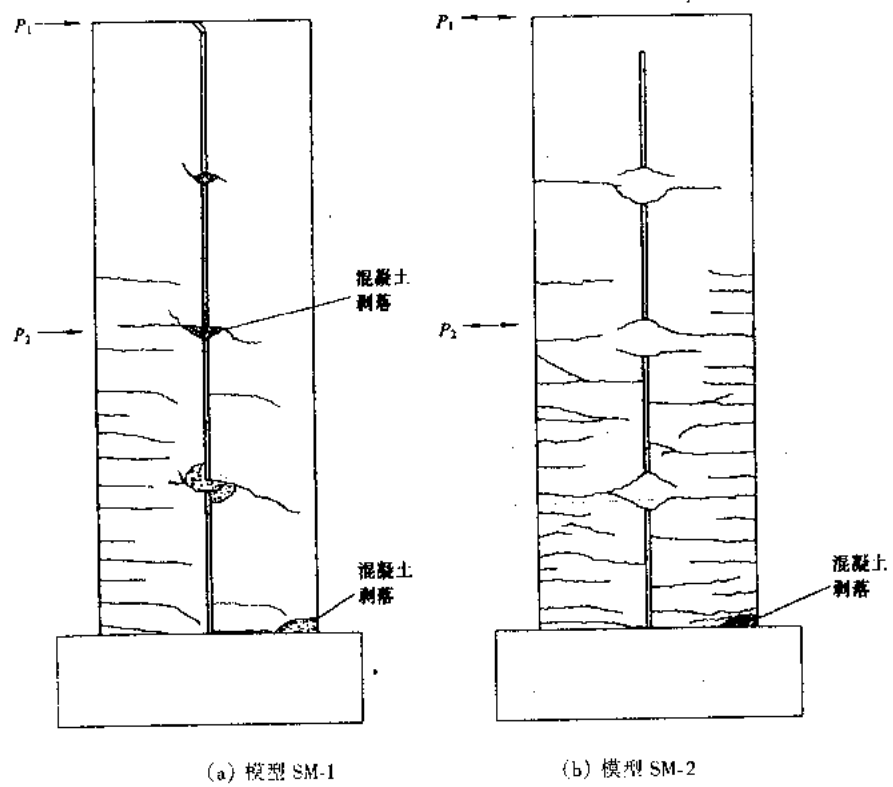


图 7.9 带竖缝剪力墙开裂和破坏模式

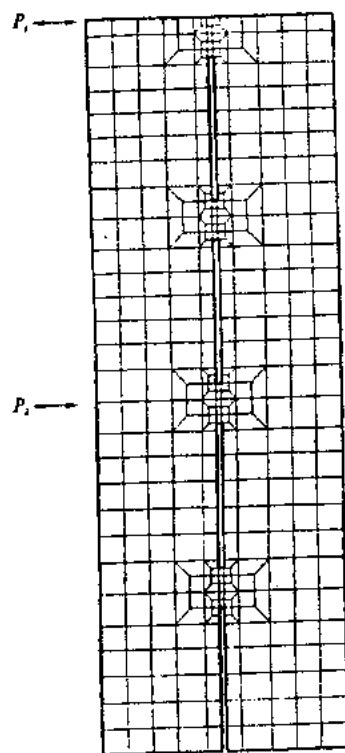


图 7.10 有限元计算模型

7.3.1 单调水平荷载作用下的带竖缝剪力墙非线性有限分析

按照单调荷载作用下的带竖缝剪力墙试验加载方式,计算模型中将 $P_1 = 2/3P$ (P 为总荷载) 的水平侧向荷载加在结构的顶部, $P_2 = 1/3P$ 水平侧向荷载加在结构的 $1/2$ 高度处,与试验时的加载情况完全一致。

按此比例加载下,当 P_1 到达 13kN 时,第一层连系梁与墙体连接处混凝土开裂; P_1 为 14kN 时,第二、三、顶层连系梁与墙体连接处混凝土开裂; P_1 为 12kN 时,受拉墙肢外侧底部单元出现开裂; P_1 到达 20kN 时,墙体的弯曲开裂延伸至受压墙肢图 7.12(a);当 P_1 到达 30kN 时,受拉墙肢外侧钢筋屈服,同时第一、二层连系梁处的混凝土单元受压方向斜对角高斯积分点上的混凝土应力 - 应变曲线进入下降段,意味着混凝土被压碎。当 P_1 到达 35kN 时,受压墙肢外侧混凝土单元压碎,结构破坏。结构顶部荷载 P_1 的计算值与试验值的对比如表 7.1 所示,图 7.11 为荷载 - 位移移曲线,图 7.12 给出了各级荷载情况下墙体开裂情况,即开裂过程。

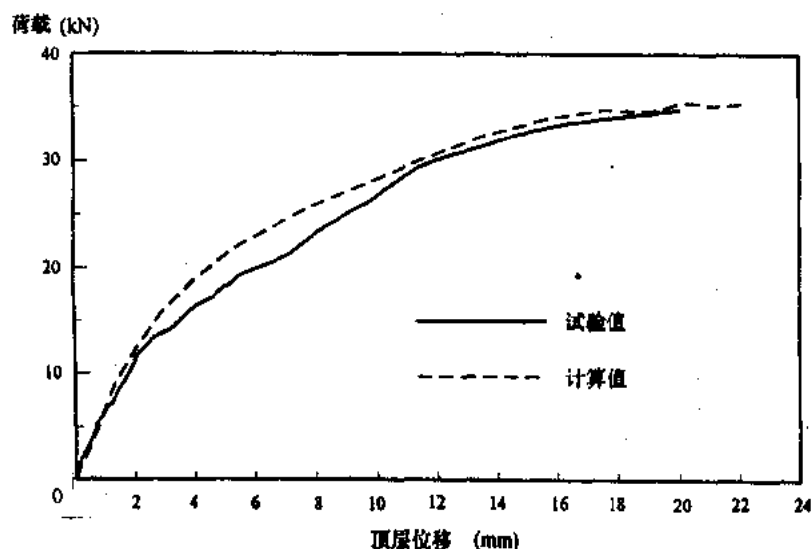


图 7.11 带竖缝剪力墙单调加载受力 - 位移曲线

表 7.1 带竖缝剪力墙单调加载计算和试验结果

项 目	一层连系梁开裂	二层连系梁开裂	三层连系梁开裂	顶层连系梁开裂	受拉墙肢开裂	受压墙肢开裂	受拉墙肢钢筋屈服	受压墙肢压碎
计算值(kN)	13	14	14	14	15	20	30	35
试验值(kN)	15	17	18	—	16	20	26	33

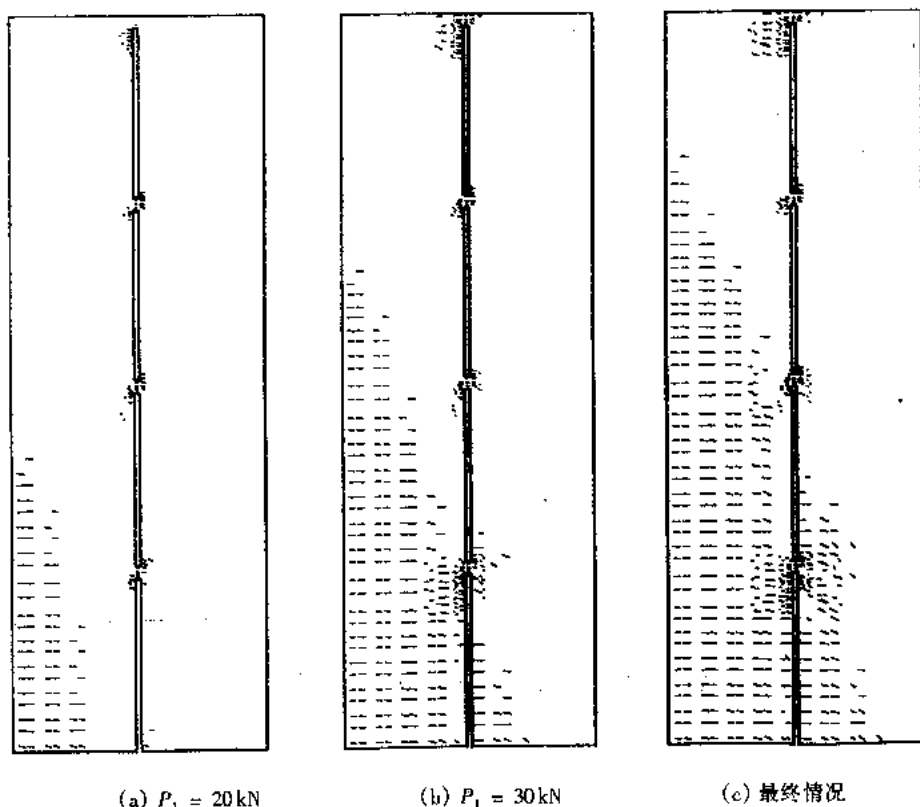


图 7.12 单调加载情况下带竖缝剪力墙开裂过程

7.3.2 反复荷载作用下的带竖缝剪力墙非线性有限元分析

按照试验加载方式和加载步骤,将 $P_1 = 2/3P$ 侧向荷载加在结构顶部, $P_2 = 1/3P$ 侧向荷载加在结构 $1/2$ 高度处。按 $P_1 = \pm 10 \text{ kN}, \pm 15 \text{ kN}, \pm 20 \text{ kN}, \pm 30 \text{ kN}, \pm 40 \text{ kN}$ 作反复加载。试验与计算得到的滞回曲线分别如图 7.13、图 7.14 所示。在第一循环加载后 ($\pm 10 \text{ kN}$), 左右墙肢外侧底边两个单元 1 ~ 2 个高斯积分点处出现开裂(试验测得可见水平裂缝为 P_1 达到 15 kN), 当 P_1 向右边载到 15 kN 时第一层的连梁出现斜裂缝(试验为 $P_1 = 15 \text{ kN}$ 时)。随着荷载值的增加, 墙肢水平裂缝由下而上逐步增多。 P_1 达到 17 kN 时第二层连梁出现斜裂缝(试验为 $P_1 = 20 \text{ kN}$ 时)。 P_1 达到 20 kN 时的墙体开裂情况见图 7.15(a), 图 7.15(b) 为计算所得的最终开裂情况 ($P_1 = \pm 40 \text{ kN}$), 开裂主要是下部墙肢的水平裂缝和连梁处的斜向裂缝。对于墙肢的水平裂缝计算结果和试验结果基本上一致。连梁处的斜向裂缝, 计算和试验得出的开裂方向基本相同, 试验产生的连梁裂缝由墙梁连接处开始并向墙肢斜向开展, 而计算所得裂缝较集中在连梁之上和连梁左右。计算和试验得出的模型最终破坏形式都为受拉墙肢边缘钢筋首先屈服, 受压墙肢受压区高逐步减小, 边缘混凝土达到极强度, 混凝土压碎, 钢筋压屈, 结构侧向刚度迅速减少, 受力 - 变形曲线接近水平。

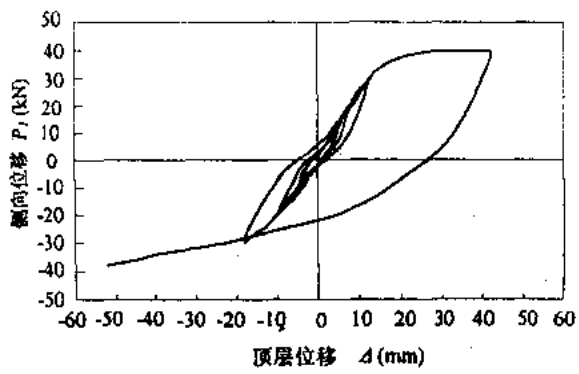


图 7.13 试验滞回曲线

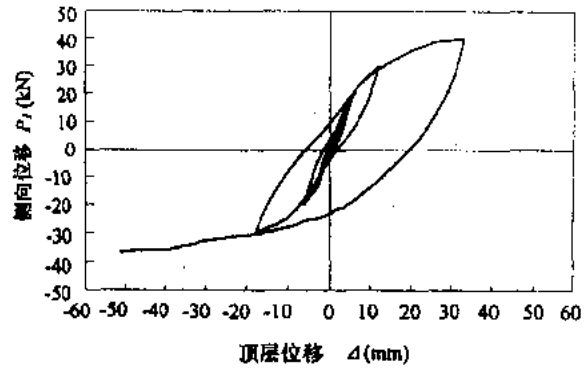
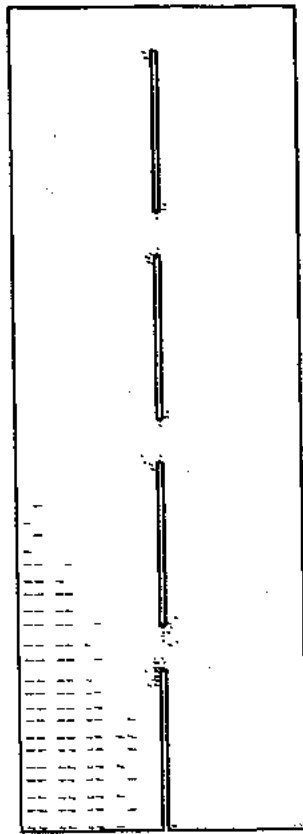
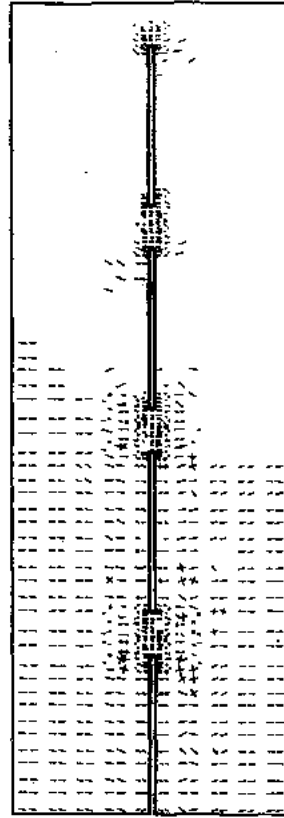


图 7.14 计算滞回曲线



(a) $P_l = 20\text{kN}$ 开裂图(向右)



(b) 最终开裂图

图 7.15 反复荷载下带竖缝剪力墙开裂情况

第八章 钢筋混凝土结构动力有限元分析

结构动力分析的主要目的是求得结构本身的动力特性及结构在动力荷载作用下的内力和变形全过程。在结构工程中,动力荷载主要包括以下几个方面:

(1) 地震作用 地震对结构的作用是通过地面运动而使结构物产生振动,当振动过大时,将会引起结构开裂、破坏甚至倒塌。进行结构抗震动力分析,可以研究结构的地震破坏机理,为结构抗震设计提供依据。

(2) 风荷载 风是空气从气压大的地方向气压小的地方流动而造成的,风速较大时会使结构产生较大的变形和振动,在结构抗风设计中,应考虑脉动风的动力作用,脉动风的强度是随时间无规则变化的。

(3) 机器振动引起的结构或构件的振动 例如,安装于楼面上的机器,在运行过程中,将不同程度地引起结构物的周期振动;重型锻压机器及打桩过程的振动也会引起结构物的过大振动。

(4) 爆炸冲击引起的结构振动 例如,核爆炸对地面及地下结构物的冲击将引起这些结构物的振动;工业设备(锅炉、压力容器等)的意外爆炸也会引起周围结构物的振动甚至破坏,等等。动力荷载的量测和数学表述是一个非常复杂的课题,本书由于篇幅所限,暂不讨论这一问题。

结构动力分析与静力分析的区别是,静力分析中的荷载与时间无关,结构的内力和变形也与时间无关(徐变等问题除外)。而动力分析中的荷载随时间变化,结构的内力和变形不仅与时间有关,而且与外荷载的特性及结构本身的动力特性有关。结构在动力荷载作用下将产生与结构本身质量有关的惯性力,以及与结构中质点的运动有关的阻尼力,这也是结构动力分析与静力分析的不同之处。用有限元法进行结构动力分析的原理与静力分析一样,也是将一个连续体离散化为一个以有限个结点的位移为广义坐标的多自由度系统,选择结点的自由度,建立单元的刚度矩阵、质量矩阵、阻尼矩阵和结点力矩阵等,建立整个系统的运动方程式并处理已知的边界条件和初始条件等,求解整个系统的动力特性(频率、振型等)和结构反应。在前几章中已介绍的有限元静力分析(包括非线性分析)的基础上,本章主要介绍与动力分析有关的问题。

8.1 动力分析的基本要求

在用有限元方法进行钢筋混凝土结构的动力分析时,除应满足静力分析中的各种要求之外,还必须考虑下列一些与结构动力特性有关的要求,这些要求对动力分析来说是必须满足的,也是最基本的要求。

8.1.1 单元尺寸大小的要求

将结构划分为有限单元之后,单元的振动频率必须高于整个结构的计算频率。若出现低于整体计算的频率时,则需将相应的单元再划分得小一些^[40]。这是因为,在结构动力计算时,总是希望首先找出整体结构的频率和振型,而不希望局部振动的频率混杂在结构整体振动的频率中。例如,在多层或高层建筑的抗震分析中,如果采用振型分解反应谱法计算地震作用,就必须求得整体结构的前几阶自振频率和振型,再计算结构在各个振型时的地震作用并进行适当的组合。在所选用的前几阶自振频率和振型中,如果有局部振动的成分,就会掩盖其他振型的贡献,从而得出不合理的结果,这种情况对于复杂体型的建筑结构需特别注意。当然,工程师对计算结果的正确判断和合理使用也是很重要的。

8.1.2 材料动力特性的适当考虑

结构材料承受动力荷载时的性能与承受静力荷载时的性能,往往有较大的差别。已有的试验研究结果表明,钢筋的屈服强度随着加载速率的增加而提高,但弹性模量基本上不变化,钢筋的极限强度只有很微小的提高。图 8.1 为钢筋的应力 - 应变曲线随加载速率的变化情况^[41]。试验也表明,屈服强度低的钢筋,当加载速率提高时,强度增加的比较显著,而屈服强度高的钢筋,在加载速率提高时强度的增长要比前者低一些,如图 8.2 所示^[41]。

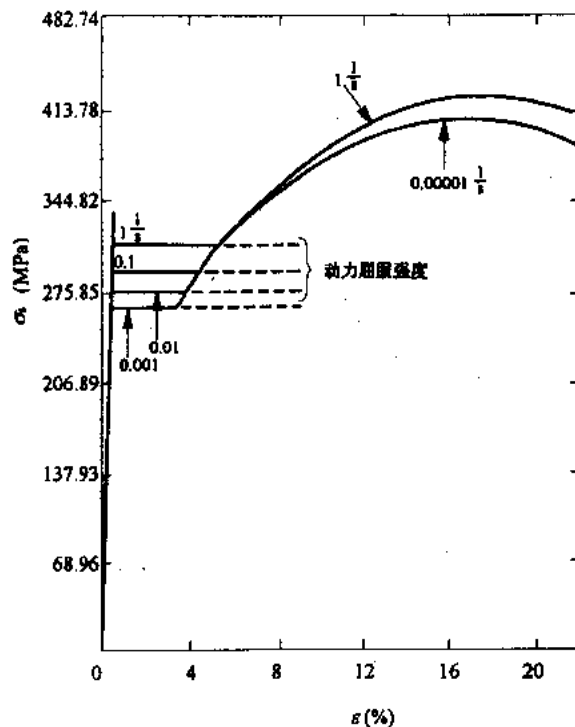


图 8.1 应变速率对钢筋 σ - ϵ 关系的影响

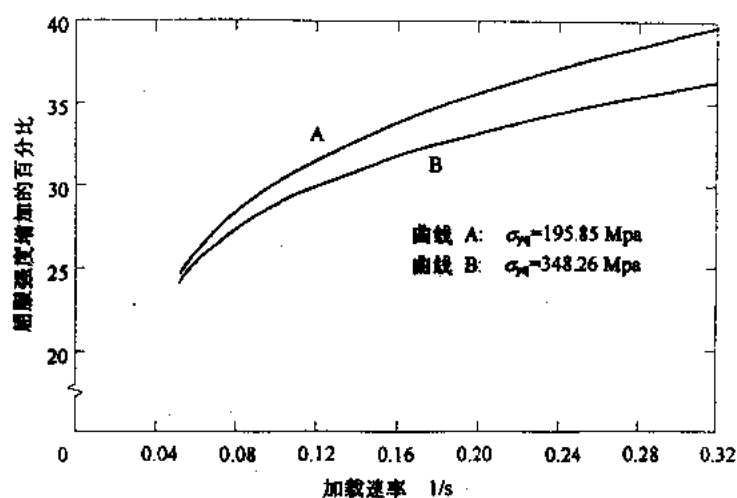


图 8.2 应变速率对屈服强度不同的钢筋的影响

混凝土的力学性质也受加载速率的影响。对在单轴受压下的混凝土应力 - 应变关系曲线的试验结果表明, 不仅最大强度、而且应力 - 应变曲线的形状也与加载速率有较大的关系, 如图 8.3 所示。加载速率对普通混凝土和微粒混凝土的影响也是不同的, 如图 8.4 所示^[41], 图中纵坐标为动力强度与静力强度之比, 本图中的数据可以作为动力计算中考虑混凝土强度受应变速率影响时的参考。

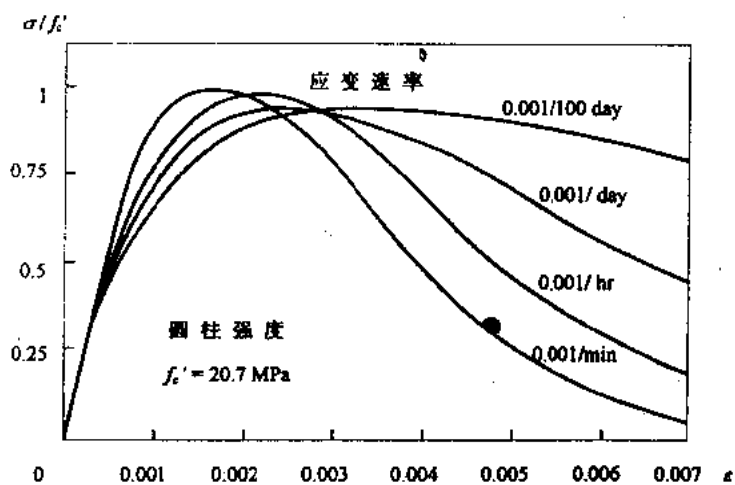


图 8.3 应变速率对混凝土应力 - 应变曲线的影响

加载速率对钢筋与混凝土之间的粘结强度以及滑移特性都有一定的影响。根据 Vos ^[41] 所做的试验结果, 对于光圆钢筋, 加载速率对粘结强度的影响很小, 而对于变形钢筋, 这种影响较为明显。因此, 在动力分析时可以根据构件的配筋情况, 来决定是否考虑加载速率对粘结强度的影响。当必须考虑这种影响时, 还需要做一些必要的试验, 为有限元计算提供更切合实际的试验数据。

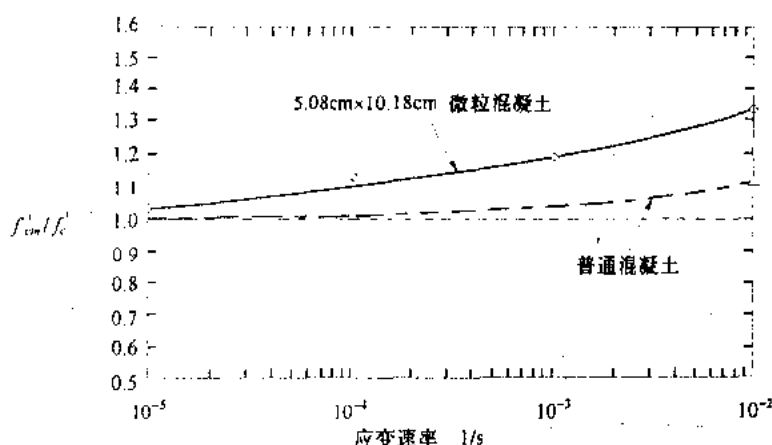


图 8.4 应变速率增加对混凝土抗压强度的影响

加载速率对结构材料力学性能影响的机理比较复杂,一般来说,徐变效应显著的材料,在动力荷载作用下其力学参数的变化比较显著。例如,应变速率对混凝土材料的影响比对钢材的影响大,对土的影响又显著大于对混凝土材料的影响。混凝土的动力强度高于静力强度的可能解释是,动力荷载作用时间短和混凝土具有较大的吸收应变能量的能力,这种提高还与混凝土中水泥石的粘滞性,以及伴随变形产生的内部微裂缝的惯性抵抗能力有关。

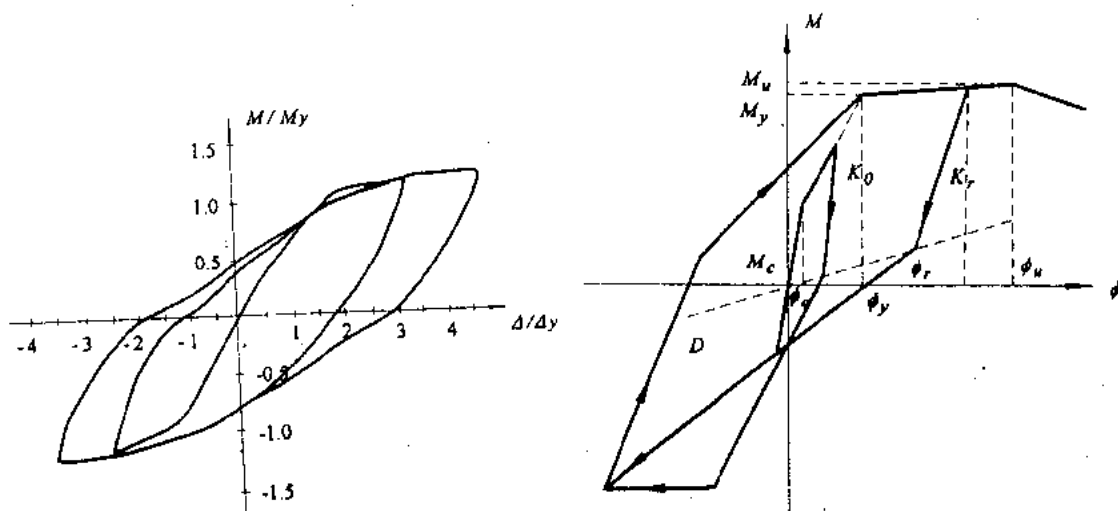
在结构动力分析中,材料动力强度的取值要根据加载速率的大小来确定,当计算冲击荷载下结构的反应时,就应考虑这种高应变速率对材料性能和计算参数的影响。但在计算结构的地震反应时,由于地震作用属于低周反复荷载,地震时结构物的应变速率通常小于 $0.001/s$,从试验曲线(图 8.4)来看,当 $\dot{\epsilon} \leq 0.001/s$ 时,应变速率对混凝土强度的影响很小,因此,可以采用静力试验得到的力学性能指标。

8.1.3 材料或构件滞回关系的选用

材料或构件在反复荷载作用下得到的力 - 变形曲线叫滞回曲线,这种滞回曲线的数学描述就是材料或构件在反复荷载作用下的本构关系或叫本构方程。在钢筋混凝土结构有限元动力分析中,材料在反复荷载作用下本构关系的选用与单元的划分和选取密切相关。计算中,若采用平面单元或实体单元,就应该在应力 - 应变层次上选用本构关系,如混凝土的 $\sigma-\epsilon$ 曲线可选用图 7.1 的形式,钢筋的 $\sigma-\epsilon$ 曲线可选用图 7.4 的形式。若在有限元模型中采用杆系单元,就应该在弯矩 - 曲率层次上选用本构关系,如构件在反复荷载下的 $M-\varphi$ 曲线可选用图 8.5 所示的形式,其中图(a)为试验曲线,图(b)为模型化的曲线。

结构构件的 $M-\varphi$ 曲线也可以按照本书第五章中所介绍的方法进行计算,这种计算可以与结构的整体分析同步进行;也可以预先计算好存储起来,在进行结构整体分析时以查表方式随时调用,这样做可以节省一些计算时间。

结构构件的 $M-\varphi$ 曲线还受到轴向力的影响。在一般情况下,梁受的轴向力比较小,可以忽略轴向力对 $M-\varphi$ 曲线的影响。但柱子是以承受轴向力为主的构件,在选取或计算柱



(a) 试验曲线

(b) 模型化曲线

图 8.5 受弯构件的 $M-\phi$ 曲线

截面的 $M-\phi$ 关系时必须反映轴向力的影响。在建筑结构的非线性地震反应分析中,柱截面上所受的轴向力是交替变化的,因此,每时每刻的 $M-\phi$ 曲线都是不同的。这样,在进行结构整体分析时,就应该根据即时的轴力来确定截面上的 $M-\phi$ 关系,以求得比较符合实际的计算结果。

已有的试验研究表明,剪力的存在对受弯构件截面的 $M-\phi$ 曲线也有明显的影响,使得滞回曲线变成“梭形”,并明显地降低了构件的耗能能力。因此,当采用杆系单元进行有限元分析时,如果在计算单元特性时没有考虑剪力的作用,就应在恢复力模型中反映这种影响。国内这方面的试验研究成果已有很多,可供读者在计算分析时选用。

8.2 动力方程及单元特性

8.2.1 动力方程

根据达朗贝尔原理,任何一个振动系统的平衡方程都可以写成下列形式:

$$\{F_I\} + \{F_D\} + \{F_S\} = \{P\} \quad (8.1)$$

式中, $\{F_I\}$, $\{F_D\}$, $\{F_S\}$ 分别表示结构的惯性力、阻尼力和弹性力向量, $\{P\}$ 为外荷载向量。其中惯性力及弹性力向量可以进一步用加速度与位移向量来表示:

$$\{F_I\} = -[M]\{\ddot{\delta}\}; \{F_S\} = -[K]\{\delta\} \quad (8.2)$$

式中, $[M]$ 和 $[K]$ 表示结构的总体质量矩阵和总体刚度矩阵。

结构的阻尼力是非常复杂的。一般来说,在任何真实结构中所看到的阻尼现象,都是由各种各样很复杂的能量耗散机理所引起的,结构材料的基本阻尼特性通常是不易确切知道的,因此,人们往往通过与同类结构中已观察到的阻尼特性进行比较,再来建立所要分析的结构的阻尼性质。当采用粘滞阻尼的假设时,阻尼力与速度成正比,这样,阻尼力向

量可以表示为

$$\{F_D\} = -[C]\{\dot{\delta}\} \quad (8.3)$$

式中, $[C]$ 为结构的总体阻尼矩阵。

在钢筋混凝土结构的有限元动力分析中, 一般情况下, 可以采用按静力方法得到的刚度矩阵, 如本书的第三章、第六章和第七章所示的计算公式, 因此, 本章主要介绍质量矩阵和阻尼矩阵的计算和处理方法。

8.2.2 质量矩阵

如材料的密度为 ρ , 则结构内单位体积的惯性力为 $-\rho\{\ddot{u}\}$, 其中 $\{\ddot{u}\}$ 为质点的加速度。对结构来说, 惯性力相当于另一种体积分布的荷载, 大小与质点的加速度成比例, 而与加速度方向相反。

按有限元方法, 以单元结点位移 $\{\delta\}^e$ 插值表示单元内部位移、速度和加速度为

$$\{u\} = [N]\{\delta\}^e; \quad \{\dot{u}\} = [N]\{\dot{\delta}\}^e; \quad \{\ddot{u}\} = [N]\{\ddot{\delta}\}^e \quad (8.4)$$

这里的形函数仍只是位置的插值函数, 与时间无关; $\{\delta\}^e, \{\dot{\delta}\}^e, \{\ddot{\delta}\}^e$ 为单元结点的位移、速度和加速度向量。

将单元内惯性力 $-\rho\{\ddot{u}\}$ 作为体积分布荷载, 按体积力与结点力的转换式进行分配, 可得到

$$\begin{aligned} \{F_I\}^e &= - \iiint_V [N]^T \rho \{\ddot{u}\} dV \\ &= - \iiint_V [N]^T \rho [N] \{\ddot{\delta}\}^e dV \\ &= - \iiint_V [N]^T \rho [N] dV \{\delta\}^e \\ &= - [M]^e \{\ddot{\delta}\}^e \end{aligned}$$

$$\text{式中} \quad [M]^e = \iiint_V [N]^T \rho [N] dV \quad (8.5)$$

式(8.5)为单元质量矩阵的一般公式, 可以按具体单元的形函数矩阵 $[N]$ 在单元内求积, 以得到 $[M]^e$ 。

当采用受弯的平面梁单元时, $[M]^e$ 是一个对称的 4×4 阶方阵, 具有如下的公式:

$$[M]^e = \frac{ml}{420} \begin{bmatrix} 156 & -22l & 54 & 13l \\ -22l & 4l^2 & -13l & -3l^2 \\ 54 & -13l & 156 & 22l \\ 13l & -3l^2 & 22l & 4l^2 \end{bmatrix} \quad (8.6)$$

式中, m 为单元长度范围内每米长度上的质量, l 为单元的长度。由于在推导(8.6)式所示的单元质量矩阵时, 采用了与推导单元刚度矩阵时同样的形函数矩阵, 因此, 这样的质量矩阵叫做一致质量矩阵。

当采用厚度为 t 的平面 4 结点矩形单元时, $[M]^e$ 是一个对称的 8×8 阶方阵, 具有如下的形式:

$$[M]^e = \frac{\rho t A}{36} \begin{bmatrix} 4 & & & & & & & \\ 0 & 4 & & & & & & \\ & 2 & 0 & 4 & & & & \\ 0 & 2 & 0 & 4 & & & & \\ 1 & 0 & 2 & 0 & 4 & & & \\ 0 & 1 & 0 & 2 & 0 & 4 & & \\ 2 & 0 & 1 & 0 & 2 & 0 & 4 & \\ 0 & -2 & 0 & 1 & 0 & 2 & 0 & 4 \end{bmatrix} \quad (8.7)$$

式中, $A = 4ab$, 单元边长为 $2a \times 2b$, t 为单元厚度。

当采用厚度为 t 的平面等参单元时, 则

$$\begin{aligned} [M]^e &= \iint [N]^T \rho [N] t dx dy \\ &= \iint [N]^T \rho [N] t |J| d\xi d\eta \end{aligned} \quad (8.8)$$

式中 $|J|$ 为雅可比矩阵的行列式值, 式中的积分可在母单元内进行。但由于 $[N]$ 矩阵的复杂性, 一般很难由直接积分求出, 常采用数值积分法, 例如高斯积分法。对式(8.8) 进行高斯数值积分, 可得

$$\{M_{ij}\}^e = \sum_{k=1}^n \sum_{l=1}^n H_k \cdot H_l \cdot \rho \cdot N_i \cdot N_j \cdot |J| \cdot t \quad (8.9)$$

在按平面问题进行建筑结构的有限元动力分析时, 还应考虑分布于各楼层的附加质量和楼板质量。这些质量应按其在有限元模型中的结点位置, 叠加到结点所对应的质量矩阵中。

8.2.3 阻尼矩阵

在结构动力分析中使用最多的是瑞利(Rayleigh) 阻尼假定, 即认为阻尼力正比于质点运动速度和应变速度, 并可以取这两种速度引起的阻尼力的线性组合。

正比于质点运动速度的单位体积上作用的阻尼力为

$$\{F_{D1}\} = -\alpha \rho \{\dot{u}\}$$

式中, α 为常数, ρ 为材料密度, 由

$$\{\dot{u}\} = [N] \{\dot{\delta}\}^e$$

可得

$$\{F_{D1}\} = -\alpha \rho [N] \{\dot{\delta}\}^e$$

按体积力与结点力转换式, 可得

$$\begin{aligned} \{F_{D1}\}^e &= \iiint_V [N]^T \{F_{D1}\} dV \\ &= -\alpha \iiint_V [N]^T \rho [N] dV \{\dot{\delta}\}^e \\ &= -[C_1] \{\dot{\delta}\}^e \end{aligned}$$

式中

$$[C_1] = \alpha \iiint_V [N]^T \rho [N] dV = \alpha [M]^e \quad (8.10)$$

正比于应变速度的阻尼应力为

$$\{\sigma\} = -\beta[D] \frac{\partial \{\varepsilon\}}{\partial t} = -\beta[D][B]\{\dot{\delta}\}^e$$

作用于单元结点上的阻尼力为

$$\begin{aligned}\{F_{D2}\}^e &= \iiint_V [B]^T \{\sigma\} dV = -\beta \iiint_V [B]^T [D] [B] dV \{\dot{\delta}\}^e \\ &= -[C_2]\{\dot{\delta}\}^e\end{aligned}$$

$$\text{式中} \quad [C_2] = \beta \iiint_V [B]^T [D] [B] dV = \beta[K]^e \quad (8.11)$$

将式(8.10)与式(8.11),相加可得

$$[C]^e = [C_1]^e + [C_2]^e = \alpha[M]^e + \beta[K]^e \quad (8.12)$$

$$\begin{aligned}\alpha &= 2\omega_i \cdot \omega_j (\zeta_j \omega_i - \zeta_i \omega_j) / (\omega_i^2 - \omega_j^2) \\ \beta &= 2(\zeta_i \omega_i - \zeta_j \omega_j) / (\omega_i^2 - \omega_j^2)\end{aligned} \quad (8.13)$$

式中, ω_i, ω_j 分别为第 i 振型、第 j 振型的圆频率, ζ_i, ζ_j 分别为第 i 振型、第 j 振型的阻尼比。通常, 称 ω_1 为系统的基频, ζ_1 为相应振型的阻尼比。对于钢筋混凝土结构, ζ_1 约为 0.05。在结构动力分析中, 阻尼比 ζ_i 一般取实测的值, 当无实测值时, 一般情况下可取 $\zeta_1 = 0.05$, 并可取前两阶振型的频率和阻尼比来计算 α 和 β 值。

瑞利阻尼表达式中质量项和刚度项的贡献以及它们组合后对阻尼矩阵的影响如图 8.6 所示^[43]。

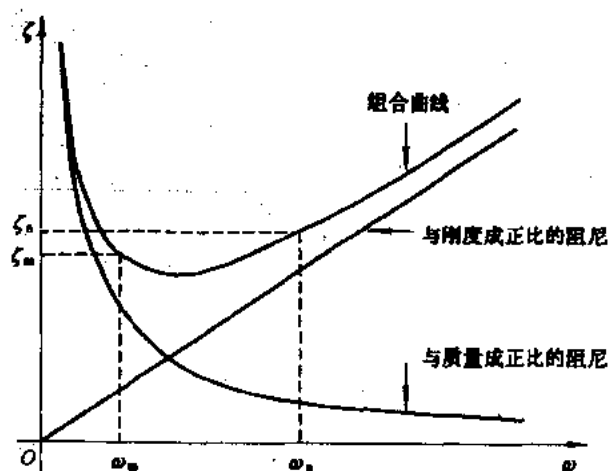


图 8.6 瑞利阻尼中阻尼比与频率的关系

8.3 动力特性的求解方法

对于式(8.1), 将惯性力、阻尼力及弹性力按单元叠加, 可得到有限元结点位移方程为

$$\sum_{e=1}^n ([M]^e \{\ddot{\delta}\}^e + [C]^e \{\dot{\delta}\}^e + [K]^e \{\delta\}^e) = \{P\} \quad (m \text{ 为单元数量})$$

$$\text{即} \quad [M]\{\ddot{\delta}\} + [C]\{\dot{\delta}\} + [K]\{\delta\} = \{P\} \quad (8.14)$$

式中, $[M]$, $[C]$, $[K]$ 分别为整个结构系统的质量矩阵、阻尼矩阵和刚度矩阵, 这些矩阵均为单元相应矩阵的叠加, 叠加的意义与前几章中介绍的刚度矩阵的叠加是相同的。荷载列阵 $\{P\}$ 为按有限元方法将集中荷载或均布荷载分配到结点上的结点荷载列阵, 也是时间的函数。

当没有外荷载作用时, $\{P\} = 0$, 如果再忽略阻尼力, 则方程(8.14)变为

$$[M]\{\ddot{\delta}\} + [K]\{\delta\} = 0 \quad (8.15)$$

式(8.15)为系统的自由振动方程。如整个结构系统的全部结点有 n 个自由度, 则式(8.15)就是 n 个自由度系统的自由振动方程。应注意, 这里的 n 个自由度, 是指经过边界条件处理过的, 即除去了边界上位移约束点自由度以后的结点位移数目, 它已排除了刚体位移部分。

设结构在自由振动时的位移为 $\{\delta\} = \{A\} \cos \omega t$, 代入(8.15)式后, 可得下列齐次方程组:

$$([K] - \omega^2[M])\{A\} = 0 \quad (8.16)$$

由于在自由振动时, 各结点的振幅 $\{A\}$ 不可能全部为零, 因此, 要使(8.16)式有非零解的条件是其系数矩阵行列式的值必须为零, 即

$$|[K] - \omega^2[M]| = 0 \quad (8.17)$$

n 个自由度的系统, 就有 n 个固有频率。对于有限元分析中自由度数目很多的结构系统, 要解 ω^2 的 n 次方程(8.17)式是非常困难的, 一般用计算机程序求解, 并将式(8.17)转换成数学上的特征值方程, 用数学上求解特征值问题的方法求解。

求解特征值问题有很多方法, 并有相应的计算机程序。本书不详细介绍这些方法的计算过程, 只是对这些方法的特点进行比较, 以供读者在选用这些方法时参考^[44]。

1. Jacobi 方法

这是目前广泛使用的求解实对称矩阵特征值问题的最古老的方法之一。其优点是计算公式简单, 程序很短, 易于掌握, 适合于计算矩阵的全部特征值及其对应的特征向量。缺点是收敛较慢, 计算耗时太多, 不宜计算阶数太高的特征值问题。

2. QR 方法

这是目前计算一般矩阵(包括复矩阵、非对称矩阵)全部特征值和特征向量的最有效方法之一。其优点是收敛速度快, 计算过程稳定。这个方法的功能比较全面, 适用范围广, 因此程序结构复杂。

3. 乘幂法及反乘幂法

前者可按模最大的特征值及对应的特征向量求解, 后者可按模最小的特征值及对应的特征向量求解。这类方法对系数矩阵的要求较低, 常用于计算结构系统的基频。其优点是计算过程简单明了, 甚至可以对低阶矩阵进行人工手算, 缺点是收敛性不稳定。

4. G-H 方法

这是求解实对称矩阵特征值问题的重要方法之一。它可以根据需要, 求解矩阵任意数目的特征值及其相应的特征向量, 具有很大的灵活性。这个方法的另一优点是计算速度快。对于同样的问题, G-H 方法的速度比 Jacobi 方法快 5 倍以上。

5. 子空间迭代法

这是求解对称矩阵前几个特征值和特征向量的有效方法。与前述的 4 种方法相比,它具有计算速度快、效率高、计算结果可靠的优点,其缺点是收敛性依赖于子空间向量的选择,与近几年来发展起来的方法相比,仍有速度较慢、解题规模小的缺点。

6. Ritz 向量法

这是一种依赖于荷载的特征值解法,它可以用来代替子空间迭代法,并能得到可靠的近似结果。其优点是不涉及迭代过程;把动力问题的求解化为一组静力问题的求解,系数矩阵([K] 矩阵)只需要进行一次三角分解就行,剩下的只是回代,所以计算速度比子空间迭代法还要快,平均快 4 倍至 10 倍。结构系统的自由度越大,这个方法的优越性越明显。其缺点是在这个方法里要事先给定一个非零的荷载向量,并且收敛速度要受到这个荷载向量的影响。

7. Lanczos 法

Lanczos 法的主要思想是采用一个 Lanczos 过程把特征值问题的系数矩阵三对角化,在三对角化以后则用一般的 QL 迭代,很快求得所需要的特征值和特征向量。这个方法的计算速度比 Ritz 向量法稍快,但精度略低,也是一个非常强有力的求解特征值问题的方法。

为了使读者编程方便,这里介绍作者多年来一直使用的广义 Jacobi 方法^[45]。广义 Jacobi 方法的基本思路是寻找合适的相似变换矩阵[P]和[Q],使

$$[P][K][Q] = \begin{bmatrix} \alpha_1 & & & & \\ & \alpha_2 & & & \\ & & \alpha_i & & \\ & & & \ddots & \\ & & & & \alpha_n \end{bmatrix}$$
$$[P][M][Q] = \begin{bmatrix} \beta_1 & & & & \\ & \beta_2 & & & \\ & & \beta_i & & \\ & & & \ddots & \\ & & & & \beta_n \end{bmatrix}$$

于是特征值 $\lambda_i = \omega_i^2 = \alpha_i / \beta_i (i = 1, 2, \dots, n)$ 。由于以上的变换为相似变换,所得的特征值即为式(8.16)的特征值。

用 C 语言编写的广义 Jacobi 法的计算程序如下:程序(code)中第一部分为主程序,有计算所用到的数组、输入、输出文件的说明等,以及调用子程序 Jaco 和打印输入等内容。输入数据有:自由度数 $n = 3$;质量矩阵为:2561,0,0,0,2545,0,0,0,559;刚度矩阵为:1446000, -903000,0, -903000,1726000, -823000,0, -823000,823000。第二部分为 Jacobi 方法的子程序;第三部分为计算结果的输出数据。

```
// Calculate Eigen Values by Jacobi Method
```

```
#include "math.h"
```



```

#include "stdlib.h"
#include "conio.h"
#include "stdio.h"
void jaco (int n, double k[], double m[], double u[], double e[]);
double max(double x, double y);
void main()
{int i, j, n = 3;
    double u[3][3], e[3];
    double temp;
    char outfile[20];
    static double m[3][3] = {{2561.0, 0.0, 0.0}, {0.0, 2545.0, 0.0}, {0.0, 0.0, 559.
                                0}};
    static double k[3][3] = {{1446000.0, -903000.0, 0.0}, {-903000.0, 1726000.0,
                                -823000.0}, {0.0, -823000.0, 823000.0}};

    FILE *fp;
    clrscr();
    gotoxy(12, 2);
    printf("PLS Enter file name of answer:");
    scanf("%s", &outfile);
    jaco(n, k[0], m[0], u[0], e);
    fp = fopen(outfile, "w"); /* store data in user's file */
    if(fp != (void *)NULL)
    {
        fprintf(fp, "\n The eigen value is: \n");
        for(i = 0; i < n; i++)
        { fprintf(fp, "%18.6f", e[i]); }
        fprintf(fp, "\n The angle frequency is: \n");
        for(i = 0; i < n; i++)
        { fprintf(fp, "%18.6f", sqrt(e[i])); }
        fprintf(fp, "\n");
        fprintf(fp, "Number of eigen vector and the eigen vector are: \n");
        for(i = 0; i < n; i++)
        { temp = 0.0;
            fprintf(fp, "%d", i + 1);
            for(j = 0; j < n; j++)
            { if(fabs(u[j][i]) > temp) temp = fabs(u[j][i]); }
            for(j = 0; j < n; j++)
            { u[j][i] = u[j][i]/temp;

```

```

        fprintf(fp, "%18.6f", u[j][i]);
    }
    fprintf(fp, " \n");
}
}
else printf("Can't open this outfile \n");
fclose(fp);
}

void jaco(int n, double k[], double m[], double u[], double e[])
{
    int i, j, il;
    int nl, p, q;
    int kmax, kaisu, kfac;
    double fmk, fmm;
    double fac, epsk, epsm;
    double d, d1, d2, d3;
    double a, ak, am, re, ru;
    double s, x, g, w, l;
    if(n == 1)
    {
        e[0] = k[0]/m[0];
        u[0] = 1.0;
    }
    nl = n - 1;
    fmk = 0.0;
    fmm = 0.0;
    for(i = 0; i < n; i++)
    {
        for(j = 0; j < n; j++)
        {
            fmk = max(fmk, fabs(k[i * n + j]));
            fmm = max(fmm, fabs(m[i * n + j]));
        }
    }
    fac = 0.00001;
    epsk = fmk * fac;
    epsm = fmm * fac;
    kfac = 5;
    kmax = kfac * n * n;
    ak = 1.0 * pow10(30);
    for(i = 0; i < n; i++)
    {
        for(j = 0; j < n; j++)

```

```

        { u[i * n + j] = 0.0; }
        u[i * n + i] = 1.0;
    }
    for(kaisu = 0; kaisu < kmax; kaisu++)
    { p = 0;
      q = 1;
      if(fmod(kaisu, 2.0) == 1.0) goto loop1;
      ak = fabs(k[1]);
      for(i = 0; i < n1; i++)
      { il = i + 1;
        for(j = il; j < n; j++)
        { a = fabs(k[i * n + j]);
          if(ak > a) goto loop6;
          p = i;
          q = j;
          ak = a;
        }
      }
      goto loop2;
    }
loop1:;
    am = fabs(m[1]);
    for(i = 0; i < n1; i++)
    { il = i + 1;
      for(j = il; j < n; j++)
      { a = fabs(m[i * n + j]);
        if(am > a) goto loop3;
        p = i;
        q = j;
        am = a;
      }
    }
loop3:;
loop2:;
    d1 = k[p * n + p] * m[p * n + q] - k[p * n + q] * m[p * n + p];
    d2 = k[q * n + q] * m[p * n + q] - k[p * n + q] * m[q * n + q];
    d3 = k[p * n + p] * m[q * n + q] - k[q * n + q] * m[p * n + p];
    d = d3 * d3 + 4.0 * d1 * d2;

```

```

s = sqrt(d);
x = (d3 + s) * 0.5;
if(d3 < 0.0) x = (d3 - s) * 0.5;
a = d2/x;
g = - d1/x;
for(j = 0; j < n; j++)
{ w = k[p * n + j];
  k[p * n + j] = k[p * n + j] + g * k[q * n + j];
  k[q * n + j] = k[q * n + j] + a * w;
  w = m[p * n + j];
  m[p * n + j] = m[p * n + j] + g * m[q * n + j];
  m[q * n + j] = m[q * n + j] + a * w;
}
for(i = 0; i < n; i++)
{ w = k[i * n + p];
  k[i * n + p] = k[i * n + p] + g * k[i * n + q];
  k[i * n + q] = k[i * n + q] + a * w;
  w = m[i * n + p];
  m[i * n + p] = m[i * n + p] + g * m[i * n + q];
  m[i * n + q] = m[i * n + q] + a * w;
  w = u[i * n + p];
  u[i * n + p] = u[i * n + p] + g * u[i * n + q];
  u[i * n + q] = u[i * n + q] + a * w;
}
if((ak < epsk) && (am < epsm)) goto loop4;

printf("matrix K or M are wrong");
loop4:;
for(i = 0; i < n; i++)
{ e[i] = k[i * n + i] / m[i * n + i]; }
for(i = 0; i < n1; i++)
{ il = i + 1;
  for(j = il; j < n; j++)
  { if(e[i] < e[j]) goto loop5;
    re = e[i];
    e[i] = e[j];
    e[j] = re;
    for(l = 0; l < n; l++)

```

```

        { ru = u[1 * n + i];
          u[1 * n + i] = u[1 * n + j];
          u[1 * n + j] = ru;
        }
loop5:;
}

double max(double x, double y)
{ double z;
  if(x > y) z = x;
  else     z = y;
  return(z);
}

```

计算结果如下:

The eigen value is:

78.919988	740.231385	1895.936270
-----------	------------	-------------

The angle frequency is:

8.883692	27.207194	43.542350
----------	-----------	-----------

Number of eigen vector and the eigen vector are:

1	0.687037	0.946396	1.000000
2	-0.998345	0.497218	1.000000
3	0.076213	-0.287762	1.000000

运用前面介绍的任何一种求解特征值问题的方法,就可以求得式(8.17)中的 n 个自振频率 $\omega_i (i = 1, 2, \dots, n)$, 以及对应的 n 个特征向量, 这些特征向量即代表了结构振动中各个质点的相对位移关系。可以证明, 当结构按其自振频率振动时, 所有质点的位移比值始终保持不变。这种特殊的振动形式称为主振型, 或简称振型。一般, 动力系统中有多少个自由度就有多少个频率, 相应地就有多少个主振型(以特征向量表示), 它们是系统的固有特性。在大多数情况下, 系统的振动曲线将包含全部的振型。

振型有一个非常重要的特点, 即任意两个频率的振型之间, 都存在着互相正交的性质, 用方程式表示如下:

$$\left. \begin{aligned} \{A\}_i^T [M] \{A\}_j &= 0 \quad (i \neq j) \\ \{A\}_i^T [M] \{A\}_j &= \bar{M}_i \quad (i = j) \end{aligned} \right\} \quad (8.18)$$

$$\left. \begin{aligned} \{A\}_i^T [K] \{A\}_j &= 0 \quad (i \neq j) \\ \{A\}_i^T [K] \{A\}_j &= \bar{K}_i \quad (i = j) \end{aligned} \right\} \quad (8.19)$$

式(8.18)称为振型的第一正交条件, 即振型关于质量矩阵的正交条件或振型关于质

量矩阵的加权正交性,式中 $\bar{M}_i$ 称为振型质量或称模态质量。

式(8.19)称为振型的第二正交条件,即振型关于刚度矩阵的正交条件或振型关于刚度矩阵的加权正交性,式中 $\bar{K}_i$ 称为振型刚度或模态刚度。

振型的正交性在结构动力反应计算中非常有用,利用这一特性可以使动力反应的求解大大简化。下一节我们将介绍利用振型正交性求解动力反应的方法。

8.4 动力反应的求解方法

求解(8.14)式的动力反应,一般有两种方法。第一种是所谓的振型分解法,利用振型的正交性,把这些联立的方程组分解为一个个相互独立的振动方程,逐个求解后再叠加,因此这个方法有时也称振型叠加法。使用这个方法需要先计算出系统的各阶振型,而且也仅适合于线性振动系统和比例阻尼的情况。第二种是数值积分方法,直接对多自由度系统的微分方程(8.14)进行积分,在积分计算中把时间历程划分为有限个微小的时段,将动力方程式化解成为矩阵形式的代数方程,用计算机逐步求解,这个方法有时也称为逐步积分法或时程分析法。这个方法可用于一般的阻尼情况,并且可以用逐段线性化的方法求解非线性动力系统的计算问题。

8.4.1 振型分解法

对于多自由度系统,结构的动力反应可以用各个振型动力反应的线性组合来表示,即

$$\{\delta(t)\} = [A]\{q(t)\} \quad (8.20)$$

式中, $\{\delta(t)\}$ 为位移向量; $\{q(t)\}$ 为广义坐标向量; 矩阵 $[A]$ 为振型矩阵, 振型矩阵中第 i 列向量 $\{A\}_i$ 即为系统的第 i 个振型向量。

将(8.20)式代入系统的动力方程式(8.14),并左乘振型向量 $\{A\}_j^T$ 后,可得

$$\begin{aligned} \{A\}_j^T [M] [A] \{\ddot{q}(t)\} + \{A\}_j^T [C] [A] \{\dot{q}(t)\} \\ + \{A\}_j^T [K] [A] \{q(t)\} = \{A\}_j^T \{P\} \end{aligned} \quad (8.21)$$

利用振型关于质量矩阵的正交性(8.18)式以及振型关于刚度矩阵的正交性(8.19)式,并假定阻尼矩阵也满足正交性条件,可以得到

$$\bar{M}_j \ddot{q}_j(t) + \bar{C}_j \dot{q}_j(t) + \bar{K}_j q_j(t) = \bar{P}_j \quad (j = 1, 2, \dots, n) \quad (8.22)$$

式中 $\bar{M}_j$ 、 $\bar{K}_j$ 分别为振型质量和振型刚度,如式(8.18)和(8.19)所示; $\bar{C}_j$ 为振型阻尼,根据假定也满足正交性条件,即 $\bar{C}_j = \{A\}_j^T [C] \{A\}_j$, 当采用瑞利阻尼时,很明显,这个条件是自然满足的; $\bar{P}_j = \{A\}_j^T \{P\}$ 称为振型结点荷载。

逐个求解(8.22)式,即可得到 n 个广义坐标 $q_j(t)$ ($j = 1, 2, \dots, n$), 代入式(8.20), 即得到了结构系统的动力反应。用振型分解法求得的结点位移 $\{\delta(t)\}$ 是时间的函数, 由它插值的单元内部位移、应力、应变的计算与静力方法一样, 不同的是这些量都是时间的函数。

用振型分解法求解结构系统的动力反应时有两个明显的优点。第一个优点是 n 个相互耦合的方程利用振型正交性解耦后相互独立, 变成了 n 个单自由度方程, 使计算过程大

大简化。第二个优点是只需按要求求解少数几个振型的方程,就可以得到满意的解答,因为在大多数情况下,结构的动力反应主要是前面几个低阶振型起控制作用。例如对于一般的高层钢筋混凝土建筑,在进行结构的地震反应计算时,通常情况下,考虑前面3~5个振型,就可获得足够精确的结果。对于体型比较复杂的高层建筑,有时需要多考虑几个振型的影响,但也不是考虑全部的振型。

8.4.2 直接积分法

在结构动力计算中,常用的直接积分法有中心差分法、线性加速度法和 Wilson- θ 法,本节将对这些方法作简单介绍。

1. 中心差分法

中心差分法的基本思路是将动力方程式中的速度向量和加速度向量用位移的某种组合来表示,将微分方程组的求解问题转换为代数方程组的求解问题,并在时间历程内求出每个微小时段的递推公式,进而逐步求得整个时程的反应。

对于(8.14)式的动力方程

$$[M]\{\ddot{\delta}\} + [C]\{\dot{\delta}\} + [K]\{\delta\} = \{P\} \quad (a)$$

其中各阶微分可以用中心差分表示为

$$\{\dot{\delta}\} = \frac{1}{2\Delta t}(\{\delta(t + \Delta t)\} - \{\delta(t - \Delta t)\}) \quad (b)$$

$$\{\ddot{\delta}\} = \frac{1}{(\Delta t)^2}(\{\delta(t + \Delta t)\} - 2\{\delta(t)\} + \{\delta(t - \Delta t)\}) \quad (c)$$

式中 Δt 为均匀的时间步长, $\{\delta(t)\}$ 、 $\{\delta(t - \Delta t)\}$ 和 $\{\delta(t + \Delta t)\}$ 分别为 t 时刻及其 Δt 前、后时刻的结点位移向量。将(b)、(c)式代入(a)式后可得到一个递推公式如下:

$$\left. \begin{aligned} [K]^* \{\delta(t + \Delta t)\} &= \{P(t)\}^* \\ [K]^* &= \left(\frac{1}{(\Delta t)^2} [M] + \frac{1}{2\Delta t} [C] \right) \\ \{P(t)\}^* &= \{P(t)\} - \left([K] - \frac{2}{(\Delta t)^2} [M] \right) \{\delta(t)\} - \left(\frac{1}{(\Delta t)^2} [M] - \frac{1}{2\Delta t} [C] \right) \{\delta(t - \Delta t)\} \end{aligned} \right\} \quad (8.23)$$

上式即为中心差分法的计算公式,在求得结构的 $[M]$ 、 $[C]$ 、 $[K]$ 和 $\{P(t)\}$ 后,就可以根据 t 时刻及 $t - \Delta t$ 时刻的结点位移,按(8.23)式推算出 $t + \Delta t$ 时刻的结点位移;并可逐步推出 $t + 2\Delta t, t + 3\Delta t, \dots, t_{end}$ 各时刻的结点位移。

式(8.23)对于 $t = 0$ 的时刻并不适用,因为一般运动的初始条件给出的是初始位移 $\{\delta(0)\}$ 和初始速度 $\{\dot{\delta}(0)\}$,而难以给出前一个 Δt 时刻的位移 $\{\delta(-\Delta t)\}$,无法直接按式(8.23)进行第一步的计算,因此,这时就要利用其他条件建立中心差分的计算公式,

$$\{\dot{\delta}(0)\} = \frac{1}{2\Delta t}(\{\delta(\Delta t)\} - \{\delta(-\Delta t)\}) \quad (d)$$

$$\{\ddot{\delta}(0)\} = \frac{1}{(\Delta t)^2}(\{\delta(\Delta t)\} - 2\{\delta(0)\} + \{\delta(-\Delta t)\}) \quad (e)$$

再利用 $t = 0$ 时刻的动力方程:


```

        exit(1);
    }
    i = 0; id = 0;
    while( !feof(fp))
        { fscanf(fp, "%f", &yg[i]);
          yg[i] = yg[i] * 10.0;
          i ++; id ++;
        }
    fclose(fp);
    cdm(n, id, dt, m[0], k[0], c[0], yg, d0, v0, fd, d, zd, v, a,
        unit, vec1, vec2, vec3, vec[0]);
}

void cdm(int n, int id, float dt, float bm[], float bk[], float bc[], float yg[],
        float d0[], float v0[], float fd[], float d[], float zd[], float v[],
        float a[], float unit[], float vec1[], float vec2[], float vec3[], float vec[])
{ int i, j, l;
  int layer, key;
  char outfile[20];
FILE * fp;
  clrscr();
  gotoxy(12, 2);
  printf("PLS enter file name of answer:");
  scanf("%s", &outfile);
  gotoxy(12, 4);
  printf("PLS enter the layer:");
  scanf("%d", &layer);
  fp = fopen(outfile, "w"); /* store data in user's file */
  if(fp != (void *) NULL)
    { fprintf(fp, "The answer of structural response \n");
      fprintf(fp, "The layer is : %5d \n", layer);
    }
    else printf("Can't open this outfile \n");
  fclose(fp);
  for(i = 0; i < n; i++) { unit[i] = - 1.0; } /* unit[] represent the unit matrix I */
  bat(bm, unit, vec1, n, 1, n);
  bat(bk, d0, vec2, n, 1, n);
  bat(bc, v0, vec3, n, 1, n);
  for(i = 0; i < n; i++) { vec1[i] = vec1[i] * yg[0] - vec2[i] - vec3[i]; }
}

```

```

for(i = 0; i < n; i++)
{ for(j = 0; j < n; j++)
  { vec[i * n + j] = bm[i * n + j]; }
}
gsn(vec, vec1, n, 1, 0.0000001, key);
for(i = 0; i < n; i++)
{ fd[i] = d0[i] - dt * v0[i] + (dt * dt / 2) * vec1[i];
  d[i] = d0[i];
}
for(l = 0; l < id; l++)
{ bat(bm, unit, vec1, n, 1, n);
  for(i = 0; i < n; i++) { vec1[i] = vec1[i] * xg[l]; }
  for(i = 0; i < n; i++)
  { for(j = 0; j < n; j++)
    { vec[i * n + j] = bk[i * n + j] - (2 / (dt * dt)) * bm[i * n + j]; }
  }
  bat(vec, d, vec2, n, 1, n);
  for(i = 0; i < n; i++)
  { for(j = 0; j < n; j++)
    { vec[i * n + j] = (1 / (dt * dt)) * bm[i * n + j] - (1 / (2 * dt)) * bc[i * n + j]; }
  }
  bat(vec, fd, vec3, n, 1, n);
  for(i = 0; i < n; i++) { vec1[i] = vec1[i] - vec2[i] - vec3[i]; }
  for(i = 0; i < n; i++)
  { for(j = 0; j < n; j++)
    { vec[i * n + j] = (1 / (dt * dt)) * bm[i * n + j] + (1 / (2 * dt)) * bc[i * n + j]; }
  }
  gsn(vec, vec1, n, 1, 0.0000001, key);
  for(i = 0; i < n; i++) { zd[i] = vec1[i]; }
  for(i = 0; i < n; i++)
  { a[i] = (1 / (dt * dt)) * (fd[i] - 2 * d[i] + zd[i]);
    v[i] = (1 / (2 * dt)) * (- fd[i] + zd[i]);
  }
  fp = fopen(outfile, "a"); /* store data in user's file */
  if(fp != (void *) NULL)
  { fprintf(fp, "%5d", l);
    fprintf(fp, "%12.4f", xg[l]);
    fprintf(fp, "%12.4f", d[layer - 1]);
  }
}

```

```

        fprintf(fp, "%12.4f", v[layer - 1]);
        fprintf(fp, "%12.4f", a[layer - 1]);
        fprintf(fp, " \n");
    }
    else printf("Can't open this outfile \n");
    fclose(fp);
    for(i = 0; i < n; i++)
    { d[i] = zd[i];
      fd[i] = d[i];
    }
}

void bat(float a[], float b[], float c[], int m, int n, int k)
{ int i, j, l;
  float s;
  for(i = 0; i < m; i++)
  { for(j = 0; j < n; j++)
    { s = 0.0;
      for(l = 0; l < k; l++)
      { s = s + a[i * k + l] * b[l * n + j]; }
      c[i * n + j] = s;
    }
  }
}

void gsn(float a[], float b[], int n, int nl, float eps, int key)
{ int i, j, k, l, m;
  float p, q;
  for(m = 0; m < n; m++)
  { p = 0.0;
    for(i = m; i < n; i++)
    { for(j = 0; j < n; j++)
      { if(fabs(a[i * n + j]) <= p) goto loop1;
        p = fabs(a[i * n + j]);
        k = i;
        l = j;
      }
    }
  }
loop1:;
}

```

```

key = 0;
if(p <= eps) return;
for(j = 0; j < n; j++)
{ q = a[k * n + j];
  a[k * n + j] = a[m * n + j];
  a[m * n + j] = q;
}
for(j = 0; j < n1; j++)
{ q = b[k * n1 + j];
  b[k * n1 + j] = b[m * n1 + j];
  b[m * n1 + j] = q;
}
for(i = 0; i < n; i++)
{ q = -a[i * n + 1] / a[m * n + 1];
  if(i == m) goto loop2;
  for(j = 0; j < n; j++)
  { a[i * n + j] = a[i * n + j] + a[m * n + j] * q; }
  for(j = 0; j < n1; j++)
  { b[i * n1 + j] = b[i * n1 + j] + b[m * n1 + j] * q; }
loop2:;
}
q = a[m * n + 1];
for(j = 0; j < n; j++)
{ a[m * n + j] = a[m * n + j] / q; }
for(j = 0; j < n1; j++)
{ b[m * n1 + j] = b[m * n1 + j] / q; }
}
for(j = 0; j < n; j++)
{ for(i = 0; i < n; i++)
  { if(a[i * n + j] < 0.5) goto loop3;
    for(k = 0; k < n; k++)
    { q = a[i * n + k];
      a[i * n + k] = a[j * n + k];
      a[j * n + k] = q;
    }
    for(k = 0; k < n1; k++)
    { q = b[i * n1 + k];
      b[i * n1 + k] = b[j * n1 + k];

```

```

        b[j * n1 + k] = q;
    }
loop3:;
    }
    }
    key = 1;
    return;
}

```

2. 线性加速度法

这个方法的基本思路是把整个振动时程分成很多个时间间隔 Δt , 并假定在 Δt 范围内加速度按直线规律变化, 在此基础上计算出 Δt 时刻内的增量位移、增量速度和增量加速度, 一步一步地求得整个时程的反应。

将(8.14)式的动力方程写成增量形式的方程:

$$[M]\{\Delta\delta(t)\} + [C]\{\Delta\dot{\delta}(t)\} + [K]\{\Delta\delta(t)\} = \{\Delta P(t)\} \quad (8.25)$$

用 t_{i-1} 时刻的 $\{\dot{\delta}\}$ 和 $\{\ddot{\delta}\}$ 表示 $\{\Delta\dot{\delta}(t)\}$ 和 $\{\Delta\ddot{\delta}(t)\}$, 代入(8.25)并整理后可得^[4]

$$\left. \begin{aligned} [K]^* \{\Delta\delta(t)\} &= \{\Delta P(t)\}^* \\ [K]^* &= [K] + \frac{3}{\Delta t}[C] + \frac{6}{\Delta t^2}[M] \\ \{\Delta P(t)\}^* &= \{\Delta P(t)\} + [M]\left\{\frac{6}{\Delta t}\{\dot{\delta}(t_{i-1})\} + 3\{\ddot{\delta}(t_{i-1})\}\right\} \\ &\quad + [C]\left\{3\{\dot{\delta}(t_{i-1})\} + \frac{\Delta t}{2}\{\ddot{\delta}(t_{i-1})\}\right\} \end{aligned} \right\} \quad (8.26)$$

在求出 $\{\Delta\delta(t)\}$ 后, $\{\Delta\dot{\delta}(t)\}$ 及 $\{\Delta\ddot{\delta}(t)\}$ 可按式求出:

$$\left. \begin{aligned} \{\Delta\dot{\delta}(t)\} &= \frac{3}{\Delta t}\{\Delta\delta(t)\} - 3\{\dot{\delta}(t_{i-1})\} - \frac{\Delta t}{2}\{\ddot{\delta}(t_{i-1})\} \\ \{\Delta\ddot{\delta}(t)\} &= \frac{6}{\Delta t^2}\{\Delta\delta(t)\} - \frac{6}{\Delta t}\{\dot{\delta}(t_{i-1})\} - 3\{\ddot{\delta}(t_{i-1})\} \end{aligned} \right\} \quad (8.27)$$

这样, t 时刻的位移、速度和加速度可按式求出:

$$\begin{aligned} \{\delta(t)\} &= \{\delta(t_{i-1})\} + \{\Delta\delta(t)\} \\ \{\dot{\delta}(t)\} &= \{\dot{\delta}(t_{i-1})\} + \{\Delta\dot{\delta}(t)\} = \frac{3}{\Delta t}\{\Delta\delta(t)\} - 2\{\dot{\delta}(t_{i-1})\} - \frac{\Delta t}{2}\{\ddot{\delta}(t_{i-1})\} \\ \{\ddot{\delta}(t)\} &= \{\ddot{\delta}(t_{i-1})\} + \{\Delta\ddot{\delta}(t)\} = \frac{6}{\Delta t^2}\{\Delta\delta(t)\} - \frac{6}{\Delta t}\{\dot{\delta}(t_{i-1})\} - 2\{\ddot{\delta}(t_{i-1})\} \end{aligned} \quad (8.28)$$

重复上述步骤, 可根据体系的初始条件, 一步一步地求得各时刻 $t_i (i = 1, 2, \dots, n)$ 时系统的动力位移、速度和加速度反应。

3. Wilson- θ 法

数值计算方法的一个基本要求是算法的收敛性好, 上一节介绍的线性加速度法当体系自振周期较短而计算步长较大时, 有可能出现计算过程发散的情况, 即计算的反应数值

越来越大,直至溢出(overflow),对于多自由度系统,其最小的自振周期可能很小,此时,计算步长 Δt 必须取得很小才能保证计算不发散。对于结构抗震分析来说, Δt 需要选得比地面运动中高频分量的周期以及结构的自振周期小很多倍(例如 10 倍以上),才能保证必要的精确度。因此,线性加速度法是一种条件收敛的算法^[45]。

Wilson- θ 法是在线性加速度法基础上改进得到的一种无条件收敛的数值方法,它的基本假定仍然是加速度按线性变化但其范围延伸到时间步长为 $\theta\Delta t$ 的区段,只要参数 θ 取得合适($\theta \geq 1.37$),就可以取得收敛的计算结果。当然, Δt 取得较大时,计算误差也将较大。

在时刻 $t + \theta\Delta t$, 多自由度系统的运动方程式为

$$[M]\{\ddot{\delta}(t + \theta\Delta t)\} + [C]\{\dot{\delta}(t + \theta\Delta t)\} + [K]\{\delta(t + \theta\Delta t)\} = \{P(t + \theta\Delta t)\}$$

根据 Wilson- θ 法的基本假定,加速度反应在 $[t, t + \theta\Delta t]$ 上线性变化,即在此区段上运用线性加速度法得到的公式,并将时间步长改为 $\theta\Delta t$,即可求得时刻 $t + \theta\Delta t$ 时的加速度反应为

$$\{\ddot{\delta}(t + \theta\Delta t)\} = \frac{6}{(\theta\Delta t)^2}(\{\delta(t + \theta\Delta t)\} - \{\delta(t)\}) - \frac{6}{\theta\Delta t}\{\dot{\delta}(t)\} - 2\{\ddot{\delta}(t)\}$$

在 $[t, t + \theta\Delta t]$ 时段内采用内插法,可以求得 $t + \Delta t$ 时刻的加速度为

$$\begin{aligned}\{\ddot{\delta}(t + \Delta t)\} &= \{\ddot{\delta}(t)\} + \frac{1}{\theta}(\{\ddot{\delta}(t + \theta\Delta t)\} - \{\ddot{\delta}(t)\}) \\ &= \{\ddot{\delta}(t)\} + \frac{1}{\theta} \left[\frac{6}{(\theta\Delta t)^2}(\{\delta(t + \theta\Delta t)\} - \{\delta(t)\}) - \frac{6}{\theta\Delta t}\{\dot{\delta}(t)\} - 3\{\ddot{\delta}(t)\} \right] \\ &= \frac{6}{\theta(\theta\Delta t)^2}(\{\delta(t + \theta\Delta t)\} - \{\delta(t)\}) - \frac{6}{\theta(\theta\Delta t)}\{\dot{\delta}(t)\} + \left(1 - \frac{3}{\theta}\right)\{\ddot{\delta}(t)\}\end{aligned}\quad (8.29)$$

根据线性加速度法的基本关系式,利用 $\{\ddot{\delta}(t + \Delta t)\}$ 可得

$$\left. \begin{aligned}\{\dot{\delta}(t + \Delta t)\} &= \{\dot{\delta}(t)\} + \frac{\Delta t}{2}(\{\ddot{\delta}(t + \Delta t)\} + \{\ddot{\delta}(t)\}) \\ \{\delta(t + \Delta t)\} &= \{\delta(t)\} + \Delta t\{\dot{\delta}(t)\} + \frac{1}{3}\Delta t^2\{\ddot{\delta}(t)\} + \frac{1}{6}\Delta t^2\{\ddot{\delta}(t + \Delta t)\}\end{aligned}\right\} \quad (8.30)$$

式(8.29)、(8.30)即为用 Wilson- θ 法计算结构动力反应的公式。根据这些公式和逐步积分的步骤,用 C 语言编制的计算程序如下:

```
// Calculate structural response with wilson-cta method
#include "math.h"
#include "stdlib.h"
#include "stdio.h"
#include "conio.h"

void stepwil(int n, float dt, float x3[], float x4[], float x5[], float x6[],
             float p[], float q[], float x0[], float v0[], float a0[], float x[],
             float v[], float a[], float bm[], float bk[], float bc[], float b0,
             float b1, float b3, float b4, float b5, float b6, float b7, float b8);

void bat(float a[], float b[], float c[], int m, int n, int k);
void gsn(float a[], float b[], int n, int n1, float eps, int key);
```

```

void wilson(int n,int id ,int id2,int ni,float dt,float cta,float x0[],
            float v0[],float a0[],float x[],float v[],float a[],
            float m[],float k[],float c[],float q[],float xg[]);

void main()
{ int i,n = 3;
  int id,id2,ni = n - 1;
  float q[3][3],xg[2100];
  float x[3],v[3],a[3];
  float dt = 0.01,cta = 1.4;
  static float x0[3] = {0.0,0.0,0.0};
  static float v0[3] = {0.0,0.0,0.0};
  static float a0[3] = {0.0,0.0,0.0};
  static float m[3][3] = {{7.48,0.0,0.0},{0.0,7.48,0.0},{0.0,0.0,7.48}};
  static float k[3][3] = {{12000.0, -6000.0,0.0},{-6000.0,12000.0, -6000.0},{0.0,
    -6000.0,6000.0}};
  static float c[3][3] = {{0.0,0.0,0.0},{0.0,0.0,0.0},{0.0,0.0,0.0}};
  FILE *fp;
  if((fp = fopen("el _ cen.dat","r")) == NULL) /* Input the EI _ centro
    value */
  { clrscr();
    printf("\n\n\n\t\t File \ 'el _ cen.dat \ 'can't open!!");
    exit(1);
  }
  i = 0; id = 0;
  while(!feof(fp))
  { fscanf (fp,"%f",&xg[i]);
    xg[i] = xg[i] * 10.0;
    i ++; id ++;
  }
  fclose(fp);
  wilson(n,id,id2,ni,dt,cta,x0,v0,a0,x,v,a,m[0],k[0],c[0],q[0],xg);
}

void wilson(int n,int id,int id2,int ni,float dt,float cta,float x0[],
            float v0[],float a0[],float x[],float v[],float a[],
            float m[],float k[],float c[],float q[],float xg[])

{ int i,jl,iw,layer;
  float tau,z0;
  float b0,b1,b2,b3,b4,b5,b6,b7,b8;

```

```

float x3[100],x4[100],x5[100],x6[100],p[100];
char outfile[20];
FILE *fp;
    clrscr();
    gotoxy(12,2);
    printf("PLS enter file name of answer:");
    scanf("%s",&outfile);
    gotoxy(12,4);
    printf("PLS enter the layer:");
    scanf("%d",&layer);
    fp = fopen(outfile,"w"); /* store data in user's file */
    if(fp! = (void *)NULL)
        { fprintf(fp,"The answer of structural response \n");
          fprintf(fp,"The layer is: %5d \n",layer);
        }
    else printf("Can't open this outfile \n");
    fclose(fp);

tau = cta * dt;
b0 = 6.0/(tau * tau);
b1 = 3.0/tau;
b2 = 2.0 * b1;
b3 = tau/2.0;
b4 = b0/cta;
b5 = - b2/cta;
b6 = 1.0 - 3.0/cta;
b7 = dt/2;
b8 = dt * dt/6.0;
for(iw = 0;iw<id;iw++)
{ z0 = xg[iw + 2] - xg[iw + 1];
  if(ni>0) z0 = z0/float(ni);
  for(jl = 0;jl<ni;jl++)
  { for(i = 0;i<n;i++)
    { p[i] = - m[i * n + i] * (xg[iw + 1] + (cta + ni - 1) * z0);
      . stepwil(n,dt,x3,x4,x5,x6,p,q,x0,v0,a0,x,v,a,m,k,c,b0,b1,b3,b4,b5,
        b6,b7,b8);
    }
  }
  fp = fopen(outfile,"a"); /* store data in user's file */
  if(fp! = (void *)NULL)

```



```

        fprintf(fp, "%5d", iw);
        fprintf(fp, "%12.4f", xg[iw + 1]);
        fprintf(fp, "%12.4f", x[layer - 1]);
        fprintf(fp, "%12.4f", v[layer - 1]);
        fprintf(fp, "%12.4f", a[layer - 1]);
        fprintf(fp, "n");
    }
    else printf("Can't open this outfile \n");
    fclose(fp);
}

void stepwil(int n, float dt, float x3[], float x4[], float x5[], float x6[],
            float p[], float q[], float x0[], float v0[], float a0[], float x[],
            float v[], float a[], float bm[], float bk[], float bc[], float b0,
            float b1, float b3, float b4, float b5, float b6, float b7, float b8)
{
    int i, j;
    int key;
    for(i = 0; i < n; i++)
    {
        for(j = 0; j < n; j++)
        {
            q[i * n + j] = bk[i * n + j] + b1 * bc[i * n + j] + b0 * bm[i * n + j];
        }
        for(i = 0; i < n; i++)
        {
            x3[i] = 2.0 * (a0[i] + b1 * v0[i]) + b0 * x0[i];
            x4[i] = b3 * a0[i] + 2.0 * v0[i] + b1 * x0[i];
        }
        bat(bm, x3, x5, n, 1, n);
        bat(bc, x4, x6, n, 1, n);
        for(i = 0; i < n; i++)
        {
            x3[i] = p[i] + x5[i] + x6[i];
        }
        gsn(q, x3, n, 1, 0.0000001, key);
        for(i = 0; i < n; i++)
        {
            a[i] = b4 * (x3[i] - x0[i] + b5 * v0[i] + b6 * a0[i];
            v[i] = v0[i] + b7 * (a[i] + a0[i]);
            x[i] = x0[i] + dt * v0[i] + b8 * (a[i] + 2.0 * a0[i]);
        }
        for(i = 0; i < n; i++)
        {
            x0[i] = x[i];
            v0[i] = v[i];
        }
    }
}

```

```

        a0[i] = a[i];
    }
}

void bat(float a[], float b[], float c[], int m, int n, int k)
{ int i, j, l;
  float s;
  for(i = 0; i < m; i++)
  { for(j = 0; j < n; j++)
    { s = 0.0;
      for(l = 0; l < k; l++)
      { s = s + a[i * k + l] * b[l * n + j]; }
      c[i * n + j] = s;
    }
  }
}

void gsn(float a[], float b[], int n, int nl, float eps, int key)
{ int i, j, k, l, m;
  float p, q;
  for(m = 0; m < n; m++)
  { p = 0.0;
    for(i = m; i < n; i++)
    { for(j = 0; j < n; j++)
      { if(fabs(a[i * n + j]) <= p) goto loop1;
        p = fabs(a[i * n + j]);
        k = i;
        l = j;
      }
    }
    loop1:;
    key = 0;
    if(p <= eps) return;
    for(j = 0; j < n; j++)
    { q = a[k * n + j];
      a[k * n + j] = a[m * n + j];
      a[m * n + j] = q;
    }
    for(j = 0; j < nl; j++)
    { q = b[k * nl + j];

```

```

        b[k * n1 + j] = b[m * n1 + j];
        b[m * n1 + j] = q;
    }
    for(i = 0; i < n; i++)
    { q = -a[i * n + 1]/a[m * n + 1];
      if(i == m) goto loop2;
      for(j = 0; j < n; j++)
      { a[i * n + j] = a[i * n + j] + a[m * n + j] * q; }
      for(j = 0; j < n1; j++)
      { b[i * n1 + j] = b[i * n1 + j] + b[m * n1 + j] * q; }
loop2:;
    }
    q = a[m * n + 1];
    for(j = 0; j < n; j++)
    { a[m * n + j] = a[m * n + j]/q; }
    for(j = 0; j < n1; j++)
    { b[m * n1 + j] = b[m * n1 + j]/q; }
    }
    for(j = 0; j < n; j++)
    { for(i = 0; i < n; i++)
      { if(a[i * n + j] < 0.5) goto loop3;
        for(k = 0; k < n; k++)
        { q = a[i * n + k]
          a[i * n + k] = a[j * n + k];
          a[j * n + k] = q;
        }
        for(k = 0; k < n1; k++)
        { q = b[i * n1 + k];
          b[i * n1 + k] = b[j * n1 + k];
          b[j * n1 + k] = q;
        }
      }
loop3:;
    }
    }
    key = 1;
    return;
}

```

上述的三种数值计算方法适用于线性和非线性系统的动力分析。对于多自由度线性

系统来说,结构或单元的刚度矩阵为常数矩阵。但对非线性系统来说,尽管可以假定系统在 Δt 时段各单元的刚度变化可略去不计,但在不同的时段,各单元的刚度在变化,从而系统的总体刚度矩阵也在变化,在对系统进行非线性动力计算时,必须考虑如何对刚度矩阵进行修正。

修正刚度矩阵的过程实质上是一个重新形成总刚度矩阵的过程,在结构系统中单元的划分不再改变的情况下,一般是对单元刚度进行修正,以反映非线性变形的影响。

当采用平面应力(或应变)单元时,应根据所采用的应力-应变的滞回关系,按照各个时刻的应力状态及应变速率确定单元的弹塑性参数(例如 E, γ, G 等),具体的修正方法可参见本书第五章及第七章的有关内容。

当采用平面受弯梁单元时,一般是选用了截面上弯矩-曲率的滞回关系,这时应根据曲率的大小和变化来确定单元的弹塑性参数(例如 EI, EA 等)。但在采用梁单元的有限元计算中,求得的变形是截面的转角 θ ,从转角求曲率 ϕ 时又要用到截面的 EI ,因此,从 $\theta \rightarrow \phi \rightarrow EI$ 时还需要进行迭代,才能得到比较合理的 EI 。

当采用层间模型时,可以采用层间力与层间位移的滞回关系,这时的单元刚度矩阵修正过程就比较简单,下面以常用的三线型刚度半退化恢复力模型为例作以介绍。

对于图 8.7 所示的恢复力模型^[45], P_c, P_y 和 P_u 分别为构件开裂、屈服和极限强度, δ_c, δ_y 和 δ_u 为相应的位移。这个模型的刚度取值规则如下:① 当计算的层间力 $P \leq P_c$ 时,加载及卸载刚度均取 k_1 ;当计算的层间剪力 $P_c < P \leq P_y$ 时,加载刚度取 k_2 ,卸载刚度取 k_1 ,并对计算的层间力进行修正。② 当计算的层间力 $P > P_y$ 时,加载刚度取 k_3 ,卸载刚度取 k_{12} ,并对计算的层间力进行修正。③ 当反向变形及 P 为负值时,加载、卸载刚度按类似规则确定。这里的加载是指变形的绝对值沿某一方向增大,卸载指变形绝对值沿该方向减小。例如当第 i 步计算的层间位移为 30mm 而第 $i+1$ 步的层间位移大于 30mm 时,称之为加载,若 $i+1$ 步计算的层间位移小于 30mm(例如 28mm)时,称之为卸载。类似地,当第 j 步计算的层间位移为 -30mm 而第 $j+1$ 步计算的层间位移为 -32mm(绝对值大于前一步

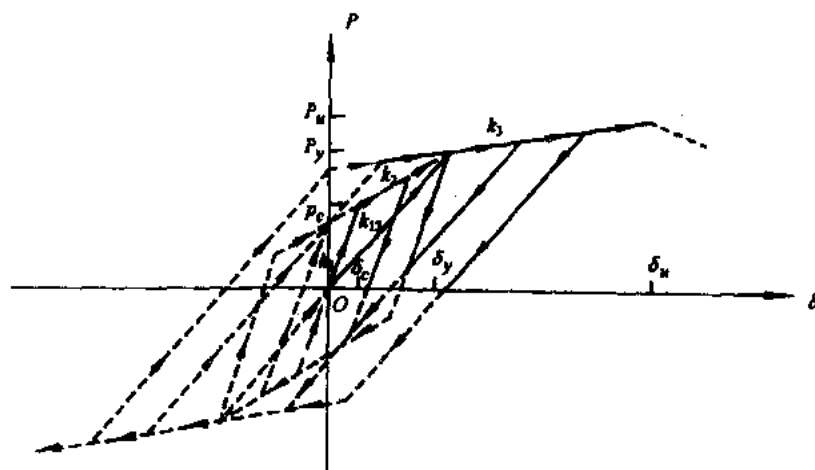


图 8.7 三线型刚度半退化恢复力模型

位移的绝对值)时,称为加载,反之称为卸载。当然,如计算得到的位移正好经过零点而无法根据位移绝对值判别时,还应根据变形速度 $\dot{x}$ 的方向来判别。非线性动力计算时的刚度修正是一个难度较大的问题,特别是出现负刚度时计算难度更大。应根据特定的问题和所选用的模型进行反复调试,直到获得满意的结果。根据作者多年的计算体会,当恢复力模型选取合适、数值计算方法正确、时间步长取得小一些时(例如 $\Delta t \leq 0.001$ 秒时),一般情况下总会得到与动力试验数据符合较好的结果。

研究表明,结构变形进入负刚度条件下各种数值积分方法的稳定性与正刚度条件下有所不同。表 8.1 是文献[47]的研究结果,可供读者在进行非线性动力计算时参考。

表 8.1 部分数值积分法的稳定性

		正刚度	负刚度
中心差分法		条件稳定	无条件稳定
Wilson- θ 法	$\theta \geq 1.37$	无条件稳定	条件稳定
	$\theta < 1.37$	条件稳定	条件稳定

8.5 动力系统的简化方法

由前一节中介绍的动力分析过程可知,结构有限元动力分析的计算相当于多次的静力计算,当时间步长 Δt 很小时,计算工作量是相当大的。特别是进行非线性分析时,由于在每一个时间步长内都需要调整结构的刚度矩阵,或者进行迭代,这样,计算工作量将更大,花费的机时将更多。为了减少计算工作量和节省计算时间,一般可以采取两种途径,第一种是选用合理的数值方法和计算程序,第二种是从结构和力学的角度简化动力方程及缩减自由度。本节将简要介绍第二种途径。

8.5.1 集中质量矩阵

在本章 8.2.2 节中介绍的单元质量矩阵称为一致质量矩阵,它们是按照位移插值的形函数矩阵,根据分布惯性力的合理分配而求得的。单元的一致质量矩阵叠加形成的整个结构的质量矩阵,一般是稀疏的、带状的,都有一定的带宽,导致所谓的质量耦合,使得动力分析时的计算量较大。于是,人们试图找出一种比较简便的办法,设想把单元的质量,按照动能相等的原理,集中在单元的结点上,这样,单元的质量矩阵和叠加后形成的总体质量矩阵就是对角矩阵,使得计算工作量大大减少。

对于平面受弯梁单元,采用集中质量时的单元质量矩阵为

$$[M]^e = \frac{ml}{2} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (8.31)$$

对于平面 4 结点矩形单元,单元的集中质量矩阵为

$$[M]^e = \frac{\rho t A}{4} [I] \quad (8.32)$$

式中, $[I]$ 为 8 阶单位矩阵。

为了研究一致质量和集中质量对结构动力特性的影响, 作者计算了一个 10 层钢筋混凝土框架结构的前 20 阶自振频率和振型。该框架为典型的 3 跨 10 层平面框架, 第一阶自振频率为 1Hz 左右, 采用平面杆系单元计算, 计算时分别采用了一致质量矩阵和集中质量矩阵, 集中质量按杆件位置集中在单元的结点上。计算结果如图 8.8 所示, 图中横坐标为振型编号, 纵坐标为自振频率 (Hz)。由图可见, 采用一致质量和集中质量法时, 前三阶的自振频率几乎没有差别, 之后采用集中质量法时的自振频率略低一些。一般来说, 采用集中质量矩阵时低估了系统的自振频率, 而用一致质量矩阵算得的频率代表结构自振频率的上限值。

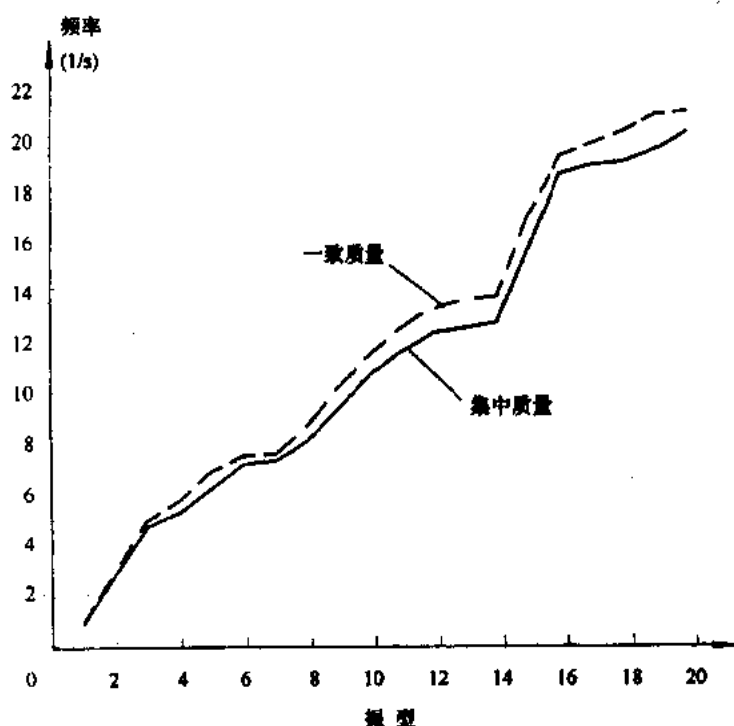


图 8.8 一致质量法与集中质量法的自振特性比较

在进行结构动力分析时, 采用集中质量法具有明显的优点, 其一是集中质量矩阵为对角矩阵, 无论是求解特征值还是动力方程的直接积分, 都可以节省很多的计算量; 其二是在某些单元中, 采用集中质量法时结点的转动自由度可以消去 (见下节介绍), 又可以进一步简化计算。

8.5.2 静力凝聚^[48]

当采用集中质量法时, 有一部分自由度方向上的集中质量为零, 如果再忽略阻尼项的

影响,就可以通过静力凝聚的方法减少动力方程的数量,以简化计算。在进行静力凝聚时,可以把有质量的结点位移集中在一起记为 $\{\delta\}_a$,没有质量的另一部分结点位移记为 $\{\delta\}_b$,将动力方程写为

$$\begin{bmatrix} M_{aa} & 0 \\ 0 & 0 \end{bmatrix} \begin{Bmatrix} \ddot{\delta}_a \\ \ddot{\delta}_b \end{Bmatrix} + \begin{bmatrix} k_{aa} & k_{ab} \\ k_{ba} & k_{bb} \end{bmatrix} \begin{Bmatrix} \delta_a \\ \delta_b \end{Bmatrix} = \begin{Bmatrix} P_a \\ P_b \end{Bmatrix}$$

或展开后写为

$$[M_{aa}]\{\ddot{\delta}\}_a + [k_{aa}]\{\delta\}_a + [k_{ab}]\{\delta\}_b = \{P\}_a \quad (8.33)$$

$$[k_{ba}]\{\delta\}_a + [k_{bb}]\{\delta\}_b = \{P\}_b \quad (8.34)$$

式(8.34)是一个代数方程,由此可以解出 $\{\delta\}_b$ 为

$$\{\delta\}_b = [k_{bb}]^{-1}(\{P\}_b - [k_{ba}]\{\delta\}_a) \quad (8.35)$$

将(8.35)式代入(8.33)式,可得到

$$\begin{aligned} & [M_{aa}]\{\ddot{\delta}\}_a + ([k_{aa}] - [k_{ab}][k_{bb}]^{-1})\{\delta\}_a \\ & = \{P\}_a - [k_{ab}][k_{bb}]^{-1}\{P\}_b \end{aligned} \quad (8.36)$$

这是一个只包括有质量的结点的位移 $\{\delta\}_a$ 的微分方程组,可以按8.4节中的方法求解。由于方程中的未知数减少了,因此计算工作量也相应地减少。

由(8.36)式解出 $\{\delta\}_a$ 后,即可代入(8.35)式,求解没有质量的结点的位移 $\{\delta\}_b$ 。这种通过求解静力平衡方程式(如(8.34)式)而达到使动力微分方程组降阶的方法称为静力凝聚。在钢筋混凝土框架、板壳类结构的有限元动力分析中,往往可以忽略对应于结点转角的转动惯量,这时可以按静力凝聚的方法把结点转角消去,只求解对应于结点线位移的降阶的微分方程。如果结点上没有外弯矩作用,式(8.34)中的 $\{P\}_b$ 为零,计算工作量还可以减少。

8.5.3 层间模型

在高层建筑的抗震动力分析中,目前国内外使用较多的是以假定楼面刚度无限大为基础的层间模型,即以每一层楼面为一个集中质量块,每层楼面考虑1~3个自由度。这种模型的优点是大大压缩了自由度的数量,适合于计算整体结构的动力特性和动力反应,找出结构的薄弱层,也适合于初步设计阶段的计算。但剪切型层间模型不能考虑整体弯曲的影响,弯剪型层间模型只能考虑相邻层的影响,而且无法判断每根杆的工作状态,因此局限性较大。

为了解决这一问题,研究人员发展了一种杆系-层间模型^[49],即每层考虑一个集中质量,而层间的刚度则由杆系形成。计算的大体过程是:首先按杆系形成每片抗侧力结构的总刚度矩阵,然后用分别在各层处作用单位水平力的方法求出阶数等于层数的抗侧力结构的片柔度矩阵;片柔度矩阵求逆即得到片刚度矩阵;利用楼板在平面内刚度无限大的假定,由各片抗侧力结构的片刚度矩阵形成整个结构的总刚度矩阵。这时,以质量集中于楼面处的层间模型来求解运动方程,得到某一时刻的总体位移后再回到杆系,考虑各片抗侧力结构的协同工作,按片刚度矩阵分配各片所承受的外力,再解出内力,以此内力来判断每根杆件所处的弹塑性(非线性)状态,重新形成每个杆件的刚度矩阵进而组

成新的空间总刚度矩阵。这样往返计算,直至达到结构强度极限或变形极限或计算完输入地震波的全部时程为止。这种杆系 - 层间模型的特点可以概括为:静按杆系,动按层间,分别判断,合并运动。关于这一模型的详细计算过程可参阅文献[49]。

尽管数值计算方法在不断地改进,计算机容量及其速度在不断地增加,但以层间模型为基础的各种空间协同模型仍是计算高层建筑的主要手段,它遍及于各种比较成熟的商用程序之中,如 TBSA, TAT, SAP84, ETABS 以及 IBM 公司与 SOM 设计公司推出的 A&ES 应用软件。了解这些商用程序的基本假定和所采用的计算模型,可以帮助使用者正确分析计算结果,以便在工程应用及研究工作中正确地使用这些计算结果。

附录 A

附录 A 钢筋混凝土剪力墙结构非线性 有限元分析源程序

F.E.Nonlinear Analysis of shear wall structure

```
# include<stdio.h>
# include<stdlib.h>
# include<math.h>
# include<graphics.h>
# include<conio.h>
# include<string.h>
# include<alloc.h>
# define pi 3.1415926
FILE *opf, *opf1, *opf2, *opf3, *opf4;
FILE      *ipf, *ipf1;
int      _n_element, _n_node, _n_fixed,      // 单元总数, 结点总数, 固定结点数
        _address[1000], _node_order[500][2], // 总刚对角地址, 结点序号排列矩阵
        _discr_fixed[20][3];                // 结点固定形式描述矩阵
float     huge * _pload_bak;
float     huge * _k;                          // 总刚矩阵数组指针
float     huge * * _cstress;                  // 混凝土应力矩阵数组指针
float     huge * * _sstress;                  // 钢筋应力矩阵数组指针
float     huge * * _strain;                   // 应变矩阵数组指针
float     huge * * _inc_strain;               // 应变增量矩阵数组指针
float     _Ec0, _Es0,                        // 混凝土初始弹性模量, 钢筋弹性模量
        _cstrain_c, _cstress_fc,            // 混凝土极限应变, 混凝土抗压强度
        _cstress_ft, _sstress_s, _u,        // 抗拉强度, 钢筋屈服强度, 泊松比
        _x[500],                            // 结点 X 坐标
        _y[500],                            // 结点 Y 坐标
double    _pload[1000];                      // 荷载、位移向量矩阵
struct    _element_describe                  // 单元几何、材料状况描述数据结构
{
    char    cstate1[4], cstate2[4],          // 混凝土 1,2 方向加、卸载状态参数
            sstate1[4], sstate2[4];         // 钢筋 r,s 方向加、卸载状态参数
```

```

        crack1[4],crack2[4],      // 混凝土 1,2 方向开裂状态参数
    int    node[4];                // 各单元所对应的 4 个结点号
    float  thick, stratio _ r, sratio _ s, sangle _ rx;
                                     // 单元厚度, r, s 方向配筋率, r 与 x 轴
                                     // 夹角
    float  Ec1[4], Ec2[4],        // 1,2 方向混凝土弹性模量
           Esr[4], Ess[4],        // r, s 方向钢筋弹性模量
           cangle[4], Gc[4],      // 第 1 主应力方向与 x 轴夹角, 混凝土
                                     // 剪切模量
           mcstress1[4], mcstress2[4], // 混凝土 1,2 方向曾达到的主应力最
                                     // 大值
           mcstrain1[4], mcstrain2[4], // 混凝土 1,2 方向曾达到的等效应变
                                     // 大值
           lcstress1[4], lcstress2[4], // 混凝土 1,2 方向曾达到的主应力最
                                     // 小值
           lcstrain1[4], lcstrain2[4], // 混凝土 1,2 方向曾达到的等效应变
                                     // 最小值
           mainstress1[4], mainstress2[4], // 混凝土 1,2 方向主应力值
           eqcstrain1[4], eqcstrain2[4], // 混凝土 1,2 方向等效应变值
           baseangle[4],            // 主应力方向基准角
           iEc1[4], iEc2[4],        // 1,2 方向混凝土加载情况下对应弹
                                     // 性模量
           dEc1[4], dEc2[4],        // 1,2 方向混凝土卸载情况下对应弹
                                     // 性模量
           iEsr[4], iEss[4],        // r, s 方向钢筋加载情况下对应弹性
                                     // 模量
           dEsr[4], dEss[4],        // r, s 方向钢筋卸载情况下对应弹性
                                     // 模量
};

struct    element _ describe huge * _ element;
void reading();                // 数据读入函数
void address _ k();            // 求一维压缩储存总刚主对角地址函
                                // 数
void form _ k();               // 生成总刚矩阵函数
void ldlt _ solve();           // LDLT 求解总刚矩阵函数
void initial _ farray(int, float * p); // 数组初始化函数
void multi _ fmatrix(int a, int b, int c, float * aa, float * bb, float * cc);
                                // 矩阵相乘函数

```

```

void form_D(int,int,float D[3][3]);           // 生成单元刚度矩阵函数
void load_vect(int,int,float);               // 生成荷载向量函数
void stress_strain(int);                      // 求应力、应变函数
void calculate_E(int,char,int);               // 求各单元高斯积分点处,各应力-应变阶段弹性模量

void main_stress(float * stressx,float * stressy,float * stressxy);
// 求主应力函数
void main_strain(float * strainx,float * strainy,float * strainxy);
// 求主应变函数
void angle_strain(float * stainx,float * strainy,float * strainxy,float angle);
// 求某角度应变函数
void angle_stress(float * stressx,float * stressy,float * stressxy,float angle);
// 求某角度应力函数
float form_B(int element,int gi,float B[3][8]); // return datj = 1 J 1
// 求几何刚度矩阵[B]及雅可比行列式
void form_Dcon(int element,int gi,float D[3][3]); // 生成单元混凝土刚度矩阵函数
void form_Dste(int element,int gi,float D[3][3]); // 生成单元钢筋刚度矩阵函数
void rejust_cstate(int);                      // 加卸状态校正,重新求解函数
main()
{
    char name[80],a[80],change = 1;
    int i,ii,iii,iiii,group_load,n_load,node[20],direction[20],nstep_load,total_step = 0;
    float b_load,e_load,inc_load[20],aa,bb = 0.0,cc = 0.0,endload,x,y;
    puts("\nInput *.nam file name:");
    gets(name);
    strcat(name,".nam");
    if((ipf = fopen(name,"r")) == NULL) // 打开数据文件组名文件
    {
        printf("Cannot %s open file! \n",name);
        exit(1);
    }
    reading();
    for(i = 0;i < n_element;i++)
    for(ii = 0;ii < 4;ii++)
    {
        _element[i].cstate1[ii] = 10;
        _element[i].cstate2[ii] = 10;
    }
}

```

```

for(i = 0; i < _n_element; i++)
for(ii = 0; ii < 4; ii++)
{
    _element[i].eqcstrain1[ii] = 0.0;
    _element[i].eqcstrain2[ii] = 0.0;
}
_inc_strain = (float huge * *)fcalloc(_n_element, sizeof(float *));
if(!_inc_strain){printf("_inc_strain allocation failure");exit(1);}
for(i = 0; i < _n_element; i++)
{
    _inc_strain[i] = (float huge *)fcalloc(12, sizeof(float));
    if(!_inc_strain[i]){printf("_inc_strain[%d] allocation failure", i);exit(1);}
}
address_k();
fscanf(ipf, "%s", name); // cstress date file
if((opf1 = fopen(name, "w+b")) == NULL){printf("\nCan not open opf1");exit(1);}
// 打开混凝土应力数据存储文件
fscanf(ipf, "%s", name); // sstress data file
if((opf2 = fopen(name, "w+b")) == NULL){printf("\nCan not open opf2");exit(1);}
// 打开钢筋应力数据存储文件
fscanf(ipf, "%s", name); // strain data file
if((opf3 = fopen(name, "w+b")) == NULL){printf("\nCan not open opf3");exit(1);}
// 打开应变数据存储文件
fscanf(ipf, "%s", name); // load data file
if((ipf1 = fopen(name, "r")) == NULL){printf("\nCan not open ipf1");exit(1);}
fscanf(ipf, "%s", name); // pd data file
if((opf4 = fopen(name, "w")) == NULL){printf("\nCan not open opf4");exit(1);}
fprintf(opf4, "0.0 0.0 \n");
fscanf(ipf, "%s", name);
fscanf(ipf1, "%d", &group_load); // 读入荷载组数
for(i = 0; i < group_load; i++)
{
    fscanf(ipf1, "%d", &n_load); // 读入荷载个数
    fscanf(ipf1, "%d", &nstep_load); // 读入加荷步数
    endlload = 0.0;
    for(ii = 0; ii < n_load; ii++)
    {
        fscanf(ipf1, "%d%d", &node[ii], &direction[ii]);
        // 荷载作用结点号,作用方向,0—X方向,1—Y方向
        fscanf(ipf1, "%f%f", &b_load, &e_load);
        // 开始荷载值,最终荷载值
    }
}

```

```

inc_load[ii] = (e_load-b_load)/nstep_load;
endload += e_load;
if(i>0)
{
    if(ii == 0)
    {
        if(aa/inc_load[0]<0.0)
            change = - 1;
        else
            change = 1;
    }
}

aa = inc_load[0];
for(ii = 0;ii<(nstep_load + 10)&&
(ii<nstep_load || (bb<endload&&inc_load[0]>0) || (bb>endload&&inc_load[0]<0));ii++)
{
    for(iii = 0;iii<2*_n_node;iii++) _pload[iii] = 0.0;
    for(iii = 0;iii<n_load;iii++)
    {
        load_vect(node[iii],direction[iii],inc_load[iii]) ;
        // 生成荷载向量
    }
}

calculate_E(i,ii,change,total_step);
if(total_step)
{
    for(iii = 0;iii<_n_element;iii++)
        farfree(_cstress[iii]);
    farfree(_cstress);
    for(iii = 0;iii<_n_element;iii++)
        farfree(_sstrs[iii]);
    farfree(_sstrs);
    for(iii = 0;iii<_n_element;iii++)
        farfree(_strain[iii]);
    farfree(_strain);
}

rejust_cstate(total_step);
for(iii = 0;iii<n_load;iii++)bb += inc_load[iii];
cc += _pload[_node_order[20][0]];
fprintf(opf4,"%f%f \ n",cc,bb);
total_step++;

```

```

    }
    fclose(ipf);
    fclose(ipfl);
    fclose(opfl);
    fclose(opf2);
    fclose(opf3);
    fclose(opf4);
    return 0;
}

void reading()
{
    char name[40],ch;
    int i,ii;
    fscanf(ipf,"%s",name);
    if((ipfl = fopen(name,"r")) == NULL)
    {
        printf("Cannot open file %s! \n",name);
        exit(1);
    }

    fscanf(ipfl,"%d",&_n_element);
    fscanf(ipfl,"%d",&_n_node);
    fscanf(ipfl,"%d",&_n_fixed);
    for(i = 0;i<_n_fixed;i++)
    {
        fscanf(ipfl,"%d",&_discr_fixed[i][0]);
        fscanf(ipfl,"%d",&_discr_fixed[i][1]);
        fscanf(ipfl,"%d",&_discr_fixed[i][2]);
    }

    fscanf(ipfl,"%f",&_Ec0);
    fscanf(ipfl,"%f",&_u);
    fscanf(ipfl,"%f",&_cstress_fc);
    fscanf(ipfl,"%f",&_cstress_ft);
    fscanf(ipfl,"%f",&_sstress_s);
    _cstress_ft = -_cstress_ft;
    _cstrain_c = 0.003;
    _element = (struct element _describe huge *)fcalloc(_n_element,sizeof(struct
        element _describe));

    // 动态内存分配,生成_element[_n_
    element] 结构数组

```

```

if(!_element){printf("_element allocation failure \n");exit(1);}
_pload_bak = (float huge *)fcalloc(2*_n_node,sizeof(float));
if(!_pload_bak){printf("_pload_bak allocation failure \n");exit(1);}
for(i = 0;i<_n_element;i++)
{
    fscanf(ipfl,"%f",&_element[i].thick);
}
for(i = 0;i<_n_element;i++)
    fscanf(ipfl,"%f",&_element[i].sangle_rx);
for(i = 0;i<_n_element;i++)
    fscanf(ipfl,"%f",&_element[i].stratio_r);
for(i = 0;i<_n_element;i++)
    fscanf(ipfl,"%f",&_element[i].sratio_s);
for(i = 0;i<_n_element;i++)
for(ii = 0;ii<4;ii++)
    fscanf(ipfl,"%d",&_element[i].node[ii]);
for(i = 0;i<_n_node;i++)    fscanf(ipfl,"%f",&_x[i]);
for(i = 0;i<_n_node;i++)    fscanf(ipfl,"%f",&_y[i]);
fclose(ipfl);
}

void initial_farray(int n,float *p)
{
    int i;
    for(i = 0;i<n;i++)
        *(p+i) = 0.0;
}

void multi_fmatrix(int a,int b,int c,float *aa,float *bb,float *cc)
{
    int i,ii,iii;
    initial_farray(a*b,cc);
    for(i = 0;i<a;i++)
    for(ii = 0;ii<b;ii++)
    for(iii = 0;iii<c;iii++)
        *(cc+b*i+ii) += *(aa+c*i+iii) * (*(bb+b*iii+ii));
}

void form_D(int element,int gi,float D[3][3])
{
    int i,ii;
    float c,s,T[3][3],aa[3][3],bb[3][3];
    c = _element[element].thick*_element[element].Ec1[gi]/(1._u*_u);

```

```

s = _element[element].thick * _element[element].Ec2[gi]/(1. - _u * _u);
if(c<0 || s<0)
    aa[0][1] = 0.0;
else
    aa[0][1] = _u * sqrt(c * s);
aa[0][0] = c; aa[0][2] = 0.0;
aa[1][0] = aa[0][1]; aa[1][1] = s; aa[1][2] = 0.0;
aa[2][0] = 0.0; aa[2][1] = 0.0;
aa[2][2] = _element[element].thick * _element[element].Gc[gi];
c = cos(_element[element].cangle[gi]);
s = sin(_element[element].cangle[gi]);
T[0][0] = c * c; T[0][1] = s * s; T[0][2] = - 2 * c * s;
T[1][0] = s * s; T[1][1] = c * c; T[1][2] = 2 * c * s;
T[2][0] = c * s; T[2][1] = - c * s; T[2][2] = c * c - s * s;
multi_fmatrix(3,3,3,t,aa,bb);
T[0][2] = c * s; T[2][0] = - 2 * c * s;
T[1][2] = - c * s; T[2][1] = 2 * c * s;
multi_fmatrix(3,3,3,bb,T,D);
initial_farray(9,aa);
aa[0][0] = _element[element].thick * _element[element].Esr[gi]
    * _element[element].stratio_r,
c = cos(_element[element].sangle_rx);
s = sin(_element[element].sangle_rx);
T[0][0] = c * c; T[0][1] = s * s; T[0][2] = - 2 * c * s;
T[1][0] = s * s; T[1][1] = c * c; T[1][2] = 2 * c * s;
T[2][0] = c * s; T[2][1] = - c * s; T[2][2] = c * c - s * s;
multi_fmatrix(3,3,3,T,aa,bb);
T[0][2] = c * s; T[2][0] = - 2 * c * s;
T[1][2] = - c * s; T[2][1] = 2 * c * s;
multi_fmatrix(3,3,3,bb,T,aa);
for(i = 0;i<3;i++)
    for(ii = 0;ii<3;ii++)
        D[i][ii] += aa[i][ii];
aa[0][0] = _element[element].thick * element[element].Ess[gi]
    * _element[element].sratio_s;
c = cos(_element[element].sangle_rx + 3.1415926/2);
s = sin(_element[element].sangle_rx + 3.1415926/2);
T[0][0] = c * c; T[0][1] = s * s; T[0][2] = - 2 * c * s;

```



```

T[1][2] = s * s;  T[1][1] = c * c;  T[1][2] = 2 * c * s;
T[2][0] = c * s;  T[2][1] = - c * s;  T[2][2] = c * c - s * s;
multi_fmatrix(3,3,3,T,aa,bb);
T[0][2] = c * s;  T[2][0] = - 2 * c * s;
T[1][2] = - c * s;  T[2][1] = 2 * c * s;
multi_fmatrix(3,3,3,bb,T,aa);
for(i = 0;i<3;i++)
for(ii = 0;ii<3;ii++)
    D[i][ii] += aa[i][ii];
}

void load_vect(int node,int direction,float inc_load)
{
    if(direction == 0)
        _pload[_node_order[node-1][0]] = inc_load;
    else
        _pload[_node_order[node-1][1]] = inc_load;
}

void stress_strain(int total_step)
{
    int i,ii,iii;
    float aa[8],bb,D[3][3],B[3][8],cc[3];
    _cstress = (float huge * *)fcalloc(_n_element,sizeof(float *));
    if(!_cstress){printf("_cstress allocation failure");exit(1);}
    _sstrress = (float huge * *)fcalloc(_n_element,sizeof(float *));
    if(!_sstrress){printf("_sstrress allocation failure");exit(1);}
    _strain = (float huge * *)fcalloc(_n_element,sizeof(float *));
    if(!_strain){printf("_strain allocation failure");exit(1);}
    for(i = 0;i<_n_element;i++)
    {
        _cstress[i] = (float huge *)fcalloc(12,sizeof(float));
        if(!_cstress[i]){printf("_cstress[%d]allocation failure",i);exit(1);}
        _sstrress[i] = (float huge *)fcalloc(12,sizeof(float));
        if(!_sstrress[i]){printf("_sstrress[%d]allocation failure",i);exit(1);}
        _strain[i] = (float huge *)fcalloc(12,sizeof(float));
        if(!_strain[i]){printf("_strain[%d]allocation failure",i);exit(1);}
    }
    if(total_step!= 0)
    {
        rewind(opfl);
        for(i = 0;i<_n_element;i++)
            fread(_cstress[i],4,12,opfl);
    }
}

```

```

rewind(opf2);
for(i = 0; i < _n_element; i++)
    fread(_sstress[i], 4, 12, opf2);
rewind(opf3);
for(i = 0; i < _n_element; i++)
    fread(_strain[i], 4, 12, opf3);
}
for(i = 0; i < _n_element; i++)
{
    for(ii = 0; ii < 4; ii++)
    {
        for(iii = 0; iii < 4; iii++)
        {
            aa[2 * iii] = _pload[_node_order[_element[i].node[iii] - 1][0]];
            aa[2 * iii + 1] = _pload[_node_order[_element[i].node[iii] - 1][1]];
        }
    }
    form _B(i, ii, B);
    multi _fmatrix(3, 1, 8, B, aa, cc); // calculator strain
    for(iii = 0; iii < 3; iii++)
    {
        cc[iii] = - cc[iii]; // extending strain is negative
        _strain[i][ii * 3 + iii] += cc[iii];
        _inc_strain[i][ii * 3 + iii] = cc[iii];
    }
    form _Dcon(i, ii, D);
    multi _fmatrix(3, 1, 3, D, cc, aa);
    for(iii = 0; iii < 3; iii++)
        _cstress[i][ii * 3 + iii] += aa[iii];
    form _Dste(i, ii, D);
    multi _fmatrix(3, 1, 3, D, cc, aa);
    for(iii = 0; iii < 3; iii++)
        _sstress[i][ii * 3 + iii] += aa[iii];
}
}
}

void main _stress(float * stressx, float * stressy, float * stressxy)
{
    float aa, bb, cc;
    if(fabs(* stressx - * stressy) < 1.0e - 30)
    {
        if(* stressxy > 1.0e - 5)
        {
            cc = pi/4.0;
        }
        else if(* stressxy < - 1.0e - 5)

```

```

    {   cc = - pi/4.0;
    }

    else // * stressxy = 0.0
    {   cc = 0.0;
    }

    else
    {   cc = 0.5 * atan(2.0 * (* stressxy)/(* stressx - * stressy));
        if(fabs(cc)<1e - 2)
        {   if(( * stressx - * stressy)>0)
                cc = 0.0;

                else
                cc = - pi/2;

        }
        else if(cc>0.0)
        {   if(( * stressx - * stressy)<0)
                cc = pi/2.0;

        }
        else
        {   if(( * stressx - * stressy)<0)
                cc += pi/2.0;

        }
    }

    aa = * stressx + * stressy;
    bb = * stressx - * stressy;
    bb = sqrt(bb * bb + 4. * (* stressxy) * (* stressxy));
    * stressxy = cc;
    * stressx = 0.5 * aa + 0.5 * bb;
    * stressy = 0.5 * aa - 0.5 * bb;
}

void main _ strain(float * strainx, float * strainy, float * strainxy)
{   float aa, bb, cc;
    if(fabs(* strainx - * strainy)<1.0e - 8)
    {   if(* strainxy>1.0e - 8)
            cc = pi/4.0;
        else if(* strainxy<1.0e - 8)
            cc = - pi/4.0;
        els // * strainxy = 0.0

```

```

        cc = 0.0;
    }
    else
    {
        cc = 0.5 * atan(( * strainxy) / (* strainx - * strainy));
        if(fabs(cc) < 1.0e - 2)
        {
            if( * strainx - * strainy > 0)
                cc = 0.0;

            else
                cc = - pi/2;
        }
        else if(cc > 0)
        {
            if(( * strainx - * strainy) < 0)
                cc -= pi/2.0;
        }
        else
        {
            if(( * strainx - * strainy) < 0)
                cc += pi/2.0;
        }
    }

    aa = * strainx + * strainy;
    bb = * strainx - * strainy;
    bb = sqrt(bb * bb + ( * strainxy) * ( * strainxy));
    * strainxy = cc;
    * strainx = 0.5 * aa + 0.5 * bb;
    * strainy = 0.5 * aa - 0.5 * bb;
}

void angle_stress(float * stressx, float * stressy, float * stressxy, float angle)
{
    float aa, bb, cc;
    aa = (* stressx + * stressy) / 2.0;
    bb = cos(2.0 * angle) * (* stressx - * stressy) / 2.0;
    cc = * stressxy * sin(2.0 * angle);
    * stressx = aa + bb + cc;
    * stressy = aa - bb - cc;
}

void angle_strain(float * strainx, float * strainy, float * strainxy, float angle)
{
    float aa, bb, cc;
    aa = (* strainx + * strainy) / 2.0;
    bb = cos(2.0 * angle) * (* strainx - * strainy) / 2.0;

```

```

cc = * strainxy * sin(2.0 * angle)/2.0;
* strainx = aa + bb + cc;
* strainy = aa - bb - cc;
}

void ldlt_solve()
{
    int i, ii, iii, n, a, b, c, d, e;
    float aa;
    n = 2 * _n_node - _n_fixed;
    for(i = 1; i < n; i++)
    {
        a = i - 1 - _address[i] + _address[i - 1] + 1; c = _address[i];
        for(ii = a + 1; ii < i + 1; ii++)
        {
            b = ii - _address[ii] + _address[ii - 1] + 1;
            if(b < a) b = a;
            e = _address[ii];
            for(iii = b; iii < ii; iii++)
            {
                aa = _k[c - i + iii];
                aa = _k[e - ii + iii];
                aa = _k[_address[iii]];
                aa = _k[c - i + iii] * _k[e - ii + iii] / _k[_address[iii]];
                _k[c - i + ii] -= aa;
            }
        }
    }
    _pload[0] = _pload[0] / _k[0];
    for(i = 1; i < n; i++)
    {
        aa = 0.0;
        a = i - _address[i] + _address[i - 1] + 1; b = _address[i]
        for(ii = a; ii < i; ii++)
        {
            c = - (i - ii) + b;
            aa += _pload[ii] * _k[c];
        }
        _pload[i] += aa;
    }
}

```

```

        _pload[i]/ = _k[b];
    }
    c = 1; d = n - 1;
    for(i = n - 2; i > 0; i --)
    {
        c ++; a = _address[i];
        while(c) (_address[d] - _address[d - 1])) { c --, d --; }
        for(ii = 1; ii < c; ii ++ )
        {
            if(ii < _address[ii + i] - _address[ii + i - 1]))
            {
                b = _address[ii + i];
                _pload[i] -= _k[b - ii] * _pload[ii + i] / _k[a];
            }
        }
    }
}

void form _ k()
{
    int i, ii, iii, iiiii, iiiii, a[8], b;
    float B[3][8], c[8][3], D[3][3], ke[8][8], datj;
    _k = (float huge *)fcalloc(_address[2 * _n_node - 1] + 1, sizeof(float));
    if(!_k) { printf(" \ n _ k allocation failure"); exit(1); }
    for(i = 0; i < _n_element; i ++ )
    {
        initial _ farray(64, ke);
        for(ii = 0; ii < 4; ii ++ )
        {
            datj = form _ B(i, ii, B); /* datj = I J * /
            form _ D(i, ii, D);
            initial _ farray(24, c);
            for(iii = 0; iii < 8; iii ++ )
                for(iiiii = 0; iiiii < 3; iiiii ++ )
                    for(iiiii = 0; iiiii < 3; iiiii ++ )
                        c[iii][iiiii] += B[iii][iii] * D[iii][iiiii];
            for(iii = 0; iii < 8; iii ++ )
                for(iiiii = 0; iiiii < 8; iiiii ++ )
                    for(iiiii = 0; iiiii < 3; iiiii ++ )

```

```

        ke[iii][iiii] += datj * c[iii][iiii] * B[iiiii][iiii];
    }
    for(ii = 0; ii < 4; ii++)
    {
        a[2 * ii] = _node_order[_element[i].node[ii] - 1][0];
        a[2 * ii + 1] = _node_order[_element[i].node[ii] - 1][1];
    }
    for(ii = 0; ii < 8; ii++)
    {
        _k[_address[a[ii]]] += ke[ii][ii];
    }
    for(iii = 0; iii < ii; iii++)
    {
        b = a[ii] - a[iii];
        if(b < 0)
            _k[_address[a[iii]] + b] += ke[ii][iii];
        else
            _k[_address[a[ii]] - b] += ke[ii][iii];
    }
}

float form_B(int element, int gi, float B[3][8]) // return datj = | J |
{
    int i, ii, node[4];
    float a, b, c, d, e, f, g, r[4] = {-1.0, 1.0, 1.0, -1.0}, s[4] = {-1.0, -1.0, 1.0, 1.0},
        rg[4] = {-0.5773503, 0.5773503, 0.5773503, -0.5773503},
        sg[4] = {-0.5773503, -0.5773503, 0.5773503, 0.5773503};
    a = 0.0; b = 0.0; c = 0.0; d = 0.0; e = 0.0;
    for(i = 0; i < 4; i++) node[i] = -_element[element].node[i] - 1;
    for(i = 0; i < 4; i++) a += r[i] * _x[node[i]] * (1 + s[i] * sg[gi])/4;
    for(i = 0; i < 4; i++) b += s[i] * _y[node[i]] * (1 + r[i] * rg[gi])/4;
    for(i = 0; i < 4; i++) c += r[i] * _y[node[i]] * (1 + s[i] * sg[gi])/4;
    for(i = 0; i < 4; i++) d += s[i] * _x[node[i]] * (1 + r[i] * rg[gi])/4;
    e = a * b - c * d; // * volume of | j | */
    initial_farray(24, B);
    for(i = 0; i < 4; i++)
    {
        f = r[i] * (1 + s[i] * sg[gi])/4;
        g = s[i] * (1 + r[i] * rg[gi])/4;
        B[0][2 * i] = (f * b - g * c)/e;
    }
}

```

```

        B[2][2 * i + 1] = B[0][2 * i];
        B[1][2 * i + 1] = (- f * d + g * a) / e;
        B[2][2 * i] = B[1][2 * i + 1];
    }
    return(e);
}

void form _ Dcon(int element, int gi, float D[3][3])
{
    int i, ii;
    float c, s, T[3][3], aa[3][3], bb[3][3];
    c = _element[element].Ec1[gi] / (1. - _u * _u);
    s = _element[element].Ec2[gi] / (1. - _u * _u);
    if(c < 0 || s < 0)
        a[0][1] = 0.0;
    else
        aa[0][1] = _u * sqrt(c * s);
    aa[0][0] = c;
    aa[0][2] = 0.0;
    aa[1][0] = aa[0][1];
    aa[1][1] = s;
    aa[1][2] = 0.0;
    aa[2][0] = 0.0;
    aa[2][1] = 0.0;
    aa[2][2] = _element[element].Gc[gi];
    c = cos(_element[element].cangle[gi]);
    s = sin(_element[element].cangle[gi]);
    T[0][0] = c * c;
    T[0][1] = s * s;
    T[0][2] = - 2 * c * s;
    T[1][0] = s * s;
    T[1][1] = c * c;
    T[1][2] = 2 * c * s;
    T[2][0] = c * s;
    T[2][1] = - c * s;
    T[2][2] = c * c - s * s;
    multi_fmatrix(3, 3, 3, T, aa, bb);
    T[0][2] = c * s;
    T[2][0] = - 2 * c * s;
    T[1][2] = - c * s;
    T[2][1] = 2 * c * s;
    multi_fmatrix(3, 3, 3, bb, T, D);
}

void form _ Dste(int element, int gi, float D[3][3])
{
    int i, ii;
    float c, s, T[3][3], aa[3][3], bb[3][3];
    initial_farray(9, aa);
    aa[0][0] = _element[element].Esr[gi];
    c = cos(_element[element].sangle_rx);
    s = sin(_element[element].sangle_rx);
    T[0][0] = c * c;
    T[0][1] = s * s;
    T[0][2] = - 2 * c * s;
}

```



```

T[1][0] = s * s;  T[1][1] = c * c;  T[1][2] = 2 * c * s;
T[2][0] = c * s;  T[2][1] = - c * s;  T[2][2] = c * c - s * s;
multi_fmatrix(3,3,3,T,aa,bb);
T[0][2] = c * s;  T[2][0] = - 2 * c * s;
T[1][2] = - c * s;  T[2][1] = 2 * c * s;
multi_fmatrix(3,3,3,bb,T,D);
aa[0][0] = _element[element].Ess[gi];
c = cos(_element[element].sangle_rx + 3.1415926/2);
s = sin(_element[element].sangle_rx + 3.1415926/2);
T[0][0] = c * c;  T[0][1] = s * s;  T[0][2] = - 2 * c * s;
T[1][0] = s * s;  T[1][1] = c * c;  T[1][2] = 2 * c * s;
T[2][0] = c * s;  T[2][1] = - c * s;  T[2][2] = c * c - s * s;
multi_fmatrix(3,3,3,T,aa,bb);
T[0][2] = c * s;  T[2][0] = - 2 * c * s;
T[1][2] = - c * s;  T[2][1] = 2 * c * s;
multi_fmatrix(3,3,3,bb,T,aa);
for(i = 0; i < 3; i++)
for(ii = 0; ii < 3; ii++)
    D[ii][ii] += aa[i][ii];
}

void address_k()
{
    in i, ii, iii, n, b, a[8];
    for(i = 0; i < _n_fixed; i++)
    {
        b = _discr_fixed[i][0] - 1;
        _node_order[b][0] = - _dirscr_fixed[i][1];
        _node_order[b][1] = - _discr_fixed[i][2];
    }
    b = 0;
    for(i = 0; i < _n_fixed; i++)
        b += _discr_fixed[i][1] + _discr_fixed[i][2];
    _n_fixed = b;
    b = 2 * _n_node - _n_fixed;
    n = 0;
    for(i = 0; i < _n_node; i++)
    {
        if(_node_order[i][0] < 0)
            _node_order[i][0] = b++;
        else

```

```

        _node_order[i][0] = n++;
    if(_node_order[i][1]<0)
        _node_order[i][1] = b++;
    else
        _node_order[i][1] = n++;
}
for(i = 0; i<2*_n_node; i++)    _address[i] = 0;
for(i = 0; i<_n_element; i++)
{
    for(ii = 0; ii<4; ii++)
    {
        a[2*ii] = _node_order[_element][i].node[ii] - 1[0];
        a[2*ii + 1] = _node_order[_element][i].node[ii] - 1[1];
    }
    while(a[0]>a[1] || a[1]>a[2] || a[2] || a[2]>a[3] || a[3]>a[4]>a[5]
    || a[5]>a[6] || a[6]>a[7])
    {
        for(iii = 0; iii<7; iii++)
            while(a[iii]>a[iii + 1]){b = a[iii]; a[iii] = a[iii + 1]; a[iii + 1] =
            b;}
    }
    if(a[7]>2*_n_node - 1)
    {
        printf("/n node number error ! /n");
        exit(1);
    }
    for(ii = 1; ii<8; ii++)
    {
        b = a[ii] - a[0] + 1;
        if(_address[a[ii]]<b)
            _address[a[ii]] = b;
    }
}
for(i = 1; i<2*_n_node; i++)
{
    _address[i] += _address[i - 1];
}
}

void reuse_cstate(int total_step)
{
    char rejust = 1, n_loop;

```

```

int i,ii,iii,n_rejust;
float aa,bb,cc,cstress1,cstress2,Gc;
for(i = 0;i<2*_n_node;i++)_pload_bak[i] = _pload[i];
n_loop = 0;
while(rejust)
{
    rejust = 0;
    n_loop++;
    form_k();
    ldlt_solve();
    farfree(_k);
    stress_strain(total_step);
    if(total_step>0)
    {
        n_rejust = 0;
        for(i = 0;i<_n_element;i++)
        for(ii = 0;ii<4;ii++)
        {
            aa = _cstress[i][3*ii];
            bb = _cstress[i][3*ii+1];
            cc = _cstress[i][3*ii+2];
            if(_element[i].lcstrain1[ii]>_cstrain_t
            &&_element[i].lcstrain2[ii]>_cstrain_t)
            {
                main_stress(&aa,&bb,&cc);
                if(_element[i].baseangle[ii]-cc>pi/4&&
                _element[i].baseangle[ii]-cc<3*pi/4)
                {
                    cstress1 = bb;
                    cstress2 = aa;
                }
                else if(cc-_element[i].baseangle[ii]>pi/4&&
                cc-_element[i].baseangle[ii]<3*pi/4)
                {
                    cstress1 = bb;
                    cstress2 = aa;
                }
                else
                {
                    cstress1 = aa;
                    cstress2 = bb;
                }
            }
        }
    }
    else
    {
        angle_stress(&aa,&bb,&cc,_element[i].cangle[ii]);
    }
}

```

```

        cstress1 = aa;
        cstress2 = bb;
    }
    if(( _ element[i].Ec1[ii]>0.0001 * _ Ec0) || ( _ element[i].
    Ec1[ii]<- 0.0001 * _ Ec0))
    {
        aa = (cstress1 - _ element[i].mainstress1[ii])/ _ element[i].
        Ec1[ii];
    }
    else
    {
        aa = _ inc _ strain[i][3 * ii];
        bb = _ inc _ strain[i][3 * ii + 1];
        cc = _ inc _ strain[i][3 * ii + 2];
        angle _ strain(&aa,&bb,&bb,_ element[i].cangle[ii]);
    }

    if(aa<- 1e - 7 || aa>1e - 7)&&
    aa/_ element[i].cstate1[ii]<0.0]
    {
        _ element[i].cstate1[ii] * = - 1;
        rejust = 1;
        n _ rejust ++;
    }

    if(( _ element[i].Ec2[ii]>0.0001 * _ Ec0) || ( _ element[i].
    Ec2[ii]<- 0.0001 * _ Ec0))
    {
        bb = (cstress2 - _ element[i].mainstress2[ii])/ _ element[i].
        Ec2[ii];
    }
    else
    {
        aa = _ inc _ strain[i][3 * ii];
        bb = _ inc _ strain[i][3 * ii + 1];
        cc = _ inc _ strain[i][3 * ii + 2];
        angle _ strain(&aa,&bb,&cc,_ element[i].cangle[ii]);
    }

    if((bb<- 1e - 7 || bb > 1e - 7)&&bb/_ element[i].cstate2[ii]<0.0)
    {
        _ element[i].cstate2[ii] * = - 1;
        rejust = 1;
        n _ rejust ++;
    }
}

```

```

}
for(i = 0; i < _n_element; i++)
for(ii = 0; ii < 4; ii++)
{
    if(_element[i].cstate1[ii] > 0)
        _element[i].Ec1[ii] = _element[i].iEc1[ii];
    else
        _element[i].Ec1[ii] = _element[i].dEc1[ii];
    if(_element[i].cstate2[ii] > 0)
        _element[i].Ec2[ii] = _element[i].iEc2[ii];
    else
        _element[i].Ec2[ii] = _element[i].dEc2[ii];
    if(_element[i].sstater[ii] > 0)
        _element[i].Esr[ii] = _element[i].iEsr[ii];
    else
        _element[i].Esr[ii] = _element[i].dEsr[ii];
    if(_element[i].sstates[ii] > 0)
        _element[i].Ess[ii] = _element[i].iEss[ii];
    else
        _element[i].Ess[ii] = _element[i].dEss[ii];
    if((_element[i].crack1[ii] > 0 && _element[i].crack2[ii] > 0) ||
    {
        aa = _element[i].Ec1[ii];
        bb = _element[i].Ec2[ii];
        cc = aa * bb;
        if(cc < 0.)
            cc = 0.0;
        else
            cc = sqrt(cc);
        _element[i].Gc[ii] = (aa + bb - 2. * _u * cc) / 4.0;
    }
}
printf("n_rejust = %d \ n", n_rejust);
if(n_loop > 5) rejust = 0;
if(rejust)
{
    for(iii = 0; iii < _n_element; iii++)
        farfree(_cstress[iii]);
}

```

```

        farfree(_cstress);
        for(iii = 0; iii < _n_element; iii++)
            farfree(_sstrs[iii]);
        farfree(_sstrs);
        for(iii = 0; iii < _n_element; iii++)
            farfree(_strain[iii]);
        farfree(_strain);
        for(i = 0; i < 2 * _n_node; i++) _pload[i] = _pload_bak[i];
    }
}

rewind(opf1);
for(i = 0; i < _n_element; i++)
    fwrite(_cstress[i], 4, 12, opf1);
rewind(opf2);
for(i = 0; i < _n_element; i++)
    fwrite(_sstrs[i], 4, 12, opf2);
rewind(opf3);
for(i = 0; i < _n_element; i++)
    fwrite(_strain[i], 4, 12, opf3);

```

附录 B 钢筋混凝土构件裂缝及变形图绘制

钢筋混凝土构件的开裂、压碎及变形过程,是对构件进行理论分析的主要依据,通过它们可对构件的机理和性能作出相应的推论,对计算分析来说,通过对钢筋混凝土构件的开裂、压碎及变形的描绘,一方面可将计算结果与试验相对照,来验证计算模型的正确性;另一方面,作为计算分析的后处理过程,将分析结果直观地显示出来,以利于提高工作效率。这些反映钢筋混凝土构件的开裂、压碎及变形过程的图形还可用来模拟试验过程,使某些试验由计算分析来替代。

以往钢筋混凝土构件的开裂、压碎及变形的绘制,通常采用调用高级算法语言的图形函数,将图形直接在屏幕上显示。但是,这种图形绘制方式由于受显示器的大小所限制,而且,它只是实际数据图形的相对图形或示意图形,精度较低,书面输出非常粗糙。为了避免这种图形制作的缺陷,可借助绘图软件来进行绘图,最常用的微机平台绘图软件当数 AutoCAD。运用 AutoCAD 绘图软件制作钢筋混凝土开裂、压碎及变形过程这类图形基本上可采用两种方法:一种是,采用高级算法语言将计算分析结果进行处理,将得到的图形信息按 AutoCAD 图形接口文件格式写入一个文本文件,然后在 AutoCAD 环境下读入该文件,生成所需图形;另一种方法是,通过 AutoCAD 的编程语言,如 ADS, AutoLISP 等,在 AutoCAD 环境下将计算结果数据读入,边分析处理,边将处理结果绘制成图。采用后一种方法绘制钢筋混凝土构件开裂图时,可形象地看到裂缝的开展过程。由 AutoCAD 编程语言作图涉及内容较多,读者有兴趣的话,可参阅 AutoCAD 的有关手册。这里仅以图形接口文件为例,说明钢筋混凝土构件裂缝的绘制方法。

AutoCAD 标准图形交换文件 DXF(Drawing Exchange File)是 AutoCAD 同外部进行数据交换和数据传输的文本文件,也可以看成是 AutoCAD 内部图形数据库的 ASCII 码映象,是一幅 AutoCAD 图形的完全等价形式。它可以从 AutoCAD 的图形编辑系统中用 DXFOUT 命令从已有图形中转换而来,也可用 DXFIN 命令调入 AutoCAD,生成相应的图形。由于它是文本代码形式的文件,所以,它的可读性较好,可以用高级算法语言程序构造一个 DXF 格式的文件,然后调入 AutoCAD,生成图形。

DXF 文件的总体结构由四个大的段(Section)组成,即标题段(HEADER SECTION)、表段(TABLES SECTION)、块段(BLOCKS SECTION)和实体段(ENTITIES SECTION),DXF 文件的总体结构如下:

```
0
SECTION                                标题段开始
2
HEADER
.....
0
```

ENDSEC	标题段结束
0	
SECTION	
2	表段开始
TABLES	
.....	
0	
ENDSEC	表段结束
0	
SECTION	
2	块段开始
BLOCKS	
.....	
0	
ENDSEC	块段结束
0	
SECTION	
2	实体段开始
ENTITIES	
.....	
0	
ENDSEC	实体段结束
0	
EOF	DXF 文件结束

标题段(HEADER)在 DXF 文件的头部,包含了图形的总的信息,在构造 DXF 文件时可将其省略。

表段(TABLES)包含了有名图项的定义:

线型表(LINETYPE TABLE)

层表 (LAYER TABLE)

格式表(STYLE TABLE)

视图表(VIEW TABLE)

在不涉及线型、图层等的情况下,表段也可省略,缺省状态下,图形实体将放在“0”层上。

块段(BLOCKS),本段包含了图块定义,这些定义描述了图形中组成各个图块的实体。构造 DXF 文件时,一般不涉及制作图块,因此,这一段也可省略掉。

实体段(ENTITIES),本段包含了图形实体及图块的调用,每一个图素,如线、点、圆等均在一 段内。

以下是一个仅有实体段的 DXF 文件实例,它包含了一条直线和一个圆的信息。

0

SECTION	
2	实体段开始
ENTITIES	
0	0 组,标识实体类型开始
LINE	直线实体
8	8 组,后跟实体所在层名
0	实体在“0”层
10	10 组,标识直线起点 X 坐标
1.0	X 坐标值
20	20 组,标识直线起点 Y 坐标
1.0	Y 坐标值
30	30 组,标识直线起点 Z 坐标
0.0	Z 坐标值
11	11 组,标识直线终点 X 坐标
2.0	X 坐标值
21	21 组,标识直线终点 Y 坐标
2.0	Y 坐标值
31	31 组,标识直线终点 Z 坐标
0.0	Z 坐标值
0	实体类型开始
CIRCLE	圆实体
8	
0	
10	10 组,标识 X 坐标
0.0	圆心 X 坐标值
20	20 组,标识 Y 坐标
0.0	圆心 Y 坐标值
30	30 组,标识 Z 坐标
0.0	圆心 Z 坐标值
40	40 组,标识半径
2.0	半径值
0	
ENDSEC	实体段结束
0	
EOF	DXF 文件结束

在 AutoCAD 环境下用 DXFIN 命令将以上文本文件调入,会生成一条直线和一个圆。

参照上述 DXF 文件的构造方法,就可以进行混凝土构件开裂、压碎及变形图的绘制。混凝土单元的开裂可用直线段表示,压碎则可用填充块(AutoCAD 中的 SOLID)来表示,表

示的位置一般在单元的各高斯积分点上。下列为构造 DXF 文件的源程序,所生成 DXF 文件用以绘制钢筋混凝土构件的开裂图。

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <math.h>
#include <alloc.h>
#define pi 3.1415926

FILE * ipf, * opf,
char _sec[] = "SECTION \n", _ent[] = "ENTITIES \n", _endsec[] = "ENDSEC \n",
    _eof[] = "EOF \n", _line[] = "LINE \n";
int _n_element, _n_node;          //单元总数, 结点总数
struct element _describe
{
    int node[4];                  //各单元四个结点号
    char crack1[4], crack2[4];    //各单元高斯积分点处开裂信息, 1—开裂, 2—
                                   //未开裂
    float cangle[4];              //开裂方向
};
struct element _describe * _element;
float _x[500], _y[500], _length[500];

void draw_crack();
void position(int element, int gi, float * x, float * y); //确定高斯坐标点函数
void draw(float x, float y, float length, float angle); //画直线段函数
void draw_element();
void main()
{
    _element = (struct element _describe *) calloc(_n_element, sizeof(struct element _describe));
    if(!_element){printf("_element allocation failure \n");exit(1);}

    char name[40];
    int i, ii;
    float aa, bb;

    printf(" \nInput FE describe data file");
```

```

scanf("%s",name);

if((ipf=fopen(name,"r"))==NULL)
{ printf("Cannot %s open file ! \n",name);
  exit(1);
}

fscanf(ipf,"%d",&_n_element);
fscanf(ipf,"%d",&_n_node);

for(i=0;i<_n_element;i++)
for(ii=0;ii<4;ii++)
    fscanf(ipf,"%d",&_element[i].node[ii]); //读入单元的4个结点编号
for(i=0;i<_n_node;i++)fscanf(ipf,"%f",&_x[i]); //读入结点X坐标
for(i=0;i<_n_node;i++)fscanf(ipf,"%f",&_y[i]); //读入结点Y坐标
fclose(ipf);

printf("\nInput cracking information data file");
scanf("%s",name);
if((ipf=fopen(name,"rb"))==NULL)
{ printf("Cannot %s open file ! \n",name);
  exit(1);
}

for(i=0;i<_n_element;i++)
{ fread(_element+i,sizeof(struct element_describe),1,ipf);
}

fclose(ipf);

for(i=0;i<_n_element;i++) //确定各单元所绘开裂线段长度
{ aa=_x[_element[i].node[0]-1];
  _length[i]=0;
  for(ii=1;ii<4;ii++)
  { bb=fabs(aa-_x[_element[i].node[ii]-1]);
    if(bb>_length[i])_length[i]=bb;
  }
}

for(i=0;i<_n_element;i++) _length[4]/=4;

```

```

printf("input DXF file name:");    //新建 DXF 文件
scanf("%s",name);
if((opf = fopen(name,"w")) == NULL)
{
    printf("Cannot %s open file! \n",name);
    exit(1);
}

fprintf(opf,"%3d\n",0);
fprintf(opf,"%s",_sec);
fprintf(opf,"%3d\n",2);
fprintf(opf,"%s",_ent);    //实体段标识

draw_element();            //画各个单元,也可只画构件轮廓图,将此函数关闭
draw_crack();              //画各个单元中的裂缝

fprintf(opf,"%3d\n",0);
fprintf(opf,"%s",_endsec);
fprintf(opf,"%3d\n",0);
fprintf(opf,"%s",_eof);    //DXF 文件结束
fclose(opf);
}

void draw_crack()
{
    int i,ii;
    float x,y;

    for(i=0;i<_n_element;i++)
    for(ii=0;ii<4;ii++)
    {
        if(_element[i].crack1[ii]>0)
        {
            position(i,ii,&x,&y);    //确定所绘裂缝坐标(高斯积分点)X,Y
            draw(x,y,_lenth[i],_element[i].canglr[ii]+pi/2); //按 X,Y 坐标、长度、开裂
                                                                    //角度,画裂缝线段
        }
        if(_element[i].crack2[ii]>0)    //画垂直与第一开裂方向的第二条裂缝
        {
            position(i,ii,&x,&y);
            draw(x,y,_length[i],_element[i].cangle[ii]);
        }
    }
}
}

```

```

void draw(float x, float y, float length, float angle)
{
    float aa, bb, x1, x2, y1, y2;

    an = length * cos(angle)/2;
    bb = length * sin(angle)/2;
    x1 = x - aa; x2 = x + aa;
    y1 = y - bb; y2 = y + bb;

    fprintf(opf, "%s", _line);
    fprintf(ope, "%3d \n", 8);
    fprintf(opf, "%d \n", 0);
    fprintf(opf, "%3d \n", 10);
    fprintf(opf, "%f \n", x1);
    fprintf(opf, "%3d \n", 20);
    fprintf(opf, "%f \n", y1);
    fprintf(opf, "%3d \n", 30);
    fprintf(opf, "%f \n", 0.0);
    fprintf(opf, "%3d \n", 11);
    fprintf(opf, "%f \n", x2);
    fprintf(opf, "%3d \n", 21);
    fprintf(opf, "%f \n", y2);
    fprintf(opf, "%3d \n", 31);
    fprintf(opf, "%f \n", 0.0);
    fprintf(opf, "%3d \n", 0);
}

void position(int element, int gi, float * x, float * y)
{
    int i;
    float n[4], rg[4] = { -0.5773503, 0.5773503, 0.5773503, -0.5773503 },
           sg[4] = { -0.5773503, -0.5773503, 0.5773503, 0.5773503 };
    n[0] = (1 - rg[gi]) * (1 - sg[gi])/4;
    n[1] = (1 + rg[gi]) * (1 - sg[gi])/4;
    n[2] = (1 - rg[gi]) * (1 + sg[gi])/4;
    n[3] = (1 + rg[gi]) * (1 + sg[gi])/4;
    *x = 0; *y = 0;
    for(i = 0; i < 4; i++)
    {
        *x += n[i] * _x[_element[element].node[i] - 1];
        *y += n[i] * _y[_element[element].node[i] - 1];
    }
}

```

```

    |
}

void draw_element()
{
    int i, ii;

    for(i = 0; i < _n_element; i++)
    {
        for(ii = 0; ii < 3; ii++)
        {
            fprintf(opf, "%s", _line);
            fprintf(opf, "%3d \n", 8);
            fprintf(opf, "%d \n", 0);
            fprintf(opf, "%3d \n", 10);
            fprintf(opf, "%f \n", _x[_element[i].node[ii] - 1]);
            fprintf(opf, "%3d \n", 20);
            fprintf(opf, "%f \n", _y[_element[i].node[ii] - 1]);
            fprintf(opf, "%3d \n", 30);
            fprintf(opf, "%f \n", 0.0);
            fprintf(opf, "%3d \n", 11);
            fprintf(opf, "%f \n", _x[_element[i].node[ii + 1] - 1]);
            fprintf(opf, "%3d \n", 21);
            fprintf(opf, "%f \n", _y[_element[i].node[ii + 1] - 1]);
            fprintf(opf, "%3d \n", 31);
            fprintf(opf, "%f \n", 0.0);
            fprintf(opf, "%3d \n", 0);
        }

        fprintf(opf, "%s", _line);
        fprintf(opf, "%3d \n", 8);
        fprintf(opf, "%d \n", 0);
        fprintf(opf, "%3d \n", 10);
        fprintf(opf, "%f \n", _x[_element[i].node[3] - 1]);
        fprintf(opf, "%3d \n", 20);
        fprintf(opf, "%f \n", _y[_element[i].node[3] - 1]);
        fprintf(opf, "%3d \n", 30);
        fprintf(opf, "%f \n", 0.0);
        fprintf(opf, "%3d \n", 11);
        fprintf(opf, "%f \n", _x[_element[i].node[0] - 1]);
        fprintf(opf, "%3d \n", 21);
        fprintf(opf, "%f \n", _y[_element[i].node[0] - 1]);
    }
}

```

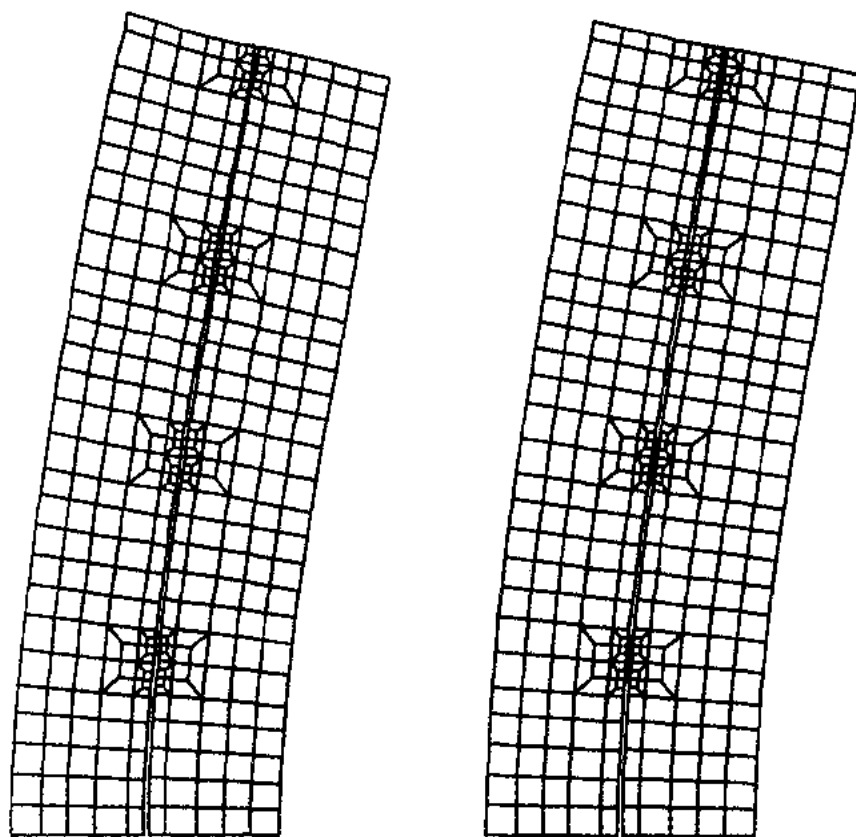
```

fprintf(opf,"%3d \n",31);
fprintf(opf,"%f \n",0.0);
fprintf(opf,"%3d \n",0);
}
}

```

用以上程序生成的 DXF 文件,经 AutoCAD 绘制成的剪力墙裂缝分布图如图 7.12、图 7.15 所示。

按照同样的 DXF 文件构造方法,可以容易地编写出用于生成描绘构件变形的 DXF 文件的源程序,附图 B1 为按计算结果所描绘的带缝剪力墙的变形图,其中图(a)为弱连系梁情况下的变形形态,图(b)为强连系梁情况下的变形形态。



(a) 弱连系梁墙体受力变形情况

(b) 强连系梁墙体受力变形情况

附图 B.1 剪力墙有限元计算模型变形图

参 考 文 献

- [1] Scordelis, A. C.. Past, Present and Future Development, Finite Element Analysis of Reinforced Concrete Structures, ASCE, 1986
- [2] Inoue, N. and Noguchi, H., Finite Element Analysis of Reinforced Concrete in Japan, Finite Element Analysis of Reinforced Concrete Structures, ASCE, 1986
- [3] 董振祥, 颜德姮. 用有限元方法研究钢筋混凝土梁的非线性性能. 同济大学学报, 1981(4)
- [4] 朱伯龙, 董振祥. 钢筋混凝土非线性分析. 上海: 同济大学出版社, 1985
- [5] 宋启根, 单炳祥, 金芷生等. 钢筋混凝土力学. 南京: 南京工学院出版社, 1986
- [6] 江见鲸. 钢筋混凝土结构非线性有限元分析. 西安: 陕西科学技术出版社, 1994
- [7] 吕西林, 吴晓涵. Dynamic Analysis of R. C. Slitted Shear Walls with Nonlinear F. E. Method, Proceedings of the Third Asia-Pacific Conference on Computational Mechanics, Sept. 1996, Seoul, Korea.
- [8] 吴晓涵, 吕西林. A Constitutive Model for Nonlinear F. E. Analysis of R. C. Shear Walls Under Cyclic Loading, Proceedings of the Third Asia-Pacific Conference on Computational Mechanics, Sept. 1996, Seoul, Korea.
- [9] Schnobrich, W. C., Behavior of Reinforced Concrete Members by Finite Element, Keynote Lecture, Proceedings of The Third Asia-Pacific Conference on Computational Mechanics, Vol. 1, Sept. 1996, Seoul, Korea.
- [10] 沈聚敏, 王传志, 江见鲸. 钢筋混凝土有限元与板壳极限分析. 北京: 清华大学出版社, 1993
- [11] 孙菊芳主编. 有限元法及其应用. 北京: 北京航空航天大学出版社, 1990
- [12] 孙钧, 汪炳鉴. 地下结构有限元法解析. 上海: 同济大学出版社, 1988
- [13] 张德兴. 有限元素法新编教程. 上海: 同济大学出版社, 1989
- [14] 滕智明, 邹离湘. RC有限元分析中的粘结单元. 见: 第四届全国结构工程学术会议论文集(上). 工程力学(增刊), 1995
- [15] 朱伯龙主编. 混凝土结构设计原理(上册). 上海: 同济大学出版社, 1992
- [16] 孙钧, 侯学渊. 地下结构. 北京: 科学出版社, 1988
- [17] D. R. J. Owen and E. Hinton, Finite Elements in Plasticity, Theory and Practice, Pineridge Press Limited, 1980
- [18] R. W. G. Blakeley, R. Park, Prestressed Concrete Sections with Cyclic Flexure, Jour. of ASCE, ST8, 1973
- [19] D. Darwin and D. A. Pecknold, Nonlinear Biaxial Stress Strain Law for Concrete, ASCE, Engineering Mechanics Division, ASCE 103(2), 1977

- [20] W. F. Chen, Plasticity in Reinforced Concrete, 1982.
- [21] N. M. Hawkins, etc., Local Bond Strength of Concrete for Cyclic Reversed Loadings"《Bond in Concrete》1982.
- [22] 混凝土结构基本理论及工程应用. 见:第三届全国建筑结构学术讨论会论文集. 长沙:湖南科学技术出版社,1993
- [23] Z.P. Bazant 等著,周氏等译. 钢筋混凝土有限元分析. 南京:河海大学出版社,1988
- [24] 王传志,滕智明主编. 钢筋混凝土结构理论. 北京:中国建筑工业出版社,1985
- [25] 刘南科,周基岳,肖允微等. 钢筋混凝土框架的非线性全过程分析. 土木工程学报, 1990,23(4)
- [26] 周基岳,刘南科. 钢筋混凝土框架非线性分析中的截面弯矩-曲率关系. 重庆建筑工程学院学报,1984(2)
- [27] 方鄂华. 高层建筑结构设计. 北京:地震出版社,1990
- [28] M. Fintel, Shearwalls—An Answer for Seismic Resistance? Point of View, 30 Years of Observations on the Performance of Buildings with Shearwalls in Earthquake, Concrete International Design & Construction, 13(17), 1991
- [29] 武藤清,结构物动力设计. 北京:中国建筑工业出版社,1984
- [30] K.Muto, Earthquake Resistant Design of 36-storied Kasumigaseki Building. Proc. 4th World Conf. on Earthquake Engng. I4, Santiago de Chile, 1969.
- [31] 吕西林, Y. K. Cheung, A. K. H. Kwan. 高层剪力墙结构抗震控制研究. 同济大学学报, 1994(24)
- [32] Saenz L. P., Disc. of Equation for the Stress-strain Curve of Concrete by Desayi and Krishnan, Journal of ACI, 1964, 61(9).
- [33] N. Shirai and T. Sato, Inelastic Analysis of Reinforced Concrete Shear Wall Structures-Material Modelling of Reinforced Concrete, IABSE Colloquium, Delft, 1981.
- [34] V. Cervenka, K. H. Gerstle, Inelastic Analysis of Reinforced Concrete Panels: Theory, Proc. of International Association for Bridge and Structural Engineers, 31(II), 1971.
- [35] R.S.H. AL-Mahaidi, Nonlinear Finite Element Analysis of Reinforced Concrete Subjected to Shear Loading. Heron, 26(1a), 1981.
- [36] J. Houde, M.S. Mirza, A Finite Element Analysis of Shear Strength of Reinforced Concrete Beams, Shear in Reinforced Concrete, 1, SP-42, ACI, Detroit, Michigan, 1974.
- [37] Y.K. Cheung, A.K.H. Kwan, and X.L. Lu, Large Scale Model Tests of Reinforced Concrete Slit Shear Walls, International Journal of Structures, 14(2), July-Dec., 1994.
- [38] 吴晓涵,吕西林. 反复荷载下混凝土剪力墙非线性有限元分析. 同济大学学报, 1996,26(1)
- [39] 邱玉春,郑卓嘉,段富编. AutoCAD < 10-11 > 使用手册. 北京:电子工业部出版社, 1993
- [40] 庄表中,梁以德,张佑启. 结构随机振动. 北京:国防工业出版社,1995
- [41] Sabris, G. M, etc, Structural Modeling and Experimental Techniques, Printice hall Inc.,

Englewood Cliffs, N. J. 07632, 1983.

- [42] 李杰, 李国强. 地震工程学导论. 北京: 地震出版社, 1992
- [43] Clough, R. W. and Penzien, J., Dynamics of Structures (2nd Edition) McGraw-Hill, Inc, 1993.
- [44] 王文亮, 杜作润. 结构振动与动态子结构方法. 上海: 复旦大学出版社, 1985
- [45] 吕西林, 周德源, 李思明. 房屋结构抗震设计理论与实例. 上海: 同济大学出版社, 1995
- [46] 李国豪主编. 工程结构抗震动力学. 上海: 上海科学技术出版社, 1980
- [47] 程民宪. 结构动力分析中几种逐步积分法在负刚度条件下的收敛性和稳定性. 工程力学, 1989(2)
- [48] 赵经文, 王宏钰. 结构有限元分析. 哈尔滨: 哈尔滨工业大学出版社, 1988
- [49] 孙业杨, 余安东, 金瑞春等. 高层建筑杆系-层间模型弹塑性动态分析. 同济大学学报, 1980(1)
- [50] 吕西林, 吴晓涵. 抗震耗能剪力墙非线性有限元分析. 同济大学学报, 1996, 26(5)